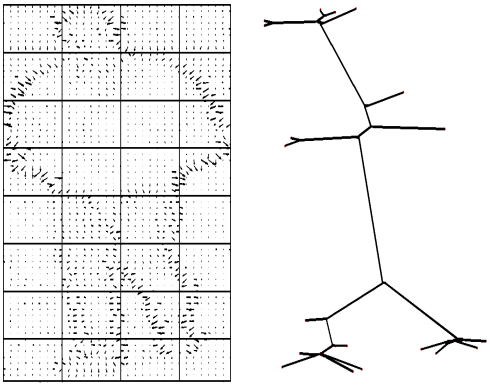
Méthodes à noyaux pour la détection de piétons

Frédéric Suard

Institut National des Sciences Appliquées



Thèse de doctorat

par Frédéric SUARD

pour obtenir le grade de

Docteur de l'Institut National
des Sciences Appliquées de Rouen

Discipline : Informatique

Méthodes à noyaux pour la détection de piétons

Kernel Machines for Pedestrian Detection

Jury :

Didier Aubert	Directeur de Recherche à l'INRETS, Versailles
Abdelaziz Bensrhair (Directeur)	Professeur à l'INSA de Rouen
Nozha Boujemaa (Rapporteur)	Directrice de Recherche à l'INRIA, Rocquencourt
Stéphane Canu	Professeur à l'INSA de Rouen
Sylvie Philipp-Foliguet (Rapporteur)	Professeur à l'ENSEA, Cergy-Pontoise
Alain Rakotomamonjy (Encadrant)	Professeur à l'Université de Rouen

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes (LITIS) - INSA de Rouen
Avenue de l'Université, 76801 Saint Etienne du Rouvray Cedex

1er décembre 2006

Table des matières

Résumé

La détection de piéton est un problème récurrent depuis de nombreuses années. La principale confrontation est liée à la grande variabilité du piéton en échelle, posture et apparence.

Un algorithme efficace de reconnaissance de formes doit donc être capable d'affronter ces difficultés. En particulier, le choix d'une représentation pertinente et discrimante est un sujet difficile à résoudre.

Dans notre cas, nous avons envisagé deux approches. La première consiste à représenter la forme d'un objet à l'aide de graphes étiquetés. Selon les étiquettes apportées, le graphe possède en effet des propriétés intéressantes pour résoudre les problèmes de variabilité de taille et de posture. Cette méthode nécessite cependant une segmentation rigoureuse au préalable.

Nous avons ensuite étudié une représentation constituée d'histogrammes locaux d'orientation de gradient. Cette méthode présente des résultats intéressants de par ses capacités de généralisation. L'application de cette méthode sur des images infrarouges complètes nécessite cependant une fonction permettant d'extraire des fenêtres dans l'image afin d'analyser leur contenu et vérifier ainsi la présence ou non de piétons.

La deuxième étape du processus de reconnaissance de formes concerne l'analyse de la représentation des données. Nous utilisons pour cela le classifieur *Support Vector Machine* bâti, entre autres, sur une fonction noyau calculant le produit scalaire entre les données support et la donnée évaluée.

Dans le cas des graphes, nous utilisons une formulation de noyau de graphes calculé sur des «sacs de chemins». Le but consiste à extraire un ensemble de chemins de chaque graphe puis de comparer les chemins entre eux et combiner les comparaisons pour obtenir le noyau final.

Pour analyser les histogrammes de gradient, nous avons étudié différentes formulations permettant d'obtenir les meilleures performances avec cette représentation qui peut être assimilée à une distribution de probabilités.

Abstract

A lot of research have been carried out in the field of pedestrian detection using images and computers. The main obstacle is related with the pedestrian himself, which could not be easily characterized, due to its high variability. In particular, we have a scale, pose and appearance variability.

To bring an issue to these variabilities, the pedestrian representation should be carefully chosen. In our case, we first use a graph based representation. In fact, a labeled graph has interesting properties to reduce particularly the scale and pose variability.

A second representation is based on histogramms of oriented gradient, which computes some local histogramms of an image. This method has a good generalization capacity against variability. To apply this method on infrared images in order to detect pedestrians, we have to design a function to extract and analyse some windows from this image.

The second part of the pattern recognition process is the classification. In our case we use the Support Vector Machine classifier, which is based on an particular function : the kernel function. The aim is to compute the inner product between data.

For the graph method, we have to design an inner product between graphs. The aim is to compare two graphs by using bag-of-paths. That is to say, we extract some paths from graphs and we compare paths between them. The final kernel is obtained by combining all comparisons between paths.

We also studied different kinds of kernel functions more specific for the histogramm-based representation in order to improve the performance of the classifier when we are using the HOG descriptor.

Remerciements

Je tiens tout d'abord à remercier mes encadrants, Alain Rakotomamonjy et Abdelaziz Bensrhair pour leur soutien, leurs conseils et leur présence durant ces années de thèse. Pour m'avoir fait découvrir la recherche et me permettre d'y prendre pied.

Je remercie les rapporteurs et les membres de mon jury de thèse, Nozha Boujemaa, Sylvie Philipp-Foliguet, Didier Aubert et Stéphane Canu pour l'intérêt qu'il ont porté à mon travail et pour l'avoir enrichi par leurs remarques et conseils.

Je remercie également les membres du laboratoire LITIS, pour m'avoir accueilli durant ces trois années de thèse, ainsi que les membres du département ASI. J'adresse des remerciements particuliers à toutes les secrétaires qui m'ont aidé dans mes diverses démarches administratives.

Durant ma thèse j'ai eu le bonheur de partager le bureau de nombreux autres thésards et associés : Vincent, Gaëlle, Olivier, Fabien, Karine, Karina, Jean-Christophe, Aurélie, Grégory et Sam, je les en remercie tous.

Je remercie enfin Valérie, Gérard, Simonne et Audrey pour leur soutien durant ces années d'études et Emilie pour m'avoir soutenu et supporté durant ces derniers mois difficiles.

Notations

Espaces Notations des espaces utilisés pour la définition des données et des fonctions.

$\mathscr{H}$ ..... Ensemble d'hypothèses, la plupart du temps, il s'agira d'un espace de Hilbert.
$\mathbb{R}$ ..... Espace des réels
$\mathbb{N}$ ..... Espace des entiers naturels
$\mathscr{X}$ ..... Espace des données exemples
$\mathscr{Y}$ ..... Espace des étiquettes
$\mathscr{A}$ ..... Espace des arcs
$\mathscr{N}$ ..... Espace des nœuds

Statistiques Notations des termes statistiques

$\mathscr{P}$ ..... Famille de lois de probabilité
$\mathbb{P}$ ..... Loi de probabilités
$\mathbb{E}$ ..... Espérance
$\mathbb{V}$ ..... Variance
L ..... Rapport de vraisemblance

Vecteurs De manière générale, les vecteurs sont notés en caractères gras et sont des vecteurs colonnes. Les vecteurs lignes sont indiqués par la présence du signe ^{T}.

$\mathbf{X} \in \mathbb{R}^{n \times d}$ ..... Ensemble de vecteurs de données
$\mathbf{x} \in \mathbb{R}^d$ ..... Vecteur de données
$x_i \in \mathbb{R}$ ..... $i_{\text{ème}}$ valeur d'un vecteur de donnée
$\mathbf{y} \in \mathbb{R}$ ..... Vecteur d'étiquettes
y_i ..... $i_{\text{ème}}$ étiquette
$\mathbf{w}$ ..... Coefficients de l'hyperplan de la fonction de décision
$\mathbf{1}^d$ ..... Vecteur de taille d contenant des 1
α, β ..... Multiplicateurs de Lagrange
H ..... Vecteur contenant des histogrammes

Fonctions Notations des fonctions utilisées dans les différents chapitres.

$\mathscr{L}$ ..... Lagrangien
ℓ ..... Fonction coût
$R(.)$ ..... Risque
$k(.,.)$ ..... Fonction noyau entre deux variables
$\text{sign}(.)$ ..... Fonction signe

$f(.)$ ..... Fonction de décision
$\underset{x\in\mathscr{X}}{\operatorname{argmin}} T(.)$ ..... Valeur de x qui donne le minimum de $T(.)$
$\underset{x\in\mathscr{X}}{\operatorname{argmax}} T(.)$ ..... Valeur de x qui donne le maximum de $T(.)$
$min(.,.)$ ..... Fonction retournant le minimum entre deux variables
$max(.,.)$ ..... Fonction retournant le maximum entre deux variables
$\langle .,. \rangle_{\mathscr{H}}$ ..... Produit scalaire dans l'espace $\mathscr{H}$
$V_\upsilon(\mathbf{x})$ ..... Voisinage au point $\mathbf{x}$ d'une taille υ
$d(.,.)$ ..... Fonction de distance
$\mathscr{O}(.)$ ..... Domination asymptotique, représente la complexité

Paramètres et cœfficients Notations des variables utilisées pour les méthodes de classification.
σ ..... Largeur de bande
C ..... Pondération des données mal classées pour le classifieur SVM
w_0 ..... Ordonnée à l'origine d'un hyperplan
n ..... Taille de la base d'apprentissage
ε ..... Terme de régularisation
ξ ..... Variable de relâchement

Graphes Notations utilisées pour la définition et la manipulation de graphes.
$A \subset \mathscr{A}$ ..... Ensemble d'arcs
$N \subset \mathscr{N}$ ..... Ensemble de nœuds
$\delta_n(N)$ ..... Ensemble d'étiquettes d'un nœud N
$\delta_a(A)$ ..... Ensemble d'étiquettes d'un arc A
$G(N, A, \delta_n, \delta_a)$ ..... Graphe constitué par un ensemble d'arc, de nœuds et d'étiquettes
h ..... Parcours dans un graphe
$|G|$ ..... Ordre d'un graphe G

Matrices
K ..... Matrice de Gram d'un noyau
M ..... Matrice d'adjacence d'un graphe

Morphologie Notations utilisées en morphologie mathématique pour le traitement d'images.
$S(.)$ ..... Squelette morphologique
B_k ..... Elément structurant de taille k
$E(., B_k)$ ..... Erosion d'un objet par un élément structurant
$D(., B_k)$ ..... Dilatation d'un objet par un élément structurant
$O(., B_k)$ ..... Ouverture d'un objet par un élément structurant

Acronymes Quelques acronymes utilisés au cours du manuscrit.
SVM ..... *Support Vector Machine* ou Séparateur à Vaste Marge
MCS ..... *Maximum Common Subgraph* ou Plus grand Sous-graphe Commun

RKHS ..... *Reproducing Kernel Hilbert Space* ou Espace de Hilbert à Noyau Reproduisant
KKT ..... Karush-Kuhn-Tucker
RBF ..... *Radial Basis Function* ou Fonction à Base Radiale
SVDD ..... *Support Vector Data Description* ou Description des données supports
GHI ..... *Generalized Intersection Histogram* ou Intersection Généralisée d'Histogrammes
HOG ..... *Histograms of Oriented Gradient* ou Histogrammes d'Orientation de Gradient

Introduction

« Les ordinateurs sont inutiles. Ils ne savent que donner des réponses. »

Pablo Picasso

Au début du 19e siècle, l'Europe a été bouleversée par la révolution industrielle. Les techniques de production ont été améliorées en grande partie par la mécanisation de la production. Sans revenir sur les aspects positifs et négatifs entrainés par ce changement, c'est surtout une philosophie nouvelle qui émerge dans l'industrie. L'homme va en effet utiliser sans réserve de nouveaux outils capables de l'assister dans ses tâches, aussi variées soient-elles.

Cette révolution connait un écho au milieu du 20e siècle, avec l'invention de l'ordinateur. Auparavant, l'homme dirigeait les machines qui ne pouvaient fonctionner de manière autonome. Dès lors, celles-ci acquièrent une certaine autonomie et sont désormais capables de gérer seules une fonction à laquelle elles sont assignées. Cette découverte ouvre ainsi la voie à la société de l'information qui voit ainsi une prédominance de l'intelligence artificielle.

Cette modification s'est étendue bien au-delà de l'industrie et de nombreux systèmes autonomes font maintenant partie du paysage quotidien : guichet bancaire, tri postal, métro. Ce dernier exemple illustre bien l'intérêt grandissant dont fait l'objet le monde du transport. Ce domaine est actuellement sujet à de nombreux travaux concernant l'intégration de systèmes d'aide à la conduite dans le cadre de la gestion de la circulation urbaine ou de la sécurité active et passive. La sécurité active est actuellement une source de travaux de recherche, car les solutions techniques deviennent de plus en plus variées et permettent de nombreuses améliorations.

1 Quelques données accidentologiques

Selon les chiffres fournis par la sécurité routière sur l'année 2005 [1], 84525 accidents corporels ont été recensés. Parmi eux, 16% des accidents concernent une voiture et un piéton. En ville, les usagers les plus vulnérables sont les piétons avec 27,7% des tués.

Il faut souligner le fait que les principaux dommages sont bien évidemment provoqués par les véhicules, le piéton ne disposant pas de protection suffisante. Pour diminuer le nombre de victimes des occupants de véhicules, des efforts importants ont été apportés sur la conception des véhicules, afin de les rendre plus résistants lors d'un choc. Les occupants d'un véhicule sont de mieux en mieux protégés car il est possible de modifier leur moyen de transport (airbags, ceintures de sécurité, structure). Dans le cas du piéton, il est impossible de le protéger personnellement. Le temps des armures est révolu et aucune solution technique n'est vraiment envisageable. Ce problème peut donc être réduit en évitant l'accident ou bien en prévoyant des protections sur les véhicules pour amortir le choc. La prédiction d'un choc est liée à la capacité de détection du piéton, en utilisant un système de reconnaissance au sein du véhicule.

[1] `http://www.securiteroutiere.gouv.fr/infos-ref/observatoire/accidentologie/`

Contrairement aux usagers des véhicules, le nombre de piétons victimes augmente durant l'hiver, lorsque les conditions de visibilité sont moins bonnes. Pour l'ensemble des usagers, le trafic nocturne représente une très grande partie des victimes à cause d'une faible visibilité. En effet, le trafic de nuit représente 10% du trafic total, mais 35% des blessés et 45% des tués.

Plus généralement, le nombre d'accidents est très élevé en semaine pour des trajets quotidiens, ce qui dénote une moindre vigilance de la part des usagers. Ici encore, l'utilisation d'un système d'aide à la conduite, dédié à la détection de piétons peut assister le conducteur. Une machine est en effet disponible en permanence, sans perte de vigilance. De plus, l'usage de systèmes d'acquisition particuliers, tels que les radars ou les caméras infrarouges, permet de réduire les problèmes de visibilité en apportant une information d'une nature différente au conducteur du véhicule. Aucun système n'est évidemment parfait, le conducteur doit également pouvoir intervenir afin de corriger une erreur du système de détection. La conception d'un tel système doit donc faire le compromis entre les contraintes de sécurité, de confort et de moyens.

Actuellement, le problème de la détection de piétons reste ouvert. De nombreux travaux ont d'ores et déjà été effectués [11, 20, 21, 23, 29, 54, 54, 67, 78] et permettent aujourd'hui d'avoir une vision précise de cette problématique et des contraintes présentes.

La mise en place d'un système de détection de piétons est confrontée à différents problèmes, de par la nature très variable du piéton. Il est en effet relativement difficile de modéliser simplement un humain. Celui-ci se présente selon différentes apparences, postures et tailles. Un piéton est également présent dans différents environnements qui peuvent générer des occultations et accentuent d'autant la variabilité du piéton.

Dans ce mémoire, nous allons étudier plus précisément la problématique de détection de piétons, au niveau de l'extraction de caractéristiques et de la discrimination. Nous allons donc orienter nos travaux selon ces deux axes :

– l'étude et l'analyse de différentes méthodes de caractérisation de piétons dans une image
– et l'etude et l'analyse de méthodes de classification performantes à associer à ces caractéristiques.

Notre objectif dans cette thèse est d'aborder le problème de la détection de piétons en apportant une attention particulière aux problèmes de variabilités. En effet, notre point de vue est qu'une caractérisation de piétons peu sensible à leurs variabilités permet d'augmenter considérablement le pouvoir de généralisation d'un système de détection construit à partir d'exemples.

2 Organisation du manuscrit

Le manuscrit est organisé de la façon suivante. Les deux premiers chapitres nous permettent de définir la problématique de la détection de piétons et les méthodes existantes. Nous présenterons ensuite le cadre de la théorie de la décision et les machines à noyaux. Le troisième chapitre introduit une première contribution dans le cas d'une représentation par graphe. Le quatrième et dernier chapitre expose notre deuxième contribution à l'aide d'une représentation constituée d'histogrammes d'orientation de gradient.

Etat de l'art pour la détection de piétons

Dans un premier temps, nous ferons donc le bilan des méthodes existantes dans le domaine de la détection de piétons. Nous nous plaçons dans le cas où les données disponibles sont des images. Nous nous focaliserons donc sur différentes approches permettant de représenter des images de piétons. La représentation peut être globale lorsqu'un seul descripteur permet de décrire l'intégralité de l'image ou locale en utilisant un ensemble de descripteurs du voisinage de points d'intérêt. Il est également possible de segmenter l'image afin de la représenter par un ensemble de régions. Quelque soit l'approche, il existe différents types de caractéristiques qui permettent ensuite de décrire l'image. Ainsi, la description d'une image peut être établie sur la valeur des pixels ou l'information de

gradient. Il est également possible d'appliquer des transformations à l'image, par exemple une décomposition en ondelettes.

Ce bilan nous permettra de préciser l'orientation des travaux et de justifier les choix de notre approche. Nous détaillerons dans ce chapitre les éléments de la problématique liée à la détection de piétons. Le piéton peut en effet apparaître à différentes échelles et positions dans l'image. Il peut également être mobile ou immobile et n'adopte donc pas la même posture selon sa vitesse et sa direction. Enfin, le piéton peut être présent sous différents aspects, les habits variant selon la saison, la mode ou les habitudes culturelles.

Nous présenterons différentes méthodes existantes, leur principe et leurs spécificités. Nous disposerons ainsi d'un échantillon de systèmes de reconnaissance complets, de systèmes d'acquisition qui permettent d'extraire des caractéristiques de l'image et des méthodes d'analyse efficaces. Nous analyserons en quoi les différentes approches peuvent résoudre le problème de variabilité.

Théorie de la décision

Nous reviendrons ensuite sur la théorie de la décision et les méthodes d'analyse pour la reconnaissance de formes. Dans la première partie, nous présenterons en effet les différentes possibilités de caractérisation et représentation du piéton. Nous devrons donc étudier les méthodes permettant d'analyser ces caractéristiques.

Après avoir présenté la théorie Bayésienne de la décision, nous détaillerons la discrimination par méthodes à noyaux. Les machines à noyaux sont actuellement les méthodes de référence pour la classification. L'apport de la fonction noyau permet en effet d'appliquer des algorithmes de classification indépendamment des données. Il est ainsi possible d'abstraire complètement la fonction de comparaison des données de l'algorithme de classification. Nous détaillerons en particulier le classifieur *Support Vector Machine* [90] utilisé par la suite dans nos travaux.

Représentation à l'aide de graphes

Dans la troisième partie, nous aborderons une méthode de représentation fondée sur des graphes. Nous supposons connaître le masque binaire définissant la région exacte des objets. Ainsi, pour comparer différents objets entre eux, nous proposons de les comparer en utilisant leur squelette morphologique. En effet, le squelette permet de représenter la forme de l'objet sous une forme compacte. Plutôt que de comparer les squelettes entre eux, nous les transformons en graphes étiquetés.

Les graphes possèdent en effet des propriétés naturelles intéressantes pour résoudre le problème de variabilité du piéton. Le but d'un graphe permet en effet de représenter la structure, l'agencement de différentes composantes entre elles. Dans notre cas, nous considérons que le graphe permet de représenter la forme d'un objet en reliant entre eux des pixels particuliers du squelette.

Selon l'étiquetage, les graphes apportent une réponse au problème de variabilité dans le cas des changements d'échelle ou d'apparence. De plus, les graphes peuvent être invariants en rotation et en translation.

Notre but, pour la détection de piétons, revient alors à effectuer une discrimination sur des graphes. Dans notre cas, nous utilisons des machines à noyau, en l'occurence le classifieur SVM. Ce classifieur est bâti sur une fonction qui détermine le produit scalaire entre chaque donnée support et la donnée évaluée. Ainsi, il est nécessaire de définir un noyau entre graphes, c'est à dire un produit scalaire entre graphes. Cependant, les graphes sont des structures complexes et la définition d'un produit scalaire est plus compliquée que celui sur des vecteurs de $\mathbb{R}^d$.

Nous étudierons donc la définition des noyaux de graphes. Dans un premier temps, nous nous baserons sur la formulation de Kashima [47], qui propose de calculer un produit scalaire en comparant les chemins générés aléatoirement dans chaque graphe. La comparaison entre chemins revient en fait à comparer les valeurs des étiquettes

des nœuds et des arcs présents sur le chemin. Nous détaillerons ainsi les possibilités offertes par cette formulation et ses limitations.

Nous étendrons ensuite la formulation de Kashima à la définition d'un noyau de graphe en considérant des «sacs de chemins». Nous proposerons ainsi deux formulations différentes pour la comparaison des chemins.

Nous validerons cette approche grâce à une base d'images de piétons dont le masque binaire est défini manuellement, ce qui nous affranchit du problème de segmentation.

Nous testerons enfin la méthode sur des images stéréoscopiques. Les méthodes de stéréovision permettent en effet de déterminer la position spatiale des objets de l'environnement. Nous pouvons donc appliquer une segmentation région en regroupant les pixels selon leur localisation. Nous proposerons de rajouter une fonction de segmentation pour extraire automatiquement les masques binaires des images. Une fois l'image segmentée, chaque région décrit un seul objet. Nous pouvons donc représenter chaque objet par un graphe pour effectuer une reconnaissance de formes.

Histogrammes d'orientation de gradient

Dans la dernière partie, nous aborderons une méthode de description globale des images. Ici, le but consiste à représenter l'intégralité d'une image à l'aide d'un seul descripteur. La description utilise des histogrammes d'orientation de gradient [21]. L'image contient donc non seulement un objet unique, mais également quelques éléments de l'environnement. Après avoir étudié en détail le fonctionnement de cette méthode, nous proposerons des améliorations en définissant un noyau spécifique. Nous appliquerons ce descripteur à la détection de piétons à l'aide d'images infrarouges.

L'utilisation de l'information de gradient permet de résoudre la variabilité d'apparence, puisque nous décrivons la forme de l'objet contenu dans l'image. La particularité de cette méthode consiste à découper l'image en cellules, puis de calculer un histogramme pour chacune d'elle. Nous obtenons ainsi une description globale constituée d'un ensemble de descriptions locales.

Après avoir détaillé le déroulement de la méthode, nous présenterons des résultats comparant cette description avec d'autres types de descripteurs, utilisant la valeur des pixels, l'information de gradient ou encore une décomposition en ondelettes.

Nous étudierons également les capacités de généralisation de la méthode, ainsi que les performances obtenues lorsque les images de piétons subissent diverses transformations telles que le redimensionnement du piéton, une translation ou une occultation. Ce test nous permettra de connaître les possibilités de cette description face au problème de variabilité.

Pour présenter et valider cette méthode, nous utilisons ainsi un SVM linéaire. Nous avons donc étudié d'autres formulations qui nous permettent d'améliorer les performances de cette méthode. Nous avons pour cela proposé différentes formulations de noyaux, afin de trouver la fonction la plus adaptée à ce descripteur.

Enfin, nous appliquerons cette méthode sur des images complètes pour la détection de piétons par infrarouge. Comme la méthode applique une description de manière globale, la détection et la localisation précise des piétons dans l'image d'une scène complète nécessite une recherche dans l'image. Cette recherche peut être résumée par l'extraction de fenêtres dans des zones contenant potentiellement un piéton. Les images extraites sont redimensionnées à la même taille définie préalablement, ce qui permet de résoudre le problème de variabilité à l'échelle. L'image contenue dans cette fenêtre est ensuite décrite par des histogrammes d'orientation de gradient, puis le descripteur est analysé. Il est ainsi possible de déterminer si la fenêtre contient ou non un piéton.

Nous présenterons donc notre approche pour extraire des fenêtres dans une image globale en utilisant les propriétés des images infrarouges et la recherche de caractéristiques significatives de la présence d'un piéton.

Nous comparerons différentes approches possibles pour l'extraction de fenêtres en évaluant les résultats sur une séquence d'images infrarouges étiquetées manuellement.

3 Contributions

Nos travaux portent sur la détection de piétons à l'aide de systèmes de vision. Le but est de travailler conjointement sur la représentation à l'aide d'images et la classification dans le cas des méthodes à noyaux.

Noyau de graphes

Tout d'abord, nous avons choisi de représenter la forme d'objets à l'aide de graphes étiquetés. Dans notre cas, les étiquettes sont des vecteurs de scalaire et peuvent être étendus à des étiquettes structurées.

De plus, cette approche de représentation à l'aide de graphes nécessite la définition d'un noyau entre graphes. Nous proposerons ainsi une extension de la formulation de Kashima, en définissant le noyau de graphes comme «sacs de chemins» que nous appelerons formulation des k-chemins.

L'idée revient ainsi à définir des noyaux mineurs pour la comparaison des chemins pour les combiner ensuite à l'aide d'un noyau majeur. Nous devons donc définir une formulation pour noyau entre chemins et la combinaison des noyaux mineurs.

Le principal écueil de la méthode de Kashima réside dans sa complexité importante. Nous verrons que la complexité est liée en partie à la définition des parcours dans les graphes. Ainsi, nous proposerons une approche différente pour définir les chemins dans un graphe en remplaçant la notion de marche aléatoire par la notion de chemin direct.

Une autre spécificité de la formulation des k-chemins réside dans la possibilité de limiter la longueur des chemins. Nous verrons ainsi qu'il n'est pas nécessaire d'utiliser des chemins de grande longueur qui n'apportent pas de gain de performance. Intuitivement, nous justifions cela par le fait que la formulation du noyau mineur est calculée à l'aide d'un produit de fonctions dont la valeur est comprise entre 0 et 1. Les chemins de grande longueur ont ainsi un poids plus faible par rapport aux chemins plus courts.

Noyau d'histogrammes

Dans la dernière partie, nous allons ainsi étudier en détails la description fondée sur des histogrammes d'orientation de gradient. Nous appliquerons cette méthode pour la détection de piétons dans des images infrarouges.

Nous étudierons ensuite différentes méthodes existant dans la littérature permettant de définir un noyau plus adapté au type de données présentes. En l'occurence, il s'agit d'histogrammes. Nous allons ainsi étudier le noyau gaussien et laplacien. Nous allons également étudier un noyau calculant les intersections entre histogrammes. Nous évaluerons également les performances obtenues par un noyau rationnel. Pour les noyaux à base radiale et rationnels, nous comparerons différentes fonctions de distances entre vecteurs de $\mathbb{R}^d$ calculées selon la norme L1, la norme L2 et la distance du χ^2. Cette dernière est en effet théoriquement plus adaptée dans le cas des histogrammes car elle permet de comparer deux distributions de probabilité.

Nous souhaitons également appliquer cette méthode sur des images infrarouges pour la détection de piétons. Nous allons donc étudier différentes méthodes permettant de rechercher des piétons dans une image infrarouge en utilisant d'une part les propriétés des images infrarouges et d'autre part sur une recherche d'éléments significatifs de la présence d'un piéton.

4 Discussion

Le thème de ce mémoire concerne la détection de piétons par un système de vision et machines à noyau. Notre approche consiste à étudier différentes méthodes de représentation afin d'obtenir une caractérisation pertinente. Le choix de la représentation sera également effectué afin d'apporter une réponse au problème de variabilité du piéton.

Le choix d'une représentation adéquate est confronté à notre approche d'imitation de l'homme. En effet, l'intuition consiste souvent à résoudre le problème comme le ferait un humain, sans tenir compte des spécificités et des capacités des machines. Les images infrarouges permettent par exemple de visualiser la scène observée lorsque la visibilité est réduite.

Nous devons également nous poser la question de la représentation unique. Certaines méthodes de représentation nous permettent d'obtenir des bonnes performances de classification de données. Elles restent cependant souvent confrontées à des hypothèses ou des contraintes sur son application qui la limite finalement à un domaine restreint. La solution serait alors de combiner différentes sources de représentation afin que chacune puisse pallier les déficiences d'une autre.

Enfin, il faut souligner la dépendance existant entre la représentation et l'analyse des données. Le choix d'une représentation pertinente est certe prépondérant, mais il ne faut pas négliger la finalité de la caractérisation des données. La pertinence et le pouvoir de représentation doivent être exploités par des algorithmes adaptés à ces méthodes. Les noyaux peuvent donc être vus comme une interface entre la représentation et l'analyse.

1

Etat de l'art

« Piéton : automobiliste descendu de sa voiture. Automobiliste : piéton remonté dans sa voiture. »

Léo Campion

Sommaire

Ce premier chapitre a pour but de faire le point sur différentes méthodes de reconnaissance de formes appliquées à la reconnaissance de piétons. Après avoir défini la reconnaissance de formes, nous illustrerons l'application pour la détection de piétons selon différentes approches. Ces approches diffèrent au niveau des étapes d'extraction de caractéristiques ou de l'analyse. Nous avons choisi de retenir les méthodes les plus représentatives et présentant à l'heure actuelle des solutions viables pour l'état de l'art de la détection de piétons.

1.1 La reconnaissance de formes

La reconnaissance de formes est un thème récurrent dans le domaine de l'intelligence artificielle.

Le but de la reconnaissance de formes consiste à analyser un ensemble de données en comparant leurs caractéristiques avec des configurations ou des modèles connus à l'aide de méthodes de classification.

Les applications de la reconnaissance de formes sont nombreuses, et certaines méthodes sont d'ores et déjà utilisées au quotidien, notamment pour la reconnaissance d'écriture et de la parole, la traduction, ou encore les prévisions météorologiques. Avec l'évolution des capacités des machines et l'amélioration des méthodes de reconnaissance de formes, de nouvelles applications peuvent ainsi être envisagées, comme, par exemple, la détection de piétons.

Le processus de reconnaissance de formes est accompli sur plusieurs étapes successives :

1. Acquisition : assimilation des données, prétraitements,
2. Représentation : extraction de caractéristiques pertinentes,
3. Analyse : discrimation de la représentation.
4. Post-traitements : validation de la décision.

Nous allons maintenant décrire brièvement les étapes de ce processus.

1.1.1 Acquisition des données

La première étape consiste à acquérir des données, c'est à dire à transformer une observation d'un élément physique effectuée par un capteur en un signal utilisable ultérieurement. A ce stade du processus de reconnaissance de formes, l'acquisition doit être la plus exhaustive, afin de conserver un maximum d'information pertinente. Une «bonne» acquisition garantit ainsi l'obtention de données fidèles à l'observation.

Lors de l'acquisition, la qualité des données recueillies dépend directement de la qualité des capteurs et de l'environnement de mesure. Le choix du ou des capteurs sera effectué en fonction du type de données en présence, selon les contraintes d'acquisition et de traitement. Dans notre cas, nous nous intéresserons principalement à des données de type image.

Si les données sont ensuite traitées par un ordinateur, le signal doit être inteprété, car sa nature peut être incompatible avec les traitemens ultérieurs. Dans le cas des images, elles doivent être numérisées.

L'acquisition peut être complétée par un prétraitement pour, par exemple, filtrer certaines composantes du signal afin de réduire l'effet du bruit ou de modifier les données pour accentuer certaines composantes.

1.1.2 Extraction de caractéristiques

L'extraction de caractéristiques est une étape majeure dans le processus de reconnaissance. Le but est de chercher l'information la plus pertinente permettant de représenter les données. Pour être la plus efficace possible, l'extraction de caractéristiques peut nécessiter des connaissances *a priori*. Tout d'abord, il est nécessaire de connaître la nature des données présentes en entrée du processus, c'est à dire l'espace dans lequel les données existent. Les méthodes employées pourront être différentes si le signal représente des images ou de la parole.

Les caractéristiques extraites peuvent être les données brutes échantillonnées. Pour permettre une représentation plus compacte, les données peuvent être compressées. Nous pouvons également extraire une information de nature différente. Par exemple, une analyse spectrale d'une image permet de convertir une information visuelle en information fréquentielle.

Si dans un premier temps aucune information n'est disponible pour valider la pertinence des caractéristiques, des méthodes peuvent être employées pour sélectionner les caractéristiques les plus pertinentes. En effet, une surabondance de données peut nuire à la performance du système. la sélection de variables permet également d'améliorer les performances de l'algorithme.

1.1.3 Analyse

L'analyse des caractéristiques permet de déterminer la classe d'appartenance des données. Il s'agit ainsi d'effectuer une classification des données.

Dans le cas d'une approche où la modélisation des données est connue, en fonction de la valeur des caractéristiques et les hypothèses définies *a priori*, la classification permettra d'affecter les données à une classe précise.

Dans une approche sans modèle, l'utilisation d'exemples, permettra de construire un modèle statistique des données après un apprentissage. Lorsque les classes des exemples sont connues, l'apprentissage est dit supervisé. Par la suite, nos travaux s'appuieront sur ce type d'apprentissage.

La définition d'un classifieur parfait est cependant impossible, le but sera donc de l'approcher au mieux selon les données disponibles et les algorithmes employés.

En disposant de suffisamment de données, la conception d'un système de reconnaissance de formes utilisera une base pour l'apprentissage et la construction du classifieur et une base de validation pour contrôler les performances de la méthode. Ce schéma permet ainsi de choisir la meilleure méthode, en contrôlant par exemple, les performances de généralisation. Nous reviendrons plus précisément sur la théorie Bayésienne de la décision dans le chapitre 2.1 et présenterons quelques méthodes de discrimination.

1.1.4 Post-traitements

Cette dernière phase permet d'exploiter le résultat obtenu par la phase d'analyse. Cette étape permet ainsi de valider la décision ou de la rejeter. Il est possible de tenir compte d'informations supplémentaires qui pourront renforcer le choix de l'acceptation du résultat.

Le contexte permet également de valider la décision. Des informations extérieures et différentes des données acquises peuvent être utilisées pour ajouter une information pertinente pour aider à la décision finale.

Il est également possible de définir des contraintes sur la confiance liée à la décision. Si celle-ci est trop faible, la décision finale sera alors repoussée ultérieurement si d'autres observations peuvent étayer le résultat.

1.2 Application à la détection de piétons

Nous allons maintenant aborder la reconnaissance de formes dans le cas de la détection de piétons. Les applications de la détection de piétons sont relativement variées : vidéosurveillance, sécurité routière. Cette variété de situations ne permet donc pas un système d'acquisition unique dont l'efficacité varie d'un système à l'autre. Les contraintes sont également très différentes, au niveau des conditions d'observation et de résultat.

Nous nous intéressons plus particulièrement au cadre applicatif de la route intelligente. Ici, nous sommes confrontés, d'une part, à des contraintes temps réel relativement importantes, et d'autre part à des conditions d'observations non stationnaires. Plusieurs systèmes de capteurs actifs peuvent être embarqués sur un véhicule : capteurs à ultrason, radar, balayage LASER [85].

Ces méthodes permettent d'acquérir une information partielle sur la présence d'un obstacle et sa distance. Ce type de capteur est désormais facilement utilisable dans de nombreuses circonstances. La précision de mesure est généralement très bonne et les traitements appliqués sont peu coûteux, donc utilisables en temps-réel. Cependant, ils n'apportent pas de véritable information sur le type d'objet présent.

Dans le cas de la détection de piéton, de par notre expérience humaine, le choix s'oriente donc plus naturellement sur des systèmes à base d'images. La perception de l'environnement fournie par ce genre de capteur est beaucoup plus riche et détaillée que les capteurs évoqués auparavant.

1.2.1 Problématique liée à la détection de piétons

Tout d'abord, il nous faut étudier les spécificités du piéton lorsque les données sont issues d'images. Sur la figure 1.1 nous présentons quelques images illustrant la détection de piéton.

Comme nous le constatons, il est difficile d'établir un modèle rigide du piéton. En effet, le piéton peut se présenter sous différentes postures et différents angles de vue. Selon les âges ou la position dans la scène observée, le piéton présente également une variabilité dans la taille. Enfin, le piéton peut se présenter sous différentes apparences notamment au niveau vestimentaire selon la saison ou les goûts.

FIG. 1.1 : Exemples d'images de piétons illustrant la variabilité du piéton.

Le choix des caractéristiques pour représenter des images de piétons est donc une étape importante.

Plusieurs approches permettent de décrire une image. L'approche globale consiste tout d'abord à décrire l'image dans son intégralité. L'image sera ainsi caractérisée par un unique descripteur. Cette approche suppose que l'image décrive un objet unique.

Un descripteur local s'appuie, lui, sur une information de voisinage. Une image sera ainsi caractérisée complètement par un ensemble de descripteurs locaux. La localisation des descripteurs locaux joue un rôle important dans cette méthode. Intuitivement, l'échantillonnage régulier de l'image est moins efficace qu'une application utilisant une recherche de points d'intérêt correspondant à des zones de variations particulières de l'image. En effet, les points d'intérêt seront présents dans chaque image contenant le même objet et seront systématiquement acquis.

Enfin, l'approche région s'appuie sur une segmentation de l'image afin de pouvoir distinguer des zones présentant des propriétés particulières dans l'image. Cette méthode permet ainsi d'extraire de l'image différentes zones dont le contenu sera par la suite caractérisé et analysé.

Nous allons maintenant décrire différentes approches proposant une méthode de reconnaissance de formes appliquée à la détection de piétons.

1.2.2 Approche région

Cette première approche consiste à segmenter l'image totale selon un critère défini préablablement. Ce critère suppose donc une connaissance *a priori* de l'information extraite de l'image. Nous allons ainsi présenter différentes approches de détection de mouvement, la segmentation en contours et la segmentation par régions.

1.2.2.1 Détection du mouvement

Actuellement, une des principales applications utilisant la détection de mouvement pour la reconnaissance de piétons concerne la vidéosurveillance. Le but consiste à détecter l'apparition de piéton dans une zone précise placée dans le champ visuel d'une caméra. L'un des principaux avantages de ces systèmes réside dans l'utilisation de caméras fixes. Il est alors possible d'utiliser des méthodes simples et rapides pour effectuer une reconnaissance de formes.

La segmentation est constituée par deux alternatives : si la caméra reste fixe et ne change jamais d'angle de vue, il est possible de connaître le fond lié à l'image et ainsi extraire facilement les objets apparus dans le champ de la caméra [33].

Si l'on cherche à détecter des objets en mouvement, il est alors possible de se baser sur cette information, en utilisant pour cela une succession d'images issues de la même séquence.

Les travaux originaux de Hoffman et al. en 1982 [41] sur l'interprétation du mouvement humain ont ainsi ouvert la voie à une littérature abondante, toujours en évolution [22, 29, 91]. Nous présentons ici les travaux de Viola et al. [91, 92] et Elzein et al. [29] utilisant le mouvement pour détecter la présence de piétons dans une séquence d'images.

1.2.2.1.1 Définition d'un masque de déplaçement Dans le domaine de la détection de mouvement, le papier de Viola et al. [91, 92] décrit un système complet et évolué pour la détection de piétons en utilisant la reconnaissance du mouvement. L'idée consiste à extraire à partir d'une séquence d'images, un masque de déplaçement, c'est à dire une succession de contours, et d'analyser ensuite ces masques. L'objectif est de pouvoir détecter les piétons avec une faible définition, le taux le plus bas de faux positifs dans un faible laps de temps. L'attention a donc été portée sur des méthodes peu coûteuses en temps de calcul et cependant fiables. Les caractéristiques sont extraites sur deux images successives et l'information tient compte non seulement du mouvement, mais également des images elles-mêmes, donc des valeurs des niveaux de gris.

Pour cela, l'extraction s'appuie sur plusieurs filtrages horizontaux, verticaux et diagonaux des paires d'images. Cet ensemble de filtrage permet ainsi de comparer les valeurs des pixels de chaque image et d'observer l'évolution spatiale et temporel des pixels. Les filtres ont été définis afin de correspondre aux mouvements possibles des piétons, mais restent cependant limités, comme le souligne les auteurs.

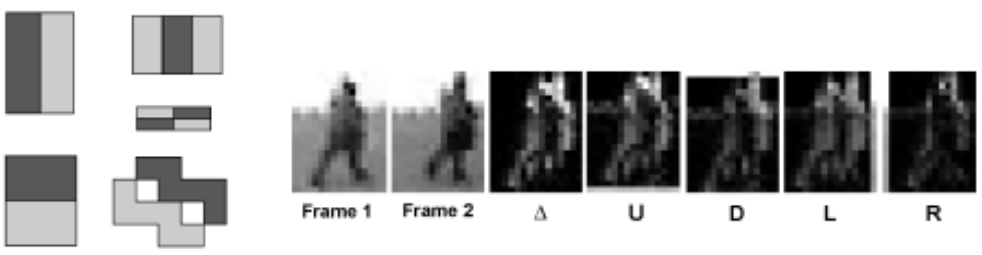


FIG. 1.2 : Exemples de filtres de mouvement (gauche) et de différences obtenues entre deux images successives présentées dans le papier [91, 92].

L'accent est ainsi porté sur la combinaison des ces caractéristiques extraites et notamment leur intégration dans un classifieur de type Adaboost [31]. La classification est donc faite par une cascade de classifieurs simples. Chaque classifieur simple a pour but de filtrer un maximum de faux positifs, afin de n'avoir, en décision finale, qu'un minimum de détections erronées et très peu de rejet de bonnes détections. La clef du bon fonctionnement de cet algorithme revient donc à réduire pendant la succession des classifieurs le taux de faux positifs beaucoup plus rapidement que le taux de détection.

Cette méthode démontre donc qu'il est possible de détecter des piétons rapidement en se basant sur des caractéristiques simples. La combinaison filtrage et cascade de classifieurs est en effet efficace et permet de résoudre ce genre de problème. Un autre intérêt est de pouvoir détecter les piétons avec une faible définition. De plus, les résultats obtenus sont bons et présagent des applications réelles à court terme.

Cependant, cette méthode ne peut fonctionner que sur des caméras fixes, et deux images successives doivent être prises dans les mêmes conditions d'acquisition. Cette limitation liée à la position de la caméra ne nous permet donc pas d'envisager une application embarquée. De plus, comme le souligne l'auteur, cette méthode n'apporte pas de solution au problème d'occlusion. Enfin, en cas d'objet statique, la reconnaissance ne peut fonctionner puisque la méthode utilise principalement la détection de mouvement.

1.2.2.1.2 Flot optique Nous restons dans de domaine de la détection de mouvement, avec la méthode proposée par Elzein et al. [29] fondée sur le principe du flot optique [42]. Le but revient à définir des zones d'intérêt en cherchant les régions de l'image contenant du mouvement, la finalité de la méthode revenant en fait à détecter les collisions potentielles. La détection d'un obstacle est donc prioritaire par rapport à la reconnaissance exacte du type d'obstacle.

Par rapport à la méthode Viola présentée ci-dessus, les traitements sont un peu plus coûteux, car le but n'est pas d'extraire des caractéristiques, mais de définir une région d'intérêt. Les efforts portent avant tout sur le calcul des différentes vitesses relatives des objets présents dans la scène observée.

En cas de collision potentielle, un algorithme de reconnaissance de formes est appliqué afin de détecter la présence ou non d'un piéton dans la zone d'intérêt. La méthode employée est celle proposée par Papageorgiou et al. [68], décrite un peu plus loin dans cette partie 1.2.3.1. Elle s'appuie sur la décomposition d'une image en ondelettes analysées ensuite par un classifieur de type *Support Vector Machines* [90].

Cette approche propose une solution intéressante pour détecter les piétons en mouvement. L'originalité réside dans l'utilisation d'une méthode de détection lorsque des zones à risque sont détectées.

Cependant, le clacul du flot optique reste très difficile à utiliser en pratique. En outre, seuls les objets en mouvement peuvent être détectés et un piéton immobile dans une zone de collision ne sera pas détecté.

1.2.2.2 Segmentation par contours

Nous allons présenter une autre méthode de segmentation de l'image. Les travaux de Gavrila et al. [23, 32]. proposent une approche reposant sur l'extraction de contours et la corrélation avec des modèles de piétons.

Contrairement aux travaux présentés précédemment de détection de mouvements et donc de séquences d'images, les contours sont extraits à partir d'une seule image. Une fois les contours déterminés, la distance euclidienne entre chaque pixel de l'image et le plus proche pixel de contour est calculé [44]. Cette image de distance sera utilisée par la suite pour rechercher des modèles de piétons.

Pour détecter la présence de piétons dans l'image, aucune information n'est disponible afin de localiser les piétons présents. La méthode propose donc de balayer l'image par une fenêtre glissante et de comparer le contenu de chaque fenêtre avec un modèle de piéton. Chaque fenêtre extraite est ainsi comparée à quelques modèles basiques

de piétons en calculant la distance de Chamfer [4] qui permet ainsi de donner le nombre de pixels communs entre l'image du modèle et l'image testée. Si le score obtenu est supérieur à un certain seuil fixé préalablement, la fenêtre contient un piéton.

Afin de préciser et donc d'améliorer la reconnaissance, un arbre hiérarchique des modèles est établi. La recherche est ainsi affinée et la modélisation, plus précise. La précision dépend de la profondeur de l'arbre hiérarchique : plus celui-ci aura de niveaux, plus le modèle trouvé sera précis. En contrepartie, le temps nécessaire à la reconnaissance sera plus important. En fonction des besoins et des contraintes, un compromis sera établi pour concevoir cet arbre hiérarchique.

Pour améliorer la reconnaissance, l'auteur propose par la suite de vérifier le résultat obtenu en utilisant des méthodes de régularisation dont les fonctions sont à base radiale [70]. Les performances sont ainsi améliorées, notamment en réduisant fortement le taux de faux-positifs.

Cette méthode propose une approche intéressante utilisable avec une seule image. En se basant sur des caractéristiques simples et quelques modèles de piétons, les résultats obtenus sont encourageants et montrent que plusieurs approches différentes permettent d'obtenir de bons résultats.

Ici, le choix de la représentation s'est porté sur des caractéristiques simples. Seule l'information de contour est utilisée, sans autre information à propos de la forme elle-même. Ainsi, aucune information de texture ou de couleur n'est apportée. La limitation de la taille de la représentation est avantageuse dans le sens où elle simplifie la méthode, mais l'information fournie pour décrire un objet manque de pertinence pour être suffisamment discriminante.

Cependant, la reconnaissance dépend fortement des modèles définis dans la base de référence ainsi que du seuil défini pour l'acceptation ou le rejet. L'utilisation de seuils peut également être source de nombreux problèmes. Il faut en effet arriver à régler correctement ce seuil de façon à rejeter un maximum d'erreurs, sans rejeter de bonnes détections. Le réglage empirique d'un tel paramètre nécessité un certain temps, en fonction du nombre de modèles utilisés, du taux d'acceptation d'erreur.

De plus, l'analyse s'appuie sur des modèles rigides. Pour couvrir le plus largement possible toutes les formes possibles du piéton, il faudrait utiliser un très grand nombre de modèles. L'accroissement de la base de référence imposant alors une augmentation de la complexité du système. La variabilité de la pose est une des principales problématiques pour la détection de piétons et cette approche semble donc difficilement pouvoir la résoudre.

Cependant, l'auteur met l'accent sur l'amélioration des résultats en renforçant la fonction de classification. Ainsi, l'importance de l'analyse des caractéristiques est démontrée et permet de souligner l'importance dans le choix de cette fonction. Les performances d'un algorithme de reconnaissances de formes dépendent non seulement de la pertinence de la représentation, mais aussi du choix de la fonction de classification.

1.2.2.3 Segmentation région

La segmentation région permet de diviser l'image originale en différentes zones. Les pixels de l'image sont regroupés selon un critère défini préalablement. La définition de ce critère joue un rôle important dans la segmentation de l'image. Si le critère est peu adapté ou mal choisi, le résultat fourni par la segmentation ne sera pas optimal et la suite du processus de reconnaissance de formes ne donnera pas les meilleurs résultats.

Nous présentons ici une approche utilisant la segmentation région dont le critère de regroupement correspond à la position des pixels dans le monde réel. Nous utilisons pour cela un système de stéréovision. Il s'agit d'un système d'acquisition conçu selon le modèle de perception humaine. L'humain a ainsi la capacité de percevoir visuellement son environnement et ajoute également la possibilité de localiser des objets dans l'espace. L'utilisation d'un couple de caméras permet donc d'extraire deux types d'information : une information visuelle pour décrire les objets présents et une information de position dans le monde réel. L'information de profondeur est obtenue en calculant

la disparité [3], c'est à dire l'écart de position pour un pixel donné entre l'image droite et l'image gauche. Plus l'écart est important plus l'objet auquel appartient le pixel est proche. Cependant, contrairement à l'humain, il est possible de définir avec précision la position des objets observés.

Cette méthode permet ainsi de définir des régions en utilisant l'information de distance. Il est donc possible de détacher les objets du fond grâce à la définition de régions qui permet de distinguer les objets en tenant compte de leur distance par rapport à la caméra. L'avantage de cette solution, permet ainsi de pouvoir caractériser ensuite uniquement des éléments pertinents appartenant à l'objet en question, sans bruiter l'information avec des caractéristiques appartenant à la scène elle-même.

Dans ce cas, les efforts sont principalement apportés au calcul de la disparité. En effet, les techniques de stéréovision sont assez compliquées et peuvent rapidement se révéler très coûteuses en temps de calcul. Il est, par exemple, très difficile de calculer directement une carte de disparité sur une image complète, le temps nécessaire serait trop important pour les moyens techniques actuels.

Dans ce domaine, Broggi et al. présentent de nombreux travaux utilisant la stéréovision [6, 11, 12]. Cette approche permet de définir très précisément des régions d'intérêt en séparant les pixels selon leur position 3D. Chaque région extraite est ensuite analysée afin de décider de la présence d'un piéton. La décision est fondée sur l'hypothèse selon laquelle un piéton est placé verticalement, sans posture particulière.

Chaque région contient un ensemble de contours verticaux définissant l'objet présent dans la région d'intérêt. L'analyse consiste à vérifier l'existence de contours symétriques et la taille des contours. Ainsi, si les proportions correspondent à celles d'un piéton et que la hauteur de la région d'intérêt est contenue dans un intervalle défini au préalable correspondant à la taille théorique d'un piéton, la région est alors classée comme piéton.

L'analyse des caractéristiques est donc une simple vérification des dimensions de l'objet détecté. Les critères pour classer un objet inconnu comme piéton ou non sont définis empiriquement afin de correspondre à la plupart des situations. Cependant, ces critères correspondent à la définition d'un piéton debout, de forme et de taille moyenne. Ils ne peuvent donc résoudre le problème de variabilité. En effet, l'utilisation de contraintes topologiques telles que la symétrie et la proportion est contradictoire avec la notion de variabilité. Par exemple, la taille minimale considérée est supérieure à la taille d'un enfant qui sont moins attentifs aux dangers extérieurs.

Pour résoudre ce problème de variabilité, la méthode de Zhao et al. [100], utilisant également sur une segmentation région obtenue par stéréovision propose, elle, l'utilisation d'une méthode d'analyse employant des réseaux de neurones.

Le calcul de la carte des disparités s'appuie sur le principe proposé par Small Vision System [51], qui a pour particularité de pouvoir être utilisé en temps réel sur une machine standard, sans spécification matérielle particulière. Le résultat de cette étape est donc un ensemble de régions distinctes définies par une boite englobante et la région exacte de l'objet.

La méthode propose également une possibilité intéressante permettant d'améliorer les résultats de la stéréovision. Dans le cas où deux objets voisins sont assimilés à la même région et que la région n'est pas détectée comme piéton une premiére fois, le système cherche à extraire les piétons présents dans la région en définissant des sous-régions dont la taille correspond à celle d'un piéton. Si la partition donne lieu à une détection de piéton, la région initiale est ainsi séparée en plusieurs sous-régions.

Ensuite, une étape de détection est implémentée. Le but est de pouvoir reconnaitre les régions contenant des piétons. Pour cela, l'analyse utilise un réseau de neurones. Ce classifieur admet en entrée un vecteur de caractéristiques, il faut donc choisir celles permettant de décrire au mieux la forme de l'objet défini par la région 3D. L'information de niveau de gris a été écartée pour résoudre le problème de la variation de l'illumination au profit de l'information de gradient. En effet, cette information reste robuste à ces variations et pertinente pour décrire la forme de l'objet. En outre, elle est plus avantageuse que l'extraction de contours, nécessitant un seuillage, fré-

quemment source d'erreurs.

Chaque région issue de la carte de disparité est ainsi transformée en une image de gradient et redimensionnée à la même taille. Ce redimensionnement garantit ainsi d'avoir la même taille pour chaque vecteur décrivant les régions présentes. Chaque pixel de cette image correspond à un neurone du réseau. La combinaison des neurones permet ainsi de définir un système de perception sur plusieurs couches, chaque couche étant totalement connectée à la suivante. Pour être appliqué réellement, le réseau de neurones nécessite une phase d'apprentissage durant laquelle des images de piétons seront analysées et les couches du réseau et les connexions vont ainsi pouvoir être ajustées en donnant un poid à chaque connexion.

FIG. 1.3 : A droite : carte de disparité fournie par Small Vision System. A gauche, réseau de neurones utilisé dans la méthode de Zhao et al. extraits du papier [100]

Cette méthode décrit donc une approche complète de reconnaissance de formes. Du système d'acquisition à la classification, toutes les étapes ont été traitées de manière homogène, sans négliger ou favoriser une étape particulière. Les résultats obtenus, notamment au niveau de la résolution des problèmes de variabilité de posture et d'échelle, d'occlusion, démontrent l'intérêt de ce type d'approche.

Cependant, les informations extraites des images, restent peu nombreuses et variées. Pour pouvoir gagner en rapidité, il a été nécessaire de restreindre fortement la dimension des données, aux dépends des résultats. Avec davantage d'informations et de caractéristiques pertinentes, les résultats peuvent être améliorés, tant au niveau du taux de reconnaissance que du taux de faux-positifs.

1.2.2.4 Régions d'intérêt

Nous allons maintenant aborder une méthode permettant de définir des régions d'intérêt. L'idée consiste à détecter dans l'image des zones particulières, significatives de la présence d'un piéton à cet emplacement. La recherche de ces régions d'intérêt permet ainsi de réduire la zone de travail dans l'image. Un exemple illustre bien cette approche : la détection de piétons par images infrarouges.

Comme nous avons pu le constater précédemment, la plupart des travaux recherchent à établir des systèmes fonctionnant selon le modèle humain. Cependant, certains capteurs ont des propriétés intéressantes, et nous permettent d'acquérir des informations qu'il nous est impossible d'obtenir autrement. La caméra infrarouge est, à ce titre, une bonne illustration. Depuis quelques années, de plus en plus de travaux [8, 20, 30, 98] mettent en avant ce type d'information, principalement dans le cadre de la détection de piétons. En effet, les propriétés de ce système permettent d'acquérir visuellement les sources de chaleur émises par les objets et les corps présents dans la scène.

L'acquisition sera donc bien sous la forme d'une image, mais la valeur des pixels sera proportionnelle, non pas à la lumière naturelle réfléchie, mais à la quantité de rayonnement infrarouge émise. Les infrarouges étant émis par des sources de chaleur, ce type de capteur permet donc de visualiser les corps chauds en présence. L'avantage

principal de ce système revient donc à détecter les piétons lorsque la visibilité reste réduite, par exemple, de nuit.

Parmi les équipes s'intéressant à la détection de piétons en images visibles, beaucoup se sont également intéressées aux images infrarouges [8, 61].

L'idée de Broggi et al. [8] consiste à localiser dans un premier temps les régions d'intérêt dont la valeur des pixels est élevée, c'est à dire représentant des objets avec une température plus élevée que l'environnement.

Le but revient donc à localiser dans l'image la position des objets. Comme pour la détection de piétons par stéréovision [11], la méthode employée consiste à définir un piéton par une fenêtre englobante. La distribution de pixels dans cette fenêtre doit correspondre à un schéma défini au préalable.

Ici, il s'agit non pas de détecter les contours verticaux, mais les colonnes contenant le plus de pixels clairs. L'hypothèse selon laquelle le piéton est placé verticalement est donc conservée. Pour chaque colonne de l'image, il faut sommer les pixels contenus dans ces colonnes. Ainsi, la présence d'un objet chaud dans l'image se traduira par une valeur plus importante dans les colonnes contenant cet objet.

La suite de la méthode permet ainsi de définir des fenêtres englobantes positionnées selon les colonnes présentant une somme supérieure à un seuil. Certaines fenêtres candidates sont supprimées lorsque leurs dimensions ne correspondent pas à celle d'un piéton.

Les fenêtres candidates restantes sont ensuite validées en les comparant avec des modèles de piétons en infrarouge. Ces modèles ont été définis manuellement et correspondent à différentes positions du piéton. L'idée est ainsi de pouvoir analyser plus finement l'attitude du piéton et éventuellement de déterminer sa direction afin d'évaluer les possibilités de collision.

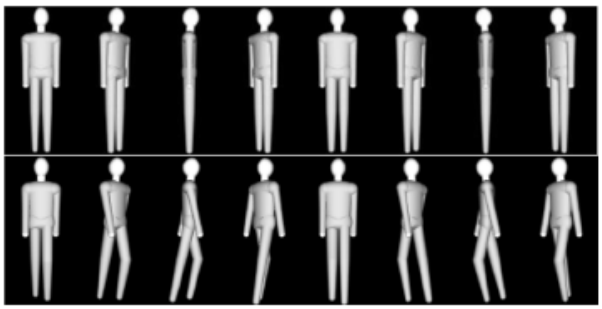


FIG. 1.4 : Exemples de modèles 3D utilisés par Broggi et al. extraits du papier [8]

L'idée originale consiste à modéliser des piétons à l'aide de modèles rigides. Cette modélisation comporte certains avantages, dont la rapidité et la simplicité d'utilisation. Elle nécessite cependant des réglages empiriques pour la validation ou le rejet des images testées. Les résultats obtenus dépendent donc très fortement des modèles définis. En l'occurence, les modèles ne sont pas issus d'images réelles, mais d'un piéton défini de manière synthétique. La difficulté de détection du piéton liée à sa variabilité de posture et d'apparence n'est pas résolue. La conception des modèles devrait être la plus exhaustive possible pour détecter tous les piétons existants. L'amélioration de la définition du modèle peut ainsi permettre de résoudre ces problèmes de variabilité.

Ainsi, dans le papier [98], Xu et al. proposent une méthode de détection de piétons par images infrarouges en conçevant un modèle statistique du piéton.

Dans un premier temps, la méthode consiste à extraire des régions d'intérêts de l'image en utilisant certaines caractéristiques de l'image. Il s'agit ici encore de repérer les zones claires de l'image, correspondant aux sources de chaleur et d'extraire autour de ces zones des boites englobantes contenant un piéton. Pour localiser les sources de chaleur, une binarisation est effectuée selon un seuillage adapatatif afin de conserver le même pourcentage de

pixels les plus clairs de l'image. Ce seuillage adaptatif est proposé pour contrer la variation présente d'une image à l'autre lors de l'apparition soudaine d'un objet chaud ou d'un éclairage.

Ensuite, pour définir exactement la taille de la fenêtre, plusieurs éléments sont pris en compte. La largeur de la fenêtre sera ainsi définie en fonction de la taille de la zone d'intérêt. Le critère retenu consiste à simplement doubler la largeur de cette zone.

Pour définir complètement la fenêtre, la méthode s'appuie sur la recherche du contact avec la route. Partant du principe qu'un piéton est posé sur le sol, la fenêtre doit donc couvrir l'intégralité de la zone allant de la région claire, qui représente la tête du piéton, jusqu'au contact avec la route.

Il faut évidemment au préalable extraire la région de l'image correspondant à la route (figure 1.5). Cette zone est définie en extrayant les contours par la méthode de Sobel [15], puis en étendant la zone présente dans le bas de l'image, à l'aide d'opérations morphologiques.

FIG. 1.5 : La figure de gauche montre les fenêtres extraites par la méthode de Xu et al. [98] et la zone correspondant à la route. La figure de droite montre une image infrarouge de piéton extraites et sa version binaire.

Pour éliminer certains candidats non désirés, une première analyse considère la taille des fenêtres et le rapport $\frac{\text{hauteur}}{\text{largeur}}$. Elle permet ainsi d'éliminer les fenêtres dont ces valeurs sont incompatibles avec la notion de piétons.

La dernière étape consiste donc à analyser le contenu de chaque fenêtre extraite de l'image originale. Ici, l'analyse fait appel à une fonction de classification *Support Vecteur Machine*.

Chaque image extraite est donc redimensionnée à une taille fixée, puis la valeur des pixels est directement utilisée comme vecteur de caractéristiques par le classifieur. Une autre alternative proposée consiste à seuiller l'imagette afin de ne retenir que des valeurs binaires dans le vecteur de caractéristiques. Cependant, cette alternative s'est révélée moins efficace. En effet, l'information fournie est nettement moins variée, et présente moins de gradation dans la valeur des pixels. Les différences d'apparence et de texture entre les différentes parties du piéton ne peuvent ainsi plus être distinguées. Cet exemple proposé par Xu illustre bien le processus de reconnaissance de formes. L'approche et les méthodes employées restent cohérentes. Les propriétés des données sont exploitées de façon à extraire correctement les caractéristiques et les images des objets présents dans la scène. La phase d'analyse n'est pas négligée non plus, de par l'utilisation d'un classifieur de type SVM.

Cependant, les caractéristiques utilisées pour décrire les images ne sont pas très pertinentes. En effet, caractériser une image simplement à l'aide de la valeur des pixels reste peu pertinent dans ce cas. La caractérisation d'une image par des valeurs binaires des pixels est utilisée principalement pour la reconnaissance de caractères. En effet, l'apparence des caractères importe peu, seule la forme générale est pertinente. Mais dans le cas présent, l'apparence du piéton n'est pas uniforme. L'utilisation de caractéristiques binaires n'est pas révélateur de la variété d'information contenue dans une image.

Enfin, pour tenter de résoudre le problème de variabilité, l'analyse des caractéristiques implique plusieurs

classifieurs, chacun d'eux étant défini selon un type de position du piéton. Cette multitude de classifieurs peut donc poser un problème de par la multiplication des classifieurs selon le nombre de postures impliquées, provoquant d'autant une augmentation de la complexité du système global.

1.2.3 Approche globale

Dans cette partie nous allons maintenant décrire plusieurs systèmes de reconnaissance de formes proposant une description globale de l'image. L'idée consiste ici à décrire l'intégralité d'une image par un seul descripteur. Cette approche suppose donc la présence d'un objet unique dans l'image décrite.

1.2.3.1 Décomposition en ondelettes

La première méthode présentée est celle de Papageorgiou et al. [67, 68] Cette méthode a été l'une des premières proposant d'appliquer le classifieur SVM [90] pour la détection de piétons.

Le but est de pouvoir représenter une image d'une taille fixée, contenant un seul objet, placé au centre de l'image. L'image est ainsi caractérisée par une décomposition en ondelettes de Haar. Ces ondelettes permettent ainsi de représenter la forme générale contenue dans l'image. En effet, les coefficients obtenus par le filtrage dépendent de la présence de contours dans l'image.

Trois configurations d'ondelettes sont utilisées afin de pouvoir caractériser les contours dans les trois directions possibles : horizontalement, verticalement et diagonalement. L'image est donc caractérisée par trois ensembles de cœfficients issus de la décomposition en ondelettes. Le nombre de cœfficients est proportionnel à la taille originale de l'image. Ces trois ensembles de cœfficients sont ensuite concaténés afin de former le vecteur de caractéristiques final.

Ce vecteur est ensuite utilisé en entrée du classifieur. Le classifieur est un SVM polynômial d'ordre 2, plus performant selon l'auteur qu'un noyau linéaire. Les résultats obtenus sont encourageants et mettent en évidence les performances fournies par la méthode de classification.

Le point noir du système concerne l'application aux images entières. En effet, pour extraire des fenêtres à partir de l'image, un balayage exhaustif de l'image est employé et chaque fenêtre extraite est ainsi analysée par le classifieur. Aucune information *a priori* ou méthode performante de localisation de piétons n'est utilisée pour réduire le nombre de fenêtres candidates.

Une spécificité de la méthode pour améliorer les résultats consiste à effectuer un rebouclage des données de validation en intégrant systématiquement dans la base d'apprentissage les erreurs fournies à la boucle précédente. Cette modification permet ainsi de réduire considérablement le taux final d'erreur.

Cette méthode a été pionnière dans l'utilisation de la classification SVM et a ainsi démontré son efficacité. Elle reste une référence pour le domaine de la détection de piétons. La caractérisation de l'image constituée par une décomposition en ondelettes est intéressante et propose ainsi une méthode pour caractériser le contenu pertinent d'une image.

Le principal inconvénient réside dans l'omission des méthodes de segmentation et d'extraction de fenêtres, ce qui peut être expliqué par la volonté de démontrer l'efficacité de l'analyse des caractéristiques. De plus, certains problèmes liés à la reconnaissance de piétons n'ont pas de solution, notamment les problèmes d'occultation qui ne trouvent pas de réponse ici. Enfin, l'amélioration des performances est possible en utilisant d'autres fonctions noyau pour le classifieur.

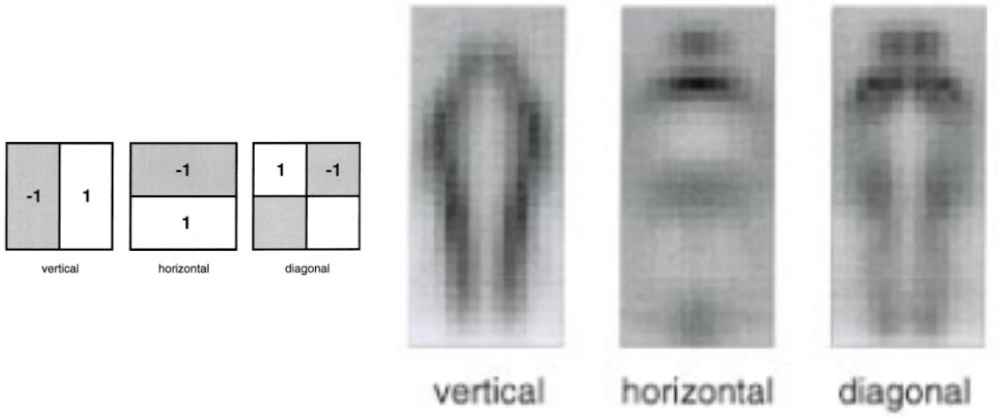


FIG. 1.6 : Différentes configuration d'ondelettes de Haar. Résultats du filtrage pour une image de piéton.

1.2.3.2 Histogrammes locaux de gradient

Une autre méthode proposant une approche globale est celle de Shashua et al. [78]. Il s'agit ici d'un système embarqué complet pour la reconnaissance de piétons. La méthode s'appuie, d'une part, sur la recherche et l'extraction de fenêtres candidates et d'autre part sur l'analyse de ces fenêtres candidates.

La localisation globale d'un piéton est effectuée en tenant compte des différences de textures, des incompatibilités de perspectives et des contraintes de taille pour définir les régions. Chaque image donne ainsi lieu à quelques dizaines de fenêtres qui sont ensuite analysées.

La caractérisation d'une image de piéton est constituée par le calcul d'histogramme d'orientation de gradient. L'image est découpée en 9 régions chacune divisée en 2×2 sous-régions. Le découpage est proposé selon la physiologie du piéton : tête, jambes, tronc (figure 1.7). Pour chaque région, une fonction de décision de type régression *ridge* est appliquée. Pour contrer le problème de variabilité de posture, la base d'apprentissage est découpée en plusieurs bases, chacune correspondant à une pose particulière du piéton. La valeur de la décision pour chaque région selon chaque base d'apprentissage est ainsi utilisée comme descripteur final.

Ce descripteur est transmis à un classifieur de type *Adaboost* [31]. Chaque descripteur étant considéré comme classifieur faible.

Selon le résultat fourni par la classification, la détection d'un piéton sera validée lorsque la confiance est suffisamment importante, ou bien retardée afin d'effectuer une nouvelle classification dans une image ultérieure.

Les résultats obtenus en appliquant réellement ce système démontrent son efficacité, notamment pour le taux de faux positifs, particulièrement bas, probablement grâce à la validation retardée de la classification qui permet ainsi de filtrer beaucoup d'erreurs ponctuelles.

La méthode présente également un intérêt particulier pour réduire le problème de variabilité en découpant l'ensemble d'apprentissage. Le système présenté est complet : il assure la localisation, l'analyse et le suivi des objets présents dans la scène.

Cependant des améliorations sont envisageables, les caractéristiques extraites sont pertinentes, mais une plus grande variété d'information permettrait également d'accroitre les performances. De plus, le découpage d'une image est proposé selon un modèle rigide qui suppose la présence d'un piéton sans posture particulière. Le problème d'occultation n'est également pas complètement résolu, sauf en envisageant dans l'ensemble d'apprentissage des images contenant des piétons masqués partiellement.

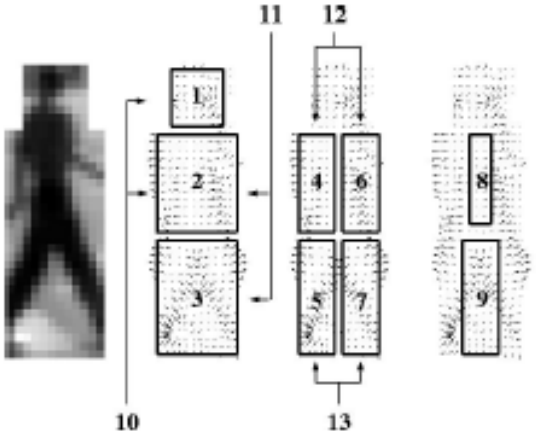


FIG. 1.7 : Découpage d'une image en différentes régions présenté dans le papier [78].

1.2.4 Approche locale

La dernière approche présentée s'appuie sur une caractérisation locale de l'image. Le but de la représentation locale consiste à échantillonner l'image afin de récupérer des motifs locaux [36, 59, 62]. La classification utilise ensuite des méthodes de type sac de mots [64]. L'apprentissage supervisé permet ainsi de définir les ensembles de motifs qui décrivent l'objet d'intérêt.

Leibe et al. [54] proposent ainsi l'utilisation de caractéristiques locaux pour la détection de piétons. L'objectif principal consiste en fait à détecter les piétons dans des situations d'occultation très prononcée. Cette approche est ainsi supposée résoudre ce problème d'occultation, car elle repose sur une caractérisation partielle des piétons.

L'échantillonnage de l'image est fonction de la détection de points d'intérêts [63]. Ces points correspondent à des zones présentant certaines particularités relévées par un détecteur. Les points d'intérêt sont localisés à l'aide d'un détecteur de type DoG (*Difference of Gaussian*) [26], qui permet également de déterminer l'influence globale du point d'intérêt. Le détecteur agit en effet selon différentes échelles sur l'image et lorsqu'un point d'intérêt est présent il est possible de connaître l'échelle correspondante.

Pour chaque point, il faut ensuite extraire le voisinage, dont la taille dépend de l'échelle du point d'intérêt. Ce voisinage est ensuite redimensionné à une taille prédéfinie et constitue un motif.

Les motifs extraits de la base d'apprentissage sont ensuite regroupés dans un dictionnaire qui sera utilisé par la suite pour effectuer la détection des piétons.

Une spécificité de cette méthode consiste à considérer la distribution spatiale des motifs. Ainsi, il est possible de déterminer des probabilités pour chaque motif d'appartenir ou non à un piéton. Ces probabilités sont ensuite utilisées pour segmenter l'image afin de pouvoir localiser les piétons.

Cette dernière étape permet ainsi de déterminer la position et le nombre de piétons, ainsi que leur posture. Pour cela, une comparaison est effectuée avec des modèles de piétons prédéfinis, en utilisant la distance de Chamfer, comme le propose Gavrila et al. [32].

Cette approche est intéressante dans le sens où l'auteur positionne la méthode de détection de façon à résoudre le problème d'occultation.

Cependant, l'utilisation d'un dictionnaire de motifs ne permet pas de détecter et localiser précisément les piétons. Il faut en effet confirmer les décisions sur plusieurs itérations et combiner cette méthode avec d'autres caractéristiques et méthodes d'analyse.

De plus, la méthode suppose l'occultation d'un piéton par un autre piéton. La présence d'un piéton dans un plan plus proche permet donc de focaliser la détection dans une zone de l'image. La détection d'un piéton occulté partiellement par un véhicule n'est pas abordée. Cette approche ne permet donc pas de résoudre complètement le problème d'occultation.

1.3 Bilan sur la détection de piétons

Dans ce premier chapitre, nous avons présenté de nombreux systèmes et applications pour la détection de piétons. Nous allons maintenant faire le bilan de cet état de l'art afin de pouvoir synthétiser les différentes méthodes et les comparer. Nous proposerons l'orientation du choix de caractéristiques et des méthodes d'analyse.

Le problème récurrent dans la détection de piétons réside dans sa très grande variabilité. Cette problématique a été maintes fois évoquée [54] et n'a toujours pas trouvé de réponse viable. Le piéton est principalement sujet à deux sortes de variablilités : la taille et la pose.

1.3.1 Variabilité à l'échelle

La variabilité de taille est provoquée par différentes causes. Tout d'abord, la variabilité liée à l'échelle : selon sa position dans la scène, le piéton apparaitra dans l'image avec une certaine taille, et donc une certaine définition. En particulier, la reconnaissance sera très difficile pour un piéton lointain qui est représenté par une faible définition dans l'image, ce qui n'autorise pas l'extraction de caractéristiques assez variées. Ensuite, le problème de taille se pose naturellement selon les différences existant entre les êtres humains, selon les individus ou les âges. Cette variabilité pose donc certaines contraintes pour définir des régions d'intérêt, ou effectuer un balayage de l'image à l'aide de fenêtres dont la taille doit être définie.

Le problème de variabilité d'échelle dépend, d'une part, du descripteur utilisé qui doit permettre la caractérisation d'un piéton quelque soit sa taille. D'autre part, l'acquisition des données influe sur cette variabilité. En effet, les méthodes décrites précédemment s'appuient sur une acquisition d'images. La définition d'une caméra étant fixe, la taille finale du piéton dépend directement de sa position réelle. La plupart des caméras actuelles permettent de représenter l'intégralité d'un piéton situé à une faible distance avec une définition suffisamment élevée pour permettre l'extraction de caractéristiques. Cependant, selon les situations ou le type de matériel, la caractérisation d'un piéton peut échouer, car nous ne disposons pas de données brutes suffisantes pour représenter visuellement le piéton.

Ce problème est donc une limitation technique résolu par l'utilisation de caméras présentant une meilleure définition. Cependant, cette augmentation de la définition pose à son tour le problème du temps de calcul. Des méthodes utilisant la multi-résolution [8, 16] proposent ainsi une approche permettant de segmenter l'image selon différentes échelles, chaque échelle permettant une représentation suffisamment précise des objets présents.

Cependant, le descripteur doit aussi pouvoir s'adapter aux différentes tailles des images représentant des piétons. Dans le cas de la segmentation région, il ne faut pas définir de contraintes à propos de la taille d'une région, afin que tous les cas possibles de taille de piéton soient envisageables. Si la méthode utilise une extraction de fenêtres lors d'un balayage de l'image, la configurations des fenêtres extraites devra présenter une variété suffisante pour extraire tous les piétons possibles.

1.3.2 Variabilité de posture

Une autre problèmatique est la variabilité de posture, plus délicate à résoudre. Contrairement à la variabilité d'échelle qui est principalement liée aux moyens techniques pour l'acquisition, la variabilité de posture est une conséquence de la nature flexible du piéton. Ainsi, le piéton peut être présent dans la scène selon différentes postures, différents angles, différentes apparences et couleurs. Il faut donc mettre au point une méthode de représentation et d'analyse suffisamment robustes pour résoudre ces nombreux cas. L'utilisation des modèles définis manuellement [8], doit donc tenir compte des postures possibles d'un piéton pour leur mise au point. Or, la plupart des méthodes fondées sur cette comparaison, ne tiennent compte que des configurations les plus courantes. La question peut donc se poser vis-à-vis de la capacité de telles méthodes pour réagir à une nouvelle posture de piéton. Prenons l'exemple d'un piéton qui décide de traverser la rue en courant. Si le modèle ne définit que les postures d'un piéton marchant à vitesse normale, le modèle n'est plus adapté lorsque le piéton est légèrement incliné, par exemple, lorsqu'il court.

La solution existe : il suffit alors de rédéfinir des nouveaux modèles. Cependant, la définition d'une base exhaustive génèrerait un trop grand nombre de modèles, ce qui accroit d'autant la taille de la base et ne permet donc plus une application simple et rapide.

Pour résoudre ce problème, il existe différentes alternatives. Tout d'abord,il faut intervenir dans le choix des caractéristiques. Si la méthode de représentation possède un fort pouvoir de généralisation, c'est à dire est capable de représenter une très grande variété de piétons à l'aide un faible échantillon en apprentissage, alors le problème peut être résolu. La caractérisation d'un piéton doit donc être déterminée par une caractérisation pertinente des piétons présents en apprentissage en combinant les informations communes discriminantes de l'ensemble des images. De même les caractéristiques extraites ne doivent être trop fidèles aux exemples utilisés dans l'apprentissage, ne permettant pas la détection de nouveaux piétons.

Face à la diversité d'apparence des humains, les caractéristiques utilisant directement les valeurs brutes [98] de l'image sont à éviter. Les différences d'un piéton à l'autre peuvent très facilement mettre en défaut cette extraction pourtant simple et relativement intuitive. Actuellement, l'extraction de l'information de gradient présente une solution prometteuse pour caractériser un piéton [21, 98]. En effet, le gradient permet de caractériser non seulement la forme générale du piéton, mais apporte également de l'information sur son apparence.

Dans le cas d'une segmentation région, le problème peut être résolu, il suffit de ne pas définir de limites sur la taille des régions. Si l'extraction s'appuie sur un balayage de l'image par une fenêtre glissante, la résolution est plus difficile [67]. Comme pour la variabilité d'échelle, la taille et la position de la fenêtre doivent donc pouvoir résoudre la variabilité de pose. Le rapport entre la hauteur et la largeur des fenêtres extraites devra être peu contraignant. Les méthodes considérant alors des notions de symétries, de rapport hauteur/largeur peuvent donc être confrontées très rapidement à ce problème de variabilité, à moins de définir des contraintes moins rigoureuses, et donc plus sensibles aux fausses alarmes.

1.3.3 Occultation

Le dernier point important pour la reconnaissance de piéton réside dans le problème de l'occultation. Ce problème est particulièrement délicat, car étant une source de danger pour les piétons. En effet, c'est justement lorsque les capacités de l'humain sont limitées que la machine devrait être capable d'intervenir et d'assister, par exemple, le conducteur. En l'occurence, très peu de méthodes sont capables de résoudre ce problème. Selon l'importance de l'occultation, le choix d'une caractérisation pertinente peut permettre d'obtenir des résultats corrects pour de faibles occultations.

Par exemple, Leibe et al. [54] propose un système capable de détecter les piétons occultés partiellement en décrivant les piétons à l'aide de descripteurs locaux extraits autours de points d'intérêt. Cependant l'occultation ne

fonctionne qu'entre des piétons et ne concerne pas l'occultation d'un piéton par tout type d'objet.

La notion d'occultation reste cependant très problématique. Il est difficile de connaître exactement le raisonnement humain qui permet de reconnaître un piéton à partir de rares indices comme la présence d'une jambe, ou une ombre portée sur la route. Pour un sytème automatique de reconnaissance de formes, la détection d'un piéton occulté dépendra du minimum de représentation qu'il est nécessaire d'acquérir. Il est ainsi possible d'évaluer la sensibilité des méthodes à l'occultation.

La solution dépend aussi des possibilités techniques. Tandis que l'humain doit prendre sa décision sans aide extérieure, les possibilités techniques actuelles et futures peuvent également supposer des solutions originales. Dans un monde qui devient de plus en plus communicant, il devient envisageable de mettre en place des collaborations entre les différents intervenants. Par exemple, dans le cas de la détection de piétons par des systèmes embarqués, la communication entre véhicule peut résoudre le problème d'occultation lorsqu'un piéton est invisible pour un véhicule mais visible pour un autre.

1.3.4 Orientation des travaux

Dans la suite, nos travaux tenterons donc d'apporter une réponse à ces différents problèmes de variabilité.

La première approche s'appuie sur la représentation d'objets à l'aide de squelettes transformés en graphes étiquetés. Cette représentation présente de nombreuses propriétés que nous nous proposons d'exploiter. Selon les étiquettes choisies, la représentation d'un objet par un graphe est indépendante de sa taille et de sa posture. En choisissant soigneusement les étiquettes, un seul graphe de piéton permettrait de représenter l'ensemble des piétons. Le problème de variabilité de posture peut ainsi être résolu en étiquetant le graphe uniquement par des distances sur les arcs. Le résultat ne dépend alors que des distances entre les nœuds et non pas des articulations entre chaque arc. De plus, si les valeurs sont normalisées, il est également possible de résoudre le problème de variabilité à l'échelle, les longueurs d'arcs étant les mêmes quelque soit la taille du piéton. Enfin, il est possible de traiter le problème de l'occultation par l'extraction de sous-graphes. Cette méthode permet ainsi de chercher la partie correspondante d'un graphe définissant un piéton partiellement occulté.

Cette méthode suppose un travail préliminaire au niveau de la segmentation afin de pouvoir extraire de l'image les objets et leur squelette.

La deuxième approche est fondée sur une représentation globale, sans segmentation préalable. Le choix s'est porté sur la représentation d'images d'objets à l'aide d'histogrammes d'orientation de gradient. Ici, la description d'une image est indépendante de l'apparence du piéton grâce à l'utilisation d'histogrammes normalisés localement.

Le calcul des histogrammes est effectué pour un ensemble de régions permettant de découper l'image. Comme ce nombre de régions est fixé préalablement, la taille du piéton n'est pas déterminante. La méthode permet donc de résoudre le problème de variabilité d'échelle. Elle reste cependant dépendante de la qualité des images fournies initialement. Pour détecter les piétons dans l'image, il nous faudra définir une méthode capable d'extraire un ensemble d'images représentant les piétons présents dans la scène. Cette méthode devra ainsi tenir compte des contraintes liées aux variabilités d'échelle et de posture.

Nous étudierons également le pouvoir de généralisation de cette méthode de représentation afin de déterminer la réponse au problème de variabilité de posture et d'occultation. Ce pouvoir peut être renforcé par la définition d'une base d'apprentissage contenant une grande variété de piétons. En effet, l'utilisation d'un classifieur de type SVM, permet de ne retenir que les exemples les plus disciminants pour la classification.

2

Discrimination et méthodes à noyaux

« Un exemple n'est pas forcément un exemple à suivre. »

Albert Camus

Sommaire

Pour l'homme, la reconnaissance de formes n'est pas un processus inné. Il apprend ainsi tout au long de sa vie à différencier des objets, avec une certaine précision. En fonction de son expérience et de l'apprentissage reçus, l'homme va ainsi être capable de catégoriser les objets de son environnement. Nous retrouvons ce même schéma d'apprentissage dans le monde numérique.

Comme nous avons pu le voir dans le chapitre précédent (1.1), l'analyse des caractéristiques est une phase essentielle lors du processus de reconnaissance de formes. Dans ce chapitre, nous allons ainsi présenter la théorie de la décision, ses principes et ses enjeux. Nous présenterons le principe de la discrimation et l'élaboration de fonctions de décisions. Nous présenterons enfin un cas particulier des machines à noyau : le classifieur SVM pour Séparateur à Vaste Marge (*Support Vector Machine* en anglais), utilisé dans les applications de détection de piétons de ce mémoire.

2.1 Théorie de la décision

Supposons que nous souhaitions mettre en place une méthode de reconnaissance de formes embarquée dans un véhicule. Le système d'acquistion nous permet d'observer les objets présents dans la scène et génère un ensemble de données $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1,n} \in \mathscr{X} \times \mathscr{Y}$, avec $\mathscr{X} = \mathbb{R}^d$, l'espace de représentation, $\mathscr{Y} = \mathbb{R}$ l'espace des étiquettes et n le nombre de données. Chaque donnée x_i est générée par une classe C_i.

Supposons que deux classes sont définies : la classe des voitures et la classe des piétons, chaque donnée étant caractérisée par sa hauteur et sa largeur. Les données sont générées à partir de sources qui définissent ainsi la classe d'appartenance C_i des données selon des probabilités conditionnelles $\mathbb{P}(\mathbf{x}|C_i),\ i = 1,n$.

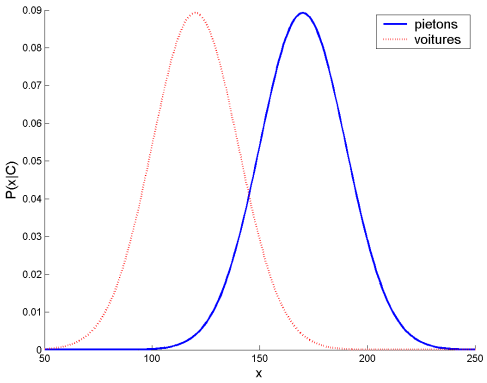


FIG. 2.1 : Illustration d'un cas idéale représentant les probabilités conditionnelles des classes voitures et piétons selon la hauteur.

Le but de la théorie Bayésienne de la décision consiste à définir un ensemble d'actions $\{a_i\}_{i=1,n}$, l'action a_i affectant la donnée $\mathbf{x}$ à la classe C_i en fonction des observations effectuées et de la règle de décision utilisée [27, 39].

$$\begin{aligned} f : \mathbb{R}^m &\longrightarrow \mathbb{R} \\ \mathbf{x} &\longmapsto a_i = f(\mathbf{x}) \end{aligned} \tag{2.1}$$

Des coûts $\ell_{(i,j)}$ sont définis pour avoir effectué l'action a_i sur la donnée $\mathbf{x}$ sachant qu'elle est générée par la classe C_j. Grace à ces coûts et en connaissant les probabilités *a posteriori* $\mathbb{P}(C = C_i|\mathbf{x})$, nous pouvons ainsi définir le risque conditionnel associé à l'action a_k :

$$R(a_k|\mathbf{x}) = \sum_{i=1}^{n} \ell_{(k,i)} \mathbb{P}(C = C_i|\mathbf{x}), \tag{2.2}$$

et le risque moyen d'une décision :

$$R(f) = \int R(f(\mathbf{x})|\mathbf{x}) \mathbb{P}(\mathbf{x}) d\mathbf{x} \tag{2.3}$$

La meilleure règle de décision f est la règle de Bayes qui minimise le risque d'une décision pour chaque donnée observée. Si la règle présente un risque minimum pour chaque donnée, alors elle présente le risque minimum par rapport à toutes les règles de décisions :

$$\begin{aligned} f_{\text{Bayes}}(\mathbf{x}) &= \operatorname*{argmin}_{f}(R(f)) \\ &= \operatorname*{argmin}_{i=1,n}(R(a_i|\mathbf{x})) \end{aligned} \tag{2.4}$$

Le choix de l'action en un point $\mathbf{x}$ revient donc à chercher l'action a_k qui présente le plus petit risque :

$$\begin{aligned} R(a_k|\mathbf{x}) &< R(a_j|\mathbf{x}) && \forall k \neq j \\ \sum_i \ell_{(k,i)} \mathbb{P}(C_i|\mathbf{x}) &< \sum_i \ell_{(j,i)} \mathbb{P}(C_i|\mathbf{x}) && \forall k \neq j \end{aligned} \tag{2.5}$$

Par exemple, dans le cas où nous disposons de 2 classes, pour un coût 0/1, l'action a_1 sera choisie lorsque :

$$\begin{aligned} & R(a_1|\mathbf{x}) \ < R(a_2|\mathbf{x}) \\ \Leftrightarrow \quad & \ell_{(1,1)}\mathbb{P}(C=C_1|\mathbf{x})+\ell_{(1,2)}\mathbb{P}(C=C_2|\mathbf{x}) \ < \ell_{(2,1)}\mathbb{P}(C=C_1|\mathbf{x})+\ell_{(2,2)}\mathbb{P}(C=C_2|\mathbf{x}) \end{aligned} \tag{2.6}$$

En supposant qu'il est plus coûteux d'effectuer une action erronée $\left(\ell_{(2,1)} > \ell_{(1,1)}\right)$, on obtient la relation du rapport de vraisemblance :

$$L(\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|C_1)}{\mathbb{P}(\mathbf{x}|C_2)} > \frac{\ell_{(2,2)}-\ell_{(1,2)}}{\ell_{(1,1)}-\ell_{(2,1)}}\frac{\mathbb{P}(C_2)}{\mathbb{P}(C_1)} = k \tag{2.7}$$

En connaissant ainsi les probabilités *a priori* $\mathbb{P}(C)$ et $P(\mathbf{x}|C)$ ainsi que les coûts ℓ, il est possible de déterminer la règle de décision en fonction du rapport de vraisemblance $L(\mathbf{x})$:

$$f(\mathbf{x}) = \begin{cases} a_1, \text{ si } L(\mathbf{x}) > k \\ a_2, \text{ si } L(\mathbf{x}) \leq k \end{cases} \tag{2.8}$$

Lorsque nous définissons un coût $0/1$, c'est à dire $\ell_{(i,i)} = 0, \ \forall i = 1,n$ et $\ell_{(i,j)} = 1, \ \forall i \neq j$, la règle de décision ne dépend alors que des probabilités *a priori* $\mathbb{P}(C_1)$ et $\mathbb{P}(C_2)$.

Dans le cas où nous disposons de plusieurs classes, le risque associé à l'action a_k est défini par :

$$\begin{aligned} R(a_k|\mathbf{x}) \ &= \sum_i \ell_{(k,i)}\mathbb{P}(C_i|\mathbf{x}) \\ &= \sum_{i\neq k} \mathbb{P}(C_i|\mathbf{x}) \\ &= 1-\mathbb{P}(C_k|\mathbf{x}) \end{aligned} \tag{2.9}$$

Nous choisirons l'action a_k lorsque le risque associé est le plus faible, c'est à dire lorsque la probabilité *a posteriori* $\mathbb{P}(C_k|\mathbf{x})$ est la plus forte.

Un exemple de méthode fondée sur la modélisation *a posteriori* est la méthode des k-ppv ou k plus proches voisins. Très intuitive et simple d'utilisation, elle est couramment utilisée comme référence avec des méthodes de discrimination plus complexes.

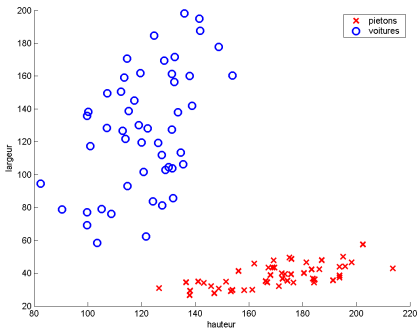


FIG. 2.2 : Représentation des données à l'aide de 2 caractéristiques : la hauteur et la largeur.

A partir d'une base d'apprentissage, le principe consiste à calculer la distance euclidienne entre un nouveau point et tous les points de cette base. La classe d'affectation est obtenue à l'aide d'un vote parmi les k plus proches voisins calculés. Le résultat est obtenu en considérant la classe des points les plus proches, c'est à dire la classe la plus probable.

En appliquant la méthode des k-ppv sur notre exemple lorsque nous disposons de deux caractéristiques (figure 2.2), nous obtenons le résultat montré sur la figure 2.3.

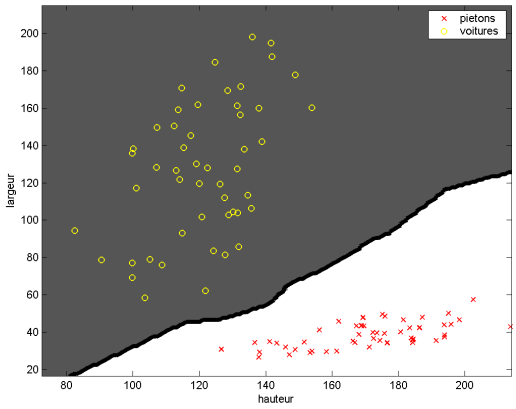


FIG. 2.3 : Fonction de décision fournie par un *k*-ppv, pour k=1, avec la base d'apprentissage vue dans la section précédente.

2.2 Modélisation de la fonction de décision

Nous allons maintenant envisager le cas où les modèles des probabilités sont inconnues, mais nous connaissons la forme des modèles ayant généré les données.

Si les données sont générées selon un modèle gaussien :

$$\begin{aligned} \mathbb{P}(\mathbf{x}|C_1) &\sim \mathcal{N}(\mu_1, \Sigma) \\ \mathbb{P}(\mathbf{x}|C_2) &\sim \mathcal{N}(\mu_2, \Sigma) \end{aligned} \tag{2.10}$$

Nous choisissons l'action a_1 si :

$$\begin{aligned} & \mathbb{P}(C_1|\mathbf{x}) > \mathbb{P}(C_2|\mathbf{x}) \\ \Leftrightarrow \quad & \mathbb{P}(\mathbf{x}|C_1)\mathbb{P}(C_1) > \mathbb{P}(\mathbf{x}|C_2)\mathbb{P}(C_2) \\ \Leftrightarrow \quad & \log\big(\mathbb{P}(\mathbf{x}|C_1)\big) + \log\big(\mathbb{P}(C_1)\big) > \log\big(\mathbb{P}(\mathbf{x}|C_2)\big) + \log\big(\mathbb{P}(C_2)\big) \end{aligned} \tag{2.11}$$

En exprimant les probabilités en fonction des paramètres du modèle de distribution nous avons :

$$\mathbf{x}^T\Sigma(\mu_1 - \mu_2) - \frac{\mu_1\Sigma^{-1}\mu_1}{2} + \frac{\mu_2\Sigma^{-1}\mu_2}{2} + \log\left(\frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}\right) > 0 \tag{2.12}$$

Nous obtenons ainsi une règle de décision linéaire pour décider l'action a_1 lorsque :

$$\mathbf{x}^T\mathbf{w} + w_0 > 0 \tag{2.13}$$

avec les cœfficients de l'hyperplan $\mathbf{w}$

$$\mathbf{w} = \Sigma(\mu_1 - \mu_2) \tag{2.14}$$

et le biais w_0

$$w_0 = -\frac{\mu_1\Sigma^{-1}\mu_1}{2} + \frac{\mu_2\Sigma^{-1}\mu_2}{2} + \log\left(\frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}\right) \tag{2.15}$$

En pratique, les cœfficients $\mathbf{w}$ et w_0 sont déterminés grâce aux observations effectuées en calculant les moyennes μ_1, μ_2 et la matrice de covariance Σ. Les probabilités $\mathbb{P}(C_1)$ et $\mathbb{P}(C_2)$ sont estimées en tenant compte du nombre d'observations dans chaque classe.

Lorsque les probabilités sont inconnues, il est possible de définir la règle de décision en estimant une fonction de décision dont le modèle est connu. Cette fonction de décision permet de définir la frontière séparant les classes

présentes. Dans cette approche, les paramètres de la fonction de décision sont appris à l'aide des données de la base d'apprentissage.

La fonction de décision sera ainsi définie :

$$\begin{aligned} f : \mathscr{X} &\longrightarrow \mathscr{Y} \\ \mathbf{x} &\longmapsto y = \text{sign}\big(f(\mathbf{x})\big) \end{aligned} \tag{2.16}$$

2.2.1 Coût et fonction objectif

Selon la règle de Bayes, la fonction de décision est celle qui minimise le coût ℓ établi selon les erreurs commises sur l'ensemble d'apprentissage. Nous comparons ainsi les valeurs obtenues pour $f(\mathbf{x}_i)$ et les valeurs des étiquettes réelles y_i par la fonction coût $\ell\big(f(\mathbf{x}),y\big)$ Il est possible d'utiliser différentes formulations pour définir la fonction coût ℓ :

– l'erreur des moindres carrés est définie par :

$$\ell(f(\mathbf{x}),y) = \|f(\mathbf{x}) - y\|^2 \tag{2.17}$$

– l'erreur charnière, pour $y \in \{-1,1\}$

$$\ell(f(\mathbf{x}),y) = \min(0, 1 - f(\mathbf{x})y) \tag{2.18}$$

– l'erreur absolue

$$\ell(f(\mathbf{x}),y) = |f(\mathbf{x}) - y| \tag{2.19}$$

– l'erreur -1/1

$$\ell(f(\mathbf{x}),y) = \big|\text{sign}\big(f(\mathbf{x})\big) - y\big| \tag{2.20}$$

S'il n'est pas possible de définir l'espérance du coût, il reste possible en pratique de calculer le risque empirique R_{emp} associé à une fonction de décision selon une fonction coût :

$$R_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{x}),y) \tag{2.21}$$

2.2.1.1 Moindres carrés

Nous allons présenter un premier exemple pour expliquer le principe de la modélisation d'une fonction de décision. Nous cherchons à modéliser les données $\mathbf{x}$ par un modèle linéaire :

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + w_0, \tag{2.22}$$

avec $\mathbf{w}$ les cœfficients du modèle et w_0 le biais.

Pour effectuer une discrimination dans le cas bi-classes, la règle de décision devient :

$$f(\mathbf{x}) = \begin{cases} a_1, \text{ si } f(\mathbf{x}) > 0 \\ a_2, \text{ si } f(\mathbf{x}) \leq 0 \end{cases} \tag{2.23}$$

Nous cherchons à minimiser l'erreur produite par la fonction de décision sur les données d'apprentissage en utilisant une fonction de coût des moindres carrés :

$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \|\mathbf{X}^T\mathbf{w} - \mathbf{y}\|^2 \tag{2.24}$$

La fonction objectif pour la méthode des moindres carrés prend ainsi en compte toutes les données afin de positionner au mieux la fonction de décision. Le critère à minimiser devient :

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i - y_i)^2 \tag{2.25}$$

Pour trouver la valeur minimum du coût, on utilise la dérivée de la fonction coût par rapport à $\mathbf{w}$, le minimum étant atteint lorsque la dérivée s'annule :

$$\begin{cases} \nabla_{\mathbf{w}} \ell = \sum_{i=1}^{n} 2\mathbf{x}_i^T \left(\mathbf{w}^T \mathbf{x}_i - y_i\right) \\ \qquad = 2\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) \end{cases} \tag{2.26}$$

$$\begin{cases} \nabla \ell_{\mathbf{w}} = 0 & \Leftrightarrow & \mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) = 0 \\ & \Leftrightarrow & \mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y} \end{cases} \tag{2.27}$$

Lorsque la matrice $\mathbf{X}^T\mathbf{X}$ est non singulière, on résoud alors le problème :

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.28}$$

Cette formulation est nécessaire, car la matrice $\mathbf{X}$ n'est pas forcément une matrice carrée et nous ne pouvons donc pas déterminer sa matrice inverse. Nous affichons le résultat obtenu pour notre exemple de discrimination piétons/voitures sur la figure 2.4.

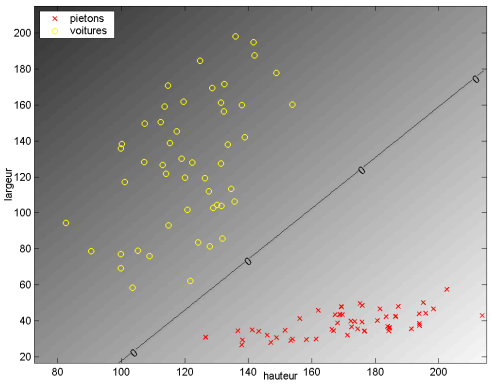


FIG. 2.4 : Fonction de décision obtenue par la méthode des moindres carrés.

2.2.1.2 Perceptron

La fonction coût du perceptron est déterminée en fonction du nombre d'erreurs commises par la fonction de décision en tenant compte de l'importance de chaque erreur. L'erreur totale est ainsi proportionnelle à la somme des distances des données mal classées par rapport à l'hyperplan de la fonction de décision :

$$\ell(\mathbf{w}) = \sum_{\mathbf{x} \in \mathscr{X}^E} (-\mathbf{w}^T \mathbf{x}), \tag{2.29}$$

avec $\mathscr{X}^E$ l'ensemble des données mal classées.

Ici la résolution de la recherche du coût minimum est effectuée par une procédure de descente de gradient. A partir d'une valeur initiale pour les cœfficients $\mathbf{w}$, le but consiste à modifier progressivement leur valeur en tenant compte de la valeur des erreurs grâce au calcul du gradient jusqu'à obtenir une convergence de la solution. L'algorithme 1 montre ainsi la résolution de cette méthode.

Initialisation($\mathbf{w}, \eta, \theta$)
i $\leftarrow$ 0
Répéter
  i $\leftarrow$ i + 1
  $\mathbf{w} \leftarrow \mathbf{w} + \eta(i) \sum_{\mathbf{x} \in \mathscr{X}^E} \mathbf{x}$
jusqu'à ce que $(|\eta(i) \sum_{x \in \mathscr{X}^E} \mathbf{x}| < \theta)$

Algorithme 1 : Algorithme de descente de gradient pour la résolution du perceptron

Il faut donc paramétrer préalablement le seuil de convergence θ, ainsi que le pas η.

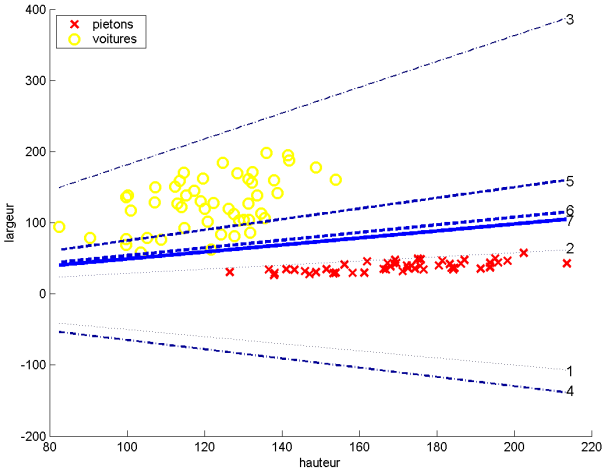


FIG. 2.5 : Evolution de la frontière de décision d'un perceptron au cours des itérations. La première fonction de décision correspond à une choix aléatoire des cœfficients de l'hyperlan.

La figure 2.5 montre l'évolution de la frontière de décision lors de la convergence. En fonction du pas et du seuil, le nombre d'itérations nécessaire peut varier. Cependant, l'initialisation ne se révèle pas déterminante, on peut donc initialiser les cœfficients de manière aléatoire.

De plus, nous constatons sur la figure 2.5 que la solution ne converge pas nécessairement à chaque itération vers la solution optimale. La solution obtenue pour une itération donnée peut s'avérer moins efficace au sens de la séparation des classes que la solution de l'itération précédente.

Il faut également noter que la solution obtenue n'est pas unique et peut varier selon les paramètres et les cœfficients du modèle initial.

2.2.1.3 Maximisation de la marge

Un des inconvénients des modèles linéaires réside dans le nombre de solutions possibles. Comme nous le voyons sur la figure 2.6, une infinité de solutions est possible dans ce cas précis.

Pour résoudre ce problème, le but consiste à définir des contraintes sur la position de l'hyperplan. Dans le cas des Séparateurs à Vaste Marge, la fonction objectif consiste à maximiser la marge séparant les différentes classes. Ici encore, le coût est défini en fonction de la distance entre la frontière de décision et les données. La fonction

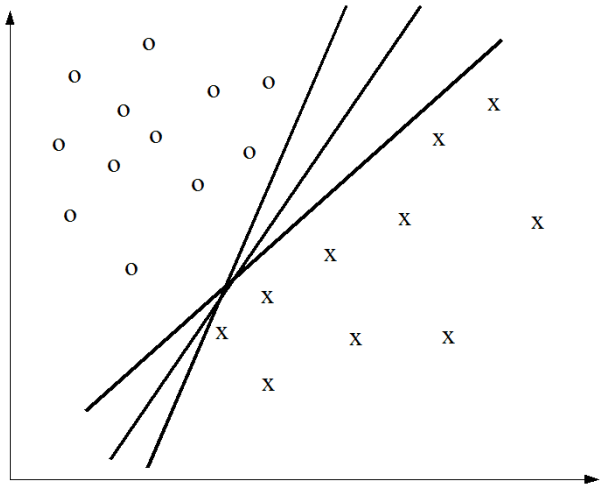


FIG. 2.6 : Exemples de fonctions linéaires parmi l'infinité de solutions possibles.

objectif est donc fondée sur la maximisation de la marge, cette contrainte permettant ainsi d'obtenir une solution unique.

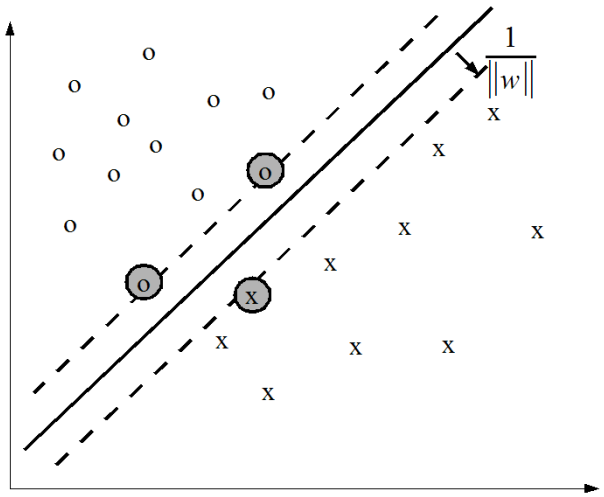


FIG. 2.7 : Fonction de décision optimale au sens de la maximisation de la marge entre les données. Les points encerclés sont les points supports (*support vector*).

Comme nous le voyons sur la figure 2.7, la taille de la marge dépend de la norme du vecteur des cœfficients de l'hyperplan et est égale à $\frac{2}{\|\mathbf{w}\|}$.

Maximiser la marge revient donc à minimiser $\frac{1}{2}\|\mathbf{w}\|$ ou de façon quadratique :

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \tag{2.30}$$

La fonction de décision doit également classer correctement les données d'apprentissage :

$$y_i \cdot f(\mathbf{x}_i) \geq 1, \quad \forall i = 1,n \tag{2.31}$$

Nous formulons donc le problème SVM de la façon suivante :

$$\begin{cases} \min & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{sous la contrainte} & y_i(\mathbf{x}_i^T\mathbf{w} + w_0) \geq 1, \quad i = 1,n \end{cases} \tag{2.32}$$

L'utilisation du Lagrangien nous permet de redéfinir ce problème, en cherchant à minimiser le Lagrangien par rapport aux cœfficients $\mathbf{w}$ et w_0 et le maximiser par rapport aux multiplicateurs de Lagrange α.

$$\begin{cases} \min\limits_{\mathbf{w},w_0} \max\limits_{\alpha} \mathscr{L}_P(\mathbf{w},w_0,\alpha) = \dfrac{1}{2}\|\mathbf{w}\|^2 - \sum\limits_{i=1}^{n} \alpha_i \left[y_i\left(\mathbf{x}_i^T\mathbf{w}+w_0\right)-1\right] \\ \qquad\qquad \text{sous la contrainte} \quad \alpha_i \geq 0, \quad i=1,n \end{cases} \tag{2.33}$$

Pour déterminer le point selle, nous calculons les dérivées partielles du Lagrangien :

$$\begin{cases} \nabla_{\mathbf{w}}\mathscr{L}_P = \mathbf{w} - \sum\limits_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \\ \nabla_{w_0}\mathscr{L}_P = -\sum\limits_{i=1}^{n} \alpha_i y_i \end{cases} \tag{2.34}$$

$$\begin{cases} \nabla_{\mathbf{w}}\mathscr{L}_P = 0 \Leftrightarrow \mathbf{w} = \sum\limits_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \\ \nabla_{w_0}\mathscr{L}_P = 0 \Leftrightarrow \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{2.35}$$

En utilisant les conditions de Karush-Kuhn-Tucker, nous obtenons la formulation duale et réecrivons donc l'équation 2.33 :

$$\begin{cases} \max\limits_{\alpha} \mathscr{L}_D(\alpha) & = \dfrac{1}{2}\sum\limits_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \sum\limits_{j=1}^{n} \alpha_j y_j \mathbf{x}_j - \sum\limits_{i=1}^{n} \left[\alpha_i y_i \mathbf{x}_i^T \mathbf{w} \alpha_i + \alpha_i y_i w_0 - \alpha_i \right] \\ & = \dfrac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum\limits_{i=1}^{n} \alpha_i - w_0 \sum\limits_{i=1}^{n} \alpha_i y_i \\ & = -\dfrac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum\limits_{i=1}^{n} \alpha_i \end{cases} \tag{2.36}$$

En écrivant sous la forme vectorielle :

$$\begin{cases} \max\limits_{\alpha} \mathscr{L}_D(\alpha) = & \alpha^T \mathbf{H} \alpha + \alpha \mathbf{1}^n \\ \text{sous la contrainte} & \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \\ \text{et} & \alpha_i \geq 0\,, \quad i=0,n \end{cases} \tag{2.37}$$

avec $\mathbf{H} = (y^T y)K$, K étant la matrice de Gram telle que $K(i,j) = \mathbf{x}_i^T \mathbf{x}_j$.

Nous aboutissons ainsi à la maximisation d'un problème quadratique, dont le problème QP peut être résolu en utilisant, par exemple, la méthode des contraintes actives [58].

Les points dont la valeur de α est nulle ne participent pas à la solution. Les autres points sont les points supports de la fonction de décision (figure 2.7).

2.3 SVM : cas général

Nous allons maintenant présenter le classifieur SVM dans le cas général, tout d'abord le cas linéaire non séparable, puis le cas non-linéaire.

2.3.1 Cas linéaire non séparable

La section précédente présente la résolution de la classification SVM, lorsque les classes définies par le problème sont parfaitement distinctes. La mise au point d'une frontière de décision peut donc être effectuée simplement, la contrainte $y_i\left(\mathbf{x}_i^T\mathbf{w}+w_0\right)\geq 1$, pour $i=1,n$ étant facilement accessible.

Cependant, les problèmes réels présentent rarement cette configuration, mais plutôt un recouvrement entre les classes, plus ou moins fort selon les cas d'étude. Pour résoudre ce problème de recouvrement, une variable de relâchement ξ est ajoutée. Cette variable permet ainsi de définir la gravité des erreurs opérées pendant la phase d'apprentissage. La contrainte de classification correcte des données devient donc :

$$f(\mathbf{x}_i)y_i \geq 1-\xi_i, \quad i=1,n \tag{2.38}$$

En tolérant des erreurs de classification en apprentissage, la fonction de décision sera moins complexe, et son pouvoir de généralisation sera ainsi conservé. La généralisation peut être définie comme la capacité de la fonction de décision à deviner la distribution sous-jacente des classes. L'accroissement de la complexité de la fonction de décision provoque cependant une diminution des capacités de cette fonction. Il ne faut pas perdre de vue, que le but d'une fonction de décision est de prédire l'appartenance de nouvelles données complètement inconnues. Une adéquation trop forte avec les données d'apprentissage peut diminuer le pouvoir de généralisation de la fonction de décision.

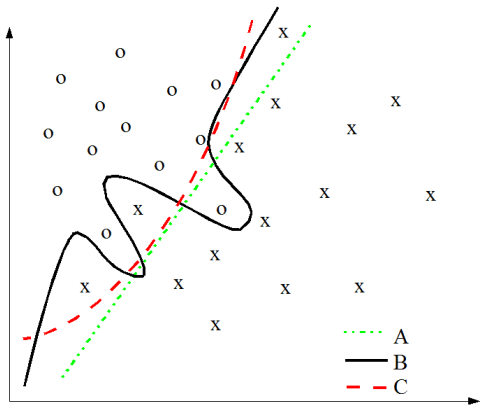


FIG. 2.8 : Fonctions de décision, (A) : fonction linéaire, commet de nombreuses erreurs en apprentissage. (B) : fonction interpolante séparant correctement les données d'apprentissage, avec un risque de sur-apprentissage. (C) : solution intermédiaire, peu d'erreurs en apprentissage, avec une bonne capacité de généralisation.

La figure 2.8 montre ainsi différentes solutions possibles selon différentes complexités. Le meilleur compromis est une fonction intermédiaire entre la fonction linéaire et l'interpolation des données d'apprentissage. Cette dernière permet notamment d'illustrer le phénomène de sur-apprentissage, lorsque la fonction colle parfaitement aux données d'apprentissage, mais ne permet pas par conséquent d'effectuer une bonne décision pour les nouvelles données. Elle ne peut en effet donner une décision que sur les données déjà apprises. Sa capacité de généralisation est très faible comparativement aux deux autres solutions illustrées.

Pour quantifier l'importance du relâchement, la variable C est utilisée. Il est ainsi possible de paramétrer l'importance apportée à la solution par les variables mal classées. Nous montrons un exemple sur la figure 2.9 où la frontière de décision admet des données mal classées.

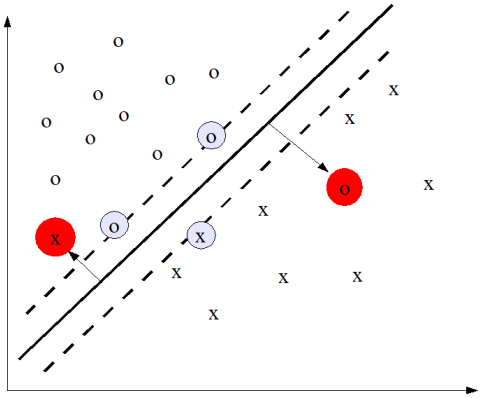


FIG. 2.9 : Exemple de points d'apprentissage mal classés. La variable de relâchement de l'équation 2.39, permet de leur accorder une moindre importance.

Le problème peut se définir de la façon suivante :

$$\begin{cases} \min\limits_{\mathbf{w},w_0} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{n}\xi_i, & \\ \text{avec} & f(\mathbf{x}_i)y_i \geq 1-\xi_i, & i=1,n \\ \text{et} & \xi_i \geq 0\ , & i=1,n \end{cases} \tag{2.39}$$

Le Lagrangien nous permet de reformuler ce problème de minimisation sous contraintes :

$$\begin{cases} \min\limits_{\mathbf{w},w_0}\max\limits_{\alpha,\beta} = \dfrac{\|\mathbf{w}\|^2}{2} - \sum\limits_{i=1}^{n}\alpha_i\left[y_i\left(\mathbf{x}_i^T\mathbf{w}+w_0\right)-1\right] + C\sum\limits_{i=1}^{n}\xi_i - \sum\limits_{i=1}^{n}\beta_i\xi_i \\ \text{avec} \quad \alpha_i \geq 0, \quad i=1,n \\ \text{et} \quad \beta_i \geq 0, \quad i=1,n \\ \text{et} \quad \xi_i \geq 0, \quad i=1,n \end{cases} \tag{2.40}$$

Pour déterminer le point selle, nous calculons les dérivées partielles du Lagrangien :

$$\begin{cases} \nabla_{\mathbf{w}}\mathscr{L}_P = \mathbf{w} - \sum\limits_{i=1}^{n}\alpha_i y_i x_i = 0 & \Leftrightarrow & \mathbf{w} = \sum\limits_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \\ \nabla_{w_0}\mathscr{L}_P = -\sum\limits_{i=1}^{n}\alpha_i = 0 & \Leftrightarrow & \sum\limits_{i=1}^{n}\alpha_i = 0 \\ \nabla_{\xi_i}\mathscr{L}_P = C - \sum\limits_{i=1}^{n}\xi_i - \sum\limits_{i=1}^{n}\beta_i = 0 & \Leftrightarrow & \sum\limits_{i=1}^{n}\beta_i = C - \sum\limits_{i=1}^{n}\xi_i \end{cases} \tag{2.41}$$

La dérivée partielle du Lagrangien par rapport à ξ_i dans l'équation 2.41, nous permet de redéfinir la contrainte sur les multiplicateurs β en modifiant la contrainte sur les multiplicateurs α par rapport à la variable C.

Les conditions des Karush-Kuhn-Tucker d'un problème de minimisation sous contraintes s'écrivent donc :

$$\begin{cases} \mathbf{w} = \sum\limits_{i=1}^{n}\alpha_i y_i x_i \\ \sum\limits_{i=1}^{n}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases} \tag{2.42}$$

Nous formulons ainsi le dual du Lagrangien :

$$\begin{cases} \max\limits_{\alpha} \mathscr{L}_D(\alpha) = -\dfrac{1}{2}\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i \\ \text{sous les contraintes} \quad \displaystyle\sum_{i=1}^{n}\alpha_i y_i = 0 \\ \text{et} \quad 0 \leq \alpha_i \leq C\,, \quad i = 0,n \end{cases} \tag{2.43}$$

En écrivant sous la forme vectorielle :

$$\begin{cases} \max_{\alpha} \mathscr{L}_D(\alpha) = \alpha^T\mathbf{H}\alpha + \alpha\mathbf{1}^n \\ \text{sous les contraintes} \quad \displaystyle\sum_{i=1}^{n}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C\,, \quad i = 0,n \end{cases} \tag{2.44}$$

2.3.2 Extension de la formulation SVM

2.3.2.1 Stratégies multiclasses

Lorsque plusieurs classes sont définies dans le problème initial, une stratégie multi-classes est définie. Il existe principalement deux approches en utilisant des classifieurs binaires [43] :

2.3.2.1.1 un contre un Pour n classes, $\dfrac{n(n-1)}{2}$ classifieurs sont entrainés, chacun opposant une classe à une autre. Pour classer des nouvelles données, celles-ci sont testées pour chaque classifieur et la classe finale est attribuée par un vote.

2.3.2.1.2 un contre tous Pour n classes, n classifieurs sont entrainés. Les ensembles d'apprentissage sont constitués par une seule classe pour les exemples positifs, toutes les autres classes sont définies comme exemples négatifs. Les nouvelles données sont classées selon la prédiction la plus forte donnée parmi tous les classifieurs.

2.3.2.2 One-class

Une variante de la formulation SVM propose ainsi de définir un classifieur dont l'hyperplan contient la frontière d'une seule classe [40, 89]. La figure 2.10 présente un exemple de frontière obtenue.

L'intérêt de cette méthode permet ainsi de résoudre le difficile problème de la conception de la base d'apprentissage, notamment pour la définition des exemples négatifs. L'idée originale a ainsi permis d'appliquer les SVM à la détection de nouveautés en ligne.

La figure 2.10 illustre ainsi l'hyperplan obtenu dans l'espace des caractéristiques. Cet hyperplan peut se résumer à la plus petite sphère englobant les données contenues dans la classe. Les points supports de la classe sont positionnés sur le contour de la sphère.

La fonction de décision s'écrit alors :

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n}\alpha_i\mathbf{x}_i^T\mathbf{x} - \rho\right) \tag{2.45}$$

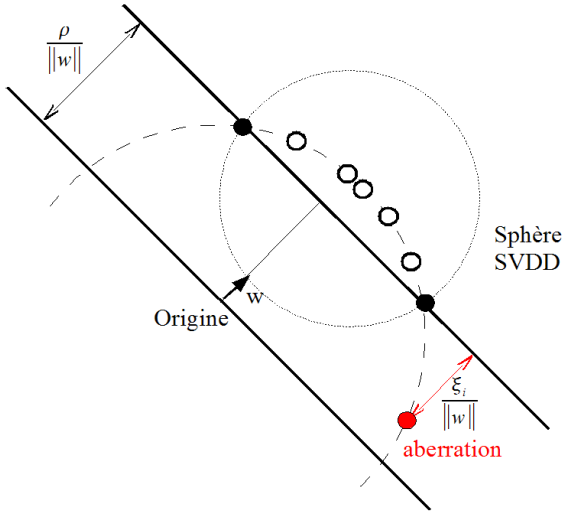


FIG. 2.10 : Schéma du classifieur une classe dans l'espace des caractéristiques.

Elle est obtenue en minimisant :

$$\begin{cases} \min\limits_{\mathbf{w},w_0,\xi,\rho} & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{1}{\nu}\sum\limits_{i=1}^{n}\xi_i, \\ \text{avec} & \mathbf{x}_i^T\mathbf{w} \geq \rho - \xi_i, \quad i = 1,n \\ & \xi_i \geq 0, \quad i = 1,n \end{cases} \tag{2.46}$$

La résolution aboutit au problème dual :

$$\begin{cases} \min\limits_{\mathbf{w},w_0,\xi,\rho} & \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j \\ \text{avec} & 0 \leq \alpha_i \leq \frac{1}{\nu}, \quad i = 1,n \\ \text{et} & \sum\limits_{i=1}^{n}\alpha_i = 1, \quad i = 1,n \end{cases} \tag{2.47}$$

2.3.3 Cas non linéaire

Nous avons considéré jusqu'à présent que la résolution des problèmes était effectuée de façon linéaire. Cependant cette linéarité n'est pas garantie, et des problèmes peuvent ne pas être résolus de cette façon.

La résolution d'un problème non linéaire peut ainsi s'effectuer en projetant les données dans un espace de grande dimension à l'aide d'une fonction Φ, d'où sont récupérées ensuite d'autres données permettant de résoudre le problème linéairement comme le montre la figure 2.11.

En associant pour chaque exemple $\mathbf{x}_i$ pour $i = 1,n$ la fonction $\Phi_i(\mathbf{x}) = k(\mathbf{x},\mathbf{x}_i)$ la fonction noyau k permet de transformer le problème dans un nouvel espace $\mathscr{H}$, appelé espace des caractéristiques.

Soit $k(.,.)$ une fonction à deux variables, symétrique et positivie. A partir de cette fonction, nous construisons l'ensemble de ses combinaisons linéaires $\mathscr{H}$:

$$\mathscr{H} = \left\{ f : \mathscr{X} \to \mathscr{Y} \mid \exists n \in \mathbb{N}, \alpha \in \mathbb{R}^n, \{\mathbf{x}_i\}_{i=1,n} \in \mathscr{X}^n \; ; \; f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i k(\mathbf{x},\mathbf{x}_i) \right\} \tag{2.48}$$

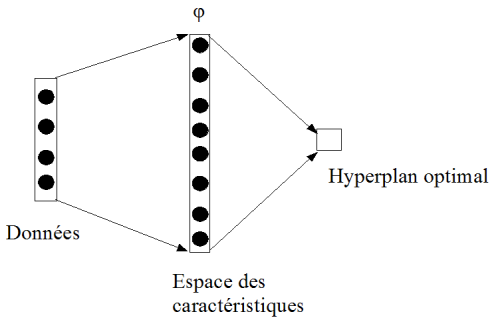


FIG. 2.11 : Projection des données par l'intermédiaire d'une fonction Φ.

Le problème SVM non linéaire peut alors être défini de la façon suivante :

$$\begin{cases} \min\limits_{f,w_0} & \frac{1}{2}\|f\|^2_{\mathscr{H}} \\ \text{avec} & y_i(f(\mathbf{x}_i)+w_0) \geq 1, \quad \forall i = 1,n \end{cases} \tag{2.49}$$

L'équation de la maximisation du dual dans l'équation 2.43 peut alors s'écrire de la façon suivante :

$$\begin{cases} \max\limits_{\alpha} & \mathscr{L}_D(\mathbf{w},w_0,\alpha,\beta) = -\frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) + \sum\limits_{i=1}^{n}\alpha_i \\ \text{avec} & \sum\limits_{i=1}^{n}\alpha_i y_i = 0 \\ \text{et} & 0 \leq \alpha_i \leq C, \quad i = 1,n \end{cases} \tag{2.50}$$

Le théorème de la représentation nous permet ainsi de redéfinir la fonction de décision qui devient :

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n}\alpha_i y_i k(\mathbf{x}_i,\mathbf{x}) + w_0\right) \tag{2.51}$$

2.4 Noyaux

Dans cette section nous allons définir formellement le noyau et les propriétés des espaces associés à une fonction noyau.

Définition - Produit scalaire : un espace vectoriel $\mathscr{X}$ est doté d'un produit scalaire s'il existe une forme bilinéaire $\langle .,. \rangle$ symétrique et à valeurs réelles telle que $\langle x,x \rangle > 0$, pour $x \in \mathscr{X}$ et $x \neq 0$.

Le produit scalaire vérifie les propriétés suivantes pour $f,g,h \in \mathscr{X}, \alpha \in \mathbb{R}$:
- $\langle f,g \rangle = \langle g,h \rangle$
- $\langle f+g,h \rangle = \langle f,h \rangle + \langle g,h \rangle$
- $\langle \alpha f,g \rangle = \alpha \langle f,g \rangle$
- $\langle f,f \rangle \geq 0 \Leftrightarrow f = 0$

Définition - Norme : la norme d'un espace $\mathscr{X}$ peut être défini en fonction du produit scalaire. Pour $f \in \mathscr{X}$:

$$\|f\|_{\mathscr{X}} = \langle f,f \rangle^{\frac{1}{2}} \tag{2.52}$$

Définition - Espace complet : un espace vectoriel $\mathscr{X}$ est dit complet si toute séquence de Cauchy $\{h_n\}_{n\geq 1}$ d'éléments de $\mathscr{X}$ converge vers un élément de $h \in \mathscr{X}$. La séquence de Cauchy est une suite dont les éléments se

rapprochent et vérifie la propriété suivante :

$$\sup_{m>n} \|h_n - h_m\|_{\mathscr{X}} \underset{n\to\infty}{\longrightarrow} 0 \tag{2.53}$$

Définition - Espace de Hilbert : un espace de Hilbert $\mathscr{H}$ est un espace vectoriel complet doté d'un produit scalaire.

Définition - Fonctionnelle d'évaluation :

soit $\mathscr{H}$ un espace de Hilbert, une fonctionnelle d'évaluation A est une fonction linéaire qui évalue une fonction $f \in \mathscr{H}$ au point $\mathbf{x}$:

$$\begin{aligned} A_{\mathbf{x}} : \mathscr{H} &\longmapsto \mathbb{R} \\ A_{\mathbf{x}} f &= f(\mathbf{x}) \end{aligned} \tag{2.54}$$

A est continue s'il existe m tel que :

$$|f(\mathbf{x})| \leq m\|f\|_{\mathscr{H}}, \quad \forall \mathbf{x} \tag{2.55}$$

Définition - Espace de Hilbert à noyau reproduisant : un espace de Hilbert à noyau reproduisant (*Reproducing Kernel Hilbert Space, RKHS*), est un espace de Hilbert de fonction à valeurs réelles sur un domaine $\mathscr{X}$, qui vérifie la propriété de continuité de la fonctionnelle d'évaluation A_x pour chaque x.

$$g \in V(f) \Rightarrow g(x) \in V(f(x)) \tag{2.56}$$

Propriété : Reproduction

soit $\mathscr{H}$ un RKHS de noyau k, la propriété de reproduction de Aronszajn [1] est définie pour $f \in \mathscr{H}$ par :

$$f(\mathbf{x}) = \langle k(\mathbf{x},.), f(.)\rangle_{\mathscr{H}} \tag{2.57}$$

En particulier :

$$k(\mathbf{x},\mathbf{x}') = \left\langle k(\mathbf{x},.), k(\mathbf{x}',.)\right\rangle_{\mathscr{H}} \tag{2.58}$$

Définition - Noyau : un noyau k peut donc être interprété comme étant un produit scalaire (équation 2.58) ou la manière d'évaluer une fonction en un point (équation 2.57).

Définition - Noyau défini positif : un noyau $k(x,x')$ sur $\mathscr{X} \times \mathscr{X}$ est dit positif
- s'il est symétrique $\forall x,\ x' \in \mathscr{X} : k(x,x') = k(x',x)$,
- s'il vérifie $\forall n \in \mathbb{N}, \forall \alpha_i \in \mathbb{R}, \forall x_i : \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(x_i,x_j) \geq 0$

Définition - Matrice de Gram : soit $k(x,x')$ un noyau positif sur $\mathscr{X}$ et $\{x_i\}_{i=1,n}$ un ensemble de points de $\mathscr{X}$. La matrice de Gram K associée au noyau k est la matrice de taille $n \times n$ de terme général :

$$K_{ij} = k(x_i,x_j) \tag{2.59}$$

Théorème - Théorème de la représentation : le théorème de la représentation (*Representer Theorem*) [49], nous permet de redéfinir l'expression de la fonction de décision dans le cas non-linéaire. Soit $\mathscr{H}$ un espace de Hilbert à noyau reproduisant et ω une fonction strictement mononote croissante, le problème de minimisation :

$$\min_{f\in\mathscr{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i),y_i) + \lambda\,\omega(\|f\|_{\mathscr{H}}) \tag{2.60}$$

a une solution de la forme :

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i,x) \tag{2.61}$$

2.4.1 Exemples de noyaux

Nous allons maintenant nous intéresser à quelques exemples de noyaux simples, définis sur $\mathbb{R}^d$. Parmi les noyaux présentés, les plus populaires aujourd'hui sont les noyaux gaussiens et affine. Le noyau gaussien est un noyau radial ou stationnaire, il est invariant en translation. Parmi les noyaux radiaux, nous trouvons également les noyaux Laplacien et rationnel. Le noyau polynômial et le noyau affine sont des noyaux projectifs et utilisent un produit scalaire entre les variables. La figure 2.12 montre les différences existant entre les noyaux.

Nom	$k(x,x')$
Gaussien	$exp^{-\frac{\|x'-x\|^2}{\sigma}}$
Laplacien	$exp^{-\frac{\|x-x'\|}{\sigma}}$
Rationnel	$1-\frac{\|x'-x\|^2}{\sigma+\|x'-x\|^2}$
χ^2	$exp\left(-\frac{1}{\sigma}\sum_i \frac{(x_i-x'_i)^2}{x_i+x'_i}\right)$
Polynômial	$(x^T x')^p$
Affine	$(x^T x'+\sigma)^p$

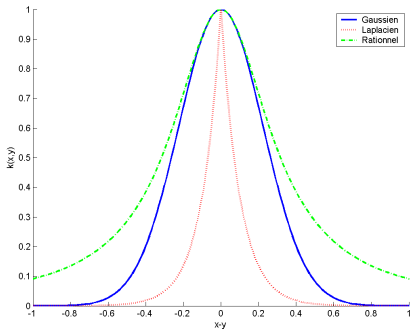


FIG. 2.12 : Exemples de valeurs de noyau radial obtenus pour sigma = 0.1.

Le terme σ est un paramètre appelé « largeur de bande ». Comme nous pouvons le remarquer sur la figure 2.13, selon sa valeur il permet de définir l'intensité d'influence du noyau. Son paramétrage s'avère très important dans la pratique, car il influe directement sur la qualité du modèle appris.

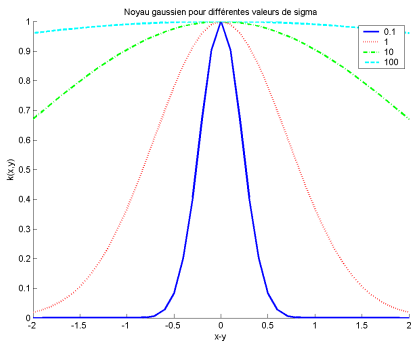


FIG. 2.13 : Différence pour un noyau gaussien selon la largeur de bande σ

2.4.2 Création d'un nouveau noyau

Selon les applications, il peut être nécessaire de définir un nouveau noyau. Les noyaux définis ci-dessus sont définis pour des données appartenant à $\mathbb{R}^d$ et ne peuvent donc traiter, par exemple, des chaines de caractères [56]. Il est donc parfois nécessaire de devoir redéfinir la notion de noyau entre des données appartenant à un espace différent.

Pour pouvoir utiliser cette nouvelle définition de noyau, il est important de garantir la positivité du noyau. Certaines propriétés sont donc intéressantes car elles garantissent la conservation de la positivité lorsque nous effectuons des combinaisons linéaires entre des noyaux eux-mêmes définis positifs [2, 18, 66].

Soient k_1 et k_2 des noyaux de $\mathscr{X} \times \mathscr{X}$, $0 \leq \lambda \leq 1$, $a \geq 0$, f une fonction à valeurs réelles dans $\mathscr{X}$, $\Phi : \mathscr{X} \to \mathscr{R}^d$ une fonction de projection, k_3 une fonction noyau de $\mathscr{R}^d \times \mathscr{R}^d$ et B une matrice $n \times n$ symétrique définie semi positive. Les fonctions suivantes sont également des noyaux :

- $k(x,x') = \lambda k_1(x,x') + (1-\lambda)k_2(x,x')$
- $k(x,x') = a \cdot k_1(x,x')$
- $k(x,x') = k_1(x,x') \times k_2(x,x')$
- $k(x,x') = f(x)f(x')$
- $k(x,x') = k_3(\Phi(x),\Phi(x'))$
- $k(x,x') = x'Bx$

Si le noyau est créé sans utiliser les concepts ci-dessus, il reste possible de vérifier sa positivité en pratique en vérifiant les valeurs propres de la matrice de Gram obtenue. Si toutes les valeurs sont positives, alors le noyau est défini positif.

L'utilisation de différents noyaux au sein d'un unique classifieur permet ainsi de combiner différentes représentations d'un même objet à reconnaître. Si les caractéristiques extraites sont de natures différentes, il est alors possible d'optimiser les performances globales du système en définissant des fonctions noyaux spécifiques à chaque type de caractéristique [83].

2.5 Conclusion

Dans ce chapitre, nous présenté la théorie de la décision. A partir d'observations de données, le but de la théorie bayésienne de la décision consiste à définir une règle de décision optimale permettant d'affecter une classe à une donnée.

Si les probabilités *a priori* des classes sont inconnues, nous pouvons utiliser des données dont la classe est connue afin de construire une règle de décision.

Dans le cas des approches sans modèle, l'apprentissage supervisé permet ainsi d'inférer une fonction de décision sans connaissance du modèle des données.

Lorsque le problème est non-linéaire, les machines à noyaux permettent de le résoudre grâce au noyau. Le noyau permet de définir un produit scalaire entre les données afin de pouvoir appliquer des algorithmes de classification.

Le classifieur SVM *Support Vector Machine* est un classifieur de type machine à noyau et constitue l'état de l'art actuel en matière de discrimination. Nous utiliserons cet algorithme dans les deux prochaines chapitres qui présentent nos travaux.

3

Noyau de graphe

« Le monde n'a peut être pas de sens, mais il a des structures, et tout est là. »

Jean-Claude Clari

Sommaire

La première partie de ce mémoire a présenté l'état de l'art concernant la détection de piétons. Comme nous l'avons souligné, une des difficultés majeures de la reconnaissance de piétons est liée avant tout à sa très grande variabilité . Dans cette partie nous proposons donc une réponse au problème de variabilité par l'utilisation des graphes.

L'idée consiste donc à représenter les objets présents à l'aide de graphes étiquetés. Cependant, pour effectuer la reconnaissance de formes par un classifieur, les graphes ne sont pas utilisables directement, aucun classifieur actuel n'étant capable de gérer ce type de données. Nous devons donc redéfinir le noyau du classifieur, c'est à dire un noyau de graphes.

Dans cette partie, nous définirons tout d'abord la théorie des graphes, présenterons leurs propriétés, leur conception. Nous décrirons ensuite le noyau de graphe, en l'occurence la formulation définie par Kashima et une extension que nous proposons. Enfin nous présenterons quelques résultats obtenus sur une application en stéréovision et sur une base d'images étiquetées manuellement.

3.1 Méthode de graphe

Depuis quelques années, l'utilisation de graphes dans de nombreux domaines révèle un engouement de plus en plus fort.

Les graphes sont utilisés comme outil en automatisme pour les réseaux de petri ou les grafcet qui représentent un déroulement séquentiel d'une application. Les graphes sont également utilisés en base de données pour la représentation conceptuelle des données et dans les systèmes d'information géographiques. Dans le domaine de la bio-informatique, les graphes permettent de manipuler des représentations moléculaires. Dans le cadre de l'analyse de textes, les graphes permettent de représenter les documents afin de mettre en évidence l'agencement spatial d'un document.

Comme nous pouvons le constater, le but principal des graphes est de permettre une représentation structurée des données. Il est ainsi possible de représenter et de manipuler plus aisément des objets complexes.

Nous proposons ici d'utiliser les graphes comme outil de représentation d'images. Nous allons donc dans un premier temps rappeler la notion de graphe.

3.1.1 Théorie des graphes

La théorie des graphes est attribuée à Leonhard Euler, notamment pour le problème des sept points de Königsberg. Plus récemment, Berge et al. a défini la théorie moderne des graphes [5], qui nous permet de spécifier formellement un graphe et présenter quelques notions relatives à la théorie des graphes.

Définition - Graphe non orienté : Soit N un ensemble de nœuds ou sommets, A un ensemble d'arêtes connectant des paires non ordonnées de nœuds tel que $A \subset N \times N$. Un graphe G est constitué d'un ensemble de nœuds et d'arcs. Un graphe est donc représenté de la façon suivante : $G(A,N)$.

Sur la figure 3.1, les nœuds sont nommés 1,2,3,4,5 et les arcs a,b,c,d,e.

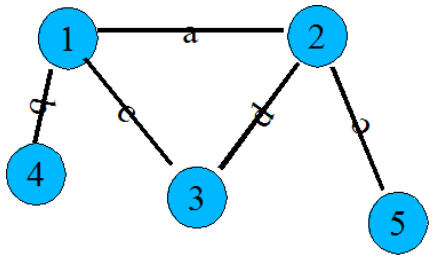


FIG. 3.1 : Exemple de graphe.

Définition - Sous-graphe : Un graphe $G_s(A_s,N_s)$ est un sous-graphe de $G(A,N)$ si G_s est contenu intégralement dans G, c'est à dire que tous les nœuds de G_s sont présents dans G, et que les relations A_s sont conservées dans G : $A_s \subset A$ et $N_s \subset N$.

Définition - Graphe orienté :

Soit N un ensemble de sommets, A un ensemble d'arcs connectant des paires ordonnées de nœuds tel que $A \subset N \times N$. Un graphe orienté est représenté de la façon suivante : $G(A,N)$ (figure 3.2).

Définition - Graphe valué : Il est possible de valuer un graphe, c'est à dire d'attribuer à ses arcs et nœuds, un

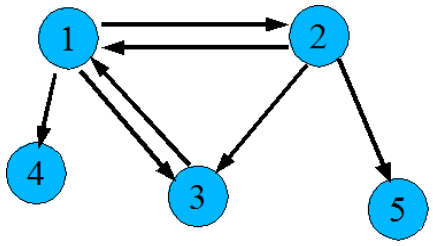


FIG. 3.2 : Exemple de graphe orienté.

ensemble de valeurs. Soit δ_a l'ensemble des étiquettes des arcs, δ_n l'ensemble des étiquettes des nœuds. G est un graphe étiqueté et se présente sous la forme : $G(A,N,\delta_a,\delta_n)$

Les étiquettes peuvent être de simples scalaires, ou bien des vecteurs appartenant à $\mathbb{R}^d$. Certaines applications nécessitent en outre des chaînes de caractères ou des images.

Définition - Adjacence :

Deux sommets sont dits adjacents ou voisins lorsqu'ils sont reliés par un arc. Le voisinage d'un nœud est constitué par l'ensemble de ses voisins. La matrice d'adjacence A d'un graphe G contenant n nœuds, est une matrice carrée de taille $n \times n$, de terme général :

$$A_{ij} = \begin{cases} 1, & \text{s'il existe un arc entre le noeud } i \text{ et le noeud } j \\ 0, & \text{sinon} \end{cases} \tag{3.1}$$

Définition - Ordre et degré : L'ordre d'un graphe G est le nombre de nœuds de ce graphe, c'est à dire la taille du graphe. Il est noté $|G|$.

Le degré d'un nœud représente la taille du voisinage de ce nœud. Le degré maximum d'un graphe G, noté $\Delta(G)$ est égal au degré maximum de l'ensemble des nœuds de G. Le degré minimum d'un graphe G, noté $\delta(G)$ est égal au degré minimum de l'ensemble des nœuds de G.

Définition - Graphe Complet : Un graphe est complet si tous les nœuds sont reliés deux à deux. Soit A la matrice d'adjacence du graphe, $\forall i,j\ A_{ij} = 1$

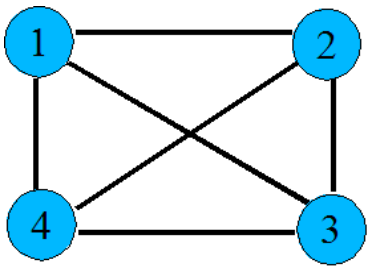


FIG. 3.3 : Graphe complet, tous les nœuds sont reliés deux à deux.

Définition - Graphe connexe : Un graphe G est dit connexe si et seulement si il existe un chemin pour toute paire de nœuds (n_i,n_j) du graphe. La figure 3.4 présente un exemple de graphe non connexe. En effet, les nœuds 1, 3 et 4 sont inaccessibles depuis les nœuds 2 et 5, et réciproquement.

Définition - Chemin : Le chemin h d'un graphe G est un sous-graphe de G. Un chemin est un parcours du graphe. Il est présenté sous la forme d'une liste ordonnée de nœuds adjacents présents sur ce chemin. Un chemin de longueur L sera présenté sous cette forme $h = \{n_1,n_2,...n_L\}$.

Le parcours $h = \{n_1,n_2,...n_L\}$ est dit *eulérien* si chaque nœud du chemin n'apparait qu'une seule fois : $\forall i \neq$

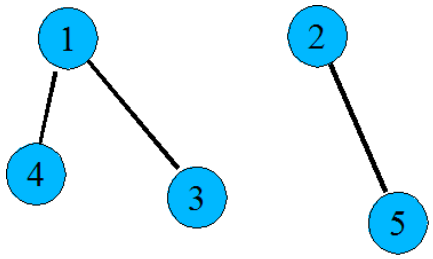


FIG. 3.4 : Exemple de graphe non connexe.

$j, n_i \neq n_j$.

Définition - Cycle :

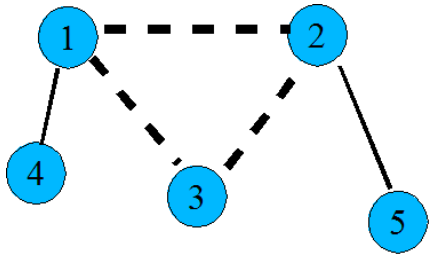


FIG. 3.5 : Graphe cyclique, le cycle est représenté par les arcs en traits pointillés.

Un cycle est un chemin dont les extrémités sont confondues (figure 3.5). Un graphe G est dit *acyclique* s'il ne contient aucun cycle. Ainsi, il existe un unique chemin entre deux nœuds du graphe G.

3.1.2 Les graphes et la reconnaissance de formes

Nous avons évoqué brièvement quelques exemples d'application des graphes. Nous allons maintenant nous attacher plus particulièrement aux applications concernant la reconnaissance de formes. Dans ces méthodes, les graphes sont utilisés comme moyen de représentation. Les graphes permettent en effet une représentation compacte et structurée des données. Les méthodes présentées ici proposent différentes manières de définir les composants du graphes : les nœuds et les arcs.

Dans le cas de la reconnaissance d'objets [19, 48, 73, 79], l'application principale concerne l'indexation d'images. Pour déterminer la classe d'appartenance du graphe, on le compare à d'autres graphes d'une base d'apprentissage à l'aide de fonctions de similarité.

Le papier de Wiskott et al. [96], présente une application pour la reconnaissance de visages utilisant des graphes. Le but consiste à modéliser un visage contenu dans une image à l'aide d'un graphe étiqueté. Le graphe est construit en cherchant des points particuliers du visage qui sont définis comme nœuds dans le graphe. Chaque nœud est étiqueté par un vecteur contenant les valeurs d'une décomposition en ondelettes, permettant ainsi de décrire le voisinage du nœud. Les arcs sont également étiquetés et apportent l'information de distance entre chaque nœud.

D'autres méthodes [69, 75], cherchent à découper une image en régions. Un graphe est établi sur l'ensemble de l'image, chaque région étant un nœud du graphe, les arcs définissant alors la relation d'adjacence entre les régions.

Dans [69], l'idée consiste donc à mettre en correspondance des graphes et de calculer ensuite leur similarité. Celle ci est déterminée à partir de la similarité entre les régions des images, d'une part, et d'autre part, sur la similarité entre les sous-graphes. Cette mesure permet ainsi de classer une image requête parmi un ensemble d'images contenues dans la base.

Dans le cas qui nous intéresse ici, c'est à dire la reconnaissance de piétons, certains travaux proposent une modélisation de l'être humain par des graphes [45, 99]. Le but revient à caractériser un humain par différentes parties : jambes, buste, tête, bras, et de représenter ensuite leurs connexions par un graphe.

3.1.3 Intérêt des graphes

Nous avons pu constater que nombreuses méthodes font appel à l'utilisation de graphes, et ce, dans les domaines les plus variés. La raison de cet engouement est du aux propriétés des graphes. Nous allons maintenant présenter quelques propriétés majeures qui justifient *a priori* l'emploi des graphes dans le cas de la reconnaissance de formes à l'aide d'images.

3.1.3.1 Propriétés topologiques

Nous pouvons retenir des propriétés intéressantes qui nous permettent de justifier l'utilisation des graphes. En effet, un graphe permet d'obtenir une représentation invariante en échelle, en rotation et en apparence.

Quelle que soit sa taille, un graphe conservera la même topologie. En effet, en augmentant ou en réduisant de façon homogène la taille d'un objet, la forme générale de l'objet ne subit pas de modification. Sachant que le graphe décrit justement cette forme, les différentes parties de la forme conservent le même agencement spatial et ne modifient donc pas la structure de graphe.

De plus, le graphe est invariant en rotation. Pour les mêmes raisons que le redimensionnement de l'objet, la rotation d'un objet ne modifie pas l'adjacence et l'agencement des différentes parties de l'objet, ce qui conserve la structure du graphe.

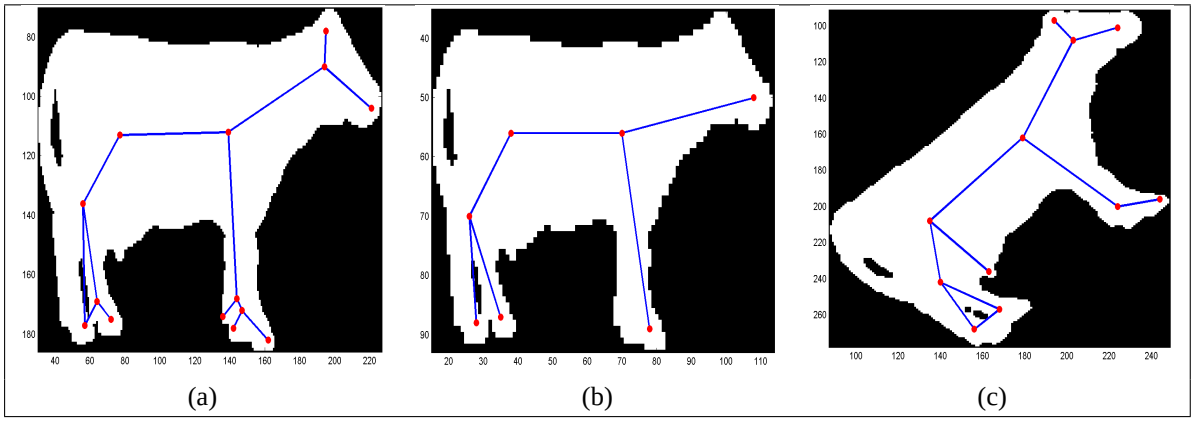


FIG. 3.6 : (a) : graphe original, (b) graphe de l'image redimensionnée par un facteur d'échelle $\frac{1}{2}$, (c) graphe de l'image pivotée de 45°

La figure 3.6 montre ainsi un exemple d'invariance en échelle et en rotation. La structure générale du graphe n'est pas modifiée, seule quelques branches sont ajoutées ou supprimées. Ces variations proviennent des modifications apportées aux images par les transformations de redimensionnement ou de rotation.

Enfin, le graphe est invariant en translation. Si l'objet subit un déplacement quelconque dans l'image, le graphe est conservé, car il n'est conçu qu'en utilisant l'objet lui-même, sans tenir compte de l'information de voisinage [38].

Le graphe possède donc des propriétés naturelles pour ces invariances, mais il faut cependant veiller à ne pas les annihiler en valuant le graphe à l'aide d'étiquettes incompatibles avec la notion d'invariance. Par exemple, utiliser la position fixe d'un nœud dans l'image va à l'encontre des trois invariances : redimensionnement, rotation

et translation.

La phase de valuation doit donc tenir compte de certaines contraintes afin d'optimiser l'utilisation des graphes pour la reconnaissance de formes.

3.1.3.2 Valuation du graphe

Les principales méthodes pour la reconnaissance de formes utilisent la transformation d'un objet en graphe par l'intermédiaire de son squelette [19, 48, 73]. Le squelette est transformé en graphe, auquel sont ajoutées par la suite des étiquettes permettant d'apporter de l'information au graphe.

La valuation du graphe permet d'ajouter des informations de différentes natures dans les arcs et les nœuds. Cela représente un avantage majeur, puisque nous sommes ainsi capables de mélanger des caractéristiques décrivant la forme elle-même avec des caractéristiques apportant une information sur le contenu de la forme, telle que sa texture ou sa couleur.

Dans [72], plusieurs étiquettes sont ainsi attribuées aux arcs permettant d'apporter un maximum d'information sur la structure de l'objet. Cependant, la multi valuation des graphes n'est pas complètement exploitée, puisque le problème se pose lors de la définition d'une méthode de correspondance entre les graphes.

3.1.3.3 Intérêt dans le cas de la détection de piétons

Comme nous l'avons évoqué plus haut, les graphes permettent une représentation structurée et compacte des données. C'est une représentation utilisable sur des images.

Dans le cadre de la détection de piétons, les propriétés des graphes vis-à-vis des invariances en échelle et en rotation nous permettent de justifier *a priori* le choix des graphes. Ces propriétés sont en effet une réponse possible aux problèmes de variabilité du piéton en échelle et en posture.

Concernant le problème de variabilité d'apparence du piéton, la réponse peut être fournie par la description de la forme du piéton. Si des étiquettes apportent une information d'une autre nature, il faudra veiller à ce qu'elle soit compatible avec cette invariance.

3.1.4 Construction des graphes à partir d'images

Nous allons maintenant aborder la construction de graphes. Cette partie nous permettra ainsi de présenter les différentes étapes permettant d'obtenir un graphe étiqueté à partir d'une image.

Comme nous l'avons évoqué plusieurs fois, le graphe permet de représenter la forme d'un objet. Il faut donc extraire la structure du graphe à partir de la forme de l'objet et définir la notion de nœud et d'arc.

Dans notre cas, nous supposons disposer du masque binaire de l'objet, c'est à dire une liste de l'ensemble des pixels composant l'objet dans l'image. Ce masque peut être obtenu de différentes manières :
- découpage région, avec éventuellement des combinaisons de différentes régions,
- définition manuelle du masque,
- calcul d'une carte de disparité à partir d'un système stéréoscopique.

Dans nos exemples, nous utiliserons soit la définition manuelle du masque qui nous permettra de valider la méthode de graphe, soit le calcul de la disparité pour une application en stéréovision.

Dans les deux cas, nous disposons de l'image de l'objet et son masque binaire. Pour définir un graphe, nous utilisons en premier lieu le squelette de l'objet. Nous allons tout d'abord définir la notion de squelette. L'étape

suivante consiste à transformer le squelette en graphe. Nous verrons comment définir les nœuds et les arcs pour construire la structure générale du graphe. Enfin, comme nous l'évoquions précédemment, l'un des principaux intérêts du graphe réside dans l'apport d'informations à travers les étiquettes des nœuds et des arcs. La dernière étape consiste donc à étiqueter les graphes à partir de différents types d'information.

3.1.4.1 Squelettisation

Cette première étape consiste donc à extraire un squelette à partir du masque binaire de l'image.

La notion de squelette a été introduite par Blum en 1967 [9]. L'objectif est de représenter un objet en minimisant la quantité d'information, sous une forme simple à extraire et à manipuler. Pour présenter cette méthode, Blum fait l'analogie avec le feu de prairie : "*Soit une prairie couverte de manière homogène par de l'herbe sèche et X un ensemble de points de cette prairie. Au départ, tous les point du contour de X sont enflammés simultanément. Le feu se propage de manière homogène et s'étend à travers la prairie à une vitesse constante. le squelette de l'ensemble de points X est défini comme le lieu des points ou les fronts enflammés se sont rencontrés.*"

Le squelette peut également être défini comme l'axe médian de l'objet. La squelettisation répond à certaines contraintes : elle doit conserver la topologie, préserver la géométrie de l'objet et avoir une épaisseur nulle en tout point.

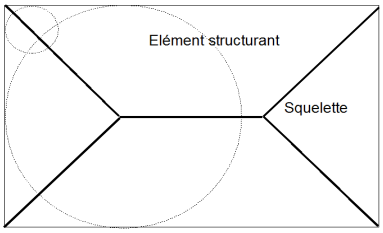


FIG. 3.7 : Exemple de squelette, les cercles représentent les disques maximaux en deux points.

Certaines propriétés sont également souhaitées : la méthode doit être insensible aux rotations et aux redimensionnement. Le squelette doit permettre de reconstituer la forme originale.

Définition - Elément structurant : un élément structurant B_k est un ensemble possédant une forme géométrique de taille**Définition** - Boule géodésique : une boule géodésique est un élément structurant de forme circulai**Définition** - Squelette : Serra [77] définit le squelette de la manière suivante : le squelette d'un objet X, $S_k(X)$ est l'union des centres des boules géodésiques maximales B contenues dans X (figure 3.7).

Le squelette peut être obtenu de différentes manières :
– amincissements successifs,
– calcul d'une carte de distance,
– ouverture et conservation des résidus.

Dans le premier cas, nous allons opérer de façon successive un amincissement de l'objet, jusqu'à obtenir un squelette correspondant aux propriétés citées ci-dessus. Dans le deuxième cas, nous calculons tout d'abord une carte de distance qui permet de donner la distance euclidienne de chaque pixel de l'objet par rapport à la frontière de l'objet. Le squelette est obtenu en cherchant les maxima locaux.

Le dernier cas correspond à la définition du squelette par Lantuéjoul [53]. Le squelette $S(X)$ d'un objet X est défini par l'union des différences entre l'érosion de X par un élément structurant de taille k et de l'ouverture adjointe de l'érosion de X :

$$S(X) = \cup_k \left[E(X,B_k) - O(E(X,B_k),B_k))\right] \tag{3.2}$$

avec B_k l'élément structurant de taille k, $E(X,B)$ l'érosion de X par un élément structurant B, $O(X,B)$ l'ouver-

ture de X par un élément structurant B.

La définition d'un squelette s'appuie sur les résidus. Cette méthode n'est cependant pas optimale, car elle ne préserve pas la topologie de l'objet. Nous étudierons donc la squelettisation par amincissements successifs et calcul de la carte de distance.

3.1.4.1.1 Amincissements successifs Tout d'abord, nous présentons un squelette obtenu par amincissements successifs (figure 3.9). L'amincissement η est le résidu de la transformation tout ou rien de la translation φ de l'objet par un élément structurant donné :

$$\eta(X) = X - \varphi(X) \tag{3.3}$$

Cette opération est accomplie jusqu'à obtenir l'idempotence du squelette, c'est à dire que l'objet original est son propre amincissement.

1	1	1
x	1	x
0	0	0

a

x	1	1
0	1	1
0	0	x

b

FIG. 3.8 : Valeur des pixels du voisinage de l'elément structurant original (a) , et pivoté de 45˚(b). x = 0,1.

Les éléments structurants doivent préserver l'homotopie de l'objet. Deux objets sont homotopes si une transformation continue permet de passer de l'un à l'autre. La figure 3.8 montre un exemple d'élément structurant. Afin d'effectuer l'amincissement dans toutes les directions possibles, il faut composer une famille d'éléments à partir de celui-ci. Cette famille est ainsi constituée par les éléments provenant de la rotation de l'élément original.

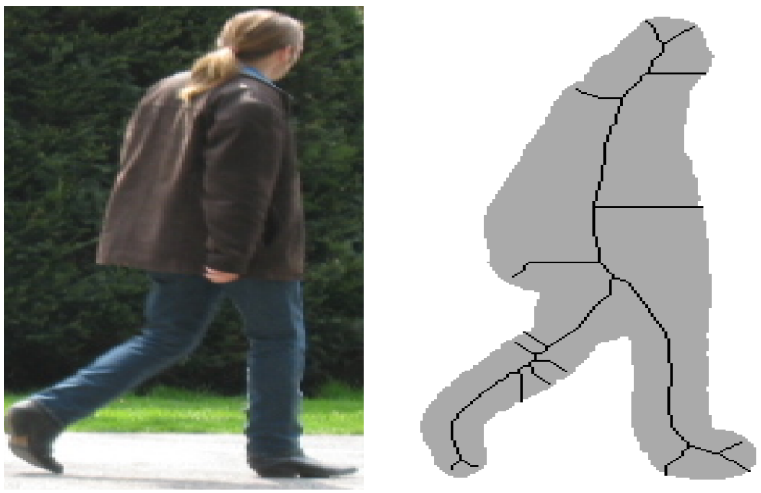


FIG. 3.9 : Exemple de squelette obtenu par amincissements successifs.

Sur la figure 3.9 nous présentons ainsi le résultat d'une squelettisation par amincissements successifs.

3.1.4.1.2 Carte de distance La deuxième méthode de squelettisation que nous étudions utilise le calcul d'une carte de distance. Elle utilise notamment la formulation de Hamilton pour mesurer les flux, issus de la propagation du feu de prairie dans la formulation de Blum. Les points de rencontre des différents flux de propagation sont ainsi les points du squelette.

Dans [25], il est démontré que ce flux peut être approximé par la distance euclidienne entre chaque pixel de l'objet et le bord de l'objet (figure 3.10). Nous calculons donc tout d'abord la carte de distance C.

Pour déterminer les points du squelette, il faut ensuite calculer la divergence du flux (figure 3.10). Le but étant de déterminer les points ne répondant pas à la contrainte de conservation de l'énergie.

a b

FIG. 3.10 : Distance euclidienne pour chaque pixel de l'objet (a) et image de la divergence du gradient de l'image de distance obtenue (b).

La divergence permet ainsi de mesurer la tendance du flux à converger ou diverger en un point. Elle est obtenue en calculant le gradient de C, en x et en y.

$$div(C) = \frac{\partial C_x}{\partial x} + \frac{\partial C_y}{\partial y}, \tag{3.4}$$

avec x et y les coordonnées cartésiennes de l'espace 2D.

Un flux négatif correspond à une perte d'énergie, c'est à dire lorsque le gradient du voisinage d'un pixel est défini selon des directions différentes. En utilisant l'analogie avec le feu de prairie, il s'agit du point de rencontre de plusieurs fronts de flammes. Ce pixel est alors un point du squelette.

Un simple seuillage permettrait donc d'extraire le squelette. Cependant, comme le montre la figure 3.11, la valeur du seuil est difficile à fixer. Si le seuil est trop faible, le squelette sera non-connecté, et inversement, lorsque le seuil est trop élevé, le squelette contiendra trop de points.

a b

FIG. 3.11 : Seuillage de la divergence. Pour un seuil trop faible (a), le squelette obtenu est non-connexe, pour un seuil trop élevé (b), l'épaisseur du squelette est trop importante.

L'idée proposée par Siddiqi et al. [25, 79], consiste donc à utiliser un seuil faible, mais en ajoutant des contraintes pour conserver la connexité du squelette. Pour cela, il faut considérer localement chaque point de l'objet qui est supérieur à ce seuil et vérifier que sa suppression ne conduit pas à la non-connexion du squelette.

Pour chaque pixel, dont la valeur de divergence est négative, un graphe local est établi entre les pixels voisins (figure 3.12). Sur cette figure, la suppression du pixel central conduirait par exemple à la non-connexion du squelette. Nous conservons donc ce pixel en tant que pixel du squelette.

0	1	0
1	1	0
1	1	0

a

0	0	1
1	1	0
1	1	0

b

FIG. 3.12 : Exemple de voisinage. Dans le cas (a), la mise à zéro du pixel central conserve la connexité du graphe de voisinage, dans le cas (b), la mise à zéro du pixel central ne conserve pas la connexité du graphe de voisinage.

La figure 3.13, montre le résultat obtenu pour cette méthode de squelettisation. Comme nous pouvons le constater, la structure comporte globalement moins de branches, le squelette permet donc de représenter la forme de l'objet, sans être perturbé par du bruit contenu dans les contours de l'objet.

FIG. 3.13 : Squelette obtenu avec la méthode de carte de distance.

3.1.4.2 Du squelette au graphe

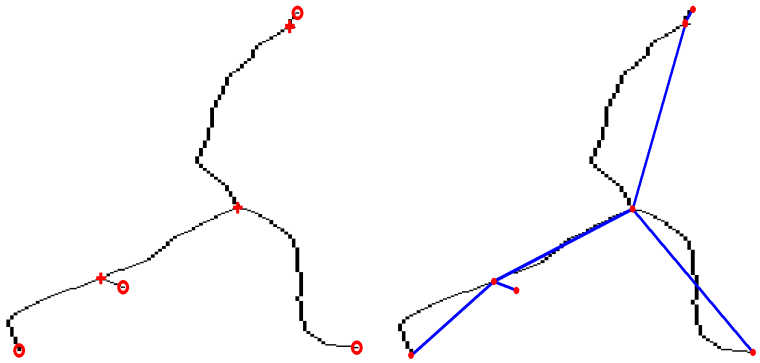
Le squelette est donc un moyen assez simple de représenter une forme générale, mais il est assez difficile de le comparer directement à d'autres squelettes. Nous proposons donc de le transformer en graphe, comme cela a déja été fait dans d'autres applications [72].

0	1	0
0	**1**	0
0	0	0

0	1	0
0	**1**	1
0	1	0

FIG. 3.14 : Exemple de pixels particuliers d'un squelette : extrémité d'une branche (gauche) et intersection entre plusieurs branches (droite). Ces pixels sont définis comme nœuds dans le graphe.

Le principe de la transformation est assez simple. Certains pixels du squelette sont particuliers comme le montre les figures 3.14 et 3.15. En effet, ces pixels sont situés soit à l'extrémité d'une branche du squelette, soit à l'intersection de plusieurs branches. Dans le premier cas, le pixel du squelette ne possède dans son voisinage proche qu'un seul autre pixel appartenant au squelette. Dans le deuxième cas, le voisinage du pixel contient d'autres pixels du squelette qui sont disposés selon une forme particulière : une intersection. Dans les deux cas, ces pixels seront pris en compte dans le graphe comme étant des nœuds du graphe. Ces nœuds sont reliés entre eux par des arcs, c'est à dire des branches du squelette.



a b

FIG. 3.15 : (a) : pixels particuliers du squelette correspondant à des nœuds dans le graphe obtenu figure (b).

Pour détecter tous ces pixels particuliers, il suffit de parcourir le squelette et analyser le voisinage du pixel. En fonction du nombre de pixels à 1 dans le voisinage, le pixel sera reconnu comme nœud ou non.

Une fois que tous les nœuds du graphes sont découverts, il faut les relier entre eux afin de pouvoir reconstituer les arcs du graphe. La méthode utilisée est analogue à celle présentée par Di Ruberto et al. [72]. En démarrant d'un nœud du squelette, il faut chercher tous les chemins possibles partant de ce nœud, qui rejoignent d'autres nœuds en passant par les pixels du squelette.

Le résultat de la transformation d'une image en graphe est montrée figure 3.15.

Nous obtenons ainsi un graphe G, tel que $G(N,A)$, avec N l'ensemble des nœuds et A, l'ensemble des arcs du graphe. Ce graphe est ainsi une autre représentation du squelette, mais n'apporte pas d'information supplémentaire.

3.1.4.3 Elagage

Comme nous l'avons présenté auparavant, les graphes issus d'images sont constitués après une squelettisation du masque binaire. Cette fonction influe sur la forme générale du squelette et donc du graphe obtenu.

Le graphe permet notamment de représenter la forme globale de l'objet, mais apporte aussi de l'information sur les formes locales de l'objet. Cette information est présente principalement dans les plus petites branches du squelette. Si trop de branches sont présentes, la surabondance d'information devient du bruit et il peut alors être nécessaire de nettoyer le squelette afin d'éliminer les branches inutiles. Cet élagage permet également de réduire la taille du graphe et donc de gagner par la suite du temps de calcul. En effet, dans la pratique, l'utilisation de la méthode de Kashima présentée dans la section 3.2.2.1, a été impossible sur des graphes de plus de 40 nœuds avec l'implémentation que nous avons utilisée.

Dans [72], l'élagage est effectué en supprimant les branches dont la taille est inférieure à un seuil, calculé selon la taille des objets. Nous avons donc défini une méthode permettant d'imposer une distance minimale entre les nœuds. Pour cela, nous regroupons les nœuds séparés par une distance inférieure à un seuil donné. L'intérêt principal de cette méthode réside dans la suppression limitée à des petites branches, c'est à dire la suppression du bruit ou de l'information locale. L'algorithme 2 présente le déroulement de l'élagage.

```
G : graphe ;
seuil : entier ;
D ← calculerDistanceEntreNoeuds(G) ;
Tant que ( n ≠ ∅) faire
    {n} ← donnerNoeudsProches(D,seuil) ;
    G ← fusionnerNoeudsLesPlusProches(G,{n}) ;
    D ← mettreAJour(D) ;
Fait
```

Algorithme 2 : Reduction de la taille d'un graphe par fusion des nœuds dont la distance est inférieure à un seuil.

Sur la figure 3.16, nous présentons un exemple d'élagage du graphe lorsque la distance minimale varie. Dans ce cas, nous ne contrôlons pas la taille obtenue pour le graphe.

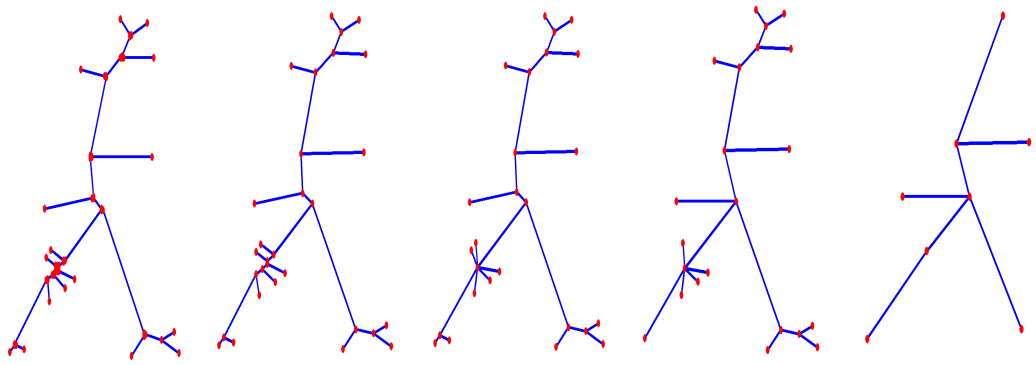


FIG. 3.16 : Exemple d'élagage d'un graphe selon une distance minimale entre chaque nœuds : graphe original (a), 2 pixels (b), 5 pixels (c), 10 pixels (d) et 20 pixels (e).

Nous proposons donc une deuxième méthode pour définir une taille maximale des graphes. Dans ce cas, nous élaguons de façon itérative le graphe, en regroupant deux à deux les nœuds les plus proches, jusqu'à obtenir une taille de graphe acceptable comme le montre l'algorithme 3. Dans ce cas, nous n'imposons pas de distance minimale entre les nœuds.

```
G : graphe ;
D ← calculerDistanceEntreNoeuds(G) ;
Tant que ( |G| > tailleMaximale) faire
    n1, n2 ← donnerNoeudsLesPlusProches(G) ;
    G ← fusionnerNoeudsLesPlusProches(G,n1,n2) ;
    D ← mettreAJour(D) ;
Fait
```

Algorithme 3 : Reduction de la taille d'un graphe par fusion des nœuds les plus proches.

La figure 3.17 présente un exemple d'élagage lorsque la taille maximale du graphe varie. Comme nous le constatons, la forme générale du graphe est conservée car l'élagage supprime prioritairement les petites branches.

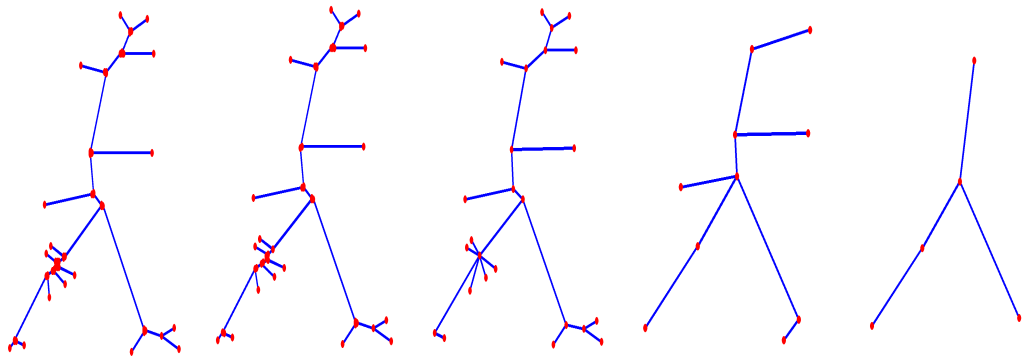


FIG. 3.17 : Exemple d'élagage d'un graphe selon la taille maximale : graphe original (a), 50 nœuds (b), 25 nœuds (c), 10 nœuds (d) et 5 nœuds (e).

3.1.4.4 Etiquetage

L'étape suivante consiste donc à ajouter d'autres informations, c'est à dire d'étiqueter le graphe. L'étiquetage consiste à valuer les arcs et les nœuds, de façon à obtenir un graphe $G(N,A,\delta_N,\delta_A)$, avec δ_N les étiquettes des nœuds et δ_A, les étiquettes des arcs.

L'étiquetage inclut différentes caractéristiques du graphe. Tout d'abord, les étiquettes possèdent des informations sur la structure de graphe. Pour caractériser la géométrie du graphe, les informations importantes sont principalement situées sur les arcs, qui déterminent ainsi la forme générale du graphe. Chaque caractéristique est extraite pour chaque arc comme le montre la figure 3.18.

Nous pouvons calculer des informations sur la forme de l'objet, c'est à dire la topologie de l'objet, sa taille son épaisseur. Nous pouvons également récupérer des caractéristiques de l'apparence de l'objet.

Ces informations sont ensuite utilisées comme étiquettes dans les nœuds δ_N et les arcs δ_A.

Parmi les caractéristiques retenues pour δ_A nous avons :
- la longueur de l'arc (L),
- la longueur de la branche du squelette (s) associée à l'arc,
- l'orientation (θ) de l'arc par rapport à l'orientation générale de l'objet : l'axe principal de l'ellipse contenant l'intégralité de l'objet,
- l'aire (A) du voisinage du squelette. Ce voisinage est construit en utilisant en chaque point du squelette la taille de l'élément structurant ou la distance euclidienne avec le contour de l'objet,
- l'écart (e) maximal entre l'arc et la branche du squelette,
- le rapport entre la longueur de l'arc et la longueur de la branche du squelette ($\frac{s}{L}$).

Pour les étiquettes δ_N des nœuds, il est possible de récupérer différentes informations :
- la taille du squelette, c'est à dire l'épaisseur de l'objet, où se situe le nœud (E),
- les coordonnées du nœud ($[X,Y]$),
- les informations du voisinage du nœud (moyenne des niveaux de gris : μ_{ng}, variance : σ_{ng}). La taille du voisinage est la taille de plus grand élément structurant contenu dans la forme de l'objet à l'emplacement du nœud ou la distance avec le contour le plus proche.

Nous avons choisi d'utiliser des étiquettes appartenant à $\mathscr{R}^d$ pour des raisons pratiques. Cependant, comme nous le verrons dans la section 3.2, les étiquettes peuvent être de nature différente.

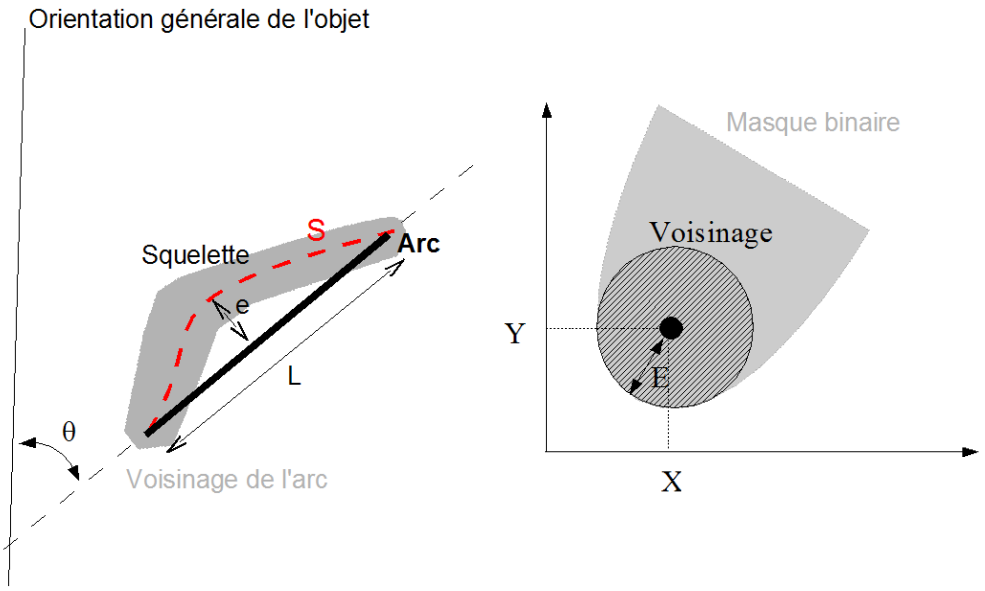


FIG. 3.18 : Extraction des caractéristiques d'un arc et d'un nœud

3.2 Comparaison de graphes

Comme nous l'avons évoqué précédemment, de nombreux domaines utilisent les graphes comme outil de représentation. Nous nous intéressons ici au domaine de la reconnaissance de formes en utilisant des images.

De nombreuses applications de graphes ont pour but l'indexation [19, 48, 69, 73, 79]. A partir d'un ensemble d'exemples constituant la base d'apprentissage, la catégorie d'un nouveau graphe est obtenue en le comparant avec les graphes de la base d'apprentissage. Pour comparer les graphes entre eux, il est nécessaire de définir une mesure de similarité. Cependant, les graphes sont des structures relativement complexes et ne permettent pas l'utilisation des distances euclidiennes, par exemple.

3.2.1 Graph Matching

Pour calculer la similarité entre graphes, il est possible d'utiliser du *graph matching* [86], littéralement effectuer la mise en correspondance de graphes. Cette approche consiste à évaluer les différences entre les graphes au niveau des nœuds et des arcs afin de pouvoir comparer les graphes entre eux.

Il existe deux types de mise en correspondance de graphes :
– la correspondance exacte,
– la correspondance inexacte.

La correspondance exacte est fondée sur le principe de l'isomorphisme de graphe. Le but consiste à chercher une fonction de projection d'un graphe à l'autre, sans modifier leur structure. La principale contrainte réside dans la conservation des liaisons entre les nœuds.

Un exemple de cette correspondance est la recherche du plus grand sous graphe commun [13]. La mesure est de la forme :

$$d(G,G') = 1 - \frac{|MSG(G,G')|}{max(|G|,|G'|)} \tag{3.5}$$

Où $MSG(G,G')$ représente le plus grand sous-graphe commun. Le principe consiste à rechercher le plus grand sous-graphe commun présent dans les deux graphes et de comparer l'ordre de ce sous-graphe avec l'ordre des graphes originaux.

La correspondance inexacte permet de mettre en correspondance des graphes dont la topologie est différente. La distance est obtenue par le coût de transformation d'un graphe à l'autre.

Dans [75] la distance entre graphes est mesurée à l'aide de la méthode *edit-distance*. Le principe consiste à appliquer différentes opérations sur les nœuds et les arcs d'un graphe G afin que celui-ci corresponde à un graphe G'. Il est ainsi possible de supprimer, valuer les arcs et les nœuds présents ou bien d'en insérer de nouveaux.

La similarité est évaluée par le nombre et l'importance des transformations. Le score obtenu pour la transformation d'un graphe en un autre est ainsi comparé avec les scores obtenus en comparant avec d'autres graphes. Le score le plus faible correspondra au graphe le plus proche, qui possède donc la structure répondant le mieux aux critères d'isomorphisme de graphes.

L'inconvénient majeur de cette solution réside dans la non symétrie de la solution [75]. En effet, les transformations pour transiter d'un graphe vers un autre ne sont pas forcément symétriques, et les coûts associés à ces transformations ne sont donc pas les mêmes.

Enfin, dans [72], la distance utilisée s'appuie sur la méthode de Gold et Rangajaran [35]. L'objectif est de calculer une matrice de permutation pour transformer un graphe en un autre. Le principe est d'affecter graduellement les nœuds d'un graphe aux nœuds du second graphe.

L'objectif est de trouver une matrice de permutation M qui minimise un coût :

$$\min_M E_{G,G'}(M) = -\frac{1}{2}\sum_{a=1}^{A}\sum_{a'=1}^{l}\sum_{b=1}^{A}\sum_{b'=1}^{l} M_{aa'}M_{bb'}C_{aa'bb'} \tag{3.6}$$

avec M la matrice de permutation :

$$M_{aa'} = \begin{cases} 1, & \text{si le nœud a de G correspond au nœud a' de G'} \\ 0, & \text{sinon} \end{cases} \tag{3.7}$$

et C_{aibj} la mesure de similarité entre l'arc (a,b) du graphe G et l'arc (a',b') du graphe G' :

$$C_{aa'bb'} = \begin{cases} 0, & \text{si } G_{ab} \text{ ou } G'_{a'b'} \text{ est nul,} \\ L\left(1 - \left|\frac{G_{ab}}{L} - \frac{G'_{a'b'}}{L}\right|\right) & \text{sinon} \end{cases} \tag{3.8}$$

avec $L = \max\left(\frac{1}{2}\sum_{a=1}^{A}\sum_{b=1}^{B} G_{ab}, \frac{1}{2}\sum_{a'}^{A'}\sum_{b'}^{B'} G'_{a'b'}\right)$, une mesure entre les arcs, G_{a_b} la valeur de la matrice d'adjacence du graphe G entre les nœuds a et b.

Cependant, cette méthode est relativement complexe. Elle nécessite en outre l'utilisation de paramètres qui influent sur la convergence du résultat obtenue par «recuit simulé» [95].

3.2.2 Noyau de graphes

Nous venons d'étudier différentes méthodes permettant de comparer des graphes entre eux. Des solutions existent pour définir des mesures de similarités entre graphes et sont utilisables dans des applications telles que l'indexation.

Nous désirons cependant étudier l'application de graphes en utilisant d'autres méthodes de classification de type machines à noyau. Dans ce cas, il est nécessaire de définir un produit scalaire entre graphe.

Les fonctions de *graph matching* présentées dans la section précédente ne sont pas toujours adaptées. Tout d'abord, certaines fonctions ne possèdent pas les propriétés du produit scalaire, comme par exemple la symétrie

avec la méthode *edit-distance* [75]. La fonction proposée par Gold et al. [35] impose une résolution par une méthode de type «recuit simulé», dont la convergence dépend de paramètres difficiles à définir.

Enfin, l'approche de Bunke et al. [13] est fondée sur la recherche du plus grand sous graphe commun. Il faut donc définir une méthode de recherche de sous-graphe.

La définition d'un produit scalaire entre graphes est confrontée à la notion même de graphe. En effet, un graphe est une structure complexe regroupant un ensemble de données pouvant être de natures différentes.

La définition d'un noyau calculé uniquement à partir des informations contenues dans les étiquettes des nœuds et des arcs est possible, mais ne prend pas en compte l'organisation du graphe. Or cette structure de graphe apporte une information pertinente en définissant les relations entre chaque élément du graphe. L'objectif consiste donc à tenir compte de cette information structurelle.

Récemment, plusieurs travaux ont proposé des noyaux constitués d'ensembles non-ordonnés [50, 94]. L'idée proposée par Wallraven et al. [94], consiste ainsi à définir un noyau mineur entre chaque élément des ensembles et de regrouper ensuite les résultats fournis par les noyaux mineurs en un noyau de plus haut niveau qui définit ainsi un produit scalaire entre les deux ensembles.

Une méthode de comparaison de graphes appuyée sur la comparaison des étiquettes de nœuds et d'arc peut donc être assimilée à des sacs de nœuds et d'arcs. En étendant cette idée, nous pouvons définir des sacs de chemins pour comparer deux graphes. En effet, un chemin issu d'un graphe permet de définir un sous-graphe et structure les étiquettes des nœuds et arcs présents sur ce chemin. En constituant les sacs avec suffisamment de chemins, il est possible d'obtenir l'intégralité du graphe et donc la structure complète du graphe.

3.2.2.1 Méthode de Kashima

Une première approche de sacs de chemins est proposée par Kashima et al. [46, 47].

Sa démarche s'appuie sur les travaux précédents de Tsuda, Smola et Gärtner [37, 88, 93] qui proposent des noyaux pour des séquences.

L'idée revient donc à considérer un graphe comme un ensemble de séquences, chaque séquence est issue d'une marche aléatoire dans le graphe. Le but est donc de comparer diverses marches aléatoires entre les graphes afin de calculer le noyau final.

Soit G et G' deux graphes étiquetés. Un chemin h de longueur L est une séquence de L nœuds et $L-1$ arcs :

$$h = (n_1, a_1, n_2, a_2, ...n_L); \tag{3.9}$$

Le noyau $K(G,G')$ est obtenu en comparant tous les chemins h et h' issus des graphes G et G' à l'aide du noyau K_c. La comparaison des chemins est pondérée par les probabilités d'effectuer chaque marche dans les graphes.

$$K(G,G') = \sum_h \sum_{h'} K_c(h,h') p(h|G) p(h'|G') \tag{3.10}$$

Pour calculer K_c, il existe deux fonctions noyau K_A et K_N définies entre les arcs et les nœuds. Dans le cas où les étiquettes sont des vecteurs de scalaire, si les arcs possèdent e_a étiquettes, K_A s'écrira de la façon suivante pour un noyau gaussien entre deux arcs a et a' :

$$K_A(a,a') = exp\left(-\frac{1}{2}\sum_{i=1}^{e_a}\frac{(a_i - a'_i)^2}{\sigma_i}\right) \tag{3.11}$$

σ_i étant la largeur de bande associée à l'étiquette i, a_i l'étiquette i de l'arc a.

De même, pour les nœuds possédant e_n étiquettes, la fonction K_N s'écrira :

$$K_N(n,n') = exp\left(-\frac{1}{2}\sum_{i=1}^{e_n}\frac{(n_i - n_i')^2}{\sigma_i}\right) \tag{3.12}$$

σ_i étant la largeur de bande associée à l'étiquette i, n_i l'étiquette i du nœud n.

Cependant, si les étiquettes sont plus complexes, les fonctions noyaux K_A et K_N peuvent être formulées différemment.

La fonction K_c peut donc s'écrire comme le produit issu de la comparaison de chaque nœud et chaque arc présents sur le chemin considéré :

$$K_c(h,h') = K_N(h_1,h_1')\prod_{i=2}^{L} K_A\left(h_{2i-2},h_{2i-2}'\right)K_N\left(h_{2i-1},h_{2i-1}'\right) \tag{3.13}$$

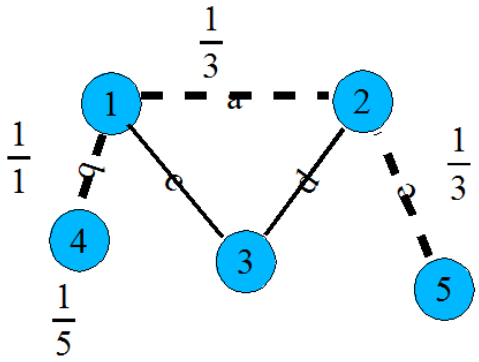


FIG. 3.19 : Exemple de chemin (4,b,1,a,2,e,5), dans un graphe.

Les probabilités $p(h|G$ et $p(h'|G')$ associées aux chemins h et h' permettent de définir l'importance des chemins dans leur graphes respectifs. Différentes probabilités sont mises en jeu. Ici, les probabilités sont uniformes et dépendent de l'ordre du graphe et du degré des nœuds. Il est évidemment possible de définir les probabilités autrement, en tenant compte, par exemple, des étiquettes des arcs et des nœuds.

Un chemin doit débuter par un nœud, la probabilité de démarrer sur un nœud préçis est de $\frac{1}{|G|}$. Par exemple, sur la figure 3.19, pour le chemin de longueur 3 : $(4,b,1,a,2,e,5)$, la probabilité de démarrer au nœud 4 est de $\frac{1}{5}$, puis, comme un seul arc part du nœud 4, la probabilité de traverser l'arc b est de $\frac{1}{1}$. De même, en arrivant au nœud 1, la probabilité de traverser l'arc a est donc de $\frac{1}{3}$, car le degré du nœud 1 est égal à 3.

Il faut noter que les probabilités ne sont pas modifiées au cours du temps. Un parcours peut donc contenir plusieurs fois les mêmes arcs et nœuds en effectuant des retours en arrière dans le graphe. Nous reviendrons par la suite sur les aspects liés à cette redondance.

La probabilité associée à un chemin h de longueur L du graphe G s'écrit donc :

$$p(h|G) = \left(p_s(n_1)\prod_{i=2}^{L} p_t(n_i|n_{i-1})p_q(n_L)\right) \tag{3.14}$$

$p_s(n_1)$, étant la probabilité de démarrer au nœud n_1, $p_q(n_L)$, la probabilité que le chemin s'arrête au nœud n_L et $p_t(n_i|n_{i-1})$, la probabilité d'aboutir au nœud n_i, sachant que le nœud précédent sur le chemin était le nœud n_{i-1}.

En théorie, la formule propose de comparer tous les chemins jusqu'à une longueur infinie. En pratique,la formulation est récursive et peut donc être remplacée par un système linéaire. La taille du système à résoudre

dépend de l'ordre des graphes : $(|G||G'|)^2$. Il faut souligner que la matrice des cœfficients du système linéaire est *sparse*. En effet, la génération des chemins est liée à la connexité des graphes. Cette matrice éparse contient moins de $(d^2|G||G'|)$ éléments non nuls, avec d le degré maximum des graphes. L'implémentation de cette méthode peut donc profiter de cette spécificité pour réduire la complexité globale.

Grâce à cette formulation, Kashima définit un critère de convergence qui permet d'arrêter le calcul lorsque suffisamment d'itérations ont été effectuées. Cette convergence se révèle être relativement intuitive. En effet, les valeurs des probabilités sont inférieures à 1, les valeurs des noyaux K_N et K_A sont normalisées et donc également comprises entre 0 et 1. De ce fait, lorsque la longueur des chemins augmente, la valeur du noyau qui compare ces chemins sera de plus en plus faible.

Revenons sur la notion des probabilités. Comme nous avons pu le constater, les probabilités permettent de pondérer la valeur des noyaux de chaque chemin. Prenons l'exemple du graphe 3.19 et calculons le nombre de chemins possibles en partant du nœud 4, pour quelques longueurs possibles :

Longueur	Nombre de chemins	Chemins
0	1	(4)
1	1	(4,1)
2	3	(4,1,3), (4,1,2), (4,1,4)
3	6	(4,1,4,1), (4,1,2,1), (4,1,2,3), (4,1,2,5),(4,1,3,1), (4,1,3,2)
...	...	...

FIG. 3.20 : Chemins générés à partir du nœud 4 pour différentes longueur dans le graphe de la figure 3.19

Comme nous pouvons le constater, dans un graphe non orienté, il est possible de définir une infinité de chemins. Mais nous devons cependant nous interroger sur la pertinence de certains éléments. Notamment, dans notre exemple, l'arc $(4,1)$ intervient de manière non négligeable dans le résultat final. L'information fournie par un tel calcul est très redondante.

Enfin, la complexité du calcul s'en trouve augmentée, ce qui ne favorise pas l'utilisation du noyau. Comme nous le voyons sur le tableau ci-dessus, le nombre de chemins comparés à chaque étape est fortement croissant.

Des alternatives sont cependant proposées. Dans [60], Mahé et al. propose de redéfinir les probabilités de transition et donc de définir les chemins à la volée. Cette solution permet ainsi d'éviter les retours en arrière, ce qui a pour conséquence non negligeable de réduire le nombre de chemins à comparer et la redondance d'information.

Nous proposons également une alternative, qui consiste à définir au préablable les chemins à comparer.

3.2.2.2 Proposition d'une alternative : les k-chemins

Nous allons donc étudier la formulation générale du noyau de graphe en tant que «sac de chemins».

3.2.2.2.1 Formulation

La méthode des k-chemins revient donc à définir le noyau de graphes par un noyau de sac de chemins [50, 94, 97]. Le noyau entre deux graphes se formulera donc :

$$K(G,G') = \sum_{i=1}^{|h|} \sum_{j=1}^{|h'|} K_c\left(h_i, h'_j\right), \tag{3.15}$$

$|h|$ étant le nombre de chemins du graphe G, $|h'|$ étant le nombre de chemins du graphe G', K_c le noyau défini entre les chemins h et h'.

Il nous est ainsi possible de définir de différentes manières le noyau mineur entre chemins K_c. Ce noyau permet ainsi d'obtenir un score pour comparer deux chemins composés de nœuds et d'arcs en utilisant pour cela des noyaux de nœuds et d'arcs.

$$K_c(h,h') = K_N(h_1,h'_1)\prod_{i=2}^{l} K_A\left(h_{2i-2},h'_{2i-2}\right)K_N\left(h_{2i-1},h'_{2i-1}\right) \tag{3.16}$$

Il est ainsi possible d'effectuer différentes combinaisons entre les noyaux de chemins, pour obtenir le noyau de graphe. Nous proposons différentes formulations.

Tout d'abord, il est possible de sommer tous les noyaux de chemins : La formulation finale (k-chemins) s'écrit donc :

$$K(G,G') = \sum_{L=1}^{LMAX}\sum_{h}\sum_{h'} K_c(h,h') \tag{3.17}$$

Nous pouvons également retenir pour chaque chemin la valeur maximum du noyau (k-cheminMax) [94] :

$$K(G,G') = \frac{1}{2}\sum_{L=1}^{LMAX}\sum_{h}\sum_{h'}\left[\max_{h'} K_c(h,h') + \max_{h} K_c(h,h')\right] \tag{3.18}$$

Cependant, cette formulation ne garantit pas la positivité du noyau (section 2.4) en théorie. En pratique, elle est néanmoins utilisable [28, 65].

3.2.2.2.2 Calcul de chemins

La formulation de Kashima utilise la notion de chemins aléatoires. En pratique les chemins sont générés au fur et à mesure, mais il pourrait être envisageable de les déterminer *a priori*.

Le calcul préalable des chemins peut ainsi permettre de réduire la complexité de la méthode en ne retenant que certains parcours du graphe. Par exemple, nous pouvons ainsi aisément supprimer les parcours qui effectuent des retours en arrière dans le graphe, ou bien limiter la longueur des chemins.

L'alternative consiste donc à définir les chemins en ne retenant que les chemins les plus courts d'un nœud à un autre nœud. La notion de marche aléatoire est donc remplacée par la notion de chemins déterministes.

En comparant les chemins possibles sur l'exemple de la figure 3.19, nous obtenons le tableau suivant :

Longueur	Kashima	K-chemins
0	(4)	(4)
1	(4,1)	(4,1)
2	(4,1,3), (4,1,2), (4,1,4)	(4,1,2), (4,1,3)
3	(4,1,4,1), (4,1,2,1), (4,1,2,3), (4,1,2,5), (4,1,3,1), (4,1,3,2)	(4,1,2,3), (4,1,2,5), (4,1,3,2)
...	...	...

Très rapidement, la différence du nombre de chemins devient importante entre les deux méthodes. De plus, si le graphe possède un cycle, la marche aléatoire générera donc un nombre conséquent de chemins, tandis que la définition des chemins les plus courts ne sera pas génée par la présence de ce cycle.

Comme les chemins sont définis au préalable, il n'est donc pas nécessaire d'utiliser des probabilités de passage qui permettaient ainsi de définir la notion de marche aléatoire.

Nous proposons donc de reformuler un chemin entre deux nœuds comme le plus court chemin entre eux. Pour définir les chemins nous nous utilisons l'algorithme de Dijkstra [24], mais d'autres algorithme pourraient être utilisés. Le but consiste à calculer la distance entre chaque nœud du graphe en interrogeant récursivement les

voisins de chaque nœud, comme le présente l'algorithme 4. Initialement, les distances sont inconnues, nous les fixons donc à une valeur infinie. Pour chaque nœud, nous considérons son voisinage et les informations de distance connues de celui ci. Nous considérons ensuite le voisinage de chaque nœud pour comparer les informations sur l'accessibilité des autres nœuds du graphe. L'information de distance est ainsi mise à jour lorsqu'un nouveau chemin est plus court.

Lorsque nous calculons les distances entre chaque nœud, nous retenons parallèlement les nœuds composant ce chemin. La finalité de cette fonction nous permet d'obtenir un ensemble de chemins reliant chaque nœuds entre eux.

3.2.3 Comparaison

Nous allons maintenant comparer la formulation de kashima et notre alternative des k-chemins.

```
G : graphe ;
MAdj : matrice d'adjacence de G ;
N : ensemble de nœuds de G ;
MDistance : matrice n × n ;
V : n,voisinage de chaque nœud ;
[Initialisation de MDistance]
Pour ∀n ∈ N faire
    MDistance(n) ← ∞ ;
Fin Pour
[Initialisation des chemins]
Pour ∀n ∈ N faire
    Chemin(n) ← n ;
Fin Pour
[Itération pour déterminer le chemin le plus court entre chaque nœud]
Tant que ( |MDistance == ∞| ≠ ∅) faire
    Pour n_D ∈ N faire
        {n_A} ← trouverNoeudsInfini(MDistance(n)) ;
        Pour n_A ∈ {n_A} faire
            Pour n_V ∈ V(n_D) faire
                Si (MDistance(n_D,n_A) > MDistance(n_V,n_A) + Distance(n_V,n_D)) Alors
                    [mise à jour de la distance entre n_D et n_A]
                    MDistance(n_D,n_A) ← MDistance(n_V,n_A) + Distance(n_V,n_D) ;
                    [Mise à jour du chemin entre n_D et n_A]
                    Chemin(n_D,n_A) ← [n_V, Chemin(n_V,n_A)] ;
                Fin Si
            Fin Pour
        Fin Pour
    Fin Pour
Fait
```

Algorithme 4 : Algorithme de calcul du plus court chemin dans un graphe.

3.2.3.1 Exemple jouet

Pour valider cette alternative, nous effectuons un premier test consistant à calculer le noyau des graphes issus de la transformation progressive d'un carre en triangle, les formes intermédiaires étant des trapèzes (figure 3.21).

Le produit scalaire normalisé entre deux graphes peut être assimilé à une distance :

$$\begin{aligned} \|x-x'\|^2 &= \langle x,x\rangle + \langle x',x'\rangle - 2\langle x,x'\rangle \\ &= 2(1-\langle x,x'\rangle) \end{aligned} \tag{3.19}$$

Nous relevons quelques résultats intéressants :
- en comparant le produit obtenu entre le carré avec le trapèze et le carré avec le triangle, le triangle semble plus proche du carré que le trapèze. Ceci est du au fait que le graphe du trapèze compte deux branches supplémentaires par rapport au triangle dans la partie supérieure.
- Nous notons également une certaine symétrie par rapport au trapèze, le triangle est aussi éloigné que le carré par rapport à la forme du trapèze.
- Le produit obtenu est symétrique : $k(G,G') = k(G',G)$.

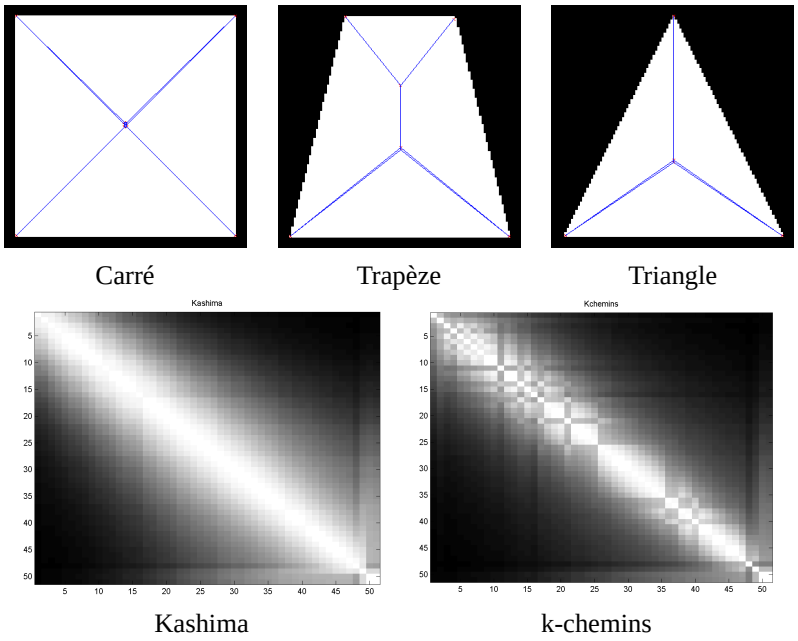


FIG. 3.21 : Partie supérieure : images des carrés, trapèzes, triangle et leur graphe associé. Partie inférieure : matrice de Gram des noyaux de Kashima et k-chemins issus des graphes provenant des formes de la transition carré en triangle.

La figure 3.21 montre ainsi les matrices de Gram obtenues pour chaque noyau : Kashima et k-chemins. Les différents paramètres tels que la largeur de bande pour les noyaux de nœuds et d'arcs sont identiques pour les deux méthodes. Le résultat montre des matrices relativement similaires, ce qui nous conforte dans la proposition d'une alternative au noyau de Kashima.

3.2.3.2 Complexité

Nous allons maintenant comparer les complexité des méthodes de Kashima et k-chemins.

La méthode de Kashima se résume sous la forme d'un système linéaire de taille $n^2 \times n^2$, pour deux graphes de taille n. La résolution d'un système linéaire de taille m a une complexité $\mathcal{O}(m^3)$. La complexité totale est donc de l'ordre de $\mathcal{O}(n^6)$.

Pour le calcul de la méthode k-chemins, nous devons tout d'abord déterminer les chemins. Nous utilisons ainsi la formulation de Dijkstra, dont la complexité est de $\mathcal{O}(n^3)$. Dans notre cas, nous utilisons une implémentation qui utilise des listes d'adjacence, ce qui permet de réduire la complexité à $\mathcal{O}(n^2 log(n))$ pour chaque graphe.

La comparaison des chemins (fonction K_c) a une complexité $\mathcal{O}(n)$, pour un chemin de longueur n au maximum, c'est à dire un chemin contenant tous les nœuds du graphe. En comparant deux graphes d'ordre n, chacun contient donc n^2 chemins. Dans le pire des cas, nous comparons tous les chemins entre eux, soit $n^2 \times n^2$ comparaisons de chemins. Nous obtenons donc une complexité de $\mathcal{O}(n^4 + 2n^2 log(n))$.

Une particularité de la formulation des k-chemins, réside dans la possibilité de réduire la longueur des chemins parcourus. Nous étudierons ainsi l'influence de la longueur des chemins sur les performances obtenues.

3.2.4 Validation du noyau de graphe

Pour valider l'utilisation des noyaux de graphes, nous proposons d'utiliser la base d'images provenant de la compétition Pascal *Visual Object Challenge* 2005 [1]. Cette base contient des images monovision couleur, chacune d'elle possédant une étiquette pour les régions définissant un piéton. Nous disposons ainsi d'un masque binaire établi manuellement. Trois catégories possèdent cette information : personne, voiture et bicyclette.

Comme nous pouvons le voir sur la figure 3.22, les formes et les postures sont relativement variées, les objets présentent parfois des occultations.

La base contient 254 piétons, 214 bicyclettes et 209 voitures.

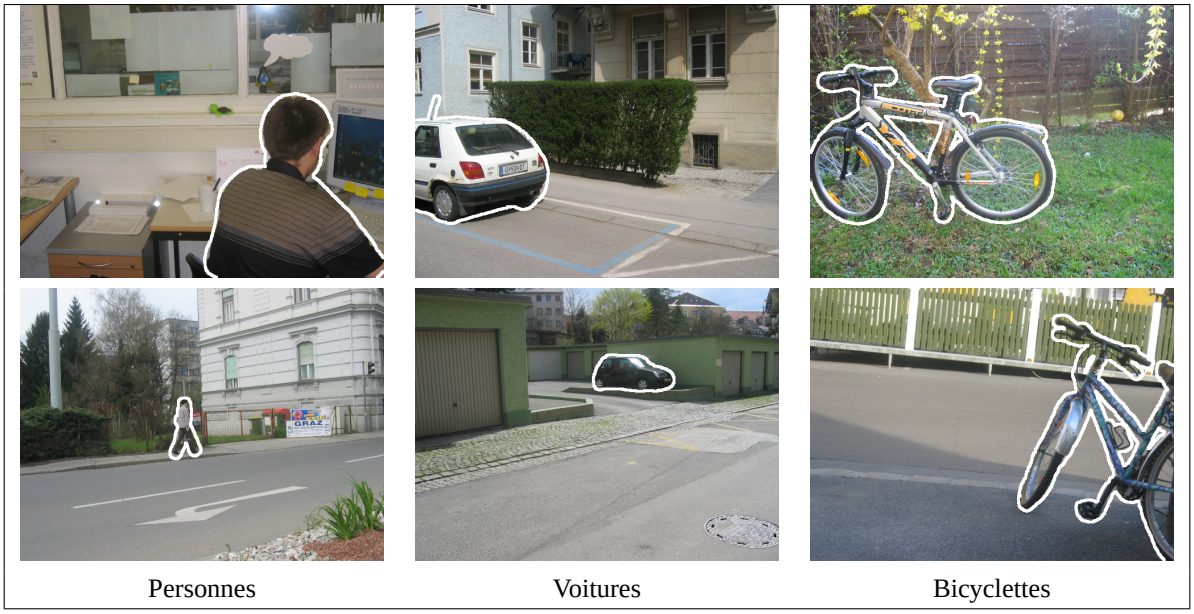


FIG. 3.22 : Exemples d'images de personnes, voitures, bicyclettes et leur masque binaire issues de la base PASCALVOC05.

Nous avons comparé les différents noyaux de graphes présentés précédemment. Nous avons dans un premier temps extrait les graphes de chaque objet de la base, puis nous avons calculé le noyau de graphe. Les graphes de piétons constituent les exemples positifs de la base, les graphes de bicyclettes et voitures représentent les exemples négatifs.

Les résultats présentés sont les courbes ROC (*Receiver Operating Characteristic*) obtenues pour ces tests. Elles

[1] `http://www.pascal-network.org/challenges/VOC/voc2005/index.html`

mettent en rapport le taux de faux positifs avec le taux de vrais positifs, comme le montre l'exemple sur la figure 3.23. Nous pouvons ainsi visualiser rapidement l'efficacité d'une méthode selon la forme de la courbe.

La courbe ROC est établie en comparant les résultats obtenus en fixant un seuil à la fonction de prédiction $f(x) \geq \theta$, $\theta \in \mathbb{R}$. Les données seront considérées comme appartenant à la classe positive si la prédiction est supérieure au seuil, à la classe négative sinon. Lorsque la valeur de θ est élevée, les fausses détections ne sont pas permises. Seules les données dont la prédiction est très forte seront alors considérées comme positives. Inversement, lorsque le seuil θ est faible, les erreurs de classifications deviennent plus fréquentes, davantage de données sont prédites dans la classe positive.

Pour comparer plus facilement les résultats obtenus, nous calculons l'aire sous la courbe ROC, appelée AUC (*Area Under Curve*). Plus l'aire est proche de 1, plus les résultats sont bons.

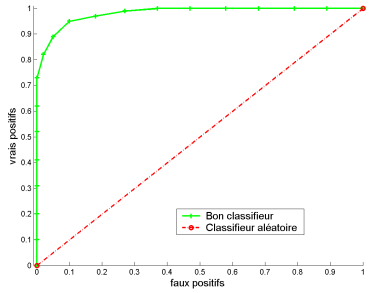


FIG. 3.23 : Exemple de courbes ROC affichant le taux de vrais positifs en fonction du taux de faux positifs. La courbe en traits pleins représente un classifieur avec de bonnes performances. La courbe en pointillés représente un classifieur erroné, qui prédit l'inverse de la classe. La courbe centrale en traits pleins et pointillés est la courbe de référence d'une prédiction aléatoire.

3.2.4.1 Paramètres, squelettisation

Les résultats présentés ont été obtenus par validation *leave-one-out*, c'est à dire que nous enlevons un objet de la base pour chaque test. Tous les autres objets sont donc présents dans la base d'apprentissage, nous apprenons la fonction de classification sur ces exemples et testons sur l'objet retiré au début. Cette séquence est ainsi reproduite pour chaque objet.

Nous avons évalué l'importance de la taille des graphes, la valeur du paramètre de largeur de bande σ pour le calcul des noyaux de nœuds et d'arcs, le nombre d'étiquettes pour les nœuds et les arcs. Les étiquettes des graphes étant normalisées, nous utilisons la même valeur de σ pour toutes les étiquettes.

Nous avons également testé les différentes méthodes de squelettisation : par amincissements et carte de distance. Pour chaque évaluation, nous avons conservé les mêmes paramètres. La valeur du paramètre de pondération des mal classés C, pour le classifieur SVM a été fixé à 10.

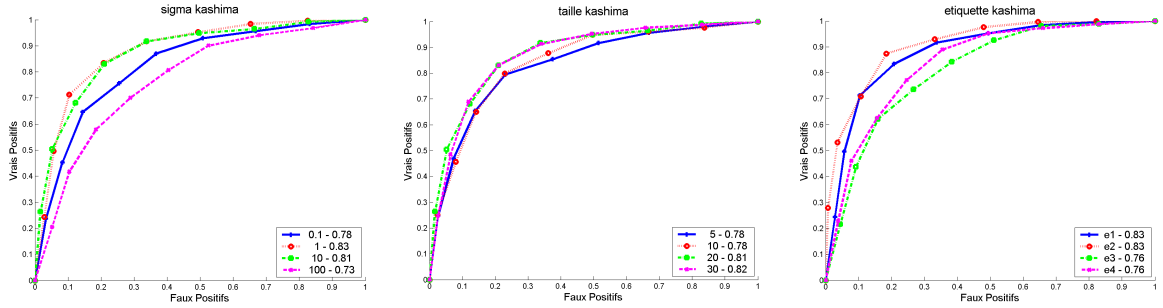
Nous avons évalué différents paramètres utilisés dans la définition du noyau de graphe :
- la largeur de bande σ : 0.1, 1, 10 et 100,
- la taille des graphes : 5, 10, 20 et 30 nœuds,
- le type d'étiquettes des nœuds et arcs :

  e1- Nœuds : Epaisseur ; Arcs : distance, orientation, épaisseur ;

  e2- Nœuds : Epaisseur, moyenne, variance ; Arcs : distance ;

  e3- Nœuds : Epaisseur ; Arcs : distance, épaisseur ;

  e4- Nœuds : Epaisseur ; Arcs : distance, épaisseur, écart.



σ taille graphe étiquettes

FIG. 3.24 : Courbes ROC obtenue avec la fonction de Kashima en fonction des différents paramètres : σ, taille maximale des graphes et types d'étiquettes. Les légendes présentent la valeur du paramètre et la valeur de l'AUC associée à ce paramètre.

		Kashima	k-chemins	k-cheminsMax
σ	0.1	0.829	0.717	0.648
	1	**0.881**	**0.779**	**0.814**
	10	0.878	0.635	0.803
	100	0.775	0.637	0.746
taille	5	0.838	**0.800**	0.816
	10	0.846	0.782	0.815
	20	**0.878**	0.779	0.814
	30	0.877	0.764	**0.822**
étiquettes	e1	0.881	0.782	0.815
	e2	**0.912**	**0.839**	**0.842**
	e3	0.816	0.745	0.778
	e4	0.839	0.763	0.782

FIG. 3.25 : Tableau récapitulatif des AUC obtenues pour les différentes méthodes de noyau de graphe selon différents paramètres mis en jeu.

Sur les figures 3.25 et 3.24, nous présentons les différents résultats obtenus pour les différents paramètres testés. Nous obtenons ainsi les valeurs optimales pour la largeur de bande σ et la taille des graphes. Nous constatons également que certains paramètres se révèlent plus déterminants que d'autres. Par exemple, le choix des étiquettes se révèle plus déterminant que le choix de σ pour la formulation proposée par Kashima, tandis que pour les formulations k-chemins ou k-cheminsMax le choix du paramètre σ est plus déterminant que le choix des étiquettes. Pour toutes les formulations, le réglage de la taille maximales des graphes n'a pas une grande importance. Les formulations k-chemins et Kashima, qui effectuent la somme de toutes les comparaisons des chemins se révèlent légèrement plus sensibles à ce paramètre, car elles tiennent compte de tous les chemins. Si nous supprimons donc du bruit dans le graphe, nous pouvons ainsi améliorer les performances. Dans le cas de la formulation k-cheminsMax, nous retenons les meilleures comparaisons pour chaque chemin. Cette formulation est donc moins sensible à ce paramètre.

Nous évaluons également l'importance des étiquettes des nœuds et des arcs. Les résultats dépendent de la squelettisation. Lorsque le squelette est obtenu par amincissements successifs, les résultats sont meilleurs lorsque les nœuds contiennent une information sur leur voisinage, mais lorsque les arcs ne sont définis que par leur longueur. Inversement, les résultats sont meilleurs avec des informations structurelles pour la squelettisation par carte de distance.

Concernant la taille des graphes, nous notons une amélioration des performances en utilisant la méthode de squelettisation par carte de distance lorsque la taille du graphe est moins importante.

Nous avons ensuite étudié l'influence de la construction des graphes, notamment lors de la construction du squelette. Sur la figure 3.26, nous avons ainsi étudié les différences obtenues par deux méthodes de squelettisation présentées dans la section 3.1.4.1.

		Amincissements successifs			Carte de distance		
		Kashima	k-chemins	k-cheminsMax	Kashima	k-chemins	k-cheminsMax
taille	5	0.838	**0.800**	0.816	0.884	**0.838**	**0.837**
	10	0.846	0.782	0.815	**0.892**	0.826	0.835
	20	**0.878**	0.779	0.814	0.866	0.769	0.793
	30	0.877	0.764	**0.822**	0.853	0.743	0.784

FIG. 3.26 : Tableau récapitulatif des AUC obtenues pour les différentes méthodes de noyau de graphe, selon différentes méthodes de squelettisation.

La modification de la méthode de squelettisation permet d'améliorer les résultats obtenus. En l'occurence, cette méthode de carte de distance, permet de supprimer de nombreuses branches du squelette, afin de ne conserver que la forme générale de l'objet. Cette méthode semble en effet être moins sensible au bruit. Les graphes obtenus nous permettent donc de définir la forme globale de l'objet, sans être perturbés par du bruit qui amène ainsi des barbules supplémentaires au niveau des contours des objets.

3.2.4.2 Longueur de chemins, pouvoir de généralisation

Une spécificité de la méthode de k-chemins réside dans la possibilité de limiter la longueur des chemins. Nous avons évoqué précédemment l'inconvénient de l'approche de Kashima qui considère tous les chemins possibles. La comparaison des chemins est calculée à l'aide de fonctions dont la valeur est inférieure à 1. Ainsi, plus le chemin est long, plus la valeur résultat de la comparaison sera faible. Nous allons donc évaluer l'importance de la longueur des chemins.

Les graphes ont été définis pour une méthode de squelettisation par carte de distance, avec la valeur de σ optimale. Les nœuds sont étiquetés avec l'épaisseur, la moyenne et la variance des niveaux de gris. Les arcs sont étiquetés avec leur longueur, orientation et épaisseur. Nous avons effectué une validation croisée *leave-one-out* sur l'ensemble des données.

	Longueur					
	0	1	2	3	4	5
k-chemins	0.8113	0.8533	0.8376	0.8279	0.8230	0.8219
k-cheminsMax	0.8317	0.8591	0.8562	0.88506	0.8455	0.8418

FIG. 3.27 : Valeurs des AUC obtenues lorsque la longueur des chemins varie pour les méthodes de k-chemin et k-cheminMax.

Nous constatons sur la figure 3.27, que les résultats obtenus en limitant la taille des chemins ne diminuent pas la performance des méthodes. Lorsque la longueur des chemins est égale à 0, les graphes sont décrits par des sacs de nœuds. Lorsque la limitation de la chemin est supérieure ou égale à 1, le calcul du noyau est effectué sur des sacs de chemins. Nous constatons ainsi que l'ajout d'une information structurelle, Ces résultats nous permettent de mettre en avant l'intérêt d'apporter une information sur la structure des nœuds.

Nous avons également évalué l'importance de la taille de la base d'apprentissage. Pour cela, nous avons comparé les résultats obtenus pour 2, 5, 10, 50, 100, 150 et 200 objets par classe dans la base d'apprentissage. Le test consiste à choisir aléatoirement les exemples de la base d'apprentissage, la base de test étant constituée des exemples restant. Nous avons effectué dix itérations pour valider les résultats.

Méthode	L \ Taille Base	2	5	10	50	100	150	200
Kashima	∞	0.5580	0.6372	0.7348	0.8649	0.8805	0.8942	0.9055
k-chemins	1	0.5638	0.6018	0.6737	0.8251	0.8505	0.8593	0.8768
	2	0.5642	**0.6033**	**0.6923**	0.8436	0.8751	**0.8773**	**0.8912**
	3	0.5659	0.6014	0.6903	**0.8449**	0.8746	0.8737	0.8876
	4	**0.5668**	0.6022	0.6889	0.8438	0.8754	0.8738	0.8855
	5	0.5658	0.6025	0.6905	0.8416	**0.8759**	0.8746	0.8852
k-cheminsMax	1	0.5653	0.6221	0.7176	0.8127	0.8155	0.8095	0.8040
	2	0.5686	0.6421	0.7237	**0.8317**	0.8581	0.8658	0.8730
	3	0.5682	0.6458	0.7232	0.8312	0.8680	0.8749	0.8834
	4	0.5710	**0.6500**	0.7259	0.8278	0.8711	**0.8760**	**0.8878**
	5	**0.5734**	0.6485	**0.7323**	0.8243	**0.8714**	0.8757	0.8872

FIG. 3.28 : Résultats obtenus lorsque la taille d'apprentissage varie. Nous évaluons également l'importance de la limitation de la taille des chemins pour les formulations k-chemins et k-cheminsMax. Les chiffres en gras montrent les meilleures performances pour une méthode donnée parmi les longueur de chemin. Les valeurs soulignées montrent les meilleures performances globales.

Nous avons également évalué l'influence de la longueur maximale pour les chemins dans les graphes avec les méthodes k-chemins et k-cheminsMax.

Les résultats obtenus sur la figure 3.28 nous montrent que les résultats obtenus pour les méthodes de k-chemins et k-cheminsMax sont comparables aux résultats de Kashima, mais restent légèrement moins performants. Nous notons également la confirmation des performances comparables pour les k-chemins et k-cheminsMax lorsque nous limitons la longueur des chemins.

3.2.4.3 Temps de calcul

Enfin, nous avons comparé le temps nécessaire pour calculer un noyau complet avec les différentes méthodes. La figure 3.29 affiche ainsi le temps nécessaire pour calculer un noyau de taille 676×676 selon les trois implémentations Kashima, k-chemins et k-cheminsMax. Le temps de référence est le temps pour le calcul des graphes de taille maximale à 5, pour la méthode de Kashima. Nous constatons que le temps nécessaire pour les méthodes des k-chemin et k-cheminsMax sont similaires, mais reste nettement inférieur par rapport à l'utilisation de la méthode de Kashima.

Le gain en temps de calcul, entre la méthode k-chemins pour une longueur maximale de chemin égale à 3, est quasiment de 4 en faveur de celle-ci par rapport à la méthode de Kashima, pour des graphes d'une taille maximale de 20 nœuds.

3.3 Application à la stéréovision

Nous allons maintenant présenter une application des noyaux de graphes.

Dans notre cas, la reconnaissance de formes s'appuie sur la description d'images. Comme nous l'avons évoqué précédemment, nous devons connaître le masque binaire associé aux objets présents dans l'image.

L'application envisagée initialement concerne un système embarqué équipé d'un système de stéréovision. L'utilisation de ce système d'acquisition nous permet de récupérer les images des objets présents, ainsi que de calculer

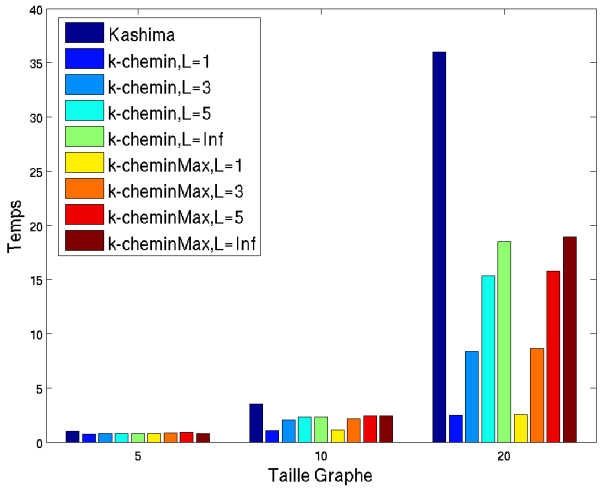


FIG. 3.29 : Illustration du temps de calcul nécessaire pour un noyau selon les différentes méthodes de Kashima, k-chemin et k-cheminMax lorsque la taille des graphes évolue.

l'information de distance des objets. Cette information nous permet ainsi de définir en pratique les masques binaires des objets.

Tout d'abord, nous présenterons la méthode de stéréovision et l'extraction d'une carte 3D. Ensuite, nous présenterons quelques résultats obtenus sur des images réelles.

3.3.1 Principe de la stéréovision

La stéréovision, utilise un système de caméras stéréoscopiques [3]. Le système d'acquisition dispose ainsi d'une paire de caméras ayant le même champ visuel.

La particularité de ce système consiste à écarter suffisamment les caméras afin de pouvoir récupérer l'information de distance pour chaque pixel des images obtenues. Le principe est fondé sur la perception humaine et utilise la différence de position respective d'un pixel dans chaque image. Cette différence permet ainsi de calculer l'information relative à la distance de l'objet duquel est extrait le pixel.

Nous utilisons le modèle sténopé (figure 3.30) pour schématiser les caméras dans le monde réel. Sur la figure 3.32 nous constatons ainsi que la différence de position d'un pixel appartenant à un objet proche est plus grande que pour un pixel appartenant à un objet éloigné dans la scène.

Pour calculer la disparité, il faut effectuer la mise en correspondance des pixels. Cela consiste à apparier les pixels d'une image avec les pixels de l'autre image [52, 55, 87]. Le calcul de la disparité permet ainsi de retrouver les coordonnées originales du point dans l'espace réel. Nous devons également souligner le fait qu'en disposant de coordonnées en deux dimensions uniquement, il est possible de récupérer des coordonnées en trois dimensions.

Dans notre cas, nous supposerons que les deux caméras sont parfaitement alignées verticalement, les centres optiques se trouvent alors sur la même ligne épipolaire. Ainsi, l'image d'un point dans le plan gauche, trouvera son correspondant stéréo sur la même ligne dans l'image droite.

Pour obtenir les coordonnées réelles d'un point présent dans les deux images nous appliquons les formules suivantes :

$$X = \frac{x \cdot e}{p \cdot \delta} \quad Y = \frac{y \cdot e}{p \cdot \delta} \quad Z = \frac{f \cdot e}{p\delta} \tag{3.20}$$

avec, (x,y) les coordonnées du point image dans le plan image, p la taille d'un pixel, δ la disparité horizontale entre les deux points stéréo, f la focale des objectifs.

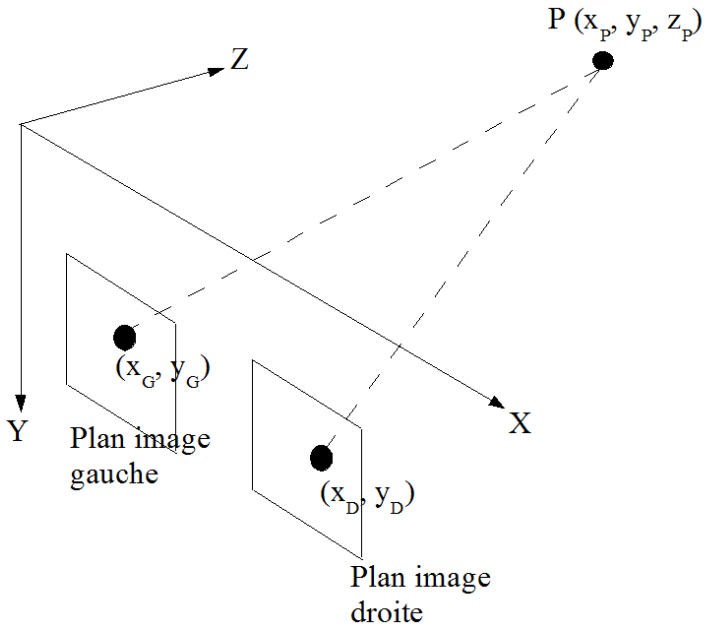


FIG. 3.30 : Configuration d'un système stéréoscopique selon le modèle sténopé.

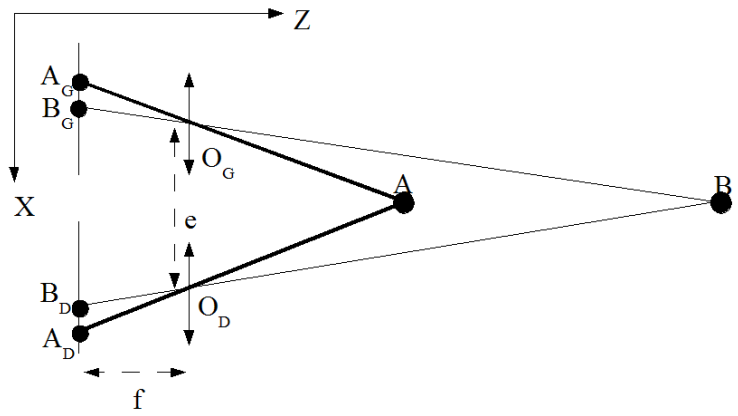


FIG. 3.31 : Schéma optique d'affichage de deux points proche et éloigné.

Lors de cet appariement, nous calculons ainsi la différence de position des pixels. Pour cela, nous calculons la position respective du pixel considéré dans chaque image.

3.3.2 De la stéréovision aux graphes

Nous disposons d'un système d'acquisition stéréoscopique, calibré de façon à avoir le même niveau pour chaque image verticalement. Actuellement, il est difficile d'obtenir des cartes de disparité de bonne qualité pour être exploitables, en utilisant simplement des méthodes de mise en correspondance. En effet, dans les zones homogènes, la mise en correspondance peut être défectueuse, et les valeurs de disparité obtenues pour des pixels voisins peuvent être complètement différentes. Par la suite, l'extraction des objets contenus dans la scène est rendue plus difficile.

Nous effectuons ainsi une segmentation en deux étapes :

1. une segmentation région,

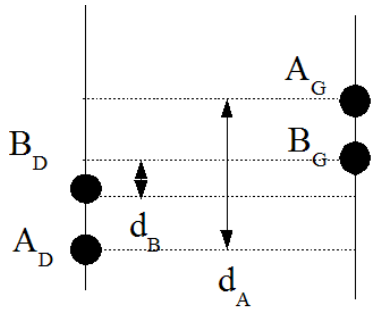


FIG. 3.32 : Exemple de différence de disparité entre deux points proche ou éloigné.

2. le calcul des disparités pour chaque région.

La segmentation région effectuée nous permet de récupérer les zones homogènes contenues dans l'image. Cependant, dans la plupart des cas, une région ne couvre pas entièrement un objet. Celui-ci peut en effet être défini par un ensemble de régions relativement différentes en terme de niveau de gris, texture. Nous utilisons une segmentation région assez simple, mais suffisante pour notre application. Nous définissons une région comme un ensemble de pixels voisins dont la différence de niveau de gris avec le niveau de la région est inférieure à un seuil. L'idée de cette application réside principalement dans la validation de notre méthode de graphe, nous n'avons donc pas utilisé de méthode trop complexe pour la segmentation. C'est une des lacunes du système actuel, qui nécessite donc des améliorations.

L'idée originale consistait à utiliser ce système avec une seule caméra, puis de regrouper les régions voisines pour former des objets entiers. Cependant, cette méthode présente de nombreux inconvénients. Le nombre de régions constituant un objet est inconnu *a priori*, nous ne pouvons donc pas fixer de critère d'arrêt sur le nombre de régions à regrouper.

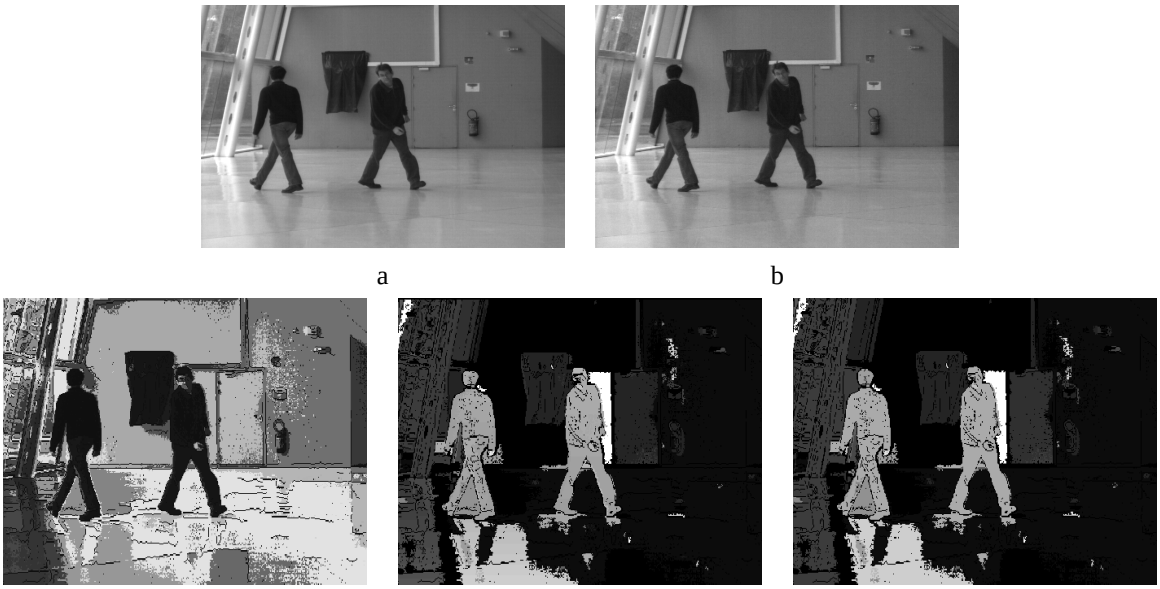
Nous avons donc choisi de définir un critère de regroupement en utilisant l'information de distance. Nous calculons ainsi pour chaque région la valeur de sa disparité, c'est à dire sa distance par rapport à la caméra. Nous pouvons ensuite regrouper les régions voisins, dont la disparité est très proche. L'image est donc découpée en un ensemble de régions, chacune définissant le masque binaire d'un objet. Nous pouvons ainsi transformer chaque objet en graphe comme nous l'avons présenté dans la section précédente 3.1.4 et effectuer la reconnaissance de chaque graphe (figure 3.34).

Nous désirons comparer les formes des objets présents. En effet, les apparences des piétons sont relativement variées et l'apport d'information de type texture ne permettrait pas de résoudre la variabilité liée à l'apparence du piéton. Nous utilisons donc des étiquettes relatives à la topologie des squelettes, telles que la taille du squelette, la longueur et l'orientation des arcs ou bien les coordonnées des nœuds.

3.3.3 Résultats

Nous évaluons maintenant les performances de la méthodes sur des images réelles de stéréovision. Nous avons effectué des acquisitions d'images stéréoscopiques, en intérieur, avec 3 piétons différents.

Nous avons ainsi extrait 110 graphes de piétons et 180 de non-piétons. Nous utilisons tous ces graphes pour l'évaluation de la méthode.



a b

c d e

FIG. 3.33 : Images gauche (a) et droite (b). Résultat de la segmentation région (c). Résultat de la disparité (d) . Résultat de la fusion entre la segmentation région et la disparité (e).

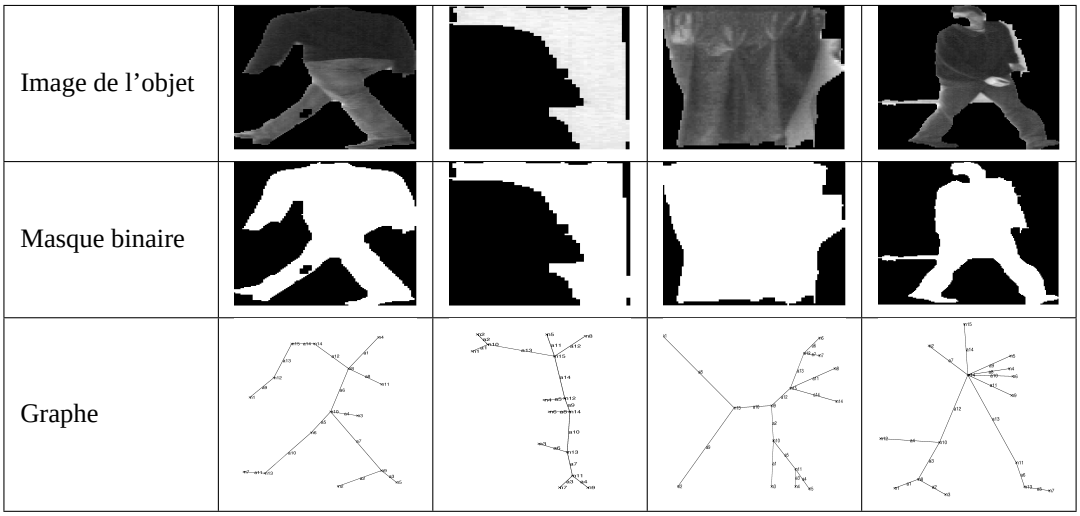


FIG. 3.34 : Exemples d'images extraites à l'aide d'un système de stéréovision, leur masque binaire associé et les graphes extraits.

3.3.3.1 Comparaison des méthodes

Nous avons évalué les performances obtenues par la méthode de Kashima et des k-chemins, selon les deux méthodes de squelettisation (voir section3.1.4.1). Pour la méthode des k-chemins, les résultats sont présentés sans limiter la longueur des chemins.

Les nœuds et les arcs sont étiquetés. Pour les nœuds, nous retenons la taille du masque binaire à l'emplacement du nœud. Pour les arcs, nous calculons la taille moyenne du masque binaire le long de la branche du squelette, la longueur et l'orientation de l'arc.

Le premier résultat (figure 3.35) a été obtenu par validation croisée *leave-one-out*, le poids des mal classés C fixé à 10. La méthode de Kashima est la plus efficace dans cette application avec une squelettisation par carte de

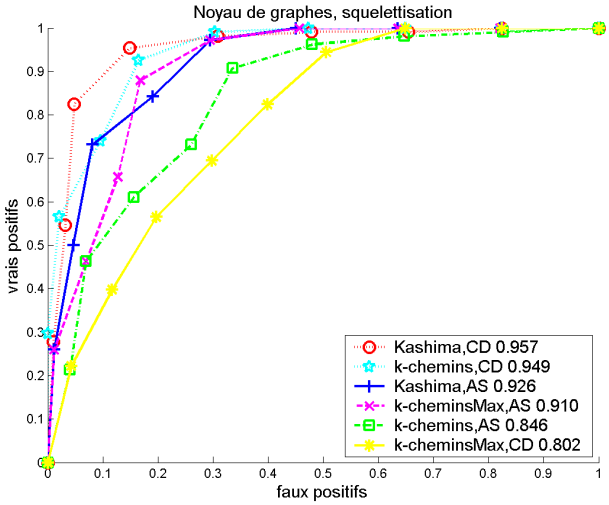


FIG. 3.35 : Résultat de la validation *leave-one-out*, des graphes de piétons et non-piétons, obtenus sur une séquence intérieure. Les courbes montrent les résultats obtenus pour la méthode de Kashima et des k-chemins, selon les deux méthodes de squelettisation (AS : amincissements successifs, CD : carte de distance). Nous notons également les valeurs des AUC obtenues.

distance. Cependant, la méthode des k-chemins donne de bons résultats également et nous notons une très forte amélioration des résultats en changeant de méthode de squelettisation. Ceci peut s'expliquer par le fait que les régions issues des cartes de disparité sont très bruitées au niveau des contours. La méthode de squelettisation à partir du calcul d'une carte de distance est moins sensible au bruit que la méthode utilisant les amincissments successifs.

3.3.3.2 Paramètrage du classifieur SVM

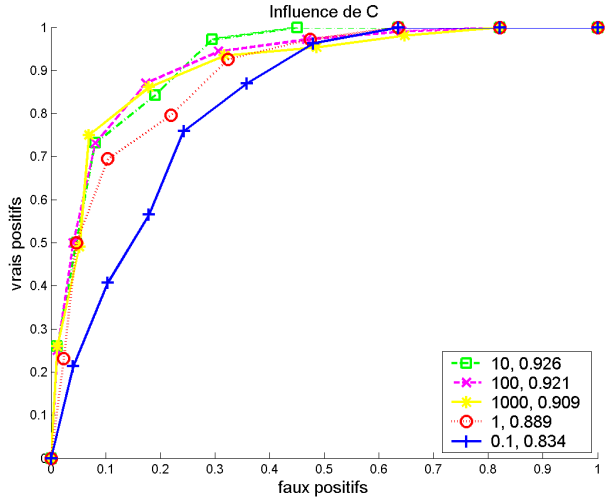


FIG. 3.36 : Courbes ROC et AUC obtenues selon la valeur du poids des mal classés pour le classifieur SVM.

Le résultat suivant (figure 3.36) permet de montrer l'influence du paramètre C du classifieur SVM qui pondère les données mal classées en apprentissage. Les meilleurs résultats sont obtenues pour des valeurs de 10 et 100.

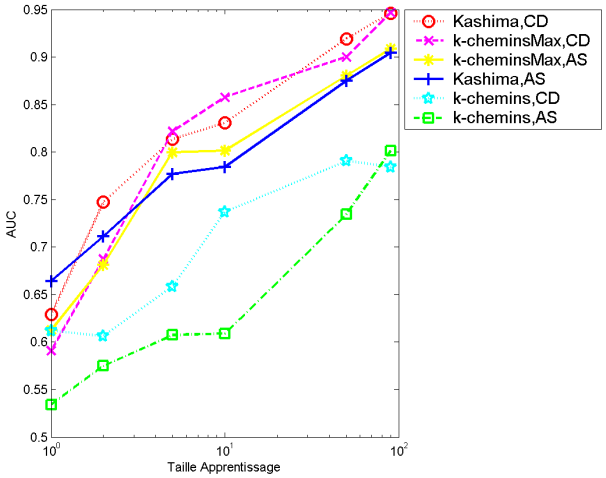


FIG. 3.37 : Variation des performances en fonction de la taille d'apprentissage pour les méthodes Kashima, k-chemins et k-cheminMax, selon les deux méthodes de squelettisation par amincissements successifs (AS) et carte de distance (CD).

3.3.3.3 Généralisation

La dernière figure (3.37) illustre l'importance de la taille d'apprentissage. Nous faisons varier la taille de la base : 1,2, 5, 10, 50 et 90 graphes de piétons et autant de non-piétons en apprentissage. Nous effectuons une validation croisée en choisissant aléatoirement les graphes présents en apprentissage et en test. Chaque test est itéré 10 fois, les bases étant les mêmes pour toutes les méthodes présentées. Les meilleurs résultats sont obtenus pour une grande base d'apprentissage, mais certaines méthodes obtiennent des résultats corrects pour seulement 5 graphes en apprentissage : la méthode des k-cheminsMax avec une squelettisation par amincissements et la méthode de Kashima avec une squelettisation par carte de distance. Cette dernière présente également de bons résultats pour seulement 2 graphes de chaque classe en apprentissage. Ce résultat nous permet de montrer la bonne capacité de généralisation de la méthode de graphe

Nous allons illustrer le pouvoir de généralisation. Voici quelques exemples de graphes bien ou mal reconnus lorsque la base d'apprentissage contient deux piétons et deux non-piétons (figure 3.38) pour la méthode de kashima.

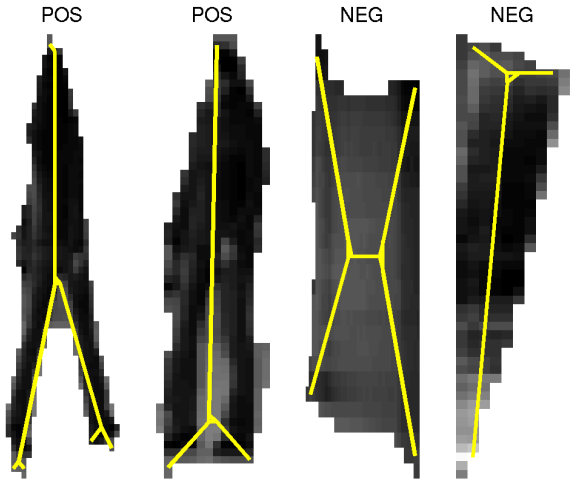


FIG. 3.38 : Images des piétons et non-piétons dans la base d'apprentissage.

La figure 3.39 présente ensuite quelques images exemples de piétons reconnus ou non, ainsi que quelques

exemples de non piétons correctement et mal classés. Sur la totalité de ce test nous obtenons une aire sous la courbe ROC de 0.70, avec 64% de bonne détection.

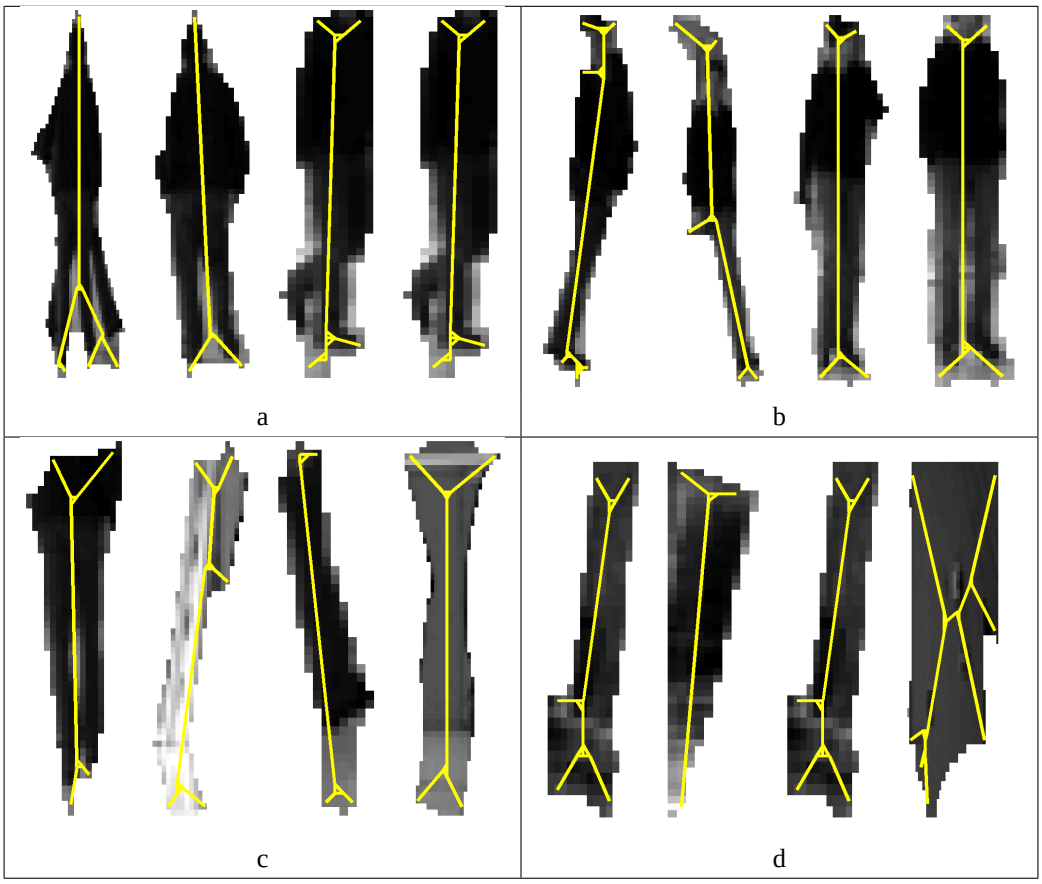


FIG. 3.39 : Exemple de vrais positifs (a), faux négatifs (b), faux positifs (c) et vrais négatifs (d).

3.4 Conclusion

Dans cette partie, nous avons présenté une méthode de représentation d'objet par graphes. Les graphes présentent en effet des propriétés intéressantes pour la représentation d'objets. Il s'agit en effet d'une représentation structurée et compacte, offrant de nombreuses possibilités d'apport d'information.

Afin de pouvoir utiliser des algorithmes de classification de type machines à noyau, il est nécessaire de définir un produit scalaire entre graphes.

La première méthode que nous utilisons est le noyau de graphe défini par Kashima [47]. Cette méthode présente une formulation intéressante du noyau de graphe, constitué de sacs de chemins. Les chemins sont en effet issus des graphes et permettent donc de représenter partiellement la structure de graphe.

Nous proposons dans un deuxième temps une formulation plus générale des «sacs de chemins». L'objectif est de pouvoir construire un noyau entre graphes qui combine un ensemble de noyaux mineurs comparant les chemins entre eux.

Nous avons ainsi remplacé la notion de marche aléatoire dans les graphes que propose Kashima par une notion de chemin direct afin de réduire la complexité de la formulation.

Nous avons pu évaluer la pertinence de l'approche de graphes sur une base d'images dont nous connaissons

le masque binaire. Ce masque est en effet nécessaire pour la construction des graphes. Nous pouvons ainsi nous détacher de la partie segmentation d'images afin de valider l'approche de noyau de graphe.

Les résultats obtenus restent favorables à la méthode de Kashima, mais la formulation k-chemins présente des performances comparables. Cependant, la particularité de l'approche par sac de chemins nous permet de limiter la taille des parcours de graphes. Nous constatons ainsi que des parcours de taille limitée permettent d'obtenir des performances comparables à la formulation sans limite, mais sont beaucoup plus avantageux en terme de temps de calcul.

Nous avons enfin appliqué ce noyau de graphe sur des images issues de séquences stéréovision [81, 82]. La stéréovision nous permet en effet de segmenter l'image en région en fonction de la position réelle des objets dans l'espace. Cette segmentation nous permet donc de déterminer en pratique le masque binaire des objets. Nous extrayons de la même manière les graphes à partir de ces régions Nous avons obtenu des résultats encourageants lors de l'évaluation de cette méthode.

Cependant, nous restons confrontés à quelques limites de cette méthode. Comme nous l'avons vu, les performances du noyau de graphes sont très liées avec la méthode de squelettisation. Si les squelettes sont très bruités, les graphes ne seront donc pas optimaux.

De plus, la complexité des méthodes reste très importante, malgré les modifications de la formulation et la limitation de la longueur des chemins.

Nous pouvons ainsi envisager des perspectives pour améliorer cette méthode, en testant de nouvelles fonctions de squelettisation, ou bien en construisant les graphes sur d'autres informations que les masques binaires. Il est par exemple envisageable de définir des graphes de régions, ou bien de points d'intérêt.

Nous avons également constaté que les performances sont liées au type d'information contenu dans les étiquettes. Une amélioration de cette méthode consisterait à déterminer automatiquement les étiquettes et les valeurs des paramètres mis en jeu.

Enfin, nous étiquetons le graphe uniquement avec des vecteurs de scalaires. La formulation du noyau de graphe nous permet d'envisager des étiquettes plus complexes, voire des étiquettes structurées. Nous pourrions ainsi rajouter des informations permettant la description du voisinage des nœuds ou des arcs.

4

Histogrammes d'orientation de gradient

« Réaliser la représentation de l'irreprésentable, voir l'invisible, toucher et percevoir l'impalpable. »

Novalis

Sommaire

Dans cette dernière partie, nous allons présenter une méthode de reconnaissance d'images de piétons. Dans la section précédente, l'utilisation des graphes était soumise à une segmentation de l'image. Un objet est défini par une région, de laquelle nous pouvons ensuite extraire des caractéristiques. Ici, nous envisageons une approche non-segmentée ou globale, comme le propose Papageorgiou et Shahua [68, 78].

La méthode que nous utilisons a été initiée par Dalal et al. [21]. Le but consiste à caractériser une image contenant un unique objet par des histogrammes locaux d'orientation de gradient.

Nous présenterons dans un premier temps l'obtention d'un vecteur de caractéristiques et quelques résultats. Puis dans un deuxième temps nous proposerons des améliorations en utilisant pour cela des fonctions noyaux dédiées aux histogrammes. Enfin nous montrerons une application de cette méthode pour la détection de piétons dans le cas de l'utilisation d'images infrarouges.

4.1 Présentation

Nous avons vu dans le premier chapitre de ce mémoire que la description des données est une étape majeure du processus de reconnaissance de formes. Nous nous plaçons dans le cas où les données sont des images.

La recherche d'une caractérisation pertinente fait ainsi l'objet d'études variées afin de comparer et déterminer les possibilités pour caractériser une image. Différentes méthodes existent actuellement pour caractériser des images de mainère globale. Le but consiste à définir une représentation pertinente qui puisse discriminer la classe de l'image.

Parmi ces représentations, certaines méthodes s'appuient sur la description par histogrammes couleurs. Ainsi, une image complète est caractérisée par un unique histogramme. Ce type de descripteur est utilisé couramment en indexation d'images [74, 80] car il permet de définir rapidement la catégorie d'une image. Cette description globale de l'image reste cependant peu adaptée dans le cas de la reconnaissance de piétons. Comme nous l'avons constaté dans la première partie de ce mémoire, la détection de piétons est confrontée à la problématique de la variabilité d'apparence du piéton. La description d'une image par un seul histogramme définissant la luminance ou la couleur de l'image n'est donc pas suffisamment discriminant pour détecter un piéton.

L'idée de caractériser l'image d'un piéton en utilisant la forme se révèle donc plus pertinente. Dans cette perspective, les travaux de Papageorgiou et al. [67], permettent de décrire la forme de l'objet représenté par une image à l'aide d'une décomposition en ondelettes. Les cœfficients obtenus permettent notamment de décrire la présence de contours dans l'image. Ce type de description obtient de très bons résultats, car il permet de caractériser la forme générale de l'objet, sans être lié à l'apparence des objets.

En 2004, Shashua et al. [78] ont présenté une méthode performante pour la détection de piétons. Ici encore, la description de l'image d'un objet permet de décrire la forme. Cependant, elle utilise des histogrammes d'orientation de gradient. Le gradient permet en effet de déterminer les variations de luminance dans l'image et donc les contours. Les histogrammes ne sont pas calculés sur l'ensemble de l'image, mais correspondent à des régions locales. L'image est ainsi découpée en différentes régions situées dans des zones spécifiques du piéton : tête, jambes, bras. Chaque région génère un histogramme, l'ensemble des histogrammes formant ainsi les données caractéristiques de l'image. Le découpage de l'image en régions reste cependant spécifique à la détection de piétons. Il est donc difficile d'étendre cette méthode à la reconnaissance d'autres catégories d'objets. Les résultats sont également liés à la position du piéton dans l'image et n'autorisent pas de variations importantes dans sa posture.

Lowe [59] utilise ainsi la description à l'aide d'orientation du gradient autour de points d'intérêt. La recherche des points d'intérêt est effectuée en multi-résolution, ce qui permet de définir une taille variable pour le voisinage des points d'intérêt. La taille dépend de l'échelle à laquelle le point d'intérêt est détecté. L'image est ainsi caractérisée par un ensemble d'histogrammes locaux. En considérant ces données comme des «sacs de mots», nous perdons alors la composante spatiale des données, ce qui réduit la pertinence de cette méthode.

La méthode introduite par Dalal et al. [21, 22], est fondée sur une description de l'image à l'aide d'histogrammes d'orientation de gradient. Ces histogrammes sont calculés sur l'ensemble de l'image, mais sont définis sur une région locale. La description d'une image est un vecteur contenant tous les histogrammes calculés. Ce vecteur apporte, d'une part, de l'information locale pour chaque région et permet, d'autre part, de conserver un agencement spatial, les vecteurs étant calculés systématiquement par le même procédé.

Nous allons décrire les différentes étapes permettant de calculer ce vecteur pour une image donnée

4.1.1 Description de la méthode

La méthode d'histogrammes d'orientation de gradient permet de définir un vecteur caractéristique d'une image. Ce vecteur est composé d'histogrammes calculés sur un nombre défini de régions selon les étapes suivantes :

1. calcul du gradient,
2. découpage de l'image en cellules,
3. calcul des histogrammes pour chaque cellule,
4. normalisation des histogrammes au sein de blocs de cellules.

Une étape est spécifique à la méthode présentée ici : la normalisation des histogrammes. Le calcul d'une norme est classique, mais son application apporte une certaine originalité. La normalisation est en effet effectuée indépendamment pour chaque cellule par rapport à son voisinage. Nous reviendrons plus tard sur cette spécificité.

4.1.1.1 Calcul du gradient

Comme le nom de la méthode le laisse entendre, la caractérisation utilise l'orientation du gradient. Pour chaque pixel de l'image, il est donc nécessaire de déterminer l'angle du gradient. Nous utilisons donc un filtrage de l'image selon les deux dimensions :

- horizontal : $(-1\ ,\ 0\ ,\ 1)$
- vertical : $(-1\ ,\ 0\ ,\ 1)^T$

Nous obtenons ainsi deux matrices G_H et G_V correspondant au gradient horiztontal et vertical (figure 4.1).

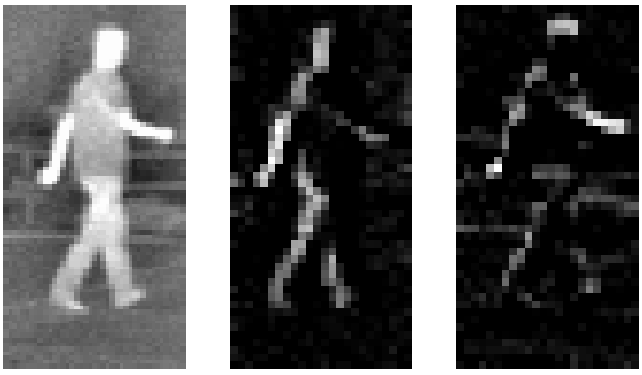


Image originale Gradient horizontal Gradient vertical

FIG. 4.1 : Image d'un piéton en infrarouge (gauche), image du gradient horizontal (milieu) et vertical (droite).

L'orientation du gradient (figure 4.2) est définie pour chaque pixel par :

$$O_G(x,y) = atan\left(\frac{G_H(x,y)}{G_V(x,y)}\right) \tag{4.1}$$

Nous calculons également la norme du gradient en chaque point (figure 4.2) :

$$N_G(x,y) = \sqrt{G_H(x,y)^2 + G_V(x,y)^2} \tag{4.2}$$

Cette norme sera utilisée lors du vote dans les histogrammes.

4.1.1.2 Découpage de l'image

L'image est ensuite découpée en cellules (figure 4.3). Le découpage est exhaustif et toutes les régions de l'image sont ainsi couvertes. La taille des cellules est fixée au préalable selon les besoins et les performances obtenues.

Pour des raisons pratiques, la taille d'une région est définie en pixels. Ainsi, pour obtenir des descripteurs de même taille pour toutes les images, celles-ci doivent être impérativement de même taille. Avant d'être traitée,

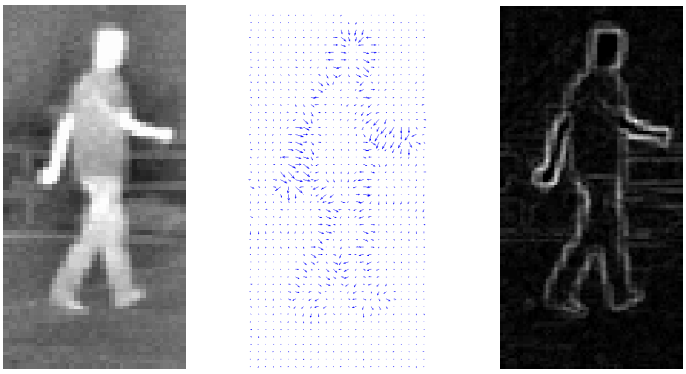


Image originale orientation du gradient Norme du gradient

FIG. 4.2 : Exemple d'image de piéton en image infrarouge (gauche), l'orientation du gradient associée à cette image (milieu) et la norme du gradient (droite).

chaque image doit donc être redimensionnée selon une taille définie. Dans notre cas, nous traitons des images de taille 128×64 pixels. Il serait cependant possible de définir la taille des cellules en fonction de la taille des images à traiter, sans avoir besoin de redimensionner ces dernières, la finalité étant d'obtenir le même nombre de cellules quelque soit la taille de l'image.

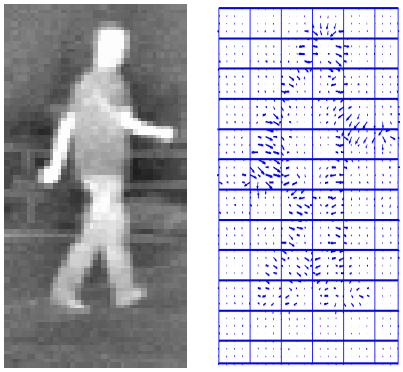


FIG. 4.3 : Découpage d'une image infrarouge (gauche) en 12×6 cellules (droite).

4.1.1.3 Calcul des histogrammes

Lorsque le découpage de l'image est effectué, nous calculons ainsi un histogramme d'orientation de gradient pour chaque cellule (figure 4.4). Chaque pixel des cellules participe au vote. Celui-ci peut être pondéré par la magnitude du gradient à l'emplacement du pixel. Pour chaque pixel de coordonnées (x,y), la valeur associée dans l'histogramme H sera :

$$H(a) = H(a) + \begin{cases} N_G(x,y) & \text{si vote pondéré} \\ 1 & \text{sinon} \end{cases} \tag{4.3}$$

avec a tel que $O_G(x,y) \in [\theta_a, \theta_{a+1}[$.

Cette pondération permet ainsi d'accorder davantage d'importance au vote d'un pixel appartenant à un contour, qui générera donc une magnitude importante, par rapport au vote d'un pixel appartenant à une région homogène. Le vote permet ainsi de tenir compte de la forme de l'objet contenu dans l'image.

Le nombre de niveaux de l'histogramme est également définissable. En fonction de la finesse de l'histogramme, nous pourrons ainsi considérer avec plus ou moins de précision l'orientation des gradients. Pour un gradient non

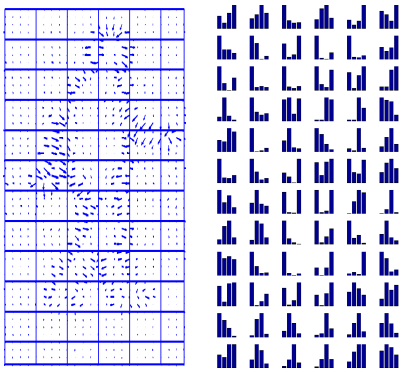


FIG. 4.4 : Calcul des histogrammes définis pour 4 niveaux d'orientation (droite) pour chaque cellule (gauche).

signé, nous avons ainsi $\theta_a = \frac{a\pi}{n}$, avec n le nombre de niveaux de l'histogramme.

De plus, nous pouvons choisir de prendre en compte le signe du gradient ou non. Cette particularité permet ainsi de caractériser indifféremment la forme de l'objet par rapport à sa couleur ou sa luminosité. La forme de l'objet sera prépondérante vis-à-vis de son apparence.

4.1.1.4 Normalisation des histogrammes

La dernière étape concerne la normalisation des histogrammes. Cette normalisation est justifiée par la réduction de la variabilité dans le cas de changements d'illumination. L'idée consiste ainsi à normaliser chaque cellule au sein d'un bloc de cellules voisines. Le nombre de cellules par bloc est un paramètre supplémentaire à fixer. Nous reviendrons par la suite sur le réglage de tous ces paramètres permettant de gérer efficacement cette méthode.

Le facteur de normalisation n_f est donc calculé en tenant compte de tous les histogrammes des cellules comprises dans le bloc. Plusieurs normes sont applicables :

- aucune normalisation : $n_f = 1$
- norme L1 : $n_f = \frac{H}{\|H\|_1 + \varepsilon}$
- norme L2 : $n_f = \sqrt{\frac{H}{\|H\|_2^2 + \varepsilon}}$

H est un vecteur contenant tous les histogrammes du bloc, ε un terme de régularisation, les valeurs contenues dans l'histogramme pouvant être nulles. Tous les histogrammes contenus dans le bloc sont ainsi normalisés et ajoutés au descripteur final.

Les blocs peuvent se recouvrir partiellement, ainsi les cellules sont normalisées en tenant compte de différents voisinages. Les histogrammes associés à ces cellules appartenant à différents blocs sont ainsi normalisés selon différents facteurs de normalisation, et ajoutés plusieurs fois dans le descripteur final. Le descripteur peut donc contenir de l'information redondante.

4.1.2 Classifieur

Une fois le descripteur obtenu, il est alors possible d'utiliser des méthodes de classification comme nous avons vu dans la section 2.1. Ici, nous utilisons le classifieur SVM linéaire (section 2.3). Pour comparer les données entre elles, nous utilisons donc un simple produit scalaire. Ce noyau a l'avantage d'être non paramétrique, ce qui nous permet d'évaluer simplement l'efficacité de la méthode. Nous pourrons ainsi envisager des améliorations par la suite, en implémentant une fonction noyau plus élaborée améliorant les performances. L'utilisation d'un noyau linéaire nous permettra également de comparer cette méthode avec d'autres types de descripteurs d'images.

La frontière de décision est donc

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}^T \mathbf{x}_i + w_0\right) \tag{4.4}$$

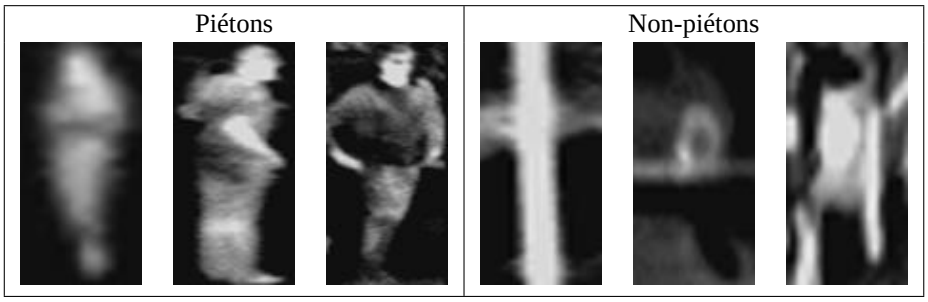
4.1.3 Paramètres

Comme nous l'avons vu dans la section 4.1.1, décrire une image à l'aide d'histogrammes fait appel à de nombreux paramètres pour gérer le découpage de l'image en cellules, le regroupement des cellules en blocs, le type de normalisation des histogrammes et les caractéristiques des histogrammes. Voici la liste des paramètres à régler :

- Cellules
  - Taille des cellules : nombre de pixels définissant une cellule.
- Blocs
  - taille : nombre de cellules par bloc,
  - chevauchement : nombre de cellules entre chaque bloc,
  - facteur de normalisation.
- Histogrammes
  - vote pondéré ou non,
  - gradient signé ou non,
  - nombre de niveaux.

Pour évaluer l'importance de chaque paramètre et régler ceux-ci de manière optimale, nous avons donc accompli un test exhaustif.

Nous disposons pour cela d'une base d'images infrarouges. Nous avons extrait manuellement environ 6000 images de piétons et non-piétons, soit plus de 3000 piétons. Comme nous l'avons précisé précédemment, les images doivent avoir les mêmes dimensions, nous avons donc redimensionné chaque image à la même taille : 128×64 pixels. Les images de non-piétons sont assez variées. Nous avons extrait des objets pouvant prêter à confusion tels que les arbres, les poteaux, mais également des objets anodins : voitures, route, mur.



Pour tester chaque paramètre, nous procédons de la façon suivante : un ensemble d'images est réservé pour l'apprentissage, nous extrayons les descripteurs HOG de chaque image en fonction des paramètres demandés, nous apprenons le classifieur SVM linéaire avec ces descripteurs et nous appliquons le classifieur appris sur les descripteurs de la base de test, calculés avec les mêmes paramètres. La base d'images contenant beaucoup de redondance, nous avons choisi de réduire sa taille à 2200 images de piétons et autant de non-piétons.

Nous retenons ainsi les AUC des résultats obtenus pour chaque ensemble de paramètres et nous comparons ensuite ces valeurs afin de conserver l'ensemble correspondant à la meilleure AUC. Pour confirmer les résultats, chaque ensemble de paramètres est évalué sur des bases d'apprentissage et de test que nous renouvelons. Nous effectuons ainsi dix itérations pour chaque test. Afin que les résultats soient comparables, nous conservons les mêmes conditions d'évaluation, c'est-à-dire que nous utilisons les mêmes images en apprentissage et en test pour

chaque itération, quelque soit l'ensemble de paramètres testé.

Voici les valeurs testées pour chaque paramètre :
- Cellules
  - Taille des cellules : 8×8, 16×16 et 32×32 pixels.
- Blocs
  - taille : 2×2, 3×3 et 4×4 cellules,
  - chevauchement : 1, 2 cellules,
  - facteur de normalisation : $L1$, $L2$, unitaire.
- Histogrammes
  - vote pondéré par la norme ou non,
  - nombre de niveaux : 4 et 8.

Nous affichons les courbes ROC (section 3.2.4) obtenues pour chaque type de paramètre, selon les variations de leur valeur sur la figure 4.5. Tous les résultats ont été obtenus avec l'ensemble de paramètres suivant :
- Cellules
  - taille des cellules : 8×8 pixels.
- Blocs
  - taille : 2×2 cellules,
  - chevauchement : 1 cellule,
  - facteur de normalisation : L2.
- Histogrammes
  - vote pondéré,
  - nombre de niveaux : 4.

Comme nous pouvons le constater sur la figure 4.5, la modification de certains paramètres se révèle plus déterminante au niveau de l'amélioration des performances globales de la méthode par rapport à d'autres paramètres. Ainsi, la taille des cellules, le facteur de normalisation et la pondération des votes se révèlent prépondérant par rapport à la taille des blocs, le nombre de niveaux dans l'histogramme et le chevauchement des blocs.

Comme nous l'avons souligné précédemment, la pondération du vote par la magnitude du gradient permet de tenir compte de la forme présente dans l'image. La normalisation des histogrammes est également importante, puisqu'elle permet de réduire les variations présentes dans l'image. Enfin la taille des cellules correspond à la taille des régions descriptives de l'image. Si les cellules sont trop grandes, elles couvrent une grande partie de l'image et décrivent donc l'image globalement. Inversement, des cellules de taille réduite apportent une information locale. Si la taille est alors trop réduite les cellules décrivent alors les pixels eux-mêmes. Dans ce cas, la description est très locale et devient sensible à la position de l'objet dans l'image. Le paramétrage optimal est donc le compromis entre ces deux cas de figure et doit alors apporter une information locale permettant de décrire la forme de l'objet, d'une taille suffisamment grande pour être insensible aux variations de position de l'objet.

Nous obtenons donc l'ensemble de paramètres suivant :
- Cellules
  - taille des cellules : 8×8 pixels.
- Blocs
  - taille : 2×2 cellules,
  - chevauchement : 1 cellule,
  - facteur de normalisation : L2.
- Histogrammes
  - vote pondéré,
  - nombre de niveaux : 8.

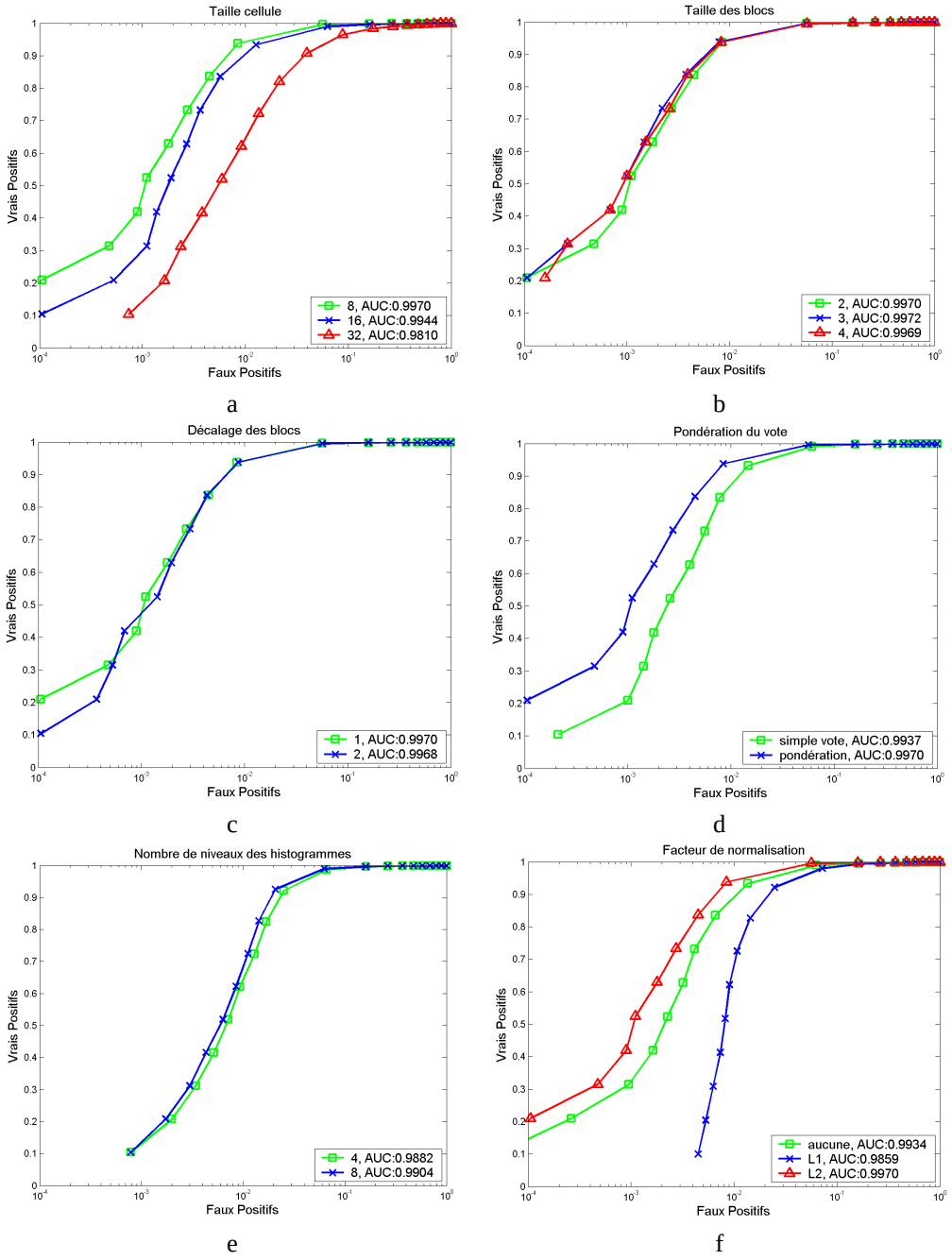


FIG. 4.5 : Courbes ROC obtenues lors du test pour le paramétrage de la méthode HOG. Les différents paramètres ont été testés : taille des cellules (a), taille des blocs (b), décalage des blocs (c), pondération du vote dans l'histogramme (d), nombre de niveaux des histogrammes (e) et type de normalisation des histogrammes dans les blocs (f).

4.1.3.1 Dimension des données

La dimension du descripteur obtenu dépend directement des paramètres définis pour la méthode. Elle dépendra ainsi de la taille des cellules, selon que l'on souhaite une description globale ou locale de l'image. Elle dépend également de la taille des blocs et du recouvrement entre les blocs, c'est-à-dire l'importance que l'on souhaite accorder à la normalisation des histogrammes. Enfin, elle est fonction du nombre de niveaux dans les histogrammes et donc de la précision que nous souhaitons pour comptabiliser les directions du gradient.

Pour des cellules de 32 pixels, des blocs de taille 1, sans recouvrement entre blocs et 4 niveaux par histogramme, la dimension du descripteur est de 32. Lorsque le découpage est plus fin, par exemple des cellules de 2×2 pixels, des blocs de largeur 2 avec une cellule de recouvrement et 16 niveaux par histogramme, la dimension obtenue est de l'ordre de 30000.

Pour notre ensemble de paramètres optimaux, la dimension est de 3360. En comparant avec le nombre de pixels, l'image est composée de 8192 pixels.

Selon les paramètres, nous pouvons donc utiliser une représentation très compacte des données et permettre ainsi de réduire le temps de calcul lorsque nous devons déterminer le produit scalaire.

4.1.4 Résultats et comparaison

Comme nous l'avons évoqué dans le premier chapitre 1.2.1, la détection du piéton est confrontée à différentes problématiques : la taille, la posture et l'apparence. Dans cette section, nous allons donc étudier la capacité de généralisation de cette méthode. De plus, nous allons évaluer la sensibilité de la méthode à la position et la taille du piéton dans l'image. Enfin, nous évaluerons l'efficacité de la méthode face au problème d'occultation.

Les résultats ont été obtenus sur la base d'images de piétons extraites manuellement d'une séquence infrarouge. Toutes les images ont été redimensionnées à la même taille de 128×64 pixels. Nous avons utilisé les paramètres optimaux définis ci-dessus dans la section 4.1.3. Chaque courbe ROC présentée est le résultat d'une validation sur dix itérations où nous choisissons aléatoirement les bases d'apprentissage et de test. La base de test contient ainsi 2200 piétons et autant de non-piétons.

4.1.4.1 Taille d'apprentissage

Nous avons évalué les performances de cette méthode lorsque la taille de la base d'apprentissage varie.

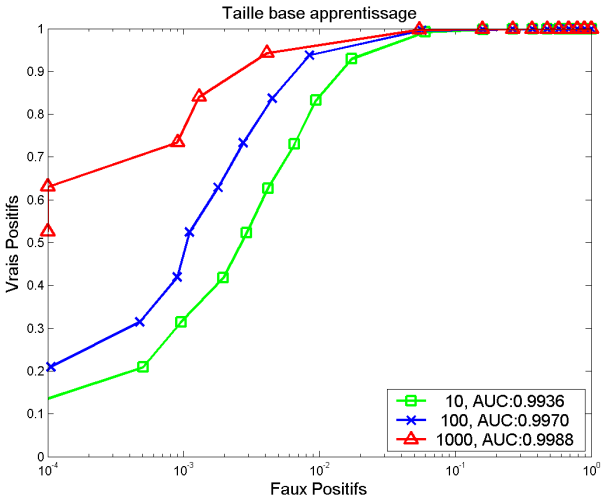


FIG. 4.6 : Courbes ROC obtenues pour la méthode HOG avec les paramètres optimaux, lorsque la taille de la base d'apprentissage varie.

Nous avons ainsi évalué les résultats obtenus avec 10, 100 et 1000 piétons dans la base d'apprentissage, avec autant de non-piétons. Les meilleurs résultats sont évidemment obtenus pour la plus grande base, mais nous pouvons souligner les bons résultats obtenus pour seulement 10 piétons en apprentissage. Ce résultat nous permet de mettre en avant la bonne capacité de généralisation de cette méthode.

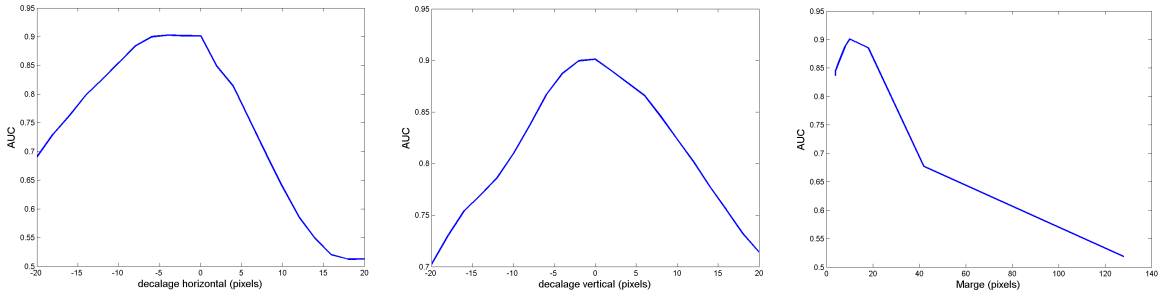
Nous expliquons cette bonne capacité de généralisation par la construction du descripteur. Le découpage de l'image en cellules permet ainsi de réduire la sensibilité à la variation de posture ou de position dans l'image.

Cette méthode présente également un taux intéressant pour les faux positifs. En considérant la courbe pour 1000 piétons en apprentissage, nous avons ainsi 1 faux positif pour 330 images, avec un taux de reconnaissance de 90 %.

4.1.4.2 Influence de la position du piéton

Nous devons dans un premier temps quantifier l'importance du positionnement d'un piéton. Ce test nous permet ainsi de justifier l'utilisation d'une méthode d'extraction automatique. Nous avons ainsi utilisé les étiquettes manuelles définies sur une séquence d'images infrarouges et extrait des images de piétons en décalant le piéton dans l'image finale. Le décalage est effectué horizontalement et verticalement. Cependant, nous ne modifions pas les images de la base d'apprentissage, lesquelles contiennent des piétons centrés manuellement.

Les résultats obtenus sur la figure 4.7 confirment ainsi l'importance du positionnement. Les résultats sont meilleurs lorsque le piéton est bien centré dans l'image, puisque les images de la base d'apprentissage contiennent des piétons centrés. Cependant, les résultats restent corrects malgré le décalage.



a b c

FIG. 4.7 : Mesure de l'influence du positionnement du piéton dans l'image, horizontalement (a), verticalement (b). Influence de la taille du piéton (c) dans l'image.

Nous avons également testé l'influence de la taille du piéton dans l'image. Lorsque les images du piéton sont extraites pour constituer les bases d'images, nous définissons une marge autour du piéton. En modifiant la taille de cette marge, nous modifions ainsi la taille du piéton. La base d'apprentissage contient des images avec une marge de 10 pixels, le meilleur résultat est donc obtenu pour la valeur d'une marge de 10 pixels.

Pour résoudre ce problème, nous proposons un test simple consistant à modifier les exemples de la base d'apprentissage en apportant le même type de modifications subies par les exemples de test (figure 4.8). Le but consiste à renforcer la base d'apprentissage afin d'améliorer les performances en classification [57].

Les transformations apportées aux images correspondent à des transformations naturelles rencontrées dans un système de détection de piétons. En effet, l'extraction manuelle peut être différente d'une extraction automatique. Les fenêtres extraites automatiquement d'une image peuvent contenir des piétons dont la position n'est pas parfaitement centrée.

Comme nous le constatons sur la figure 4.9, les résultats obtenus sont améliorés en utilisant une base dont les exemples fournis en apprentissage subissent des transformations aléatoires. La base d'apprentissage contient 500 piétons, 500 non-piétons et nous évaluons le classifieur sur 1000 images.

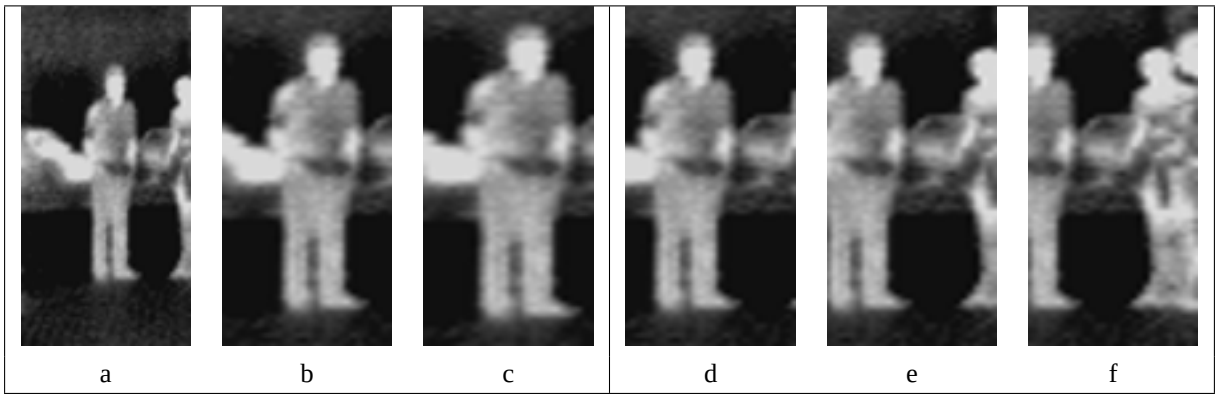


FIG. 4.8 : Exemples de transformations subies par un piéton. Nous évaluons tout d'abord l'importance de la taille du piéton dans l'image : 64 (a), 100 (b) et 120 (c) pixels en hauteur. Les autres images présentent le résultat d'un décalage horizontal du piéton dans l'image : 5 (d), 10 (e) et 15 (f) pixels vers la gauche.

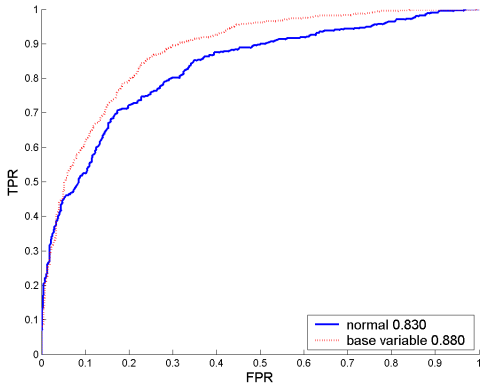


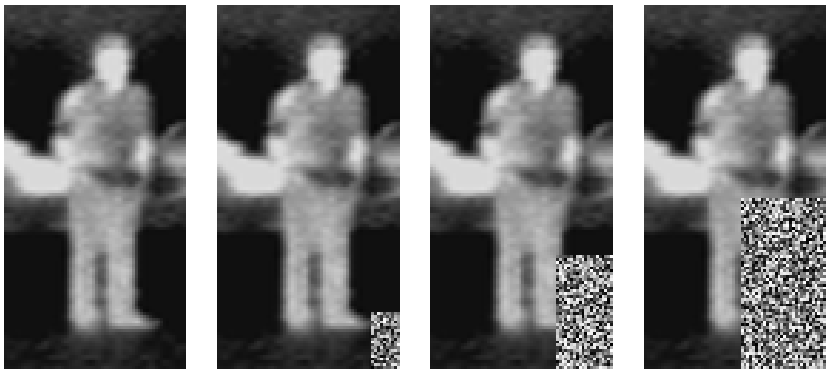
FIG. 4.9 : Résultats obtenus sur une base de test ayant subi des transformations (modification de la position et de la taille du piéton). Nous comparons une base d'apprentissage constituée d'images de piétons extraites manuellement (trait plein) et une base d'apprentissage constituée d'images ayant subi le même genre de transformations que la base de test (trait pointillé).

4.1.4.3 Problème d'occultation

Nous avons évalué l'influence des problèmes d'occultation pour la méthode. Pour pouvoir mesurer l'influence de l'occultation, il est nécessaire de pouvoir quantifier également leur importance. Nous avons donc mis en place un test simple consistant à masquer progressivement les images de la base de test. Nous utilisons donc en apprentissage des images complètes et testons la performance du classifieur sur des images occultées.

Pour simuler une occultation, nous perturbons une zone de l'image selon une taille paramétrable (figure 4.10). La zone occultée est remplacée par du bruit. Ceci ne représente pas une occultation réelle, au sens où un piéton serait masqué par un autre piéton ou un véhicule, mais elle nous permet de connaître les capacités de la méthode.

En effet, nous définissons de façon arbitraire la zone occultée afin de mesurer son importance. Mais les cas réels d'occultation ne sont pas aussi critiques. Si le piéton est masqué par un autre piéton, la détection du premier piéton se révèle plus importante, puisque celui-ci est plus proche du véhicule, donc plus en danger qu'un piéton plus lointain. Le problème d'occultation se pose réellement lorsque le piéton est masqué par un véhicule et s'apprête, par exemple, à traverser la rue. Le test que nous proposons est relativement objectif, car la zone d'occultation peut revêtir de nombreux aspects. Il faut alors pouvoir évaluer la surface minimum nécessaire pour que le piéton soit détecté par le système.



a b c d

FIG. 4.10 : Exemple de piéton subissant une occultation de 0% (a), 2.5% (b), 10% (c) et 20% (d).

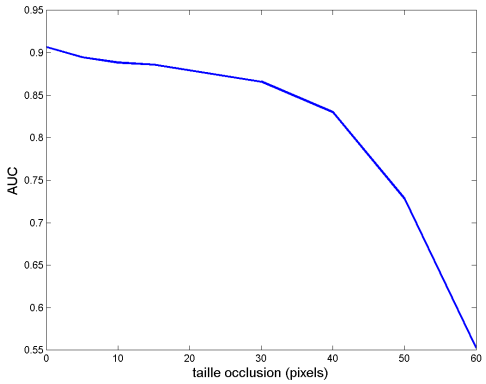


FIG. 4.11 : Variations de l'aire sous la courbe ROC en fonction de la largeur de l'occultation.

Nous constatons sur la figure 4.11 que les performances restent sensiblement les mêmes pour une taille d'occultation jusqu'à 20×40 pixels, soit 10% de l'image. Les performances diminuent légèrement jusqu'à une région 30×60 pixels, quasiment un quart de l'image.

La méthode présente donc des résultats intéressants pour le problème d'occultation. Ici encore, l'explication peut être donnée par la construction du descripteur. Le découpage en cellules de petite taille permet en effet de réduire l'impact de la zone occultée qui ne concerne qu'une partie de l'image, les cellules restantes permettant de détecter le piéton.

Dans la pratique, ce problème peut être abordé en constituant une base d'apprentissage permettant de résoudre ce problème. Il faudrait ainsi intégrer des images contenant des piétons masqués ou des groupes de piétons.

4.1.4.4 Comparaison

Nous allons maintenant comparer cette méthode avec d'autres descripteurs présentés dans la première partie de ce manuscrit.

Dans [98], Xu et al. utilise les données brutes pour caractériser les images. Le premier descripteur contient donc uniquement les valeurs des pixels (figure 4.12).

Le deuxième descripteur est proposé selon la méthode de Papageorgiou et al. [68, 71]. Il est déterminé à partir des cœfficients obtenus par une décomposition en ondelettes. La figure 4.12 montre ainsi les cœfficients obtenus pour une image infrarouge en utilisant des ondelettes de Haar selon les directions horizontale, verticale et

diagonale.

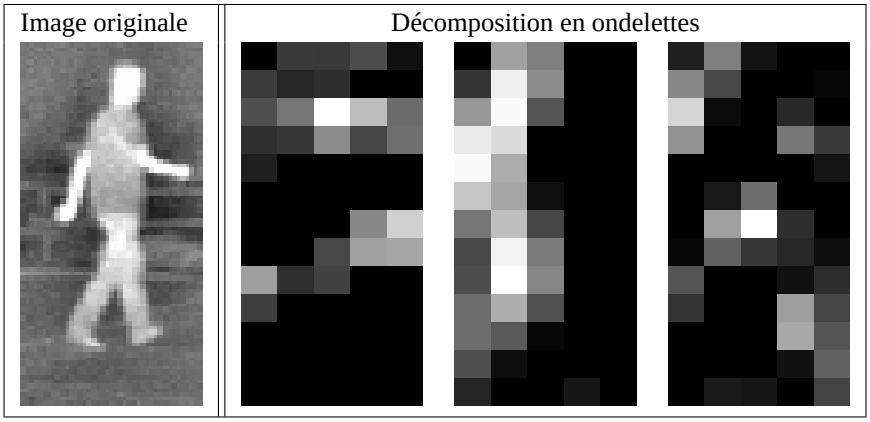


FIG. 4.12 : Image originale d'un piéton en infrarouge (gauche), décomposition en ondelettes pour un filtrage horizontal, vertical, diagonal (images centrales).

Le troisième descripteur utilise la valeur de la norme du gradient en chaque pixel de l'image (figure 4.13). Nous allons pouvoir comparer le pouvoir de représentation entre la valeur brute des pixels et le calcul du gradient.

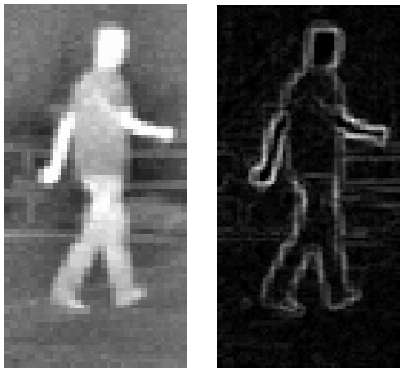


FIG. 4.13 : Image d'un piéton en image infrarouge (gauche) et la norme du gradient associé (droite).

Le quatrième descripteur est inspiré de la méthode de Shashua et al. [78]. La figure 4.14 montre ainsi le découpage utilisé, inspiré par le papier de Shashua et al. Nous définissons 5 régions basiques, auxquelles nous ajoutons la combinaison des régions 2-3 et 4-5. Chaque région définie de cette manière est découpée en 4 sous-régions, pour lesquelles nous calculons un histogramme d'orientation de gradient. Nous obtenons ainsi 28 histogrammes d'orientation de gradient pour composer le descripteur final. Cependant, nous n'utilisons pas de régression *Ridge* pour analyser les données, mais le même classifieur que pour tous les descripteurs. Nous cherchons en effet simplement à comparer l'efficacité de la description des images.

Le dernier descripteur est constitué selon la méthode HOG décrite ci-dessus dans la section 4.1.1. Nous avons utilisé les paramètres optimaux de la méthode.

Le test s'appuie sur la même base lors de l'apprentissage et du test pour toutes les méthodes comparées. Nous avons ainsi évalué les performances pour 100 et 1000 images en apprentissage et évalué les méthodes sur une base de test de 1000 images. Afin de valider les résultats obtenus, nous avons effectué dix itérations en renouvelant aléatoirement les bases d'apprentissage et de test.

La figure 4.15 présente les résultats obtenus. Comme nous pouvons le constater, les descripteurs utilisant l'information de gradient présentent les meilleurs résultats. En particulier, l'utilisation d'histogrammes améliore fortement les performances La décomposition de l'image en ondelettes n'améliore pas significativement les résultats

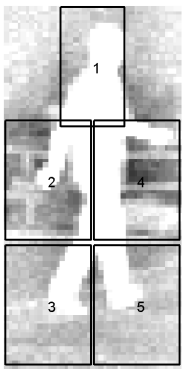
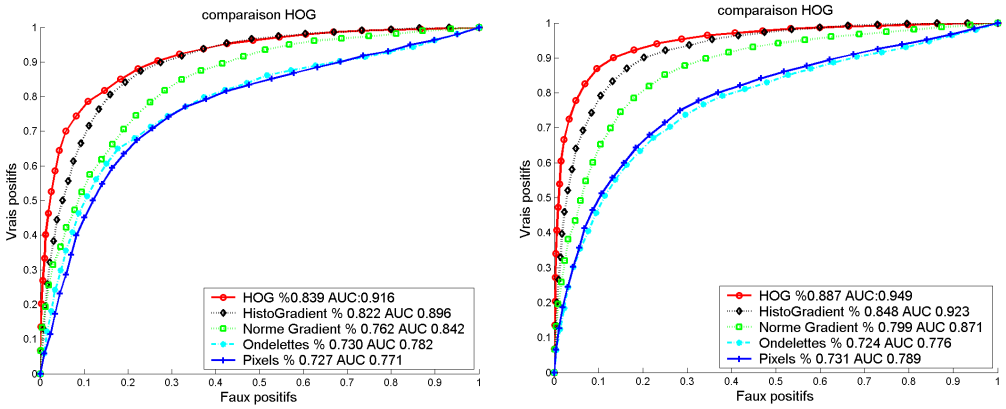


FIG. 4.14 : Découpage en régions selon la méthode de Shashua.



a b

FIG. 4.15 : Comparaison de la méthode HOG avec d'autres descripteurs : valeurs des pixels, filtrage par ondelettes, norme du gradient, histogrammes locaux de gradient. La figure (a) montre les résultats obtenus pour 100 images en apprentissage, 1000 sur la figure (b).

par rapport à la description issue de la valeur brute des pixels. L'explication peut tenir en partie sur le type d'images employées dans ce test. En effet, les images infrarouges contiennent des régions particulières suffisamment pertinentes pour décrire un piéton.

Concernant le pouvoir de généralisation des méthodes, l'augmentation de la taille de la base d'apprentissage améliore les performances de toutes les méthodes, en particulier pour les méthodes constituée par l'information de gradient, mais ne modifie pas le classement.

4.2 Noyau pour le HOG

Nous avons donc présenté dans la section précédente 4.1 la méthode d'orientation d'histogrammes de gradient. Cette méthode permet de caractériser l'image d'un objet à l'aide d'un vecteur d'histogrammes. Pour valider cette méthode nous avons utilisé un classifieur SVM linéaire, dont le noyau est un simple produit scalaire. Ce noyau présente l'intérêt d'être non paramétrique et donc de pouvoir déterminer précisément l'importance du descripteur, sans que le paramétrage du classifieur soit contraignant.

4.2.1 Présentation des noyaux

Nous allons maintenant étudier les possibilités d'amélioration en travaillant sur la méthode de classification. Pour cela, nous allons analyser différentes fonctions noyaux plus adaptées au type de descripteur, en l'occurence des histogrammes.

Nous allons ainsi présenter et comparer les noyaux :
- linéaire,
- intersection d'histogramme,
- RBF,
- rationnel.

Le noyau linéaire est obtenu par le produit scalaire entre les vecteurs x et $x' \in \mathbb{R}^n$:

$$k_{lin}(x,x') = \langle x,x' \rangle = \sum_{i=1}^{n} x_i.x'_i \tag{4.5}$$

Nous avons utilisé ce noyau précédemment pour présenter la méthode HOG. Sa simplicité et son utilisation rapide présentent un grand intérêt pour obtenir des performances en classification correctes.

Le noyau d'intersection d'histogramme, appelé *Generalized Intersection Histogram*, est présenté par Boughorbel et al. [10] dans le cas de description d'images à l'aide d'histogrammes.

$$k_{GHI}(x,x') = \sum_{i=1}^{n} min\left\{ |x_i|^\beta, |x'_i|^\beta \right\}, \tag{4.6}$$

β étant un paramètre, tel que $\beta \geq 0$.

Nous utilisons ensuite le noyau RBF (*Radial Basis Function*) :

$$k_{RBF}(x,x') = exp^{\left(-\frac{d(x,x')}{\sigma}\right)}, \tag{4.7}$$

σ étant le paramètre correspondant à la largeur de bande, $d(x,x')$ une fonction de distance entre les vecteur x et x'.

Nous définissons enfin le noyau rationnel [17, 34] :

$$k_{rationnel}(x,x') = 1 - \frac{d(x,x')}{d(x,x') + \sigma}, \tag{4.8}$$

σ un paramètre à fixer, $d(x,x')$ une fonction de distance entre x et x'.

Pour les noyaux RBF et rationnel, nous avons ainsi besoin de définir les fonctions de distance entre vecteurs. Nous utilisons la norme L_1 :

$$d_{L_1}(x,x') = \left|x - x'\right| = \sum_{i=1}^{n} \left|x_i - x'_i\right|, \tag{4.9}$$

la norme L_2 :

$$d_{L_2}(x,x') = \left\|x - x'\right\|^2 = \sum_{i=1}^{n} \left(x_i - x'_i\right)^2, \tag{4.10}$$

et une approximation de la fonction de χ^2 [14, 76]. La formulation initiale de la fonction de χ^2 est définie pour comparer une distribution pratique xp avec une distribution théorique xt connue : $d_{\chi^2 T} = \sum_{i=1}^{n} \frac{(xt_i - xp_i)^2}{xt_i}$. La formulation modifiée [76] permet de comparer deux histogrammes réels :

$$d_{\chi^2(x,x')} = \sum_{i=1}^{n} \frac{(x_i - x'_i)^2}{x_i + x'_i} \tag{4.11}$$

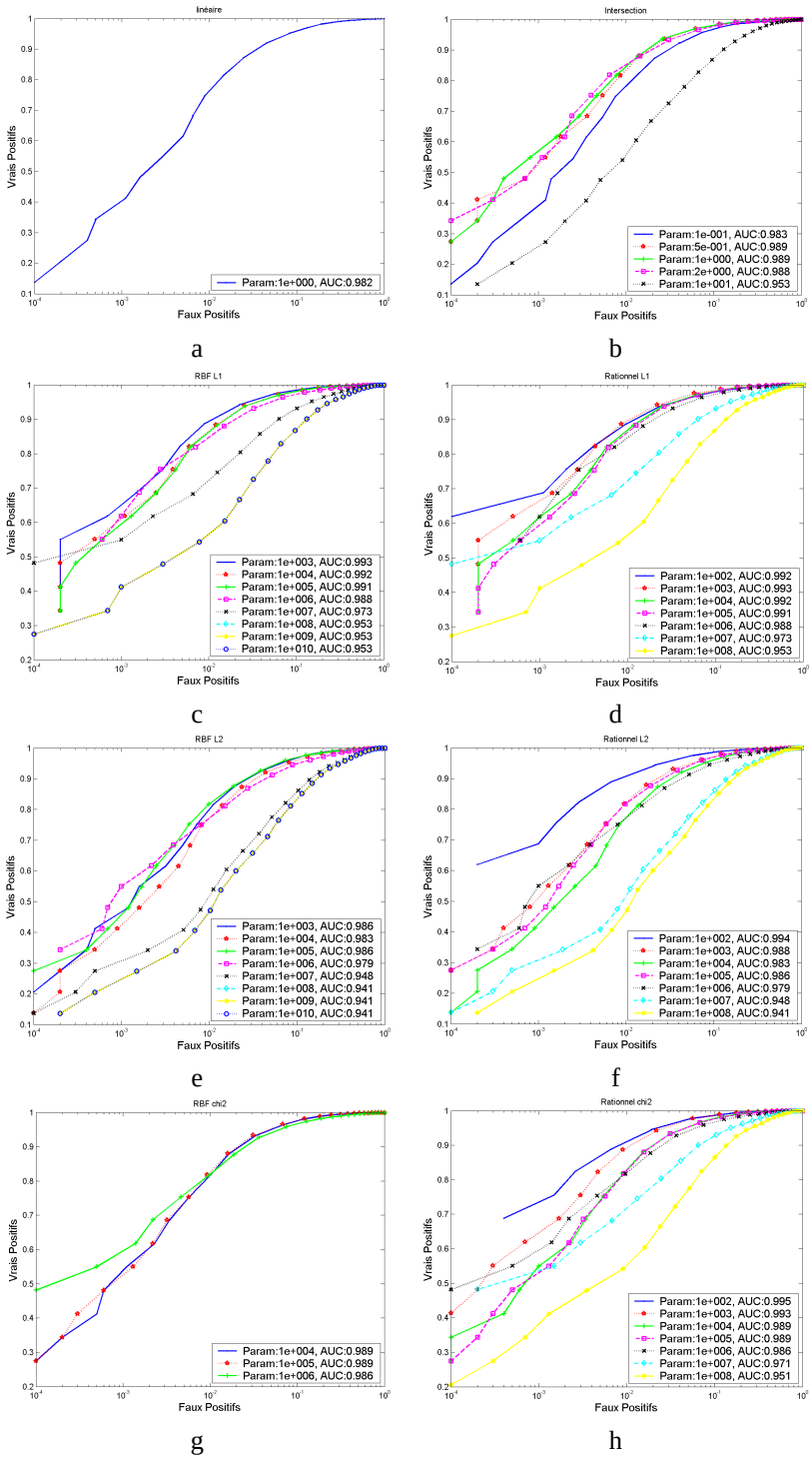


FIG. 4.16 : Courbes ROC obtenues pour évaluer les paramètres de chaque fonction noyau. (a) : noyau linéaire, aucun paramètre. (b) noyau GHI qui fait appel à une puissance. (c) est le noyau Laplacien, (d) le noyau rationnel L_1, (e) le noyau Gaussien, (f) le noyau rationnel L_2, (g) le noyau RBF χ^2 et (h) le noyau rationnel χ^2. Ces fonctions nécessitent un paramètre : la largeur de bande σ.

4.2.2 Evaluation des performances

Nous avons évalué indépendamment chaque fonction noyau, afin de retenir les meilleurs paramètres pour chacune d'elle.

La base d'images est celle utilisée pour évaluer les paramètres du HOG dans la section 4.1.3. Nous avons effectué dix itérations pour chaque paramètre, chacune d'elle utilisant une base d'apprentissage et de test choisies aléatoirement. Les bases sont définies préalablement afin que les conditions d'évaluation soient comparables pour tous les paramètres.

Le noyau linéaire ne nécessite aucun paramétrage. Les autres méthodes n'utilisent qu'un seul paramètre. Pour le noyau d'intersection d'histogrammes il s'agit ainsi du facteur de puissance β. Pour les noyaux RBF et rationnel, il s'agit du paramètre de largeur de bande σ.

Nous utilisons le noyau linéaire comme référence, car il ne dépend d'aucun paramètre. En effet, nous constatons sur la figure 4.16 que certaines fonctions se révèlent moins performantes lorsque le paramètre est mal choisi. Nous avons ainsi retenu les meilleurs paramètres pour chaque fonction.

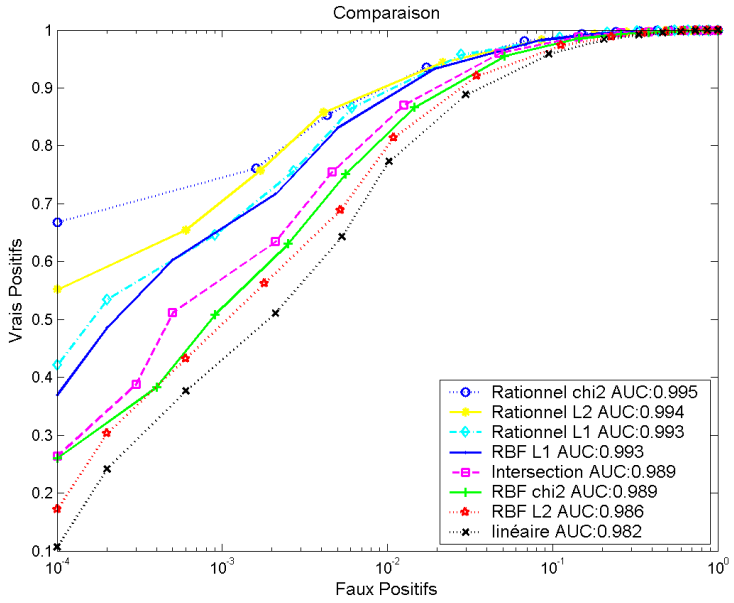


FIG. 4.17 : Courbes ROC des meilleurs résultats obtenus pour différentes fonctions noyau.

La figure 4.17 affiche les meilleurs résultats de chaque fonction, ce qui nous permet de les comparer. Les noyaux rationnels se révèlent les plus performants, notamment avec la fonction de distance χ^2. Le noyau GHI, est légèrement meilleur que le RBF L_2, mais moins performant que le RBF L_1.

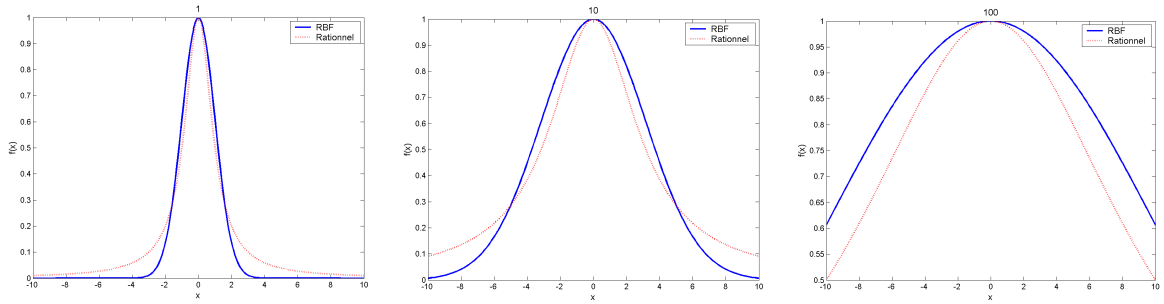


FIG. 4.18 : Cette figure montre les différences entre les noyaux RBF et rationnel en norme L2 lorsque la largeur de bande σ varie : 1 (a), 10 (b) et 100 (c).

Lorsque nous comparons la même fonction de distance avec un noyau rationnel et RBF (figure 4.18), nous expliquons le gain de performance par le fait que le noyau rationnel privilégie les valeurs proches sans pénaliser

cependant les valeurs plus éloignées. Dans son papier [76], Schiele et al. oppose ainsi une distance d'intersection d'histogrammes à une fonction χ^2. Les résultats obtenus ici sont similaires et montrent l'intérêt de l'utilisation de la fonction χ^2, adaptée aux distributions de probabilité, telles que les histogrammes.

Le noyau de référence, c'est à dire le noyau linéaire obtient des résultats corrects, mais est dépassé par toutes les fonctions présentées, ce qui nous conforte dans la définition d'un noyau plus adapté à la méthode HOG.

4.3 Application : détection de piétons à l'aide d'images infrarouges

Nous avons présenté une méthode de caractérisation globale d'images à l'aide d'histogrammmes d'orientation de gradient. Nous allons maintenant appliquer cette méthode à un système de détection de piétons par images infrarouges.

4.3.1 Contexte

Les systèmes d'acquisition d'images infrarouges deviennent de plus en plus attractifs, leurs prix devenant désormais accessibles pour des applications en série. Ce type de capteur est particulièrement intéressant pour le domaine de la route intelligente qui peut ainsi accéder à une nouvelle nature d'information sur l'environnement routier.

Une caméra infrarouge permet en effet de récupérer les informations liées à la chaleur produite par les objets en présence dans la scène observée. Ce type d'information peut non seulement ajouter de nouvelles données à des images du domaine visible, mais permet également de pallier les déficiences des caméras normales qui sont inopérantes lorsque les conditions de visibilité sont réduites, notamment la nuit.

L'application envisagée concerne la détection de piétons [8, 20, 30]. Nous allons ainsi présenter un système complet permettant de détecter les piétons à l'aide d'une seule caméra infrarouge en utilisant la méthode d'histogrammes d'orientation de gradient présentée précédemment dans la section 4.1.

4.3.2 Schéma de l'application

La méthode d'histogrammes d'orientation de gradient permet de décrire l'image d'un objet. Cependant, l'image doit contenir un seul objet, placé au centre et d'une certaine taille dans l'image. Nous avons vu dans la section 4.1.4.2 que la méthode donne des résultats optimaux lorsque la position et la taille du piéton des images testées correspond à la taille et à la position des images de la base d'apprentissage. Ainsi, la performance de la méthode sur des images complètes dépendra de l'extraction des piétons.

Il serait bien évidemment possible de redéfinir la base d'apprentissage en incluant des exemples de piétons non centrés et de taille différentes, mais cette modification se ferait au prix d'une base trop exhaustive. Nous avons donc préféré nous baser sur une méthode d'extraction de fenêtres fonctionnelles, permettant de détecter dans une seule image les piétons présents.

De plus, la définition exacte de la fenêtre contenant le piéton nous permet ensuite de localiser plus précisément le piéton et de mettre en place, par exemple, un suivi de piéton.

La figure 4.19 montre le déroulement de l'application. A partir d'une image complète, nous cherchons des fenêtres à extraire. Chaque image extraite est ensuite redimensionnée à une taille fixée puis est caractérisée à l'aide d'histogrammes de gradient. Ces vecteurs de caractérisques sont ensuite analysés par le classifieur SVM qui donne ainsi la classe d'appartenance des images extraites.

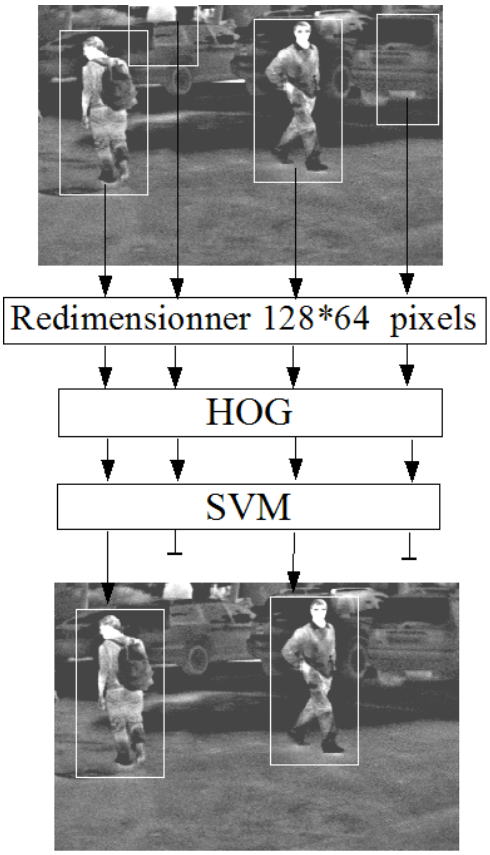


FIG. 4.19 : Schéma général de détection de piétons par images infrarouges.

4.3.3 Extraction d'images

Comme il s'agit d'une approche globale [68, 78], la détection des piétons dans l'image d'une scène complexe doit donc être précédée d'un balayage de l'image. L'approche intuitive consisterait à extraire de manière exhaustive toutes les fenêtres potentielles en balayant l'intégralité de l'image pour différentes tailles de fenêtres. Cette approche permet de s'affranchir des connaissances *a priori* sur l'image ou le contenu de la scène.

Cette solution n'est cependant pas viable car ce balayage exhaustif nécessiterait un temps beaucoup trop important pour être utilisable en des temps raisonnables. Une solution consistant à limiter la zone de balayage et la taille des fenêtres comme le propose Szarvas et al. [85] va également à l'encontre de la problématique de la détection de piéton liée à sa variabilité en taille et pose.

4.3.3.1 Détection des zones d'intérêt

Une autre approche consiste donc à localiser dans l'image des éléments propres aux piétons afin de limiter la zone de recherche. La question revient alors à définir ces éléments caractéristiques.

L'utilisation des images infrarouges est justifiée principalement par la possibilité de détecter les piétons de nuit. Les applications [8, 98] proposent d'exploiter les propriétés des images infrarouges qui permettent notamment de détecter les zones émettant de la chaleur. En effet, le piéton se distingue de son environnement par sa chaleur plus importante. Nous faisons l'hypothèse que la tête du piéton, ou une partie est découverte. Elle peut alors être différenciée très simplement du reste de l'image et des éléments de l'environnement. Ceux-ci n'émettent pas de chaleur et sont donc beaucoup plus sombres. La valeur des pixels est proportionnelle à la chaleur reçue. Un pixel

sombre déterminera ainsi une zone froide, tandis qu'un pixel clair sera caractéristique d'une zone chaude.

4.3.3.1.1 Seuillage L'intuition concernant les images infrarouges consiste à définir une simple méthode de binarisation de l'image pour séparer le piéton de son environnement. Nous allons donc dans un premier temps rechercher une méthode permettant d'extraire les piétons par un simple seuillage de l'image [7].

L'approche consiste donc à rechercher des régions d'intérêt dans l'image, la région étant définie comme un ensemble de pixels clairs. Ceci est effectué au moyen d'un simple seuillage. Le seuil est réglé de façon à détecter toutes les zones d'intérêt présentant des piétons sans générer des zones trop importantes en intégrant un voisinage très important autour de chaque zone.

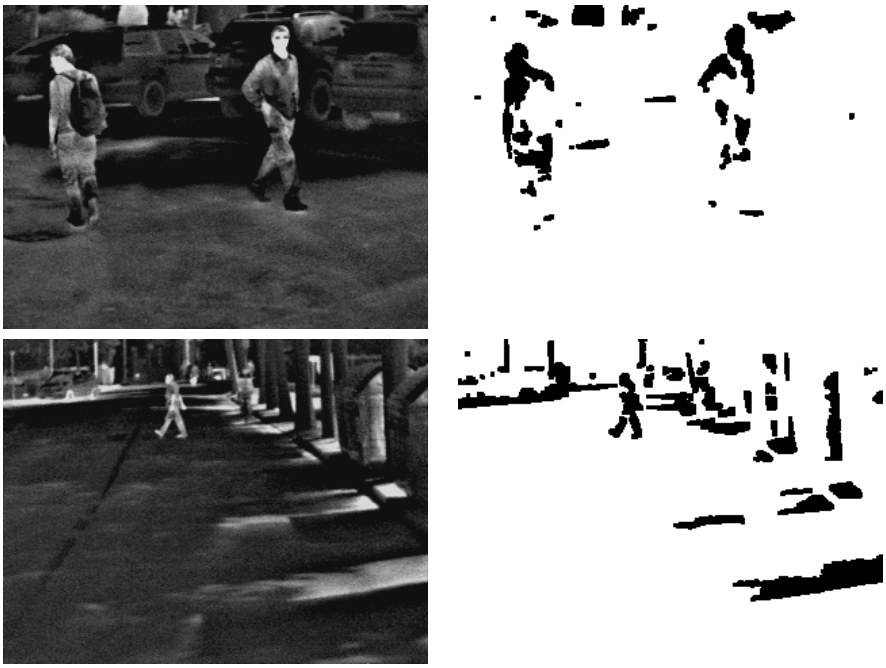


FIG. 4.20 : Illustration du seuillage d'une image infrarouge.

Cependant, cette hypothèse suppose que le piéton émet une chaleur homogène. Les images infrarouges permettent de visualiser un piéton placé loin de la caméra. Pour cette approche, l'hypothèse est vérifiée, le piéton présentant une homogénéité globale.

Cette hypothèse n'est malheureusement pas valable lorsque le piéton est proche de la caméra, les vêtements du piéton ne présentant pas une surface homogène, comme le montre la figure 4.20. De près, la situation est donc comparable à des images visibles.

4.3.3.1.2 Extraction de fenêtres de taille fixée

Nous proposons donc de détecter dans un premier temps les régions d'intérêt, puis d'extraire autours de ces régions d'intérêt des fenêtres contenant potentiellement des piétons. La figure 4.21 montre ainsi un exemple d'extraction de zones d'intérêt.

Nous faisons l'hypothèse que la région d'intérêt représente la tête du piéton, celle-ci étant fréquemment découverte, donc plus facilement détectable.

Pour définir une fenêtre autour de cette région, le bord supérieur de la fenêtre coïncidera donc avec le bord supérieur de la région. En supposant également que la tête du piéton est centré dans l'image, nous pouvons positionner les frontières verticales de part et d'autre de la zone d'intérêt.

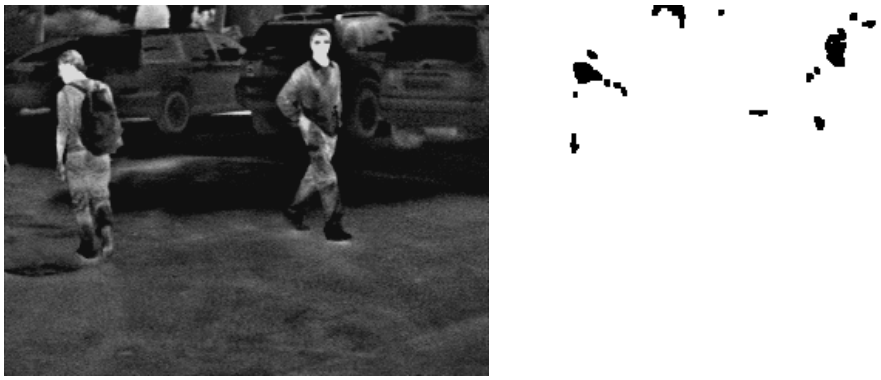


FIG. 4.21 : Zones d'intérêt dans l'image binaire obtenue par seuillage (droite) obtenue à partir d'une image infrarouge (gauche).

Cependant, nous ne disposons d'aucune information *a priori* sur la taille ou la position du piéton. Nous devons donc définir au préalable différentes configurations possibles de fenêtres, c'est à dire différentes largeurs et hauteurs de fenêtres.

Nous combinons ainsi les différentes tailles de largeur et hauteur afin d'extraire un ensemble de fenêtres. Le but consiste à extraire suffisamment de fenêtres pour couvrir les postures et tailles possibles du piétons, sans aller jusqu'au balayage exhaustif.

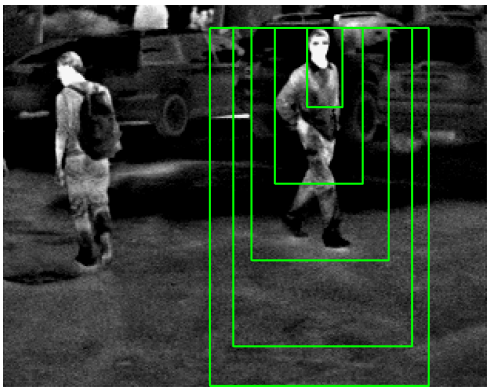


FIG. 4.22 : Image infrarouge (gauche) et exemples de fenêtres dont la taille est fixée au préalable (gauche).

4.3.3.1.3 Recherche de critères déterminants

Il est donc important de pouvoir mettre en place une méthode qui ne s'appuie sur aucun élément connu. Cela consiste donc à définir les frontières des fenêtres, en cherchant à nouveau des éléments caractéristiques. L'extraction devra ainsi tester plusieurs configurations de fenêtres autour des zones d'intérêts.

Nous utilisons la même approche pour définir les fenêtres que dans la méthode précédente, mais au lieu de définir des tailles rigides de fenêtres, nous utilisons une information locale afin de positionner au mieux les frontières des fenêtres.

Sur la figure 4.21, nous montrons les zones d'intérêts extraites de l'image. Une fois que les zones d'intérêt sont localisées, il faut ensuite extraire des fenêtres autour de ces zones.

Le piéton étant différent de son environnement nous pouvons observer des variations au niveau des pixels. Nous pouvons alors extraire des informations relatives aux contours du piéton afin de définir les frontières des fenêtres. Sur la figure 4.23, nous montrons ainsi l'image de la norme du gradient ainsi que la segmentation par contours en

utilisant la méthode de Sobel [15].

Pour définir les frontières des fenêtres nous utilisons l'information de gradient. L'idée consiste à détecter les zones importantes de variations dans l'image afin de définir la position des fenêtres [8, 30].

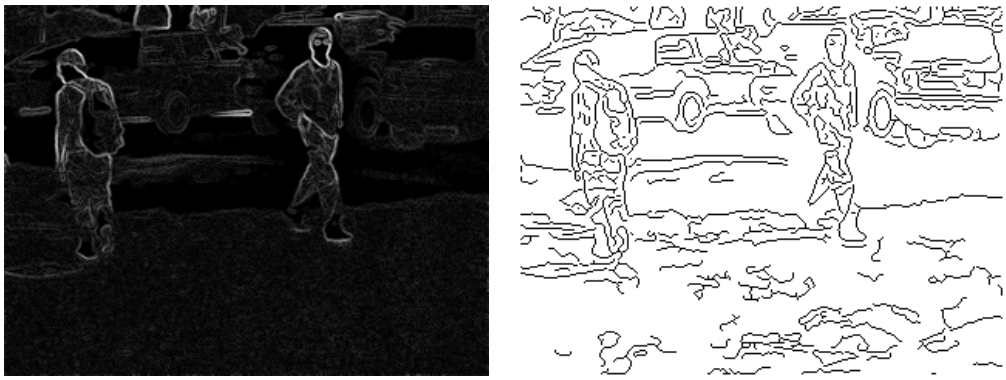


FIG. 4.23 : Image de la norme du gradient (gauche) et contours extraits par la méthode de Sobel (gauche).

Pour illustrer la suite de cette méthode, nous nous focaliserons sur une seule région d'intérêt : la tête du piéton de droite. Nous utilisons également la description appuyée sur la norme du gradient.

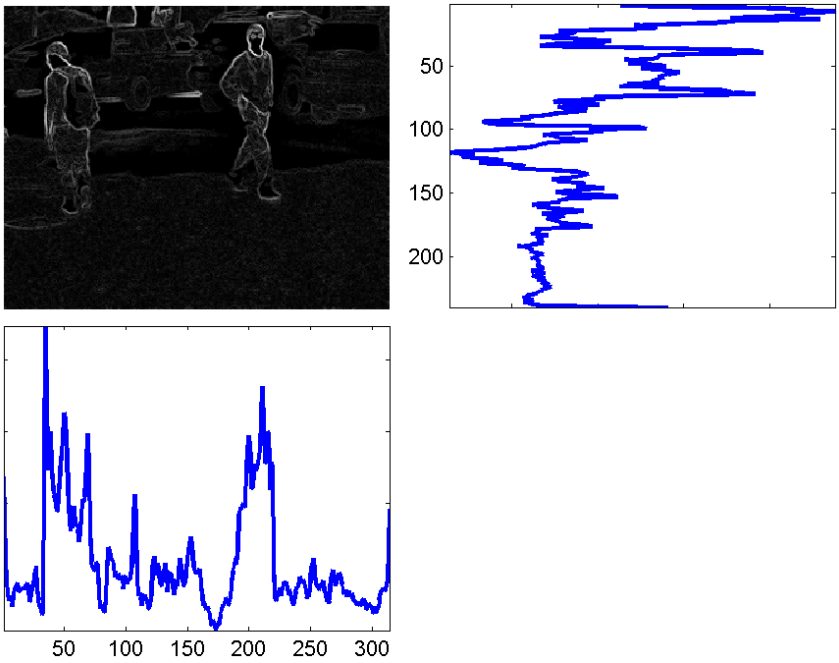


FIG. 4.24 : Somme verticale (image inférieure) et horizontale (image supérieure droite) de la norme du gradient (image supérieure gauche).

Nous ne disposons d'aucune information *a priori* sur la position, la forme ou la taille du piéton. Nous allons donc extraire un ensemble de fenêtres autour de cette zone d'intérêt. Pour cela, nous allons chercher des frontières verticales et horizontales en tenant compte des variations dans l'image.

Nous calculons ainsi la somme verticale de la norme du gradient. Nous appliquons la même méthode lorsque la description est calculée à partir des contours. Les colonnes contenant le plus de variations dans l'image auront ainsi de fortes valeurs pour la norme du gradient. Ces colonnes correspondent donc aux contours verticaux des objets présents dans l'image. Nous cherchons les maxima locaux pour définir plus précisément les colonnes potentielles (figure 4.25).

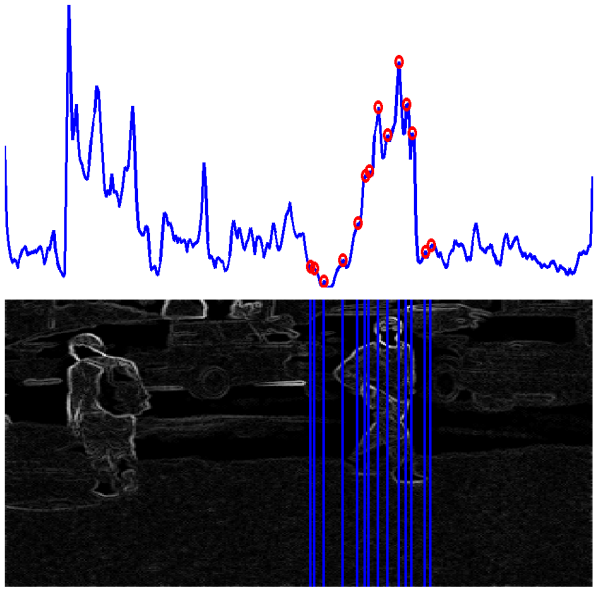


FIG. 4.25 : Calcul des maxima locaux de la somme verticale du gradient de l'image.

Nous définissons ainsi les frontières gauches et droites grâce aux maxima locaux de la somme verticale du gradient.

Nous opérons de la même manière pour extraire les maxima locaux horizontaux et obtenons ainsi un ensemble de frontières gauche, droite et inférieure.

Nous supposons que la zone d'intérêt est la tête du piéton. Nous pouvons ainsi déterminer la frontière supérieure des fenêtres par rapport à cette zone d'intérêt.

La figure 4.26 montre ainsi les ensembles des frontières verticales et horizontales extraites autour d'une région particulière.

FIG. 4.26 : Définition des frontières pour la région d'intérêt encadrée : gauches et droites (image de gauche), frontière supérieure et inférieures (image de droite).

Enfin, nous combinons ces frontières entre elles, afin de définir un ensemble de fenêtres autour de la région d'intérêt. Nous obtenons ainsi un certain nombre de fenêtres qui nous permettent de tester la présence d'un piéton dans l'image.

Nous extrayons ainsi l'image associée à chaque fenêtre définie, nous calculons le descripteur à l'aide de la

méthode HOG, puis nous appliquons la classification de ces descripteurs.

4.3.4 Evaluation

Nous avons présenté plusieurs approches pour extraire les fenêtes d'une image infrarouge. Pour tester et comparer les différentes méthodes mises en place, nous avons mis au point une fonction permettant d'évaluer leur efficacité.

Nous avons en effet deux critères à prendre en compte. Pour évaluer la performance de la méthode HOG sur une base d'images, nous calculons la courbe ROC obtenue pour l'ensemble des images. C'est à dire que nous comparons les valeurs des prédictions pour différencier les classes. Cependant, nous avons un critère supplémentaire qui concerne la définition de la fenêtre. Il faut tenir compte de la pertinence du positionnement d'une fenêtre. L'évaluation doit donc tenir compte du recouvrement des piétons détectés.

Pour cela, nous comparons les fenêtres extraites automatiquement avec les fenêtres définies manuellement. Celles-ci n'englobent pas systématiquement au mieux le piéton mais permettent de donner un point de comparaison.

Nous avons donc défini un score qui prend en compte la prédiction de la classification des fenêtres, mais également le recouvrement des fenêtres avec leur définition théorique.

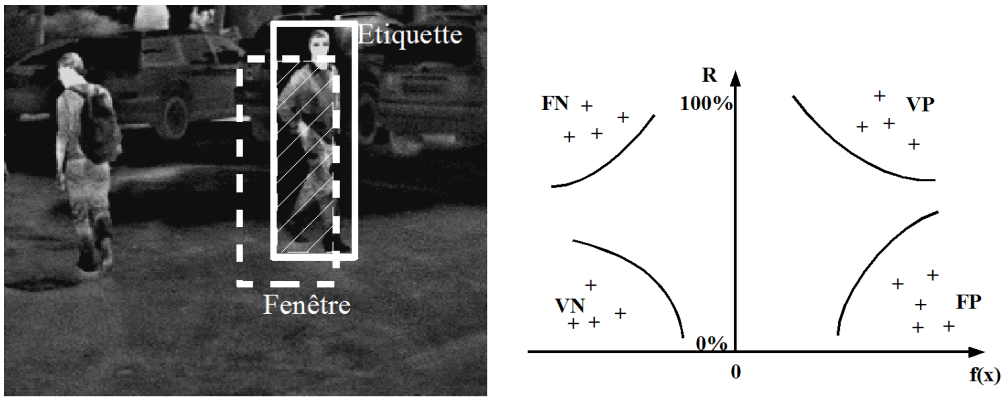


FIG. 4.27 : L'image de gauche (a) présente un exemple de recouvrement entre une fenêtre définie manuellement (traits pleins) et une fenêtre extraite automatiquement (traits pointillés). A droite (b), le schéma présente l'évaluation d'une méthode d'extraction automatique de fenêtres.

Sur la figure 4.27, nous illustrons le calcul du recouvrement. Nous déterminons la région qui forme l'intersection (région hachurée) entre la fenêtre définie de façon théorique (traits pleins) et une fenêtre extraite de l'image (traits pointillés). Le recouvrement est calculé de la façon suivante :

$$R = \frac{S_{\cap}}{S_T + S_P}, \tag{4.12}$$

où $S_{\cap}$ est la surface en pixels de l'intersection entre la surface S_T de la fenêtre théorique et S_P la fenêtre extraite en pratique.

Cette figure 4.27, présente également l'évaluation des méthodes d'extraction automatique. Nous pouvons ainsi positionner chaque fenêtre dans un graphique, dont les coordonnées dépendent en abscisse de la valeur de la prédiction du classifieur et en ordonné de la valeur de recouvrement de la fenêtre théorique. Cette figure nous permet ainsi de visualiser non seulement les résultats du classifieur, mais également l'efficacité de la méthode d'extraction.

Nous pouvons ainsi évaluer le nombre de vrais positifs, c'est à dire les fenêtres qui recouvrent au mieux le piéton et obtiennent une valeur de prédiction élevée, les faux positifs définis par un faible recouvrement malgré une valeur de prédiction élevée, les vrais négatifs qui ne recouvrent pas un piéton et obtiennent une valeur de prédiction faible et enfin les faux négatifs qui recouvrent correctement le piéton mais obtiennent une valeur de prédiction faible.

Pour retenir la meilleure méthode, nous conservons le schéma classique, ce sera donc celle qui maximise le nombre de vrais positifs et vrais négatifs, tout en minimisant le nombre de faux négatifs et faux positifs.

Nous allons maintenant présenter quelques résultats obtenus sur une séquence d'images infrarouges.

4.3.5 Comparaison des méthodes d'extraction

Pour évaluer la méthode la plus pertinente d'extraction de fenêtres, nous avons ainsi comparé les différentes approches pour chercher les critères de position des frontières. Le but est double, évaluer la pertinence et la performance des différentes méthodes.

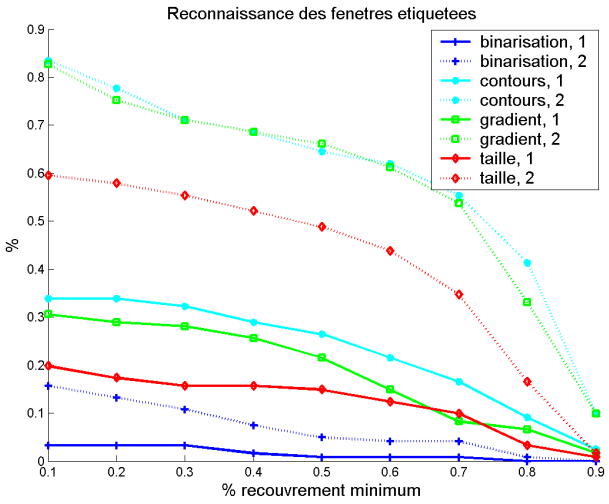


FIG. 4.28 : Taux de bonne reconnaissance des fenêtres étiquetées manuellement en fonction du pourcentage de recouvrement minimum pour chaque méthode d'extraction avec une classification 1 ou 2 classes.

Nous avons ainsi comparé les différentes méthodes d'extraction de fenêtres. Nous avons retenu 80 images complètes, soit 121 piétons au total, présents sous différentes formes. Pour comparer les méthodes, nous faisons varier le pourcentage minimum de recouvrement entre les fenêtres théoriques et extraites, puis nous comptabilisons le nombre de bonnes détections et d'erreurs pour chaque méthode.

La figure 4.28 présente les résultats obtenus sur des images étiquetées manuellement. La figure 4.29 présente les performances au niveau du nombre de faux positifs.

Comme nous le constatons, les méthodes appuyées sur la recherche de caractéristiques significatives telles que la norme du gradient ou la présence de contours permettent d'obtenir des meilleurs résultats que le simple seuillage des images ou la définition préalable des tailles de fenêtres.

Nous comparons également la classification *one-class* et bi-classes. L'avantage du *one-class* permet de s'affranchir de la définition de la base d'apprentissage des exemples négatifs. Elle obtient ainsi un taux de faux positifs plus intéressant que la classification bi-classes, mais est moins performante pour détecter les piétons.

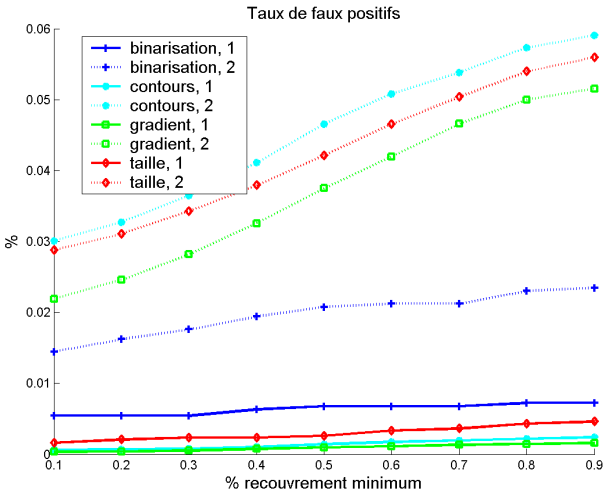


FIG. 4.29 : Taux de faux positifs en fonction du pourcentage de recouvrement minimum pour chaque méthode d'extraction avec une classification 1 ou 2 classes.

4.3.6 Résultats

Nous allons maintenant illustrer les résultats obtenus par quelques exemples.

La figure 4.30 présente ainsi quelques résultats obtenus sur des images complètes en utilisant une extraction de fenêtres à l'aide de l'information de gradient. Nous constatons ainsi qu'il est possible de détecter les piétons quelque soit leur taille dans l'image.

Cependant, nous avons quelques fausses alarmes. En particulier, nous notons que des parties de voitures ou des piétons tronqués sont reconnues comme piétons. Ceci peut être expliqué en partie par le fait que la base d'apprentissage contient des piétons proches d'une voiture. La base contient également des piétons tronqués lorsqu'il sont proches de la caméra et n'apparaissent pas intégralement dans l'image.

De plus, nous ne tenons pas compte de l'ensemble des résultats, chaque fenêtre est indépendante. Nous pourrions donc améliorer les résultats en considérant l'ensemble de l'image et en supprimant ainsi les détections redondantes. L'implémentation d'une méthode de suivi permettrait également d'améliorer les performances.

4.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode permettant de caractériser une image selon une approche globale. A partir d'une image contenant un objet unique centré, nous calculons des histogrammes locaux d'orientation de gradient [21, 22]. Ces histogrammes sont ensuite utilisés comme descripteurs pour analyser l'image.

En utilisant un classifieur SVM linéaire, nous avons pu constater que cette méthode obtient de bons résultats en classification sur une base d'images extraites manuellement.

Nous avons ensuite confronté ce descripteur à la problématique principale, c'est à dire la variabilité du piéton. Le descripteur présente de bonnes capacités de généralisation, car le calcul du descripteur est effectué sur des régions locales dont la taille est suffisamment petite pour éviter certaines perturbations apparaissant dans une autre partie de l'image. De plus, l'information retenue pour caractériser l'image est l'orientation du gradient. Nous ne tenons donc pas compte de l'apparence de l'objet décrit, mais principalement sa forme.

Nous avons également comparé cette approche à différentes méthodes de description globale, qui sont actuel-

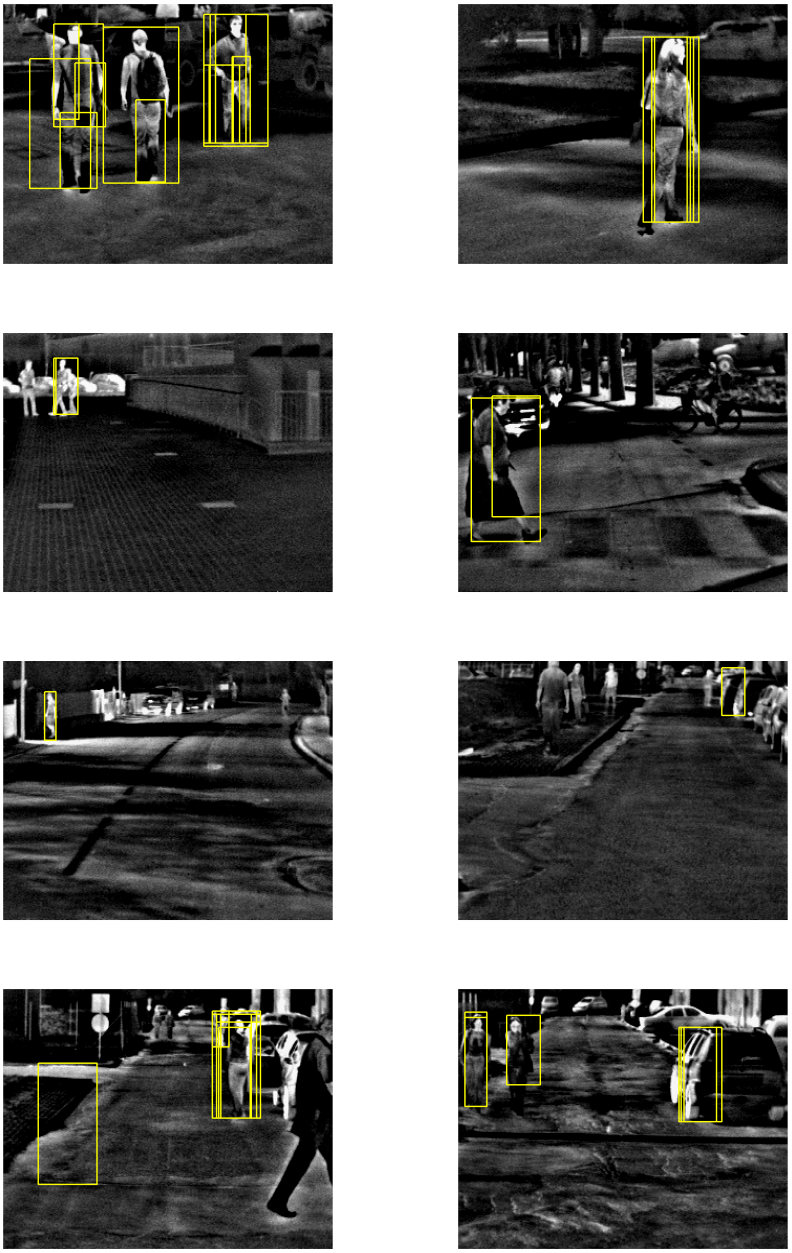


FIG. 4.30 : Résultats obtenus sur des images infrarouges à l'aide d'une extraction de fenêtres utilisant l'information de gradient.

lement les références dans le domaine de la détection de piétons.

Pour améliorer les performances de cette méthode, nous avons testé différentes formulations de noyaux, certaines étant plus adaptées à des descripteurs d'histogrammes que le produit scalaire.

Cependant, ce descripteur ne peut être appliqué que sur des images contenant un objet unique. L'application de cette méthode pour la détection de piétons sur une image complète nécessite donc l'extraction de fenêtres dans cette image [84]. Ces fenêtres sont par la suite décrites à l'aide de la méthode HOG, puis analysées afin de détecter la présence ou non d'un piéton.

Nous avons appliqué cette méthode sur des images infrarouges [8, 20, 30]. Les propriétés des images infra-

rouges permettent en effet de visualiser la scène observée en fonction des chaleurs émises. Ainsi, un piéton peut être distingué de son environnement, car il présente une différence de température. Nous pouvons ainsi utiliser cette propriété pour extraire de l'image des régions d'intérêt qui nous permettent de localiser globalement le piéton dans l'image. Il est ensuite nécessaire d'extraire des fenêtres autours de ces régions d'intérêt. Cette approche nous permet ainsi d'éviter un balayage exhaustif de l'image [68, 85]. Un balayage exhaustif n'apporte pas de solution idéale pour contrer la variabilité du piéton. Il faudrait en effet extraire des fenêtres de toutes les tailles possibles.

Nous proposons donc d'utiliser des caractéristiques significatives de la présence d'un piéton en cherchant des zones d'intérêt telles que les sources de chaleur, puis la présence de contours dans l'image afin de positionner plus précisément les fenêtres dans l'image.

Nous constatons donc quelques limitations à cette approche. Tout d'abord elle permet de décrire une image non segmentée, mais n'est pas applicable directement pour effectuer une détection de piétons des images complètes.

Les performances de la méthode sont optimales lorsque la position du piéton dans l'image évaluée est la plus proche de la position générale des piétons dans les images de la base d'apprentissage. Si nous ne souhaitons pas développer la base d'apprentissage pour résoudre ce problème, la solution consiste à extraire au mieux les images des piétons. Cela nécessite donc une méthode performante pour l'extraction des fenêtres.

La perspective consiste donc à tester une approche différente pour localiser les piétons dans l'image, avec une approche de type *coarse-to-fine*, par exemple.

Une autre perspective consiste à définir une description permettant de tenir compte des changements de posture des piétons. Le but serait ainsi de calculer le produit scalaire entre deux descripteurs en cherchant par exemple la cellule la plus proche dans un voisinage, au lieu de mettre en correspondance les cellules à la même position. Cette approche permettrait ainsi de réduire davantage la taille de la base d'apprentissage et d'améliorer encore les capacités de généralisation.

Conclusion

« En essayant continuellement on finit par réussir. Donc : plus ça rate, plus on a de chance que ça marche. »

Jacques Rouxel

Le domaine de reconnaissance de formes appliquée à la détection de piétons est confrontée à la nature même du piéton. Différentes variabilités entrent en compte : la variabilité à l'échelle, de posture et d'apparence. En effet, dans le cas des données de type image, le piéton peut être positionné à divers emplacements dans l'image, sous différents aspects et points de vue. Nous avons donc choisi d'orienter nos travaux concernant la détection de piétons afin de tenter une résolution de ces problèmes.

Contributions

Une partie de la réponse à ces problèmes de variabilité peut être apportée par le choix pertinent de la représentation. Mais le bon déroulement d'un algorithme de reconnaissance de formes est également étroitement lié à la méthode de classification utilisée. Dans notre cas, nous faisons appel à des machines à noyaux, en l'occurence le classifieur SVM (*Support Vector Machine*), reconnu actuellement pour ses bonnes performances.

Nos travaux ont donc porté sur l'étude d'une méthode de caractérisation adéquate. Tout d'abord, nous exploitons la représentation à l'aide de graphes qui possèdent de nombreuses propriétés topologiques intéressantes pour résoudre le problème de variabilité. S'agissant de graphes étiquetés, le choix des informations valuant le graphe se révèle prépondérant afin de conserver les propriétés inhérentes aux graphes. Il est ainsi possible de résoudre la variabilité à l'échelle, car la forme du graphe n'est pas modifiée, à la posture si les étiquettes sont choisies soigneusement. De plus, nous décrivons la forme de l'objet. L'apparence aura donc un impact très faible sur la représentation. La représentation par graphes nécessite cependant des travaux préliminaires afin d'extraire de l'image un masque binaire pour chaque objet à reconnaître. Nous construisons en effet le graphe à partir du squelette morphologique du masque binaire.

La classification s'appuie sur un noyau qui permet de calculer le produit scalaire entre les données. Cette fonction est primordiale pour le classifieur. Cependant, les graphes sont des structures complexes incompatibles avec les fonctions noyaux classiques tels que le noyau gaussien ou laplacien. Il est donc nécessaire de définir un produit scalaire entre graphes. Nous considérons pour cela les graphes comme des ensembles de chemins. Un chemin est en effet un sous-graphe et contient donc une information structurée. Un graphe peut ainsi être décomposé en différents chemins. La comparaison entre deux graphes se résume donc à la comparaison de sacs de chemins.

Pour appliquer cette méthode pratiquement, nous proposons d'utiliser un système d'acquisition stéréovision qui nous permet de segmenter l'image en séparant les objets selon leur position dans le monde réel. Cette position est calculée en fonction de la différence de position des pixels dans les images gauche et droite. La différence sera ainsi proportionnelle à la profondeur.

Pour s'affranchir des problèmes de segmentation, nous avons alors étudié une seconde méthode qui utilise une caractérisation globale de l'image. La représentation utilise ici des histogrammes locaux d'orientation de gradient. Comme l'image décrite par cette méthode ne peut contenir qu'un objet unique, la recherche de plusieurs piétons dans une image entière nécessite une extraction préalable d'imagettes dont le contenu doit être analysé.

Nous avons dans un premier temps expérimenté les possibilités de cette représentation et l'avons comparé à d'autres méthodes de description globale. Elle s'est révélée très performante et possède un pouvoir de généralisation intéressant pour notre application de détection de piétons. Bien que la conception du descripteur ne soit pas spécifiquement dédié à résoudre la variabilité du piéton, les performances obtenues permettent d'utiliser cette méthode avec de bonnes performances. La raison de cette capacité de généralisation réside partiellement dans la division de l'image en cellules, réduisant ainsi l'impact d'une altération partielle de l'image.

De plus, pour améliorer les performances obtenues en classification, nous avons remplacé la formulation initiale fondée sur un noyau linéaire par un noyau plus adapté aux descripteurs représentant des distributions de probabilités. Nous avons ainsi étudié différentes formulations de noyaux afin de pouvoir améliorer les performances et déterminer la fonction la mieux adaptée à cette représentation.

Nous avons enfin choisi d'appliquer cette méthode à la reconnaissance de piétons à l'aide d'images infrarouges. Pour cela, nous avons dû concevoir une méthode permettant d'extraire des fenêtres candidates de l'image entière, car la description ne porte que sur des images contenant un unique objet. Nous avons comparé différentes approches d'extraction de fenêtres en utilisant les propriétés des images infrarouges et des éléments significatifs de la présence d'un piéton, comme par exemple les contours. Cette application permet également de résoudre le problème de variabilité à l'échelle, car les fenêtres extraites sont redimensionnées à la même taille. De plus, nous avons pris en compte les contraintes liées à la variabilité d'échelle et de posture pour concevoir la méthode d'extraction, afin de ne pas limiter la détection aux piétons debouts, de taille normale, sans posture particulière.

Limites des approches proposées

Dans le cas de la représentation par graphes, la principale limitation est liée à la construction des graphes. Tout d'abord, la méthode de segmentation est relativement complexe à mettre en œuvre. L'utilisation d'un système stéréoscopique est en effet coûteux en temps de calcul et nécessite la mise au point d'algorithmes très efficaces pour exploiter de manière optimale les possibilités offertes par la stéréovision. De plus, nous avons vu que l'obtention des graphes est relative à la squelettisation des masques binaires. Ici encore, les résultats obtenus sont différents selon la définition du squelette et l'implémentation utilisée. Nous n'avons cependant pas étudié d'autres approches pour concevoir les graphes. Les graphes auraient aussi bien pu être utilisés pour décrire un agencement de régions ou de points d'intérêt.

Notre approche initiale consistait à décrire la forme uniquement. Nous avons donc choisi de définir les graphes à partir des squelettes morphologiques et de ne retenir que des informations sur la topologie de l'objet dans les étiquettes des nœuds et des graphes. Les résultats obtenus sont prometteurs, mais l'ajout d'informations de nature différente nous aurait certainement permis d'améliorer les performances de cette méthode.

Une autre limitation dans l'utilisation des graphes réside dans la complexité de la formulation du noyau de graphe. Les modifications que nous proposons en remplaçant les marches aléatoires par des chemins directs nous permettent de réduire cette complexité. Mais l'application ne peut être envisagée que sur des graphes de taille réduite. L'utilisation d'une formulation de type «sac de chemins» n'est peut être pas la meilleure solution pour définir un produit scalaire entre graphes lorsque ceux-ci présentent un ordre élevé.

Dans le cas de la description à l'aide d'histogrammes d'orientation de gradient, l'application s'affranchit théoriquement de la segmentation. En pratique, une telle méthode est difficile à utiliser sans connaissance *a priori* sur l'image ou des informations contextuelles. La recherche des piétons dans une image complète se révèle en effet

très coûteuse en temps de calcul car il n'est pas envisageable de réduire le nombre de fenêtres extraites sans retomber dans les problèmes inhérents à la variabilité du piéton tels que la taille ou la posture. L'utilisation optimale consisterait donc à combiner une recherche complète ainsi qu'un travail sur des séquences vidéos afin d'éviter la redondance des recherches inutiles et de focaliser l'extraction sur des zones intéressantes.

Perspectives

Ces limitations nous permettent ainsi d'envisager des améliorations et des pistes de recherche futures.

Il faudrait dans un premier temps étudier d'autres approches de construction de graphes afin d'approfondir les capacités de cette méthode de représentation. Nous n'avons également pas complètement exploité les possibilités offertes par les graphes. Nous utilisons actuellement des vecteurs de scalaires pour étiqueter les graphes, mais nous envisageons d'intégrer des structures plus complexes dans ces étiquettes. Nous pourrions intégrer une information sur le voisinage des nœuds plus complète en utilisant, par exemple, une description constituée d'histogrammes de gradient. Il est également possible d'envisager des étiquettes structurées, telles que des graphes, dans les nœuds.

Il existe une grande variété d'étiquettes possibles pour les nœuds et les arcs. Afin de retenir les informations les plus pertinentes, il faudrait déterminer automatiquement les paramètres liés aux étiquettes et sélectionner les informations discriminantes.

Une dernière perspective appuyée sur la représentation par graphe est la recherche de sous-graphe. Le but consiste à définir la partie d'un graphe de référence qui correspond le mieux à un graphe requête issu d'un objet occulté, par exemple. Un algorithme de type *forward-backward* permettrait ainsi d'élaguer le graphe de référence jusqu'à obtenir le graphe le plus proche du graphe requête.

Nous avons également quelques perspectives pour améliorer la description par histogramme de gradient. Nous envisageons tout d'abord d'étudier de nouvelles méthodes de segmentation pour améliorer la recherche des piétons dans l'image. Il est par exemple envisagé d'utiliser des méthodes de suivi pour améliorer la détection et prédire les dangers potentiels pour les piétons. Nous utilisons actuellement une seule description, mais il est envisageable de la combiner avec des descripteurs supplémentaires afin d'améliorer encore les performances.

Nous envisageons également de définir un noyau permettant de comparer différemment les histogrammes entre deux descripteurs. Actuellement, le calcul d'un produit scalaire entre deux descripteurs consiste à comparer les histogrammes des cellules dont les positions respectives sont les mêmes. Tout d'abord, nous proposons de prendre en compte la position de la cellule dans l'image, pour pondérer certains histogrammes moins importants car contenant l'environnement du piéton. Nous envisageons également de définir un noyau permettant de comparer chaque cellule d'une image avec le voisinage correspondant dans l'autre image. Ainsi, lorsque la position du piéton subit une translation dans l'image, le résultat serait le même que s'il était centré. Cette approche permettrait donc d'améliorer les capacités de généralisation et donc de définir une base d'exemples pour l'apprentissage moins importante.

Enfin, nous continuerons d'étudier de nouvelles approches possibles pour la détection de piétons afin d'apporter de nouveaux éléments pour répondre au problème de variabilité. Les méthodes fondées sur les points d'intérêt présentent notamment des ouvertures intéressantes pour une approche sans segmentation.

Bibliographie

[1] Nachman Aronszajn and Kennan Smith. Functional spaces and functional completion. *Annales de l'institut Fourier*, 6 :125–185, 1956.

[2] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML'04 : Proceedings of the twenty-first international conference on Machine learning*, page 6, New York, NY, USA, 2004. ACM Press.

[3] Stephen Barnard and Martin Fischler. Computational stereo. *Computing Surveys*, 14 :553–572, December 1982.

[4] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Boles, and Helen C. Wolf. Parametric correspondence and chamfer matching : Two new techniques for image matching. In *IJCAI*, pages 659–663, 1977.

[5] Claude Berge. *Théorie des Graphes et ses Applications*. Collection Universitaire de Mathématiques, Dunod, 1958.

[6] Massimo Bertozzi, Alberto Broggi, Gianni Conte, and Alessandra Fascioli. Stereo vision system performance analysis. In Sergio Taraglio and Vincenzo Nanni, editors, *Enabling Technologies for the PRASSI Autonomous Robot*, pages 68–73. ENEA, Rome, Italy, jan 2002.

[7] Massimo Bertozzi, Alberto Broggi, Michael Del Rose, and Andrea Lasagni. Infrared Stereo Vision-based Human Shape Detection. In *Procs. IEEE Intelligent Vehicles Symposium 2005*, pages 23–28, Las Vegas, USA, June 2005.

[8] Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli, Thorsten Graf, and Marc-Michael Meinecke. Pedestrian detection for driver assistance using multiresolution infrared vision. *IEEE Trans. on Vehicular Technology*, 53(6) :1666–1678, nov 2004.

[9] Harry Blum. A transformation for extracting new descriptors of shape. In Wathen Dunn, editor, *Proceedings of Models for the Perception of Speech and Visual Form*, pages 362–380, 1967.

[10] Sabri Boughorbel, Jean-Philippe Tarel, and Nozha Boujemaa. Generalized histogram intersection kernel for image recognition. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, volume III, pages 161 – 164, Genova, Italy, 2005.

[11] Alberto Broggi, Massimo Bertozzi, Roland Chapuis, Frédéric Chausse Alessandra Fascioli, and Amos Tibaldi. Pedestrian localization and tracking system with kalman filtering. In *Procs. IEEE Intelligent Vehicles Symposium 2004*, pages 584–589, Parma, Italy, jun 2004.

[12] Alberto Broggi, Massimo Bertozzi, Alessandra Fascioli, and Gianni Conte. *Automatic Vehicle Guidance : the Experience of the ARGO Vehicle*. World Scientific, Singapore, apr 1999.

[13] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recogn. Lett.*, 19(3-4) :255–259, 1998.

[14] Olivier Chapelle, Patrick Haffner, and Vladimir Vapnik. Svms for histogram based image classification. *IEEE Transactions on Neural Networks*, 9, 1999.

[15] Jean-Pierre Cocquerez and Sylvie Philipp. *Analyse d'images : filtrage et segmentation*. Masson, 1995.

[16] Donatello Conte, Pasquale Foggia, Jean-Michel Jolion, and Mario Vento. Un algorithme multirésolution pour la gestion des occlusions basé sur les pyramides de graphes. In *Compression et Représentation des Signaux Audiovisuels, CORESA 2005, Rennes*, November 2005.

[17] Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational kernels : Theory and algorithms. *Journal of Machine Learning Research*, 5 :1035–1062, 2004.

[18] Nello Cristianini and John Shawe-Taylor. *Introduction to Support Vector Machines*. Cambridge Univeristy Press, 2000.

[19] Christopher M. Cyr and Benjamin B. Kimia. 3d object recognition using shape similarity-based aspect graph. In *ICCV*, pages 254–261, 2001.

[20] Congxia Dai, Yunfei Zheng, and Xin Li. Layered representation for pedestrian detection and tracking in infrared imagery. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 13, Washington, DC, USA, 2005. IEEE Computer Society.

[21] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, June 2005.

[22] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision, Austria*, pages 428–441, May 2006.

[23] Jan Giebel Dariu Gavrila and Stefan Munder. Vision-based pedestrian detection : the protector+ system. In *Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy*, pages 13–18, June 2004.

[24] Edsger Dijkstra. A note on two problems in connexion with graphes. *Numerische Mathematik*, 1 :269–271, 1959.

[25] Pavel Dimitrov, Carlos Phillips, and Kaleem Siddiqi. Robust and efficient skeletal graphs. In *Computer Vision and Pattern Recognition, Hilton Head Island, USA*, volume 1, pages 417–423, 2000.

[26] Gyuri Dorkó and Cordelia Schmid. Maximally stable local description for scale selection. In *European Conference on Computer Vision, Graz, Austria*, 2006.

[27] Richar Duda, Peter Hart, and David Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[28] Jan Eichhorn and Olivier Chapelle. Object categorization with svm : kernels for local features. Technical report, MPIK, July 2004.

[29] Hadi Elzein, Sridhar Lakshmanan, and Paul Watta. A motion and shape-based pedestrian detection algorithm. In *Intelligent Vehicles Symposium, 2003*, pages 500–504, June 2003.

[30] Yajun Fang, Keiichi Yamadac, Yoshiki Ninomiya, Berthold Horn, and Ichiro Masaki. Comparison between infrared-image-based and visible-image-based approaches for pedestrian detection. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 505–510, June 2003.

[31] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[32] Dariu Gavrila. Pedestrian detection from a moving vehicle. In *Proceedings of European Conference on Computer Vision, Dublin, Ireland*, pages 37–49, June 2000.

[33] Camillo Gentile, Octavia I. Camps, and Mario Sznaier. Segmentation for robust tracking in the presence of severe occlusion. In *CVPR (2)*, pages 483–488, 2001.

[34] Marc Genton. Classes of kernels for machine learning : A statistics perspective. *Journal of Machine Learning Research*, 2 :299–312, 2001.

[35] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4) :377–388, 1996.

[36] Kristen Grauman and Trevor Darrell. Pyramid match kernels : Discriminative classification with sets of image features. Technical report, Massachusetts Institute of Technology, march 2005.

[37] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels : Hardness results and efficient alternatives. *Lecture Notes in Artificial Intelligence*, 2777(1) :129–143, 2003.

[38] Vincent Guigue, Alain Rakotomamonjy, and Stéphane Canu. Classification de signaux invariante en translation. In *20e colloque GRETSI sur le traitement du signal et des images*, Louvain, 2005.

[39] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, August 2001.

[40] Paul Hayton, Bernhard Schölkopf, Lionel Tarassenko, and Paul Anuzis. Support vector novelty detection applied to jet engine vibration spectra. In *Neural Information Processing Systems*, pages 946–952, 2000.

[41] D. D. Hoffman and B. E. Flinchbaugh. The interpretation of biological motion. *Biological Cybernetics*, pages 195–204, 1982.

[42] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[43] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multi-class support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

[44] Daniel P. Huttenlocher, Gregory A. Klanderman, and William Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9) :850–863, 1993.

[45] Sergey Ioffe and David Forsyth. Probabilistic methods for finding people. *Int. J. Comput. Vision*, 43(1) : 45–68, 2001.

[46] Hisashi Kashima and Yuta Tsuboi. Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In *Proceedings of* 21^{th} *International Conference on Machine Learning*, 2004.

[47] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieh International Conference on Machine Learning*, 2003.

[48] Yakov Keselman, Ali Shokoufandeh, Fatih Demirci, and Sven Dickinson. Many-to-many feature matching using spherical coding of directed graphs. In *Proceedings, 8th European Conference on Computer Vision, Prague, Czech Republic*, pages 322–335, May 2004.

[49] George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33 :82–95, 1971.

[50] Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *Proc. of ICML-2003, Washington DC.*, 2003.

[51] Kurt Konolige. Small vision system : Hardware and implementation. In *Eighth International Symposium on Robotics Research, Japan.*, 1997. URL `http://www.ai.sri.com/konolige/svs/Papers`.

[52] Raphael Labayrade, Didier Aubert, and Jean-Philippe Tarel. Real time obstacle detection on non flat road geometry through 'v-disparity' representation. In *Proceedings of IEEE Intelligent Vehicle Symposium*, pages 646–651, Versailles, France, June 2002.

[53] Christian Lantuejoul and Serge Beucher. On the use of the geodesic metric in image analysis. *Journal of miscrocopy*, 121(1) :39–49, 1981.

[54] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society.

[55] Vincent Lemonde and Michel Devy. Détection d'obstacles par stéréovision sur véhicules intelligents. In *Congrès des jeunes chercheurs en vision par ordinateur, ORASIS' 05, Fournol, France*, 2005.

[56] Huma Lodhi, John Shawe-Taylor, Nello Cristianini, and Christopher Watkins. Text classification using string kernels. In *Neural Information Processing Systems*, pages 563–569, 2000.

[57] Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. Technical report, INSA de Rouen, november 2005.

[58] Gaëlle Loosli, Stéphane Canu, S.V.N. Vishwanathan, Alexander J. Smola, and Monojit Chattopadhyay. Une boîte à outils rapide et simple pour les svm. *CAp 2004 - Conférence d'Apprentissage*, pages 113–128, 2004.

[59] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[60] Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. Extensions of marginalized graph kernels. In R. Greiner and D. Schuurmans, editors, *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, pages 552–559. ACM Press, 2004.

[61] Mirko Mählisch, Matthias Oberländer, Otto Löhlein, Dariu Gavrila, and Werner Ritter. A multiple detector approach to low-resolution fir pedestrian recognition. In *IEEE Intelligent Vehicles Symposium, Las Vegas, Nevada*, pages 325–330, June 2005.

[62] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10) :1615–1630, 2005.

[63] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2) :43–72, 2005.

[64] Eric Nowak, Frederic Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*. Springer, 2006.

[65] Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander Smola. Learning with non-positive kernels. In *Proceedings of the 21st International Conference on Machine Learning*, pages 639–646, 2004.

[66] Cheng Soon Ong, Alexander J. Smola, and Robert C.Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6 :1043–1071, 2005.

[67] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 193, Washington, DC, USA, 1997. IEEE Computer Society.

[68] Constantine Papageorgiou and Tomaso Poggio. Trainable pedestrian detection. In *Proceedings of the 1999 International Conference on Image Processing*, pages 35–39, 1999.

[69] Sylvie Philipp-Foliguet and Mohamed Lekkat. Recherche d'images à partir d'une requête partielle utilisant la disposition des régions. In *Actes du colloque RFIA'04*, Toulouse, January 2004.

[70] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. In *Proceedings of the IEEE*, volume 78-9, pages 1481–1497, 1990.

[71] Alain Rakotomamonjy and Frédéric Suard. Sélection de variables par svm : application à la détection de

piétons. In *RFIA04*, 2004.

[72] Cecilia Di Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37(1) :21–31, 2004.

[73] Cecilia Di Ruberto and Giuseppe Rodriguez. Recognition of shapes by morphological attributed relational graphs, September 2002. Atti del VIII Convegno AIIA - Associazione Italiana Intelligenza Artificiale.

[74] Nozha Boujemaa Sabri Boughorbel and Constantin Vertan. Soft color signatures for image retrieval by content. In *Eusflat'2001*, page 2001, 394–401.

[75] Bertrand Le Saux and Horst Bunke. Combining svm and graph matching in a multiple classifier system for image content recognition, August 2006. Workshop on Statistical Pattern Recognition (S+SSPR'06) of the IAPR International Conference on Pattern Recognition (ICPR'06), Hong Kong, China.

[76] Bernt Schiele and James L. Crowley. Object recognition using multidimensional receptive field histograms. In *ECCV (1)*, pages 610–619, 1996.

[77] Jean Serra. Morphologie mathématique. *Traité d'Informatique Géologique*, 6(6) :194–238, 1972.

[78] Amnon Shashua, Yoram Gdalyahu, and Gaby Hayon. Pedestrian detection for driving assistance systems : Single-frame classification and system level performance. In *Proceedings of IEEE Intelligent Vehicles Symposium*, 2004.

[79] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, and Steven W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, pages 13–32, 1999.

[80] Markus Stricker and Michael Swain. The capacity of color histogram indexing. In *CVPR94*, pages 704–708, 1994.

[81] Frédéric Suard, Vincent Guigue, Alain Rakotomamonjy, and Abdelaziz Bensrhair. Pedestrian detection using stereo-vision and graph kernels. In *Intelligent Vehicles Symposium, Las Vegas, Nevada*, pages 267–272, June 2005.

[82] Frédéric Suard, Alain Rakotomamonjy, and Abdelaziz Bensrhair. Détection de piétons par stéréovision et noyaux de graphes. In *GRETSI05, Louvain-la-Neuve, Belgique*, pages 686–686, 2005.

[83] Frédéric Suard, Alain Rakotomamonjy, and Abdelaziz Bensrhair. Object categorization using kernels combining graphs and histogram of gradients. In *International Conference on Image Analysis and Recognition, Póvoa de Varzim, Portugal*, pages 23–34, September 2006.

[84] Frédéric Suard, Alain Rakotomamonjy, Abdelaziz Bensrhair, and Alberto Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *Intelligent Vehicles Symposium, Tokyo, Japan*, pages 206–212, June 2006.

[85] Máté Szarvas, Utsushi Sakai, and Jun Ogata. Real-time pedestrian detection using lidar and convolutional neural networks. In *Intelligent Vehicles Symposium, Tokyo, Japan*, pages 213–218, June 2006.

[86] Andrea Torsello. *Matching Hierarchical Structures for Shape Recognition*. PhD thesis, University of York, 2004.

[87] Gwenaëlle Toulminet. *Extraction des contours 3D des obstacles par stéréovision pour l'aide à la conduite automobile*. PhD thesis, Laboratoire PSI, INSA de Rouen, France, 2002.

[88] Koji Tsuda, Taishin Kin, and Kiyoshi Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(Suppl 1) :268–275, 2002.

[89] Runar Unnthorsson, Thomas Philip Runarsson, and Magnus Thor Jonsson. Model selection in one class nu-svms using rbf kernels. In *16th conference on Condition Monitoring and Diagnostic Engineering Management*, August 2003.

[90] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y, 1995.

[91] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Computer Vision and Pattern Recognition, Kauai, Hawaii*, 1 :511–518, 2001.

[92] Paul Viola, Michael Jones, and Daniel Snow. Pedetrian using patterns of motions and appearance. In *IEEE Int. Conf on Computer Vision*, pages 734–741, 2003.

[93] S. V. N. Vishwanathan and Alexander J. Smola. Fast kernels on strings and trees. In *Neural Information Processing Systems Conference*, volume 15, 2002.

[94] Christian Wallraven, Barbara Caputo, and Arnulf Graf. Recognition with local features : the kernel recipe. In *Ninth IEEE International Conference on Computer Vision (ICCV'03)*, volume 1, pages 257–265, 2003.

[95] Gérard Weisbuch. *Dynamique des Systèmes Complexes*. CNRS,InterEditions, 1989.

[96] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. In *Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel*, volume 1296, pages 456–463, Heidelberg, 1997. Springer-Verlag.

[97] Lior Wolf and Amnon Shashua. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4 :913–931, 2003.

[98] Fengliang Xu, Xia Liu, and Kikuo Fujimura. Pedestrian detection and tracking with night vision. *IEEE Transactions on Intelligent Transportation Systems*, 6–1 :63–71, march 2005.

[99] Liang Zhao. *Dressed Human Modeling, Detection, Detection, and Parts Localization*. PhD thesis, Robotic Institute, Canergie Mellon University, 2001.

[100] Liang Zhao and Charles Thorpe. Stereo and neural network based pedestrian detection. *IEEE Trans. on Intelligent Vehicles Transportation systems*, 1(3) :148–154, 2000.

Index

Résumé

La détection de piéton est un problème récurrent depuis de nombreuses années. La principale confrontation est liée à la grande variabilité du piéton en échelle, posture et apparence.

Un algorithme efficace de reconnaissance de formes doit donc être capable d'affronter ces difficultés. En particulier, le choix d'une représentation pertinente et discrimante est un sujet difficile à résoudre.

Dans notre cas, nous avons envisagé deux approches. La première consiste à représenter la forme d'un objet à l'aide de graphes étiquetés. Selon les étiquettes apportées, le graphe possède en effet des propriétés intéressantes pour résoudre les problèmes de variabilité de taille et de posture. Cette méthode nécessite cependant une segmentation rigoureuse au préalable.

Nous avons ensuite étudié une représentation constituée d'histogrammes locaux d'orientation de gradient. Cette méthode présente des résultats intéressants de par ses capacités de généralisation. L'application de cette méthode sur des images infrarouges complètes nécessite cependant une fonction permettant d'extraire des fenêtres dans l'image afin d'analyser leur contenu et vérifier ainsi la présence ou non de piétons.

La deuxième étape du processus de reconnaissance de formes concerne l'analyse de la représentation des données. Nous utilisons pour cela le classifieur *Support Vector Machine* bâti, entre autres, sur une fonction noyau calculant le produit scalaire entre les données support et la donnée évaluée.

Dans le cas des graphes, nous utilisons une formulation de noyau de graphes calculé sur des «sacs de chemins». Le but consiste à extraire un ensemble de chemins de chaque graphe puis de comparer les chemins entre eux et combiner les comparaisons pour obtenir le noyau final.

Pour analyser les histogrammes de gradient, nous avons étudié différentes formulations permettant d'obtenir les meilleures performances avec cette représentation qui peut être assimilée à une distribution de probabilités.

Abstract

A lot of research have been carried out in the field of pedestrian detection using images and computers. The main obstacle is related with the pedestrian himself, which could not be easily characterized, due to its high variability. In particular, we have a scale, pose and appearance variability.

To bring an issue to these variabilities, the pedestrian representation should be carefully chosen. In our case, we first use a graph based representation. In fact, a labeled graph has interesting properties to reduce particularly the scale and pose variability.

A second representation is based on histogramms of oriented gradient, which computes some local histogramms of an image. This method has a good generalization capacity against variability. To apply this method on infrared images in order to detect pedestrians, we have to design a function to extract and analyse some windows from this image.

The second part of the pattern recognition process is the classification. In our case we use the Support Vector Machine classifier, which is based on an particular function : the kernel function. The aim is to compute the inner product between data.

For the graph method, we have to design an inner product between graphs. The aim is to compare two graphs by using bag-of-paths. That is to say, we extract some paths from graphs and we compare paths between them. The final kernel is obtained by combining all comparisons between paths.

We also studied different kinds of kernel functions more specific for the histogramm-based representation in order to improve the performance of the classifier when we are using the HOG descriptor.