Une large base de données pour la détection de segments de vidéos TV

Van-Hao Le, Mathieu Delalandre, Donatello Conte

Une large base de données pour la détection de segments de vidéos TV

Van-Hao Le[1] Mathieu Delalandre[1] Donatello Conte[1]

[1] Laboratoire LIFAT, Université de Tours
64 avenue Jean Portalis, 37200 Tours, France
prénom.nomdefamille@univ-tours.fr

Résumé

Cet article est en lien avec la problématique d'évaluation de performance pour la détection de segments de vidéos. De nombreuses bases de données ont été proposées à partir de vidéos sur Internet. Cependant, la détection de segments est un problème inhérent à la diffusion en continu de flux vidéo. Une alternative est de constituer les bases de données à partir de vidéos TV afin de gagner en échelle. Nous proposons dans cet article une base de données de vidéos TV nommée STVD. Un protocole est introduit garantissant une capture à l'échelle et une génération robuste de vérité terrain. STVD est la base publique la plus importante de la littérature sur la tâche couvrant quatre-vingt seize mille vidéos d'une durée de plus de neuf mille heures.

Mots Clef

segments, détection, TV, base de données, évaluation

Abstract

This paper is interested with the performance evaluation of the partial video copy detection. Several public datasets exist designed from web videos. The detection problem is inherent to the continuous video broadcasting. The alternative is then to process with TV datasets offering a deeper scalability. We propose in this paper a TV dataset called STVD. It is designed with a protocol ensuring a scalable capture and robust groundtruthing. STVD is the largest public dataset on the task with $96k$ *videos of* $9,250$ *hours.*

Keywords

partial video copy, detection, TV, dataset, evaluation

1 Introduction

Au cours de la dernière décennie, l'utilisation de la télévision s'est fortement digitalisée. Ceci est lié d'une part à la démocratisation des équipements numériques (smart TVs, téléphones) et d'autre part à l'émergence des réseaux hauts débits. Ce contexte rend aujourd'hui possible le développement de nouveaux services numériques couvrant un large éventail d'applications comme l'analytique TV, le résumé automatique de vidéos pour le replay, la détection d'éléments pour la télévision augmentée.

Dans cet article, nous abordons le problème de la détection de segments de vidéos. Cette détection vise à trouver une vidéo requête courte apparaissant dans une vidéo plus longue sous contraintes de dégradation. Cela a de multiples applications pour la télévision comme le journalisme de données ou la détection de publicités [11]. La Figure 1 donne quelques exemples de contenu à la télévision pouvant faire l'objet d'une détection de segments.

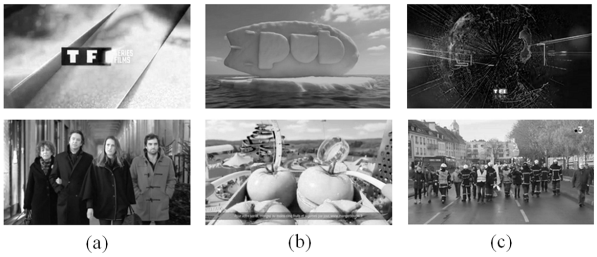


FIGURE 1 – Contenu répété à la télévision (a) générique de série (b) jingle et publicité (c) JT et reportage

Cet article est en lien avec la problématique d'évaluation de performance et les bases de données. Cette dernière est bien connue dans le domaine de la vision par ordinateur. De nombreuses bases de données pour la détection de segments ont été proposées majoritairement à partir de vidéos sur Internet [12, 7, 6]. La détection de segments est un problème inhérent à la diffusion en continu de flux vidéo. Une alternative est de constituer les bases de données à partir de vidéos TV. Cette démarche est en meilleure adéquation au problème et permet de gagner en échelle. Très peu de bases de données TV ont été proposées dans la littérature.

Nous proposons dans cet article une large base de données pour la détection de segments de vidéos TV nommée STVD (pour "large-Scale TV Dataset"). La section 2 donne un état de l'art. La section 3 présente notre protocole pour la capture et production de la vérité terrain. Nos expérimentations sont présentées dans la section 4 et la section 5 donne nos conclusions et perspectives. La Table 1 liste les principaux symboles utilisés dans l'article.

Symboles	Description
$\mathbf{h_1}, \ldots, \mathbf{h_q}$	liste de clés de hachage
$\mathbf{t}, \hat{\mathbf{t}}$	horodatage programmé et détecté d'un programme
$\mathbf{L} \in [\mathbf{L}_{\min}; \mathbf{L}_{\max}]$	$\mathbf{L} = \hat{\mathbf{t}} - \mathbf{t}$ est la latence, $\mathbf{L}_{\min} < \mathbf{0}$, $\mathbf{L}_{\max} > \mathbf{0}$ les bornes $\min$ et $\max$
$\mathbf{L}^- < 0, \mathbf{L}^+ > 0$	des valeurs de latence négative et positive
$\mathbf{W} = \mathbf{W}^- + \mathbf{W}^+$	la fenêtre de capture pour la détection à grosse maille
$\mathbf{D} \in [\mathbf{D}_{\min}; \mathbf{D}_{\max}]$	$\mathbf{D} > 0$ est une durée de programme, $\mathbf{D}_{\min}, \mathbf{D}_{\max}$ les durées $\min$ et $\max$
$\mathbf{T_0}, \ldots, \mathbf{T_6}$	les méthodes de dégradations et transformations
$\mathcal{S}$	une séquence extraite par $\mathbf{T_0}$ démarrant à $\mathbf{s}$ et terminant à $\mathbf{e}$ avec $\mathbf{s} = \mathbf{t} - \lvert\mathbf{L}^-\rvert$ et $\mathbf{e} = \mathbf{t} + \mathbf{D} + \mathbf{L}^+$
α, β	des paramètres pour le contrôle de la dégradation

TABLE 1 – Principaux symboles utilisés dans l'article

2 État de l'art

Deux bases de données TV ont été proposées dans la littérature pour la détection de segments de vidéos [2, 8]. Elles sont détaillées dans la Table 2. Ces bases fournissent les fichiers vidéo contenant les segments avec une vérité terrain associée. La vérité terrain labélise les vidéos et peut fournir des valeurs d'horodatage et métadonnées (titre, catégorie, résumé, etc.). En complément, des vidéos tests, sans corrélation avec les segments à retrouver, sont fournies pour les besoins d'évaluation de performance. Cela permet d'éprouver les systèmes contre les fausses détections. Il est d'usage de qualifier les vidéos segments et tests de cas Vrais Positifs (VP) et Vrais Négatifs (VN).

Base de données	**2007** [8]	**2014** [2]
Vidéos VP + VN	5 500	$>$ 20 000 000
Durée totale (h)	44 000 heures	380 000 heures
Métadonnées	aucune	peu
Résolution	352×288	320×240
Publique	non	non

TABLE 2 – Comparaison des bases de données TV

Un protocole doit être défini afin de garantir la fiabilité de la vérité terrain. Celui-ci doit prévenir de l'apparition de cas Faux Positifs (FP) et Faux Négatifs (FN) dans la base de données. Les FP correspondent à des vidéos labélisées présentant un contenu visuel différent des segments recherchés. Les FN sont des segments qui auraient été manqués par le protocole apparaissant dans les vidéos tests.

Le système [8] exploite deux sources de capture afin d'obtenir les VP et VN. Une capture longue est utilisée pour extraire aléatoirement des courtes séquences correspondant aux VP. Ces séquences sont ensuite altérées synthétiquement avec des méthodes de dégradation. Une capture courte, obtenue à partir d'une chaîne de télévision étrangère, est utilisée pour obtenir les VN. L'absence de cas FP, FN entre les deux sources est vérifiée par expertise métier. L'expertise métier ne peut garantir une complète fiabilité de la base de données. De plus, les segments sont extraits aléatoirement sans signification métier. Les multiple occurrences des segments ne sont pas détectées. La base de données est donc davantage dédiée à la reconnaissance de vidéos. Finalement, aucune métadonnée n'est fournie.

Une approche différente est utilisée dans [2]. Chaque jour une base d'empreintes (descripteurs temporels, de Fourier) est constituée à partir de la capture des chaînes. Cette base est auto-comparée pour détecter les segments et leurs occurrences correspondant aux VP. Pour répondre à la contrainte d'échelle, une unique occurrence VP est sélectionnée et comparée avec l'archive d'empreintes des jours antérieurs. Les intervalles où aucune occurrence n'est détectée servent pour constituer les VN. De cette manière, la vérité terrain est produite de manière incrémentale. Les métadonnées des vidéos sont renseignées manuellement par un utilisateur à partir d'une interface graphique.

Cette approche permet une grande variabilité dans la constitution des segments (générique, publicité, reportage, etc.). Cependant, elle est sujette à l'erreur compte-tenu de la stratégie de recherche pleine engagée. Finalement, uniquement un faible échantillon de métadonnées peut être obtenu avec l'interaction utilisateur.

Les protocoles proposés par [8, 2] présentent des limites. La fiabilité de la vérité terrain ne peut-être garantie et peu de métadonnées sont associées. Ces bases de données n'ont jamais été rendues publiques ni utilisées dans la littérature pour de l'évaluation de performance. Nous proposons dans cet article un nouveau protocole et une base de données TV présentés dans la section 3.

3 La base de données STVD

Nous présentons dans cette section notre protocole pour la construction de la base de données STVD. La Figure 2 détaille notre chaîne de traitement. Nous utilisons 5 composants (**C1**) à (**C5**). Nous procédons tout d'abord avec une capture de vidéo TV (**C1**). Ensuite, nous avons procédé de manière incrémentale pour la construction de la vérité terrain à l'image de [2]. Cependant, afin de minimiser les cas FP nous avons mis en place une détection à deux niveaux de granularité (**C2**) et (**C4**) où le premier niveau utilise les métadonnées de programmation connues a priori. Les vidéos VN sont obtenues par sélection aléatoire (**C3**) et utilisées pour paramétrer la détection à maille fine (**C4**). Pour le filtrage des FN, nous utilisons deux sources de capture comme dans le système [8]. Un dernier composant (**C5**) est utilisé pour la dégradation de vidéo. Les sections 3.1 à 3.5 détaillent nos différents composants.

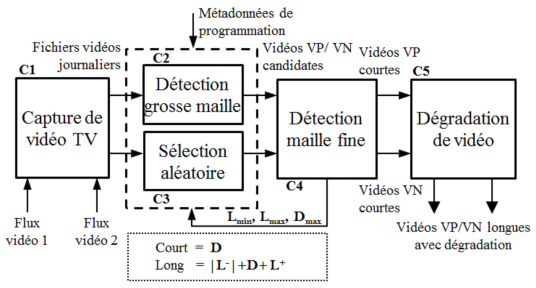


FIGURE 2 – Chaîne de traitement pour la base STVD

3.1 La capture vidéo (C1)

Un large volume de données vidéo TV est nécessaire pour la construction de la base STVD. Ceci nécessite l'utilisation d'une plateforme de capture dédiée. Nous avons utilisé la station TV [3, 10]. Cette station capture les vidéos à partir du signal TNT. Elle embarque des cartes de Avermedia CL332-HN [1] permettant la capture matérielle des flux vidéo. La station peut enregistrer jusqu'à 8 chaînes en simultané. Nous détaillons notre protocole dans la Table 3 et le paragraphe suivant.

1 mois						
FPS	**Résolution**	**kbps**	**Durée (h)**	**Chaînes**	**Fichiers**	**Taille (To)**
30	320×240	560	4 800 h	8	240	1, 23
3 mois au total			**14 400 h**	**24**	**720**	**3,69**

TABLE 3 – Protocole pour la capture TV

Nous avons paramétré la station pour l'enregistrement de 24 chaînes françaises non payantes durant une période de 3 mois. La capture a été ordonnancée en tourniquet (i.e. chaînes #1-8 sur le 1^{er} mois ...#17-24 sur le 3^{e} mois). Afin de minimiser le stockage, nous avons paramétré la capture à une résolution de 320×240 pixels et un débit de 560 kbps [1] pour la compression. La résolution 320×240 a été paramétrée comme suggéré en [2]. Le débit de 560 kbps a été arrêté sur la base des recommandations Avermedia [1]. Nous avons limité la capture à 20h par jour [2] de par la faible programmation nocturne. Les pistes audio ont été supprimées compte-tenu que les cartes Avermedia CL332-HN n'offrent pas de support matériel. La télévision française est délivrée à une cadence de 25 FPS en standard SECAM. Nous avons cependant paramétré la capture à une cadence 30 FPS en adéquation avec le standard NTSC et les vidéos en-ligne. Notre avons obtenu une base de données de 14 400 heures composée de 720 fichiers vidéo journaliers pour une taille de 3,69 To Table 3.

3.2 La détection à grosse maille (C2)

La détection incrémentale des VP requiert une recherche sur de longues séquences de capture [2]. Elle est sujette aux cas FP compte-tenu du problème d'échelle. Une programmation a priori peut être obtenue sur les programmes TV à partir des EPGs [3]. Cette programmation fournit des métadonnées sur les principaux programmes comme les dates de début et de fin, les titres et résumés. Elle peut être utilisée pour un premier niveau de détection des programmes.
Nous avons développé un robot d'indexation pour collecter quotidiennement des métadonnées de programmation à partir du site d'un magazine [4]. Nous avons généré des clés de hachage $\mathbf{h_1}, \ldots, \mathbf{h_q}$ pour chacun des programmes collectés. Ces clés ont été constituées à partir des noms de programmes et identifiants des chaînes. Un encodage SHA-3 512 bits a été appliqué afin de minimiser les collisions. Un programme répété est obtenu pour toute multiple occurrence d'une clé.
Les titres des programmes peuvent présenter des variations dans les EPGs. Ceci peut induire des erreurs à la génération des clés. Nous avons déployé un protocole pour correction. Un prétraitement du texte [5] a tout d'abord été appliqué. Nous avons ensuite inspecté visuellement les titres à l'aide d'heuristiques [6] pour une correction à la main. Finalement, uniquement les programmes journaliers et hebdomadaires ont été inspectés afin de borner le travail de correction.
Les métadonnées d'un programme indiquent les heures $\mathbf{t}$ de programmation. Cependant, aucune information n'est donnée sur les localisations des contenus répétés comme les génériques de début et de fin. De plus, une forte latence peut-être observée entre la programmation et la diffusion. Nous avons paramétré notre détection pour la capture des génériques d'introduction Figure 3. Afin de prendre en compte la latence, nous avons utilisé une fenêtre de dimension $\mathbf{W} = \mathbf{W}^- + \mathbf{W}^+$. Le paramètre $\mathbf{W}^-$ a été fixé pour garantir la latence minimale $\mathbf{W}^- \geq |\mathbf{L}_{\min}|$. $\mathbf{W}^+$ a été défini à partir de la durée et de la latence maximale $\mathbf{W}^+ \geq \mathbf{L}_{\max} + \mathbf{D}_{\max}$. Il est d'usage d'avoir $\mathbf{W}^+ \gg \mathbf{W}^-$.

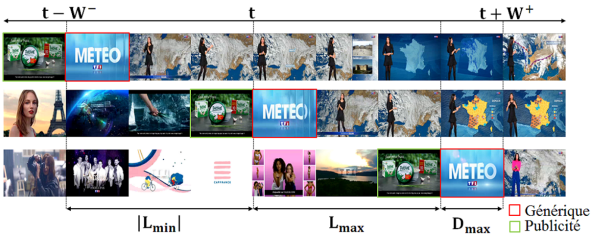


FIGURE 3 – La détection à grosse maille

La capture est donc effectuée sur l'intervalle $[\mathbf{t} - \mathbf{W}^-, \mathbf{t} + \mathbf{W}^+]$. Les valeurs $\mathbf{L}_{\min}, \mathbf{L}_{\max}$ et $\mathbf{D}_{\max}$ ont été établies en première instance par programmation expert Figure 2. Elles ont ensuite été ajustées à partir de la détection à maille fine (**C4**).

3.3 La sélection aléatoire (C3)

Le composant (**C2**) est en charge d'extraire les vidéos VP. Pour les vidéos VN, nous avons déployé un composant parallèle (**C3**) sur notre chaîne de traitement. Comme précisé en section 2 et illustré dans la Figure 2, nous avons utilisé deux flux de capture séparés 1, 2 pour obtenir les vidéos VP et VN. Ceci minimise la probabilité d'apparition de cas FN. Ensuite, nous avons adopté une stratégie de sélection en contenu en-dehors des plages de génériques. Nos deux étapes de sélection sont détaillées ci-dessous.
Étape 1 : nous avons extrait la liste des programmes du flux 2 à l'aide du composant (**C2**). Ensuite, nous avons rendu invalide pour la sélection toutes les séquences dans lesquelles un générique pouvait apparaître dans les in-

1. kilobits per second
2. De 6h du matin jour$_i$ à 2h du matin jour$_{i+1}$
3. Electronic Program Guide
4. https ://www.telerama.fr/
5. Passage en minuscule, suppression des caractères spéciaux, etc.
6. Recouvrement des plages horaires, similarité entre les titres

tervalles $[\mathbf{t} - \mathbf{W}^-, \mathbf{t} + \mathbf{W}^+]$. Durant cette étape, les séquences avec recouvrement ont été agrégées.

Étape 2 : les séquences valides ont été découpées en intervalles successifs de durée $\mathbf{W}$. À l'intérieur de tout intervalle valide, nous avons sélectionné à $\mathbf{t} = \mathbf{W}^-$ une vidéo VN candidate de durée aléatoire $\mathbf{D} \in [\mathbf{D}_{\min}; \mathbf{D}_{\max}]$.

3.4 La détection à maille fine (C4)

La détection (**C2**) fournit des séquences de capture ou des vidéos VP ont été identifiées à partir des métadonnées. Un second niveau de détection (**C4**) est nécessaire pour confirmer la présence des vidéos et les horodater. Une stratégie est de rechercher automatiquement tout contenu répété entre les séquences. Cependant, des contenus répétés sans lien avec les programmes peuvent apparaître comme les publicités Figure 3. En effet, la fenêtre de capture $\mathbf{W}$ est à large intervalle puisque les valeurs de latence $|\mathbf{L}_{\min}| + \mathbf{L}_{\max}$ sont beaucoup plus grandes que la durée maximale des génériques $\mathbf{D}_{\max}$.

Une connaissance métier est nécessaire pour ce second niveau de détection. Nous avons développé une interface graphique pour segmenter à la main une instance pour chacun des génériques. Comme pour la vérification des métadonnées, seuls les programmes journaliers et hebdomadaires ont été considérés requérant une faible interaction utilisateur. Les valeurs $\mathbf{D}_{\min}, \mathbf{D}_{\max}$ ont été obtenues à partir du générique le plus court et le plus long, respectivement.

Nous avons apparié chaque instance de générique aux séquences identifiées en (**C2**) en utilisant la mesure **ZNCC** (Zero-mean Normalized Cross-Correlation) Figure 4. Cette mesure est robuste au bruit, invariante au contraste et se prête bien au problème de détection [4, 10].

Les instances de génériques ont été appariées sur l'ensemble des frames pour un horodatage fin. L'appariement a été effectué à une résolution minimale 64×48 pixels pour compromis entre robustesse et optimisation. La détection a été calculée sur GPU avec une implémentation optimisée.

Le score d'appariement global $\overline{\mathbf{ZNCC}}$ a été obtenu par moyenne pondérée des mesures **ZNCC** de chacune des frames. Les poids ont été fixés à partir des mesures d'entropie des frames de référence. L'entropie se présente comme une bonne mesure de sélection [5] et estimateur de robustesse **ZNCC** [10] en absence de données d'apprentissage.

Le seuil de détection a été paramétré à partir de résultats d'appariements avec des vidéos VN. Pour ce faire, des vidéos VN ont été utilisées en sortie de (**C3**) vers (**C4**) Fig. 2. Nous avons obtenu une bonne séparabilité du problème avec cette approche comme détaillé en section 4. La différence entre la valeur d'horodatage détectée et programmée donne la latence $\mathbf{L} = \hat{\mathbf{t}} - \mathbf{t}$.

3.5 La dégradation de vidéo (C5)

Les vidéos VP/VN obtenues via les composants (**C1**) à (**C4**) correspondent à des captures en conditions réelles présentant un faible niveau de dégradation. Afin de stresser les systèmes et évaluer leur degré de robustesse, une

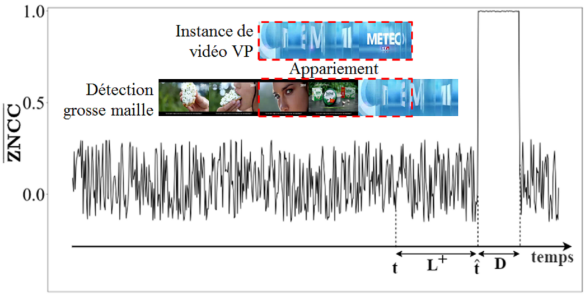


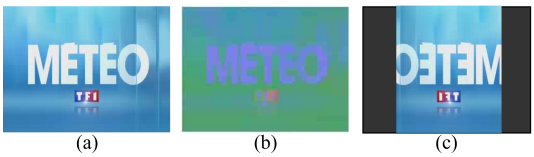
FIGURE 4 – Détection à maille fine

approche communément utilisée dans la littérature est l'application de bruit synthétique [2, 6, 8, 12]. Nous avons sélectionné des méthodes de dégradation représentatives détaillées dans la Table 4. Ces méthodes sont labélisées $\mathbf{T_0}$ à $\mathbf{T_6}$ et appliquées conjointement aux vidéos VP/VN.

Label	Méthode	Déscription
$\mathbf{T_0}$	extraction de séquence	utilise la distribution de latence pour extraire une séquence vidéo autour du segment de durée $\lvert\mathbf{L}^-\rvert + \mathbf{L}^+$
$\mathbf{T_1}$	codage des couleurs	sélectionne aléatoirement un espace couleur $\in$ {RGB ; YUV ; YIQ ; CIE} pour le codage [9]
$\mathbf{T_2}$	réduction d'échelle	applique une réduction d'échelle aléatoire $\alpha \in [0,1; 0,7]$ pour obtenir des fenêtres de dimensions 32×34 à 224×238 selon les critères [13]
$\mathbf{T_3}$	compression	applique un paramètre aléatoire $\beta \in [0,25; 0,75]$ aux débits kbps recommandés $\in \{35; 140; 320; 560\}$ à résolutions données [1]
$\mathbf{T_4}$	inverser	inverse aléatoirement (oui/non) la vidéo
$\mathbf{T_5}$	rotation	applique aléatoirement une rotation verticale / horizontale $\in \{0; \frac{\Pi}{2}; \Pi; \frac{3}{2}\Pi\}$ de la vidéo
$\mathbf{T_6}$	bords noirs ou étirement	sélectionne aléatoirement un paramètre $\frac{w}{h} \in \{0,46; 0,56; 0,63; 0,75; 1; 1,33; 1,6; 1,78; 2,17\}$ avec ($\frac{w}{h} < 1$) bord droit/gauche ($\frac{w}{h} > 1$) haut/bas **OU** étire l'image

TABLE 4 – Méthodes de dégradation de vidéos

Nous utilisons tout d'abord une méthode $\mathbf{T_0}$ pour obtenir les séquences contenant les segments. Les méthodes $\mathbf{T_{1,6}}$ se déclinent en deux catégories Figure 5 pour l'altération de pixels $\mathbf{T_{1,3}}$ (b) et les transformations globales $\mathbf{T_{4,6}}$ (c).



(a) (b) (c)

FIGURE 5 – dégradations (a) image de référence (b) altération de pixels (c) transformations globales

Dans la littérature, les vidéos sont dégradées avec une faible corrélation aux conditions réelles de capture [2, 6, 8]. Dans notre système, nous avons paramétré nos méthodes afin de conserver un caractère réaliste aux dégradations.

$\mathbf{T_0}$ est paramétrée à l'aide de la distribution de latence obtenue via (**C4**). Cette transformation extrait de longues séquences $\mathcal{S}$ englobant les vidéos VP et VN Fig. 6 (a). Considérant une vidéo VN horodatée à $\mathbf{t_i} = \mathbf{W}^-$ par (**C3**), ou un générique détecté à $\mathbf{t_i}$ par (**C4**), la séquence $\mathcal{S}_\mathbf{i}$ est extraite

à $\mathbf{s_i} = \mathbf{t_i} - |\mathbf{L}^-|$ et $\mathbf{e_i} = \mathbf{t_i} + \mathbf{D_i} + \mathbf{L}^+$ avec $|\mathbf{L}^-|, \mathbf{L}^+$ des valeurs de latence negative et positive, respectivement.

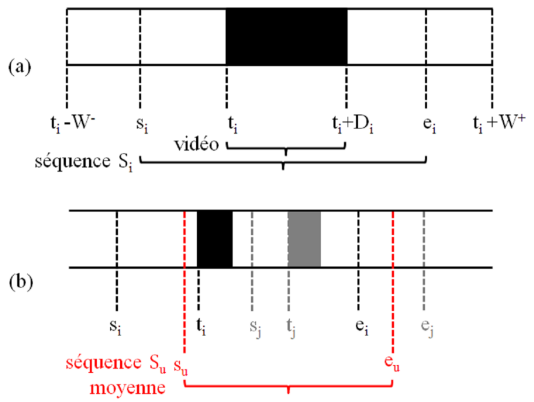


FIGURE 6 – (a) séquence $\mathcal{S}$ (b) cas du recouvrement

Une séquence $\mathcal{S}$ est extraite pour chacune des vidéos VN. La selection des vidéos à $\mathbf{t} = \mathbf{W}^-$ au sein d'intervalles de durée $\mathbf{W}$ par (**C3**) garantit le non recouvrement. Des recouvrements peuvent cependant apparaître pour les vidéos VP Fig. 6 (b). Considérant deux génériques successifs programmés à $\mathbf{t_i}, \mathbf{t_j}$ il est courant d'avoir $\mathbf{t_j} - \mathbf{t_i} \ll \mathbf{W}^+$. Dans ce cas, une séquence $\mathcal{S}_\mathbf{u}$ moyenne est établie et conservée si $\forall \mathcal{S}_\mathbf{i} \in \mathcal{S}_\mathbf{u}, \mathbf{s_u} < \mathbf{t_i} + \mathbf{D_i} < \mathbf{e_u}$.

La méthode $\mathbf{T_1}$ utilise des codages vidéo standard [9]. Les bornes du paramètre α dans $\mathbf{T_2}$ ont été arrêtées sur la base des recommandations [13] pour le traitement vidéo robuste à basse résolution et sous contrainte d'optimisation. $\mathbf{T_3}$ applique un paramètre de dégradation borné β aux débits kbps recommandés à résolutions données [1].

$\mathbf{T_4}$ et $\mathbf{T_5}$ appliquent des transformations géométriques réalistes comme l'étirement et les rotations verticales / horizontales. Les valeurs pour le paramètre $\frac{w}{h}$ dans $\mathbf{T_6}$ ont été définies à partir de résolutions standard [7].

Nous avons combiné les méthodes de dégradation $\mathbf{T_0}$ à $\mathbf{T_6}$ afin de générer différents jeux de test A à E détaillés dans la Table 5. Le jeu de test A donne une capture brute sans dégradation pour la détection de segments. Les jeux B et C altèrent les pixels à l'aide des méthodes $\mathbf{T_1}$ à $\mathbf{T_3}$ à deux niveaux de dégradation contrôlés par les paramètres α, β. Le jeu de test D est attrait aux transformations globales via les méthodes $\mathbf{T_4}$ à $\mathbf{T_6}$. Finalement, le jeu de test E combine toutes les degradations $\mathbf{T_0}$ à $\mathbf{T_6}$.

Jeu	$\mathbf{T_0}$	$\mathbf{T_{1-3}}$	$\alpha \in$	$\beta \in$	$\mathbf{T_{4-6}}$
Jeu A	√				
Jeu B	√	√	$[0,4;0,7]$	$[0,5;0,75]$	
Jeu C	√	√	$[0,1;0,4]$	$[0,25;0,5]$	
Jeu D	√				√
Jeu E	√	√	$[0,1;0,4]$	$[0,25;0,5]$	√

TABLE 5 – Jeux de données

7. Ordinateur fixe, tablette et téléphone https ://gs.statcounter.com/

4 Expérimentations et résultats

Nous présentons dans cette section nos expérimentations pour la génération de la base de données STVD. Ces expérimentations ont été conduites par déploiement de notre protocole détaillé en section 3 et Figure 2. La Table 6 donne l'organisation de la base. Nous avons séparé notre base de capture, obtenue par le composant (**C1**), en deux bases de 4 800 et 9 600 heures. Ces bases ont été constituées à partir de la capture de 8 et 16 chaînes sans recouvrement. Nous avons ensuite traité ces deux bases à l'aide de nos composants (**C2**) à (**C5**) pour obtenir les vidéos VP et VN.

	C1	**C2, C3, C4**		**C5**	
	Durée	**Vidéos**	**Durée**	**Vidéos**	**Durée**
VP	4 800 h	3 675	1, 9 h	13 425	1 584 h
VN	9 600 h	16 455	9, 7 h	82 275	7 666 h

TABLE 6 – Organisation de la base STVD

Pour les vidéos VP, nous avons tout d'abord analysé avec le composant (**C2**) le nombre de clés de hachage $\mathbf{h_1}, \ldots, \mathbf{h_q}$ et leurs occurrences Figure 7 (a). Nous avons obtenu un millier de clés environ avec des occurrences de 1 à 175 pour un volume total de 7 310 vidéos. Nous avons ensuite filtré les programmes journaliers et hebdomadaires Figure 7 (b) pour des questions de fiabilité [8]. Nous avons obtenu 405 programmes à partir desquels nous avons sélectionné les 255 plus occurrents pour l'interaction utilisateur[8]. Ces programmes couvrent $\simeq 80\%$ des cas VP pour un nombre total de 5 730 vidéos et 255 clés.

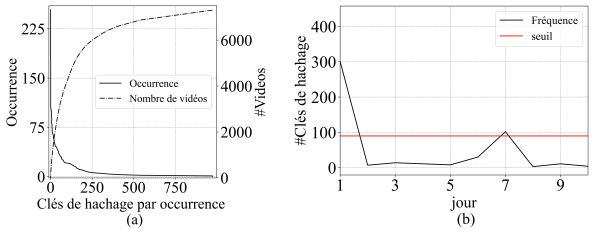


FIGURE 7 – (a) clés de hachage (b) fréquence programmes

Nous avons segmenté à la main, pour chacune des 255 clés, un générique pour l'appariement à maille fine (**C4**). Ces génériques ont été comparés aux séquences identifiées par la détection à grosse maille (**C2**). Nous avons obtenu une distribution intra-classe compacte $\overline{\mathbf{ZNCC}} \in [0,95;1]$ caractérisant une faible dégradation à la capture Figure 8 (a). Cet appariement a aussi permis de filtrer les génériques alternatifs (i.e. avec un contenu visuel différent). Nous avons retenu 3 675 vidéos sur les 5 730 pour les 255 clés.

Les résultats d'appariement ont permis ensuite de fixer les paramètres de durée $\mathbf{D}$ et de latence $\mathbf{L}$. Nous avons obtenu une durée $\mathbf{D} \in [1;5]$ secondes par générique pour une durée totale de 1, 9 heures pour les 3 675 vidéos candidates. Nous avons observé une distribution gaussienne de la latence avec $\mathbf{L} \in [-500;700]$ secondes Figure 8 (b). Cette

8. Comme discuté en sections 3.2 et 3.4.

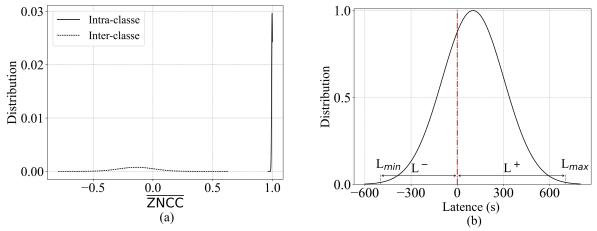


FIGURE 8 – (a) distribution $\overline{\mathbf{ZNCC}}$ (b) latence

distribution a été utilisée pour paramétrer le générateur de nombre aléatoire dans $\mathbf{T_0}$ pour obtenir les séquences $\mathcal{S}$ Figure 6. Nous avons obtenu 2 685 séquences pour les vidéos VP suite à la détection des recouvrements.
Pour les vidéos VN, nous avons rendu invalide pour la sélection toutes les séquences où un générique pouvait apparaître comme détaillé dans la section 3.3. Nous avons obtenu 16 455 séquences valides. Les vidéos VN ont été sélectionnées au sein de chacune des séquences à $\mathbf{t} = \mathbf{W}^-$ avec une durée $\mathbf{D}$ aléatoire pour une durée de $9,7$ heures. Une faible fraction de ces séquences $\simeq 100$ a été comparée aux 255 génériques à l'aide du composant (**C4**) afin de fixer le seuil de détection des VP. Nous avons observé aucun appariement pour $\overline{\mathbf{ZNCC}} \in]0,6; 0,95[$ garantissant une bonne robustesse pour le filtrage des FP Figure 8 (a).
Dans une dernière étape, nous avons utilisé le composant (**C5**) pour obtenir les 5 jeux de test A à E détaillés dans la section 3.5. Nous avons obtenu quatre-vingt seize mille vidéos environ composées de $5 \times 2\,685$ et $5 \times 16\,455$ vidéos VP et VN, respectivement. Considérant la distribution de latence Figure 8 (b), la transformation $\mathbf{T_0}$ appliquée pour chacune des vidéos s'est traduite par une durée moyenne $|\mathbf{L}^-| + \mathbf{L}^+$ de $5,6$ minutes. Nous avons obtenu une durée totale de plus de neuf mille heures pour la base avec 1 584 et 7 666 heures de vidéos VP et VN Table. 6. STVD est aujourd'hui la base publique la plus importante de la littérature sur la tâche de détection de segments. Elle est en moyenne quatre fois plus importante que les plus grandes bases constituées à partir de vidéos en ligne [6, 7]. La base a été rendue publique en ligne [9] pour les besoins recherche.

5 Conclusions et perspectives

Nous avons proposé dans cet article une base de données TV, nommée STVD, pour l'évaluation de la détection de segments de vidéos. Un protocole est introduit garantissant une capture à l'échelle et une génération robuste de vérité terrain. STVD est la base publique la plus importante de la littérature sur la tâche couvrant quatre-vingt seize mille vidéos d'une durée de plus de neuf mille heures.

Références

[1] AVerMedia. Avermedia capture card software developement kit. Technical Report 4.2, AVerMedia Technologies, Inc. www.avermedia.com, 2015.

9. http://mathieu.delalandre.free.fr/projects/stvd/index.html

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

[3] M. Delalandre. A workstation for real-time processing of multi-channel tv. In *International Workshop on AI for Smart TV Content Production (AI4TV)*, pages 53–54, 2019.

[4] Z.J. Guzman-Zavaleta and C. Feregrino Uribe. Partial-copy detection of non-simulated videos using learning at decision level. *Multim. Tools Appl*, 78(2) :2427–2446, 2019.

[5] Y. Hou, X. Wang, S. Liu, and Y. Zhang. Video copy detection based on uniform local binary pattern. In *DEStech Transactions on Computer Science and Engineering*, 2018.

[6] Q.Y. Jiang, Y. He, G. Li, J. Lin, L. Li, and W.J. Li. Svd : A large-scale short video dataset for near-duplicate video retrieval. In *International Conference on Computer Vision (ICCV)*, pages 5281–5289, 2019.

[7] Y.G. Jiang and J. Wang. Partial copy detection in videos : A benchmark and an evaluation of popular methods. *IEEE Transactions on Big Data*, 2(1) :32–42, 2016.

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

[9] Y.S. Kahu, B.R. Raut, and K.M. Bhurchandi. Review and evaluation of color spaces for image/video compression. *Color Research and Application*, 44 :8–33, 2019.

[10] V.H. Le, M. Delalandre, and D. Conte. Real-time detection of partial video copy on tv workstation. In *International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–4, 2021.

[11] M. Li, Y. Guo, and Y. Chen. Cnn-based commercial detection in tv broadcasting. In *International Conference on Network, Communication and Computing (ICNCC)*, pages 48–53, 2017.

[12] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, W. Kraaij, A. Smeaton, and G. Quénot. Trecvid 2010 - an overview of the goals, tasks, data, evaluation mechanisms and metrics. NIST, https ://www.nist.gov/, 2010.

[13] J. Su, D.V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *Transactions on Evolutionary Computation (TEVC)*, 23(5) :828–841, 2019.

Remerciements

Les auteurs souhaitent remercier le laboratoire LIFAT et l'école d'ingénieur Polytech Tours (départements informatique et informatique industrielle) pour le financement du projet station TV.