Détection et suivi d'objets mobiles par caméras fixes

Lionel Robinault

Université Lumière Lyon 2

Année 2021

Mémoire d'Habilitation à Diriger des Recherches

Présenté et soutenu publiquement par

Lionel Robinault

Le 09 juin 2021

Détection et suivi d'objets mobiles par caméras fixes

Préparée au sein du laboratoire LIRIS

Composition du jury

M. DARMONT Jérôme	Rapporteur	(Professeur des Universités)
M. CHATEAU Thierry	Rapporteur	(Professeur des Universités)
M. BREMOND François	Rapporteur	(Directeur de recherche)
Mme ACHARD Catherine	Examinatrice	(Professeure des Universités)
M. BOUWMANS Thierry	Examinateur	(Maître de conférences HDR)
Mme TOUGNE Laure	Garante	(Professeure des Universités)

Avant-propos

Ce manuscrit décrit mes activités de recherche et d'encadrement depuis mon arrivé au sein du Laboratoire LIRIS et de la création de la société Foxstream. Il se compose en deux parties.

La première partie présente mon curriculum vitae détaillé ainsi qu'une synthèse des travaux de recherche et d'encadrement. L'essentiel de mes recherches a été réalisé dans le cadre d'une société privée dont je suis l'un des cofondateurs. J'ai cependant gardé des liens étroits avec le laboratoire LIRIS de Lyon dont je suis membre associé depuis plus de 15 ans. Les canons de la recherches industrielles, et plus encore dans une PME, sont différents de ceux de la recherche académique notamment en ce qui concerne les publications. Cependant, au sein de la société Foxstream, nous avons fait le choix de ne pas hésiter à publier nos travaux avec le risque que nos concurrents puissent en bénéficier ou qu'ils aient connaissance de nos orientations.

La seconde partie présente une synthèse des méthodes de détection et de suivi des objets d'intérêt mobile dans un contexte spécifique de détection d'intrusions, c'est-à-dire avec un certain nombre de contraintes fortes, que nous détaillons par la suite. Ces contraintes nous amènent à faire différents compromis. Nous proposons un cadre théorique et un formalisme mathématique permettant de définir une détection d'intrusion à partir d'une représentation continue d'un monde en 3D + t via une représentation discontinue en 2D + t. Nous présentons rapidement les étapes en nous focalisant sur les deux aspects les plus importants à savoir la détection proprement dite et le suivi temporel des objets détectés. Pour ces deux étapes, nous présentons les différentes approches de l'état de l'art que nous avons dans la plupart des cas expérimentées et nous détaillons les principales difficultés que doivent surmonter les algorithmes mis en œuvre. Nous avons fait le choix, dans ce manuscrit, de ne présenter que les algorithmes basés sur des descripteurs ou plus généralement sur le traitement du signal. Ce choix est motivé par les recherches que nous avons menées ces quinze dernières années. Toutefois, pour les aspects détection et suivi, nous présentons quelques travaux basés sur l'apprentissage profond. Cette seconde partie s'intéresse également à la validation et à l'évaluation des algorithmes. Nous montrons les difficultés à proposer des métriques pertinentes en fonction de l'application visée. Nous présentons les évaluations pour chacune des étapes prise indépendamment ainsi que l'évaluation au plus haut niveau de la solution globale de détection d'intrusions.

Abstract

This manuscript describes my research and supervision activities since my arrival at the LIRIS Laboratory and the creation of the Foxstream company. It is composed of two parts.

The first part presents my detailed curriculum vitae as well as a summary of the research and supervision work. Most of my research was carried out within a private company of which I am one of the two founder. However, I have remained close relationships with the LIRIS laboratory in Lyon, of which I have been an associate member for more than 15 years.

The second part presents a synthesis of the methods of detection and monitoring of mobile objects of interest in a specific context of intrusion detection, that is to say with a certain number of strong constraints, we will detail below. These constraints lead to make various compromises. We propose a theoretical framework and a mathematical formalism allowing to define an intrusion detection from a continuous representation of a world in 3D + t *via* a discontinuous representation in 2D + t. We quickly present the steps by focusing on the two most important aspects, namely the detection itself and the temporal tracking of the detected objects. For these two steps, we present the different state-of-the-art approaches that we have in most cases experienced and we detail the main difficulties that the implemented algorithms must overcome. This second part is also interested in the validation and evaluation of algorithms. We show the difficulties in proposing relevant metrics according to the targeted application. We present the assessments for each of the steps taken independently as well as the highest level assessment of the overall intrusion detection solution.

We have chosen, in this manuscript, to present mainly algorithms based on descriptors or more generally on signal processing. However, with regard to the performances obtained by machine learning and in particular CNNs, there is a field of research that we cannot neglect and that we seriously consider. This choice is motivated by the research carried out over the past fifteen years which is based on an analysis that we can qualify as "descriptive" compared to a "predictive" analysis.

Descriptive analysis consists of defining a comprehensible mathematical model which describes the phenomenon to observe. This involves collecting data on a process, formulating hypotheses on the models and validating these hypotheses by comparing the result of the descriptive models with the actual result. The production of such models, however, is difficult and incomplete because there is always a risk of variables or phenomena that scientists and engineers neglect to include due to ignorance or inability to understand certain complex phenomena, hidden or not intuitive.

Predictive analysis involves the discovery of rules underlying a phenomenon and form a predictive model that minimizes the error between the actual outcome and the expected outcome considering all possible interfering factors. Machine learning rejects the paradigm of the programming descriptive traditional where the problem analysis is replaced by a process of training and the system is powered by a large number of examples known he learns and uses to calculate new models. Deep learning is a subset of machine learning based largely, today, on artificial neural networks and in the case of images, on convolutional neural networks (CNNs). This type of network achieves performance in most areas of data analysis applications. The demands of the video protection market are such that they cannot tolerate more than one false alarm per week per camera. With a typical scan rate of 5 frames per second

and assuming the scan is enabled twelve hours per day, this corresponds to an error for approximately one million thousand scanned images. This is the level of performance that we are currently achieving with our descriptive analysis algorithms. To manage a complex problem such as intrusion detection with such a low error rate, there is no other solution today than deep learning.

The term “video surveillance” or its more politically correct version “video protection” are generic terms which group the entire video processing chain in the context of the surveillance of goods and people. Among all the elements of this processing chain we can find the acquisition, broadcasting of video streams, visualization and recording. The majority of video protection installations are limited to these stages. In this type of installation, the video can be displayed live on one or more monitors so that an operator can visualize the scene and can act live. More generally, the video is simply recorded continuously to be consulted later in order to verify an event reported elsewhere. The real-time analysis of video streams, which is called "Intelligent Video Surveillance" to stand out from simple recorders, is ultimately little used. This analysis can be used in different contexts such as the public highway, a store, an airport or around a building.

As part of our work, we are mainly interested in intrusion detection algorithms and more specifically in perimeter detection. The algorithms we present are not reserved exclusively for this type of detection. The background modeling, which is one of the important steps, is used in several other fields. But in most cases, algorithms will need to be optimized for this task and use time constants appropriate to the reaction time needed to properly support detection.

Generally speaking, intrusion detection consists of analyzing normal activity and identifying abnormal or suspicious activities. This framework applies to video surveillance as well as to the surveillance of an information system. In the case of video surveillance, the purpose is to detect and report the presence or unauthorized penetration of certain classes of objects in an area during a period of time defined by the user. Its function may be to detect several types of intrusion such as crossing, approaching or even raiding if we consider the time dimension. The notion of unauthorized class can be quite difficult to define since it is not necessarily possible to predict clearly how the intrusion will occur: on foot, by bicycle, by car or other vehicle, to speak only of the "classic" intrusion detection in video protection. And even when it comes to people, standing is not necessarily the mode of travel adopted during the intrusion. As a result, a pedestrian detector assuming a standing person may not be able to correctly detect a crawling person. It is then more efficient to define the class or category of authorized objects such as birds or small mammals and to trigger an alarm on all the others, even if this list is extended depending on the site to be secured.

Perimeter detection is a special case of intrusion detection since it is deployed mainly on the periphery or on the outline of a building and aims to detect an intrusion upstream of the area to be protected.

In a fairly general way, an algorithm for detecting moving objects in a video stream acquired by a fixed camera comprises two important steps: the phase of extracting the objects of interest and the temporal tracking of these objects. Pre-processing steps can be placed upstream of the chain to change the color space, resize the image or smooth the acquisition noise. Similarly, post-processing steps can be add ed downstream to remove objects that will fit not under detection.

The detection of moving objects in a video is one of the first steps of an algorithm in many applications. It is particularly well suited to segmenting objects of interest when they are in relative motion with respect to the background of the scene. This detection is based either on motion detection or on motion-based segmentation. The distinction is a bit subtle, but detection aims to decide which pixels or groups of pixels belong to moving objects while motion-based segmentation applies to partitioning the image into regions with common displacement characteristics. Detection produces a bit map indicating the presence or absence of motion while segmentation produces a multi-tag map.

There are three main families of approaches, which are commonly found in the literature and which allow the discovery of moving objects in a fixed scene:

- Optical flow detection: optical flow characterizes the apparent displacement of the intensity of a pixel in an image caused by the relative movement of objects in relation to the scene.
- Block matching detection: this approach is based on matching regions from one image to another.
- Detection by background model: the principle of this approach is to build and, in most cases, to update a model characterizing the pixels belonging to the scene devoid of any object of interest.

We can also add the case of specialized detectors. This class of algorithm does not use the notion of motion or video and generally does not keep any memory of previous detections. This approach can effectively detect classes of special objects including pedestrians. Indeed, when the objects of interest are known *a priori*, it is possible to use a detector constructed or trained specifically for this task. In this type of approach, the image is generally traversed by a sliding window in which the detector is applied. The presence or absence of the object depends on the response of the detector. Recent work now makes it possible to analyze the whole image in a single iteration and automatically extract a set of bounding boxes.

As part of intrusion detection, the class of the most popular algorithm is modeling the background. The general principle of these approaches is to characterize, in a time window, the evolution or more generally the distribution of the colorimetric components of each pixel of the image. The idea is thus to be able to delimit the space or the colorimetric spaces corresponding to the scene devoid of any object of interest. These color spaces correspond to what is commonly referred to as the background. If for a given pixel, the value of its colorimetric components at time t does not belong to any of the subspaces of the scene, that is to say to the background, then it is considered to be part from the foreground.

A common approach to modeling the background is to assume that the samples of the background that compose it are generated by a random variable and therefore corresponding to a given probability density function. It is then sufficient to estimate the parameters of the density function to determine whether a new sample belongs to the same distribution.

Even in the simplest cases, the notion of a fixed background is open to debate. The scene itself can be fixed but with more or less dynamic parts such as wind-driven tree branches, reflections on the surface of the water, or even rain or snow even if this case is less common. Moreover, the term “fixed” is not necessarily the most suitable for talking about the background. What we mean by fixed (or movement in opposition) implies that the colorimetric components of a pixel do not change significantly over time. However, even in the case of a fixed scene in which there is no moving object, the colorimetric components of the pixels can still evolve globally (change in camera gain) or locally (change in camera gain). brightness, shadow cast by an out-of-view

object, *etc.).* The “fixed bottom / moving object” approximation is therefore a semantic shortcut which is not always true in practice. However, it is still commonly used in the community.

Moreover, the objects of interest, as we define them, are not necessarily in motion during the whole time sequence is studied. For instance, this will be the case when a vehicle is parked in a parking space, or on the contrary, leaves its place. This relatively simple case already poses a certain number of problems when it comes to updating the model.

Tracking is another essential element in the processing chain leading to the identification of an action or behavior in a video. In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in the image plane as the object moves in a scene. In other words, a tracking algorithm works to assign a unique tag to each tracked object in different frames of a video. In a video protection context, the task of a tracking algorithm therefore consists in associating each blob detected at time t with the appropriate tracked object at time t-1, so as to preserve the identity of the real world objects through the video footage. During the process, the algorithm should also create new object models and update existing models as necessary. To summarize, we can thus define tracking as: the estimation over time of the state of several moving objects using an observation set.

The literature distinguishes two main categories of tracking methods that we will find under the acronyms VOT for “Visual Object Tracking” and MOT for “Multi Object Tracking”. The first category gathers the algorithms is not the objective is to follow a single target in fact indeed the assumption that the vector of state of the object to follow is known at the moment t_0 and does not require detector (segmentation or specialized detector). The tracking process then consists of looking for the region of the image that minimizes the distance to the target's signature. The second category is more general and ultimately includes all the other algorithms that do not fit into the first category. However, it mainly integrates the methods referring to the association of data. As part of our work, we were particularly interested in the problem of multi-object tracking since, with a preliminary modeling / segmentation step of the foreground, we have at our disposal a set of blobs detected at time t.

Just like the detection step, an efficient tracking algorithm must be able to take into account a set of elements likely to disrupt its operation. These elements can be linked to the context of the scene, to the nature of the objects, to their evolution in the scene but also to the performance of the detection phase. In the latter case, the nature of false detections such as total or partial omissions will not necessarily be identical.

The way of approaching the follow-up strongly depends on the constraints related to the targeted application. Most of our work is mainly focused on real-time intrusion detection. A first constraint that we have to face is ???? that we cannot use the whole sequence or the video but only a few images preceding the instant t. The second constraint is that we cannot determine the number of targets to track. However, since we use background modeling and segmentation to extract foreground blobs, we have more freedom over the object's state vector and are not limited to just bounding boxes.

Under these conditions, the general iterative monitoring process breaks down for each new image as follows :

- **Prediction / propagation:** this first step is based on the estimated change of the object state vector from the previous observations. This makes it possible to limit the search space.

- **Matching:** this second step consists in evaluating the correspondence between the estimated state vector in the search space and updating it according to the observations.

In the particular case where we have a set of targets at time t-1 and a set of detections at time t, the tracking then turns into an assignment problem. This principle is mainly used in the case of intrusion detection based on the modeling of a background image due to the fact that the sets of tracked objects and of detected blobs comprise relatively few elements and therefore that the space of search is reduced.

In order to set up an effective tracking algorithm, the following elements should be precisely defined:

- **A state model** that corresponds to the representation of the tracked objects. This state model describes the object on the basis of information obtained from the image but can also include a kinematic component making it possible to introduce the speed and possibly the acceleration.
- **A dynamic model** which makes it possible to mainly characterize the movement of the object in order to limit the search space but which can also take into account the evolution of the state vector as a whole.
- **A distance measure** that allows to evaluate the correspondence of the tracked objects.
- **An estimation strategy** which allows, from observations and the dynamic model, to infer future states.

The first two elements refer to the prediction / propagation phase and the next two to the matching phase.

Finally, we bring attention to the evaluation of algorithms. Evaluation is an essential step in any development and must be studied with the greatest care to ensure that the result of this evaluation is as relevant as possible. The quality and accuracy of the data set is essential. The case of video protection, as we defined it, in this particular that when a system is deployed, there is a high probability that there will be no real detection during the entire exploitation of the system. It is therefore important, during the evaluation, to ensure that the system is able to detect an intrusion. This first check is relatively simple to perform. However, it is also necessary to verify that the system is robust to a significant amount of variation and unwanted movement. This second check is more difficult to perform with a relatively small data set. The dataset must therefore be sufficiently representative. Equally important is the quality and accuracy of the reference. Care must also be taken with the metrics used. These metrics must be in line with the objective.

Table des matières

I - Synthèse des activités de recherche

1 Curriculum Vitae

1.1 Statut actuel

Depuis 2003	**Responsable Recherche** à Foxstream SA (co-fondateur) et **Chercheur Associé** au Laboratoire LIRIS

1.2 Titres universitaires

2009	**Doctorat informatique et image**, Ecole doctorale Info-Math, Université Lumière Lyon 2, Laboratoire LIRIS UMR 5205 **Sujet :** Mosaïque d'images multi-résolution et applications
2005	**Master recherche** IGI (Informatique Graphique et Image) Université Lumière Lyon 2 Analyse d'images (acquisition, compression, représentation, indexation, reconnaissance) et synthèse d'images (rendu, photoréalisme, réalité virtuelle, réalité augmentée)
1994	**DESS EEA** (Electronique, Electrotechnique, Automatique) Université Claude Bernard Lyon1, spécialité Micro-Electronique Conception microélectronique (numérique et analogique), Physique des matériaux semi-conducteur et technologie des dispositifs ainsi que des microsystèmes
1992	**Maîtrise EEA** (Electronique, Electrotechnique, Automatique) Université Claude Bernard Lyon1

Acquisition des connaissances fondamentales en électronique (numérique, analogique et RF), physique pour l'électronique, automatique, traitement du signal et architecture des circuits.

1991

Licence EEA (Electronique, Electrotechnique, Automatique) Université Claude Bernard Lyon1

1990

Deug A' Université de Savoie Chambéry

Acquisition des connaissances fondamentales en mathématique, physique, chimie, économie, informatique et électronique.

1.3 Expériences professionnelles

Depuis 2003

Responsable Recherche et Développement et co-fondateur à Foxstream SA

Recherche en vision par ordinateur, analyse d'images, détection et identification des objets en mouvement en temps réel.

Co-encadrement de thèses, modélisation de la scène, détection d'événements rares, détection des dépendances temporelles.

Management d'une équipe d'ingénieurs et de chercheurs, mise en œuvre méthode SCRUM, définition des orientations et gestion de planning, veille technologique.

1998 à 2003

Chef de Projet chez Prodys Equipement puis Norack (co-fondateur)

Réalisation de machines spéciales : logiciels de pilotage, suivi mécanique, schémas électrotechniques, étude et réalisation des cartes électroniques, mesure, mise au point.

Étude et rédaction des devis Consultation et suivi des fournisseurs et sous-traitants

1994 à 1998

Responsable assurance qualité : rédaction et mise en place des procédures qualité, certification ISO 9001 approuvée par l'AFAQ en février 1998.

Ingénieur d'étude et développement électronique et informatique : transmetteur de mesure, console de programmation

1.4 Synthèse des activités de recherche

1.4.1 Axes de recherche actuels

- Modélisation des composantes statiques dans une séquence vidéo
    - Détection automatique et apprentissage profond
- Recherche des dépendances temporelles dans un flux de données hétérogène

1.4.2 Co-encadrements de thèse

- **Apprentissage profond non supervisé de représentations spatio-temporelles pour la vidéo**
  Doctorant : Lohani Devashish
  Encadrants : Pr. Laure Tougne, MCF Carlos Crispim Junior, Dr. Lionel Robinault
  Soutenance : prévue fin 2022

- **Détection des relations spatio-temporelles dans un système de capteurs hétérogènes**
  Doctorant : Amine El Ouassouli
  Encadrants : MCF – HDR Marian Scuturici, Dr. Lionel Robinault
  Soutenance : Le 24 Juillet 2020

- **Utilisation du contexte pour la détection et le suivi d'objets en vidéosurveillance**.
  Doctorant : Matthieu Roger
  Encadrants : Pr. Laure Tougne, Dr. Lionel Robinault
  Soutenance : Le 09 juin 2015

- **Segmentation spatio-temporelle et classification « one-class » pour la détection d'intrusion**
  Encadrants : Pr. Laure Tougne, Mcf. Emmanuelle Reynaud, Dr. Lionel Robinault
  Soutenance : Le doctorant a interrompu sa thèse pour rejoindre son épouse à Bordeaux

1.4.3 Thèse de doctorat

Sujet Mosaïque d'images multi-résolution et applications
Laboratoire LIRIS, UMR5205 – Université Lumière Lyon 2
Adresse 5, Av. Pierre Mendes-France – 69676 Bron Cedex
Encadrants Pr. Serge Miguet et MCF. Stéphane Bres
Soutenance 08/09/2009
Jury

Rapporteurs :
Mme Michèle Rombaut Professeur des Universités,
Mme Nicole Vincent Professeur des Universités,

Examinateurs :
M. Thierry Chateau Maître de conférences,
M. Jean-Baptiste Ducatez Ingénieur,

Président :

M. Patrick Perez	Directeur de recherche INRIA,

Directeurs :

M. Serge Miguet	Professeur des Universités,
M. Stéphane Bres	Maître de conférences.

Résumé : Notre travail de thèse s'est articulé autour de l'utilisation de caméras motorisées qui, dans le langage courant, sont appelées caméras PTZ. La première partie de notre travail de recherche a été consacrée à la construction et au rendu d'un panorama. Un panorama est une représentation étendue d'une scène construite à partir d'une collection d'image. Nous avons présenté différents modèles de représentations de ces panoramas ainsi qu'une optimisation du parcours de la scène observée permettant de limiter le nombre de prises de vue. Nous avons présenté également différents traitements permettant d'améliorer sensiblement le rendu et de corriger la plupart des défauts liés à l'assemblage d'une collection d'images acquises avec des paramètres de prise de vue différents. La deuxième partie a été consacrée aux applications de détection et de suivi d'objets en mouvement avec une caméra PTZ seule ou en association avec une caméra omnidirectionnelle. Par rapport aux applications classiques de suivi de cible en mouvement avec une caméra PTZ, notre approche se différencie par le fait que nous réalisons une segmentation fine des objets permettant leur classification.

1.4.4 Publications :

- 3 articles dans des revues internationales
- 1 édition scientifique d'ouvrage
- 12 communications à des colloques internationaux
- 3 communications à des colloques nationaux

1.5 Synthèse de l'activité d'enseignement

2003 - 2005	Tice, Cursus L1 - Lyon2, Volume 120h
Depuis 2012	Séminaire industriel "métiers de la R&D" - Insa 5IF & 5TC

2 Recherches académiques

2.1 Stage Master Recherche (2005)

La recherche que nous avons menée au sein du laboratoire LIRIS, en coopération avec la société Foxstream a eu pour but d'analyser et de décrire les objets en mouvement dans une séquence vidéo. Cette étude a eu pour fil directeur, l'analyse du comportement des souris de laboratoire soumises à un stress. Ce stage s'est déroulé en deux temps. Dans un premier temps, nous avons

étudié deux grandes approches utilisées dans la littérature pour estimer le mouvement des objets dans une image : le calcul du flot optique qui consiste à estimer le mouvement de chaque pixel de l'image ou de chaque groupe de pixels et une approche plus globale qui consiste à estimer, par une segmentation, le mouvement d'une ou de plusieurs régions de l'image. Nous avons orienté notre recherche sur la deuxième méthode de façon à extraire les contours des objets en mouvement. Cette segmentation spatio-temporelle a été réalisée à partir d'une méthode probabiliste classique basée sur les mélanges de gaussiennes à laquelle nous avons apporté quelques améliorations. Notre principale contribution a été de conditionner la mise à jour des distributions en fonction du résultat de la segmentation précédente. Ceci nous a permis de contrôler la prise en compte des objets en mouvement dans l'image et notamment lorsque ces objets restent un long moment immobile. A partir de l'extraction des points du contour de chaque objet, nous avons calculé un vecteur d'état. Ce vecteur d'état étant composé essentiellement de différents paramètres calculés à partir des moments et des premiers descripteurs de la transformée de Fourier. Ce vecteur d'état et le calcul de sa dérivée par différence finie, nous a permis d'améliorer le suivi des objets sur plusieurs images successives même en cas d'occultation et de fusion entre objets.

La deuxième partie du stage a été consacrée plus précisément à l'étude du comportement des souris soumises à un stress. Les descripteurs que nous avons calculés à partir des vecteurs d'état des objets nous ont permis de déterminer de façon fiable le moment de l'impulsion électrique, les phases d'immobilité et les phases d'enfouissement. Les résultats ont ensuite été présentés sous la forme d'un chronogramme afin d'obtenir une vision globale de l'activité du cobaye. Un exemple de résultat est donné ci-dessous (Figure 1). Le premier chronogramme correspond l'étiquetage manuel d'une vidéo réalisé par un expert. Le second a été tracé à partir des descripteurs que nous avons proposés. Ces chronogrammes se lisent de la gauche vers la droite. Le premier trait correspond à la stimulation électrique. Les traits bleus (ou foncés sur impression en niveau de gris) correspondent aux phases d'immobilité et les traits verts (ou clairs) correspondent aux phases d'enfouissement. Le taux de fiabilité de nos descripteurs était supérieur à 85%, ce qui était très largement suffisant pour l'application visée.

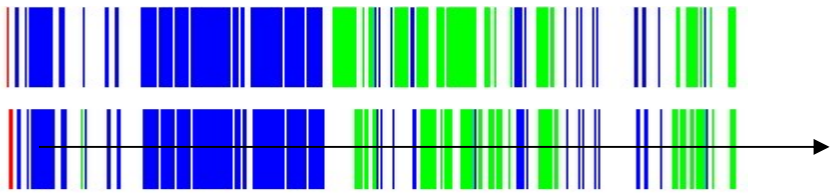


Figure 1 : chronogrammes du comportement d'une souris soumise à un stress

2.2 Travaux de thèse

La formation et l'exploitation d'images panoramiques ont des implications importantes dans plusieurs domaines comme la robotique, la vision par ordinateur et la réalité virtuelle. Dans ce travail de thèse, financé par la société Foxstream et que nous avons réalisé au sein de l'UMR LIRIS, nous avons étudié les différentes étapes qui entrent dans le processus de réalisation et d'exploitation d'une image panoramique : l'acquisition, la projection des prises de vue dans

l'image panoramique (Figure 2), l'amélioration du rendu, la visualisation et les applications de détection et de suivi des objets en mouvement dans une scène.

Figure 2 : exemple de représentation panoramique (360° suivant le plan horizontal et 120° sur le plan vertical)

L'utilisation d'une caméra « Pan-Tilt-Zoom » (PTZ) permet de contrôler interactivement la prise de vue. L'un des intérêts des caméras PTZ est que leur centre optique est relativement peu éloigné de leur centre de rotation. Ceci permet, sous certaines conditions, de considérer que l'ensemble de la scène est acquis à partir d'un centre de projection unique simplifiant ainsi la transformation mathématique. Toutefois, il n'est pas possible d'obtenir une notion de la profondeur même en multipliant les prises des vues pour des angles différents. Puisqu'aucune notion de profondeur n'est possible dans ces conditions, nous pouvons donc considérer que tous les points sont à la même distance du centre c'est-à-dire sur la surface d'une sphère de rayon inconnu. Notre problématique est donc de parcourir l'ensemble des points d'une sphère à partir de la projection d'images rectangulaires. Nous avons réalisé une étude théorique de la projection des bords des images dans une sphère. A partir de ces équations, dont un exemple est donnée ci-après (Figure 3), nous avons proposé un parcours de la sphère permettant de minimiser le nombre de prises de vue [1].

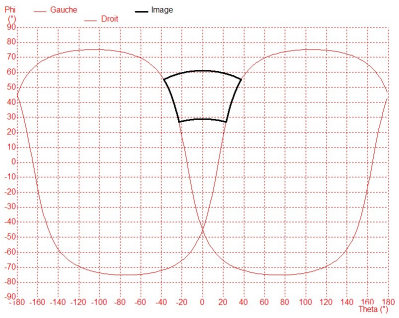


Figure 3 : représentation graphique des équations des bords droit et gauche d'une image

Dans la mesure où les images utilisées pour la construction du panorama ne sont pas acquises simultanément, nous avions plusieurs problèmes à résoudre afin d'améliorer le rendu : correction de la luminosité, suppression des fantômes (objets en mouvement dans la scène pendant les prises de vue), défauts d'alignement et surtout recalage des images. En effet, idéalement une caméra PTZ devrait nous permettre de disposer des paramètres exacts de la prise de vue. Dans la réalité, ce n'est pas forcément le cas, d'autant qu'avec certains types de matériel, il n'est tout simplement pas possible de disposer de ces paramètres. Avec d'autres caméras, comme celle que nous avons utilisé, la précision de la commande n'était pas suffisante

pour obtenir un recalage robuste avec simplement les données de la caméra. Le but du recalage d'images est d'aligner géométriquement deux images ou plus, de sorte que des pixels respectifs ou leurs dérivés (bords, coins, *etc.*) représentant la même structure fondamentale puissent être mis en correspondance. L'idée sous-jacente est de mettre en correspondance les images selon leurs propriétés radiométriques ou géométriques en utilisant une fonction spécifique permettant d'évaluer la qualité de la mise en correspondance. Dans un premier temps, nous avons testé plusieurs méthodes décrites dans la littérature. N'ayant pas obtenu des résultats conformes à nos attentes, nous avons commencé par étudier plus précisément les mesures de similarité (Figure 4) puis nous avons proposé une méthode de recalage à la fois robuste et rapide [2].

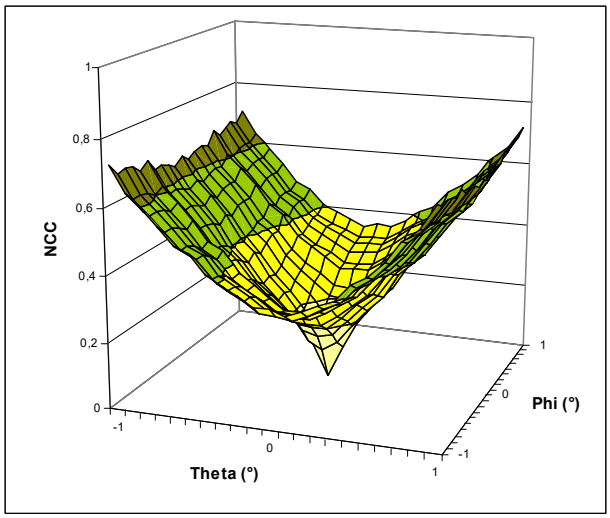


Figure 4 : étude du comportement de la mesure de corrélation croisée normalisée dans l'espace de recherche

Nous avons ensuite développé plusieurs applications de suivi et de classification d'objets en mouvement. Avant d'aborder le problème du suivi, nous avons eu à résoudre celui de la détection des objets en mouvement. Dans une activité de vidéo surveillance, l'intérêt des caméras PTZ est de permettre une vision très élargie du périmètre à surveiller. Leur principal défaut est que lorsque la caméra est orientée dans une direction, elle ne permet pas de voir ou de surveiller le reste de la scène. Plusieurs auteurs utilisent dans ce cas deux caméras. Une caméra fixe avec un objectif grand angle ou de type fish-eyes ou encore une caméra associée à un miroir sphérique ou parabolique et une caméra PTZ asservie par la première. L'apport de notre technologie est de proposer une solution basée sur l'utilisation d'une seule caméra PTZ associée à un miroir sphérique [3]. En position de repos, la caméra vise le miroir sphérique (Figure 5). La vision totale et permanente de l'espace permet d'assurer qu'un événement isolé sera automatiquement détecté. A partir de l'analyse des pixels en « mouvement » dans cette image omnidirectionnelle, il nous est possible de déterminer la position dans l'espace d'un objet d'intérêt et de piloter la caméra en conséquence. Dans le cadre de cette première application, nous avons proposé une calibration automatique de l'ensemble Camera/Miroir permettant de faciliter la mise en œuvre de notre solution.

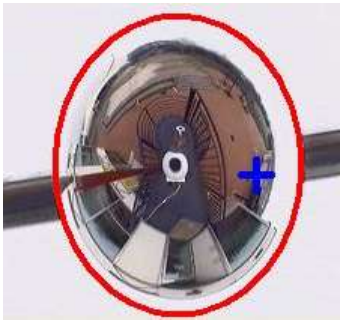


Figure 5 : visualisation du miroir en position repos

Cette première phase de détection et de focalisation terminée, nous nous sommes intéressés à la segmentation des objets en mouvement et à leur suivi. Une segmentation fine des objets permet une classification de ces objets à partir de caractéristiques extraites de leur contour. La plupart des algorithmes de segmentation d'objet en mouvement ont été développés pour les caméras fixes et ne peuvent pas être utilisés directement sur des caméras PTZ. Parmi ces algorithmes de détection, nous nous sommes intéressés plus particulièrement aux algorithmes basés sur la modélisation d'une image de fond ne comportant que les composantes statiques de la scène. Parmi ces algorithmes, les mélanges de gaussiennes ont prouvé leur efficacité. Nous avons proposé une généralisation de ces algorithmes, appliquée aux caméras PTZ en mouvement. Une fois le modèle de fond projeté sur le plan de l'image courante, il peut être utilisé comme un modèle de caméra fixe. Nous avons ainsi pu utiliser les méthodes de segmentation, de suivi et d'identification décrites dans la littérature pour les caméras fixes. Cette généralisation nous a permis d'obtenir de bons résultats pour le suivi des personnes dans une scène en mouvement [3].

3 Recherches industrielles et encadrements

3.1 Contexte

En octobre 2003, j'ai rejoint le groupe de fondateur de la startup Foxstream, qui sera officiellement crée en juin 2004, afin de prendre en charge le développement des fonctionnalités d'analyse vidéo. La startup était alors incubée par le laboratoire LIRIS dans les locaux de l'université Lyon 2 à Bron. C'est donc tout naturellement que j'ai rejoint le laboratoire. Avec le développement de l'activité de l'entreprise, j'ai progressivement eu la charge du management de l'équipe de recherche et développement tout en gardant une forte implication dans le laboratoire dont je suis toujours chercheur associé. Dix-sept ans plus tard, le groupe Foxstream est composé de deux entités et regroupe plus d'une trentaine de personnes. Foxstream est présent essentiellement sur le marché de la vidéo protection en extérieur et dans le domaine bancaire.

Les problématiques liées à l'analyse vidéo en extérieur sont nombreuses. La plupart des algorithmes que nous avons développés au sein de la société sont basés sur la segmentation des objets d'avant plan. J'ai donc été amené à étudier et proposer différents algorithmes de modélisation des composantes statique de la scène ou plus exactement proposer des modèles caractérisant l'évolution des composantes colorimétrique des pixels correspondant à la scène

dénuée d'objet d'intérêt. Ces travaux de recherche ont donné lieu cinq thèses soutenues, une quinzaine de publications et l'organisation d'un challenge en collaboration avec des collègues de l'université de Clermont Ferrand.

La segmentation des objets d'avant plan n'est qu'une première étape. Afin de proposer des fonctionnalités d'analyses complètes, de la capture du flux jusqu'à la détection des situations d'alarmes, j'ai été amené à étudier et à proposer des solutions sur l'ensemble de la chaîne de traitement : prédiction et suppression des ombres, utilisation du contexte, suivi des objets, détection et reconnaissance des plaques minéralogiques.

Ces dernières années, les performances en apprentissage profond en lien avec l'accélération des calculs nous ont amené à nous intéresser fortement à ces problématiques. Nous avons réalisé un premier travail exploratoire avec la proposition d'une solution d'indexation vidéo basée sur l'utilisation d'un réseau profond. Ce travail a donné lieu à une publication [4]. Fort de cette première incursion réussie dans le monde de l'apprentissage profond, je co-encadre actuellement une thèse sur l'apprentissage non supervisé de représentations spatio-temporelles pour la vidéo.

Par ailleurs, j'ai régulièrement et activement participé à des manifestations proposées par le laboratoire : séminaires, démonstrations, journées des doctorants, rencontre laboratoire / entreprise, *etc.* Dans le cadre de mon implication dans le laboratoire, j'ai participé à la rédaction de plusieurs projets ANR ou région. J'ai été sollicité comme relecteur en vue de la préparation de différentes conférences nationales et internationales. Je suis également membre de l'association ANDES et je suis intervenu lors des journées de promotion du doctorat organisées par l'association.

Depuis 2012, je participe à deux séminaires proposés par l'INSA aux étudiants de 5ième année. Ces séminaires ont pour objet l'univers des startups et la recherche dans les PME.

3.2 Recherche en cours

Je travaille actuellement sur deux projets de recherche. Le premier est un projet de recherche bénéficiant d'un financement de la région Auvergne-Rhône-Alpes en collaboration avec le laboratoire LIRIS, l'IFSTAR et la société Elistair. Ce projet vise à proposer une solution de monitoring du trafic routier à l'aide d'un drone captif. Le deuxième est intégralement financé par la société Foxstream. Dans ce projet, nous exploitons les récentes avancées des techniques d'apprentissage profond que nous appliquons dans un contexte bancaire pour détecter et analyser les files d'attentes, prévenir les incivilités, détecter les comportements anormaux près des distributeurs de billets, *etc*. Nous nous intéressons particulièrement la classification du genre et de la classe d'âge des personnes.

3.2.1 Projet Station'air (depuis 2017)

Ce projet vise à proposer une solution de monitoring du trafic routier à l'aide d'un drone captif. Equipé d'une caméra vidéo embarquée, le drone est relié par un câble à une station au sol. Le rôle de ce câble est triple. Il permet principalement de rendre le drone captif et donc de limiter

les contraintes liées à son déploiement (gestion de la sécurité, procédures administratives, *etc*.) mais il permet également de fournir une alimentation continue et de récupérer le flux vidéo en temps réel. Un drone, ainsi équipé, peut effectuer un vol statique non intrusif de plusieurs heures, positionné à un observatoire stratégique. Dans cette configuration, le système peut assurer une surveillance continue jusqu'à 70 m de haut. Pour la sécurité, une zone, d'un rayon de 70 m est délimitée sous le drone pour se tenir à l'écart des zones habitées et prévenir tout danger. En collaboration avec le laboratoire LIRIS, nous nous intéressons principalement à la stabilisation, l'extraction des trajectoires et la catégorisation des véhicules à partir du flux vidéo.

D'un point de vue traitement d'images, la problématique considérée dans le cadre de ce projet est celle du suivi multi-objets (objets posés au sol) à l'aide d'une caméra embarquée sur le drone. Une classification des véhicules en quelques classes (véhicules légers, véhicules intermédiaires, poids lourds, *etc*.) est également nécessaire.

Les difficultés inhérentes au suivi multi-objets sont traditionnellement les suivantes :

- pas d'hypothèse sur le nombre d'objets à suivre, ce qui implique la gestion des entrées-sorties des objets suivis dans la scène ;
- gestion des changements d'apparence / de taille dus à l'éloignement et à la position des objets par rapport à la caméra ;
- gestion des occultations des objets les uns par rapport aux autres et éventuellement par rapport à des éléments fixes de la scène ;
- gestion des ambiguïtés résultant du suivi d'objets similaires (deux véhicules de la même couleur par exemple) ;
- qualité des détections (omissions ou fausses détections).

La prise d'altitude avec l'aide du drone permet d'amoindrir un certain nombre de difficultés rencontrées dans le cas d'une caméra de vidéosurveillance classique comme la gestion des occultations, les changements d'apparence, de taille, etc. En revanche, de nouvelles difficultés, et donc de nouveaux verrous scientifiques, apparaissent parmi lesquels :

- Image non fixe due au mouvement du drone, même si on peut supposer que le mouvement sera relativement faible [5];
- Des objets plus petits à détecter, à suivre et à catégoriser.

Dans la littérature, les travaux s'intéressant au suivi des objets à l'aide de drone restent encore très minoritaires et sont récents. On peut par exemple citer l'article [6] dans lequel les auteurs s'intéressent à des portions de route extrêmement petites, ce qui ne correspondent pas au contexte de notre projet dans lequel, au contraire, nous cherchons à caractériser la circulation sur des zones importantes comme des nœuds. Dans [7], les auteurs ont pour objectif le suivi d'un objet en mouvement à l'aide d'un drone autonome dans des conditions bien différentes des nôtres (intérieur).

Dans le cadre de ce projet nous avons à traiter deux situations très différentes : les intersections plus ou moins complexes dont la taille est inférieure à 100m et les nœuds autoroutiers qui peuvent s'étendre sur 1km^2. Dans le premier cas la taille apparente des véhicules dans l'image est suffisante pour utiliser des détecteurs monolithiques classiques (haar, HOG, CNN). Ceci

permet d'extraire et de classer efficacement les différents véhicules. Dans le cas des nœuds plus importants, même avec des caméras à haute résolution, la taille apparente des véhicules dans l'arrière-plan de l'image ne dépassera pas quelques pixels. Dans ces conditions l'utilisation des détecteurs monolithiques n'est pas envisageable. Nous devrons donc utiliser des méthodes de segmentations basée sur une modélisation des composantes statiques de la scène. Cela suppose donc l'utilisation d'algorithmes de recalage extrêmement précis et d'un modèle de fond robuste aux congestions. Ce sont, les deux grands défis de ce projet.

3.2.2 Classification du genre et de la tranche d'âge (depuis 2019)

La classification du genre et de la tranche d'âge des personnes sur des images de visages est un sujet difficile qui a été largement étudié ces dernières années et dont les applications sont nombreuses : indexation de vidéos, recherche d'informations, applications commerciales. En 2013, nous avions déjà travaillé le sujet de façon plus classique: détection du visage, pré-traitement, extraction de descripteur de Weber spatial [8], classification.

Si la classification du genre et de la classe d'âge est un sujet aussi difficile, c'est qu'elle est soumise à de nombreuses contraintes. En plus de la variabilité importante des visages humains (couleur de peau, taille, forme), et de la présence ou non de caractéristiques particulières comme les cheveux, les lunettes, la moustache, il faut également tenir compte de la prise de vue de la caméra, de la luminosité, de l'inclinaison des visages ou de l'occultation par d'autres objets. Il est donc presque impossible d'avoir des résultats parfaits atteignant 100% de bonne détection et classification, même si de nombreuses techniques permettent de pallier ces limites.

L'utilisation ces dernières années des techniques d'apprentissage profond, rendues possible par la très grande capacité de calcul des cartes graphiques, a permis d'améliorer significativement les résultats. Nous avons donc commencé par étudier les architectures d'apprentissage profond existantes puis nous les avons appliquées dans le cadre de la classification du genre et de la classe d'âge. Ce travail nous a permis d'améliorer significativement nos résultats de 2013 puisque sur la classification du genre nous sommes passés de 72% de bonne détection à plus de 90%. Ces résultats encourageant nous poussent à étudier les couches de convolutions intermédiaires afin de tester la possibilité de faire de la ré-identification des personnes.

3.3 Travaux de recherches récents

Je présente ici, deux travaux de recherches majeurs conduits ces trois dernières années. Le premier travail a été entièrement financé par Foxstream et vise à proposer une solution permettant d'estimer le temps d'attente principalement aux postes aux frontières des aéroports. Le deuxième travail a été conduit dans le cadre d'un projet de recherche collaboratif financé en partie grâce aux Fonds Unique Interministériel en partenariat avec Aximum, l'IFSTAR, Genesys et le CERAMA.

3.3.1 Calcul du temps d'attente (2016-2020)

Les contraintes de capacité, l'augmentation du trafic aérien et la concurrence entre aéroports internationaux stimulent l'investissement des infrastructures aéroportuaires. Mettre en place

dans les aéroports des solutions technologiques avancées est essentiel pour maximiser l'efficacité opérationnelle, améliorer l'expérience client et la qualité des services proposés par les compagnies aériennes. Les aéroports se repositionnent dans la chaîne de valeur en devenant le centre de l'excellence opérationnelle des acteurs du métier visant à satisfaire le voyageur. Les aéroports fixent de nouveaux indicateurs opérationnels et commerciaux centrés sur « l'expérience client » en vue de limiter les temps d'attente dans divers points stratégiques, d'améliorer la ponctualité des compagnies aériennes, de diminuer les délais et d'augmenter les revenus des activités non aéronautiques.

La principale problématique dans le transport aérien est le respect des horaires. Tout retard a des répercussions énormes en termes de gestion du trafic et bien sûr en termes de coût. Une meilleure gestion des files d'attentes permet de limiter une source de retard. L'amélioration de la gestion des files d'attentes passe entre-autre par l'information auprès des voyageurs. L'information permet de diminuer le stress des voyageurs et leur permet de mieux tolérer l'attente. L'information est également utile pour les opérationnels afin d'ouvrir ou de fermer des comptoirs d'enregistrement et ainsi optimiser la ressource en personnel tout en fluidifiant le trafic. Cependant, il est plus efficace de mettre en place les moyens le plus en amont possible dès l'apparition des premiers signes d'engorgement ou d'allongement de l'attente.

Dans le cadre de ce projet, notre activité de recherche a consisté tout d'abord à détecter les personnes dans une zone étendue à partir d'un réseau de caméra et de détecter automatiquement les files d'attentes de façon à pourvoir informer individuellement le temps d'attente au sein de chaque file. Nous avons finalisé la première partie (Figure 6) avec les fonds propres de l'entreprise et nous avons étudié la deuxième à travers une thèse CIFRE [9].

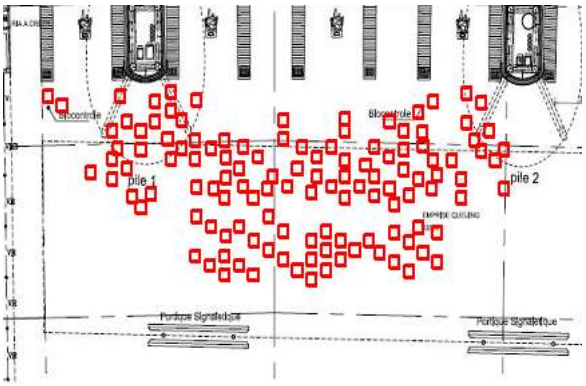


Figure 6 : projection sur un plan des positions des personnes détectées à partir d'un réseau de caméras

Le principe général de la détection des personnes consiste à projeter une carte de mouvement 3D de chaque caméra dans un plan pour différente valeur de Z. Les recouvrements entre caméras sont gérés en utilisant l'algorithme du chemin le plus court. C'est-à-dire que pour un point du plan vu par deux caméras, c'est sur la caméra la plus proche du point qu'est récupérée l'information de mouvement.

Cette solution nécessite 3 grandes étapes :

- Une calibration 3D de la caméra permettant de projeter la carte de mouvements sur un plan pour une valeur de Z donnée,

- L'estimation du mouvement vu par chaque caméra (Figure 7),
- L'estimation du nombre total de personnes dans la scène.

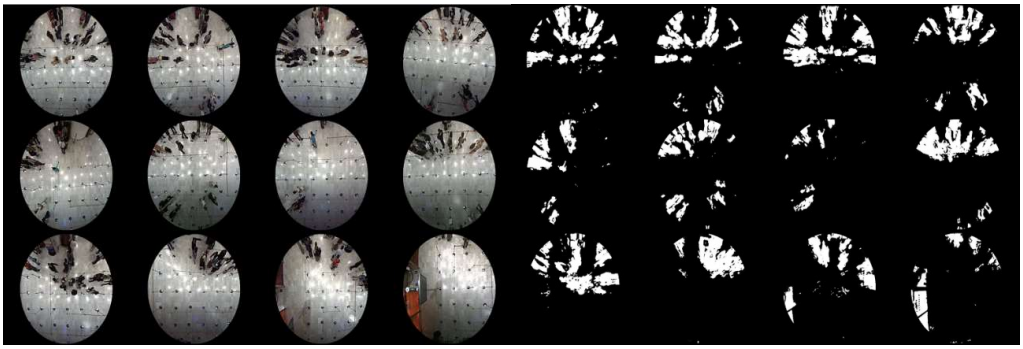


Figure 7 : exemple d'images issues de plusieurs caméras et des masques de mouvements associés

La calibration 3D est l'étape la plus délicate. Elle permet de faire correspondre un pixel d'une caméra à un point de l'espace 3D. D'un point de vue mathématique, c'est une étape assez simple puisqu'il s'agit de déterminer une matrice de projection. La petite difficulté est la mise en œuvre de cette calibration que nous avons décomposée en plusieurs étapes :

- Correction de la distorsion liée à l'objectif des caméras
- Définition du plan de projection et sélection de points clés
- Estimation de l'homographie entre différentes caméras
- Calcul d'une mosaïque d'images (Figure 8)
- Projection de la mosaïque sur le plan (calibration 2D de la scène)
- Calibration Z de chaque caméra.

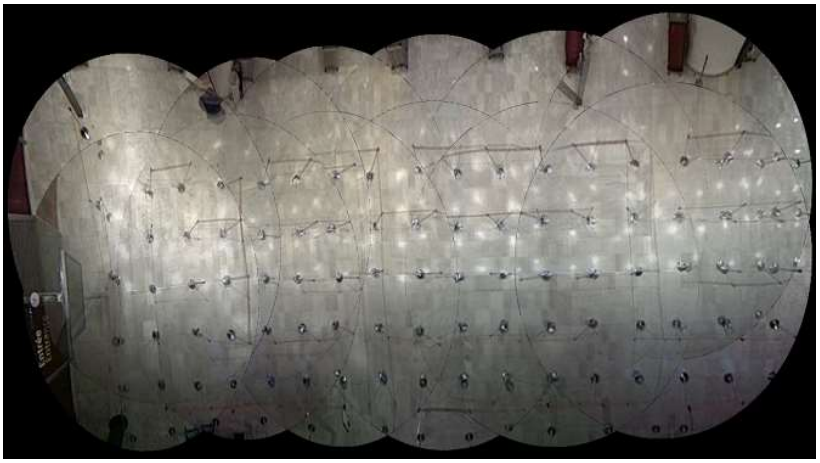


Figure 8 : exemple de mosaïque d'image réalisée à partir de la projection de 17 caméras sur un même plan de référence

Une fois que la position de chaque personne est déterminée à chaque image, nous effectuons la détection automatique des files d'attente et pour cela nous cherchons à détecter des tendances de mouvement dans la scène. Plusieurs approches sont possibles pour répondre à cette problématique dont l'approche classique basée sur le suivi d'objets. Cette approche étant difficile à mettre en œuvre ici, nous avons choisi une piste sans géographique différente, utilisée dans le domaine des télécoms (sur des données d'alarmes), fondée sur la détection d'épisodes fréquents dans une séquence d'évènements typés. Appliqués au contexte de la détection de trajectoires, de tels algorithmes permettent de trouver les motifs fréquents de zones de l'image

faisant l'objet d'activité (de mouvement). De manière abstraite, une trajectoire physique peut être perçue comme une succession d'évènements décrivant le changement d'état d'une zone de l'image (un ensemble de pixels) par rapport à un état statique. Cette approche permet, ainsi, de s'affranchir de l'information géographique et de ne voir l'image que comme un ensemble de « types de zone » susceptible de changer d'état et donc de créer une alarme, un événement. L'analyse de cette succession d'événements, qui correspondent donc à l'activité présente dans la vidéo, permet de trouver des séquences correspondant à des trajectoires.

3.3.2 Protection des hommes en jaune (2015-2018) :

Le projet YELLOW avait pour objectif d'améliorer la sécurité des agents d'exploitation et de gestion des autoroutes (appelés « hommes en jaune ») et des usagers de la route. Les données d'accidents de l'année 2018[1] montrent que les « hommes en jaune » travaillant pour les sociétés d'autoroutes françaises et intervenant au cœur des flux de circulation sont une population vulnérable. Les accidents matériels et humains occasionnés par l'intrusion d'un usager de la route dans la zone de chantier sont particulièrement fréquents et significatifs pour les exploitants (et les usagers). Face à ce bilan, les sociétés d'autoroute qui visent le « zéro accident » ont pour objectif de réduire l'exposition aux risques du personnel, notamment à travers le développement de nouveaux matériels. Le projet YELLOW portait sur la conception de systèmes embarqués de détection et d'alerte d'une intrusion d'un usager de la route dans le périmètre d'un chantier et du risque de collision avec les opérateurs et les équipements de chantier.

Dans le cadre de ce projet, nous nous sommes intéressés particulièrement à la détection des véhicules entrant dans la zone de chantier ou d'intervention ayant un fort risque de collision avec des engins, des équipements ou des hommes en utilisant des techniques de segmentation et de suivi. L'extraction des objets de premier plan reste un problème ouvert que nous avions suffisamment étudié pour permettre une segmentation fine. Dans le même ordre d'idée, le suivi des objets avait également bien progressé, notamment avec les travaux de thèse de Matthieu Roger [10] que j'ai encadrée.

Une contrainte particulière était liée à son usage. Pour un fonctionnement optimal, l'analyse nécessitait souvent d'être paramétrée soigneusement en fonction du contexte. C'est une étape délicate dont dépend le résultat. Comme il est difficile d'avoir sur chaque chantier un expert en analyse vidéo dédié au paramétrage, il était nécessaire que la solution développée puisse s'auto-configurer et ne nécessiter aucune intervention manuelle sur site. L'idéal étant que les opérateurs n'aient absolument rien à paramétrer lorsqu'ils arrivent sur un chantier. Cela impliquait que le système puisse lui-même analyser la scène par un apprentissage non supervisé de façon à rapidement déterminer les zones de passage des véhicules et soit capable de distinguer les trajectoires « normales » des trajectoires « anormales ». Afin de pouvoir répondre

[1] https://www.autoroutes.fr

à cette problématique nous avons proposé un modèle paramétrique de chaussée et un algorithme de détection automatique de ces paramètres.

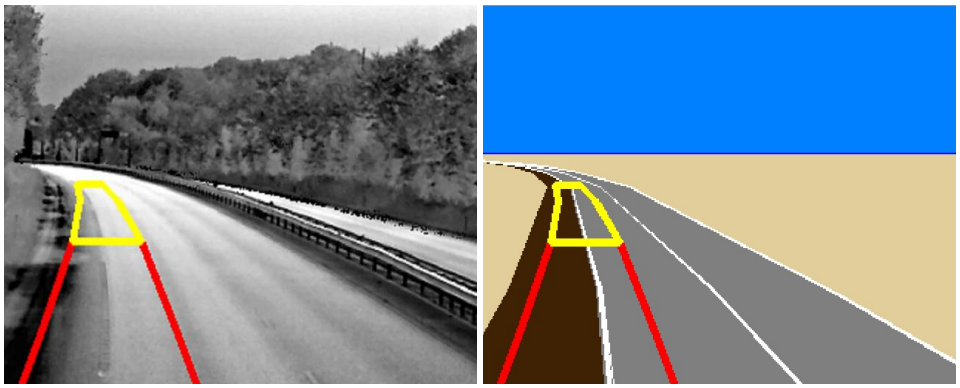


Figure 9 : exemple de scène captée par la caméra (à gauche) et représentation synthétique du modèle de chaussée après détection automatique. Le polygone vert représente la zone de pré-alarme et les deux lignes rouges délimitent la zone d'alarme.

3.4 Encadrements de thèses

Depuis sa création, la société Foxstream a toujours eu la volonté d'investir dans la recherche. A ce jour, cinq thèses ont été soutenues en lien avec le laboratoire LIRIS et une 6ième thèse est en cours. Nous pouvons ajouter une 7ième qui n'a pas été menée jusqu'à son terme mais qui a donné lieu à plusieurs publications. Je présente dans cette section les travaux de thèse que j'ai officiellement co-encadrés.

2009 - 2011 - Co-encadrement de la thèse de Corentin Lallier avec la Pr. Laure Tougne. Détection d'intrusions.

Les travaux de recherche se sont organisés autour de deux axes. Le premier axe était celui de l'extraction d'objets dans la vidéo, avec une première phase de bibliographie, notamment sur les algorithmes de soustraction de fond. Puis, dans une seconde phase, l'implémentation d'un algorithme choisi (le Vumètre) avec évaluation des forces et des faiblesses de cet algorithme par rapport aux différentes conditions et proposition de diverses améliorations. Dans une dernière phase, il a été proposé un environnement de comparaison de cet algorithme avec ceux déjà existants au sein de la société. Cet environnement était composé d'une interface de tests, d'une mesure de comparaison d'images binaires et de différentes bases de vidéos étiquetées. Cet environnement a donné lieu à un article [11]. Le second axe était en rapport avec la classification *one-class* (ou mono-classe) dans le cadre de la détection d'intrusions. La classification était effectuée en fonction des différentes caractéristiques (descripteurs de formes, position, *etc.*) calculées à partir des objets extraits de la vidéo. La classification one-class repose sur le principe de la détection de nouveauté. Le classifieur doit apprendre un modèle représentatif des objets fréquents, typiquement les bruits, afin de catégoriser tout nouvel objet comme intrusif. Le but est de pouvoir classer un objet comme intrusion dans le cas où il est très difficile (pour des raisons technique, financière ou autre) voire impossible de définir une intrusion de manière précise. Le doctorant a malheureusement interrompu sa thèse pour rejoindre son épouse à Bordeaux.

2011 - 2015 - Co-encadrement de la thèse de Matthieu Rogez avec la Pr. Laure Tougne. Utilisation du contexte pour la détection et le suivi d'objets en vidéosurveillance. Soutenue le 9 juin 2015.

Dans cette thèse [10], nous nous sommes concentrés essentiellement sur les tâches de détection et de suivi des objets mobiles à partir d'une caméra fixe. Contrairement aux méthodes basées uniquement sur les images acquises par les caméras, notre approche a consisté à intégrer un certain nombre d'informations contextuelles à l'observation afin de pouvoir mieux interpréter ces images. Ainsi, nous avons proposé de construire un modèle géométrique et géolocalisé de la scène et de la caméra. Ce modèle est construit directement à partir des études de pré-déploiement des caméras et peut notamment utiliser les données OpenStreetMap afin d'établir les modèles 3D des bâtiments proches de la caméra. Nous avons complété ce modèle en intégrant la possibilité de prédire la position du soleil tout au long de la journée et ainsi pouvoir calculer les ombres projetées des objets de la scène. Cette prédiction des ombres a été mise à profit afin d'améliorer la segmentation des piétons par modèle de fond en supprimant les ombres du masque de mouvement [12].

Une autre contribution majeure de ce travail concerne le suivi des objets mobiles, où nous avons utilisé le formalisme des automates finis afin de modéliser efficacement les états et évolutions possibles d'un objet. Ceci nous a permis d'adapter le traitement de chaque objet selon son état. Nous avons géré les occultations inter-objets à l'aide d'un mécanisme de suivi collectif (suivi en groupe) des objets le temps de l'occultation et de ré-identification de ceux-ci à la fin de l'occultation. Notre algorithme s'adapte à n'importe quel type d'objet se déplaçant au sol (piétons, véhicules, etc.) et s'intègre naturellement au modèle de scène développé [13].

Nous avons également développé un ensemble de "rétroactions" tirant parti de la connaissance des objets suivis afin d'améliorer les détections obtenues à partir d'un modèle de fond. En particulier, nous avons abordé le cas des objets stationnaires, souvent intégrés à tort dans le fond, et avons revisité la méthode de suppression des ombres du masque de mouvement en tirant parti de la connaissance des objets suivis. L'ensemble des solutions proposées a été implémenté dans le logiciel de l'entreprise Foxstream et est compatible avec la contrainte d'exécution en temps réel nécessaire en vidéosurveillance.

2016 - 2020 - Co-encadrement de la thèse d'Amine El-Ouassouli avec Mcf. Marian Scuturici. Détection de relations spatio-temporelles dans un système de capteurs hétérogènes. Soutenue le 24 juillet 2020

Ce travail sort du cadre de nos problématiques de recherches habituels est a été motivé par, d'une part, la multiplication des données que nous pouvions extraire d'un ou de plusieurs flux vidéo et d'autre part, par une quantité sans cesse croissante d'informations pouvant être mise à notre disposition. Qu'il s'agisse d'informations ponctuelles simples comme un contact de porte, des informations plus détaillées comme le ticket d'une caisse enregistreuse, ou encore des horaires de train ou des données météo. En effet, les avancées significatives qu'ont connu les technologies de capteurs, leur utilisation croissante ainsi que leur intégration dans les systèmes d'information permettent d'obtenir des descriptions temporelles riches

d'environnements réels. L'information générée par de telles sources de données peut être qualifiée d'hétérogène sur plusieurs plans : types de mesures physiques, domaines et primitives temporelles, modèles de données, *etc.* Dans ce contexte, l'application de méthodes de fouille de motifs constitue une opportunité pour la découverte de relations temporelles non-triviales, directement utilisables et facilement interprétables décrivant des phénomènes complexes.

Nous avons proposé d'utiliser un ensemble d'abstraction temporelles pour construire une représentation unifiée, sous forme des flux d'intervalles (ou états), de l'information générée par un système hétérogène. Cette approche a permis d'obtenir une description temporelle de l'environnement étudié à travers des attributs (ou états), dits de haut niveau, pouvant être utilisés dans la construction des motifs temporelles. A partir de cette représentation, nous nous sommes intéressés à la découverte de dépendances temporelles quantitatives (avec information de délais) entre plusieurs flux d'intervalles. Nous avons introduit un modèle de dépendances « Complex Temporal Dependency » (CTD) défini de manière similaire à une forme normale conjonctive. Ce modèle a permis d'exprimer un ensemble riche de relations temporelles complexes. Pour ce modèle de dépendances nous avons proposé des algorithmes efficaces de découverte : CTD-Miner [9] et ITLD - Interval Time Lag Discovery [14] .

2019-(2022 ?) – Co-encadrement de la thèse de Devashish Lohani avec Pr. Laure Tougne, MCF. Carlos Crispim Junior avec l'assistance de Sarah Bertrand et Quentin Barthélémy. Apprentissage profond non supervisé de représentations spatio-temporelles pour la vidéo. Soutenance prévue en septembre 2022.

L'objectif de cette thèse est de capturer, de façon non supervisé ou faiblement supervisé, la nature spatio-temporelle des vidéos dans un modèle génératif efficace, et non de traiter indépendamment les dimensions spatiales (images) et la dimension temporelle. Au même titre que les premières couches des réseaux convolutionnels 2D encodent des descripteurs locaux spécialisés pour les images, nous souhaitons apprendre des descripteurs spatio-temporels permettant des modéliser les vidéos. Une fois appris, les descripteurs de ce réseau génératif pourront être utilisés comme entrée d'un réseau discriminatif appris pour une tâche supervisée, telle que la reconnaissance d'action.

Pour cette première étape de capture des descripteurs locaux spatio-temporelle plusieurs architectures peuvent être envisagées comme les auto-encodeurs variationnels (VAE) [15] ou les réseaux génératifs adversariaux (GAN) [16]. Nous avons fait le choix d'expérimenter les auto-encodeurs covolutionnels 3D. Les premiers résultats que nous avons obtenus [17] semblent montrer que cette approche modélise assez bien les caractéristiques spatio-temporelles des vidéos.

3.5 Suivi de thèses

Dans cette section, je présente les travaux de thèses que j'ai suivies en tant que responsable recherche et développement de l'entreprise Foxstream.

2004-2007 : Suivi de la thèse de Nicolas Thome. Représentations hiérarchiques et discriminantes pour la reconnaissance des formes, l'identification des personnes et l'analyse des mouvements dans les séquences d'images. Soutenue le 11 juillet 2007

Dans ce travail [18], nous avons abordé le problème de l'analyse des séquences d'images autour de deux thèmes principaux : la lecture de plaques minéralogiques et l'analyse du mouvement humain, en nous focalisant sur l'interprétation de vidéos monoculaires et dans un contexte algorithmique proche du temps réel.

Au début des années 2000, la lecture de plaques minéralogiques était une problématique qui suscitait l'intérêt de la communauté scientifique et du monde industriel depuis de nombreuses années, à la fois pour les thématiques de recherches soulevées par les questions de reconnaissance de formes que pour les applications pratiques qui en découlaient. A cette période déjà une quantité importante de solutions logicielles étaient commercialisées sur le marché, à des prix abordables et présentant des taux de reconnaissance corrects bien que très en deçà de ce qu'il est possible de faire aujourd'hui avec les techniques d'apprentissage profond. Les performances observées dans les situations réelles s'avéraient nettement dégradées en pratique par un ensemble de conditions d'observation difficiles : éclairage variable, plaques sales, distance et orientations relatives de la caméra par rapport au véhicule, *etc*. La méthode que nous avons proposée s'est décomposée autour des étapes suivantes : localisation de la plaque, binarisation, extraction des caractères, reconnaissance et moyennage temporel [19]. Le cœur de la technologie, constitué par la reconnaissance optique de caractères, a été prise en charge par un réseau de neurones hiérarchique qui analyse la forme des descripteurs de Fourier des contours extraits sur chaque image binaire. Le découplage des deux niveaux logiques du classifieur permet un apprentissage fiable qui limite au maximum les phénomènes de sur apprentissage, et rend possible la reconnaissance de caractères altérés : contours internes bouchés, topologie inexacte, *etc*. D'autre part, nous avons mis en place à plusieurs points critiques de l'algorithme une coopération verticale entre niveaux logiques. Celle-ci permet à une couche donnée d'analyser la pertinence de l'interprétation effectuée aux niveaux inférieurs, offrant la possibilité de suivre plusieurs hypothèses en cas d'ambiguïtés. Malgré la simplicité apparente de la problématique et la nature relativement standard des caractères à identifier, l'identification des véhicules par l'interprétation du contenu de leur plaque minéralogique nécessite la mise en place de stratégies évoluées.

A l'opposé, l'analyse du mouvement humain apparaît d'emblée comme un problème difficile. La nature complexe et articulée du corps humain, les auto-occultations entre membres, la forte variabilité de l'apparence ou la faible observabilité des degrés de liberté en profondeur sont autant d'éléments qui rendent la détection, le suivi et l'interprétation du comportement des personnes dans les séquences d'images difficiles. Pour aborder ce problème complexe, nous avons proposé, dans un premier temps, une approche de détection et de suivi assez basique, reposant sur une extraction des régions en mouvement qui sont ensuite dynamiquement mises en correspondance au cours du temps. Cette première partie de l'algorithme, qui ne tient pas compte de l'aspect déformable de la projection du corps humain dans l'image, permet cependant un traitement rapide et rend possible la détection de situations de fusion ou de séparation de régions. Ce premier niveau d'analyse est générique et est applicable à un type

d'objet quelconque. Dans les cas où le suivi peut être assuré sans ambiguïté, nous passons à une échelle plus fine, en proposant une méthode pour détecter et étiqueter les différentes parties visibles du corps de chaque personne à partir de la silhouette extraite en mouvement. Cette étape est menée à bien par une technique de mise en correspondance entre un graphe calculé à partir du squelette 2D image et un modèle 3D du squelette humain. L'ensemble des candidats pour les membres est identifié en introduisant une information *a priori* qui fixe des contraintes d'assemblement, utilise uniquement l'information morphologique et topologique de la silhouette, et est donc applicable dans des conditions générales, indépendamment du point de vue, de la pose de la personne, de la géométrie ou de l'apparence des membres. Elle permet à chaque instant de capturer un ensemble de caractéristiques de forme, de couleur et de texture sur chaque membre, conduisant à la mise à jour dynamique d'un modèle d'apparence articulé pour chaque personne suivie. Ce dernier est ensuite utilisé dans des situations difficiles pour effectuer une identification des personnes et assurer une poursuite robuste du suivi. L'approche propose donc de mettre à jour une caractéristique dont le contenu informatif est ensuite rétro propagé au niveau de la détection région pour contraindre le suivi. L'approche dans son ensemble permet un fonctionnement temps réel, et utilise un niveau de détail adaptatif en fonction des difficultés rencontrées lors des différentes étapes. Nous avons appliqué cet algorithme dans le cadre du maintien à domicile des personnes âgées et plus spécifiquement pour permettre la détection des chutes.

2007-2010 Suivi de la thèse de Ionel Pop. Détection des événements rares dans des vidéos. Soutenue le 23 septembre 2010

Dans ce travail de thèse [20], qui se place dans le contexte de l'analyse automatique de vidéos, nous avons proposé plusieurs algorithmes permettant d'identifier des événements inhabituels, en faisant l'hypothèse que ces événements ont une faible probabilité. Nous avons abordé plusieurs types d'événements, de l'analyse des zones en mouvement à l'analyse des trajectoires des objets suivis.

Nous nous sommes focalisés essentiellement sur les objets suivis et nous avons proposé plusieurs mesures de similarité entre des trajectoires. Ces mesures, basées sur DTW (Dynamic Time Warping), estiment la similarité des trajectoires prenant en compte différents aspects spatiaux, mais aussi temporels, pour être en mesure de faire la différence entre des trajectoires qui ne sont pas parcourues de la même façon (en termes de vitesse de déplacement). Nous avons construit des modèles de trajectoires, permettant de représenter les comportements habituels des objets afin de pouvoir ensuite détecter ceux qui s'éloignent de la normale. Pour pallier les défauts de suivi qui apparaissent dans la pratique, nous avons analysé les vecteurs de flot optique et nous avons construit une carte de mouvement. Cette carte modélise sous la forme d'un codebook les directions privilégiées qui apparaissent pour chaque pixel, permettant ainsi d'identifier tout déplacement anormal, sans avoir pour autant la notion d'objet suivi. En utilisant la cohérence temporelle, nous avons apporté une amélioration significative du taux de détection, affecté par les erreurs d'estimation de flot optique.

3.6 Encadrement d'étudiants en Master Recherche

2006 : Automatisation du test d'enfouissement défensif à l'aide de l'analyse vidéo, Mathieu Lesourd, 4 mois, M2 Sciences Cognitives et Applications, Université Nancy 2

L'objet du stage a été de récupérer l'ensemble des données fournies par l'analyse vidéo, d'en extraire des informations pertinentes pour le chercheur et de rendre ces informations facilement exploitables grâce à une interface intuitive (Figure 10).

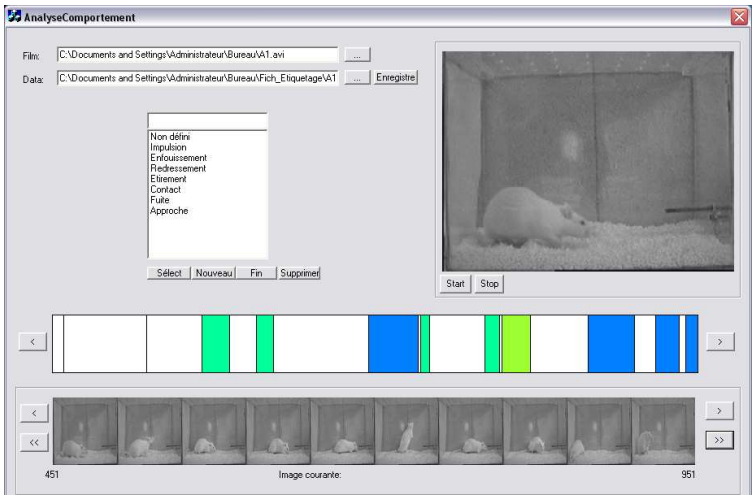


Figure 10 : capture d'écran du logiciel « AnalyseComportement »

2006 : Suivi d'objet en mouvement à l'aide d'une caméra dôme, Nadir Benmounah, 6 mois, M2 Informatique Graphique et image, Lyon 2

Ce stage, très en lien avec mon début de thèse, a permis de mettre en lumière certaines carences de la caméra. Outre les défauts liés à l'objectif, le principal défaut mis en évidence a été la faible précision des informations de position donnée par la caméra. Une contribution originale de ce stage a été la proposition d'une correction des changements de luminosité basée sur le modèle colorimétrique HSV.

2008 : Développement d'un classifieur pour application de vidéo surveillance temps réel, Clément Metge, 6 mois, M2 sciences cognitives, Lyon 2

Il s'agissait d'étudier plusieurs classifieurs et systèmes d'apprentissages. Une approche descriptive des vecteurs de signature des formes a été utilisée afin de connaitre leur structure et leur composition et orienter nos recherches vers le classifieur le plus approprié (Figure 11). Cette étude nous a rapproché des cartes auto-organisatrices de Kohonen qui ont été adaptées à notre problématique.

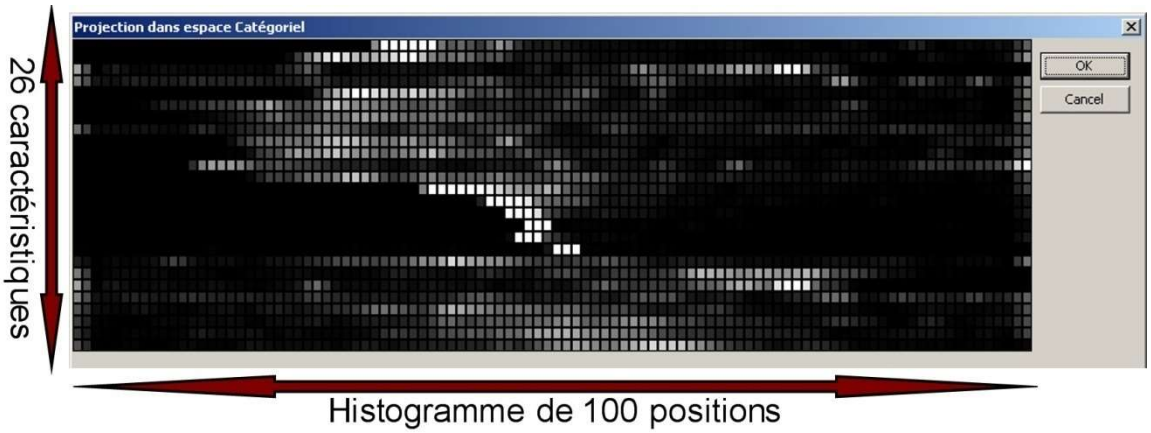


Figure 11 : projection des histogrammes des 26 composantes du vecteur de forme

2009 : Architecture réseau dans un système de Vidéo-Surveillance, Simon Meyffret, 4 + 2 mois, M2 Systèmes informatiques et réseaux + PFE Insa Lyon

Le stage s'est déroulé en deux phases afin de permettre à l'étudiant de valider son double parcours. Une première phase de recherche d'une durée de 4 mois et une seconde phase d'industrialisation. Lors de la première partie, une analyse des besoins en sécurité dans un système de vidéo-surveillance a été réalisée. Ceci a permis de mettre en relief des contraintes fortes concernant l'intégrité et la confidentialité des données, mais également concernant la difficulté liée au chiffrement des données multimédia (données volumineuses, fréquentes, *etc*.). Cette étude a été l'occasion d'évaluer les risques pour le système et d'optimiser les mesures de sécurité à prendre pour diminuer les attaques.

2010 : Analyse et extraction de caractéristiques à partir de flux vidéo H264, Laurent Charpentier, 4 + 2 mois, M2 Systèmes informatiques et réseaux + PFE Insa Lyon

L'objectif de ce stage a été d'étendre les notions de détection, de localisation et d'extraction des caractéristiques du mouvement dans un flux vidéo H.264, à partir du disque ou en temps réel, sans décompresser le flux. Le principe de détection réside en l'étude de la taille des macro-blocks (Figure 12) et de leur évolution au cours du temps (Figure 13). Ce travail a été valorisé par une publication [21].

Figure 12 : localisation de deux macro-blocs MB1 et MB2 dans l'image.

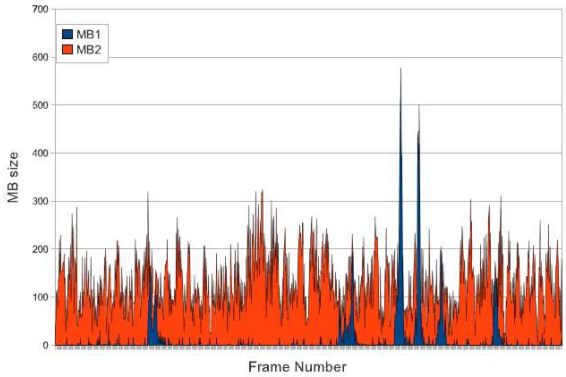


Figure 13 : évolution au cours du temps de la taille des deux macro-blocs sélectionnés dans l'image. Les deux pics supérieurs à 400 pour MB1 correspondent au passage d'un véhicule.

2017 : Comptage de foule basé sur la régression par processus gaussien, Nina Ghigo, PFE INSA parcours recherche

Malgré toutes les avancées qui ont été faites dans le domaine de la détection, isoler des individus dans une foule dense reste une opération très complexe et relativement peu fiable. Le comptage par régression ne se base ni sur la détection des individus ni sur le suivi de ceux-ci, et est donc une approche qui peut être employée dans des foules plus denses. Le comptage par régression consiste à estimer la foule en se basant sur des propriétés holistiques de l'image. Nous avons donc effectué une association entre ces caractéristiques et le nombre de personnes dans la foule, *i.e.* nous avons cherché à estimer la fonction qui lie les entrées (descripteurs) aux sorties (nombre de personne) tout en minimisant l'erreur.

2018 : Indexation vidéo par apprentissage profond, Tom Duran, PFE INSA parcours recherche

Dans ce travail, nous avons proposé un système unifié d'indexation et de recherche d'événements dans une vidéo basée sur les découvertes récentes en apprentissage profond. Nous avons montré qu'un détecteur d'objets tel que YOLOv2 peut être utilisé comme un outil très efficace pour la détection d'événements, la sélection d'images clés et la reconnaissance de scènes. Notre approche a été motivée par le fait que les cartes de caractéristiques calculées par le détecteur profond codent non seulement la catégorie des objets présents dans l'image, mais également leur emplacement, éliminant ainsi automatiquement les informations d'arrière-plan. Les résultats expérimentaux sur différents ensembles de données de vidéo surveillance ont démontré l'efficacité du système proposé. Ce travail a été valorisé par une publication [4].

3.7 Encadrements de projets de fin d'étude

2002 : Profilomètre laser, Stéphane Robert, 6 mois, PFE 5^{ième} année ingénieur.

L'objectif de ce stage a été d'étudier la faisabilité d'un profilomètre laser portable permettant de mesurer avec une précision de 50µm l'usure des rails de tramway. Le relevé du profil était effectué par l'acquisition, à l'aide d'une caméra, d'une trace laser projetée sur le rail (Figure 14).

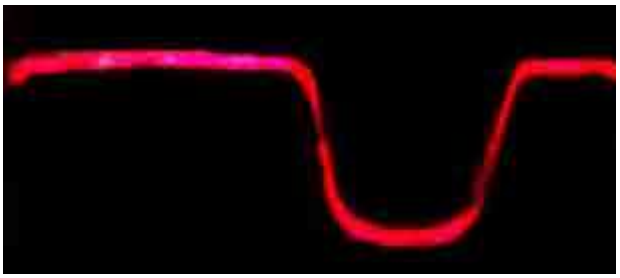


Figure 14 : image de la trace laissée par la projection d'un plan laser sur un rail de tramway

2003 : Traitement du flux vidéo, Abdelkader Bennat, 6 mois, PFE 5ième année ingénieur, Istase Saint-Etienne

Dans le cadre de ce stage, nous avons réalisé un état de l'art et une évaluation de différentes approches de segmentation de l'avant plan et étudié de différents espaces colorimétriques.

2004 : Segmentation de 2 objets en mouvement dans des applications de vidéosurveillance, Cyril Chambeyron, 5 mois, PFE 5ième année ingénieur, INPG Grenoble

Ce travail portait sur de la problématique liée au croisement des objets en mouvement dans une vidéo. Les modèles classiques de segmentation de l'avant plan ne permettant pas de séparer les objets en mouvement qui se croisent. Nous avons mis en place une segmentation basée sur l'algorithme MVFAST avec une recherche optimisée en diamant.

2005 : Détection de personnes, Vincent Prélot, 5 mois, PFE 5ième année ingénieur, INSA Lyon

Nous avons étudié et implémenté l'algorithme de Viola et Jones [22], basé sur l'utilisation des ondelettes de Haar et d'un classifieur AdaBoost. Au cours de ce stage, nous avons utilisé 3 classifieurs en cascades et permis d'améliorer significativement la détection des personnes par rapport à la solution originale proposée par les auteurs.

2006 : FoxTool, outil d'aide au placement des caméras, Kevin Frugier, 5 mois, PFE 5ième année ingénieur, INSA Lyon

Les algorithmes de vidéo protection nécessitent des prérequis indispensables qui doivent absolument être respectés. Afin d'assurer une couverture optimale des sites à sécuriser, l'objet du stage a été de réaliser un outil (Figure 15) d'aide au placement des caméras sur un plan 2D intégrant les différentes contraintes intrinsèques et extrinsèques des caméras : focale, taille du capteur, position, orientation, *etc.,* ainsi que les contraintes liées aux types d'algorithmes utilisés.

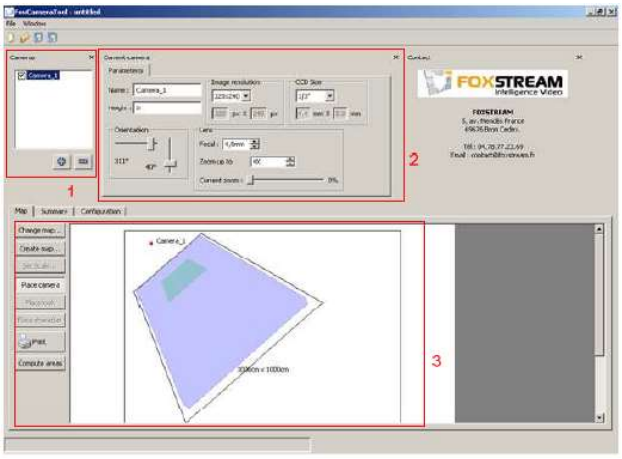


Figure 15 : interface de l'outil FoxTool d'aide au placement des caméras

2006 : Lecture de plaque minéralogique, Thibaut Abgrall, 6 mois, PFE 5ième année ingénieur, ENSERG Grenoble

L'objectif du stage a été d'améliorer et de poursuivre les développements du module de détection et de reconnaissance des plaques d'immatriculation (Figure 16). Un travail conséquent a été réalisé afin d'améliorer les performances de l'OCR basé sur un réseau de neurones. Une contribution originale a été apportée pour calculer une chaîne de caractères moyenne à partir de plusieurs résultats successifs du réseau de neurones.

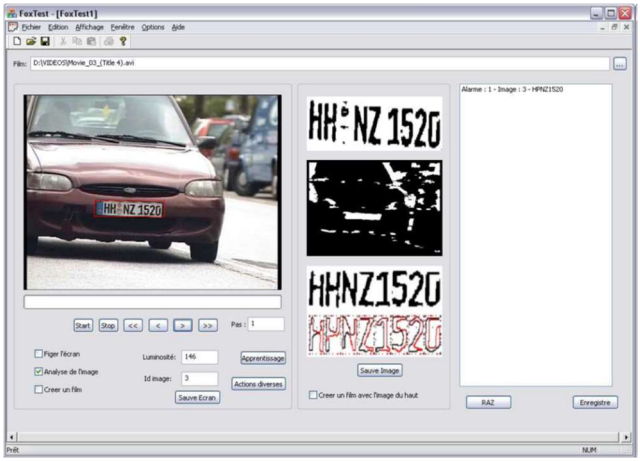


Figure 16 : interface de test du module de détection et de reconnaissance des plaques d'immatriculation

2009 : FoxTool, outil d'aide au placement des caméras, David Clapot, 3 mois, 4ième année ingénieur, INSA Lyon

L'objectif de ce stage a été d'ajouter de nouvelles fonctionnalités au logiciel existant et d'optimiser le calcul des projections.

2010 : Conception d'application de traitement vidéo : acquisition, Jennifer Renoux, PFE 5ième année ingénieur, INSA Lyon

Il s'agissait essentiellement d'un travail d'intégration de plusieurs cartes d'acquisitions et d'optimisation des transferts de flux dans le logiciel FoxVigi.

2010 : Monitoring d'une application d'analyse vidéo, Matthieu LANGLOIS, PFE 5ième année ingénieur, INSA Lyon

Dans le cadre de ce stage, nous avons développé plusieurs outils permettant d'évaluer la qualité du fonctionnement du logiciel sur site et de visualiser les éventuelles défaillances afin de réagir au plus vite. Ceci afin de permettre aux superviseurs (utilisateurs du logiciel) de surveiller l'état du logiciel à distance et de générer des rapports d'activité du logiciel exportables dans un format standard et facilement interprétable.

2012 : Amélioration de la détection et de la reconnaissance de plaque d'immatriculation, Sarha Driai, 4ieme année INSA

Le but de ce stage a été de reprendre les différentes briques qui composent un système de lecture de plaques minéralogiques et d'en améliorer les performances. Nous avons ainsi travaillé sur la détection de la plaque dans l'image en étudiant différents paramètres de texture d'Haralik. Nous avons proposé plusieurs améliorations dans la phase de binarisation et sur l'extraction des caractères (Figure 17). Enfin, nous avons travaillé sur l'amélioration des performances du réseau de neurones essentiellement par l'enrichissement de la base d'apprentissage.

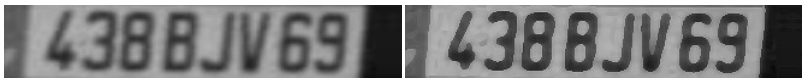


Figure 17 : exemple de plaque avant et après amélioration

2013 : Détection de visages et classification hommes-femmes sur des images et des vidéos, Emeline Amouroux, PFE 5ieme année, ISIMA.

La classification du genre sur des images de visages est un sujet difficile qui a été largement étudié ces dernières années et dont les applications sont nombreuses : indexation de vidéos, recherche d'information, applications commerciales. Cette reconnaissance du genre se divise en plusieurs étapes : détection du visage, pré-traitement, classification (extraction de caractéristiques puis classification de ces caractéristiques). Nous avons travaillé sur l'ensemble de ces étapes lors de ce projet. Pour la détection, nous avons utilisé les algorithmes de Viola et Jones [23] , et de Christophe Garcia [24], que nous avons comparés et testés sur des vidéos. Une fois le visage détecté, nous lui avons appliqué un pré-traitement simple avec des méthodes de base : conversion en échelle de gris et redimensionnement. Un pré-traitement plus évolué sera étudié par la suite pour améliorer les résultats. Ensuite, pour la première étape de la classification, nous avons choisi le descripteur de Weber spatial [25], puis nous avons testé deux classifieurs : la distance minimale avec la mesure Chi-2 et les SVMs. Pour valider notre descripteur, nous avons utilisé, dans un premier temps, des images de texture de la base Brodatz [26]. Puis nous avons appliqué et testé notre méthode complète sur des images de la base FERET [27] et sur une nouvelle base d'images de visages issus de plusieurs bases et d'Internet. Cette dernière nous a permis de nous rendre compte des limites d'un tel sujet : luminosité des images, prise de vue de la caméra, orientation des visages.

II - La détection périmétrique

1 Détection d'intrusions

1.1 Introduction et contexte

Avant d'aborder les problématiques de recherche, nous présentons rapidement dans ce premier chapitre le concept de détection d'intrusion tel qu'il est défini dans le monde industriel, ses contraintes ainsi que les solutions qui peuvent être proposées.

Le travail de recherche que nous avons mené depuis 2003 a été réalisé au sein de la société Foxstream et du laboratoire LIRIS. Le groupe Foxstream est composé des sociétés Foxstream, Foxstream Inc. et Cossilys21.

Foxstream est une société d'édition logicielle, fondée en 2004, spécialisée dans l'analyse et le traitement automatique en temps-réel du contenu d'images vidéo. Foxstream offre des solutions capables d'extraire et de transmettre une information pertinente à partir d'un flux vidéo. Foxstream est présent essentiellement sur le marché de la sécurité (vidéosurveillance), et sur le marché de la gestion de flux (comptage, files d'attente, etc.) pour des aéroports, commerces, etc. Sa filiale Foxstream Inc. est basée à Miami, USA.

Cossilys21 est une société de haute technologie ayant pour vocation l'innovation et la production de systèmes intelligents de vidéo protection. Depuis plus de 20 ans, Cossilys21 s'impose comme référence sur le marché de la vidéo-protection notamment dans le secteur bancaire pour lequel Cossilys21 équipe de grandes banques nationales et régionales. Cossilys21 intervient également sur de nombreux secteurs d'activité comme le retail ou encore l'industrie.

Le LIRIS est une unité mixte de recherche (UMR 5205), regroupant plus de 300 membres en 2019 et dont les tutelles sont le CNRS, l'INSA de Lyon, l'Université Claude Bernard Lyon 1, l'Université Lumière Lyon 2 et l'Ecole Centrale de Lyon. Le champ scientifique de l'unité est l'Informatique et plus généralement les Sciences et Technologies de l'Information.

Les activités scientifiques de ses 14 équipes de recherche sont structurées en 6 pôles de compétences, de 15 à 30 permanents, reconnues au niveau international :

- Vision intelligente et reconnaissance visuelle
- Géométrie et modélisation
- Data Science (Science des données)
- Services, Systèmes distribués et Sécurité

- Simulation, virtualité et sciences computationnelles
- Interactions et cognition

Le LIRIS réalise une activité de recherche de fond sur ces 6 pôles de recherche, tout en développant un savoir-faire au service de la société en liaison étroite avec les disciplines Ingénierie, Sciences Humaines et Sociales, Sciences de l'environnement et Sciences de la Vie.

1.2 Description d'un module de détection d'intrusion

Le terme « vidéo-surveillance » ou sa version plus politiquement correcte « vidéo-protection » sont des termes génériques qui regroupent l'ensemble de la chaine de traitement de la vidéo dans le cadre de la surveillance des biens et des personnes. Parmi l'ensemble des éléments de cette chaîne de traitement nous pouvons trouver l'acquisition, la diffusion des flux vidéo, la visualisation et l'enregistrement. La très grande majorité des installations de vidéo-protection se limite à ces étapes. Dans ce type d'installation, la vidéo peut être affichée en direct sur un ou plusieurs moniteurs afin qu'un opérateur visualise la scène et puisse agir en direct. Plus généralement la vidéo est simplement enregistrée en continu pour être consultée plus tard afin de vérifier un événement rapporté par ailleurs. L'analyse en temps réel des flux vidéo, que l'on nomme « Vidéo Surveillance Intelligente » (VSI) pour bien se démarquer des simples enregistreurs, n'est finalement que très peu utilisée. Cette analyse peut être utilisée dans différents contextes comme par exemple la voie publique, un magasin, un aéroport ou autour d'un bâtiment.

Dans le cadre de notre travail, nous nous intéressons essentiellement aux algorithmes de détection d'intrusion et plus spécifiquement à la détection périmétrique. Les algorithmes que nous allons présenter ne sont pas réservés exclusivement à ce type de détection. La modélisation de fond, qui est l'une des étapes importantes, est utilisée dans plusieurs autres domaines. Mais dans la plupart des cas, les algorithmes devront être optimisés pour cette tâche et utiliser des constantes de temps en adéquation avec le délai de réaction nécessaire à une bonne prise en charge de la détection.

De façon assez générale, la détection d'intrusion consiste à analyser l'activité normale et à repérer les activités anormales ou suspectes. Ce cadre s'applique aussi bien à la vidéo surveillance qu'à la surveillance d'un système d'information. Dans le cas de la vidéo surveillance, la finalité est de détecter et de signaler la présence ou la pénétration non autorisée de certaines classes d'objets dans une zone et dans une période de temps définies par l'utilisateur. Elle peut avoir pour fonction de déceler plusieurs types d'intrusion comme le franchissement, l'approche ou même le maraudage si l'on considère la dimension temporelle. La notion de classe non autorisée peut être assez difficile à définir dans la mesure où l'on ne peut pas forcément prédire de façon claire la façon dont l'intrusion va se produire : à pied, à vélo, en voiture ou autre véhicule pour ne parler que de la détection d'intrusion « classique » en vidéo protection. Et même en ce qui ne concerne que les personnes, la station debout n'est pas forcément le mode de déplacement adopté pendant l'intrusion. Si bien qu'un détecteur de piétons faisant l'hypothèse d'une personne debout ne sera peut-être pas capable de détecter correctement une personne qui rampe ou qui se déplace à « quatre pattes ». Il est alors plus efficace de définir

la classe ou la catégorie des objets autorisés comme les oiseaux ou les petits mammifères et de déclencher une alarme sur tous les autres quitte à allonger cette liste en fonction du site à sécuriser.

La détection périmétrique est un cas particulier de la détection d'intrusion puisqu'elle se déploie essentiellement sur la périphérie ou sur le contour d'un bâtiment et vise à détecter une intrusion en amont de la zone à protéger.

Aujourd'hui, dans des conditions générales d'utilisation, les systèmes proposés par la société Foxstream garantissent moins d'une fausse alarme par semaine et par caméra. Ce qui correspond tout de même à moins d'une fausse alarme pour 3 millions d'images analysées (dans le cas de 5 images analysées par seconde). Ce taux est assez remarquable. Cependant pour un petit site équipé de 8 caméras, cela correspond à 1 fausse alarme par jour. La plupart des installations sont gérées à distance par des télésurveilleurs qui assurent la levée de doute 24h sur 24. Les télésurveilleurs gèrent ainsi plusieurs sites. Si un opérateur de télésurveillance assure la sécurité de seulement 200 comptes, il reçoit donc un flux de 200 fausses alarmes par jour. Le traitement moyen d'une fausse alarme étant d'une minute, cela correspond à plus de 3h par jour perdues par le télésurveilleur ! Le traitement des fausses alarmes a donc un coût économique très important qu'il faut bien évidement chercher à limiter.

De façon assez simpliste, un système de VSI réalise les fonctions suivantes :

- Acquisition du flux vidéo
- Extraction du flux vidéo des objets en mouvement dans l'image
- Filtrage et classification (objets réels vs objets hors gabarit et immatériels : ombre au sol, réflexion...)
- Déclenchement d'une alarme si un objet possède les caractéristiques des critères de détection (zone, gabarit, trajectoire…)

Dans le cas de la société Foxstream, les performances d'un module de détection, lorsque les prérequis sont respectés, sont indiquées ci-dessous :

- Délai de détection : en général une seconde, au maximum 2 secondes. Il peut y avoir un délai de latence supplémentaire dû au délai d'acquisition (par exemple pour les acquisitions IP). L'objet à détecter doit avoir une taille suffisante.
- Taux de détection : > 99% hors phénomène météo extrême et contraste suffisant (supérieur à 5, indice colorimétrique sur les 3 canaux)
- Taux de fausses alarmes : une par jour pour les caméras classiques, une par semaine pour les caméras thermiques
- Déplacement minimum de l'objet : afin d'être détecté, l'objet doit se déplacer d'au moins 15 pixels dans l'image analysée.

1.3 Architecture d'un système de vidéo protection

Un module de détection d'intrusion seul n'a pas beaucoup d'intérêt et n'est qu'un élément dans une architecture plus globale qui peut être synthétisé par le schéma suivant :

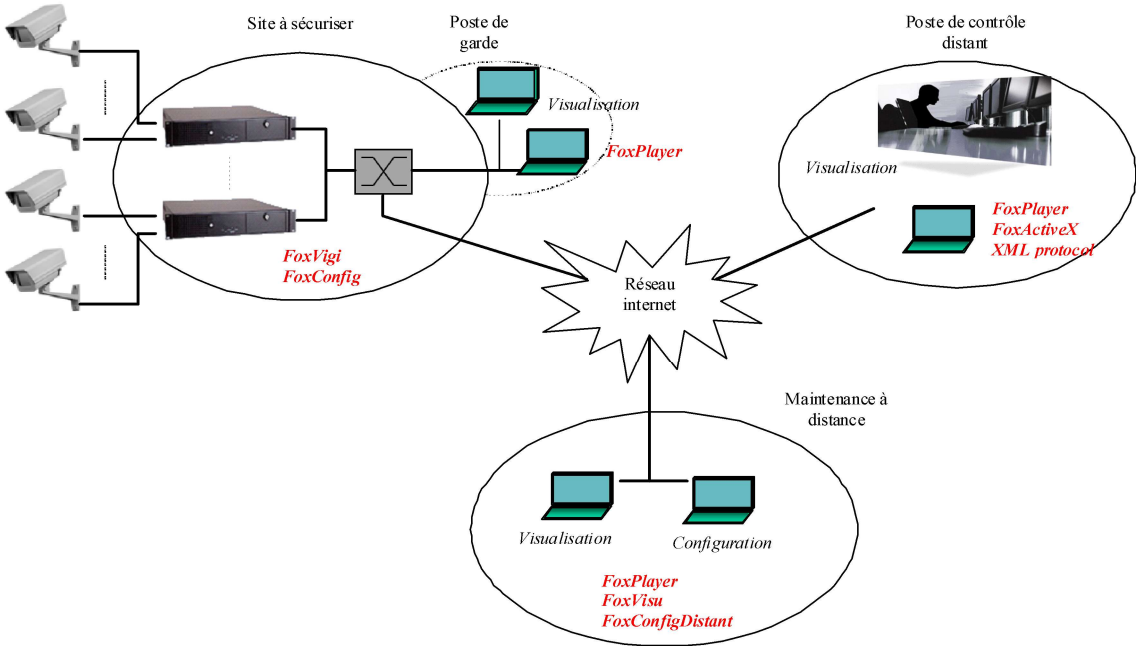


Figure 18 : exemple d'architecture d'un système global de vidéo-protection à partir de produit de la société Foxstream

En fonction de la taille et des possibilités offertes sur le site, certains éléments de ce schéma ne sont pas déployés. Que le site soit important ou relativement modeste, nous pouvons considérer 3 sites géographiques distincts :

- le site à sécuriser avec éventuellement un poste de garde,
- un poste de contrôle distant (généralement un télésurveilleur),
- un site de maintenance.

1.3.1 Site à sécuriser

C'est le site sur lequel sont installés les caméras et les serveurs d'analyses. C'est le site pour lequel une société comme Foxstream apporte une véritable valeur ajoutée. Il existe plusieurs architectures possibles suivant la complexité du site et le nombre de serveur. En principe, un serveur d'analyse a la capacité de traiter jusqu'à 32 flux. Si le nombre de flux est supérieur alors plusieurs serveurs sont utilisés. En règle générale, les serveurs d'analyse sont tous regroupés dans un même local, voir même dans la même baie informatique. Cependant, suivant la nature du site et son importance, les serveurs d'analyses peuvent être disséminés sur l'ensemble du site. Le site à sécuriser peut éventuellement disposer d'un poste de garde avec du personnel sur place habilité à visualiser les flux vidéo et réaliser la levée de doute sur alarme.

1.3.2 Poste de contrôle distant

Le poste de contrôle distant est un poste sur lequel des opérateurs sont habilités à regarder les flux vidéo et lancer des opérations en cas de remonté d'alarme. Il se distingue du poste de garde par le fait que la connexion avec le site à sécuriser est réalisée à travers le réseau internet. Les flux vidéo peuvent être visualisés en temps réel ou en temps différé. Dans certain cas, le flux se limite à une séquence d'image. Cependant, tous les postes de contrôle ne permettent pas la visualisation des flux ni même des images. Le poste de contrôle gère plusieurs sites qui peuvent

être sécurisés avec d'autre type de matériel que la solution Foxstream. Et même si le site à sécuriser est équipé de la solution Foxstream, cette dernière ne gère pas les contrôles d'accès, les alertes incendies ou la température des chambres froides pour n'en citer que quelques-uns. Le poste de contrôle distant est donc équipé d'un logiciel de supervision capable de gérer toutes ces entrées et fournir à l'opérateur toutes les informations et la procédure à suivre en cas d'alerte.

1.3.3 Maintenance à distance

Le poste de maintenance à distance est le poste sur lequel sont envoyés des rapports de fonctionnement de l'application et ou des opérateurs sont habilités à intervenir sur le paramétrage de l'application. Sur ce site, les opérateurs doivent avoir suivi une solide formation pour être capable d'assurer la maintenance et d'ajuster les paramètres si nécessaire. Ce poste doit donc disposer d'outils permettant de gérer les rapports envoyés par les équipements, de détecter automatiquement les disfonctionnements et de prévenir les opérateurs. Ce poste doit également disposer d'outils permettant de gérer un parc machine et de modifier les paramètres.

2 Réglementation

2.1 Voie publique

La législation française encadre les pratiques de télésurveillance, en vertu du respect des libertés et de la vie privée ainsi que du traitement des données. Seules les autorités publiques (mairies, préfectures…) sont ainsi en droit de filmer la voie publique mais elles sont soumises à une autorisation préfectorale.

Les caméras peuvent être installées pour prévenir des atteintes à la sécurité des personnes où pour sécuriser des lieux particulièrement exposés aux vols ou aux agressions. Les dispositifs de vidéo protection sont soumis au code de la sécurité intérieure[2]. Ces caméras peuvent aussi permettre de constater des infractions aux règles de circulation ou de stationnement.

Lorsque la préfecture autorise l'installation de caméras sur la voie publique, en aucun cas celles-ci ne peuvent visualiser l'intérieur des immeubles d'habitation. Dans le cadre privé, des procédés de masquage irréversible de la vidéo doivent être utilisés.

Seule des personnes habilitées par l'autorisation préfectorale peuvent visualiser les images en direct ou les enregistrements. Ces personnes doivent en outre être formées et sensibilisées aux règles et au respect de la vie privé.

[2] Articles L223-1 et suivants (lutte contre le terrorisme), articles L251-1 et suivant

La durée de sauvegarde des enregistrements ne doit pas excéder un mois. En règles générales, quelques jours suffisent amplement pour être informé d'un événement et prendre les dispositions pour visualiser et extraire des séquences. Les séquences ne peuvent être extraites que sur demande dans le cadre d'une procédure de justice et peuvent alors être conservées pendant toute la durée de la procédure.

2.2 Cadre privé

Dans un lieu privé ou dans des locaux professionnels qui n'accueillent pas de public, aucune autorisation préfectorale n'est nécessaire pour l'installation de caméras. Dans ce cadre privé, les dispositifs de vidéo-protection ne sont donc pas soumis aux règles de code de sécurité intérieur.

Toutefois, les entreprises comme les particuliers n'ont pas le droit de filmer la voie publique. Ils ne peuvent filmer que les abords immédiats de leurs bâtiments. Lorsque la voie publique ou des immeubles d'habitation sont visibles sur les images, ils doivent être masqués de façon irréversible à la visualisation et à l'enregistrement.

Le code civil règlemente également les règles à respecter quant à l'enregistrement des vidéos dans le cadre privé[3]. Contrairement aux lieux ouverts au public, la conservation des images n'a pas de limite de durée. Toutefois, il est interdit de filmer ses salariés à leur insu. Il est donc obligatoire de les informer sur l'emplacement des caméras et sur la durée de conservation des images.

2.2.1 Au travail

Au travail, la vidéo protection est particulièrement encadrée. Pour s'en convaincre, il suffit de regarder la liste des articles du code du travail ou du code pénal qui en font référence.

Un employeur peut être amené à installer un dispositif de vidéo protection à l'extérieur de ses locaux mais aussi à l'intérieur. Par contre, il doit obligatoirement consulter les instances représentatives du personnel et en informer ses salariés afin de définir précisément l'objectif de cette installation. Il reste néanmoins un certain nombre de lieux qui ne peuvent en aucun cas être filmés comme les zones de repos, les toilettes ou encore l'accès au locaux syndicaux ou aux représentants du personnels. De même, il est interdit de filmer les salariés sur leur poste de travail sauf dans des cas très précis comme par exemple les employés manipulant de l'argent ou des biens de valeur. Dans le cas d'un caissier, la caméra doit plus spécifiquement visualiser la caisse que le caissier.

[3] Article 9 du code civil (protection de la vie privé), article 226-1 (enregistrement de l'image d'une personne à son insu dans un lieu privé)

A l'instar des systèmes de vidéo installés sur la voie publique, seules des personnes habilitées sont autorisées à visualiser les images en directe ou enregistrées. Elles doivent en outre être formées et sensibilisées aux règles de mise en œuvre d'un système de vidéo protection.

C'est à l'employeur de fixer la durée de conservation des images et celle-ci doit être en adéquation avec l'objectif visé tel qu'il a été présenté aux salariés.

Si le lieu est ouvert au public il est nécessaire d'obtenir une autorisation par le préfet. Si le lieu n'est pas ouvert au public ou s'il ne filme que des espaces dédiés aux personnels comme les zones de stockage par exemple, il n'y a aucune formalité à entreprendre auprès de la préfecture ou de la CNIL (Commission Nationale de l'Informatique et des Libertés).

2.2.2 Particulier

Les particuliers ont également de plus en plus recours à des caméras pour sécuriser leur habitation et éviter ou limiter les cambriolages. La législation est plus souple car ces installations ne sont pas soumises aux règles de la protection des données personnelles ni à celle du code de la sécurité intérieure.

Par contre, un particulier ne peut filmer que l'intérieur de sa propriété. En aucun cas, il ne peut filmer la voie publique même pour sécuriser sa voiture garée devant son domicile. Il doit en outre, veiller à ne pas porter atteinte à la vie privée des personnes qui accèdent à son habitation et doit également respecter le droit à l'image. Si un particulier emploie des personnes à domicile ou si des professionnels interviennent régulièrement (par exemple personnel de santé), ceux-ci doivent être informés de l'installation des caméras.

2.3 Textes de référence

Nous listons ici quelques-uns des principaux textes de référence qui régissent l'installation, l'usage et le recours abusif aux systèmes de vidéo-surveillance. Ce tableau n'est pas exhaustif. Une liste plus complète peut être trouvée sur le site de la CNIL[4].

Code	Article	Commentaire
Code de la sécurité intérieure	L223-1 (et suivant)	Lutte contre le terrorisme
	L251-1 (et suivant)	
Code du travail	L2323-47	Information/consultation des instances représentatives du personnel
	L1221-9 et L1222-4	Information individuelle des salariés

[4] www.cnil.fr

	L1121-1	Principe de proportionnalité
Code civil	9	Protection de la vie privée
Code pénal	226-1	Enregistrement de l'image d'une personne à son insu dans un lieu privé
	226-18	Collecte déloyale ou illicite
	226-20	Durée de conservation excessive
	226-21	Détournement de la finalité du dispositif
	R625-10	Absence d'information des personnes

Tableau 1 : Table des principaux textes de référence qui régissent l'installation, l'usage et le recours abusif aux systèmes de vidéo surveillance

3 Technologie et limites

Il existe un grand nombre de solutions permettant d'assurer la protection périmétrique d'un site. Chaque système a ses avantages et ses inconvénients et ne peuvent pas forcément être appliqués dans tous les cas. Nous présentons dans ce chapitre, les principales solutions déployées dans le cas d'installations classiques.

3.1 Capteur électrique ou mécanique

Une solution économique consiste à fixer directement sur la clôture un câble piézo-électrique qui sera sensible aux vibrations, choc ou déformation de la clôture. Ces systèmes ont généralement une bonne qualité de détection, sont très simple à installer (lorsque la clôture le permet) et nécessite que très peu de réglage. Les risques de ces systèmes sont les déclenchements intempestifs lorsqu'il y a du vent.

Une solution équivalente est encore plus économique consiste à utiliser des fils tendus associé à un capteur d'écrasement. Ces dispositifs détectent très bien les tentatives d'escalades ou d'écartement ainsi que les coupures. Par contre, ils nécessitent une clôture de bonne qualité. Certains fabricants proposent ce type de solution directement intégré à la clôture.

Dans cette catégorie de capteur, nous pouvons également citer les systèmes enterrés constitués de tube rempli d'un liquide antigel sous pression. Ces systèmes détectent les variations de pression apportées par les personnes lorsqu'elles s'approchent des tubes. Ces dispositifs ont l'avantage d'être très discrets et s'adaptent à la plupart des sols. Par contre, en fonction du niveau de sensibilité, ils ne peuvent pas différencier une personne d'un animal.

3.2 Capteur à ondes électromagnétiques

Nous rangeons dans cette catégorie les principaux capteurs basés sur l'émission et la réception d'une onde électromagnétique. Il existe trois grandes classes de ce type de capteur. Les plus utilisés sont les barrières infrarouges. Classiquement un émetteur infrarouge est placé sur un

poteau et un récepteur est placé sur un autre poteau à quelque mètre. Lorsqu'une personne passe entre les deux poteaux, elle obstrue le rayon de lumière qui n'est plus détecté par le récepteur ce qui déclenche une alarme. Ce sont des capteurs particulièrement performants avec des taux de détection proche de 100% mais qui ne sont pas discriminant. Pour limiter les détections liées au passage des animaux, les constructeurs proposent des barrières avec plusieurs capteurs placés à différentes hauteurs. Seule la « coupure » simultanée de plusieurs faisceaux déclenche une alarme. Bien évidement pour que ces systèmes soient opérationnels il ne faut pas qu'il y ait d'obstacle entre les poteaux. Une autre limitation de ces dispositifs est qu'ils sont relativement sensibles au brouillard.

La technologie radar est également utilisée pour la détection périmétrique. Dans ce cas, c'est l'effet doppler qui est exploité pour permettre la localisation. La qualité de détection est elle aussi très bonne avec une portée de détection pouvant être supérieure à 100m. Contrairement aux barrières infrarouges, les radars ne sont pas sensibles au brouillard et gardent un bon niveau de détection même en cas de forte pluie. Par contre, ce ne sont pas des technologies adaptées lorsque le relief est accidenté.

La troisième classe de capteurs correspond aux barrières hyperfréquences. L'ensemble est composé d'un émetteur générant un champ électromagnétique et d'un récepteur analysant les modifications de ce champ lors du passage d'une personne. Ces barrières hyperfréquence présentent les mêmes avantages et inconvénients que les radars par contre elles ne permettent pas la localisation. En contrepartie, elles sont un peu plus simples à déployer pour un coût moindre.

3.3 Caméras

Il existe donc un éventail assez large de solutions, hors caméra, permettant de réaliser la fonction de détection périmétrique mais toutes les solutions présentées jusqu'à présent ont la même limitation. Elles ne permettent pas de faire, ce que l'on nomme traditionnellement dans le monde de la vidéo surveillance, la levée de doute. Nous avons vu qu'une barrière infra rouge à un taux de détection de 100% ou très approchant et un taux très faible de fausse alarme. Pour simplifier, cela signifie que tout ce qui obstrue le signal est détecté. Ce système, bien que très performant, n'est donc pas discriminant. Lors du déclenchement d'une alarme, il n'est pas possible de savoir ce qui a déclenché l'alarme. Afin de pouvoir être en mesure de qualifier l'alarme et ainsi éviter de se déplacer à chaque fois qu'un lapin passe entre les barrières, la prise d'image devient nécessaire.

Les caméras sont donc un bon moyen de faire cette levée de doute soit en direct soit par la relecture des enregistrements. Finalement, puisque des caméras doivent être installées pour faire de la levée de doute, autant les utiliser aussi pour la détection.

Les caméras sont depuis longtemps utilisées dans le cadre de la vidéo surveillance, mais elles sont restées longtemps utilisées pour la visualisation en directe et pour l'enregistrement. Ce n'est que depuis la fin des années 1990 qu'elles ont commencées réellement à être utilisées pour la tâche de détection d'intrusion. La raison principale de cette utilisation assez récente est

que l'analyse vidéo est une tâche qui demande des ressources de calcul relativement importantes et qu'il a fallu attendre l'arrivée de processeurs suffisamment puissants pour permettre d'analyser plusieurs flux en temps réel pour un coût raisonnable.

3.3.1 Caméra monochrome / couleur

Il existe plusieurs types de caméra utilisés dans le cadre de la video-protection. Indépendamment de la façon dont le flux vidéo peut être transmis (analogique, IP, IEEE1394, *etc.*) nous pouvons grossièrement les classer en trois catégories : classique monochrome ou couleur, infra rouge et thermique.

Les premières caméras réellement utilisées pour la tache de détection ont été les caméras dites « classiques », qui restituent une image en couleur ou en niveau de gris, parce que ce sont celles qui étaient déjà majoritairement utilisées dans le cas de la vidéo-surveillance sans analyse. L'intérêt de ces caméras est qu'elles restituent une image visuellement proche de ce que nous pourrions voir à l'œil nu. De plus, lorsque la résolution est suffisante, elles peuvent également permettre l'identification des personnes ou des objets. En contrepartie, elles ont besoin d'un certain niveau de luminosité dans la scène. De fait, la nuit, elles ont besoins d'un éclairage additionnel pour que le capteur puisse restituer une image suffisamment contrastée. Les constructeurs de caméra ont beaucoup investi dans la capacité de leur caméra à proposer une image propre même lorsque la luminosité est faible. Une des techniques que l'on retrouve encore aujourd'hui consiste à proposer une image couleur lorsque la luminosité est relativement forte et de basculer dans un mode en niveau de gris lorsque la luminosité est plus faible. Ce basculement radical n'est évidemment pas sans poser de problème puisqu'un même algorithme doit pouvoir fonctionner avec une image couleur et une image en niveau de gris. Quel que soit le mode utilisé, les caméras sont également amenées à corriger le gain du capteur de façon à exploiter au mieux sa dynamique. Cette correction peut être faite de façon brusque ou progressive sur l'image. Pour les caméras les plus élaborée, la correction peut se faire indépendante sur plusieurs zones de l'image. Toutes ces transformations ne sont pas sans avoir une incidence sur l'analyse.

Par ailleurs si théoriquement la détection dépend de la profondeur de champ et donc peut être infinie, dans la pratique, la portée de détection reste limitée à 50m voir 100m. Il y a plusieurs raisons à cela et l'une des principales et bien sur la résolution et la taille apparente de l'objet dans l'image. Comme l'image est discrétisé, il faut un nombre minimum de pixels pour une bonne détection. Comme nous l'avons vue, il faut également une luminosité suffisante dans la scène. Mais finalement, le facteur le plus discriminant que l'on néglige trop souvent concerne les conditions climatiques. Une caméra équipée d'un objectif avec un facteur de zoom important peut très bien permettre de visualiser, avec une résolution suffisante, une personne à 100m par temps clair. Par contre dans cette même configuration, si les conditions climatiques se dégradent (brouillard, pluie, neige), la portée peut rapidement être limitée et la détection impossible.

3.3.2 Caméra infra rouge

Afin de palier le problème de la luminosité, les constructeurs de caméra ont proposé des caméras infrarouges. En réalité, c'est un peu un abus de langage parce qu'en fait la plupart des capteurs des caméras classiques sont capables d'acquérir de l'information dans le proche infrarouge allant jusqu'à des longueurs d'onde de 900nm voir 1000 nm (Figure 19). Pour s'en convaincre, il n'est pas rare que les constructeurs ajoutent un filtre IR devant le capteur de façon à limiter la bande passante à celle de l'œil humain, c'est-à-dire entre 400nm et 700nm environ. L'intérêt de ces caméras est alors de les coupler avec un éclairage infra-rouge. Ceci permet donc d'éclairer la scène dans une plage de longueur d'onde qui n'est pas visible par œil humain et ainsi limiter les nuisances. En contrepartie, ces éclairages étant souvent très directif, il faut les placer dans l'axe et dans la direction de la caméra. Ainsi, plusieurs constructeurs ont proposé des caméras avec un éclairage IR intégré.

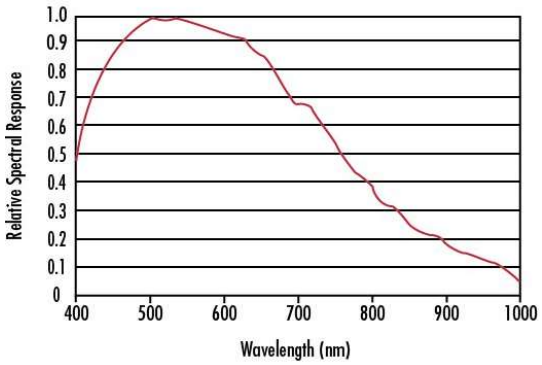


Figure 19 : réponse spectrale dur capteur ICX 285 de Sony où l'on voit que la réponse spectrale du capteur s'étend bien au-delà des 700nm du domaine « visible »

Ces caméras, qui ont inondée le marché de la vidéo surveillance en France autour de l'année 2010, ont été une fausse bonne idée parce que de nombreux problèmes sont apparus. Pour les installateurs elles étaient pourtant intéressantes dans la mesure où elles limitaient les frais d'installation. Pour l'analyse, les choses ont été plus compliquées. Certes, l'éclairage directif permet d'offrir un meilleur contraste lorsqu'une personne passe devant la caméra. Les objets en mouvement étant éclairés face à la caméra apparaissent beaucoup plus clairs que le fond de l'image. Si cela permet une meilleure détection des personnes c'est en revanche plus problématique lorsqu'il pleut. Les gouttes d'eau passant devant l'objectif de la caméra sont alors particulièrement visibles et laissent apparaitre des trainées dans l'image (Figure 20 : gauche). Même s'il existe un certain nombre d'algorithme permettant de filtrer ces détections intempestives, comme nous le verrons par la suite, la multiplicité des événements qui peuvent s'étaler sur plusieurs heures et sur plusieurs caméras d'un même site, engendre immanquablement un nombre non négligeable de fausse alarme.

Par ailleurs, ces caméras ont également été la source d'un autre désagrément inattendu, qui s'il peut paraître anecdotique a néanmoins nécessité des actions correctives. L'éclairage infrarouge intégré aux caméras est à la fois source de luminosité mais aussi de chaleur. Il attire les insectes volant mais également les araignées qui tissent leur toile devant l'objectif (Figure 20 : droite). La toile en elle-même est source de perturbations mais lorsque l'araignée se déplace devant

l'objectif son corps est fortement éclairé. Comme la caméra corrige son gain pour améliorer la dynamique, le reste de la scène devient très sombre et limite la capacité de détection.

Figure 20 : exemple de capture d'une caméra avec éclairage infrarouge intégré : pluie (à gauche), araignée (à droite).

3.4 Caméra thermique

La dernière catégorie de caméras est dite « thermique ». Ces caméras opèrent dans une gamme de rayonnement infrarouge compris généralement entre 3 et 15 μm. Dans le cas de la peau humaine d'une personne en bonne santé, le pic de luminance est observé aux alentours de 9μm. Ces caméras existent depuis de nombreuses années et ont été principalement utilisées pour la thermographie. Elles ont véritablement été utilisées de façon massive dans la vidéo surveillance intelligente depuis le début des années 2010. L'intérêt de ces caméras est qu'elles ne nécessitent pas d'éclairage et sont beaucoup moins sensibles aux conditions météorologiques. Ceci permet d'augmenter sensiblement la distance couverte par une seule caméra et donc de limiter leur nombre. Par ailleurs, elles ne captent pas les ombres portées ce qui simplifie également les algorithmes de traitement. Cependant, elles ne présentent pas que des avantages. L'inconvénient principal est qu'une caméra thermique restitue image dont l'intensité d'un pixel est corrélée à une température. Avec une telle image, il n'est pas possible d'identifier une personne. Par ailleurs, les longueurs d'onde exploitées par les caméras thermiques sont moins énergétiques que dans le domaine visible. Suivant le modèle de caméra et sa qualité, le temps d'exposition peut être allongé, le signal plus ou moins bruité et les corrections de gain plus fréquence que sur des caméras classiques.

4 Analyse de la vidéo

L'analyse automatique de la vidéo est une tâche importante de la vidéo-protection parce qu'elle permet de se passer d'un opérateur humain pour réaliser la tache fastidieuse de détection. L'opérateur humain n'est cependant pas totalement exclu du processus de décisions parce que dans la plupart des installations, il reste l'élément terminal qui valide ou non l'alarme et déclenche si besoin les opérations correctrices. Cependant, l'opérateur n'a plus à visualiser en permanence l'ensemble des vidéos. Le système sélectionne automatiquement les séquences

d'images pertinentes et le rôle de l'opérateur consiste à faire la levée de doute sur ces séquences. Dans le cadre restreint de la détection périmétrique, l'algorithme d'analyse est donc chargé de réaliser la détection de tous les objets en mouvement dans le champ de la caméra, de filtrer et de classer les objets afin de limiter les fausses alarmes et de remonter les événements à l'opérateur ou à un système d'indexation de la vidéo.

4.1 Exploitation des données

L'analyse des vidéos est une tâche qui nécessite une certaine capacité de calcul et dans le cas d'une détection périmétrique, elle doit de plus être réalisée en temps réel. La notion de temps réel dépend fortement de l'application visée et du temps de réaction tout au long de la chaine de traitement de l'alarme. Typiquement, en vidéo protection, l'analyse se fait au rythme de 5 images par secondes environ. C'est généralement la fréquence qui permet de décrypter une action humaine dans le champ couvert par la caméra. Sauf mention contraire, lorsque nous parlerons de « temps réel » dans ce manuscrit, nous serons dans ce cadre-là. Cependant, certaines situations peuvent nécessiter des fréquences plus élevées lorsque les mouvements à détecter sont rapides et près de la caméra.

L'analyse peut être réalisée au plus près de la prise de vue voire même directement dans la caméra elle-même. Dans le cas de la protection périmétrique, plusieurs fabricants de caméras proposent des algorithmes de détection ou proposent un mécanisme permettant d'ajouter un algorithme tiers dans leurs caméras. Cependant, les capacités de calcul dans la caméra restant relativement limité, cette solution est souvent moins efficace qu'une analyse sur un serveur dédié.

De même la plupart des algorithmes analysent le flux à la volé sans connaissance de l'avenir. Là encore, il s'agit de limiter la puissance de calcul et l'espace mémoire. Cependant comme nous le verrons par la suite, ce n'est pas une généralité et plusieurs solutions ne traitent pas le flux image par image mais une séquence entière. Dans ce cas, la séquence à analyser est sélectionnée par un autre système qui peut être un capteur, comme ceux présenté plus haut, ou une analyse de mouvement plus basique. L'analyse d'une séquence présélectionnée est aussi un moyen de déporter l'analyse sur un serveur correctement dimensionné.

Enfin, nous pouvons également citer le cas où l'analyse est réalisée à postériori. Dans ce cas, la contrainte de « temps réel » peut être relâchée de même que l'avenir peut être exploité. Par « l'avenir », nous entendons que la sanction sur une image t peut être déterminée à partir des images t + n.

4.2 Prérequis

La détection d'intrusion, en particulier en extérieur, est une activité d'analyse vidéo très sensible à l'environnement. Le matériel, les conditions de prise de vue, l'éclairage, peuvent avoir un impact direct sur les performances et la qualité des résultats. Le capteur de détection extérieure par analyse vidéo le plus adapté est la caméra thermique. Afin d'optimiser la pertinence de la détection, il n'est pas rare de rappeler un certain nombre de précautions à prendre en compte :

- L'intrus doit représenter une taille minimale dans l'image et doit être vue en entier.
- Eviter de visualiser le ciel et régler la casquette de la caméra de façon à réduire les problèmes d'éblouissement de la caméra et les gouttes de pluie sur l'objectif. Cela permet aussi de maximiser la zone de détection. Le produit type « mini-dôme » fixe est donc particulièrement à éviter.
- Un nettoyage régulier des caméras est recommandé pour prévenir des dégradations de l'optique à cause des salissures/insectes.
- La caméra doit être fixée sur un mur ou un mât et ne pas subir de trop fortes vibrations.
- Ne pas utiliser de caméras « dôme motorisé » car les opérateurs pourraient l'utiliser, et la fiabilité et la précision de pré-positionnement peut-être un élément aléatoire. En plus de provoquer des fausses alarmes, cela ne permet pas de garantir la couverture de la zone de détection (quand une caméra dôme regarde une zone, elle ne regarde pas ailleurs). La caméra doit idéalement être en position fixe pour réaliser l'analyse de l'image. La caméra fixe est donc l'outil idéal pour la fonction de détection extérieure par analyse vidéo.

Dans le cas des caméras « classiques » c'est-à-dire qui opèrent dans le domaine visible, des précautions supplémentaires peuvent être exigées notamment en ce qui concerne l'éclairage. Le champ de détection de la caméra doit être correctement éclairé de façon à ce qu'une intrusion de nuit soit visible.

4.3 Etude de pré-déploiement

Avant de placer les caméras sur un site, il est souvent nécessaire de faire une étude du projet et avoir un aperçu concret du matériel nécessaire pour le réaliser afin d'assurer la meilleure protection périmétrique possible. L'étude permet également de se rendre compte des obstacles qui peuvent impacter la visibilité des caméras, comme les murs, poteaux, haie, *etc*. (Figure 21)

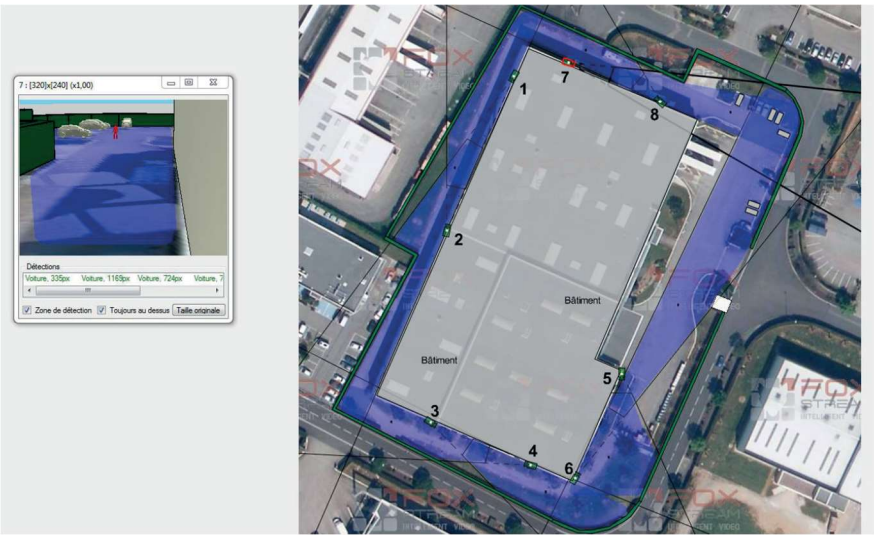


Figure 21 : exemple d'étude d'un site avec l'outil FoxTool la société Foxstream

Lors de cette étude, des fonctionnalités 3D permettent de simuler les conditions réelles propres à chaque site en ajoutant des objets tels que des voitures, murs, bâtiments et d'observer leur impact sur la détection, pour ajuster si nécessaire l'emplacement des caméras, le choix du modèle ou celui de la focale.

Afin que cette étude soit la plus réaliste possible, l'opérateur est amené à renseigner les champs suivants :

- Résolution : déterminant pour la taille finale d'une personne/d'un objet à l'écran
- Le type de capteur CCD utilisé. Cette information est importante car elle intervient dans le calcul du champ de vision
- La focale de base de la lentille.
- La position 2D de la caméra sur le plan,
- La hauteur où est fixée la caméra
- L'orientation et l'inclinaison de la caméra

A partir de ces informations et d'un modèle de projection adapté, il est possible de reproduire les conditions de prise de vue (Figure 22).

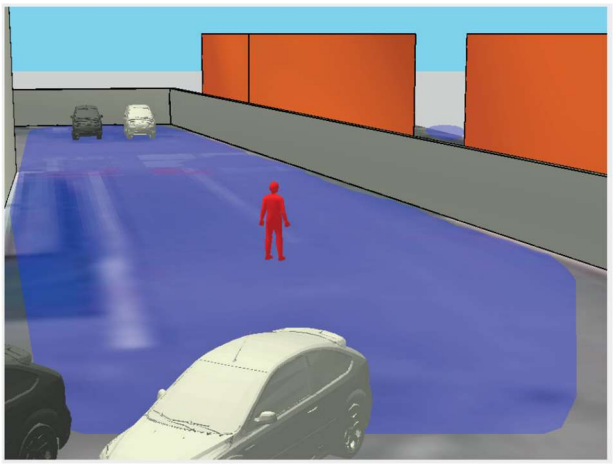


Figure 22 : exemple de projection simulée avec l'outil FoxTools

III - Modèles Mathématiques et définitions

1 Introduction

Afin de pouvoir appliquer des traitements mathématiques aux flux vidéo, il est nécessaire de pouvoir s'appuyer sur un certain nombre de concept de bas niveau ainsi que sur de modèles mathématiques. L'objectif de ce chapitre est de rappeler ces derniers. Les modèles que nous allons utiliser vont volontairement être simplifiés afin de répondre aux contraintes et aux exigences du domaine d'applications dans lequel nous évoluons. Ces simplifications vont immanquablement engendrer une perte ou une déformation de l'information originale, mais elles restent cependant valables dans les situations que nous cherchons à caractériser et limite la complexité algorithmique. Nous allons utiliser différents modèles à chaque étape du processus d'acquisition d'un flux vidéo numérique que nous pouvons représenter comme sur la Figure 23 :

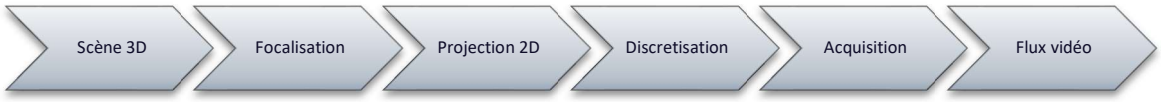


Figure 23 : processus d'acquisition d'une scène par une caméra numérique

Ce processus est très simplifié mais il permet de montrer les étapes essentielles de la restitution d'un flux vidéo. Il va notamment nous permettre de définir la place des différents modèles mathématiques et des concepts de base que nous allons ensuite utiliser. Nous avons indiqué ici le processus d'acquisition d'une caméra numérique. Nous pouvons facilement modifier cette représentation pour nous placer dans le cas des caméras analogiques. Comme le flux vidéo analogique doit être digitalisé pour que nous puissions le traiter, il suffit d'inverser les étapes d'acquisition et de discrétisation dans le schéma ci-dessus.

2 Focalisation

Nous rappelons dans ce chapitre quelques éléments fondamentaux de l'optique géométrique ainsi que quelques définitions. Après ces quelques notions simples, nous présentons un résumé du modèle sténopé. Pour plus d'information, le lecteur pourra se référer aux ouvrages [28], [29], et [30].

2.1 Les bases de l'optique géométrique

Le concept de rayon de lumière introduit par Euclide est l'élément de base de l'optique géométrique. Ce rayon de lumière n'a pas d'existence physique, mais il indique la direction de propagation de la lumière. La propagation de la lumière est étudiée en lui associant une infinité de rayons de lumière indépendants les uns des autres. Un ensemble de rayons de lumière provenant d'une même source est appelé faisceau. Cette représentation a pour principal intérêt de dissocier le trajet de la lumière de l'onde électromagnétique dont elle est composée, ceci afin de permettre de traiter les problèmes d'optique par de simples constructions géométriques. Dans le langage courant, il n'est pas rare d'entendre parler de « rayons lumineux ». C'est bien sûr un abus de langage et il n'y a qu'au cinéma que l'on voit des rayons de lumière qui soit lumineux. Il peut arriver que l'on voie la trace d'un faisceau de lumière mais dans ce cas c'est parce qu'il y a des particules dans le milieu de propagation qui diffractent une partie des rayons.

Avec le concept de rayon de lumière, l'optique géométrique repose sur quelques principes et lois simples. Tout d'abord, le principe de la propagation rectiligne dans un milieu transparent, homogène et isotrope. C'est l'un des premiers principes à avoir été énoncé. L'air, par exemple, est un milieu qui ne répond pas à cette définition puisque son indice absolu varie en fonction de la température et de la pression. Une autre limite du domaine de validité de la propagation rectiligne de la lumière correspond au phénomène de diffraction que l'on peut observer lorsque l'on fait passer un faisceau de lumière à travers un trou mince. Un autre principe important est l'indépendance des rayons de lumière. Pour simplifier, si deux faisceaux de lumière provenant de deux sources différences et dont les directions se croisent en un point, le point d'intersection ne provoque pas d'interférence particulière entre les rayons. Un dernier principe fondamental correspond au *retour inverse* de la lumière. Le trajet emprunté par la lumière dans un sens est le même dans l'autre sens (réciprocité du rayon incident). Parmi les lois de l'optique géométrique, celles de Snell-Descartes sont particulièrement importantes. Elles décrivent les phénomènes lumineux engendrés par l'impact de la lumière sur la matière : la réflexion et la réfraction. Pour reprendre la vision de Descartes, la réflexion correspond au « rebond » de la lumière sur une surface plane. De façon plus formelle, le rayon réfléchi est dans le plan d'incidence, plan défini par le rayon incident et la normale à la surface réfléchissante. La réfraction est la déviation des rayons lumineux passant obliquement d'un milieu transparent à un autre milieu transparent d'indice différent. Lors de la réfraction sur la surface dioptrique, le rayon de lumière coupe la normale et se réfracte selon une direction définie par l'angle i_2 lié à l'angle d'incidence i_1 par la relation des sinus :

$$n_1 \cdot sin\, i_1 = n_2 \cdot sin\, i_2$$

où n_1 et n_2 sont les indices de réfraction des deux milieux.

2.2 Systèmes optiques

Un système optique est un ensemble de milieux transparents, homogènes et isotropes d'indices de réfraction différents disposés les uns à la suite des autres. Ils sont séparés par des surfaces polies, appelées dioptres, de formes géométriques simples (plans, sphères) ou plus complexes

(paraboloïdes, ellipsoïdes, hyperboloïdes). Ces surfaces peuvent être réfringentes (ou réfractantes). C'est-à-dire qu'une fraction plus ou moins importante de l'énergie incidente est transmise d'un milieu à l'autre. Dans le cas où la *quasi*-totalité de l'énergie incidente est réfléchie, on parle alors de surface réfléchissante (miroir). Les systèmes optiques peuvent être classés en trois catégories :

- les systèmes dioptriques ne comportant que des surfaces réfringentes. Ces systèmes ont une face d'entrée et une face de sortie distincte ;
- les systèmes catoptriques ne comportant que des surfaces réfléchissantes ;
- les systèmes catadioptriques comportant à la fois des éléments réfringents et réfléchissants.

2.3 Conditions de Gauss

Afin de simplifier l'étude des phénomènes optiques, nous nous plaçons généralement dans les conditions de Gauss. C'est en publiant ses « Dioptrische Untersuchungen » entre 1838 et 1841 que Carl Friedrich Gauss (1777 – 1855) introduit les notions de plans principaux et points principaux. Son objectif est, sous réserve de certaines conditions (dites conditions de Gauss), de disposer d'un outil mathématique permettant justement de faciliter l'étude des instruments optiques (télescope, jumelle). L'approximation de Gauss consiste, en lumière monochromatique, à n'utiliser que les trajets des rayons pour lesquels le système fonctionne dans des conditions de stigmatisme approché. Le stigmatisme réel stipule que l'image d'un point est un point. Le stigmatisme est dit approché lorsque l'image d'un point est une tache suffisamment petite pour être considérée comme un point. Pour être dans les conditions de Gauss :

- les points objets doivent être voisins de l'axe optique ;
- les rayons utilisés pour la formation des images sont très peu inclinés.

2.4 Grandissement

Si l'on se place dans les conditions de Gauss, alors on considère que la diffraction de la lumière et les aberrations de la lentille sont négligeables. On peut tracer le diagramme de la Figure 24 :

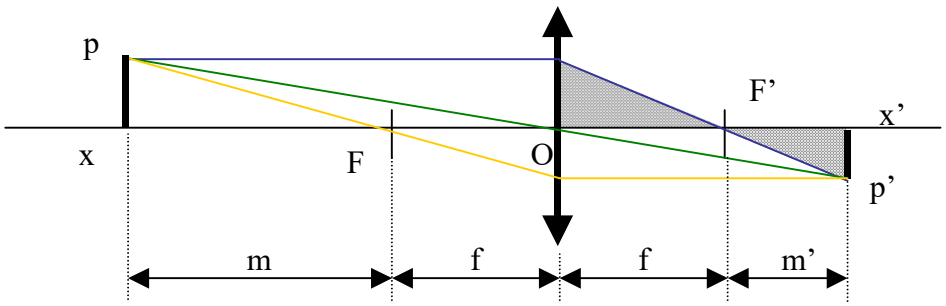


Figure 24 : diagramme de Gauss

Où F est le foyer objet, F' le foyer image et f la distance focale.

De ce diagramme, nous pouvons également noter quelques propriétés importantes :

- Un rayon parallèle à l'axe optique qui atteint la lentille passe par le foyer image en sortant,
- Un rayon qui passe par le centre optique de la lentille n'est pas dévié,
- Un rayon qui passe par le foyer objet et qui atteint la lentille ressort parallèle à l'axe optique.

On note g le grandissement correspondant au rapport entre la projection de l'objet sur le plan image et l'objet.

$$g = \frac{\overline{x'p'}}{\overline{xp}}$$

Le grandissement peut également s'exprimer s'exprime aussi de la forme :

$$g = \frac{m + f'}{m + f}$$

Sur le diagramme, on remarque également que les deux triangles grisés sont semblables. Il en va de même pour les deux triangles associés côté objet. Le grandissement peut donc s'exprimer :

$$g = \frac{m'}{f} = \frac{f}{m}$$

De cette dernière relation, on en déduit :

$$m \times m' = f^2$$

2.5 Ouverture ou Diaphragme

Le nombre d'ouverture n, contrôlé par le diaphragme, détermine la quantité de lumière qui atteint le capteur CCD. Il détermine aussi la profondeur de champ. Le nombre d'ouverture correspond au rapport entre la distance focale et le diamètre du faisceau utile au niveau de la lentille.

$$n = \frac{f}{D}$$

où n est le nombre d'ouverture, f la distance focale et D le diamètre du faisceau au niveau de la lentille.

Lorsqu'on ferme le diaphragme (Figure 25), on augmente le nombre d'ouverture. A l'inverse, lorsqu'on ouvre le diaphragme (Figure 26), on diminue le nombre d'ouverture.

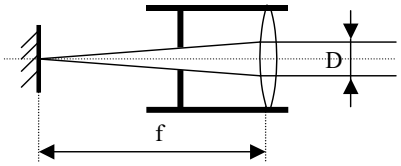


Figure 25 : diaphragme fermé, ouverture grande

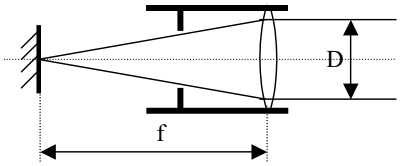


Figure 26 : diaphragme ouvert, ouverture petite

2.6 Netteté et profondeur de champ

Si l'on place le plan image à une distance m' du foyer image, alors seuls les points situés à une distance m du foyer objet seront nets. Dans le diagramme suivant (Figure 27), l'objet est placé plus loin que la distance m. On remarque alors que le point P ne se projette plus sur un point p unique sur le plan image, mais sur un petit disque. On a donc une image floue. Nous attirons l'attention du lecteur sur la signification du terme « image » que nous employons. En optique, une image est forcément nette, sinon c'est une tache. Dans ce qui suit, nous utilisons le terme « image » suivant sa signification dans la vie courante : reproduction visuelle d'un objet réel.

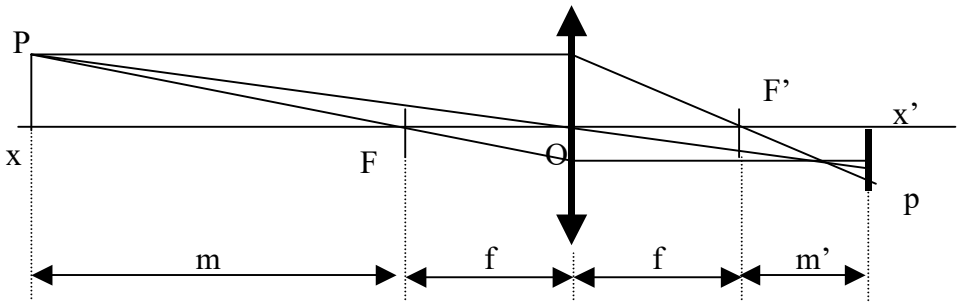


Figure 27 : exemple de représentation de Gauss où l'objet est n'est pas placé à la bonne distance et apparait flou dans l'image

La profondeur de champ est l'intervalle de distance pour lequel l'image reste nette. En toute rigueur, cette distance est donc nulle quelle que soit la distance focale. Dans la pratique, on se donne une tolérance (Figure 28). On considère que l'image est nette tant que le diamètre de la tache correspondant à la projection d'un point p unique est inférieur à ε. En photographie, pour une pellicule 24 x 36, on fixe généralement :

$$\varepsilon = 30\mu m$$

Dans le cas d'une caméra numérique, la valeur ε sera celle du diamètre du cercle inscrit dans une cellule correspondant à un pixel. Pour un capteur ¼' et 752 pixels par ligne, $\varepsilon = 4.3\mu m$

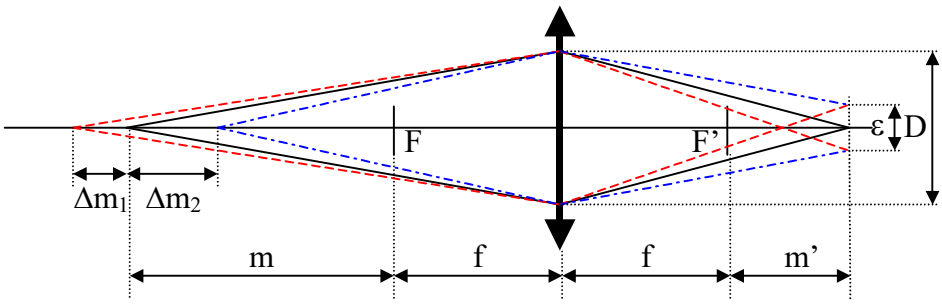


Figure 28 : représentation de la profondeur de champ dans le diagramme de Gauss

Si Δm1 et Δm2 correspondent aux incréments de distance pour lequel la projection d'un point sur le plan image est inférieure ou égale à ε, la profondeur de champ est donnée par :

$$\Delta m_1 + \Delta m_2 = \frac{2 \cdot f^2 \cdot \varepsilon \cdot n \cdot m(m+f)}{f^4 - \varepsilon^2 \cdot n^2 \cdot m^2} \quad \text{avec } n = \frac{f}{D}$$

De cette équation, nous pouvons tirer les enseignements suivants :

- si l'ouverture augmente, c'est à dire si on ferme le diaphragme, la profondeur de champ augmente,
- si la focale diminue, la profondeur de champ augmente,
- si la distance de mise au point m augmente, la profondeur de champ augmente.

3 Projection 2D

3.1 Camera obscura

L'acquisition d'une image à partir d'un capteur CCD est un phénomène complexe. Le phénomène optique est plus simple à comprendre puisqu'il s'agit d'un phénomène lumineux naturel que l'on peut expérimenter facilement. C'est l'expérience très simple de la chambre noire. Aristote est vraisemblablement le premier à avoir expérimenté ce phénomène. Le 25 janvier 1544 à Louvain, Reinerus Gemma-Frisius a utilisé le principe de la chambre noire ou « caméra obscura » pour observer une éclipse du soleil. Il utilisera plus tard l'illustration suivante (Figure 29) dans son livre « de radio astromomica et geometrico » publié en 1545. Il semble que ce soit la première illustration de la chambre noire.

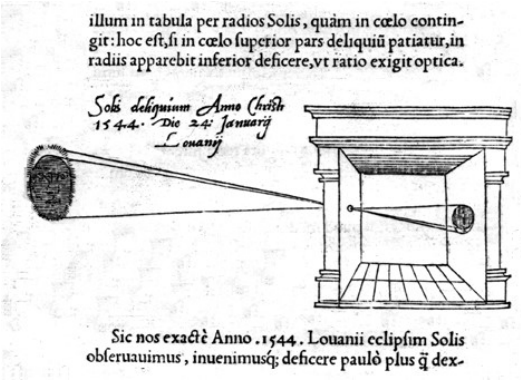


Figure 29 : illustration de la « Camera obscura » - Reinerus Gemma-Frisius – 1545

Les peintres du XVI^{ième} siècle connaissaient bien ce phénomène et comprirent très vite le bénéfice qu'ils pouvaient en retirer. Ils construisirent d'immenses chambres noires, parfois transportables, qui leur permettaient, comme avec un calque, de reproduire des paysages extérieurs. Le trou, appelé sténopé, ne devait pas être trop grand pour donner une projection nette, mais pas trop petit non plus pour laisser passer suffisamment de lumière. Il fut employé par de nombreux artistes, dont Giovanni Baptista della Porta, Vermeer, Guardi et Antonio Canal, dit Canaletto, qui l'utilisa notamment pour mettre en perspective ses célèbres paysages des canaux de Venise. Le dispositif fut amélioré en utilisant une lentille convergente qui donnait le même résultat avec une luminosité et une netteté meilleure. Le système se perfectionna en plaçant des boîtes coulissantes les unes dans les autres et par l'utilisation d'un miroir incliné à 45° pour redresser l'image. On essaya à maintes reprises de fixer l'image sur un support. En 1826, Nicéphore Niepce réussit à obtenir, après une exposition de 8 heures, une image stable sur une plaque d'étain enduite de bitume de Judée. La photographie venait de naître.

3.2 Définitions

Le modèle sténopé (pinhole) est le modèle projectif le plus souvent utilisé (Figure 30). Il permet un certain nombre de simplifications. Ce modèle est défini par deux éléments : un point, appelé centre optique et un plan ne contenant pas le point, appelé plan rétinien.

Dans ce modèle, le repère de la scène et le repère de la caméra sont composés d'axes orthogonaux. Le repère de la scène W est défini par son origine O_w et par ses trois axes (X_w, Y_w, Z_w). Le repère de la caméra C est défini par son origine O_c et par ses trois axes (X_c, Y_c, Z_c). Le point origine O_c correspond également au centre optique de la caméra. L'axe optique est normal au plan rétinien appelé aussi plan image Π, et coupe le plan au point p. A des fins de simplification, nous considérons que l'axe optique est confondu avec l'axe Z_c. Le plan Π est défini par deux vecteurs orthogonaux u et v. Dans ce repère, le point p a pour coordonnées (u_0, v_0). La distance O_cP correspond à la distance focale f.

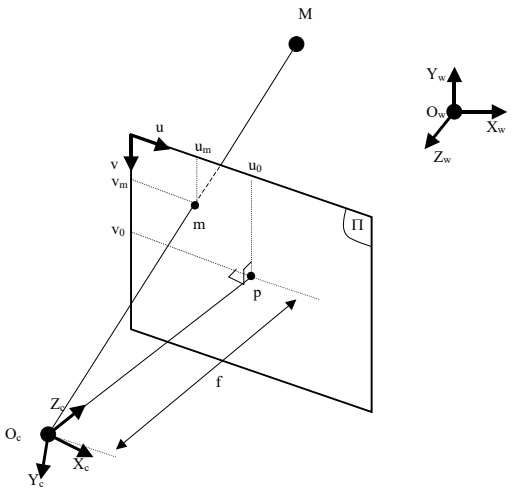


Figure 30 : représentation simplifiée du model sténopé

La projection d'un point M de la scène de coordonnées (x_w, y_w, z_w) sur le plan image correspond au point m de coordonnée (u_m, v_m).

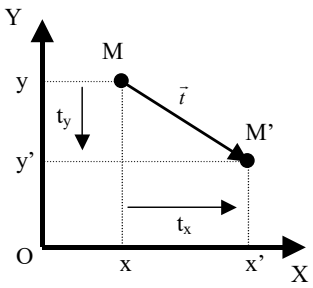
Dans cette représentation simplifiée du modèle sténopé, les paramètres extrinsèques sont les paramètres permettant le changement de repère de la scène vers le repère caméra. Les paramètres intrinsèques sont la distance focale f et la position du point p dans le repère image. Il est à noter que le point p ne correspond pas forcément au centre de l'image. Dans la pratique, les constructeurs de caméras s'efforcent de faire correspondre ces deux points, mais des imperfections dans la réalisation de l'optique et le jeu mécanique nécessaire au montage engendrent un léger décalage dont il faudra tenir compte en fonction de la précision visée.

3.3 Paramètres extrinsèques

Le point M est défini par ses coordonnées (x_w, y_w, z_w) dans le repère de la scène, mais il peut également être défini par ses coordonnées (x_c, y_c, z_c) dans le repère de la caméra. Le plan image étant orthogonal à l'axe optique qui est lui-même confondu à l'axe Zc, un changement de repère est inévitable. Les paramètres extrinsèques sont donc les paramètres qui permettent le calcul du changement de repère entre le repère de la scène et le repère de la caméra.

3.3.1 Transformation rigide 2D

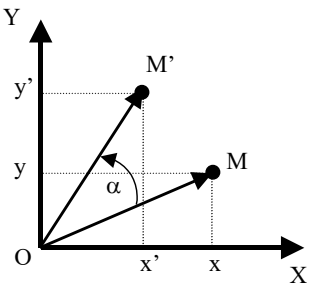
Dans le cas simple d'une transformation 2D, la translation de vecteur $\vec{t}$ d'un point M, s'écrit $\overrightarrow{OM'} = \overrightarrow{OM} + \vec{t}$,



soit en notation matricielle : $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$

En notation matricielle, la rotation α d'un point M, s'écrit :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\ \alpha & -sin\ \alpha \\ sin\ \alpha & cos\ \alpha \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix}$$



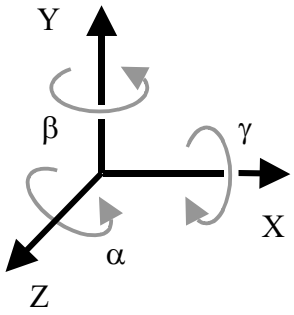
La transformation rigide 2D, composée d'une translation et d'une rotation, s'écrit donc :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\ \alpha & -sin\ \alpha \\ sin\ \alpha & cos\ \alpha \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

3.3.2 Transformation rigide 3D

Dans le cas 3D qui nous intéresse, la transformation rigide se compose de trois translations et d'une décomposition des rotations autour des trois axes.

- L'angle α autour de l'axe Z (de X vers Y) correspond au roulis (roll)
- L'angle β autour de l'axe Y (de Z vers X) correspond au lacet (pan)
- L'angle γ autour de l'axe X (de Y vers Z) correspond au tangage (tilt)



En notation matricielle, la décomposition des rotations s'écrit :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Le développement de l'expression ci-dessus donne le résultat suivant :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\alpha\cdot\cos\beta & (\cos\alpha\cdot\sin\beta\cdot\sin\gamma - \sin\alpha\cdot\cos\gamma) & (\sin\alpha\cdot\sin\gamma + \cos\alpha\cdot\sin\beta\cdot\cos\gamma) \\ \sin\alpha\cdot\cos\beta & (\cos\alpha\cdot\cos\gamma + \sin\alpha\cdot\sin\beta\cdot\sin\gamma) & (\sin\alpha\cdot\sin\beta\cdot\cos\gamma - \cos\alpha\cdot\sin\gamma) \\ -\sin\beta & \cos\beta\cdot\sin\gamma & \cos\beta\cdot\cos\gamma \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Cette matrice de transformation est notée R.

Nous pouvons remarquer que cette matrice R est une matrice orthogonale puisque quels que soient les angles α, β et γ, le produit :

$$R^tR = Id$$

où Id est la matrice identité.

Intuitivement, nous pouvons formuler que, quelle que soit la rotation que l'on fait subir au trièdre formant le repère, l'orthogonalité des axes formant ce repère est conservée après la transformation. Cette propriété d'orthogonalité de la matrice R, nous permet également d'écrire que la matrice inverse est égale à sa transposée :

$$R^{-1} = R^t$$

La transformation rigide 3D s'écrit donc :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} \cos\alpha\cdot\cos\beta & (\cos\alpha\cdot\sin\beta\cdot\sin\gamma - \sin\alpha\cdot\cos\gamma) & (\sin\alpha\cdot\sin\gamma + \cos\alpha\cdot\sin\beta\cdot\cos\gamma) \\ \sin\alpha\cdot\cos\beta & (\cos\alpha\cdot\cos\gamma + \sin\alpha\cdot\sin\beta\cdot\sin\gamma) & (\sin\alpha\cdot\sin\beta\cdot\cos\gamma - \cos\alpha\cdot\sin\gamma) \\ -\sin\beta & \cos\beta\cdot\sin\gamma & \cos\beta\cdot\cos\gamma \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

ou de manière simplifiée, en introduisant les coordonnées homogènes :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

En utilisant la notation $r_i = (r_{i1} \quad r_{i2} \quad r_{i3})$, cette transformation s'écrit :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_1 & t_x \\ r_2 & t_y \\ r_3 & t_z \\ 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Les trois angles α, β et γ sont dits du cadran. Il ne faut pas les confondre avec les angles d'Euler (Figure 31). Les angles d'Euler permettent également la transformation d'un référentiel OXYZ en référentiel Oxyz. Cependant, les trois angles ψ, θ et φ portent des noms liés à leur application en astronomie.

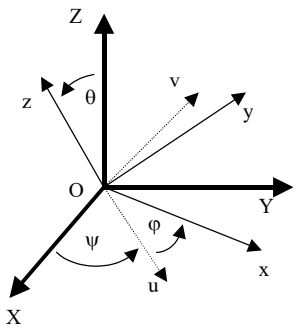


Figure 31 : représentation des angles d'Euler

Où ψ est l'angle de précession, θ est l'angle de nutation et φ est l'angle de la rotation propre.

Cette décomposition fait intervenir un référentiel intermédiaire Ouvz.

Les paramètres extrinsèques sont donc *in fine* au nombre de 6 : trois angles de rotation α, β, γ et trois translations t_x, t_y, t_z.

3.4 Paramètres intrinsèques

Une fois le changement de repère effectué, nous avons besoin de calculer la projection des points du repère caméra sur le plan image. Dans le cas du modèle simplifié, on rappelle que l'axe optique est confondu avec l'axe Z_c et que le plan image est orthogonal à cet axe, c'est-à-dire parallèle au plan X_cY_c (Figure 32). Un calcul supplémentaire permet de convertir les unités métriques du repère de la caméra en unités pixels de l'image. Les paramètres intrinsèques de la caméra sont donc les paramètres qui permettent ces transformations.

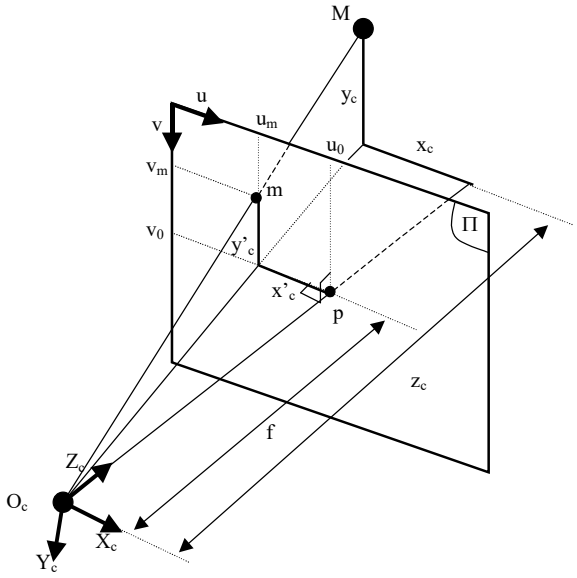
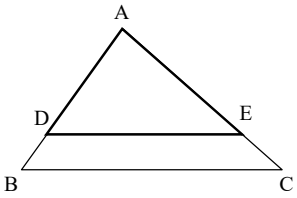


Figure 32 : modèle sténopé

3.4.1 Projection perspective

D'après le théorème de Thales, si dans un triangle ABC, D est un point de [AB], E est un point de [AC] et si les segments [BC] et [DE] sont parallèles alors on a la relation :



$$\frac{AD}{AB} = \frac{AE}{AC} = \frac{DE}{BC}$$

Donc d'après Thales :

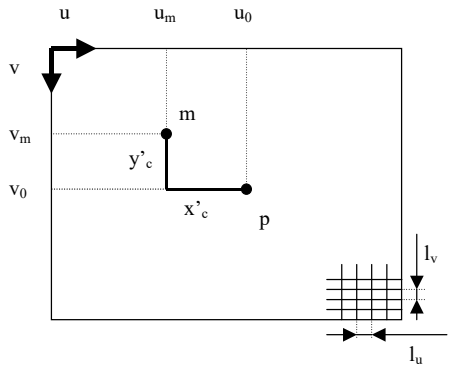
$$\begin{cases} \frac{x_c'}{f} = \frac{x_c}{z_c} \\ \frac{y_c'}{f} = \frac{y_c}{z_c} \end{cases} \Rightarrow \begin{cases} x_c' = f\frac{x_c}{z_c} \\ y_c' = f\frac{y_c}{z_c} \end{cases}$$

3.4.2 Changement d'unité

Le changement d'unité permet la transformation de l'unité métrique du repère caméra à l'unité pixelique du repère (u,v) de l'image. Contrairement au repère caméra, le repère de l'image n'est pas homogène du fait de la nature même de la matrice du capteur.

Cette transformation s'écrit simplement :

$$\begin{cases} u_m = u_0 + \frac{x_c'}{l_u} \\ v_m = v_0 + \frac{y_c'}{l_v} \end{cases}$$



où l_u est la largeur d'un pixel, l_v la hauteur d'un pixel, u_0 et v_0 les coordonnées du point p, centre optique de l'image.

En regroupant les deux expressions, on obtient :

$$\begin{cases} u_m = u_0 + \frac{f}{l_u} \cdot \frac{x_c}{z_c} \\ v_m = v_0 + \frac{f}{l_v} \cdot \frac{y_c}{z_c} \end{cases} \Rightarrow \begin{cases} u_m = u_0 + k_u \cdot \frac{x_c}{z_c} \\ v_m = v_0 + k_v \cdot \frac{y_c}{z_c} \end{cases} \text{ avec } \begin{cases} k_u = \frac{f}{l_u} \\ k_v = \frac{f}{l_v} \end{cases}$$

Les paramètres intrinsèques sont donc au nombre de 4 :

- k_u, la distance focale en pixels horizontaux,
- k_v, la distance focale en pixels verticaux,
- u_0, v_0, les coordonnées du centre optique de l'image.

De façon à rendre homogènes les matrices des paramètres intrinsèques et extrinsèques, les expressions des paramètres intrinsèques peuvent se mettre sous la forme :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \text{ avec } \begin{cases} u = \frac{U}{W} \\ v = \frac{V}{W} \end{cases}$$

Soit, en notation simplifiée :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = [A] \bullet \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

où A est la matrice des paramètres intrinsèques.

3.5 Formulation du modèle sténopé

En regroupant les expressions des paramètres intrinsèques et extrinsèques, la formulation du modèle sténopé s'écrit :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

L'expression de la projection perspective s'écrit généralement sous la forme :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

avec :

$$\begin{cases} m_{11} = k_u \cdot \cos\alpha \cdot \cos\beta - u_0 \cdot \sin\beta \\ m_{12} = k_u \cdot (\cos\alpha \cdot \sin\beta \cdot \sin\gamma - \sin\alpha \cdot \cos\gamma) + u_0 \cdot \cos\beta \cdot \sin\gamma \\ m_{13} = k_u \cdot (\sin\alpha \cdot \sin\gamma + \cos\alpha \cdot \sin\beta \cdot \cos\gamma) + u_0 \cdot \cos\beta \cdot \cos\gamma \\ m_{14} = k_u \cdot t_x + u_0 \cdot t_z \\ m_{21} = k_v \cdot \sin\alpha \cdot \cos\beta - v_0 \cdot \sin\beta \\ m_{22} = k_v \cdot (\cos\alpha \cdot \cos\gamma + \sin\alpha \cdot \sin\beta \cdot \sin\gamma) + v_0 \cdot \cos\beta \cdot \sin\gamma \\ m_{23} = k_v \cdot (\sin\alpha \cdot \sin\beta \cdot \cos\gamma - \cos\alpha \cdot \sin\gamma) + v_0 \cdot \cos\beta \cdot \cos\gamma \\ m_{24} = k_v \cdot t_y + v_0 \cdot t_z \\ m_{31} = -\sin\beta \\ m_{32} = \cos\beta \cdot \sin\gamma \\ m_{33} = \cos\beta \cdot \cos\gamma \\ m_{34} = t_z \end{cases}$$

Dans cette expression, on remarque que les coefficients m_{31}, m_{32}, m_{33} et m_{34} ne dépendent pas des paramètres intrinsèques de la caméra.

3.6 Distorsion de l'image

Dans la plupart des applications, seule la distorsion radiale pose réellement un problème. Cette distorsion de l'image se manifeste généralement par la courbure vers l'extérieur de l'image des droites verticales et horizontales (Figure 34). L'effet de la distorsion est d'autant plus visible que l'on s'éloigne du centre de l'image. L'exemple de distorsion radiale que l'on peut observer facilement est celui du juda des portes d'habitation. Lorsque nous regardons à travers, toutes les lignes droites apparaissent courbes. Ces aberrations sont essentiellement dues à la géométrie de la lentille. Les rayons de lumière qui passent par les bords de la lentille sphériques classiques convergent en un point légèrement décalé par rapport au rayon qui passe par le centre. Ce phénomène bien connu est également appelé « aberration sphérique » (Figure 33).

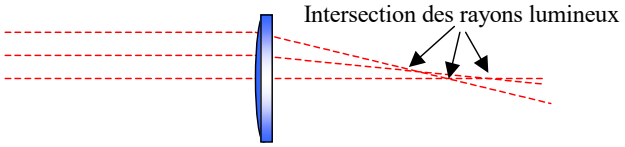


Figure 33 : distorsion radiale avec les lentilles sphériques classiques

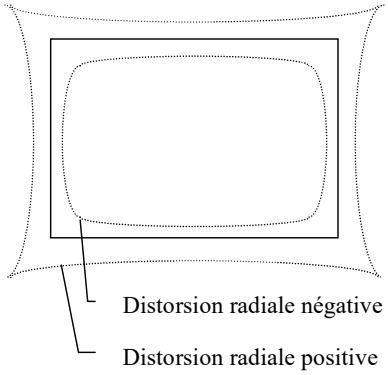


Figure 34 : effet de la distorsion radiale sur l'image

Ce type d'aberration peut être corrigé analytiquement. Nous verrons par la suite un modèle mathématique permettant la correction. Cependant, cette correction se fait avec une perte d'information. Une autre solution consiste à diminuer l'angle d'ouverture des lentilles de façon à retomber dans le cas de la dioptrique de Gauss. L'inconvénient est que cette solution diminue la luminosité des images. La plupart des constructeurs proposent aujourd'hui des lentilles asphériques. Une surface non sphérique permet de faire converger les rayons lumineux du bord de la lentille et du centre vers un foyer unique.

La distorsion radiale peut être modélisée par un polynôme de degré n. Dans la pratique, on utilise un polynôme de degré 4 en ne gardant que les puissances paires. Cette simplification est réputée pour être suffisante. La formulation de la distorsion radiale est donc la suivante :

$$\begin{cases} \hat{u} = u + (u - u_0) \cdot \left(k_1 \cdot r^2 + k_2 \cdot r^4\right) \\ \hat{v} = v + (v - v_0) \cdot \left(k_1 \cdot r^2 + k_2 \cdot r^4\right) \end{cases}$$

où $r = \sqrt{(u - u_0)^2 + (v - v_0)^2}$ et (k_1, k_2) sont les coefficients de la distorsion radiale.

3.7 Formulation complète

Ces aberrations sont essentiellement dues à la géométrie de la lentille. La distorsion peut être modélisée de la façon suivante :

$$\begin{cases} u_m = u_0 + k_u \bullet \dfrac{x_c + \Delta x_c}{z_c} \\ v_m = v_0 + k_v \bullet \dfrac{y_c + \Delta y_c}{z_c} \end{cases}$$

avec :

$$\begin{cases} \Delta x_c = k_1 \cdot x \cdot (x_c^2 + y_c^2) + p_1 \cdot (3x_c^2 + y_c^2) + 2p_2 \cdot x_c \cdot y_c + s_1 \cdot (x_c^2 + y_c^2) \\ \Delta y_c = k_2 \cdot y \cdot (x_c^2 + y_c^2) + p_2 \cdot (3x_c^2 + y_c^2) + 2p_1 \cdot x_c \cdot y_c + s_2 \cdot (x_c^2 + y_c^2) \end{cases}$$

où : (k_1, k_2) sont les coefficients de la distorsion radiale, (p_1, p_2) sont les coefficients de la distorsion tangentielle due au décentrage de l'axe optique et (s_1, s_2) sont les coefficients de la distorsion tangentielle due à la nature du prisme. 90 % de la distorsion est due au paramètre k_1.

4 Discrétisation

4.1 Capteur numérique

Les équations que nous venons de présenter permettent de modéliser la projection de l'image de la scène focalisée par l'objectif sur le capteur. Dans le cas des caméras numériques, la nature intrinsèque du capteur va imposer une discrétisation de l'image. Il existe deux grandes familles de capteur : CCD et CMOS.

4.1.1 Capteur CCD (Charge Coupled Device)

Le capteur CCD (Figure 35), ou dispositif à transfert de charge, est le composant photosensible d'un appareil photo ou d'une caméra numérique. Inventé par W.S. Boyle et George Smith, de Bell Laboratories en 1969, ce composant, sous forme matricielle, transforme un faisceau de lumière en signal électrique. Chaque élément du capteur est une micro-cuvette d'une taille de quelques microns. Lors de l'exposition lumineuse, chaque cuvette se remplit d'électrons « fabriqués » lors de l'excitation des électrons du silicium par les photons de la source de lumière. Une fois le temps d'exposition écoulé, le CCD est vidé. Les techniques de vidage peuvent varier mais en général, les électrons sont transférés en fin de colonne, pour être transmis colonne par colonne vers le convertisseur analogique numérique. Malgré un tri très sévère des capteurs, plusieurs éléments du capteur, voir même des lignes entières, peuvent être défectueuses. Ces éléments défectueux sont dits aveugles. Plusieurs stratégies sont mises en place par les fabricants pour palier le défaut de ces éléments aveugles. Une solution simple consiste à dupliquer la valeur d'un pixel ou d'une ligne voisine. La solution généralement employée procède par interpolation. Sans aller jusqu'à être aveugle et bloquer la transmission, certains éléments sont trop sensibles. C'est-à-dire que le « courant de noir », correspondant à l'absence d'illumination, est trop important par rapport aux autres éléments du capteur. Ce défaut se traduit par l'apparition de points blancs sur les zones de l'image faiblement éclairées ou correspondant aux ombres. Là aussi, les stratégies diffèrent en fonction des fabricants. Certains considèrent ces éléments comme aveugles et leur appliquent la stratégie correspondante. D'autres vont leur affecter une correction dynamique. Ce processus, breveté par Thomson, soustrait la différence de « courant de noir » pour chaque pixel défectueux, par comparaison avec les pixels voisin. La difficulté est que l'intensité du « courant de noir » est liée à la température ambiante. Ce courant double lorsque la température s'élève de 8°C. C'est pourquoi, pour les caméras très sensibles ou certaines caméras thermiques, le capteur est maintenu à une température de 0° C (Leaf Camera Back) voire jusqu'à –50°C (scanner d'Ex-Machina). L'intérêt du refroidissement est donc de réduire les défauts du capteur mais aussi de le maintenir à un niveau constant. L'augmentation de la température n'influence pas seulement le « courant de noir », mais augmente également l'agitation des électrons. Cette agitation croissante rend les électrons plus « détachables ». Ils sont donc plus facilement entraînés par

les électrons contenus dans chaque puits et augmente donc le bruit du capteur. Le refroidissement du capteur permet également de réduire ce bruit. L'un des principaux avantages des CCD par rapport aux tubes qui équipaient les caméras vidéo et leur faible encombrement. Un autre avantage est l'absence de rémanence et de marquage lors d'une illumination trop forte ou d'une surexposition locale provoquée par les réflexions métalliques. Par contre, dans ce dernier cas un phénomène de « blooming » peut apparaître. Ce défaut provient du débordement des électrons d'un puit du capteur vers les puits voisins.

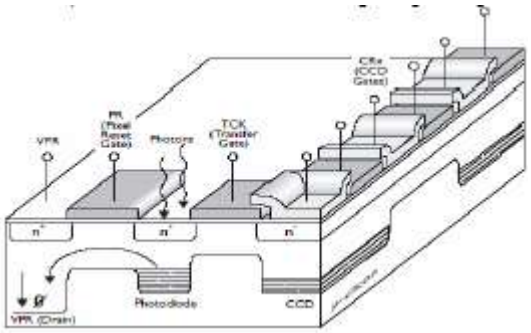


Figure 35 : structure d'un capteur CCD

4.1.2 Capteur CMOS (Complementary Metal Oxyde Semiconductor)

Depuis le milieu des années 1990, de nouveaux capteurs ont fait leur apparition. Ces capteurs se basent sur la technologie CMOS (Complementary Metal Oxyde Semiconductor). Si la technologie CMOS est connues depuis les années soixante, ce n'est qu'en 1993 [31] qu'elle a été utilisée par la NASA dans le domaine de l'imagerie numérique. Contrairement au capteur CCD, chaque puits contient un photo-détecteur et un transistor pour lire le signal. La technologie CMOS est plus délicate à mettre en œuvre que celle des capteurs CCD classiques, par contre sa dynamique est 250 fois plus grande sans saturation des pixels grâce à une compression logarithmique du signal arrivant sur chaque pixel, d'où l'intérêt des fabricants pour ce type de technologie. Autres avantages : vitesse d'acquisition plus grande, faible consommation électrique et encombrement encore plus réduit par rapport au CCD.

Le tableau suivant (Tableau 2) indique la taille en mm des principaux formats de capteurs CCD.

Format CCD	Hauteur (mm)	Largeur (mm)	Diagonale (mm)
1/4''	2.4	3.2	4
1/3''	3.6	4.8	6
1/2''	4.8	6.4	8
2/3''	6.6	8.8	11
1''	9.6	12.8	16

Tableau 2 : Taille des principaux capteurs CCD

4.2 Restitution des couleurs

Les éléments photosensibles du capteur CCD, tout comme ceux du capteur CMOS, ne sont pas sensibles à une longueur d'onde donnée mais à l'ensemble de la gamme visible ou infrarouge. Pour restituer les couleurs, deux stratégies sont principalement utilisées. Les caméras ou appareils photos numériques haut gamme vont être équipés de trois capteur CCD (ou CMOS) et d'un dispositif de prismes qui va décomposer le signal lumineux et l'orienter vers les différents capteurs en fonction de la longueur d'onde. Dans le cas des caméras classiques, un seul capteur est utilisé. Il est alors recouvert d'une matrice de filtre coloré. Ainsi, chaque cellule devient sensible à l'intensité d'une bande de fréquence donnée. Le filtre généralement utilisé est la mosaïque Bayer (Figure 36), du nom de son inventeur l'ingénieur Bryce Bayer de Eastman Kodak. Cette mosaïque est constituée d'une succession de ligne Vert/Bleu et de ligne Vert/Rouge.

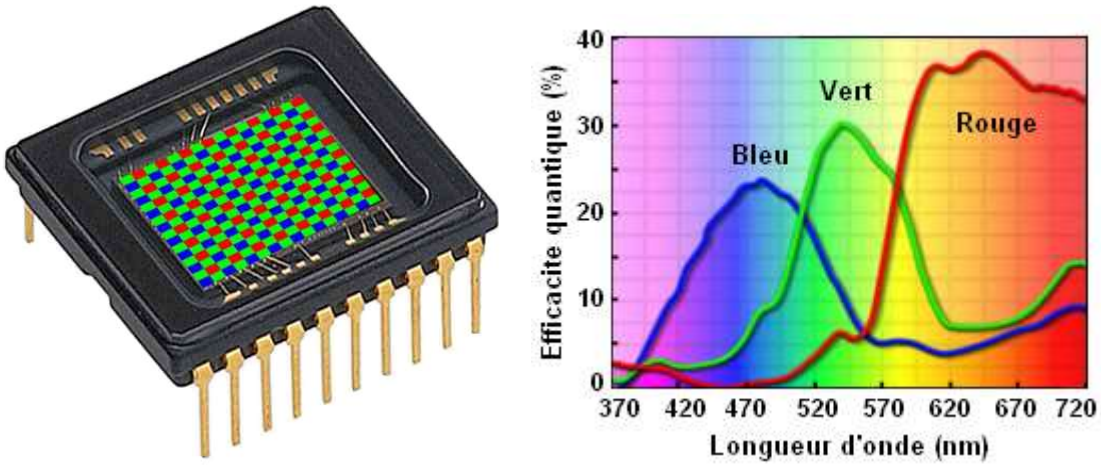


Figure 36 : matrice Bayer et sa sensibilité en fonction de la longueur d'onde

Le filtre est donc constitué de 50% de vert, 25% de bleu et 25 % de rouge. Cette répartition permet de respecter au mieux la sensibilité de l'œil humain. On obtient ainsi un codage classiquement appelé « 422RGB ».

Les données brutes issues du capteur sont ensuite traitées numériquement par différents procédés de façon à restituer une image selon une grille régulière. Ces traitements visent à, d'une part limiter les déficiences du capteur comme indiqué précédemment et, d'autre part, à améliorer le rendu de l'image en compensant les surexpositions ou les sous-expositions.

Les caméras « jour/nuit » que nous avons mentionnées dans le chapitre précédent utilisent un même capteur quel que soit le niveau de luminosité. Lorsque la luminosité est suffisante, l'image couleur est restituée suivant le procédé que nous venons de présenter. Lorsque la luminosité est plus faible, le niveau de gris de chaque pixel est simplement une somme pondérée des cellules correspondants aux trois composantes colorimétriques du pixel.

4.3 Pixel

L'image transmise par le capteur est donc généralement une suite de valeur numérique placée les unes à la suite des autres dans un espace mémoire et organisées ligne par ligne et colonne

par colonne selon une grille régulière. Le plus petit élément d'une image est un pixel qui décrit une ou plusieurs composantes colorimétriques (également appelé canal). Un pixel est une variable aléatoire discrète qui peut prendre 256 états. De façon générique, un pixel peut être défini comme suit :

$$p = (v_1, v_2, \dots, v_C)$$

où v_c est la $c^{ième}$ composante colorimétrique avec $c \in [1:C]; C \in \mathbb{N};\ v_k \in \mathbb{R}$.

Il existe plusieurs modes de représentation colorimétrique que nous ne détaillerons pas ici. Les plus utilisés étant la représentation en niveau de gris avec une seule composante colorimétrique, la représentation RGB (ou BGR suivant l'ordre des couleurs) avec 3 composantes colorimétriques, la représentation HSV qui peut avoir un intérêt si on souhaite s'affranchir des problèmes de luminosité en ne considérant que les composantes H et S ou encore la représentation YUV422 qui est le format natif des codeurs/décodeurs H264.

Il est a noté que la dynamique du capteur peut être très largement supérieure aux 256 niveaux de chaque composante colorimétrique de l'image.

4.4 Image

Une image est généralement considérée comme un plan orthonormé discret et fini de pixels. Le pavage, c'est-à-dire le partitionnement du plan euclidien en cellule élémentaire, est considéré comme régulier avec des cellules carrées qui ne se chevauchent pas. Cette représentation correspond au format des images tel qu'il est transmis pas le dispositif de capture. Une image est ainsi généralement représentée par une matrice à deux dimensions de pixels où chaque pixel est un carré (Figure 37). Nous attirons le lecteur sur le fait que cette représentation ne correspond pas à la réalité physique du système de capture. En effet sur le capteur, les pixels ne sont pas carrés et ne sont pas tous alignés en ligne et en colonne. Par exemple, sur la matrice Bayer il n'y a pas une répartition équitable des couleurs.

Une image I de dimension H * W, où est H le nombre de ligne et W le nombre de colonne avec H ∈ ℕ et W ∈ ℕ, est représentée comme :

$$I = \begin{pmatrix} p_{0,0} & \cdots & p_{0,W-1} \\ \vdots & & \vdots \\ p_{H-1,0} & \cdots & p_{H-1,W-1} \end{pmatrix}$$

Un pixel $p_{w,h,}$ avec $w \in [0:W-1]$ et $h \in [0:W-1]$, possède donc une adresse constituée d'un couple de coordonnées entières.

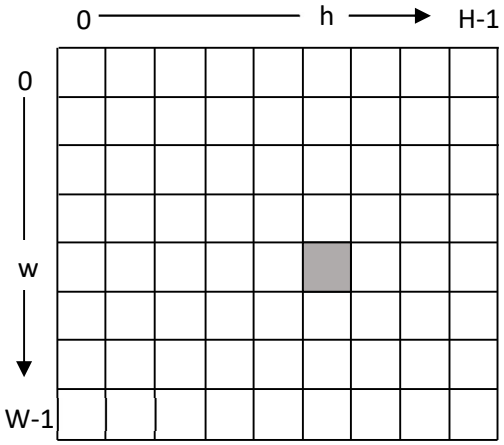


Figure 37 : organisation matricielle d'une image

Un pixel de coordonnée (w,h) peut être définie par 1 où plusieurs composantes colorimétriques. Dans le cas, ou plus d'une composante colorimétrique sont utilisées, le format de stockage peut varier suivant les outils utilisés ou même le format de représentation (cas YUV422 par exemple).

5 Acquisition

5.1 Progressive ou entrelacé

Le processus d'acquisition vise à figer une image dans le temps et à extraire la valeur de chaque pixel. Dans le cas idéal, nous aimerions que tous les pixels de l'image soient captés au même instant t mais ce n'est pas toujours le cas et cela dépend essentiellement de la technologie utilisée. Il en existe deux grandes classes : entrelacée et progressive. La première est ancienne et est liée aux premières heures de la télévision. Elle a été élaborée pour s'adapter au balayage ligne par ligne des tubes cathodiques et à la durée de maintien des luminophores. Afin d'éviter le scintillement de l'écran, les images sont transmises en deux parties comprenant les lignes paires puis les lignes impaires. Les premiers capteurs CCD ont donc été développés pour répondre à cette exigence. Cependant, comme les deux demi-images ne sont pas acquise en même temps, des « déchirures » peuvent apparaitre sur l'image reconstituée entre les lignes successives lorsque le mouvement de la caméra ou le mouvement d'un objet dans la scène est important. On appelle cela l'effet de peigne. Cette technologie était encore très présente au début des années 2000 et reste encore d'actualité sur les caméras analogiques. L'« œil » humain est capable de reconstituer l'image sans cet effet de peigne qui n'est alors visible que lorsque l'on fait un arrêt sur image. Cependant, cet effet reste bien présent et peut poser des difficultés lors de l'analyse automatique des images.

Avec le balayage progressif, chaque ligne de pixels est obtenue de façon séquentielle de façon à construire une seule image. C'est la technique utilisée sur les capteurs CMOS mais elle est également disponible sur certains capteurs CCD. L'intérêt évident de cette technologie est qu'il n'y a plus d'effet de peigne. Toutefois, il faut également tenir compte du type d'obturateur (shutter) qui peut être « global » ou « rolling ». Dans le cas du « rolling shutter », les colonnes sont exposées puis lues les unes après les autres ce qui peut entrainer un flou dans l'image. Dans

le cas du « global shutter », toutes les cellules du capteur sont exposées et obturées quasiment simultanément. Le capteur est ensuite « vidé » ce qui permet d'obtenir un véritable instantané.

5.2 Vidéo ou séquence d'images

Afin de nous placer au plus près des cas d'utilisations, nous introduisons le temps. La dimension temporelle est très importante dans le cas de la détection d'intrusion parce que l'utilisateur final ne résonne pas en termes d'indice d'image dans une vidéo mais en termes de date et d'heure et en temps de réaction du système.

Une vidéo est une séquence d'image, c'est-à-dire une succession d'images ordonnées temporellement. Une vidéo V de N images est définie comme suit :

$$V = \{I_t\}_{t=t_{start}}^{t_{stop}}$$

où I_{t_n} est la n[ième] image de la séquence prise à l'instant t_n avec $n \in [1:N]; N \in \mathbb{N};\ t_n < t_{n+1}$, t_{start}, la date du début de la séquence vidéo et t_{stop}, la date de la fin de la séquence et $t_{\text{start}} \leq t \leq t_{\text{stop}}$.

Cette première définition est générale et ne tient pas compte de la résolution des images ou de la stabilité de la fréquence d'acquisition. Dans certaines applications de vidéoprotection, il est possible d'enregistrer un flux vidéo en continu avec une fréquence d'acquisition et une résolution d'image faible et d'augmenter la fréquence et/ou la résolution lorsqu'il y a un intérêt (intervention d'un utilisateur, détection de mouvement par système externe ou analyse, *etc.*). Ce mode particulier d'enregistrement est généralisable à tous types d'application dès lors que les capacités de stockage sont limitées. Si dans la plupart des applications la résolution des images est identique dans l'ensemble de la séquence vidéo, le délai entre deux images successives est, en pratique, rarement constant sur une longue période. Plusieurs paramètres influent sur la fréquence de diffusion et de réception des images, comme les problèmes réseau (coupure, bande passante, qualité de la transmission) ou les corrections de gains notamment en ce qui concerne les caméras thermiques. Sauf mention contraire, considérons que la résolution des images est constante dans une séquence et nous utiliserons la définition suivante :

$$V = \{I_t \in \mathbb{R}^{H \times W \times C}\}_{t=t_{start}}^{t_{stop}}$$

Une vidéo peut ainsi être représentée par un tenseur à 4 dimensions de N images de résolution W x H et comportant C canaux.

6 Définitions

6.1 Eléments de géométrie discrète

La géométrie est très largement utilisée en traitement d'image pour détecter des lignes ou des courbes, extraire le contour des formes, calculer le périmètre d'une forme, sa convexité ou sa circularité. Toutes ces opérations de bas niveau réalisées sur l'image sont alors des points

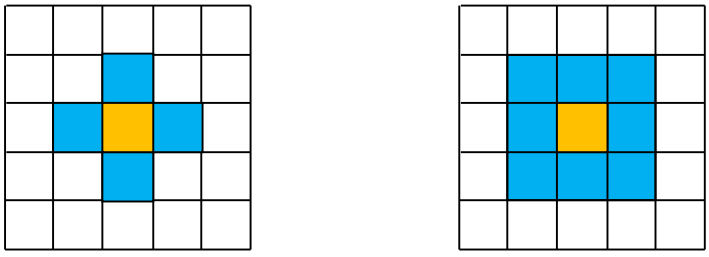
d'entrées pour extraire des informations de plus haut niveau. La nature discrète de l'image pose un certain nombre de problèmes théoriques intéressants du point de vue de la géométrie et peut-être source de paradoxe. Par exemple, deux droites discrètes peuvent se croiser sans avoir de pixel en commun. De nombreux travaux ont été réalisés afin d'étudier de façon théorique et algorithmique les objets constitués d'un ensemble de cellules. Ces travaux font l'objet d'un champ de recherche à part entière appelé « géométrie discrète ». Nous reprenons ici quelques définitions de base de la géométrie discrète emprunté dans [32] qui nous seront utiles par la suite pour des définitions de plus haut niveau.

Avant de présenter ces quelques définitions, nous rappelons que nous nous plaçons dans le cadre du pavage d'un plan euclidien organisé selon une grille régulière à partir de cellules carrées correspondantes aux pixels. Nous ne faisons pas d'hypothèse sur l'origine du repère, ainsi nous nous plaçons dans le cadre général où l'adresse d'un pixel est constitué d'un couple de coordonnée entière appartenant à $\mathbb{Z}^2$

Distance entre deux pixels : la distance entre deux pixels p et q de $\mathbb{Z}^2$ peut être définie comme la somme des différences absolues de chacune de leurs coordonnées prises une à une. De façon générale, on peut définir la distance d(p; q) entre deux pixels p et q de $\mathbb{Z}^2$ par :

$$d(p,q) = \sum_{i=1}^{2} |p_i - q_i|$$

r-Voisinage : deux pixels p et q de $\mathbb{Z}^2$ sont r-voisins si, prise deux à deux, au moins r de leurs coordonnées sont égales et les autres ne différent que de un. Un pixel à donc quatre 1-voisins (Figure 38-a) et huit 0-voisins (Figure 38-b).

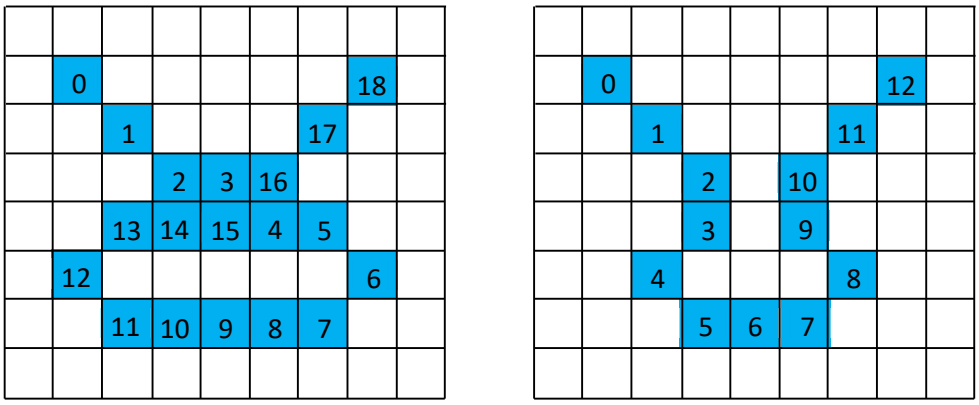


(a) *(b)*

Figure 38 : en bleu les quatre 1-voisins (a) et les 8 0-voisins (b) du pixel orange au centre

r-Chemin : un r-chemin est une séquence (p_0, …, p_{n-1}) de n pixels de $\mathbb{Z}^2$ tels que pour tout i = 0, …, n-2, p_i et p_{i+1} sont r-voisins. Un r-chemin est dit fermé si en plus, p_0 et p_{n-1} sont r-voisins (Figure 39-a).

r-Courbe: une r-courbe est une séquence (p_0, …, p_{n-1}) de n pixels de $\mathbb{Z}^2$ tels que pour tout i = 1, …, n-2, p_i à exactement deux r-voisins p_{i-1} et p_{i+1}. Une r-courbe est dite fermée si en plus, p_0 et p_{n-1} ont aussi exactement deux r-voisins, respectivement p_{n-1}, p_1 et p_{n-2}, p_0 (Figure 39).



(a) *(b)*

Figure 39 : un 0-chemin (a) et une 0-courbe (b)

Composante r-connexe : soit P un ensemble de pixels de $\mathbb{Z}^2$. P est une composante r-connexe si et seulement si pour tout couple de pixels p et q appartenant à P, il existe un r-chemin contenu dans P et reliant p à q.

r-bord : le bord d'une composante r-connexe P est l'ensemble des points de P dont le r-voisinage intersecte à la fois P et $\overline{\mathrm{P}}$.

6.2 Définitions des concepts de niveau intermédiaire

A partir des définitions de base de la géométrie discrète, nous pouvons à présent définir des concepts de plus haut niveau permettant de caractériser différents éléments de la scène. Pour cela, nous devons reformuler certain terme de la géométrie discrète. De même par soucis de généralité nous donnons les définitions relatives aux positions des pixel dans l'image dans le domaine des nombres réels.

Zone d'intérêt : La zone d'intérêt Z correspond à un sous ensemble de pixel de l'image.

$$Z \subset \mathbb{R}^{W \times H}$$

A noter que dans Z, les pixels ne sont pas forcément connexes. Communément, Z correspond à la (aux) zone(s) à protéger.

Catégorie : la catégorie Cat permet de définir les différents types d'objets qui peuvent être détectés par le système ou plus généralement présents dans la scène. Le terme catégorie doit être pris au sens large, c'est-à-dire que nous ne nous limitons pas à des classes de haut niveau comme : personne, véhicule, chat, *etc*. Une catégorie peut aussi être définie par une combinaison de critère géométrique et/ou colorimétrique. Par exemple, nous pouvons nous intéresser qu'à une catégorie d'objet ayant une certaine taille relative dans l'image et une vitesse de déplacement limitée. Dans le cadre de notre problématique, nous pouvons déjà définir deux sous-ensembles. La catégorie des objets autorisés Cat_a et la catégorie des objets non autorisés Cat_{na} avec :

$$Cat_a \cup Cat_{na} = Cat \text{ et } Cat_a \cap Cat_{na} = \emptyset$$

Dans la pratique, l'ensemble de toutes les catégories peut être difficile voire impossible à caractérisé. En fonction des applications, seulement l'une des deux sous catégories (autorisée ou non autorisée) est clairement définie. L'autre sous-catégorie est alors composée des objets ne correspondant pas à la première.

Blob : un blob b est un ensemble 0-connexes de n pixels $\{p_0, ... p_{n-1}\}$ de $\mathbb{R}^2$ ayant certaines propriétés communes (couleur, texture, mouvement, *etc.*)

Moment spatial d'un blob : les moments d'un blob sont des caractéristiques de premier ordre calculées à partir des positions des pixels appartenant au blob. Soit $b = \{p_0, ..., p_{n-1}\}$, un objet de n pixels (w_i, h_i) les coordonnées de chaque pixel et (u, v) des exposants entiers de puissance, le moment spatial m_{uv} s'écrit:

$$m_{uv} = \sum_{i=0}^{N-1} \sum_{j}^{N-1} w_i^u h_j^v$$

La somme u+v s'appelle « ordre du moment ».

A partir des différents moments, il est possible de calculer un certain nombre de caractéristiques géométriques comme la surface, les coordonnées du centre de masse, l'excentricité, la circularité ou les axes principaux obtenus en calculant les vecteurs propres de la matrice d'inertie des moments d'ordre 2.

Position d'un blob : La position d'un blob dans l'image est déterminée par les coordonnées d'un point représentatif du blob. Le plus souvent nous utilisons le centre de masse de l'objet $(\bar{w}, \bar{h})$ que nous pouvons évaluer à partir des moments :

$$\bar{w} = \frac{m_{10}}{m_{00}}, \bar{h} = \frac{m_{01}}{m_{00}}$$

Toutefois, dans certaine situation, il peut être judicieux de considérer le point de contact du blob avec le sol.

Objet d'intérêt : Dans une image, un objet d'intérêt est un blob ou un ensemble de blob qui appartient à la projection d'un objet de la scène. A noter que dans cette définition, nous n'imposons pas de connexité entre les blobs. Un objet est également défini par sa position $p \in \mathbb{R}^2$ dans l'image. A l'instar des blobs, la notion de position d'un objet peut être différente d'une application à l'autre et devra être correctement définie. Comme point de référence, nous pouvons utiliser le centre de masse calculé à partir des positions des pixels appartenant à chaque blob, un point particulier du contour ou de l'enveloppe convexe, *etc*. Nous avons volontairement étendue l'appartenance de la position à $\mathbb{R}^2$ et non à $\mathbb{R}^{W \times H}$, c'est-à-dire que nous autorisons un objet à être défini en dehors de l'espace image. Un objet est également défini par sa classe d'appartenance. Comme pour la position, et afin d'utiliser les bons algorithmes, il est indispensable de définir correctement ce qu'est un objet d'intérêt et donc l'ensemble des classes. Soit Nbo le nombre total d'objet d'intérêt dans une séquence d'image, un objet d'intérêt

$i \in [1 : Nbo]$ est donc défini dans les images horodatées d'une séquence par une classe d'appartenance, un ensemble de blob et un point de référence de la façon suivante :

$$O_i = \left(Cl_i, \{b_{i,t}\}, p_{i,t}\right)_{i \in [1:Nbo],\ \mathrm{t_{start}} \le t \le \mathrm{t_{stop}}}$$

Si l'objet i n'est pas visible à l'image t de la séquence alors $\{b_{i,t}\} = \emptyset$ et $p_{i,t}$ = NAN (indéfini)

Objet en mouvement : Nous considérons qu'un objet i est en mouvement à l'instant t si la norme du gradient de son point de référence est supérieure à zéro. C'est-à-dire :

$$\|grad(p_{i,t})\| > 0$$

Trajectoire d'un objet : La trajectoire d'un objet est l'ensemble des positions successives du point de référence (centre de masse, point en contact avec le sol, …) de cet objet au cours du temps.

Masque des objets d'intérêts : Pour une image n, le masque des objets d'intérêts ou MOI est une carte binaire de même résolution que l'image, c'est-à-dire de même taille et de même maillage $MOI \in \mathbb{R}^{W \times H}$. Chaque pixel de ce masque est une variable aléatoire discrète pouvant prendre deux états : 0 lorsque le pixel correspondant appartient au fond et 1 lorsque le pixel correspondant appartient à un objet.

$$MOI_n = \left\{\left(Cl_i, \{b_{i,n}\}\right), Cl_i \in Cl_a\right\}_{i=1}^{Nbo}$$

Suivi : Soit E^i_n le vecteur de caractéristiques associé à un objet i à l'image n et $\Gamma = \{ E^i_1, \dots E^i_N\}$ l'ensemble des états de l'objet au cours d'une séquence temporelle de N frames. A noter que E_n peut être égale à $\{\emptyset\}$ si l'objet n'est pas présent ou n'est pas visible dans l'image n. L'objectif du suivi visuel des objets dans une séquence d'image est de mettre à jour le vecteur d'états Γ en fonction des observations collectées en tenant compte des observations passées et à venir lorsque l'application visée le permet.

6.3 Concept de haut niveau

Les concepts de haut niveau sont plus difficiles à formaliser et dépendent fortement de l'application. Bien que cela ne devrait pas être le cas, la formulation de ces concepts est également contrainte par les limites des techniques pouvant être appliquées pour y répondre.

Nous donnons ici une définition générale de la détection d'intrusion.

- Soit $I_t \in \mathbb{R}^{H \times W \times C}$ une image horodatée de résolution H x W et de C canaux
- Soit $\mathcal{V} = \{I_t \in \mathbb{R}^{H \times W \times C}\}_{t \in T}$ une séquence vidéo d'images horodatées de résolution H x W et de C canaux définie sur un intervalle de temps $T = \left[\mathrm{t_{start}}, \mathrm{t_{stop}}\right]$ avec t_{start}, la date du début de la séquence et t_{stop}, la date de la fin de la séquence et $\mathrm{t_{start}} \le t \le \mathrm{t_{stop}}$.

- Soit $Z \subset \mathbb{R}^{H \times W}$, un sous ensemble de pixel des images de la séquence correspondant à la (aux) zone(s) à protéger.
- Soit $\mathcal{T}$ un ensemble d'intervalles de temps durant lesquels une zone doit être protégée.
$$\mathcal{T} = \{[t_b, t_e], t_{start} \leq t_b, t_b \leq t_e, t_e \leq t_{stop}\}$$
- Soit $\mathrm{Cl}_{na} \in \mathrm{Cl}$, l'ensemble des classes d'objets non autorisées pendant les intervalles de temps précédemment définis. Nous considérons ici que les classes non autorisées sont les mêmes dans tous les intervalles de temps où la zone est à protéger afin de ne pas alourdir la définition.
- Soit $O_i = \left(Cl_i, \{b_{i,t}\}, p_{i,t}\right)_{i \in [1:Nbo],\ \mathrm{t_{start}} \leq t \leq \mathrm{t_{stop}}}$, un objet défini comme une classe, un ensemble de blobs et ses coordonnées dans l'image où Nbo est le nombre total d'objets dans la séquence.

Evénement d'intrusion : nous définissons un événement d'intrusion IE(t) comme étant le déplacement d'au moins un objet non autorisé dans une zone définie sur l'image pendant une fenêtre temporelle donnée, ce que nous pouvons traduire ainsi :

$$\mathrm{IE(t)} = \begin{cases} 1 & si\ t \in T\ \wedge \exists\ o_i, cl_i \in cl_{na} \wedge p_{i,t} \in Z \wedge \left\| grad(p_{i,t}) \right\| > 0 \\ 0 & sinon \end{cases}$$

Séquence d'intrusion : une séquence d'intrusion est définie comme l'intervalle de temps maximum encadrant une succession continue d'événements d'intrusions.

$$IS = [t_{bs}, t_{es}], \{IE(t) = 1\}_{t=t_{bs}}^{t_{es}} \wedge IE(t_{bs-1}) = 0\ \wedge IE(t_{es+1}) = 0$$

Avec $t_{bs} \in \mathcal{T}, t_{bs} > t_{start}$ et $t_{es} \in \mathcal{T}, t_{es} < t_{stop}$, où $\mathrm{t_{start}}$ et $\mathrm{t_{stop}}$ sont respectivement les dates de début et de fin de la vidéo. Une vidéo contient un ensemble $\mathcal{S}$ de séquences d'intrusion.

7 Conclusion

Ce chapitre est une synthèse des outils que nous allons utiliser dans la suite du document pour détecter les blobs en mouvement dans l'image, les regrouper, si nécessaire, pour représenter les objets en mouvement dans la scène, les suivre au cours du temps et déterminer leur trajectoire. Ces informations de niveau intermédiaires ainsi obtenues vont ensuite être analysées afin de nous permettre de calculer des comportements de plus haut niveau. La première étape de ce travail consiste à détecter les objets d'intérêts dans une séquence vidéo.

IV - Détection d'objets en mouvement

1 Introduction

La détection d'objets en mouvement dans une vidéo est une des premières étapes d'un algorithme dans bon nombre d'applications. Elle est particulièrement bien adaptée pour segmenter les objets d'intérêt lorsque ceux-ci sont en mouvement relatif par rapport au fond de la scène. Cette détection repose soit sur la détection du mouvement soit sur la segmentation basé mouvement. La distinction est un peu subtile mais la détection vise à décider quels pixels ou groupes de pixels appartiennent à des objets en mouvement alors que la segmentation basé mouvement s'applique à partitionner l'image en régions comportant des caractéristiques de déplacement communes. La détection produit une carte binaire indiquant la présence ou l'absence de mouvement alors que la segmentation produit une carte multi-labels.

Dans cette section, nous allons présenter les trois grandes familles d'approches, que l'on retrouve communément dans la littérature et qui permettent d'extraire les d'objets en mouvement dans une scène fixe :

- Détection par flot optique : le flot optique caractérise le déplacement apparent de l'intensité d'un pixel d'une image causé par le mouvement relatif des objets par rapport à la scène.
- Détection par appariement de blocs : cette approche est basée sur la mise en correspondance de régions d'une image sur une autre.
- Détection par modèle de fond : le principe de cette approche est de construire et, dans la plupart des cas, de mettre à jour un modèle caractérisant les pixels appartenant à la scène dénuée d'objet d'intérêt.

Dans cette section, nous aborderons également le cas des détecteurs spécialisés. Cette classe d'algorithme n'utilise pas la notion de mouvement ou de vidéo et, ne garde généralement aucune mémoire des détections précédentes. Cependant, nous l'intégrons dans ce chapitre puisqu'avec l'essors de l'apprentissage profond, cette approche permet efficacement de détecter des classes d'objets particulières et notamment les piétons. En effet, lorsque les objets d'intérêts sont connus *a priori*, il est possible d'utiliser un détecteur construit ou entraîné spécifiquement pour cette tâche. Dans ce genre d'approche, l'image est généralement parcourue par une fenêtre glissante dans laquelle le détecteur est appliqué. La présence ou l'absence de l'objet est fonction de la réponse du détecteur. Des travaux récents permettent désormais d'analyser l'ensemble de l'image en une seule itération et d'extraire automatiquement un ensemble de boîtes englobantes.

2 Détection de mouvement

Nous présentons tout d'abord les approches basées sur le flot optique et sur l'appariement de bloc. Ce ne sont pas les méthodes les plus utilisées dans le cas de la vidéo protection mais elles présentent néanmoins quelques intérêts. Nous aborderons ensuite plus en détail l'approche basée sur la modélisation du fond ou de l'arrière-plan.

2.1 Flot optique

L'analyse du flot optique est à classer dans la catégorie des approches permettant de faire la segmentation basée mouvement. Comme indiqué dans [33], le mouvement est un puissant signal de perception permettant de segmenter des objets dans une vidéo. Le terme de flot optique a été inventé par le psychologue James J. Gibson dans une étude de la vision humaine. Le calcul du flot optique consiste à extraire un champ de vitesses dense à partir d'une séquence d'images. Au début des années 80, Horn et Schunk [34] ont proposé une première estimation du flot optique. Leurs travaux ont été suivis d'un grand nombre de publications. Parmi les différentes approches possibles, les deux plus populaires sont :

- L'approche différentielle
- L'approche fréquentielle,

La plupart des solutions, quelle que soit leur approche, peuvent être traitées de façon hiérarchique afin d'améliorer la détection et permettre la recherche de déplacements plus grand que la taille du noyau.

Les méthodes différentielles permettant de déterminer le flot optique font l'hypothèse que l'intensité d'un point de l'image en mouvement est constante le long de la trajectoire. Si p(r, c, t) représente l'intensité lumineuse d'un point au temps t, alors p(r(t), c(t), t) = constante. Cela se traduit par l'égalité suivante :

$$p(r, c, t) = p(r + dr, c + dc, t + dt)$$

De la résolution de cette équation, on obtient l'expression de la contrainte du flot optique (CFO) :

$$\frac{\partial p}{\partial r}\frac{dr}{dt} + \frac{\partial p}{\partial c}\frac{dc}{dt} + \frac{\partial p}{\partial t} = 0$$

où

- $\left(\frac{\partial p}{\partial c}, \frac{\partial p}{\partial r}\right)$ représente le gradient spatial ∇u
- $\left(\frac{dr}{dt}, \frac{dc}{dt}\right)$ représente le champ vitesse que l'on cherche à déterminer
- $\frac{\partial p}{\partial t}$ représente le gradient temporel

L'approche de Lucas-Kanade [35] est avec celle de Horn-Schunck [34] l'une des méthodes les plus classiques de détermination du flot optique, même si, à l'origine, cette méthode était utilisée pour le recalage d'image. Cette méthode fait l'hypothèse supplémentaire que le flot est

localement constant sur un voisinage et cherche à résoudre par la méthode des moindres carrés la contrainte de flot optique dans un voisinage W(p). Ce voisinage pondéré par une Gaussienne peut être considéré comme un processus de diffusion linéaire. Plusieurs auteurs ont eu l'idée d'utiliser un processus non linéaire pour obtenir les coefficients du tenseur lissé. Spies et *al.* [36] rehaussent les structures temporelles en utilisant les valeurs propres issues du tenseur de structure. Brox et *al.* [37] proposent une méthode différente basée sur un tenseur de diffusion et montrent que les tenseurs non linéaires anisotropes donnent des localisations plus précises.

Classiquement, le flot optique est utilisé pour extraire des trajectoires ponctuelles [20] sur une longue période qui sont ensuite regroupées. Pour Ochs et *al.* [38], cette analyse sur une longue période leur permet de d'obtenir un regroupement plus cohérent dans le temps. Chen et *al.* [39] proposent un graphe qui utilise les informations de mouvement locales et globales codées par les trajectoires de points denses suivis.

Les approches fréquentielles d'estimation du flot entre deux images sont basées sur l'équivalence translation / déphasage de la transformée de Fourier.

$$I(r,c) \xrightarrow{TF} F(u,v)$$

$$I(r+dr, c+dc) \xrightarrow{TF} F(u,v)e^{2i\pi(udx+vd\ )/wh}$$

Il suffit donc de considérer ce déphasage pour deux couples (u,v) pour calculer (dr, dc).

Cette technique est relativement rapide grâce au calcul de la FFT (Fast Fourier Transform). Elle est robuste car toutes les fréquences contribuent au calcul du flot. Par contre, elle est sensible au bruit et aux changements d'illumination et est, en général, utilisée pour calculer le mouvement global de l'image. Plusieurs exemples d'approches fréquentielles sont décrits dans les travaux de Heeger [40], Spinei [41] et Torralba [42].

Comme la plupart des problèmes en vision par ordinateur, le flot optique peut également être estimé à partir du réseau de neurones convolutionnels (CNN) entrainé spécifiquement pour réaliser cette tâche. Dosovitskiy et *al.* [43] ont dès 2015 proposés un réseau appelé FlowNet.

2.2 Appariement de blocs

Les techniques basées sur l'appariement de blocs ou *block matching* recherchent une maximisation de la mesure de corrélation entre deux blocs d'une même image ou de deux images différentes. Contrairement aux approches différentielles présentées plus haut, ces techniques ne sont pas basées sur l'hypothèse de l'illumination constante et sont plus simples à mettre en œuvre. Elles sont notamment utilisées dans les algorithmes de compression d'image ou de vidéo.

Wu et *al.* [44] ont présenté une méthode pour extraire les objets en mouvement directement à partir du flux H.264 compressé. Le codage H.264 exploite la redondance temporelle des images pour arriver à une bonne compression en codant uniquement la différence entre 2 images

successives ou bien la différence entre la prédiction et l'image de référence. Les auteurs effectuent une classification des vecteurs de mouvement dans les catégories de contour, fond, forme ou bruit. Ils obtiennent alors un champ de vecteurs de mouvement soumis au modèle MRF (Markov Random Field), déterminant quel sous blocs 4X4 sont en mouvement. Leur approche produit de bons résultats mais elle utilise de nombreux paramètres pour les seuils de la classification et les coefficients de pondération du MRF nécessitant des réglages fins selon les séquences vidéo.

Solana-Cipres et *al.* [45] ont proposé l'utilisation de logique floue pour segmenter les régions en mouvement. Cette approche requière peu d'information et est basée sur deux caractéristiques de la norme H.264 qui sont les modes de décision et les vecteurs de mouvement. La logique floue est utilisée pour décrire la position, la vitesse et la taille des régions en mouvement. L'algorithme est efficace en temps réel car peu de données sont nécessaires et il montre des résultats encourageants dans diverses situations.

Une approche originale développée par Poppe et *al.* [46] se différencie des autres travaux en analysant le champ de vecteurs de mouvement pour trouver les objets en mouvements. Cette approche se base sur le fait que le format H.264 compresse plus les parties de l'arrière-plan que celles de l'avant-plan contenants des objets en mouvement. La détection est alors basée sur la taille (en bits) des macro-blocs codant l'image. Les auteurs construisent le modèle de fond MB_{model} en sauvegardant la taille maximale de chaque macro-bloc pendant la phase d'apprentissage. Pendant la phase de détection, un macro-bloc est considéré comme « avant plan » si sa taille s remplit le critère suivant :

$$s > MB_{model} + T_{mb}$$

Où T_{mb} est un seuil fixé pour un MB particulier. Le système permet des vitesses d'exécution très importantes, jusqu'à 20 fois plus rapides que les autres approches de détection.

Dans [21], nous avons proposé une amélioration du travail de Poppe et *al.* Dans notre approche, nous avons intégré un modèle adaptatif de mélanges de Gaussiennes permettant de caractériser l'évolution de la taille des macro-blocs au cours du temps. Le modèle proposé par Poppe et *al.* a plusieurs limitations. Il fait notamment l'hypothèse que le fond de la scène est stationnaire ; c'est-à-dire qu'aucune mise à jour du modèle n'est faite pendant la phase de détection. Même si ce choix peut être pratique pour plusieurs applications de vidéosurveillance, un modèle d'apprentissage du fond adaptatif est généralement une meilleure approximation. En effet, ce modèle basé sur la taille des macro-blocs devrait évoluer pendant le temps de traitement de la séquence vidéo complète.

Par conséquent, dans notre approche, nous représentons la distribution des tailles d'un macro-bloc donné par un modèle de mélange de Gaussiennes. Nous avons par la suite intégré d'autres modèles de fond adaptatifs de la littérature basés sur des représentations d'intensités multimodales telles que le codebook [47] et les distributions discrètes [48]. Dans le cas des mélanges de Gaussienne, une fois la taille s (en bits) d'un macro-bloc MB_t extraite du flux H.264 à un instant t, nous évaluons si s appartient au modèle Gaussien en considérant chaque distribution G_i (avec la loi normale $N(\mu_i, \sigma_i)$) :

$$s - \mu_i \leq k.\sigma_i$$

Dans cette équation, k est une constante qui est généralement égale à 2 ou 3. Si s n'appartient pas à une des distributions Gaussiennes du modèle, alors le macro-bloc associé est considéré comme appartenant à l'avant plan.

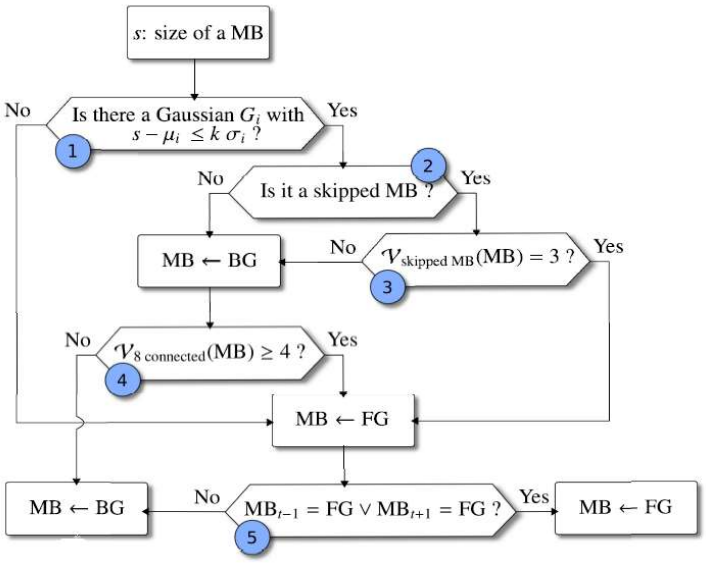


Figure 40 : schéma général de notre approche de détection d'objets dans un flux vidéo H264

Si l'équation précédente est respectée pour une Gaussienne G_i, alors nous vérifions si le macro-bloc est un « skipped MB » ou pas. Ces macro-blocs particuliers, définis dans la norme H.264/AVC [49], représentent un macro-bloc pour lequel aucune donnée résiduelle n'a été produite lors de l'encodage. Dans ce cas, nous devons considérer un ensemble de macro-blocs du voisinage que nous appellerons $v_{MB\ skipped}$. En effet, ces macro-blocs sont utilisés pendant le décodage pour reconstruire un « skipped MB ». Dans ce test, nous supposons, comme dans [46], que si les 3 macro-blocs connexes sont classifiés comme appartenant à l'avant plan, alors le « skipped MB » l'est aussi, sinon nous le classifions comme appartenant à l'arrière-plan. Dans ce dernier cas, nous appliquons un filtre spatial qui part du principe que, si au moins 4 macro-blocs appartiennent au premier plan parmi les 8 voisins connectés, alors le macro-bloc traité sera aussi classifié comme appartenant à l'avant plan. La Figure 40 présente le schéma général de notre approche de détection d'objets dans un flux vidéo H264.

Nous avons obtenu des résultats encourageant que le lecteur pourra retrouver dans [21]. Toutefois, cette piste n'a pas été retenue par la suite. L'une des raisons est que les blocs ne permettent pas une segmentation fine des objets. La méthode permet de détecter du mouvement de façon correcte et d'évaluer des trajectoires lorsque les objets en mouvement dans la scène ne s'occultent pas les uns les autres. Par contre, la segmentation est trop grossière pour classer ou identifier ces objets. Une application que nous avions également envisagée était d'utiliser cette approche pour réaliser une indexation rapide des vidéos en récupérant la boite englobante des objets en mouvement ainsi qu'un histogramme de couleur. L'idée étant de faire ensuite des requêtes sur la couleur des objets et sur leur position dans l'image. Notre approche

permet effectivement d'extraire les boites englobantes. Cependant, l'histogramme de couleur est plus difficile à obtenir puisqu'il faut pour cela décompresser tous les blocs qui interviennent dans la construction des blocs de l'objet en mouvement et cela jusqu'à la dernière I-frame reçue.

Ces techniques d'estimation du flot optique sont intéressantes mais s'avèrent délicates à manipuler pour extraire de l'image une personne qui marche par exemple. En effet le balancement des bras ne suit pas forcement le sens de la marche. Une approche différente consiste à estimer, par une segmentation, le mouvement global d'une ou de plusieurs régions de l'image.

2.3 Modélisation du fond

Cette dernière catégorie d'algorithmes est la plus populaire en vidéo protection tout simplement parce qu'elle est la plus efficace. En termes d'efficacité nous considérons le rapport entre le résultat obtenu et le temps de calcul.

Le principe général de ces approches est de caractériser, dans une fenêtre temporelle, l'évolution ou plus généralement la répartition des composantes colorimétrique de chaque pixel de l'image. L'idée étant ainsi de pouvoir délimiter l'espace ou les espaces colorimétriques correspondant à la scène dénuée d'objet d'intérêt. Ces espaces colorimétriques correspondent à ce qui est communément appelé le fond ou l'arrière-plan. Si pour un pixel donné, la valeur de ses composantes colorimétriques à l'instant t n'appartient à aucun des sous espaces de la scène, c'est-à-dire à l'arrière-plan, il est alors considéré comme faisant partie de l'avant-plan.

Comme indiqué dans [50], une approche courante de la modélisation de l'arrière-plan consiste à supposer que les échantillons de l'arrière-plan qui le composent sont générés par une variable aléatoire et correspondent donc à une fonction de densité de probabilité donnée. Il suffit alors d'estimer les paramètres de la fonction de densité pour déterminer si un nouvel échantillon appartient à la même distribution.

Un état de l'art très complet des méthodes de modélisation de fond peut être trouvé dans les travaux de Thierry Bouwmans[5] et en particulier dans [51]. Dans ce manuscrit, nous allons simplement présenter les approches les plus emblématiques en vidéo protection et surtout conformes à la contrainte de l'exécution en temps réel. Ces méthodes seront reprises par la suite pour illustrer les difficultés liées à l'évaluation de ces méthodes et pour proposer des approches génériques d'amélioration.

GMM : La première approche que nous présentons est appelée mélanges de gaussienne ou GMM pour « Gaussian Mixture Model ». Il est difficile, voire même inconvenant, de ne pas présenter cette approche proposée par Stauffer et Grimson [52] étant donné le nombre

[5] https://sites.google.com/site/backgroundsubtraction/overview

d'articles scientifiques qui y font référence. L'idée générale de cette approche et d'approximer la variation de chaque pixel par une ou plusieurs distributions gaussiennes représentées par une moyenne, un écart type et un poids. Ce choix de distribution permet une représentation multimodale prenant en compte plusieurs couleurs pour chaque pixel (en général entre 3 et 5). Elle gère donc les cas de mouvements périodiques dans le fond (mouvement des branches d'un arbre à cause du vent par exemple). Cette méthode a été constamment améliorée notamment par Dar-Shyang [53] qui propose une solution permettant d'améliorer la stabilité puis par Kaewtrakulpong et Bowden [54] qui ont intégré une procédure de détection des ombres et amélioré la vitesse d'apprentissage du modèle. Une des principales difficultés des GMM consiste à décider quelles distributions du modèle appartiennent à l'arrière-plan. Aussi, Fradi et Dugelay [55] proposent l'incorporation d'un modèle de mouvement uniforme. Si les GMM restent encore populaires aujourd'hui c'est que cette méthode probabiliste permet d'obtenir des performances satisfaisantes grâce à sa capacité de modéliser assez simplement des arrières plans complexes.

CB : En 2005, Kim et al [47] proposent une méthode originale appelée « Codebook ». Le principe de cette méthode est d'associer un dictionnaire de mots de code à chaque pixel de l'image. Un mot de code est composé d'un vecteur v = (R; G; B) ou chaque composante colorimétrique correspond à la moyenne des composantes colorimétrique de chaque pixel attribué à ce mot et d'un ensemble de 6 valeurs

$$aux = \{I_m, I_M, f, \lambda, p, q\}$$

où I_m et I_M sont la luminosité minimale et maximale de tous les pixels attribués à ce mot de code, f est la fréquence à laquelle le mot de code s'est produit, λ est la longueur maximale d'un cycle négatif définie comme étant l'intervalle le plus long de la période d'apprentissage pendant lequel le mot de code n'a pas été répété ; p et q sont respectivement le premier et le dernier accès au mot de code. Les auteurs proposent également un modèle de couleur permettant de séparer les composantes couleurs et la luminosité (Figure 41). L'intérêt de ce modèle est d'être moins sensible aux ombres portées et aux changements d'éclairage de la scène.

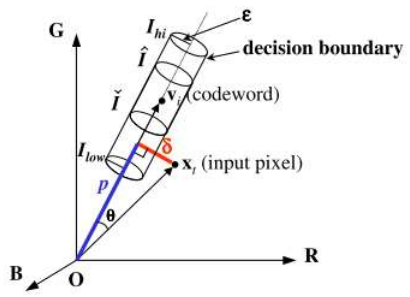


Figure 41 : représentation du modèle de couleur proposé dans l'algorithme du Code Book [47]

Une condition, nécessaire mais pas suffisante, d'appartenance d'un pixel à l'un des codes du dictionnaire est que sa luminosité I soit comprise entre les valeurs limites I_{low} et I_{high}. Ces valeurs sont calculées à partir des valeurs I_m et I_M ainsi que deux facteur α et β :

$$I_{low} = \alpha I_M \text{ et } I_{high} = \min(\beta I_M, \frac{I_m}{\alpha})$$

Cette approche proportionnelle des limites haute et basse n'est pas optimale puisqu'elle a tendance à rendre l'algorithme plus sensible, comme indiqué dans [56] lorsque les valeurs colorimétriques sont faibles et *a contrario* moins sensible lorsque les valeurs colorimétrique sont plus forte. En d'autres termes, l'algorithme détecte mieux les objets sombres que les objets clairs. Or, les auteurs n'apportent aucune justification sur la dynamique qui serait plus ou moins forte en fonction de la luminosité.

L'algorithme original a également été amélioré au cours du temps. Dans [57] les auteurs proposent notamment une stratégie de mise à jour du modèle, ce qui n'était pas le cas dans [47]. Dans [56], les auteurs proposent de remplacer le modèle colorimétrique RGB par le modèle HSV alors que dans [58] les auteurs lui préfèrent le modèle YUV. L'article de Doshi et *al.* [56] apportent une contribution supplémentaire en replaçant le cylindre de l'algorithme original par un volume hybride composé d'un cône à la base et d'un cylindre. Wu et *al.* [59] intègrent une dépendance spatiale entre les pixels voisins ainsi qu'une dépendance temporelle avec un modèle de Markov. Enfin, plusieurs auteurs utilisent des solutions hybrides où l'algorithme de CB est associé à d'autres méthodes. Par exemple [60] utilisent un noyau gaussien pour éviter la quantification inhérente à l'algorithme d'origine ; Wu et al [61] proposent une solution multicouche avec une première couche modélisée par des motifs binaire locaux (LBP) et une couche de raffinement utilisant l'algorithme CB ; Mousse et *al.* [62] associent le CB avec un détecteur de gradients. Après avoir appliqué l'algorithme de CB, les auteurs recherchent l'enveloppe convexe de chaque contour sur l'image originale. Un pixel d'avant plan détecté par l'algorithme de CB est effectivement considéré comme tel s'il est à l'intérieur de l'enveloppe convexe d'un contour.

Une variante du CB est proposée par Saint Charles et al [63] avec l'algorithme PAWCS. Dans leur solution, les auteurs utilisent également le concept de dictionnaire de mots mais en modifiant la nature des mots. Dans leur cas, un mot est constitué d'un vecteur couleur, de trois entiers permettant de garder la fréquence, ainsi que la date du premier et du dernier accès comme pour le CB. Toutefois, ils ajoutent deux octets pour mémoriser une information de texture. Les auteurs utilisent le descripteur LBSP (« Local Binary Similarity Pattern ») proposé dans [64]. Contrairement au CB classique, il n'y a pas de regroupement des valeurs d'un pixel au sein d'un même mot de code. Dans l'algorithme CB, le mot qui correspond le mieux à la valeur du pixel courant est mis à jour en modifiant sa couleur moyenne et éventuellement les valeurs I_m et I_M. Dans le cas de l'algorithme PAWCS, la caractéristique de texture ne peut pas être moyennée aussi facilement que peut l'être la couleur. La stratégie des auteurs est donc de conserver des mots couvrant un espace de représentation plus faible avec une plus faible persistance au lieu de conserver des mots plus généraux avec une persistance plus forte. Par ailleurs, les auteurs proposent également un dictionnaire global qui permet de mémoriser des mots correspondant à l'avant plan ainsi qu'une boucle de rétroaction permettant d'ajuster les seuils de façon automatique. Dans [65], les auteurs indiquent que l'algorithme PAWCS est l'une des approches les plus abouties dans la catégorie des modélisations du fond. A noter toutefois que dans PAWCS les auteurs intègrent une étape de post traitement sur le masque de mouvement avec, entre-

autre, une fonction de lissage, des opérations de morphologie mathématique et même un remplissage des trous à l'intérieur des blobs. Ces mêmes post-traitements appliqués au masque de mouvement des autres algorithmes permettent également d'améliorer significativement leurs performances.

VM : L'algorithme du VuMètre proposé par Goyat [48] n'est pas le plus connu ni le plus utilisé des algorithmes de modélisation du fond. Toutefois, il est très facile à mettre en œuvre et offre des performances honorables comme nous le verrons dans la section « évaluation ». C'est un modèle non paramétrique basé sur une modélisation discrète de la densité de probabilité de chaque pixel. Au lieu d'utiliser des noyaux gaussien comme dans [66], les auteurs utilisent une fonction de Kronecker. La mise à jour proposée par les auteurs permet de maintenir la normalisation de l'histogramme et ainsi d'éviter les dérives. Par ailleurs un poids défini par l'utilisateur permet simplement d'adapter la vitesse de mise à jour du modèle en fonction du contexte.

ViBe : L'algorithme ViBe a été proposé par Barnish et *al.* [67]. Comme indiqué dans [65], c'est un modèle particulièrement efficace et assez simple d'implémentation. Dans un sens, on peut le considérer comme une méthode basée sur un dictionnaire à l'instar du CB, que nous avons présenté précédemment, mais avec une caractérisation des mots réduite à sa plus simple expression. En effet, chaque pixel du modèle de fond est défini par un ensemble de N échantillons (typiquement N = 20). Ce qui rend la méthode ViBe originale est que la politique de sélection des échantillons du modèle de fond découle d'un processus aléatoire. Les auteurs proposent un mécanisme de propagation spatiale, basé également sur un processus aléatoire, qui permet d'intégrer dans le modèle de fond un objet d'avant plan qui reste statique. Le processus de décision permettant de déterminer si un pixel d'une nouvelle image appartient à l'arrière-plan est simple. Il suffit de trouver au moins n échantillons suffisamment proches, selon la distance euclidienne, pour classer le pixel dans l'arrière-plan. Typiquement n = 2. Il n'y a pas de notion temporelle dans le modèle proposé par Barnish. C'est simplement la fréquence d'apparition associé au tirage aléatoire qui permet une représentation plus ou moins importante de certaine valeur et qui en détermine leur durée de vie.

Cet algorithme a été rapidement modifié par Van Droogenbroeck et *al.* [50] qui ont proposé l'algorithme ViBe+. Les auteurs ont proposé comme modification mineure de réduire le facteur de mise à jour mais surtout de le rendre adaptatif de façon à prendre en compte plus rapidement un changement global brusque. Cependant leur contribution majeure consiste à proposer un algorithme en deux passes. La première passe permet de calculer le masque de mouvement et la deuxième s'appuie sur ce masque pour mettre à jour le modèle. Toutefois entre la première et la deuxième passe, le masque de mouvement subit un certain nombre de traitements classiques (filtrage, opération de morphologie mathématique) permettant de combler les trous, de fusionner les régions connexes ou de supprimer les blobs de petite taille. Le masque ainsi traité est alors utilisé pour conditionner la mise à jour des pixels de l'arrière-plan. D'après les auteurs, ceci permet d'améliorer la cohérence de la mise à jour sur les composantes connexes mais également de limiter la propagation spatiale. Ce dernier point est particulièrement intéressant lorsque l'utilisateur souhaite prolonger la détection des objets statiques de l'avant

plan. Enfin comme autre contribution, les auteurs utilisent la mesure *distcolor*, proposée dans CB [47] à la place de la distance euclidienne pour déterminer le nombre d'échantillons.

Dans [68], les auteurs proposent l'algorithme SubSens basé sur le même principe de sélection aléatoire des échantillons. La différence majeure entre les deux algorithmes est que dans SubSens, chaque échantillons est caractérisé par sa couleur et un descripteur de texture LBSP [64].

3 Les défis de la modélisation de l'arrière-plan

La modélisation de l'arrière-plan est une technique efficace pour extraire les objets en mouvement dans la scène. Si, de plus, le fond est fixe ou considéré comme tel, l'extraction des objets en mouvement est facilitée. La vidéo protection telle que nous l'avons définie se place exactement dans ce cadre-là. Pour rappel, les caméras sont fixes et, à part quelques cas particuliers, le fond de la scène à sécuriser est fixe. Il s'agit essentiellement de zones de parking, de chemins ou d'étendues d'herbe ou de végétaux dans lesquels il ne se passe rien ou presque. Quelques animaux peuvent se déplacer dans la zone à sécuriser mais généralement ils doivent être ignorés par le système. Le système doit par contre être opérationnel pendant de très longue période de plusieurs heures voire plusieurs jours, de jour comme de nuit, et quelques soient les conditions météorologiques. La détection de l'intrusion, quant à elle, doit se faire rapidement et sur quelques images seulement. Les algorithmes que nous avons présentés ont été spécialement étudiés pour modéliser ce type de scénario, mais il nous est arrivé de les utiliser dans d'autres cas comme par exemple la détection de péniche sur un canal ou la détection d'intrusions sur un front de mer.

Même dans les cas les plus simples, la notion de fond fixe est sujette à discussion. La scène en elle-même peut être fixe mais avec des parties plus ou moins dynamique comme des branches d'arbres mues par le vent, les reflets à la surface de l'eau, ou encore de la pluie ou de la neige même si cela est moins fréquent. Par ailleurs, le terme « fixe » n'est pas forcément le plus adapté pour parler du fond de la scène ou de l'arrière-plan. Comme nous l'avons déjà précisé, les pixels qui représentent une image ne bougent évidemment pas. Ce que nous entendons par fixe (ou mouvement par opposition) sous-entend que les composantes colorimétriques d'un pixel n'évoluent pas significativement au court du temps. Cependant, même dans le cas d'une scène fixe dans laquelle il n'y a pas d'objet en mouvement, les composantes colorimétriques des pixels peuvent quand même évoluer de façon globale (changement de gain de la caméra) ou locale (changement de luminosité, ombre portée d'un objet hors champ, *etc.*). L'approximation « fond fixe / objet en mouvement » est donc un raccourci sémantique qui n'est pas toujours vrai en pratique. Cependant, elle reste communément utilisée dans la communauté.

Par ailleurs, les objets d'intérêts, tels que nous les avons définis dans la section précédente, ne sont pas forcément en mouvement pendant l'ensemble de la séquence temporelle étudiée. Ce sera par exemple le cas lorsqu'un véhicule se gare sur une place de parking, où au contraire, quitte sa place. Nous verrons que ce cas, pourtant relativement simple, pose déjà un certain nombre de problèmes quant à la mise à jour du modèle. En vidéo surveillance, il y a un autre cas plus courant de situation où nous souhaitons maintenir la détection d'une personne même

lorsqu'elle s'arrête temporairement. Un algorithme de segmentation qui ne se focalise que sur la détection du mouvement peut ne pas convenir. D'autres mécanismes devront alors être mis en œuvre.

Dans un contexte de vidéo surveillance un certain nombre de situations rendent la modélisation du fond difficile. Plusieurs auteurs Toyama et *al.* [69], Brutzer et *al.* [70], Bouwmans [71] ont identifié une quinzaine de situations critiques. A cette liste, que nous allons reprendre rapidement, nous pouvons ajouter d'autres situations (identifiées par un astérisque) auxquelles nous avons été confrontés. Nous pouvons regrouper l'ensemble de ces situations critiques dans trois grandes catégories. Une première catégorie est spécifique au capteur ou à la caméra dans son ensemble. Il s'agit essentiellement de perturbations temporaires ou persistantes liées à la capture du signal vidéo. La deuxième catégorie correspond à la scène dans sa globalité et la troisième est liée aux objets d'intérêt à détecter.

3.1 Défis liés à la capture de la scène

Cette première catégorie de défis est liée à la capture et à la transmission du signal vidéo. Dans l'idéal, la suppression ou l'atténuation des problèmes que nous allons exposer peut être obtenue par un meilleur réglage des paramètres de la caméra, son remplacement ou par une meilleure fixation. Cependant dans la plupart des situations réelles, le coût ne doit pas être trop élevé et un système de vidéo protection robuste doit en tenir compte.

Bruit : le bruit se caractérise par une variation aléatoire de plus ou moins fortes amplitudes des composantes colorimétriques d'un pixel. En vidéosurveillance, il y a deux sources principales de bruit. Le bruit lié à la qualité du capteur et le bruit lié à la compression. Dans le cas du capteur, le bruit est souvent plus présent lorsque la luminosité de la scène est faible (Figure 42). Le bruit du capteur est également plus présent sur les caméras dites thermiques. Le bruit de compression est lié à l'encodage et dépend de la bande passante disponible sur le réseau (Figure 43).

Figure 42 : exemple de bruit lié au capteur. Il se manifeste par la présence aléatoire de grain dans l'image

Figure 43 : exemple de bruit lié à la compression. Dans cette image il est plus visible sur le ciel et se manifeste par la coupure franche et aléatoire entre les blocs mais également d'un « GoP » à l'autre.

Ajustement du gain : Les caméras essayent, dans la mesure du possible, de maximiser la dynamique de leur capteur de façon à présenter une image correctement contrastée. Par ailleurs, les capteurs ont également tendance à dériver en fonction de la température. Ce phénomène est d'autant plus important sur les caméras thermiques. En conséquence, un préamplificateur associé au capteur ajuste le gain pour restituer un signal correct. Cette correction peut être brutale ou progressive en fonction de la qualité de la caméra. Visuellement cela se traduit par un changement global de la luminosité (Figure 44), qui n'est pas forcément linéaire, entre deux ou plusieurs frames successives.

Figure 44 : exemple de correction de gain d'une caméra thermique sur une même séquence. En s'approchant de la caméra, la personne, dont la température est significativement différente du reste de la scène, occupe de plus en plus de place dans l'image. Afin d'améliorer le contraste sur la personne, la caméra assombrit tout le reste de la scène.

Vibrations : Sous le terme de vibrations, nous regroupons tous les mouvements intempestifs de la caméra qui entrainent un déplacement du centre optique. Comme indiqué dans nos hypothèses de départ, nous nous plaçons dans le cas d'une caméra fixe où, et en toute rigueur, le centre optique et l'ensemble des conditions de prise de vue ne doivent pas être modifiées au cours de la séquence à analyser. Dans les installations, il n'est cependant pas rare que le support de la caméra ne soit pas correctement adapté voire même que l'ensemble de la structure sur laquelle est placée la caméra subisse des vibrations. Des solutions mécaniques ou logicielles peuvent être utilisées pour corriger ce défaut. Si ce n'est pas le cas ou si ces solutions s'avèrent insuffisantes, un même pixel de coordonnée (r,c) sur deux images successives ne représente plus la même structure dans la scène (cf Figure 45).

Figure 45 : exemple d'une mauvaise caractérisation des pixels en mouvement liée aux vibrations entre deux images successives. Le masque de mouvement, construit à partir de la différence entre les deux images, montre des détections de mouvement sur les zones à fort gradient.

Objectif sale ou mal réglé* : à l'instar des problèmes liés aux vibrations, les problèmes liés à un objectif sale ou mal réglé peuvent être et doivent être réglés par un dimensionnement correct et un paramétrage adéquat. Par ailleurs une maintenance préventive est également en mesure d'en atténuer les effets. Malgré ces recommandations, le nettoyage des caméras peut ne pas être simple à réaliser surtout lorsque la caméra est en hauteur ou, plus généralement, difficilement accessible. Lorsque les caméras sont à l'extérieur, les projections de pluies, la poussière ou les intempéries de manière générale vont progressivement dégrader la qualité de l'image. De même, l'objectif peut être correctement ajusté lors de l'installation mais se dérégler au cours du temps. Il en résulte que les images acquises par la caméra perdent progressivement en netteté. L'image est alors un peu floue. Ce problème ne doit pas être pris à la légère puisqu'il correspond à un grand nombre de cas réels. Une image floue entraine une perte significative de l'information contenue dans la scène. Cela affecte notamment les textures mais également les couleurs.

Cette liste n'est bien sûr pas exhaustive. Nous pourrions ajouter le cas des caméras dites jour/nuit qui transmettent des frames en couleur lorsque la luminosité de la scène est suffisante puis basculent en niveau de gris lorsque la luminosité est faible. Dans ce cas, une solution est de proposer deux modélisations de fond, une pour le mode couleur et une pour le mode niveau de gris. Nous pourrions également ajouter le cas des caméras avec éclairage infra-rouge intégré que nous avons rapidement présentées précédemment ou les problèmes d'éblouissement par les phares des véhicules.

3.2 Défis liés à la complexité de la scène

Cette deuxième catégorie de défis est liée à la complexité de la scène dans sa globalité, indépendamment de la présence ou de l'absence d'objets d'intérêt. Dans un contexte de vidéo surveillance en extérieur, nous pouvons être confrontés à une multitude de situations qui peuvent aller d'un parking goudronné sans arbre, c'est-à-dire une surface bien lisse et uniforme, à un champ d'herbes hautes bordé d'arbres, voir même le lit d'une rivière. De plus, les algorithmes doivent également prendre en compte différentes conditions météorologiques et différentes conditions d'éclairage. Enfin, les algorithmes doivent également être performants sur une durée très longue pouvant s'étaler sur plusieurs mois voire plusieurs années. Dans ces conditions, la scène est susceptible d'être modifiée par l'ajout, le déplacement ou la suppression

de certaines composantes dites statiques. Ces différentes situations et conditions peuvent être résumées par l'ensemble des cas suivants.

Variation lumineuse soudaine : ce type de variation correspond à un brusque changement de luminosité entre deux images successives. Ces variations peuvent être locales ou globales dans l'image. Ce type de variation correspond généralement à l'activation ou la désactivation d'un éclairage extérieur mais il peut également être lié à l'introduction rapide d'un objet massif dans la scène (Figure 46). C'est notamment le cas du passage d'un train dans une entrée ou sortie d'un tunnel ferroviaire (émergence). Associé à ce phénomène rapide, nous devrons également tenir compte du changement de gain que la caméra risque de faire intervenir et qui lui peut être plus ou moins progressif.

Figure 46 : exemple de variation lumineuse sur une émergence d'un métro. Sur les deux images successives d'une même séquence, nous pouvons remarquer un changement busque de la luminosité provoquée par l'entrée du train dans le tunnel. A droite, le masque de mouvement associé est pratiquement saturé.

Variation lumineuse progressive : ces variations, qui peuvent également être locales ou progressives, sont généralement provoquées par le changement des conditions climatiques qui évoluent au cours du temps et par la course du soleil (Figure 47). Elles peuvent également être provoquées par le passage des nuages. A l'instar des variations brusques, ce type de variation peut aussi entrainer un ou plusieurs ajustements du gain de la caméra qui, à l'inverse du cas précédent, seront plus rapide que la variation lumineuse.

Figure 47 : exemple de variations lumineuses sur une même journée

Fond dynamique ou multimodal : comme nous l'avons déjà précisé, les méthodes de suppression de fond s'appliquent principalement au cas de la détection d'objets mobiles à partir d'une caméra fixe. Dans ce contexte, il est alors possible en première approximation de distinguer le "fond" par son caractère "fixe" en opposition aux objets d'intérêt qui sont mobiles. Cette première approximation "fond fixe/objets mobiles" n'est cependant pas toujours vraie en pratique. En effet, certaines parties de la scène voire la quasi-totalité peuvent être dynamiques sans pour autant que cela ne reflète le mouvement d'un objet d'intérêt. Ces cas de fond

dynamique sont nombreux et nous pouvons citer en exemple les branches d'un arbre mue par le vent, un champ d'herbes hautes, les reflets à la surface de l'eau, les flocons de neige, *etc*. (Figure 48)

Figure 48 : exemples de scènes avec des fonds dynamiques

Objet du fond déplacé : ce problème apparaît lorsqu'un objet considéré comme statique dans la scène est déplacé. Le modèle de fond doit être en mesure de réinitialiser toute la zone précédemment occultée par l'objet. Si le modèle ne prend pas en compte ce cas, la silhouette de l'objet reste présente sur le masque de mouvement à l'emplacement de l'objet déplacé (Figure 49). Cette silhouette résiduelle est souvent appelée « ghost » dans la littérature.

Figure 49 : exemple de mauvaise gestion des objets déplacés. Les deux images en partant de la gauche sont issues de la même séquence prise à quelques secondes d'intervalles. La première est prise juste avant le départ d'un camion et la suivante juste après. Le masque de mouvement correspondant à la deuxième image fait bien apparaitre le camion sur le départ mais laisse la trace résiduelle du camion lors de son stationnement.

Objet inséré dans le fond : à l'inverse du cas précédent, ce problème apparait lorsqu'un objet est intégré dans la scène. Ce défi est un peu plus complexe à résoudre que le précédent parce qu'il faut tenir compte de l'intérêt de l'objet. Suivant le cas, on voudra pouvoir intégrer rapidement un objet qui n'est pas d'intérêt mais continuer à détecter un objet d'intérêt qui s'immobilise temporairement dans la scène. Un autre problème lié à l'insertion des objets dans la scène apparait également avec les « ghosts ». Comme la plupart des algorithmes de modélisation, et c'est notamment le cas avec ceux que nous avons présentés, traitent les pixels indépendamment les uns les autres sans tenir compte du voisinage (ou de façon très limité), les pixels d'un même objet ne sont pas forcément intégrés dans le fond au même instant. Nous pouvons donc observer sur le masque de mouvement une forme qui se désagrège petit à petit (Figure 50).

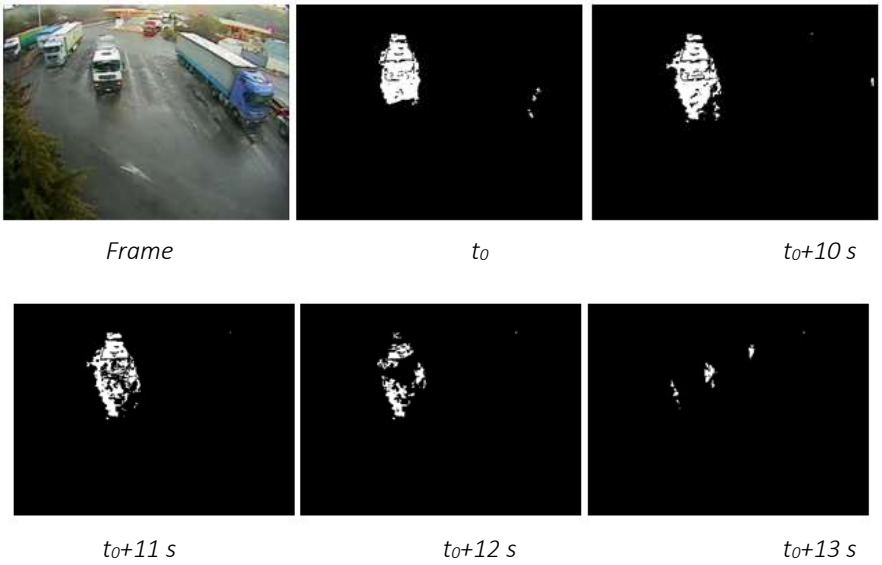


Figure 50 : exemple d'intégration d'un objet dans le fond de la scène. A t_0, un camion que nous pouvons voir sur l'image de gauche, s'immobilise sur une place de parking. Le masque de mouvement à t_0 représente bien la silhouette du véhicule. A t_0 + 10 s, le véhicule n'est pas encore intégré dans le fond de la scène et sa silhouette est toujours reconnaissable (malgré la détection de l'ombre portée sur le devant du véhicule). A partir de t0 + 11 s, il est progressivement intégré au fond. Sa silhouette se désagrège progressivement.

Ombres : la gestion des ombres portées est un autre grand défi de la modélisation du fond. Les ombres projetées par les objets d'intérêt sont en général classifiées à tort comme faisant partie de l'avant-plan (Figure 51), du fait de leur dissemblance avec le fond, compliquant ainsi les étapes ultérieures d'identification et de suivi des objets. Par ailleurs, l'ombre des arbres peut aussi poser quelques difficultés lorsqu'il y a du vent et que l'ombre est projetée sur la zone d'intérêt. On se retrouve alors dans le cas d'un problème de fond dynamique. De façon plus marginale, l'ombre projetée par les objets de la scène peut poser des difficultés lorsque les temps d'intégration du modèle sont très longs ; cas du bagage abandonné par exemple. L'utilisation de caméras thermiques permet naturellement de s'affranchir de cette problématique de l'ombre portée sur les objets d'intérêts.

Figure 51 : exemple classique d'une mauvaise classification de l'ombre projetée par les objets. Sur l'image de gauche nous pouvons remarquer que les quatre individus sont bien isolés les uns des autres. Sur le masque de mouvement (à droite), ils sont, par contre, rassemblés sous un même blob.

Perspective * : ce défi est lié au fait que nous souhaitons détecter aussi bien des objets d'intérêt qui sont proches de la caméra que des objets qui en sont éloignés. Or un même objet n'aura pas la même taille apparente en fonction de sa distance par rapport à la caméra. Par ailleurs, les conditions météorologiques peuvent atténuer le contraste entre l'objet d'intérêt et le fond de la scène en fonction de la distance rendant la détection plus délicate (Figure 52). De même que

la taille apparente diminue en fonction de l'éloignement, la vitesse apparente du déplacement de l'objet dans l'image diminue également. Nous verrons dans la section suivante que le contraste, la taille et la vitesse de déplacement des objets d'intérêt peuvent être problématiques. Un effet de perspective important implique de trouver un compromis entre une bonne détection au loin et un faible taux de fausses alarmes de près.

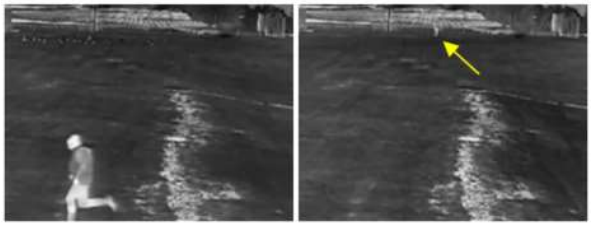


Figure 52 : exemple de scène avec un effet de perspective prononcé. Sur l'image à droite, le bout de la flèche indique la position de la personne dans l'image

3.3 Défis liés aux objets d'intérêt

Cette dernière catégorie de défis concerne plus spécifiquement les objets que l'on cherche à détecter ainsi que leur comportement dans la scène. Ces défis sont donc beaucoup plus liés à la problématique que l'on cherche à résoudre. Toutefois, dans le contexte spécifique de la vidéoprotection, le modèle de fond sera plus ou moins confronté aux défis suivants.

Camouflage : le camouflage, qu'il soit volontaire ou non, se manifeste lorsque le contraste entre l'objet d'intérêt et le fond de la scène est faible (Figure 53). C'est un défi particulièrement difficile à relever mais aussi à évaluer. Il est souvent possible de régler suffisamment bien les paramètres d'un algorithme pour optimiser ses performances sur une vidéo très courte. Nous pouvons donc arriver à détecter à peu près correctement les objets camouflés en augmentant la sensibilité. Cependant, lorsque l'on garde ces paramètres très sensibles sur un site surveillé 24h sur 24, le nombre de fausses alarmes devient vite impossible à gérer.

Figure 53 : exemple de camouflage. Il y a trois personnes dans cette image. Saurez-vous les retrouver ?

Trou dans l'objet d'intérêt : ce problème apparait lorsque des objets de couleur relativement uniformes et peu texturés se déplacent lentement dans la scène ou dans l'axe de la caméra. En fonction des constantes de temps du modèle de fond, les pixels à l'intérieur de l'objet sont plus difficiles à détecter parce qu'ils peuvent être intégrés dans le fond de la scène. Une façon simple pour illustrer ce problème est d'utiliser un modèle de fond « différence » basé sur la simple

différence entre l'image courante et l'image précédente comme nous pouvons le voir sur la figure suivante (Figure 54).

Figure 54 : exemple de détection incomplète d'une personne qui marche dans l'axe de la caméra. Dans cet exemple nous avons volontairement utilisé une constante de temps faible pour illustrer le problème. La couleur de la robe étant uniforme, elle est apprise par le modèle de fond.

Ralentissement ou arrêt : ce défi est très similaire à celui présenté dans la section précédente et que nous avions nommé « Objet inséré dans le fond ». Contrairement à ce que nous avons dit alors, il s'agit ici, de ne pas intégrer dans le modèle de fond l'objet d'intérêt de façon à pouvoir maintenir la détection. La modélisation du fond n'a pas vocation à définir ce qui est réellement un objet d'intérêt pour l'utilisateur ou pas. Son rôle est de permettre la segmentation des objets d'avant plan et donc, au niveau du modèle de fond, tout objet en mouvement est un objet d'intérêt. Il est donc traité de la même manière. C'est finalement à l'utilisateur de fixer les constantes de temps propres à son application et de mettre en place des stratégies de plus haut niveau permettant de gérer le suivi et l'intégration des objets d'avant plan dans le fond. Nous reviendrons plus tard sur ce défi.

Surface (apparente) des objets : le nombre de pixels représentant l'objet d'intérêt dans la scène peut être une source de difficulté. Notamment lorsque l'objet d'intérêt n'occupe que très peu de pixel dans l'image ou à l'inverse lorsque l'objet est très gros.

Parmi les autres défis que nous n'avons pas mentionnés, il y a la vitesse de déplacement des objets. Si la vitesse est faible, on se retrouve dans le cas du « Ralentissement ou arrêt » que nous avons mentionné. Si la vitesse est importante, il peut y avoir un problème lié au capteur, mais les techniques basées sur la modélisation du fond ne sont pas affectées, bien au contraire.

4 Prise en compte du contexte

4.1 Introduction

Afin de rendre un peu plus robuste les algorithmes de détection, nous avons étudié, dans le travail de thèse de Mathieu Roger [10], la possibilité de prendre en compte des informations liées au contexte de la scène dans l'espace ou dans le temps.

Greenhill et *al.* [72] précisent que les travaux de la littérature sur le sujet « apprennent la scène » pendant une certaine durée via une phase d'apprentissage. Contrairement à ce principe, qui peut être coûteux en temps et source d'erreurs, nous avons proposé de prendre en compte un certain nombre de données extérieures connues *a priori*. Ces données sont par exemple : la

position GPS de la caméra, son orientation, ses paramètres intrinsèques et extrinsèques, la météo du jour, des événements spécifiques (passage de trains à proximité par exemple), des lieux spécifiques (parking avec des objets qui peuvent s'immobiliser), mobilier urbain, *etc*. Il s'agit tout simplement de données existantes et plus ou moins faciles à récupérer soit lors de la mise en place de la caméra, soit via Internet.

Grâce à ces données, nous avons non seulement amélioré le processus de segmentation mais également le suivi des objets et, de manière générale, l'ensemble de la chaine de traitements notamment en diminuant le nombre de faux positifs. Ces travaux nous permettent de mieux appréhender le comportement des objets dans la scène en fonction de leur position et de leur apparence mais également en fonction de leur trajectoire.

Dans un contexte de vidéo protection, les caméras ne sont pas placées au hasard mais leur position et leurs orientations sont étudiées en amont de l'installation de façon à assurer une couverture optimale de la zone à surveiller. Comme nous l'avons vu précédemment, des outils de simulations peuvent être mis à disposition des installateurs. Sur la base des principes mathématiques que nous avons présentés, nous avons développé l'application Foxtools. Pour que le rendu soit le plus réaliste possible, l'utilisateur fourni un plan de l'installation ainsi que les différents modèles de caméra qui vont être utilisés. Il renseigne alors les positions des caméras sur le plan, la hauteur à laquelle elles seront installées ainsi que leurs orientations. L'ensemble de ces informations permettent de modéliser la scène, de prédire la zone couverte par chaque caméra et surtout de vérifier la taille apparente d'un objet dans l'image afin de s'assurer que la résolution sera suffisante pour une bonne détection. Afin d'améliorer encore le rendu, l'outil permet également de placer des bâtiments sur plan. Ceci vise à prédire les éventuelles occultations. Pour être efficace, ce travail de pré-déploiement doit être le plus précis possible. Il est alors tentant de pouvoir le réutiliser dans le but d'améliorer la détection et limiter les fausses alarmes.

4.2 OpenStreetMap

Afin d'enrichir notre simulateur, nous nous sommes intéressés à la base de données « OpenStreetMap[6] ». OpenStreetMap est un projet de cartographie du monde qui met à disposition de tous et de manière gratuite une très grande quantité de données géographiques. Parmi ces données géographiques, on retrouve des frontières (pays, régions, villes, etc.), des routes, divers points d'intérêts (station essences, magasins, etc.) et, ce qui nous intéresse particulièrement ici, des bâtiments. Cette richesse d'information est possible grâce aux contributions faites par la communauté construite autour de ce projet. En effet, à l'instar du projet Wikipedia[7], toute personne peut contribuer à l'amélioration de la base de données

[6] http://www.openstreetmap.org

[7] https://www.wikipedia.org

OpenStreetMap. L'utilisation principale de cette base de données géographique est de produire des cartes à partir des données géographiques collectées comme illustré sur la Figure 55.

OpenStreetMap, encode les informations géographiques à l'aide des 4 primitives suivantes :

- Un nœud (node) représente une position d'un point sur terre. Il est décrit par une latitude, une longitude, un identifiant et optionnellement une élévation. C'est la primitive fondamentale permettant d'exprimer des coordonnées sur terre.
- Un chemin (way) est une ligne brisée composée d'un ou plusieurs nœuds. Lorsque le premier et le dernier nœud d'un chemin sont identiques, le chemin est dit fermé, autrement c'est un chemin ouvert. Les chemins ouverts servent principalement à représenter les routes, alors que les chemins fermés permettent de délimiter des régions, en particulier les contours simples des bâtiments.
- Une relation (relation) permet de définir un lien logique ou géographique entre un ou plusieurs nœuds et/ou chemins. Pour ce qui nous intéresse, une relation permet de regrouper les contours d'un même bâtiment en précisant si le contour est intérieur ou extérieur quand le bâtiment a une forme complexe à trous.
- Un tag (tag) est une paire clé/valeur encodant une méta-information à propos d'une autre primitive. En ce qui nous concerne, cela permet de préciser qu'un contour est un contour de bâtiment, et éventuellement de quel type de bâtiment il s'agit (appartements, maison, hôtel, cathédrale, etc.).

Il est important de noter que toutes les informations ne sont pas nécessairement renseignées dans la base de données OpenStreetMap. Il n'y a en effet aucune garantie ni sur l'exactitude, ni sur l'exhaustivité des données renseignées. A minima, les bâtiments auront tous au moins un contour délimitant leur implantation sur le sol. Dans certains cas, la hauteur ou le nombre d'étages seront renseignés, mais cela reste marginal.

Figure 55 : exemple de carte générée à partir de la base de données OpenStreetMap. Sur cette carte figurent les rues avec leur nom, et les bâtiments éventuellement avec leur numéro (Roger, 2015)

À l'échelle mondiale, un peu moins de 355 millions de bâtiments (Août 2019) sont répertoriés, et ce, sur l'ensemble des continents. Si l'on s'intéresse plus particulièrement au cas de la France,

ce sont 47 millions de bâtiments qui sont répertoriés. Le cas de la France est un peu particulier. Cette singularité s'explique par les contributions significatives faites grâce à la libération des données cadastrales par la direction générale des impôts et par la direction générale des finances publiques. La base de données est en constante évolution et de nouveaux bâtiments sont ajoutés continuellement. En 2014, nous avions recensé environ 120 millions de bâtiments et 5 ans plus tard, il y en avait 3 fois plus.

La base de données OpenSteetMap nous permet ainsi d'obtenir une description géométrique des bâtiments entourant la caméra. Nous utilisons un modèle délibérément simple, car il est entièrement construit à partir de connaissances préalables (c'est-à-dire des données spécifiées par l'utilisateur, et non à partir de l'observation et de l'apprentissage par caméra réelle). Nous supposons que le sol est horizontal et plat sur toute la scène : il n'y a pas de trous, ni de pente de terrain. Dans notre formulation, le sol est défini comme le plan z = 0. Ce choix de modélisation simple est motivé par le fait qu'il s'applique à la plupart des situations rencontrées en pratique sans nécessiter de renseignements de la part de l'utilisateur. Deuxièmement, nous modélisons les bâtiments par des polygones extrudés verticalement. En d'autres termes, un bâtiment se compose d'un contour polygonal et d'une hauteur (le toit est plat), comme illustré à la figure suivante (Figure 56).

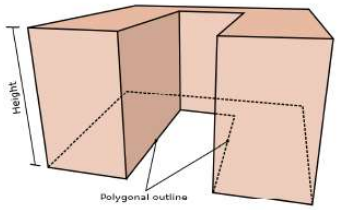


Figure 56 : exemple de représentation d'un bâtiment à l'aide de polygones extrudés

4.3 Carte statique de paramètres

A partir des paramètres intrinsèques et extrinsèques de la caméra et d'un modèle de scène issue de données d'OpenStreetMap, nous pouvons donc produire une image de synthèse correspondant à la projection de la scène sur le capteur de la caméra comme montré dans l'image suivante (Figure 57).

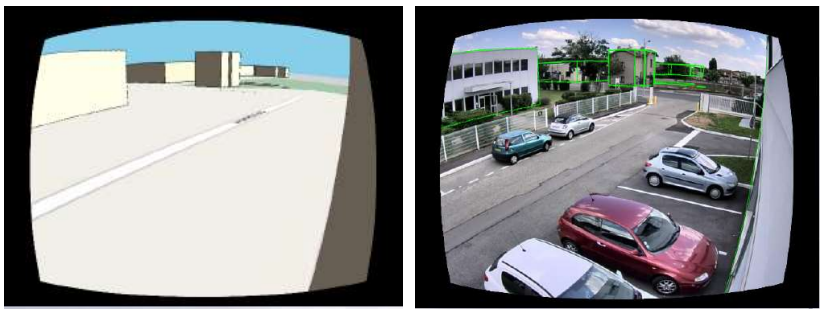


Figure 57 : rendu synthétique d'une caméra et projection des contours des bâtiments (en jaune) dans une vue réelle. Lors du calcul de la projection, nous tenons compte de la distorsion de la caméra.

La projection dans l'image réelle est intéressante dans la mesure où elle permet de vérifier la bonne calibration de la caméra et le bon modèle de scène. Cependant, l'intérêt de ce type de

projection ne s'arrête pas à la simple simulation mais s'avère très utile pour l'analyse vidéo. Le modèle de projection et l'intégration des bâtiments dans la scène virtuelle permettent de définir le plan d'évolution des objets dans la scène, dans la mesure où ceux-ci se déplacent sur le sol, ainsi que les zones d'apparition et de disparition des objets. Par exemple sur la figure suivante (Figure 58), qui est une représentation sémantique de l'image précédente, la zone de circulation est représentée en rouge, les bâtiments en vert et les zones d'apparition et de disparition des objets en bleu. Les pixels noirs correspondent aux zones de la scène où nous n'avons pas d'information. Cela correspond au ciel ou aux zones de l'image qui sont trop éloignées de la zone utile à analyser.

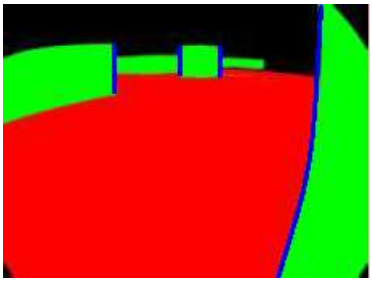


Figure 58 : représentation sémantique d'une scène où le plan de circulation est représenté en rouge, les bâtiments en vert et les accès en bleu.

L'accès à cette représentation sémantique de la scène s'avère très intéressante notamment dans la phase de suivi. Cette carte permet notamment d'identifier les zones potentielles d'apparition ou de disparition des objets dans la scène. Par exemple, dans l'image précédente (Figure 58), un objet qui touche un trait bleu est susceptible d'entrer ou de sortir de la scène en passant derrière un bâtiment. Cette information peut être mise à profit pour adapter le processus de suivi.

4.4 Suppression des ombres

4.4.1 Exposé du problème

Nous avons débuté nos travaux sur l'utilisation du contexte pour proposer une solution permettant de gérer convenablement les ombres portées. En effet, les ombres portées des objets en mouvement sont souvent très mal dissociées de l'objet qui les a projetées, voire pas du tout, en particulier pour les objets qui évoluent sur le sol, tels que les voitures ou les piétons. Sanin et *al.* [73] présentent un état de l'art assez complet des techniques de détection des ombres et classent les algorithmes de détection des ombres en fonction des caractéristiques utilisées :

- Les méthodes basées sur la chromaticité utilisent généralement un modèle d'atténuation de couleur linéaire : un pixel ombré diminue son intensité (c'est-à-dire s'assombrit) sans modifier la chromaticité. Dans ce contexte, il est souvent souhaitable d'utiliser un espace colorimétrique facilitant la séparation intensité / chromaticité telle que le HSV [74] ou le CIELAB [75].
- Les méthodes basées sur la physique utilisent les propriétés physiques des sources de lumière et/ou les propriétés physiques des surfaces des matériaux afin de détecter les ombres. Nadimi et Bhanu [76], par exemple, modélisent à la fois les contributions du soleil (lumière blanche) et du ciel (lumière bleue) pour construire leur modèle

d'atténuation de couleur. Huang et Chen [77] utilisent un modèle d'éclairage plus général appelé modèle de réflexion dichromatique bi-illuminant pour détecter les ombres. Finlayson et *al.* [78] émettent des hypothèses sur le capteur de la caméra afin de dériver une image intrinsèque et de supprimer les ombres.

- Les méthodes basées sur la géométrie déduisent les ombres projetées de la description géométrique des objets composant la scène courante. Ces méthodes sont bien adaptées à des ombres d'objets spécifiques telles que les voitures [79] ou les piétons. Ils supposent qu'il n'y a qu'une seule source de lumière et que l'ombre est projetée sur une surface plane.
- Les méthodes basées sur la texture supposent que les caractéristiques de texture d'une région donnée sont généralement préservées lorsqu'elles sont situées dans l'ombre d'un objet. Ces méthodes fonctionnent généralement en deux étapes: d'abord, elles sélectionnent les pixels d'ombre candidats (un détecteur d'ombre faible convient parfaitement), puis mettent en corrélation la texture de la région candidate de l'image en cours avec celle du modèle de fond [80].

4.4.2 Modélisation

Dans le travail de thèse de Matthieu Roger [10], nous avons proposé une autre approche permettant de traiter le problème de l'ombre. Nous avons exploré la possibilité d'utiliser des connaissances contextuelles facilement disponibles, telles que les coordonnées GPS de la caméra ainsi que la date et l'heure de l'observation, pour prédire les pixels de l'image susceptibles de faire partie de l'ombre lié à l'ensoleillement.

La première étape de notre modèle de prévision des ombres estime les ombres projetées par les bâtiments environnants. Un aspect novateur de notre travail consiste à inclure les données OpenStreetMap afin de créer un modèle de scène géométrique. La construction d'un modèle géométrique pour son utilisation en vidéo n'est en soit pas nouvelle. En effet, Jackson et *al.* [81] construisent leur modèle de scène en l'apprenant à partir de l'observation des occultations des objets en mouvement. D'autres auteurs [82] [83] utilisent des modèles 3D de scènes virtuelles hautement détaillées. Dans notre cas, nous utilisons un modèle de caméra sténopé classique et un environnement 3D basé sur les données d'OpenStreepMap. Les bâtiments sont donc modélisés à partir de polygones extrudés comme indiqué précédemment. Cette première étape permet d'indiquer les zones de l'image qui sont dans l'ombre des bâtiments et dans lesquelles nous n'avons pas à rechercher l'ombre des objets en mouvement.

La deuxième étape consiste à estimer l'ombre portée des objets en mouvement de façon à la supprimer du masque de mouvement. Cette deuxième étape est un peu plus complexe mais utilise le même principe de projection des ombres.

Par souci de simplicité, nous ne considérons que les ombres causées par le soleil et projetées sur le sol. De plus, nous supposons que le soleil se comporte comme une lumière directionnelle, c'est-à-dire une lumière ponctuelle placée à l'infini. Cette simplification est justifiée par le fait que la distance entre la terre et le soleil est beaucoup plus grande que les distances impliquées dans la scène.

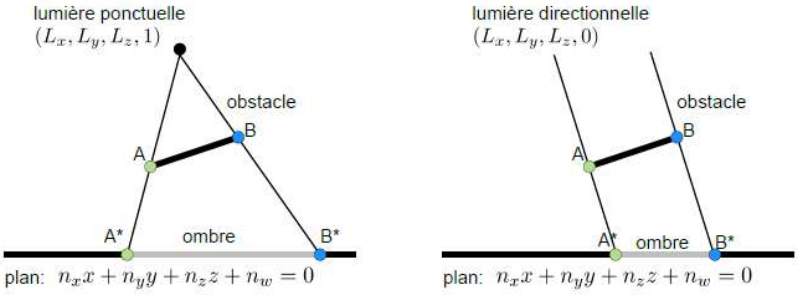


Figure 59 : illustration de la technique de projection plane des ombres.

Avec l'ensemble de ces hypothèses ainsi que la paramétrisation des sols plats mentionnée précédemment, les ombres peuvent facilement être calculées en utilisant une projection parallèle comme décrit dans [84]. Nous projetons des sommets sur le plan de sol (z = 0) parallèlement à la direction du soleil. Une telle projection peut être réalisée avec la matrice P suivante :

$$P = \begin{pmatrix} L_z & 0 & -L_x & 0 \\ 0 & L_z & -L_y & 0 \\ 0 & 0 & 0 & L_z \end{pmatrix}$$

où $\vec{L} = (L_x, L_y, L_z)$ est le vecteur de la direction du soleil. Puisque nous ne nous intéressons qu'à la direction de L pour la projection de l'ombre, nous pouvons imposer une contrainte de normalisation et la paramétrer avec seulement deux angles : azimut et altitude, définis par rapport au nord (respectivement horizon) et positifs vers l'est (respectivement zénith).

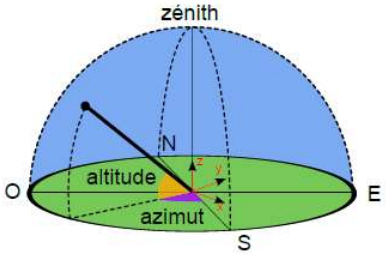


Figure 60 : système de coordonnées horizontales permettant d'exprimer la direction d'observation du Soleil. Nous avons également fait figurer le repère de la scène (en rouge) afin de faciliter la visualisation de la conversion donnée.

Dans ce système, l'altitude est repérée par rapport au plan horizontal local d'observation et varie entre 0° (horizon) et 90° (zénith). L'azimut est, quant à lui, repéré par rapport au Sud (0°) et varie positivement vers l'Ouest (+90°), négativement vers l'Est (-90°). Notons que d'autres conventions sont également couramment utilisées pour paramétrer l'azimut, notamment en repérant la direction à partir du Nord.

Pour convertir ces angles d'azimut (Γ) et d'altitude (a) en une direction (L_x, L_y, L_z), exprimée dans le repère de la scène, on utilisera l'équation suivante :

$$\begin{cases} L_x = \cos(a)\cos(\Gamma_N + \pi - \Gamma) \\ L_y = \cos(a)\sin(\Gamma_N + \pi - \Gamma) \\ L_z = \sin(a) \end{cases}$$

Où Γ_N désigne l'angle entre l'axe des abscisses du repère de la scène et la direction du nord.

Pour calculer la projection des ombres, il ne nous reste plus qu'à déterminer la direction du soleil en fonction des coordonnées GPS, de la date et de l'heure. Le calcul de la position du soleil n'est pas trivial car elle est affectée par de nombreuses perturbations : influence de la lune, diminution de la vitesse de rotation de la Terre, réfraction atmosphérique, *etc*. Plusieurs auteurs ont proposé divers algorithmes [85][86][87][88] traduisant un compromis différent entre l'exactitude de la prédiction (au cours d'une période de validité donnée) et la complexité du modèle. Nous avons basé notre travail sur l'un des algorithmes proposés par Grena [89] qui permettait de trouver le meilleur compromis entre précision (erreur maximale de 0,009°) et complexité. Outre les coordonnées de position et l'heure d'observation, cet algorithme corrige la position apparente du soleil en fonction de la température, de la pression et de l'irrégularité de la rotation de la terre ΔT. La pression et la température sont utilisées pour la correction de la réfraction. Dans notre implémentation, nous avons choisi de fixer ces trois derniers paramètres à des approximations raisonnables, respectivement : 25°C, 1 atm et 67 s.

4.4.3 Prédiction des ombres liées à la scène

Nous avons à présent tous les éléments permettant de calculer l'ombre portée des bâtiments situés dans le voisinage de la caméra. Afin de valider notre modèle, nous avons comparé les masques obtenus avec des images réelles [10]. L'image suivante (Figure 61) montre une série d'acquisitions réalisées entre le 18/07/2012 14h et le 19/07/2012 11h sur le parking bordant les locaux de la société Foxstream (GPS : N45.7539326 E4.925457). Dans cette séquence, nous constatons que les contours des bâtiments ainsi que leurs ombres sont globalement correctement prédits.

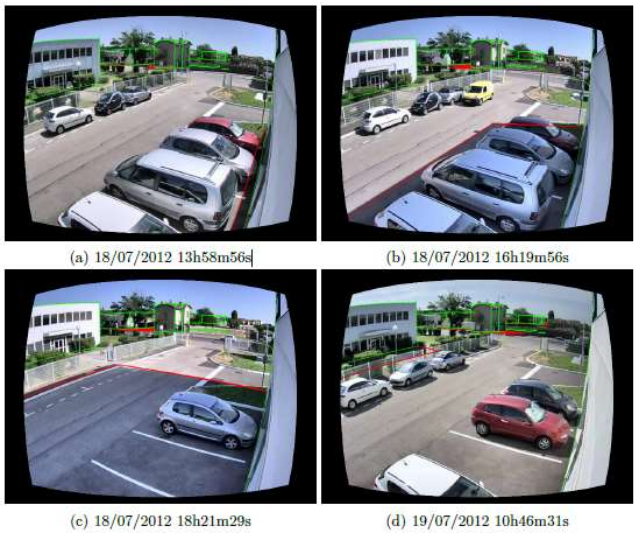


Figure 61 : exemple de prédiction des ombres sur différents moments de la journée. Les contours des bâtiments sont représentés en vert et les contours de l'ombre prédite en rouge.

4.4.4 Suppression de l'ombre des piétons

Le calcul de l'ombre portée des bâtiments nous a surtout permis de valider l'ensemble de la chaine de modélisation et de rendu 3D. Fort des excellents résultats obtenus, nous avons

poursuivi ce travail pour supprimer l'ombre des piétons du masque de mouvement. La difficulté supplémentaire que nous avons ici et que, contrairement aux bâtiments, nous n'avons pas la position des personnes dans la scène. Nous ne disposons que d'un masque de mouvement comme présenté dans la figure suivante (Figure 62) et d'un modèle de projection.

Figure 62 : exemple d'une image avec son masque de mouvement associé

Par souci de simplicité, nous modélisons les piétons par un cylindre de 60 cm de diamètre et d'1m75 de hauteur, posé sur le sol. L'utilisation d'un cylindre permet de nous affranchir de l'orientation des piétons. A partir de ce modèle de piéton et de la calibration de la caméra, nous pouvons calculer la projection d'une personne debout sur l'ensemble de la scène et calculer le rectangle englobant de la silhouette dans l'image.

Figure 63 : exemple de projection d'un cylindre représentant un piéton dans une scène préalablement calibrée. La grille (en bleu) permet de visualiser le plan pour z=0 (le sol). Le cylindre (en jaune) est la projection de notre modèle cylindrique de personne et le rectangle rouge représente le rectangle englobant.

Nous utilisons le rectangle englobant de façon à accélérer la recherche des zones du masque de mouvement susceptibles de contenir la projection des personnes avec leur ombre. En effet, l'utilisation de boîtes rectangulaires alignées sur les axes de l'image se prête bien à des détections accélérées grâce aux images intégrales [22]. Une fois les zones candidates détectées, nous pouvons reprendre notre modèle cylindrique de personne pour le projeter dans la scène mais surtout pour estimer l'ombre portée d'une personne. Nous pouvons alors supprimer les pixels détectés en mouvement qui se superposent à l'ombre calculée. Un certain nombre de précautions sont prises avant de supprimer un pixel, notamment, nous vérifions si le pixel n'appartient pas à une autre personne. La figure suivante (Figure 64) reprend les différentes étapes du processus.

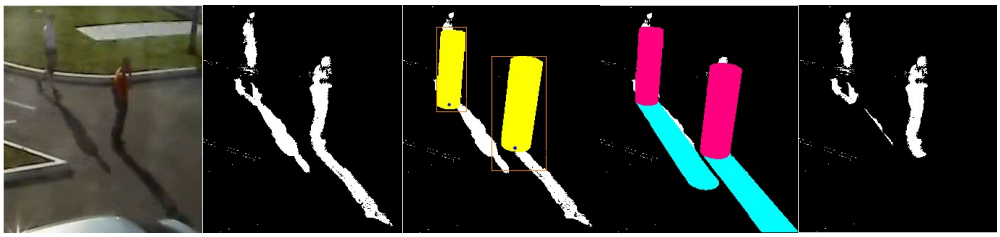


Figure 64 : illustration des principales étapes de notre algorithme de soustraction de l'ombre des piétons. Avec de gauche à droite : l'image courante, le masque de mouvement associé, le calcul des régions candidates (cylindre vert), l'estimation de l'ombre portée par ces régions candidates et le résultat final.

Sur cet exemple, nous pouvons remarquer que la solution proposée est robuste lorsque le masque de mouvement initial est de mauvaise qualité. Par ailleurs, la méthode reste également robuste même lorsque les paramètres intrinsèques et extrinsèques de la caméra ne sont pas connus avec exactitude. Sur la figure suivante (Figure 65), nous présentons rapidement quelques résultats sur différentes séquences vidéo. Pour les deux premières vidéos, nous avions à notre disposition l'ensemble des informations permettant de générer les modèles de caméra et de scène. Pour les deux suivantes, qui sont issues du challenge PETS 2009[8], nous avons pu les estimer simplement et les résultats sont satisfaisants.

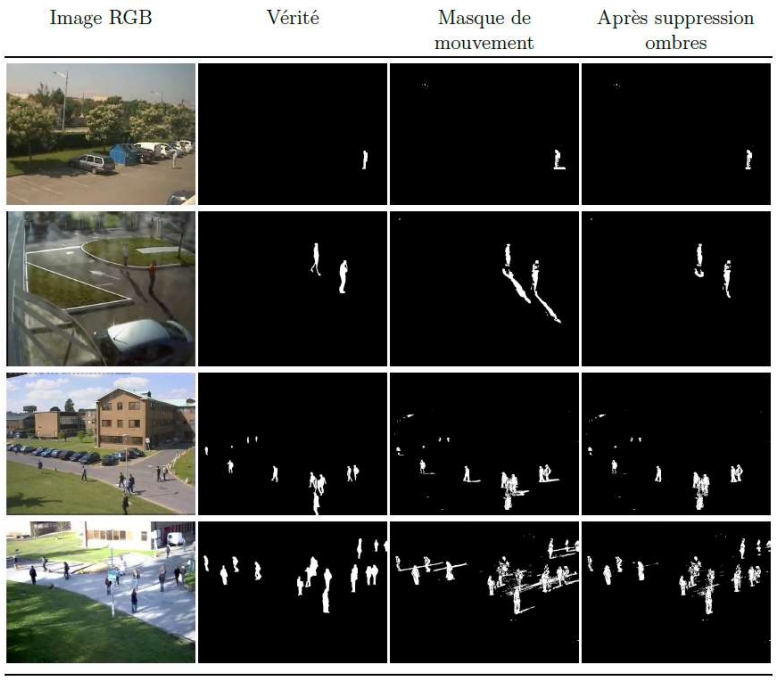


Figure 65 : résultats de la suppression d'ombre sur différentes vidéos.

Nous avons principalement axé notre travail sur la suppression de l'ombre des piétons qui constituent une cible majeure en vidéo protection mais la méthode proposée s'applique à tout autre type d'objet dans la mesure où un gabarit standard peut être utilisé (Figure 66). Nous

[8] http://www.cvg.reading.ac.uk/PETS2009/

l'avons d'ailleurs utilisé pour supprimer l'ombre des véhicules mais pour cela nous devons faire face à une autre difficulté. En effet, contrairement aux piétons, nous pouvons difficilement nous affranchir de l'orientation du véhicule. Nous aurions pu utiliser le flot optique pour déterminer la direction de circulation du véhicule et orienter notre gabarit en conséquence mais nous avons préféré utiliser les informations de suivi (position, taille, vitesse) obtenues à partir des images précédentes.

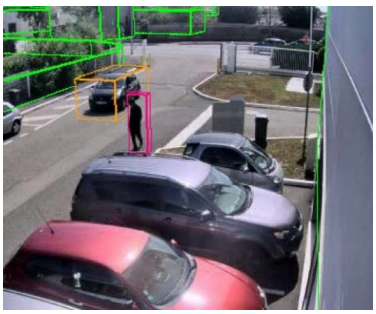


Figure 66 : illustration du gabarit (orange) utilisé pour un véhicule.

L'intégralité de la méthode proposée ainsi que le processus de validation sont décrits dans [10]. Par ce travail, nous avons pu montrer que l'utilisation du contexte apporte une véritable valeur ajoutée au processus d'analyse.

5 Détecteurs spécialisés

5.1 Introduction

Dans le cadre de notre activité de recherche, nous nous sommes principalement attachés à étudier et proposer des solutions de détection basées sur la modélisation de fond. Les raisons principales de ce choix sont que cette approche est bien adaptée aux objets en mouvement dans une scène fixe et que son temps de calcul est faible. La première raison permet de ne pas faire d'hypothèses sur le type ou la taille des objets à détecter et de se focaliser sur le mouvement des objets dans la scène. La deuxième raison permet de multiplier les analyses sur une même machine et donc de limiter les coûts de déploiement. Cependant, il y a plusieurs situations où le type d'objet est clairement défini et où l'utilisation d'un détecteur spécialisé est pertinente.

Dans le cas de la vidéo protection, les deux types d'objets qui représentent un intérêt certain sont les personnes et les visages. Il est donc tentant d'utiliser des détecteurs spécialisés, c'est-à-dire des détecteurs spécialement entrainés pour extraire un type d'objet particulier. Cependant, un détecteur de piétons n'est pas entièrement adapté à la plupart des situations de détection périmétrique. En effet, dans ce cas, nous ne pouvons pas présager de la façon dont la personne va essayer de s'introduire dans le site. La personne peut par exemple ramper, courir, se dissimuler sous un drap, *etc*. Pour utiliser un système basé sur un détecteur spécialisé, il faudrait un détecteur pour les personnes debout, un autre pour les personnes qui rampent, un autre pour les personnes accroupies et un autre encore pour les personnes qui marchent à quatre pattes. En théorie, il pourrait y en avoir qu'un seul. Il suffirait de l'entrainer avec suffisamment d'exemple de personnes dans différentes positions.

Par contre, la détection de visages joue un rôle primordial dans certaines applications de sécurité. Elle conditionne de façon directe le bon fonctionnement de systèmes comme la reconnaissance faciale, la vidéo-surveillance ou l'analyse d'émotions. La détection de visage est par ailleurs largement utilisée dans les appareils photos, avec l'implémentation des travaux de Viola & Jones [22] depuis plus d'une dizaine d'années. Néanmoins, malgré les récents progrès, les problématiques d'occultation, de conditions d'éclairage et de variations de positions sont autant de challenges que la détection de visage n'est capable de résoudre que partiellement. C'est dans ce contexte que les réseaux de neurones à convolution (CNNs) ont réussi à se démarquer, initiés par AlexNet [90], pour la classification d'objets [91].

Dans [92], les auteurs différentient quatre groupes d'algorithmes de détection de visage : les méthodes fondées à partir de la connaissance humaine (*i.e.* caractéristiques construites "à la main"), l'utilisation de caractéristiques invariantes, la mise en correspondance de modèle et les méthodes basées sur l'apparence. En 2015, Zafeiriou et *al.* [93], se limitent à deux catégories de détecteurs. Les méthodes dites à modèle rigides dont l'approche de Viola Jones, les CNNs ainsi que les méthodes d'extraction de caractéristique (HoG, Webber, ...) et les méthodes fonctionnant sur le principe de modèles de structures déformables comme les « Deformable Parts-based Model » (DPM).

Cependant, depuis 2015, les travaux se sont largement dirigés vers les CNNs, désormais regroupables en deux familles, laissant de côté la plupart des méthodes désuètes citées précédemment. Nous allons rapidement présenter les méthodes évoquées dans [92], avant de consacrer une large partie aux deux familles principales de réseaux de neurones : les réseaux basés sur la proposition de région et les réseaux "end-to-end" dans lesquels la détection est gérée en une seule étape.

5.2 Détecteurs « à l'ancienne »

5.2.1 Méthode de Viola et Jones

Proposée en 2001, c'est sur la base des travaux de Viola et Jones [22] que reposent de nombreux modèles encore utilisés aujourd'hui dans nos caméras. Elle était révolutionnaire à l'époque car elle permettait pour la première fois la détection en temps réel.

Globalement, le principe est de parcourir l'image en entier et de discriminer petit à petit des sous-zones via des caractéristiques très simples utilisées comme classifieur en cascade (Figure 67).

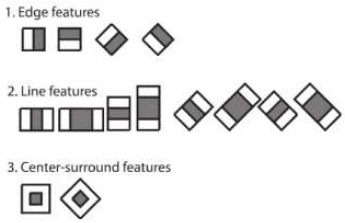


Figure 67 : exemple de classifieurs basiques utilisés dans l'algorithme de Viola et Jones issu de [22]

L'algorithme proposé par Viola et Jones s'est distingué par plusieurs innovations majeures :

- L'image intégrale : en chacun des points de l'image est associée la somme des pixels située à sa gauche et au-dessus de lui. Cette opération simple améliore considérablement l'efficacité du système, puisque le calcul de la somme d'une zone rectangulaire alignée sur les axes de l'image (opération typiquement utilisée pour les descripteurs de bas niveau) ne demande qu'une addition ou soustraction de 4 valeurs.
- La méthode d'AdaBoost Learning : il s'agit de combiner plusieurs classifieurs "faibles" pour aboutir à un classifieur "fort". Le modèle est entrainé pour trouver les meilleurs seuils de caractéristiques permettant de séparer les visages des échantillons négatifs.
- La cascade de classifieurs : chacun des classifieurs est appliqué de manière séquentielle et binaire. Pour chaque classifieur, les échantillons négatifs sont rejetés définitivement. L'algorithme applique le classifieur suivant tant que l'échantillon est positif. Cette architecture en cascade permet de rejeter la grande majorité des candidats rapidement (Figure 68). Les classifieurs sont par ailleurs appliqués par complexité croissante : Les premiers seront les plus simples, donc les plus faciles à calculer, et permettront de discriminer efficacement l'immensité des candidats négatifs.

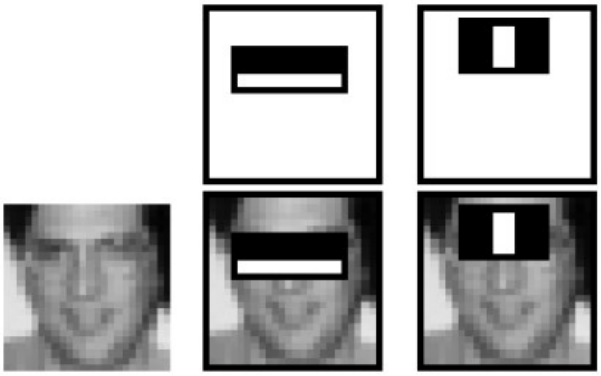


Figure 68 : exemple des premières caractéristiques du classifieur : La première évalue la différence d'intensité entre les yeux et les joues. Si la zone des yeux est plus foncée, l'algorithme passe au classifieur suivant. La seconde compare l'intensité entre les yeux et le nez. De la même façon, si les zones oculaires sont plus foncées, l'algorithme passe au classifieur suivant. Illustration issue de [23].

5.2.2 Modèles à partir d'extraction de caractéristiques

Inspiré par la méthode de Viola & Jones, plusieurs travaux ont utilisé des architectures similaires avec des caractéristiques à extraire (initialement les Haar-like features) et un algorithme d'apprentissage par boosting. L'objectif était d'établir des caractéristiques de plus en plus efficaces, et d'adapter le boosting pour le rendre plus performant.

Initialement, les travaux se sont concentrés sur les ondelettes de Haar proposées par Viola & Jones, avec des rectangles pivotés à 45° [94], ou des caractéristiques combinées entre elles [95]. Cependant, les caractéristiques pseudo-Haar ont rapidement montré leurs limites avec notamment une sensibilité trop accrue aux conditions d'éclairage. C'est dans ce contexte que les motifs binaires locaux (Local Binary Patterns, LBP)[96] ont été proposés pour la détection de visages et de personnes. L'intérêt des LBP est qu'il s'agit de descripteurs robustes aux problèmes d'éclairage, qui comparent la luminance d'un pixel avec son voisinage. D'autres caractéristiques de plus en plus sophistiquées ont été proposées, certaines à base de modèle de visage [97], ou d'autres à base d'histogrammes avec notamment les histogrammes de gradient orienté [98]. Un récapitulatif assez complet de modèle d'extraction de caractéristiques est donné dans [93]. Il est à noter que la plupart de ces méthodes ne s'appliquent pas seulement à la détection de visages ou de piétons mais peuvent très bien se généraliser à d'autres types d'objets. La contrainte forte reste quand même que les objets doivent toujours présenter le même profil. Mais il s'agit ici d'une simple limitation liée à l'apprentissage. Dans le cas des visages, nous nous attendons à avoir un visage vu de face. L'apprentissage est donc réalisé avec ce type de prise de vue. Il est tout à fait possible d'apprendre un détecteur de visages vus de profil. Par contre, le détecteur aura plus de difficulté à détecter les personnes vues de face.

A l'instar des recherches qui ont été faites sur les descripteurs, plusieurs travaux se sont concentrés à améliorer l'algorithme d'apprentissage et de boosting. Dans [93], le lecteur pourra également trouver un tableau assez exhaustif des différentes approches proposées dans la littérature.

5.2.3 Modèles déformables

Une dernière catégorie de modèles utilisés dans [99] a largement démontré son efficacité, avec des résultats atteignant l'état de l'art jusqu'en 2015. Le principe est basé sur les « pictorial structures ». L'objet ou le visage à détecter est défini comme un ensemble de sous-parties modélisées indépendamment (Figure 69). La structure globale est formée de ces sous-parties reliées entre elles par paires.

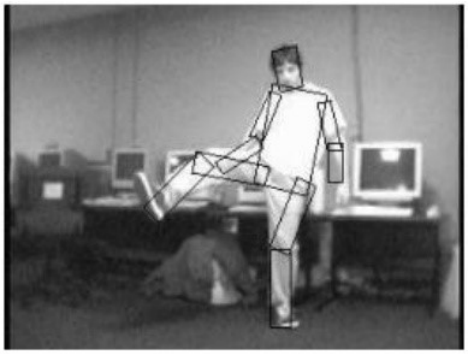


Figure 69 : exemple de "pictorial structures" utilisées pour la détection de personnes [99]

Avec cette approche, un visage peut être modélisé avec des sous-parties comme les yeux, le nez, la bouche. La structure globale doit alors respecter une forme de visage humain c'est-à-dire, les yeux au-dessus du nez, l'écart entre la bouche et les yeux pas trop grand *etc*. Chaque sous-partie utilise ses propres classifieurs. Ces classifieurs sont alors combinés, ce qui permet de prendre en

compte la localisation spatiale relative à chaque sous partie du visage. Par exemple une, configuration qui indique une bouche au-dessus du nez sera largement pénalisée. En d'autres termes, la déformation globale de la structure doit être minimisée.

Malgré leurs relatives bonnes performances en termes de détection, les DPM souffrent de deux défauts majeurs. Le premier est leur besoin d'une forte capacité de calcul comme pour les CNNs avec un nombre de paramètres souvent très élevé. Le deuxième concerne l'effort d'annotation particulièrement trop coûteux. Pour ces modèles fortement supervisés, il est nécessaire d'annoter la localisation de chacune de nos sous-parties de visage dans la base de données d'entrainement. Par contre, l'approche DPM nécessite beaucoup moins de données d'entraînement qu'un CNN classique.

Les modèles DPM se sont montrés particulièrement efficace jusqu'en 2015. Ils ne sont néanmoins plus d'actualité aujourd'hui car dépassés par les réseaux de neurone à convolution.

5.3 Convolutional Neural Network (CNN)

La présentation du réseau AlexNet [90], au concours ImageNet en 2012 a rapidement relégué les autres approches de classification d'objet au second rang, avec des résultats significativement supérieurs. Une telle fracture n'a pas eu lieu dans le domaine de la détection de visages, pour plusieurs raisons. D'une part, les données d'entraînement spécifique aux visages annotés étaient, jusqu'à récemment, de taille trop réduite pour être suffisantes à l'apprentissage d'un réseau profond. D'autre part, les modèles d'extraction de caractéristiques étaient, comme nous l'avons vu, déjà plutôt efficaces, car adaptés à une tâche précise. Ce n'est qu'à partir de 2014 avec l'approche proposée par Zhang et *al*. [100] que des premiers résultats convaincants sont apparus. Zhang n'est pas le premier à proposer une architecture à base de réseau de neurone à convolution pour la tache de détection de visage. En effet, dès 2002, Garcia et *al.* [24] ont proposé ce type d'architecture avec des résultats équivalents à ceux de Viola & Jones, voire même un peu plus robuste à l'orientation du visage.

Nous allons aborder les deux grandes familles d'architecture de CNN utilisées pour la détection de visages et pour la détection d'objets de manière générale. Il s'agit de l'architecture basée sur la localisation de région puis sur la classification de ces régions et, de l'architecture dite « single shot » où les deux opérations sont réalisées en même temps.

5.3.1 Les réseaux avec proposition de région et classification

Une approche assez intuitive pour la détection est de faire glisser une fenêtre sur une image pour localiser les objets, avec différentes tailles et échelles. Grâce à un système de recherche sélective [101], l'algorithme génère des régions d'intérêt qui sont ensuite classées par un autre mécanisme. C'est cette architecture qui est notamment utilisée dans le réseau RCNN proposé par Girshick et *al.* [102] (Figure 70).

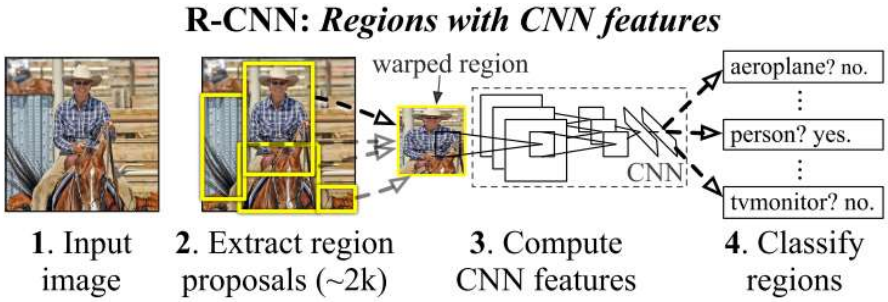


Figure 70 : architecture du modèle RCNN [102]. *Une première étape de recherche sélective permet de proposer une carte de régions candidates. Dans une deuxième étape, différentes couches de convolutions permettent d'extraire des caractéristiques. La couche terminale est ensuite passée à un réseau entièrement connecté qui déterminent la boite englobante des objets et la classe associée.*

L'algorithme a ensuite été régulièrement optimisé par l'équipe de recherche qui a finalement proposée l'architecture Faster-RCNN [103] (Figure 71). Parmi les différentes optimisations l'une des principales est l'utilisation d'un premier réseau à base de convolution dont les sorties vont être utilisées à la fois pour proposer des régions candidates et pour la phase de classification.

Le mécanisme de proposition de région est lui-même un réseau de neurone appelé RPN (Region Proposal Network). Ce RPN prend donc en entrée les cartes de caractéristiques de sortie du premier réseau de convolution. Il utilise des filtres 3 × 3 qu'il fait glisser sur les cartes de caractéristiques pour faire des propositions de régions agnostiques par classe en utilisant un réseau convolutionnel. Dans le cas de Faster-RCNN les auteurs ont utilisé le réseau ZF [104]. D'autres réseaux tels que VGG net [105] ou ResNet [106], peuvent être utilisés pour une extraction plus complète au détriment de la vitesse. Le réseau ZF génère 256 valeurs, qui sont alimentées en 2 couches séparées et entièrement connectées pour prévoir une boîte englobante et 2 classes possibles : une pour la catégorie « avoir un objet » et une « sans objet ».

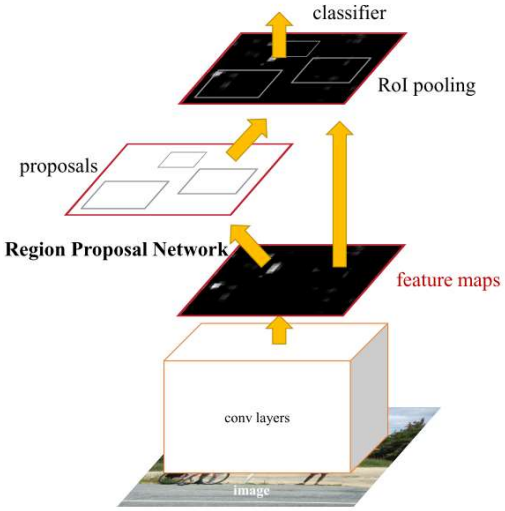


Figure 71 : architecture simplifiée de « Faster RCNN » [103]. *Un premier réseau de convolution est utilisé comme entrée du réseau de proposition de région et du réseau de classification*

Dans [107] les auteurs proposent différentes améliorations de Faster-RCNN dédiées spécifiquement à la détection de visage. Ils utilisent notamment les techniques de « Hard

Negative Mining » déjà utilisée par Wan et *al.* [108] dans un contexte similaire, la concaténation des cartes de fonctionnalité [109] et réalisent un apprentissage multi-échelles.

5.3.2 Les réseaux dit « single shot »

La seconde grande famille des détecteurs d'objets se base sur une architecture assez différente. Les réseaux comme Faster R-CNN sont dits à deux étages. Un premier étage sélectionne des régions susceptibles de contenir un objet et un deuxième étage détermine la classe de l'objet contenu dans chaque région candidate. Dans cette deuxième famille de détecteurs, le principe est de réunir ces deux étapes en une seule, en proposant les boites englobantes et la classe associée en même temps, d'où le terme de "single shot".

Dans un réseau de convolution type R-CNN, le réseau de convolution principal renvoie 5 paramètres : les coordonnées de la boite englobante, et un indice de confiance (y a-t-il un objet ou non dans la boite englobante). Ici, un tel réseau va en plus renvoyer les probabilités pour chacune des classes potentielles. Au lieu des 5 paramètres, le réseau retourne donc pour chaque prédiction 5+k paramètres où k représente le nombre de classes possibles (typiquement de l'ordre de la centaine ou du millier). L'avantage majeur de ce principe est d'accélérer considérablement le processus (la proposition de région étant assez coûteuse) au détriment d'une précision souvent un peu diminuée.

Il existe deux principales références dans la détection à un étage sur lesquels les travaux continuent de se baser aujourd'hui : SSD [110] et YOLO [111]. La grande majorité des détecteurs de visage à un étage s'inspirent du détecteur SSD.

L'approche SSD [110] est basée sur un réseau de convolution qui produit une collection de taille fixe de boîtes englobantes et de scores indiquant la présence de classe d'objets dans ces boîtes, suivie d'une étape de filtrage pour ne garder que les détections finales. Les premières couches du réseau reposent sur une architecture standard VGG net [105] utilisée pour la classification des images de haute qualité (tronquée avant toute couche de classification), que les auteurs appellent le réseau de base. Ce réseau permet classiquement d'extraire des cartes de caractéristiques. Une structure auxiliaire est ajoutée à ce réseau tronqué pour produire les détections. Cette structure auxiliaire a été spécialement définie pour permettre une détection multi-échelles. Chaque couche convolutive produit une carte de caractéristiques pour la couche suivante mais est aussi utilisée par le module de décision final pour définir les régions d'intérêt et la classe de l'objet associé.

Le principe de YOLO [111] est un peu différent. Le but reste le même, à savoir : utiliser un même réseau neuronal pour détecter, classifier et isoler les objets dans des boites englobantes. La différence est qu'ici, l'image d'entrée est découpée en une grille régulière. Si le centre d'un objet tombe dans une grille, la cellule est responsable de la classification de l'objet. Comme une cellule ne peut avoir qu'un nombre fini de d'objet, le nombre total d'objets détectables dans une image est limité. A l'instar de SSD, la première partie du réseau se base sur une architecture standard (Figure 72). Dans le cas de YOLO, les auteurs utilisent les premières couches d'un réseau GoogLeNet [112]. Ce réseau tronqué est suivi de deux couches entièrement connectées.

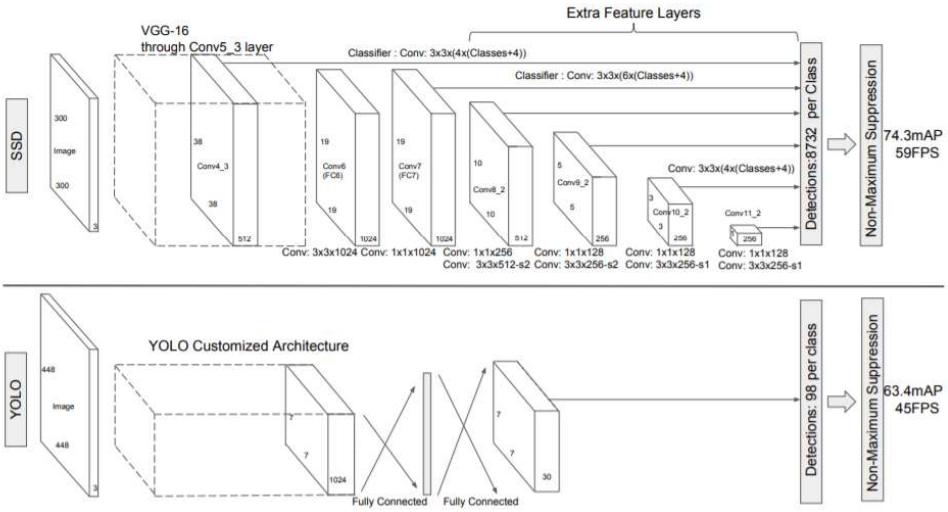


Figure 72 : comparaison des architectures SSD et Yolo. Sur cette figure, nous pouvons remarquer l'aspect multi-échelle mis en avant par les auteurs de SSD [110]

Dans le cas spécifique de la détection de visages, Li et *al.* [113] ont proposé le réseau « Dual Shot Face Detector- DSFD ». Leur approche est basée sur SSD. La tâche mono-classe de détection de visage permet aux détecteurs SSD d'être particulièrement efficaces et de rivaliser avec les détecteurs à base de RPN en termes de précision, contrairement à la détection d'objets où ils sont souvent choisis pour leur rapidité plutôt que pour la justesse de leur détection, notamment dans le choix de la classe d'objets.

5.3.3 Comparaison des deux familles

En 2017, une équipe de chercheurs de Google a réalisé une large étude comparative des détecteurs à SSD et ceux basés sur la proposition de région [114], la plupart des informations de cette partie en sont tirées.

Pour choisir le réseau de détection idéal pour une application donnée, il faut prendre en compte plusieurs critères :

- Le compromis précision/vitesse d'exécution : dans [114] les auteurs indiquent par exemple que les systèmes les plus précis de détection d'objets sont basés sur une architecture Faster RCNN, alors que les plus rapides sont de type SSD. Le choix du réseau dépend donc fortement du cahier des charges et des besoins de l'application visée.
- La taille des objets/visages à détecter : les systèmes type SSD sont moins performants pour de petits objets que les systèmes à RPN. Pour les objets de grande taille, les détecteurs one-shot ont une précision similaire aux détecteurs à 2 étapes pour une vitesse d'exécution supérieure. Pour des détections d'objets plus grand en temps réel, préférez les systèmes basés sur du SSD/YOLO.
- D'autres paramètres secondaires entrent en jeu : la résolution des images entrantes aide à la précision du système mais le ralentit considérablement. L'hyperparamètre du nombre de propositions générés dans les réseaux type RCNN peut aussi avoir un large impact sur le temps d'exécution pour une précision quasi-similaire.

Bien que d'architectures différentes, ces deux familles utilisent des concepts communs, que nous pouvons résumer ainsi :

- « Non-max suppression » : utilisé pour supprimer la plupart des boites englobantes qui se superposent et pour garder une seule hypothèse solide par objet
- « Data augmentation » : augmentation artificiellement la taille des bases de données d'entraînement en réalisant des transformations simples des images (rotation, réflexion, rognage, *etc.*)
- Normalisation : la normalisation des lots d'entrainement permet d'assurer une meilleure stabilité du réseau
- Les innovations apportées par AlexNet : ReLU, dropout, *etc.*

Concernant plus particulièrement les réseaux de détection de visages, les deux familles de réseaux coexistent de façon équitable au niveau de la précision atteinte pour l'état de l'art. La tendance des recherches actuelles est cependant à l'avantage des détecteurs "one-shot" avec notamment l'apparition des extracteurs de caractéristiques à pyramide, comme PyramidBox [115].

5.4 Conclusion

Parmi tous les algorithmes de détection, la modélisation de fond est la solution généralement utilisée en vidéo protection. Son succès vient de sa relative facilité de mise en œuvre mais surtout de son faible temps de calcul. Ceci permet d'embarquer l'algorithme, dans la caméra, au plus près du capteur ou de réaliser plusieurs analyses simultanées et en temps réel sur un même serveur.

Afin de lisser le bruit de détection et améliorer la définition des contours de l'objet, plusieurs opérations de bas niveau peuvent être appliquées sur le masque de mouvements. Les plus communes sont les opérations de morphologie mathématique qui sont généralement utilisées pour supprimer les pixels isolés et remplir les trous. La prise en compte de la perspective peut également être utilisée pour ajuster les seuils de détection ainsi que la taille des masques de convolutions.

Par ailleurs, des traitements peuvent aussi être appliqués en amont de la modélisation. Par exemple, il est possible de détecter le changement de gain de la caméra ou tout changement brusque de la luminosité avant de mettre à jour le modèle de fond. Il en va de même pour les vibrations qui peuvent être détectées et éventuellement atténuées par un recalage d'images.

La modélisation de fond est une étape très importante dans une application de vidéo protection parce qu'il s'agit de l'étape qui permet la détection. Si l'objet n'est pas détecté à ce niveau, il ne pourra pas l'être par la suite. Finalement, toutes les autres étapes du processus vont être mises en place pour filtrer ce que la modélisation aura détecté de façon à isoler les événements importants.

L'une de ces étapes est le suivi des objets que nous allons aborder plus précisément dans le chapitre suivant.

V - Suivi d’objets d’intérêts mobiles

1 Introduction

Le suivi est, après la détection, un élément indispensable dans la chaîne de traitements aboutissant à l’identification d’une action ou d’un comportement dans une vidéo. Dans sa forme la plus simple, le suivi peut être défini comme le problème de l’estimation de la trajectoire d'un objet dans le plan image lorsque l’objet se déplace dans une scène. En d'autres termes, un algorithme de suivi s’attache à attribuer une étiquette unique à chaque objet suivi dans différentes images d'une vidéo. Dans un contexte de vidéo-protection, la tâche d’un algorithme de suivi consiste donc à associer chaque blob détecté à l’instant t à l'objet approprié suivi à l’instant t-1, de manière à préserver l'identité des objets du monde réel à travers la séquence vidéo (Figure 73). Au cours du processus, l'algorithme doit également créer de nouveaux modèles d'objets et mettre à jour les modèles existants si nécessaire. Pour résumer, nous pouvons ainsi définir le suivi comme : l’estimation à travers le temps de l’état de plusieurs objets en mouvement en utilisant un ensemble d’observations.

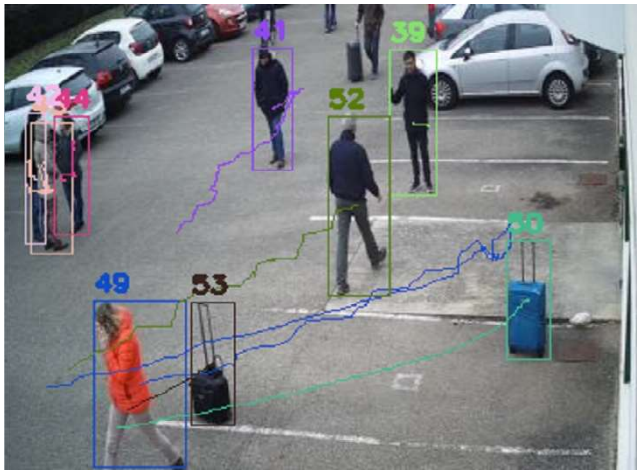


Figure 73 : exemple de résultat d’un algorithme de suivi où chaque objet suivi est affublé d’un identifiant unique. Sur cette illustration les trajectoires et les boites englobantes des objets sont représentées avec la même couleur que l’identifiant.

Au cours des différents travaux de recherche que nous avons étudiés ou encadrés nous avons dû résoudre différentes problématiques de suivi. Dans [19] l’objectif était de détecter des plaques d’immatriculation et de les suivre au cours du temps de façon à renforcer le résultat de la reconnaissance de caractères. Dans [18] le suivi des objets avait pour but de détecter un

comportement de plus haut niveau comme les chutes de personnes. Dans [20], le suivi était à la base de l'extraction des trajectoires permettant de détecter des comportements rares. Dans [116] et dans [10], le suivi était un élément indispensable pour la détection d'une intrusion. Enfin dans les travaux [14], le suivi est indirectement utilisé pour fournir des flux de données au système.

Le suivi est une étape de niveau intermédiaire dans le sens où elle s'appuie sur des caractéristiques calculées à partir des pixels et de leur localisation. Comme nous l'avons présenté précédemment, l'objectif du suivi visuel des objets dans une séquence d'images consiste à mettre à jour le vecteur d'état associé à chaque objet en fonction des observations collectées à chaque frame. Lorsqu'il n'y a qu'un seul objet d'intérêt dans la scène et qu'il est correctement détecté lors des étapes précédentes, cette mise à jour est finalement assez simple. Cependant, à l'instar de la modélisation du fond, l'étape de suivi est confrontée à un certain nombre de difficultés que nous aborderons dans le paragraphe suivant. Il en résulte que l'algorithme de suivi doit composer avec un certain nombre d'incertitudes, telles que l'incertitude liée au bruit des mesures, les fausses alarmes, les détections manquées, l'apparition et la disparition de cibles.

En fonction de l'application visée, il peut être possible de fixer des contraintes sur le mouvement et sur le changement d'apparence des objets suivis. La plupart des algorithmes font l'hypothèse que la trajectoire des objets est relativement lisse sans changement brusque. De même, une calibration même très simple de la scène permet de fixer le gabarit des objets à suivre et de fixer des contraintes sur leur déformation entre deux ou plusieurs images successives. L'ensemble de ces connaissances *a priori* permet de simplifier le problème.

La littérature abonde d'articles sur le suivi, ce qui montre à la fois l'intérêt de la communauté pour cette problématique mais aussi le fait que cela reste encore un domaine ouvert. Parmi les travaux particulièrement remarquables sur l'état de l'art et dans notre domaine d'application, nous pouvons citer l'article de Yilmaz et *al.* [117] ainsi que l'article de Smeulders et *al.* [118]. Ces deux études ne sont bien évidement pas les seules mais elles permettent de se faire une idée assez précise des différentes approches. A noter que dans le cas de l'article de Smeulders et *al.,* les auteurs se limitent au cas des « traqueurs » utilisant des boites englobantes.

A l'instar du chapitre précédent, nous n'allons pas faire ici une étude exhaustive de toutes les méthodes de suivi mais simplement reprendre les quelques grandes approches utilisées traditionnellement dans le cas de la vidéo protection. Notre objectif n'est pas de nous focaliser sur un problème particulier mais bien d'étudier les différents aspects du suivi auxquels nous pouvons être confrontés.

2 Défis

Au même titre que l'étape de détection, un algorithme de suivi efficace doit être en mesure de prendre en compte un ensemble d'éléments susceptibles de perturber son fonctionnement. Ces éléments peuvent être liés au contexte de la scène, à la nature des objets, à leur évolution dans la scène mais également aux performances de la phase de détection. Dans ce dernier cas, la

nature des fausses détections comme des omissions totales ou partiels ne sera pas forcément identique.

2.1 Défis d'ordre général

2D vs 3D : l'une des principales difficultés des algorithmes de suivi est liée à la projection 2D sur le plan image de la scène 3D. Le mouvement apparent de l'objet dans l'image ne peut être représenté qu'avec des degrés de libertés limités. La notion de profondeur est mal estimée avec la projection 2D ce qui engendre un certain nombre d'ambiguïtés.

Occultation : les occultations peuvent être totales ou partielle et de durée plus ou moins longue. Elles peuvent être causées par la présence d'objets connus de la scène (Figure 74) ou liées à la présence d'autre objets en mouvement. Dans le premier cas, l'algorithme peut prendre en compte la présence de ces objets occultants dans le processus de décision. Dans le second cas, cette prise en compte peut être plus délicate du fait de la projection 2D.

Figure 74 : exemple d'occultation partielle d'un véhicule passant derrière un poteau électrique. Le masque de mouvement (à droite) présente deux blobs distincts.

Changement apparent de la taille : ce problème est directement lié à la projection 2D et à l'effet de perspective. Plus les objets s'éloignent de la caméra plus leur taille apparente diminuent dans l'image (Figure 75).

Figure 75 : exemple de changement de la taille apparente d'une personne dans une scène.

Changement d'apparence : au-delà du changement de taille que nous venons d'évoquer, les objets en mouvement peuvent également changer d'apparence en fonction de leur orientation vis-à-vis de la caméra. Un véhicule vu de face n'a pas la même apparence lorsqu'il est vu de profil

(Figure 76). Ce changement d'apparence peut être accentué par la nature déformable ou articulée de l'objet d'intérêt.

Figure 76 : exemple de changement d'apparence d'un objet en fonction de son orientation. Sur cet exemple, nous pouvons également anticiper une difficulté supplémentaire où sur l'image de droite, le véhicule transporte une palette.

Objet entrant dans la scène : Lorsque qu'un objet entre dans la scène, il arrive fréquemment que sur les toutes premières images, il ne soit pas vu en totalité. Cela est déjà un problème en soit lié à l'apparence comme nous l'avons déjà indiqué. Mais plus encore, il arrive que l'objet apparaisse en plusieurs sous-blobs (Figure 77). Nous avons alors deux ou plusieurs étiquettes pour le même objet qu'il faudra ensuite fusionner lorsque l'objet apparaitra en totalité ou tout au moins en un seul blob.

Figure 77 : exemple d'objet entrant dans la scène. Sur l'image de gauche, deux objets distincts semblent entrer dans la scène alors que sur l'image de droite, ils correspondent à deux parties d'un même objet.

Qualité de la détection : comme nous l'avons vu dans le chapitre précédent, la détection des objets d'intérêt n'est pas une tâche aisée et plusieurs problèmes peuvent venir dégrader la qualité de la détection voire l'empêcher temporairement. Un algorithme de suivi robuste doit donc tenir compte de cette contrainte. Les défauts liés à la détection dépendent bien évidement du type d'algorithme utilisé pour cette détection.

Faible résolution de la cible : lorsque la projection de la cible est très petite, généralement à cause de l'éloignement de la cible par rapport à la caméra, le nombre de pixels représentant la cible est très faible. Ce nombre restreint de pixels peut dégrader les performances des mesures de similarité basées sur la forme ou sur des caractéristiques colorimétriques.

Mouvement rapide : le déplacement rapide d'un objet dans la scène, dont l'effet est amplifié lorsque l'objet est relativement proche de la caméra, impose de prendre en compte un espace de recherche plus grand. Il est en effet d'usage, dans la plupart des algorithmes de suivi, de limiter le voisinage de recherche d'un objet. Outre un gain en temps de calcul, cette pratique

permet de limiter les faux appariements. Dans certain cas, nous le verrons par la suite, un recouvrement des rectangles englobants de la cible entre deux images successives peut être imposé.

(a)

(b)

Figure 78 : exemple du déplacement rapide d'une personne sur deux images successives. Sur la séquence (a), le déplacement du centre de gravité de la personne est de 57 pixels. Sur la séquence (b), le déplacement du centre de gravité est de 25 pixels.

Flou de mouvement : le flou de mouvement dégrade la qualité visuelle de l'objet. Ce flou lié au mouvement se manifeste particulièrement sur les caméras dites entrelacées (Figure 79). L'extraction de caractéristiques visuelles peut être perturbée.

Figure 79 : exemple d'effet de flou lié au déplacement rapide d'un objet. Sur cette image, le piéton et le fond de la scène sont net alors qu'un effet de « peigne » caractéristique des caméras « entrelacées » se manifeste sur le motocycliste.

Variation de l'illumination de l'objet : un même objet peut être illuminé différemment en fonction de sa position dans l'image. Par exemple, au cours de son déplacement, il peut passer de l'ombre à une zone éclairée (Figure 80).

Figure 80 : exemple de changement d'exposition lumineuse d'une personne sur images d'une même séquence.

Nous pouvons également ajouter dans cette liste, le bruit de l'image et les erreurs de segmentation ou de détection qui dépendent du modèle utilisé.

2.2 Défis imposés par la modélisation du fond

Blobs parasites : les blobs parasites concernent globalement tous les blobs qui ont été proposés par l'algorithme de modélisation et qui ne correspondent à aucun objet d'intérêt. Nous allons principalement retrouver ici tous les problèmes classiques liés aux changements de luminosité, aux « fantômes », aux mouvements de la végétation dus au vent, intempéries, *etc.*

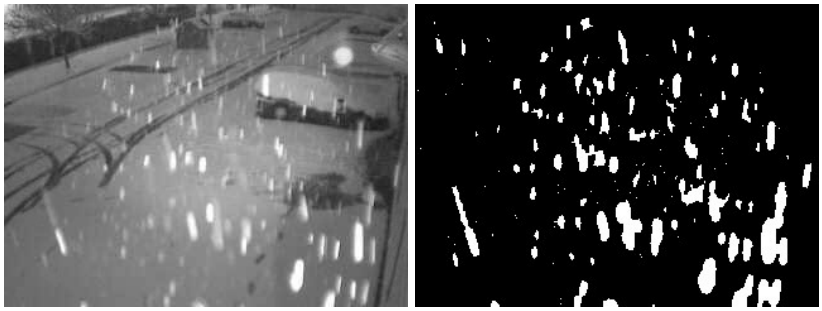


Figure 81 : exemple de blobs parasites liés à la neige et générés par le modèle de fond. Sur le devant de la scène les blobs peuvent être facilement filtrés parce que trop petits par rapport au gabarit d'un personne. Par contre sur le fond de la scène certains blobs ont une taille et même une forme compatible avec une personne.

Non détection : l'objet d'intérêt peut ne pas être détecté, soit à cause du problème de camouflage que nous avons déjà évoqué, soit parce qu'il stationne dans la scène et qu'il est intégré dans le modèle de fond.

Blobs divisés : ce cas se retrouve bien plus souvent que la non-détection. Il est plus ou moins lié aux mêmes phénomènes. La conséquence est que l'objet de la scène est partiellement segmenté en plusieurs blobs.

Fusion d'objets : ce problème de fusion d'objets apparaît lorsque des objets sont très proches les uns des autres dans l'image ou qu'un objet en occulte partiellement un autre et qu'ils ne forment qu'un seul blob. Ils peuvent être éloignés de plusieurs mètres dans la scène, mais leur projection dans le plan image fait qu'ils ne forment qu'un ensemble de pixels connexes en mouvement.

2.3 Défis en lien avec l'utilisation de détecteurs spécialisés

Dans le cas des détecteurs spécialisés le problème est souvent plus binaire et nous pouvons nous limiter aux deux types de problèmes suivants.

Non-détection : à l'instar de la segmentation basée sur un modèle de fond, la non-détection d'un objet à partir d'un détecteur spécialisé peut avoir les mêmes causes liées aux occultations totale ou partielle, au camouflage ou autre. Cependant, une particularité des détecteurs spécialisés est qu'il arrive également que l'objet ne soit pas détecté alors qu'il est parfaitement visible dans l'image. L'effet « boite noire » des détecteurs spécialisés fait qu'il est très difficile de comprendre les raisons de la non-détection. C'est une vraie problématique parce que sur l'ensemble d'une séquence donnée, un même objet ne sera peut-être pas correctement détecté sur l'ensemble des frames alors qu'il apparaît pourtant en totalité.

Classe erronée : ce problème apparait lorsque le classifieur détecte correctement un objet mais commet une erreur de classification.

Fusion d'objet avec ou sans changement de classe : ce problème apparait lorsque deux ou plusieurs objets sont très proches les uns des autres et que le classiffieur n'est plus en mesure de les séparer et les regroupe sous une seule étiquette. Par exemple, un ensemble de personnes qui peuvent être détectées indépendamment comme « personne » lorsqu'ils sont suffisamment éloignés les uns des autres et qui sont étiquetés tous ensemble sous le même label « groupe de personne » lorsqu'ils se regroupent.

3 Le suivi d'objets

3.1 Classification des méthodes de suivi

La littérature distingue deux grandes catégories de méthodes de suivi que nous allons retrouver sous les acronymes anglais VOT pour « Visual Object Tracking » et MOT pour « Multi Object Tracking ». La première catégorie regroupe les algorithmes dont l'objectif est de suivre une cible unique en faisant bien souvent l'hypothèse que le vecteur d'état de l'objet à suivre est connu à l'instant t_0 et ne nécessite pas de détecteur (segmentation ou détecteur spécialisé). Le processus de suivi consiste alors à rechercher la région de l'image qui minimise la distance avec la signature de la cible. La deuxième catégorie est plus générale et regroupe finalement tous les autres algorithmes qui ne rentrent pas dans la première catégorie. Toutefois, elle intègre essentiellement les méthodes faisant référence à l'association de données. Dans le cadre de nos travaux, nous nous sommes particulièrement intéressés au problème du suivi multi objet puisque, avec une étape préalable de modélisation/segmentation de l'avant-plan, nous avons à notre disposition un ensemble de blobs détectés à l'instant t. Cependant, le détecteur peut ne pas être parfait et des objets suivis à l'instant t-1 peuvent ne pas avoir de correspondant dans l'ensemble des blobs à l'instant t. Pour ces objets « gelés » (nous reviendrons sur ce terme par

la suite), une stratégie de type « VOT » peut leur être appliquée afin de pallier les défauts du détecteur ou s'assurer de leur réelle absence dans la scène.

De nombreuses approches de suivi d'objets dans les vidéos sont décrites dans la littérature et plusieurs auteurs ont proposé une classification de ces différentes méthodes. Yilmaz et *al.* [117], proposent une classification en trois catégories (Figure 82) basée sur le type de descripteurs. Cette étude est un peu ancienne mais elle est intéressante dans le sens où les auteurs présentent différentes approches permettant la description des objets et ne se limitent pas aux boites englobantes. La taxinomie proposée par les auteurs se base justement sur le type de représentation de l'objet : points, primitives géométrique et apparence.

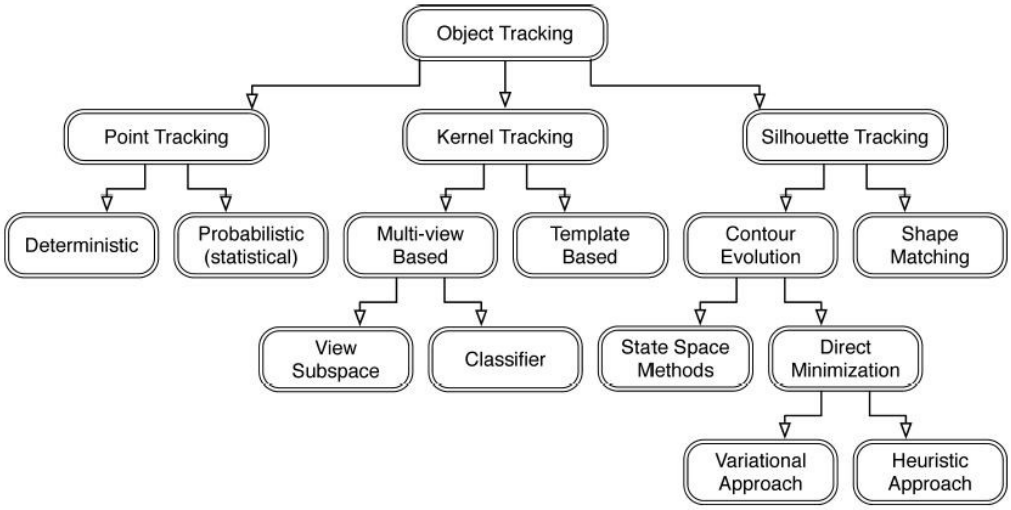


Figure 82 : taxonomie des méthodes de suivis proposée par Yilmaz [117]

Dans le cas du suivi par points, les objets sont représentés par des points et le problème peut être formulé comme un problème de mise en correspondance de points d'intérêts. Cette première approche a été principalement étudiée dans les années 80 et 90 lorsque les capacités de calcul étaient particulièrement limitées.

L'approche par noyaux fait référence à la forme géométrique (gabarit) et à l'apparence de l'objet. Le noyau peut être un rectangle ou une ellipse avec un histogramme associé. Les objets sont suivis en calculant le mouvement du noyau dans des images consécutives. Plusieurs modèles d'apparence ont été proposés dans la littérature comme nous le verrons par la suite. Le suivi par noyau est particulièrement bien adapté lorsque la détection est réalisée par un détecteur spécialisé.

Les méthodes de suivi de silhouettes utilisent les informations encodées à l'intérieur de la région de l'objet. Ces informations peuvent prendre la forme de modèles de densité et d'apparence de forme. Compte tenu des modèles d'objets, les silhouettes sont suivies soit par correspondance de formes soit par évolution de contours. Cette approche est généralement utilisée après une modélisation de fond et une segmentation du masque de mouvement.

L'étude de Smeulders et *al.* [118], plus récente, se concentre sur les méthodes de suivi à base de boites englobantes. Les auteurs ne proposent pas réellement de taxonomie des différents

traqueurs. Leur étude est plus particulièrement orientée sur leur évaluation. Néanmoins, les auteurs les regroupent dans quatre grandes classes en fonction de la complexité du modèle d'apparence utilisé ou de la méthode de recherche. La première catégorie concerne les traqueurs basés sur des modèles d'apparences simples associés à des mesures de similarités, comme par exemple une simple imagette de l'objet et une mesure de corrélation croisée, ou une comparaison d'histogrammes. La deuxième catégorie concerne les modèles d'apparence étendus soit en conservant plusieurs apparences de l'objet au cours du temps, soit en utilisant plusieurs modèles dont des modèles de mouvements ou d'état. Pour ces deux premières catégories, la stratégie de recherche consiste à explorer un voisinage et à ne garder que la position pour laquelle la similarité est la plus grande. La troisième catégorie proposée concerne les traqueurs qui utilisent des modèles d'apparence relativement simples mais pour laquelle la stratégie de recherche est basée sur la mise en commun de plusieurs échantillons épars. Les traqueurs de cette catégorie sont essentiellement dérivés du traditionnel filtre à particules. Enfin la quatrième catégorie proposée par les auteurs regroupe les approches basées sur l'apprentissage d'un classificateur de la cible par rapport à l'arrière-plan. Par exemple, dans cette catégorie, les auteurs incluent le traqueur proposé par Nguyen et *al*. [119] où un classificateur discriminant linéaire est formé à partir d'un vecteur de caractéristiques de texture de Gabor de la région cible en s'appuyant sur un vecteur de caractéristiques dérivés du fond local entourant la cible.

Dans [120] les auteurs proposent également une taxonomie (Figure 83) assez complète des méthodes récentes de suivi intégrant également des approches basée sur l'apprentissage profond.

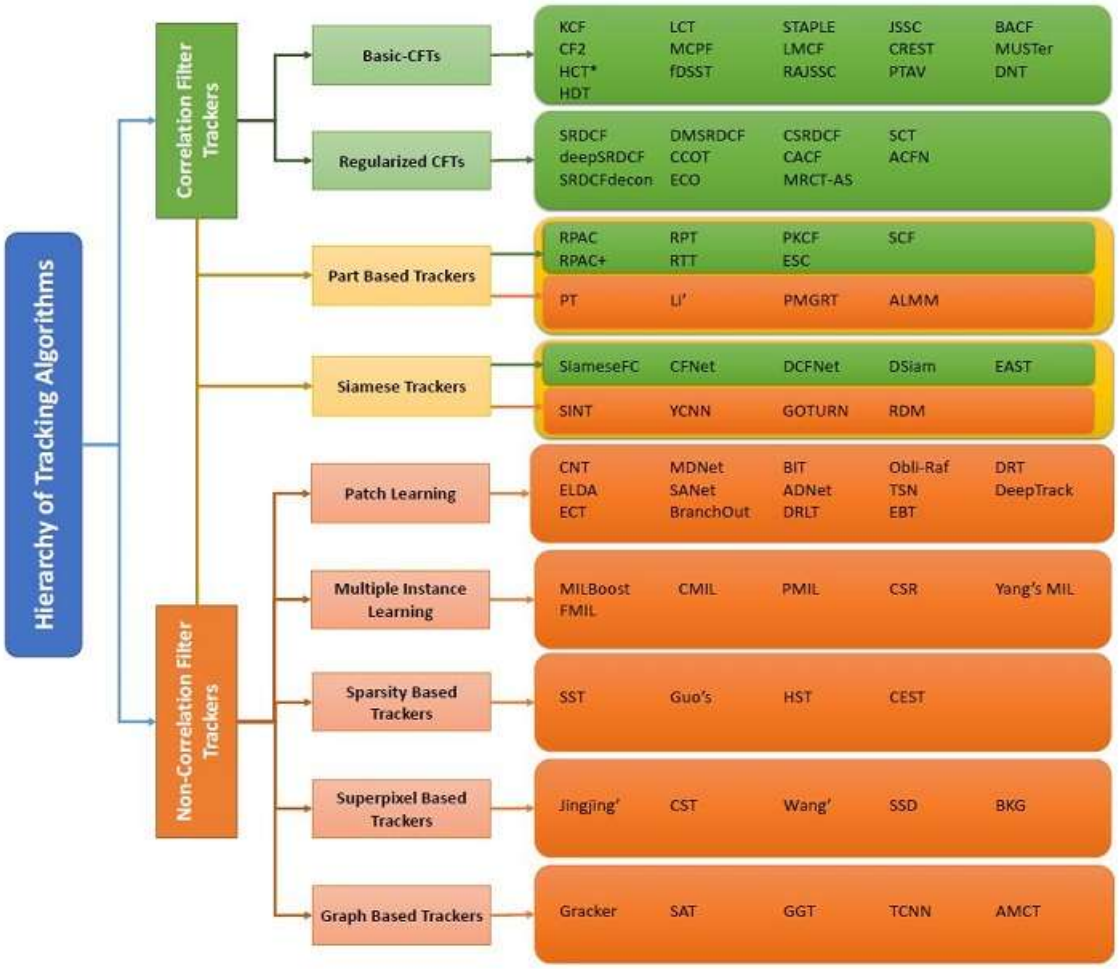


Figure 83 : taxonomie des méthodes de suivi proposée par Fiaz et al dans [120]

Nous pouvons également classer les différentes méthodes en simplement deux catégories générales. La première faisant référence aux approches où une détection préalable est disponible à chaque itération, soit à travers un détecteur spécialisé, soit via une modélisation de fond et une segmentation. La seconde regroupant les approches où seul le vecteur d'état de l'objet est donné pour la première image de la séquence.

3.2 Processus général

La façon d'appréhender le suivi dépend fortement des contraintes liées à l'application visée. L'essentiel de nos travaux est principalement axé sur la détection d'intrusions en temps réel. Une première contrainte à laquelle nous devons faire face et que nous ne pouvons pas utiliser l'ensemble de la séquence où de la vidéo mais uniquement quelques images précédent l'instant t. La deuxième contrainte est que nous ne pouvons pas déterminer le nombre de cibles à suivre. Toutefois, comme nous utilisons une modélisation de l'arrière-plan et une segmentation pour extraire les blobs de l'avant- plan, nous avons une plus grande liberté sur le vecteur d'état de l'objet et ne sommes pas limités aux seules boîtes englobantes.

Dans ces conditions le processus général itératif du suivi se décompose à chaque nouvelle image de la façon suivante :

- **Prédiction / propagation :** cette première étape se base sur l'évolution estimée du vecteur d'état de l'objet à partir des observations précédentes. Ceci permet de limiter l'espace de recherche.
- **Mise en correspondance :** cette deuxième étape consiste à évaluer la correspondance entre le vecteur d'état estimé dans l'espace de recherche et à le mettre à jour en fonction des observations.

Dans le cas particulier où nous disposons d'un ensemble de cible à l'instant t-1 et d'un ensemble de détections à l'instant t, le suivi se transforme alors en un problème d'affectation. Le but est alors pour chaque objet O_i ($i \in [1 ; N]$) dont l'état est connu à l'instant t-1, de trouver la correspondance avec un ensemble de blobs B_j ($j \in [1 ; M]$) détectés à l'instant t. Nous pouvons déjà remarquer une limite de ce type d'approches. Si la distance entre les vecteurs d'état des deux candidats est trop importante, il y a un risque que la mise en correspondance échoue et donc un risque de décrochement irréversible du processus de suivi. Ce principe est cependant majoritairement repris dans le cas de la détection d'intrusion basée sur la modélisation d'une image de fond du fait que les ensembles d'objets suivis et de blobs détectés comportent relativement peu d'éléments et donc que l'espace de recherche est réduit.

Afin de mettre en place un algorithme de suivi efficace, il convient de définir précisément les éléments suivants :

- **Un modèle d'état** qui correspond à la représentation des objets suivi. Ce modèle d'état décrit l'objet à partir d'informations issues de l'image mais peut également inclure une composante cinématique permettant d'introduire la vitesse et éventuellement l'accélération.
- **Un modèle dynamique** qui permet de caractériser principalement le déplacement de l'objet afin de limiter l'espace de recherche mais qui peut également prendre en compte l'évolution du vecteur d'état dans son ensemble.
- **Une mesure de distance** qui permet d'évaluer la mise en correspondance des objets suivis.
- **Une stratégie d'estimation** qui permet, à partir des observations et du modèle dynamique d'inférer les états futurs.

Les deux premiers éléments font référence à la phase de prédiction/propagation et les deux suivantes à la phase de mise en correspondance.

La mise à jour du modèle d'état est un élément important dont il faudra bien souvent tenir compte pour obtenir un suivi robuste. Comme précisé dans [121], il y a en général deux types de modifications de l'apparence d'un objet suivi : intrinsèque et extrinsèque. La pose et/ou la déformation de la forme de l'objet sont considérées comme des modifications intrinsèques alors que les changements d'éclairage ou de luminosité, le mouvement de la caméra, le changement de point de vue et les occultations sont à classer dans les modifications extrinsèques.

Les différents algorithmes de suivi, qui traitent de ce problème particulier, que nous allons retrouver dans la littérature vont principalement se différencier sur le choix du modèle d'apparence, de la stratégie de mise à jour/propagation et la stratégie d'association de données.

3.3 Les modèles d'état

Le modèle d'état est un élément important de la chaine de traitement et son choix est directement dicté par l'application et les capacités de calcul du matériel sur lequel il va être déployé. Le vecteur d'état désigne toutes les caractéristiques de couleur, de forme, de texture et plus généralement tout traitement mathématique calculé sur une région de l'image et permettant de définir la signature d'un objet. Cette signature doit être la plus discriminante possible et la moins gourmande en temps de calcul.

Sur ce dernier point, nous pouvons toutefois nuancer un peu le propos. Dans un contexte de calcul en temps réel sur une multitude de flux vidéo, nous avons largement évoqué les nécessités d'optimiser et de limiter l'utilisation des ressources de calcul. Toutefois, pour l'étape de suivi, nous nous plaçons dans un contexte d'association de données où les deux ensembles à analyser ont relativement peu d'éléments chacun. Le nombre de comparaisons est donc relativement limité et peut encore être réduit par un pré-filtrage sur la position comme nous le verrons dans le paragraphe suivant. En conséquence, nous pouvons relâcher un peu la contrainte sur les ressources de calcul dans la mesure où cela permet une meilleure spécificité.

3.3.1 Approche pixel

Cette première catégorie permet de construire la signature de l'objet directement à partir des pixels correspondant à la zone d'intérêt. Cette signature peut ainsi prendre la forme d'un simple vecteur et une approche naïve pour comparer deux signatures consiste à calculer une norme L2 classique. Cette approche naïve n'est en général pas satisfaisante parce qu'elle nécessite des vecteurs de même dimension mais surtout elle n'est pas robuste aux changements de luminosité. Pour pallier ce dernier défaut, une approche encore relativement simple est de changer d'espace de couleurs et d'utiliser un espace du type HSV qui peut être plus robuste que l'espace RGB. Afin d'améliorer encore la robustesse, Silvera et *al*. [122] proposent un alignement colorimétrique des images invariant au changement d'illumination. L'intérêt de leur approche est que le calcul d'alignement se fait une seule fois sur l'ensemble des pixels de l'image ce qui permet ensuite de gérer la recherche de plusieurs cibles sans appliquer la transformation sur chaque cible.

Une approche un peu différente est de ne pas utiliser directement les pixels bruts mais les gradients qui sont réputés plus robustes aux changements d'illuminations. Il y plusieurs façons d'exploiter les gradients et nous pouvons citer par exemple les LBP (Local Binary Pattern) utilisés par Alismail et *al.* [123] ou les Histogrammes de Gradient Orienté (HOG : histogram of oriented gradients). L'introduction des HOG par Dalal et *al.* [98] a amélioré significativement les détecteurs de personnes. Cette caractéristique est calculée sur un ensemble de cellules carrées de petite taille (8x8 pixels), où chaque pixel de la cellule vote pour une orientation en fonction de l'orientation du gradient en ce point. Ce vote est pondéré par l'intensité du gradient du point considéré. Le vecteur de caractéristiques est formé par la concaténation des valeurs de chaque bloc après normalisation afin de rendre le descripteur plus robuste aux changements d'illumination. Depuis leur introduction en 2005 pour la détection de personnes, les HOG ont largement été utilisés dans d'autres domaines et notamment dans le cas du suivi [124]

Ces quelques premiers exemples utilisent la même idée simple pour la mise à jour de la cible. En effet, après l'étape de mise en correspondance ou de recherche suivant le type d'algorithme (respectivement MOT ou VOT), le vecteur d'état de l'objet (hors caractéristiques cinématiques) est remplacé par le vecteur d'état du blob correspondant. Cette approche simple peut souffrir de certaines limitations lorsque la localisation de l'objet est imparfaite ou que celui-ci change d'apparence quelle qu'en soit la raison. Pour pallier ce défaut, une solution proposée est de mettre à jour graduellement le vecteur d'état en gardant un certain historique. A partir d'un ensemble d'observations d'un même objet, Li et *al*. [121] construisent un tenseur puis calculent un sous espace propre. Dans cette approche, les auteurs considèrent l'ensemble des pixels bruts d'une région rectangulaire comme une simple matrice de valeur qu'ils empilent dans un tenseur. Le sous-espace vectoriel constitue alors la signature de l'objet. Une approche similaire avait déjà été proposée par Black et *al.* [125].

Une autre approche de modèle d'apparence dynamique consiste à utiliser un gabarit ou « Template ». Cette stratégie a eu un certain succès dans les années 2000 pour le suivi des personnes. Dans ce type d'approche, le modèle d'apparence est appris « à la volée ». Cette solution est d'autant plus efficace lorsque la détection des objets repose sur une modélisation du fond et une segmentation. A partir des travaux de Davis et Bodick [126], Haritaoglu et *al.* [127] ont proposé un modèle d'apparence appliqué au suivi humain. Le modèle d'apparence est constitué en fait de deux gabarits. Le premier est relatif à la forme, et représente la probabilité qu'un pixel lui correspondant fasse effectivement partie de l'objet suivi. Ceci permet de pondérer les régions où une partie de l'objet a été fréquemment détectée, ce qui a pour intérêt de représenter la forme d'une manière simple mais efficace. Le second gabarit est corrélé au premier et capture des informations d'intensité lumineuse (cf. Figure 84). Sur cette version proposée par Haritaoglu, le modèle d'apparence n'incorpore pas l'information de couleur. Une extension dans ce sens a été proposé par Zhao et *al.* [128].

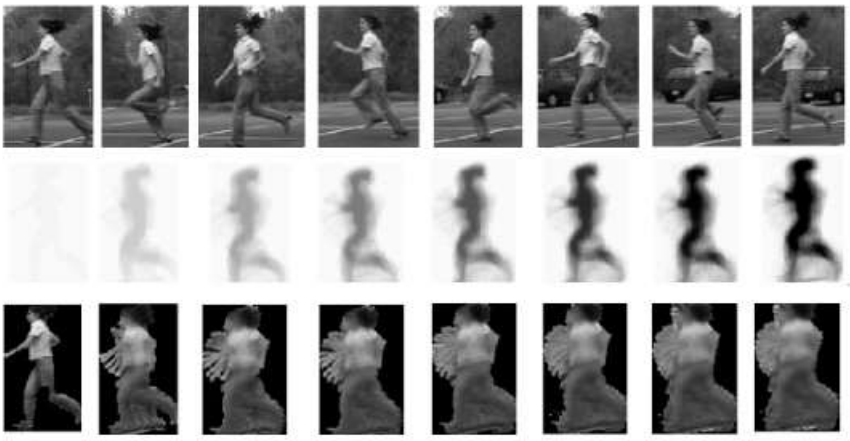


Figure 84 : exemple de mise à jour du gabarit d'après Haritaoglu et al. [127]. *La ligne du haut correspond aux images de la séquence vidéo, la ligne du milieu correspond au gabarit relatif à la forme (plus le pixel est sombre, plus son poids est important) et la ligne du bas correspond au gabarit d'intensité.*

Dans [18], nous avons également proposé un modèle d'apparence basé sur l'utilisation de gabarit et adapté au suivi individuel de personnes dans les séquences d'images. Notre approche consiste à détecter et étiqueter les différentes parties visibles du corps de chaque de chaque

personne à partir des blobs extraits d'une modélisation du fond (Figure 85). Les régions segmentées en mouvement sont mises en correspondance au cours du temps, afin de former un ensemble de liens basés sur des considérations géométriques simples.

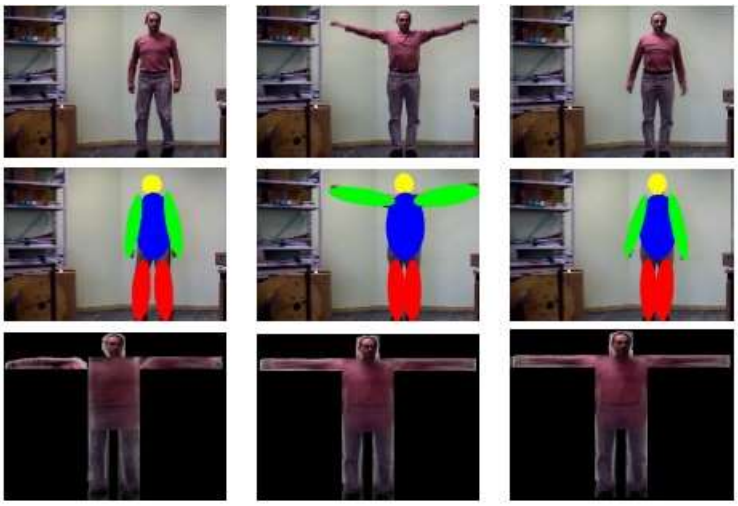


Figure 85 : modèle d'apparence articulé proposé dans [18]*. La ligne du haut correspond aux images de la séquence. La ligne du milieu correspond à la détection des différents éléments du modèle. La ligne du bas correspond au modèle d'apparence après sa mise à jour.*

Cette étape est menée à bien par une technique de mise en correspondance entre un graphe calculé à partir du squelette 2D image et un modèle 3D du squelette humain. L'ensemble des candidats pour les membres est identifié en introduisant une information *a priori* qui fixe des contraintes d'assemblement et utilise uniquement l'information morphologique et topologique de la silhouette. Nous l'avons appliqué au cas des personnes mais elle est transposable à un type d'objet articulé quelconque. Nous avons également proposé une stratégie qui consiste à tirer profit des situations où l'association des régions est suffisante pour effectuer un suivi non ambigu afin de chercher à générer et mettre à jour un modèle d'apparence qui est ensuite utilisé dans des conditions difficiles (cas des occultations par exemple). Une fois la silhouette correspondant à la personne suivie extraite, nous formons un ensemble de segments qui constituent les candidats pour les différents membres à identifier (cf. Figure 85). La méthode proposée n'utilisant que l'information relative à la forme de la silhouette et à la topologie des segments, elle est donc applicable dans un ensemble de configurations très génériques. En particulier, la mise en correspondance ne dépend pas de la pose de la personne, du point de vue, de la géométrie ou de l'apparence des membres. Elle est ainsi, entre autres, invariante à la taille de la personne dans l'image (Figure 86). Une fois chaque membre identifié dans l'image, le modèle d'apparence qui lui est associé est mis à jour. La caractéristique formée permet à la fois de capturer une information discriminante et robuste. L'utilisation combinée du suivi de régions et du modèle d'apparence permet de suivre un nombre indéterminé de personnes dans la vidéo, de les identifier et d'interpréter leurs interactions.

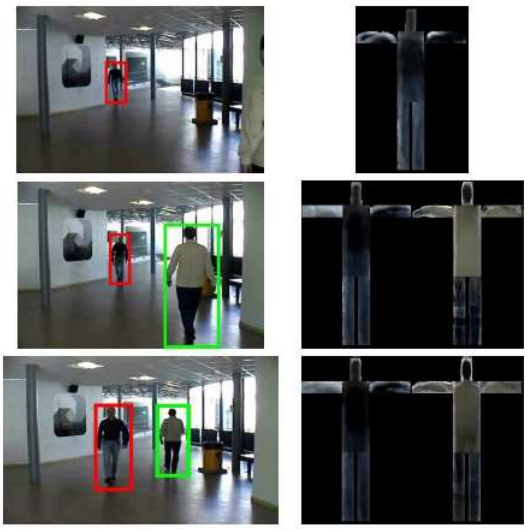


Figure 86 : exemple de l'invariance du modèle à la taille des personnes dans une séquence d'image.

Dans le cadre des travaux de Rogez [10] nous avons également utilisé le modèle d'apparence proposé par Ilyas et *al.* [129]. Le vecteur d'état proposé par les auteurs est obtenu en calculant la trace couleur selon un axe prédéfini. Dans le cas des personnes, la direction privilégiée est celle de l'axe vertical. Le vecteur d'état est construit à partir de l'imagette comportant l'objet et du masque binaire de mouvements (Figure 87) à l'instar de ce qui est proposé par Haritaoglu et *al.* [127]. La hauteur de l'imagette est normalisée à une valeur h afin que les vecteurs d'état aient tous la même dimension quelle que soit la taille apparente de l'objet dans l'image. Le vecteur est ensuite calculé en intégrant les couleurs sur l'axe horizontal en ne tenant compte que des pixels de l'avant plan. Le calcul peut être fait en utilisant la moyenne, la médiane ou en gardant la valeur la plus fréquente d'une densité de probabilités. Les auteurs proposent une transformation de l'espace des couleurs où chaque canal est centré sur la valeur moyenne et normalisé par l'écart type afin d'obtenir un vecteur d'état robuste aux changements de luminosité. Dans le cadre de notre implémentation, nous avons opté pour un espace de couleur R_cG_c plus simple à obtenir. Les valeurs R_c et G_c sont directement calculées à partir des valeurs chromatiques RGB du pixel considéré, à partir des formules suivantes :

$$R_c = \frac{R}{R+G+B} \text{ et } G_c = \frac{G}{R+G+B}$$

Ainsi, les valeurs de R_c et G_c représentent la part de rouge et de vert dans la couleur d'origine indépendamment de la luminance. L'espace chromatique ainsi défini introduit quelques pertes (il n'est pas possible de revenir à l'espace RGB complet d'origine). Cependant, l'espace n'inclut pas les informations sur la luminance ce qui le rend plus robuste aux changements de luminosité et réduit la dimension du vecteur d'état.

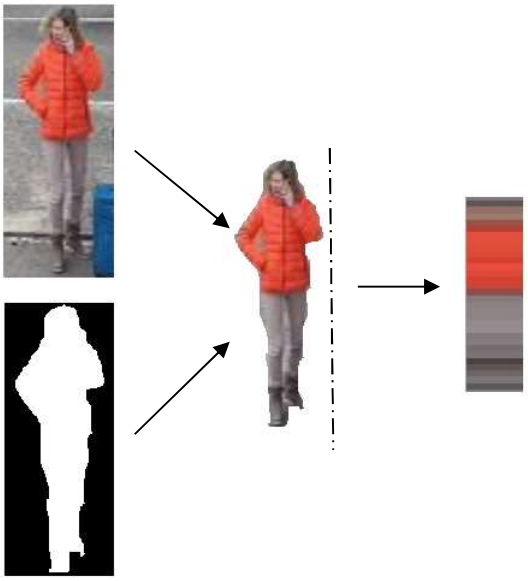


Figure 87 : illustration de la construction du vecteur d'état proposé par Ilyas et al. [129]

3.3.2 Approche par densité de probabilités

Une autre grande famille de descripteurs très largement utilisée dans la tache de suivi est basée sur la comparaison de densité de probabilités de caractéristiques de couleurs (Figure 88). Les histogrammes de couleurs sont en effet largement utilisés en raison de leur robustesse et de leur efficacité à capturer des caractéristiques visuelles à l'intérieur d'une région. De façon générale et par défaut, un histogramme de couleurs est calculé à partir d'une partition en plusieurs classes d'intervalles égaux. Toutefois, il est possible d'utiliser des intervalles de longueurs différentes qui tiennent compte de la distribution des couleurs. Un histogramme de couleurs se définit comme $H = \{h_i\}$ avec $i \in [1, n]$ où n est le nombre de classe de l'histogramme. Un intérêt non négligeable est que, dans le cas de région rectangulaire, le calcul des histogrammes de couleur peut être optimisé [130] en utilisant des images intégrales [131] associées à de simples fonctions de seuil.

Une première solution minimaliste consiste à utiliser un simple histogramme de couleurs calculé sur l'ensemble de la région considérée. C'est la solution proposée par Bradski et *al*. [132] qui utilisent un histogramme de couleurs dans l'espace HSV calculé sur une région rectangulaire.

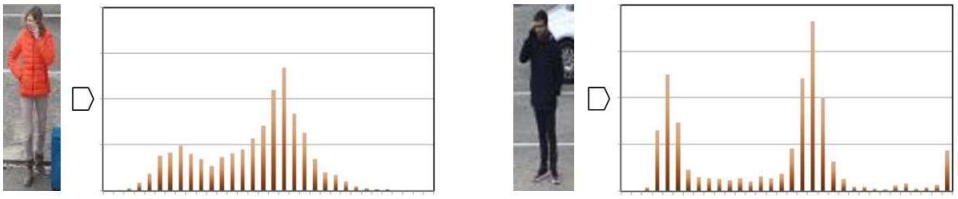
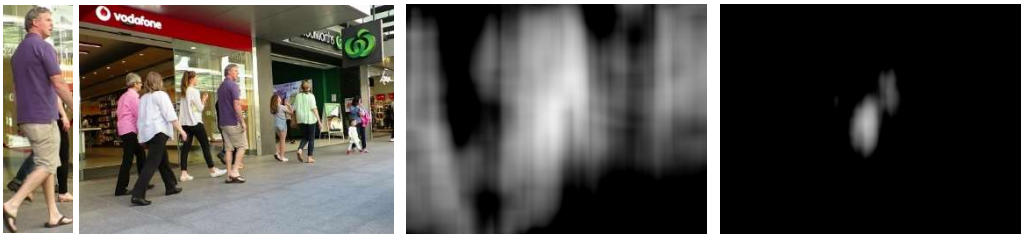


Figure 88 : exemple de représentation d'un histogramme en niveau de gris de deux personnes

Si l'histogramme d'une région rectangulaire est simple à estimer et peut facilement être optimisé, il n'en reste pas moins qu'il n'est pas forcement adapté à tous les objets. Il peut introduire dans la signature des éléments du fond de la scène qui ne sont pas représentatifs de l'objet. Ce biais dans la signature peut avoir plusieurs conséquences malheureuses lors de la

mise en correspondance. Si deux objets ont intrinsèquement des signatures relativement similaires, le poids du fond de la scène capturée dans l'histogramme peut être suffisant pour provoquer une inversion dans la mise en correspondance. De même si le fond de la scène sur lequel l'objet se projette change radicalement, le suivi peut également échouer. Afin de limiter le poids du fond de la scène dans le calcul de l'histogramme, Birchfield et *al.* [133] proposent de délimiter la région d'intérêt par une ellipse. Comaniciu et *al*. [134] étendent ce modèle ellipsoïde en pondérant la couleur d'un pixel par rapport à sa distance au centre de l'objet. Plus précisément le modèle proposé favorise les pixels proches du centre de l'ellipse par rapport à ceux situés à proximité de la périphérie. Cette solution sera reprise par plusieurs auteurs [135][136][137][138][139][140]. Une autre solution consiste à reprendre le principe du masque de mouvement proposé par Haritaoglu et *al*. [127].

Les histogrammes de couleurs sont simples à mettre en œuvre et permettent relativement bien de caractériser un objet mais leur principale faiblesse est que l'information spatiale est perdue lors du calcul. L'approche de Comaniciu et *al*. [134] utilise cette information spatiale lors de l'estimation de l'histogramme mais simplement pour donner moins de poids aux pixels situés à la périphérie de la région considérée et donc moins susceptible d'appartenir à l'objet. Leur approche ne prend notamment pas en compte l'orientation. Intuitivement nous pouvons supposer que le fait de garder une trace d'une disposition spatiale des couleurs, même grossière, peut être bénéfique pour améliorer les performances d'un algorithme de suivi. Une solution simple, reprise par plusieurs auteurs [141][130][138] [142] consiste à diviser la région suivie en sous-régions et de calculer la distribution des couleurs pour chaque sous-région. La signature de l'objet est alors simplement l'agrégation des différents histogrammes. C'est cette stratégie que nous avons utilisée dans la thèse de Matthieu Rogez [10] pour le suivi des piétons où la boite englobante des objets suivis est coupée en deux dans le sens de la hauteur. Deux histogrammes sont alors calculés représentant pour l'un le haut du corps et pour l'autre, le bas du corps. L'intérêt de cette approche est illustré sur l'exemple suivant (Figure 89). Pour construire cet exemple, nous avons utilisé la vidéo MOT17-09 [143] et nous avons arbitrairement sélectionné une cible (a) sur l'image 235 de la vidéo et nous recherchons cette cible dans l'image 237 (b) de cette même vidéo. Nous faisons glisser une fenêtre rectangulaire de la taille de la cible sur l'ensemble de l'image. A chaque position, nous extrayons la signature et nous calculons la similarité de cette signature et la signature de la cible. Ceci nous permet de calculer une carte de similarité où l'intensité du pixel est proportionnelle à la similarité et représente la probabilité que le centre de la cible se trouve à cette position. L'image (c) correspond à une signature avec un seul histogramme alors que l'image (d) correspond à une signature divisée en deux histogrammes, un pour la partie haute de la cible et un pour la partie basse. Les parties les plus claires de ces cartes correspondent aux régions où la cible est susceptible de se trouver. Sur ce simple exemple, nous pouvons remarquer que cette zone est beaucoup plus restreinte sur la carte à deux histogrammes (d).



(a) *(b)* *(c)* *(d)*

Figure 89 : illustration de l'efficacité d'une signature à plusieurs histogrammes. (a) représente la cible issue de l'image 235 de la vidéo MOT17-09 [143]. *(b) correspond à l'image 237 de cette même vidéo. (c) est la carte de similarité calculée à partir d'une signature à 1 histogramme. (d) est la carte de similarité calculée à partir d'une signature à deux histogrammes.*

Une fois la stratégie de construction de l'histogramme ou des histogrammes choisie, une autre étape importante est la sélection de la mesure de comparaison. Les caractéristiques de cette mesure de comparaison sont les mêmes quelle que soit finalement l'approche générale choisie. La mesure doit être suffisamment discriminante entre deux signatures de deux objets différents même lorsque la répartition des couleurs est proche. Dans un même temps, il est nécessaire que la mesure soit suffisamment robuste pour assurer la correspondance entre deux signatures bruitées d'un même objet. Il existe plusieurs mesures dans la littérature. Meshgi et *al.* [144] ont proposé une comparaison de plusieurs de ces mesures dans le cadre du suivi d'objet. La plupart des mesures utilisées pour la tâche de suivi sont basées sur une comparaison classe par classe. La plus populaire étant dérivée du coefficient Bhattacharya. Elle est utilisée par [134][137] [130][142][13] [145][140] pour ne citer que quelques auteurs. En règle générale, les algorithmes de suivi recherchent la minimisation d'une distance. De fait la **distance de Bhattacharya** entre deux histogramme H = {h_i} et K = {k_i} avec i ∈ [1, n] s'exprime ainsi :

$$D_B(H,K) = 1 - \sum_i \sqrt{h_i . k_i}$$

La popularité de cette mesure est liée d'une part à sa simplicité mais également au fait que, dans certaines conditions, l'interprétation géométrique du coefficient est simple. En effet, si les histogrammes sont normalisés, le coefficient $(\sum_i \sqrt{h_i . k_i})$ représente le cosinus de l'angle entre les deux vecteurs.

Une autre distance couramment utilisée [133] [146] est l'**Intersection d'histogrammes** proposée par Swain et *al.* [147]. Elle se définit comme suit :

$$D_\cap(H,K) = 1 - \frac{\sum_i \min(h_i, k_i)}{\sum_i k_i}$$

L'intersection d'histogrammes est attrayante en raison de sa capacité à gérer des correspondances lorsque les histogrammes ne sont pas normalisés.

Dans cette catégorie de mesures nous pouvons également citer la **distance Minkowski** qui est également très simple à calculer puisqu'elle se défini comme suit :

$$D_{Lr}(H,K) = \left(\sum_i |h_i - k_i|^r\right)^{1/r}$$

Une limitation importante de ces mesures classe-par-classe est qu'elles font l'hypothèse que le domaine des histogrammes est aligné. En pratique, cette hypothèse n'est pas toujours vérifiée. La situation la plus courante qui viole cette hypothèse est le changement de luminosité qui peut être lié au changement de gain de la caméra ou lorsque que l'objet passe d'une zone d'ombre à une zone plus éclairé ou inversement. Le simple glissement de l'un des histogrammes diminue considérablement les performances des mesures classe-par-classe. Une solution consiste alors à croiser les informations. La **distance du cantonnier** ou du terrassier (EMD pour « **Earth Mover's Distance** ») proposée par Rubner [148] entre dans cette catégorie. Cette distance fait référence au transport optimal, c'est-à-dire qu'elle mesure la quantité de travail minimal nécessaire pour passer d'une distribution à une autre. Cette mesure de distance est utilisée par Adam et *al.* [138].

Une liste plus complète des mesures de distance ou de similarité entre des fonctions de densité de probabilité est proposée par Cha [149].

3.3.3 Autres approches

Les approches pixels et histogrammes de couleurs sont majoritairement utilisées dans le cadre du suivi. La raison de leur succès est qu'elles sont simples à calculer et ne nécessitent pas de ressources de calcul trop importantes. La plupart font cependant l'hypothèse que la luminosité est constante. Dans la pratique, cette hypothèse n'est pas forcément vérifiée sur une séquence longue mais dans le cas d'un algorithme de suivi temps réel avec une fréquence d'acquisition d'au moins 5 images par seconde, les changements de luminosité sont globalement acceptables. Pour que le suivi soit robuste, une stratégie de mise à jour du modèle doit être intégrée.

Il existe toutefois d'autres approches pour résoudre le problème du suivi. Dans les travaux de Nicolas Thome [18], nous avions étudié une approche basée sur le flot optique. L'intérêt de flot optique est qu'il permet de donner une direction et un sens au mouvement des différentes parties des objets et que cela facilite le suivi des objets qui se croisent (Figure 90).

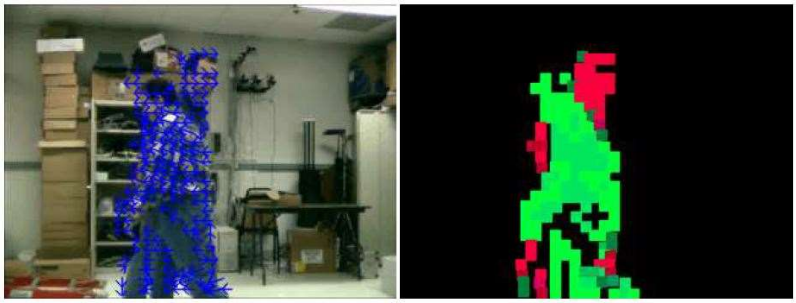


Figure 90 : exemple d'application du flot optique pour la tache de suivi. Sur l'image de gauche, deux personnes se croisent et le flot optique est calculé sur les blobs en mouvement. L'image de gauche représente le masque de mouvement colorisé en fonction de l'orientation du flot optique.

Les méthodes de suivi basées sur le flot optique ont eu un certain succès autour des années 2000 [150] [151] puis elles sont un peu tombées dans l'oubli. Certains auteurs ont toutefois poursuivi dans cette voie [152][153]. Les progrès réalisés dans l'estimation du flot optique à partir des techniques d'apprentissage profond [154][155] relancent l'intérêt pour ce type d'approche dans le cas du suivi.

Une approche encore différente consiste à utiliser des informations obtenues à partir du contour des objets. Cette approche présente l'avantage de pouvoir plus facilement s'adapter au suivi des objets non rigides. La représentation à partir des contours actifs est assez largement utilisée [156][157][158][159]. En règle générale, les auteurs utilisent le modèle, ou une adaptation du modèle, proposé par Kass [160]. Dans ce modèle, l'évolution d'un contour actif est contrainte par une fonction d'énergie ϕ composée de trois termes : un terme d'énergie « image » qui mesure la probabilité que les données images appartiennent à la classe de l'objet de premier plan, un terme de régularisation (tension, rigidité élasticité) et un terme d'énergie interactive qui caractérise le déplacement global du contour actif. La fonction ϕ est une somme pondérée de ces trois termes. Le principe de base est de chercher à minimiser cette fonction d'énergie entre deux images successives.

Les contours actifs ne sont pas la seule approche permettant d'exploiter les informations du contour. Dans [116], la signature des objets est déterminée à partir des points du contours des blobs de premier plan. Une trentaine de caractéristiques sont ainsi calculées dont le périmètre, la surface, la compacité, l'excentricité, la position du centre de masse, les moments normalisés (jusqu'à l'ordre 3), les premiers coefficients de la transformée de Fourier des points du contour, *etc*. Nous avons également ajouté une pondération à ces différents termes de façon à donner plus d'importance à certaines caractéristiques en fonction du contexte d'utilisation. L'intérêt principal de cette approche est qu'elle s'adapte bien au contexte de la vidéo protection. En effet, une fois que les contours sont extraits, les caractéristiques peuvent être calculées rapidement. Si la complexité des calculs peut être plus importante que l'extraction d'un histogramme, le nombre de points à traiter est bien inférieur. Les autres intérêts non négligeables sont que cette approche ne nécessite pas d'ajustement en fonction du type de caméra (couleur, niveau de gris, thermique, *etc*.), que la signature est insensible aux changements de luminosité et qu'elle permet de suivre des objets même avec des résolutions très faibles.

3.4 Recherche et mise en correspondance

La construction de la signature et la mesure de similarité sont deux points importants à prendre en considération pour construire un algorithme de suivi efficace. Cette signature dépend d'un certain nombre de facteurs et de contraintes liées à l'environnement et aux ressources de calculs disponibles. Un autre point très important, qui est soumis aux mêmes contraintes, est la stratégie de recherche de la cible et/ou de la mise en correspondance. Comme nous l'avons indiqué en introduction de ce chapitre, nous pouvons considérer deux grandes catégories d'algorithmes de suivi. Les algorithmes de type VOT où le but est de rechercher une cible, dont la position de départ est connue, dans une image ou dans une succession d'images et les algorithmes de type MOT, qui généralement, essaient de mettre en correspondance des objets connus à l'instant t-1 avec des objets détectés à l'instant t. Dans les deux cas, un modèle

dynamique associé à un filtre de Kalman peut être appliqué à l'objet ou aux objets afin de tenir compte de son évolution et ainsi de limiter l'espace de recherche.

Par ailleurs, une solution complète de suivi n'est pas forcément limitée à une seule de ces deux approches mais peut être une combinaison des deux. Ceci est surtout vrai pour les algorithmes de type MOT. En effet, le processus de détection placé en amont du suivi dans la chaine de traitement peut ne pas être parfait et peut ne pas être en mesure de détecter tous les objets d'intérêts à chaque nouvelle image. Il y a évidemment plusieurs raisons à cela notamment la qualité intrinsèque du détecteur mais également des situations d'occultation totales ou partielles. A l'issue de la phase de mise en correspondance d'un algorithme de type MOT, il peut être intéressant d'associer un algorithme de type VOT sur les objets suivis à l'instant t-1 pour lesquels aucun correspondant n'a été trouvé à l'instant t.

Pour rappel, le cadre de ce travail s'inscrit dans le cas particulier de la détection d'intrusions en ligne, c'est-à-dire que les images de la séquence vidéo sont traitées dans l'ordre d'acquisition et dans l'intervalle entre deux acquisitions. Dans ces conditions particulières et pour simplifier, nous pouvons considérer les approches VOT comme des algorithmes de recherche d'une cible et les algorithmes MOT comme des problèmes d'affectation.

3.4.1 Recherche de cible

Dans cette première catégorie d'algorithmes, nous considérons que la cible est connue, c'est-à-dire que nous disposons de sa signature dans au moins une image précédente. L'objectif est alors de rechercher une signature équivalente dans l'image courante.

Une première alternative est d'utiliser un algorithme de type CONDENSATION (Conditional Density Propagation) proposé initialement par Isard et Blake [161][156] et plus généralement connu sous le nom de filtrage particulaire (Figure 91). C'est un processus itératif composé de trois étapes : mutation, pondération et sélection. Le filtre est tout d'abord initialisé en générant un certain nombre d'échantillons appelés « particules » à partir de la position d'origine. A chaque itération, dans la première étape de mutation, les particules explorent indépendamment l'espace de recherche. Ensuite, une étape de pondération consiste à affecter un poids à chaque particule en fonction de son adéquation avec les observations. Le poids de chaque particule est proportionnel à la mesure de vraisemblance ou de similarité par rapport à la cible. Enfin, l'étape de sélection permet d'éliminer ou de multiplier des particules en fonction de leur degré de vraisemblance. A l'issue du processus itératif, le résultat est une moyenne pondérée des différentes particules. La redistribution des particules est généralement effectuée de manière aveugle mais peut être conditionnée par un noyau gaussien de façon à forcer l'apparition de nouvelles particules dans une région de plus forte vraisemblance ou liée à des contraintes particulières (mouvement, forme, *etc.*).

Figure 91 : illustration du principe du filtre particulaire.

Le filtre particulaire est fréquemment utilisé comme algorithme de base dans le cas du suivi d'objets [141][124][140]. Il peut également être une solution alternative pour les algorithmes basés sur l'association de données afin de retrouver dans l'image un objet qui n'a pas été détecté ou plus exactement qui n'a pas trouvé de correspondant satisfaisant dans la liste des objets détectés.

Dans [18], nous avons adapté un filtre particulaire pour gérer les occultations et plus spécifiquement le cas où des personnes se croisent comme dans le cas de la figure suivante (Figure 92).

(a) suivi de deux personnes

(b) Détection du début de l'occultation

(c) Gestion de l'occultation par filtre particulaire

(d) Fin de l'occultation et reprise du suivi classique

Figure 92 : exemple de gestion des occultations à partir d'un filtre particulaire [18]

En première approximation, l'algorithme de suivi que nous avons développé est basé sur une mise en correspondance d'une liste d'objets suivis à l'instant t-1 et d'une liste de « blobs » détectés à l'instant t à partir du modèle d'apparence articulé que nous avons présenté précédemment. Pour la détection des « blobs », nous utilisons un algorithme de segmentation basé sur une modélisation de l'arrière-plan. En l'absence d'occultation (a), les deux personnes sont détectées et suivies sans faire intervenir un processus additionnel. Lorsque l'une des deux personnes commence à être occultée par une autre (b), l'étape de modélisation du fond/segmentation génère un seul blob regroupant les deux personnes. Dans ce cas, la mise en correspondance échoue, par contre nous avons une intersection forte entre les deux objets suivis et un blob qui n'ont pas trouvé de correspondant. Cette situation déclenche le processus de suivi par filtre particulaire indépendamment sur les deux objets en limitant l'espace de recherche sur la région occupée par le blob (c). Le vecteur d'état correspond au centre de la boîte englobante de l'objet et à son facteur d'échelle. Nous cherchons à estimer la densité de probabilité de ces paramètres, à partir d'une position initialisée juste avant l'occultation. L'étape de mutation comporte deux termes : un terme correspondant à la vitesse estimée de l'objet (nous utilisons un modèle du premier ordre, remis à jour à chaque pas de temps), et un terme de bruit qui propage les particules aléatoirement. Notons un point important dans le cadre de notre application : puisque nous cherchons à estimer la position d'une personne au sein d'une région (qui a fusionné), la position des particules est délimitée par les bords de la région, ce qui contraint la propagation des particules. L'étape de mesure de la similarité basée sur des données image est effectuée en attribuant un poids pour chaque particule, évaluée par une mesure de corrélation entre l'apparence de la particule et le modèle de l'objet. Dans notre cas, nous calculons une distance euclidienne basée sur la couleur pour chaque pixel des régions d'intérêt. Les rectangles fins correspondent à la position des particules propagées pour chaque personne, et le rectangle plus épais représentent l'état moyen du filtre, c'est-à-dire la somme pondérée des particules par leur poids, ce qui constitue l'estimation de la position de la personne. Nous pouvons remarquer que le suivi reste très précis malgré les hypothèses relativement simplistes formulées pour l'utilisation du filtre (suivi d'une zone rectangulaire de taille fixe, ce qui revient à considérer les personnes rigides ; pondération des particules effectuée par une simple mesure de corrélation couleur ; modèle de mouvement à vitesse constante). Ceci illustre les capacités de flexibilité du filtre à particules et notamment sa capacité à estimer des distributions multi modales. Enfin, à la fin de l'occultation (d), lorsque la segmentation « sépare » correctement les deux personnes en deux « blobs » distincts, les personnes sont à nouveau reconnues grâce à la

minimisation des distances entre les modèles d'apparences. Le suivi peut se poursuivre de manière cohérente.

Une autre alternative largement utilisée pour le suivi d'une cible est une approche de type « mean-shift ». La méthode « mean-shift » est un estimateur du gradient de densité non paramétrique développé par Fukunaga et Hostetler [162]. Pour simplifier, le « mean-shift » est un algorithme itératif qui a pour objectif de faire converger un point vers le maximum local le plus proche. Cette convergence vers un mode local a été établie par Comaniciu et Meer [163] pour une large classe de noyaux sur des fenêtres fixées. Les étapes de la convergence sont les suivantes. La première étape consiste à construire la carte de probabilités de la présence de la cible dans l'ensemble de l'image. Nous pouvons pour cela reprendre le processus basé sur les histogrammes utilisé dans l'exemple de la Figure 89. Une fois que la carte est disponible, une fenêtre ou un noyau est initialisé (position et taille) en fonction des informations connues ou estimées. L'étape suivante est un processus itératif où il convient de calculer le barycentre des pixels inclus dans la fenêtre (ou le noyau) et de déplacer la fenêtre (ou le noyau) sur ce barycentre. Ces deux opérations (calcul du barycentre et déplacement de la fenêtre) sont répétées tant qu'un nombre d'itérations n'est pas atteint ou tant que le déplacement de la fenêtre est supérieur à un seuil. L'utilisation de l'algorithme « mean-shift » pour le suivi d'objet a été proposé par Comaniciu et Meer [164]. Les auteurs avaient déjà utilisé cette approche pour réaliser la segmentation des images couleurs [165]. A noter que dans les faits, nous ne calculons jamais l'intégralité de la carte de probabilité. A chaque itération, seuls les points à l'intérieur de la fenêtre d'observation ou du noyau doivent être évalués et une optimisation très classique consiste à mémoriser les points calculés puisqu'il y a forcément un recouvrement partiel de la fenêtre d'observation entre deux itérations.

Parallèlement à ce travail et dans la même période, Bradski [132] a proposé l'algorithme « CamShift » (Continuously Adaptive Mean Shift) qui encapsule le « mean-shift » et ajuste la taille de la fenêtre de suivi. L'algorithme « Cam-shift » est, lui aussi, un algorithme itératif. A chaque itération, un algorithme « mean-shift » classique est appliqué sur l'image. Une fois le point de convergence atteint, la taille de la fenêtre d'observation est adaptée en fonction du moment d'ordre 0. Dans l'article de Bradski, le ratio entre la hauteur et la largeur est fixé à une certaine valeur adaptée spécifiquement au cas du suivi des visages mais le principe se généralise simplement.

Les algorithmes « mean-shift » et « camshift » sont fréquemment associé à des caractéristiques basées sur les histogrammes [132][163][166] [167] mais ils peuvent être associés à d'autres types de caractéristiques comme la texture [168][142] ou le flot optique [169]. Une contrainte que l'on attribut régulièrement à ces deux approches [117] est qu'il doit y avoir un recouvrement partiel entre la position de l'objet à l'instant t-1 et sa position à l'instant t. Si un processus de prédiction/propagation a été mis en place, la contrainte est alors qu'il doit y avoir un recouvrement partiel entre la position prédite et la position réelle. Dans la très grande majorité des situations de vidéoprotection, avec une fréquence d'analyse de 5 images par seconde, cette condition est remplie et une approche « mean-shift » reste valable. Quand bien même, cette condition ne serait pas remplie, rien n'interdit d'utiliser une fenêtre d'observation plus grande.

Dans [10], nous avons adapté l'algorithme « mean-shift » pour améliorer la localisation des piétons en présence d'ombres portées (Figure 93). Dans ce travail, la carte de probabilités est calculée à partir du masque de mouvement et d'un gabarit.

Figure 93 : Utilisation du « mean-shift » pour améliorer la localisation des personnes. Sur la personne à droite de l'image nous pouvons visualiser les 4 itérations de l'algorithme de la position précédente en rouge à la position courante en jaune.

Les algorithmes de suivi par filtre particulaire et par une approche « mean-shift » sont les plus utilisée et les plus faciles à mettre en œuvre. Ils sont notamment tous les deux disponibles dans la librarie opencv [9]. À modèle d'apparence égal, Pérez et *al.* ont montré dans [141] l'intérêt des méthodes stochastiques de type filtrage particulaire face aux méthodes déterministes telles que le « mean-shift » pour le suivi visuel. Pour avoir abondamment expérimenté et utilisé ces deux approches, l'algorithme « mean-shift » permet d'obtenir un suivi plus précis et plus stable que le filtrage particulaire. Par contre, le filtrage particulaire est plus robuste aux occultations. Des résultats sont présentés dans le chapitre suivant.

3.4.2 Problème d'affectation

Dans cette deuxième approche, nous considérons que nous avons à notre disposition une liste B_j ($j \in [1 ; M]$) de blobs détectés à l'instant t que nous souhaitons mettre en correspondance avec une liste O_i ($i \in [1 ; N]$) d'objets suivis à l'instant t-1. Les deux listes n'ont pas forcément le même nombre d'éléments et quand bien même les deux listes auraient le même nombre d'éléments, il n'y a pas forcément de correspondance entre tous les éléments des deux listes. A partir des signatures extraites sur les objets et les blobs, nous pouvons calculer une mesure de similarité $S_{j,i}$ entre les blobs et les objets et construire un graphe biparti pondéré comme représenté sur la Figure 94.

[9] https://opencv.org/

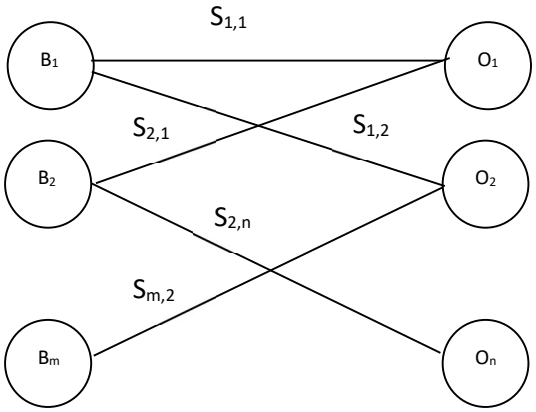


Figure 94 : représentation du problème sous la forme d'un graphe biparti pondéré

Dans ce graphe, nous pouvons remarquer qu'il n'y a pas d'arête entre le blob B_2 et l'objet O_2. Pour simplifier le problème, il est d'usage de limiter l'espace de recherche de chaque blob dans une zone de l'image afin de ne calculer les similarités qu'entre un blob et seulement les objets susceptibles de se retrouver dans son voisinage spatial. De même, il est possible de limiter la complexité du graphe en ne représentant pas les arêtes dont la similarité est inférieure à un seuil défini par l'utilisateur ou par l'application visée.

La recherche des correspondances entre blobs et objets peut alors se résumer à un problème d'affectation puisqu'il s'agit de trouver, parmi les couplages de cardinal maximal, celui qui a le poids maximum.

Une façon non optimale mais très simple à mettre en œuvre consiste à calculer la matrice d'association $A = [S_{j,i}]$ et à utiliser un algorithme glouton. A chaque itération, l'algorithme réalise l'association entre le blob j et l'objet i dont la similarité $S_{j,i}$ est maximale et supprime toutes les références au blob j et à l'objet i. L'algorithme se termine lorsque le nombre objet ou de blob restant est nul ou lorsque la similarité maximale est inférieure à un seuil indiquant qu'il n'y a plus d'association significative. Cette approche naïve est généralement suffisante dans le cas de la détection d'intrusions où d'une part le nombre de cibles à suivre est limité et où d'autre part, la sanction doit être immédiate et limitée dans le temps et dans l'espace.

Ce problème d'affectation peut être résolue plus efficacement à l'aide de l'algorithme de Munkres [170] ou plus exactement sa version un peu plus récente proposée par Bourgeois et *al.* [171] permettant de gérer les matrices rectangulaires, c'est-à-dire les matrices dans lesquelles le nombre d'objets n et de blobs m est différent.

Ce problème d'affectation, et plus généralement d'association de données, peut également être résolu à partir d'algorithmes plus complexe comme le filtre d'association de données probabiliste commune (JPDAF) [172] ou le suivi multi-hypothèse (MHT) proposée par Reid [173] et revisité dans [174][175]. L'algorithme JPDAF consiste à calculer un ensemble de probabilités jointes entre un objet i et un sous ensemble de blobs probables. Toutes les probabilités d'associations sont normalisées et utilisées pour la mise à jour de chaque objet cible. L'algorithme MHT est plus complexe et garde en mémoire plusieurs hypothèses de toutes les associations de données possibles dans une fenêtre temporelle afin de résoudre plus

efficacement les ambiguïtés. La complexité de l'algorithme et les coûts de calcul de cette recherche exhaustive sont très importants. En pratique, des techniques d'élagage sont généralement combinées avec le MHT pour restreindre le nombre croissant d'hypothèses.

3.4.3 Gestion des objets suivis

Un algorithme de suivi de cibles multiples ne se limite pas à la gestion des associations. Il doit également gérer l'apparition de nouvelles cibles dans la scène comme la disparition des cibles lorsqu'elles quittent la scène. L'algorithme doit également tenir compte des imperfections de l'étape de détection, que ce soit les faux positifs comme les faux négatifs, ainsi que tous les problèmes d'occultation totale ou partielle et cela pendant plusieurs images. Cette gestion proprement dite n'est généralement pas détaillée dans les articles. Les auteurs mettent plus en avant leur travail sur la signature des objets ou sur la gestion des associations.

Dans [10], nous avons proposé une méthode de suivi multi objets. Cette méthode s'appuie sur les travaux de Di Lascio et *al.* [176] qui formulent classiquement le problème de suivi multi-objets en un problème récursif d'associations entre détections et objets suivis. La particularité de la méthode qu'ils proposent est, premièrement, de prendre explicitement en compte les défauts qui peuvent survenir lors de la phase de détection, en particulier les défauts courants liés à l'utilisation d'un masque de mouvement (fusion, morcellement, non détection). Deuxièmement, leur algorithme gère les occultations inter-objets via l'utilisation de groupes qui permettent notamment de suivre collectivement les objets occultants et occultés. Troisièmement, ils modélisent l'évolution d'un objet par un automate à états finis permettant ainsi de synthétiser de manière compacte et intelligible l'état courant d'un objet ainsi que les transitions possibles pour celui-ci. Cette modélisation permet d'adapter le comportement de l'algorithme en fonction de l'état de chaque objet. Toutefois, l'algorithme proposé par Di Lascio est spécifique à une catégorie d'objets (piétons, valises) et nécessite un apprentissage spécifique pour chacune des classes d'objets à suivre. Nous avons donc proposé une généralisation de cette approche à tout objet se déplaçant dans la scène.

L'automate à états finis (voir Figure 95) décrit les différents états d'un objet et les conditions qui déclenchent la transition vers un état différent. L'état est utilisé à la fois pour influencer les étapes de traitement effectuées sur chaque objet et pour sélectionner la valeur la plus appropriée pour certains des paramètres impliqués dans le traitement. La grande force de l'automate à états finis est qu'il peut être adapté simplement pour différents scénarios en ajoutant des états et les transitions associés.

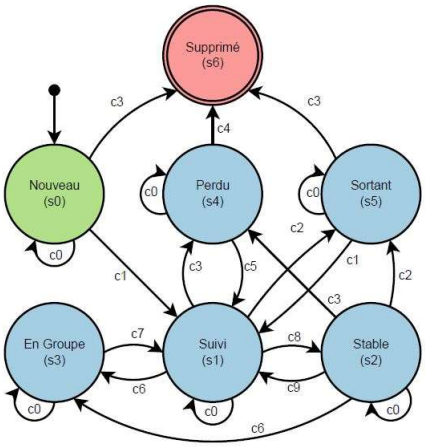


Figure 95 : automate à états finis tel que proposé par Di lasco [176]

Dans la version de base, les états de l'automate étaient les suivants :

- Nouveau : l'objet vient d'être créé ou est situé aux bords de la scène. S'il entre complètement, et ne touche pas les bords du cadre, il devient « suivi » (transition C1). S'il quitte la scène, il est supprimé (transition C3).
- Suivi : l'objet est entièrement dans la scène, mais la mise en correspondance n'est pas considérée comme fiable. Si un objet « suivi » est mis en correspondance avec un blob avec un niveau de similarité suffisant sur l'image suivante, il devient « Stable » (transition C8). Si l'objet n'est pas mis en correspondance avec un blob ou n'est pas retrouvé dans l'image (transition C3), il devient « perdu », et s'il touche le bord de la scène, il devient « sortant » (transition C2).
- Stable : l'objet est fiable. Si un objet « stable » n'est pas mis en correspondance avec un blob avec un niveau de similarité suffisant sur l'image suivante, il devient « suivi » (transition C9). Si l'objet n'est pas du tout mis en correspondance avec un blob ou n'est pas retrouvé dans l'image (transition C3), il devient « perdu », et s'il touche le bord de la scène, il devient « sortant » (transition C2).
- Perdu : l'objet n'est pas détecté. Il peut y avoir plusieurs raisons à cela, soit parce qu'il est complètement ou partiellement occulté par un élément de fond ou un autre objet, soit parce que le détecteur est en défaut. Si l'objet redevient visible (transition C4), il devient « suivi ». Si l'objet reste « perdu » pendant une durée supérieure à un seuil T_d, il est « supprimé » (transition C4). L'état « perdu » permet d'éviter qu'un objet ne soit oublié trop tôt lorsqu'il disparaît temporairement.
- Sortant : l'objet est localisé sur les bords de la scène. S'il disparaît, il devient « supprimé » (transition C3). S'il rentre à nouveau dans la scène, il devient « suivi » (transition C1). Cette transition n'est pas disponible dans l'automate de Di Lascio *et al.* mais elle permet d'éviter le cas où un objet quittant la scène change de direction pour rester dans la scène.
- Supprimé : l'objet n'est plus suivi et sera supprimé lors de la prochaine itération.

Avec cette première programmation, l'automate permet de gérer efficacement le suivi des objets. Dans le cas d'un scénario de détection d'intrusions, les étapes de haut niveau placées après l'automate n'ont plus qu'à gérer le cas des objets « stables ». Tous les autres objets sont dans des états transitoires et/ou leurs caractéristiques ne sont pas suffisamment fiables pour prendre une décision. Ainsi dans le cas de la détection d'intrusion, si un objet « stable » est dans une zone de l'image censée ne pas être autorisée, une alarme peut être levée.

L'automate peut facilement être adapté afin de gérer des scénarii plus complexes. Par exemple, nous avons programmé une version légèrement différente pour gérer la détection d'un bagage abandonné et une autre version pour gérer les files d'attentes. Dans les deux cas, nous avons simplement ajouté un état particulier ainsi que les transitions associées.

4 Apprentissage profond

4.1 Apprentissage des représentations spatio-temporelles

La révolution opérée par les réseaux de neurones profonds dans le domaine de la détection et la classification d'objets suite aux excellents résultats obtenus sur des grandes bases d'images telle ImageNet [177], a tout naturellement amené la communauté scientifique à appliquer ces méthodes au suivi visuel. A l'instar de la reconnaissance visuelle, les premières architectures ont été proposées à partir de réseaux neuronaux convolutionnels « classiques » (CNN) mais très rapidement de nouvelles architectures ont été étudiées telles que les réseaux siamois (SNN), les réseaux neuronaux récurrents (RNN), les réseaux contradictoires génératif (GAN), les auto-encodeurs (AE), *etc*.

L'utilisation des CNN pour le suivi a été motivé par le fait que les recherches récentes sur la vision par ordinateur et la reconnaissance des modèles ont mis en évidence les capacités des réseaux neuronaux convolutionnels (CNN) à résoudre des tâches difficiles telles que la classification, la segmentation et la détection d'objets. Bien que les détecteurs spécialisés basés sur les CNN et détournés pour le suivi permettent d'extraire des caractéristiques robustes, ces caractéristiques ne sont pas spécifiques au suivi mais à la tâche première qui est la localisation et la classification des objets. Les réseaux siamois ont alors été introduits afin de pouvoir réellement disposer d'un vecteur de caractéristiques adapté à la tache de suivi. Un réseau siamois, dans sa version la plus simpliste, se composent de deux CNN « classiques » reliés par une ou plusieurs couches de sorties. Ces réseaux jumeaux calculent la même fonction, pas forcément avec les mêmes poids, pour produire une carte de similarité. Ils peuvent alors être entraînés avec des données de suivi, typiquement une imagette de la cible et une imagette requête (correspondant ou non à la cible) et ainsi extraire des caractéristiques propres à la tache de suivi [178]. Les réseaux de neurones récurrents sont une autre classe de réseaux, qui permettent, dans une certaine mesure, de propager des informations dans le temps. A son niveau le plus fondamental, un RNN est simplement un type de réseau de neurones densément connecté. Cependant, la principale différence par rapport aux réseaux à « action directe » est l'introduction du temps. Dans les faits, dans un réseau neuronal récurrent, la sortie de la couche cachée est réinjectée sur l'entrée elle-même.

Plusieurs études de ces nouvelles architectures sont proposées dans la littérature [179][180]. Nous n'allons pas faire ici l'étude de toutes les architectures proposées dans la littérature mais nous présentons rapidement trois algorithmes de suivi : SORT [181], GOTURN [182] et ROLO, basés sur des approches différentes.

4.2 Présentation de quelques architectures

L'algorithme SORT proposé par Bewley et *al.* [181] est simple et efficace. Le principe de l'algorithme repose principalement sur l'analyse de l'inférence d'un détecteur spécialisé. Les auteurs utilisent le détecteur Faster R-CNN proposé par [183], mais précisent que leur approche s'adapte à n'importe quel détecteur d'objet. En effet, le suivi est assuré en maximisant l'intersection entre les boites englobantes des objets détectés dans deux images successives. Poursuivant leur travaux, les auteurs ont proposé une amélioration [184] en intégrant un modèle d'apparence profond basé sur un CNN et appris sur la base MARS [185]. La base MARS est une base utilisée pour la ré-identification de personnes. En conséquence, l'évolution proposée par les auteurs se spécialise au cas du suivi de personnes.

L'algorithme GOTURN a été proposé par Held et *al.* [182]. L'architecture de leur réseau est construite de façon à associer l'apparence au mouvement (Figure 96). L'apprentissage est réalisé hors ligne et s'adapte à n'importe quel type d'objet contrairement à l'algorithme SORT qui s'appuie sur un classifieur. Dans les faits, un premier réseau reçoit l'imagette de la cible et en déduit un vecteur de caractéristiques. Un deuxième réseau reçoit une imagette correspondant à une région candidate et en déduit un autre vecteur de caractéristiques. Ces deux vecteurs sont ensuite passés à un réseau entièrement connecté qui permet la localisation de la cible dans l'imagette candidate. D'après les auteurs, l'algorithme est assez rapide mais cela dépend fortement du nombre de cibles à suivre, puisque chaque cible doit être traitée indépendamment.

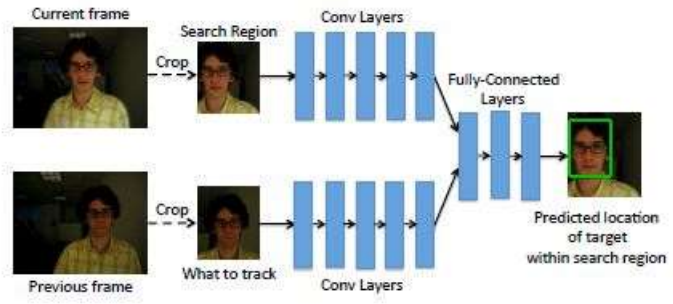


Figure 96 : architecture du réseau GOTURN. En entrée le réseau reçoit une imagette de la cible issue de l'image précédente et une imagette correspondant à la région de recherche dans l'image courant. La sortie du réseau indique la position de la cible dans la région candidate. [182]

L'algorithme ROLO proposé par Ning et *al*. [186] utilise également un détecteur spécialisé, en l'occurrence Yolo [111], pour extraire des caractéristiques visuelles riches et robustes ainsi que les inférences préliminaires de localisation (Figure 97). Cette première étape permet de gérer les dimensions spatiales. La deuxième étape est basée sur une architecture LSTM (long short term memory) permettant de tirer parti de la cohérence temporelle et assurer le suivi.

Les LSTM sont un type particulier de réseau de neurones récurent (RNN), capables d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter et *al.* [187]. Cette classe de réseau permet, dans une certaine mesure, de propager l'information dans le temps.

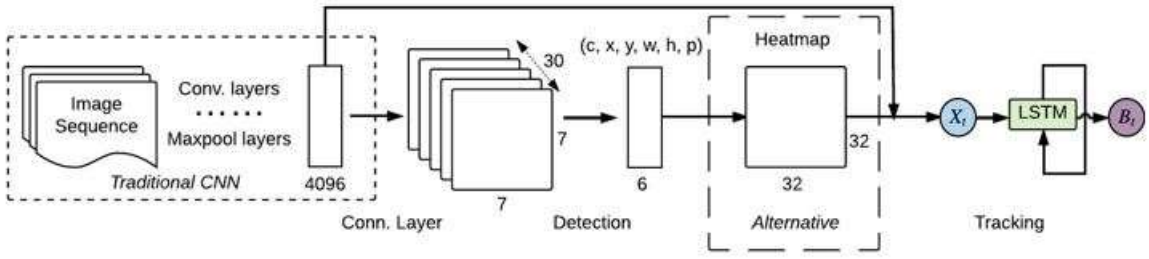


Figure 97 : vue simplifiée de l'architecture ROLO telle que présentée par les auteurs [186]

Parmi les réseaux populaires, nous pourrions citer MDNnet [188] qui a remporté le chalenge VOT2015 [189]. Comme pour SORT, les auteurs utilisent un CNN appris hors ligne comme extracteur de caractéristiques puis des couches spécifiques pour le suivi. L'algorithme VITAL, proposé par Song et *al.* [190], quant à lui utilise une architecture basée sur les réseaux génératifs contradictoires.

4.3 Limites des approches présentées

Comme nous avons pu le voir sur ces quelques exemples d'architectures, les réseaux neuronaux divisent l'analyse des données spatio-temporelles des vidéos en deux parties. Ils apprennent tout d'abord des structures spatiales sur chaque cadre vidéo (en utilisant un CNN 2D), puis ils apprennent des modèles temporels en utilisant des réseaux récurrents comme LSTM ou « convolutionnel LSTM » (ConvLSTM)[191]. Cette approche fait l'hypothèse que les dimensions spatiales et temporelles sont indépendantes et peuvent être traitées séquentiellement. En outre, certains auteurs montrent qu'un réseau LSTM n'est pas optimale pour traiter les données séquentielles [192].

Une alternative possible est alors d'utiliser une architecture différente basée sur des convolutions 3D avec cette première interrogation : peut-on prouver que les convolutions 3D sont le groupe de symétries adaptées pour capturer les invariances de la donnée spatio-temporelle ? Comme nous le verrons en toute fin de ce document, c'est une des questions dont nous allons tenter de répondre avec la dernière thèse que nous avons engagée.

5 Conclusion

Bien que des progrès indéniables ont été réalisés ces dernières années, le suivi reste un sujet ouvert et très actif. Les techniques d'apprentissages profond proposent des performances impressionnantes dans la détection et la segmentation des objets mais la tâche de suivi reste difficile, à plus forte raison lorsqu'il y a plusieurs cibles à suivre simultanément dans la scène.

Les tâches de détection et de suivi des objets d'intérêt dans une séquence vidéo peuvent paraître indépendantes mais elles sont en réalité indissociables et complémentaires. La solution à l'un des problèmes implique généralement la résolution implicite ou explicite d'un autre problème. En résolvant efficacement le problème de la segmentation des objets, il est plus facile de trouver une solution au problème de suivi. Des résultats de segmentation précis fournissent des observations d'objets fiables pour le suivi, ce qui peut résoudre des problèmes tels que

l'occultation, la déformation, la mise à l'échelle, *etc.*, et permet d'éviter fondamentalement les défaillances du suivi. D'autre part, des résultats de suivi d'objet précis peuvent également guider l'algorithme de segmentation pour déterminer la position de l'objet, ce qui réduit l'impact du mouvement rapide de l'objet, de l'arrière-plan complexe, des objets similaires, *etc.*, et améliore les performances de la segmentation de l'objet comme nous l’avons proposé dans [10].

Les tâches de détection et de suivi d’objet sont deux étapes clés dans la recherche d’une solution efficaces de vidéo-protection mais plus largement dans toutes les applications qui exploitent l’estimation du déplacement des objets en mouvement dans la scène. L’évaluation de ces deux étapes doit être particulièrement rigoureuse.

VI - Evaluation des étapes de la détection d'intrusions

1 Avant-propos

L'évaluation des algorithmes est une tâche primordiale que ce soit dans l'industrie ou dans le monde de la recherche académique, avec toutefois une différence dans les objectifs visés. L'industriel veut s'assurer que « ça marche » et le chercheur veut s'assurer qu'il « fait mieux » que ses confrères. Ce schéma est un peu manichéen et la frontière entre les deux est bien sûr un peu plus poreuse. Cependant, dans tous les cas, cela présuppose de définir correctement les données d'entrées, les données de sorties et le domaine de validité de l'algorithme testé. Les données d'entrées sont composées des caractéristiques du flux vidéo à analyser mais également des différents paramètres ou réglages qui peuvent être fixés par un opérateur ou déduit d'un prétraitement de la séquence vidéo. Les données de sorties sont classiquement les résultats fournis par l'algorithme ou par une étape particulière de l'algorithme. Pour comparer les résultats de plusieurs algorithmes entre eux, il est nécessaire que leurs données de sorties soient du même format ou alors, qu'il y ait une étape de mise en conformité. Ce sera notamment le cas pour comparer les performances de détection d'un algorithme basé sur une modélisation du fond et un détecteur spécialisé. Le premier génère en sortie une carte binaire de mouvement alors que le second fourni une liste de boites englobantes. Il est donc nécessaire de trouver une métrique commune ou bien de réaliser une segmentation du masque de mouvements pour extraire les boites englobantes. Le domaine de validité est, quant à lui, plus difficile à définir mais il est très important. Par exemple, tout au long de nos recherches, nous nous sommes particulièrement intéressés au cas des caméras fixes, c'est-à-dire telles que les paramètres intrinsèques et extrinsèques n'évoluent pas pendant la séquence vidéo. En toute rigueur, une caméra fixée sur un mat qui vibre légèrement en fonction du vent ne répond pas à cette définition. Cependant, dans un contexte de vidéo-protection, cela arrive fréquemment et un algorithme de détection devra être robuste à ce genre de situation. De même, la notion de temps réel est très suggestive. Elle dépend du contexte d'utilisation et des ressources qui peuvent être mise à disposition. En fonction de ces différents paramètres, des choix d'architecture ou des compromis entre précision et rappel sont pris en compte lors de l'élaboration d'un algorithme. Ces quelques situations montrent que la comparaison des algorithmes est une tâche difficile et que les résultats doivent être correctement interprétés.

Comme précisé dans [193], l'évaluation d'un algorithme peut se concevoir à différents niveaux :

- qualitatif : l'algorithme me fournit-il bien le résultat escompté ?
- quantitatif : quelle est la qualité du résultat obtenu ?
- fonctionnel : quel temps dois-je attendre pour obtenir le résultat ?
- robustesse : l'algorithme rend-il toujours un résultat correct si l'information à extraire dans l'image est altérée (bruit, contraste, *etc.*) ?

Dans ce chapitre, nous nous intéressons principalement à l'aspect quantitatif du résultat, bien que l'aspect robustesse sera également traité à travers des vidéos longues. Nous allons essentiellement nous focaliser sur les deux étapes clés d'un algorithme de détection d'intrusion à savoir la détection (plus précisément la modélisation du fond) et le suivi. Nous terminerons ce chapitre par l'évaluation des alarmes d'intrusions telles que nous l'avons définie dans le chapitre III, en présentant un cadre et des métriques originales.

2 Modélisation du fond

2.1 Bases de données

Une façon assez classique d'estimer et de comparer l'efficacité des méthodes de modélisation de l'arrière-plan est d'étudier le masque binaire de mouvements. La difficulté est alors de disposer d'une vérité terrain (GT) avec des masques de mouvement générés manuellement ou à partir d'outils plus sophistiqués. Plusieurs auteurs proposent quelques séquences vidéo avec les masques associés en lien avec leurs articles. Ces vidéos sont relativement courtes et ont été générées pour faire apparaître un problème particulier que les auteurs cherchent à compenser.

Comme nous l'avons déjà fait dans les chapitres précédents, nous n'allons pas présenter une liste exhaustive des bases de données spécifiques à l'étude des masques de mouvement. Dans [194] les auteurs présentent rapidement quelques jeux de données utilisés pour l'évaluation. De même sur son site WEB[10], Thierry Bouwmans référence une trentaine de jeux de données. Nous présentons quelques jeux de données auxquels nous nous sommes intéressés et que nous avons utilisés.

Wallflower [195] est l'une des toutes premières bases de vidéos proposée pour l'évaluation des algorithmes de modélisation de fond. Elle contient sept vidéos, chacune représentant un défi spécifique tel que le changement d'éclairage, le mouvement d'arrière-plan, *etc*.

MuHAVi [196] est un jeu de données vidéo principalement destiné aux algorithmes de reconnaissance d'action humaine basés sur des silhouettes (Figure 98). Un ensemble de silhouettes annotées manuellement a été spécialement préparé pour l'évaluation de ce type de méthodes. Néanmoins, les silhouettes annotées peuvent être utilisées comme masques de

[10] https://sites.google.com/site/backgroundsubtraction/Home

mouvements pour l'évaluation des méthodes de modélisation de l'arrière-plan et plus généralement des méthodes de segmentation. Le jeu de données se compose de 17 classes d'actions acquises à partir de 8 caméras. La base a été enrichie en 2014.

Figure 98 : images issues du jeu de donnée MuHAVI illustrant quelques-unes des actions et des prises de vue

CDnet [194] : la base CDnet est l'une des base les plus utilisées pour l'évaluation des modèles de fond. Elle a été créée en 2012 et enrichie en 2014 pour des ateliers en marge de deux éditions de la conférence CVPR. La version 2012 comportait 90000 images correspondant à 31 séquences vidéo (couleur et thermique) regroupées en 6 catégories. En 2014, 22 nouvelles vidéos ont été ajoutées avec des résolutions plus ou moins faibles et des fréquences d'acquisitions différentes. L'étiquetage de la base CDnet est plus riche que la simple annotation binaire que l'on retrouve habituellement (Figure 99). Dans un masque, un pixel peut prendre 5 valeurs indiquant s'il appartient au fond de la scène, à l'objet en mouvement dans la zone d'intérêt, à un objet en mouvement en dehors de la zone d'intérêt, ou à de l'ombre. La cinquième valeur est utilisée pour marquer le bord de l'objet, c'est-à-dire la limite entre l'objet lui-même et le fond. Cette distinction est intéressante parce qu'il est en général très difficile lors de l'étiquetage d'indiquer clairement la limite entre l'objet et le fond, sauf si l'on utilise des vidéos synthétiques.

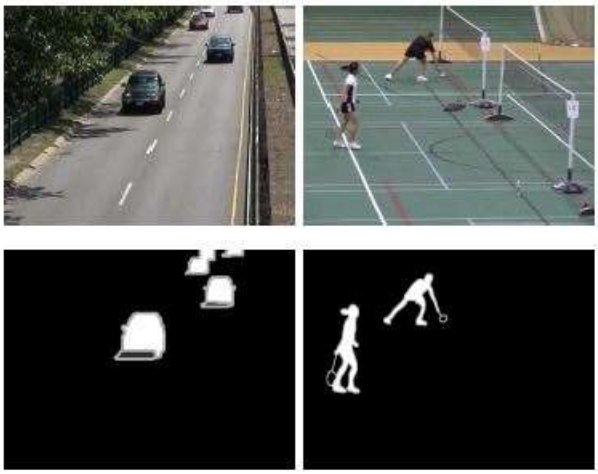


Figure 99 : exemple d'image et de masque de mouvement du jeu de données CDnet où nous pouvons remarquer les différentes valeurs de l'étiquetage sur le masque en bas à gauche, ainsi que la finesse notamment en ce qui concerne la raquette sur le masque en bas à droite.

Dans le cadre des deux ateliers, les auteurs ont donné les résultats des différents algorithmes de détection de mouvement.

BMC2012[197]: dans le cadre d'un atelier en marge de la conférence ACCV, nous avons participé à l'organisation et à l'élaboration de la base de vidéos utilisées pour l'évaluation des

algorithmes. Cette base comportait 10 vidéos synthétiques et 9 vidéos réelles dont certaines très longues. Les vidéos synthétiques (Figure 100) ont été réalisées à partir du simulateur SiVIC [198]. Cet outil a permis de générer des vidéos dans un contexte urbain en simulant différents types d'événements : ensoleillé, nuageux, brumeux, vent et bruit d'acquisition. Avec cet outil, nous avons pu fournir le masque de mouvement de la vérité terrain pour chaque image à la fréquence de 25 images par seconde.

Figure 100 : exemple de vidéos synthétiques avec masque d'avant-plan de notre jeu de donnée.

La deuxième partie du jeu de données dans BMC était composée de vidéos réelles (Figure 101) acquises à partir de caméras fixes dans des contextes de vidéosurveillance réels. Ce jeu de données a été construit afin de tester la fiabilité des algorithmes dans le temps et dans des situations difficiles telles que les scènes extérieures. Ainsi, nous avons proposé des vidéos longues (environ une heure et jusqu'à quatre heures) qui peuvent présenter un ou plusieurs changements de luminosité ou différentes évolutions de l'arrière-plan comme des véhicules qui stationnent ou quittent une zone de parking. Dans ces vidéos longues, seules quelques images ont été étiquetés manuellement.

Figure 101 : exemples d'images extraites de vidéos réelles dans notre jeu de données.

DAVIS [199] : cette base de données comporte 50 séquences différentes en résolution HD (1080p) à 25 images pas secondes et un total de près de 3500 images annotées. La vérité terrain est donnée sous la forme d'un masque binaire. Les vidéos ne contiennent à chaque fois qu'un seul blob correspondant à un ou plusieurs objets connectés spatialement. Un exemple de plusieurs objets spatialement connectés est donné sur la figure suivante (Figure 102) où sur la dernière image une femme, une poussette et un enfant dans la poussette sont segmentés sous la forme d'un seul blob. Les auteurs justifient le choix de n'avoir qu'un seul blob par séquence afin de pouvoir comparer équitablement les méthodes de segmentation opérant sur un seul objet ou sur plusieurs objets.

Figure 102 : exemple de séquence de la base de données DAVIS [199] *avec le masque de la vérité terrain en surimpression.*

La base DAVIS n'est pas la base la plus adaptée pour évaluer les approches de modélisation de l'avant-plan parce que dans la plupart des vidéos les conditions de prise de vue évoluent au cours de la séquence. Par contre les auteurs de l'évaluation introduisent des métriques intéressantes que nous verrons prochainement.

Une vérité terrain basée sur un masque de mouvement n'est pas la seule approche permettant de vérifier la qualité de la modélisation et il existe d'autre bases utilisant le principe de boîte englobante. Toutefois ces bases sont plus adaptées pour estimer la qualité du suivi des objets comme nous le verrons dans le paragraphe suivant.

2.2 Métriques d'évaluation

De façon générale, la qualité des masques de mouvement générés par un algorithme de modélisation est évaluée en comparant pixel à pixel le masque de mouvement avec un masque issu de la vérité terrain. Chaque pixel est alors associé à une des quatre catégories suivantes que nous utiliserons par la suite à l'aide de leur acronyme anglais :

- Vrai positif (TP) : si le pixel est correctement détecté comme un pixel d'avant-plan,
- Vrai négatif (TN) : si le pixel est correctement détecté comme un pixel d'arrière-plan,
- Faux positif (FP) : si le pixel de l'arrière-plan est détecté comme un pixel d'avant-plan,
- Faux négatif (FN) : si le pixel de l'avant-plan est détecté comme un pixel de l'arrière-plan.

Les FP sont des pixels qui sont détectés à tort comme étant en mouvement et qui vont ajouter du bruit dans le masque. Sans une attention particulière, les pixels correspondant à l'ombre portée des objets en mouvement sont souvent détectés comme faux positif. Les FN correspondent aux pixels qui n'ont pas été détectés comme étant en mouvement. Ils peuvent entrainer des omissions ou plus fréquemment des trous, voire la fragmentation d'un objet en plusieurs blobs.

Ces quatre classes permettent de construire une matrice de confusion, appelée également table de contingence, où les deux lignes correspondent à la vérité terrain et les deux colonnes à la classe estimée.

		Mesure	
		Avant-plan	Arrière-plan
Vérité	Avant-plan	TP	FN
	Arrière-plan	FP	TN

A partir des données de la matrice de confusion, il est possible de calculer un certain nombre de mesures. Les plus communes sont les suivantes :

- Rappel (Re) : $TP / (TP + FN)$
- Précision (Pr) : $TP / (TP + FP)$
- Spécificité (Sp) : $TN / (TN + FP)$
- Taux de faux positif (FPR) : $FP / (FP + TN)$
- Taux de faux négatif (FNR) : $FN / (TN + FP)$
- *F*-mesure: 2 . (Pr . Re) / (Pr + Re)

Le rappel permet d'évaluer la capacité du modèle à détecter les pixels de l'avant-plan et la précision permet d'évaluer la capacité du modèle à ne détecter que les pixels de l'avant plan. Afin de pouvoir comparer deux modèles ayant des valeurs de précision et de rappel différentes, il est d'usage d'utiliser la F-mesure qui correspond à la moyenne harmonique des deux valeurs. L'utilisation de la moyenne harmonique à la place de la moyenne arithmétique permet de punir d'avantage les mesures extrêmes. Ces trois mesures sont régulièrement utilisées par les auteurs pour évaluer leur modèle ou pour comparer plusieurs modèles entre eux [200][201] [194][202].

Par exemple, pour les évaluations de CDnet [203], ces mesures sont calculées pour chaque masque de mouvements. Pour chacune de ces mesures, une valeur moyenne est calculée sur l'ensemble de la séquence. Les auteurs justifient ce choix (par opposition à la mise en commun de tous les pixels dans la séquence, puis la moyenne) par le fait qu'il empêche les biais qui se produiraient si certaines séquences étaient beaucoup plus grandes en termes de résolution ou de nombre d'images.

Les mesures précision, rappel, F-Mesure, sont certes intéressantes pour avoir une vision globale de la qualité de la segmentation, mais elles ne permettent pas de repérer la plupart des défauts tels que la fusion, la fragmentation ou tout simplement la non-détection des objets ; à plus forte raison lorsque ces mesures sont moyennées sur l'ensemble de la séquence ce qui est généralement le cas. Afin de palier ces défaut, d'autres mesures ont été introduites.

Dans DAVIS [199] , les auteurs utilisent l'indice de Jaccard entre un masque et une vérité terrain à chaque image ainsi que la mesure F pour les points du contour. Ces deux mesures permettent une estimation de la qualité de la segmentation à chaque image. Afin d'évaluer la stabilité temporelle de la segmentation tout au long de la séquence, les auteurs introduisent une mesure de stabilité basée sur l'évolution du contour de l'objet. Le masque des objets est transformé en polygones représentant les contours sur lesquels sont calculés un descripteur de formes SCD [204]. La stabilité du contour entre deux images successives est estimée en utilisant la mesure DTW pour « dynamic time warping » appliquée sur le descripteur de forme. La mesure DTW permet d'estimer une distance point à point entre deux courbes. Nous l'avons utilisée dans [20]

pour estimer la correspondance entre deux trajectoires. Les auteurs calculent le coût moyen par point comme mesure de stabilité. Si la transformation est fluide et précise, le coût est relativement faible et, est simplement lié à la déformation de l'objet. Un coût important permet, d'après les auteurs, de mesurer efficacement les oscillations et les inexactitudes de l'extraction du contour.

Dans le BMC2012 [197], nous avons présenté trois types de mesures. Tout d'abord les mesures traditionnelles de rappel, précision et F-mesure calculé sur le masque de mouvement auxquelles nous avons ajouté une mesure du rapport signal / bruit (PSNR : « Peak Signal Noise Ratio ») :

$$PSNR = \frac{1}{N}\sum_{i=1}^{N} 10.Log_{10}\frac{M}{\sum_{j=1}^{M}\|S_i(j) - G_i(j)\|^2}$$

où $S_i(j)$ est le $j^{ième}$ pixel du masque i de taille M dans la séquence S composée N images et G la vérité terrain. Cette première série de mesures permet de comparer le comportement « brut » de chaque algorithme.

Nous avons également utilisée une mesure perceptuelle (SSIM : « Structural SIMilarity ») donnée par [205].

$$SSIM = \frac{1}{N}\sum_{i=1}^{N}\frac{(2\mu_{s_i}\mu_{g_i} + c_1)(2cov_{S_i,G_i} + c_2)}{(\mu_{S_i}^2 + \mu_{G_i}^2 + c_1)(\sigma_{S_i}^2 + \sigma_{G_i}^2 + c_2)}$$

où μ_S, μ_G sont les valeurs moyennes, σ_s, σ_g les écarts types et $cov_{S,G}$ la covariance du masque S et de la vérité G associée. Les deux valeurs c_1 et c_2 sont des constantes correspondant à la dynamique des images d'entrées. L'introduction d'une mesure perceptuelle pour l'évaluation d'un masque de mouvement a été motivée en supposant que la perception visuelle humaine est fortement adaptée pour extraire les informations structurelles d'une image.

Enfin dans le cadre de BMC2012, nous avons de nouveau utilisé la mesure D-Score que nous avions présenté dans [11]. Le but de cette mesure est de pénaliser les erreurs de classification des pixels en fonction de la position réelle des objets. Comme la distance de Baddeley [206], il s'agit d'une mesure de similarité appliquée aux images binaires et basées sur la transformation en distance (Figure 103). Pour calculer cette mesure, nous ne considérons que les erreurs (faux positifs et faux négatif) du masque de mouvements. Chaque coût d'erreur dépend de la distance du pixel mal classé avec le pixel positif le plus proche dans la vérité terrain. Nous avons introduit cette mesure pour pénaliser les erreurs de classification proches des objets en mouvement et susceptible déformer leur contour ou de fusionner des objets entre eux.

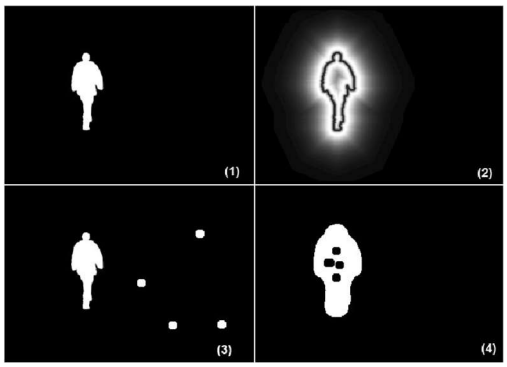


Figure 103 : illustration de la mesure D-Score. (1) Vérité terrain. (2) Carte du coût. (3) Exemple d'erreurs à longue portée. D-score = 0.003 (4) Omissions avec des erreurs de moyenne portée. D-score = 0.058.

Formellement, le D-Score est donné par :

$$DScore(x) = \begin{cases} \exp\left(-\left(\ln(2.DT(x)) - \frac{5}{2}\right)^2\right) & si\ DT(x) > 0 \\ 0 & sinon \end{cases}$$

où DT(x) est la distance entre le pixel x et le pixel positif le plus proche dans la vérité terrain.

Cette mesure est un bon exemple de métriques introduites en fonction du contexte d'exploitation du résultat de la segmentation issue de la modélisation. En effet, nous utilisions les points du contour pour calculer un vecteur de caractéristiques associés à chaque objet. Ce vecteur de caractéristiques était ensuite utilisé comme signature pour suivre les objets au cours de la séquence mais également pour permettre une classification. Dans ces conditions, une attention particulière devait être apportée à la détection du contour des objets, alors que les trous dans un objet de même que la détection à tort de blobs loin des objets, n'avaient que peu ou pas d'incidence.

3 Evaluation des algorithmes de suivi

3.1 Jeux de données

Dans le cadre du suivi d'objets, le nombre de jeux de données est plus fournis que pour l'évaluation de la modélisation de fond. Il peut y avoir plusieurs explications à cela. Dans le cas de la phase de détection, il existe plusieurs solutions comme la segmentation des objets en mouvement via une modélisation du fond ou l'utilisation d'un détecteur spécialisé. Or, dans ces deux cas, les données de sorties sont différentes, puisque la segmentation se base sur un masque de mouvements et le détecteur génère une liste de boîtes englobantes. Certes, à l'issue de la segmentation, il est possible de calculer les boîtes englobantes pour l'évaluation mais au prix d'une perte d'information. Dans le cas du suivi, l'évaluation à partir de boîtes englobantes est très largement utilisée puisqu'elle s'adapte à la plupart des algorithmes de détection. Dans le cas du suivi, c'est essentiellement la position et l'identifiant de l'objet que l'on va scruter mais pas la silhouette. Il y a donc un consensus à l'utilisation des boîtes englobantes ce qui rend l'étiquetage manuel un peu plus facile.

Il est également intéressant de voir comment la complexité des jeux données a évolué au cours de ces dernières années. Les scénarios des premiers ensembles de données de référence pour le suivi dans le cadre de la surveillance, tels que les bases de données VIVID [207] ou CAVIAR [208], sont relativement simples avec une ou deux cibles à suivre alors que pour le défi MOT 2020 [209], ce sont près de 200 cibles qu'il faut suivre simultanément.

VIVID [207] : dans le cas de **VIVID**, il s'agit de suivre des véhicules, c'est-à-dire des objets qui ne sont pas déformable et dont les trajectoires sont relativement simples à modéliser et à prédire. Il y a cependant dans ce jeu de données des situations complexes comme la gestion des occultations (Figure 104).

Figure 104 : exemple de scène issue du jeu de données VIVID [207] *avec occultation totale d'un objet en mouvement*

CAVIAR [208][11] : le jeu de données issu du projet Caviar propose des scénarios un peu plus complexes dans la mesure où ils mettent en scène des personnes et quelques interactions. Parmi les scénarios, nous pouvons trouver des personnes marchant seules, s'arrêtant un court instant, faisant des allées-retours ou croisant d'autres personnes. Sur la plupart des vidéos, la vérité terrain est donnée sous la forme d'une boite englobante associée à un identifiant pour chaque objet en mouvement. Cependant, sur certaine vidéo, la vérité terrain est plus complexe (Figure 105) et contient des informations complémentaires sur les personnes comme la position de la tête, des mains et des pieds ainsi que la direction du regard.

[11] https://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

Figure 105 : exemple d'étiquetage d'une vidéo du jeu de donnée CAVIAR [208] *où les personnes sont représentées par une boite englobante et où la position de la tête, des mains et des pieds sont repérées par un cercle.*

PETS2009 [210]: le jeu de données PETS 2009 a largement été utilisé par la communauté scientifique pendant plusieurs années. Il a été réalisé sur un site à l'aide de plusieurs caméras synchronisées permettant d'obtenir plusieurs points de vue d'une même scène. A l'origine, ce jeu de données a été conçu sur la base de trois types de scénarios : nombre de personnes et estimation de la densité (Figure 106), suivi des personnes (Figure 107), analyse des flux et reconnaissance d'événements.

Figure 106 : exemple d'une séquence issue de PETS 2009 [210] *et utilisée pour l'évaluation des algorithmes d'estimation de la densité*

Figure 107 : exemple d'une séquence issue de PETS 2009 [210] *et utilisée pour l'évaluation des algoritmes de suivi*

A noter que pour ce jeu de donnée, les concepteurs ont fourni un certain nombre d'informations sur les modèles de caméras, leur position, leur orientation, permettant de réaliser une calibration du système. Par contre, il n'y a pas de vérité terrain. Certaines vidéos ont été labellisées par la suite[12]. Ce jeu de données est plus complexe que les précédents. Dans le cas des scénarios de suivi par exemple, une vingtaine de personnes se déplacent dans tous les sens, se croisent, opèrent des demi-tours ou interagissent. Par ailleurs, sur certaines séquences, des problèmes d'ombres portées apparaissent.

[12] http://www.milanton.de/data.html

VOT Challenge [189][13] : les défis VOT pour « Visual Object Tracking » ont été initiés en 2013 dans le but d'évaluer différents traqueurs. Dans le cadre de ce défi, il s'agit de suivre une cible unique pour laquelle la boite englobante est donnée pour la première image. Le jeu de données propose une grande variété de cibles et de situations comme par exemple suivre une fourmi dans une boite, un joueur de basket pendant un match ou encore une moto sur une piste (Figure 108).

Figure 108 : exemple de situations différentes dans le jeu de données VOT [189]*. Lorsque plusieurs objets sont en mouvement dans la scène, un seul est initialisé et suivi.*

Le point de vue est rarement unique et la taille de la cible comme son orientation peuvent changer très significativement au cours d'une même séquence (Figure 109). Le but ici est d'évaluer la qualité d'un traqueur basée sur des données visuelles.

Figure 109 : exemple de changement de taille et d'orientation d'une cible à suivre dans une même séquence dans le jeu de données VOT [189]

Le jeu de données utilisé par les défis VOT a évolué au cours du temps en fonction des performances des traqueurs. Des vidéos simples ont progressivement été enlevées au profit de vidéos plus complexes. De même, afin de répondre à l'intérêt croissant pour l'imagerie thermique, un défi spécifique a été créé à partir de 2015. En 2020, les organisateurs ont également proposé un jeu de données avec une vérité terrain basée sur des masques de segmentations à la place des boites englobantes.

MOTChallenge [143][14] : le défi MOT pour « Multiple Object Tracking » est basé sur la même idée que VOT, c'est-à-dire offrir à la communauté un ensemble de données et une plateforme de tests commune permettant l'évaluation des algorithmes de suivi. Dans le cas de MOT et contrairement à VOT, il s'agit d'évaluer le suivi de plusieurs cibles et plus précisément, d'évaluer

[13] https://votchallenge.net/

[14] https://motchallenge.net/

le suivi de plusieurs personnes. Le jeu de données est principalement constitué de scènes urbaines.

Figure 110 : exemples de scènes du jeu de données MOT 2016 [143]

Le premier défi MOT a été lancé en 2014 et a évolué au cours du temps afin de proposer des scènes de plus en plus complexe. En 2016, il s'agissait de suivre une vingtaine de personnes sur une séquence (Figure 110). Pour le défi 2020 [209], de nouvelles vidéos ont été proposées dont une avec plus de 200 cibles à suivre sur une même image (Figure 111).

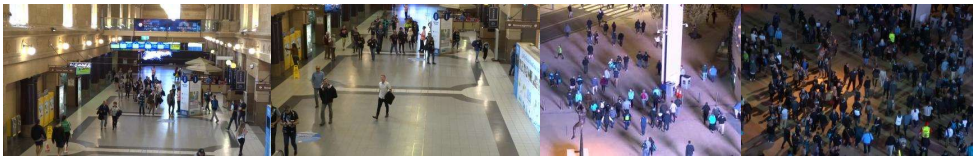


Figure 111 : exemples de scènes du jeu de données MOT 2020 avec une augmentation significative de la densité par rapport au défi 2016

Ces quelques exemples de jeu de donnée montrent l'intérêt accordé à la tache de suivi et montre surtout, que malgré les indéniables progrès réalisés ces dernières années, le suivi reste un sujet ouvert et très actif. Il existe bien d'autres jeux de données et analyses comparatives sur le sujet comme par exemple le défi MOTS [211][15] qui inclut la segmentation des objets ou encore le défi OTB 2013 [212] sur le suivi d'objets en temps réel.

3.2 Métriques d'évaluations

L'analyse comparative des traqueurs est assez simple lorsqu'il n'y a qu'un seul objet à suivre et à localiser. Dans ce cas, les protocoles d'évaluation de l'ensemble de données OTB 2013 [212] et VOT 2016 [213] sont principalement utilisés.

Pour le suivi de plusieurs objets, l'évaluation est plus complexe. En effet, l'algorithme de suivi ne doit pas seulement localiser précisément les objets dans la scène mais il doit également préserver une identité unique à chaque objet tout au long de la séquence.

Plusieurs travaux ont été menés afin de proposer des métriques pour ce type d'évaluation du suivi de plusieurs objets. Dans [214], les auteurs proposent une métrique basée sur la localisation des objets alors que dans [215], les auteurs mettent l'accent sur le maintien de

[15] http://www.vision.rwth-aachen.de/page/mots

l'identité de l'objet tout au long de la séquence. Dans [216], les auteurs proposent une approche spatiotemporelle pour l'évaluation des systèmes de suivi.

Suite aux travaux conduits dans le cadre du projet CLEAR « Classification of Events, Activities and Relation-ships » et l'article de Bernardin et *al.* [217], un consensus au sein de la communauté scientifique s'est dégagé autour de l'utilisation des métriques CLEAR MOT. Ces métriques sont notamment utilisées pour évaluer les algorithmes de suivi dans les défis MOT [143] ou PETS [210] ou encore dans plusieurs états de l'art sur le sujet [39][218][219].

Avant de poursuivre la description de ces métriques, nous définissons les notations suivantes :

- $\mathcal{G}_t = \{g_{i,t}\}_{i=1}^{N_t}$ est l'ensemble des N_t objets présents dans la scène à l'instant t (vérité terrain)
- $\mathcal{H}_t = \{h_{j,t}\}_{j=1}^{M_t}$ est l'ensemble des M_t objets suivis par l'algorithme évalué à l'instant t (hypothèses).

Il s'agit alors de trouver, à chaque instant t, les correspondances entre les éléments de G_t et ceux de H_t. A noter que les mises en correspondance ne sont autorisées que si la distance entre un objet $g_{i,t}$ et une hypothèse $h_{j,t}$ est relativement faible, c'est-à-dire inférieure à un seuil T défini. Cette notion de distance peut être évaluée à partir des positions du centre des rectangles englobants mais est plus généralement évalué à partir du recouvrement spatial entre les boites englobantes des objets annotés et des objets suivis.

$$d_{i,j,t} = \frac{|g_{i,t} \cap h_{j,t}|}{|g_{i,t} \cup h_{j,t}|}$$

A partir de cette mise en correspondance, il est possible d'évaluer :

- le nombre de correspondances trouvées noté c_t ;
- parmi ces correspondances, celles pour lesquelles l'identité de l'objet (selon l'algorithme de suivi) a changé depuis l'instant précédent. Leur nombre est noté is_t ;
- le nombre d'éléments de $\mathcal{G}_t$ non associés, ce nombre correspondant au nombre d'objets non détectés est noté fn_t ;
- le nombre d'éléments de $\mathcal{H}_t$ non associés, ce nombre correspondant au nombre de fausses détections est noté fp_t ;
- l'erreur totale commise sur la position de chacun des couples objet/hypothèse établis à l'instant t, noté d_t.

A partir de ces informations, Bernardin et *al.* [217] proposent les métriques suivantes :

Multiple Object Tracking Precision (MOTP) :

$$MOTP = \frac{\sum_t d_t}{\sum_t c_t}$$

Cette quantité mesure l'erreur moyenne commise sur les positions des objets suivis. Elle permet d'évaluer la précision du suivi indépendamment des autres sources d'erreur (changement d'identité, non détection, fausse détection). Pour un algorithme idéal, MOTP = 0, et sa valeur

est croissante à mesure que l'erreur d'estimation des positions augmente. Comme indiqué précédemment, la mise en correspondance n'est autorisée que si la distance entre une hypothèse et une vérité terrain est inférieure à un seuil T. Dans ces conditions, la mesure MOTP telle qu'elle est définie ci-dessus est bornée entre [0 ; T]. Afin de faciliter la comparaison entre les méthodes, la mesure MOTP est souvent reformulée pour varier entre 0 et 1 avec une valeur de 1 pour un algorithme idéal :

$$MOTP = 1 - \frac{1}{T} \cdot \frac{\sum_t d_t}{\sum_t c_t}$$

C'est cette définition que nous utilisons dans la suite de ce manuscrit.

Multiple Object Tracking Accuracy (MOTA):

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + is_t)}{\sum_t N_t}$$

Cette quantité rassemble en un indicateur unique l'ensemble des types d'erreurs commises par l'algorithme de suivi (non-détection, fausse détection et changement d'identité). Pour un algorithme idéal, MOTA = 1 et sa valeur est décroissante à mesure que le nombre d'erreurs augmente. Il est à noter que ce score peut être négatif.

False Positive (FP) :

$$FP = \sum_t fp_t$$

Cette quantité correspond au nombre total de fausses détections (objets suivis ne correspondant pas à un objet réellement existant). Pour un algorithme idéal, FP = 0.

False Negative (FN) :

$$FN = \sum_t fn_t$$

Cette quantité correspond au nombre total d'omissions (objet réellement existant non suivi). Pour un algorithme idéal, FN = 0.

ID Switch (IDs) :

$$IDs = \sum_t is_t$$

Cette quantité correspond au nombre total de changements d'identité au cours de la séquence. Pour un algorithme idéal, IDs = 0.

Cette première série de métriques permet d'évaluer efficacement les algorithmes de suivi tant sur l'aspect localisation avec MOTP que sur la capacité des algorithmes à détecter et suivre fidèlement les cibles avec MOTA. Dans une application de vidéo-surveillance, il peut être

intéressant d'avoir une estimation sur la capacité de l'algorithme de suivi à fournir des trajectoires fiables. Li et al. [220] proposent de classifier les trajectoires selon le pourcentage de points de la trajectoire effectivement suivis.

Mostly Tracked (MT) : si plus de 80% des points d'une trajectoire ont été correctement estimés, la trajectoire est considérée comme "majoritairement suivie".

Partially Tracked (PT) : si entre 20% et 80% des points d'une trajectoire ont été correctement estimés, la trajectoire est considérée comme "partiellement suivie".

Mostly Lost (ML) : si moins de 20% des points d'une trajectoire ont été correctement estimés, la trajectoire est considérée comme "majoritairement perdue".

4 Evaluation de la détection d'intrusion

4.1 Les jeux de données

L'évaluation globale d'un algorithme de détection d'intrusions est plus complexe que celles des étapes intermédiaires que nous venons de voir, parce qu'elle fait appel à des informations de plus haut niveau. Il existe quelques jeux de données avec des scénarios de vidéo-protection mais il est très difficile d'en trouver qui soient réellement réalistes. Par exemple, le laboratoire d'intelligence artificielle, robotique et vision de l'université du Minnesota met à disposition un jeu de données **UMN**[16] avec notamment quelques vidéos sur la détection de bagages abandonnés. Une des vidéos proposées « Abandoned object detected while moving person is not. Shows computed foreground blobs used in detection process" est une séquence de 10 secondes où l'on voit une personne seule déposer un sac dans une rue déserte. Cette vidéo n'est clairement pas représentative du défi lié à la problématique du « bagage abandonné » tel qu'il est perçu par les exploitants. Par exemple, en 2019, la SNCF (Société nationale des chemins de fer français) a mis en place un test en situation réelle dont le scénario était de détecter la présence d'un bagage abandonné, *i.e.* un bagage séparé de son propriétaire (hors champ) et immobilisé pendant au moins 2 minutes dans un hall de gare ou dans les couloirs d'accès aux voies.

En 2006, l'INRIA a proposé le jeu de données **ITESEO**[17]. Le corpus de vidéo est assez limité en durée, un peu moins de 3h, mais propose quelques défis intéressants comme les problèmes d'ombre et de luminosité ainsi que des scènes acquises avec des caméras thermiques. Comme précisé dans [221], la gestion des ombres peut être étudiée sous au moins trois problèmes différents: ombres à différents niveaux d'intensité (c'est-à-dire faiblement ou fortement

[16] http://mha.cs.umn.edu/

[17] https://www-sop.inria.fr/orion/ETISEO/

contrastées), ombres au même niveau d'intensité mais avec différents types de fonds en termes de couleur et de texture et ombres avec différentes sources d'éclairage en termes de position de la source et de longueurs d'ondes. Le point particulièrement intéressant de ce jeu de données et le soins mis par les auteurs pour proposer différentes métriques d'évaluation [221].

Pour évaluer les algorithmes sur des vidéos plus longues, Adam et al. [222] proposent, dans le jeu de données **SUBWAY,** deux vidéos totalisant un peu plus de deux heures d'enregistrement. Le défi **TRECVid** 2008 [223] va encore un peu plus loin et propose une centaine d'heures de vidéo. Cette quantité de données relativement importante est intéressante pour estimer les performances de détection d'événements à faible fréquence. Ces vidéos ont été prises en intérieur dans un aéroport particulièrement fréquenté sur une période de dix jours et pendant deux heures chaque jour.

Plus récemment, dans [224], les auteurs proposent d'utiliser le jeux de données **USCD**[18] pour la détection d'anomalies. Le jeu se composent de deux séries de vidéos d'une même scène totalisant en tout une centaine de séquences, dont environ la moitié est destinée à l'entrainement et l'autre moitié aux tests. Cependant, ces séquences ne comportent que 150 à 200 images, ce qui correspond à un total (entrainement et test) de 25mn d'enregistrement environ. C'est une durée beaucoup trop faible pour estimer la capacité d'un algorithme à éviter de générer des fausses alarmes lorsqu'il sera déployé sur un site. A noter que le masque de mouvements est donné pour quelques-unes des séquences de ce jeu de données.

Le jeu de données **VIRAT** [225], proposé en 2011 s'approche encore un peu plus des besoins de la vidéo protection. Il a été créé dans le but de permettre l'évaluation des algorithmes de reconnaissance d'événements visuels, tels que détecter une personne qui court, qui marche, qui flâne, qui monte ou descend d'un véhicule, *etc*. Au total, 23 types d'événements sont étiquetés. Le jeu de données se compose d'environ 25h de vidéo réparties en 6 scènes différentes prises avec des caméras fixes. Une autre partie du jeu de données se compose de vidéos (environ 4h) avec prise de vue aérienne, ce qui sort du cadre de notre étude. Concernant les caméras fixes, les auteurs du jeu de données proposent plusieurs séquences pouvant aller jusqu'à une dizaine de minutes. Certaines de ces séquences ont été prises les unes à la suite des autres ce qui permet un enregistrement continu de plus d'une heure. D'autres séquences ont été prises à différents moments de la journée ce qui laisse apparaître des problèmes liés aux ombres portées. Ce jeu de données est intéressant pour évaluer les interactions entre personnes, véhicules et autre mais il n'est pas encore suffisant pour l'évaluation des algorithmes de détection d'intrusions.

[18] http://www.svcl.ucsd.edu/projects/anomaly/dataset.html

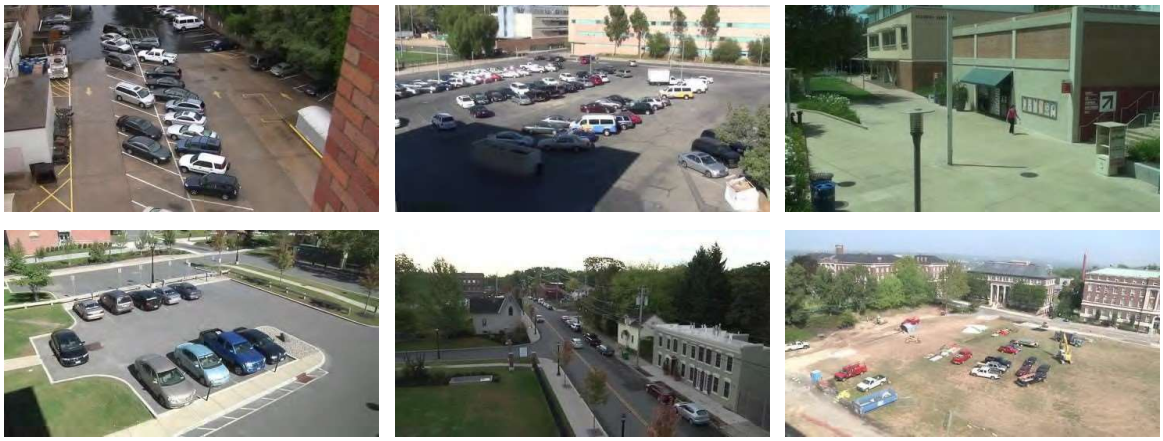


Figure 112 : exemples des six scènes fixes du jeu de données VIRAT [225]

Tous ces jeux de données ont été proposés pour répondre à une problématique et sont toutes intéressantes à leur niveau. Cependant, dans plusieurs applications, et la vidéo-protection en fait partie, la difficulté n'est pas seulement de détecter correctement un événement : intrusion, bagage abandonné, *etc.*, mais il est également important de ne pas détecter à tort, c'est-à-dire de limiter le nombre de fausses alarmes. C'est d'ailleurs la raison qui nous a poussé à proposer des vidéos de plus d'une heure lors du défi BMC[197]. Un jeu de données qui répond plus directement aux défis inhérents à la vidéo protection est le jeu de données **I-Lids**[19]. Depuis 2006, le ministère de l'intérieur britannique met disposition ce jeu de données destiné à évaluer les systèmes de détection intelligents basés sur l'analyse de données vidéo. Plusieurs scénarios sont proposés : détection de bagages abandonnés, détection de véhicules stationnés, surveillance de portes d'accès et de zones stériles, *etc*. Les vidéos sont issues de différents modèles de caméras : couleur, avec infra-rouge intégré, ou thermique (Figure 113). Le jeu de données comporte plus d'un Téra octet de vidéo. Chaque scénario comprend environ 24 heures de séquences enregistrées dans des conditions différentes : jour, nuit, météo, niveau d'activité du fond de la scène. Sur chaque scène, des événements d'alarmes sont joués par des acteurs et le reste des séquences ne comprend qu'une activité « normale » d'arrière-plan ne contenant pas d'alarme.

Figure 113 : exemple de scènes du jeu de données I-LIDS

La grande force de ce jeu de donnée, par rapport à la plupart des autres jeux de données, est qu'il permet tester un algorithme de détection d'intrusions dans des conditions réelles en

[19] https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems

évaluant la capacité de l'algorithme à isoler les événements pertinents et à ne pas lever d'alertes sur le « bruit de fond » de la scène.

4.2 Les Métriques usuelles

L'évaluation d'un algorithme de détection d'intrusions dépend beaucoup de la façon dont l'information va être traitées ainsi que des données qui vont être transmises. Dans la mesure où il y a peu de jeux de données spécifiques liée aux défis de la détection d'intrusions, il y a de fait peu de métriques définies dans l'état de l'art. Le type d'évaluation qui s'approche le plus est celui de la reconnaissance d'événement.

Dans le cadre du projet ETISEO [221], les auteurs ont proposé différentes mesures permettant d'évaluer les performances d'un système de vidéo surveillance pour chaque tache de traitement : détection d'objet, suivi, classification et reconnaissance d'événements. Dans le cas spécifique de la reconnaissance d'événements, et après avoir essayé diverses fonctions de correspondance, les auteurs ont constaté que le choix des fonctions de correspondance entre séquences temporelles n'affectait pas beaucoup les résultats de l'évaluation. Par conséquent, dans l'évaluation du jeu de données ETISEO, les auteurs évaluent les intervalles de temps de détection par rapport à la vérité à partir du coefficient de Dice. Soit RD un intervalle pendant lequel un événement doit être détecté et C, l'intervalle pendant lequel l'événement est détecté. Le coefficient de Dice, D1, est donné par :

$$D1 = 2 . \frac{card(RD \cap C)}{card(RD) + card(C)}$$

Nous pouvons retrouver des métriques équivalentes dans [226], PETS 2009 [210]. Comme il n'y a pas de définition d'une intrusion, cette métrique permet de mesurer la qualité globale de la classification entre les images d'alarme et les autres images de la séquence mais ne permet pas réellement de déterminer si un algorithme a fait une ou plusieurs omissions. A noter toutefois que cette métrique a été proposé en lien avec un jeu de données particulier dans lequel les séquences d'images relativement courte ne comportaient que très peu d'événements voire qu'un seul. Dans ces conditions, cette métrique est tout à fait pertinente.

Dans les jeux de données où la durée des vidéos commence à être suffisamment importante pour contenir plusieurs événements d'intrusions différents espacés dans le temps, l'évaluation peut être basée sur la matrice de confusion. Il est alors nécessaire de définir les conditions permettant de déterminer les « vrais positifs », « faux positifs » et « faux négatifs ». Les « vrais négatifs » ne sont généralement pas déterminés parce qu'ils ne rentrent pas dans le calcul du rappel, de la précision et du F-score. Dans le cas de l'évaluation du jeu de donnée VIRAT [225] comme TRECvid [223], les valeurs de la matrice de confusion sont calculées à partir des intervalles temporel entre la vérité terrain et la détection. Un « vrai positif » est comptabilisé lorsque l'intersection entre un intervalle de la vérité et un intervalle de détection divisé par l'intervalle de la vérité est supérieur à un seuil. Les « vrais positifs » sont comptabilisés par rapport à la vérité terrain et chaque intervalle de la vérité ne contribue au plus qu'à un seul « vrai positif ». Un faux négatif est comptabilisé lorsque aucun intervalle de détection ne

contribue à un « vrai positif » pour un intervalle de la vérité donné. Enfin un « faux positif » est comptabilisé lorsqu'un intervalle de détection n'intersecte aucun intervalle de la vérité. Les mesures de rappel et de précision qui sont ensuite calculées permettent d'évaluer la qualité de la détection et la pertinence de la détection ainsi que le nombre de non-détections. Toutefois, il reste un point important que ces métriques ne permettent pas d'évaluer qui est le délai entre l'apparition d'un événement et sa détection.

Dans le cadre du projet I-LIDS, la définition de la détection d'intrusion est plus encadrée et répond à un cas d'usage précis. En effet, le gouvernement britannique impose que les systèmes d'analyse vidéo répondent à un certain niveau performance pour pouvoir répondre aux appels d'offres des marchés publics. Le protocole de test est clairement défini. L'information d'alarme donnée par le système en cours d'évaluation se limite à une sortie relais. L'état du relais, ouvert ou fermé, indique un état d'alarme. Les systèmes ont dix secondes pour signaler un état d'alarme (changement d'état du relais) après qu'un événement d'alarme apparait dans les images d'évaluation. Pendant ce délai, seul le premier changement d'état est considéré. Si d'autres alarmes sont générées pendant ce laps de temps, elles sont ignorées. Après cette fenêtre de dix secondes, toute autre alarme signalée est considérée comme un « faux positif », c'est-à-dire que le système ne doit pas continuer à envoyer des notifications d'alarmes pendant la durée des événements d'alarme. Si une succession rapide de fausses alarmes est notifiée, une seule fausse alarme est enregistrée toutes les cinq secondes. Si une alarme n'est pas notifiée pendant ce délai de 10 secondes, l'événement est alors considéré comme un « faux négatif ». Il est également précisé que toutes les notifications d'alarme signalées dans les cinq premières minutes de chaque nouvelle séquence sont ignorée afin de permettre au système en cours d'évaluation de se stabiliser. Un exemple d'une séquence d'état d'alarmes est donné ci-après (Figure 114).

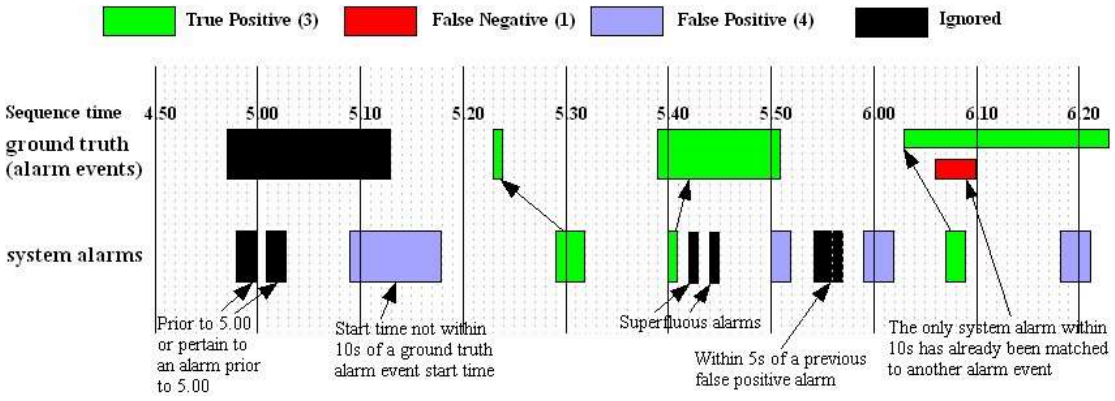


Figure 114 : exemple de séquence d'états d'alarme pour l'évaluation du jeu I-LIDS

Ce protocole ainsi défini permet de comptabiliser sur l'ensemble des séquences de test le nombre de « vrais positifs », « faux positifs » et « faux négatifs », et de calculer les mesures de rappel Rp et de précision Pr. La note finale est une combinaison de ces deux mesures appelée F1 définie ainsi :

$$F_1 = \frac{(\alpha + 1) \cdot Rp \cdot Pr}{Rp + \alpha \cdot Pr}$$

où α est une pondération du rappel par rapport à la précision et défini pour chaque scénario. Cette pondération détermine l'influence du taux de détection (rappel) par rapport à celui du taux de fausses alarmes sur la valeur de la F1. Dans les faits, la valeur de α dépend de l'usage du système et plus précisément, du traitement de l'alarme. Si une levée de doute rapide peut être faite alors il est possible de tolérer quelques fausses alarmes afin de privilégier le taux de détections. Si, à chaque notification d'alarme, tous les accès sont verrouillés, l'électricité coupée, et que la procédure de redémarrage nécessite un quart d'heure de manipulations (c'est un cas réel que nous avons eu à traiter), il est évident qu'une attention particulière sera donnée à la limitation des fausses alarmes.

4.3 Notre système d'évaluation

Dans le cadre de notre activité, nous avons mis en place une procédure pour l'évaluation des algorithmes de détection d'intrusions. Cette procédure est définie pour les étapes intermédiaires du processus de détection telles que le masque de mouvements et le suivi. Pour l'étape finale, cette détection, qui consiste à notifier une alarme, nous nous appuyons tout d'abord sur la définition d'un événement d'intrusion telle que nous l'avons présentée au chapitre III. De cette définition, nous avons défini une séquence d'événements. Pour rappel, une séquence d'intrusions est l'intervalle de temps maximum encadrant une succession d'événements d'intrusion :

$$IS = [t_{bs}, t_{es}], \{IE(t) = 1\}_{t=t_{bs}}^{t_{es}} \wedge IE(t_{bs-1}) = 0 \wedge IE(t_{es+1}) = 0$$

Avec $t_{bs} \in \mathcal{T}, t_{bs} > t_{start}$ et $t_{es} \in \mathcal{T}, t_{es} < t_{stop}$, où t_{start} et t_{stop} sont respectivement les dates de début et de fin de la vidéo. Une vidéo contient un ensemble $\mathcal{S}$ de séquences d'intrusion. La vérité terrain se construit également sur la base de cette définition. La vérité terrain, de même que le résultat d'un algorithme de détection d'intrusions, peut se représenter graphiquement sous la forme d'intervalles ou de chronogramme comme présenté sur la figure suivante (Figure 115).

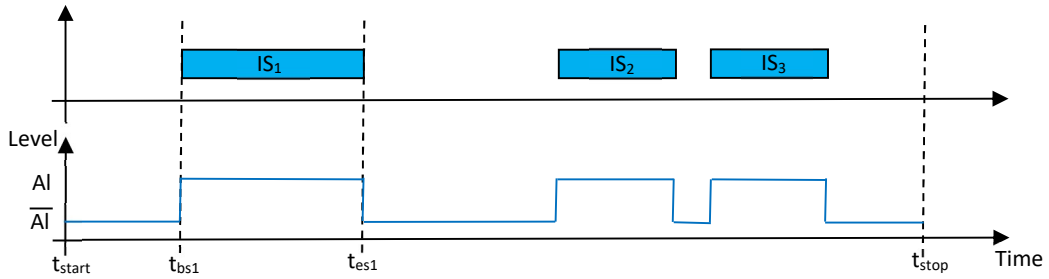


Figure 115 : exemple de représentation graphique d'un ensemble de séquences d'intrusions sous forme d'intervalles (en haut) et de chronogramme en bas.

La vérité terrain se compose donc d'un ensemble d'intervalle $\mathcal{S}_g$ labellisés par un ou plusieurs opérateurs humains ou par tout autre moyen ou ensemble de moyens. Le système de détections à évaluer peut fournir soit un événement impulsionnel indiquant le début de l'intrusion, soit un état actif pendant toute la durée de l'intrusion et inactif sinon, soit une impulsion pour le début

et une impulsion pour la fin. Dans ce dernier cas, nous considérons que l'information est de type intervalle puisque nous avons la date du début et de fin de l'intrusion.

4.3.1 Notification de l'alarme

Nous souhaitons tout d'abord vérifier que le système détecte, dans un délai raisonnable, le début d'une séquence d'intrusion. Nous nous plaçons dans un cas similaire à l'évaluation I-Lids avec toutefois une particularité liée au fait, que dans la phase d'exploitation, la notification d'alarme est complétée par une séquence d'images qui est envoyée à un télésurveilleur. Nous imposons que cette séquence de quelques images comporte la première seconde de l'intrusion.

Nous introduisons tout d'abord deux paramètres à notre évaluation, T_{pre} et T_{post}. Ces deux paramètres correspondent à une tolérance ajoutée, dans la vérité terrain, respectivement en début et fin de chaque séquence d'intrusion. Ces deux paramètres ont des valeurs très faibles, généralement inférieures à une seconde, et ont deux intérêts. Le premier est qu'ils permettent de prendre en compte l'incertitude de l'étiquetage. Il est en effet très difficile de définir avec précision sur quelle image d'une vidéo débute ou finie une intrusion. Le deuxième intérêt est que cela permet de fusionner deux séquences très proches l'une de l'autre. Deux séquences S_n et S_{n+1} sont fusionnées si :

$$t_{es,n} + T_{post} \geq t_{bs,n+1} + T_{pre}$$

De façon marginale, l'intérêt de fusionner les séquences est que cela permet également de prendre en compte l'incertitude de l'étiquetage, à plus forte raison lorsque l'étiquetage n'est pas réalisé par un opérateur. L'intérêt ici est plutôt d'éviter de comptabiliser deux intrusions lorsqu'il y a un élément de la scène qui provoque des occultations passagères (Figure 116).

Figure 116 : exemple de situation où la fusion de deux séquences est envisageable. La personne correctement détectée sur l'image de gauche et occultée sur l'image du milieu puis de nouveau détectée sur l'image de droite.

Le risque de cette approche est de fusionner deux événements distincts. Cependant, nous considérons que la fréquence d'apparition des séquences d'intrusions, tout comme leur durée, sont très faible au regard de la taille de la vidéo. La probabilité de fusionner deux événements distincts est donc très faible. Les deux paramètres T_{pre} et T_{post} sont fixés pour une vidéo donnée et n'ont pas vocation à être modifiés en fonction des algorithmes à tester.

Nous introduisons également le paramètre T_d qui correspond au délai maximum de détection à partir du début de la séquence d'intrusions. Ce délai pourra être modifié de façon à évaluer la capacité de l'algorithme à détecter rapidement une intrusion. Dans l'idéal, nous aimerions que T_d tende vers zéro. En pratique, la fréquence d'analyse est de cinq images par seconde et il est

d'usage que nous attendions la confirmation d'une intrusion sur au moins trois images. En conséquence nous pouvons déjà affirmer que nos algorithmes ne seront pas compétitifs si T_d est inférieur à 600 ms.

La première étape de l'évaluation consiste donc à redéfinir la vérité terrain en fonction de ces trois paramètres. Un exemple est donné sur la figure suivante (Figure 117) où nous avons trois séquences d'intrusions labellisées (Figure 117 : a). Nous pouvons observer que sur la deuxième et la troisième séquences, il y a une intersection entre le T_{post} de la deuxième et le T_{pre} de la troisième. Ces deux séquences sont alors fusionnées en une seule (Figure 117 : b). L'évaluation de l'algorithme se fait sur la vérité terrain ainsi redéfinie. Typiquement lorsque l'analyse est réalisée à la fréquence de cinq images par seconde, T_{pre} est fixé à 400 ms, ce qui correspond à deux images et T_{post} est fixé à 2 s, soit dix images.

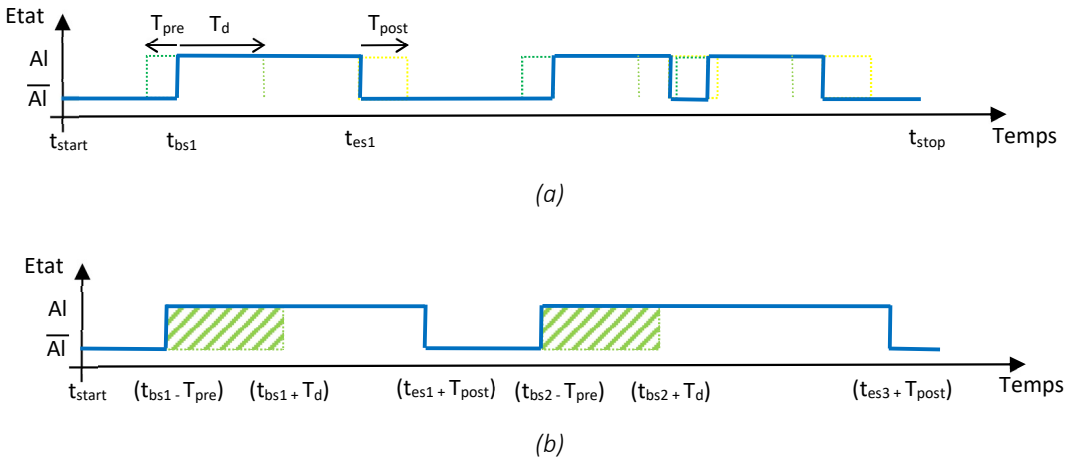


Figure 117 : exemple de vérité terrain d'une vidéo. La figure (a) correspond à la vérité terrain telle qu'elle a été labellisée. Sur la première séquence d'intrusions nous représentons les paramètres T_{pre}, T_{Post} et T_d. La figure (b) correspond à la vérité terrain telle qu'elle est utilisée par l'évaluateur, avec les décalages liés aux paramètres T_{pre}, T_{Post}. Les zones hachurées correspondent aux zones où une notification d'alarme est attendue.

Soit $\mathcal{S}_g = \{IS_{g,i}\}_{i\in[1:nbg]}$, un ensemble de séquences correspondant à la vérité terrain où nbg est le nombre de séquences d'intrusion labellisées et $\mathcal{S}_h = \{IS_{h,j}\}_{j\in[1:nbh]}$, un ensemble de séquences détectées par un algorithme où nbh est le nombre de séquences d'intrusion détectées. Nous comptabilisons les « vrais positifs » (TP), les « faux positifs » (FP) et les « faux négatifs » (FN) de la façon suivante :

- TP : pour une séquence d'intrusions de la vérité $SI_{g,i}$ donnée, un TP et un seul est comptabilisé si $\exists\, IS_{h,j}, t_{bs_{g,i}} - T_{pre} \leq t_{bs_{h,j}} \leq t_{bs_{g,i}} + T_d$
- FP : pour une séquence d'intrusions de la vérité $SI_{g,i}$ donnée, un FP est comptabilisé si $\nexists\, IS_{h,j}, t_{bs_{g,i}} - T_{pre} \leq t_{bs_{h,j}} \leq t_{bs_{g,i}} + T_d$
- FN : pour une séquence d'intrusions détectée $SI_{h,j}$ donnée, un FN est comptabilisé si $\nexists\, IS_{g,i}, t_{bs_{g,i}} - T_{pre} \leq t_{bs_{h,j}} \leq \max(t_{es_{g,i}} + T_{post}, t_{bs_{g,i}} + T_d)$

Nous pouvons remarquer que nous ne nous intéressons ici qu'au front montant de chaque détection. Par ailleurs si, pour une séquence d'intrusions de la vérité $SI_{g,i}$ donnée, $t_{es_{g,i}} + T_{post} < t_{bs_{g,i}} + T_d$, une détection peut être comptabilisée comme un vrai positif alors qu'il n'y a plus d'intrusion dans la vérité. Nous pouvons également remarquer que si $t_{es_{g,i}} + T_{post} > t_{bs_{g,i}} + T_d$, les fronts montants des détections qui pourraient se produire dans l'intervalle $[t_{bs_{g,i}} + T_d, t_{es_{g,i}} + T_{post}]$ sont ignorés.

A partir de ces données, nous calculons les métriques standards de précision, rappel et F-score. Cette première série de mesures nous permet d'évaluer la capacité des algorithmes à détecter le début de chaque intrusion.

4.3.2 Qualité de la détection

L'évaluation de la notification d'alarme permet de vérifier que le système est en capacité de fournir les premières images d'une intrusion, de détecter toutes les intrusions et de ne pas émettre de fausses alarmes. Par contre, le protocole que nous venons de présenter ne permet pas de s'assurer que le système est capable de maintenir une alerte pendant la totalité d'une intrusion. Pour cela, nous proposons d'utiliser une mesure de confiance que nous exploitons dans nos travaux sur les dépendances temporelles [9]. Cette mesure de confiance entre deux ensembles de séquence $\mathcal{S}_g$et $\mathcal{S}_h$notée $conf(\mathcal{S}_g \rightarrow \mathcal{S}_h)$ est donnée par :

$$conf(\mathcal{S}_g \rightarrow \mathcal{S}_h) = \frac{len(\mathcal{S}_g \cap \mathcal{S}_h)}{len(\mathcal{S}_g)}$$

où $len(\mathcal{S}_g)$, longueur de $\mathcal{S}_g$, est la somme de la durée des intervalles de $\mathcal{S}_g$ et $len(\mathcal{S}_g \cap \mathcal{S}_h)$ est la longueur de l'intersection de $\mathcal{S}_g$ et de $\mathcal{S}_h$. Dans cette définition nous utilisons traditionnellement la vérité terrain « brute » sans les transformations du protocole précédent mais avec le risque d'obtenir une confiance un peu supérieure pour un algorithme qui maintiendrait une alarme plus longtemps que nécessaire.

5 Conclusion

L'évaluation est une étape indispensable de tout développement et doit être étudié avec le plus grand soin afin de s'assurer que le résultat de cette évaluation est le plus pertinent possible. Comme nous l'avons vu, la qualité et la justesse du jeu de données est primordiale. Le cas de la vidéo-protection, tel que nous l'avons défini, à ceci de particulier que lorsqu'un système est déployé, il y a une forte probabilité pour qu'il n'y ait aucune réelle détection pendant toute la durée de l'exploitation. Il est donc important, au cours de l'évaluation, de s'assurer que le système soit en capacité de détecter une intrusion. Cette première vérification est relativement simple à réaliser. Cependant, il est également nécessaire de vérifier que le système est robuste à une quantité importante de variations et de mouvements intempestifs. Cette deuxième vérification est plus délicate à réaliser avec un jeu de données relativement réduit. Le jeu de données doit donc être suffisamment représentatif. La qualité et la justesse de la référence sont tout aussi important.

Un soin doit également être apporté aux métriques utilisées. Ces métriques doivent être en adéquation avec l'objectif visée. Comme nous l'avons vu, nous avons introduit la métrique D-score pour évaluer les maques de mouvements. Cette métrique a été proposée pour répondre à un besoin particulier lié au reste de la chaine de traitement, pour laquelle un soin particulier devait être apporté sur les contours des objets alors que la présence de blobs parasites pouvait facilement être supprimer par la suite. De la même manière, le cadre que nous proposons pour l'évaluation des intrusions est dicté par l'exploitation qui est faite du résultat de la détection.

VII – Conclusion et Perspectives

Dans ce manuscrit, nous avons présenté une vue globale de la détection d'intrusions basée sur l'analyse de séquences vidéo et plus généralement, de la détection et du suivi d'objets en mouvement. Dans le cadre de notre activité, nous avons travaillé et proposé des solutions pour répondre aux défis des différentes étapes de l'algorithme, et tout particulièrement concernant la modélisation de l'arrière-plan et le suivi. A ce jour, l'essentiel de notre travail est basé sur une analyse que nous pouvons qualifier de « descriptive » par rapport à une analyse « prédictive ».

L'analyse descriptive consiste à définir un modèle mathématique compréhensible qui décrit le phénomène que nous voulons observer. Il s'agit de recueillir des données sur un processus, de formuler des hypothèses sur les modèles et de valider ces hypothèses en comparant le résultat des modèles descriptifs avec le résultat réel [227]. La production de tels modèles est toutefois difficile et incomplète parce qu'il existe toujours un risque de variables ou de phénomènes que les scientifiques et les ingénieurs négligent d'inclure en raison de l'ignorance ou de l'incapacité de comprendre certains phénomènes complexes, cachés ou non intuitifs [228].

L'analyse prédictive implique la découverte de règles qui sous-tendent un phénomène et forment un modèle prédictif qui minimise l'erreur entre le résultat réel et le résultat prévu compte tenu de tous les facteurs d'interférence possibles [227]. L'apprentissage automatique rejette le paradigme de la programmation descriptive traditionnelle où l'analyse des problèmes est remplacée par un processus de formation et où le système est alimenté par un grand nombre d'exemples connus qu'il apprend et utilise pour calculer de nouveaux modèles [228]. L'apprentissage profond est un sous ensemble de l'apprentissage automatique basé en grande partie, aujourd'hui, sur les réseaux neuronaux artificiels et dans le cas des images, sur les réseaux de neurones convolutionnels (CNN). Ce type de réseau atteint des performances inégalées dans la plupart des domaines d'applications de l'analyse de données. Comme nous l'avons rapidement présenté, un réseau de neurones est une cascade de filtres linéaires et de non-linéarités. Le but du réseau est de fournir une fonction f de régression ou de classification de données de grandes dimensions.

En 2012, notre vie a changé. On ne l'a pas vraiment vu venir. C'était un peu comme une marée sur une plage de l'ile d'Oléron. Au début, une marée significativement plus haute que les autres. Nous avons naïvement pensé que cela allait en rester là mais l'eau a continué de monter et à tout submergé. Notre ancien monde s'est écroulé. Fini le temps où l'on passait des mois à proposer et à expérimenter un nouveau descripteur, à mettre au point une modélisation des composantes statiques d'une scène vidéo, à extraire des caractéristiques pertinentes pour le suivi et la ré-identification, où encore à calibrer des caméras pour estimer une profondeur. Aujourd'hui, il suffit de mettre en cascade une centaine de couches de convolutions (opérations

arithmétique niveau primaire qui consiste à faire une somme de multiplication terme à terme), de les intercaler avec une couche non linéaire type « Rectified Linear Unit » (encore un bien grand mot pour une fonction qui consiste à garder toutes les valeurs positives et mettre à zéro les valeurs négative en s'affranchissant de la dérivée au point zéro) et de terminer par un réseau entièrement connecté. Pour l'apprentissage, rien de plus simple, vous prenez 15 millions d'images étiquetées manuellement que vous passez à votre réseau, vous attendez quelques heures ou un mois en fonction de vos capacités de calcul et vous êtes capable de détecter une tasse de thé (ou de café selon votre gout) sur le rebord de votre bureau. Un nouveau monde s'offre à nous.

Cependant, il y a eu une vie avant 2012 et plusieurs solutions ont été proposées pour extraire de l'information à partir d'une image ou d'une vidéo, avec une certaine efficacité. Il y a encore aujourd'hui des tâches où l'analyse « prédictive », basée les réseaux de neurones profonds, présente des performances très en deçà de ce que nous pouvons obtenir avec une analyse « descriptive ».

Toutefois, au regard des performances obtenues par l'apprentissage automatique et notamment les CNN, il y a là un champ de recherche que nous ne pouvons pas négliger et que nous considérons sérieusement. Le succès des CNN est attribué à leur capacité d'apprendre une représentation hiérarchique des données d'entrées brutes, sans s'appuyer sur des fonctionnalités conçues à la main. Au fur et à mesure que les entrées sont traitées par les couches réseau, le niveau d'abstraction des entités résultantes augmente. Les couches les moins profondes saisissent les informations locales tandis que les couches plus profondes utilisent des filtres dont les champs récepteurs sont beaucoup plus larges et capturent donc des informations sémantiques abstraites plus globales [229]

Bien que les CNN-2D soient particulièrement bien adaptés pour apprendre des représentations appropriées pour les tâches de reconnaissance et de détection d'images, ils ne sont pas capables de capturer l'information temporelle codée dans les images consécutives d'une vidéo [230]. Un noyau 3D peut alors être utilisé [231] pour extraire à la fois les caractéristiques spatiales et temporelles d'une vidéo en convoluant un volume formé en empilant des images temporellement contiguës de la vidéo. C'est pourquoi, nous souhaitons étudier la possibilité de faire apprendre de manière non-supervisée un réseau profond encodant des représentations pour les vidéos [232] à partir de vidéos non étiquetées. L'objectif est de capturer la nature spatio-temporelle des vidéos dans un modèle génératif efficace, et non de traiter indépendamment les dimensions spatiales (images) et la dimension temporelle. Au même titre que les premières couches des réseaux convolutionnels 2D encodent des descripteurs locaux spécialisés pour les images, nous souhaitons apprendre des descripteurs spatio-temporels permettant de modéliser les vidéos. Une architecture de type auto-encodeur, se prête bien dans le cadre de la vidéo où les données étiquetées sont plus rares, et particulièrement pour la détection d'anomalie [230][233]. Utilisé dans ce cadre particulier, un tel auto-encodeur 3D apprend, dans ses cartes de fonctionnalités, les représentations qui sont invariantes aux changements spatio-temporels. L'idée est d'apprendre les informations visuelles régulières (normales) à partir de séquences vidéo. L'intuition est alors que l'auto-encodeur ainsi formé est capable de reconstruire, avec une faible erreur, les signatures de mouvements fréquents

présents dans les vidéos, mais est incapable de reconstruire avec précision les mouvements rares. Dans [17], nous avons décidé de tester cette architecture sur notre problème de détection d'intrusions. Inspirée par les travaux de Nogas et *al.* [234] sur la détection de chutes, nous formons notre auto-encodeur sur des vidéos sans intrusion, afin d'apprendre la « normalité » et nous testons le modèle avec des vidéos sans intrusion et avec intrusions. Une intrusion est détectée lorsque nous avons une mauvaise reconstruction de la vidéo, c'est-à-dire une erreur de reconstruction élevée. Cependant, un système de détection d'intrusion basé sur la détection des « anormalités » n'est pas adapté au marché de la vidéoprotection. Par exemple, les chutes de neige sont très rares sur la côte d'azur, ce n'est pas pour autant qu'elles doivent déclencher une alarme. Au-delà de cet exemple un peu particulier, il n'en reste pas moins qu'une détection d'intrusion, comme nous l'avons spécifiée au chapitre III, obéit à un ensemble de facteurs : temporels, spatiaux, types d'objet, *etc.*, qui pris individuellement ne sont pas forcément rare mais qui collectivement le sont.

Les exigences du marché de la vidéoprotection sont telles, qu'elles ne tolèrent pas plus d'une fausse alarme par semaine par caméra. Avec une fréquence d'analyse classique de 5 images par secondes et en supposant que l'analyse est activée douze heures par jour, cela correspond à une erreur pour environ un million cinq cent mille images analysées. C'est le niveau de performance que nous atteignons actuellement avec nos algorithmes d'analyses descriptives. Pour gérer un problème aussi complexe que la détection d'intrusion, il n'existe pas aujourd'hui de solution basée uniquement sur l'apprentissage profond avec un taux d'erreur aussi faible. Il semble toutefois que l'analyse descriptive, telle que nous l'avons pratiquée jusqu'à présent, ait atteint une forme de limite. Il devient en effet de plus en plus difficile d'améliorer significativement les performances des algorithmes basée sur cette technologie. En attendant que l'analyse « prédictive » atteigne, voir dépasses, le niveau de l'analyse « descriptive », une approche intégrant le meilleur de ces deux mondes peut être une alternative intéressante.

Nous avons déjà expérimenté une solution hybride mêlant ces deux mondes. Il s'agissait de détecter, segmenter et suivre les objets à partir des algorithmes classiques que nous avons présentés dans ce manuscrit et basés sur une analyse descriptive puis de confirmer l'information d'alarme en vérifiant la classe d'appartenance de l'objet à l'aide d'un classifieur basé sur un réseau de neurone profond. Une architecture similaire est également proposée par Kim et al. [235]. Cette idée n'était pas nouvelle puisque nous avions déjà utilisé ce principe avec un arbre de décision [116]. Cependant, cette solution hybride n'est pas réellement satisfaisante. Certes, elle permet de limiter significativement le nombre de fausses alarmes mais elle ne permet pas d'améliorer la détection. Le réseau de neurones, ou le classifieur en général, étant utilisé pour filtrer les alarmes générées par l'algorithme descriptif. Par ailleurs, le paramétrage de cet algorithme descriptif doit toujours être réalisé par un expert. Un autre point négatif, est que le réseau de neurones utilisé pour l'inférence nécessite une certaine résolution d'image ; c'est-à-dire que pour déterminer la classe d'un objet, il est nécessaire que la taille apparente de l'objet dans l'image soit suffisante afin qu'il y ait suffisamment de détails. Les différents modèles que nous avons expérimentés (Resnet 101, SSD mobile net, Faster RCNN inception) ne nous ont pas permis descendre en deçà de 250 pixels (rectangle de 25x10) pour obtenir une classe fiable alors que nous obtenons de très bons résultats avec deux fois moins de pixels en utilisant une approche basée uniquement sur l'analyse classique. Le niveau de résolution est un paramètre

important lors du déploiement d'une système de vidéo-protection parce que cela conditionne le positionnement et le nombre de caméras à installer sur le site. Enfin, au niveau matériel, l'utilisation d'un GPU est nécessaire pour obtenir l'inférence du réseau avec un délai compatible avec le temps réel tel que l'avons défini dans le cas de la détection d'intrusions. Dans ces conditions, l'utilisation d'une telle architecture hybride, n'est pas réellement intéressante. Pour qu'elle le soit, il serait nécessaire que le gain soit particulièrement significatif.

C'est dans ce but que nous avons initié le travail de thèse de monsieur Devasish Lohani sur les auto-encodeur et sur la détection d'anomalie. Ce travail préparatoire nous permet de mieux comprendre les représentations spatio-temporelles ainsi que les architectures les plus à même de les capturer. Bien que la fonction de détection d'anomalies ne soit pas la plus adaptée à la vidéo-protection, le modèle utilisé, basé sur un auto-encodeur, peut être appris à partir d'un volume important de données faiblement supervisées. Une fois appris, les descripteurs des premières couches de l'étape de codage pourront être utilisées comme entrée d'un réseau discriminatif appris pour une tâche supervisée mais nécessitant moins de données labélisées.

Une autre piste que nous souhaitons explorer, dans le cas de la détection d'intrusions, consiste à utiliser la sortie d'un algorithme basé sur une analyse descriptive et paramétré par un expert comme résultats attendu d'un modèle prédictif. L'idée étant à terme que le modèle prédictif, une fois appris, ne nécessite plus la présence d'un expert lors du déploiement.

A tous ces aspects pratiques, il y a un tout de même un frein à l'utilisation d'un système basé uniquement sur une architecture de type « apprentissage profond ». En effet, si dans beaucoup de domaines, les réseaux profonds montrent plus d'efficacité, leur compréhension est plus difficile et reste un problème ouvert, notamment sur l'interprétation des résultats de mauvaises classifications. Comme nous l'avons vu, les fausses alarmes ont un coût économique important, mais les non-détections sont encore bien moins tolérées. Lorsque cela survient, il est donc important d'en comprendre la raison, d'être en mesure d'en expliquer l'origine et éventuellement de mettre en place des actions correctives. Il faut, pour cela, assurer la stabilité des prédictions aux déformations locales des données et expliquer les cas de mauvaises classifications.

Un ensemble de travaux théoriques résumés dans [236], montre qu'un réseau multicouches initialisé sur des fonctions ondelettes, encodant des invariants (tels que la translation, l'échelle, la rotation ou des groupes plus généraux de transformations), fournit des contractions (au sens de Lipschitz) multi-échelles, permettant de réduire la dimension et la variabilité des données sans perdre en séparabilité des classes et capacité d'estimation de la fonction de régression. Cependant la grande majorité de ces travaux ne s'appliquent qu'au cas d'un réseau déjà initialisé avec des filtres d'ondelettes, et non à un réseau avec les poids des filtres appris sur un ensemble d'entrainement.

Poursuivant cette réflexion sur l'invariance aux groupes de transformations, les auteurs [237] utilisent l'outil mathématique des noyaux pour caractériser la stabilité et la capacité de généralisation des réseaux convolutionnels. Ce travail est effectué dans un cadre d'apprentissage classique, où les filtres sont appris à partir de la donnée. Notamment, il est

montré que la notion de norme permet de contrôler la stabilité et la capacité de généralisation du réseau, et que ces conditions de stabilité sont reliées aux paramètres architecturaux du réseau, telle que la profondeur (nombre de couches), la taille des filtres, le facteur de sous-échantillonnage ou la fonction d'activation. Si la norme est explicitement régularisée lors de l'apprentissage d'un réseau à noyaux, ce n'est pas le cas des réseaux convolutionnels classiques appris via l'algorithme de rétro-propagation du gradient.

Ce point permettrait d'expliquer la sensibilité surprenante des réseaux convolutionnels profonds aux exemples antagonistes. Ces derniers sont des données d'entrainement sur lesquelles on ajoute intentionnellement un bruit non-aléatoire imperceptible afin de perturber arbitrairement le résultat de la classification [238]. Ces exemples sont intensivement étudiés, puisque leur existence tendrait à prouver que la variété (sous-espace non-linéaire) représentée par le réseau n'est pas régulière, et que les stratégies usuelles d'augmentation de données (déformations des données d'entrainement) ne permettent pas d'échantillonner correctement ce sous-espace, ce qui empêche sa généralisation. Certains travaux [239][240] étudient quels sont les paramètres du réseau qui sont sensibles à ces exemples : régularisation implicite, autre type de pénalisation, autre fonctionnelle de coût, normalisation versus orthogonalité, *etc.*, ou quels sont les liens avec les propriétés statistiques de sur-apprentissage ou de sur-généralisation des différents modes des distributions d'entrées [241].

Ceci nous amène à étudier les propriétés de stabilité des réseaux convolutionnels dans le cadre de l'apprentissage profond, et de la manière de contrôler cette stabilité couche par couche, ainsi que des liens avec les propriétés statistiques des données. En guise de conclusion de ce manuscrit, nous posons les trois questions suivantes :

1 - Les données vidéo sont usuellement regardées comme un cube, contenant 2 dimensions spatiales (images) et une dimension temporelle. Peut-on prouver que les convolutions 3D sont le groupe de symétries adaptées pour capturer les invariances de la donnée spatio-temporelle ?

Si oui, est-ce que ces invariances sont globales ou uniquement locales ? Autrement dit, est-ce qu'un réseau purement convolutif est suffisant, ou faut-il rajouter par-dessus un réseau récurrent (type LSTM) pour capturer les dépendances à long-terme ?

2 - Les 3 dimensions n'ont pas le même taux d'acquisition : N pixels selon l'axe horizontal ne représentent pas la même distance de déplacement de l'objet dans la scène que N pixels selon l'axe vertical. De plus, ces 2 quantités ne sont pas comparables à N échantillons de l'axe temporel, puisque n'étant pas de même nature. Ainsi, le déplacement d'un objet dans ce cube n'est pas isotrope.

Est-ce que des convolutions anisotropiques [242] seraient adaptées pour la donnée spatio-temporelle?

Ainsi, si l'on ajoute des contraintes sur les dimensions, peut-on obtenir des caractérisations plus fines ?

a) La dimension temporelle peut être relue *a posteriori* de manière non-causale (*ie*, du futur vers le passé). Mais, indépendamment de cette considération sur les traitements post-acquisition, la vidéo a enregistré les images d'un monde causal. Est-ce que le fait que l'axe temporel soit orienté a un impact ?

b) Les deux dimensions spatiales ne sont pas orientées : les objets peuvent se déplacer dans les deux directions de chaque dimension spatiale. Cependant, les deux axes spatiaux ne sont pas identiques pour autant. Un objet qui se déplace selon l'axe horizontal reste à distance constante de la caméra, alors qu'un objet qui se déplace selon l'axe vertical s'éloigne ou s'approche. Est-ce que cette différence a un impact ?

Dans les deux cas ci-dessus, quand on parle d'impact, il peut s'agir de considérations :

- théoriques, comme des restrictions sur le groupe de symétries ;
- ou pratiques : noyau 3D rectangulaire (par exemple, un noyau plus large que haut), valeurs du stride et/ou du pas non égales sur les 3 dimensions, convolution dilatées / « à trous » sur certaine dimension plutôt que d'autre, convolutions séparables, etc.

3 - Le taux d'acquisition temporel (en image/seconde) est un paramètre ayant une influence critique sur l'analyse vidéo. En effet, un taux trop élevé générerait des images très corrélées entre elles, alors qu'un taux trop faible donnerait des images sur lesquelles on ne pourrait plus faire de suivi d'objets.

Ainsi, le taux d'acquisition doit être ajusté au phénomène le plus rapide que nous souhaitons ne pas manquer. Ceci est similaire au théorème de Shannon qui stipule que la fréquence d'acquisition doit être au moins deux fois supérieure à la fréquence maximale à observer. La vitesse maximale d'un être humain est aux alentours de 45 km/h (Usian Bolt en vitesse de pointe). En fonction de la disposition de la caméra (hauteur, angle de vue, *etc.*), peut-on déterminer automatiquement le taux d'acquisition optimal, *ie* qui permet de ne pas manquer une intrusion, mais qui évite des calculs inutiles ?

VIII - Bibliographie

[1] L. Robinault, S. Bres, and S. Miguet, “Panoramic mosaicing optimization,” in *Proceedings - 14th International conference on Image Analysis and Processing, ICIAP 2007*, 2007, pp. 548–553.

[2] L. Robinault, S. Bres, and S. Miguet, “Real time foreground object detection using ptz camera,” in *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*, 2009, vol. 1.

[3] L. Robinault, I. Pop, and S. Miguet, “Self-calibration and control of a PTZ camera based on a spherical mirror,” in *6th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009*, 2009.

[4] T. Durand, X. He, I. Pop, and L. Robinault, “Utilizing deep object detector for video surveillance indexing and retrieval,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11296 LNCS, pp. 506–518.

[5] P. Lemaire, C. F. Crispim-Junior, L. Robinault, and L. Tougne, “Jitter-Free Registration for Unmanned Aerial Vehicle Videos,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019.

[6] S. Kamate and N. Yilmazer, “Application of Object Detection and Tracking Techniques for Unmanned Aerial Vehicles,” in *Procedia Computer Science*, 2015.

[7] I. N. Thiang and H. M. Tun, “Vision-Based Object Tracking Algorithm With AR. Drone,” *Int. J. Sci. Technol. Res.*, vol. 4, no. 8, pp. 135–139, 2015.

[8] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza, “Gender recognition from face images with local WLD descriptor,” in *2012 19th International Conference on Systems, Signals and Image Processing, IWSSIP 2012*, 2012.

[9] A. El Ouassouli, “Discovering Complex Quantitative Dependencies between Interval-based State Streams,” INSA Lyon, 2020.

[10] M. Rogez, “Utilisation du contexte pour la détection et le suivi d’objets en vidéosurveillance,” Université Lumière Lyon2, 2015.

[11] C. Lallier, E. Reynaud, L. Robinault, and L. Tougne, “A testing framework for background subtraction algorithms comparison in intrusion detection context,” in *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2011*, 2011.

[12] M. Rogez, L. Tougne, and L. Robinault, “A prior-knowledge based casted shadows prediction model featuring OpenStreetMap data,” in *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications*, 2013, vol. 1.

[13] M. Rogez, L. Robinault, and L. Tougne, “A 3D tracker for ground-moving objects,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8888.

[14] A. El Ouassouli, L. Robinault, and V.-M. Scuturici, “Mining Quantitative Temporal Dependencies Between Interval-Based Streams,” in *DaWaK 2019. Lecture Notes in Computer Science, vol 11708*, 2019.

[15] Y. Pu *et al.*, “Variational autoencoder for deep learning of images, labels and captions,” in *Advances in Neural Information Processing Systems*, 2016.

[16] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, “Generative Adversarial Nets,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, 2017.

[17] D. Lohani, C. Crispim-Junior, Q. Barthélemy, S. Bertrand, L. Robinault, and L. Tougne, “Spatio-Temporal Convolutional Autoencoders for perimeter intrusion detection by video protection,” pp. 1–15.

[18] N. Thome, “Représentations hiérarchiques et discriminantes pour la reconnaissance des formes, l’identification des personnes et l’analyse des mouvements dans les séquences d’images,” Université Lumière Lyon2, 2007.

[19] N. Thome, A. Vacavant, L. Robinault, and S. Miguet, “A cognitive and video-based approach for multinational License Plate Recognition,” *Mach. Vis. Appl.*, vol. 22, no. 2, 2011.

[20] I. Pop, “Détection des événements rares dans des vidéos,” INSA Lyon, 2010.

[21] A. Vacavant, L. Robinault, S. Miguet, C. Poppe, and R. Van De Walle, “Adaptive background subtraction in H.264/AVC bitstreams based on macroblock sizes,” in *VISAPP 2011 - Proceedings of the International Conference on Computer Vision Theory and Application*, 2011.

[22] P. Viola and M. Jones, “Robust Real-time Object Detection,” in *Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing, and Sampling*, 2001.

[23] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *Int. J. Comput. Vis.*, vol. 57, pp. 137–154, 2004.

[24] C. Garcia and M. Delakis, “A neural architecture for fast and robust face detection,” in *Proceedings - International Conference on Pattern Recognition*, 2002.

[25] J. Chen *et al.*, “WLD: A robust local image descriptor,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1705–1720, 2010.

[26] I. Hersey and P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. NY: Dover Publications, Inc, 1966.

[27] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, 2000.

[28] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric View-point*. MIT-Press, 1993.

[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[30] B. Baland, *Optique géométrique : Imagerie et instruments*. Broché, 2007.

[31] E. Fossum, “CMOS lmagc Sensors : Electronic Camera on A Chip,” *IEEE Int. Electron Devices Meet. Tech. Dig.*, 1995.

[32] D. Coeurjolly, A. Montanvert, and J.-M. Chassery, *Géométrie discrète et images numériques*, Collection. Hermès - Lavoisier, 2007.

[33] J. Shi and J. Malik, “Motion segmentation and tracking using normalized cuts,” in *Proceedings of the IEEE International Conference on Computer Vision*, 1998.

[34] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Tech. Appl. Image Underst.*, 1981.

[35] B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision.,” *Int. Jt. Conf. Artif. Intell.*, 1981.

[36] H. Spies, H. Scharr, H. Spies, and H. S. De, “Accurate Optical Flow in Noisy Image Sequences,” in *ICCV 2001*, 2001, no. September, pp. 587–592.

[37] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek, “Nonlinear structure tensors,” *Image Vis. Comput.*, 2006.

[38] P. Ochs, J. Malik, and T. Brox, “Segmentation of moving objects by long term video analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, 2014.

[39] L. Chen, J. Shen, W. Wang, and B. Ni, “Video Object Segmentation Via Dense Trajectories,” *IEEE Trans. Multimed.*, vol. 17, no. 12, pp. 2225–2234, 2015.

[40] D. J. Heeger, “Optical flow using spatiotemporal filters,” *Int. J. Comput. Vis.*, vol. 1, pp. 279–302, 1988.

[41] A. Spinei, D. Pellerin, and J. Hérault, “Spatiotemporal energy-based method for velocity estimation,” *Signal Processing*, vol. 65, no. 3, pp. 347–362, 1998.

[42] A. B. Torralba and J. Hérault, “An efficient neuromorphic analog network for motion estimation,” *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 46, no. 2, pp. 269–280, 1999.

[43] A. Dosovitskiy *et al.*, “FlowNet: Learning optical flow with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[44] X. Wu, W. Xu, N. Zhu, and Z. Yang, “A fast motion estimation algorithm for H.264,” in *2010 International Conference on Signal Acquisition and Processing, ICSAP 2010*, 2010.

[45] C. Solana-Cipres, G. Fernandez-Escribano, L. Rodriguez-Benitez, J. Moreno-Garcia, and L. Jimenez-Linares, “Real-time moving object segmentation in H.264 compressed domain based on approximate reasoning,” *Int. J. Approx. Reason.*, vol. 51, no. 1, pp. 99–114, 2009.

[46] C. Poppe, S. De Bruyne, T. Paridaens, P. Lambert, and R. Van de Walle, “Moving object detection in the H.264/AVC compressed domain for video surveillance applications,” *J. Vis. Commun. Image Represent.*, vol. 20, no. 6, pp. 428–437, 2009.

[47] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real-time foreground-background segmentation using codebook model,” *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[48] Y. Goya, T. Chateau, L. Malaterre, and L. Trassoudaine, “Vehicle trajectories evaluation by static video sensors,” in *Vehicle trajectories evaluation by static video sensors*, 2006.

[49] T. Wiegand, G. Sullivan, G. Bjontegaard, and G. Luthra, “Overview of the H.264/AVC video coding standard,” *EEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[50] M. Van Droogenbroeck and O. Paquot, “Background subtraction experiments and improvements for vibe,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, no. June, pp. 32–37.

[51] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Computer Science Review*, vol. 11–12. pp. 31–66, 2014.

[52] C. Stauffer and W. Grimson, “Adaptative background mixture models for real-time tracking,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 246–252, 1999.

[53] L. Dar Shyang, “Effective Gaussain Mixture Learning for Video Background Substraction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, p. 2005, 2005.

[54] P. KaewTraKulPong and R. Bowden, “An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection,” in *Video-Based Surveillance Systems*, 2011.

[55] H. Fradi and J. L. Dugelay, “Robust foreground segmentation using improved Gaussian mixture model and optical flow,” *2012 Int. Conf. Informatics, Electron. Vision, ICIEV 2012*, pp. 248–253, 2012.

[56] A. Doshi and M. Trivedi, “‘Hybrid cone-cylinder’ codebook model for foreground detection with shadow and highlight suppression,” in *Proceedings - IEEE International Conference on Video and Signal Based Surveillance 2006, AVSS 2006*, 2006.

[57] A. Ilyas, M. Scuturici, and S. Miguet, “Real time foreground-background segmentation using a modified codebook model,” in *6th IEEE International Conference on Advanced*

Video and Signal Based Surveillance, AVSS 2009, 2009.

[58] X. Cheng, T. Zheng, and L. Renfa, “A fast motion detection method based on improved codebook model,” *J. Comput. Dev.*, vol. 47, pp. 2149–2156, 2010.

[59] M. Wu and X. Peng, “Spatio-temporal context for codebook-based dynamic background subtraction,” *AEU - Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, 2010.

[60] Y. Li, F. Chen, W. Xu, and Y. Du, “Gaussian-Based Codebook Model for Video Background Subtraction,” in *Advances in Natural Computation*, 2006, pp. 762–765.

[61] Y. Wu, D. Zeng, and H. Li, “Layered video objects detection based on LBP and codebook,” in *Proceedings of the 1st International Workshop on Education Technology and Computer Science, ETCS 2009*, 2009.

[62] M. A. Mousse, E. C. Ezin, and C. Motamed, “Foreground-background segmentation based on codebook and edge detector,” in *Proceedings - 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2014*, 2015, pp. 119–124.

[63] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, “A self-adjusting approach to change detection based on background word consensus,” *Proc. - 2015 IEEE Winter Conf. Appl. Comput. Vision, WACV 2015*, no. January, pp. 990–997, 2015.

[64] G. A. Bilodeau, J. P. Jodoin, and N. Saunier, “Change detection in feature space using local binary similarity patterns,” in *Proceedings - 2013 International Conference on Computer and Robot Vision, CRV 2013*, 2013.

[65] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Deep neural network concepts for background subtraction: A systematic review and comparative evaluation,” *Neural Networks*, vol. 117, pp. 8–66, 2019.

[66] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric Model for Background Subtraction,” in *ECCV 2000. Lecture Notes in Computer Science*, 2000.

[67] O. Barnich and M. Van Droogenbroeck, “ViBe: A universal background subtraction algorithm for video sequences,” *IEEE Trans. Image Process.*, vol. 20, pp. 1709–1724, 2011.

[68] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, “Flexible background subtraction with self-balanced local sensitivity,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[69] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: principles and practice of background maintenance,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

[70] S. Brutzer, B. Höferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[71] T. Bouwmans, “Recent Advanced Statistical Background Modeling for Foreground Detection - A Systematic Survey,” *Recent Patents Comput. Sci.*, vol. 4, pp. 147–176, 2011.

[72] D. Greenhill, J. Renno, J. Orwell, and G. A. Jones, “Learning the Semantic Landscape: Embedding scene knowledge in object tracking,” *Real-Time Imaging*, vol. 11, no. 3, pp. 186–203, 2005.

[73] A. Sanin, C. Sanderson, and B. C. Lovell, “Shadow detection: A survey and comparative evaluation of recent methods,” *Pattern Recognit.*, vol. 45, no. 4, pp. 1684–1695, 2012.

[74] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, “Improving shadow suppression in moving object detection with HSV color information,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2001.

[75] J. F. Lalonde, A. A. Efros, and S. G. Narasimhan, “Estimating natural illumination from a single outdoor image,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[76] S. Nadimi and B. Bhanu, “Physical models for moving shadow and object detection in video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1079–1087, 2004.

[77] J. Bin Huang and C. S. Chen, “Moving cast shadow detection using Physics-based features,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009.

[78] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, “On the removal of shadows from images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 55–68, 2006.

[79] M. J. Leotta and J. L. Mundy, “Learning background and shadow appearance with 3-D vehicle models,” in *BMVC 2006 - Proceedings of the British Machine Vision Conference 2006*, 2006.

[80] A. Leone and C. Distante, “Shadow detection for moving objects based on texture analysis,” *Pattern Recognit.*, vol. 40, no. 4, pp. 1222–1233, 2007.

[81] B. Jackson, R. Bodor, and N. Papanikolopoulos, “Learning static occlusions from interactions with moving figures,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[82] J. Martín, D. Vázquez, D. Gerónimo, and A. M. López, “Learning appearance in virtual scenarios for pedestrian detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[83] B. Kaneva, A. Torralba, and W. T. Freeman, “Evaluation of image features using a photorealistic virtual world,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[84] J. Blinn, “Me and my (fake) shadow,” *IEEE Comput. Graph. Appl.*, vol. 8, pp. 82–86, 1988.

[85] J. J. Michalsky, “The Astronomical Almanac’s algorithm for approximate solar position (1950-2050),” *Sol. Energy*, vol. 40, no. 3, pp. 227–235, 1988.

[86] M. Blanco-Muriel, D. C. Alarcón-Padilla, T. López-Moratalla, and M. Lara-Coira, “Computing the solar vector,” *Sol. Energy*, vol. 70, no. 5, pp. 431–441, 2001.

[87] I. Reda and A. Andreas, “Solar position algorithm for solar radiation applications,” *Sol. Energy*, vol. 76, no. 5, pp. 577–589, 2004.

[88] R. Grena, “An algorithm for the computation of the solar position,” *Sol. Energy*, vol. 82, no. 5, pp. 462–470, 2008.

[89] R. Grena, “Five new algorithms for the computation of sun position from 2010 to 2110,” *Sol. Energy*, vol. 86, no. 5, pp. 1323–1337, 2012.

[90] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012.

[91] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object Detection with Deep Learning: A Review,” *IEEE Transactions on Neural Networks and Learning Systems*. 2019.

[92] M. H. Yang, D. J. Kriegman, and N. Ahuja, “Detecting faces in images: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, 2002.

[93] S. Zafeiriou, C. Zhang, and Z. Zhang, “A survey on face detection in the wild: Past, present and future,” *Comput. Vis. Image Underst.*, vol. 138, pp. 1–24, 2015.

[94] R. Lienhart and J. Maydt, “An extended set of Haar-like features for rapid object detection,” in *IEEE International Conference on Image Processing*, 2002.

[95] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, “Statistical learning of multi-view face detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002.

[96] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Gray scale and rotation invariant texture classification with local binary patterns,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000.

[97] P. Wang and Q. Ji, “Learning discriminant features for multi-view face and eye detection,” in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.

[98] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.

[99] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vis.*, vol. 61, pp. 55–79, 2005.

[100] C. Zhang and Z. Zhang, “Improving multiview face detection with multi-task deep convolutional neural networks,” in *2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014*, 2014.

[101] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *Int. J. Comput. Vis.*, vol. 104, pp. 154–171, 2013.

[102] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014.

[103] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[104] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[105] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” pp. 1–14, 2014.

[106] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[107] X. Sun, P. Wu, and S. C. H. Hoi, “Face detection using deep learning: An improved faster RCNN approach,” *Neurocomputing*, vol. 299, no. 19, pp. 42–50, 2018.

[108] S. Wan, Z. Chen, T. Zhang, B. Zhang, and K. Wong, “Bootstrapping Face Detection with Hard Negative Examples,” pp. 1–7, 2016.

[109] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[110] W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[111] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[112] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “InceptionNet/GoogLeNet - Going Deeper with Convolutions,” *Conf. Comput. Vis. Pattern Recognit.*, vol. 91, no. 8, pp. 2322–2330, 2016.

[113] J. Li and all, “DSFD: Dual Shot Face Detector,” in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5055–5064.

[114] J. Huang *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[115] X. Tang, D. K. Du, Z. He, and J. Liu, “PyramidBox: A context-assisted single shot face detector,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[116] L. Robinault, “Mosaïque d’image multi-résolution et applications,” Lyon 2, 2009.

[117] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, 2006.

[118] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2014.

[119] H. T. Nguyen and A. W. M. Smeulders, “Robust tracking using foreground-background texture discrimination,” *Int. J. Comput. Vis.*, vol. 69, pp. 277–293, 2006.

[120] M. Fiaz, A. Mahmood, and S. K. Jung, “Tracking Noisy Targets: A Review of Recent Object Tracking Approaches,” 2018.

[121] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, “Robust visual tracking based on incremental tensor subspace learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[122] G. Silveira and E. Malis, “Real-time visual tracking under arbitrary illumination changes,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[123] H. Alismail, B. Browning, and S. Lucey, “Robust tracking in low light and sudden illumination changes,” in *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 2016.

[124] L. Zhang, J. Varadarajan, P. N. Suganthan, N. Ahuja, and P. Moulin, “Robust visual tracking using oblique random forests,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[125] M. J. Black and A. D. Jepson, “EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation,” *Int. J. Comput. Vis.*, vol. 26, pp. 63–84, 1998.

[126] J. W. Davis and A. F. Bobick, “The representation and recognition of action using temporal templates,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[127] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: Real-time surveillance of people and their activities,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, 2000.

[128] T. Zhao and R. Nevatia, “Tracking multiple humans in complex situations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1208–1221, 2004.

[129] A. Ilyas, M. Scuturici, and S. Miguet, “Inter-camera color calibration for object re-identification and tracking,” in *Proceedings of the 2010 International Conference of Soft Computing and Pattern Recognition, SoCPaR 2010*, 2010.

[130] F. Porikli, “Integral histogram: A fast way to extract histograms in Cartesian spaces,” in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.

[131] F. C. Crow, “Summed-area tables for texture mapping,” in *SIGGRAPH ’84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 1984, pp. 207–212.

[132] G. R. Bradski, “Real time face and object tracking as a component of a perceptual user interface,” in *Proceedings - 4th IEEE Workshop on Applications of Computer Vision, WACV 1998*, 1998.

[133] S. Birchfield, “Elliptical head tracking using intensity gradients and color histograms,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.

[134] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, 2003.

[135] R. T. Collins, “Mean-shift blob tracking through scale space,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[136] G. D. Hager, M. Dewan, and C. V. Stewart, “Multiple kernel tracking with SSD,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[137] T. L. Liu and H. T. Chen, “Real-Time Tracking Using Trust-Region Methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 3, pp. 397–402, 2004.

[138] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[139] K. Cannons and R. Wildes, “Spatiotemporal oriented energy features for visual tracking,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007.

[140] I. A. Iswanto and B. Li, “Visual Object Tracking Based on Mean-shift and Particle-Kalman Filter,” *Procedia Comput. Sci.*, vol. 116, pp. 587–595, 2017.

[141] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002.

[142] J. Ning, L. Zhang, D. Zhang, and C. Wu, “Robust object tracking using joint colour texture histogram,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1245–1263, 2009.

[143] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, “MOT16: A Benchmark for Multi-Object Tracking,” pp. 1–12, 2016.

[144] K. Meshgi and S. Ishii, “Expanding histogram of colors with gridding to improve tracking accuracy,” in *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015*, 2015.

[145] J. H. Yoon, C. R. Lee, M. H. Yang, and K. J. Yoon, “Online multi-object tracking via structural constraint event aggregation,” in *Proceedings of the IEEE Computer Society Conference*

on Computer Vision and Pattern Recognition, 2016.

[146] W. Zhong, H. Lu, and M. H. Yang, “Robust object tracking via sparsity-based collaborative model,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.

[147] M. J. Swain and D. H. Ballard, “Color indexing,” *Int. J. Comput. Vis.*, vol. 7, pp. 11–32, 1991.

[148] Y. Rubner, C. Tomasi, and L. J. Guibas, “Earth mover’s distance as a metric for image retrieval,” *Int. J. Comput. Vis.*, vol. 40, pp. 99–121, 2000.

[149] S.-H. Cha, “Comprehensive survey on distance/similarity measures between probability density functions,” *Int. J. Math. Model. Methods Appl. Sci.*, vol. 1, 2007.

[150] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Comput. Vis. Image Underst.*, vol. 63, no. 1, pp. 75–104, 1996.

[151] M. Irani, “Multi-frame optical flow estimation using subspace constraints,” in *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

[152] Y. Wu and J. Fan, “Contextual flow,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009.

[153] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, “PROST: Parallel robust online simple tracking,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[154] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of optical flow estimation with deep networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[155] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, “Unsupervised deep learning for optical flow estimation,” in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.

[156] M. Isard and A. Blake, “CONDENSATION - Conditional Density Propagation for Visual Tracking,” *Int. J. Comput. Vis.*, vol. 29, pp. 5–28, 1998.

[157] J. Maccormick and A. Blake, “Probabilistic exclusion principle for tracking multiple objects,” *Int. J. Comput. Vis.*, vol. 39, pp. 57–71, 2000.

[158] D. Cremers, “Dynamical statistical shape priors for level set-based tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1262–1273, 2006.

[159] X. Sun, H. Yao, and S. Zhang, “A novel supervised level set method for non-rigid object tracking,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[160] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1988.

[161] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional

density," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1996.

[162] K. Fukunaga and L. D. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[163] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[164] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2000.

[165] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[166] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces," in *Visual Information Processing 2003, Proceedings of the Pan-Sydney Area*, 2003.

[167] D. Exner, E. Bruns, D. Kurz, A. Grundhöfer, and O. Bimber, "Fast and robust CAMShift tracking," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, 2010.

[168] J. Yin, Y. Han, J. Li, and A. Cao, "Research on real-time object tracking by improved CamShift," in *Proceedings - 1st International Symposium on Computer Network and Multimedia Technology, CNMT 2009*, 2009.

[169] K. Abughalieh, W. Qadi, K. Melkon, B. Fakes, B. Sababha, and A. Al-Mousa, "A compact portable object tracking system," in *2014 5th International Conference on Information and Communication Systems, ICICS 2014*, 2014.

[170] J. Munkres, "Algorithms for the Assignment and Transportation Problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[171] F. Bourgeois and J. C. Lassalle, "An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices," *Commun. ACM*, vol. 14, no. 12, 1971.

[172] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[173] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automat. Contr.*, vol. 24, no. 6, pp. 843–854, 1979.

[174] H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density," in *Sixth International Conference of Information Fusion*, 2003, pp. 800–806.

[175] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[176] R. Di Lascio, P. Foggia, G. Percannella, A. Saggese, and M. Vento, “A real time algorithm for people tracking using contextual reasoning,” *Comput. Vis. Image Underst.*, vol. 117, no. 8, pp. 892–908, 2013.

[177] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.

[178] Y. Gu *et al.*, “Efficient Online Training Method for SiamFC Network in Object Tracking,” in *26th International Conference on Neural Information Processing*, 2019.

[179] P. Li, D. Wang, L. Wang, and H. Lu, “Deep visual tracking: Review and experimental comparison,” *Pattern Recognit.*, vol. 76, pp. 323–338, 2018.

[180] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung, “Handcrafted and deep trackers: Recent visual object tracking approaches and trends,” *ACM Comput. Surv.*, vol. 52, no. 2, 2019.

[181] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proceedings - International Conference on Image Processing, ICIP*, 2016.

[182] D. Held, S. Thrun, and S. Savarese, “Learning to track at 100 FPS with deep regression networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[183] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[184] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proceedings - International Conference on Image Processing, ICIP*, 2018.

[185] L. Zheng *et al.*, “Mars: A video benchmark for large-scale person re-identification,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[186] G. Ning *et al.*, “Spatially supervised recurrent convolutional neural networks for visual object tracking,” in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2017.

[187] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[188] H. Nam and B. Han, “Learning Multi-domain Convolutional Neural Networks for Visual Tracking,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[189] M. Kristan *et al.*, “The Visual Object Tracking VOT2015 Challenge Results,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[190] Y. Song *et al.*, “VITAL: VIsual Tracking via Adversarial Learning,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[191] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *NIPS’15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, pp. 802–810.

[192] S. Bai, J. Z. Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,” 2018.

[193] C. Rosenberger, “Contribution à l’évaluation d’algorithmes de traitement d’images,” Université d’Orléans, 2006.

[194] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “CDnet 2014: An expanded change detection benchmark dataset,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 393–400, 2014.

[195] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: principles and practice of background maintenance,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 255–261.

[196] S. Singh, S. A. Velastin, and H. Ragheb, “MuHAVi: A multicamera human action video dataset for the evaluation of action recognition methods,” in *Proceedings - IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010*, 2010.

[197] A. Vacavant, L. Tougne, L. Robinault, and T. Chateau, *Workshop on background models challenge*, vol. 7728 LNCS, no. PART 1. 2013.

[198] D. Gruyer, C. Royère, N. Du Lac, G. Michel, and J. M. Blosseville, “SiVIC© and RTMaps©, interconnected platforms for the conception and the evaluation of driving assistance systems,” in *13th World Congress on Intelligent Transport Systems and Services*, 2006.

[199] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[200] D. H. Parks and S. S. Fels, “Evaluation of background subtraction algorithms with post-processing,” in *Proceedings - IEEE 5th International Conference on Advanced Video and Signal Based Surveillance, AVSS 2008*, 2008.

[201] A. Vacavant, L. Tougne, L. Robinault, and T. Chateau, *Workshop on background models challenge*, vol. 7729 LNCS, no. PART 2. 2013.

[202] A. Shahbaz, J. Hariyono, and K. H. Jo, “Evaluation of background subtraction algorithms for video surveillance,” *2015 Front. Comput. Vision, FCV 2015*, no. February 2019, pp. 1–4, 2015.

[203] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “changedetection.net: A new change detection benchmark dataset,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[204] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape

contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[205] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[206] A. Baddeley and I. Molchanov, "Averaging of Random Sets Based on Their Distance Functions," *J. Math. Imaging Vis.*, vol. 8, pp. 79–92, 1998.

[207] R. Collins, X. Zhou, and S. K. Teh, "An Open Source Tracking Testbed and Evaluation Web Site," *IEEE Int. Work. Perform. Eval. Track. Surveill.*, 2005.

[208] R. B. Fisher, "The PETS04 surveillance ground-truth data sets," in *Sixth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2004.

[209] P. Dendorfer *et al.*, "MOT20: A benchmark for multi object tracking in crowded scenes," pp. 1–7, 2020.

[210] A. Ellis, A. Shahrokni, and J. M. Ferryman, "PETS2009 and Winter-PETS 2009 results: A combined evaluation," in *Proceedings of the 12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS-Winter 2009*, 2009.

[211] P. Voigtlaender *et al.*, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.

[212] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013.

[213] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[214] J. C. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Trans. Multimed.*, vol. 8, no. 4, pp. 761–774, 2006.

[215] J. Black, T. Ellis, and P. Rosin, "A Novel Method for Video Tracking Performance Evaluation," in *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003.

[216] K. Smith, D. Gatica-Perez, J. Odobez, and Sileye Ba, "Evaluating Multi-Object Tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[217] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *Eurasip J. Image Video Process.*, 2008.

[218] W. Luo, J. Xing, X. Zhang, X. Zhao, and T.-K. Kim, "Multiple Object Tracking: A Review," *CoRR*, 2015.

[219] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in*

Bioinformatics), 2016.

[220] Y. Li, C. Huang, and R. Nevatia, “Learning to associate: Hybridboosted multi-target tracker for crowded scene,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009.

[221] A. Nghiem *et al.*, “ETISEO , performance evaluation for video surveillance systems,” in *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2007, pp. 476–481.

[222] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, “Robust real-time unusual event detection using multiple fixed-location monitors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 550–560, 2008.

[223] T. Rose, J. Fiscus, P. Over, J. Garofolo, and M. Michel, “The TRECVid 2008 Event Detection evaluation,” in *2009 Workshop on Applications of Computer Vision, WACV 2009*, 2009.

[224] W. Li, V. Mahadevan, and N. Vasconcelos, “Anomaly detection and localization in crowded scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 18–32, 2014.

[225] S. Oh *et al.*, “A large-scale benchmark dataset for event recognition in surveillance video,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[226] D. Hall *et al.*, “Comparison of target detection algorithms using adaptive background models,” in *Proceedings - 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS*, 2005.

[227] G. Bonaccorso, *Machine Learning Algorithms: Popular algorithms for data science and machine learning*, 2nd Editio. Ltd, Packt Publishing, 2018.

[228] N. O. Mahony, T. Murphy, K. Panduru, D. Riordan, and J. Walsh, “Improving controller performance in a powder blending process using predictive control,” in *2017 28th Irish Signals and Systems Conference, ISSC 2017*, 2017.

[229] F. Milletari, N. Navab, and S. A. Ahmadi, “V-Net: Fully convolutional neural networks for volumetric medical image segmentation,” in *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 2016.

[230] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X. S. Hua, “Spatio-temporal AutoEncoder for video anomaly detection,” in *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, 2017.

[231] S. Ji, W. Xu, M. Yang, and K. Yu, “3D Convolutional neural networks for human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2013.

[232] B. R. Kiran, D. M. Thomas, and R. Parakkal, “An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos,” *J. Imaging*, vol. 4, no. 2, 2018.

[233] X. Wang, W. Xie, and J. Song, “Learning Spatiotemporal Features with 3DCNN and ConvGRU for Video Anomaly Detection,” in *International Conference on Signal Processing Proceedings, ICSP*, 2019.

[234] J. Nogas, S. S. Khan, and A. Mihailidis, “DeepFall: Non-Invasive Fall Detection with Deep Spatio-Temporal Convolutional Autoencoders,” *J. Healthc. Informatics Res.*, vol. 4, pp. 50–70, 2020.

[235] S. H. Kim, S. C. Lim, and D. Y. Kim, “Intelligent intrusion detection system featuring a virtual fence, active intruder detection, classification, tracking, and action recognition,” *Ann. Nucl. Energy*, vol. 112, pp. 845–855, 2018.

[236] S. Mallat, “Understanding deep convolutional networks,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 374, no. 2065, 2016.

[237] A. Bietti and J. Mairal, “Group invariance, stability to deformations, and complexity of deep convolutional representations,” *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1–49, 2019.

[238] C. Szegedy *et al.*, “Intriguing properties of neural networks,” in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[239] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[240] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, “Parseval networks: Improving robustness to adversarial examples,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.

[241] M. Belkin, D. Hsu, and P. P. Mitra, “Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate,” in *Advances in Neural Information Processing Systems*, 2018.

[242] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, “Learning shape correspondence with anisotropic convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2016.