Établissement d'un graphe de liens au sein d'une collection de vidéos par comparaisons d'histoires

Thierry Malon, Sylvie Chambon, Alain Crouzil, Vincent Charvillat

▶ **To cite this version:**

Thierry Malon, Sylvie Chambon, Alain Crouzil, Vincent Charvillat. Établissement d'un graphe de liens au sein d'une collection de vidéos par comparaisons d'histoires. Congrès Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP 2020), Jun 2020, Vannes, France. pp.1-9. hal-02950728

HAL Id: hal-02950728
https://hal.science/hal-02950728

Submitted on 28 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:
http://oatao.univ-toulouse.fr/26390

Official URL

https://cap-rfiap2020.sciencesconf.org/data/RFIAP_2020_paper_11.pdf

To cite this version: Malon, Thierry and Chambon, Sylvie and Crouzil, Alain and Charvillat, Vincent *Établissement d'un graphe de liens au sein d'une collection de vidéos par comparaisons d'histoires.* (2020) In: Congrès Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP 2020), 23 June 2020 - 26 June 2020 (Vannes, France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Estimation des liens de recouvrement dans une collection de vidéos par comparaison d'histoires

T. Malon S. Chambon A. Crouzil V. Charvillat

IRIT, Université de Toulouse, Toulouse, France

thierry.malon@irit.fr

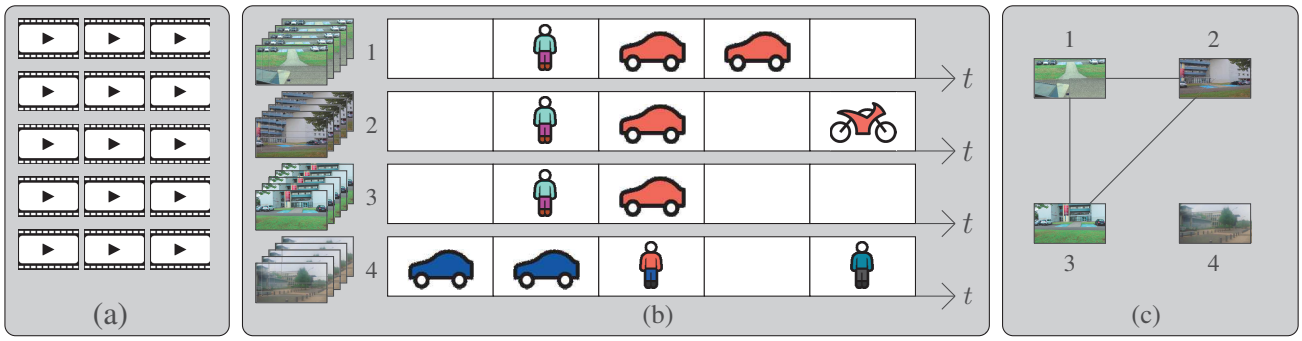


FIGURE 1 – Approche proposée pour la construction d'un graphe de liens – Pour chaque vidéo d'un ensemble de vidéos statiques (a), les objets présents sont détectés pour formuler les *histoires* des vidéos (b), puis des liens sont formés entre les vidéos aux histoires similaires pour construire un graphe de liens (c) où chaque arête indique une paire de vidéos avec des champs de vue qui se recouvrent.

Résumé

À partir d'une collection de vidéos filmées depuis des caméras fixes couvrant la même période de temps, nous présentons une méthode permettant de trouver quels groupements de vidéos présentent du recouvrement dans leurs champs de vue. Chaque vidéo est d'abord traitée individuellement : les objets présents dans le temps sont détectés à intervalles réguliers, puis une catégorie et un descripteur d'apparence leur sont attribués. La vidéo est ensuite découpée en cellules à différentes résolutions et nous assignons à chaque cellule son histoire : la liste des objets qui y ont été détectés au cours du temps. Les liens entre cellules de différentes vidéos sont alors déterminés en comparant leurs histoires : deux cellules sont liées si, dans le temps, elles présentent des détections simultanées d'objets de la même catégorie avec des apparences similaires. Les paires de vidéos avec suffisamment de liens sont alors considérées comme ayant des champs de vue en recouvrement. Les liens sont finalement visualisés dans un graphe dont chaque nœud représente une vidéoet où les arêtes indiquent des paires de vidéos en recouvrement.

Mots Clef

Réseau de caméras, vidéos avec recouvrement, appariement, multi-résolution, liens de recouvrement.

Abstract

From a collection of videos acquired from static cameras covering the same time interval, we present a method for finding groups of videos presenting overlap in their fields of view. Each video is first processed individually: objects present on time are detected at regular time steps, and a category and an appearance descriptor are assigned to each of them. Next, the video is split into cells at different resolutions and we assign each cell its story: it consists in the list of objects detected in the cell over time. Once the stories are established for each video, the links between cells of different videos are determined by comparing their stories: two cells are linked if they show simultaneous detections of objects of the same category with similar appearances over time. If enough links are found between two videos, they are considered to have overlapping fields of view. The links are finally visualized into a graph where each node represents one video, and the edges indicate pairs of overlapping videos.

Keywords

Camera network, overlapping videos, matching, multiresolution, overlap links.

1 Introduction

Dans le cas d'un événement particulier tel qu'une compétition, une manifestation, ou un attentat, de très nombreuses vidéos sont acquises par les systèmes de surveillance ou les outils d'acquisition grand public, comme les caméras de smartphone. Cette source d'information peut être capitale, notamment dans le cadre d'une enquête, mais est également difficile à exploiter. En effet, en raison de l'hétérogénéité des sources vidéos, il n'y a aucune garantie quant à la présence ou la validité d'informations externes telles que la position et l'orientation de la caméra, ou sur les liens entre différentes vidéos : observent-elles la même scène ? Aperçoit-on des objets en commun dans plusieurs caméras ? Dans le cas le plus général, les utilisateurs ne disposent que des fichiers vidéos datés selon un référentiel temporel commun.

Naviguer dans une telle collection de vidéos sans connaissance *a priori* des liens qui existent entre elles est une tâche difficile. Depuis une caméra donnée, un utilisateur peut être intéressé par un élément de la scène, un objet ou une personne par exemple, mais ne pas disposer de toute l'information qui l'intéresse en raison du point de vue de la caméra : l'objet peut par exemple être occulté ou apparaître trop petit pour être bien observable. L'utilisateur peut alors chercher une autre vidéo lui permettant de mieux observer cette même scène, depuis un point de vue qui lui soit plus favorable. Sans aucune connaissance des liens entre vidéos, il est contrait de visualiser les autres vidéos sans garantie qu'il y en ait vraiment une qui observe la même scène.

Même en visualisant un court extrait de toutes les vidéos, il n'est pas toujours évident de déterminer si deux vidéos observent la même scène. L'arrière-plan peut ne pas être assez discriminant (par exemple les devantures de deux magasins d'une même enseigne) ou être très différents même si les deux vidéos montrent la même scène (par exemple pour deux caméras qui se font face). Il faut donc prendre en compte la concordance entre les objets qui apparaissent dans chacune des vidéos au cours du temps, ce que nous appelerons *l'histoire* de la vidéo. S'il existe des régions aux histoires proches, c'est-à-dire si des objets aux apparences similaires y sont observés systématiquement et simultanément au cours du temps, il est très probable que ces deux régions se correspondent et soient dans les champs de vue des deux caméras et donc que les deux vidéos observent, au moins partiellement, la même scène. Il paraît en effet improbable qu'un tel scénario soit le résultat d'une série de coïncidences.

Ainsi, estimer ces liens au sein d'une collection de vidéos nécessite la visualisation des éléments au premier plan des vidéos et pas uniquement la visualisation de l'arrière-plan. Ce fait rend d'autant plus pénible la tâche de recherche de liens pour un opérateur humain. L'idée de cet article est d'automatiser cette tâche en exploitant le principe décrit ci-dessus : deux vidéos observent la même scène si elles présentent des régions qui racontent la même *histoire*, c'est-à-dire dans lesquelles nous détectons simultanément des objets de la même catégorie (véhicule, personne ou autre) et d'apparences similaires au cours du temps. L'objectif est alors, à partir d'un ensemble de vidéos issues de caméras fixes couvrant le même intervalle de temps, de construire un graphe dont chaque nœud est un fichier vidéo et où chaque arête représente un lien entre des régions communes aux champs de vue de deux caméras. Ce processus est illustré par la figure 1.

Nos contributions sont les suivantes :

1. Nous proposons une méthode d'estimation des liens de recouvrement d'une collection de vidéos par établissement et comparaisons d'histoires.
2. Nous évaluons cette méthode sur un ensemble de 63 vidéos issues de plusieurs jeux de données. Les résultats obtenus sont encourageants et les graphes de liens obtenus contiennent peu d'erreurs.

Nous présentons les travaux connexes aux nôtres dans la section 2, avant de positionner nos travaux et de formuler le problème dans la section 3. Nous exposons ensuite notre approche dans la section 4. Après avoir défini la terminologie et le concept d'*histoire* dans la section 4.1, puis la distance entre histoires dans la section 4.2, nous décrivons notre algorithme de calcul du graphe de liens d'une collection de vidéos dans la section 4.3. Cette méthode est évaluée dans la section 5, avant de conclure en section 6.

2 Travaux connexes

Les travaux exploitant plusieurs vidéos s'appuient généralement sur trois types d'hypothèses :

1. Les liens entre vidéos sont connus *a priori*. C'est le cas de [1], où toutes les vidéos proviennent d'un même campus. Les auteurs proposent de ré-identifier des piétons entre différentes caméras et de calculer conjointement la topologie du réseau de caméras en déterminant les transitions les plus courantes d'une caméra à une autre. La topologie initiale est calculée à partir des ré-identifications les plus fiables, puis ils renforcent itérativement la ré-identification en s'appuyant sur la topologie et inversement.
2. Des informations complémentaires sont disponibles (géo-localisation des caméras ou paramètres de calibrage). C'est le cas de [2] où les paramètres des caméras sont supposés connus et où l'hypothèse est faite que les vidéos observent toutes la même scène.
3. Un mécanisme permet de passer d'une vidéo à une autre, soit par des interactions de l'utilisateur, soit automatiquement. Dans un contexte sportif, [3] attribue des scores à chaque vue à partir de l'activité des objets (calculée comme la proportion de pixels ne faisant pas partie de l'arrière-plan), leurs tailles, leurs positions et leurs nombres, ainsi qu'en fonction d'événements qui y sont détectés grâce au mouvement des objets. La vue affichée est régulièrement et automatiquement mise à jour pour montrer la vue avec le score le plus élevé. L'utilisateur n'a aucun contrôle et ne peut ni intéragir pour

changer manuellement de vue, ni visualiser les liens entre les différentes vues. Dans [2], les auteurs proposent à l'utilisateur de sélectionner une région d'intérêt dans une vidéo parmi une collection de vidéos qui observent une même scène et de le rediriger vers la vidéo dans laquelle les objets situés dans la région de la requête occupent une plus grande partie de l'image que dans la vidéo initiale.

Peu de travaux se placent dans des conditions où aucun *a priori* n'est supposé quant aux relations entre les différentes vidéos. Nous pouvons citer les travaux [4, 5] qui découpent les vidéos en régions selon une grille puis cherchent à mettre en correspondance des régions de différentes vidéos. Ils décrivent pour cela un processus d'exclusion qui consiste en ce que deux régions qui observent la même partie d'une scène soient systématiquement occupées ou non occupées aux mêmes instants, sous réserve que les vidéos soient synchronisées temporellement. Ils supposent initialement des liens entre toutes les paires de régions et retirent progressivement des liens entre deux régions dès qu'un objet est observé dans une région alors qu'aucun objet n'est observé dans l'autre région. Du fait qu'initialement, tous les liens sont supposés, cette méthode laisse de nombreux faux liens entre des régions qui ne se correspondent pas, notamment entre des régions où aucun objet n'est apparu, rendant peu fiable une navigation entre vidéos qui serait basée sur ces liens.

3 Positionnement et formulation du problème

Dans nos travaux précédents [6], nous avons repris l'idée de lier des régions de vidéos qui présentent des profils d'activité similaires, à savoir des régions où des objets apparaissent simultanément de façon synchronisée, et nous avons enrichi cette approche en tenant compte de la catégorie des objets. Contrairement à l'approche citée précédemment [4, 5], ici les liens entre régions ne sont pas binaires (un lien ou pas de lien), mais quantifiés par un score qui traduit le degré de cohérence entre les profils d'activité des régions. Plus des objets de même catégorie sont apparus simultanément dans les deux régions, plus ce score est élevé. Ces scores nous permettent de mettre en place des cartes de correspondance entre régions. Nous proposons ensuite à l'utilisateur de formuler une requête de trajectoire dans une vidéo et, en exploitant les cartes de correspondance entre régions, la méthode proposée permet de reformuler la trajectoire en son équivalent dans les autres vues, puis de classer les autres vidéos qui offrent une meilleure visualisation de la trajectoire requête, permettant ainsi une navigation entre vidéos. L'utilisateur n'a cependant aucune vue d'ensemble des différentes vidéos. De plus, seule la catégorie des objets est prise en compte alors que l'apparence pourrait aussi être exploitée. Dans la continuité de ce travail préliminaire, les travaux que nous présentons dans cet article s'inscrivent donc à la frontière entre :

- L'appariement d'images, dont le but est de retrouver les primitives qui se correspondent d'une image à l'autre. Pour estimer ces liens, nous établissons des correspondances entre cellules extraites des vidéos en appariant les objets qui y sont détectés. Nous utilisons des méthodes de détection de l'état de l'art : SSD [7] *Single Shot Multi-Box Detector*, Mask-RCNN [8] *Regional Convolutional Neural Network*, ou YOLO [9], *You Only Look Once*.
- L'estimation de la topologie d'un réseau de caméras, qui consiste à estimer les positions relatives d'un ensemble de caméras observant une même scène. Nous souhaitons obtenir un graphe donnant les liens entre les différentes vidéos. Ces liens sont liés à la proportion d'éléments communs entre les vidéos et, en ce sens, nous déterminons une certaine forme de topologie.
- La ré-identification, qui consiste à retrouver, à partir d'une image requête d'un objet, à classer les potentielles autres occurrences de cet objet dans d'autres vidéos en exploitant des indices visuels.

Il est important de préciser que la ré-identification propose un classement, mais que son rôle n'est pas d'affirmer que deux imagettes correspondent ou non au même objet : il est généralement supposé que ce sera un opérateur humain qui statuera, aidé par le classement que lui propose la ré-identification. Ainsi, hormis de rares travaux tels que [10] qui traitent d'approches de ré-identification en milieu ouvert, les approches classiques sont testées sur des jeux de données où chaque objet apparaît dans au moins deux caméras, pour ne pas avoir à traiter le cas des objets qui n'apparaissent que dans une seule caméra. Elles sont évaluées par des métriques qui mesurent la proportion d'imagettes résultats qui correspondent au même objet que la requête dans les N premières réponses avec $N = 1, 5, 10...$

Dans le contexte applicatif que nous envisageons, rien ne garantit qu'un objet apparaisse dans plusieurs caméras. Nous ne cherchons pas à ré-identifier précisément les objets détectés entre caméras, mais à déceler une cohérence globale entre les apparences des objets qui apparaissent à des instants proches dans différentes vidéos pour en déduire ou non un lien. En cela, notre approche diffère considérablement de la tâche de ré-identification, tout en empruntant ses principes généraux (détection d'objets, calcul de descripteurs d'apparence et classement en fonction de la similarité entre descripteurs).

À présent, nous pouvons formuler le problème que nous souhaitons résoudre. À partir d'une collection de N vidéos issues de caméras fixes et couvrant la même période de temps, le but est de déterminer quels groupes de vidéos observent la même scène, c'est-à-dire quelles vidéos présentent du recouvrement dans leurs champs de vue. En sortie, nous souhaitons obtenir un graphe de liens entre vidéos. Plus précisément, chaque nœud de ce graphe est une vidéo et chaque arête représente un lien de recouvrement entre les champs de vue des caméras (voir figure 1).

4 Histoire d'une vidéo

Dans cette partie, la notion d'*histoire* est introduite. Cette notion est un concept clé dans notre approche de construction du graphe de relations entre vidéos. Une mesure de distance entre histoires est ensuite définie. Cette distance entre histoires de deux régions de deux vidéos différentes est d'autant plus faible que des objets de même catégorie et d'apparences proches y sont simultanément détectés au cours du temps. Enfin, nous mettons en place une approche multi-résolution à différentes échelles spatiales et temporelles pour éviter de comparer de nombreuses paires de régions de résolution fine lorsque la concomitance des détections a déjà été établie à une résolution plus grossière. À l'inverse, cette approche multirésolution permet d'éviter d'effectuer des recherches à des résolutions plus fines dès que la dissemblance ne fera plus de doute.

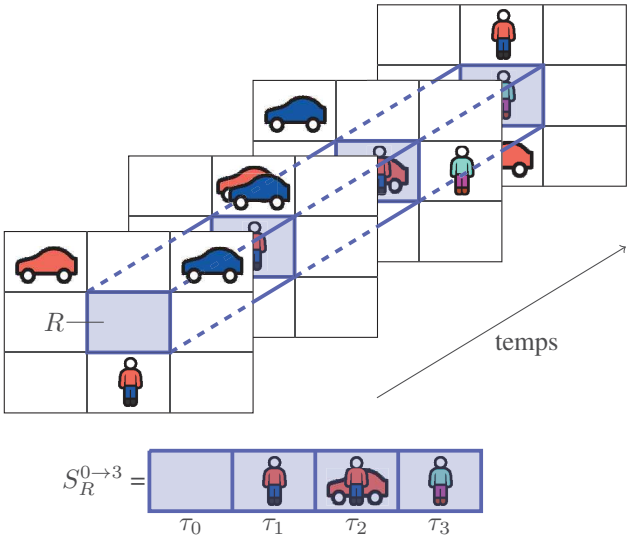


FIGURE 2 – Histoire d'une région – À des pas de temps réguliers, les objets sont détectés dans chaque région de la vidéo et se font assigner une catégorie et un descripteur d'apparence. L'histoire d'une région donnée R apparaît en bleu. Elle raconte qu'au temps τ_0, il n'y a aucun objet dans R, puis qu'à τ_1, il y a une personne, et ainsi de suite.

4.1 Terminologie employée et définition

Lorsque l'on raconte une histoire, on décrit généralement les personnages, leurs apparences, les lieux qu'ils traversent et leurs actions, avec plus ou moins de détails, et dans l'ordre chronologique. Pour ces raisons, nous avons choisi le terme générique « histoire » pour regrouper des propriétés variées telles que la catégorie, la position et l'apparence d'objets détectés dans une vidéo au fil du temps.
Considérons que chaque vidéo est découpée en régions, par exemple en cellules selon un quadrillage régulier. Pour calculer l'histoire d'une région, des objets y sont régulièrement détectés tous les pas de temps $\tau_i = \tau \times i$ par des algorithmes de détection de l'état de l'art comme ceux cités dans la section 3. L'histoire de la région consiste alors en une liste temporelle qui, à chaque pas de temps, contient l'ensemble des objets détectés dans cette région, comme illustré par la figure 2. Nous parlons de l'histoire de la vidéo lorsque la région considérée occupe tout le cadre.
Nous assignons chaque détection à la cellule qui contient son *point de contact au sol*, c'est-à-dire le pixel situé au milieu de l'arête horizontale inférieure de sa boîte englobante, comme dans [11] (voir figure 3). Ce choix est motivé par le fait que cela ancre le pixel dans le plan du sol, ce qui permet d'éviter des ambiguïtés dues à la profondeur : de manière générale, un même pixel peut être occupé par des objets situés à des distances différentes de la caméra, mais dans ce cas leurs pixels au niveau du sol sont différents.

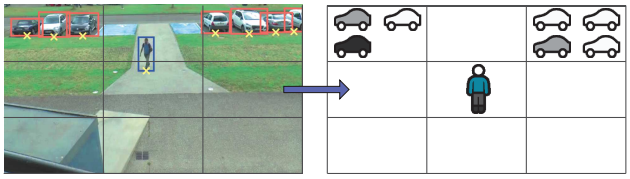


FIGURE 3 – Établissement des histoires. À gauche : la fenêtre vidéo est découpée en 9 régions et on y détecte les objets présents et leurs catégories (les rectangles bleu et rouges correspondent aux catégories respectivement « humain » et « voiture »). Le *point de contact au sol* est marqué d'une croix jaune. À droite : l'histoire de chaque région est constituée de la liste des objets qui y apparaissent au niveau de leur *point de contact au sol*, et sont caractérisés par une catégorie et un descripteur d'apparence.

L'histoire d'une région R au cours de l'intervalle de temps $\tau_0, ..., \tau_i$ est notée $S_R^{0\to i}$. L'histoire en un instant τ_j est notée S_R^j et l'histoire sur toute la durée de la vidéo est notée S_R. Nous notons V la région correspondant au cadre complet et T le nombre de pas de temps total. Cette définition permet de calculer des histoires à différentes résolutions spatiales et temporelles en ne considérant que les objets détectés dans un certain intervalle de temps et dans une certaine région du cadre (voir figure 4). L'utilisation d'une analyse multi-résolution permet d'estimer les liens entre régions de façon robuste et de réduire les temps de calculs, comme détaillé dans la section 4.3. La prochaine étape que nous décrivons consiste à définir une distance entre histoires.

4.2 Distance entre histoires

Afin de définir la distance entre deux histoires de même longueur S_R et $S_{R'}$, nous introduisons d'abord la notion d'objet commun aux deux histoires. Un objet O est commun aux deux histoires s'il apparaît dans la région R de la première vidéo au temps τ_i et si un objet O' de même catégorie que O et d'apparence similaire apparaît dans la région R' ou dans son voisinage spatial (par exemple, dans son 8-voisinage), au temps τ_i ou dans son voisinage temporel $\{\tau_{i-k}, ..., \tau_i, ..., \tau_{i+k}\}$, comme illustré par la figure 5. La similarité d'apparence est établie si la distance entre

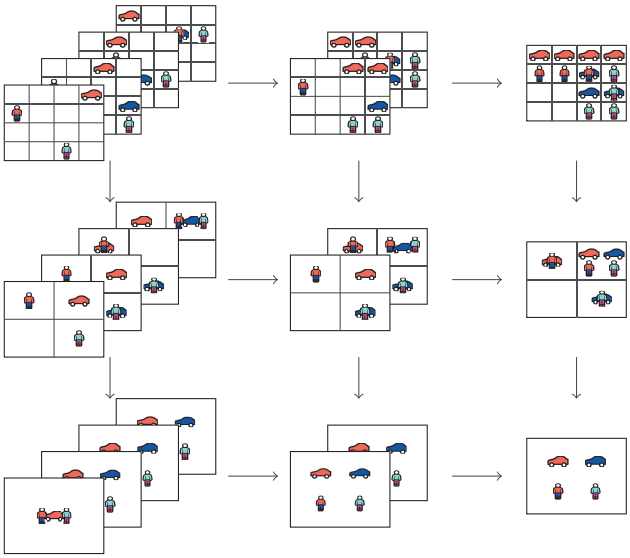


FIGURE 4 – La multirésolution – Un exemple d'histoire à différentes résolutions d'espace et de temps. De gauche à droite : la résolution temporelle est réduite en prenant l'union des histoires successives deux à deux. De haut en bas : la résolution spatiale est réduite en prenant l'union de cellules adjacentes. L'histoire en haut à gauche est ainsi la plus détaillée tandis que l'histoire en bas à droite est un résumé de tous les objets qui apparaissent dans la vidéo, peu importe leur emplacement ou l'instant où ils apparaissent.

les descripteurs d'apparence est inférieure au seuil d'apparence σ_{app}. On dit alors que O a trouvé un correspondant dans R'.

Aucune ré-identification d'objets par paire n'est effectuée : plusieurs objets différents de R peuvent trouver un même correspondant dans R'. C'est en cela que notre approche cherche à repérer une cohérence globale entre objets détectés au cours du temps sans chercher à ré-identifier précisément les objets. Notons $C(S_R, S_{R'})$ la proportion d'objets apparaissant dans R qui ont trouvé un correspondant dans R' et $|S_R|$ (respectivement $|S'_R|$) le nombre d'objets apparaissant dans l'histoire S_R (respectivement S'_R). Nous définissons la distance $d(S_R, S_{R'})$ entre deux histoires par :

$$d(S_R, S_{R'}) = 1 - \frac{C(S_R, S_{R'}) + C(S_{R'}, S_R)}{|S_R| + |S_{R'}|} \qquad (1)$$

Cette distance peut s'interpréter comme le pourcentage d'objets issus des deux histoires confondues qui ne trouvent pas de correspondant. Pour éviter de compter une multitude de fois les objets statiques de la scène, tels que les voitures garées, nous mesurons l'indice de Jaccard entre toutes les paires d'objets de même catégorie aux temps τ_i et τ_{i+1}. Les paires de détections présentant un indice supérieur à 0.9 ne sont pas prises en compte dans le calcul des histoires.

Notre dernière contribution concerne l'établissement du graphe de liens entre vidéos à partir des distances entre histoires des régions qui les composent.

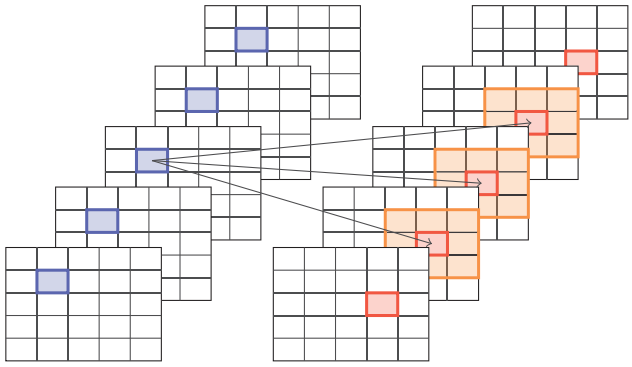


FIGURE 5 – Illustration du calcul de dissimilarité entre deux histoires S_R (en bleu) et $S_{R'}$ (en rouge) – Pour chaque objet apparaissant dans S^i_R, on cherche un correspondant de même catégorie et d'apparence similaire dans le voisinage spatio-temporel de $S^i_{R'}$ (les régions en orange). La distance $(S_R, S_{R'})$ est alors le pourcentage d'objets de S_R et $S_{R'}$ qui ne trouvent pas de correspondant.

4.3 Calcul du graphe de liens

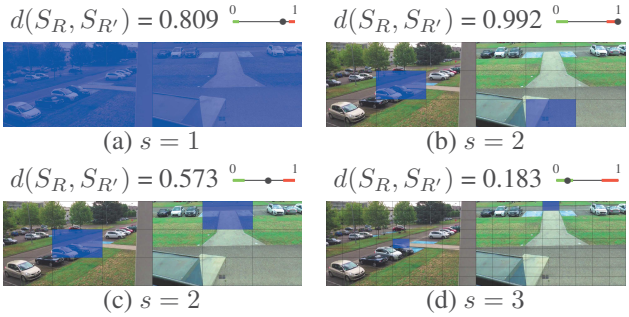
Dans cette section, l'algorithme de recherche de recouvrement entre vidéos est détaillé (voir algorithme 15). Pour établir que deux vidéos présentent du recouvrement, nous calculons les distances entre leurs histoires à différentes résolutions, de la résolution la plus grossière jusqu'à la résolution la plus fine, si nécessaire. Pour chaque paire de vidéos, les différentes résolutions contribuent au résultat comme suit. Les deux vidéos sont d'abord comparées à la résolution la plus grossière $s = 1$ en prenant en compte tous les objets qui apparaissent sur de larges intervalles de temps. Si la distance entre leurs histoires dépasse un seuil σ^s_{rejet}, dépendant du niveau de résolution s, les vidéos sont considérées comme n'ayant rien à voir et ne présentent donc pas de recouvrement. Elles ne sont pas comparées à une résolution plus fine dans ce cas. Si, au contraire, la distance entre histoires est inférieure à un seuil σ^s_{accept}, les vidéos présentent des objets similaires au cours du temps et il n'est alors pas nécessaire de chercher des correspondances à une résolution plus fine : le recouvrement est établi. Le plus souvent, la distance entre les histoires S_R et S'_R tombe entre les deux seuils. Nous subdivisons alors les régions en cours de comparaison et toutes les paires de sous-régions sont étudiées à leur tour. Les seuils σ^s_{accept} et σ^s_{rejet} s'adaptent en fonction de la résolution car à mesure que les comparaisons deviennent plus fines, il faut être plus sévère envers les paires de cellules qui ne se correspondent pas pour éviter de trop subdiviser et à l'inverse plus clément envers les paires qui se correspondent car le voisinage de recherche autorisé se réduit (voir figure 6).

Le processus est répété jusqu'à ce qu'il ne reste plus aucune paire de régions à comparer. Nous disposons alors d'une liste de liens entre régions de la paire de vidéos. Nous allons maintenant évaluer le graphe obtenu par cette approche.

Algorithme 1 : Recouvrement entre deux vidéos. La variable *matches* est l'ensemble des paires de régions qui se correspondent. La variable *candidates* est l'ensemble des triplets composés de deux régions à comparer et d'un niveau de résolution. Lorsqu'un candidat (R_1, R_2, s) est rejeté (instruction *null*), aucune correspondance n'est cherchée à une résolution plus fine.

Entrée : deux histoires de deux vidéos S^{V_1} et S^{V_2}
Sortie : matches : liens entre V_1 et V_2

matches = [ ]
candidates = {(V_1, V_2, 1)}
tant que *candidates* $\neq \emptyset$ **faire**
  (R_1, R_2, s) = candidates[0]
  si $d(S_{R_1}, S_{R_2}) \leqslant \sigma^s_{accept}$ **alors**
    matches = matches $\cup$ {(R_1, R_2)}
  sinon
    si $d(S_{R_1}, S_{R_2}) \geqslant \sigma^s_{rejet}$ **alors**
      candidates = candidates $\cup$ { (r_1, r_2, s+1)
       $\forall$ (r_1, r_2) $\in$ Subdiv(R_1) $\times$ Subdiv(R_2) }
    sinon
      null
    fin
  fin
  supprimer candidates[0]
fin



(a) $s = 1$ (b) $s = 2$ (c) $s = 2$ (d) $s = 3$

FIGURE 6 – Estimation du recouvrement entre deux vidéos. En vert : zone d'acceptation. En rouge : zone de rejet. (a) Les histoires sont d'abord comparées à une échelle globale. Leur distance ne tombant dans aucune des deux zones d'arrêt, la fenêtre est subdivisée et on compare toutes les paires de sous-régions. (b) Une paire de sous-régions de (a) dont la distance entre histoires dépasse le seuil σ^2_{reject} : on ne comparera pas leurs sous-régions. (c) Une autre paire de sous-régions dont la distance entre histoires tombe entre les deux zones d'arrêt. Toutes les paires de sous-régions sont comparées. (d) Une paire de sous-régions de (c) dont la distance entre histoires est inférieure au seuil σ^3_{accept}. Ces sous-régions sont considérées comme se correspondant.

5 Expérimentations

5.1 Jeux de données

L'approche présentée requiert des vidéos avec recouvrement sur une période de temps commune suffisamment longue pour accumuler des preuves de corrélation entre les objets qui apparaissent. Nous utilisons donc des séquences vidéos qui durent au moins quelques minutes et dans lesquelles apparaissent au moins trois objets différents. Seuls quelques jeux de données offrent des vues multiples qui satisfont à ces conditions. Au total, nous avons pu trouver 63 vidéos issues de 4 jeux de données publics :

- 7 vidéos sont des séquences issues de vidéos de caméras de surveillance en direct disponibles publiquement sur Youtube [1]. Deux paires présentent du recouvrement.
- 12 vidéos du jeu de données MEVA [12] disposées en 6 groupements de 2 vidéos. Ces séquences sont des cas difficiles car la zone de recouvrement est faible et les piétons se déplacent en groupes.
- 19 vidéos du jeu de données de l'EPFL intitulé Multi-camera Pedestrian Videos [13], disposées en 5 groupements de 3 à 4 vidéos chacun. Jusqu'à 7 piétons y apparaissent simultanément.
- 25 vidéos issues du jeu de données ToCaDa que nous avons proposé dans le but de contribuer à l'essor de ce type de base de données dans [14]. Parmi ces vidéos, 19 filment l'entrée d'un même bâtiment. Les 6 autres ont des points de vue disjoints. Une trentaine d'objets (véhicules et piétons) y apparaissent.

1. Les auteurs n'indiquent aucune manière de citer la source de leurs vidéos. Nous avons choisi de ne pas indiquer directement la source. Les résultats présentés respectent l'anonymat des personnes filmées.

Les nombres, durées et résolutions des vidéos sont indiqués dans les premières colonnes du tableau 1. Le graphe théorique des liens est illustré par la figure 7. Notons que, dans les jeux de données utilisés, chaque vidéo d'un groupement est liée à toutes les autres vidéos du groupement.

5.2 Évaluation

Notre approche exploite la cohérence globale entre détections d'objets pour former un graphe de liens entre vidéos. Bien que nous ne proposions pas de contribution en terme de détection d'objets, nous évaluerons dans un premier temps les résultats obtenus en termes de détections, car c'est à partir de ces détections qu'est calculé le graphe de liens. Trois méthodes de l'état de l'art sont évaluées, à savoir SSD [7], Mask-RCNN [8] et YOLOv3 [9]. Pour chaque méthode, nous détectons les objets présents dans chaque vidéo tous les pas de temps $\tau = 500ms$. Les détections sont évaluées par la précision, le rappel et le score F_1 [15], en considérant une boîte englobante comme bien détectée si la catégorie de l'objet est correctement déterminée et si l'indice de Jaccard de cette boîte englobante avec la vérité terrain est supérieur à 0,5 (comme dans le Pascal VOC challenge [16]). D'autre part, seules les détections avec une confiance supérieure à 0,5 sont gardées. Les résultats sont présentés dans le tableau 1. De manière générale, les trois méthodes de détection font peu de fausses détections, ce qui se traduit par un rappel proche de 100%.

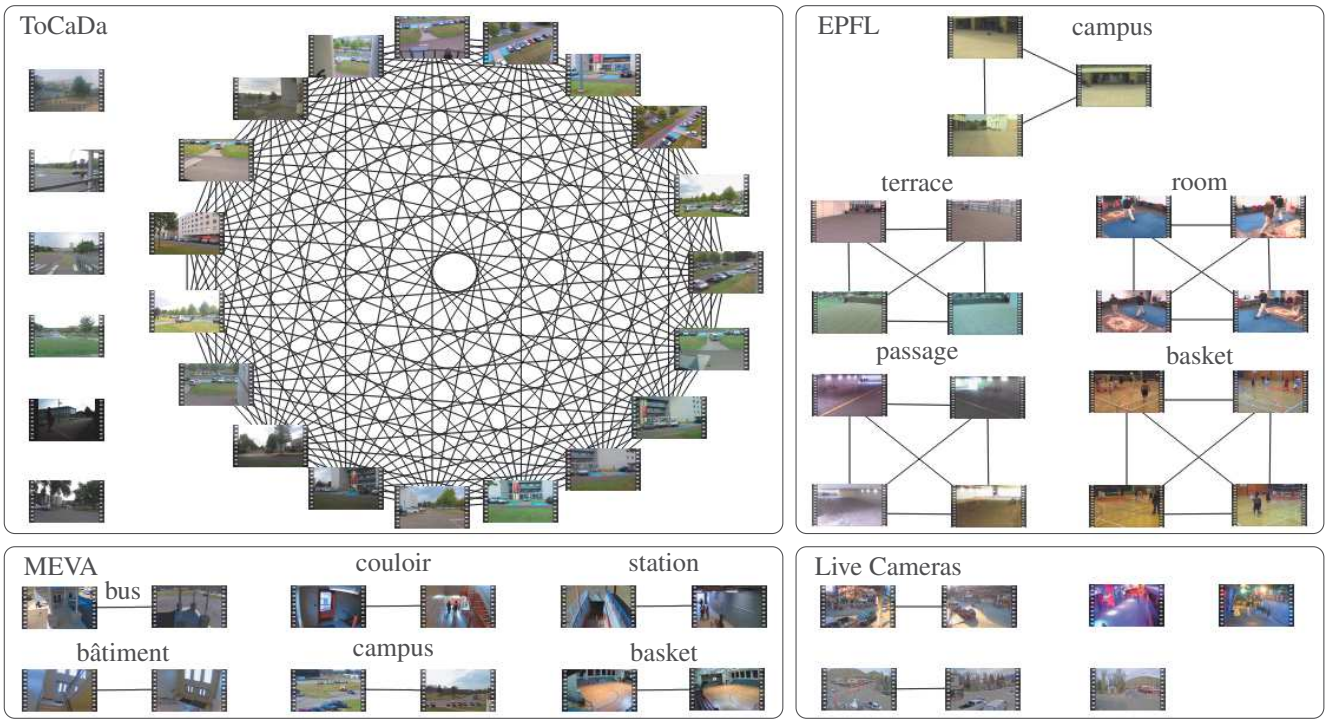


FIGURE 7 – Graphe de liens théorique des 63 vidéos issues de divers jeux de données publics. Une arête entre deux vidéos indique qu'elles présentent du recouvrement dans leurs champs de vue.

En terme de précision, SSD présente des scores très inférieurs aux autres méthodes, car la plupart des vidéos des jeux de données utilisés contiennent des objets petits et nombreux, que l'algorithme ne parvient pas à détecter. YOLOv3 donne en moyenne les meilleurs scores. On remarque que Mask-RCNN obtient les meilleurs résultats sur les vidéos de haute résolution.

Évaluons à présent l'approche de formulation du graphe de liens. Trois niveaux de résolutions spatiales et temporelles sont utilisés. À la résolution spatiale la plus grossière, on considère la fenêtre dans sa globalité. À la résolution la plus fine, on considère un quadrillage régulier de 9×9 cellules. La résolution temporelle varie de $\pm 3\tau$ à la résolution la plus grossière jusqu'à $\pm 1\tau$ à la résolution la plus fine. Pour chaque boîte englobante, trois descripteurs différents sont calculés sur l'imagette qu'elle contient. Le premier est le descripteur *ColorNames* [17]. Il découpe l'imagette en 6 tranches horizontales et associe à chaque tranche une distribution de 16 couleurs saillantes. Les descripteurs sont ensuite comparés en utilisant la métrique KISSME, *Keep It Simple and Straightforward MEtric* [18]. Le deuxième descripteur employé est HOG, *Histogram of Oriented Gradients*, l'histogramme des gradients orientés [19]. La métrique utilisée est une distance euclidienne. Enfin, le troisième type de descripteur est une représentation latente de l'imagette dans l'espace latent d'un réseau de neurones convolutif. Nous avons choisi le réseau ResNet18 [20], préentrainé sur la base d'images ImageNet [21]. Ces descripteurs sont comparés en distance cosinus.

Nous calculons alors le graphe de liens en faisant varier les seuils d'apparence σ_{app} et d'acceptation σ^1_{accept} entre 0 et 1 par pas de $0,05$, et prenons $\sigma^s_{accept} = 1 - (1 - \sigma^1_{accept})^s$. σ^s_{rejet} est pris égal à $(1+\sigma^s_{accept})$ / 2. Plusieurs variantes de la méthode sont testées :

- en considérant toutes les détections simultanées, qu'importent la catégorie et l'apparence des objets ;
- en ne prenant en compte que les détections simultanées d'objets de la même catégorie ;
- en prenant en compte la catégorie et l'apparence des détections simultanées. Les trois types de descripteurs sont testés séparément.

Pour chaque paire de seuils $(\sigma_{app}, \sigma^s_{accept})$, nous comptabilisons les liens correctement trouvés (les vrais positifs), les mauvais liens trouvés (les faux positifs) et les liens manquants (les faux négatifs), puis nous calculons le score F_1 associé. Le tableau 2 indique les valeurs maximales de F_1 obtenues pour chaque jeu de données. Au niveau des détecteurs, Mask-RCNN et YOLOv3 présentent, comme pour les résultats de détection, de bons résultats, tandis que la méthode SSD donne des scores moins élevés. Nous expliquons ce résultat par la mauvaise précision de SSD pendant la phase de détection. De manière générale, la prise en compte de la catégorie améliore le score. L'effet est moindre sur les jeux de données EPFL et MEVA car ils contiennent essentiellement une seule catégorie d'objets (des personnes). L'utilisation de l'apparence est également bénéfique et, parmi les trois types de descripteurs testés, la représentation latente donne les meilleurs résultats.

	Propriétés			SSD [7]			Mask-RCNN [8]			YOLOv3 [9]		
Jeu de données	#vid	Durée	Résolution	Prc	Rpl	F_1	Prc	Rpl	F_1	Prc	Rpl	F_1
Live Cameras	7	5min	1920×1080	7,5	99,2	13,8	84,6	95,3	89,2	83,8	99,5	**90,8**
MEVA (global) [12]	12		1920×1080	17,9	99,4	25,9	84,9	98,7	**90,9**	72,9	98,2	82,9
MEVA basket	2	5min	1920×1080	1,9	100	3,6	85,8	98,6	**91,6**	58,1	98,3	73,0
MEVA couloir	2	5min	1920×1080	61,9	100	76,4	92,9	100	**96,2**	91,0	100	95,2
MEVA bâtiment	2	5min	1920×1080	19,6	100	32,7	73,3	100	**83,5**	61,7	95,5	73,9
MEVA bus	2	5min	1920×1080	14,0	96,4	24,4	88,9	99,4	**93,7**	80,4	99,3	88,7
MEVA campus	2	5min	1920×1080	4,2	100	7,7	83,0	100	**90,6**	64,7	96,1	76,8
MEVA station	2	5min	1920×1080	5,8	100	10,4	85,8	94,1	**89,8**	81,4	100	89,6
EPFL (global) [13]	19		360×288	54,4	99,2	67,1	74,7	95,4	83,4	92,6	99,5	**95,9**
EPFL room	4	1min58	360×288	76,0	99,2	85,9	84,7	99,3	91,4	95,0	100	**97,4**
EPFL campus	3	3min55	360×288	74,4	98,9	84,9	83,8	97,8	90,2	95,3	100	**97,6**
EPFL basket	4	6min14	360×288	26,9	99,4	41,9	66,7	95,5	78,5	94,9	99,6	**97,2**
EPFL passage	4	1min40	360×288	28,2	100	43,5	61,8	85,1	70,2	90,2	99,1	**94,3**
EPFL terrace	4	2min59	360×288	66,4	98,7	79,3	76,8	99,3	86,6	87,6	98,9	**92,9**
ToCaDa [14]	25	4min48	960×540	13,5	99,8	22,4	74,8	96,5	83,1	80,5	99,4	**88,2**
Tous	63			25,7	99,5	35,3	77,7	96,4	85,2	83,0	99,2	**89,7**

TABLE 1 – Résultats des détections – Nous présentons également les propriétés des différents jeux de données : nombre (#vid), durée et résolution des vidéos. Pour l'évaluation de la détection, nous indiquons la précision moyenne (Prc), le rappel moyen (Rpl) et la moyenne des scores F_1. Les résultats sont donnés en pourcentage.

	SSD [7]					Mask-RCNN [8]					YOLOv3 [9]				
Jeu de données	base	ctg	ctgcn	ctghg	ctglr	base	ctg	ctgcn	ctghg	ctglr	base	ctg	ctgcn	ctghg	ctglr
Live Cameras	19	22	27	25	25	80	**100**	**100**	**100**	**100**	80	**100**	**100**	**100**	**100**
MEVA (global)	28	29	32	30	35	59	60	67	62	**80**	59	59	65	68	76
EPFL (global)	56	56	64	59	67	61	61	71	65	79	63	63	74	66	**81**
ToCaDa	25	29	34	32	35	83	87	88	87	88	85	88	89	88	**90**
Tous	16	19	21	20	23	32	36	43	39	47	35	38	47	44	**51**

TABLE 2 – Scores F_1 les plus élevés des graphes de liens obtenus en fonction de trois méthodes de détection et les variantes de l'approche proposée. L'approche de base (base) prend en compte toutes les détections simultanées, peu importe la catégorie ou l'apparence. L'approche par catégorie uniquement (ctg) ne prend pas en compte l'apparence. Les approches ctgcn, ctghg et ctglr prennent en compte la catégorie et l'apparence en utilisant respectivement des descripteurs : Color Names, HoG et des descripteurs latents d'un réseau convolutif. Les résultats sont donnés en pourcentage.

6 Conclusion

À partir d'un ensemble de vidéos issues de caméras fixes couvrant la même période de temps, la méthode proposée parvient à estimer le graphe des liens de recouvrement entre vidéos. Il s'agit d'un graphe où chaque nœud correspond à un fichier vidéo et où les paires de vidéos dont les champs de vue se croisent sont liées par une arête. Notre méthode s'appuie sur des détections concomitantes d'objets dans des paires de vidéos. Nous l'avons évaluée sur différents jeux de données en prenant en compte ou non la catégorie des objets détectés et en testant différents descripteurs d'apparence. Les graphes obtenus sur chaque jeu de données sont précis et le graphe obtenu sur l'ensemble des jeux de données reste correct. Les meilleurs résultats ont été obtenus en utilisant YOLOv3 pour la détection d'objets, et des descripteurs latents pour caractériser l'apparence.
Nos futurs travaux chercheront à améliorer l'estimation de la topologie du réseau de caméras en exploitant l'arrière-plan des vidéos. En effet, une fois établi que des caméras filment la même scène, il nous paraît intéressant de préciser si elles offrent des points de vue similaires (positions et angles de vue des caméras proches) ou très différents (l'angle de vue varie significativement, comme pour deux caméras qui se font face). Enfin, nous souhaitons affiner le graphe via des liens indiquant un délai de transition entre la disparition dans une caméra et l'apparition dans une autre.

Références

[1] Y.-J. Cho, S.-A. Kim, J.-H Park, K. Lee et K.-J. Yoon, Joint person re-identification and camera network topology inference in multiple cameras, *Computer Vision and Image Understanding*, volume 180, pages 34-46, 2019.

[2] A. Carlier, L. Calvet, P. Gurdjos, V. Charvillat et W. T. Ooi, Querying Multiple Simultaneous Video Streams with 3D Interest Maps, *Visual Content Indexing and Retrieval with Psycho-Visual Models*, pages 125-144, 2017.

[3] F. Daniyal, M. Taj et A. Cavallaro, Content and task-based view selection from multiple video streams, *Multimedia tools and applications*, volume 46, pages 235-258, 2010.

[4] A. Van Den Hengel, A. Dick et R. Hill, Activity topology estimation for large networks of cameras, *IEEE International Conference on Video and Signal Based Surveillance*, pages 44-44, 2006.

[5] A. Van Den Hengel, A. Dick, H. Detmold, A. Cichowski et R. Hill, Finding camera overlap in large surveillance networks, *Asian Conference on Computer Vision*, pages 375-384, 2007.

[6] T. Malon, S. Chambon, V. Charvillat et A. Crouzil, Estimation of correspondent trajectories in multiple overlapping synchronized videos using correlation of activity functions, *IEEE International Conference on Image Processing*, pages 994-998, 2019.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu et A. C. Berg, SSD : Single Shot MultiBox Detector, *arXiv*, 2015.

[8] K. He, G. Gkioxari, P. Dollár et R. Girshick, Mask R-CNN, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2961-2969, 2020.

[9] J. Redmon et A. Farhadi, YOLOv3 : An Incremental Improvement, *arXiv*, 2018.

[10] S. Chan-Lang, Q.-C. Pham et C. Achard, Closed and open-world person re-identification and verification, *International Conference on Digital Image Computing : Techniques and Applications*, pages 1-8, 2017.

[11] S. Khan et M. Shah, Consistent labeling of tracked objects in multiple cameras with overlapping fields of view, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25, pages 1355-1360, 2003.

[12] Multiview Extended Video with Activities (MEVA) dataset, *https://mevadata.org/*

[13] F. Fleuret, J. Berclaz, R. Lengagne et P. Fua, Multi-Camera People Tracking with a Probabilistic Occupancy Map, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30, pages 267-282, 2008.

[14] T. Malon, G. Roman-Jimenez, P. Guyot, S. Chambon, V. Charvillat, A. Crouzil, A. Péninou, J. Pinquier, F. Sèdes et C. Sénac, Toulouse campus surveillance dataset : scenarios, soundtracks, synchronized videos with overlapping and disjoint views, *ACM Multimedia Systems Conference*, pages 393-398, 2018.

[15] K. Murphy1, A. Torralba, D. Eaton et W. Freeman, Object detection and localization using local and global features, *Toward Category-Level Object Recognition*, pages 382-400, 2006.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn et A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, *International Journal of Computer Vision*, volume 88, pages 303-338, 2010.

[17] Y. Yang, J. Yang, J. Yan, S. Liao et S. Z. Li, Salient Color Names for Person Re-Identification, *European Conference on Computer Vision*, pages 536-551, 2014.

[18] M. Kostinger, M. Hirzer, P. Wohlhart, P.M. Roth et H. Bischof, Large scale metric learning from equivalence constraints, *Conference on Computer Vision and Pattern Recognition*, pages 2288-2295, 2012.

[19] N. Dalal et B. Triggs, Histograms of oriented gradients for human detection, *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886-893, 2005.

[20] K. He, X. Zhang, S. Ren et J. Sun, Deep residual learning for image recognition, *Conference on Computer Vision and Pattern Recognition*, pages 770-778, 2016.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li et L. Fei-Fei, Imagenet : A large-scale hierarchical image database, *Conference on Computer Vision and Pattern Recognition*, pages 248-255, 2009.