# Labellisation du Comportement de Descripteurs Locaux pour la Détection de Copies Vidéo

J. Law-To[1 2] V. Gouet-Brunet[2] O. Buisson[1] N. Boujemaa [2]

[1] INA Direction de la Recherche et Expérimentation, Bry Sur Marne
[2] INRIA Rocquencourt Equipe IMEDIA, Rocquencourt

{jlawto@ina.fr, valerie.gouet@inria.fr, obuisson@ina.fr, nozha.boujemaa@inria.fr}

## Résumé

*Ce papier présente une approche efficace d'indexation et de recherche dans de grandes bases de vidéos. Cette indexation automatique exploite un ensemble de descripteurs locaux et leurs trajectoires à travers la séquence vidéo. Cette méthode permet d'une part de réduire la redondance temporelle intrinsèquement liée à la vidéo et d'ajouter d'autre part un contexte de comportement à ces descripteurs. Ainsi, en partant d'une description bas-niveau du signal, notre approche permet d'aboutir à une représentation de plus haut niveau, associant une tendance de comportement aux descripteurs locaux. La description obtenue est d'une part plus compacte, non redondante et d'autre part peut être rendue spécifique à la vidéo en fonction de l'application de recherche désirée. Une application cruciale dans la gestion de patrimoines numériques est la traçabilité du catalogue vidéo et nous proposons dans cet article un système de détection de copie par le contenu et son évalution. L'évaluation montre une nette amélioration des performances face à une technique état de l'art tout en présentant une meilleure flexibilité. Elle est de plus temps réel sur une base vidéo importante (plusieurs centaines d'heures).*

## Mots clefs

Indexation vidéo par le contenu, détection de copies vidéos, descripteurs locaux, trajectoires.

# 1 Introduction

La croissance des contenus audiovisuels, et en particulier vidéo nécessite la création d'outils de recherche par similarité ou copies. La traçabilité des contenus audiovisuels est une nécessité pour les professionnels des archives et les détenteurs de droits vidéo. La détection de copie par le contenu (en anglais *Content Based Copy Detection, CBCD)* est une alternative au tatouage d'images pour tracer un fond d'archives vidéo. Les méthodes de recherche par le contenu et en particulier sur la détection de copies dans les fonds vidéo consistent généralement à extraire des éléments caractéristiques de la vidéo appelées signatures et à les comparer à une base. Plusieurs approches existent dans la littérature : dans [1, 2], les auteurs utilisent des signatures temporelles alors que dans [3], les auteurs comparent des méthodes basées sur des descriptions globales au niveau image (couleurs) et temporelles (mouvement, distribution de l'intensité). Ces descriptions globales (temporelles ou spatiales) ont l'avantage de caractériser les séquences de manière légère (1 vecteur par trame) mais sont peu robustes et peu discriminantes. En effet, la notion de copie dépasse la réplique exacte et inclu tout nouveau montage issus d'une vidéo. Les modifications sont de types divers (changement de la luminance, insertion d'éléments, décalage de l'image, remontage, etc). Ainsi la rechercher de copie apparait comme un sous ensemble du vaste dommaine de la recherche par similarité. La figure 1 illustre cette idée : elle montre l'exemple de deux vidéos beaucoup plus similaires en terme de contenu qu'une vidéo et sa copie.


Similaires mais non copies (les cravates sont différentes)


Copies (l'une est faite à partir de l'autre)
Source video : *Gala du Midem.* G. Ulmer 1970 (c) Ina

Figure 1 – *Copie / similarité.*

Ces contraintes nous ont orientés vers une *description locale* de la vidéo et donc vers les points d'intérêt. L'utilisation de signatures basées sur ceux ci a prouvé son efficacité

pour retrouver des images [4] ou des vidéos [5].
Le concept que nous proposons est basé sur l'estimation et la caractérisation de trajectoires de points d'intérêt le long de la séquence vidéo. Il présente deux avantages : tout d'abord, la redondance temporelle des descriptions locales, intrinsèquement liée à la vidéo, est éliminée avec une perte d'information réduite, comme cela a déjà été fait notamment dans [6, 7]. Dans un deuxième temps, l'analyse de ces trajectoires fait ressortir des tendances de comportements locaux et permet donc d'enrichir chaque descripteur local en lui ajoutant une information sur le comportement spatial et temporel du point. Cette description permet d'assigner des labels de comportements aux descripteurs locaux. L'objectif est d'obtenir une description de la vidéo par le contenu plus *riche*, plus *compacte* tout en restant *générique*. De tels labels, associés à des tendances de comportement des points, peuvent être interprétés comme un contexte cinématique associé aux descripteurs locaux. La notion de contexte associé à une description locale a été récemment proposé pour les images fixes, en ajoutant une information semi-globale autour du point [8] ou caractérisant les relations spatiales entre voisins [9].

# 2 Description bas-niveau par descripteurs locaux

Cette section présente la description bas niveau des séquences vidéos que nous avons choisies. Elle se fait en deux étapes : l'extraction de la description locale sur chaque trame et le suivi de cette description le long de la séquence vidéo. Les techniques présentées ici sont classiques et ne représentent pas la contribution majeure de notre travail.

**Descripteurs locaux.** Les points d'intérêt ont d'abord été développés pour la Vision par Ordinateur puis pour la recherche d'images par le contenu. De nombreuses approches de détection de points et de description locale ont été proposées. Le lecteur peut trouver une évaluation des méthodes les plus connues dans [10]. Les points d'intérêt ont été étendus au niveau spatio-temporel [11].
Les points d'intérêt sont pertinents pour une recherche précise et locale dans l'image comme des détails ou des objets. Associés à un vote spécifique, ils sont robustes aux occultations, aux décalages et à certaines transformations géométriques et par conséquent sont pertinents pour la détection de copies. Pour évaluer notre algorithme, nous avons utilisé le détecteur de Harris [12] associé à une description locale classique (jet local) sur quatre positions autour du point détecté $\vec{s_i} = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial^2 I}{\partial x \partial y}, \frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial y^2}\right)$. Nous obtenons donc une signature par points de dimension 20. Cette description est ensuite normée pour être invariante à un changement affine de luminance. Cet espace de description est designé $S_{Harris}$. Nous n'avons pas utilisé la populaire approche SIFT [13], car elle implique un espace de signature trop important (128 dimensions par points) ce qui appliqué à la vidéo devient très vite problématique (une heure de vidéo représente $25 * 3600$ image aboutissant à $3 \times 10^6$ descripteurs). Nous n'utilisons pas ce type de points car ils ne permettent pas de décrire certaines informations de contexte temporel pourtant pertinentes, comme par exemple les points de décor qui décrivent une information non traitée par les points d'intérêt spatio-temporels.

**Construction des trajectoires.** Les trajectoires sont ensuite construites sur le même principe que le classique algorithme KLT [14] : on apparie les descriptions locales de trame en trame. La différence est que nous effectuons cette mise en correspondance sur les 15 trames précédentes et suivantes, de manière à être robuste à d'éventuelles ruptures de trajectoires, non négligeables dans les vidéos où la qualité d'image est plutôt faible. A noter que cette méthode de suivi est générique et pourrait être appliquée à tout autre type de descripteurs.

# 3 Vers les labels de comportement

Cette section décrit la manière d'aboutir à une description de plus haut niveau, comportant donc plus de sémantique que la simple description bas-niveau précédente.

## 3.1 Description du signal sur une trajectoire

A chaque trajectoire de points, on souhaite associer une description du signal que l'on qualifie de description bas niveau. Pour cette description, nous calculons la moyenne de chaque composante des descriptions locales des points de la trajectoire. La description obtenue est notée $\vec{S}_{mean}$. Lors de la construction des trajectoires, l'appariement étant fait de proche en proche, la valeur du descripteur local peut théoriquement dériver largement. Pour vérifier la pertinence de la description choisie, plusieurs valeurs de $\vec{S}_{mean}$ ont été analysées sur une séquence vidéo de 1 heure : pour chaque trajectoire, nous avons calculé la distance de chacun des points avec la valeur de $\vec{S}_{mean}$ correspondant. Dans 95 % des cas, cette distance est plus petite que le seuil utilisé dans l'étape d'appariement de la construction des trajectoires. Cette expérience confirme que $\vec{S}_{mean}$ caractérise bien le signal le long d'une trajectoire. Une approche similaire est décrite dans [7] : les auteurs observent que sur la trajectoire construite à partir d'un changement d'angle de vue progressif, la description SIFT a une variation quadratique ; les auteurs prennent également la moyenne de la description comme description finale. Dans la suite du papier, nous appelons cet espace de description $S_{Signal}$ ; il est de même nature que $S_{Harris}$ (même dimension, même type de distribution).

## 3.2 Description cinématique des trajectoires

Une description de plus haut niveau peut être obtenue en associant un contexte spatio-temporel au descripteur $\vec{S}_{mean}$. Ce contexte est obtenu en récupérant les propriétés des trajectoires, qui sont de nature spatiale et temporelle, donnant une information cinématique sur le comportement du point d'intérêt le long de sa trajectoire. Nous considérons les propriétés suivantes, calculées pour chaque trajec-

toire et stockées durant la partie d'indexation hors-ligne :
– Time code de début et de fin : $[tc_{in}, tc_{out}]$ ;
– Variation spatiale : $[x^{min}, x^{max}], [y^{min}, y^{max}]$.
Cette espace de description est noté $S_{Traj}$ par la suite. L'association de $S_{Signal}$ et $S_{Traj}$ permet d'enrichir la description de la vidéo, qui reste générique.

### 3.3 Definition des labels

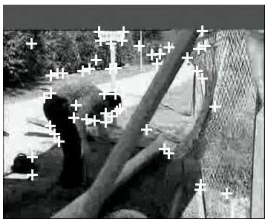
A partir des propriétés définies plus haut, il est possible de déterminer des tendances de comportements. Considérons par exemple les points ayant les caractéristiques suivantes :
– En mouvement / immobile ;
– Persistant / rare (persistance = 1) ;
– Mouvement rapide / lent ;
– Mouvement horizontal / vertical.
Cette liste ne représente que quelques exemples de labels que l'on peut attribuer. En classant les descripteurs locaux en fonction de leur comportement, il devient donc possible d'étiqueter chaque descripteur de $S_{Signal}$. Pour l'instant, les labels et catégories de comportement sont obtenus par simples seuillages globaux.
Cette annotation constitue une description de *haut niveau* car elle implique une interprétation de la vidéo : le choix d'un label plutôt qu'un autre est *spécifique* au contenu de la vidéo. Le potentiel des labels obtenus est multiple : dans un premier temps, il vont servir à sélectionner des sous-ensembles de trajectoires pertinents afin de réduire de manière efficace l'espace de description et dans un deuxième temps, ils vont être exploités spécifiquement dans une fonction de vote pour améliorer la recherche.
Dans ce travail, nous nous attachons à la détection de copie et les labels que nous allons considérer sont : les points immobiles et persistants qui définissent un label *Décor* et les points persistants et en mouvement qui définissent le label *Mouvement*. Les points de décor apportent de la *robustesse* à la description tandis que les points en mouvement sont propres à la vidéo et donc très *discriminants*. Dans la section suivante, nous détaillons comment ces différents espaces de description venant d'être présentés vont être exploités dans un algorithme de recherche de copie vidéo.

## 4 Algorithme de recherche

Cette section présente la méthode de recherche de vidéos à partir de l'indexation décrite précédemment. Le cas particulier de la détection de copie est développé ici.

### 4.1 Une technique asymétrique

Contrairement à la plupart des méthodes de recherche aussi bien images que vidéos, nous n'effectuons pas les mêmes opérations sur les vidéos requêtes (VR) que sur les vidéos sources (VS). Le calcul des trajectoires n'est pas appliqué à la VR pour les raisons suivantes : la première est d'ordre pratique, en effet l'indexation hors ligne des VS nécessite un long temps de calcul (voir section 5.3 pour des ordres de grandeur) alors qu'un système de détection de copies pour assurer la traçabilité d'archives audiovisuelles doit être au minimum temps réel (les VR étant constituées du flux de toutes les chaines de TV). Ce constat pénalise d'ailleurs la plupart des méthodes utilisant des descriptions par tubes spatio-temporels (voir [15] par exemple) et si de plus on souhaite être efficace sur des volumes conséquents, on ne peut se permettre les couts de calculs de l'indexation hors ligne sur les VR. Une deuxième raison plus fondamentale est que l'on veut être robuste au remontage, à l'utilisation d'extraits et dans ces cas,les trajectoires peuvent être tronquées. Les VR sont donc échantillonées dans un espace de description similaire à $S_{Harris}$ selon 2 paramètres :
– la periode $p$ du choix de d'image extraite du flux ;
– le nombre $n$ de points de Harris choisis.
Pour l'instant $p$ et $n$ sont constants et fixés par un opérateur mais on peut imaginer par la suite un choix dynamique de ces valeurs. L'avantage de la méthode asymétrique est qu'elle est rapide mais surtout, elle permet un choix en ligne de la qualité et de la granularité temporelle des détections. Le principal challenge de cette méthode est que l'on a d'un cotés des points de Harris (VR) dans l'espace de description $S_{Harris}$ et de l'autres des trajectoires (VS) avec les espaces $S_{Signal}$ et $S_{Traj}$. La figure 2 l'illustre bien : sur l'image de gauche, les croix représentent les requêtes tandis que les propriétés des trajectoires associées à la vidéo source sont représentées sur l'image de droite.



Vidéo Requête
Les + representent les sous-requêtes.
(Points d'interêt)

Vidéo Source
Les boites representent $S_{Traj}$
$[x^{min}, x^{max}], [y^{min}, y^{max}]$

Figure 2 – *Illustration des espaces de descriptions utilisés pour la méthode asymétrique.*

### 4.2 Recherche spécifique pour la détection de copies

Nous détaillons ici les différentes étapes de notre algorithme de recherche basé sur la méthode de description présentée. Dans ce travail, l'algorithme est dédié à la détection de copies, mais il peut s'appliquer à d'autres applications de recherche par contenu dans les vidéos, en choisissant les labels appropriés.

**Recherche bas-niveau des plus proches voisins.** La vidéo requête a donc été echantillonée en K sous-requêtes ayant chacune une description dans $S_{Harris}$, un time code $tc_k$ et une position $(x_k, y_k)$ ($k \in [1, K]$). La première étape de la recherche permet de ramener les plus proches voisins dans un rayon donné de l'espace de description $S_{Signal}$.

Chaque voisin ramené est une trajectoire et possède donc en plus de sa description dans $S_{Signal}$, une description dans $S_{Traj}$ ainsi qu'un ou plusieurs labels de comportement. Cette étape utilise un algorithme de recherche approximative probabiliste non détaillé ici (voir [5]). A partir de ces voisins, une recherche spécifique en fonction de la robustesse nécessaire et de la similarité désirée sera effectuée. Les choix faits pour la détection de copies sont ici présentés.

**Choix des labels.** Pour la détection de copies, nous considérons les deux types de comportements des points introduits dans la section 3.3 pour leur pertinence : les points labellisés *Décor* et les points labellisés *Mouvement*. Ce choix a tout d'abord pour conséquence de réduire le volume de l'espace de description des points ; dans un deuxième temps, ces deux types de points vont permettre un recalage des sous-requêtes dans les trajectoires, comme décrit ci-après.

**Recalage spatio-temporel.** Un vote par comptage simple des candidats ramenés précédemment n'est pas assez discriminant pour la détection de copie. Le vote que nous avons développé va tenter de recaler l'ensemble des sous requêtes sur les voisins ramenés qui sont des trajectoires. Le recalage spatio-temporel consiste à évaluer le décalage temporel et spatial entre la vidéo dans la base de données et le flux requête. Pour cela, on utilise la description de l'espace $S_{Traj}$. Les détails de ce recalage ne sont pas explicités ici car ce n'est pas l'objet du papier mais le principe est de faire un premier recalage par image et par label de comportement puis de fusionner ce recalage par image pour enfin propager dans le temps le décalage estimé. Chaque recalage se fait par comptage du nombre de requêtes compatibles à un décalage donné. Ce vote donne une grande robustesse au système : robustesse au décalage,à l'insertion de cadre, au remontage (en utilisant des extraits). La détection de copie par le contenu utilisant ce vote spécifique est évaluée dans la section suivante.

# 5 Evaluation pour le CBCD

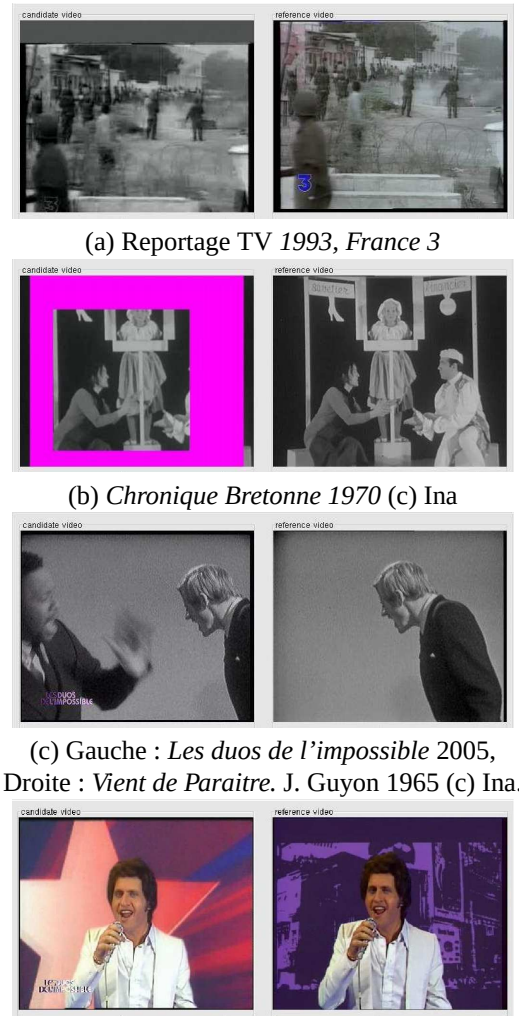
Cette section présente notre méthode d'évaluation sur des cas simulés et des cas réels, et donne des indices de performance comparés à une méthode de l'état de l'art.

## 5.1 Cadre de l'évaluation

**Base vidéo.** Toutes les expériences sont réalisées sur une base de données vidéo réelles de 320 heures : 300 heures aléatoirement choisies dans les archives de l'INA [1] et 20 heures correspondant aux vidéos nécessaires à la dernière expérimentation (voir 5.4). Ces vidéos encodées en *MPEG-1* (25 im/s) présentent toute sorte de contenus (journaux TV, sports, émissions de variétés etc...) de différentes époques (émissions couleurs ou noir et blanc).

**Définition des transformations.** Afin de tester la robustesse du système, nous avons défini un certain nombre

[1] Institut Nationnal de l'Audiovisuel




(a) Reportage TV *1993, France 3*

(b) *Chronique Bretonne 1970* (c) Ina

(c) Gauche : *Les duos de l'impossible* 2005, Droite : *Vient de Paraitre.* J. Guyon 1965 (c) Ina.

(d) Gauche : *Les duos de l'impossible* 2005, Droite : *Système deux.* C. Fayard 1975 (c) Ina.

Figure 3 – *Exemples de détection de copies. A gauche, les Vidéos Requêtes (vidéos avec des transformations aléatoires ou émission tv). A droite : vidéo de la base (VS).*

de transformations potentielles voulues (translations, insertions, recadrage, modification du gamma) ou non voulues (bruits, dégradations colorimétriques) comme on peut le voir sur les images (a) et (b) de la figure 3. La robustesse du système sera testée en prenant des vidéos de la base et en les transformant artificiellement.

**Technique de référence.** Afin d'évaluer notre technique de détection de copie, nous avons besoin d'une référence. Nous avons choisi de comparer notre méthode à celle décrite dans [5]. Les auteurs utilisent aussi des descripteurs locaux et obtiennent de très bons résultats, même sur des grandes bases de vidéos (10 000 heures). Il y a cependant deux différences fondamentales : nous avons indexé toutes les images sans limitation du nombre de descripteurs à priori, tandis que la technique de référence indexe uniquement 20 points sur des images clefs. La deuxième diffé-

rence est que l'on ajoute un contexte de comportement des points à notre description. Les techniques utilisant des descripteurs globaux ne nous semblent pas assez robustes à de grandes transformations comme les insertions. Par exemple dans [3], les performances sont moindre surtout pour des extraits courts alors que la base de donnée est très petite. Les auteurs de [5] nous ont confié leur code pour effectuer cette comparaison dans les mêmes conditions de tests.

## 5.2 Evaluation sur un jeu de test

Notre méthode est tout d'abord testée sur deux jeux de test : *Bench1min* et *Bench30* construits comme représenté sur la figure 4. Nous avons selectionnés aléatoirement 40 extraits vidéo de notre base puis nous les avons transformés artificiellement en utilisant des paramètres aléatoires. Ces segments ont une durée de 1 minute pour *Bench1min* et une durée aléatoire comprise entre 5 images et 30 secondes pour *Bench30*. Ces segments sont ensuite insérés au hasard dans un flux de 7 heures de flux vidéos de différentes chaines de TV.

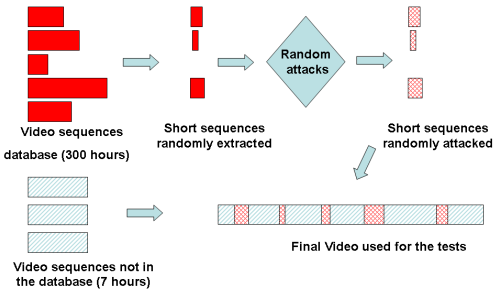



Figure 4 – *Construction du jeu de test.*

Ces expériences permettent d'évaluer le système dans une situation "réelle" simulée : des segments vidéos attaqués sont inclus dans un flux important de vidéos. Le but étant de détecter ces segments et de les retrouver le plus préciément possible. Nous avons utilisé deux jeux de paramètres requêtes :

– $p = 30$ and $n = 20$ pour avoir le même nombre de sous-requêtes que la référence,
– $p = 15$ and $n = 50$ pour tester l'amélioration possible.

Les figures 5 et 6 présentent les courbes précision/rappel pour les deux jeux de tests. Le second test est plus difficile car il met en jeu des segments très courts (moins de 1s pour certains). Plusieurs remarques peuvent être faites :

– Sur le test *Bench1min*, les techniques sont très performantes avec un rappel supérieur à 90% pour une précision de 95%. En augmentant le nombre de requêtes, on retrouve toutes les séquences (100% comparé à 97% pour la référence).
– Pour le test *bench30*, la chute du rappel apparait à une précision de 52% pour la référence tandis qu'elle apparait à 64% pour notre technique avec le même nombre de requêtes.

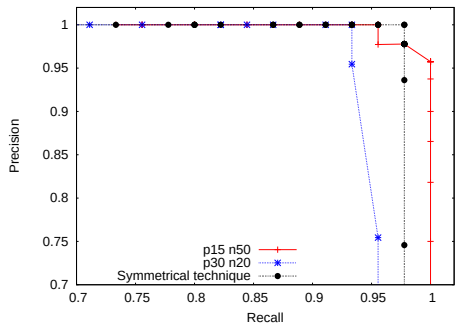



Figure 5 – *Precision/rappel pour Bench1min.*

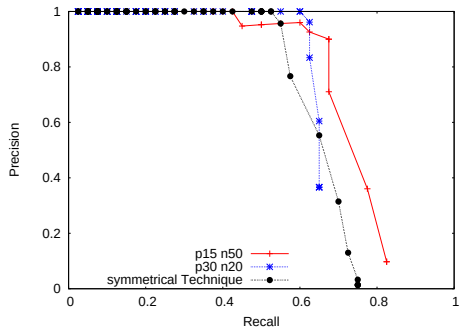



Figure 6 – *Precision/rappel pour bench30.*

– L'augmentation du nombre de requêtes permet d'augmenter le rappel mais cause parfois une baisse de la précision du fait de l'apparition de certaines fausses alarmes : la précision chute plus tot (44% du rappel).

En conclusion pour ces 2 tests, le rappel pour une précision acceptable (supérieure à 90%), est meilleur pour notre technique et d'autant meilleur que les segments à retrouver sont courts (100% comparé à 97% et 71% comparé à 55%) ce qui est particulièrement intéressant pour l'utilisation d'images d'archives dans des reportages par exemple.

## 5.3 Temps de calculs

Le fait de travailler sur de gros volumes de vidéos pose le problème des temps de calculs. Le système final doit être temps réel car dans la pratique les vidéos requêtes sont infinies (flux TV 24h/24). Nous donnons ici les valeurs mesurées sur l'expérimentation précédente en utilisant la fonction *time* de linux (table 1). L'indexation hors ligne est au final 1.5 fois plus lente que le temps réels sur un PC standard (Pentium IV, 2.5 GHz, 1 Go RAM). C'est le calcul des points de Harris qui prend la majorité du temps CPU mais nous n'avons pour le moment pas optimisé ce code. L'avantage de notre méthode est que ce calcul n'est effectué qu'*une seule fois* et qu'à partir de cette description, nous pouvons extraire très rapidement les descripteurs que l'on souhaite utiliser en fonction de l'application. La recherche en ligne est très rapide : 6 fois le temps réel alors

que la partie la plus gourmande est la partie image (extraction de points de Harris requêtes).

| Indexation hors ligne | 320 heures de vidéos | |
|---|---|---|
| Calcul de $S_{Traj}$ et $S_{Signal}$ | 460 hours | 0.7 T.R. |
| Construire l'espace | 5 min | 3600 T.R. |
| Détection en ligne | 7 heures de requêtes | |
| Calcul des requêtes | 45 min | 9 T.R. |
| Recherche et vote | 22 min | 19 T.R. |
| Total | 67 min | 6 T.R. |

Tableau 1 – *Temps de calcul : Temps mesuré et comparé au temps réel (T.R.).*

### 5.4 Un cas réel difficile

De nombreuses émissions TV utilisent des images d'archives et certaines effectuent en post-production de très fortes transformations de l'image comme l'insertion de personnes dans une vidéo d'archive associée à une forte translation (voir image (c) de la figure 3) ou même l'élimination du décors (voir image (d) de la figure 3). Ces transformations sont des cas extrèmes ; elles vont nous permettre d'illustrer la force des descriptions locales pour la détection de copies. En effectuant un test sur 3 heures d'émissions en requête et notre base de 320 heures de vidéos, nous avons comparé les résultats à la technique de référence (voir la table 2).

| Segments retrouvés avec la technique de référence | 43 |
|---|---|
| Segments retrouvés par notre technique | 82 |
| Temps de recherche avec la technique de référence | 7min53s |
| Temps de recherche par notre technique | 10min44s |

Tableau 2 – *Résultats de détection de copie sur un cas réel.*

Comme précédemment, notre technique montre son avantage sur les segments courts (la taille moyenne des segments supplémentaires détectés est 4.3 s) et au final la quantité de vidéo retrouvée en plus est importante : 2min 51s, ce qui correspond à un gain de 36 %, non négligeable dans un but de traçabilité du patrimoine audiovisuel.

## 6 Conclusion et perspectives

Ce papier présente une méthode robuste et efficace d'indexation et de recherche par le contenu de vidéos. Il décrit deux contributions complémentaires : la première est une description du comportement de descripteurs locaux dans une vidéo en leur attribuant un contexte. La deuxième est l'utilisation de ce contexte pour effectuer une recherche spécifique de vidéo illustrée dans ce papier pour la détection de copies. L'utilisation de labels de comportement permet de rendre l'espace de description plus compact tout en améliorant la robustesse et la discriminance du système. L'évaluation de cette méthode sur un jeu de test difficile et sur un cas réel extrème montre l'efficacité de la méthode et l'amélioration des résultats face à des systèmes état de l'art tout en étant temps réel. Les travaux futurs seront axés sur l'utilisation de descripteurs locaux complémentaires utilisant la généricité de nos algorithmes afin d'améliorer encore les performances. Une autre perspective est l'amélioration de la méthode de définition des labels de comportement en utilisant des méthodes de classification non supervisées. Enfin, les applications de ce type d'indexation sont multiples et le développement de détections de type d'émission par similarité est envisagé.

## Références


[1] P. Indyk, G. Iyengar, et N. Shivakumar. Finding pirated video sequences on the internet. Technical report, Stanford University, 1999.

[2] X-S. Hua, X. Chen, et H-J. Zhang. Robust video signature based on ordinal measure. Dans *ICIP*, 2004.

[3] A. Hampapur et R. Bolle. Comparison of sequence matching techniques for video copy detection. Dans *Conference on Storage and Retrieval for Media Databases*, pages 194–201, 2002.

[4] S-.A. Berrani, L. Amsaleg, et P. Gros. Robust content-based image searches for copyright protection. Dans *ACM Intl. Workshop on Multimedia Databases*, pages 70–77, 2003.

[5] A. Joly, C. Frelicot, et O. Buisson. Feature statistical retrieval applied to content-based copy identification. Dans *ICIP*, 2004.

[6] J. Sivic et A. Zisserman. Video Google : A text retrieval approach to object matching in videos. Dans *ICCV*, volume 2, pages 1470–1477, Octobre 2003.

[7] M. Grabner et H. Bischof. Extracting object representations from local feature trajectories. Dans *1st Cognitive Vision Workshop*, 2005.

[8] E. N. Mortensen, H. Deng, et L. Shapiro. A sift descriptor with global context. Dans *CVPR*, 2005.

[9] J. Amores, N. Sebe, et P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. Dans *CVPR*, 2005.

[10] Krystian Mikolajczyk et Cordelia Schmid. A performance evaluation of local descriptors. *ICPR*, 2003.

[11] I. Laptev et T. Lindeberg. Space-time interest points. Dans *ICCV*, 2003.

[12] C. Harris et M. Stevens. A combined corner and edge detector. Dans *4th Alvey Vision Conference*, pages 153–158, 1988.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. Dans *IJCV*, 2004.

[14] C. Tomasi et T. Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, Avril 1991.

[15] D. DeMenthon et D. Doermann. Video retrieval using spatio-temporal descriptors. Dans *ACM international conference on Multimedia*, 2003.