Real-time detection of partial video copy on TV workstation

Van-Hao LE, Mathieu Delalandre and Donatello Conte
LIFAT Laboratory
Tours city, France
firstname.lastname@univ-tours.fr

***Abstract*—A system for the real-time copy detection of live TV videos is presented. A TV workstation supports the real-time and multichannel processing. Real-time NCC features are used for matching. A key-frame selection method ensures the robustness, the response and processing time optimization. Experiments are reported for time processing and accuracy on a public dataset against competitive methods.**

***Index Terms*—Real-time, detection, video copy, TV, workstation, multichannel, NCC, key-frame**

I. Introduction

Over the last ten years, connected TVs (smart TVs, TV apps on mobile) became dominant in the industry and market. Compared to traditional TVs, the devices offer new services to users as interactive media and on-demand content. The recent trends for parallel computing and artificial intelligence make possible the design of systems for the live TV analysis. This could offer a wide range of applications as the video-based soccer analysis or the detection of text, logo and advertising.

In this paper, we address the problem of partial video copy detection. The goal is to find one or more segments of a query video which have transformed copies [1]. The detection of partial copies can be done for web videos or for the TV. This has practical applications such as the TV analytic [2] or the commercial detection [3]. Fig. 1 gives examples of TV content targeted while applying the detection.

Several works have been published in the literature for the off-line detection of partial video copy on TV [2], [3]. However, little work addresses the problem for the live TV [4]. The detection of partial video copy for the live TV raises two main open problems. The first is the real-time processing and capture of the video signal. The second problem is the processing of multiple videos streams as the TV is delivered as a set of channels. At the best of our knowledge, the real-time and multichannel detection of partial video copy has neither been addressed in the literature.

Figure 1: Repeated content on TV (a) jingle and generic for series (b) advertising (c) news

We propose here a system to address these issues. A TV workstation is presented in section II to support the real-time and multichannel processing. Section III is dedicated to our real-time detection method. Experiments are reported in section IV and section V will provide conclusions.

II. The TV workstation

For a real-time processing, the TV signal must be captured with a minimum latency. Another constraint is the processing of multiple TV channels. These aspects are little discussed in the literature. To support these processes, we present here a TV workstation Fig. 2. A first version of the station has been presented in [5]. We remind here the architecture while developing the updates.

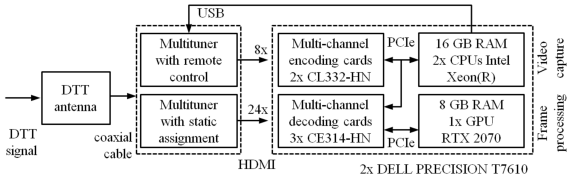

Figure 2: The TV workstation

The TV streams are broadcasted over different networks including the IPTV, DTT and SaT signals[1]. The IP based streams suffer from a big latency and jitter compared to DTT and SaT signals [6]. This is a critical point for the real-time applications. To solve this problem, the main capture in our workstation is driven from the DTT signal. This signal is processed with a multiple tuner. It demodulates the DTT signal into multiple video streams corresponding to the channels.

The workstation is composed of two DELL PRECISION T7610 computers having a dual-core architecture. They are set with two Intel Xeon(R) CPUs $E5-2620$ 2GHz and a GPU RTX 2070. The GPUs offer a deep processing ability but suffer from latency [7]. They are suitable for the soft real-time. For the hard real-time, CPU processing is recommended. It becomes competitive with GPU while using key capabilities [8].

For the real-time issue, the video streams are processed with capture cards Table I. These cards process the video streams at the hardware level for decoding, control of the FPS, downscaling, color-space conversion and transfer to the main memory. The workstation embeds two levels of card for the video capture and processing. We have selected the Avermedia cards CL332−HN and CE314−HN. The workstation has a processing capacity of 32 channels (24 for processing, 8 for capture).

Cards	Task	N	Channels	FPS
2x CL332−HN	capture	2	4 = 2x 2	30
3x CE314−HN	processing	4	12 = 3x 4	

N is the number of video streams processed per card

Table I: Configuration of a T7610

[1]Internet Protocol, Digital Terrestrial Television and Satellite

III. Real-time detection of video copy

The detection of partial video copy must process in real-time and with multiple channels. To fit with this requirement, time-efficient features must be computed at the frame level. Section III-A introduces our features whereas section III-B is dedicated to the key-frame selection problem.

A. Real-time features for frame matching

Several features have been investigated to characterize the frames for the video copy detection such as the CNN [9], SURF [10], BRIEF [11] features and the LBP [12]. These features can handle major degradations but have a little real-time ability.

For the real-time video copy detection, features processed in the compressions domain [13] and the methods for image matching [14] can be used. We have investigated the image matching that can be supported by our workstation for spatial decoding. A large number of distances and metrics have been proposed for image matching. We have considered the Zero-mean Normalized Cross-Correlation ($\mathbf{ZNCC}$) that is robust to noise, contrast-invariant and fits well with the detection problem.

$\mathbf{ZNCC}$ is given in Eq. (1) For simplification, we use here the one dimensional notation. $\mathbf{I}(\mathbf{x})$ is a discrete function taking values in an image. $\mathbf{ZNCC}(\mathbf{I}, \mathbf{I}^*)$ compares the image $\mathbf{I}$ to an image $\mathbf{I}^*$. $\sigma_{\mathbf{I}}$ and $\sigma_{\mathbf{I}^*}$ are the standard deviations of the two images and $\bar{\mathbf{I}}$, $\bar{\mathbf{I}}^*$ the image means. $\mathbf{ZNCC} \in [-1, 1]$ where 1 is the perfect correlation. For short, it is common to refer $\mathbf{ZNCC}$ as $\mathbf{NCC}$.

$$\mathbf{ZNCC}(\mathbf{I}, \mathbf{I}^*) = \frac{\sum_{\forall \mathbf{x}} (\mathbf{I}(\mathbf{x}) - \bar{\mathbf{I}}(\mathbf{x}))(\mathbf{I}^*(\mathbf{x}) - \bar{\mathbf{I}}^*(\mathbf{x}))}{\sigma_{\mathbf{I}} \sigma_{\mathbf{I}^*}} \quad (1)$$

Computing the $\mathbf{NCC}$ could be time consuming as it requires adding, timing, rooting and summing operations. Methods have been proposed to speed up the computation Table II. The methods [15], [16] require fingerprinting not compatible with the processing of multiple video streams. The upper-bounding [17] ensures a strong optimization but offers a little predictability on the execution time. For our implementation, we have selected the parallel $\mathbf{NCC}$ [18]. It reformulates the brute-force

matching with a specific pipeline and SSE[2] instructions for optimization. It can be applied to multiple video streams as no fingerprinting is needed and the processing is achieved with predictability.

Methods	Time efficience	Fingerprinting	Predictability
Pyramid [15]	++	Pyramid	no
Block matching [16]	++	Image Integral	yes
Upper-Bounding [17]	+++	None	no
Parallel [18]	+	None	yes

Table II: Fast $\mathbf{NCC}$ computation
$+$ is the worst / $+++$ is the best

B. Key-frame selection

Representative frames have to be selected for robustness and time optimization. This is referred as the key-frame selection problem in the literature.

A traditional approach is to measure the distortion between consecutive frames in the reference videos. The frames with a significant visual change are selected as key-frames. The distortion could be measured with deep features [9], the $\mathbf{NCC}$ difference [14] or the maximum entropy [19]. An alternative is to minimize the distortion between reference and altered videos. In [20] altered videos are used to design key-frame fingerprinting robust to scale invariance, timeshifting and video cropping.

We propose here a key-frame selection method dedicated to the detection of live TV videos. The selection is driven either for robustness and either for response and processing time optimization. Our goodness criterion for selection is given in Eq. (2)[3]. This criterion maximizes the $\mathbf{NCC}$ intra and inter-class distance. $\mathbf{X}$ is a reference frame, $\tilde{\mathbf{X}}_\mathbf{0}, \ldots, \tilde{\mathbf{X}}_\mathbf{s}, \ldots, \tilde{\mathbf{X}}_\mathbf{m}$ and $\mathbf{Y_0}, \ldots, \mathbf{Y_t}, \ldots, \mathbf{Y_n}$ are the sets of true positive and negative frames. $\mathbf{NCC}_{\min}(\mathbf{X}, \tilde{\mathbf{X}}_\mathbf{s})$ and $\mathbf{NCC}_{\max}(\mathbf{X}, \mathbf{Y_t})$ are then the minimum and maximum distances between the frame $\mathbf{X}$ and the true positive and negative sets.

$$\phi(\mathbf{X}) = \mathbf{NCC}_{\min}(\mathbf{X}, \tilde{\mathbf{X}}_\mathbf{s}) - \mathbf{NCC}_{\max}(\mathbf{X}, \mathbf{Y_t}) \quad (2)$$

[2]Streaming SIMD Extensions
[3]For simplification, we consider $\mathbf{NCC}(\mathbf{I}, \mathbf{I}^*) \geq 0$

For optimization, the number of key-frames must be minimized. Our approach is depicted in Fig. 3. To guaranty a response time for detection, only a first interval Δ of the video is used for selection. The frames with $\phi(\mathbf{X}) < 0$ are ignored for precision. Selection is then driven with peak detection. Low peaks can be still candidate and an automatic threshold has to be found. For solving, we interpolate a function $\mathbf{F}(\beta)$ from the cumulative distribution $\phi(\mathbf{X_k})$. The threshold is obtained with the local derivative $\frac{\partial^2 \mathbf{F}(\beta)}{\partial \beta^2}$ and zero-crossing. None zero-crossing could appear with a low Δ. The minimum threshold over all the videos is then used. A Non-maximum Suppression (NMS) is applied to remove the low peaks close in the time domain.

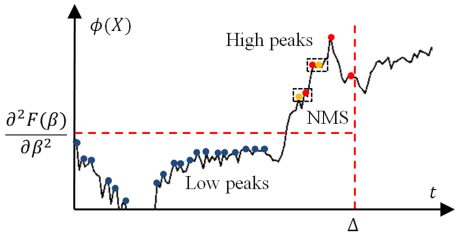

Figure 3: Key-frame selection

IV. Experiments and Results

We provide in this section experiments. Section IV-A investigates the time processing for frame matching. The aspects of performance evaluation and datasets are provided in section IV-B.

A. Time processing for frame matching

Time experiments are reported here for frame matching with the real-time $\mathbf{NCC}$. We have considered the CPUs for processing for a hard real-time implementation [7], [8]. For acceleration, multi-threading was applied. 12 threads were launched per CPU E5-2620 with a total of 48 for the workstation. The experiments have been driven using the 24 channels with 2 threads per channel.

Table III gives the results as maximum numbers of matched frames. These numbers have been fixed in order to respect a rate of 30 FPS i.e. none matching of a frame could exceed an execution time of $1/30$ s. Our workstation supports several tens

thousands of matching considering low resolution frames $\in [32^2, 96^2]$. A near thousand of matching could be computed per channel.

Frame size	32^2	64^2	96^2
Workstation	88.8K	17.2K	8K
Channel	3700	720	340

K is thousand

Table III: Matched frames at 30 FPS hard real-time
4 CPUs E5 − 2620 / 24 channels

B. Datasets and performance evaluation

This section gives a performance evaluation of our system. At the best of our knowledge, a single dataset exists for the detection of TV repeated content [2]. However, this dataset is not public available. We have adapted the SVD dataset [21] for the online video copy detection. It is the largest public dataset having a near duration of 3 thousands of hours. It contains videos captured from mobile devices. The degradations are linked to online rendering such as the cropping or the rotation. For adaptation to the TV use-case, we propose a pipeline in Fig. 4 illustrated in Fig. 5.

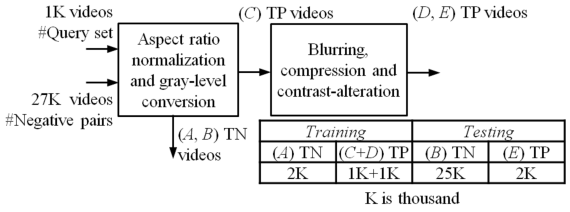

Figure 4: Pipeline to process the SVD dataset
K is thousand

(a) (b) (c)

Figure 5: A TP video (a) *#Query set* (b) normalized in set *C* (c) vs. near-duplicate in sets *D, E*

We have used the *#Query set* and *#Negative pairs* composed of 1K and 27K videos. The negative pairs are sets of videos very similar to the queries but not near-duplicates. We have applied an aspect ratio normalization to get videos at a same resolution Fig. 5 (b). For real-time processing Table III, we have re-scaled the frames at sizes 32×24. The obtained videos from the *#Query set* and *#Negative pairs* are true positives (TP) and true negatives (TN) respectively, dispatched into three sets *A*, *B* and *C*.

We have altered the set *C* of TP videos with blurring, compression and contrast alteration to get near-duplicates Fig. 5 (c). These alterations fit well with the TV detection. The videos have been dispatched in two sets *D*, *E* for training and testing.

Table IV gives the P/R and F-measure of our system. These results are achieved with a threshold maximizing the F score. We have set different maximum response times Δ for detection. We report too the number of selected key-frames.

Method	Δ (ms)	P (%)	R (%)	F (%)	#KF
Proposed	300	99.14	98.67	98.91	1026
	1000	99.06	99.16	99.11	1351
	3000	99.30	99.02	99.20	2582
CNN [9]	1000	97.44	99.07	98.25	1351
SURF [10]		93.85	91.59	92.71	1356
BRIEF [11]		94.31	91.37	92.82	1353
NCC diff [14]		95.35	92.37	93.84	1350
Max entropy [19]		97.71	97.71	97.71	1353

Table IV: P, R and F scores

Comparative results are provided with competitive methods for key-frame selection. As discussed in section III-B, the standard approach is to select the key-frames having significant visual changes in the reference videos. This could be obtained with the CNN features [9], the **NCC** difference [14] or the maximum entropy [19]. For a further comparison, we have considered too the SURF and BRIEF features used for frame matching [10], [11]. The setting of the methods is done to reach an equal response time Δ and number of key-frames.

Our key-frame selection achieves the strongest detection with a score $F \simeq 0.99$. This results from goodness criterion using the **NCC** features for the selection similar to the frame matching. A higher Δ

parameter achieves a better detection but relaxes the response time. A near 1 to 3 key-frames are selected per video. Competitive detections are obtained at a slightly gap $F \simeq 0.98$ with the CNN features [9] and the max entropy [19] for selection, considering a same level complexity and response time. As a general trend, our system is able to detect in real-time a near 80K of reference videos from 24 channels with a strong accuracy.

V. Conclusions

A system for the real-time partial copy detection of live TV videos was presented. It is based on a workstation, real-time NCC features and a key-frame selection method. It detects in real-time a near 80K reference videos on 24 channels with a strong detection accuracy $F \simeq 0.99$. Our key-frame selection outperforms the competitive methods at a same level complexity and response time with a slightly gap. The detection problem for the live TV is characterized by a low-level degradation and a high scalability. This requires to design large-scale and public TV datasets for further experiments.

References

[1] Y. Hu, Z. Mu and X. Ai, "STRNN: End-to-End Deep Learning Framework for Video Partial Copy Detection," Journal of Physics: Conference Series (JPCS), vol. 1237, no. 2, 2019.

[2] J.H. Chenot and G. Daigneault, "A large-scale audio and video fingerprints-generated database of TV repeated contents," International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1-6, 2014.

[3] M. Li, Y. Guo and Y. Chen, "CNN-based Commercial Detection in TV Broadcasting," International Conference on Network, Communication and Computing (ICNCC), pp. 48-53, 2017.

[4] Y. Zhang, Q. Li, H. Tong, J. Badilla, Y. Zhang and D. Wang, "Crowdsourcing-based Copyright Infringement Detection in Live Video Streams," International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 367-374, 2018.

[5] M. Delalandre, "A Workstation for Real-Time Processing of Multi-Channel TV," International Workshop on AI for Smart TV Content Production, Access and Delivery (AI4TV), pp. 53-54, 2019.

[6] M. Dabrowski, R. Kolodynski and W. Zielinski, "Analysis of Video Delay in Internet TV Service over Adaptive HTTP Streaming," Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 143-15, 2015.

[7] V. Golyanik, M. Nasri and D. Stricker, "Towards Scheduling Hard Real-Time Image Processing Tasks on a Single GPU," International Conference on Image Processing (ICIP), pp. 4382-4386, 2017.

[8] L. Lacassagne, D. Etiemble, H. Zahraee, A. Dominguez and P. Vezolle, "High Level Transforms for SIMD and Low-Level Computer Vision Algorithms," Workshop on Programming Models for SIMD/Vector Processing (WPMVP), pp. 49-56, 2014.

[9] C. Zhang, B. Hu, Y. Suo, Z. Zou, and Y. Ji, "Large-Scale Video Retrieval via Deep Local Convolutional Features," Advances in Multimedia, vol. 2020, no. 8, pp. 1-8, 2020.

[10] G. Ozbulak, F. Kahraman and S. Baykut, "Robust Video Copy Detection in Large-Scale TV Streams using Local Features and CFAR based Threshold," Conference on Digital Signal Processing (DSP), pp. 124-128, 2016.

[11] Y. Zhang and X. Zhang, "Effective Real-Scenario Video Copy Detection," International Conference on Pattern Recognition (ICPR), pp. 3940-3945, 2016.

[12] M. Saddique, K. Asghar, U.I. Bajwa, M. Hussain and Z. Habib, "Spatial Video Forgery Detection and Localization using Texture Analysis of Consecutive Frames," Advances in Electrical and Computer Engineering, vol. 19, no. 3, pp. 97-108, 2019.

[13] K.W. Liang, Y.C. Chen, Z.Y. Chen and P.C. Chang, "Video Copy Detection based on HEVC intra Coding Features," International Conference on Consumer Electronics (ICCE), pp. 108-111, 2015.

[14] Z.J. Guzman-Zavaleta and C. Feregrino-Uribe, "Towards a Video Passive Content Fingerprinting Method for Partial-Copy Detection Robust against Non-Simulated Attacks," PLOS one, vol. 11, no. 11, pp. 1-19, 2016.

[15] Y. Fouda and K. Ragab, "An efficient Implementation of Normalized Cross-Correlation Image Matching based on Pyramid," International Joint Conference on Awareness Science and Technology (iCAST) & Ubi-Media Computing (UMEDIA), pp. 98-103, 2013.

[16] G. Facciolo, N. Limare and E. Meinhardt-Llopis, "Integral Images for Block Matching," Image Processing On Line (IPOL), vol. 4, pp. 344-369, 2013.

[17] G. Adhikari, S.K. Sahani, M.S. Chauhan and B.K. Das, "Fast Real-Time Object Tracking based on Normalized Cross-Correlation and Importance of Thresholding Segmentation," International Conference on Recent Trends in Information Technology (ICRTIT), pp. 1-5, 2016.

[18] R. Moller, M. Horst and D. Fleer, "Illumination Tolerance for Visual Navigation with the Holistic Min-Warping Method," Robotics, vol. 3, pp. 22-67, 2014.

[19] Y. Hou, X. Wang, S. Liu and Y. Zhang, "Video Copy Detection based on Uniform Local Binary Pattern," DEStech Transactions on Computer Science and Engineering, 2018.

[20] O. Cirakman, B. Gunsel, N.S. Sengor and S. Kutluk, "Content-based Copy Detection by a Subspace Learning based Video Fingerprinting Scheme," Multimed Tools Appl, vol. 71, pp. 1381-1409, 2014.

[21] Q.Y. Jiang, Y. He, G. Li, J. Lin, L. Li and W. Li, "SVD: A Large-Scale Short Video Dataset for Near-Duplicate Video Retrieval," International Conference on Computer Vision (ICCV), pp. 5280-5288, 2019.

Acknowledgement

The authors wish to thank the LIFAT Laboratory and the Polytech Tours institute for their funding support for the TV workstation.