Combining Features at Search Time: PRISMA at Video Copy Detection Task

Juan Manuel Barrios
PRISMA Research Group
Department of Computer Science, University of Chile
jbarrios@dcc.uchile.cl

Benjamin Bustos
PRISMA Research Group
Department of Computer Science, University of Chile
bebustos@dcc.uchile.cl

Xavier Anguera[†]
Telefonica Research
Torre Telefonica Diagonal 00
08019 Barcelona, Spain
xanguera@tid.es

ABSTRACT

Most of current Video Copy Detection systems (VCD) perform a multimodal detection by dividing the system into subsystems. Each subsystem performs a copy detection using a different feature (either visual or audio), and the sets of candidates are combined (fused) to create the final result.

We present a VCD system that fuses visual and audio descriptors at the similarity search level. The system produces the copy candidates by comparing video segments using visual and audio descriptors instead of fusing copy candidates from independent subsystems.

We submitted four Runs to TRECVID 2011 CCD task:

- `PRISMA.m.balanced.EhdGry`: a combination of two visual global descriptors. Two detection candidates per query.
- `PRISMA.m.balanced.EhdRgbAud`: a combination of two visual global descriptors and one audio descriptor. Two detection candidates per query.
- `PRISMA.m.nofa.EhdGry`: a combination of two visual global descriptors. One detection candidate per query.
- `PRISMA.m.nofa.EhdRgbAud`: a combination of two visual global descriptors and one audio descriptor. One detection candidate per query.

Our Runs achieve good detection effectiveness, especially for NoFA profile, and they are among the fastest Runs. To the best of our knowledge, this is the first VCD system that successfully fuses audio and visual descriptors at an earlier stage than decision level.

Additionally, we have performed a joint submission with Telefonica Research team, under the name `Telefonica-research.m.balanced.joint`, which tests the combination at the decision level of Telefonica's local descriptor, audio descriptor, and PRISMA's `EhdRgb` global descriptors.

1. INTRODUCTION

Most of current Video Copy Detection systems (VCD) perform a multimodal detection by dividing the system into subsystems. Each subsystem performs a copy detection using a different feature (either visual or audio), and the sets of candidates are combined (fused) to create the final result.

For instance, let A and B be two VCD subsystems: A uses visual descriptors and B uses audio descriptors. A and B perform a copy detection process, each one producing a set of copy candidates C^A and C^B, respectively. Then, a fusion step takes both sets C^A and C^B and combines them to create the final detection list C. This late fusion approach creates C by: a) joining or intersecting C^A and C^B; b) calculating a detection score for each candidate in C according to the detection scores from each subsystem; and c) fixing the copy excerpt limits (start/end times) using the candidates' time limits. Some systems with this fusion approach are the TRECVID-CCD teams: INRIA-TEXMEX 2010, Telefonica 2010, IBM 2010 (fuses three visual descriptors), NTT-SCL 2010, KDDI 2010, NII 2009, THU-IMG 2009, and others.

We present a VCD system that fuses visual and audio descriptors at the similarity search level: i.e. the system generates the copy candidates as a result of searching similar video segments according to both visual and audio descriptors at the same time, instead of fusing copy candidates from independent subsystems.

2. P-VCD SYSTEM 2011

P-VCD is our system developed for TRECVID 2010. P-VCD system divides the copy detection process in five tasks: Preprocessing, Video Segmentation, Feature Extraction, Similarity Search, and Copy Localization.

P-VCD 2010 is deeply detailed and analyzed in [1]. This year we have introduced some low-level changes that improved the detection performance and decreased the processing time (by optimizing the code and adjusting parameters). However, the most remarkable improvement has been the introduction of audio features and their novel combination with visual descriptors (P-VCD 2010 only worked with visual descriptors). We have used the audio descriptor that Telefonica Research used for TRECVID 2009, which is in turn based on the descriptor in [2]. We have adapted that audio descriptor to fit the metric approach, and we have divided the Similarity Search task into a multi-step search to efficiently resolve audio+video searches.

CCD evaluation considered a reference video collection of 11,485 video files with a total extension of 406 hours. The query video collection contained 1,608 visual queries and 1,407 audio queries, which combined produced 11,256 audio+visual queries with a total extension of 226 hours.

2.1 Preprocessing

This task has two objectives: to normalize the quality of query and reference videos, and to diminish the effect of transformations on query videos by detecting and reverting

[†]Reference implementation for audio descriptor.

picture-in-picture (PIP) and camcording. This task is similar to our participation in TRECVID 2010 except for some implementation changes that decreased the processing time.

The normalization process: 1) removes frames whose intensities show low variance, 2) detects and removes letter-, pillar-, or window-boxing, and 3) removes outlier frames. We stated that a frame is an outlier when both differences between current frame and previous frame and current frame and next frame are high, and the difference between previous frame and next frame is small.

A PIP detection is performed on every query video by detecting a persistent rectangle. When a PIP is detected two new queries are created: the foreground video (each frame cropped to the detected rectangle) and the background video (each frame with the detected rectangle filled with black pixels). A camcording detection is performed on query videos by detecting a wrapping quadrilateral. When camcording is detected a new query video is created by mapping the detected quadrilateral to the video corners.

New queries are treated as independent queries up to the copy localization task, where all the detections are combined. As a result of this task, the number of visual query videos processed by next tasks increased from 1,608 to 5,147, thus the number of a+v queries increased from 11,256 to 36,029.

The audio track for all the reference and audio queries were resampled to 8KHz-mono using FFmpeg tools.

2.2 Video Segmentation

Every query and reference video has been partitioned into segments of 333 ms length. Last year we tested variable-length segments depending on visual similarity. However, this year we preferred a fixed-length segmentation in order to simplify the fusion of audio with visual descriptors.

The 11,485 reference videos produced 4,522,262 visual segments and 4,441,717 audio segments (some videos have different lengths for audio and visual tracks). The 5,147 visual queries produced 1,120,455 visual segments, and the 1,407 audio queries produced 306,304 audio segments.

2.3 Feature Extraction

For each frame in a visual segment we calculated three global descriptors: Edge Histogram (**Ehd**), Gray Histogram (**Gry**), and Color Histogram (**Rgb**); and for each audio segment we calculated one audio descriptor (**Aud**):

- **Ehd** divides a frame into 4×4 blocks, for each block measures the distribution of 10 orientations of edges (using 2×2 masks from [3]). This produced a 160-d vector, quant. 1 byte/dim.
- **Gry** divides a frame into 4×4 blocks, for each block calculates a 12-bins histogram of intensities. This produced a 192-d vector, quant. 1 byte/dim.
- **Rgb** divides a frame into 4×4 blocks, for each block calculates a 4-bins histogram for each of the Red, Green and Blue channels. This produced a 192-d vector, quant. 1 byte/dim.
- **Aud** is based on Telefonica research implementation of the descriptor in [2]. Originally, the descriptor is calculated with a FFT of the acoustic data every 10 ms over a sliding window of 32 ms, then the frequency bins are converted into a Mel scale of 16 bands, and a 15-bit fingerprint is calculated by comparing the energies of consecutive bands (1=increase, 0=decrease). However, for this work, we extracted a Mel scale of 160 bands, normalized the sum of energies to 1, and then we averaged the normalized energies for all the windows inside the audio segment. This produced a 160-d vector, quant. 4 bytes/dim (we did not test 1 byte/dim).

The descriptor for a visual segment is the average descriptor for all of its frames. Note that the preprocessing task should have removed most of the noisy frames.

The adapted audio descriptor enabled us to measure the degree of similarity between any two short audio signals (instead of just searching collisions), which is a requirement for applying the metric approach.

The following table shows the space required by these four descriptors:

	reference	**query**
Ehd	691 MB	212 MB
Gry	829 MB	206 MB
Rgb	829 MB	206 MB
Aud	2,7 GB	187 MB

2.4 Similarity Search

For measuring the degree of similarity between any two segments we combined the distance between their descriptors. We used Manhattan distance to compare any of the four descriptors:

$$L_1(\vec{x}, \vec{y}) = \sum_{i=0}^{dim} |x_i - y_i|$$

2.4.1 Visual-Based Search

Let $\mathcal{Q}_v$ be the set of visual segments for query videos, and $\mathcal{R}_v$ be the set of visual segments for reference videos ($|\mathcal{Q}_v|$=1,120,455 and $|\mathcal{R}_v|$=4,522,262). The visual-based search aims for retrieving for each object in $\mathcal{Q}_v$ the k most similar objects in $\mathcal{R}_v$ according to a distance $d_v(q, r)$.

Visual Distance For any two visual segments q and r, their similarity was measured using the distance function:

$$\begin{aligned} d_v(q,r) &= \frac{w_1}{\tau_1} * L_1(\text{Ehd}(q), \text{Ehd}(r)) \\ &+ \frac{w_2}{\tau_2} * L_1(\text{Gry}(q), \text{Gry}(r)) \end{aligned} \quad (1)$$

where the normalization factors τ_1, τ_2 and the weighting factors w_1, w_2 are automatically calculated using the Weighting by Max-τ algorithm with parameter α=0.001. The α-normalization algorithm first calculates τ_1 as the value that:

$$\mathbb{P}[L_1(\text{Ehd}(q), \text{Ehd}(r)) \leq \tau_1] = \alpha$$

for any two randomly-selected objects in $q, r \in \mathcal{Q}_v \cup \mathcal{R}_v$ (τ_2 is calculated analogously). Both w_1 and w_2 are then calculated as the values that maximize the value that α-normalizes d_v. More details on these algorithms in [1].

Approximate Search Once defined the distance, the approximate search retrieves the k=10 nearest neighbors using an approximation parameter T=1% with $\mathcal{P}$=5 pivots. This approximate search first selects $\mathcal{P}$ objects in $\mathcal{Q}_v$ and calculates the distance d_v with every object in $\mathcal{R}_v$. For any two objects an estimation of $d_v(q, r)$ can be calculated using the triangle inequality:

$$d_v(q,r) \approx \max_{p \in \mathcal{P}} |d_v(q,p) - d_v(p,r)|$$

This estimation can efficiently be evaluated with only 5 operations, and only for the $T^*|\mathcal{R}_v|$=45,222 objects with lowest estimations the real d_v was calculated. Finally, the 10 closest objects to q were selected.

Copy Localization Using the k approximate nearest neighbors for each visual query segment a copy localization is performed. This algorithm searches for chains of nearest neighbors belonging to the same reference video and offset. Each located chain obtains a score depending on the distance and the rank of each voter segment. Then, the chains for all the query videos created by the preprocessing task are combined. Finally, the chain with the highest score for each query video is reported in `PRISMA.m.nofa.EhdGry`, and the two chains with highest scores are reported in `PRISMA.m.balanced.EhdGry`.

2.4.2 Audio-Based Search

Let $\mathcal{Q}_a$ be the set of audio segments for query videos, and $\mathcal{R}_a$ be the set of audio segments for reference videos ($|\mathcal{Q}_a|$=306,304 and $|\mathcal{R}_a|$=4,441,717). The audio-based search aims for retrieving for each object in $\mathcal{Q}_a$ the k most similar objects in $\mathcal{R}_a$ according to a distance $d_a(q,r)$.

Audio Distance For any two audio segments q and r, their similarity was measured using the distance function:

$$d_a(q,r) \quad = \quad L_1(\text{Aud}(q), \text{Aud}(r)) \tag{2}$$

We have not submitted any Run for this audio-based search. However, the approximate search process is analogous to the Visual-Based search but replacing d_v with d_a.

2.4.3 Audio+Visual Search

Let $\mathcal{Q}_{av}$ be the set of a+v-segments for query videos, and $\mathcal{R}_{av}$ be the set of a+v-segments for reference videos. We created $\mathcal{Q}_{av}$ by combining sets $\mathcal{Q}_a$ and $\mathcal{Q}_v$ and their descriptors following the script `tv11.make.av.queries.sh`, which produced $|\mathcal{Q}_{av}|$=7,840,587 a+v-segments. We created $\mathcal{R}_{av}$ by combining sets $\mathcal{R}_a$ and $\mathcal{R}_v$ and their descriptors, producing $|\mathcal{R}_{av}|$=4,387,633 a+v-segments. The combination process requires that visual-segments and audio-segments have the same length to create an a+v-segment, and it guarantees that every created a+v-segment has all the visual and audio descriptors, i.e. an a+v-segment is discarded when only has one type of descriptor.

The audio+visual search aims for retrieving for each object in $\mathcal{Q}_{av}$ the k most similar objects in $\mathcal{R}_{av}$ according to a distance $d_{av}(q,r)$.

Audio+Visual Distance For any two a+v-segments q and r, their similarity was measured using the distance function:

$$\begin{aligned} d_{av}(q,r) \quad &= \quad \frac{w_1}{\tau_1} * L_1(\text{Ehd}(q), \text{Ehd}(r)) \\ &+ \quad \frac{w_2}{\tau_2} * L_1(\text{Rgb}(q), \text{Rgb}(r)) \\ &+ \quad \frac{w_3}{\tau_3} * L_1(\text{Aud}(q), \text{Aud}(r)) \end{aligned} \tag{3}$$

The normalization and weighting factors were calculated with the same algorithms from Equation 1. Then, we manually decrease w_3 because we already knew from TRECVID 2010 that audio was not as feasible as visual to detect copies. TRECVID CCD guidelines ensures a copy exists in both visual and audio tracks at the same time, however there are some valid copies which audio track is ruined by the audio transformation.[1]

Multi-Step A+V Search For Audio+Visual Search, we did not applied directly the approximate search from Section 2.4.1. The two major drawbacks for applying that approximate search to d_{av} are:

- The query set size has increased almost 7 times, thus the search time will also increase almost 7 times. Then, the search parameters T and $\mathcal{P}$ should be adjusted to reduce the search time (decreasing the accuracy).
- The distance function d_{av} is more difficult to approximate than d_a and d_v because involves more independent underlying metrics[2]. Then, T and $\mathcal{P}$ should be adjusted to increase the accuracy (increasing the search time).

To overcome these issues we have defined a two-steps search. Given a query video Q, the first step performs approximate searches with d_a and d_v to collect candidate reference videos $R(Q)$. The second step performs exact searches using d_{av} between every segment Q and reference segments $\mathcal{R}'_{av}(Q)$.

First Step A+V Search For the collecting step, we performed approximate searches with both distances d_a and d_v. In the case of d_a, we performed the approximate search using parameters k=30, T=2%, and $\mathcal{P}$=5 pivots. In the case of d_v, we performed the approximate search using parameters k=10, T=1%, and $\mathcal{P}$=5 pivots using the following distance (instead of Equation 1):

$$\begin{aligned} d_v(q,r) \quad &= \quad \frac{w_1}{\tau_1} * L_1(\text{Ehd}(q), \text{Ehd}(r)) \\ &+ \quad \frac{w_2}{\tau_2} * L_1(\text{Rgb}(q), \text{Rgb}(r)) \end{aligned} \tag{4}$$

For each query video Q, a unique list $kNN(Q)$ including the k nearest neighbors for each segment $q \in Q$ according to d_a or d_v is calculated. All duplicated reference segments are removed from $kNN(Q)$, then a voting procedure is performed, where each segment in $kNN(Q)$ gives one vote to the reference video that owns it. Finally, the set $R(Q)$ is created with the D most voted reference videos that received at least 2 votes. We defined parameter D=40 based exclusively on resulting search time. The output of this step is a set of reference videos $R(Q)$ for each query video Q, where $|R(Q)| \leq D$.

Second Step A+V Search For every a+v-segment q from query video Q, this step performs an exact kNN search using d_{av}. The search space $\mathcal{R}'_{av}(Q)$ is defined as all the a+v-segment from reference videos in $R(Q)$ (thus $|\mathcal{R}'_{av}(Q)| \ll |\mathcal{R}_{av}|$). With this reduction of the search space (which

[1] Moreover, after analyzing the ground truth, we have realized that an (maybe unintended) audio transformation replaces the audio track from a valid copy with an audio track from an unrelated video. This creates some a+v queries where visual track matches the original while the audio track does not match (i.e. the copy exists only in the visual track), e.g. query 8960.mpg with original video id 288. This also happens with queries: 6008.mpg, 7468.mpg, 7996.mpg, 9261.mpg, 9472.mpg, 9517.mpg, and maybe others.

[2] This behavior is analyzed in [1], however it is still an open issue to quantify it.

only includes the a+v-segments from the D most promising videos) the exact search with d_{av} can be efficiently resolved.

Copy Localization Analogous to Section 2.4.1, the copy localization is performed using the (exact) kNN lists according to d_{av}. The chain with the highest score for each query video is reported in `PRISMA.m.nofa.EhdRgbAud`, and the two chains with highest scores are reported in `PRISMA.m.balanced.EhdRgbAud`.

3. RESULTS ANALYSIS

Evaluated visual transformations were: **T1**: simulated camcording; **T2**: picture-in-picture (PIP) original video in foreground; **T3**: insertion of pattern; **T4**: strong reencoding; **T5**: change of gamma; **T6**: three transformations between blur, change of gamma, frame dropping, contrast, reencoding, ratio, and white noise; **T8**: three transformations between crop, shift, contrast, caption, mirroring, insertion of pattern, and PIP original video in background; **T10**: random combination of three previous transformations; **T7** and **T9** were not evaluated.

Evaluated audio transformations were: **A1**: no transformation; **A2**: mp3 compression; **A3**: mp3 compression and multiband companding; **A4**: bandwidth limit and single-band companding; **A5**: mix with speech; **A6**: mix with speech and multiband compress; **A7**: bandpass filter mix with speech and compress.

Query videos were generated by defining 201 base queries and applying the 8 visual transformations $\times$ 7 audio transformations.

The evaluation of a submitted Run relied on three measures:

- **NDCR**: Measures the effectiveness of the detection, weighting the probability of missing a detection and the probability to falsely indicate that there is a copy for a query video (NDCR=$P_{\text{MISS}} + \beta \cdot P_{\text{FA}}$). The closer to zero the better the effectiveness, a trivial NDCR of 1.0 can be obtained by submitting an empty Run, thus a good result should not be greater that 1.0.
- **F1**: Measures the accuracy in localization after a copy has been correctly detected. The closer to 1.0 the better the accuracy.
- **Mean processing time**: Measures the efficiency for processing queries.

NDCR was evaluated for two profiles: Balanced (β=200) and No False Alarms (NoFA, β=200,000). TRECVID calculates these measures separately for each of the 56 transformations, and for comparison purposes we include the average result for all transformations. These three measures are calculated at a submitted decision threshold. Additionally, Optimal NDCR and Optimal F1 are calculated by cutting the Run at the optimal decision score.

Twenty-two teams participated in the evaluation. Each team submitted at most 4 Runs, which resulted in 32 submissions for NoFA profile and 41 submissions for Balanced profile. Then, we classified every Run into audio-only (A), visual-only (V), and audio+video (AV). We stated that a Run is visual-only when its results for NDCR and F1 are identical for the 7 audio transformations in a same visual transformation (thus, its results are not influenced by changes in audio). Analogously, we stated that a Run is audio-only when NDCR and F1 are identical for the 8 visual transformations in a same audio transformation. Despite our submitted decision threshold was fairly accurate for NoFA profile, we will base our analysis on Optimal NDCR and Optimal F1 rather than NDCR and F1.

Figures 1 and 2 summarize the results for the NoFA and Balanced profiles for all the submitted Runs. We achieve better results for NoFA profile than for Balanced profile (6th and 10th of 32), which confirms our conclusion from our last participation that global descriptors perform better for NoFA profile than for Balanced profile [1]. Non-copies are usually easier to discard for global descriptors than for local descriptors, thus global descriptor may detect more correct correct copies before the first false alarm. The NDCR decreases for Balanced profile because copies with complex transformations may be undetectable for global descriptors affecting the detection rate.

Comparing processing time, our submissions are between the fastest Runs with good detection performance. This satisfactory balance between effectiveness and efficiency is due to approximate search parameters properly adjusted. Note also that the processing time for audio+video search only increases about 1.3 times (instead of the 7 times for the naive approach) due to the multi-step search.

Figure 3 averages the results by audio transformation and video transformation. This figure shows the good performance at localizing copies for `EhdRgbAud` Runs, which both achieve the highest Optimal F1 for T3, T4 and T8.

Figure 4 shows the results provided by TRECVID. The system performs better than the median for every transformation for NDCR, F1 and processing time.

We ran our tests on an Intel Core i7-2600K CPU (3.4 GHz $\times$ 4 cores) with 8 GB RAM on a GNU/Linux 2.6.38. Our system is implemented in C using OpenCV and FFmpeg libraries.

4. CONCLUSIONS

One question that rose from TRECVID 2010 was the feasibility of performing multimodal fusion at an earlier stage than the decision level. We have shown with this work that it is indeed possible and also that it can be performed efficiently. We envisioned this kind of fusion as a conclusion of our work for TRECVID 2010, however, to fulfill this issue we have required to adapt an audio descriptor and to design a multi-step search.

This novel multi-step search not only enables to fuse global descriptors with audio descriptors, but also to fuse with local descriptors. We plan to work on this triple fusion in a future. Other issues we plan to address are: an improved algorithm for selecting reference videos in the multi-step search, and to improve the detection performance when the copied audio track does not match the original audio track.

5. REFERENCES

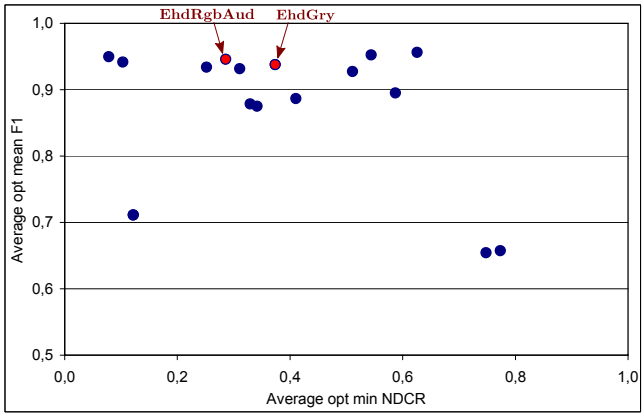
[1] J. M. Barrios and B. Bustos. Competitive content-based video copy detection using global descriptors. *Multimedia Tools and Applications*, pages 1–36, 2011.

[2] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *Proc. of the int. symp. on Music Information Retrieval (ISMIR'02)*, pages 107–115, 2002.

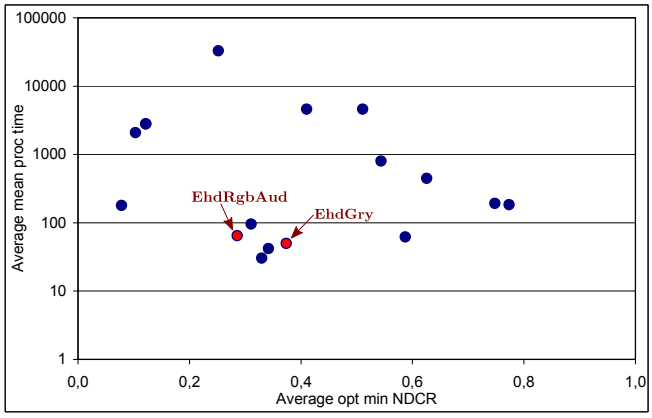
[3] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

#	Run	Average opt min NDCR	Average opt mean F1	Average mean time	Type
1	cascade	0.078	0.950	179	AV
2	dodo	0.103	0.942	2079	AV
3	V48A66T160	0.122	0.711	2792	AV
4	V48A66T60	0.122	0.711	2792	AV
5	tyche	0.252	0.934	32848	AV
6	**EhdRgbAud**	**0.286**	**0.946**	**64**	**AV**
7	0	0.311	0.931	96	AV
8	1	0.330	0.879	30	AV
9	3	0.342	0.875	42	AV
10	**EhdGry**	**0.374**	**0.938**	**50**	**V**
11	AudioOnly	0.410	0.887	4589	A
12	orange1	0.511	0.927	4589	AV
13	bhgccd	0.544	0.952	801	V
14	wsyVA	0.587	0.895	62	AV
15	bgccd	0.626	0.956	445	V
16	Uvote	0.748	0.654	191	V
17	Wvote	0.773	0.658	184	V
18	4sys	13.807	0.682	4	V
19	videoonly1	14.243	0.785	727	AV
20	brnoccd	23.812	0.709	1575	AV
21	base	27.236	0.683	1	V
22	2sys	27.237	0.621	2	V
23	videoonly2	27.590	0.773	719	V
24	multimodal	57.768	0.948	601	AV
25	test	107.79	0.000	4	V
26	mfh	117.32	0.000	1	AV
27	ITUMSPR1	137.98	0.370	953	A
28	ch4of12	275.33	0.930	110	A
29	zhVideo	400.57	0.914	62	V
30	VideoNOFA7	401.45	0.054	142	V
31	VideoNOFA8	401.45	0.054	142	V
32	1	9999.1	0.000	50	V

(a) Average results for all submissions to No False Alarms profile. Sorted by Average optimal min NDCR.



(b) Average optimal min NDCR (0–1) vs Average optimal mean F1 (0.5–1).

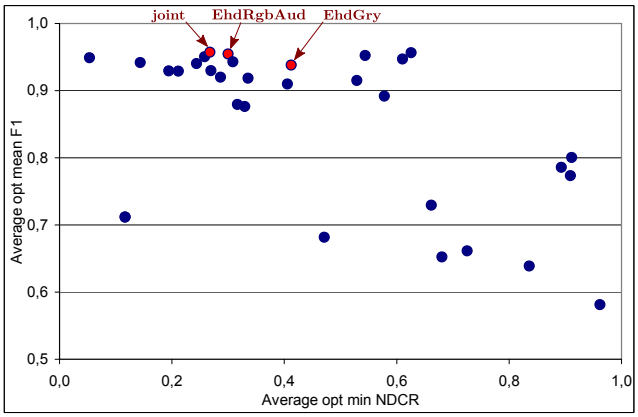


(c) Average optimal min NDCR (0–1) vs Average mean processing time (1–100000, log scale).

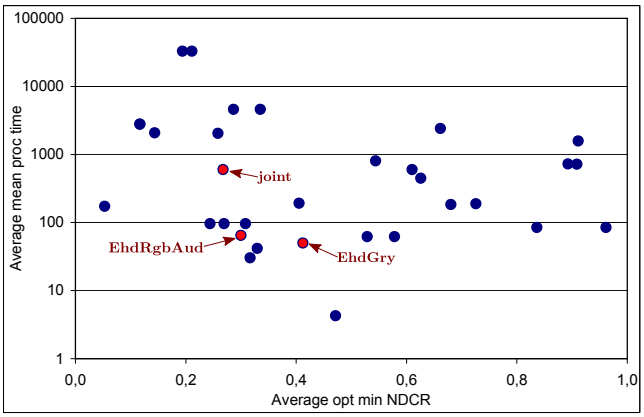
Figure 1: Results for No False Alarms Profile using the optimal decision threshold. Values averaged for the 56 transformations. Runs `PRISMA.m.nofa.EhdGry` and `PRISMA.m.nofa.EhdRgbAud` highlighted.

#	Run	Average opt min NDCR	Average opt mean F1	Average mean time	Type
1	cascade	0.053	0.949	172	AV
2	V48A66T58B	0.117	0.712	2792	AV
3	V48A66T65B	0.117	0.712	2792	AV
4	dodo	0.144	0.942	2079	AV
5	zozo	0.194	0.929	32848	AV
6	themis	0.211	0.929	32848	AV
7	1	0.244	0.940	96	AV
8	deaf	0.258	0.950	2041	V
9	**joint**	**0.268**	**0.957**	**601**	**AV**
10	2	0.270	0.930	96	AV
11	orange3	0.287	0.920	4589	AV
12	**EhdRgbAud**	**0.300**	**0.955**	**64**	**AV**
13	3	0.309	0.943	96	AV
14	2	0.317	0.879	30	AV
15	4	0.330	0.876	42	AV
16	VideoOnly	0.335	0.918	4589	V
17	audioonly	0.406	0.910	192	A
18	**EhdGry**	**0.412**	**0.938**	**50**	**V**
19	4sys	0.471	0.682	4	V
20	zhVideo	0.529	0.915	62	V
21	bhgccd	0.544	0.952	801	V
22	wsyVA	0.578	0.892	62	AV
23	multimodal	0.610	0.947	601	AV
24	bgccd	0.626	0.956	445	V
25	mask	0.662	0.729	2393	AV
26	Wvote	0.681	0.652	184	V
27	Uvote	0.726	0.661	188	V
28	fsift	0.836	0.639	84	V
29	videoonly1	0.893	0.786	727	AV
30	videoonly2	0.909	0.773	719	V
31	brnoccd	0.911	0.800	1575	AV
32	fsift2	0.962	0.581	84	V
33	ch4of12	1.005	0.930	110	A
34	ITUMSPR2	1.125	0.417	953	AV
35	mfh	1.193	0.000	1	AV
36	chth	1.256	0.928	248	A
37	VideoBal5	1.395	0.054	142	V
38	VideoBal6	1.395	0.054	142	V
39	chcode	1.989	0.929	120	A
40	1	19.087	0.000	49	V
41	test	19.089	0.242	4	AV

(a) Average results for all submissions to Balanced profile. Sorted by Average optimal min NDCR.



(b) Average optimal min NDCR (0–1) vs Average optimal mean F1 (0.5–1).



(c) Average optimal min NDCR (0–1) vs Average mean processing time (1–100000, log scale).

Figure 2: Results for Balanced Profile using the optimal decision threshold. Values averaged for the 56 transformations. Runs `PRISMA.m.balanced.EhdGry`, `PRISMA.m.balanced.EhdRgbAud` and `Telefonica-research.m.balanced.joint` highlighted.

	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T8**	**T10**	**AVG**
Optimal NDCR	0.217	0.213	0.304	0.329	0.261	0.330	0.347	0.551	0.271	0.110	0.150	0.386	0.351	0.147	0.321	0.286
Global Rank	6^{th}	5^{th}	6^{th}	6^{th}	7^{th}	8^{th}	7^{th}	10^{th}	6^{th}	7^{th}	8^{th}	15^{th}	11^{th}	4^{th}	5^{th}	6^{th} of 32
Optimal F1	0.950	0.951	0.945	0.947	0.947	0.939	0.944	0.897	0.947	0.971	0.961	0.932	0.932	0.970	0.957	0.946
Global Rank	5^{th}	5^{th}	6^{th}	6^{th}	5^{th}	6^{th}	5^{th}	12^{th}	5^{th}	1^{st}	1^{st}	13^{th}	8^{th}	1^{st}	2^{nd}	5^{th} of 32

(a) Run `PRISMA.m.nofa.EhdRgbAud`.

	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T8**	**T10**	**AVG**
Optimal NDCR	0.374	0.374	0.374	0.374	0.374	0.374	0.374	0.761	0.321	0.119	0.239	0.403	0.515	0.209	0.425	0.374
Global Rank	11^{th}	11^{th}	9^{th}	10^{th}	10^{th}	10^{th}	8^{th}	13^{th}	7^{th}	8^{th}	9^{th}	16^{th}	16^{th}	6^{th}	11^{th}	10^{th} of 32
Optimal F1	0.938	0.938	0.938	0.938	0.938	0.938	0.938	0.887	0.945	0.966	0.949	0.952	0.924	0.934	0.944	0.938
Global Rank	10^{th}	9^{th}	8^{th}	8^{th}	7^{th}	7^{th}	7^{th}	14^{th}	8^{th}	2^{nd}	6^{th}	5^{th}	10^{th}	10^{th}	5^{th}	7^{th} of 32

(b) Run `PRISMA.m.nofa.EhdGry`.

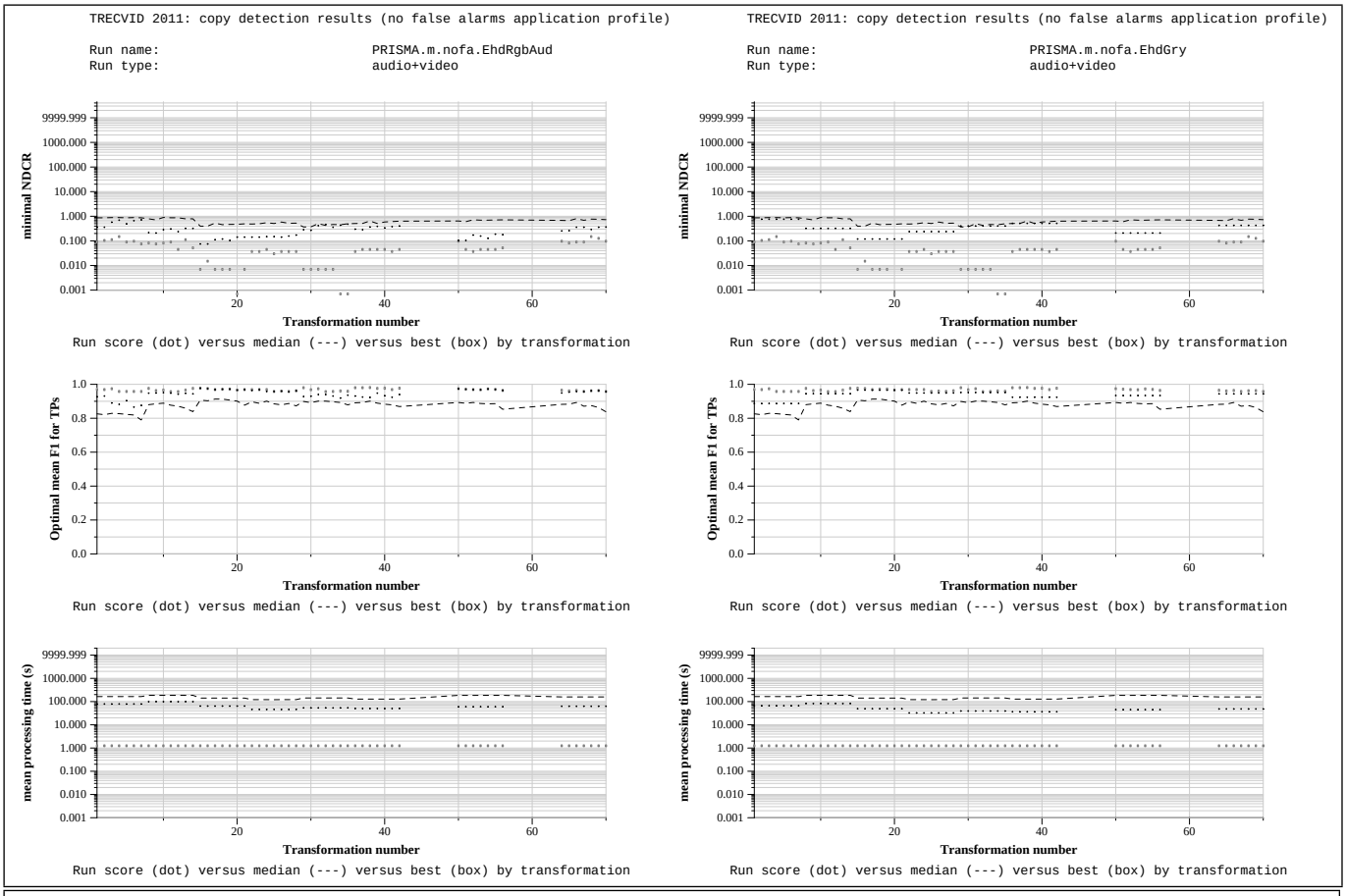
	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T8**	**T10**	**AVG**
Optimal NDCR	0.223	0.222	0.315	0.341	0.278	0.357	0.363	0.541	0.252	0.109	0.195	0.437	0.377	0.181	0.307	0.300
Global Rank	6^{th}	6^{th}	11^{th}	15^{th}	11^{th}	16^{th}	14^{th}	16^{th}	9^{th}	9^{th}	8^{th}	20^{th}	17^{th}	5^{th}	11^{th}	12^{th} of 41
Optimal F1	0.957	0.959	0.956	0.952	0.954	0.952	0.953	0.899	0.948	0.973	0.974	0.957	0.954	0.976	0.957	0.955
Global Rank	5^{th}	1^{st}	4^{th}	6^{th}	3^{rd}	4^{th}	3^{rd}	19^{th}	7^{th}	1^{st}	1^{st}	3^{rd}	4^{th}	1^{st}	2^{nd}	3^{rd} of 41

(c) Run `PRISMA.m.balanced.EhdRgbAud`.

	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T8**	**T10**	**AVG**
Optimal NDCR	0.412	0.412	0.412	0.412	0.412	0.412	0.412	0.761	0.313	0.104	0.323	0.495	0.607	0.293	0.403	0.412
Global Rank	18^{th}	18^{th}	14^{th}	18^{th}	17^{th}	17^{th}	15^{th}	22^{th}	10^{th}	8^{th}	14^{th}	21^{th}	22^{th}	16^{th}	16^{th}	18^{th} of 41
Optimal F1	0.938	0.938	0.938	0.938	0.938	0.938	0.938	0.887	0.945	0.966	0.950	0.952	0.926	0.936	0.941	0.938
Global Rank	15^{th}	13^{th}	13^{th}	13^{th}	9^{th}	10^{th}	9^{th}	24^{th}	10^{th}	3^{th}	7^{th}	7^{th}	16^{th}	15^{th}	7^{th}	11^{th} of 41

(d) Run `PRISMA.m.balanced.EhdGry`.

Figure 3: Results for NDCR and F1 averaged by Audio Transformation and Visual Transformation.



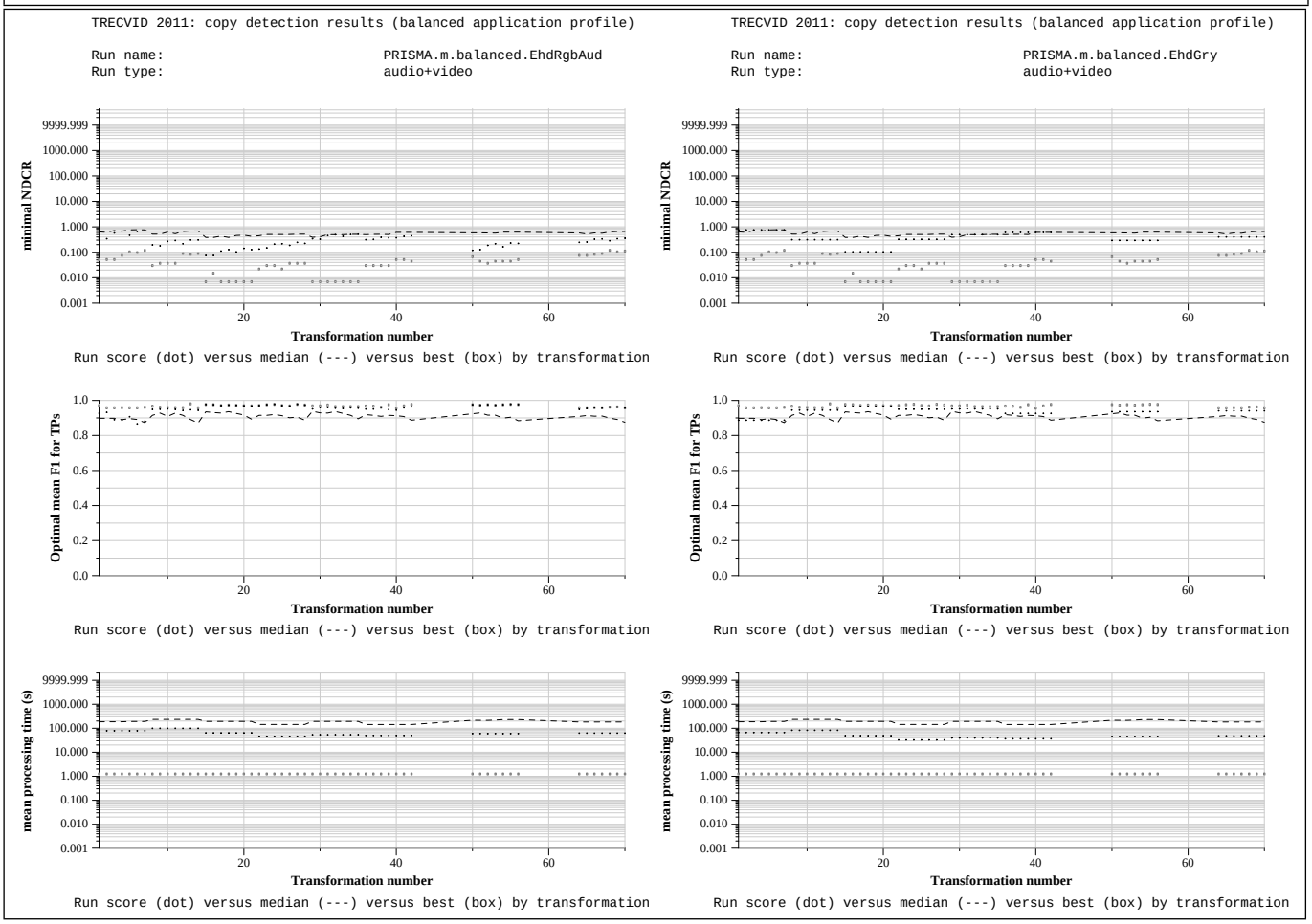


Figure 4: TRECVID results for our four submitted Runs.