HAL open science

Détermination de la vitesse limite par fusion de données vision et cartographiques temps-réel embarquées

Anne-Sophie Puthon

To cite this version:

Anne-Sophie Puthon. Détermination de la vitesse limite par fusion de données vision et cartographiques temps-réel embarquées. Autre. Ecole Nationale Supérieure des Mines de Paris, 2013. Français. NNT : 2013ENMP0042 . pastel-00957392

HAL Id: pastel-00957392

https://pastel.hal.science/pastel-00957392v1

Submitted on 10 Mar 2014

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

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

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n°432 : Science des Métiers de l'Ingénieur

Doctorat ParisTech

THESE

pour obtenir le grade de docteur par

Mines ParisTech

Spécialité « Informatique, temps réel, robotique et automatique »

présentée et soutenue publiquement par

Anne-Sophie PUTHON

le 2 Avril 2013

Détermination de vitesse limite par fusion de données vision et cartographiques temps-réel embarquées

Speed limit determination by real-time embedded visual and cartograhical data fusion

Directeur de thèse : **Fawzi NASHASHIBI**
Co-encadrement de la thèse : **Fabien MOUTARDE**

Jury :

Mme Véronique BERGE-CHERFAOUI, Maître de Conférence, UTC	Rapporteur
M. Roger REYNAUD, Professeur, Université Paris-Sud	Rapporteur
M. Fabrice MERIAUDEAU, Professeur, Le2i	Président du jury
M. Jean-Philippe LAUFFENBURGER, Maître de Conférence, UHA	Examinateur
M. Fawzi NASHASHIBI, Directeur de recherche, INRIA Rocquencourt	Directeur de Thèse
M. Fabien MOUTARDE, Maître de Conférence, Mines ParisTech	Co-encadrant de Thèse
M. Benazouz BRADAI, Docteur, VALEO	Invité

THÈSE

MINES ParisTech
Centre de Robotique
60 bd Saint Michel, 75006 PARIS - France

Remerciements

Je tenais tout d'abord à remercier mon directeur de thèse, M. Fawzi Nashashibi, qui m'a prodigué ses conseils tout au long de ma thèse. Il a fait preuve d'une grande écoute et de compréhension lors de nos discussions portant sur mes travaux ou le projet Speedcam. Son soutien et son enthousiasme m'ont beaucoup aidée à surmonter les difficultés inhérentes à tout travail de thèse.

Je souhaiterais exprimer ma gratitude à mon encadrant, M. Fabien Moutarde. Toujours disponible et ouvert, il m'a permis d'aborder mon travail sous d'autres perspectives et d'éviter de perdre de vue mes objectifs.

Je remercie également tout les membres de mon jury qui ont accepté d'évaluer mon travail et notamment M. Fabrice Meriaudeau, président du jury, et mes rapporteurs, Mme Véronique Berge-Cherfaoui et M. Roger Reynaud. Leur remarques et questions m'ont été d'une grande aide pour améliorer mon rapport. Ils m'ont permis d'aborder mes travaux avec un œil neuf, nécessaire après trois années, et de concilier les points de vue académiques, industriels et utilisateur final.

Ma thèse s'est déroulée principalement dans le cadre du projet Speedcam et je souhaite adresser ma reconnaissance aux différents partenaires, notamment M. Benazouz Bradai et Philippe Gougeon, de l'équipe Valeo Driving Assistance. Leur expérience du domaine, leur expertise et leur niveau d'exigence ont donné une dimension industrielle à mon projet. Je n'oublie pas de mentionner M. Ulrich Kressel de Daimler qui m'a fourni des bases de données importantes pour évaluer mes travaux.

En dernier lieu, je souhaite vivement remercier tous mes collègues du CAOR pour tous les bons moments partagés pendant ma thèse. Une bonne ambiance et des personnes toujours prêtes à aider et faire partager leur connaissance sont des éléments indispensables à tout thésard.

Table des matières

1 Introduction **11**

2 Détection de panonceaux **15**
2.1 Motivations . . . 15
2.2 État de l'art . . . 17
2.2.1 *Template-matching* . . . 18
2.2.2 Utilisation de la transformée de Hough . . . 19
2.2.3 *Template-matching* dans l'espace de Hough . . . 20
2.3 Approche proposée . . . 21
2.3.1 Principe de la croissance de régions . . . 22
2.3.2 Sélection des graines contrastées . . . 24
2.3.3 Recherche de zones uniformes . . . 28
2.3.4 Choix des paramètres . . . 28
2.4 Réalisation d'un benchmark . . . 30
2.4.1 Approches frontière . . . 30
2.4.2 Approches par seuillage . . . 31
2.4.3 Approche par région . . . 33
2.5 Étude comparative . . . 36
2.5.1 Bases de données . . . 36
2.5.2 Mesures utilisées . . . 37
2.5.3 Analyse des résultats . . . 38
Mesure de Jaccard . . . 38
Centrage . . . 39
Recouvrement . . . 39
Étude des faux positifs . . . 40
Robustesse aux dégradations d'image . . . 41
2.6 Conclusion . . . 42

3 Reconnaissance de panonceaux **45**
3.1 Introduction . . . 45
3.2 Classification . . . 45
3.2.1 Qu'est-ce que la classification ? . . . 45
Approche syntaxique . . . 46
Template-matching . . . 47
Réseaux de neurones . . . 47
Autres techniques d'apprentissage statistique . . . 50
3.2.2 Comment représenter les motifs ? . . . 53

Approches locales . . . 54
Approches globales . . . 56
3.2.3 Domaine des panonceaux . . . 58
3.3 Méthode proposée . . . 59
3.3.1 Analyse du problème . . . 59
3.3.2 Définition de macro-catégories . . . 60
3.3.3 Utilisation de descripteurs globaux . . . 61
3.4 Évaluation . . . 63
3.4.1 Base de données . . . 63
3.4.2 Performances . . . 65
Justification du paramétrage utilisé pour le PHOG . . . 65
Justification de l'architecture . . . 66
Analyse des résultats . . . 69
3.5 Conclusion . . . 71

4 Fusion de données **73**
4.1 Introduction . . . 73
4.2 Théorie sur la fusion de données . . . 73
4.2.1 Principes généraux . . . 73
Définitions . . . 73
Imperfections des données . . . 74
Modèles . . . 75
Architectures . . . 75
4.2.2 Théorie des probabilités . . . 78
4.2.3 Théorie des possibilités . . . 79
Généralités . . . 79
Application à la fusion . . . 80
4.2.4 Théorie de Dempster-Shafer . . . 81
Généralités . . . 81
Combinaison . . . 82
4.2.5 Affaiblissement - Renforcement . . . 83
4.2.6 Décision . . . 83
4.3 État de l'art de la fusion de données dans les systèmes ISA . . . 83
4.3.1 Contexte bayésien . . . 84
4.3.2 Prise en compte du contexte de situation . . . 86
4.3.3 Critères liés au GPS . . . 88
4.4 Méthode proposée . . . 90
4.4.1 Estimation de la fiabilité de la navigation . . . 92
4.4.2 Estimation des fonctions de masse . . . 93
4.4.3 Détermination de la vitesse limite . . . 98
4.5 Évaluation . . . 99
4.5.1 Bases de données . . . 99
4.5.2 Comparaison avec la méthode de [Lauffenburger et al., 2008] . . . 99
4.5.3 Importance de la fusion de données . . . 101
4.6 Conclusion . . . 102

5 Système complet **107**
5.1 Introduction . . . 107
5.2 Présentation du système complet . . . 107
5.3 Intégration des panonceaux . . . 108
5.3.1 Comparaison de détecteurs . . . 108
5.3.2 Étude de la taille . . . 111
5.3.3 Suivi spatio-temporel . . . 112
5.3.4 Analyse fonctionnelle . . . 112
5.3.5 Performances temporelles . . . 114
5.4 Intégration des marquages . . . 115
5.4.1 Intérêt . . . 115
5.4.2 Mise en œuvre . . . 116
Machine d'état . . . 116
Files de panneaux . . . 117
5.5 Évaluation . . . 119
5.5.1 Implémentation réalisée . . . 120
5.5.2 Performances générales . . . 121
5.5.3 Apport de la gestion des panonceaux et marquages . . . 122
5.5.4 Cas d'une vision parfaite . . . 122
5.6 Conclusion . . . 122

6 Conclusion **127**

A Publications **129**

Bibliographie **131**

Liste des Abréviations

ADAS *Advanced Driver Assistance System*, Systèmes d'Aide à la Conduite

DFG *Data Fusion Group*, Groupe de Fusion de Données

FC *Functional Class*, Classe Fonctionnelle

FPPP Faux Positifs Par Positif

GT *Ground Truth*, Vérité Terrain

HDOP *Horizontal Dilution of Precision*, Coefficient d'Affaiblissement de la Précision Horizontale

HOG *Histogram of Oriented Gradients*, Histogramme de Gradients Orientés

ISA *Intelligent Speed Adaptation*, Adaptation Intelligente de la Vitesse

JDL *Joint of Directors of Laboratories*, Comité de Directeurs de Laboratoires

MLCP *Most Likely Candidate Probability*, Probabilité du Candidat le Plus Vraisemblable

OCR *Optical Character Recognition*, Reconnaissance Optique de Caractères

PHOG *Pyramid of Histograms of Oriented Gradients*, Pyramide d'Histogrammes de Gradients Orientés

RBF *Radial Basis Function*, Fonction de Base Radiale

SIFT *Scale Invariant Feature Transform*, Transformation de Caractéristiques Visuelles Invariantes à l'Échelle

SURF *Speeded-Up Robust Feature*, Caractéristiques Robustes Accélérées

SVM *Support Vector Machine*, Machines à Vecteurs de Support

SVM-BTA *Support Vector Machine with Binary-Tree Architecture*, Machines à Vecteurs de Support avec Architecture Hiérarchisée

TSR *Traffic Sign Recognition*, Reconnaissance de Panneaux Routiers

Chapitre 1

Introduction

D'après l'ONISR (Observatoire National Interministériel de la Sécurité Routière), la vitesse (inadaptée ou excessive) serait une des causes principales de plus de 40% des accidents mortels. La capacité de perception visuelle diminue à mesure que la vitesse augmente, en même temps que la distance de freinage en cas de danger. Ainsi, un piéton dispose d'un champ visuel de 180°, un automobiliste roulant à 40 km/h, de 100° et à 130 km/h, l'angle d'ouverture n'est plus que de 30°. Une vision et une marge de réaction limitées rendent le traitement et l'analyse des informations plus complexes et peuvent être à l'origine de situations dangereuses. De plus, la gravité de l'accident dépend de l'énergie du choc et donc de la vitesse des véhicules. En 2011 [1], 3963 personnes ont trouvé la mort dans un accident de la route. Outre l'aspect humain, le coût économique représenté est considérable tant pour la société que pour les conducteurs [2].

L'intérêt de proposer un système de régulation automatique de la vitesse est alors évident. Le concept est déjà à l'œuvre dans de nombreux véhicules récents grâce à l'utilisation du régulateur de vitesse. Ces systèmes soulagent le conducteur qui peut ainsi reporter son attention sur la route plutôt que sur son compteur mais doivent être activés et mis à jour manuellement. Entièrement automatiques, ils amélioreraient notablement le confort des utilisateurs tout en garantissant une certaine anticipation des modifications de vitesse de croisière, évitant ainsi les à-coups et réduisant la consommation.

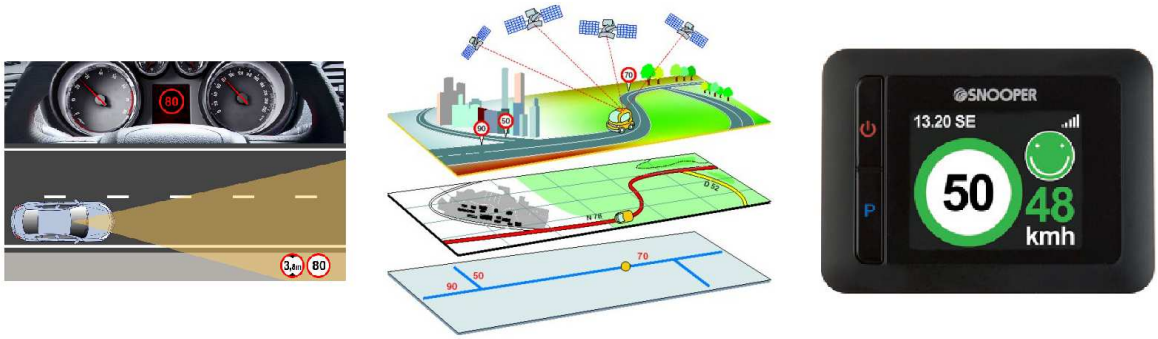
Le nombre croissant de projets liés à cette thématique témoigne bien de l'intérêt grandissant pour ce domaine, tant dans la recherche académique que dans le monde industriel [3]. Depuis quelques années, des véhicules sont même équipés de modules de détermination de vitesse limite de plus en plus sophistiqués (figure 1.1). La majorité repose sur une caméra monoculaire qui détecte et reconnaît les panneaux présents sur les bords des routes. Ces derniers correspondent à des vitesses limites **locales** ou temporaires, dénotant la présence de zones dangereuses. Toutefois, de tels systèmes n'ont aucune connaissance des limitations **implicites** liées au contexte de conduite, comme le type de route ou la présence en ville ou non. De plus, leurs performances sont sensiblement réduites lorsque les conditions climatiques se dégradent.

1. wwww.securite-routiere.gouv.fr/la-securite-routiere/l-observatoire-national-interministeriel-de-la-securite-routiere

2. En 2011, plus de 10 millions de contraventions pour excès de vitesse ont été émises, pour une recette totale de 539 millions d'euros. La majorité concerne des infractions de moins de 20 km/h en ville, pénalisées par une amende de 135 euros et un retrait d'un point de permis

3. Le projet LAVIA en France [Ehrlich et al., 2003] l'étude de [Jamson, 2006] au Royaume-Uni ou celle de [Vlassenroot et al., 2007] en Belgique ont permis d'évaluer l'impact qu'auraient des systèmes automatiques de détermination de vitesse limite sur la conduite et leur acceptation par les utilisateurs.

Une autre approche consiste à se baser uniquement sur un capteur de navigation qui associe à chaque position du véhicule une vitesse limite stockée dans une base de données statique. De nombreux projets, dont le projet LAVIA (Limiteur s'Adaptant à la VItesse Autorisée) [Ehrlich et al., 2003], ont été menés afin d'étudier la faisabilité, l'acceptation et l'utilité de tels systèmes. Des dispositifs commerciaux existent même, comme le Snooper MySpeed [4]. Les attraits d'applications basées sur la navigation sont indéniables : simplicité de mise en œuvre, réseau entièrement renseigné sur la vitesse limite, absence de modifications nécessaires de l'infrastructure. Néanmoins, il est nécessaire, pour leur bon fonctionnement, que l'ensemble du réseau soit maintenu à jour précisément et que la couverture satellitaire soit toujours suffisante pour tous les véhicules. Les canyons urbains, les tunnels ou même les nuages, sources de mauvaise réception des signaux, risquent alors de demeurer des zones d'ombre pour ces systèmes. La combinaison des deux capteurs apparaît naturellement comme la solution la plus adaptée pour obtenir fiabilité et robustesse.



(a) Système Opel Eye (b) Projet LAVIA (c) Dispositif MySpeed

FIGURE 1.1 – Quelques exemples de systèmes de détermination de vitesse limite. (a) Le système Opel Eye vise à détecter et reconnaître, au moyen d'une caméra, les panneaux de limitation de vitesse. (b) L'étude menée dans le cadre du projet LAVIA (source : [Ehrlich et al., 2003]) repose sur un capteur de navigation et une base de données précisément renseignée dans une zone géographique restreinte. À une position donnée correspondent un segment de route et une vitesse limite. (c) L'appareil MySpeed fonctionne selon le même principe que le projet LAVIA. Le véhicule qui en est équipé est ainsi informé en temps réel de la limitation actuelle grâce à une antenne GPS (*Global Positioning System*) intégrée.

Notre travail est consacré à la fusion de données des deux types de systèmes afin de pallier à leurs défaillances individuelles tant du point de vue conceptuel que pratique. En effet, de par sa nature, la navigation ne peut pas lire les panneaux présents au bord de la route et la vision n'a aucune connaissance du contexte de conduite et des limitations associées. De plus, la signalisation routière ne repose pas entièrement sur les panneaux mais est souvent complétée par des panonceaux et s'applique sur des voies spécifiques, délimitées par des marquages. Ces trois informations sont indissociables pour traiter correctement les limitations locales. Les objectifs de cette thèse sont donc multiples : améliorer de façon individuelle les performances des modules vision et cartographique et réaliser la meilleure combinaison possible. Le projet Speedcam [5], dans le cadre duquel elle s'est déroulée, apporte une dimension industrielle supplémentaire et des préoccupations plus applicatives.

4. http ://www.snooper.fr/MySpeed.Snooper.My-Speed.html
5. http ://www.systematic-paris-region.org/fr/projets/speedcam

Le plan de ce mémoire est divisé en deux parties : traitement d'images en vue de détecter et reconnaître les panonceaux (chapitres 2 et 3) et implémentation d'un système complet basé sur la fusion de données cartographiques et vision (chapitres 4 et 5). Dans le chapitre 2, nous présentons une approche inédite de détection de panonceaux en deux phases, la recherche de régions contrastées à l'aide d'un algorithme de reconstruction morphologique, puis l'extraction du panonceau par croissance de régions. Nous comparons notre méthode à trois autres techniques issues de l'état de l'art et mettant en œuvre contours, colorimétrie et régions. L'étape de reconnaissance est ensuite détaillée dans le chapitre 3. Nous définissons pour cela plusieurs macro-catégories de panonceaux classifiées à l'aide de descripteurs globaux, les PHOG (*Pyramid of Histograms of Oriented Gradients* - Pyramides d'Histogrammes de Gradients Orientés), et d'un arbre de SVMs (*Support Vector Machines* - Séparateurs à Vaste Marge). Nous évaluons ensuite le module complet de gestion de panonceaux. Dans un second temps, nous étudions la fusion de données entre la navigation et la vision en améliorant tout d'abord la gestion des informations provenant du capteur de navigation (chapitre 4). Nous comparons les résultats obtenus avec une méthode similaire de la littérature. Enfin, le système complet de détermination de vitesse limite développé est implémenté et évalué dans le chapitre 5 avec notamment l'ajout des marquages.

Chapitre 2

Détection de panonceaux

2.1 Motivations

FIGURE 2.1 – Illustrations de situations pour lesquelles la vitesse limite dépend du contexte spécifié par le panonceau. À gauche : La vitesse limite de 70 km/h ne s'applique qu'aux véhicules de plus de 3.5 tonnes tandis que le panneau 90 s'applique à toutes les autres catégories de véhicules. L'information "Rappel" n'apporte aucune information supplémentaire. À droite : La vitesse limite de 70 km/h ne concerne que la voie de sortie située à droite, comme spécifié par le panonceau "Flèche". Les véhicules qui circulent sur la route principale ne sont pas affectés.

De nombreuses recherches ont été réalisées pour la détection de panneaux routiers dans le but d'assister le conducteur et de l'alerter en cas de situations dangereuses. Malheureusement, peu d'entre elles se sont intéressées au cas des panonceaux, ces rectangles situés sous un panneau et dont ils précisent la signification. Ils apparaissent en diverses occasions, sous de nombreuses formes et leur interprétation est essentielle pour implémenter un système conforme aux règles de conduite (figure 2.1). Il semble en effet inutile d'informer le conducteur d'une restriction concernant une autre voie que celle où il circule ou une autre catégorie de véhicule. Nous nous sommes plus particulièrement intéressés à la signalisation relative aux limitations de vitesse. Imaginons un véhicule roulant sur autoroute et passant à proximité d'une voie de sortie, sans l'emprunter. Des panneaux, combinés la plupart du temps à des panonceaux "Flèche", sont présents pour indiquer que la vitesse y est réduite. Si le véhicule était pourvu d'un système de détermination de vitesse basée sur la vision mais **sans** prise en compte des panonceaux, une alerte pour réduire sa vitesse risquerait d'être déclenchée, chaque fois que ce type de situation serait rencontré. Cela entraînerait la distraction, voire l'agacement, du conducteur.

D'après [Instruction Interministérielle Relative à la Signalisation de Direction, 1982], contrairement aux panneaux, les dimensions des panonceaux ne sont pas standardisées mais dépendent plutôt de l'information portée. La variété des directives est très grande et elles peuvent être divisées en différents types :

- les **flèches** qui spécifient la voie sur laquelle s'applique la limitation de vitesse. Elles se rencontrent en majorité en sortie d'autoroute ;
- les **pictogrammes** qui précisent la catégorie de véhicules concernés (camions, bus, caravanes, motos, etc.) ;
- le **texte** qui permet l'affichage d'une quantité infinie d'informations, notamment des tonnages, distances, étendues, restrictions en fonction de conditions climatiques ou de plages horaires ;
- les panonceaux **mixtes**, combinant du texte et une autre catégorie, comme un pictogramme.

Cette catégorisation est valable dans les deux pays pour lesquels nous avons développé le système, à savoir la France et l'Allemagne. Quelques exemples de panonceaux français et allemands sont présentés dans le tableau 2.1.

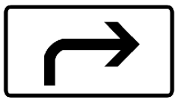
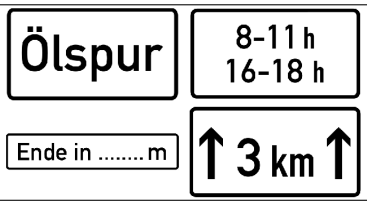
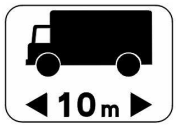
Type de panonceau	France	Allemagne
Flèche		
Pictogramme		
Texte		
Mixte		

TABLE 2.1 – Exemples de panonceaux français et allemands. Nous les avons divisés en quatre grandes catégories en fonction du type d'information portée. Les flèches spécifient la voie sur laquelle s'applique la limitation, les pictogrammes indiquent la catégorie de véhicules concernée tandis que les instructions plus complexes sont fournies soit sous forme de texte soit grâce à une combinaison des types précédents.

L'approche que nous avons privilégiée est à l'image de la plupart des applications de type TSR (*Traffic Sign Recognition* - Reconnaissance de Panneaux Routiers) qui procèdent en trois étapes. Tout d'abord, une phase de **détection** permet de délimiter dans l'image les régions d'intérêt où se trouvent potentiellement les objets recherchés. Dans notre cas, cette phase consiste en la détection de rectangles que nous détaillerons dans la suite de ce chapitre. Les objectifs sont multiples, notamment réduction du temps de calcul et du nombre de fausses alarmes, simplification et modularité du processus. Ensuite, la **reconnaissance** (chapitre 3) vise à séparer les bons des mauvais candidats,

la plupart du temps grâce à un apprentissage préalable, et à retourner le type d'objet détecté. À l'issue de cette étape, le système fournit le nombre de panonceaux présents dans l'image, le plus souvent associés à une mesure de confiance, ainsi que leur message. Enfin, le **suivi temporel** est utilisé pour valider une hypothèse si un même objet a été correctement détecté un nombre déterminé d'images. Combiné à un modèle de caméra, il peut également permettre d'introduire un *a priori* sur la position d'un objet dans l'image suivante. La figure 2.2 illustre le processus complet.

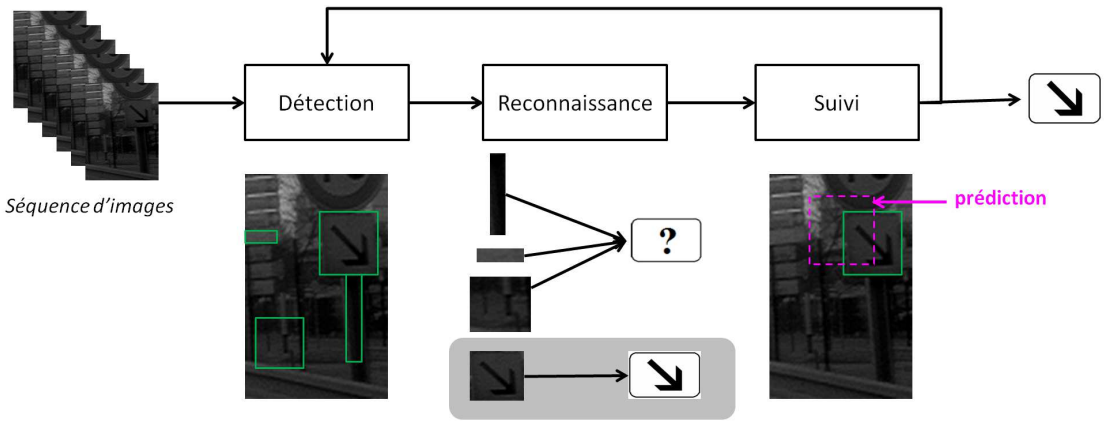


FIGURE 2.2 – Schéma général d'un système de reconnaissance de panonceaux. La détection recherche dans les images fournies par la caméra les zones où se trouvent potentiellement les panonceaux (à l'aide d'informations de couleur, de forme, etc.). Un algorithme de reconnaissance permet ensuite d'éliminer les fausses détections et de déterminer précisément le type de panonceau. Enfin, le suivi temporel vise à estimer la position de l'objet dans l'image suivante pour réduire le temps de calcul et augmenter la confiance de la prochaine détection du même objet.

Dans ce chapitre, nous présentons la première partie du système de reconnaissance de panonceaux, la détection des régions d'intérêt. Nous commençons par établir l'état de l'art, assez restreint, dans ce domaine puis nous décrivons notre approche inédite dont le cœur est un algorithme de croissance de régions basé sur une sélection de graines contrastées. Ensuite, nous détaillons les différents algorithmes de segmentation d'image que nous avons implémentés dans le but de réaliser une étude comparative des performances. Ce *benchmark* exhaustif indique que notre méthode fournit des résultats prometteurs.

2.2 État de l'art

Contrairement à la détection de panneaux qui requiert l'image complète, la recherche de panonceaux se limite à une zone située sous ces derniers. La petite taille de la région de recherche rend tentante l'idée de se dispenser de détecter des rectangles afin de procéder directement à la reconnaissance, via des fenêtres glissantes et une approche multi-échelle. Ces approches ont été notamment implémentées pour la détection de plaques d'immatriculation [Huang et al., 2008] ou de manière plus générale pour la reconnaissance d'objets très divers [Bosch et al., 2007a, Lampert et al., 2008, Tuytelaars and Mikolajczyk, 2007]. Néanmoins, quelle perte de temps de chercher un objet alors qu'il n'y a rien ! De plus, la grande diversité de panonceaux, en terme de dimensions, de ratios hauteur/largeur et de types, rend complexe la tâche de classification directe. Ce constat explique pourquoi l'ensemble des techniques de reconnaissance de panonceaux comportent une phase de détection [Hamdoun et al., 2008, Nienhüser et al., 2010, Liu et al., 2011] que nous pouvons

séparer en trois familles : *template-matching*, transformée de Hough et *template-matching* dans l'espace de Hough.

2.2.1 *Template-matching*

Actuellement, le peu de recherches réalisées dans le domaine de détection de rectangles fonctionnent selon le même principe, segmenter l'image à partir des contours. La façon la plus naturelle de procéder consiste à extraire les segments de l'image puis à réaliser un *template-matching*. La détection de rectangles de [Miura et al., 2000] repose simplement sur des projections de l'image de contours selon deux directions orthogonales. Un contour vertical correspondra ainsi à un pic dans l'histogramme de projection horizontale (figure 2.3). Cette technique, très simple à mettre en œuvre, est toutefois très sensible au bruit, aux occlusions ainsi qu'à une inclinaison éventuelle du panonceau par rapport à l'horizontale. L'utilisation des projections ne garantit pas non plus que le segment retourné soit continu.

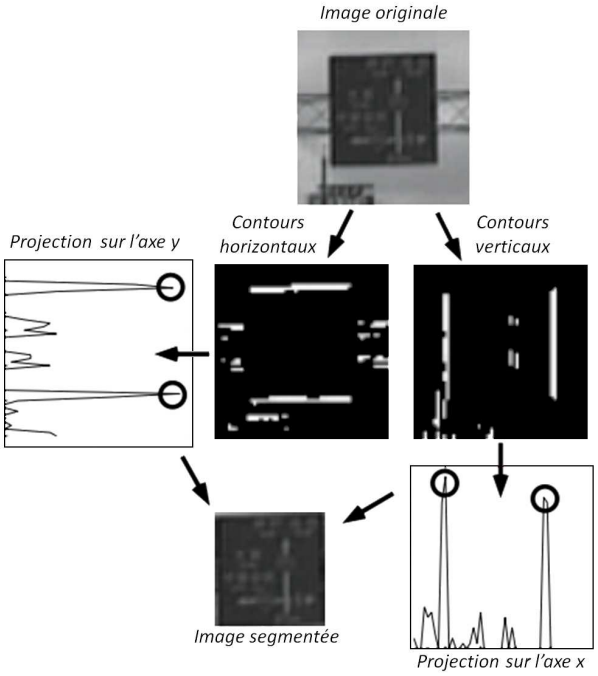


FIGURE 2.3 – Détection de rectangles à l'aide de projections horizontales et verticales de segments (source : [Miura et al., 2000]). La région de l'image située entre deux pics dans chaque histogramme de projections est segmentée et correspond à un panneau.

[Gavrila, 1999] implémente une technique plus évoluée de *template-matching*, la Transformée de Distance. Le principe est illustré par la figure 2.4. En premier lieu, un patron de primitives T est généré dans lequel les pixels blancs dénotent la présence d'un contour. Ce gabarit correspond à la forme recherchée. Les primitives sont extraites de l'image originale S dans une image binaire I. T est alors modifiée (rotation, mise à l'échelle, translation) jusqu'à obtenir une erreur de recouvrement minimum par rapport à I. La mesure de similarité D entre les deux images correspond à la moyenne

des distances de chaque primitive de T à la plus proche de I.

$$D(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t) \tag{2.1}$$

où $|T|$ correspond au nombre de primitives de T et $d_I(t)$, la distance entre une primitive t de T et la plus proche de I. Un modèle est apparié si la mesure est inférieure à un certain seuil τ.

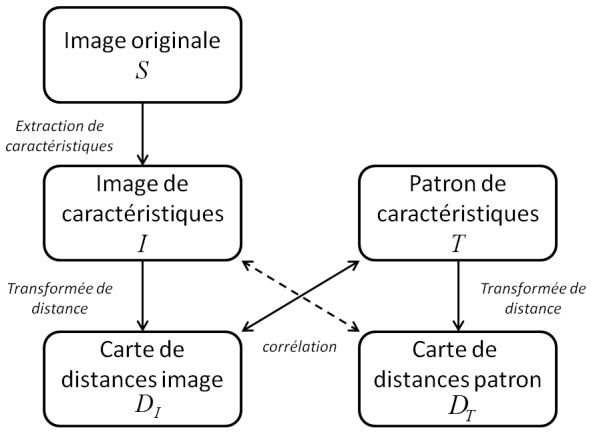


FIGURE 2.4 – Algorithme de la Transformée de Distance (source : [Gavrila, 1999]). La forme recherchée est déterminée par un ensemble de primitives, contours ou coins, stocké dans un patron T. Les mêmes indices visuels sont extraits de l'image originale S dans I qui est ensuite balayée par des fenêtres de tailles différentes. À chaque itération, l'image patron subit une transformation géométrique pour coller au mieux à la région de recherche de l'image I. Si la distance entre les deux ensembles de primitives est inférieure à un seuil, cela signifie qu'un objet de la forme recherchée est présent.

2.2.2 Utilisation de la transformée de Hough

Une autre méthode se base sur une généralisation de la transformée de Hough pour détecter les polygones réguliers à n côtés. À partir de l'image de gradients, [Loy and Barnes, 2004] proposent de déterminer le type de polygone ainsi que la position de son centre et son rayon. Le premier est obtenu grâce à l'orientation des gradients qui, le long du contour d'un polygone régulier, sont tous multiples d'une même valeur, $\frac{2\pi}{n}$ (figure 2.5).

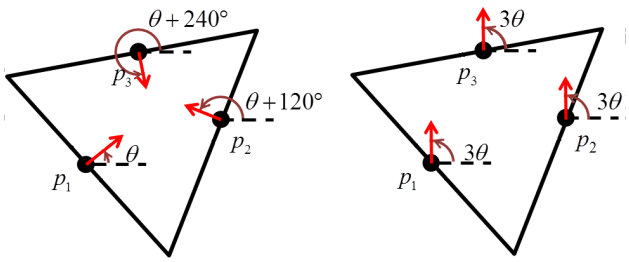


FIGURE 2.5 – Détermination du nombre de côtés d'un polygone (source : [Loy and Barnes, 2004]). Les orientations d'un polygone à n côtés (ici $n = 3$) sont toutes égales modulo $2\pi/n$.

Centre et rayon sont obtenus à l'aide d'une matrice d'accumulation (figure 2.6). Un pixel de contour p vote en effet pour un ensemble de centres potentiels selon une ligne orthogonale à la

direction de son gradient $g_n(p)$ et pour plusieurs valeurs de rayons r. La longueur de cette ligne de vote est liée au rayon et au type de polygone. Les pixels situés sur cette ligne sont impactés différemment suivant leur distance au point considéré, positivement au centre et négativement sur les extrémités. Cette méthode a été reprise par [Keller et al., 2008] pour la détection de panneaux américains. Pour améliorer la précision du système, ils proposent de procéder en plusieurs itérations pour déterminer l'orientation ainsi que la taille des rectangles. Après avoir éliminé les faux positifs grâce à un algorithme Adaboost entraîné sur une base de données de négatifs, ils recommencent l'opération de détection pour tous les candidats restants dans une fenêtre plus petite.
[Liu et al., 2011] ont eux adapté cette technique à la détection de panonceaux chinois avec une phase de post-traitement pour déterminer l'angle de rotation.

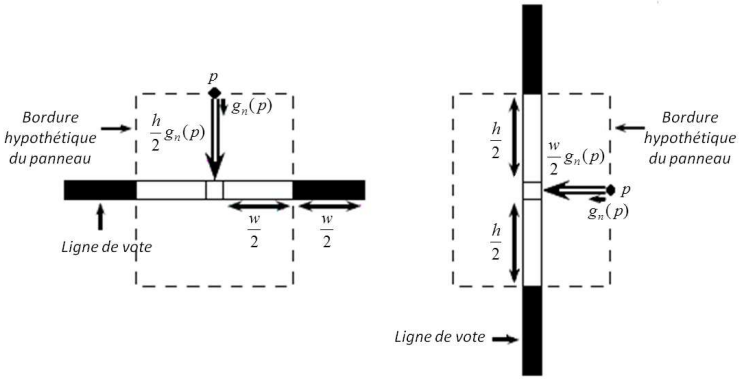


FIGURE 2.6 – Ligne de vote utilisée pour la détection de polygones réguliers (source : [Keller et al., 2008]). Chaque pixel de contour p ajoute son vote selon une ligne orthogonale à la direction de son gradient $g_n(p)$. Les pixels recevant un vote positif (resp. négatif) apparaissent en blanc (resp. noir). La longueur de la ligne ($2w + 1$ si horizontale et $2h + 1$ si verticale) est ici directement relative aux dimensions du rectangle recherché.

2.2.3 *Template-matching* dans l'espace de Hough

Une troisième manière d'aborder la détection de rectangles est celle présentée par [Nienhüser et al., 2010], inspirée de [Jung and Schramm, 2004]. L'idée est de réaliser un *template-matching* directement dans l'espace de Hough en exploitant les différentes relations géométriques vérifiées par les contours d'un rectangle. Cela revient à détecter un motif particulier constitué de quatre points, correspondant aux côtés du rectangle, répartis comme sur la figure 2.7, et tels que :

- les paires de segments parallèles génèrent deux points de même orientation θ et de même intensité ;
- les points sont symétriques deux à deux par rapport à un axe horizontal. Cette condition implique que le centre du rectangle recherché soit à l'origine du repère ;
- les deux paires de segments sont orthogonales, d'où une différence de 90° entre les deux paires de points.

L'inconvénient majeur de cette méthode est qu'elle est très facilement perturbée par n'importe quel contour n'appartenant pas au rectangle. Cela nécessite de délimiter une fenêtre de recherche relativement précise autour du panonceau (figure 2.8). Il faut donc être très attentif à la première étape de détection de contours.

Les inconvénients majeurs des techniques de détection basées sur les contours sont la sensibilité au bruit et aux occlusions. En effet, en présence de bruit, les contours, qui sont par définition des motifs de hautes fréquences, seront fortement perturbés. On peut notamment citer le bruit dû au mouvement du véhicule ou le flou généré par des conditions climatiques dégradées. Les méthodes présentées ne sont pas robustes aux occlusions, *i.e.* ne prennent pas en compte les cas où seulement trois côtés du rectangle seraient visibles. Quant à l'utilisation de gabarits, les tailles et ratios des panonceaux très variés la rendent difficile. En outre, il n'y a à ce jour aucune base de données publique qui permettrait de comparer les différents algorithmes et chaque équipe propose sa propre méthode d'évaluation sans fournir de détails sur les mesures ou les bases utilisées. Nous nous sommes alors demandés si d'autres approches de segmentation ne pourraient pas s'appliquer à ce domaine.

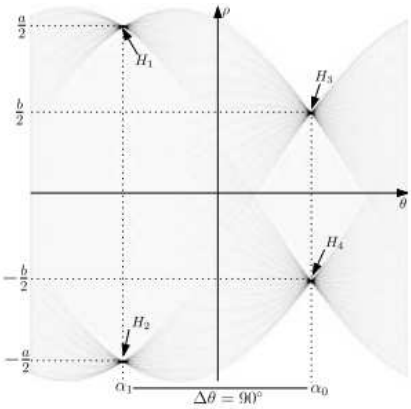
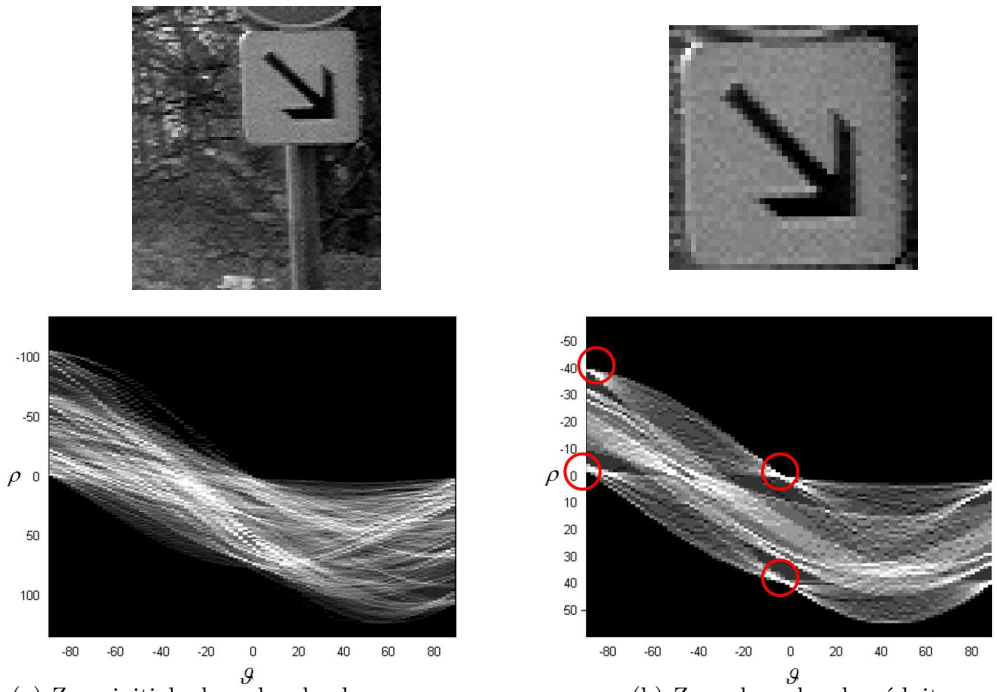


FIGURE 2.7 – Transformée de Hough obtenue pour un rectangle centré à l'origine (source : [Jung and Schramm, 2004]). Les couples de points avec la même valeur de θ, (H_1, H_2) et (H_3, H_4), représentent les côtés parallèles du rectangle. De plus, l'intensité de deux points du même couple dans la matrice d'accumulation est égale puisqu'un nombre égal de points y a contribué. La différence entre les deux couples $\Delta\theta = 90°$ témoigne de leur orthogonalité.

2.3 Approche proposée

Après avoir étudié les différentes techniques existantes, nous avons décidé de développer une approche inédite basée sur la croissance de régions. Ce choix est motivé par plusieurs constats :

- La région de recherche est de taille limitée car située au-dessous d'un panneau de limitation de vitesse préalablement détecté. Cela nous permet d'utiliser cette approche, pourtant assez gourmande en ressources.
- Un panonceau correspond à un rectangle de couleur homogène sur lequel est inscrit un message qui doit être suffisamment contrasté par rapport au fond pour être lisible par le conducteur. Une technique de croissance de régions représente une bonne alternative aux approches basées contour en recherchant des zones uniformes autour de pixels fortement contrastés. En effet, sans message lisible, un panonceau n'a pas d'intérêt.
- Contrairement aux approches type décomposition/fusion, la croissance de régions ne nous oblige pas à segmenter toute l'image, inutile lorsque l'objet n'est pas présent dans l'image. De plus, il est possible de ne faire croître qu'une région à la fois.



(a) Zone initiale de recherche de panonceaux. (b) Zone de recherche réduite.

FIGURE 2.8 – (a) Transformée de Hough obtenue pour la région initiale de recherche de panonceaux. L'image contient beaucoup plus d'informations que nécessaire conduisant à une transformée de Hough bruitée. Extraire les lignes relatives au panonceau se révèle difficile voire impossible. (b) Appliquer la transformée de Hough sur une région très réduite centrée sur le panonceau fait plus nettement apparaître le motif présenté par [Jung and Schramm, 2004]. Les quatre cercles rouges de la figure correspondent aux pics de la matrice d'accumulation liés au rectangle.

2.3.1 Principe de la croissance de régions

La croissance de régions est une approche ascendante introduite par [Zucker, 1976]. Il s'agit de faire croître des régions initiales R_0, les graines (aussi appelées germes), en ajoutant itérativement les pixels du voisinage qui répondent à certains critères d'homogénéité. La première étape critique est donc la sélection des graines, qui doivent nécessairement appartenir aux régions à segmenter. Leurs caractéristiques doivent également être les plus proches possible de celles des objets recherchés. Il peut s'agir de pixels seuls aussi bien que de régions connexes. Ensuite, les différents critères de croissance doivent être soigneusement choisis. Pour chaque région R_n, les pixels candidats C_n à l'agglomération doivent répondre à certaines conditions, notamment d'adjacence et d'homogénéité locale. Ces dernières permettent respectivement de générer des régions connexes et de limiter le nombre de postulants à chaque itération. Sont alors ajoutés à la région R_{n+1} ceux qui vérifient un critère d'homogénéité globale.

$$R_{n+1} = R_n + C_n \tag{2.2}$$

Enfin, la convergence est atteinte dès qu'il n'est plus possible d'ajouter aucun pixel à la région.

La croissance de régions est une méthode très répandue en traitement d'image et qui existe en de nombreuses variantes. Elles peuvent être distinguées de deux manières principalement, selon le choix des germes ou leurs critères de croissance. En ce qui concerne la sélection des régions initiales, il est possible de procéder :

- manuellement.
  L'opérateur définit lui-même les graines, ce qui se révèle rapidement limité dans le cas de grandes bases de données.
- au hasard.
  À chaque étape, une région est formée à partir d'un germe choisi aléatoirement dans l'image. Une fois la convergence atteinte, un nouveau germe est sélectionné et ainsi de suite jusqu'à ce que toute l'image soit recouverte. Cette technique nécessite souvent une étape supplémentaire de fusion de régions.
- par un seuillage sur les intensités des pixels si on possède un *a priori* sur les objets recherchés. [Garza-Jinich et al., 1999] y ajoutent un critère de concentration pour éliminer les pixels isolés.
- par une segmentation grossière de l'image à l'aide d'atlas et de recalage.
  En médecine notamment, des modèles d'organes permettent de localiser approximativement les différentes zones recherchées.

Généralement, les pixels p susceptibles d'être ajoutés sont adjacents à la région R_n pour garantir sa connexité.

$$C_n = \{p \in \bar{R_n} \mid N_v(p) \cap R_n = \emptyset\} \tag{2.3}$$

$\bar{R_n}$ représente l'ensemble des pixels non segmentés. Le voisinage N_v définit le type de connexité de la région. Certaines techniques proposent également de supprimer de la région ceux qui ne répondent plus ou pas assez aux critères d'homogénéité [Revol and Jourlin, 1997].

Enfin, l'utilisateur peut faire le choix de faire croître les régions successivement ou de façon concurrentielle, ce qui conditionne les critères d'homogénéité utilisés. Dans le cas de croissance indépendante, le critère le plus simple serait d'agréger deux pixels p et q si leurs intensités I sont proches, au sens d'un seuil γ fixé.

$$|I(p) - I(q)| \leq \gamma \tag{2.4}$$

Il est toutefois impossible de garantir que les régions ainsi obtenues soient homogènes. [Sekiguchi et al., 1994] proposent de prendre également en compte le germe s.

$$|I(p) - I(q)| \leq \gamma \tag{2.5}$$

$$|I(p) - I(s)| \leq \delta \tag{2.6}$$

La difficulté provient du choix des seuils γ et δ qui modifient fortement la segmentation finale. Leurs valeurs peuvent être évaluées à l'aide d'une base de données d'apprentissage ou en procédant à plusieurs segmentations avec des seuils différents parmi lesquelles le meilleur résultat seul est retenu.

Il est possible de faire croître plusieurs régions simultanément. Dans ce cas, l'image est partitionnée en N germes qui grossissent de façon concurrentielle. Lorsqu'un pixel est candidat pour plusieurs régions, il est aggloméré à celle qui propose le meilleur critère. Cette technique présente certaines similitudes avec celles à base de graphe qui cherchent la segmentation présentant le meilleur compromis.

2.3.2 Sélection des graines contrastées

La qualité des régions initiales conditionne celle de la segmentation finale. Comme la croissance classique ne permet pas la suppression de pixels, il est nécessaire de bien évaluer dès le départ les caractéristiques de l'objet recherché. Un nombre trop important de graines risque d'augmenter considérablement le temps de calcul et de sur-segmenter l'image. Au contraire, si leur nombre est trop faible, la probabilité de ne pas détecter l'objet augmente.

Les rectangles de type panonceaux se caractérisent par un fond clair sur lequel est affiché un message généralement de couleur sombre. L'approche la plus intuitive consiste à seuiller directement l'image pour en extraire les pixels clairs. Ses limites sont cependant rapidement atteintes lorsque les conditions d'acquisition ne sont pas contrôlées et que la variance dans la luminosité et le contraste des images est très importante. Il est donc difficile d'estimer la plage d'intensité relative aux panonceaux seuls sans segmenter le fond. Une alternative possible est d'introduire un *a priori* en utilisant la région du panneau détecté au préalable (figure 2.9). Cela nécessite un pré traitement du centre du panneau pour en extraire les valeurs désirées.

Il nous est donc apparu plus fiable de rechercher des zones fortement contrastées, ou trous. Pour la recherche de ces pixels sombres, nous nous sommes intéressés à la reconstruction morphologique [Vincent, 1993].
En binaire, la reconstruction $\rho_I(M)$ de I à partir de l'image de marqueurs M est l'union des composantes connexes de I qui contiennent au moins un pixel de M.

$$\rho_I(M) = \bigcup_{k, M \cap I_k \neq \emptyset} I_k \tag{2.7}$$

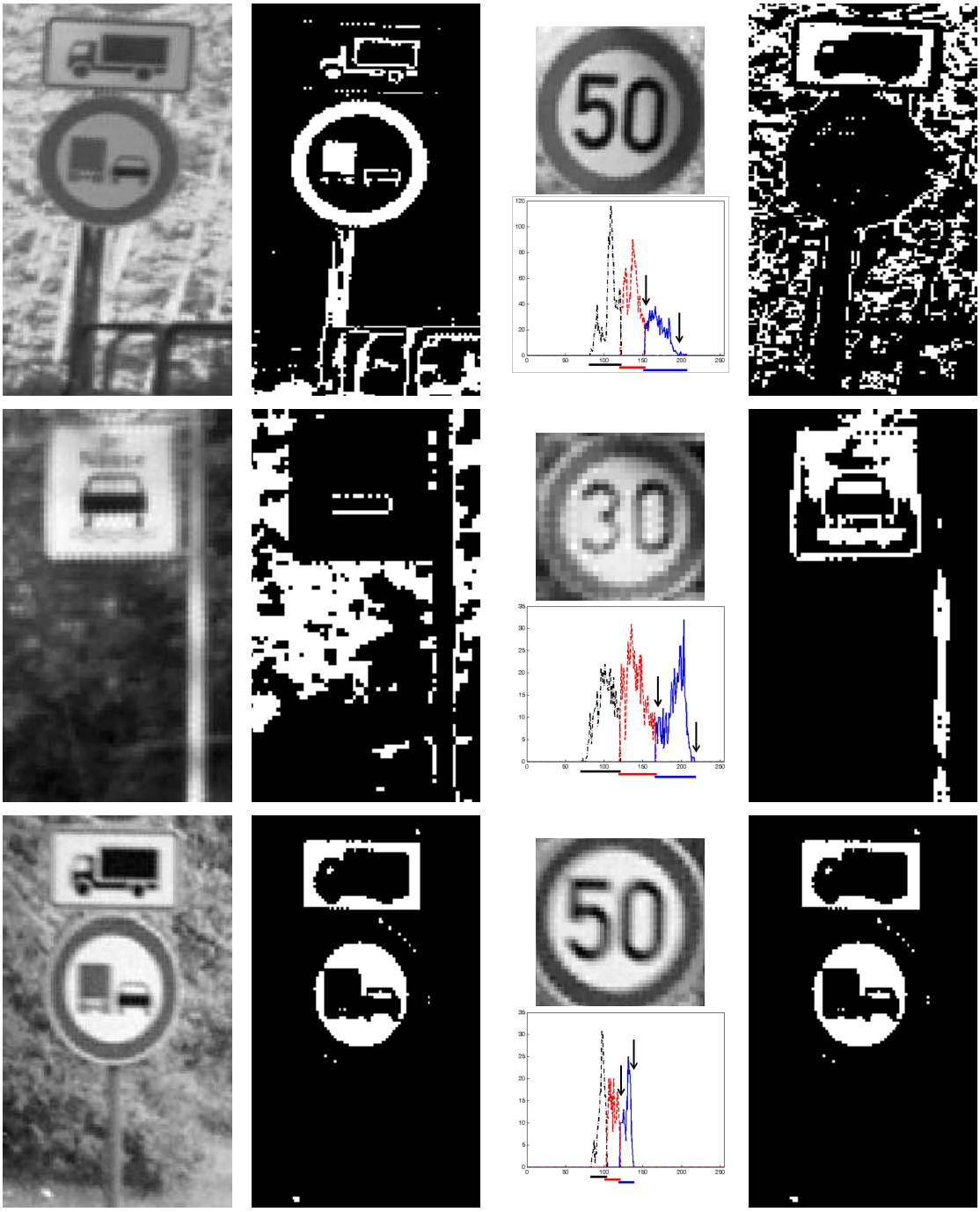
La transposition à l'espace des fonctions passe par la notion de distance géodésique. La distance géodésique d_X entre deux points de X correspond à la longueur du plus court chemin les joignant tout en restant dans X. La définition de la dilatation géodésique $\delta_X^{(n)}(Y)$ de taille n de Y dans X est alors immédiate et correspond à l'ensemble des pixels p de X dont la distance géodésique à Y est inférieure à n.

$$\delta_X^{(n)}(Y) = \{p \in X \mid d_X(p, Y) \leq n\} \tag{2.8}$$

La reconstruction peut alors s'écrire en fonction de ce nouvel opérateur.

$$\rho_I(M) = \bigcup_{n \geq 1} \delta_I^{(n)}(M) \tag{2.9}$$

La reconstruction est une opération de morphologie mathématique itérée jusqu'à idempotence, *i.e.* jusqu'à ce que la reconstruction de l'image donne l'image elle-même. Comme illustré par la figure 2.10, la fonction résultante correspond à l'originale "aplatie" au niveau du marqueur. Par soustraction à l'originale, il est possible d'extraire les pics de l'image. Dans notre cas, il s'agit de retourner les vallées, correspondant aux zones sombres, en appliquant l'opérateur au complémentaire de l'image.



(a) Image originale (b) Graines obtenues pour un seuil **fixe** (c) Panneau et histogramme correspondant (d) Graines obtenues pour un seuil calculé à partir d'***a priori***

FIGURE 2.9 – Graines obtenues par deux méthodes différentes. (a) Les zones de recherche de panonceaux sont situées au-dessous des panneaux de limitation de vitesse. (b) La méthode de sélection de pixels appartenant au panonceau la plus simple et rapide consiste à choisir un seuil **fixe**. Malheureusement, les résultats sont de qualité très variable, fonction de la luminosité de l'image. (c) Une autre possibilité est d'utiliser un *a priori* sur les intensités, notamment grâce à l'histogramme du panneau précédemment détecté. Pour cela, nous lui appliquons l'algorithme k-means, avec $k = 3$, puis nous sélectionnons la plage d'intensités du cluster le plus clair (en bleu sur l'histogramme, compris entre les deux flèches). (d) En blanc, les graines obtenues à partir d'***a priori*** déduit de l'histogramme du panneau situé au-dessus paraissent plus satisfaisantes.

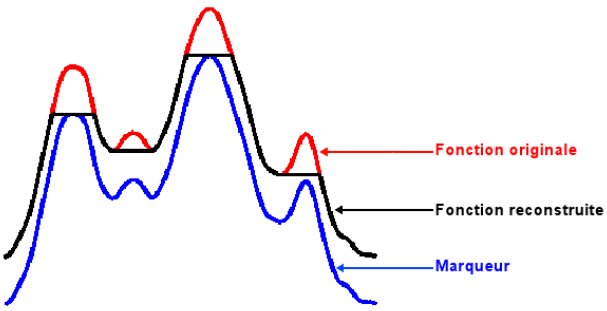


FIGURE 2.10 – Résultat (en noir) d'une reconstruction morphologique sur une fonction (en rouge) par un marqueur (en bleu). Les pics de la fonction originale sont ainsi supprimés.

Après reconstruction et soustraction à l'originale, l'image résultante I_s doit être filtrée pour ne retenir que les composantes connexes les plus contrastées. Pour cela, nous procédons en deux étapes. Tout d'abord, l'image est seuillée selon deux valeurs t_1 et t_2 telles que :

1. t_1 et t_2 dépendent de la moyenne μ et de l'écart-type σ de l'intensité de l'image.

$$t_1 = \mu + \kappa_1 \sigma \tag{2.10}$$
$$t_2 = \mu + \kappa_2 \sigma \tag{2.11}$$
$$\kappa_1 > \kappa_2 \tag{2.12}$$

   De cette façon, le seuillage est plus robuste aux changements de luminosité ou de contraste.

2. t_1 est choisi de façon à sélectionner les pixels avec un fort contraste tandis que t_2 permet de récupérer l'ensemble des pixels appartenant aux régions sélectionnées. Plus t_1 est grand, moins nous risquons d'avoir de faux positifs et plus t_2 est grand, plus les graines sélectionnées seront concentrées autour des pics. Nous obtenons alors deux images I_{t_1} et I_{t_2} en seuillant l'image I_s :

$$I_{t_i} = \{i \in \{1,2\}, p \in I_s \mid I_s(p) \geq t_i\} \tag{2.13}$$

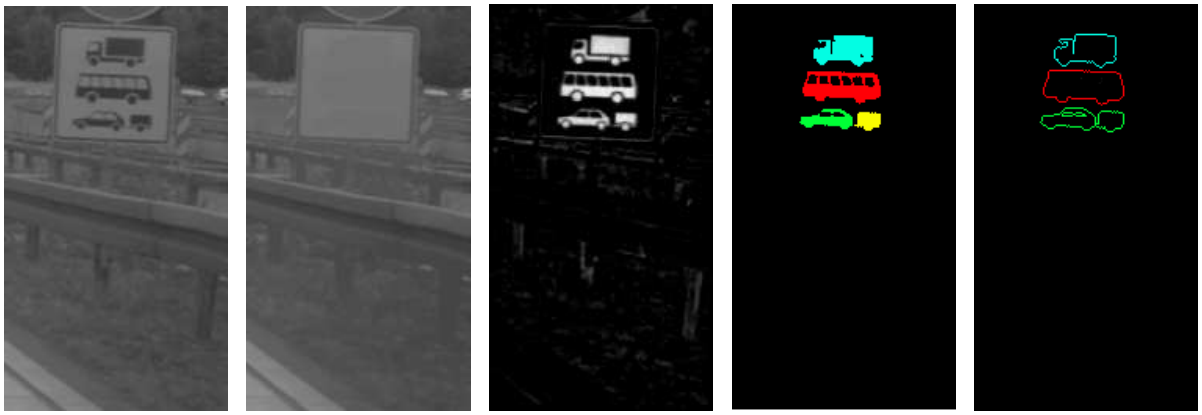
3. Les ensembles E finalement conservés correspondent à l'intersection des composantes connexes CC_1 et CC_2 de I_{t_1} et I_{t_2}.

$$E = \{CC \in CC_1 \cap CC_2 \mid Card(CC_1) > 3\} \tag{2.14}$$

   $Card(CC_1) > 3$ nous permet d'éliminer les composantes trop faiblement contrastées qui pourrait s'apparenter à du bruit.

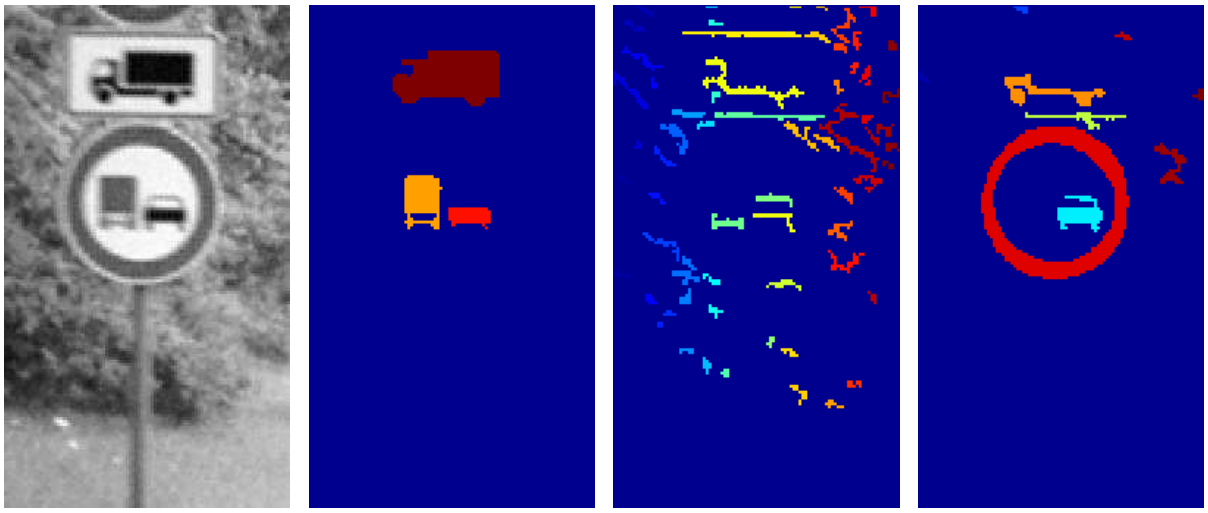
Les graines initiales R_0 servant de point de départ à la croissance de régions sont alors définies comme l'ensemble de pixels connexes formant la couronne extérieure de chaque composante connexe restante. La méthode complète est illustrée par la figure 2.11.

L'avantage de cette technique par rapport à celle, plus classique, du *bottom-hat* [Dougherty, 1992] est que les paramètres t_1 et t_2 ne dépendent pas de la taille des objets à segmenter. Nous avons comparé les résultats obtenus en utilisant la reconstruction et le *bottom-hat* pour extraire les pixels sombres de l'image (figure 2.12). Le résultat de cette seconde approche est fortement influencée par la taille de l'élément structurant r_{ES}. De plus, elle ne retourne pas la composante connexe complète mais seulement les pixels situés sur la frontière des régions contrastées.



(a) Image originale (b) Reconstruction (c) Soustraction à l'originale (d) Composantes connexes après seuillage (e) Graines résultantes

FIGURE 2.11 – Exemple de sélection de graines contrastées à l'aide de la reconstruction. (a) Image originale. (b) L'opérateur de reconstruction morphologique retourne une image pour laquelle l'intensité des pixels sombres a été mise à jour par rapport à celle des pixels environnants. (c) Les pixels ressortent d'autant mieux, lorsque nous soustrayons les deux images, qu'ils sont sombres par rapport à leur voisinage. (d) L'image est ensuite seuillée pour ne conserver que les composantes connexes parmi les plus contrastées. (e) Finalement, les graines générées par notre approche correspondent à la couronne extérieure des composantes précédentes. Des quatre composantes connexes de (c), nous générons seulement trois ensembles initiaux de graines car les frontières extérieures des composantes verte et jaune ont fusionné.



(a) Image originale (b) Reconstruction (c) *Bottom-hat* pour $r_{ES} = 3$ (d) *Bottom-hat* $r_{ES} = 7$

FIGURE 2.12 – Composantes connexes contrastées obtenues par la technique de reconstruction (b) et pour deux tailles différentes r_{ES} d'élément structurant pour le *bottom-hat*. L'utilisation de l'opérateur de reconstruction nous permet d'extraire directement la totalité des composantes connexes sombres sur fond clair de l'image. En revanche, un mauvais choix de taille r_{ES} d'élément structurant pour le *bottom-hat* conduit à une sélection des pixels externes des objets sombres ayant une taille au moins égale à r_{ES}.

2.3.3 Recherche de zones uniformes

Après avoir défini les régions initiales, il nous faut déterminer les critères permettant leur croissance. Nous avons opté pour une croissance non concurrentielle, *i.e.* une seule région croît à chaque étape. Ainsi, nous aurons autant de rectangles potentiels qu'il y a de graines. Notre choix de critères s'apparente à celui de [Sekiguchi et al., 1994]. À chaque itération ne sont concernés que les pixels du voisinage immédiat N_v de la région courante R_n. N_v peut correspondre à un voisinage de connexité 4 ou 8.

Pour faire partie de la liste des candidats C_n, les pixels doivent donc vérifier trois critères :

- **adjacence**.
  À chaque itération, nous ne considérons que les pixels du voisinage immédiat N_v de la région courante R_n. Nous notons l'ensemble E_{adj_n}.

$$E_{adj_n} = \{p \in I \setminus R_n \mid N_v(p) \cap R_n \neq \emptyset\} \tag{2.15}$$

- homogénéité **locale**.

$$E_{L_n} = \{p \in E_{adj_n} \mid \exists\, q \in R_n, |\frac{I(p)}{I(q)} - 1| \leq \kappa_L\} \tag{2.16}$$

- homogénéité **globale**.

$$E_{G_n} = \{p \in E_{adj_n} \mid |\frac{I(p)}{\mu_0} - 1| \leq \kappa_G\} \tag{2.17}$$

  μ_0 correspond à la moyenne des pixels de la région initiale R_0. L'intervalle de valeurs sur la région sera au final limité à $2\kappa_G\mu_0$.

L'ensemble des pixels candidats C_n est obtenu par l'intersection des deux derniers.

$$C_n = E_{L_n} \cap E_{G_n} \tag{2.18}$$

La combinaison des aspects local et global est essentielle pour obtenir un résultat correct. Le premier nous permet d'éliminer rapidement des pixels de contours ou des artefacts, ce qui permet d'accélérer le processus. Ces derniers correspondent en effet à des zones de forts gradients dans l'image. L'information locale assure également une croissance plus graduelle de la région. En revanche, l'aspect global évite de tomber dans l'excès inverse car il existe presque toujours un chemin connexe de couleur proche qui relie deux points d'une image. Une fois les candidats sélectionnés, nous les ajoutons directement à la région et recommençons l'opération jusqu'à ce qu'aucun pixel ne puisse plus être ajouté.

Les critères ont été choisis de manière adaptative afin d'être plus robuste aux changements d'illumination et de contraste. Les valeurs de κ_L et κ_G doivent toutefois être bien choisies car :

- trop grandes, la région générée risquerait de "déborder" et d'englober des pixels du fond ;
- trop petites, l'algorithme serait trop sensible au bruit.

La figure 2.13 présente un résultat de segmentation pour différentes valeurs de κ_L et κ_G.

2.3.4 Choix des paramètres

Quatre paramètres principaux conditionnent les performances de notre méthode de croissance de région : κ_1, κ_2, κ_L et κ_G. Les deux premiers sont utilisés pour sélectionner les graines de départ. La figure 2.14 présente le rappel (équation 2.19) en fonction du nombre de Faux Positifs par Positif FPPP (équation 2.20) pour la base de données db_D (tableau 2.2), pour différents couples

(a) Image originale (b) Critères surévalués (c) Critères bien évalués (d) Critères sous-évalués

FIGURE 2.13 – Influence du choix des critères κ_L et κ_G sur la qualité de la segmentation. (b) Une surévaluation d'une valeur fait croître la région au-delà des limites attendues. (d) En revanche, une sous-évaluation arrête le processus bien trop tôt.

de valeurs κ_2_κ_1 ($\kappa_L = 0.15$ et $\kappa_G = 0.05$). Cette courbe est obtenue en modifiant le seuil de *rectangularité* des régions générées par notre méthode, c'est-à-dire le rapport entre son aire et celle de sa boîte englobante, et en considérant une région correcte d'après sa mesure de Jaccard $\mathscr{J} \geq 0.5$ (équation 2.25). Le rappel et le nombre de Faux Positifs par Positif FPPP sont des mesures utilisées pour évaluer les performances des algorithmes de détection.

$$\text{Rappel} = \frac{\text{Nombre de panonceaux correctement détectés}}{\text{Nombre de panonceaux de la base de données}} \tag{2.19}$$

$$\text{FPPP} = \frac{\text{Nombre de rectangles retournés}}{\text{Nombre de panonceaux de la base de données}} \tag{2.20}$$

Nous voyons que le rappel n'est que peu modifié par le choix de ces paramètres et a une valeur proche de 0.7. L'influence de κ_1 sur le nombre de faux positifs apparaît clairement et passe de FPPP = 6.2 pour $\kappa_1 = 1.0$ à FPPP = 2.8 pour $\kappa_1 = 2.5$. En effet, κ_1 permet de sélectionner uniquement les composantes de l'image reconstruite qui sont suffisamment contrastées par rapport au fond.

Le paramètre κ_2 détermine les pixels conservés dans la région initiale R_0. S'il est trop faible, les candidats retenus seront trop nombreux et risquent de déborder hors du panonceau. Une valeur trop grande générera des régions trop restreintes avec de fortes contraintes sur les intensités tolérées pour les régions finales car la mesure d'homogénéité globale dépend aussi de l'intensité moyenne μ_0 des pixels de R_0. Elle est donc intrinsèquement liée à κ_L et κ_G. Nous avons déterminé empiriquement leurs valeurs à $\boldsymbol{\kappa_1 = 2.5}$, $\boldsymbol{\kappa_2 = 0.8}$, $\boldsymbol{\kappa_L = 0.22}$ et $\boldsymbol{\kappa_G = 0.12}$.

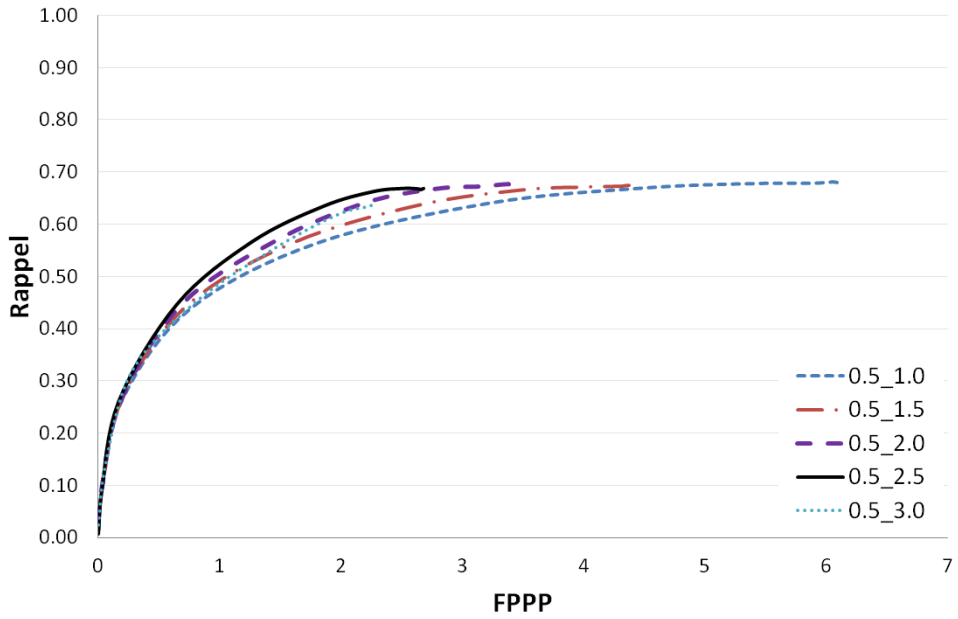


FIGURE 2.14 – Rappel en fonction du nombre de Faux Positifs Par Positif (FPPP) pour différents couples $\kappa_2_\kappa_1$ pour la base allemande db_D. Les paramètres κ_L et κ_G sont fixés à 0.15 et 0.05. Cette courbe est obtenue en faisant varier le seuil de *rectangularité* des régions en sortie de notre algorithme. La taux de faux positifs est inversement proportionnel à κ_1 et vaut 6.2 pour $\kappa_1 = 1.0$ et 2.8 pour $\kappa_1 = 2.5$. Nous avons choisi $\boldsymbol{\kappa_1 = 2.5}$.

2.4 Réalisation d'un benchmark

Pour valider notre approche, nous avons décidé de la comparer à trois autres méthodes appartenant aux trois familles principales de techniques de segmentation : frontière, seuillage et région.

2.4.1 Approches frontière

Au vu de la littérature, la majorité des méthodes implémentées dans le domaine de la détection de panonceaux est basée sur les contours. Le principe est qu'il existe une délimitation franche entre l'objet à segmenter et le fond. Tout d'abord, une carte de contours est établie, dans le domaine de l'image ou de Hough. Ensuite, il s'agit d'extraire ceux qui font partie de la forme recherchée, par *template-matching* ou en faisant appel à des techniques de vote. La première approche, la plus intuitive, nécessite de définir précisément des gabarits des formes recherchées puis de calculer une mesure de similarité entre ces derniers et l'image. La seconde est plus complexe et fonctionne selon le même principe que la transformée de Hough pour détecter des polygones réguliers. Elles souffrent néanmoins toutes les deux d'une grande sensibilité au bruit et aux occlusions.

Une autre façon d'aborder le problème est d'utiliser des modèles déformables, guidés par une énergie déterminée par l'adéquation entre le contour, les données et la forme recherchée. Plus robustes par nature, ils sont aussi plus complexes à mettre en œuvre et à maintenir [Sethian, 1997, Mumford and Shah, 1985, Chan and Vese, 2001, Rousson and Paragios, 2002].

Nous avons étudié l'approche développée par [Herbin-Sahler et al., 2007] orientée *template-matching*. La recherche des panonceaux s'effectue à partir de la carte de contours obtenus à l'aide du filtre de Canny et fait appel à différentes heuristiques pour extraire les rectangles. Les étapes

de l'algorithme sont les suivantes :

1. **Application du filtre de Canny** pour extraire les pixels de contour.
2. **Filtrage des contours suivant leur orientation**.
   En supposant que les panonceaux sont majoritairement horizontaux, il est possible de filtrer la carte obtenue pour ne conserver que les contours horizontaux et verticaux. L'intervalle correspondant est $[-\theta; +\theta] \cup [90-\theta; 90+\theta]$, où θ correspond à la tolérance sur les orientations et permet de prendre en compte de petites rotations.
3. **Regroupement des contours en segments**.
   Cette étape sert d'intermédiaire entre bas- (pixel) et haut-niveau (rectangle). Il s'agit de déterminer les segments dans l'image à partir d'un ensemble d'heuristiques et de paramètres, comme l'espacement entre deux contours pour les considérer comme appartenant au même segment. Ces derniers doivent être soigneusement choisis pour garder un compromis robustesse/précision acceptable.
4. **Association des segments parallèles en paires**.
   Pour réduire la complexité de la tâche de recherche de rectangles, [Herbin-Sahler et al., 2007] extraient tout d'abord les paires de segments parallèles. Certains *a priori* sur la taille des objets recherchés permettent encore de réduire le nombre de candidats.
5. **Formation des rectangles**.

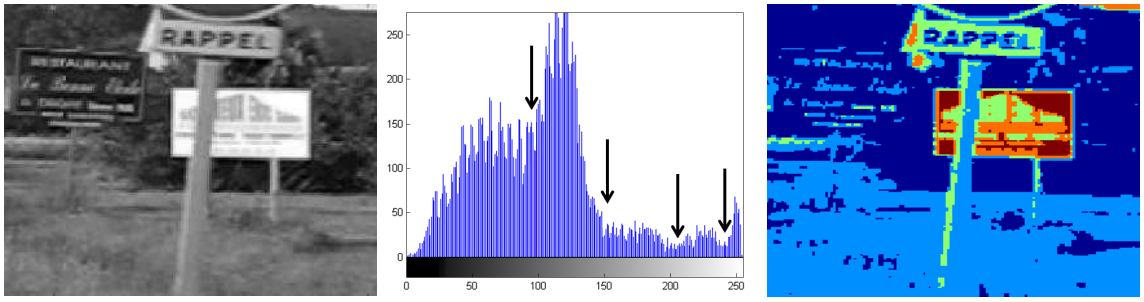
Le principal avantage de cette méthode est sa rapidité par rapport à d'autres approches basées contour. Les nombreuses restrictions utilisées permettent en effet de réduire la recherche dans l'image mais rendent son utilisation complexe et peu modulaire. Elle souffre aussi des inconvénients liés aux méthodes frontière.

2.4.2 Approches par seuillage

Principalement rencontré dans le domaine de la détection de panneaux, le seuillage exploite le fait que la couleur des objets à segmenter est relativement homogène et perceptible dans l'image. Ce postulat se vérifie car les panneaux sont de couleurs normalisées et spécialement conçus pour être visibles à grande distance par le conducteur. Après le traitement de l'image pour faire ressortir les pixels d'intérêt, un seuillage et une recherche de composantes connexes permettent d'extraire les objets recherchés. Pour détecter les panneaux de limitation de vitesse, cerclés de rouge, [Ruta et al., 2010] proposent de traiter l'image RGB de manière à renforcer la composante rouge puis de seuiller l'image de façon récursive à l'aide d'un QuadTree.

Plus généralement, l'étude de l'histogramme peut fournir des informations intéressantes sur la répartition colorimétrique de l'image. Il est possible d'en extraire différentes plages d'intensité auxquelles appartiennent les objets (figure 2.15). Bien que simple en apparence, cette technique risque de ne pas fonctionner lorsque les objets sont trop texturés, ou que les frontières de l'objet se mêlent au fond. De plus, cette segmentation ne contient aucune information spatiale par définition et nécessite donc une phase de filtrage géométrique en post-traitement pour affiner les résultats.

Les panonceaux sont des objets relativement homogènes ce qui rend possible une approche globale basée sur la couleur. En effet, si les panonceaux ont des couleurs relativement distinctes du reste de l'image, ils devraient apparaître comme des pics dans l'histogramme. Pour segmenter l'image, nous avons implémenté la technique de [Zhang et al., 2003] de décomposition d'histogramme. En supposant que ce dernier soit la somme d'un nombre inconnu M de gaussiennes,



(a) Image originale (b) Histogramme (c) Résultat après seuillage

FIGURE 2.15 – Illustration d'une technique de seuillage manuel d'une image de panonceau. (a) Zone de recherche originale du panonceau en niveau de gris. (b) Histogramme de l'image. Quatre seuils (représentés par des flèches noires) ont été définis manuellement. Ils correspondent visuellement aux limites des différentes gaussiennes composant la distribution d'intensités. (c) Image obtenue après seuillage. Aucune information spatiale n'est prise en compte, un post-traitement est donc nécessaire pour extraire les composantes connexes et filtrer les objets ne répondant pas aux critères géométriques recherchés.

l'objectif est d'estimer les paramètres $\theta_m = (\mu_m, \Sigma_m), m \in [1; M]$ de chaque composante. La distribution peut alors s'écrire :

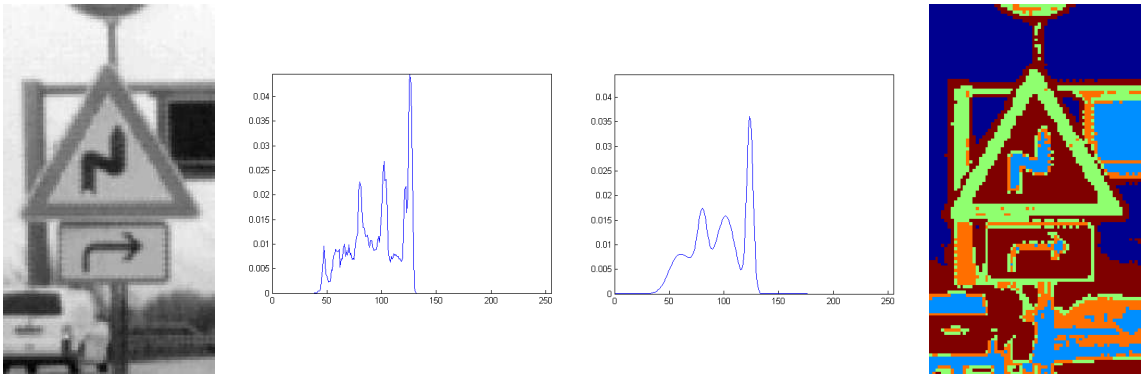
$$p(x \mid \Theta) = \sum_{m=1}^{M} \pi_m p(x \mid \theta_m) \tag{2.21}$$

où $\Theta = (\pi_m, \theta_m), m \in [1; M]$ et π_m représente la proportion de la composante m dans la distribution. Le modèle est estimé à l'aide de deux étapes principales itérées jusqu'à convergence, une phase d'espérance-maximisation et une de décomposition/fusion. La première permet d'estimer, à nombre fixe de gaussiennes , la distribution qui approche le mieux les données. La seconde consiste à modifier un certain nombre de modes sélectionnés par rapport à deux critères, de fusion et de décomposition. Deux gaussiennes seront ainsi fusionnées si elles sont suffisamment corrélées et inversement, un mode considéré comme inadapté à la distribution sera subdivisé en deux.

La mise en œuvre de cette technique pour la détection de panonceaux est la suivante. L'histogramme de la région de recherche est calculé puis décomposé en un nombre M de gaussiennes suivant le principe détaillé précédemment. Lorsque les intensités des différents objets de l'image sont suffisamment distinctes, la modélisation par des gaussiennes permet une bonne segmentation (figure 2.16). En revanche, si la distribution est trop compacte, le résultat obtenu ne sera pas exploitable pour notre application (figure 2.17).

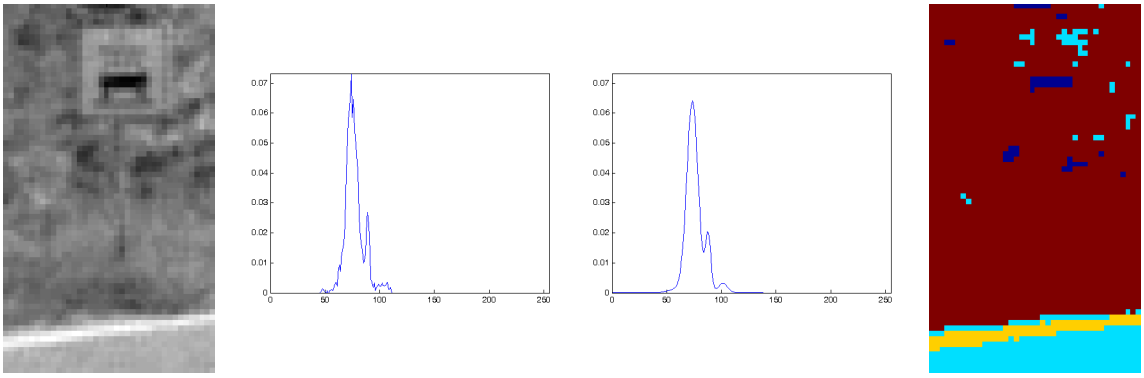
Une phase de post-traitement est ensuite nécessaire, d'une part pour récupérer les composantes connexes, d'autre part pour sélectionner uniquement le mode correspondant au panonceau. En effet, après avoir estimé la distribution de gaussiennes la plus probable, il reste à choisir quel mode correspond à celui de l'objet recherché. Pour cela, nous supposons que l'intensité du panonceau est similaire à celle du centre du panneau détecté au préalable. Nous appliquons donc un *k-means* sur le panneau (avec $k = 3$) pour en extraire la composante la plus claire qui donne l'ordre de grandeur recherché correspondant au fond du panonceau (figure 2.18).

Le principal intérêt de cette approche est son absence de paramètres, notamment sur le nombre de modes. En effet, nous ne disposons pas de cette connaissance *a priori* puisqu'elle dépend de l'image à traiter. Malheureusement, cela entraîne un temps de traitement parfois long, si l'image est complexe ou si l'initialisation est trop éloignée du résultat. De plus, nous travaillons sur des images en niveaux de gris, rendant difficile la distinction du panonceau et du fond (figure 2.19).



(a) Image originale (b) Histogramme original (c) Approximation de la distribution (d) Segmentation

FIGURE 2.16 – Segmentation obtenue par la technique de [Zhang et al., 2003]. L'histogramme de l'image originale (b) a été approximé par un mélange de cinq gaussiennes (c). (d) L'image ainsi segmentée n'est pas encore exploitable. Les pixels ont été chacun affectés à une classe sans contrainte de connexité et un seul mode nous intéresse en réalité.



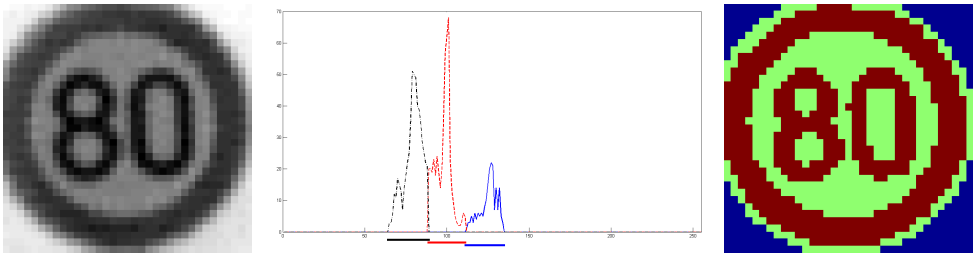
(a) Image originale (b) Histogramme original (c) Approximation de la distribution (d) Segmentation

FIGURE 2.17 – Exemple de mauvaise segmentation obtenue par la technique de [Zhang et al., 2003]. (c) L'approximation par une distribution de quatre gaussiennes de l'histogramme original ne permet pas d'extraire le panonceau. Ce dernier a en effet une texture proche de celle du fond rendant difficile l'utilisation de cette méthode.

2.4.3 Approche par région

Cette dernière catégorie consiste à manipuler l'image en terme de régions connexes ce qui facilite l'analyse. L'approche que nous avons proposée dans la partie 2.3 se rapporte à cette catégorie. Outre la croissance de régions, nous pouvons trouver celles dites de décomposition/fusion, qui consistent au contraire à diviser l'image en régions homogènes. Le risque de sur-segmentation étant assez élevé, une étape de fusion est ensuite mise en place en vue de fusionner les zones adjacentes qui présentent des caractéristiques communes.

Les méthodes à base de graphes appartiennent également à cette famille d'algorithmes. En modélisant l'image sous forme d'un graphe dont les nœuds seraient les régions connexes et les



(a) Panneau (b) Histogrammes (c) Segmentation obtenue

FIGURE 2.18 – Recherche du mode correspondant à l'intensité du panonceau à partir du panneau à l'aide d'un algorithme *k-means*. (a) Région du centre du panneau de limitation de vitesse. (b) Histogrammes des différentes classes. Nous avons choisi un nombre de *clusters* $k = 3$ correspondant, par valeurs moyennes d'intensité croissantes, aux chiffres, au cercle bordant le panneau et au centre. (c) Segmentation obtenue.

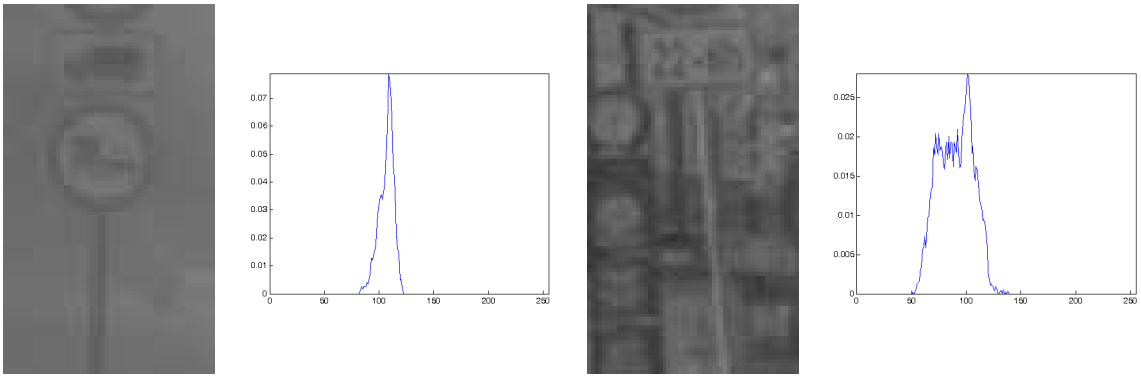


FIGURE 2.19 – Exemples de situations où le panonceau et le fond sont difficilement séparables d'un point de vue colorimétrique.

arêtes les mesures de similarité entre régions, il est possible de voir la segmentation comme une coupe de graphe qui minimiserait une énergie. L'attrait de ce type de technique est toutefois contrebalancé par le coût algorithmique rapidement prohibitif.

Une technique très utilisée en segmentation d'images est celle de [Felzenszwalb and Huttenlocher, 2004]. L'image est considérée comme un graphe dont les nœuds V représentent les régions, homogènes au sens de certains critères, et les arêtes E de poids w la mesure de différence entre ces régions. Elle peut s'apparenter à une méthode proche de la croissance de régions concurrentielle car elle recherche des régions connexes, homogènes et distinguables par rapport aux autres. Au cours du processus, les arêtes seront supprimées (resp. conservées) entre deux régions R_1 et R_2 si la valeur du prédicat P vaut 0 (resp. 1). Ce dernier est basé sur une mesure de dispersion interne Σ d'une même région et dissemblance Δ entre des pixels situés de chaque côté de la frontière entre deux régions.

$$\Delta(R_1, R_2) = min_{v_1 \in R_1, v_2 \in R_2, (v_1, v_2) \in E} w(v_1, v_2) \tag{2.22}$$

$$\Sigma(R) = max_{e \in MST(R,E)} w(e) \tag{2.23}$$

$$P(R_1, R_2) = \begin{cases} 1 & \text{si } \Delta(R_1, R_2) > min(\Sigma(R_1) + \tau(R_1), \Sigma(R_2) + \tau(R_2)) \\ 0 & \text{sinon} \end{cases} \tag{2.24}$$

$\Delta(R_1, R_2)$ mesure la dissemblance de deux régions et correspond au poids minimum des arêtes reliant deux nœuds situés de chaque côté de leur frontière.
$\Sigma(R)$ représente la dispersion interne de R, vue comme le plus grand poids des arêtes e du sous-arbre $MST(R, E)$ (*Minimum Spanning Tree* - Arbre Couvrant Minimal).
$\tau(R) = \frac{k}{|R|}$ permet d'éviter de sur-segmenter l'image. En effet, les régions de petite taille ($| R | \rightarrow 1$) seront plus facilement homogènes que les grandes. La constante k joue le rôle d'un potentiomètre, plus sa valeur sera grande et plus les composantes de grande taille seront autorisées.

Dans notre implémentation, $k = \kappa_A * Aire(image)$ dépendra de la taille de la zone où nous recherchons le panonceau. De cette façon, les rectangles recherchés occuperont plus ou moins le même espace que le panonceau soit près ou loin du véhicule.

La figure 2.20 illustre le rappel en fonction de la valeur du nombre de faux positifs pour la base de données db_D (tableau 2.2) et pour différentes valeurs de κ_A. Une région est correcte si $\mathscr{J} \geq 0.5$ (équation 2.25). Nous observons que pour $\kappa_A = 0.01$, le nombre de faux positifs est important (FPPP = 9) car la segmentation s'arrête trop tôt, les rectangles obtenus sont de petite taille et en grand nombre pour une image donnée. Pour de fortes valeurs, c'est le contraire. Seules les grandes régions sont conservées, conduisant à un nombre plus faible de faux positifs et invariablement un rappel plus faible aussi. Nous choisissons $\kappa_{\mathbf{A}} = \mathbf{0.03}$ qui garantit un rappel de 75% pour un nombre de faux positifs raisonnable de FPPP = 6.2. Au stade de la détection, l'enjeu est d'avoir le meilleur rappel pour éviter de perdre trop de panonceaux. L'étape de classification permettra ensuite d'éliminer les faux positifs restants.

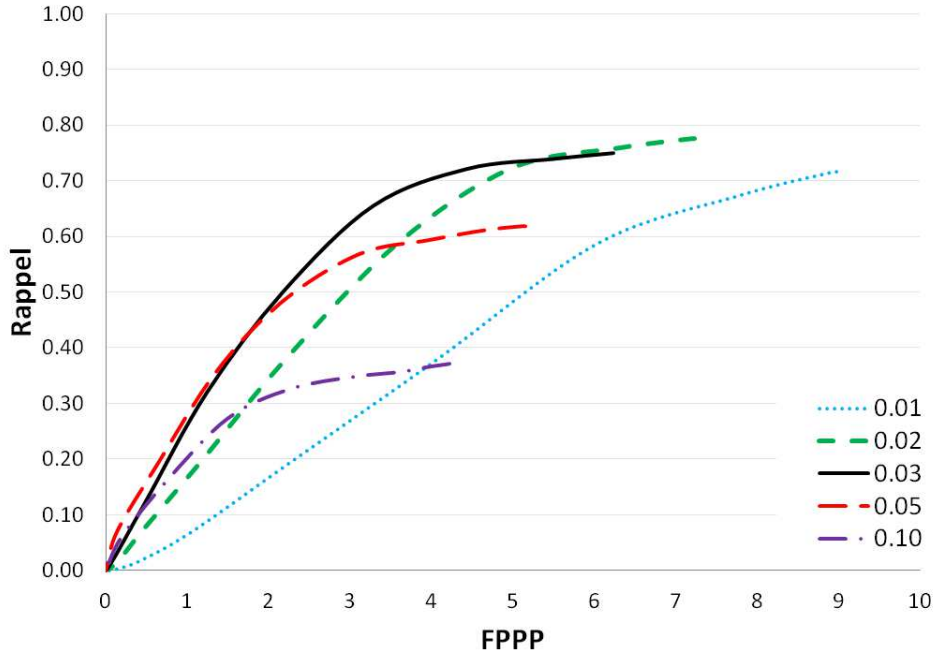


FIGURE 2.20 – Rappel en fonction du nombre de Faux Positifs Par Positif (FPPP) lorsque nous faisons varier le paramètre κ_A. Le nombre de faux positifs diminue à mesure que la valeur de κ_A augmente, car plus la taille minimale autorisée des rectangles recherchés est faible, plus nous aurons de candidats et donc potentiellement de fausses alarmes. En revanche, une trop grande valeur de κ_A est inadaptée car nous perdons les panonceaux de faible taille d'où une chute du rappel. Nous avons choisi $\kappa_{\mathbf{A}} = \mathbf{0.03}$ car offrant le meilleur compromis Rappel-FPPP.

Pour assurer un temps de traitement quasi linéaire, contrairement à la majorité des applications mettant en œuvre des graphes, [Felzenszwalb and Huttenlocher, 2004] ordonnent les arêtes par poids croissant à l'initialisation et traitent ensuite les candidats dans cet ordre. Il n'y a pas de réarrangement des nœuds à chaque itération en fonction de la segmentation courante. Les expérimentations réalisées semblent assurer un minimum d'erreur avec cette technique malgré son côté "glouton".

Toutes les régions générées par les trois méthodes ne sont pas des candidats potentiels car elles segmentent toute la zone, que nous avons volontairement choisie assez large. Nous procédons donc, après l'étape de détection, à une présélection des rectangles en fonction de leur hauteur h, largeur L, ratio r et distance d au centre de la zone de recherche. Elles doivent vérifier $\mathbf{h} \geq \mathbf{10}$, $\mathbf{L} \geq \mathbf{10}$, $\mathbf{r} \in [\mathbf{0.5}; \mathbf{5.0}]$ et $\mathbf{d} \leq \mathbf{0.25} \times \mathbf{L}(\mathbf{zone})$.

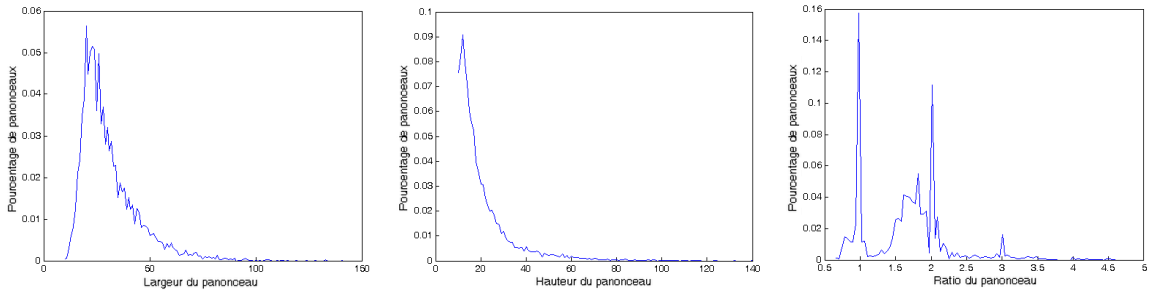
2.5 Étude comparative

2.5.1 Bases de données

Actuellement, aucune base de données publique n'existe qui permettrait de comparer notre approche à la littérature. Nous avons donc évalué les techniques présentées dans les sections 2.3 et 2.4 sur deux bases privées. Elles ont été acquises dans les deux pays concernés par le projet Speedcam, l'Allemagne et la France. Cela introduit une première source de variabilité des panonceaux principalement parce qu'ils sont délimités par une bordure noire en Allemagne et non en France. La variété des panonceaux présents en terme de hauteur, largeur et ratio est également importante (figure 2.21). Il est à noter que la taille minimale de 10 pixels correspond à un panonceau vu à une distance de 15 mètres. De plus, certaines séquences ont été acquises dans des conditions climatiques dégradées, ce qui rend la détection plus difficile puisque l'image est nettement plus floue. Un récapitulatif des bases est donné dans le tableau 2.2.

Base	Pays	Nombre	Largeur	Hauteur	Ratio
db_F	France	1040	$[10; 42]$	$[10; 42]$	$[0.88; 1.80]$
db_D	Allemagne	12127	$[10; 141]$	$[10; 140]$	$[0.64; 4.61]$

TABLE 2.2 – Bases de données utilisées pour l'évaluation de la détection de panonceaux.



(a) Largeur (b) Hauteur (c) Ratios

FIGURE 2.21 – Histogrammes présentant le pourcentage de la base de données db_D en fonction de la largeur (a), hauteur (b) et ratio (c). La grande variabilité est principalement dûe à la taille apparente des panonceaux qui augmente au cours du temps et à la diversité des informations affichées.

2.5.2 Mesures utilisées

Pour évaluer et comparer les différentes méthodes, une vérité terrain a été réalisée à la main. Chaque panonceau présent sous un panneau de limitation de vitesse a été délimité et annoté. La figure 2.22 montre un exemple d'image obtenue avec notre outil de vérité terrain SamGT.

FIGURE 2.22 – Création de vérité terrain à l'aide de SamGT. Chaque objet d'intérêt, ici des panonceaux, est associé à une boîte englobante (en jaune) et un identifiant unique.

Pour des raisons de simplification, nous noterons GT la région d'intérêt spécifiée par la vérité terrain (en anglais, *Ground Truth*) et $ALGO$ le résultat obtenu par les différents algorithmes. Dans la plupart des recherches du domaine, l'évaluation est réalisée à l'aide de la mesure de Jaccard, notée $\mathscr{J}$. Elle correspond au rapport entre l'intersection et l'union de GT et $ALGO$ (figure 2.23).

$$\mathscr{J}(GT, ALGO) = \frac{GT \cap ALGO}{GT \cup ALGO} = \frac{I}{U} \tag{2.25}$$

Une forte valeur de $\mathscr{J}$ témoigne donc à la fois d'un bon recouvrement entre GT et $ALGO$ et d'un faible débordement.

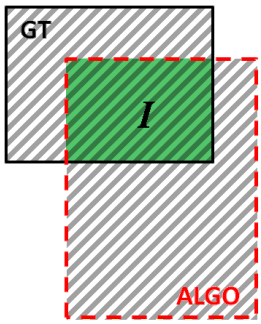


FIGURE 2.23 – Mesure de Jaccard $\mathscr{J}$. L'intersection $I = GT \cap ALGO$ correspond au recouvrement de la vérité terrain GT et de la région $ALGO$ fournie par la détection de rectangles. $U = GT \cup ALGO$ représente l'union des deux.

Cependant, cette mesure ne donne pas assez d'importance au centrage de $ALGO$ par rapport à GT. Pour une même valeur de $\mathscr{J}$, la qualité de la segmentation du point de vue du centrage peut être très différente (figure 2.24). La mesure de Jaccard seule ne nous permet alors pas de comparer correctement des algorithmes de détection de rectangles. En effet, il ne s'agit que de la première étape d'un système de reconnaissance de panonceaux. La classification qui suit a pour but

de supprimer les mauvaises détections et de déterminer le type des panonceaux restants. Conserver la zone centrale du panonceau, porteuse de l'information, est un indice important de qualité des méthodes comparées. Nous introduisons donc deux mesures supplémentaires, le centrage $\mathscr{C}$ et le recouvrement $\mathscr{R}$.

$$\mathscr{C} = \frac{distance(centre(GT), centre(ALGO))}{diag(GT)/2} \tag{2.26}$$

$$\mathscr{R} = \frac{GT \cap ALGO}{|GT|} \tag{2.27}$$

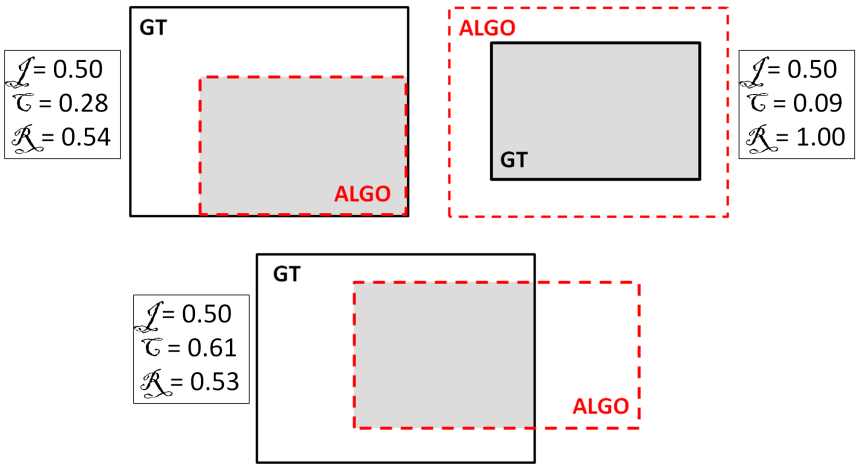


FIGURE 2.24 – Exemples de trois configurations différentes de GT et $ALGO$ qui conduisent à une même valeur de $\mathscr{J}$ mais différentes valeurs de $\mathscr{R}$ et $\mathscr{C}$. La combinaison des trois mesures permet d'avoir une meilleure vision de la qualité de la détection.

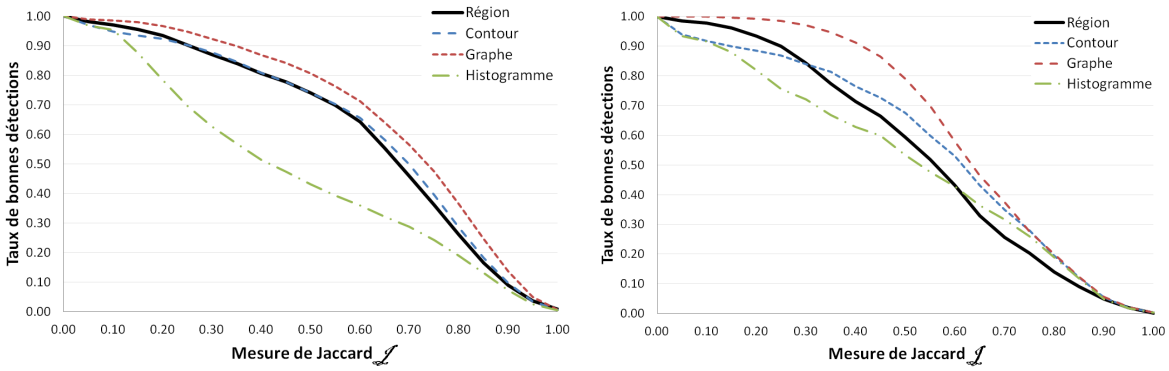
2.5.3 Analyse des résultats

Mesure de Jaccard

Dans un premier temps, nous comparons les techniques implémentées avec la mesure de Jaccard. Généralement, le taux de bonnes détections d'un algorithme correspond au nombre de régions telles que $\mathscr{J} \geq 0.5$. La figure 2.25 compare les différentes techniques d'après la mesure de Jaccard. Une valeur de $\mathscr{J}$ trop faible n'est pas discriminante car la majorité des régions trouvées auront un faible recouvrement avec la vérité terrain. En revanche, si $\mathscr{J}$ est proche de 1, la tolérance sur les régions détectées sera très faible, ce qui explique les mauvais taux.

La méthode à base de graphe apparaît comme la meilleure et dépasse de près de 10% toutes les autres sur la base allemande db_D. Près de 80% des panonceaux sont détectés pour une mesure de Jaccard de 0.5. La segmentation colorimétrique est de loin la plus mauvaise pour db_D, ce qui peut s'expliquer par le fait que les arrières-plans des panonceaux présentent une intensité très proche de ces derniers. Notre technique à base de croissance de régions ne se distingue pas ici. Elle conduit à des résultats similaires à celle à base de contours. Le taux de bonnes détections est de 75% pour $\mathscr{J} \geq 0.5$ et de 45% pour $\mathscr{J} \geq 0.7$.

Les performances sont plus mitigées pour la base française, où seulement 55% des panonceaux sont détectés pour $\mathscr{J} \geq 0.5$. Il est à noter que nous avons fixé les mêmes paramètres pour les deux bases de données. Il semble évident que notre méthode à base de croissance de régions nécessiterait



(a) Base de données db_D (b) Base de données db_F

FIGURE 2.25 – Comparaison des différentes techniques d'après la mesure de Jaccard $\mathscr{J}$ pour les deux bases de données. Les courbes représentent le taux de bonnes détections en fonction de $\mathscr{J}$. Les deux meilleures approches sont celles basées sur les graphes et les contours. La méthode à base de graphe de [Felzenszwalb and Huttenlocher, 2004] dépasse notre méthode pour la base française de presque 10%.

un jeu de paramètres par base contrairement à la méthode à base de graphe. Les résultats obtenus sont toutefois à modérer car la taille de db_F est dix fois plus faible que celle de db_D. La gamme de panonceaux rencontrés est de plus limitée au type "Flèche".

Centrage

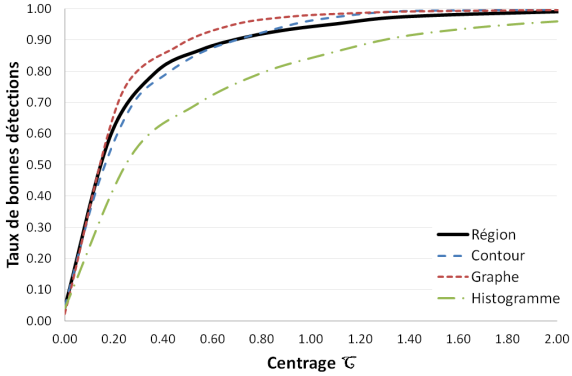
La figure 2.26 montre les performances des différents algorithmes en terme de centrage $\mathscr{C}$. L'essentiel de l'information étant situé au centre du rectangle, cet aspect présente une grande importance pour la phase de classification. Pour les techniques "Graphe", "Région" et "Contour", plus de 90% des régions vérifient $\mathscr{C} \leq 1$ sur les deux bases de données, ce qui signifie que leur centre est distant de moins d'une demie-diagonale de celui de la vérité terrain.

En terme de centrage, la technique à base de graphe montre une fois de plus sa supériorité. Pour 98% des panonceaux, nous avons $\mathscr{C} \leq 1$, résultat prometteur pour la phase de classification qui suit. La méthode à base de régions donne, ici aussi, des résultats semblables à celle à base de contours. Nous aurions pu prétendre à bien mieux puisque les graines utilisées sont obtenues à partir des zones contrastées, situées au centre des panonceaux. Cependant, les valeurs des paramètres ont conduit à un débordement trop important, et surtout non uniforme, des régions initiales.

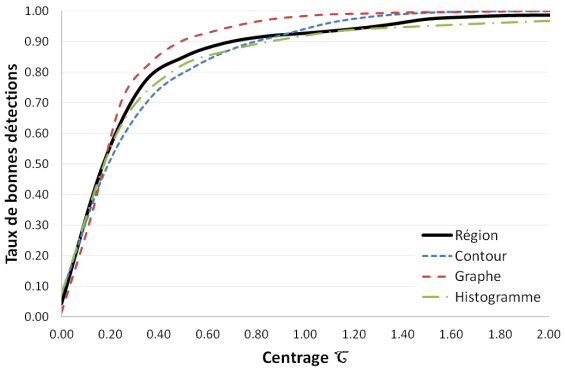
Recouvrement

Du point de vue du recouvrement $\mathscr{R}$, les méthodes "Graphe" et "Région" se classent en tête pour la base allemande (figure 2.27). 90% des rectangles ont un taux de recouvrement supérieur à 50%, ce qui, compte-tenu des résultats de centrage, augure de bons candidats pour la classification. Pour la base française en revanche, le taux de bonnes détections de "Région" chute rapidement et 75% seulement des rectangles ont un recouvrement supérieur à 50%.

Les méthodes "Graphe" et "Contour" présentent des résultats plus homogènes entre les deux bases avec une très nette avance pour la première. Du point de vue statistique, la base allemande est toutefois plus fiable car la variété de panonceaux est beaucoup plus grande, tant du point de vue de l'aspect géométrique (taille, ratio, etc.) que des conditions climatiques.

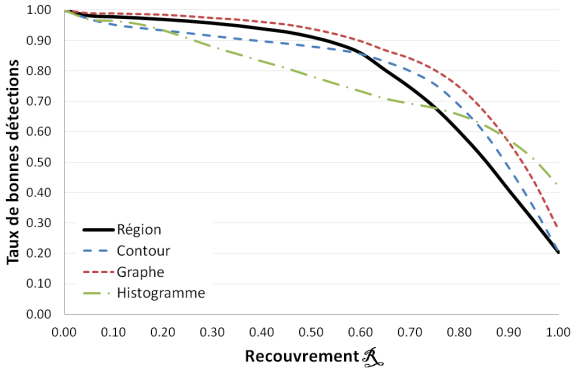


(a) Base de données db_D

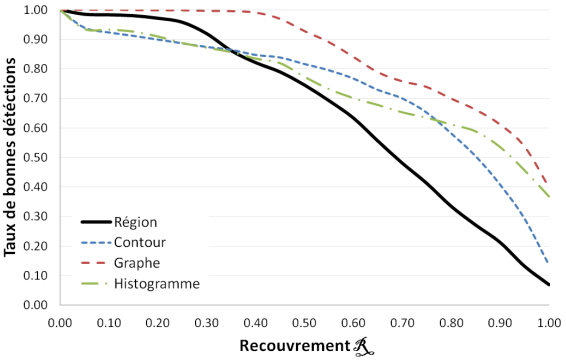


(b) Base de données db_F

FIGURE 2.26 – Comparaison des différentes techniques d'après la mesure de centrage $\mathscr{C}$ pour les deux bases de données. Les courbes représentent le taux de bonnes détections en fonction de $\mathscr{C}$. La technique à base de graphe remporte la première place pour les deux bases de données. Pour plus de 90% des rectangles détectés, nous avons $\mathscr{C} \leq 1$, indiquant que leur centre n'est distant de celui de la vérité terrain que d'une demie-diagonale ou moins. Notre méthode "Région" atteint 94% pour $\mathscr{C} \leq 1$ pour la base db_D.



(a) Base de données db_D



(b) Base de données db_F

FIGURE 2.27 – Comparaison des différentes techniques d'après la mesure de recouvrement $\mathscr{R}$ pour les deux bases de données. Les méthodes "Graphe" et "Région" arrivent en tête pour la base allemande avec près de 90% des rectangles tels que $\mathscr{R} > 0.5$.

Étude des faux positifs

Nous allons maintenant comparer les méthodes "Graphe" et "Région" en terme de faux positifs sur la base de données db_D. Les courbes Rappel en fonction du nombre de Faux Positifs Par Positif (FPPP) sont présentées dans la figure 2.28 et sont obtenues en faisant varier un critère de *rectangularité*, *i.e.* le rapport entre l'aire réelle de la région et celle de sa boîte englobante. Les bonnes détections correspondent aux régions telles que $\mathscr{J} \geq 0.5$. Du point de vue du rappel, les résultats corroborent ceux de la section 2.5.3 et la méthode "Graphe" surpasse "Région" de presque 15%. En revanche, le nombre de faux positifs est 1.5 fois plus grand. Cet écart s'explique par le fait que notre approche ne segmente une image que si des zones fortement contrastées, dont les panonceaux font partie, ont été détectées. Cela réduit le nombre d'opérations et de faux positifs *a fortiori*. L'approche "Graphe" est opérée pour toutes les images envoyées en entrée du système.

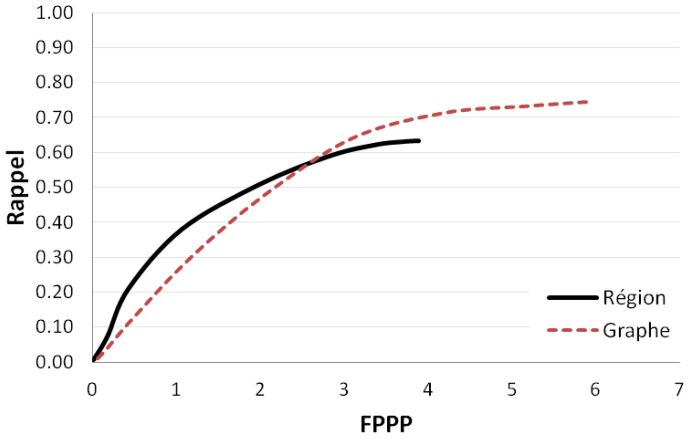


FIGURE 2.28 – Rappel en fonction du nombre de Faux Positifs Par Positif (FPPP) pour les méthodes "Graphe" et "Région" obtenu en faisant varier la *rectangularité* des régions des algorithmes et pour la base de données db_D. Une bonne détection correspond à une mesure de Jaccard $\mathscr{J} \geq 0.5$. La méthode "Graphe" est supérieure à "Région" en terme de Rappel de près de 15%, ce qui est en accord avec les observations précédentes (section 2.5.3). Néanmoins, le nombre de faux positifs est 1.5 fois plus grand. La méthode "Région" ne segmente que les zones détectées comme fortement contrastées alors que celle à base de graphes recherche dans toutes les images.

Les phases de post-traitement n'auront pas les mêmes contraintes en fonction de la méthode choisie. Dans le cas de la méthode "Graphe", il faut être à même d'éliminer un maximum de faux positifs lors de la classification pour éviter les fausses alarmes, d'où une bonne précision. Il faut aussi garder à l'esprit que segmenter toutes les images conduit à un temps de calcul "incompressible" pour la phase de détection. Pour l'approche "Région", la reconnaissance doit surtout avoir un bon rappel pour ne pas rejeter les panonceaux correctement détectés.

Robustesse aux dégradations d'image

Nous nous sommes ensuite intéressés à la robustesse des algorithmes "Graphe" et "Région" aux changements de luminosité L et contraste C sur la base de données db_D. Pour cela, nous avons artificiellement dégradé les images en faisant varier ces deux paramètres. Nous sommes alors à même d'étudier le comportement du système dans des conditions pouvant survenir à différents moments de la journée (pénombre, contre-jour, etc.). La figure 2.29 illustre ces dégradations volontaires sur un exemple. Au premier abord, la reconnaissance de panonceau semble plus délicate lorsque la luminance est faible et le contraste fort, l'image étant trop sombre pour y distinguer quoi que ce soit.

Les performances des deux algorithmes "Graphe" et "Région" sont présentées dans la figure 2.30 et correspondent au taux de bonnes détections au sens $\mathscr{J} \geq 0.5$. La qualité originale de l'image correspond au point (100,100). L'approche "Graphe" semble assez robuste aux dégradations imposées avec plus de 70% de bonnes détections pour une assez grande plage de valeurs. Pour $C < 70$, la luminance ne semble pas jouer sur les résultats et pour de faibles valeurs de L, l'algorithme ne détecte que 10% des panonceaux.

Les meilleures performances de la méthode "Région" se situent, quant à elles, autour de la diagonale mais se détériorent ensuite rapidement. Les régions recherchées devant présenter un fort contraste, les résultats obtenus lorsque C diminue s'expliquent. De même, lorsque la luminance

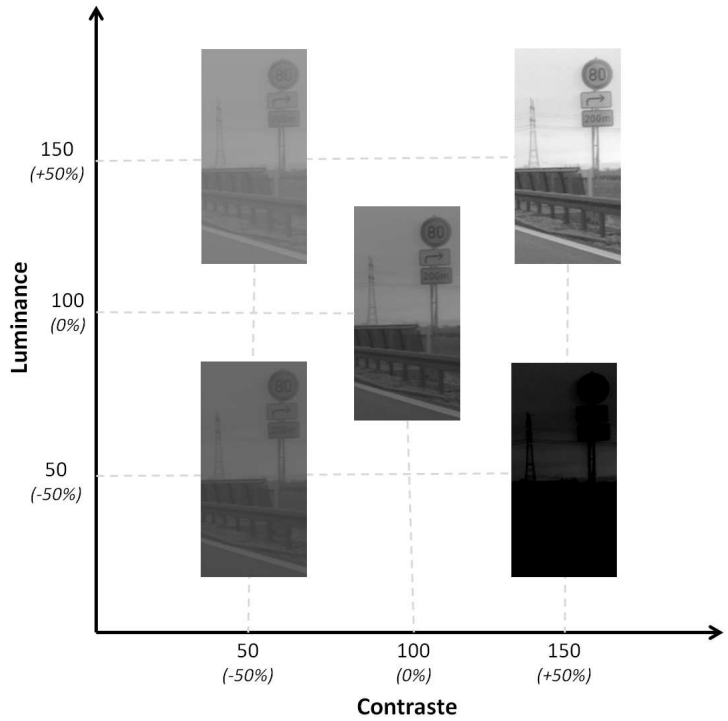


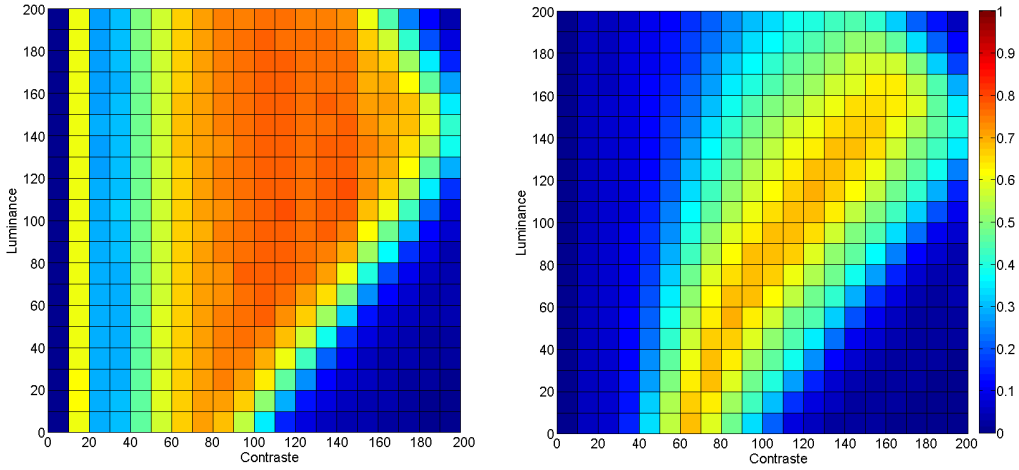
FIGURE 2.29 – Exemples d'une image de panonceau dégradée pour différentes valeurs de contraste et de luminance. L'image originale se situé en (100,100). Visuellement, lorsque le contraste est fort et la luminosité faible, la détection semble compromise car le contraste est réévalué sur l'image assombrie et le panonceau se fond complètement dans l'arrière-plan.

est faible, nous avons vu que les objets étaient difficilement visibles pour un œil humain. La comparaison des deux méthodes donne donc l'avantage à la technique "Graphe" car sa robustesse aux dégradations est plus grande. En revanche, il faut garder à l'esprit que lorsque l'image est trop altérée, la reconnaissance du panonceau devient complexe.

2.6 Conclusion

Dans cette partie, nous avons présenté une méthode inédite de détection de rectangles à contenu informatif. Un algorithme de reconstruction morphologique est utilisé pour ne sélectionner que les régions fortement contrastées de l'image. Une croissance de régions est alors opérée à partir de ces graines afin d'extraire les panonceaux. Cette approche a été évaluée et comparée à un benchmark de trois techniques de segmentation basées sur les frontières, la colorimétrie et les graphes. La majorité d'entres elles n'ont pas fait l'objet d'une mise en œuvre dédiée à ce domaine d'application et ont dûes être adaptées à notre problématique.

Plusieurs mesures ont été réalisées pour confronter toutes ces méthodes, de la mesure de Jaccard, communément utilisée, au centrage et recouvrement. De manière générale, la méthode à base de graphes, inspirée de [Felzenszwalb and Huttenlocher, 2004] se distingue nettement des autres et donne des résultats très prometteurs. Cependant, le nombre de faux positifs générés est important par rapport à notre méthode "Région". Cette dernière présente des résultats corrects, certes moins bons que "Graphe", et une précision supérieure. En limitant la segmentation aux zones contrastées, nous réduisons en effet le nombre de fausses alarmes et, nous l'espérons, le temps de calcul d'une



(a) Méthode "Graphe" (b) Méthode "Région"

FIGURE 2.30 – Performances des méthodes (a) "Graphe" et (b) "Région" pour différentes valeurs de luminance L et de contraste C pour la base de données db_D. L'approche basée "Graphe" apparaît beaucoup plus robuste aux changements de conditions extérieures que notre technique "Région". La plage de valeurs pour lesquelles les résultats dépasse 50% de bonnes détections est assez importante pour la première alors que pour la seconde elle se situe autour de la diagonale. Pour de faibles contrastes ou luminances, les panonceaux sont toutefois nettement moins visibles même pour un humain.

technique, somme toute assez complexe.

Une vision plus globale des avantages et inconvénients des techniques proposées dans le cadre de la reconnaissance de panonceaux sera obtenue après combinaison avec un classifieur. L'objectif final étant de développer un système robuste et fiable, la comparaison finale des méthodes devra tenir compte des aspects pratiques liés à un système complet. Tous les panonceaux physiques seront-ils détectés et correctement classifiés ?

Chapitre 3

Reconnaissance de panonceaux

3.1 Introduction

L'étape de détection de rectangles a permis d'extraire de l'image les régions s'approchant le plus d'un panonceau. Toutefois, cet algorithme est loin d'être parfait. Ces candidats doivent maintenant être filtrés pour ne retenir que les panonceaux réels et supprimer toutes les fausses alarmes qui pourraient alerter inutilement le conducteur. L'objectif de la reconnaissance est ainsi double, éliminer les négatifs et classifier les panonceaux pour déchiffrer l'information qui y est contenue. Quelques exemples de panonceaux pouvant être placés sous les panneaux de limitation de vitesse sont présentés dans la figure 3.1. Idéalement, le système sera à même de distinguer les flèches des camions voire de lire les informations textuelles.

Dans ce chapitre, nous commençons par présenter de manière générale les différents types de classifieurs et de descripteurs, puis l'existant dans le domaine plus spécifique des panonceaux. Ensuite, nous décrivons notre méthode de reconnaissance basée sur une architecture en arbre et un mode de représentation globale du panonceau. Enfin, nous évaluons les performances de notre algorithme tout en justifiant nos choix.

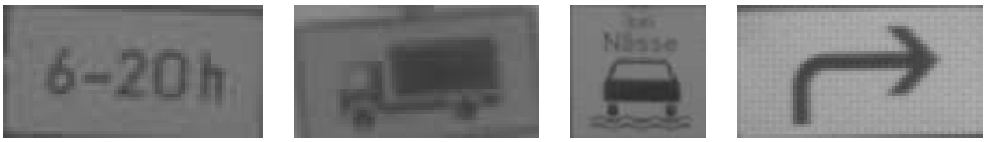


FIGURE 3.1 – Exemples de panonceaux de différents types pouvant être rencontrés sous les panneaux de limitation de vitesse.

3.2 Classification

3.2.1 Qu'est-ce que la classification ?

L'objectif est de définir une règle permettant d'assigner tout motif inconnu $\mathbf{x}$ d'un certain espace X à une classe donnée l à partir de connaissances *a priori*. Ces dernières sont fournies sous forme d'une base d'apprentissage, représentative de la tâche de classification. Elle représente la mémoire, ou l'expérience, du système. Elle doit notamment contenir un ensemble significatif de N motifs $\mathbf{x_i}, i \in [1, N]$ de chacune des catégories à distinguer. Si chaque motif $\mathbf{x_i}$ a été au préalable associé à classe, étiquetée l_i, l'algorithme de classification vise à séparer au mieux l'espace entre

les L classes. L'apprentissage est dit *supervisé*. En revanche, si cette distribution est inconnue, il est dit *non-supervisé* et l'objectif est de trouver le nombre optimal de classes permettant la meilleure séparation au sens de critères donnés.

La qualité du classifieur peut être appréciée selon trois aspects particulièrement importants :
- une représentation des motifs alliant **simplicité** et **discriminabilité**.
  Un motif sera représenté par un ensemble de k caractéristiques qui contiennent toute son information intrinsèque.
- une séparation de l'espace entre les classes telle qu'elle **maximise la distance inter-classes et minimise la distance intra-classe**.
  Plus une classe sera compacte dans l'espace de représentation, plus l'estimation de sa frontière sera facilitée. De même, si deux classes se superposent, il sera plus difficile, voire impossible, d'assigner un motif à une classe plutôt qu'une autre sans risque d'erreur.
- une bonne capacité à estimer les classes des exemples inconnus avec le minimum d'erreur, ou **capacité de généralisation**.

[Jain et al., 2000] proposent de décomposer les techniques de reconnaissance en quatre familles : approche syntaxique, *template-matching*, réseau de neurones et les autres techniques d'apprentissage statistique. Bien que la distinction entre les deux dernières catégories soit discutable, nous reprenons cette séparation ci-dessous.

Approche syntaxique

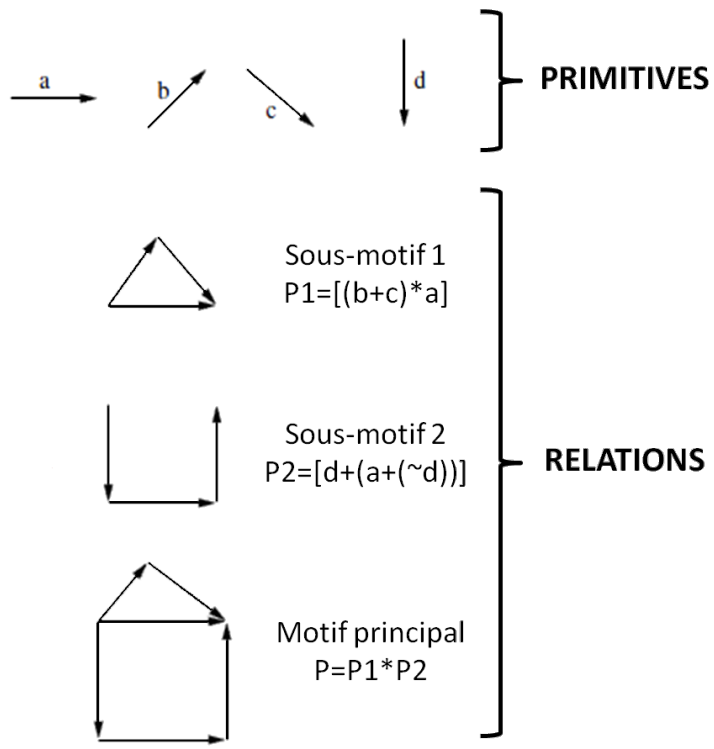


FIGURE 3.2 – Exemple d'une décomposition d'un motif en primitives et relations (source : [Bunke, 1990]).

Dans l'approche syntaxique, chaque motif peut être décomposé en un ensemble de primitives, sous-motifs élémentaires, et de relations les unissant [Bunke, 1990, Chanda and Dellaert, 2004]. La représentation peut se faire sous forme d'arbres ou de graphes. Les primitives sont à l'origine de tous les motifs et forment un dictionnaire et les relations, la syntaxe. Pour une image par exemple, les primitives peuvent être les coins, les contours, etc. La figure 3.2 illustre la décomposition d'un motif en primitives et relations.

Cette approche est particulièrement adaptée aux situations où les motifs ont une structure très présente, comme l'analyse des encéphalogrammes, constitués d'un ensemble de signaux élémentaires [Wang et al., 2001], mais aussi la reconnaissance d'objets [Lin and Fu, 1986, Lagunovsky and Ablameyko, 1997, Gdalyahu and Weinshall, 1999]. Toutefois, les performances de tels systèmes sont fortement altérées par la présence de bruit qui rend plus difficile la détection de primitives. Les règles grammaticales peuvent également être très complexes à définir.

Template-matching

Le *template-matching* repose sur deux principes fondamentaux, des modèles explicites représentent la connaissance sur les classes et la reconnaissance est effectuée par correspondance, les exemples étant comparés aux modèles de la base. La génération des modèles doit être réalisée de manière à avoir un système suffisamment robuste aux changements d'illumination, de forme, d'aspect, et plus généralement à tout ce qui relève de la variabilité de chaque classe. Ils doivent permettre d'approcher au mieux la distribution réelle des motifs. La mesure de similarité correspond à la distance d'un exemple inconnu à un modèle de la base. Il s'agit le plus souvent d'une mesure de corrélation. Un exemple simple consiste à prendre la liste des pixels comme représentation d'une image et de calculer une distance euclidienne avec un vecteur de la base. Plus cette valeur sera faible, plus grande sera la similitude.

Cette méthode, assez simple en apparence, est complexe à mettre en pratique. La base de données des modèles doit être complète et la méthode de recherche exhaustive. Le coût de calcul est alors rapidement prohibitif. Elle semble plus appropriée aux problématiques de ré-identification pour lesquels modèles et exemples sont semblables. C'est le cas notamment des systèmes de Reconnaissance Optique de Caractère [Mori et al., 1992] ou de détection de panneaux routiers [Paclik et al., 2006, Piccioli et al., 1996, Gavrila, 1999]. Dans [Gavrila, 1999], la transformée de distance est utilisée pour représenter les panneaux et les comparer avec les modèles. Pour augmenter la vitesse, une approche multirésolutions est mise en œuvre à l'aide d'une hiérarchie de modèles de différentes tailles (figure 3.3).

Réseaux de neurones

Introduits par [McCulloch and Pitts, 1943], ces modèles de calcul s'inspirent du fonctionnement des neurones biologiques. Apparentés aux techniques d'intelligence artificielle, ils permettent de gérer des problèmes complexes, pour la résolution desquels un raisonnement logique ne suffirait pas. Le mécanisme de décision s'appuie davantage sur la perception et la mémoire. La structure est composée d'un ensemble d'unités de traitement interconnectées, les neurones, qui communiquent entre elles au moyen de signaux.

Un réseau de neurones est constitué de cellules de trois types : celles de la couche d'entrée N_i reçoivent les informations provenant de l'extérieur, celles de la couche de sortie N_o envoient les données hors du réseau, et entre les deux les cellules de la couche cachée N_h (figure 3.4). Plus

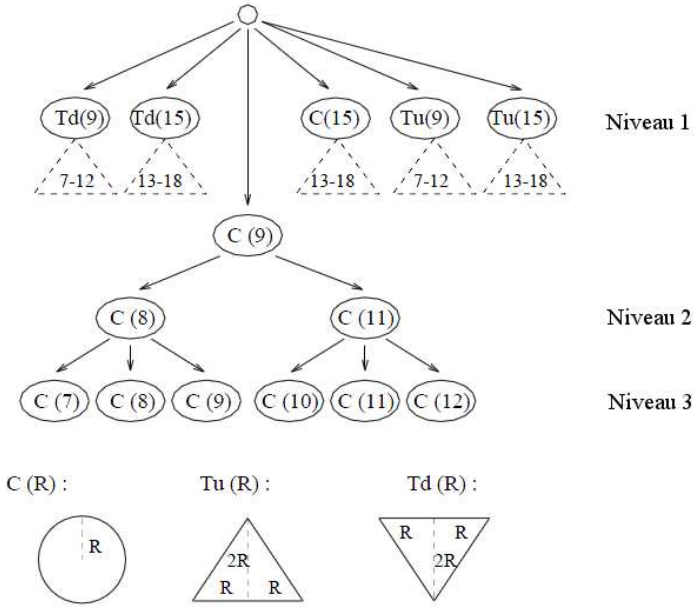


FIGURE 3.3 – Hiérarchie de modèles multi-échelles utilisée par [Gavrila, 1999] afin d'améliorer les performances du *template-matching* pour la reconnaissance de panneaux. Au premier niveau de résolution, des motifs grossiers sont utilisés. En cas de bonne détection, une granularité plus fine permet de raffiner le résultat et d'avoir une meilleure estimation du rayon de l'objet. Le cercle $C(9)$ du premier niveau correspond au second niveau à deux motifs $C(8)$ et $C(11)$ et au niveau le plus fin à un ensemble de cercles de rayons compris entre 7 et 12.

précisément, une unité de traitement k reçoit des données y_j de N cellules en amont (figure 3.5). Les connexions sont pondérées d'un facteur w_{jk} déterminant l'effet du signal de l'unité j sur l'unité k. Le potentiel s_k de k à un instant t est ainsi calculé comme une somme pondérée de ses entrées $y_j(t)$ et d'un biais $\theta_k(t)$, définie comme étant la règle de propagation du réseau.

$$s_k(t) = \sum_j w_{jk} y_j(t) + \theta_k(t) \tag{3.1}$$

La sortie de l'unité est ensuite déterminée à l'aide d'une fonction d'activation $\mathscr{F}_k$.

$$y_k(t+1) = \mathscr{F}_k(s_k(t)) \tag{3.2}$$

La plupart du temps, les fonctions d'activation choisies sont des sigmoïdes (comme la tangente hyperbolique) ou des seuillages.

L'apprentissage consiste à adapter les poids des connexions en fonction des données et de la topologie du réseau. Si l'information entrant dans une unité ne provient que des neurones situés en amont, le réseau est **sans rebouclage**. Le modèle du perceptron introduit par [Rosenblatt, 1959] appartient à cette catégorie. Pour adapter les poids, l'apprentissage des réseaux neuronaux utilise une méthode de gradient visant à minimiser l'écart entre les sorties obtenues et celles correspondant aux données. Cette descente de gradient se fait généralement à l'aide d'une astuce de calcul, nommée **rétropropagation du gradient** [Rumelhart et al., 1986]. Le principe est de propager en arrière l'erreur commise en sortie sur toutes les connexions des unités ayant pris part

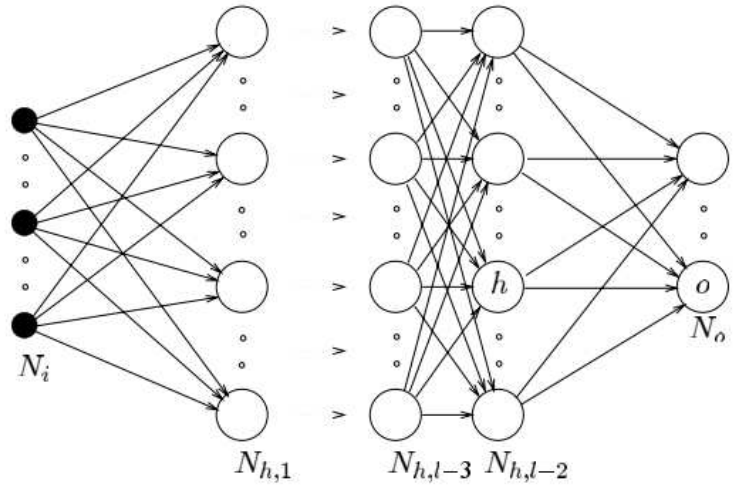


FIGURE 3.4 – Réseau de neurones à $l-2$ couches cachées (source : [Kröse and van der Smagt, 1996]).

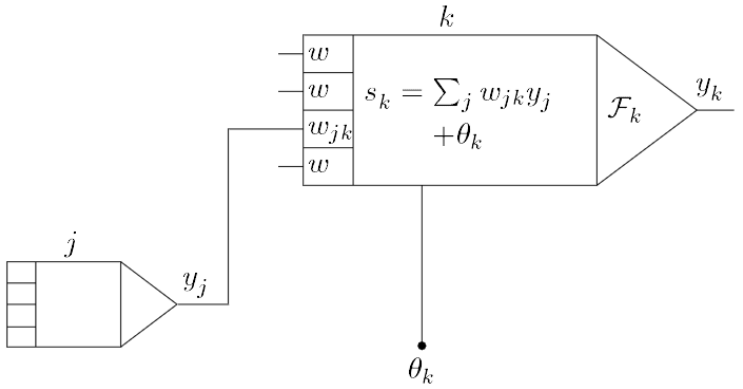


FIGURE 3.5 – Unité de traitement k d'un réseau de neurones. Les données reçues des cellules en amont y_j et le biais θ_k sont pondérés par les coefficients w_{jk}. Le niveau de sortie y_k est fonction de cette somme s_k et d'une fonction d'activation $\mathscr{F}_k$ (source : [Kröse and van der Smagt, 1996]).

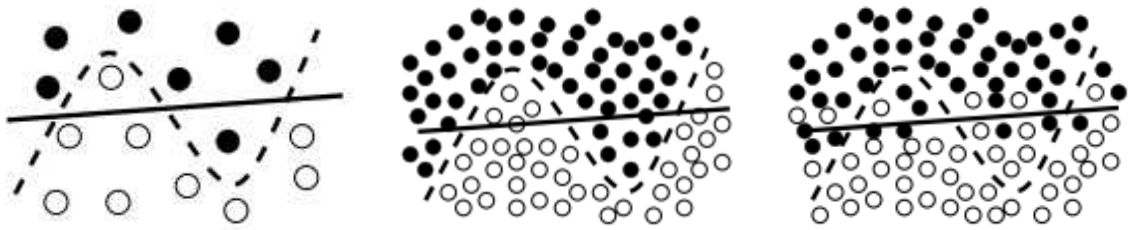
à cette décision. L'importance de la correction dépend de l'influence de l'élément dans la prise de décision.

Les domaines d'application sont aussi variés que les types de réseaux eux-mêmes, dépendant du choix de la cellule élémentaire, de l'architecture du réseau aussi bien que de sa dynamique. Les avantages de tels systèmes sont leur simplicité et leur rapidité de traitement. Le réseau est qualifié de "boîte noire", la classe d'appartenance de la donnée d'entrée étant directement calculée par celui-ci. L'inconvénient est que ce résultat, de fait, n'est plus interprétable par l'utilisateur. De plus, la rapidité du système est contrebalancée par la durée de la phase d'apprentissage qui peut être longue. Même si les réseaux neuronaux sont historiquement plus anciens que d'autres méthodes d'apprentissage comme les SVMs (*Support Vector Machines* - Séparateurs à Vaste Marge) et leur processus d'apprentissage moins déterministe et plus délicat, ils présentent l'avantage de permettre de traiter très facilement des problèmes multiclasses (en mettant par exemple un neurone de sortie par classe). C'est l'une des raisons, ajoutée à leur simplicité, pour laquelle ils demeurent un outil

apprécié dans les domaines impliquant des systèmes intelligents, comme les systèmes d'aide à la conduite. Dans le cadre de la détection de panneaux, de nombreuses études mettent en œuvre des réseaux de neurones pour la phase de classification [Gavrila, 1999, Garcia-Garrido et al., 2006]. Elles s'appuient généralement sur des démarches holistiques basées sur la reconnaissance du panneau dans son ensemble. L'invariance par rotation, translation ou changement d'échelle n'est donc par assurée et les performances risquent d'en être altérées. Les solutions proposées les plus classiques sont d'intégrer dans la base d'apprentissage des images déformées [de la Escalera et al., 1997], de redresser l'image au préalable [Escalera and Radeva, 2004] ou de décrire l'image à l'aide de caractéristiques possédant ces invariances, comme la Transformée de Fourier [Kang et al., 1994].

Autres techniques d'apprentissage statistique

Les frontières peuvent également être déterminées via les distributions de probabilité des motifs pour chaque classe. Dans le cas d'un problème à L classes $\mathscr{L} = \{l_i\}, i \in [1, L]$ pour lequel les motifs sont représentés par un vecteur de k caractéristiques, il s'agit d'estimer une fonction $f : \mathbb{R}^k \to \mathscr{L}$ à partir de la base d'apprentissage qui permet de classifier correctement des motifs inconnus. La principale difficulté réside dans la recherche d'un compromis entre la complexité du classifieur et sa capacité de généralisation. Il pourrait être tentant en effet de trouver la meilleure frontière au vu de la base d'apprentissage, qui représente, seule, la vision du monde pour le classifieur. Une erreur empirique minimale sur cette base conduit cependant souvent à un modèle complexe et finalement mal adapté au monde réel, d'où un risque de **sur-segmentation**. À l'inverse, un modèle trop simplifié ne sera pas efficace et présentera une erreur de classification importante. C'est la **sous-segmentation**. La figure 3.6 présente ces deux problèmes.



(a) Deux types de frontières obtenues après apprentissage

(b) Sous-segmentation

(c) Sur-segmentation

FIGURE 3.6 – Illustration du dilemme de complexité (source : [Müller et al., 2001]). (a) Les deux frontières obtenues après apprentissage présentent une faible erreur de classification bien que celle en pointillé soit nettement plus complexe que l'autre. (b) La base de test s'adapte beaucoup mieux dans ce cas à la frontière en pointillé. La seconde a donc sous-estimé la distribution réelle des exemples, d'où la sous-segmentation. (c) En revanche, pour une autre base de données, c'est la frontière en pointillé qui n'est pas adaptée et qui, en cherchant une erreur empirique minimale, s'est montrée trop rigide. L'erreur de généralisation est donc grande et le modèle nettement plus complexe que la réalité, c'est la sur-segmentation.

Pour estimer la frontière entre les différentes classes, il est possible d'utiliser toute la base de données ou seulement un sous-ensemble représentatif.

Dans la première catégorie, nous retrouvons les techniques de **plus proche voisin** comme les arbres de décision [Quinlan, 1986] ou les forêts aléatoires (*Random Forests*) [Breiman, 2001].

Un arbre K-d est un arbre de recherche binaire permettant de séparer des motifs représentés par un vecteur de caractéristiques de dimension K. Chaque nœud, hormis les feuilles, sépare les données en deux selon la i-ème caractéristique f_i dont la variance pour le sous-ensemble concerné est maximale. Pour s'assurer d'obtenir un arbre équilibré, la médiane m_i est utilisée telle que le sous-arbre gauche vérifie la propriété $f_i < m_i$ et inversement à droite. L'arbre K-d est un algorithme de recherche du plus proche voisin. Un exemple est en effet classifié en descendant l'arbre jusqu'à atteindre une feuille, c'est-à-dire une classe donnée (figure 3.7). Une des premières limitations de ce type d'arbre est sa sensibilité au bruit, une simple erreur à un embranchement et la classification est fausse.

Les forêts aléatoires proposent une solution à ce problème en générant un ensemble d'arbres aléatoires. Aléatoires dans le sens où :

- un sous-ensemble de la base d'apprentissage est choisi aléatoirement avec remise.
- à chaque nœud, un sous-ensemble de caractéristiques est choisi. Un compromis sur sa taille doit être trouvé entre un nombre trop élevé qui conduirait à une sur-segmentation et un trop faible qui augmenterait la vitesse d'apprentissage mais aussi le risque de sous-segmentation.

La classification d'un exemple inconnu est réalisée en combinant les probabilités *a posteriori* obtenues pour chaque arbre.

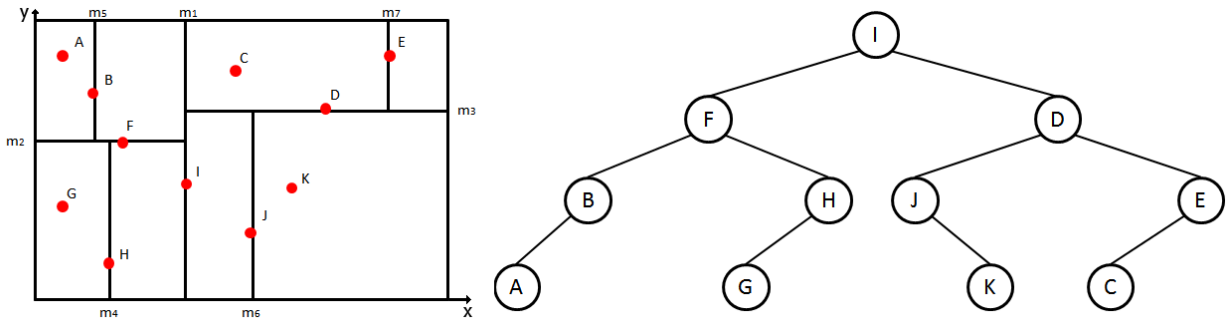


FIGURE 3.7 – Illustration de la construction d'un arbre K-d. L'espace des données est successivement découpé selon les axes x et y. Le nœud racine est l'exemple I car il maximise la variance. La construction s'arrête lorsque tous les exemples ont été utilisés.

De par leur facilité d'utilisation, les forêts aléatoires sont employées dans bon nombre d'applications. Elles sont bien adaptées dans le cas de bases de données déséquilibrées et permettent même un classement des caractéristiques par importance. [Bosch et al., 2007a] les utilisent pour la classification d'images et mettent en avant leur coût algorithmique faible. Une équipe de notre laboratoire a proposé une méthode de classification de panneaux routiers [Zaklouta et al., 2011] avec un taux de reconnaissance de 97%.

La seconde catégorie ne prend en compte qu'un **sous-ensemble représentatif** de la base de données pour établir les frontières. Moins sensibles aux *outliers* (données aberrantes), ces algorithmes présentent une meilleure capacité de généralisation en cherchant plutôt les tendances suivies par les données.

Nous y trouvons les techniques Adaboost [Freund and Schapire, 1997] ou les SVMs [Boser et al., 1992]. Ces derniers sont des classifieurs binaires. Considérons un ensemble de N exemples linéairement séparables $x_i \in \mathbb{R}^k$ étiquetés $l_i \in \{-1; 1\}$. La séparation entre les deux classes est

modélisée par un hyperplan H :

$$H : w\mathbf{x} + b = 0 \tag{3.3}$$

La distance minimale entre H et l'origine vaut donc $\frac{|b|}{\|w\|}$ (figure 3.8). Les points les plus proches de l'hyperplan pour chaque classe sont appelés les **vecteurs supports** et sont situés à une distance $d-$ (resp. $d+$) pour la classe -1 (resp. +1). Ils forment deux hyperplans $H+$ et $H-$ choisis de sorte que leur distance $\frac{2}{\|w\|}$ soit maximale. Les performances de la classification sont optimales lorsque la marge est maximale, ce qui revient à minimiser $\frac{\|w^2\|}{2}$. Ce problème peut être résolu via les multiplicateurs de Lagrange $\alpha_i \geq 0$.

$$L = \frac{\| w^2 \|}{2} - \sum_{i=1}^{m} \alpha_i l_i (w\mathbf{x_i} + b) - 1 \tag{3.4}$$

La fonction de classification pour un exemple inconnu x est alors :

$$f(\mathbf{x}) = sgn(w\mathbf{x} + b) = sgn(\sum_{i=1}^{m} \alpha_i l_i (\mathbf{x_i}.x) + b) \tag{3.5}$$

Les exemples de la base d'apprentissage vérifient alors :

$$\begin{cases} w\mathbf{x_i} + b \geq 1 & \text{si} \quad l_i = 1 \quad \Rightarrow l_i(w\mathbf{x_i} + b) - 1 \geq 0 \\ w\mathbf{x_i} + b \leq -1 & \text{si} \quad l_i = -1 \end{cases} \tag{3.6}$$

En présence de bruit, le SVM peut être rendu plus tolérant aux exemples mal classifiés par l'ajout de variables "molles" ξ_i.

$$\begin{cases} w\mathbf{x_i} + b \geq 1 - \xi_i & \text{si} \quad l_i = 1 \\ w\mathbf{x_i} + b \leq -1 + \xi_i & \text{si} \quad l_i = -1 \end{cases} \tag{3.7}$$

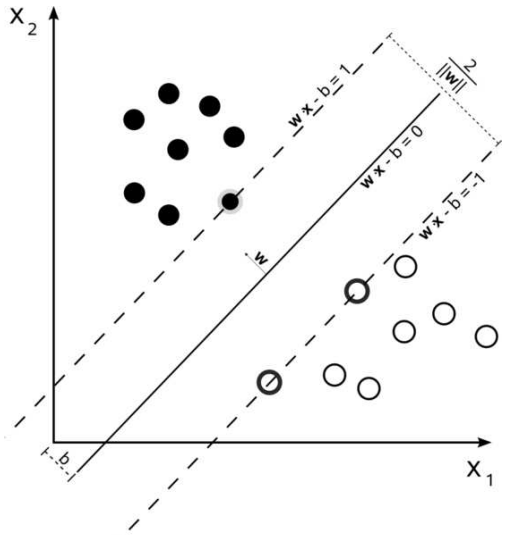


FIGURE 3.8 – Séparation linéaire des données de la base d'apprentissage par l'hyperplan $H : w\mathbf{x} + b = 0$ (source : [Wikipedia, 2008]). Les vecteurs de support sont les points ayant une distance minimale avec H.

Toutefois, les données ne sont pas nécessairement séparables de façon linéaire. Elles sont alors transformées dans un espace de dimension supérieure grâce à une fonction non linéaire $z = \Phi(\mathbf{x})$.

La séparation linéaire sera effectuée dans ce nouvel espace. Pour éviter le calcul explicite de Φ, le produit scalaire $\Phi(\mathbf{x_i}).\Phi(\mathbf{x_j})$ est remplacé par un noyau K. Pour déterminer l'étiquette d'un exemple x , nous étudions le signe de $w\Phi(\mathbf{x}) + b$.

$$f(\mathbf{x}) = sgn(w\Phi(\mathbf{x}) + b) \tag{3.8}$$

$$= sgn(\sum_{i=1}^{m} \alpha_i l_i(\Phi(\mathbf{x_i}).\Phi(\mathbf{x})) + b) \tag{3.9}$$

$$= sgn(\sum_{i=1}^{m} \alpha_i l_i K(\mathbf{x_i}, \mathbf{x}) + b)) \tag{3.10}$$

Parmi les noyaux les plus utilisés, nous retrouvons :
- le noyau linéaire $K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i}\mathbf{x_j}$.
- le noyau polynômial $K(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i}\mathbf{x_j} + r)^d \quad \gamma \in \mathbb{R}^+, r \in \mathbb{R}, d \in \mathbb{N}$.
- le noyau gaussien $K(\mathbf{x_i}, \mathbf{x_j}) = exp(-\gamma \parallel \mathbf{x_i} - \mathbf{x_j} \parallel^2)$.

Pour gérer les problèmes multiclasses, il existe deux solutions, utiliser des SVMs de type *un-contre-un* ou *un-contre-tous*. Dans le premier cas, chaque classe est modélisée contre une autre, c'est-à-dire que tous les éléments de l_i sont considérés comme positifs alors que tous ceux de la classe l_j sont vus comme négatifs. Pour N classes, il y aura au final $N(N-1)/2$ classifieurs. Un exemple inconnu sera associé à la classe présentant la confiance la plus forte. Dans le second cas, une classe est cette fois séparée de toutes les autres.

Le principal avantage des SVMs est leur capacité de généralisation, ainsi que la relative rapidité de leur apprentissage, et le déterminisme de ce dernier. Ils cherchent à minimiser la limite sur l'erreur de généralisation du modèle plutôt que minimiser l'erreur sur la base d'apprentissage. Ainsi, les exemples d'apprentissage situés loin de l'hyperplan ne changeront pas les vecteurs de support, indiquant par là une meilleure classification d'exemples non rencontrés. Ils sont de plus peu sensibles à la dimension de l'espace des caractéristiques, contrairement aux réseaux de neurones et la classification est possible même avec des bases de données de petite dimension. Ils sont utilisés dans de nombreuses applications, reconnaissance d'écriture manuscrite, classification de texte, détection d'objet, etc.

En résumé, le choix du classifieur semble beaucoup moins critique que celui des caractéristiques utilisées pour décrire les objets à classifier. Il dépend essentiellement de la base de données (taille, ratio négatifs/positifs), du type d'application, des performances attendues et des connaissances de l'expert. Les approches syntaxiques nécessitent l'élaboration de règles grammaticales complexes, pour chaque catégorie d'objets. Elles sont donc difficiles à mettre en œuvre pour un problème multiclasses. Les approches par *template-matching* semblent plus adaptées à la ré-identification et au suivi. Enfin, les forêts aléatoires offrent un bon compromis performances/temps avec toutefois de forts besoins en mémoire et un risque de sur-segmentation tandis que les SVMs atteignent de meilleures performances en général au prix d'un coût algorithmique non négligeable [Meyer et al., 2003].

3.2.2 Comment représenter les motifs ?

Une des parties les plus critiques dans la reconnaissance consiste à extraire de chaque motif un ensemble de caractéristiques propres à sa classe et la distinguant de façon nette des autres.

L'objectif est d'augmenter ainsi la variance inter-classes tout en diminuant la variance intra-classe afin d'obtenir une représentation plus compacte du motif. En prenant tous les pixels d'une image comme vecteur de caractéristiques, la complexité est de $O(NM)$ avec (N, M), les dimensions de l'image. De plus, les invariances aux rotations, translations et changements d'échelle ne sont pas assurées par ce descripteur. De nombreuses autres méthodes ont ainsi vu le jour. Nous nous intéressons au cas où les motifs sont des images pour lequel deux grands types d'approches existent : locales et globales.

Approches locales

Le motif est décrit autour de points particuliers "facilement" repérables, *i.e.* sur de petites régions situées autour de points d'intérêt. Ces zones sont considérées comme riches en information et traçables sur différentes instances du même objet. Il faut prêter attention d'une part à la détection des points d'intérêt et d'autre part à une bonne représentation locale de l'image.

Parmi les détecteurs de points d'intérêt, les plus connus sont Harris [Harris and Stephens, 1988] et ses dérivés Harris-Laplace et Hessian-Laplace [Mikolajczyk and Schmid, 2001], invariants aux changements d'échelle. Ils recherchent les régions de l'image pour lesquelles les propriétés, comme l'intensité, la couleur ou la texture, varient localement de façon significative par rapport à leur voisinage. Dans la figure 3.9, la majorité des points de Harris extraits pour deux images tournées à 90° sont communs, démontrant ainsi l'invariance par rotation du détecteur.

Un bon détecteur se caractérise par :
- la **répétabilité**, un maximum de points doivent être retrouvés pour un même objet vu selon différents points de vue ;
- la **distinguabilité**, les zones repérées sont suffisamment variées pour permettre leur mise en correspondance pour deux objets de la même catégorie ;
- la **localité**, pour réduire les risques d'occlusion et permettre une bonne correspondance sous d'autres angles ;
- la **quantité** ;
- la **précision** ;
- l'**efficacité**.

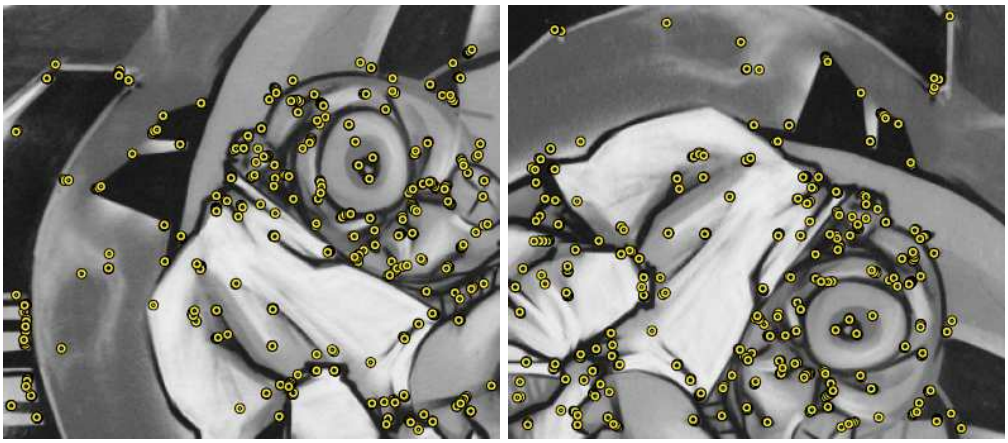


FIGURE 3.9 – Points d'intérêt détectés par Harris sur deux images identiques à une rotation près. L'invariance par rotation assure que les mêmes points seront retrouvés sur une image retournée (source : [Tuytelaars and Mikolajczyk, 2007]).

La description locale de la forme peut être effectuée à l'aide des points de Haar [Viola and Jones, 2001] qui consistent en une somme pondérée des pixels situés autour du point d'intérêt (figure 3.10). Un exemple sera compté positivement si cette somme dépasse un certain seuil θ.

$$\sum(\text{régions claires}) - \sum(\text{régions sombres}) > \theta \tag{3.11}$$

Classifieurs faibles, un grand nombre d'entre eux sont nécessaires pour reconnaître un objet. Ils ne sont pas non plus robustes aux différentes transformations géométriques.

D'autres descripteurs plus complexes tels les SIFT [Lowe, 2004] et les SURF [Bay et al., 2008] utilisent les informations locales de gradient. [Lowe, 2004] détermine l'orientation et l'amplitude des gradients en chaque point d'une région donnée, les pondère par une gaussienne 2D et les concatène ensuite sur une sous-région en histogrammes d'orientations (figure 3.11). Le SURF décrit la répartition des gradients autour d'un point en sommant les projections selon les axes x et y.

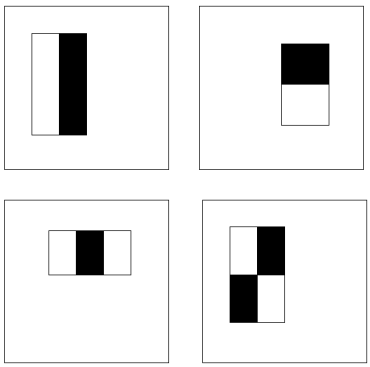


FIGURE 3.10 – Un descripteur Haar [Viola and Jones, 2001] est composé d'un ensemble de zones rectangulaires telles que les pixels situés dans les zones claires sont comptés positivement, et ceux des zones sombres, négativement. Si la somme excède un seuil fixé θ, le point est positif.

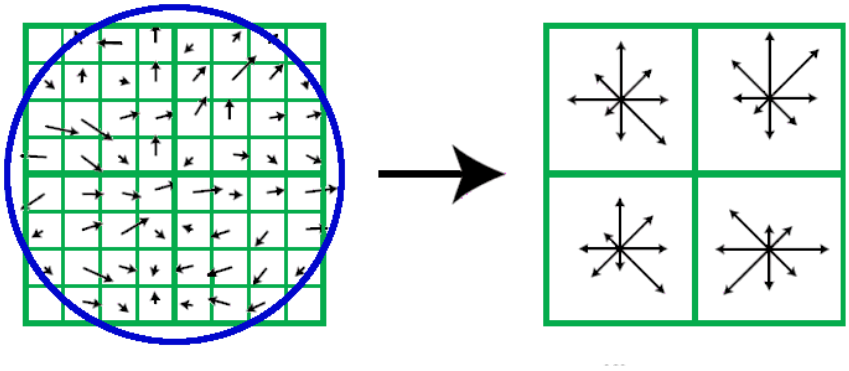


FIGURE 3.11 – Descripteur SIFT [Lowe, 2004]. Orientation et amplitude des gradients sont calculées en chaque point de la région définie autour du point d'intérêt. Elles sont pondérées par une gaussienne 2D, modélisée par le cercle bleu. Un histogramme d'orientations est ensuite obtenu pour chaque sous-région en concaténant toutes les informations des pixels.

La représentation de l'objet ainsi obtenue est compacte et ne concerne que les parties suffisamment discriminantes. Elle est également robuste aux occlusions car constituée d'un ensemble de

petits descripteurs locaux. La majorité des descripteurs utilisés sont également rapides à calculer. En revanche, l'aspect spatial est perdu, la position relative des points n'est pas conservée et il faut pouvoir extraire de l'image des points d'intérêt, ce qui nécessite une bonne résolution de départ sur l'image.

Approches globales

Les caractéristiques sont dites globales lorsque l'ensemble de l'objet, image ou contour, est utilisé pour établir le descripteur. Le plus connu, et le plus facile à mettre en œuvre, est l'histogramme qui fournit la distribution de pixels par intervalles de valeurs. Nous pouvons notamment citer les histogrammes de couleurs ou les HOG (*Histograms of Oriented Gradients* - Histogrammes de Gradients Orientés) utilisés pour la reconnaissance de piétons par [Dalal and Triggs, 2005] et illustrés par la figure 3.12. Pour calculer ces derniers, l'image est divisée en blocs, eux-mêmes divisés en cellules. Dans chacune d'elle, l'amplitude et l'orientation du gradient de chaque pixel sont calculées. Un histogramme d'orientations est ensuite formé, l'amplitude servant de poids. Décomposer l'image en blocs et cellules permet de conserver l'information spatiale perdue dans le cas d'un simple histogramme.

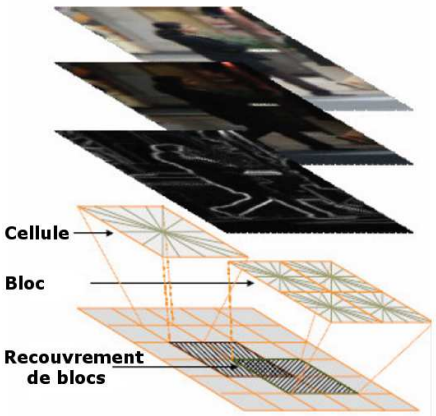


FIGURE 3.12 – Calcul des HOG (source : [Dalal and Triggs, 2005]). L'image est subdivisée en blocs, qui peuvent se chevaucher, eux-mêmes décomposés en cellules. L'histogramme est obtenu en regroupant les gradients de chaque pixel suivant leur orientation. Leur amplitude sert de mesure, une valeur forte comptant plus qu'une faible.

Le PHOG (*Pyramid of Histograms of Oriented Gradients* - Pyramide d'Histogrammes de Gradients Orientés) est un descripteur plus précis que le HOG, il lui apporte un aspect spatial supplémentaire. La figure 3.13 explique le calcul du PHOG sur $L = 3$ niveaux de décomposition. Au niveau l, l'image est subdivisée en 4^l sous-régions. Dans chacune d'elle, l'image des contours est synthétisée en un histogramme d'orientations de gradients pondérés par leur amplitude de K intervalles. Le descripteur global correspond à la concaténation des vecteurs obtenus pour chaque niveau, soit une dimension totale $K \sum_{l \in L} 4^l$. Enfin, une normalisation est effectuée sur le vecteur complet. Contrairement à l'approche proposée par [Dalal and Triggs, 2005] pour la détection de piétons, il n'y a pas de recouvrement des blocs, ni de partitionnement plus fin en cellules.

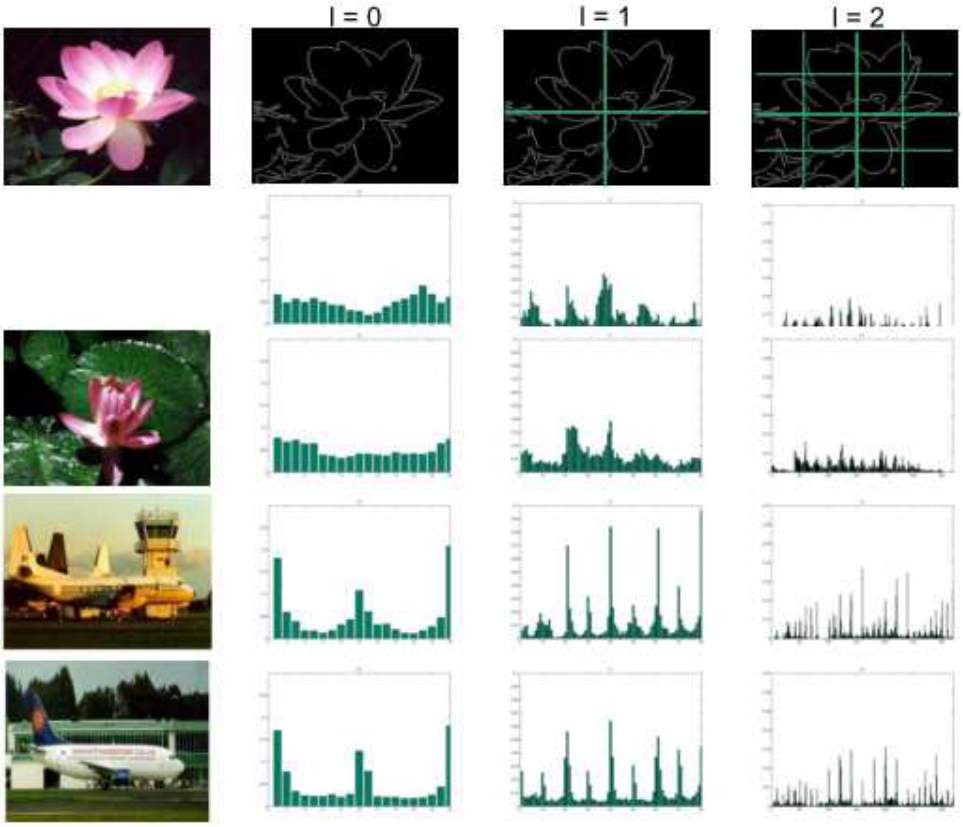


FIGURE 3.13 – Génération d'un PHOG pour quatre images (source : [Bosch et al., 2007b]). Dans un premier temps, l'image de gradients est calculée et orientation et amplitude sont stockées pour tous les points de contour. À chaque niveau l, l'image est décomposée en 4^l sous-régions pour lesquelles un histogramme d'orientations de gradients sur K intervalles est obtenu. La concaténation de tous les vecteurs de chaque niveau forme le descripteur final.

D'autres opérateurs plus complexes ont été élaborés comme les moments géométriques de Hu [Hu, 1962] et de Zernike [Teague, 1980]. Ces derniers représentent les propriétés spatiales de la distribution des pixels dans l'image, parmi lesquelles l'aire de l'objet, la position de son centre de gravité, la variance, etc. Une discrimination suffisante nécessite parfois un certain nombre d'ordres, longs à calculer. Les filtres de Gabor ou les ondelettes sont quant à eux particulièrement adaptés pour la classification de textures [Idrissa and Acheroy, 2002, Li and Shawe-Taylor, 2005] car ils permettent de discriminer les motifs d'un point de vue fréquentiel. Les descripteurs de Fourier permettent de reconnaître un objet d'après son contour [Roth and Winter, 2008] et il est également possible de projeter l'objet dans un espace plus restreint à l'aide de l'Analyse en Composantes Principales.

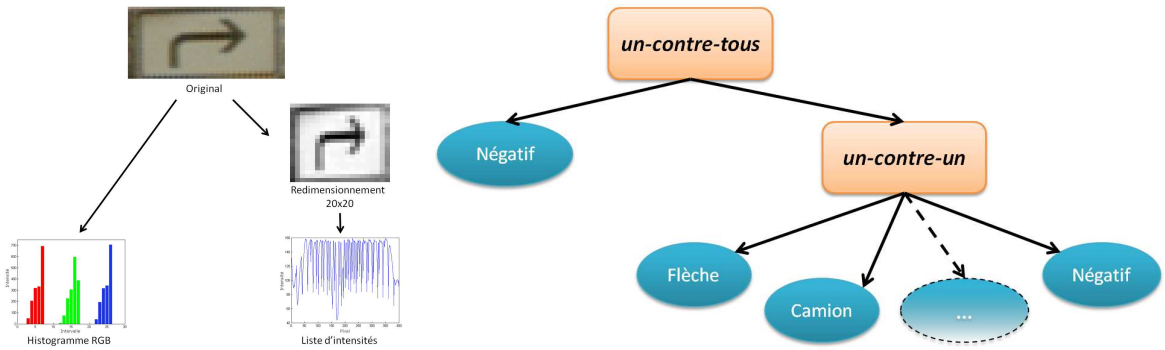
Les approches globales visent à transformer l'image dans un autre espace de manière à en faire ressortir les caractéristiques. Leur avantage principal est qu'elles fournissent une représentation complète de l'image. Toutefois, elles ne sont pas souvent robustes aux occlusions ou transformations affines. De plus, aucun moyen ne permet de distinguer l'objet de son fond car les données utilisées appartiennent aux deux catégories.

3.2.3 Domaine des panonceaux

Seules quelques rares techniques ont été publiées sur le sujet mais leur comparaison est difficile étant donné qu'elles sont évaluées sur des bases indépendantes et privées. Les recherches existantes ont toutes en commun d'utiliser des descripteurs globaux et une même méthodologie : une phase de détection de rectangles suivie d'une classification et d'un éventuel suivi. Parmi les recherches sur le sujet, nous pouvons citer [Hamdoun et al., 2008, Nienhüser et al., 2010, Liu et al., 2011].

[Nienhüser et al., 2010] ont développé un système de reconnaissance de panonceaux à partir de deux types de descripteurs et une architecture en forme d'arbre (figure 3.14). Dans un premier temps, un histogramme RGB de l'image est calculé et utilisé pour distinguer les négatifs des vrais panonceaux. Ensuite, la liste des pixels de l'image en niveaux de gris redimensionnée en 20x20 est employée pour reconnaître la catégorie du panonceau à l'aide d'un ensemble de SVMs de type *un-contre-tous*.

La méthode employée donne de bonnes performances sur la base de données présentée avec une précision de 97%. Le temps de calcul est, de plus, relativement faible avec un temps moyen de 4.4ms (langage C, sur un Intel Core 2 Quad Processor Q8200). Toutefois, le problème est limité à quatre catégories de panonceaux, à savoir "Flèche", "Camion", "Bei Nässe" ("Par temps de pluie") et "7.5t", qui figurent parmi les plus représentées en Allemagne. Le redimensionnement des images peut entraîner une dégradation des données et les descripteurs utilisés ne résolvent pas le problème de la sensibilité aux translations, rotations et changements d'échelle. De surcroît, la base de données est trop petite pour apprécier pleinement les résultats présentés (1633 images dans la base d'apprentissage dont 505 positifs et 525 images pour la base de test).



(a) Caractéristiques (b) Classifieurs

FIGURE 3.14 – Architecture employée pour la classification de panonceaux par [Nienhüser et al., 2010]. (a) Le vecteur de caractéristiques est constitué d'une part de l'histogramme RGB du panonceau original, d'autre part de la liste des intensités des pixels de l'image en niveau de gris redimensionnée à la taille 20x20. (b) La classification se fait à l'aide d'un arbre de classifieurs. Les deux étapes principales sont la séparation des négatifs du reste des vrais panonceaux puis la reconnaissance du type.

[Liu et al., 2011] proposent une autre structure en forme d'arbre, le SVM-BTA (*SVM with Binary Tree Architecture* - SVM avec une Architecture d'Arbres Binaires) (figure 3.15). Le principe est de décomposer un problème multiclasses en un sous-ensemble de problèmes à deux classes. L'espace des classes possibles est ainsi décomposé en macro-classes et à chaque itération, une classification binaire attribue l'exemple inconnu à l'une ou l'autre de ces macro-classes. L'avantage de cette architecture est qu'elle rejette naturellement les négatifs sans pour autant les

avoir préalablement inclus dans la base d'apprentissage. Les caractéristiques utilisées pour décrire les panonceaux sont une combinaison des descripteurs de Fourier et des ondelettes de Gabor. Les vecteurs ainsi obtenus seront robustes aux différentes transformations. Malheureusement, les évaluations proposées ne semblent avoir été réalisées que sur trois types de panonceaux chinois, qui ne sont pas précisés. Bases de données et temps de calcul ne sont pas non plus exploitables.

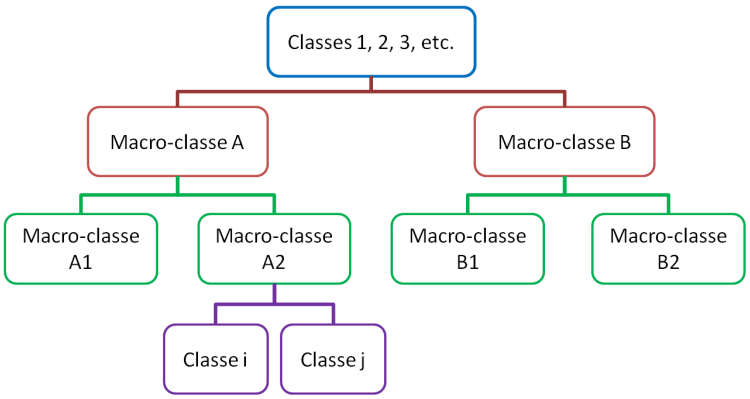


FIGURE 3.15 – Architecture d'un SVM-BTA (source : [Liu et al., 2011]). À chaque nœud, l'ensemble des classes restantes est scindé en deux sous-groupes tels que chaque exemple arrivant dans ce sous-arbre sera classifié entre les classes présentes. Au niveau des feuilles, il ne reste plus qu'une seule classe.

[Hamdoun et al., 2008] font la classification des flèches françaises uniquement à l'aide d'un réseau de neurones entraîné sur les intensités des pixels d'une imagette du panonceau. Ces travaux ont été menés au sein de notre laboratoire du CAOR à l'École des Mines de Paris. Les résultats sont prometteurs bien que limités à une seule catégorie. La base de données utilisée pour les évaluations ne comporte que 18 instances de panonceaux pour un taux final de reconnaissance de 78%. Les réseaux de neurones sont un outil puissant mais leur interprétation n'est pas évidente et l'utilisateur n'a pas facilement accès à ses rouages. De plus, utiliser directement les pixels de l'imagette (redimensionnée) en entrée du classifieur n'est guère robuste par rapport aux défauts de cadrage de la phase de détection.

3.3 Méthode proposée

3.3.1 Analyse du problème

La reconnaissance des panonceaux est un problème complexe pour de nombreuses raisons. Tout d'abord, les informations présentées sont très diverses et les techniques développées auparavant restreignent le problème à la reconnaissance de quelques classes seulement. Ne sont pris en compte que les panonceaux les plus répandus ou critiques pour la détermination de la vitesse limite. Ensuite, la résolution des caméras utilisées est très faible. De ce fait, les panonceaux sont souvent très petits dans les bases de données et de granularité parfois trop grossière pour permettre la reconnaissance de certaines catégories comme le texte. Enfin, la diversité des tailles et ratio (chapitre 2, figure 2.21) limite le choix de la technique de reconnaissance. Une analyse par *template-matching* requerrait ainsi un trop grand nombre de modèles pour espérer une classification en temps-réel.

3.3.2 Définition de macro-catégories

Nous avons choisi de définir des macro-catégories pour simplifier la tâche de classification et parce qu'il semble difficile, voire impossible, de déterminer immédiatement l'information située sur le panonceau. Le choix de ces super-classes est dicté par l'expertise et la similitude observée entre les panonceaux. Nous avons ainsi sélectionné les classes **"Flèche", "Texte", "Pictogramme"** et **"Mixte"**. Une fois cette distinction faite, un classifieur plus spécifique pourra être utilisé pour raffiner le résultat. Un algorithme d'OCR (*Optical Character Recognition* - Reconnaissance Optique de Caractères), comme Tesseract [HPLabs, 2005], permettrait ainsi de déchiffrer les panonceaux de type "Texte". Cette architecture en arbre se rapproche de celles présentées par [Nienhüser et al., 2010] et [Liu et al., 2011].

L'architecture générale de la classification, illustrée par la figure 3.16, est constituée de deux étapes de classification. La première vise à séparer négatifs et positifs de manière à réduire le temps de traitement lorsqu'aucun panonceau n'est présent et aussi à améliorer la précision du système en éliminant des fausses détections. Un classifieur SVM *un-contre-un* avec un noyau RBF (*Radial Basis Function* - Fonction à Base Radiale) est employé. Les images considérées comme négatifs sont alors éliminées de la chaîne. La seconde étape doit permettre de déterminer la macro-catégorie des panonceaux restants. Un ensemble de six classifieurs SVMs de type *un-contre-un* à noyau RBF permet de répartir les exemples entre les quatre macro-catégories.

Nous utilisons la bibliothèque LibSVM de [Chang and Lin, 2011] qui comporte de nombreuses implémentations de SVMs. Elle permet, entre autres, de générer des probabilités d'appartenance aux différentes classes en se basant sur les travaux de [Wu et al., 2004]. Grâce à cela, nous éliminons, au cours de la classification, une grande partie des négatifs :

- La probabilité p_{pos} qu'un exemple appartienne à la classe des positifs doit être supérieure à un seuil τ_{pos} pour le valider. Il est éliminé dans le cas contraire.
- La différence entre les deux meilleures probabilités doit dépasser un second seuil Δ_{1-2} pour que l'exemple soit associé à la meilleure classe, sinon il est également éliminé.

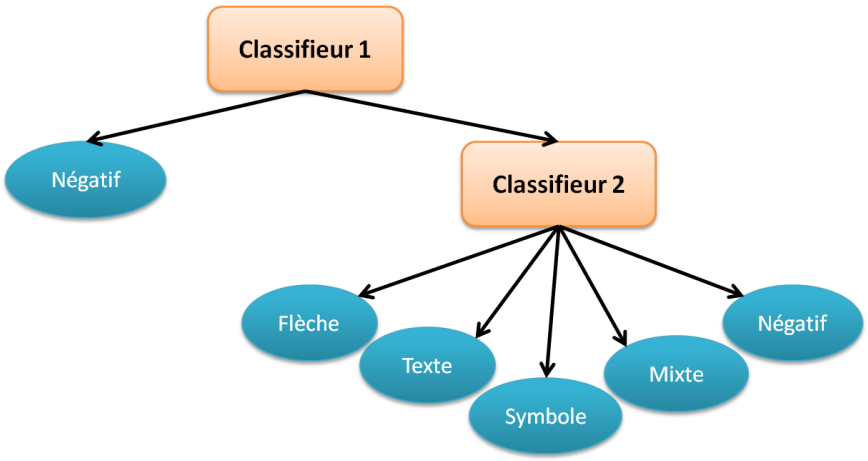


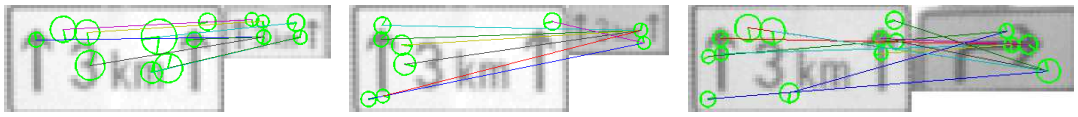
FIGURE 3.16 – Hiérarchie de classifieurs utilisée pour la reconnaissance des panonceaux. Au premier niveau, les exemples sont séparés de façon binaire entre négatifs, qui sont éliminés, et positifs. Ces derniers font ensuite l'objet d'une seconde classification afin de déterminer leur macro-catégorie. Pour raffiner encore le processus, il est possible de rajouter une troisième couche pour reconnaître en détail l'information présente (le texte par exemple).

3.3.3 Utilisation de descripteurs globaux

Utiliser des descripteurs locaux est une pratique relativement courante en classification. Néanmoins, dans notre application, ils présentent certains inconvénients qui les rendent inappropriés.

Tout d'abord, la taille des panonceaux conduit à l'extraction d'un faible nombre de points d'intérêt. En effet, même si certains détecteurs sont basés uniquement sur un point, les descripteurs utilisent une région autour de ce dernier pour permettre la mise en correspondance. Les descripteurs SIFT par exemple sont définis à partir d'une grille de 16x16 pixels. La figure 3.17 illustre la détection de points d'intérêt pour deux images du même panonceau vu à des distances différentes. Le nombre de points repérés est trop faible pour permettre une bonne description de l'image, ce qui rend la mise en correspondance difficile.

Ensuite, l'usage de descripteurs locaux nous obligerait à classifier chaque catégorie de panonceaux existante de manière indépendante. Les points d'intérêt extraits pour les panonceaux "Camion" risquent de ne rien avoir de commun avec ceux des "Motos", de même qu'un "200m" n'aura rien de commun avec "bei Nässe". Le problème deviendrait rapidement trop complexe et spécifique, notamment au pays auquel appartiennent les panonceaux.



(a) Même panonceau vu à des distances différentes (b) Panonceaux appartenant à la même catégorie (c) Panonceaux de types différents

FIGURE 3.17 – Illustration de la difficulté d'utiliser des descripteurs locaux (ici des SIFT) pour la reconnaissance de panonceaux. (a) La mise en correspondance entre deux occurrences du même panonceau vu à des distances différentes est correcte malgré le faible nombre de points d'intérêt. (b) Pour deux panonceaux du même type mais vus à différents moments et lieux, les erreurs d'association sont très importantes. (c) Le résultat de mise en correspondance de deux panonceaux de types complètement différents, présentant des résolutions moyennes, est peu convaincant, les points d'intérêt étant mal appariés. De tels descripteurs pourraient donc être utilisés pour le suivi de panonceaux mais pas pour l'étape de classification.

Nous avons donc privilégié l'usage de différents types de descripteurs globaux.

Le premier se rapproche des descripteurs utilisés par [Hamdoun et al., 2008] et [Nienhüser et al., 2010]. Il consiste à normaliser et redimensionner l'image à une taille fixe, de 20x20 pixels dans notre cas, quelle que soit sa catégorie ou sa taille initiale, puis de faire la liste, ligne par ligne, des intensités normalisées des pixels (figure 3.18). Normaliser l'image a pour but de rendre le descripteur indépendant de la luminosité ambiante. Redimensionner assure que tous les vecteurs auront la même taille malgré une grande variété de panonceaux, vus à des distances et sous des angles différents. Comme le montre la figure 3.19, les vecteurs représentant les panonceaux possèdent une certaine cohérence dûe à la présence d'une texture, d'un motif répété. Ils oscillent autour d'une valeur moyenne constante, conséquence de l'apparence binaire des panonceaux, les valeurs extrêmes étant quasiment constantes sur l'image. Les négatifs sont moins structurés, la valeur moyenne sur une ligne change au cours du parcours de l'image ainsi que les extrema.

Le second descripteur, le PHOG, est plus précis et apporte un aspect spatial au HOG classique. Bien qu'il nous semble essentiel de conserver l'aspect spatial pour la classification, un découpage trop fin rendrait le descripteur trop sensible aux erreurs de centrage et plus long à calculer. De plus, la faible taille des panonceaux ne justifie pas ce niveau de précision. La figure 3.20 illustre le calcul du PHOG pour un panonceau de type "Flèche".

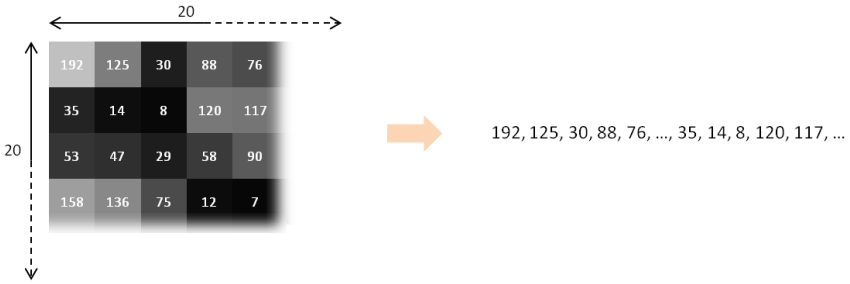
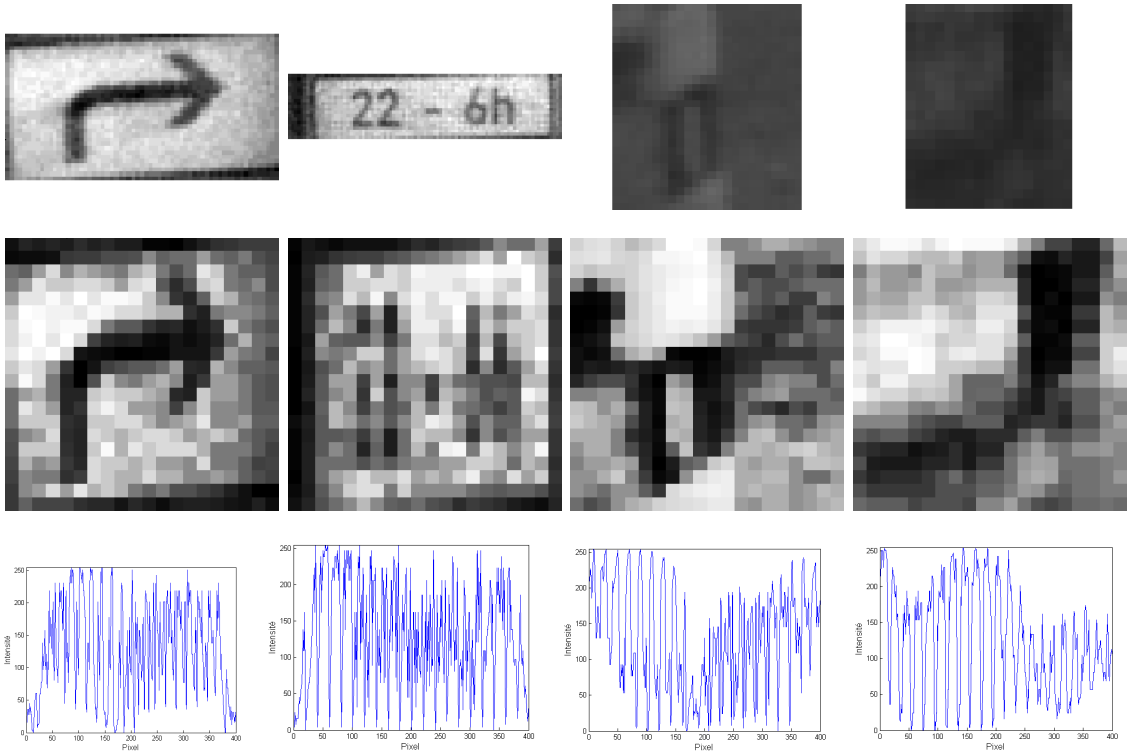


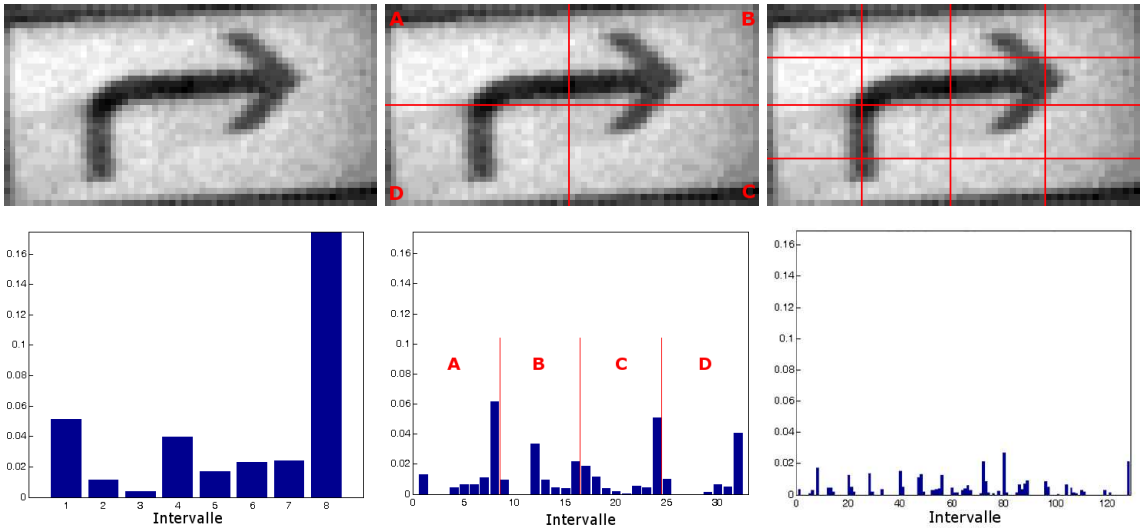
FIGURE 3.18 – Vecteur de caractéristiques de type "Liste". L'image est d'abord redimensionnée à une taille fixe de 20x20 pixels. La valeur des pixels ainsi obtenus est concaténée en une liste de 400 éléments.



(a) "Flèche" (b) "Texte" (c) "Négatif" (d) "Négatif"

FIGURE 3.19 – Descripteurs de type "Liste" obtenus après redimensionnement de l'image originale (haut) à une taille fixe de 20x20 pixels (milieu). Les vecteurs obtenus pour les panonceaux positifs, comme les flèches (a) ou les textes (b), présentent une structure répétée sur chaque ligne. Idéalement, il n'y a que deux couleurs sur les panonceaux, noir et blanc, ce qui limite les valeurs entre deux extrêmes correspondant à la couleur dominante du texte et celle du fond. Le vecteur oscille autour d'une valeur moyenne contrairement aux négatifs (c) et (d).

Nous combinons ce descripteur à un autre, simple, permettant de qualifier la texture en étudiant dans chacune des sous-régions précédentes le ratio de pixels clairs sur sombres. Nous le nommerons "Proportion" par la suite. Pour cela, l'image est tout d'abord binarisée à l'aide de la même technique qui a permis d'extraire les graines pour la croissance de région (section 2.3.2). Un algorithme de



(a) $l = 0$ (b) $l = 1$ (c) $l = 2$

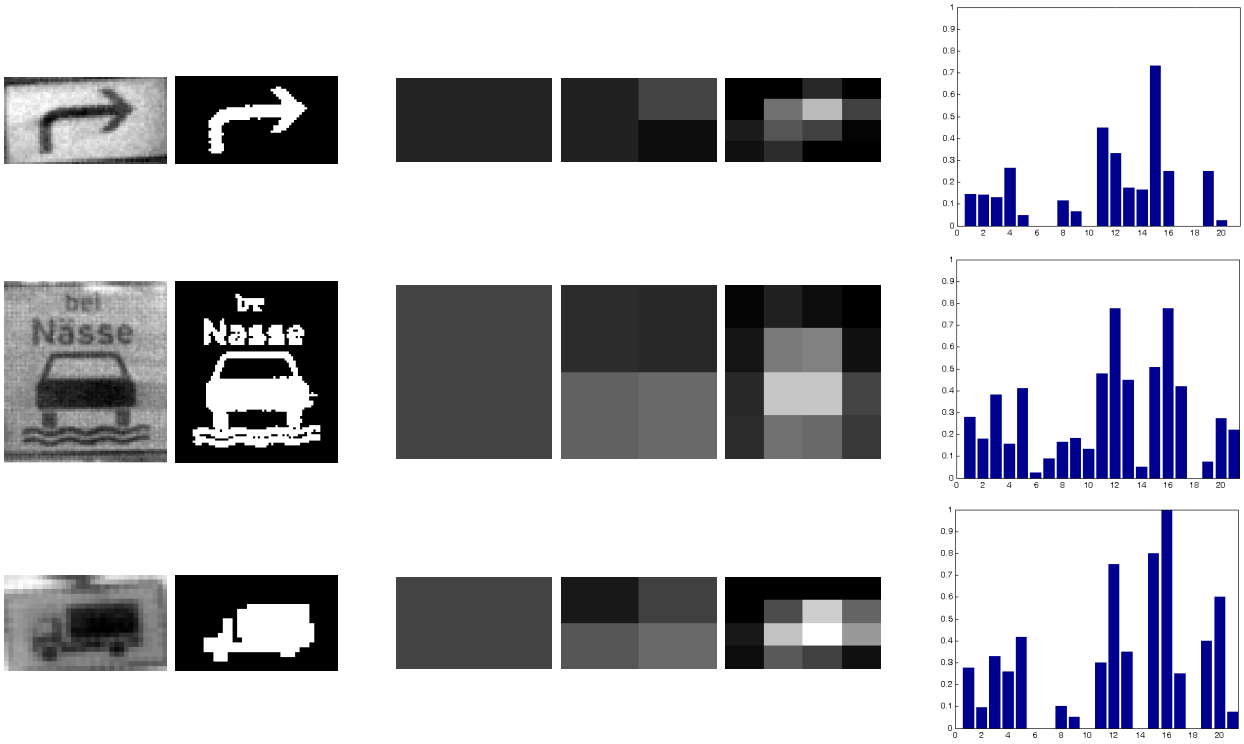
FIGURE 3.20 – Illustration du calcul d'un Pyramid HOG pour un panonceau de type "Flèche" sur trois niveaux. Les orientations des gradients de chaque sous-région sont ajoutées à un histogramme avec un facteur relatif à leur amplitude. Les histogrammes ont chacun $K = 8$ intervalles et sont obtenus pour des orientations entre 0 et π.

reconstruction morphologique est utilisé afin de trouver les pixels fortement contrastés de l'image. Une sélection est ensuite opérée à l'aide de deux seuils t'_1 et t'_2. Le premier permet de conserver les pixels appartenant à ces maxima locaux tandis que le second nous aide à éliminer ceux qui présentent une valeur moyenne de contraste trop faible. Les seuils utilisés sont moins stricts que pour la détection de rectangles, $t'_1 = \mu + 0.5\sigma$ et $t'_2 = \mu + 1.5\sigma$. Il est ainsi possible de connaître grossièrement la forme de l'objet. La figure 3.21 illustre le calcul du vecteur "Proportion" pour trois types de panonceaux. L'information présente est tout d'abord séparée du fond grâce à la binarisation. L'image est ensuite découpée en plusieurs blocs, de la même façon que le PHOG, et le ratio de pixels blancs sur noirs est calculé dans chacun d'eux. Au premier niveau, la valeur obtenue nous donne une indication sur la densité globale de pixels noirs. La flèche par exemple présente moins de pixels noirs que le camion. Aux niveaux plus fins, nous pouvons déterminer la localisation des informations de manière plus précise. Pour le panonceau "bei Nässe", nous constatons que la densité est plus forte pour les blocs situés en bas et au centre alors que pour la flèche elle serait localisée en haut à droite.

3.4 Évaluation

3.4.1 Base de données

La base de données utilisée pour la reconnaissance a été acquise en Allemagne avec une caméra couleur de résolution 752 x 480, 10 bits. Les panonceaux utilisés pour l'apprentissage et l'évaluation ont été extraits de la vérité terrain, elle-même obtenue par annotation manuelle image par image. La répartition des catégories de panonceaux dans les deux types de bases est présentée dans le tableau 3.1. La diversité des informations au sein d'une macro-catégorie est importante et chaque sous-type n'est pas nécessairement représenté de manière équilibrée. Pour le texte par exemple, les messages existants concernent le tonnage, la distance d'application, la plage



(a) (b) (c) (d) (e) (f)

FIGURE 3.21 – Calcul du vecteur "Proportion" lié à la répartition de pixels clairs et sombres dans différentes sous-régions de l'image binarisée. (a) Image originale. (b) Image binarisée à l'aide de la reconstruction morphologique présentée dans la section 2.3.2. (c) (d) (e) Ratio de pixels clairs sur sombres dans l'image de niveau $l = 0$, $l = 1$ et $l = 2$. (f) Vecteur obtenu par concaténation de toutes les valeurs des niveaux précédents.

horaire concernée, etc. et le nombre d'images de chaque type varie en fonction de la fréquence d'apparition du panonceau.

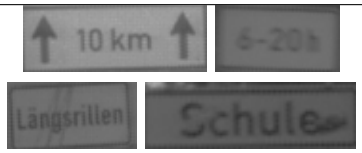
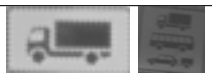
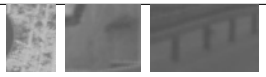
Catégorie	Exemples	Apprentissage	Test
"Texte"		1693	697
"Flèche"		267	99
"Pictogramme"		2030	473
"Mixte"		1293	521
"Négatif"		13856	4944

TABLE 3.1 – Bases de données utilisées pour la classification de panonceaux.

Pour déterminer si un panonceau a été correctement classifié ou non, nous utilisons les mesures de précision et de rappel.

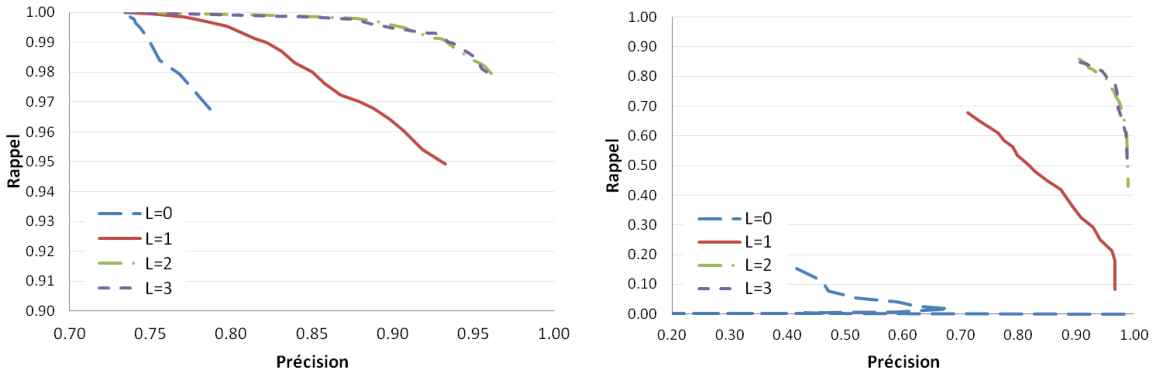
$$\text{Précision} = \frac{\text{Nombre de panonceaux correctement affectés à la classe } l}{\text{Nombre de panonceaux affectés à la classe } l} \tag{3.12}$$

$$\text{Rappel} = \frac{\text{Nombre de panonceaux correctement affectés à la classe } l}{\text{Nombre de panonceaux de la classe } l} \tag{3.13}$$

3.4.2 Performances

Justification du paramétrage utilisé pour le PHOG

L'utilisation de PHOG nécessite de définir certains paramètres, comme le nombre de niveaux L de la pyramide, la prise en compte ou non des gradients signés et le nombre d'intervalles de l'histogramme K. Intéressons-nous tout d'abord au niveau de définition de la pyramide. Un faible nombre de niveaux fournit un descripteur de petite dimension mais plus sensible aux problèmes de centrage. En revanche, un trop grand nombre de niveaux rendrait le descripteur trop spécifique, risquant ainsi la sur-segmentation. Sa dimension serait également trop importante. [Bosch et al., 2007b] préconisent d'utiliser jusqu'à quatre niveaux de résolution sur leur base de données. Nous avons comparé les performances obtenues pour des niveaux $L = \{0, 1, 2, 3\}$ pour un nombre d'intervalles $K = 8$ avec des gradients non-signés dans la figure 3.22. Les meilleures correspondent aux plus grandes valeurs de L, comme attendu. Pour $L = 2$, nous atteignons un rappel de 99% pour une précision de 95% pour les négatifs. Pour les positifs, nous obtenons un rappel de 80% pour une précision de 97%. Toutefois, le gain entre $L = 2$ et $L = 3$ est trop faible pour justifier le choix d'un descripteur de dimension quatre fois plus grande (pour $L = 2$, $21 * 8 = 168$ et pour $L = 3$, $85 * 8 = 680$).



(a) Classe des négatifs (b) Classe des positifs

FIGURE 3.22 – Influence du nombre L de niveaux de la pyramide du PHOG pour la classification des panonceaux. Les PHOG ont été calculés avec $K = 8$ et des gradients non-signés. À partir du niveau de définition $\mathbf{L} = \mathbf{2}$, les résultats sont similaires. En revanche, pour une résolution plus faible, les performances sont bien moindres avec un rappel proche de 0% pour $L = 0$.

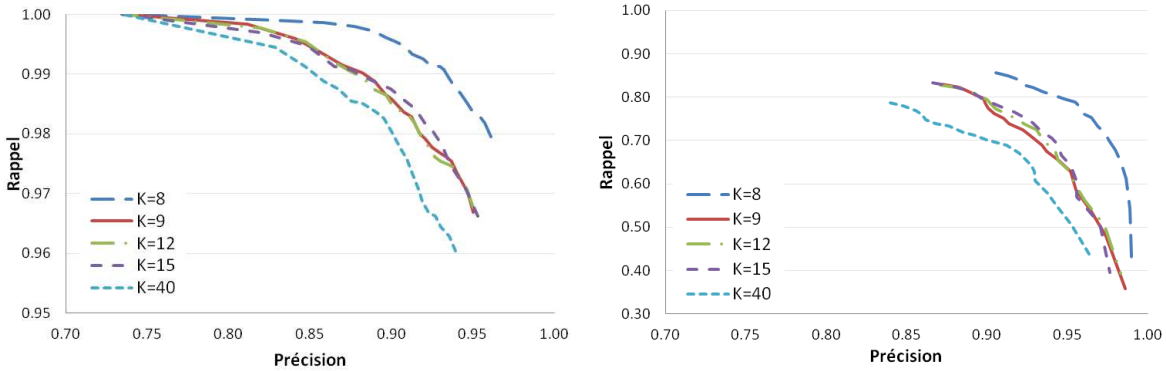
Une fois défini le niveau L de résolution de la pyramide, nous examinons l'influence du choix du nombre d'intervalles K du PHOG. Nous choisissons d'utiliser des gradients non-signés, *i.e.* compris entre 0° et 180°. Les performances obtenues pour des gradients signés sont négligeables comparées au choix de K [Bosch et al., 2007a], et pour une même résolution, il faudrait doubler le nombre d'intervalles, et donc la taille du descripteur. D'après la figure 3.23, il apparaît qu'une résolution trop grande en angle n'est pas un choix judicieux car le descripteur devient trop sensible aux problèmes de mauvaise orientation. Pour $K = 40$ par exemple, la résolution angulaire est de $180°/40 = 4.5°$. Les meilleures performances correspondent à la valeur $K = 8$, soit une résolution de $180°/8 = 22.5°$. Cela s'explique par l'aspect même des panonceaux qui comportent des symboles assez simples géométriquement, principalement composés de lignes droites horizontales et verticales. Nous choisissons finalement $\mathbf{L} = \mathbf{2}$ et $\mathbf{K} = \mathbf{8}$ pour notre PHOG.

Justification de l'architecture

Pour justifier l'utilisation d'un arbre de classification, nous avons implémenté différentes combinaisons de descripteurs et d'architectures. Commençons par comparer les performances de systèmes basés sur la liste de pixels, les PHOG et les "PHOG+Proportion". Nous cherchons à déterminer à quelle macro-catégorie appartient un panonceau et l'apprentissage est effectué sur l'ensemble des positifs et des négatifs. Cela correspond au sous-arbre droit de la figure 3.16.

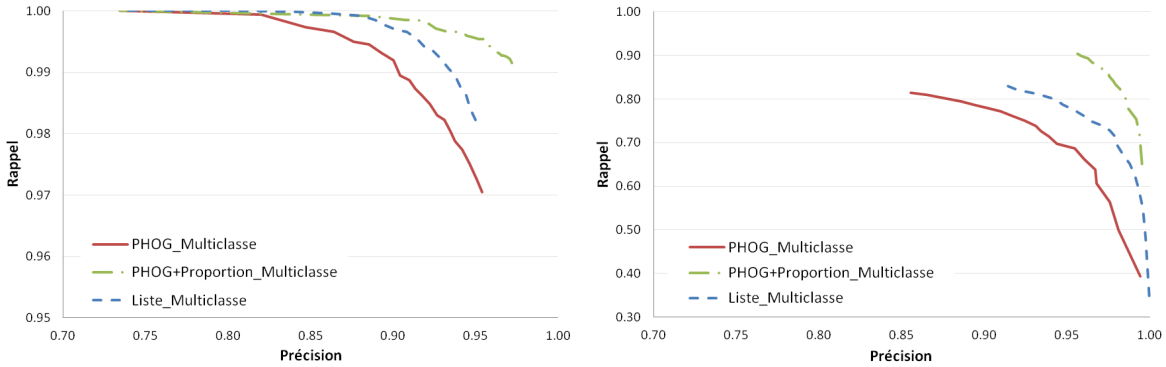
La figure 3.24 présente les courbes précision-rappel pour les trois descripteurs employés pour la classification directe de toutes les classes. Pour générer ces fonctions, nous avons fait varier l'écart minimum Δ_{1-2}. Lorsque la distance entre les deux classes est inférieure au seuil, l'image est considérée comme un négatif. Le descripteur combiné "PHOG+Proportion" s'avère le meilleur en terme de rappel. L'apport de l'utilisation du descripteur "Proportion" est évident au regard des courbes et le gain obtenu atteint 20%.

Afin d'améliorer les performances, nous avons réalisé une classification binaire permettant d'éliminer une grande partie des négatifs. En effet, la plupart des rectangles issus de notre algorithme de détection risque de ne pas être aussi idéaux que les exemples proposés dans cette partie.



(a) Classe des négatifs (b) Classe des positifs

FIGURE 3.23 – Influence du nombre d'intervalles K du PHOG pour la classification des panonceaux. Les PHOG ont été calculés sur deux niveaux ($L = 2$) et pour des gradients non-signés, *i.e.* des angles appartenant à $[0, 180]$. Le meilleur descripteur, pour les négatifs comme les positifs, correspond à $K = 8$ et le moins bon à $K = 40$. Les performances similaires des cas $K = 9$, $K = 12$ et $K = 15$ semblent indiquer que la quantité d'informations supplémentaires apportée en augmentant K est faible.

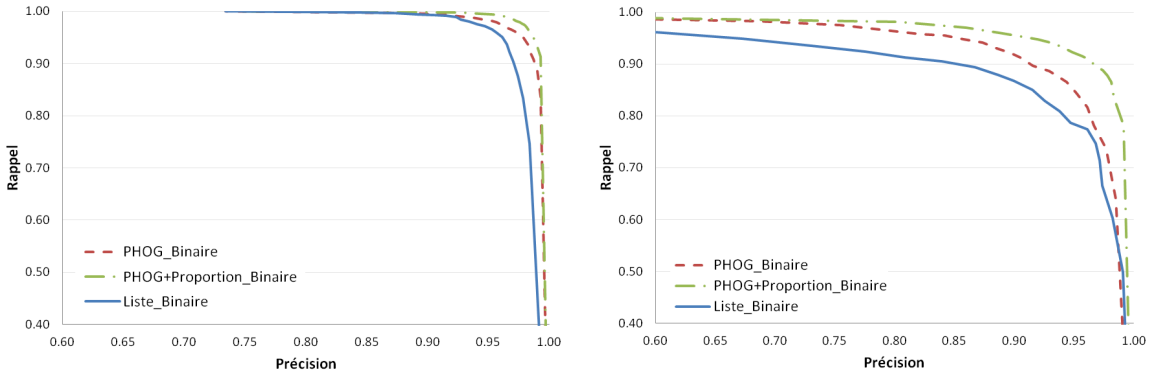


(a) Classe des négatifs (b) Classe des positifs

FIGURE 3.24 – Courbes précision-rappel obtenues pour les trois descripteurs -"Liste", "PHOG" et "PHOG+Proportion"- pour la classification de toutes les classes. Le meilleur descripteur s'avère être la combinaison "PHOG+Proportion" qui présente de meilleurs résultats tant en précision qu'en rappel, pour les négatifs comme pour les positifs. Les performances atteignent un rappel de 90% et une précision de 95%. Toutefois, l'architecture actuelle est figée et ne permet pas de jouer sur le compromis précision-rappel pour adapter le système aux attentes.

Il faut garder à l'esprit que nous aurons à ce stade une limite haute, des performances optimales, pour notre système complet. Cela présente deux avantages : réduire le temps nécessaire à la reconnaissance en supprimant la majorité des négatifs et utiliser un classifieur plus précis entraîné uniquement sur les positifs pour améliorer les performances. Nous comparons une fois de plus les trois descripteurs (figure 3.25) et le meilleur semble être la combinaison "PHOG+Proportion". En effet, les performances attendues correspondraient à celle d'un classifieur excluant tous les négatifs et conservant tous les positifs, soit un rappel pour la classe "Positif" de 1 et une précision pour la classe "Négatif" de 1 également. Avec "PHOG+Proportion", nous atteignons, pour la classe des négatifs, un rappel de 97% pour une précision de 98% et pour celle des positifs, 94% pour les

deux.



(a) Classe des négatifs (b) Classe des positifs

FIGURE 3.25 – Performances obtenues pour la classification des panonceaux en négatifs ou positifs. Le meilleur des trois descripteurs utilisés est la combinaison "PHOG+Proportion". L'objectif est de supprimer le maximum de négatifs tout en conservant la quasi totalité des positifs, soit un rappel proche de 1 pour les positifs et une précision de 1 pour les négatifs.

Nous voyons ensuite deux possibilités pour déterminer la classe l_x d'un objet inconnu x à l'aide des deux étapes, positifs-négatifs puis macro-catégorie. La première serait de multiplier directement les probabilités obtenues à chaque étape puis de sélectionner la classe avec la plus grande valeur (méthode "PHOG+Proportion_Fusion" de la figure 3.26). Il s'agit d'une architecture linéaire et non en arbre, les deux classifieurs étant appliqués sur le même ensemble de données. Elle pourrait être rapprochée d'une fusion de données pour laquelle chaque classifieur donne son appréciation du problème, laissant au système le soin de combiner les informations pour faire le "bon" choix.

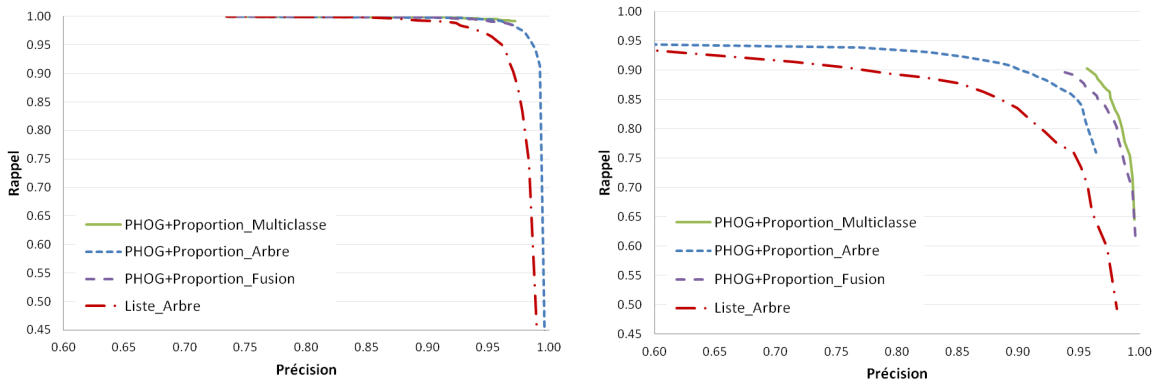
$$l \in Pos = \{Texte, Flèche, Pictogramme, Mixte\} \cup Neg = \{-1\} \tag{3.14}$$

$$l_1 \in Pos \quad p(l_x = l_1) = p_1(l_x \neq -1).p_2(l_x = l_1) \tag{3.15}$$

$$p(l_x = -1) = p_1(l_x = -1) + p_1(l_x \neq -1).p_2(l_x = -1) \tag{3.16}$$

$$l_x^* = \text{argmax}_l\{p(l_x = l)\} \tag{3.17}$$

La seconde consiste à fixer deux seuils, l'un pour déterminer si un exemple est positif τ_{pos}, l'autre pour sélectionner la meilleure classe Δ_{1-2}. De cette façon, de nombreux négatifs sont éliminés dès la première étape, ce qui réduit le temps de calcul. De plus, le compromis entre rappel et précision est plus facilement ajustable car un seuil fort pour la première étape permet de supprimer un maximum de fausses alarmes, augmentant ainsi la précision, alors qu'un seuil faible assure une bonne valeur de rappel. Nous avons donc choisi cette seconde solution avec comme descripteur "PHOG+Proportion" pour les deux étapes de classification. Nous présentons les résultats de la classification avec cette architecture dans la figure 3.26. Les courbes sont obtenues en faisant varier le seuil τ_{pos}. La classification directe avec "PHOG+Proportion" discrimine en une seule étape les négatifs et chacune des macro-catégories. Elle offre une très bonne précision pour les positifs de 97% pour un rappel de 85%. En revanche, elle risque de conduire à plus de fausses alarmes. L'utilisation du descripteur "Liste" n'est pas convaincante, ses performances se situent bien en-dessous des trois autres. L'utilisation du "PHOG+Proportion" et de la hiérarchie de classifieurs apparaît comme plus flexible, la gamme de valeurs Précision-Rappel étant plus large.



(a) Classe des négatifs (b) Classe des positifs

FIGURE 3.26 – Comparaison de différentes architectures possibles pour la classification des panonceaux, du classifieur seul qui permet la classification en macro-catégories et négatifs (PHOG+Proportion_Multiclasse) à l'architecture en arbre proposée pour deux descripteurs (PHOG+Proportion_Arbre et Liste_Arbre) en passant par la fusion des deux étapes par multiplication directe des probabilités (PHOG+Proportion_Fusion). Les courbes sont obtenues en faisant varier le seuil τ_{pos}. La meilleure solution semble être l'utilisation d'une architecture en arbre avec deux classifieurs et un descripteur de type "PHOG+Proportion" (PHOG+Proportion_Arbre). Elle permet d'atteindre, pour $\tau_{pos} = 0.2$, une précision de 87% pour un rappel de 92% pour les positifs

Analyse des résultats

La figure 3.27 montre les résultats de la classification pour une architecture en arbre avec le descripteur "PHOG+Proportion". Ces courbes ont été obtenues en faisant varier le seuil de probabilité τ_{pos} d'appartenir aux positifs. La matrice de confusion du tableau 3.2 expose les performances au sens du meilleur compromis précision-rappel, obtenu pour $\tau_{pos} = 0.1$. La précision pour les négatifs atteint 98%, soit 1 fausse détection pour 20 correctes, et un rappel de 96%. Peu de panonceaux sont confondus avec une autre classe, les erreurs correspondent majoritairement à une confusion avec les négatifs.

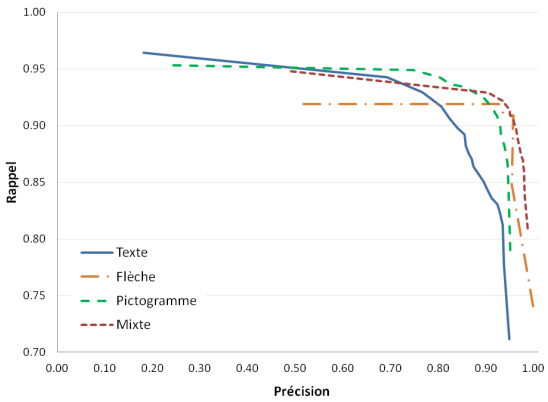


FIGURE 3.27 – Courbes Précision-Rappel obtenues pour l'architecture en arbre de classifieurs pour chaque macro-catégorie. Ces courbes ont été réalisées en faisant varier le seuil de probabilité τ_{pos} d'appartenir à la classe des positifs. Plus ce seuil est haut, meilleure sera la précision et pire le rappel. Le choix du point de fonctionnement dépendra de l'application.

		Classifieur					
		"Négatif"	"Texte"	"Flèche"	"Pictogramme"	"Mixte"	Rappel
Vérité Terrain	"Négatif"	4766	107	0	49	22	**96%**
	"Texte"	48	632	2	13	2	**91%**
	"Flèche"	2	4	90	2	1	**91%**
	"Pictogramme"	23	8	0	440	2	**93%**
	"Mixte"	24	14	0	9	474	**91%**
	Précision	**98%**	**83%**	**98%**	**86%**	**95%**	**95%**

TABLE 3.2 – Matrice de confusion obtenue pour le classifieur en deux étapes et le descripteur "PHOG+Proportion". Le seuil τ_{pos} a été fixé à **0.2** et la différence Δ_{1-2} entre les deux meilleures probabilités des macro-catégories à **0.05**.

La catégorie "Texte" présente la moins bonne précision (83%) ce qui s'explique par la présence dans la classe des négatifs de nombreuses images de rambardes. La texture régulière ainsi que l'alignement des motifs facilitent la confusion entre les deux classes (figure 3.28). Concernant les pictogrammes, certains panneaux représentant des véhicules, comme "Interdiction de doubler", font l'objet de fausses détections (figure 3.29). Sur l'ensemble de la base de données, le rappel atteint 95%, ce qui est un résultat plus que satisfaisant étant donné l'utilisation de caractéristiques globales simples.

FIGURE 3.28 – Fausses détections de type "Texte". Les négatifs mal classifiés présentent souvent un aspect texturé proche, comme les rambardes de sécurité.

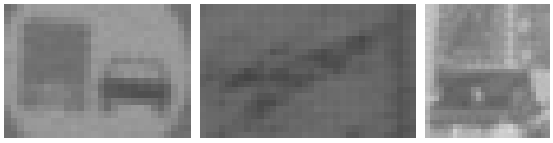


FIGURE 3.29 – Fausses détections de type "Pictogramme". Les négatifs mal classifiés correspondent notamment à des panneaux représentant des véhicules ou ressemblent à un objet sombre sur fond clair.

Pour améliorer ces résultats, nous pourrions tout d'abord prendre en compte la présence ou non d'un panneau d'un autre type en-dessous de celui de limitation de vitesse. Les fausses détections dues, par exemple, au panneau "Interdiction de doubler" seraient éliminées (figure 3.29). Ensuite, en augmentant la résolution de la caméra, nous pourrions distinguer plus facilement les caractères des panonceaux "Texte" qui sont, la plupart du temps, flous pour notre base de données. En outre, [Nienhüser et al., 2010] profitent de l'information couleur pour séparer les négatifs des panonceaux. Cet aspect semble très intéressant car ces derniers possèdent une dominante noir/blanc (ou noir/jaune dans le cas de signalisation temporaire) qui les rend visibles de loin et limite la confusion avec l'arrière-plan. Sur nos images, cette information a été volontairement supprimée pour des raisons de temps de calcul mais pourrait être une piste supplémentaire à étudier.

3.5 Conclusion

Pour mener à bien la reconnaissance des panonceaux, nous avons proposé une méthode à base d'une hiérarchie de SVMs associée à des descripteurs globaux. Les deux niveaux de classification, inspirés des travaux de [Nienhüser et al., 2010], ont montré leur intérêt par rapport à un classifieur seul. Elle nous permet entre autres de supprimer en amont un grand nombre de négatifs et donc de réduire le temps de calcul. Nous avons de plus proposé de décomposer l'ensemble des panonceaux en macro-catégories -"Texte", "Flèche", "Pictogramme" et "Mixte"- afin de simplifier la tâche de classification et par là les descripteurs. En effet, cette approche permet de procéder en plusieurs phases, à savoir Positifs/Négatifs, Macro-catégorie, Catégorie, en vue d'éliminer le maximum de négatifs à chaque étape et peut-être de mettre en œuvre une architecture dédiée pour chaque sous-partie. Cet aspect modulaire favoriserait également le parallélisme de l'algorithme.

De très bons résultats préliminaires ont été obtenus pour les différentes classes. Le rappel atteint au minimum 91% et la précision 83% sur un ensemble de 1790 panonceaux et 4944 négatifs. Ces tests ont été réalisés sur des exemples "parfaits" et sont donc encourageants mais correspondent au cas d'un détecteur "parfait" également. D'un point de vue applicatif, nous étudierons dans le chapitre 4 un système complet de reconnaissance de panonceaux, offrant une nouvelle perspective à notre méthode. Problème de centrage, recouvrement trop faible ou au contraire débordement trop important seront ainsi mis en relation avec les performances de notre architecture.

Chapitre 4

Fusion de données

4.1 Introduction

Dans le cadre de la détermination de la vitesse limite en vigueur, les deux principales raisons pour privilégier un système basé sur la fusion de deux capteurs sont d'une part la non-complémentarité des sources utilisées et d'autre part la volonté d'obtenir un système robuste. En effet, une caméra ne fournit des informations que sur des conditions locales ou temporaires, *i.e.* les panneaux de limitation, et pas sur le contexte général de conduite. Cependant, dans la majorité des situations, seule la connaissance de l'environnement permet de déterminer la vitesse réglementaire. De plus, baser un système ADAS (*Advanced Driver Assistance System* - Système Avancé d'Aide à la Conduite) sur un seul capteur le rend moins robuste aux éventuelles défaillances logicielles ou pannes matérielles. En fusionnant plusieurs capteurs, les chances augmentent d'obtenir des informations redondantes. Pour ces raisons, nous avons opté pour un système basé sur deux capteurs, une caméra et un GPS. La vitesse limite sera obtenue suite à la fusion des données des deux sources.

Dans ce chapitre, nous présentons tout d'abord les aspects théoriques de la fusion au travers de trois techniques majeures, l'inférence bayésienne, la théorie des possibilités et la théorie de l'évidence. Ensuite, nous décrivons les systèmes ISA (*Intelligent Speed Adaptation* - Adaptation Intelligente de la Vitesse) existants basés sur la fusion de données issues de la navigation et de la vision mono-caméra. Après cela, nous présentons notre approche qui consiste en une amélioration de la gestion d'attributs de la cartographie liés au contexte de conduite Nous proposons de les séparer en deux classes afin de quantifier indépendamment la fiabilité du capteur de navigation et la confiance dans chaque vitesse limite. Enfin, nous procédons à l'évaluation et à l'analyse du système que nous comparons à l'existant.

4.2 Théorie sur la fusion de données

4.2.1 Principes généraux

Définitions

Le concept de fusion de données est très général par essence et a donné lieu à de nombreuses applications dans des domaines aussi variés que la médecine, l'aérospatial ou la défense. Commençons par quelques définitions.

- La fusion d'informations consiste à combiner des informations issues de plusieurs sources afin d'améliorer la prise de décision [Bloch and Maitre, 1998].

- La fusion de données est le processus de combinaison de données qui permet d'affiner les estimations et la prédiction [Steinberg et al., 1998].
- La fusion multicapteurs est définie comme le processus de combinaison des entrées de capteurs avec les informations d'autres capteurs, blocs de traitement d'information, bases de données, en une seule représentation [Kokar and Kim, 1994].

Bien que similaires, ces définitions mettent l'accent sur différents aspects de la fusion. La première met en avant le résultat attendu, prendre une décision quant au problème posé au moyen de plusieurs sources. Dans la seconde, la modélisation du monde est mise en avant et réalisée au moyen de données, mais pas nécessairement de sources, différentes. Enfin, la troisième se concentre sur l'architecture générale du système et la manière d'extraire les informations hétéroclites à disposition, qu'il s'agisse de données brutes, de caractéristiques ou de connaissances. Ainsi, en confrontant plusieurs sources, notre compréhension du monde est améliorée.

Imperfections des données

Pour réaliser une fusion efficace, il ne suffit pas de combiner de façon hasardeuse des informations issues de différents capteurs. Il est en effet nécessaire d'avoir à la fois une bonne connaissance du problème à résoudre et de l'aptitude des sources à répondre à ce dernier. Les principaux défauts de qualité des sources sont :

- l'**incertitude** qui caractérise la qualité des informations fournies ou l'assurance de la source en ces dernières.
- l'**imprécision** qui concerne le contenu de l'information et mesure un défaut quantitatif.
- l'**incomplétude**, ou la caractérisation de l'absence d'information apportée par la source sur certains aspects du problème, qui représente l'une des raisons principales pour choisir la fusion. En général, les capteurs utilisés n'ont qu'une vision partielle du monde qui ne met en évidence que certains aspects du problème.
- le **conflit** qui apparaît lorsque deux ou plusieurs informations conduisent à des interprétations contradictoires et donc incompatibles. Pour le gérer, il est possible notamment de supprimer les sources considérées comme non fiables, d'ajouter des informations ou de retarder la fusion jusqu'à l'obtention d'informations concordantes.
- l'**ambiguïté**, ou la capacité d'une information à conduire à plusieurs interprétations, qui peut provenir des imperfections précédentes, comme d'une imprécision qui ne permet pas de choisir entre deux situations. Un des objectifs de la fusion est de lever ces ambigüités en combinant différentes sources.

Toutefois, il faut garder à l'esprit que la fusion n'est pas une solution miracle. Les sources doivent répondre à certains critères pour être éligibles. Tout d'abord, elles doivent être **indépendantes** cognitivement, c'est-à-dire qu'elles ne consultent pas les autres capteurs pris à partie pour en déduire leurs données. Ensuite, elles doivent être **fiables** à deux points de vue, fournir une information et en garantir la certitude. Enfin, deux qualités opposées sont également essentielles, la **redondance** et la **complémentarité**. La redondance caractérise deux sources apportant la même information qui, par le processus de fusion, se trouve renforcée. Imprécision et incertitude s'en trouvent réduites au prix d'une quantité d'information utile plus faible. La complémentarité en revanche quantifie le fait que les sources apportent des informations sur des grandeurs différentes. Cela permet de limiter les situations ambigües.

Modèles

Le concept de fusion de données est très vaste et englobe de nombreuses techniques en fonction du domaine d'application, du problème à résoudre et du type de capteurs utilisés. Afin d'apporter un cadre plus spécifique et de permettre aux chercheurs de modéliser, comparer, évaluer leurs systèmes à base de fusion de données, plusieurs modèles ont vu le jour.

Le premier a été proposé par [Dasarathy, 1997] et se compose de trois niveaux.

- **Bas-niveau**.
  Les informations issues du capteur sont combinées directement sans traitement, comme associer des signaux de même nature pour obtenir une meilleure qualité.
- **Moyen-niveau**.
  À ce premier niveau d'abstraction, les données issues des sources sont interprétées, des caractéristiques sont extraites des données puis fusionnées.
- **Haut-niveau**.
  Il s'agit du plus haut niveau d'abstraction, les données sont souvent très hétérogènes et concernent différents aspects du problème. Il requiert la formulation d'hypothèses issues d'un expert ou d'un système.

Un modèle plus connu a été développé par le groupe DFG (*Data Fusion Group* - Groupe de Fusion de Données) du consortium JDL (*Joint of Directors of Laboratories* - Comité de Directeurs de Laboratoires) dans les années 1990 [White, 1988]. Originalement consacrée aux applications militaires, cette architecture a été élargie par Steinberg *et al.* en 1998 [Steinberg et al., 1998] et en 2004 [Steinberg and Bowman, 2004] pour la rendre plus accessible aux autres communautés scientifiques. Elle est composée de cinq niveaux (figure 4.1) :

Niveau 0 Évaluation des données bas-niveau.
Estimation et prédiction des états des signaux/objets observables à partir des informations brutes des sources.

Niveau 1 Évaluation des objets.
Estimation et prédiction de l'état de chaque entité à partir de son suivi, de sa cinématique, et de l'estimation discrète de son état.

Niveau 2 Évaluation de la situation.
Estimation et prédiction des relations entre les entités (relations physiques, de communication, d'influence, perceptuelles, ...).

Niveau 3 Évaluation de l'impact.
Estimation et prédiction des effets sur la situation des actions planifiées, estimées ou prédites des objets.

Niveau 4 Perfectionnement du processus.
Adaptation des processus d'acquisition et de traitement des informations afin d'atteindre les objectifs fixés.

Cette approche est toutefois nettement orientée vers le domaine militaire et il n'est pas forcément évident de décomposer tous les systèmes de la même façon. Les niveaux peuvent être imbriqués et dépendants les uns des autres.

Architectures

Définissons d'abord quelques notations. Nous disposons de l sources S_j avec $j \in [1; l]$. Chaque source doit prendre une décision sur une observation x dans un ensemble de n décisions $d_1, ..., d_n$. La décision d_i correspond au fait que l'observation x vérifie une hypothèse H_i. Chaque source S_j

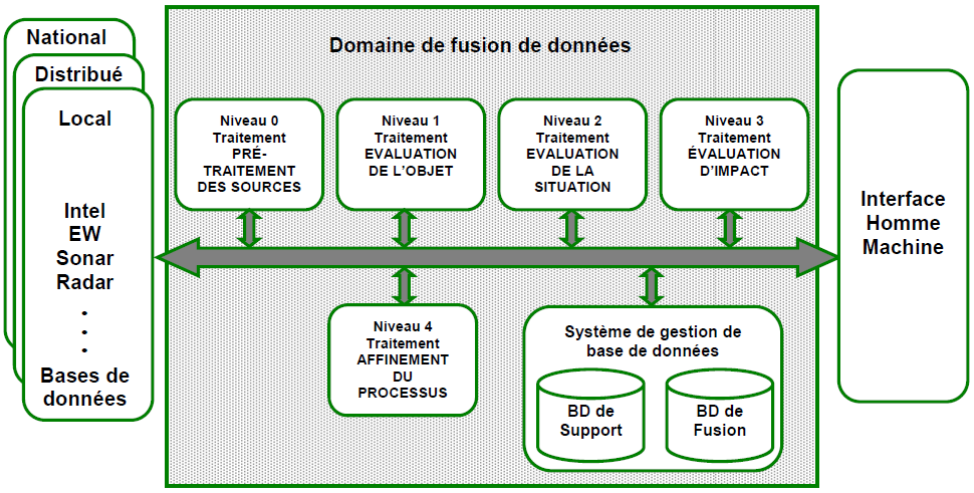


FIGURE 4.1 – Architecture du JDL DFG (source : [Steinberg et al., 1998]).

fournit une information sur la décision d_i pour l'observation x représentée par $M_i^j(x)$.

Les modèles évoqués précédemment se rapportent plus à l'aspect théorique et général de la fusion de données. D'un point de vue pratique se pose la question des moyens utilisés pour exprimer les informations dans le même référentiel pour les combiner. Ce procédé est constitué de quatre étapes : modélisation, estimation, combinaison et décision. La modélisation concerne le choix du formalisme, ou comment vont être représentées les informations à fusionner. Elle est guidée par les connaissances externes d'experts du domaine ou les capteurs utilisés. Elle permet de déterminer M_i^j qui peut être une distribution, une fonction coût, etc. L'estimation n'est pas une étape obligatoire et dépend de la modélisation choisie. La combinaison permet de regrouper les informations à l'aide d'un opérateur défini et compatible avec le formalisme retenu. Enfin, l'étape de décision sert à sélectionner l'hypothèse la plus probable compte tenu des données et d'un critère. Un indice de qualité est calculé pour relativiser l'importance de la décision.

L'architecture du système complet dépendra de la façon dont ces étapes s'organisent et interagissent. Nous pouvons distinguer quatre grands types : centralisé, décentralisé, orthogonal et hybride.

- L'approche, dite globale, correspond à une architecture **centralisée** (figure 4.2). Toutes les informations M_i^j issues des sources S_j convergent vers un seul centre de décision. Ce processus est idéal en théorie car il considère l'information disponible dans son ensemble mais en pratique, il se révèle trop complexe à gérer et peu fiable. En effet, si le centre décisionnel échoue, le système complet est paralysé.
- L'architecture **décentralisée** comprend au contraire plusieurs centres de décisions (figure 4.3). Chaque source S_j agit de manière indépendante et sélectionne une décision $d(j)$ au vu des informations dont elle dispose. Une décision d est prise à partir des m décisions locales. Cette approche est plus modulaire car il est facile d'ajouter des sources et de gérer des arrivées décalées d'informations. Elle apporte donc une réponse rapide grâce à des procédures spécifiques à chaque source. Cette modélisation est particulièrement intéressante lorsque les sources ne fournissent pas d'informations simultanément. Cependant, aucune relation entre les sources n'est prise en compte de cette façon d'où un risque d'apparition de conflits. De plus, cela impose que toutes les sources aient des connaissances sur le même type de données.

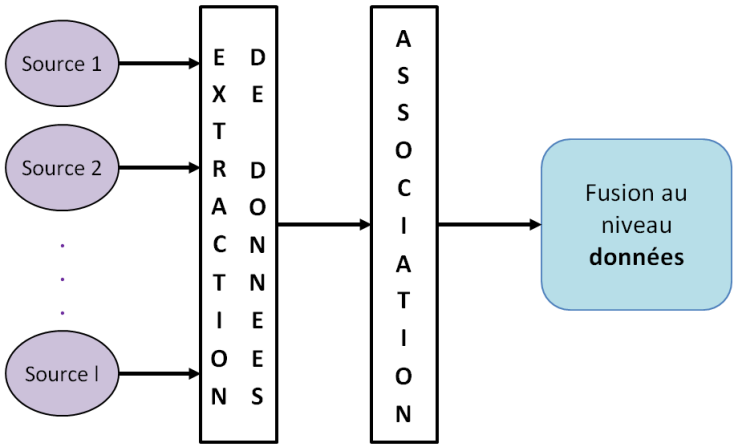


FIGURE 4.2 – Architecture de fusion **centralisée**. Les informations issues des capteurs sont combinées en un seul centre de décision.

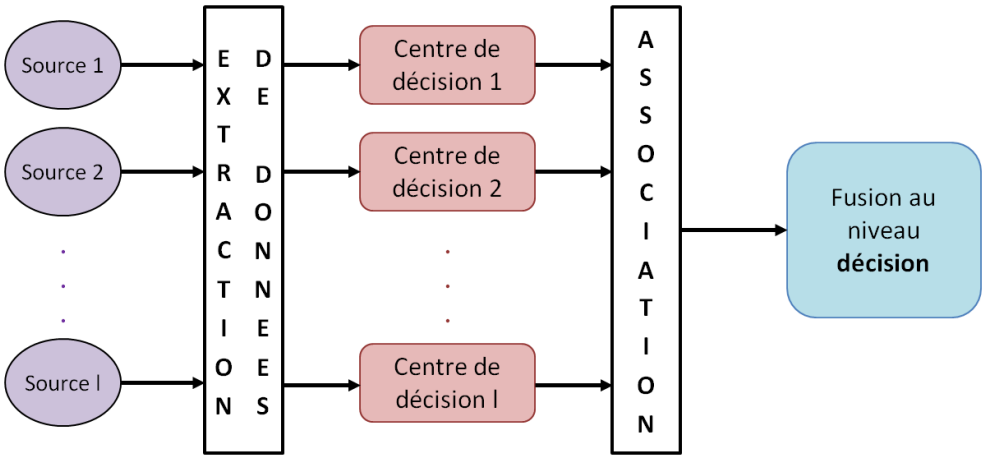


FIGURE 4.3 – Architecture de fusion **décentralisée**. À chaque source est associé un centre de décision indépendant qui sélectionne le meilleur candidat. La fusion consiste alors à combiner ces décisions pour en déduire la plus vraisemblable.

- Le modèle **orthogonal** correspond à un compromis entre les deux précédents. Les données des l sources sont associées pour chaque décision d_i. Aucune décision intermédiaire n'est prise et les informations sont manipulées dans le formalisme choisi jusqu'à la dernière étape, diminuant ainsi les conflits et contradictions.
- L'approche **hybride** consiste à choisir de manière adaptative les informations nécessaires pour un problème donné en fonction des spécificités des sources. Elle fait souvent intervenir des connaissances symboliques sur les sources ou les objets et est très utilisée dans les systèmes à base de règles.

Parmi les principales méthodes de fusion de données, nous pouvons citer la théorie des probabilités, ou inférence bayésienne, la théorie des possibilités, provenant de la logique floue, et la théorie de l'évidence, dite de Dempster-Shafer.

4.2.2 Théorie des probabilités

Les connaissances, teintées d'incertitude, de chaque source S_j sur la probabilité de voir la décision d_i se réaliser sont modélisées par des probabilités conditionnelles $M_i^j = p(d_i|S_j)$. Les solutions possibles au problème sont obligatoirement représentées par des singletons, induisant un espace de définition Ω exclusif, car une seule hypothèse peut être vraie à la fois, et exhaustif, la solution au problème étant obligatoirement présente dans Ω.

Les distributions $p(d_i|S_j)$ sont rarement connues, il faut donc les estimer à l'aide de la règle de Bayes.

$$p(d_i|S_j) = \frac{p(S_j|d_i)p(d_i)}{p(S_j)} \tag{4.1}$$

$p(d_i)$ représente la probabilité *a priori*, sans connaissance des preuves apportées par S_j. La valeur $p(S_j|d_i)$ correspond à la probabilité conditionnelle que S_j ait raison si d_i est vérifiée. Nous pouvons l'estimer par dénombrement, c'est-à-dire à l'aide des fréquences sur une base d'apprentissage. La probabilité marginale $p(S_j)$ est la probabilité que S_j fournisse des preuves données, pour toutes les hypothèses d_i, soit $p(S_j) = \sum_i p(S_j|d_i)p(d_i)$.

L'opération de combinaison peut ensuite se faire de deux façons, au niveau de la modélisation ou par la règle de Bayes. Ainsi, la probabilité *a posteriori* d'une décision d_i connaissant les états des sources S_j peut s'écrire directement :

$$p(d_i|S_1, ..., S_l) = \frac{p(S_1, ..., S_l|d_i)p(d_i)}{p(S_1, ..., S_l)} \tag{4.2}$$

Dans ce cas, les différents termes doivent être estimés par apprentissage. Il est également possibles d'introduire, au fur et à mesure qu'elles arrivent, les informations des sources dans le modèle.

$$p(d_i|S_1, ..., S_l) = \frac{p(S_1|d_i)p(S_2|S_1, d_i)...p(S_l|S_1, ..., S_{l-1}, d_i)p(d_i)}{p(S_1)p(S_2|S_1)...p(S_l|S_1, ..., S_{l-1})} \tag{4.3}$$

Cependant, estimer ces densités de probabilité de même que réaliser l'apprentissage à partir de plusieurs capteurs est complexe. L'hypothèse d'indépendance des sources est souvent posée pour simplifier le problème et l'équation devient :

$$p(d_i|S_1, ..., S_l) = \frac{\prod_{j=1}^{l} p(S_j|d_i)p(d_i)}{\prod_{j=1}^{l} p(S_j)} \tag{4.4}$$

La dernière étape consiste à choisir l'hypothèse d_k la plus vraisemblable, à l'aide des probabilités précédemment calculées. Le critère le plus utilisé est le maximum de probabilité *a posteriori*.

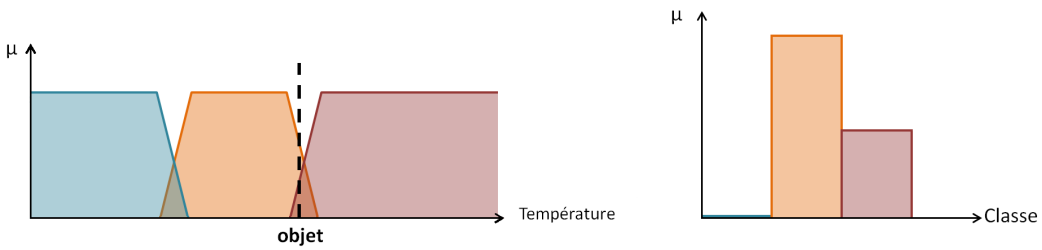
$$p(d_k|S_1, ..., S_l) = \max_{i \in \{1,...,n\}} p(d_i|S_1, ..., S_l) \tag{4.5}$$

La théorie des probabilités repose sur une base mathématique solide et un mode de représentation bien connu. De plus, elle propose une bonne représentation de l'incertitude et l'ajout ou la suppression de sources sont bien gérés. En revanche, elle souffre d'une mauvaise représentation de l'imprécision et de l'absence d'ignorance. Les conflits entre les sources ne sont pas pris en compte et il est souvent difficile d'estimer les probabilités *a priori* $p(d_i)$ sans passer par des hypothèses simplificatrices fortes.

4.2.3 Théorie des possibilités

Généralités

La théorie possibiliste développée par Dubois et Prade [Dubois and Prade, 1988] s'appuie sur la notion de sous-ensemble flou, introduite par Zadeh en 1965 [Zadeh, 1965]. Elle a pour but de modéliser les différentes imprécisions du système, principal défaut de la théorie bayésienne. Elle est particulièrement adaptée lorsque la connaissance *a priori* sur le système est de nature experte. Cette nature subjective se traduit par la définition de classes sans borne stricte ou par l'appartenance partielle ou graduelle d'une donnée à une classe. Prenons par exemple un système permettant de savoir si un objet est chaud, tiède ou froid. Il est possible, soit de modéliser directement les classes de manière floue comme un ensemble de températures, soit de laisser l'expert déterminer l'appartenance de l'objet aux différentes classes (figure 4.4).



(a) Classes avec des bornes floues

(b) Appartenance partielle d'un objet aux différentes classes

FIGURE 4.4 – Exemple d'un système de détermination de température d'un objet. (a) Les classes peuvent être définies de manière floue. Un objet sera d'autant plus froid que sa température baisse. (b) L'expert peut lui-même affecter à l'objet différentes probabilités d'appartenir aux classes froid, tiède et chaud.

Soient S un ensemble net et s, une observation. Un sous-ensemble flou A de S est complètement défini par sa fonction d'appartenance μ de S dans $[0, 1]$. $\mu(s)$ représente le degré d'appartenance de s au sous-ensemble flou A.

$$\mu : \quad S \rightarrow [0, 1] \tag{4.6}$$

$$s \rightarrow \mu(s) \tag{4.7}$$

Un tel sous-ensemble se caractérise par :

- son support $Supp$, ou l'ensemble des éléments de S qui appartiennent en partie à la classe ;
- sa hauteur h, qui correspond au plus fort degré d'appartenance d'un élément de S à la classe ;
- son noyau, ensemble de tous les éléments appartenant de façon absolue (avec un degré 1) à la classe.

La théorie des possibilités permet d'exprimer l'imprécision et l'incertitude d'une hypothèse à l'aide de la distribution de possibilité Π et de nécessité N définies sur $P(S)$, l'ensemble des parties de S dans $[0, 1]$.

Les propriétés de ces deux distributions sont les suivantes :

$$\Pi(\emptyset) = 0 \quad \Pi(S) = 1 \tag{4.8}$$
$$N(\emptyset) = 0 \quad N(S) = 1 \tag{4.9}$$
$$\Pi(A) < N(A) \tag{4.10}$$
$$N(A) = 1 - \Pi(\bar{A}) \tag{4.11}$$
$$\forall i \in I \subset \mathbb{N}, \forall A_i \subseteq S, \quad \Pi(\cup_{i \in I} A_i) = sup_{i \in I} \Pi(A_i) \tag{4.12}$$
$$\forall i \in I \subset \mathbb{N}, \forall A_i \subseteq S, \quad N(\cap_{i \in I} A_i) = inf_{i \in I} N(A_i) \tag{4.13}$$

Plus la valeur de possibilité est proche de 1, plus la réalisation de l'évènement concerné est possible. Plus la nécessité est proche de 1, plus la certitude dans la réalisation de l'évènement est importante.

$$\Pi(A) = 0 \quad \Leftrightarrow \quad \text{A est impossible} \tag{4.14}$$
$$\Pi(A) = 1 \quad \Leftrightarrow \quad \text{A est complètement possible mais non certain} \tag{4.15}$$
$$N(A) = 1 \quad \Leftrightarrow \quad \text{A est certain} \tag{4.16}$$

L'étape de définition de $P(S)$ peut être évitée par l'utilisation de distributions de possibilités π qui attribuent un degré de possibilité à tout élément s de S et non plus de $P(S)$. Elles doivent vérifier la propriété de normalisation suivante :

$$\pi : S \to [0, 1], \quad \sup_{s \in S} \pi(s) = 1 \tag{4.17}$$

$\pi(s)$ indique le degré de possibilité pour que l'observation x soit égale à s. Cette condition correspond à une hypothèse de monde fermé, dans lequel au moins un élément de S est complètement possible. Dans le cas fini, une mesure de possibilité peut être construite à partir d'une distribution de possibilités par la formule :

$$\forall A \subseteq S, \Pi(A) = \sup_{s \in A} \pi(s) \tag{4.18}$$

Application à la fusion

À partir d'une observation x, une source S_j fournit une information, notée $M_i^j(x)$, sur la décision d_i. Deux modélisations sont alors possibles : exprimer le degré d'exactitude de la décision d_i prise pour x selon S_j (équation 4.19) ou représenter la possibilité $\pi_x^j(d_i)$ que d_i soit valide pour x (équation 4.20).

$$M_i^j(x) \quad = \mu_i^j(x) \tag{4.19}$$
$$M_i^j(x) \quad = \pi_x^j(d_i) \tag{4.20}$$

Toutefois, quelle que soit la modélisation choisie, la phase d'estimation se révèle délicate. Elle peut s'inspirer des méthodes statistiques d'apprentissage, d'heuristiques ou de minimisation de critères de classification.

La combinaison des différentes sources réalisée ensuite varie en fonction de l'importance du conflit, correspondant au maximum de l'intersection entre deux distributions de possibilités. Si le degré de conflit est faible, les opérateurs conjonctifs, assimilables aux "ET logiques" sont utilisés. Toutes les solutions données comme possibles par une source sont conservées, ce qui suppose

qu'au moins une d'entre elles est fiable. Au contraire, les opérateurs disjonctifs sont utilisés dans le cas de sources discordantes. Les opérateurs adaptatifs permettent de gérer ces deux situations, en agissant comme un maximum en cas de fort conflit et de minimum sinon. Toutefois, le choix de ces opérateurs est complexe et nécessite une bonne connaissance à la fois des propriétés de l'opérateur et de son comportement en terme de décision et de réaction face au conflit. La solution retenue finalement pour l'observation x correspond au maximum des degrés d'appartenance (équation 4.21).

$$x \in d_k^*, k^* = argmax_k\{\mu_k, 1 \leq k \leq n\} \tag{4.21}$$

La qualité de la décision dépendra de sa netteté, par comparaison du degré d'appartenance maximum avec un seuil fixé, et de sa capacité de discrimination, par rapport à la deuxième plus forte valeur.

Le principal avantage de cette technique par rapport aux probabilités est sa souplesse qui permet de modéliser les connaissances imprécises d'expert. Toutefois, elle est également un inconvénient par la difficulté de définir les fonctions utilisées. De plus, l'incertitude n'est pas explicitement définie et seulement accessible par déduction.

4.2.4 Théorie de Dempster-Shafer

Généralités

La théorie de l'évidence, ou théorie des croyances, est introduite par Dempster en 1967 [Dempster, 1967] et formalisée mathématiquement par Shafer en 1976 [Shafer, 1976]. Elle généralise la théorie des probabilités en introduisant la notion d'ignorance. Ce formalisme est particulièrement adapté pour gérer les incertitudes et imprécisions car, contrairement à la théorie des probabilités, le modèle s'applique aussi bien aux singletons qu'aux sous-ensembles de décisions. Le degré de croyance en un évènement est modélisé par une fonction de masse m, rendant possible l'évaluation des conflits entre sources ainsi que leur fiabilité respective. Les n solutions possibles d_i d'un problème forment le *cadre de discernement* Ω. L'ensemble de toutes les disjonctions possibles de solutions s'appelle le référentiel de fonctions de masse 2^Ω.

$$\Omega = \{d_1, d_2, ..., d_n\} \tag{4.22}$$

$$2^\Omega = \{\emptyset, \{d_1\}, \{d_2\}, ..., \{d_n\}, \{d_1 \cup d_2\}, ..., \Omega\} \tag{4.23}$$

La fonction de masse $m_j(A)$ modélise le niveau de croyance de la proposition $A \subset 2^\Omega$ fourni par la source S_j. Elle est définie sur $[0, 1]$ et satisfait à la relation :

$$\sum_{A \in 2^\Omega} m_j(A) = 1 \tag{4.24}$$

Si les critères d'exclusivité et d'exhaustivité sont respectés, toutes les décisions sont comprises dans Ω et le système est qualifié de *monde fermé* vérifiant l'équation :

$$m_j(\emptyset) = 0 \tag{4.25}$$

La masse attribué à Ω, quant à elle, est interprétée comme l'absence de connaissance de la source, d'autant plus grande que la fiabilité du capteur diminue. Tout sous-ensemble A de Ω tel que $m(A) > 0$ est nommé élément focal. La réunion de tous les éléments focaux forme le noyau.

Deux fonctions quantifient les informations fournies par les sources, la crédibilité Cr et la plausibilité Pl. La première est une fonction croissante de 2^Ω dans $[0,1]$ et mesure la croyance minimum en A, c'est-à-dire à quel point les sources soutiennent la proposition A. La fonction de plausibilité, ou croyance maximum en A, permet de mesurer à quel point les informations issues d'une source ne s'opposent pas à la proposition A.

$$\forall A \in 2^\Omega, Cr_j(A) = \sum_{B \subseteq A, B \neq \emptyset} m_j(B) \tag{4.26}$$

$$\forall A \in 2^\Omega, Pl_j(A) = \sum_{B \cap A \neq \emptyset} m_j(B) = 1 - Cr_j(\bar{A}) \tag{4.27}$$

Les relations entre crédibilité, plausibilité et masses de croyance sont obtenues par la formule d'inversion de Möbius.

$$\forall A \in 2^\Omega, m(A) = \sum_{B \subset A} (-1)^{|A|-|B|} Cr(B) \tag{4.28}$$

$$\forall A \in 2^\Omega, m(A) = \sum_{B \subset A} (-1)^{|A|-|B|+1} Pl(\bar{B}) \tag{4.29}$$

Une hypothèse est définie sans ambiguïté lorsque sa crédibilité est supérieure à la plausibilité de tout autre hypothèse. L'intervalle $[Cr_j(A), Pl_j(A)]$ est appelé intervalle de confiance et sa longueur représente parfois une mesure de l'ignorance en un évènement A. Lorsque les masses sont uniquement affectées aux singletons, les trois fonctions m, Cr et Pl sont égales et correspondent à une probabilité. Dans le cas contraire, la théorie des croyances permet une modélisation souple et riche, en particulier pour gérer les ambigüités et hésitations entre classes. La théorie des probabilités apparaît donc comme la limite de la théorie de Dempster-Shafer.

Combinaison

De nombreux opérateurs de combinaison existent pour fusionner les informations issues de l capteurs. Parmi les plus utilisés, nous pouvons citer les règles conjonctives et disjonctives. La première s'appuie sur la règle orthogonale de Dempster et permet de combiner des sources indépendantes et fiables.

$$\forall A \in 2^\Omega, m_1 \oplus m_2 \oplus ... \oplus m_l(A) = \sum_{B_1 \cap B_2 \cap ... \cap B_l = A} m_1(B_1) m_2(B_2) ... m_l(B_l) \tag{4.30}$$

En général, la masse affectée à l'ensemble vide $\emptyset$ n'est pas nulle et peut s'interpréter comme le conflit K entre les sources.

$$K = \sum_{B_1 \cap B_2 \cap ... \cap B_l = \emptyset} \prod_{j=1}^{l} m_j(B_j) \tag{4.31}$$

Ce dernier peut avoir différentes origines : une non-exhaustivité de l'espace Ω, un manque de fiabilité ou un désaccord des sources. Si la modélisation du système assure que toutes les solutions du problème sont dans Ω, une opération de normalisation est alors mise en place.

$$m(A) = \frac{1}{1-K} \sum_{B_1 \cap ... \cap B_l = A} \prod_{j=1}^{l} m_j(B_j) \qquad \text{si } A \neq \emptyset \tag{4.32}$$

$$m(\emptyset) = 0 \tag{4.33}$$

La combinaison disjonctive s'applique sur des sources indépendantes dont l'une au moins est fiable.

$$\forall A \in 2^{\Omega}, m_1 \ominus m_2 \ominus ... \ominus m_l(A) = \sum_{B_1 \cup B_2 \cup ... \cup B_l = A} m_1(B_1) m_2(B_2) ... m_l(B_l) \tag{4.34}$$

De nombreux autres opérateurs ont été définis dans la littérature [Sentz and Ferson, 2002] et dépendent de l'application choisie.

4.2.5 Affaiblissement - Renforcement

En 1976, [Shafer, 1976] introduit l'opération d'affaiblissement, permettant de prendre en compte la fiabilité d'une source d'information S_j. Cette opération est contrôlée par un coefficient $\alpha_j \in [0, 1]$ appelé taux d'affaiblissement. Les fonctions de masse affaiblies $\bar{m_j}$ correspondantes sont alors modifiées de la façon suivante :

$$\begin{cases} \bar{m_j}(A) = \alpha_j m_j(A) \\ \bar{m_j}(\Omega) = (1 - \alpha_j) + \alpha_j m_j(\Omega) \end{cases} \tag{4.35}$$

Plus la valeur de α_j est faible, plus les masses du cadre de discernement sont affaiblies et l'essentiel de la masse est transféré sur Ω. Dans le cas où $\alpha_j = 0$, la source S_j n'est pas du tout fiable, toute la masse est affectée à Ω. La valeur $\alpha_j = 1$ caractérise une source fiable, les évènements A concentrent toute la masse et aucune ambiguïté ne demeure entre les classes. La distinction établie entre le facteur de confiance α_j et la fonction de masse m_j témoigne du fait que ces deux grandeurs peuvent être obtenues indépendamment l'une de l'autre, par des experts différents ou à partir de deux points de vue différents.

4.2.6 Décision

Plusieurs critères de décision existent pour déterminer l'hypothèse la plus vraisemblable ${d_i}^*$, en fonction de la rigueur du résultat attendu. Le maximum de plausibilité, quant à lui, permet de déterminer la meilleure hypothèse au sens du degré de confiance et constitue un critère de décision optimiste. Toutes les hypothèses qui ne contredisent pas ${d_i}^*$ seront prises en compte.

$$Pl({d_i}^*)(x) = \max_{1 \leq k \leq n} Pl(d_k)(x) \tag{4.36}$$

Le maximum de crédibilité détermine la meilleure hypothèse au vu de son degré de croyance et constitue un critère de décision pessimiste. Il ne prend en effet en compte que les hypothèses qui soutiennent entièrement ${d_i}^*$.

$$Cr({d_i}^*)(x) = \max_{1 \leq k \leq n} Cr(d_k)(x) \tag{4.37}$$

4.3 État de l'art de la fusion de données dans les systèmes ISA

Dans le cadre de notre application, la détermination de vitesse limite, la fusion de données voit son intérêt dans la combinaison de deux capteurs complémentaires, la caméra et le GPS. Chacun des deux possède une vision partielle du problème. Le premier détecte les limitations locales ou temporaires tandis que le second a une connaissance plus générale de l'environnement.

L'accent a cependant été principalement mis ces dernières années sur le développement de systèmes de détection de panneaux [Bahlmann et al., 2005, de La Escalera et al., 2003, Ruta et al., 2010]. De très bonnes performances ont même été atteintes dans ce domaine. Malheureusement, il

ne s'agit là que de vitesses temporaires ou locales, une caméra ne permettant pas de connaître les vitesses implicites, *i.e.* ayant cours en l'absence de panneaux et précisées par le Code de la Route. Un dispositif visant à déterminer la vitesse limite doit donc être nanti à la fois d'un système de vision et d'un dispositif permettant une connaissance des vitesses liées à la situation de conduite, via un capteur de navigation. Toutefois, l'imperfection des capteurs et les confrontations avec des situations inhabituelles, comme la présence de limites variables en cas de travaux, risquent de faire naître des conflits. L'utilisation simultanée de ces deux capteurs nécessite donc l'usage d'un processus de fusion pour établir la limite de vitesse la plus probable. La décision consistera à choisir parmi les vitesses possibles la plus vraisemblable en fonction des informations que le système aura récupérées de son environnement. Nous allons présenter les quelques systèmes existants basés sur l'utilisation conjointe d'une caméra et d'un système de navigation qui diffèrent tant par le formalisme choisi que les informations prises en compte.

4.3.1 Contexte bayésien

Le système développé par [Bahlmann et al., 2008] met en jeu le cadre bayésien, supposant que les sorties des deux capteurs correspondent à des mesures statistiques. Le schéma du système complet est donné par la figure 4.5. À l'évidence, l'architecture est centralisée puisque la décision n'est prise qu'après l'estimation des probabilités de chaque limite de vitesse par les deux sources.

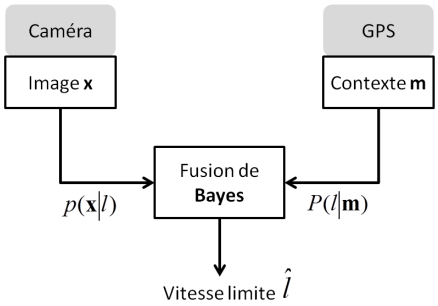


FIGURE 4.5 – Schéma du système de fusion entre vision et navigation mis en œuvre par [Bahlmann et al., 2008].

D'un côté, le système de vision détecte dans l'image les instances de panneaux et fournit, à partir des observations $\mathbf{x}$, une vitesse $l \in \mathbb{L} = \{5, 10, 20, ..., 130, 999\}$ (la valeur 999 correspond à une absence de limitation) et un résultat $u_l(\mathbf{x})$ obtenu généralement par un classifieur statistique et assimilable à la probabilité *a posteriori* $p(l|\mathbf{x})$. La règle de Bayes nous permet ensuite de calculer la vraisemblance $p(\mathbf{x}|l)$.

$$\forall l \in \mathbb{L}, p(l|\mathbf{x}) \propto P(l)p(\mathbf{x}|l) \tag{4.38}$$

De l'autre côté, les informations issues de la cartographie, type de route m_{ST} et limitation de vitesse m_{SL}, forment un t-uple $\mathbf{m}$ associé à une probabilité $P(l|m)$ qui modélise le contexte cartographique.

$$m_{ST} \in \mathbb{M}_{ST} = \{\emptyset, \text{autoroute, route nationale, route rurale, ville, zone calme}\} \tag{4.39}$$

$$m_{SL} \in \mathbb{M}_{SL} = \{\emptyset, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 999\} \tag{4.40}$$

$\emptyset$ signifie qu'aucune information n'est disponible, par exemple si la route n'a pas été cartographiée ou que les champs utiles n'ont pas été renseignés.

Soit une image obtenue par le système de vision $\mathbf{x}$ et un t-uple $\mathbf{m}$ de contexte, le système de [Bahlmann et al., 2008] assigne $\mathbf{x}$ à la classe $\hat{l}$ avec la plus forte probabilité *a posteriori*.

$$\hat{l} = argmax_l\{P(l|\mathbf{x},\mathbf{m})\} \tag{4.41}$$

$P(l|\mathbf{x},\mathbf{m})$ ne peut pas être mesurée directement mais déduite des valeurs de $p(\mathbf{x}|l)$ et $P(l|\mathbf{m})$. Puisque les connaissances issues de la navigation ne modifient pas la vraisemblance de voir $\mathbf{x}$, les deux capteurs peuvent être considérés comme indépendants, ce qui conduit à $p(\mathbf{x}|l,\mathbf{m}) = p(\mathbf{x}|l)$.

$$P(l|\mathbf{x},\mathbf{m}) \propto p(\mathbf{x}|l,\mathbf{m})P(l|\mathbf{m}) \tag{4.42}$$

$$= p(\mathbf{x}|l)P(l|\mathbf{m}) \tag{4.43}$$

$p(\mathbf{x}|l)$ étant obtenue directement par l'application de l'équation 4.38 sur la sortie du classifieur statistique du système de vision, il reste à modéliser $P(l|\mathbf{m})$, la modélisation probabiliste du contexte du GPS. La première possibilité serait d'estimer $P(l|\mathbf{m})$ à partir d'un ensemble d'observations $\{(l^1,\mathbf{m}^1),...,(l^M,\mathbf{m}^M)\}$. Cependant, l'ensemble des évènements possibles contient $L.|\mathbb{M}_{ST}|.|\mathbb{M}_{SL}|$ éléments. Pour obtenir des estimations fiables, il faudrait d'énormes quantités de segments de route étiquetés, données qui ne sont pas disponibles. La seconde solution utilise des connaissances *a priori* fournies par un expert sur la vitesse limite l en vigueur dans un contexte particulier $\mathbf{m}$. $P(l|\mathbf{m})$ dérive d'un modèle basé sur des règles de régulation de trafic. Le cas français est décrit ci-dessous.

1. Le domaine des vitesses limites autorisées en fonction du type de route est le suivant :
   $l \in \{50,...,130\}$ si m_{ST} = autoroute
   $l \in \{50,...,110\}$ si m_{ST} = route nationale
   $l \in \{50,...,90\}$ si m_{ST} = route rurale
   $l \in \{30,...,80\}$ si m_{ST} = ville
   $l \in \{5,...,30\}$ si m_{ST} = zone de prudence
2. En présence d'un panneau de fin de limitation de vitesse, les limites en vigueur sont mises à :
   $l = 130$ si m_{ST} = autoroute
   $l = 110$ si m_{ST} = route nationale
   $l = 90$ si m_{ST} = route rurale
   $l = 50$ si m_{ST} = ville
   $l = 30$ si m_{ST} = zone calme
3. Les hypothèses suivantes sont faites sur les relations entre la vraie limite et celle fournie par la carte :
   (a) la vraie limite l^* n'est jamais plus élevée que celle donnée par la navigation m_{SL}. Cette hypothèse repose sur l'idée que l^* et m_{SL} diffèrent seulement en zone de travaux, lorsque la limite est variable et sur les routes qui ne possèdent aucune information ($m_{SL} = \emptyset$).
   (b) la classe $l \in \mathbb{L}$ qui correspond à la vitesse fournie par la navigation m_{SL} possède la probabilité la plus forte $P(l \mid m) = a$.
   (c) toutes les autres limites de vitesses l telles que $l \neq m_{SL}$, autorisées par la règle 1 possèdent une probabilité plus faible, $P(l \mid m) = b, b < a$.

Les configurations exclues par le modèle ont alors une probabilité de 0. Cela permet d'éliminer un certain nombre de mauvaises classifications du système de vision. La contrainte $a > b$ favorise les vitesses limites en accord avec la navigation. Le résultat de la fusion sera donc particulièrement

dépendant du choix de ces critères.

Plusieurs limitations apparaissent dans cette approche. Tout d'abord, Bahlmann *et al.* évoquent le problème de la synchronisation des données que nous retrouverons dans toutes les autres applications. En effet, le contexte de navigation **m** se réfère au segment courant alors que la vision détecte des panneaux avant qu'ils ne soient atteints, *i.e.* avant que la limitation ne soit effective. Une solution consiste à réaliser la fusion avec un "retard" pour que le véhicule se situe sur la bonne portion de route. Ensuite, il n'est pas fait mention de l'origine des informations de la navigation. Il semble que toutes les données utiles soient bien stockées dans la base et que le capteur soit supposé parfaitement fiable. Cependant, si le type de route n'est pas accessible, la source devient totalement aveugle. Son rôle est donc relativement réduit et ne sert qu'à "assister" la vision. Enfin, aucune gestion des conflits n'est faite. Si la vision fait une mauvaise détection qui ne correspond à aucun élément focal de la navigation, la fusion échoue.

4.3.2 Prise en compte du contexte de situation

[Nienhüser et al., 2009] proposent de prendre en compte la fiabilité de chaque capteur pour améliorer la fusion. Ils mettent en avant les principales causes de défaillance. Les zones de travaux ou à message variable pénalisent la navigation qui souffre de son aspect statique. Les mauvaises conditions météorologiques ou de luminosité, comme le contrejour, font elles chuter les performances de la vision. L'identification de telles conditions permettra de modifier en temps-réel la fiabilité des sources et d'améliorer la prise de décision.

Le système de navigation de Nienhüser *et al.* interroge en temps-réel une base de données qui fournit la situation de conduite, dont le type de route, la vitesse réglementaire et les messages variables, si présents. La caméra a pour but de détecter les panneaux de limitation ainsi que les zones de travaux ou les conditions climatiques (tableau 4.1).

Source	Variable	Valeur
Caméra	Zone de travaux	oui, non
	Période de la journée	jour, aube, nuit
	Conditions climatiques	sec, brouillard, pluie
Navigation	Limites variables	oui, non

TABLE 4.1 – Variables environnementales affectant la fiabilité des capteurs caméra et navigation utilisées dans [Nienhüser et al., 2009].

Le cadre de fusion est la théorie de Dempster-Shafer. Les masses de confiance m_{cam} et m_{nav}, de la caméra et de la navigation respectivement, sont combinées à l'aide de la règle conjonctive normalisée (équation 4.32). La cadre de discernement utilisé Ω comprend toutes les vitesses possibles.

$$\Omega = \{5, 10, 20, 30, 45, 50, 60, 70, 80, 90, 100, 110, 120, 130, 999\} \tag{4.44}$$

Les masses de confiance relatives aux différents capteurs sont ensuite calculées. Le type de route, la vitesse limite v_{nav} et la présence de messages variables conditionnent la fonction de masse m_{nav}. Le premier fournit des restrictions quant à l'ensemble des vitesses limites possibles en fonction de la seconde à l'aide d'*a priori*. Si aucune vitesse limite n'est donnée explicitement, la vitesse implicitement liée au type de route est alors utilisée par Nienhüser *et al.* (tableau 4.2).

Attributs	Fonctions de masse
type = ville $v_{nav} = \emptyset$	$m_{nav}(\{50\}) = a$ $m_{nav} = (\{30, ..., 70\}) = b$
type = ville $v_{nav} = 70$	$m_{nav}(\{70\}) = a$ $m_{nav}(\{30, ..., 60\}) = b$
type = autoroute $v_{nav} = 130$	$m_{nav}(\{130\}) = a$ $m_{nav}(\{50, ..., 120, 999) = b$
type = autoroute v_{nav} = variable	$m_{nav}(\{50, ..., 130, 999\}) = b$

TABLE 4.2 – Exemples de fonctions de masse m_{nav} pour les vitesses limites en fonction de la situation de conduite "type" selon [Nienhüser et al., 2009]. Dans le premier cas, la navigation nous localise en ville mais aucune vitesse limite n'est stockée dans la base. La valeur par défaut est $v_{nav} = 50$ d'où une masse $m_{nav}(\{50\}) = a$. Dans le second cas, pour le même type de situation, nous avons une vitesse limite stockée $v_{nav} = 70$. La masse de confiance est donc cette fois-ci $m_{nav}(\{70\}) = a$. Dans le dernier cas, v_{nav} = variable signifie que nous nous trouvons dans une zone à message variable.

Les valeurs a et b ajustent les fonctions de masse à la situation et s'apparentent à celles utilisées par [Bahlmann et al., 2008]. Lorsque la navigation indique une zone comprenant des messages variables (v_{nav} = variable), seul un sous-ensemble de vitesses possibles en fonction du contexte bénéficiera d'une masse de confiance non-nulle (tableau 4.2, cas 4).

$$a = \begin{cases} \mu_a & \text{Limites variables = non} \\ & \text{Zone de travaux = non} \\ 0 & \text{sinon} \end{cases}$$
$$b = \begin{cases} \mu_b & \text{Limites variables = non} \\ & \text{Zone de travaux = non} \\ \mu_a + \mu_b & \text{sinon} \end{cases}$$

où μ_a et μ_b sont des constantes déterminées empiriquement.

Le système de vision fournit à la fois des vitesses limites v_{cam}, liée aux panneaux détectés, et un indice de confiance $p(v_{cam})$, dépendant du résultat de la reconnaissance et du suivi temporel des panneaux.

$$\begin{cases} m_{cam}(\emptyset) = 0 \\ m_{cam}(\{v_{cam}\}) = c.p(v_{cam}) \\ m_{cam}(\Omega) = 1 - c \end{cases} \tag{4.45}$$

La valeur c adapte la fiabilité à la situation. La caméra risque de voir son crédit diminuer en cas de forte variation de luminosité ou de mauvaises conditions climatiques.

$$c = \begin{cases} \mu_{nuit} & \text{Période de la journée = nuit} \\ \mu_{aube} & \text{Période de la journée = aube} \\ \mu_{climat} & \text{Conditions climatiques } \neq \text{ sec} \\ \mu_{cam} & \text{sinon} \end{cases} \tag{4.46}$$

avec μ_{nuit}, μ_{aube}, μ_{climat} et $\mu_{cam} \in [0, 1]^4$ déterminées empiriquement.

Lorsque le conflit K est trop important ou que la meilleure solution, au sens du maximum de crédibilité obtenu après fusion $Cr_{cam,nav}$, n'est pas assez discriminante, la règle de décision ne

permet pas de conclure quant à la vitesse légale $\hat{l}$.

$$\hat{l} = \begin{cases} \emptyset & \text{si} \quad K > c_{conflit} \\ \emptyset & \text{si} \quad argmax_{l \in \Omega} Cr_{cam,nav}(\{l\}) < c_{min} \\ argmax_{l \in \Omega} Cr_{nav,cam}(\{l\}) & \end{cases} \tag{4.47}$$

Le principal avantage de cette technique est l'utilisation de la théorie de Dempster-Shafer qui permet de bien gérer les conflits entre sources et d'attribuer des masses de confiance à des sous-ensembles et non plus seulement à des singletons. Nienhüser *et al.* mettent en avant les principales sources de conflit et proposent une solution pour les détecter et adapter le système. Des modules spécialisés sont ainsi développés pour déterminer les zones de travaux ou les conditions climatiques. Toutefois, cette implémentation conduit à un autre problème, la non-indépendance des capteurs. En effet, l'essentiel des modules se base sur le capteur de vision pour étudier l'environnement et adapter la fiabilité du capteur... de vision ! Ce dernier est donc considéré comme fiable pour déterminer le contexte mais pas pour reconnaître les panneaux. De même, l'information issue de la navigation est considérée "parfaite" et n'est pas remise en cause.

4.3.3 Critères liés au GPS

[Lauffenburger et al., 2008] mettent également en pratique la théorie de Dempster-Shafer mais traitent de façon plus détaillée les informations issues de la navigation. Le cadre de discernement regroupe toutes les vitesses légales en France (équation 4.44). Les vitesses limites de 5 à 30km/h s'appliquent généralement aux zones calmes (voisinage d'école, parkings, etc.). La valeur 999 se réfère aux situations où aucune limite n'est imposée.

La figure 4.6 montre comment les informations sont extraites des deux sources d'information et les masses de confiance m_{cam} et m_{nav}. Le capteur de vision, comme dans la majorité des applications TSR (*Traffic Sign Recognition* - Reconnaissance de Panneaux Routiers), détecte et reconnaît les panneaux de limitation de vitesse présents dans l'image et retourne la masse de confiance associée m_{cam}. Aucune vitesse n'est retournée lorsqu'un signe ne peut pas être détecté (à cause d'un obstacle notamment). La fonction de masse est calculée à partir des résultats de la reconnaissance et de l'intégration temporelle et diminue au cours du temps, modélisant le fait qu'un panneau n'est applicable que sur une distance limitée (en réalité, jusqu'à la prochaine intersection).

À partir de la navigation, [Lauffenburger et al., 2008] proposent d'extraire six critères pour évaluer le contexte de conduite et les masses de confiance relatives aux différentes vitesses.

- C_1, indice de confiance dans le positionnement.
  La position du véhicule, déterminée à l'aide du GPS, est fusionnée avec les mesures odométriques et inertielles du véhicule puis mise en correspondance avec la carte. La pertinence du résultat est donné par l'indice MLCP (*Most Likely Candidate Probability* - Probabilité du Candidat le Plus Vraisemblable).
- C_2, qualité de numérisation du réseau routier.
  Il indique si le chemin considéré est de définition suffisante pour permettre son utilisation dans une application ADAS.
- C_3, classe fonctionnelle de la route.
  Les routes sont classées en fonction de leur importance selon différents niveaux. Européennes et autoroutes appartiennent par exemple au niveau FC_1.
- C_4, type de route (européenne, autoroute, nationale, etc.).

– C_5, situation de conduite et contexte (ville, intersections, sortie de route).
– C_6, activation/désactivation du mode guidage.

La vitesse limite SL_{nav} spécifiée dans la base de données est ensuite utilisée pour déterminer un ensemble d'éléments focaux (tableau 4.3). Cette approche a pour but de prendre en compte les imprécisions et incertitudes de la navigation (erreurs de positionnement, de numérisation de la carte, données obsolètes, etc.).

Vitesse limite de la base (km/h)	Élément focaux supplémentaires (km/h)
5	50
10	50
20	50
30	50
45	30, 50
50	30, 70, 100, 110, 120, 130, 999
60	$\emptyset$
70	50, 80, 90
80	50, 60, 70, 90
90	50, 70
100	50, 70, 999
110	50, 70
120	50, 70, 999
130	50, 70
999	$\emptyset$

TABLE 4.3 – Liste des vitesses jugées possibles (éléments focaux) en fonction de la vitesse spécifiée dans la carte numérique SL_{nav} selon [Lauffenburger et al., 2008]. L'objectif est d'intégrer les imprécisions du GPS, comme l'erreur de positionnement, l'obsolescence de certaines données, etc.

La méthode de [Lauffenburger et al., 2008] calcule les fonctions de masse m_{nav} des éléments focaux SL du système de navigation via une somme des différents critères $C_i, i \in [1;6]$ pondérés par des coefficients α_i, en fonction de leur importance respective. Ces poids sont généralement définis par expérience ou expertise.

$$m_{nav}(SL) = \frac{\sum_{i=1}^{6} \alpha_i C_i(SL)}{\sum_{i=1}^{6} \alpha_i} \tag{4.48}$$

[Daniel and Lauffenburger, 2011] ont proposé une adaptation de cette méthode basée sur des sources spécialisées [Rombaut, 1998], *i.e.* n'ayant de connaissance qu'en un seul évènement A. Cette modélisation permet d'éviter la définition de masses de confiance dans des propositions antagonistes et les seules définies seront A, $\bar{A}$ et Ω. Ils proposent également une fusion décentralisée, les modules de vision et de navigation sélectionnant en amont la vitesse limite la plus probable à l'aide des informations dont ils disposent (figure 4.7).

Au niveau de la caméra, les panneaux détectés dans l'image permettent de déterminer directement la vitesse limite. Pour la navigation, Daniel *et al.* considèrent tous les attributs comme autant de sources spécialisées qu'ils combinent ensuite à l'aide de la règle de combinaison conjonctive. L'objectif de cette étape est de repérer le plus tôt possible des incohérences au sein de ce capteur qui pourraient générer des conflits. Les vitesses les plus probables de chaque module,

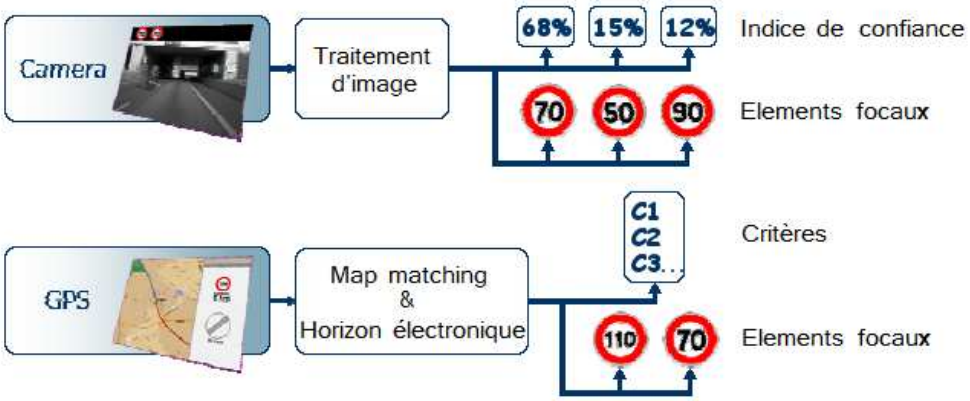


FIGURE 4.6 – Données extraites des deux capteurs selon [Lauffenburger et al., 2008]. La vision indique les panneaux rencontrés associés à un indice de confiance. La navigation extrait de la cartographie des critères liés au segment de route, ainsi qu'un ensemble de vitesses possibles. La fusion est ensuite réalisée à l'aide de la théorie de Dempster-Shafer.

associées à une mesure de fiabilité du capteur, sont finalement combinées. Une opération de redistribution du conflit est réalisée avant de prendre la décision sur la vitesse limite à l'aide du maximum de crédibilité.

Cette technique exploite les avantages de celle de [Lauffenburger et al., 2008], l'utilisation d'attributs de navigation, tout en y apportant certaines améliorations. La prise en compte de la fiabilité permet de relativiser la qualité des informations fournies par les différents capteurs et ainsi de mieux estimer l'origine des conflits. L'utilisation de sources spécialisées est une alternative élégante à la somme pondérée (équation 4.48). Toutefois, la sélection d'une seule vitesse au niveau de chaque module peut présenter un risque, celui d'éliminer une solution, certes moins probable au sens des masses de confiance, mais offrant le meilleur consensus.

4.4 Méthode proposée

Après avoir étudié les différents systèmes de fusion vision-navigation existants dans la littérature, nous avons opté pour une modélisation proche de celle de [Lauffenburger et al., 2008]. La théorie de Dempster-Shafer semble la plus appropriée au problème, de par sa bonne gestion des conflits et de l'ignorance. Les premiers sont fréquents car la navigation et la vision sont par nature des capteurs complémentaires, gérant respectivement les limites règlementaires, ou implicites, et temporaires/locales. La seconde apparaît notamment lorsqu'une des informations est inaccessible, en l'absence de panneaux ou de mauvaise réception satellite.

De plus, il nous est facilement possible de prendre en compte la fiabilité des capteurs dans le modèle via l'affaiblissement. [Nienhüser et al., 2009] et [Daniel and Lauffenburger, 2011] ont mis en avant l'importance du contexte dans le crédit accordé aux sources. Ainsi, la confiance accordée à la vision est fortement diminuée en cas de conditions climatiques dégradées. De même, en cas de travaux ou de limites variables, la limitation fixée par la cartographie risque de ne pas être à jour. La mesure de la fiabilité $c_{cap}(\Delta t)$ des différents capteurs pondère donc la fonction de masse brute $m_{cap}(SL)$ issue du capteur pour une proposition SL. De cette façon, un capteur peu fiable,

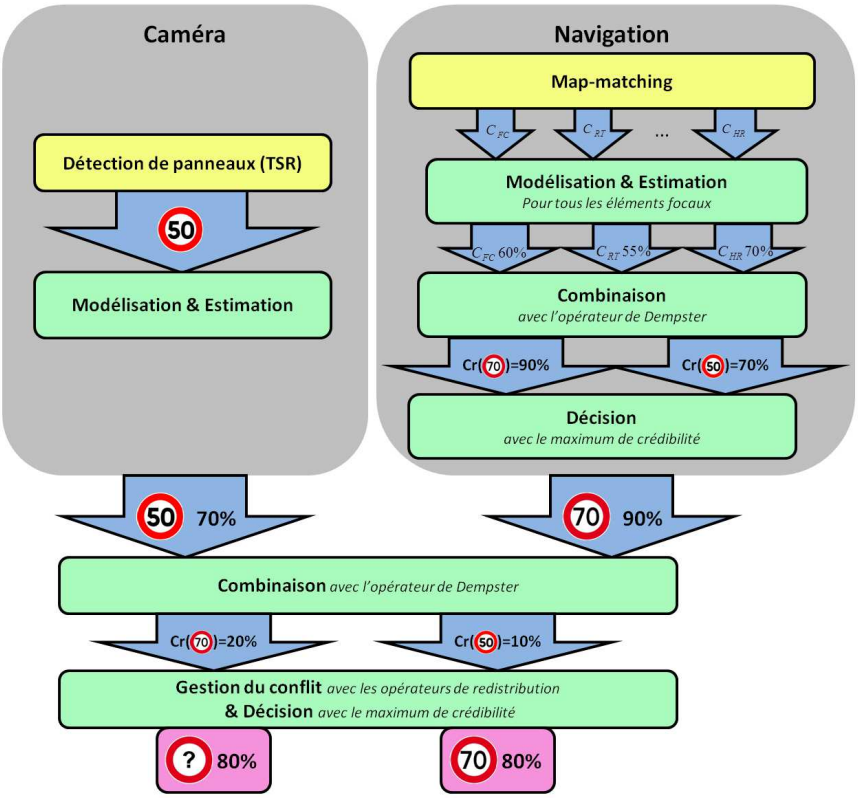


FIGURE 4.7 – Méthode de fusion à architecture décentralisée proposée par [Daniel and Lauffenburger, 2011]. La vitesse la plus probable au niveau de chaque capteur est d'abord déterminée. La fusion est ensuite réalisée entre les décisions intermédiaires.

i.e. avec une valeur c_{cap} proche de 0, ne jouera qu'un faible rôle lors de la fusion.

$$\begin{cases} \bar{m}(SL) = c_{cap}(\Delta t) m_{cap}(SL) & \text{si } SL \subset \Omega \\ \bar{m}(\Omega) = (1 - c_{cap}(\Delta t)) + c_{cap}(\Delta t) m_{cap}(\Omega) \end{cases} \tag{4.49}$$

Δt correspond au temps écoulé depuis la dernière mise à jour des données par la source. Une information qui n'a pas été renouvelée depuis longtemps aura ainsi moins d'influence qu'une autre, plus récente.

$$c_{cap}(\Delta t) = c_{cap}(0)(1 - \frac{\Delta t}{T_{max}}) \tag{4.50}$$

Au bout de $T_{max} = 60s$, la fiabilité du capteur est nulle. Cette valeur correspond à une distance parcourur $d = 833m$ en ville et $d = 2167m$ sur autoroute.

Enfin, la modélisation des données de navigation sous forme de critères nous paraît plus complète et correcte que les autres. Elle nécessite moins de règles, de connaissances d'expert pour parvenir aux masses de confiance, contrairement à [Bahlmann et al., 2008]. Notre approche consiste à diviser les critères en deux classes, la première quantifiant la fiabilité de la navigation et la seconde la confiance dans une vitesse donnée. Cela nous permet de modéliser les masses de confiance de la source indépendamment de sa fiabilité. Ces deux étapes sont détaillées dans les sections 4.4.1 et 4.4.2. Cette approche a été l'objet d'une publication [Puthon et al., 2010]. Le schéma général est donné dans la figure 4.8. Contrairement à [Daniel and Lauffenburger, 2011], nous ne prenons

de décision qu'à la fin du processus. Les deux capteurs calculent indépendamment les masses de confiance de chaque vitesse limite qui sont ensuite combinées grâce à la théorie de Dempster-Shafer.

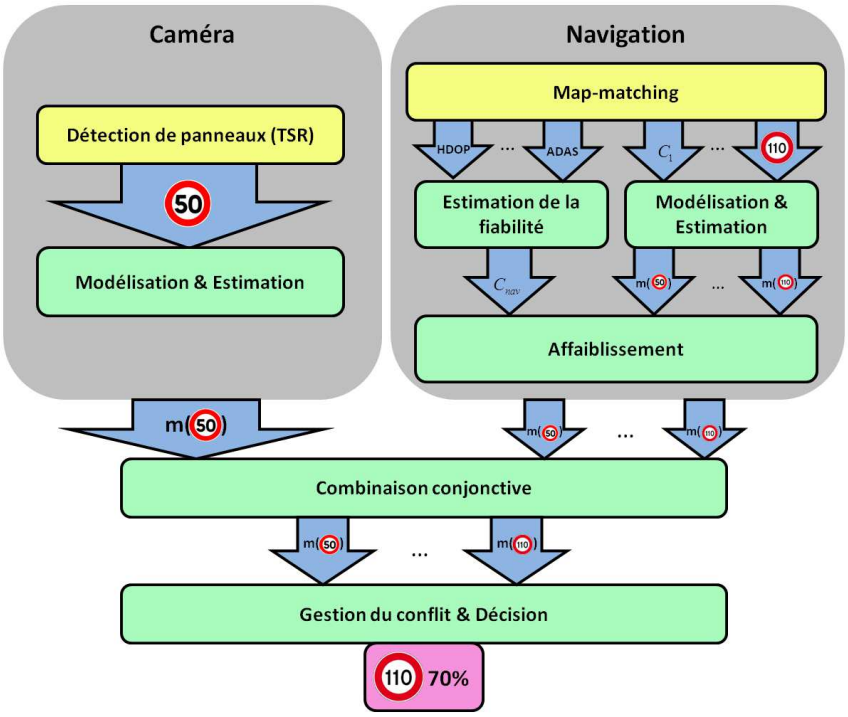


FIGURE 4.8 – Modèle de fusion de données proposé. L'architecture est de type centralisé, un seul centre de décision sélectionne la vitesse limite la plus vraisemblable à partir des masses de confiance déterminées au niveau de chaque capteur.

4.4.1 Estimation de la fiabilité de la navigation

Pour estimer la fiabilité du capteur de navigation, nous nous sommes basés sur l'approche de [Daniel and Lauffenburger, 2011]. En reprenant le processus permettant d'extraire les attributs de la base de données liés au segment de route (figure 4.9), trois étapes critiques apparaissent :

- la détermination de position du véhicule sur la Terre, qui correspond à la sortie brute du capteur GPS ;
- le *map-matching* qui affecte cette position à une route spécifique du réseau routier ;
- l'extraction des informations de la base de données.

Elles sont sujettes à différentes imprécisions que nous essayons de modéliser dans le critère de fiabilité c_{nav}.

$$c_{nav}(0) = (1 - \frac{c_{HDOP}}{c_{HDOP_{max}}}).c_{MLCP}.c_{ADAS} \tag{4.51}$$

c_{HDOP} (*Horizontal Dilution Of Precision* - Coefficient d'Affaiblissement de la Précision Horizontale) est directement lié à la qualité du positionnement par GPS, elle-même fonction de la configuration satellitaire au moment de l'acquisition. Plus cette erreur est faible, meilleure sera la précision sur la position du véhicule. La figure 4.10 compare l'évolution de l'indice HDOP et du nombre de satellites utilisés pour localiser le véhicule pour la séquence *Paris_Rouen* (tableau 4.6). Il est intéressant de noter la correspondance entre les fortes valeurs de HDOP et les zones sans couverture satellitaire.

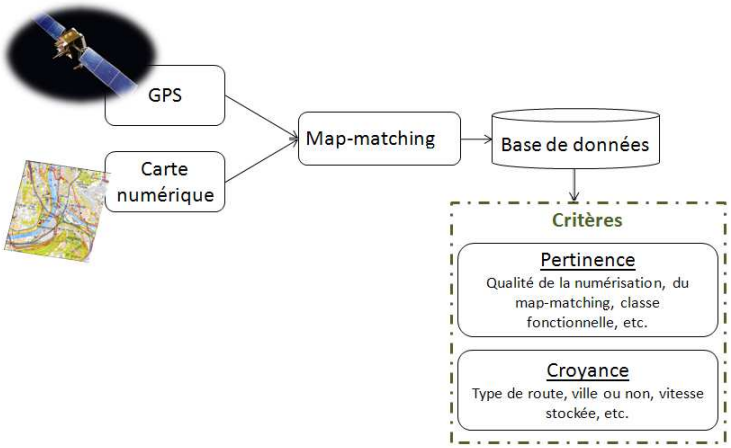


FIGURE 4.9 – Extraction d'attributs liés à la route à partir du capteur de navigation.

c_{MLCP} dépend du processus de *map-matching*, et notamment de la configuration du réseau routier autour du point. Une valeur faible de c_{MLCP} dénote une mauvaise association route-position, qui peut être dûe à un réseau dense autour du point ou une erreur de positionnement trop importante.

c_{ADAS} correspond à la précision de la carte numérique. Lorsque $c_{ADAS} = 0$, le segment de route associé à la position du véhicule n'a pas une définition suffisante pour une application ADAS.

La figure 4.11 présente l'évolution de la confiance c_{nav} au cours du temps en fonction des indices c_{HDOP}, c_{ADAS}, et c_{MLCP}. Nous pouvons noter une certaine corrélation entre ces trois valeurs, notamment à la fin de la séquence. Ainsi, les fortes valeurs de c_{HDOP} dénotent une mauvaise localisation GPS et sont associées à de faibles valeurs de c_{ADAS} (définition insuffisante des routes) et de c_{MLCP} (mauvais *map-matching*). De manière plus générale, la confiance c_{nav} oscille autour d'une valeur moyenne 0.65 (std. $\pm$ 0.25).

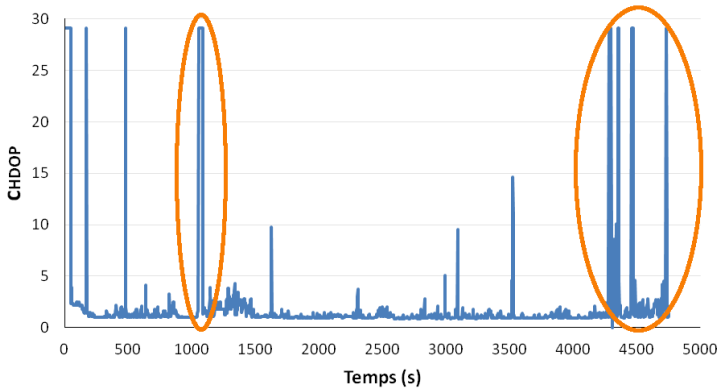
4.4.2 Estimation des fonctions de masse

La seconde étape consiste à estimer les masses de confiance pour chaque vitesse $SL \in \Omega$ en fonction des attributs issus de la base de données.

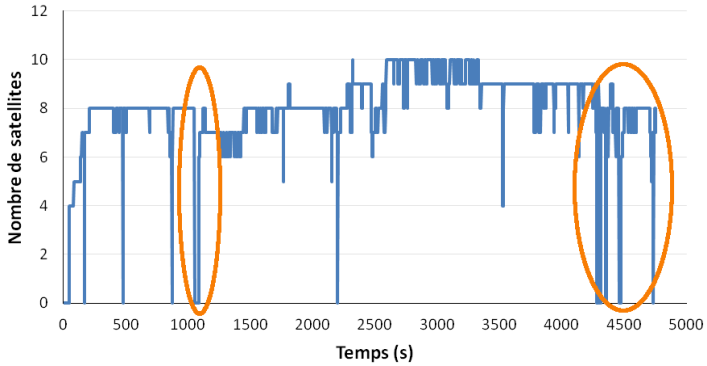
$$\Omega = \{5, 10, 20, 30, 45, 50, 60, 70, 80, 90, 100, 110, 120, 130\} \tag{4.52}$$

En l'absence de limitations locales, le Code de la Route spécifie des vitesses règlementaires en fonction du contexte. Le tableau 4.4 présente les limitations en vigueur pour un véhicule de tourisme.

Parmi les critères restants, le type de route et l'indicateur de zones urbaines semblent les plus utiles à l'estimation des masses de confiance. La figure 4.12 montre la répartition des vitesses limites en fonction de ces deux critères pour la séquence *Paris_Rouen* (tableau 4.6). Les résultats sont conformes à nos attentes. Sur autoroute, les vitesses importantes sont les plus représentées avec une nette prépondérance pour 130 km/h (70% des routes). Les routes nationales sont limitées à 110 km/h (95%) et aucune vitesse ne prédomine pour les autres catégories. Pour le second critère "Ville", la répartition est moins tranchée même si la vitesse la plus représentée en ville est 50 km/h. Une des explications à ce manque de précision est que les zones urbaines sont définies



(a) c_{HDOP}



(b) Nombre de satellites

FIGURE 4.10 – Comparaison de l'évolution de l'indice c_{HDOP} (a) au cours du temps avec le nombre des satellites (b) pour la séquence *Paris_Rouen*. Notons que les fortes valeurs de c_{HDOP} correspondent à des zones sans couverture, encerclées sur la figure.

Type de route	Ville	Vitesse limite (km/h)
Autoroute	-	130
Nationale avec séparation centrale	-	110
-	Non	90
-	Oui	50

TABLE 4.4 – Limitations de vitesse en vigueur en France pour un véhicule de tourisme d'après le Code de la Route.

comme des polygones et que tous les segments qui l'intersectent sont considérés comme urbains. Or, il arrive souvent que les villes soient entourées de voies rapides (périphériques, contournement) qui sont ainsi étiquetées comme "Ville".

Les deux critères précédents ne sont donc pas suffisants pour établir des masses de confiance correctes. Les autres attributs fournis par le capteur, classe fonctionnelle, présence d'intersections et de sorties, apportent-ils une information supplémentaire pertinente ? La figure 4.13 montre que oui, pour une partie du moins. Classe fonctionnelle et vitesse présentent une certaine corrélation. Plus la catégorie de la route est importante (FC1 ou FC2), plus la limitation y est élevée. De façon

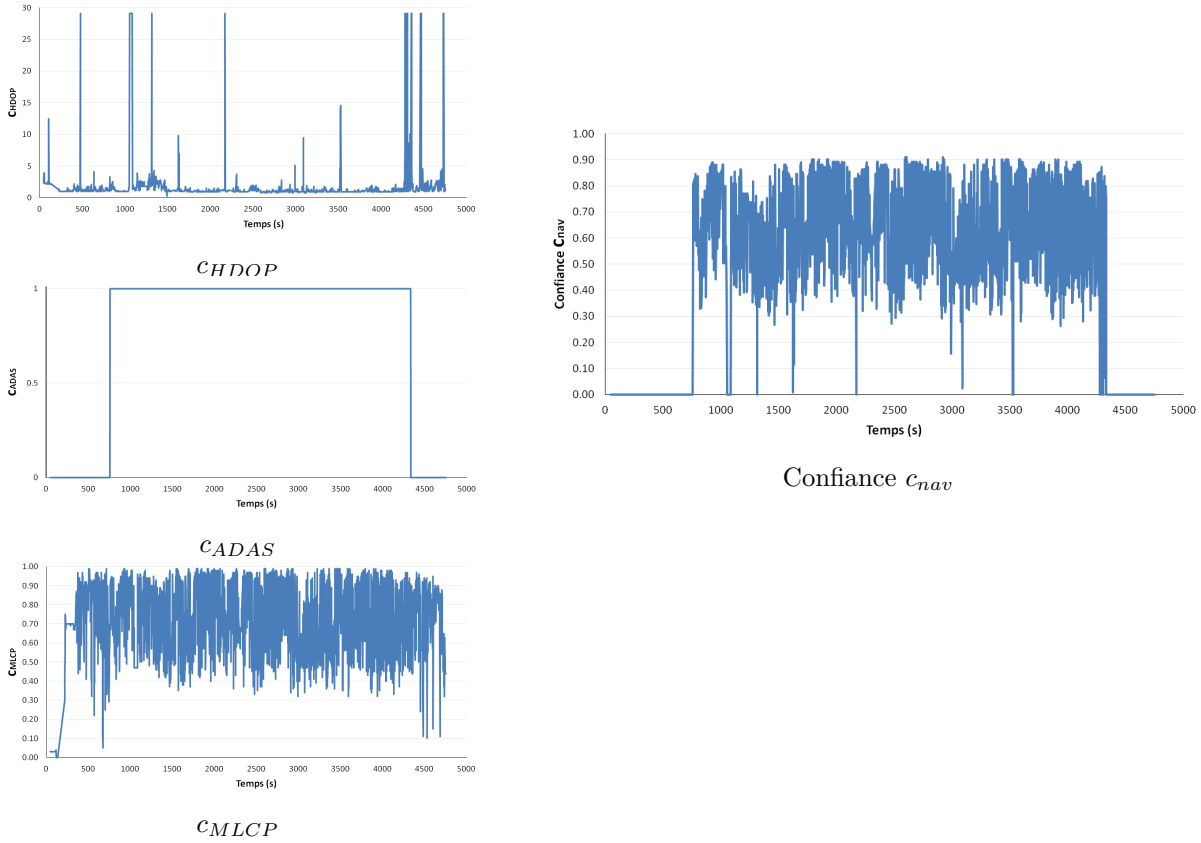
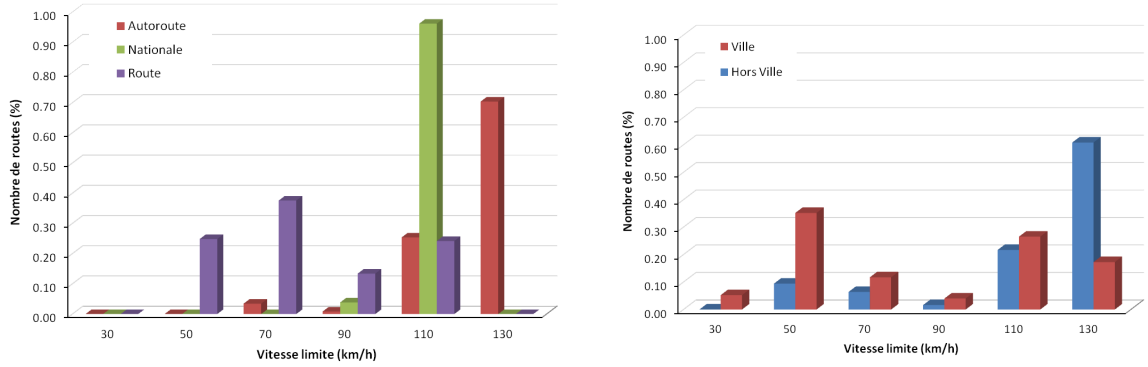


FIGURE 4.11 – Confiance c_{nav} dans le capteur de navigation à partir de c_{HDOP}, c_{ADAS} et c_{MLCP} pour la séquence *Paris_Rouen*. En début et fin de séquence, le réseau n'est pas suffisamment défini d'où de faibles valeurs de c_{ADAS} et donc de c_{nav}. De même, lorsque c_{HDOP} présente de fortes valeurs (une mauvaise réception de signal), la confiance chute.

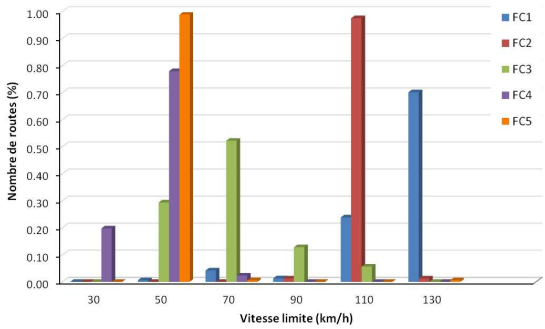


(a) En fonction du type de route

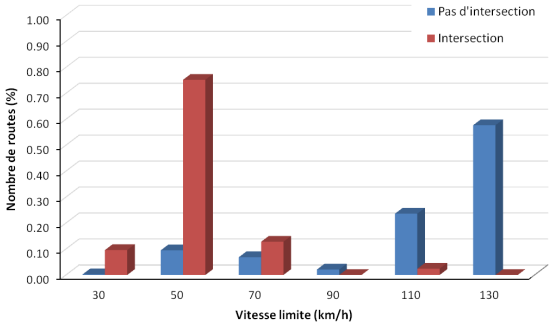
(b) En fonction du contexte urbain

FIGURE 4.12 – (a) Pourcentage de routes en fonction de la vitesse limite pour un type de route donné pour la séquence *Paris_Rouen*. (b) Pourcentage de routes en fonction de la vitesse limite et de la localisation en ville ou non. La répartition obtenue est conforme au Code de la Route. Sur autoroute, la vitesse la plus probable est 130 km/h et sur nationale 110 km/h. En ville, les faibles vitesses sont privilégiées bien que moins nettement.

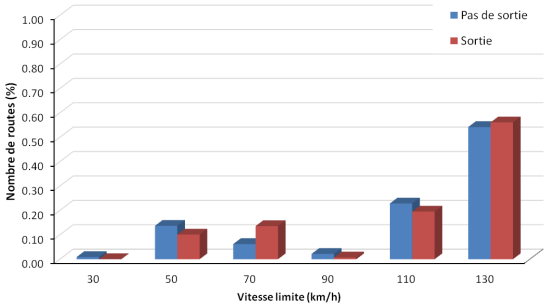
similaire, en présence d'intersections, les vitesses les plus fréquentes se situent autour de 50 km/h. Cela peut s'expliquer par l'absence de croisements sur les routes à grande vitesse contrairement aux zones urbaines où ils se rencontrent fréquemment. En revanche, l'attribut relatif à la présence de sortie n'apporte aucune information substantielle puisque chaque vitesse est également répartie entre les deux types de situations.



(a) En fonction de la classe fonctionnelle



(b) En fonction de la présence d'intersections



(c) En fonction de la présence de sorties

FIGURE 4.13 – (a) La répartition des vitesses limite (pour la séquence *Paris_Rouen*) en fonction de la classe fonctionnelle indique que les routes aux limitations les plus hautes sont également celles qui appartiennent aux premières classes (FC1 ou FC2). (b) En présence d'intersections, la vitesse est plus faible principalement parce que sur les routes à grande vitesse ce type de situation ne se rencontre que rarement. (c) L'étude de la répartition des vitesses en présence de sorties n'apporte aucune information utile. Contrairement à ce qui est attendu, aucune tendance ne s'en dégage et chaque vitesse est représentée de façon égale.

Un dernier facteur à prendre en compte est la vitesse stockée dans la base de données. La figure 4.17 représente l'évolution de cette vitesse par rapport à la vérité terrain. Sur une portion non négligeable de la séquence, la vitesse est fausse. Ces erreurs ont pour origine les défauts déjà énoncés du capteur de navigation : imprécision, fréquence de mise à jour trop faible, perte de signal, etc. Même si, de nos jours, la précision des cartes et la couverture du réseau sont de plus en plus importantes, utiliser uniquement ce capteur pour déterminer la vitesse limite ne serait donc pas suffisamment fiable. Pour parer à ses imperfections, les systèmes existants ont choisi d'utiliser cette donnée pour déterminer un ensemble de vitesses probables. Contrairement à ces approches, nous avons choisi de calculer les masses de confiance pour toutes les vitesses de l'ensemble de solutions mais de renforcer celle qui est spécifiée dans la base.

Finalement, nous avons retenu les critères suivants, similaires à ceux de [Lauffenburger et al., 2008] hormis c_{BDD} :

- c_{type}, le type de route ;
- c_{ville}, la présence en zone urbaine ou non ;
- c_{FC}, la classe fonctionnelle de la route ;
- c_{int}, la présence ou non d'une intersection ;
- c_{BDD}, la vitesse spécifiée dans la base de données.

Les valeurs choisies pour chaque vitesse limite sont déterminées empiriquement (tableau 4.5) à partir des expériences réalisées auparavant. Les valeurs du critère c_{type} sont élevées pour les vitesses cohérentes avec le type de route (autoroute, nationale ou route). Ainsi, sur nationale, les vitesses les plus probables sont 110km/h et 90km/h. c_{ville} présente de fortes valeurs pour les vitesses situées autour de 50km/h, c_{FC} pour les routes de catégories FC1 ou FC2 et donc les hautes vitesses. En ce qui concerne c_{BDD}, les vitesses stockées dans la base de données verront leur masse de confiance renforcée, reflétant la connaissance supplémentaire apportée.

Les masses de confiance sont ensuite obtenues par une somme pondérée par l'importance relative des différents critères dans le choix de la vitesse limite $SL \in \Omega$.

$$m_{nav}(SL) = \alpha_{type}c_{type}(SL) + \alpha_{ville}c_{ville}(SL) + \alpha_{FC}c_{FC}(SL) + \alpha_{int}c_{int}(SL) + \alpha_{BDD}c_{BDD}(SL) \tag{4.53}$$

$$\alpha_{type} + \alpha_{ville} + \alpha_{FC} + \alpha_{int} + \alpha_{BDD} = 1 \tag{4.54}$$

Le classement des coefficients dépend fortement de l'appréciation de l'expert et des observations réalisées. À partir des études sur les répartitions des vitesses en fonction du contexte, nous avons privilégié la hiérarchie suivante :

$$\begin{matrix} \alpha_{type} \\ \alpha_{ville} \\ \alpha_{BDD} \end{matrix} > \alpha_{FC} > \alpha_{int} \tag{4.55}$$

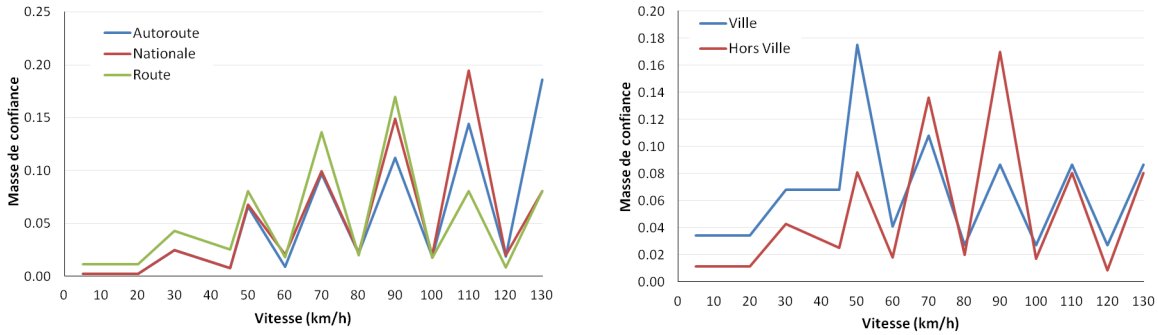
Nous avons finalement choisi les valeurs $\alpha_{type} = \alpha_{ville} = \alpha_{BDD} = \frac{6}{21}$, $\alpha_{FC} = \frac{2}{21}$ et $\alpha_{int} = \frac{1}{21}$.

Critères	Limitations de vitesse													
	5	10	20	30	45	50	60	70	80	90	100	110	120	130
C_{type} : Autoroute	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.3	0.1	0.3	0.1	0.6	0.1	1.0
C_{type} : Nationale	0.0	0.0	0.0	0.0	0.0	0.2	0.1	0.3	0.1	0.6	0.1	1.0	0.1	0.0
C_{type} : Route	0.1	0.1	0.1	0.2	0.1	0.4	0.1	0.8	0.1	1.0	0.1	0.0	0.0	0.0
C_{ville} : Oui	0.1	0.1	0.1	0.3	0.3	1.0	0.1	0.3	0.0	0.0	0.0	0.0	0.0	0.0
C_{ville} : Non	0.0	0.0	0.0	0.0	0.0	0.3	0.1	0.6	0.1	1.0	0.1	1.0	0.1	1.0
C_{FC} : Catégorie haute	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.1	1.0	0.1	1.0
C_{FC} : Catégorie basse	0.1	0.1	0.1	1.0	0.3	1.0	0.1	1.0	0.1	0.0	0.0	0.0	0.0	0.0
$C_{intersection}$: Oui	0.1	0.1	0.1	0.8	0.8	0.8	0.1	0.8	0.0	0.0	0.0	0.0	0.0	0.0
$C_{intersection}$: Non	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.8	0.1	0.8	0.1	0.8
C_{BDD} : Oui	1.0													
C_{BDD} : Non	0.0													

TABLE 4.5 – Critères utilisés pour l'évaluation des masses de croyance pour les différentes vitesses limites.

La figure 4.14 présente les masses de confiance obtenues pour les différentes vitesses du cadre de discernement Ω en fonction du contexte de conduite. Plus la valeur des masses est grande, plus la vitesse associée est considérée comme fréquente dans la situation donnée. Sur autoroute, les fortes limitations sont ainsi privilégiées, notamment la vitesse règlementaire de 130 km/h. Toutefois,

d'autres vitesses plus faibles apparaissent également, dans des zones dangereuses ou aux abords des péages et sorties. Sur route nationale et ordinaire pour lesquelles les vitesses légales sont de 110 km/h et 90 km/h, même constat. En zone urbaine enfin, les faibles vitesses sont les plus favorisées et principalement 50 km/h. Lorsqu'une vitesse SL est spécifiée dans la base de données, sa masse de confiance est augmentée de $\alpha_{BDD}C_{BDD}(SL)$.



(a) Influence du type de route

(b) Influence du contexte urbain

FIGURE 4.14 – (a) Masses de confiance obtenues pour chaque vitesse en fonction du type de route. Les vitesses définies par le Code de la Route bénéficient dans chaque situation de la masse la plus forte. Ainsi, les types Autoroute, Nationale et Route correspondent respectivement aux limitations 130 km/h, 110 km/h et 90 km/h. Les dents de scie viennent du fait que certaines vitesses sont beaucoup moins fréquentes que d'autres. (b) Masses de confiance obtenues pour chaque vitesse en ville et hors-ville. Les faibles vitesses sont privilégiées dans le premier cas, principalement 50 km/h, alors que dans le second cas, les limitations se situent autour de 90 km/h. Nous considérons dans les deux cas qu'aucune vitesse n'est stockée dans la base de données.

4.4.3 Détermination de la vitesse limite

Comme détaillé dans la figure 4.8, nous réalisons une fusion globale, la décision n'étant prise qu'à la fin à l'aide de toutes les fonctions de masse estimées par les différentes sources.

Concernant la vision, m_{cam} correspond à la confiance qu'a le système dans son résultat de classification. La fiabilité c_{cam} est telle que $c_{cam}(0) = 0.9$ et diminue linéairement au cours du temps selon l'équation 4.50. L'information ne sera prise en compte qu'une fois le panneau dépassé dans un souci de synchronisation des données, puisque les panneaux ne sont applicables qu'à partir de leur position. Les masses de confiance de la navigation sont calculées à l'aide des attributs précédemment décrits de la même façon que sa fiabilité. Notons que seuls les singletons sont définis dans notre méthode, de la même façon que [Lauffenburger et al., 2008] ou [Daniel and Lauffenburger, 2011].

Une combinaison conjonctive normalisée est ensuite réalisée.

$$m_{cam} \oplus m_{nav}(SL) = \frac{1}{1-K} \sum_{SL_{cam} \cap SL_{nav} = SL} m_{cam}(SL_{cam}) m_{nav}(SL_{nav}) \tag{4.56}$$

$$K = \sum_{SL_{cam} \cap SL_{nav} = \emptyset} m_{cam}(SL_{cam}) m_{nav}(SL_{nav}) \tag{4.57}$$

La décision finale, au sens du maximum de crédibilité, correspondra à la vitesse limite en vigueur

SL^*. Nous intégrons toutefois une condition supplémentaire liée à la valeur du conflit.

$$Cr(SL^*) = \max_{SL} Cr(SL) \tag{4.58}$$

$$K < K_{max} \tag{4.59}$$

Nous choisissons $\mathbf{K_{max} = 0.2}$.

4.5 Évaluation

4.5.1 Bases de données

L'absence de bases de données publiques et de résultats détaillés des techniques existantes rend difficile la comparaison. Dans l'état de l'art, aucune évaluation complète n'a été réalisée. Les seuls résultats correspondent à des situations ponctuelles et il est difficile de se faire une idée précise des performances d'un système complet opérationnel. Nous avons donc décidé d'évaluer notre approche sur des bases de données acquises par un partenaire du projet Speedcam, Valeo. Leurs caractéristiques sont détaillées dans le tableau 4.6. Les séquences ont été annotées manuellement. La vérité terrain correspond à la vitesse limite effective du véhicule, estimée à partir des panneaux de limitations ainsi que du contexte de conduite que l'opérateur peut conjecturer des images.

Les capteurs utilisés pour réaliser les séquences sont :

- une caméra monochrome 10 bits avec une résolution de 752 x 480 pixels ;
- une *SensorBox* de NavTeq contenant un GPS Trimble Lassen SK-II couplé à des gyroscopes et au bus CAN du véhicule.

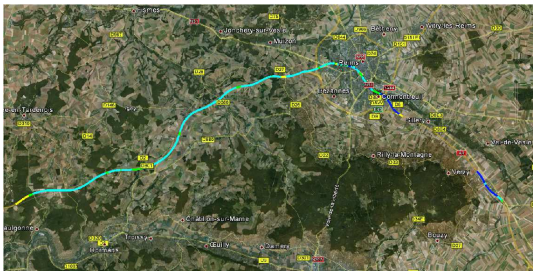
La figure 4.15 représente les tracés des différentes séquences obtenus à partir des trames GPS. Nous constatons avec ces images satellitaires plusieurs pertes de signal, notamment pour les séquences *A4_Reims* et *Cherisy*, afin de simuler des défaillances du système de navigation. La fusion de données devra être robuste face à ces problèmes pendant lesquels la caméra sera le seul capteur fiable.

Séquence	Durée (h :mm :ss)	Longueur (km)	Ville (%)	Autoroute (%)	Sorties	Panneaux
A4_Reims	0:36:13	64.8	0	99	11	57
A4_A86	0:35:53	20.6	0	99	11	57
Paris_Rouen	1:19:03	124.2	11	88	27	113
Cherisy	0:51:42	55.2	20	0	20	67

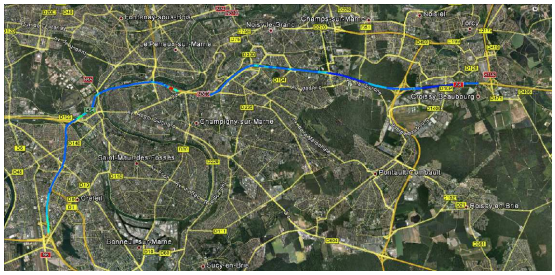
TABLE 4.6 – Séquences utilisées pour nos évaluations.

4.5.2 Comparaison avec la méthode de [Lauffenburger et al., 2008]

Nous avons comparé notre technique à celle de [Lauffenburger et al., 2008] à laquelle nous avons apporté une amélioration notable en séparant les attributs en deux classes. Dans le tableau 4.7, nous présentons les performances des systèmes de navigation, vision, fusion de données selon [Lauffenburger et al., 2008] et notre méthode. Les résultats correspondent au pourcentage de temps pendant lequel le véhicule a roulé avec la bonne vitesse. De manière générale, notre approche s'avère meilleure que celle de [Lauffenburger et al., 2008] pour toutes nos séquences. Cela provient essentiellement des cas de mauvaise localisation ou de données obsolètes de la base cartographique. Nous pouvons néanmoins observer que la fusion directe entre la navigation et le détecteur de panneaux n'est pas une solution miracle. Il arrive que des limitations détectées par la vision avec une

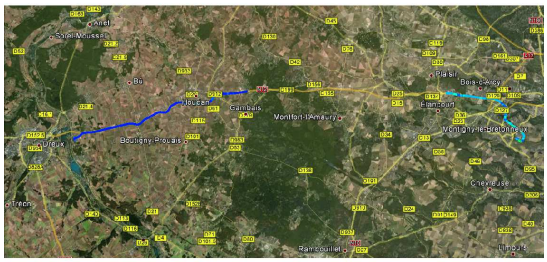


(a) *A4_Reims*



(b) *A4_A86*

(c) *Paris_Rouen*



(d) *Cherisy*

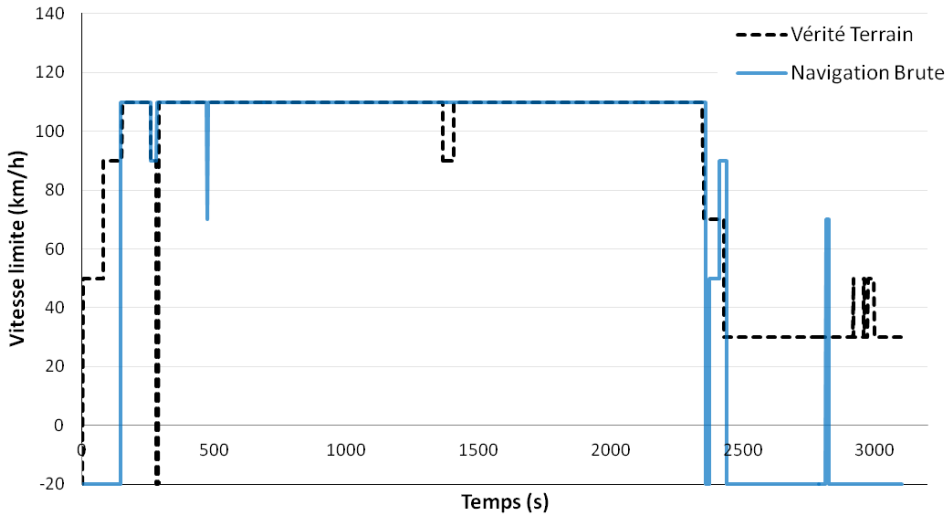
FIGURE 4.15 – Images satellitaires, réalisées à partir des trames GPS, des séquences employées pour l'évaluation de nos algorithmes de fusion. Les couleurs correspondent au nombre de satellites ayant participé à la localisation, les plus claires pour les faibles réceptions. Nous notons une perte de signal sur une partie du trajet des séquences (a) *A4_Reims* et (d) *Cherisy* qui simule une défaillance du système de cartographie. De plus, les tracés de (c) *Paris_Rouen* et (d) *Cherisy* traversent des zones urbaines contrairement aux deux autres, essentiellement sur autoroute.

bonne confiance induisent le système en erreur car ces dernières ne s'appliquent pas au véhicule. Ces situations ne peuvent pas être résolues en l'état actuel mais nous proposons des solutions dans le chapitre 5.

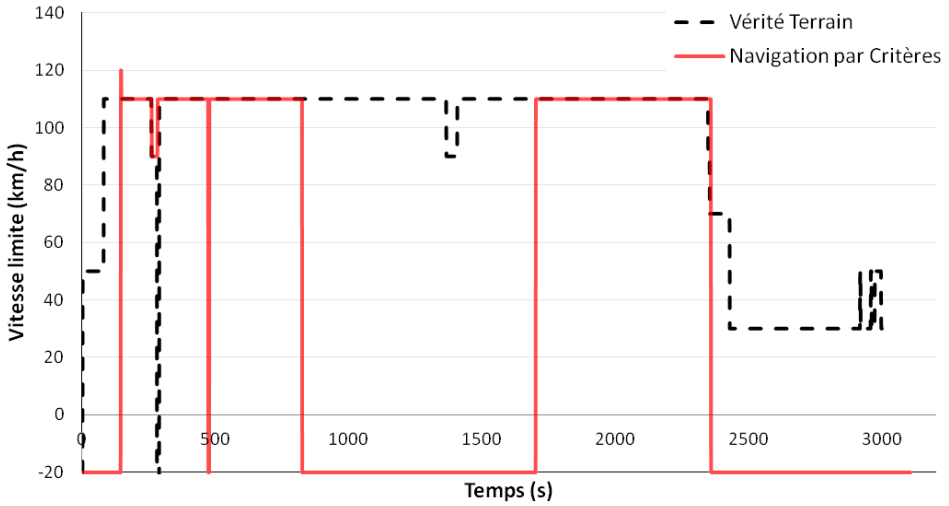
Séquence	Navigation	Vision	Fusion Lauffenburger	Fusion Puthon
A4_Reims	86%	42%	58%	80% (*+22%*)
A4_A86	98%	79%	92%	97% (*+5%*)
Paris_Rouen	74%	46%	69%	72% (*+3%*)
Cherisy	71%	59%	56%	69% (*+13%*)

TABLE 4.7 – Comparaison des sorties des systèmes navigation, vision, fusion de données selon [Lauffenburger et al., 2008] et fusion de données selon notre méthode (Fusion Puthon). Sont indiqués les pourcentages de temps pendant lequel le système donne la bonne limite de vitesse.

La figure 4.16 illustre le traitement opéré sur les données de navigation. Nous ne prenons pas en compte directement la vitesse stockée dans la base de données, ce qui pourrait générer des erreurs, en cas de travaux ou de perte de signal. À la fin de la séquence *Cherisy*, la base de données est erronée et la prise en compte des critères donne une sortie à -20, ce qui équivaut à une ignorance du système de navigation quant à la vitesse limite.



(a) Sortie brute de la Navigation



(b) Sortie "traitée" de la Navigation

FIGURE 4.16 – (a) Sortie brute du capteur de navigation pour la séquence *Cherisy*. La vitesse prise en compte est celle associée à la portion de route sur laquelle circule le véhicule et est stockée dans la base de données. Si aucune information n'est disponible, la sortie vaut -20. (b) Sortie du capteur de navigation après gestion des attributs (critères). Nous sélectionnons les vitesses avec la masse de confiance la plus élevée. L'ignorance dûe à la perte de signal en milieu de parcours est ici représentée par une sortie à -20. La fin de parcours est mal définie dans la carte, d'où $c_{nav} = 0$, et une sortie à -20.

4.5.3 Importance de la fusion de données

D'après l'observation des sorties brutes des capteurs en terme de vitesse limite, il apparaît évident qu'un système de type TSR ou basé sur la navigation seule ne sera pas suffisamment performant. La figure 4.17 montre les vitesses limites fournies par les deux sources d'information pour la séquence *Paris_Rouen*. Hormis les début et fin de séquence, la navigation est généralement proche de la vérité terrain, la base de données étant à jour et le réseau routier peu complexe

évitant les erreurs de *map-matching*. La caméra, elle, permet de pallier efficacement aux défaillances précédentes de la navigation. Tout l'intérêt de la fusion de données apparaît. Les situations pour lesquelles la navigation ne dispose que de peu de données ou d'une faible fiabilité sont résolues par la combinaison des deux capteurs, comme le montre la figure 4.17. Les résultats obtenus pour les séquences *A4_Reims* et *Cherisy* sont donnés dans la figure 4.18.

Contrairement à la navigation, les erreurs entre le système de vision et la vérité terrain sont nettement plus nombreuses. En effet, chaque panneau de limitation de vitesse rencontré est pris en compte sans évaluer le contexte environnant (sortie, panonceau, etc.). Les panonceaux de type "Flèche" ou les marquages pourraient alors apporter une information très utile dans ces circonstances. Il en est de même des limitations spécifiques à certaines catégories de véhicules ou conditions climatiques. Quelques situations problématiques sont illustrées dans la figure 4.19. Nous retrouvons la situation de passage au niveau d'une sortie non empruntée, signalée par un panneau avec flèche et d'une limitation spécifique aux transports de matières dangereuses. Les panonceaux seront gérés par le système défini dans les chapitres 2 et 3 tandis que les marquages seront évoqués dans le chapitre 5.

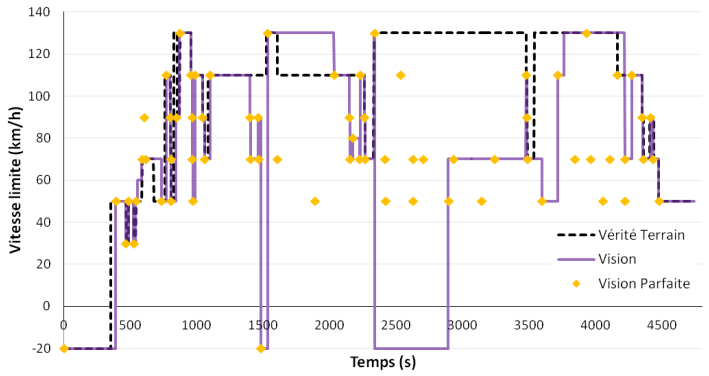
4.6 Conclusion

Le système de détermination de vitesse limite proposé est prometteur. La fusion de données vision et cartographiques semble en effet nécessaire pour obtenir de bonnes performances au vu des résultats de chaque capteur pris indépendamment. Le capteur de navigation est utilisé pour connaître le contexte de conduite (type de route, environnement urbain ou non, etc.). Pour cela, nous nous sommes inspirés de la méthode de [Lauffenburger et al., 2008] et avons extrait de la base de données un ensemble d'attributs qui donne ces indications sur l'environnement. L'amélioration principale que nous avons proposée concerne l'utilisation de la technique d'affaiblissement de Dempster-Shafer afin de prendre en compte la fiabilité du capteur de navigation.

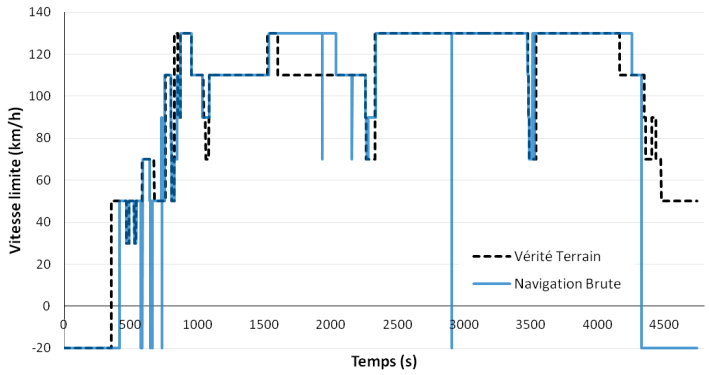
Les attributs ont été séparés en deux classes, l'une relative à la qualité des informations provenant de la source et l'autre aux masses de confiance des différentes vitesses limites. Cette distinction nous permet de gérer efficacement les situations de perte de signal ou de mauvaise définition de la carte sans altérer les confiances associées aux vitesses en fonction du contexte. En comparant notre approche à celle de Lauffenburger *et al.*, nous avons pu observer une progression de presque 20% pour certaines séquences de notre base de données.

Toutefois, les résultats sont entachés par le module de vision qui ne prend en compte que les panneaux de limite de vitesse. Aucune gestion des panonceaux ou des marquages n'est réalisée. De nombreuses fausses alarmes sont donc déclenchées et génèrent des erreurs, aux abords des sorties notamment. C'est pourquoi nous proposons dans le chapitre 5 d'ajouter au système ces deux fonctionnalités. La première a fait l'objet des deux chapitres précédents tandis que la seconde sera développée dans le chapitre suivant.

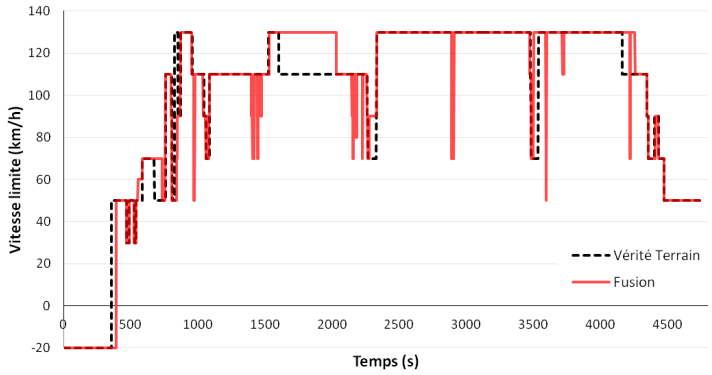
Une autre piste à exploiter pour perfectionner notre système concerne le choix des valeurs des critères. Il a été fait empiriquement par faute de bases de données suffisamment complètes. Il serait plus correct d'étudier les limites de vitesse pour un grand nombre de séquences et la valeur des critères correspondante.



(a) Vision

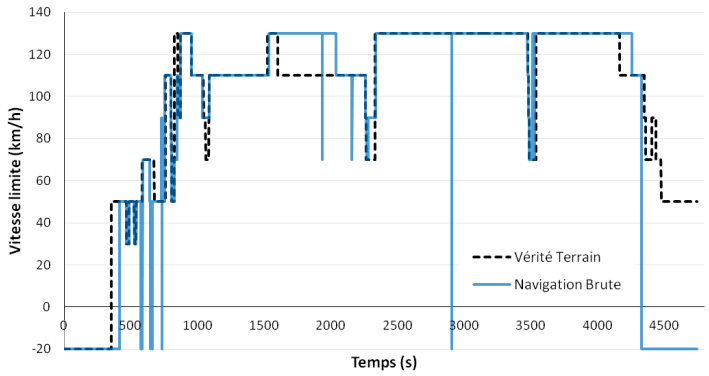


(b) Navigation

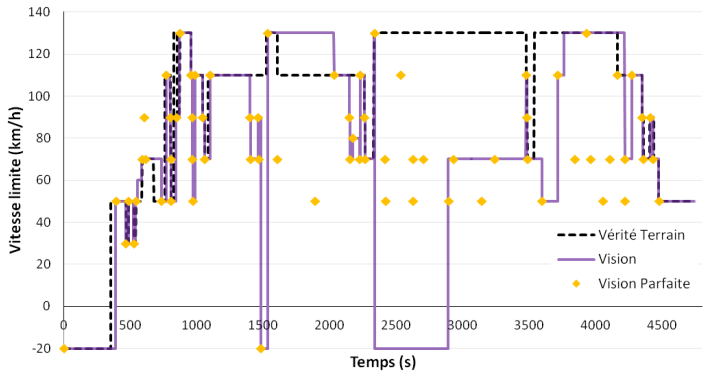


(c) Fusion

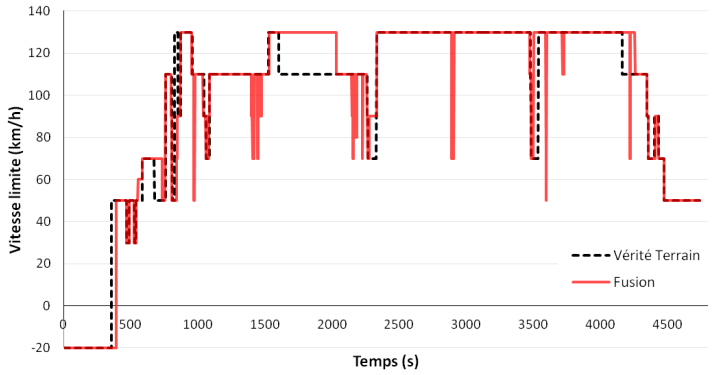
FIGURE 4.17 – Sorties brutes des capteurs vision (a) et navigation (b) et fusion de données (c) comparées à la vérité terrain pour la séquence *Paris_Rouen*. (a) De nombreux panneaux (illustrés par des losanges jaunes) sont présents au bord de la route et un système de type TSR ne serait correct que 42% du temps. Toutefois, des fausses détections surviennent fréquemment, la plupart du temps lorsque le panneau est trop éloigné du véhicule ou visible trop peu de temps pour être validé. Ces erreurs servent parfois le système, comme à l'instant $t = 4000$ où les panneaux 70 et 50 sont heureusement ignorés. (b) Au début et à la fin de la séquence, les informations fournies par la base de données sont erronées. Cela correspond aux périodes où la fiabilité du capteur est la plus faible. De plus, au cours de la séquence, quelques pics apparaissent à 70 km/h pour la navigation alors que la vérité terrain vaut respectivement 130 et 110 km/h. (c) Autour de $t = 1500$, le véhicule passe à proximité de plusieurs sorties qui sont prises en compte par le système. La confiance de la vision dans ces panneaux est en effet très élevée et la navigation ne les rejette pas non plus puisque ces vitesses peuvent survenir sur autoroute.



(a) Navigation



(b) Vision



(c) Fusion

FIGURE 4.18 – Sorties des différents modules pour la séquence *Paris_Rouen*. (a) La navigation donne de bons résultats la plupart du temps, la base de données est donc correctement à jour. En fin de séquence cependant, les limitations temporaires sont données par les panneaux. (b) Le module de vision prend en compte tous les panneaux, sa précision n'atteint que 42%, mais permet de bien gérer la fin de séquence. (c) Résultat de la fusion de données. Certaines situations, ,otamment au début et à la fin de la séquence, ont bénéficié de la combinaison des deux capteurs, alors complémentaires. Quelques erreurs doivent encore être corrigées, notamment par la prise en compte des panonceaux et marquages.

(a) Situation correcte.

(b) Situations incorrectes.

FIGURE 4.19 – Illustrations de situations correctement et faussement gérées par le système de vision. (a) Panneau correctement interprété par le système de vision. (b) Situations pour lesquelles le panneau est mal interprété. La sortie située à droite n'est pas empruntée par le véhicule, mais en l'absence de détection de panonceaux et de marquages, le système de vision valide le panneau. De même pour les limitations spécifiques à certains véhicules.

Chapitre 5

Système complet

5.1 Introduction

Après avoir amélioré de manière indépendante les deux sources utilisées pour la détermination de la vitesse limite, nous allons maintenant mettre en place le système complet. Tout d'abord, un diagramme illustrant les différentes interactions entre les modules de vision et de cartographie est présenté. Nous détaillons ensuite l'implémentation du système de reconnaissance des panonceaux. Dans un troisième temps, nous abordons un aspect nouveau, la gestion des marquages. Cette partie est essentielle pour s'assurer que les limites de vitesses temporaires soient associées à la voie à laquelle elles s'appliquent. Enfin, nous discutons des résultats obtenus pour le système complet de fusion de données.

5.2 Présentation du système complet

L'importance de la fusion des capteurs de vision et de navigation est indéniable pour garantir la robustesse et la fiabilité du système. Cependant, nous avons constaté dans le chapitre 4 que la connaissance seule des panneaux ne permettait pas de gérer des situations telles les sorties d'autoroute de manière correcte. En effet, toutes les limites variables seront prises en compte qu'elles s'appliquent ou non à la voie sur laquelle circule le véhicule. D'autres informations, comme les panonceaux et les marquages, sont nécessaires pour attribuer les panneaux aux bonnes routes. Pour cela, nous avons implémenté un système complet incorporant tous ces éléments en nous appuyant sur le raisonnement humain (figure 5.1).

Trois types d'évènements sont susceptibles de modifier la vitesse limite : un changement de contexte de conduite, la détection d'un nouveau panneau et une traversée de marquage. Le premier est fourni par la navigation et concerne l'arrivée/la sortie de ville, le changement de route ou la modification de la vitesse limite stockée dans la base. Les deux autres évènements sont générés par le système de vision. Les panneaux et panonceaux sont détectés, reconnus puis associés à la bonne voie, grâce à la connaissance des marquages. Lorsque le véhicule arrive au niveau d'un panneau qui le concerne, le système met à jour la vitesse limite. La traversée d'un marquage témoigne, elle, d'un changement de voie (une sortie par exemple) et donc des limitations associées.

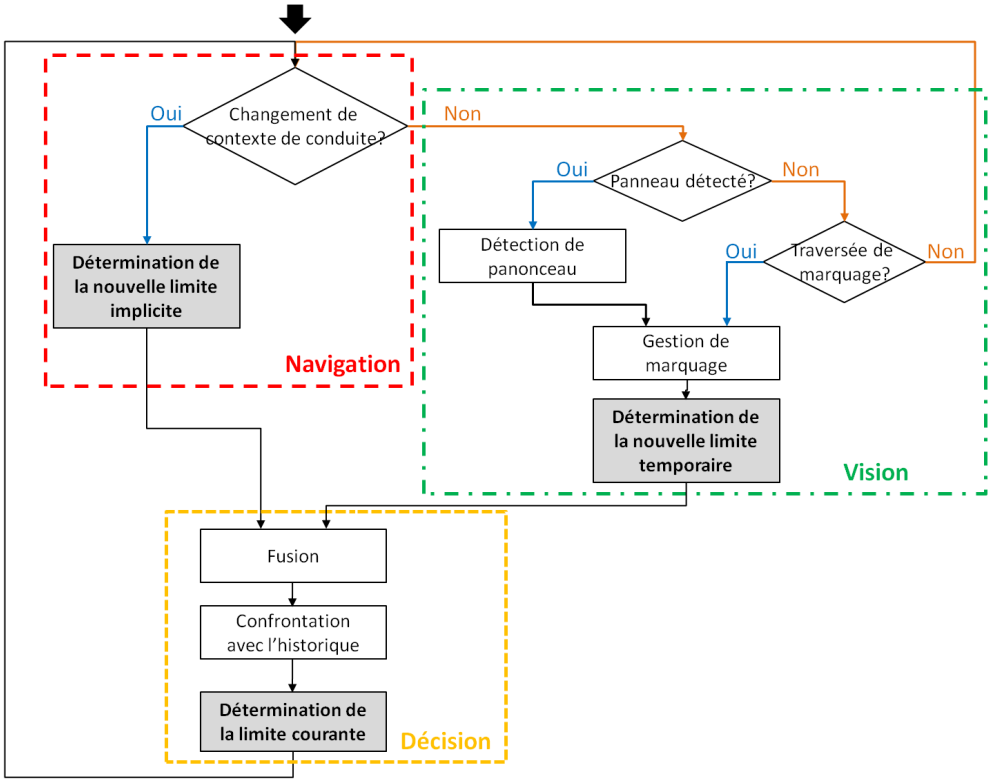


FIGURE 5.1 – Diagramme du système complet implémenté composé de trois parties principales : les systèmes de navigation et de vision et le centre décisionnel. Le processus de mise à jour de la vitesse limite est déclenché par trois types d'évènements (modélisés par des losanges). Les changements dans le contexte de conduite sont gérés par la navigation. Les panneaux nouvellement détectés ou les traversées de marquages concernent tous deux la vision. Les informations mises à jour par ces deux systèmes sont ensuite fusionnées pour déterminer la vitesse limite réelle. Le résultat est finalement comparé à l'historique pour détecter les sauts dans les vitesses qui révèleraient des erreurs.

5.3 Intégration des panonceaux

Après avoir évalué indépendamment les algorithmes de détection et de classification de panonceaux, nous allons les combiner pour estimer les performances du système complet. Le tableau 5.1 résume les proportions des quatre macro-catégories précédemment définies dans la base de données. Les panonceaux sont dénombrés à la fois image par image et par occurrence, un objet devant être visible au moins $N_D = 3$ images pour être valide.

5.3.1 Comparaison de détecteurs

Dans un premier temps, nous allons comparer les résultats obtenus pour deux configurations de détecteurs, "Graphe" et "Région", du chapitre 2 et le classifieur développé dans le chapitre 3 dont nous rappelons les paramètres sélectionnés (tableau 5.2).

La figure 5.2 présente le rappel et le nombre de faux positifs pour les deux détecteurs en fonction de la probabilité minimum τ_{pos} d'appartenir à la classe des positifs et de l'écart Δ_{1-2} entre les deux meilleures probabilités des macro-catégories p_{best} et p_{second}. Comme attendu, le nombre de faux positifs est plus faible pour la méthode "Région" que pour "Graphe". Il oscille entre 0 et 1.5 pour la première et atteint presque 3 pour la seconde. En revanche, le rappel pour "Graphe" conserve une

Catégorie	Nombre (image)	Occurrence physique
"Texte"	1899	102
"Flèche"	329	23
"Pictogramme"	1096	55
"Mixte"	620	49

TABLE 5.1 – Bases de données utilisées pour le système complet de reconnaissance de panonceaux.

Détecteur	
Région Graphe	$\kappa_1 = 2.5 \quad \kappa_2 = 0.8 \quad \kappa_L = 0.22 \quad \kappa_G = 0.12$ $\kappa_A = 0.03$
Classifieur	
PHOG Proportion	$K = 8 \quad L = 2$ $\kappa'_1 = 0.5 \quad \kappa'_2 = 1.5$

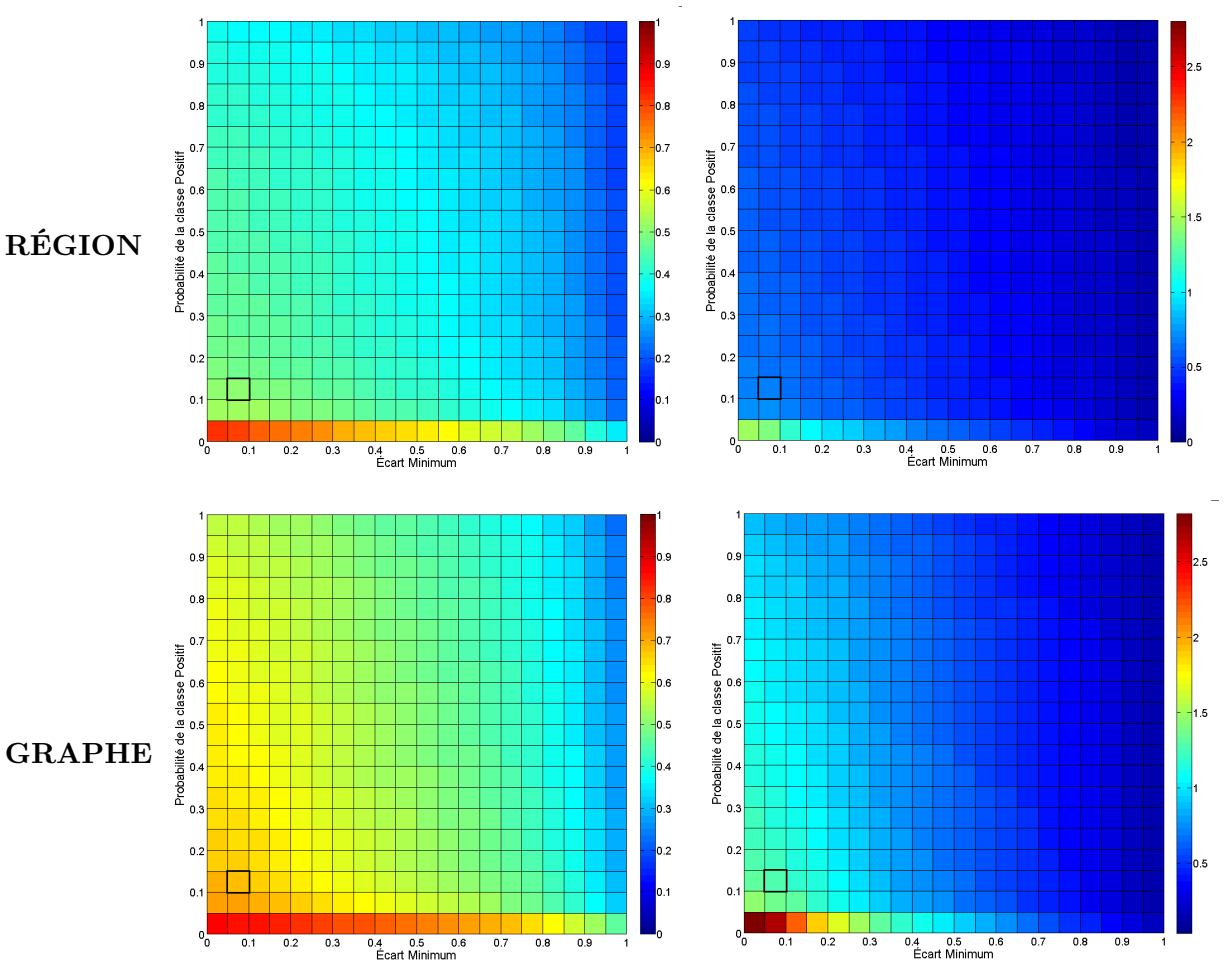
TABLE 5.2 – Paramètres utilisés pour la détection et la reconnaissance des panonceaux. Nous comparons les deux meilleurs détecteurs du chapitre 2 et les descripteurs du chapitre 3.

valeur supérieure à 0.6 pour une plus grande plage de valeurs. Lorsque τ_{pos} et Δ_{1-2} augmentent, le nombre de faux positifs diminue au détriment du rappel. Toutefois, l'influence de τ_{pos} sur les résultats nous fait penser que le classifieur binaire n'est pas approprié. Le nombre de faux positifs éliminés grâce à cette étape est trop faible par rapport aux pertes de bonnes détections. Les raisons pour ces mauvais résultats peuvent être un sous-apprentissage du classifieur binaire ou un mauvais choix de la base de données d'apprentissage. Au vu de ces résultats, nous choisissons $\tau\mathbf{pos} = \mathbf{0.01}$ et $\mathbf{\Delta_{1-2}} = \mathbf{0.05}$.

Le détail par macro-catégorie est donné dans la figure 5.3 en terme de rappel et de FPPP pour $\tau_{pos} = 0.1$ et en fonction de Δ_{1-2}. Les résultats obtenus par la méthode "Graphe" sont satisfaisants pour cette base de données. Pour $\Delta_{1-2} = 0.05$, le rappel atteint 86% pour la classe "Texte", 45% pour "Pictogramme", 48% pour Mixte mais 12% seulement pour "Flèche" ! Pour l'approche "Région", le taux de classification est nettement plus faible avec seulement 66% pour "Texte", 32% pour Pictogramme, 30% pour "Mixte" et 11% pour "Flèche". Du point de vue des FPPP, les performances des deux méthodes semblent similaires sauf pour "Texte" qui présente une valeur proche de 1.2 pour "Graphe" et de seulement 0.6 pour "Région". La première phase de classification visant à séparer positifs et négatifs n'est pas suffisante dans cette situation et de nombreux négatifs sont classifiés comme du "Texte".

Les matrices de confusion des deux détecteurs sont données dans les tableaux 5.3 et 5.4 pour les paramètres $\tau_{pos} = 0.1$ et $\Delta_{1-2} = 0.05$. Les faux positifs sont comptabilisés par rapport aux panonceaux de la vérité terrain présents au même moment. Imaginons que le système reconnaît dans l'image un objet de type "Flèche" alors que la vérité terrain donne un "Texte" et un "Pictogramme". Nous aurons alors un demi-faux positif "Flèche" pour "Texte" et un demi-faux positif pour "Pictogramme".

Les performances sont très variables en fonction des classes. La classe "Flèche" bénéficie d'un rappel de seulement de 12% pour "Graphe" et est principalement confondue avec la classe "Texte". Sa précision est de 94%, la confiance que nous pouvons avoir dans le classifieur est donc forte pour cette macro-catégorie. Un panonceau classifié comme "Flèche" aura de fortes chances d'être correct. Pour les classes "Mixte" et "Pictogramme", la tendance est la même avec un rappel de 52% (resp. 48%) et une précision de 57% (resp. 58%). Étant donné que ces deux classes correspondent directement à un type donné de panonceaux, ces résultats ne sont pas gênants. Inversement, le

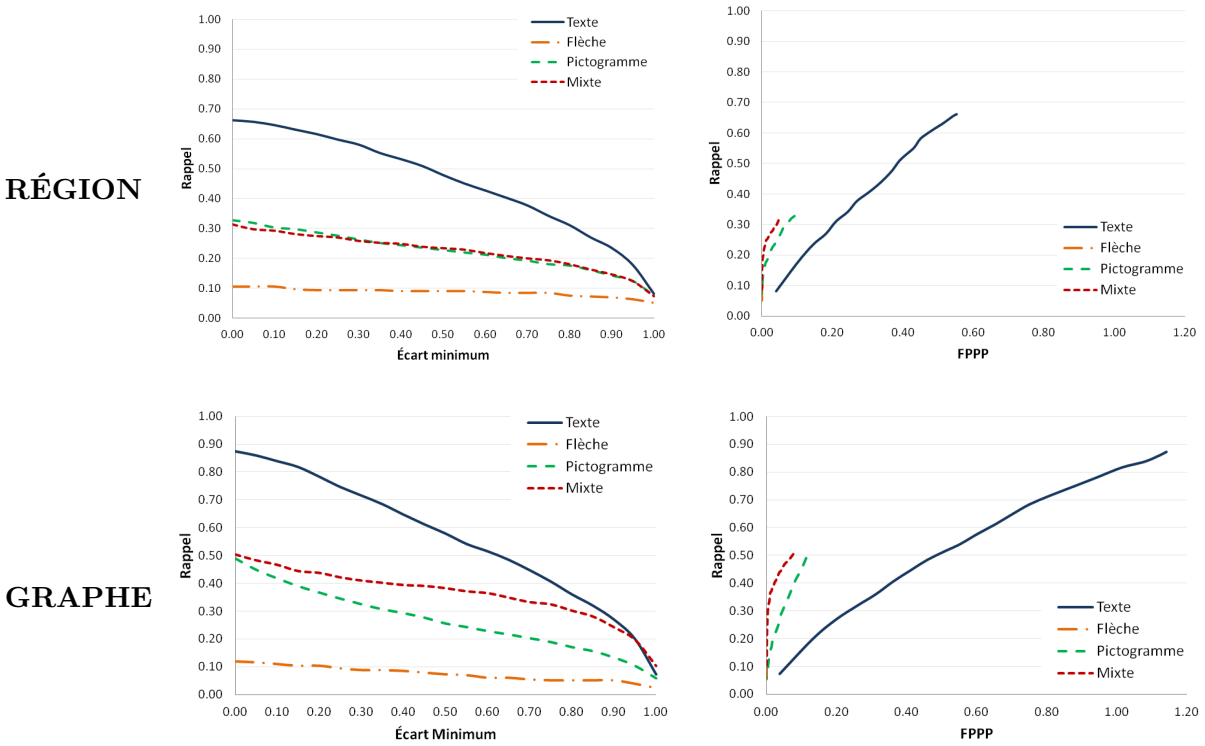


(a) Rappel (b) FPPP

FIGURE 5.2 – Évolution (a) du rappel et (b) du nombre de Faux Positifs Par Positif (FPPP) en fonction de la probabilité minimum τ_{pos} d'appartenir à la classe Positif et Δ_{1-2} l'écart entre les deux meilleures probabilités de macro-catégories. Les performances chutent rapidement à mesure que les deux paramètres augmentent et que les faux positifs sont éliminés. La méthode "Graphe" donne de meilleurs résultats mais souffre d'un plus grand nombre de FPPP. Nous choisissons finalement $\tau_{\mathbf{pos}} = \mathbf{0.1}$ et $\mathbf{\Delta_{1-2}} = \mathbf{0.05}$.

rappel de la classe "Texte" est de 94% et la précision de 29%. L'information de cette catégorie doit être encore affinée dans une étape ultérieure qui nous permettra de faire un tri supplémentaire. Ces mauvais résultats proviennent certainement de la base de données d'apprentissage, trop déséquilibrée, et présentant beaucoup plus d'exemples de cette catégorie que les autres.

La confusion entre "Mixte" et "Texte" ou "Pictogramme" s'explique aisément puisque cette classe est une combinaison des deux. Pour améliorer les performances, il faudrait ajouter un module pour gérer les positions relatives des objets les uns par rapport aux autres.



(a) Rappel (b) FPPP

FIGURE 5.3 – Évolution du rappel et du nombre de Faux Positifs Par Positif (FPPP) pour chaque type de panonceau en fonction de Δ_{1-2} l'écart entre les deux meilleures probabilités de macro-catégories. La classe "Texte" présente le meilleur rappel, 86% pour $\Delta_{1-2} = 0.05$, et la classe "Flèche" le pire avec un rappel de 12%. En revanche, le FPPP vaut presque dix fois plus pour "Texte" que pour "Pictogramme" pour la méthode "Région". C'est donc la classe la moins bien discriminée par la première étape de classification puisqu'il reste un grand nombre de négatifs qui n'ont pas été détectés.

		Classifieur					
		"Négatif"	"Texte"	"Flèche"	"Pictogramme"	"Mixte"	Rappel
Vérité Terrain	"Négatif"	7518	634	6	175	99	**89%**
	"Texte"	498	1401	2	25.5	23.8	**74%**
	"Flèche"	294	489	35	31	7	**11%**
	"Pictogramme"	742	611	0	354	26.2	**32%**
	"Mixte"	429	399	0	89.5	191	**31%**
	Précision	**79%**	**40%**	**81%**	**52%**	**55%**	

TABLE 5.3 – Matrice de confusion de la classification des régions retournées par le détecteur "Région" sur la base de données du tableau 5.1. La précision la plus importante est obtenue pour la classe "Flèche" au prix d'un faible rappel de 11%. La classe "Texte" possède le meilleur rappel (74%) et la moins bonne précision (40%).

5.3.2 Étude de la taille

Les performances dépendent-elles de la taille, et donc de la distance, des panonceaux ? La figure 5.4 apporte un élément de réponse. Elle fournit le nombre de bonnes détections par intervalle

		Classifieur					
		"Négatif"	"Texte"	"Flèche"	"Pictogramme"	"Mixte"	Rappel
Vérité Terrain	"Négatif"	13514	1954	1	248	158	**85%**
	"Texte"	97	1781	0	18	5	**94%**
	"Flèche"	290	566	39	29	17	**12%**
	"Pictogramme"	565	1326	0	531	62	**48%**
	"Mixte"	297	544	0	86	323	**52%**
	Précision	**91%**	**29%**	**98%**	**58%**	**57%**	

TABLE 5.4 – Matrice de confusion de la classification des régions retournées par le détecteur "Graphe" sur la base de données du tableau 5.1. La tendance est la même que pour "Région" (tableau 5.3) avec globalement de meilleures performances tant en rappel qu'en précision. Le rappel pour la classe "Texte" lui est ainsi supérieur de 20% (bien que sa précision soit inférieure).

de hauteur et de largeur pour chaque catégorie pour la méthode "Région". Pour la classe "Texte", il semble que oui. Plus le panonceau est proche, plus les caractères se détachent les uns des autres et permettent de le distinguer d'un "amas" de pixels. Pour les autres classes, à l'exception de "Mixte", pour laquelle le nombre de panonceaux de grande taille est trop faible pour permettre une bonne estimation, la tendance est la même. La confiance dans la sortie du classifieur croît donc à mesure que le panonceau se rapproche.

5.3.3 Suivi spatio-temporel

Afin de valider un panonceau, nous devons être capable de le suivre dans le temps. La confiance que nous aurons dans un panonceau dépend du nombre de fois où il a été détecté. La figure 5.5 illustre, pour deux exemples de panonceaux "Texte" et "Flèche", les sorties du système. La probabilité des exemples classifiés est comptée positivement si la catégorie prédite est correcte, négativement s'ils sont affectés à une autre classe et nulle si l'exemple est considéré comme négatif. Le numéro d'occurrence se réfère à l'image dans laquelle est présent le panonceau, 0 correspondant à la première apparition. Dans le premier cas, toutes les occurrences du panonceau ont été correctement validées et reconnues. Il n'y a aucun faux négatif. Dans le second cas, le panonceau n'est correctement détecté qu'à la fin, lorsqu'il est le plus visible. Le reste du temps, il est confondu avec d'autres catégories. Ce résultat n'est pas surprenant puisque nous avons vu précédemment que la classe "Flèche" présentait le rappel le plus bas.

À droite, nous voyons l'évolution du centre des différents rectangles. Globalement, la trajectoire suivie par les positifs semble cohérente alors que les fausses alarmes se déplacent de façon plus hasardeuse (du moins pour le premier cas). Un suivi temporel permettrait d'éliminer un certain nombre de fausses alarmes résiduelles.

5.3.4 Analyse fonctionnelle

Quelles sont les conditions principales pour valider un panonceau ? Il faut s'assurer qu'il a été bien détecté un certain nombre de fois avant de le considérer comme positif. Pour cela, nous "suivons" chaque panonceau détecté, c'est-à-dire que nous associons à un même panonceau physique tous les objets détectés au cours du temps tels que leur trajectoire soit cohérente. Le modèle de projection de la caméra est approximé par un modèle sténopé [Miura et al., 2000]. Ce dernier n'est valide que sous les hypothèses d'une trajectoire rectiligne du véhicule et d'une vitesse constante entre deux acquisitions, vérifiées car la fréquence de la caméra est de 25 images par seconde.

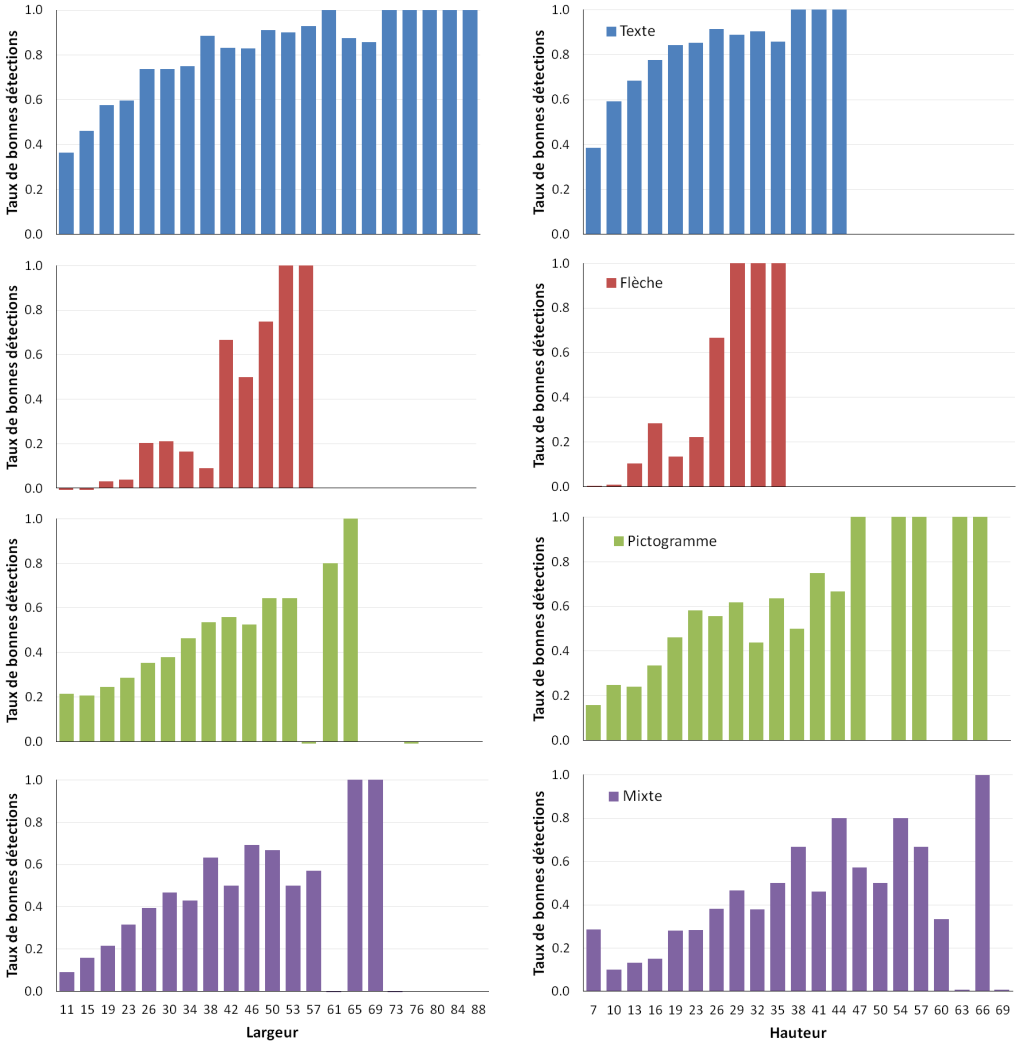
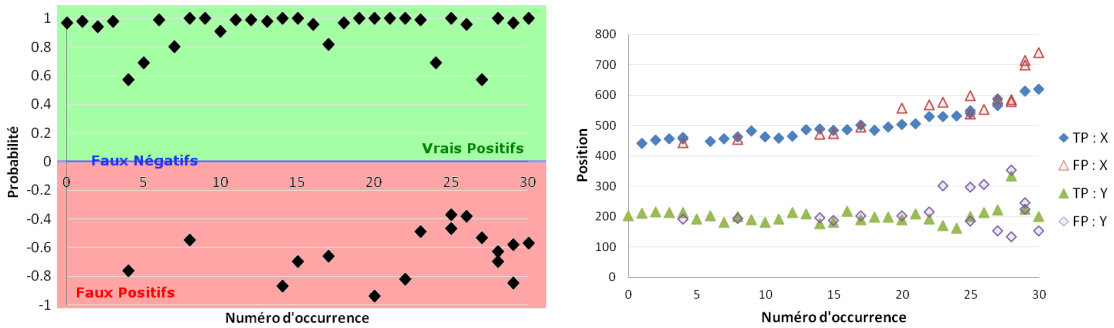
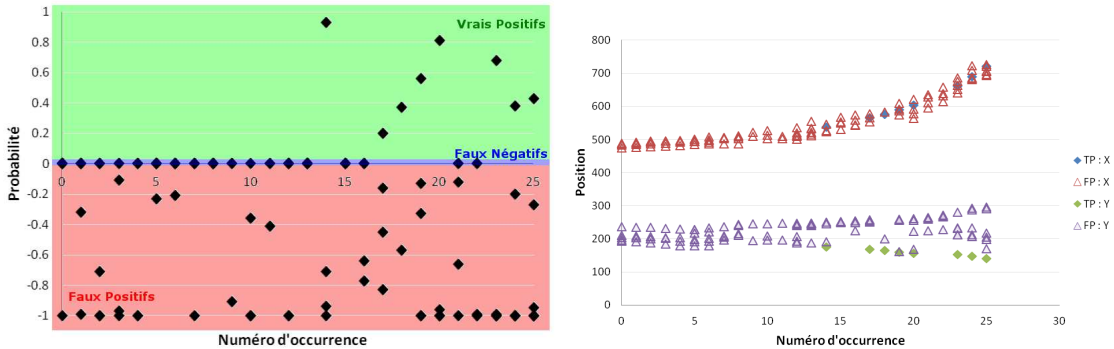


FIGURE 5.4 – Taux de bonnes détections de la méthode "Région" en fonction de la taille (extraite de la vérité terrain) des panonceaux. Globalement, le taux de détections augmente avec la taille, hauteur ou largeur, à l'exception de la classe "Mixte" pour laquelle les résultats sont plus mitigés. À noter que l'apprentissage a été réalisé sur des panonceaux de hauteur ou largeur supérieure à 10 pixels mais que nous parvenons à reconnaître des objets plus petits.

Le tableau 5.5 recense les panonceaux correctement détectés au moins $N_D = 3$ fois. Les performances obtenues à la fin de notre chaîne sont correctes à l'exception de la classe "Flèche". Notre base de données contient des panonceaux très variés, tant dans leur aspect géométrique que sémantique, contrairement à la plupart des études sur le sujet. Ces résultats sont, de plus, obtenus sans réel suivi. Nous n'avons pas pu ajouter de module de *tracking* qui aurait permis de prédire la position future d'un panonceau déjà détecté. Pour améliorer les résultats de la classe "Flèche", nous pourrions dans un premier temps, rééquilibrer notre base d'apprentissage car les panonceaux de type "Texte" semblent favorisés pour le moment. L'utilisation des forêts aléatoires, adaptées pour ce cas de figure, peut être également envisagée.



(a) Exemple de détection d'un panonceau de type "Texte"



(b) Exemple de détection d'un panonceau de type "Flèche"

FIGURE 5.5 – Exemples de résultats de détection pour deux panonceaux de type "Texte" (a) et "Flèche" (b). À gauche, les probabilités de la meilleure classe sont présentées, les bonnes détections sont en positif et les fausses alarmes en négatif. À droite, nous avons l'évolution de la position en X et en Y des bonnes et mauvaises détections. Nous voyons qu'une grande partie des faux positifs pourraient être éliminés par le suivi des positions des panonceaux, leur trajectoire n'étant pas cohérente dans le temps.

	Nombre total	Bonnes détections ($N_D \geq 3$)	
		Graphe	Région
Texte	102	98 (*96%*)	89 (*87%*)
Flèche	23	4 (*17%*)	4 (*17%*)
Pictogramme	55	47 (*86%*)	38 (*69%*)
Mixte	49	36 (*74%*)	21 (*43%*)

TABLE 5.5 – Résultats de détection de panonceaux en terme d'occurrence physique pour les deux détecteurs "Graphe" et "Région".

5.3.5 Performances temporelles

Les différents modules de reconnaissance de panonceaux ont été développés en C++ et implémentés pour être utilisés avec le logiciel RtMaps®. L'évaluation a été réalisée sur un Sony Vaio VPCZ1 Intel i5 Core et pour l'ensemble de notre base de données. Nous avons mesuré les temps de calcul des deux approches de détection de rectangles "Graphe" et "Région" ainsi que notre méthodes de classification, recensés dans le tableau 5.6. En l'état actuel, il est impossible d'envisager l'utilisation de la détection "Région" pour une application temps-réel. De même, l'extraction des caractéristiques globales, PHOG et Proportion, présente un coût algorithmique important. Une

optimisation serait donc nécessaire avec une gestion hiérarchisée des graines de départ qui éviterait de rechercher des régions dans des zones déjà traitées ou l'utilisation d'images intégrales pour le calcul des caractéristiques.

Notre système complet ne pourra donc pas intégrer ces modules immédiatement. La reconnaissance de panonceaux utilisée par la suite correspondra à un algorithme développé par un partenaire de notre projet Speedcam, Daimler. Dans la section 5.5, nous présentons ses caractéristiques et performances.

	Détection		Classification	
	Graphe	Région	Extraction des caractéristiques	SVM
Moyenne	79	722	926	65
Écart-type	65	729	1025	35
$[Min; Max]$	$[9; 713]$	$[81; 11019]$	$[69; 12220]$	$[6; 280]$

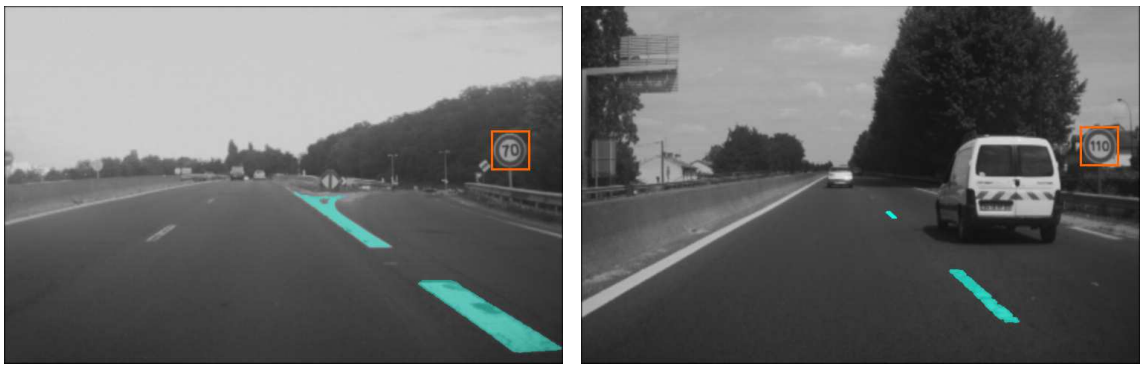
TABLE 5.6 – Temps de calcul (en ms) des différents modules implémentés de reconnaissance de panonceaux. Les valeurs des écarts-types sont très importantes par rapport à la moyenne associée, indiquant la dispersion des mesures effectuées. Concernant la détection des rectangles, la méthode à base de graphe est dix fois plus rapide que celle à base de croissance de régions, rendant alors impossible son utilisation en l'état. Du côté de la classification, l'extraction des caractéristiques (PHOG et Proportion) n'est pas non plus optimale pour le moment.

5.4 Intégration des marquages

5.4.1 Intérêt

Détecter les panneaux de limitation présents dans les images ne suffit pas pour affirmer qu'ils s'appliquent au véhicule. Généralement, les sorties du module de vision correspondent au type de panneau rencontré, celui du panonceau éventuellement associé ainsi que la distance latérale par rapport au véhicule. Différentes situations peuvent donc être à l'origine des mêmes données et il est nécessaire de connaître au maximum l'environnement pour prendre la bonne décision. La figure 5.6 illustre deux situations rencontrées pour lesquelles les informations du module vision caractérisant le panneau sont similaires et conduisent à la même décision. Toutefois, la prise en compte du contexte, des marquages présents notamment, permettrait d'éliminer la première limitation comme s'appliquant à la sortie. Un problème identique se pose lorsque les panonceaux "Flèche" sont pris en compte et que nous ne disposons d'aucun moyen permettant de savoir si le véhicule a effectivement pris la voie de sortie. Pour cela, nous proposons d'introduire un module de détection et classification de marquages que nous associerons à notre module de gestion de panneaux.

Longueur, largeur et espacement des lignes déterminent le type des marquages (figure 5.7). Nous utilisons, pour cette caractérisation, un module réalisé par un partenaire du projet Speedcam, Valeo. Les lignes de bords de route (T4 ou T6) et les lignes de sortie de voie (T2) sont celles qui nous intéressent le plus pour notre étude. Leur présence, à droite ou à gauche du véhicule, ou leur traversée nous donnent une indication précieuse sur la situation. Notre objectif est de maintenir à jour le statut du véhicule afin d'attribuer à la bonne voie le panneau rencontré. À cette fin, nous avons implémenté une machine d'états actualisée grâce aux marquages et un ensemble de files de panneaux.



(a) Limite applicable à la sortie (b) Limite applicable à la route principale

FIGURE 5.6 – Ces deux situations différentes devraient conduire aux mêmes sorties du module de vision basé uniquement sur les panneaux et panonceaux. Le panneau de limite de vitesse est situé à la même distance latérale par rapport au véhicule bien que le contexte soit différent. (a) Le panneau rencontré s'applique à la sortie située à droite du véhicule mais n'est pas accompagné d'un panonceau "Flèche". Seuls les marquages et la connaissance du contexte (un panneau avec flèche rencontré plus tôt ou une information de navigation) permettent d'attribuer la limite à la voie latérale. (b) Le panneau s'applique bien à la route principale dans cette situation mais sa distance latérale est grande car le véhicule se trouve sur la voie la plus à gauche de l'autoroute.

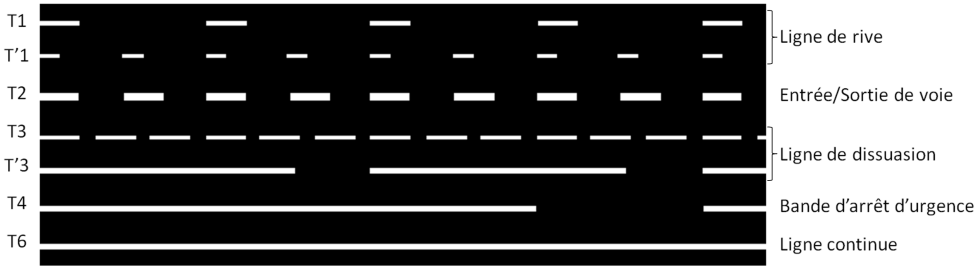


FIGURE 5.7 – Types de marquages existants en France. Chaque type est caractérisé par trois mesures : la largeur, la longueur et l'espacement entre deux traits. Les lignes de rive (T1 et T'1) séparent les voies du même sens de circulation. Les marquages T2 indiquent la présence d'une entrée ou sortie de route. Les lignes de dissuasion (T3 et T'3) interdisent le dépassement des autres véhicules (sauf très lents). Les bords de route sont indiqués par les types T4 (sur autoroute) et T6.

5.4.2 Mise en œuvre

Le modèle proposé est constitué de deux parties, une machine d'états et un ensemble de files de panneaux.

Machine d'état

La première est mise à jour grâce aux informations de marquages. Elle est composée de quatre états, illustrés par la figure 5.8 :

1. **Normal** : le véhicule n'occupe pas de position spécifique sur la route. La situation la plus classique correspond à des marquages de type T1 de chaque côté.
2. **Bord de route** : lorsque la voie de circulation est située tout à droite ou à gauche de la route, un marquage de type T4-T6 y est détecté.

3. **Sortie à proximité** : à proximité d'une sortie, un marquage T2 apparaît.
4. **Sortie** : cet état intermédiaire survient à la traversée d'un marquage de sortie et est conservé quelques mètres avant de mettre à jour l'état en fonction des nouvelles lignes détectées.

(a) **Normal** (b) **Bord de route** (c) **Sortie à proximité**

FIGURE 5.8 – Illustration des types de marquages (en rouge) générant un changement d'état. (a) L'état **Normal** correspond à un état par défaut, typiquement lorsque les marquages à droite et à gauche correspondent à des démarcations de voies de type T1. (b) **Bord de route** (resp. (c) **Sortie à proximité**) est obtenu lorsqu'un marquage de type T4-T6 (resp. T2) est détecté d'un côté du véhicule.

Une fenêtre spatiale de taille fixe est utilisée pour valider les marquages détectés par le module, c'est-à-dire qu'ils ne sont validés qu'après détection du même type sur une certaine distance. Le passage d'un état à l'autre s'opère une fois le type connu et si certaines conditions sont remplies, représentées par la figure 5.9. Il dépend des marquages situés de chaque côté du véhicule. Certains changements sont impossibles, notamment **Normal** → **Sortie** car le second état suppose que nous avons traversé, ou à défaut, détecté d'un côté puis de l'autre, un marquage T2, impliquant un état transitoire par **Sortie à proximité**. Nous avons illustré deux cas de figure : une sortie d'autoroute empruntée (figure 5.10) et un changement de voie puis dépassement de sortie d'autoroute (figure 5.11).

Files de panneaux

Afin de gérer le positionnement des panneaux sur la route et conserver un certain historique des détections précédentes, nous divisons la route en trois espaces - Sortie Gauche, Normal et Sortie Droite -, modélisés par des files. La séparation de la route est illustrée par la figure 5.12. Les panneaux rencontrés sont suivis dans un premier temps sur un nombre donné d'images pour éliminer les trajectoires incohérentes. Une fois validés, ils sont affectés à la voie en fonction de certains critères liés au contexte de position p et de distance d par rapport au centre de la voie de circulation. Pour appartenir à la voie principale, un panneau doit vérifier une des conditions suivantes :

1. État = **Normal** ET $d < d_{Normal}$ ET "Flèche" = NON
   Il s'agit de la situation classique où le véhicule croise un panneau de limitation temporaire sans panonceau "Flèche".
2. État = **Bord de route** ET ($d < d_{Bord}$ OU $p(panneau) \neq p(T4 - T6)$) ET "Flèche" = NON
   En bord de route, les panneaux détectés peuvent s'appliquer à une route parallèle sauf s'ils sont suffisamment proches ou situés de l'autre côté du marquage de bord de route.
3. État = **Sortie à proximité** ET $p(panneau) \neq p(T2)$
   Au niveau d'une entrée ou sortie, un panneau sera pris en compte s'il est situé du côté opposé à cette voie.
4. État = **Sortie** ET $d < d_{Bord}$ ET "Flèche" = OUI
   Lorsque le véhicule a emprunté la sortie, il peut détecter les panneaux affectés à l'ancienne voie ($d \geq d_{Bord}$).

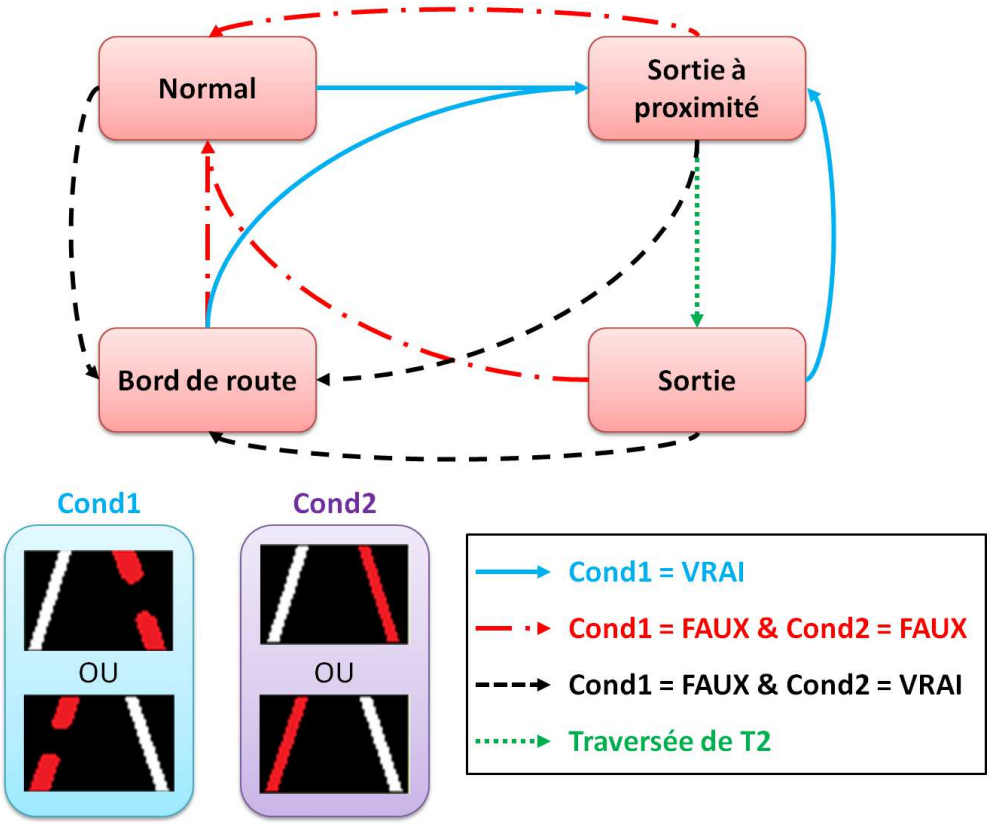


FIGURE 5.9 – Machine d'état employée pour connaître la situation du véhicule sur la route en fonction des informations de marquage. La condition Cond1 correspond à la détection d'un marquage de sortie T2 à droite ou à gauche du véhicule. Cond2, de manière similaire, est vraie dès qu'un marquage de type bord de route T4-T6 est perçu.

Le cas échéant, l'association sera faite avec la voie située du côté p du panonceau. Les distances d_{Bord} et d_{Normal} sont choisies de façon à rejeter les panneaux situés trop loin pour appartenir à la route. Dans l'état **Normal**, la situation la plus extrême correspond à un véhicule situé au centre d'une autoroute à quatre voies avec un panneau détecté au bord de la route. En bord de voie, la distance moyenne est nettement plus faible. Nous avons défini $d_{Bord} = 2.0 * L$ et $d_{Normal} = 3.5 * L$ où $L = 3.5$ m correspond à la largeur standard d'une voie en France.

En cas de franchissement d'un marquage de type T2, toutes les files sont décalées du côté opposé au mouvement du véhicule (figure 5.13). Les panneaux sont finalement appliqués une fois dépassés.

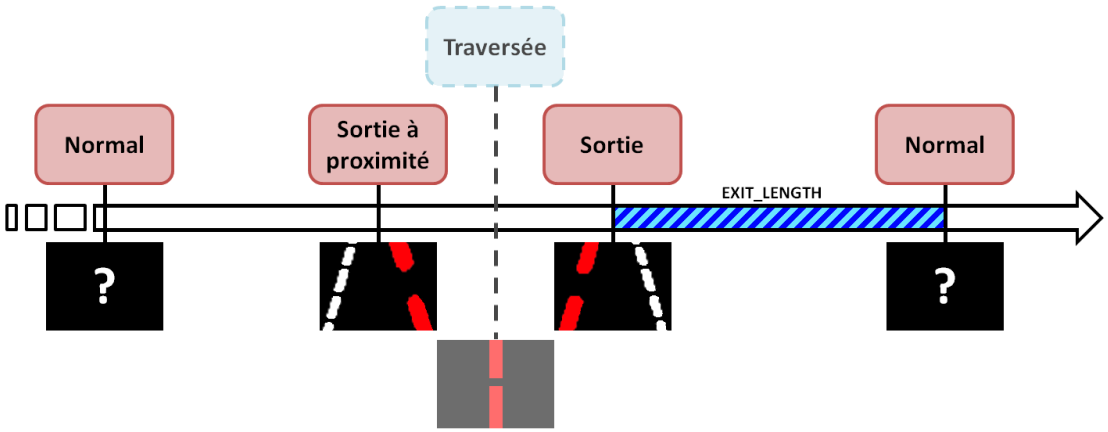


FIGURE 5.10 – Situation de sortie d'autoroute. Le véhicule est par défaut dans l'état **Normal** en l'absence de connaissance sur les marquages. La détection d'un type T2 à droite nous amène à l'état **Sortie à proximité** tant qu'il est présent et n'a pas été traversé. Une alerte est ensuite levée lorsque la ligne est franchie ou que le module détecte que le marquage a été successivement détecté de chaque côté du véhicule. La machine passe alors à **Sortie** et y reste pendant une distance EXIT_LENGTH ou tant qu'aucun nouveau marquage ne provoque un changement d'état.

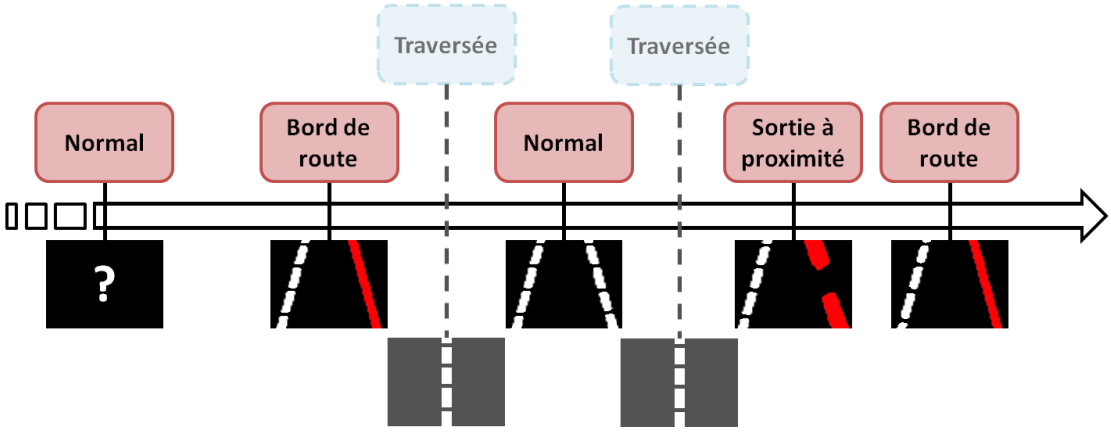
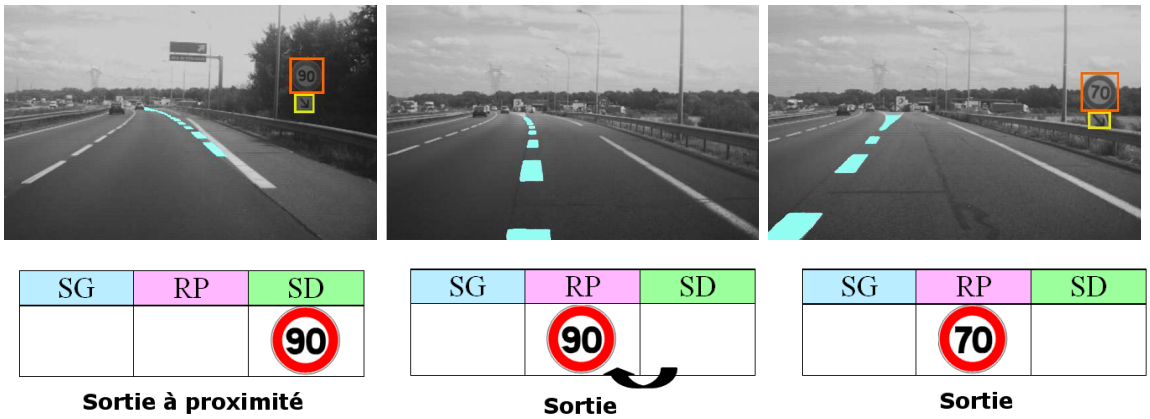


FIGURE 5.11 – Situation typiquement rencontrée sur autoroute avec changement de voie et passage au voisinage d'une sortie. Après avoir circulé sur la voie la plus à droite de la route (marquage de type T4-T6), le véhicule traverse une première fois un marquage de démarcation passant de **Bord de route** à **Normal** puis traverse une seconde fois au voisinage d'une sortie. L'état est alors **Sortie à proximité** tant que le marquage de sortie T2 est présent puis de nouveau **Bord de route**.

5.5 Évaluation

Nous avons ensuite évalué l'apport des différents modules dans la détermination de la vitesse limite. Dans un premier temps, nous résumons les architectures implémentées ainsi que les différents modules mis en œuvre. Ensuite, nous présentons les performances générales obtenues pour chaque source d'informations et pour la fusion que nous comparons aux résultats du chapitre 4. Enfin, nous étudions le cas idéal d'un module de vision gérant parfaitement panneaux, panonceaux et marquages pour mettre en perspective l'utilité de notre fusion.

FIGURE 5.12 – Séparation de la route en trois espaces. La Route Principale (RP) correspond à l'espace sur lequel circule actuellement le véhicule et les panneaux appartenant à cette file seront donc ceux qui s'appliqueront. Les voies de Sortie Gauche (SG) et Droite (SD) sont associées aux panneaux détectés sur les voies latérales.



(a) Le panneau détecté est associé à la voie de sortie

(b) Franchissement du marquage de sortie

(c) Le panneau est applicable à la voie courante

FIGURE 5.13 – Illustration d'une situation de franchissement de marquage de sortie. (a) Un panneau a été détecté à droite de la route lorsque le véhicule est dans l'état **Sortie à proximité**. La présence du panonceau "Flèche" ainsi que le marquage de sortie T2 situé du même côté permettent d'associer la vitesse limite à la sortie droite (SD). (b) Après franchissement du marquage, toutes les files de panneau sont décalées vers la gauche et la limite précédente se retrouve être appliquée à la route principale (RP). L'état devient **Sortie**. (c) Le panneau 70 se trouve du côté opposé au marquage de sortie et possède un panonceau "Flèche". Puisque l'état courant est **Sortie**, la vitesse est appliquée à la route principale.

5.5.1 Implémentation réalisée

Toutes nos évaluations ont été réalisées avec le logiciel RtMaps® sur un Sony Vaio VPCZ1 Intel i5 Core et pour l'ensemble de notre base de données (tableau 4.6). Notre système complet se présente sous la forme d'un diagramme comprenant différents blocs, échangeant entre eux des données standards, comme des images, des ROI ou des mesures. En tant qu'utilisateur, nous pouvons développer nos propres algorithmes en C++ et intégrer ces modules au diagramme.

La majorité des éléments ont été développés pendant cette thèse à l'exception du module de reconnaissance de panneaux/panonceaux final et celui de caractérisation des marquages. En effet, la classification en l'état actuel n'étant pas suffisante pour une application embarquée, nous avons utilisé un module développé par Daimler, partenaire de notre projet Speedcam. Les principales

catégories de panonceaux sont reconnus dont flèches, camions, distances, tonnages et textes sur une ou plusieurs lignes. Toutefois, les messages ne sont pas déchiffrés et le système peut seulement dire si le panonceau appartient aux différentes classes. Un suivi temporel est également incorporé et les sorties disponibles correspondent aux panneaux et panonceaux une fois validés. De même, la détection des marquages n'est pas le cœur de notre sujet et nous avons préféré utiliser un module réalisé par Valeo.

L'évaluation de ces deux éléments est difficile car le détail de l'implémentation n'est pas accessible. Les seules choses mesurables sont le nombre de panneaux et panonceaux physiques correctement détectés ainsi que le pourcentage de temps pendant lequel les marquages sont correctement classifiés. Les résultats sont rassemblés dans le tableau 5.7. Nous constatons que les résultats ne sont pas encore parfaits et que bon nombre de limitations temporaires sont mal gérées. Ces tendances se retrouvent dans les performances générales des différents modules du tableau 5.8.

Bases de données	Panneaux			Panonceaux			Marquages	
	GT	Vision		GT	Vision		GT (s)	Vision
		Précision	Rappel		Précision	Rappel		
A4_Reims	57	87%	58%	18	83%	28%	2173	1520 (70%)
A4_A86	57	93%	47%	28	93%	50%	2149	741 (34%)
Paris_Rouen	113	96%	62%	27	100%	48%	4715	3410 (82%)
Cherisy	67	77%	51%	34	89%	24%	3068	1552 (51%)

TABLE 5.7 – Évaluation des modules développés par les partenaires du projet Speedcam et utilisés dans notre implémentation. Les panneaux et panonceaux sont détectés par le même module. Les performances ne sont pas encore optimales puisqu'encore beaucoup de limitations temporaires sont mal gérées (valeurs de rappel assez faibles pour les panneaux et panonceaux) et que les changements de marquages sont mal détectés.

5.5.2 Performances générales

Le tableau 5.8 montre les résultats obtenus en ajoutant les modules de gestion des panonceaux et des marquages au système complet. Nous intégrons de plus la gestion des conditions climatiques grâce aux informations du bus CAN. Nous avons évalué notre système en terme de durée pendant laquelle la sortie correspond à la vérité terrain. L'amélioration obtenue atteint les 20% pour la séquence *Cherisy*. Ces meilleures performances s'expliquent principalement par les modifications apportées au capteur de navigation et la bonne gestion des panneaux de sortie de route. Le premier aspect a été évoqué dans le chapitre 4. Le second est évident au vu des résultats. En effet, un module de gestion de panneaux seul prenait en compte TOUS ceux qu'il rencontrait alors qu'avec les marquages un tri plus sélectif est réalisé. Les séquences *A4_A86* et *Paris_Rouen* voient ainsi les performances du module de vision augmenter de presque 20% par rapport à la vision seule. En revanche, pour *Cherisy*, aucune différence notable du point de vue temporel du moins, et pour *A4_Reims*, la tendance est même inversée. En effet, les capteurs ne sont pas parfaits et la combinaison des deux modules de détection de panneaux/panonceaux et de marquages peut mener à certaines erreurs.

La figure 5.14 compare les sorties des différents modules (vision et navigation) puis la fusion de données à la vérité terrain. Le module de vision est de plus évalué par rapport à un capteur **parfait**, qui traiterait de manière idéale toutes les limitations temporaires rencontrées. En début de séquence, la navigation est "aveugle", la définition du réseau étant insuffisante. C'est donc la vision qui prend le dessus. Autour de $t = 2000$, en revanche, la navigation l'emporte ce qui assure

de bonnes performances générales.

Séquence	**Performances (temps)**					
	Navigation	Vision	Marquages	Fusion	*Par rapport à*	
					Navigation	*Fusion (chapitre 4)*
A4_Reims	86%	42%	36%	88%	*+2%*	*+8%*
A4_A86	98%	79%	97%	99%	*-1%*	*+2%*
Paris_Rouen	74%	46%	65%	87%	*+13%*	*+15%*
Cherisy	71%	58%	59%	91%	*+20%*	*+22%*

TABLE 5.8 – Résultats obtenus avec la fusion de données de navigation et de vision, dont les panneaux, les panonceaux et les marquages. Les sorties sont comparées à la vérité terrain en terme de temps. Nous estimons également les améliorations apportées par rapport au meilleur capteur, la navigation, en terme de performances et par rapport au système précédemment développé dans le chapitre 4. La fusion de données donne de meilleurs résultats pour toutes les séquences, l'amélioration pouvant atteindre 20% pour la séquence *Cherisy* par rapport à la navigation.

5.5.3 Apport de la gestion des panonceaux et marquages

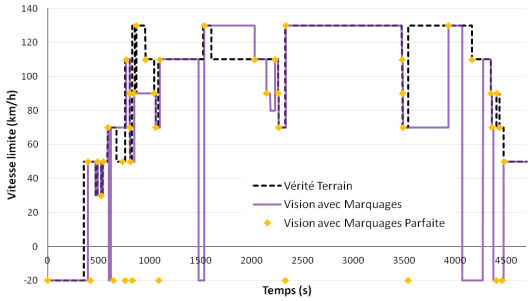
La figure 5.15 illustre les sorties du module de vision avant et après ajout des marquages et des panonceaux pour la séquence *Cherisy*. Nous les comparons à la vérité terrain mais également au cas d'un module **parfait**, c'est-à-dire qui détecterait et gérerait correctement tous les éléments rencontrés. Lorsque seuls les panneaux sont pris en compte, de nombreuses erreurs sont générées, causées notamment par les sorties d'autoroute croisées. En ajoutant la connaissance sur les panonceaux et les marquages, le capteur serait correct jusqu'à 87% du temps dans le cas idéal. Pour la séquence *Cherisy*, leur apport est évident. Au début de la séquence, entre les instant $t = 0$ et $t = 350$, aucun panneau n'est présent et la vitesse limite réelle ne dépend pas des panneaux mais de la navigation contrairement au reste de la séquence.

5.5.4 Cas d'une vision parfaite

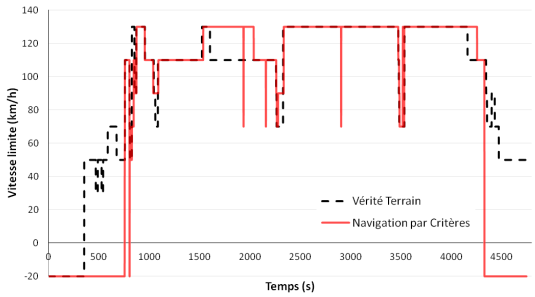
Si nous étudions les sorties parfaites des modules de vision, l'apport des marquages et panonceaux est encore plus manifeste (tableau 5.9). Dans ce cas, les performances dépendent uniquement de la signalisation temporaire présente sur la route. Pour la séquence *A4_A86*, des panneaux sont présents 97% du temps pour rappeler aux automobilistes les limitations de vitesse. En ajoutant à cela une très bonne couverture cartographique, nous pouvons facilement expliquer les bons résultats de notre système. Pour les autres séquences, la prise en compte du contexte de conduite est nécessaire pour assurer la bonne gestion des limitations de vitesse.

5.6 Conclusion

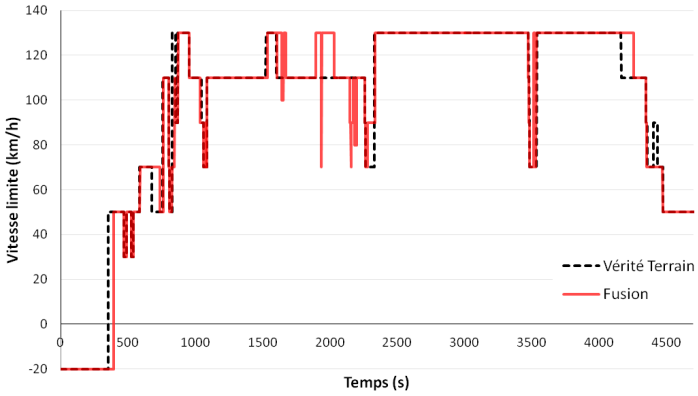
Le système complet, intégrant panonceaux, marquages et fusion de données, a démontré son efficacité. Nous avons vu en effet que la détection des panneaux de limitation de vitesse et des panonceaux éventuels ne suffisait pas. La prise en compte des marquages est nécessaire pour associer les bonnes limites aux voies qu'elles concernent. En la combinant à une machine d'états, nous sommes ainsi capables de gérer l'historique en modifiant dynamiquement les limitations de vitesse en fonction de la voie de circulation. Les performances atteignent, pour nos bases de données, 87% du temps de parcours avec la bonne vitesse et jusqu'à 99% pour la meilleure (figure 5.16).



(a) Vision avec Marquages et Panonceaux



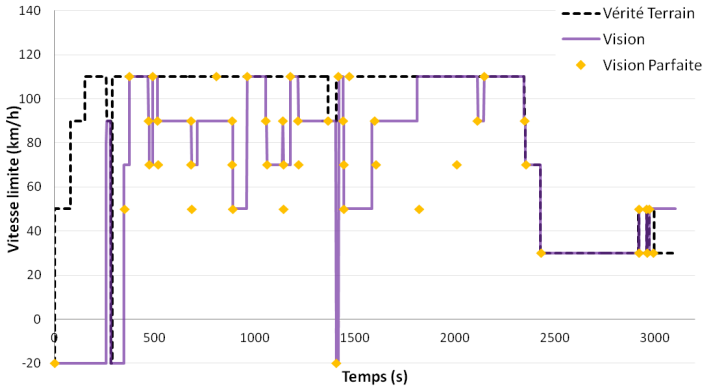
(b) Navigation



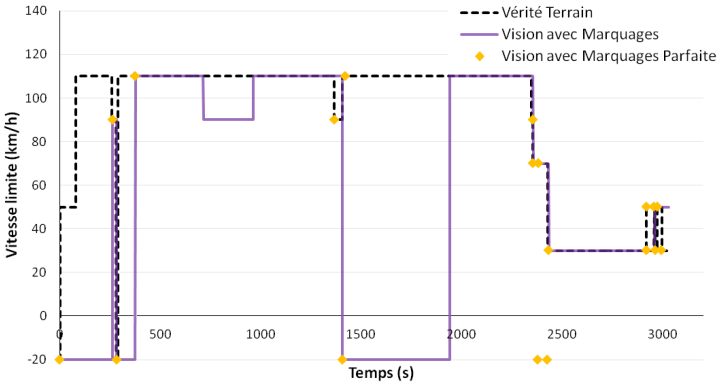
(c) Fusion de données

FIGURE 5.14 – (a) La sortie du module de vision pour la séquence *Paris_Rouen* est comparée à la vérité terrain et à un capteur parfait. Lorsque le véhicule franchit un marquage d'entrée de voie, en cas d'entrée sur autoroute par exemple, le module n'a aucune connaissance de la nouvelle limitation en vigueur et affiche donc -20 (ce qui correspond à l'ignorance). Un fonctionnement parfait donnerait une courbe passant par tous les losanges jaunes. (b) Sortie du module de navigation après gestion des attributs extraits de la cartographie. Les erreurs du début de séquence correspondent à des situations de limitations temporaires, dépendant des panneaux. (c) Résultats de la fusion de données. Le début de la séquence est correctement traité grâce au module de vision tandis que la navigation prend le pas vers la fin de la séquence et pallie aux défaillances de la vision. En $t = 1500$, bien que les deux sources d'information donnent une vitesse de 130 km/h, la sortie réelle est de 110 km/h à cause des conditions climatiques. Pour prendre en compte la pluie, nous intégrons un capteur supplémentaire, le bus CAN, qui nous indique si les essuie-glaces sont enclenchés. Le résultat de la fusion est alors correct et la vitesse limite est 110 km/h.

Pour obtenir des résultats en total accord avec le cadre de fusion de données, il serait intéressant d'estimer la fiabilité du détecteur de lignes ainsi que du module d'assignation des panonceaux. En calculant des probabilités relatives aux types de marquages ainsi qu'aux différentes associations panonceaux-voies de circulation, nous aurions des masses de confiance plus fiables.



(a) Panneaux seuls.

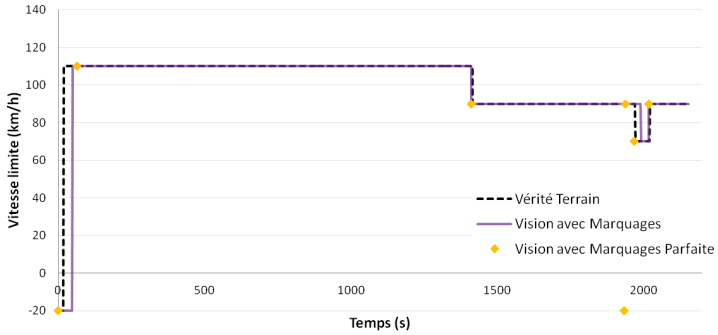


(b) Avec gestion des marquages et panonceaux.

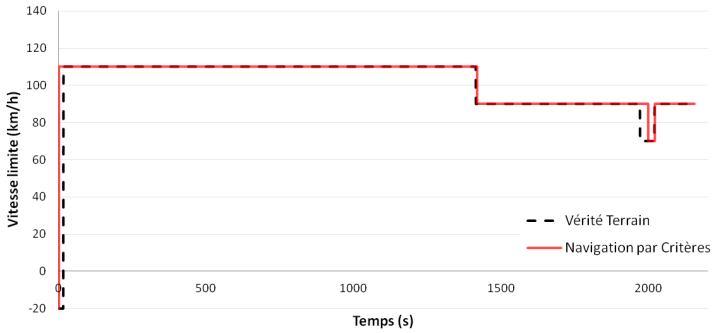
FIGURE 5.15 – (a) Sortie du module de vision avec panneaux seuls (continu violet) comparée à un module de vision parfait (losanges jaunes) et à la vérité terrain (pointillés noirs) pour la séquence *Cherisy*. Lorsque le système ne prend en compte que les panneaux, les fausses alarmes sont nombreuses et dûes en grande partie aux sorties d'autoroute. Il est à noter que le module est loin d'être parfait et manque certaines limitations de vitesse (losanges jaunes).(b) Sortie du module de vision avec marquages. Les différences principales avec le module parfait se situent aux instants $t = 715$ et $t = 1400$, correspondant respectivement à un panneau 90 pris en compte à tort et un 110 ignoré. En $t = 1950$, un second panneau 110 est cette fois pris en compte et la sortie du module vision est alors rectifiée.

	Vision parfaite Panneaux	Vision parfaite Marquages et Panneaux/Panonceaux
A4_Reims	52%	61% (*+9%*)
A4_A86	50%	97% (*+47%*)
Paris_Rouen	43%	72% (*+29%*)
Cherisy	46%	87% (*+41%*)

TABLE 5.9 – Comparaison des performances des modules de vision avec et sans gestion des marquages et panonceaux dans le cas d'un fonctionnement **parfait**. Les améliorations apportées sont sensibles entre les deux versions. Le plus grand gain concerne la séquence *A4_A86* et s'explique par le nombre important de sorties croisées mais non empruntées, situations correctement gérées avec panonceaux et marquages.



(a) Vision



(b) Navigation

FIGURE 5.16 – (a) Sortie du module de vision avec marquages (continu violet) comparée à un module de vision parfait (losanges jaunes) et à la vérité terrain (pointillés noirs) pour la séquence *A4_A86*. Chaque changement dans la vérité terrain est signalé par un panneau de limitation de vitesse, ce qui facilite le travail de la fusion. (b) Sortie du module de navigation en fonction du contexte de conduite. Les deux capteurs sont d'accord sur la quasi totalité du parcours, d'où une précision de 99%.

Chapitre 6

Conclusion

Les objectifs principaux de cette thèse étaient de développer un module de reconnaissance de panonceaux, d'améliorer l'interprétation et le traitement des données de navigation et de fusionner efficacement les deux sources (vision et navigation) en vue de déterminer la vitesse limite d'un véhicule.

La première partie de notre étude concernait les panonceaux car peu de travaux s'y consacrent alors que leur prise en compte est essentielle pour la gestion des limitations temporaires. Nous avons donc développé un système complet de détection et reconnaissance de ces objets.

Dans un premier temps (chapitre 2), nous avons proposé une méthode inédite de détection de rectangles en combinant un algorithme de reconstruction morphologique et une croissance de régions. Cette association nous a permis de ne rechercher dans l'image que les régions fortement contrastées, *i.e.* celles contenant une information, et de les utiliser comme graines de départ pour notre croissance de régions. Nous avons comparé notre technique à un benchmark de trois autres approches de traitement d'image basées sur les contours, la colorimétrie et les graphes. Elle s'est avérée donner de moins bons résultats en terme de rappel que celle à base de graphes, inspirée de [Felzenszwalb and Huttenlocher, 2004], avec environ 75% de bonnes détections contre 80% pour la seconde sur une base de données de plus de 12000 panonceaux. Toutefois, le nombre de fausses alarmes est nettement plus faible de par notre sélection de régions initiales.

Dans un second temps (chapitre 3), nous avons conçu une architecture de classifieurs à base de descripteurs globaux et de SVMs. Pour simplifier la reconnaissance, nous avons proposé de décomposer les panonceaux en quatre macro-catégories, "Texte", "Flèche", "Pictogramme" et "Mixte". Chacune d'entre elles a été caractérisée à l'aide d'un *Pyramid HOG* associé à un vecteur "Proportion" correspondant au ratio de pixels sombres sur clairs. Un rappel supérieur à 90% et une précision supérieure à 80% augurent de bonnes performances pour le système complet pour une base de test d'environ 5200 positifs et 13800 négatifs.

Nous nous sommes ensuite intéressés à la gestion des informations provenant de la navigation (chapitre 4). Cette source apporte des renseignements sur le contexte de conduite, comme le type de route ou la présence en ville ou non, résumés sous forme d'attributs. Grâce à ces derniers, nous calculons la confiance qu'à le capteur en chaque limitation de vitesse en fonction de l'environnement. Notre principale contribution a consisté à prendre en compte la fiabilité de cette source en fonction du nombre de satellites visibles, de la qualité de numérisation de la carte et du degré de précision des informations stockées. La fusion de données est ensuite réalisée à l'aide de la théorie de Dempster-Shafer. Nous avons confronté cette nouvelle approche à celle de [Lauffen-

burger et al., 2008] sur un ensemble de quatre séquences pour un total de plus 3h20 d'acquisitions et 263km. Une nette amélioration a été constaté, passant de 58% à 80% pour la meilleure séquence.

Enfin, un système complet a été développé, mettant en œuvre panonceaux, marquages et fusion de données (chapitre 5). La connaissance des voies situées autour de la voiture permet la gestion des panneaux et associe les limitations de vitesse à la bonne route. Nous avons alors implémenté une machine d'états, alimentée par les types de lignes détectés et les panneaux/panonceaux présents, capable de mettre à jour la limitation temporaire en vigueur. Toutes ces améliorations combinées nous ont permis d'atteindre de très bonnes performances. Même si les capteurs ne sont pas parfaits, le système fournit la bonne limitation de vitesse plus de 87% du temps.

De nombreuses améliorations sont toutefois possibles. Concernant le capteur de vision, le module de reconnaissance de panonceaux est pour le moment incomplet puisqu'il manque encore une étape de classification, notamment pour les classes "Texte" et "Pictogramme". Un algorithme OCR pourrait être utilisé pour déchiffrer les premiers par exemple, à condition d'avoir une résolution de caméra suffisante. Une fois correctement détectées, les informations sémantiques présentes sur les panonceaux pourraient permettre de savoir la durée pendant laquelle la vitesse s'applique ou si la catégorie représentée concerne le véhicule.

L'estimation des attributs a été réalisée de manière empirique. Pour obtenir des valeurs plus représentatives, nous pourrions utiliser une grande base de données GPS afin de relier la probabilité d'occurrence des différents attributs aux vitesses limites.

Enfin, une méthode d'estimation de fiabilité des capteurs pourrait être ajoutée au système. Pour cela, nous pourrions comparer les sorties du module de navigation à celles obtenues avec un GPS de type RTK et une base de données à jour. Pour la caméra, il faudrait prendre en compte les conditions climatiques, car en temps de pluie ou de brouillard, il semble évident que les performances vont chuter.

Annexe A

Publications

1. *Improvement of Multisensor Fusion in Speed Limit Determination by Quantifying Navigation Reliability*, Anne-Sophie Puthon, Fawzi Nashashibi, Benazouz Bradai, IEEE International Conference on Intelligent Transportation Systems (ITSC), 2010, pp 855-860.
2. *A Complete System to Determine the Speed Limit by Fusing a GIS and a Camera*, Anne-Sophie Puthon, Fawzi Nashashibi, Benazouz Bradai, IEEE International Conference on Intelligent Transportation Systems (ITSC), 2011, pp 1686-1691.
3. *Subsign Detection with Region-Growing from Contrasted Seeds*, Anne-Sophie Puthon, Fabien Moutarde, Fawzi Nashashibi, IEEE Intelligent Transportation Systems (ITSC), 2012, pp 969-974.

Bibliographie

[Bahlmann et al., 2008] Bahlmann, C., Pellkofer, M., Giebel, J. and Baratoff, G., 2008. Multi-Modal Speed Limit Assistants : Combining Camera and GPS Maps. In : IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, pp. 132–137.

[Bahlmann et al., 2005] Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M. and Koehler, T., 2005. A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information. In : IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, pp. 255–260.

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L., 2008. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding 110(3), pp. 346–359.

[Bloch and Maitre, 1998] Bloch, I. and Maitre, H., 1998. Fusion of Image Information under Imprecision. Aggregation and Fusion of Imperfect Information pp. 189–213.

[Bosch et al., 2007a] Bosch, A., Zisserman, A. and Muñoz, X., 2007a. Image Classification using Random Forests and Ferns. In : IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, pp. 1–8.

[Bosch et al., 2007b] Bosch, A., Zisserman, A. and Munoz, X., 2007b. Representing Shape with a Spatial Pyramid Kernel. In : ACM International Conference on Image and Video Retrieval, New York, NY USA, pp. 401–408.

[Boser et al., 1992] Boser, B. E., Guyon, I. M. and Vapnik, V. N., 1992. A Training Algorithm for Optimal Margin Classifiers. In : ACM Workshop on Computational Learning Theory, New York, NY, USA, pp. 144–152.

[Breiman, 2001] Breiman, L., 2001. Random Forests. Machine Learning 45(1), pp. 5–32.

[Bunke, 1990] Bunke, H., 1990. Syntactic and Structural Pattern Recognition : Theory and Applications. World Scientific.

[Chan and Vese, 2001] Chan, T. F. and Vese, L. A., 2001. Active Contours without Edges. Transactions on Image Processing 10(2), pp. 266–277.

[Chanda and Dellaert, 2004] Chanda, G. and Dellaert, F., 2004. Grammatical Methods in Computer Vision : An Overview. Technical report.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J., 2011. LIBSVM : A Library for Support Vector Machines. Transactions on Intelligent Systems and Technology 2(3), pp. 1–27.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. In : IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, pp. 886–893.

[Daniel and Lauffenburger, 2011] Daniel, J. and Lauffenburger, J.-P., 2011. Conflict Management in Multi-sensor Dempster-Shafer Fusion for Speed Limit Determination. In : IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, pp. 985–990.

[Dasarathy, 1997] Dasarathy, B. V., 1997. Sensor Fusion Potential Exploitation - Innovative Architectures and Illustrative Applications. Proceedings of the IEEE 85(1), pp. 24–38.

[de La Escalera et al., 2003] de La Escalera, A., Armingol, J. M. and Mata, M., 2003. Traffic Sign Recognition and Analysis for Intelligent Vehicles. Image and Vision Computing 21(3), pp. 247–258.

[de la Escalera et al., 1997] de la Escalera, A., Moreno, L. E., Salichs, M. A. and Armingol, J. M., 1997. Road Traffic Sign Detection and Classification. Transactions on Industrial Electronics 44(6), pp. 848–859.

[Dempster, 1967] Dempster, A. P., 1967. Upper and Lower Probabilities Induced by a Multivalued Mapping. The Annals of Mathematical Statistics 38(2), pp. 325–359.

[Dougherty, 1992] Dougherty, E. R., 1992. An Introduction to Morphological Image Processing. Society of Photo Optical.

[Dubois and Prade, 1988] Dubois, D. and Prade, H., 1988. Possibility Theory : An Approach to Computerized Processing of Uncertainty. Plenum Press.

[Ehrlich et al., 2003] Ehrlich, J., Marchi, M., Jarri, P., Salesse, L., Guichon, D., Dominois, D. and Leverger, C., 2003. LAVIA , The French ISA Project : Main Issues and First Results on Technical Tests. In : IEEE Intelligent Transport Systems, pp. 1–11.

[Escalera and Radeva, 2004] Escalera, S. and Radeva, P., 2004. Fast Greyscale Road Sign Model Matching and Recognition. Recent Advances in Artificial Intelligence Research and Development pp. 69–76.

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. F. and Huttenlocher, D. P., 2004. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision 59(2), pp. 167–181.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E., 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55(1), pp. 119–139.

[Garcia-Garrido et al., 2006] Garcia-Garrido, M. A., Sotelo, M. A. and Martin-Gorostiza, E., 2006. Fast Traffic Sign Detection and Recognition under Changing Lighting Conditions. In : IEEE International Conference on Intelligent Transportation Systems, Toronto, Ont., Canada, pp. 811–816.

[Garza-Jinich et al., 1999] Garza-Jinich, M., Meer, P. and Medina, V., 1999. Robust Retrieval of Three-Dimensional Structures from Image Stacks. Medical Image Analysis 3(1), pp. 21–35.

[Gavrila, 1999] Gavrila, D. M., 1999. Traffic Sign Recognition Revisited. In : DAGM Symposium für Mustererkennung, Bonn, Germany, pp. 86–93.

[Gdalyahu and Weinshall, 1999] Gdalyahu, Y. and Weinshall, D., 1999. Flexible Syntactic Matching of Curves and its Application to Automatic Hierarchical Classification of Silhouettes. Transactions on Pattern Analysis and Machine Intelligence 21(12), pp. 1312–1328.

[Hamdoun et al., 2008] Hamdoun, O., Bargeton, A., Moutarde, F., Bradai, B. and Chanussot, L., 2008. Detection and Recognition of End-of-Speed-Limit and Supplementary Signs for Improved European Speed Limit Support. In : World Congress on Intelligent Transport Systems, New York, NY, USA, pp. 7–10.

[Harris and Stephens, 1988] Harris, C. and Stephens, M., 1988. A Combined Corner and Edge Detector. In : Alvey Vision Conference, Manchester, UK, pp. 147–152.

[Herbin-Sahler et al., 2007] Herbin-Sahler, A., Chanussot, L. and Moutarde, F., 2007. Procédé de Détection d'un Objet Cible.

[HPLabs, 2005] HPLabs, 2005. Tesseract-OCR.

[Hu, 1962] Hu, M.-K., 1962. Visual Pattern Recognition by Moment Invariants. Transactions on Information Theory 8(2), pp. 179–187.

[Huang et al., 2008] Huang, H., Gu, M. and Chao, H., 2008. An Efficient Method of License Plate Location in Natural-Scene Image. In : IEEE International Conference on Fuzzy Systems and Knowledge Discovery, Jinan, Shandong, China, pp. 15–19.

[Idrissa and Acheroy, 2002] Idrissa, M. and Acheroy, M., 2002. Texture Classification using Gabor Filters. Pattern Recognition Letters 23(9), pp. 1095–1102.

[Instruction Interministérielle Relative à la Signalisation de Direction, 1982] Instruction Interministérielle Relative à la Signalisation de Direction, 1982. Technical report, Service de l'exploitation routière et de la sécurité ; Ministère de l'Intérieur et décentralisation ; Direction de la réglementation et contentieux.

[Jain et al., 2000] Jain, A. K., Duin, R. P. W. and Mao, J., 2000. Statistical Pattern Recognition : A Review. Transactions on Pattern Analysis and Machine Intelligence 22(1), pp. 4–37.

[Jamson, 2006] Jamson, S., 2006. Would Those Who Need ISA, Use it ? Investigating the Relationship between Drivers' Speed Choice and their Use of a Voluntary ISA System. Transportation Research 9(3), pp. 195–206.

[Jung and Schramm, 2004] Jung, C. R. and Schramm, R., 2004. Rectangle Detection based on a Windowed Hough Transform. In : IEEE Brazilian Symposium on Computer Graphics and Image Processing, Curitiba, Brazil, pp. 113–120.

[Kang et al., 1994] Kang, D. S., Griswold, N. C. and Kehtarnavaz, N., 1994. An Invariant Traffic Sign Recognition System Based on Sequential Color Processing and Geometrical. In : IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 88–93.

[Keller et al., 2008] Keller, C. G., Sprunk, C., Bahlmann, C., Giebel, J. and Baratoff, G., 2008. Real-Time Recognition of U.S. Speed Signs. In : IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, pp. 518–523.

[Kokar and Kim, 1994] Kokar, M. M. and Kim, K., 1994. Preface to the Special Section on Data Fusion : Architectures and Issues. Control Engineering Practice 2(5), pp. 803–809.

[Kröse and van der Smagt, 1996] Kröse, B. and van der Smagt, P., 1996. Introduction to Neural Networks.

[Lagunovsky and Ablameyko, 1997] Lagunovsky, D. and Ablameyko, S., 1997. Fast Line and Rectangle Detection by Clustering and Grouping. In : International Conference on Computer Analysis of Images and Patterns, Kiel, Germany, pp. 503–510.

[Lampert et al., 2008] Lampert, C. H., Blaschko, M. B. and Hofmann, T., 2008. Beyond Sliding Windows : Object Localization by Efficient Subwindow Search. In : IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, pp. 1–8.

[Lauffenburger et al., 2008] Lauffenburger, J.-P., Bradai, B., Basset, M. and Nashashibi, F., 2008. Navigation and Speed Signs Recognition Fusion for Enhanced Vehicle Location. In : World Congress of the International Federation of Automatic Control, Seoul, Republic of Corea, pp. 2069–2074.

[Li and Shawe-Taylor, 2005] Li, S. and Shawe-Taylor, J., 2005. Comparison and Fusion of Multiresolution Features for Texture Classification. Pattern Recognition Letters 26(5), pp. 633–638.

[Lin and Fu, 1986] Lin, W.-C. and Fu, K.-S., 1986. A Syntactic Approach to Three-Dimensional Object Recognition. Transactions on Systems, Man, and Cybernetics 16(3), pp. 405–422.

[Liu et al., 2011] Liu, W., Lv, J., Gao, H., Duan, B., Yuan, H. and Zhao, H., 2011. An Efficient Real-Time Speed Limit Signs Recognition Based on Rotation Invariant Feature. In : IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, pp. 998–1003.

[Lowe, 2004] Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), pp. 91–110.

[Loy and Barnes, 2004] Loy, G. and Barnes, N., 2004. Fast Shape-based Road Sign Detection for a Driver Assistance System. In : IIEEE/RSJ nternational Conference on Intelligent Robots and Systems, Sendai, Japan, pp. 70–75.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W., 1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics 5(4), pp. 115–133.

[Meyer et al., 2003] Meyer, D., Leisch, F. and Hornik, K., 2003. The Support Vector Machine under Test. Neurocomputing 55(55), pp. 169–186.

[Mikolajczyk and Schmid, 2001] Mikolajczyk, K. and Schmid, C., 2001. Indexing based on Scale Invariant Interest Points. In : IEEE International Conference on Computer Vision, Vancouver, BC, Canada, pp. 525–531.

[Miura et al., 2000] Miura, J., Kanda, T. and Shirai, Y., 2000. An Active Vision System for Real-Time Traffic Sign Recognition. In : IEEE International Conference on Intelligent Transportation Systems, Dearborn, MI, USA, pp. 52–57.

[Mori et al., 1992] Mori, S., Suen, C. Y. and Yamamoto, K., 1992. Historical Review of OCR Research and Development. Proceedings of the IEEE 80(7), pp. 1029–1058.

[Müller et al., 2001] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K. and Schölkopf, B., 2001. An Introduction to Kernel-based Learning Algorithms. Transactions on Neural Networks 12(2), pp. 181–202.

[Mumford and Shah, 1985] Mumford, D. and Shah, J., 1985. Boundary Detection by Minimizing Functionals. In : IEEE Conference on Computer Vision and Pattern Recognition, San Fransisco, CA, USA, pp. 137–154.

[Nienhüser et al., 2009] Nienhüser, D., Gumpp, T. and Zöllner, J. M., 2009. A Situation Context Aware Dempster-Shafer Fusion of Digital Maps and a Road Sign Recognition System. In : IEEE Intelligent Vehicles Symposium, Xi'an, Shaanxi, China, pp. 1401–1406.

[Nienhüser et al., 2010] Nienhüser, D., Gumpp, T., Zöllner, J. M. and Natroshvili, K., 2010. Fast and Reliable Recognition of Supplementary Traffic Signs. In : IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, pp. 896–901.

[Paclik et al., 2006] Paclik, P., Novovicova, J. and Duin, R. P. W., 2006. Building Road Sign Classifiers using a Trainable Similarity Measure. Transactions on Intelligent Transportation Systems 7(3), pp. 309–321.

[Piccioli et al., 1996] Piccioli, G., de Micheli, E., Parodi, P. and Campani, M., 1996. Robust Method for Road Sign Detection and Recognition. Image and Vision Computing 14(3), pp. 209–223.

[Puthon et al., 2010] Puthon, A.-S., Nashashibi, F. and Bradai, B., 2010. Improvement of Multisensor Fusion in Speed Limit Determination by Quantifying Navigation Reliability. In : IEEE International Conference on Intelligent Transportation Systems, Funchal, Portugal, pp. 855–860.

[Quinlan, 1986] Quinlan, J. R., 1986. Induction of Decision Trees. Machine Learning 1(1), pp. 81–106.

[Revol and Jourlin, 1997] Revol, C. and Jourlin, M., 1997. A New Minimum Variance Region Growing Algorithm for Image Segmentation. Pattern Recognition Letters 18(3), pp. 249–258.

[Rombaut, 1998] Rombaut, M., 1998. Decision in Multi-Obstacle Matching Process using Dempster-Shafer's Theory. Advances in Vehicle Control and Safety pp. 63–68.

[Rosenblatt, 1959] Rosenblatt, F., 1959. Principles on Neurodynamics. Spartan Books.

[Roth and Winter, 2008] Roth, P. M. and Winter, M., 2008. Survey of Appearance-based Methods for Object Recognition. Technical report.

[Rousson and Paragios, 2002] Rousson, M. and Paragios, N., 2002. Shape Priors for Level Set Representations. In : European Conference on Computer Vision, Vol. 2351, Copenhagen, Denmark, pp. 78–92.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., 1986. Learning Representations by Back-Propagating Errors. Nature 323(6088), pp. 533–536.

[Ruta et al., 2010] Ruta, A., Li, Y. and Liu, X., 2010. Real-Time Traffic Sign Recognition from Video by Class-Specific Discriminative Features. Pattern Recognition 43(1), pp. 416–430.

[Sekiguchi et al., 1994] Sekiguchi, H., Sano, K. and Yokoyama, T., 1994. Interactive 3-Dimensional Segmentation Method based on Region Growing Method. Systems and Computers in Japan 25(1), pp. 88–97.

[Sentz and Ferson, 2002] Sentz, K. and Ferson, S., 2002. Combination of Evidence in Dempster-Shafer Theory. Technical report.

[Sethian, 1997] Sethian, J. A., 1997. Level Set Methods : An Act of Violence.

[Shafer, 1976] Shafer, G., 1976. A Mathematical Theory of Evidence. Princeton University Press.

[Steinberg and Bowman, 2004] Steinberg, A. N. and Bowman, C. L., 2004. Rethinking the JDL Data Fusion Levels. In : MSS National Symposium on Sensor and Data Fusion, SENSIAC, Columbia, SC, USA.

[Steinberg et al., 1998] Steinberg, A. N., Bowman, C. L. and White, F. E., 1998. Revisions to the JDL Data Fusion Model. In : Society of Photo-Optical Instrumentation Engineers - Sensor Fusion : Architectures, Algorithms, and Applications III, SPIE, Orlando, FL, USA, pp. 430–441.

[Teague, 1980] Teague, M. R., 1980. Image Analysis via the General Theory of Moments. Journal of Optical Society of America 70(8), pp. 920–930.

[Tuytelaars and Mikolajczyk, 2007] Tuytelaars, T. and Mikolajczyk, K., 2007. Local Invariant Feature Detectors : A Survey. Foundations and Trends in Computer Graphics and Vision 3(3), pp. 177–280.

[Vincent, 1993] Vincent, L., 1993. Morphological Grayscale Reconstruction in Image Analysis : Applications and Efficient Algorithms. Transactions on Image Processing 2(2), pp. 176–201.

[Viola and Jones, 2001] Viola, P. and Jones, M., 2001. Robust Real-Time Object Detection. In : IEEE Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing and Sampling, Vancouver, BC, Canada, pp. 137–154.

[Vlassenroot et al., 2007] Vlassenroot, S., Broekx, S., De Mol, J., Int Panis, L., Brijs, T. and Wets, G., 2007. Driving with Intelligent Speed Adaptation : Final Results of the Belgian ISA-Trial. Transportation Research 41(3), pp. 267–279.

[Wang et al., 2001] Wang, F., Quiniou, R., Carrault, G. and Cordier, M.-O., 2001. Learning Structural Knowledge from the ECG. In : Medical Data Analysis, Madrid, Spain, pp. 288–294.

[White, 1988] White, F. E., 1988. A Model for Data Fusion. In : National Symposium on Sensor Fusion, pp. 149–158.

[Wikipedia, 2008] Wikipedia, 2008. Support Vector Machines.

[Wu et al., 2004] Wu, T.-F., Lin, C.-J. and Weng, R. C., 2004. Probability Estimates for Multiclass Classification by Pairwise Coupling. Machine Learning Research 5, pp. 975–1005.

[Zadeh, 1965] Zadeh, L. A., 1965. Fuzzy Sets. Information and Control 8(3), pp. 338–353.

[Zaklouta et al., 2011] Zaklouta, F., Stanciulescu, B. and Hamdoun, O., 2011. Traffic Sign Classification using K-d Trees and Random Forests. In : IEEE International Joint Conference on Neural Networks, San Jose, CA, USA, pp. 2151–2155.

[Zhang et al., 2003] Zhang, Z., Chen, C., Sun, J. and Luk Chan, K., 2003. EM Algorithms for Gaussian Mixtures with Split-and-Merge Operation. Pattern Recognition 36(9), pp. 1973–1983.

[Zucker, 1976] Zucker, S. W., 1976. Region Growing : Childhood and Adolescence. Computer Graphics and Image Processing 5(3), pp. 382–399.

Détermination de vitesse limite par fusion de données vision et cartographiques temps-réel embarquées

Résumé : Les systèmes d'aide à la conduite sont de plus en plus présents dans nos véhicules et nous garantissent un meilleur confort et plus de sécurité. Dans cette thèse, nous nous sommes particulièrement intéressés aux systèmes d'adaptation automatique de la vitesse limite. Nous avons proposé une approche alliant vision et navigation pour gérer de façon optimale l'environnement routier.

Panneaux, panonceaux et marquages sont autant d'informations visuelles utiles au conducteur pour connaître les limitations temporaires en vigueur sur la route. La reconnaissance des premiers ont fait l'objet ces dernières années d'un grand nombre d'études et sont même commercialisés, contrairement aux seconds. Nous avons donc proposé un module de détection et classification de panonceaux sur des images à niveaux de gris. Un algorithme de reconstruction morphologique associé à une croissance de régions nous ont permis de concentrer la segmentation sur les zones fortement contrastées de l'image entourées d'un ensemble de pixels d'intensité similaire. Les rectangles ainsi détectés ont ensuite fait l'objet d'une classification au moyen de descripteurs globaux de type PHOG et d'une structure hiérarchique de SVMs. Afin d'éliminer en dernier lieu les panonceaux ne s'appliquant pas à la voie sur laquelle circule le véhicule, nous avons pris en compte les informations de marquages à l'aide d'une machine d'états.

Après avoir élaboré un module de vision intégrant au mieux toutes les informations disponibles, nous avons amélioré le système de navigation. Son objectif est d'extraire d'une base de données embarquée, le contexte de conduite lié à la position du véhicule. Ville ou non, classe fonctionnelle et type de la route, vitesse limite sont extraits et modélisés sous forme d'attributs. La fiabilité du capteur est ensuite calculée en fonction du nombre de satellites visibles et de la qualité de numérisation du réseau. La confiance en chaque vitesse limite sera alors fonction de ces deux ensembles.

La fusion des deux sources au moyen de Demspter-Shafer a conduit à de très bonnes performances sur nos bases de données et démontré l'intérêt de tous ces éléments.

Mots clés : fusion de donnée ; Dempster-Shafer ; croissance de régions ; SSR (Sub-Sign Recognition) ; critères de navigation

Speed limit determination by real-time embedded visual and cartograhical data fusion

Abstract: ADAS (Advanced Driver Assistance Systems) are more and more embedded in vehicles and aim at improving drivers' comfort and safety. In this thesis, we focus on ISA (Intelligent Speed Adaptation) and propose an approach combining vision and navigation to optimally manage the environment.

Traffic signs, sub-signs and markings are visual cues helping the driver to know about the current temporary speed limit. Numerous studies were already dedicated to the recognition of the former leading to commercialized products. We thus developed an approach for the detection and recognition of sub-signs using greyscale image processing. A morphological reconstruction filter is first used to select highly contrasted regions followed by a growing region algorithm. The classification is then performed with global features like PHOG and a SVM-BTA (SVM with Binary Tree Architecture). As all the encountered traffic signs are not applicable to the current lane, we integrate marking information.

After the vision module, we improve the navigation one by extracting from the database the driving context related to the vehicle position and the reliability of the sensor. Data fusion is finally performed with the discounting scheme and the normalized conjunctive rule. Very good performances are achieved for several hours of driving and show the importance of each element.

Keywords: data fusion; Dempster-Shafer; growing region; SSR (Sub-Sign Recognition); navigation criteria

MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY