



**HAL**  
open science

# Assistance à la navigation d'aéronefs dans les zones aéroportuaires par vision multi-spectrale

Claire Meymandi-Nejad

► **To cite this version:**

Claire Meymandi-Nejad. Assistance à la navigation d'aéronefs dans les zones aéroportuaires par vision multi-spectrale. Robotique [cs.RO]. INSA Toulouse, 2021. Français. NNT: . tel-03400886v1

**HAL Id: tel-03400886**

**<https://laas.hal.science/tel-03400886v1>**

Submitted on 25 Oct 2021 (v1), last revised 4 Apr 2022 (v2)

**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.



# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

*l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*

---

---

Présentée et soutenue le *06/07/2021* par :

**CLAIRE MEYMANDI-NEJAD**

**Assistance à la navigation d'aéronefs dans les zones  
aéroportuaires par vision multi-spectrale**

---

---

### JURY

ROMUALD AUFRÈRE	Maître de Conférences	Rapporteur
OUIDDAD LABBANI-IGBIDA	Professeur des Universités	Rapporteur
YASSINE RUICHEK	Professeur des Universités	Examineur
DAMIEN VIVET	Ingénieur Chercheur	Examineur
MICHEL DEVY	Directeur de Recherche Émérite	Directeur de thèse
ARIANE HERBULOT	Maître de Conférences	Directeur de thèse

---

**École doctorale et spécialité :**

*EDSYS : Robotique 4200046*

**Unité de Recherche :**

*LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes (UPR 8001)*

**Directeur(s) de Thèse :**

*Michel DEVY et Ariane HERBULOT*

**Rapporteurs :**

*Romuald AUFRÈRE et Ouiddad LABBANI-IGBIDA*



## Remerciements

Je tiens tout d'abord à remercier Michel Devy et Ariane Herbulot, mes directeurs de thèse, qui m'ont permis de mener à bien cette thèse, par leur présence, leur encadrement et leur accompagnement. Je les remercie de la confiance qu'ils m'ont accordée pour développer mes travaux.

Je remercie également Salwa El Kaddaoui, Housseem-Eddine Benseddik et Esteban Perrotin avec qui j'ai eu la chance de travailler et partager des séances de réflexions au cours de ma thèse. Merci à Julien Miranda, avec qui j'ai partagé conférences et bureau, pour son aide et ses points de vue techniques appréciés.

Merci à Ouiddad Labbani-Igbida et Romuald Aufrère d'avoir accepté de prendre de leur temps et d'être rapporteur.e.s et membres de jury de soutenance de ma thèse. Leurs remarques judicieuses m'ont apporté le recul nécessaire pour finaliser ma thèse. Je remercie aussi Damien Vivet et Yassine Ruichek, président du jury, pour avoir accepté de faire partie du jury de soutenance en temps qu'examineurs.

Je souhaite remercier l'équipe pédagogique du département Génie Électrique et Informatique de l'INSA de Toulouse avec qui j'ai eu l'occasion de travailler en tant que vacataire et qui m'ont donné par la suite l'envie de candidater pour le poste d'ATER. Je remercie en particulier Thierry Monteil, Claude Baron, Élodie Chanthery et Pierre-Emmanuel Hladik pour leur soutien et leurs lettres de recommandations qui ont, de mon point de vue, joué pour l'obtention du poste d'ATER.

Merci à toutes les personnes que j'ai côtoyé lors de mon séjour au LAAS, que ce soient les doctorantes, post-docs, stagiaires ou ingénieures, qu'elles appartiennent aux équipes RAP, RIS, DISCO ou Gepetto du LAAS, pour m'avoir permis de garder le sourire pendant ces années de thèse. Merci à elles d'avoir su me remonter le moral quand il le fallait, me pousser à donner le meilleur de moi-même, me donner le courage de mener à bien cette thèse. Merci à elles pour les moments de détente que nous avons passés ensemble, les nombreux gâteaux partagés, les parties de cartes jouées et pour l'amitié qu'elles m'ont offerte. Merci à Aurélie, Diane et Dinesh pour m'avoir accompagnée lors de nombreuses séances de sport afin de faire retomber la pression. Merci à Martin pour ces trajets en vélo matin et soir au bord du canal. Bien sûr, un merci tout particulier aux membres du bureau B-181 et affiliés pour m'avoir accueillie et prêté une place lorsque Julien ne partageait pas mon bureau.

Merci à tous mes amis de m'avoir aidée et accompagnée durant cette thèse. Merci à mes parents, à toute ma famille pour leur soutien, leur confiance et leur présence lors de la soutenance de thèse. Enfin, merci à Thomas, mon compagnon, sans qui je ne serais pas arrivée jusqu'ici. Merci à lui pour ses relectures jusqu'au petit matin et son aide massive en cette fin de thèse.



# Table des matières

Liste des figures	v
Liste des tableaux	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Problématiques . . . . .	2
1.2 Objectifs et contributions . . . . .	7
1.3 Plan du manuscrit . . . . .	8
1.4 Publications . . . . .	9
<b>2 Contexte général</b>	<b>11</b>
2.1 Présentation des projets . . . . .	12
2.2 Normes et recommandations aéronautiques . . . . .	16
2.3 État de l'art général sur la vision embarquée . . . . .	18
2.4 Solution proposée au cours de cette thèse . . . . .	20
2.5 Conclusion . . . . .	30
<b>3 Pré-traitements des données</b>	<b>31</b>
3.1 Introduction . . . . .	32
3.2 Traitement des images infra-rouges . . . . .	34
3.3 Fusion de points de vues multi-caméras . . . . .	39
3.4 Pré-traitements colorimétriques . . . . .	51
3.5 Conclusion . . . . .	93
<b>4 Détection de motifs dans les zones aéroportuaires</b>	<b>95</b>
4.1 Introduction . . . . .	96
4.2 État de l'art sur la détection de marquages . . . . .	98
4.3 Détection par filtrage particulière . . . . .	103
4.4 Amélioration du filtre particulière . . . . .	114
4.5 Comparaison des performances . . . . .	118
4.6 Conclusion . . . . .	132
<b>5 Détection d'obstacles dans les zones aéroportuaires</b>	<b>133</b>
5.1 Introduction . . . . .	134
5.2 Techniques de base . . . . .	136
5.3 État de l'art sur la détection d'obstacles . . . . .	138
5.4 Estimation de mouvement . . . . .	147
5.5 Méthode géométrique - Carte d'obstacles . . . . .	156
5.6 IPM multi-plans . . . . .	164

5.7	Conclusion . . . . .	170
<b>6</b>	<b>Adaptation aux conditions de vision dégradées par le brouillard</b>	<b>171</b>
6.1	Introduction . . . . .	172
6.2	État de l'art des méthodes de débrumage . . . . .	174
6.3	Application d'algorithmes de débrumage au contexte aéronautique . .	180
6.4	Prise en compte de l'infrarouge . . . . .	185
6.5	Proposition d'amélioration de méthodes . . . . .	194
6.6	Conclusion . . . . .	196
<b>7</b>	<b>Conclusion</b>	<b>197</b>
7.1	Synthèse . . . . .	197
7.2	Perspectives . . . . .	199
<b>A</b>	<b>Représentation des résultats des différences de couleur pour les couples (<i>EspaceCouleur</i>, <i>FonctionDistance</i>)</b>	<b>217</b>
	<b>Résumé - Abstract</b>	<b>225</b>

# Liste des figures

1.1	Exemple de trafic dense sur un aéroport. . . . .	4
1.2	Principe d'un système EVS, combiné avec un affichage HUD. . . . .	5
1.3	À gauche, un boîtier EVS, contenant des caméras VIS, SWIR (1,1 à $2,5\mu\text{m}$ ) et LWIR (8 à $14\mu\text{m}$ ). À droite, le boîtier monté sous le nez d'un aéronef (extrait de [71]). . . . .	5
1.4	Images acquises par le boîtier EVS (extrait de [71]). . . . .	5
1.5	Affichage dans le cockpit pendant la phase de taxiage : à gauche le dispositif eTACS, et à droite le dispositif OANS (copyright AIRBUS). . . . .	6
2.1	Intégration matérielle réalisé dans le cadre du projet SART : à gauche la campagne de mesures réalisée sur l'aéroport de Blagnac avec les capteurs montés sur un camion et à droite le boîtier des caméras VIS/LWIR (long wavelength infrared) et kits de développement sur FPGAs. . . . .	13
2.2	Représentation de l'espace de couleur XYZ avec le triangle sRGB (en bleu), les bornes du jaune de référence données par la norme OACI (en noir) et la correction empirique (en rouge). . . . .	17
2.3	Le spectre électromagnétique (extrait de [91]). . . . .	23
2.4	Exemple de combinaison de technologies (caméras, radars, ultrasons) pour l'aide à la conduite d'un véhicule (Mercedes Classe S <sup>2</sup> ). . . . .	24
2.5	Positionnement de la caméra visible (en bleu) et infra-rouge (en rouge) dans le boîtier d'acquisition. . . . .	26
2.6	Représentation du repère caméra (en rouge) et du repère avion (en bleu). . . . .	27
2.7	Briques fonctionnelles de la solution générale proposée pour répondre à la problématique de l'aide à la navigation d'aéronef. . . . .	27
2.8	Étapes détaillées de la brique algorithmique "Analyse de scène" pour une image à un temps $t$ donné. . . . .	28
2.9	Projection sur le plan du sol (extrait de [90]). . . . .	29
3.1	Exemple de deux images infrarouges converties par une normalisation d'histogramme globale (à gauche) et converties grâce au CLAHE (à droite). . . . .	36
3.2	(a) Image infrarouge après traitement par l'algorithme de CLAHE et (b) image visible. . . . .	38
3.3	Homographies liant les images réelles, les vues construites (BEV et vue fusionnée) et la carte du sol. . . . .	43
3.4	Représentation du repère caméra (en rouge/vert/bleu pour $(X_C/Y_C/Z_C)$ ) et du repère de la carte du sol (en rouge pour $(X,Y,Z)$ ). . . . .	44

3.5	Création de la carte du sol par transformation IPM depuis l'image originelle (VIS ou IR). . . . .	45
3.6	(a) Image originelle en RGB, (b) Résultat de la transformation IPM, appelé carte du sol. . . . .	46
3.7	Représentation des différentes images obtenues grâce à la multiplicité des caméras embarquées sur un avion. . . . .	47
3.8	Représentation des repères caméras (en rouge/bleu/vert) et du repère de la carte du sol (en rouge et indicé $w$ ). . . . .	48
3.9	Carte de contours basée Canny. . . . .	53
3.10	Carte de contours basée modèle. . . . .	54
3.11	Carte de couleur pour les marquages au sol. . . . .	55
3.12	Carte de couleur pour les balises. . . . .	55
3.13	(a) Carte d'observation pour les marquages, (b) et carte binarisée associée. . . . .	56
3.14	(a) Carte du sol et (b) carte d'observation binarisée de la délimitation tarmac/herbe. . . . .	57
3.15	Représentation du diagramme de chromaticité de l'espace XYZ [39]. . . . .	65
3.16	Représentation de l'espace de couleur (a) HSV et (b) HSL [173]. . . . .	67
3.17	Représentation de l'espace de couleur (a) YIQ [174] et (b) YUV [175]. . . . .	68
3.18	Exemple de création de la base de données depuis des images réelles et simulées. Chaque masque de couleur correspond à une classe, avec la couleur marron pour les pixels non labellisés, la couleur magenta pour les pixels de la classe 'Non Défini' et la couleur orange pour la classe 'Herbe'. Un exemple est donné pour (a) une image acquise en conditions réelles avec (b) le masque associé et pour (c) une image issue du simulateur avec (d) le masque associé. . . . .	71
3.19	Classes 'Sombre', 'Clair', 'Non Défini', 'Gris', 'Rouge', 'Bleu', 'Vert', 'Herbe' et 'Jaune' de la base de données. En dessous, on retrouve les pourcentages de représentativité des classes sous la forme : représentativité dans la base de données - représentativité dans les images. . . . .	71
3.20	Exemple de répartition des différences de couleur d'un couple ( <i>EspaceCouleur</i> , <i>FonctionDistance</i> ) pour la classe 'Jaune' (en jaune) par rapport à l'ensemble des différences de couleur (en gris). . . . .	72
3.21	Représentation des différences de couleur d'un couple ( <i>EspaceCouleur</i> , <i>FonctionDistance</i> ) pour la classe 'Jaune' (en rouge) par rapport à l'ensemble des différences de couleur (en bleu) pour (a) $\Delta E^*_{CMC}$ , (b) $\Delta E_H$ , (c) $\Delta C$ et (d) $\Delta E_{XYZ}$ . . . . .	73
3.22	Boîtes à moustaches représentant la distribution des différences de couleur par classe pour quelques couples ( <i>EspaceCouleur</i> , <i>FonctionDistance</i> ). . . . .	75
3.23	Résultats du masquage pour une image réelle (a-b) et pour une image simulée (c-d). . . . .	78

3.24	Cartes binaires (gauche) et cartes de distance (droite) pour deux images réelles provenant du cockpit (a) et de la dérive (b) d'un avion. Première ligne : $\Delta E^*_{ab}$ , $\Delta E_{AB}$ , $\Delta E_B$ , $\Delta E^*_{00}$ . Seconde ligne : $\Delta E_{LCH}$ , $\Delta C_{LCH}$ , $\Delta E_{CH}$ , $\Delta E^*_{CMC}$ , vérité terrain. Troisième ligne : $\Delta E_{CbCr}$ , $\Delta E_{Cb}$ , $\Delta E_{IQ}$ , $\Delta E_H$ (HSV/HSL). Cet ordre est conservé pour les quatre blocs. . . . .	79
3.25	Impact du SNR sur le $F_1$ score pour différentes paires candidates. Les lignes pleines représentent les résultats obtenus pour des valeurs seuil idéales tandis que les lignes en pointillés correspondent à ceux obtenus en utilisant la valeur seuil estimée par l'étude colorimétrique.	81
3.26	Pertinence des attributs fournie par la méthode LAMDA+MEMBAS pour une classification (a) à deux classes et (b) à neuf classes. . . . .	88
3.27	Représentation des arbres de profondeur moyenne pour une classification (a) à deux classes et (b) à neuf classes. Le premier nœud explique la façon dont est créé l'arbre. . . . .	90
3.28	(a) Image provenant d'une caméra placée dans la dérive d'un avion et (b) labellisation manuelle. (c) Labellisation par un arbre de décision. (d) Labellisation par logique floue. . . . .	91
3.29	Schéma en cascade de classifieurs utilisés pour la labellisation semi-automatique d'images réelles. Le premier classifieur ( $C_1$ ) associe une classe à chaque pixel et le second classifieur ( $C_2$ ) détecte une ligne suivant le motif de couleur fourni en entrée. . . . .	92
4.1	Exemple de motifs d'intérêt des zones aéroportuaires : motifs de marquages au sol complexes, balises de bord de pistes, lignes doubles, panneaux... . . . . .	96
4.2	Présentation d'images d'entrée du filtre particulière : (a) image $IM$ originelle et (b) carte du sol, résultat de la transformation IPM. . . . .	103
4.3	Principe du filtre particulière. . . . .	105
4.4	Exemple de carte binarisée. Les segments initiaux servant à l'initialisation d'un cluster de particules sont indiqués par les flèches bleues et les lignes rouges indiquent les hauteurs de la carte dans lesquelles on détecte de nouveaux départs de clusters. . . . .	107
4.5	Politique de ré-échantillonnage pour 5 particules. . . . .	107
4.6	Sélection des futures particules. . . . .	108
4.7	Cas de la détection de ligne : (a) Fusion de groupement de particules, (b) Détection de nouveaux débuts de groupements de particules. . . . .	109
4.8	Groupements de particules à fusionner en bleu et en vert, occultation simulée en rouge. . . . .	111
4.9	(a) Positions de particules de différents groupements après passage du filtre particulière sur l'intégralité de l'image, rapporté à l'image originelle, (b) Combinaison de l'estimation des marquages (vert) et des balises (rouge). . . . .	112

4.10	(a) Détection de balises, (b) Positions de particules de différents groupements après passage du filtre particulaire sur l'intégralité de la carte du sol, (c) Positions de particules de différents groupements après passage du filtre particulaire sur l'intégralité de la carte du sol transposée, (d) Détection de marquages rapportée à l'image originelle, (e) Détection de marquages par la carte du sol transposée, rapportée à l'image originelle. . . . .	113
4.11	Exemple de zones aéroportuaires. . . . .	114
4.12	Résultats d'une propagation du haut vers le bas de l'image pour les méthodes de propagation (a) initiale, (b) compte tenu de l'état précédent et (c) à pas variable. . . . .	115
4.13	Schématisation de motifs observables par un filtre particulaire fenêtré pour cinq particules avec une rotation fixée comme nulle de la fenêtre. . . . .	117
4.14	Définition des trois méthodes utilisées. . . . .	118
4.15	Performance des trois méthodes sur l'image globale. . . . .	122
4.16	Impact de la transformation IPM sur les performances d'images provenant de caméras placées (a) dans la dérive et (b) à l'avant de l'avion. . . . .	124
4.17	Performances des trois méthodes en fonction de la portée de détection. . . . .	125
4.18	Impact des conditions météorologiques dégradées sur (a) les images de la caméra placée dans la dérive et (b) les images de la caméra placée à l'avant de l'avion. . . . .	126
4.19	(a) Image en conditions météorologiques dégradées et zoom (zone jaune) sur les résultats (bleu pour la transformée de Hough, rouge pour le filtre particulaire et vert pour LaneNet) par (b) temps clair et (c) conditions météorologiques dégradées. . . . .	127
4.20	(a) Image réelle d'une caméra placée à l'avant de l'avion. Vérité terrain et résultats (b) pour la transformée de Hough, (c) pour le filtre particulaire et (d) pour LaneNet. . . . .	128
4.21	(a) Image réelle d'une caméra placée dans la dérive de l'avion. Vérité terrain et résultats (b) pour la transformée de Hough, (c) pour le filtre particulaire et (d) pour LaneNet. . . . .	129
4.22	Résultats sans puis avec IPM d'une image simulée provenant d'une caméra placée à l'avant de l'avion (respectivement (a) et (b)) pour la transformée de Hough, (respectivement (c) et (d)) pour le filtre particulaire et (respectivement (e) et (f)) pour LaneNet. . . . .	130
5.1	Schéma des interactions entre délimitations du tarmac (frontière vert-gris), balises (en bleu) et marquages au sol (en jaune). Les éléments peuvent être utilisés seuls pour former un motif ou en combinaison (par exemple marquage + liste de balises et leur position en fonction du marquage) pour créer des motifs plus complexes qui seront appariés pour estimer la matrice de mouvement. . . . .	147
5.2	Représentation des classes utilisées pour définir les différents motifs d'intérêt utilisés pour l'estimation de mouvements. . . . .	150
5.3	Représentation d'une matrice de coût (calcul de similitude) entre des motifs de la carte du sol à l'instant $t - 1$ (en bleu) et des motifs de la carte du sol à l'instant $t$ (en rouge). . . . .	150

5.4	Création des cartes du sol $IPM$ ainsi que de la carte d'obstacles $Obst$ depuis les images originelles $IM$ . . . . .	157
5.5	Représentation des erreurs induites par la transformation IPM lors de la projection sur le plan du sol pour (a) un prisme droit et (b) un obstacle pour lequel l'avant se trouve au dessus du sol. . . . .	159
5.6	(a) $IM_{t-1}$ l'image originelle en niveaux de gris à l'instant $t - 1$ , (b) $IM_t$ l'image originelle en niveaux de gris à l'instant $t$ , (c) $IPM_{t-1}$ la carte du sol à l'instant $t - 1$ , (d) $IPM_t$ la carte du sol à l'instant $t$ , (e) $IPM_{(t-1)/t}$ la carte du sol à l'instant $t - 1$ exprimée dans le repère de la carte du sol à l'instant $t$ , (f) $IM_{(t-1)/t}$ l'image reconstruite après projection inverse de $IPM_{(t-1)/t}$ , (g) $Obst$ la carte d'obstacles corrigée exprimée dans le repère du sol et (h) $IM_{Obst}$ l'image non corrigée de la présence d'obstacles sans passage par la transformation IPM. . . . .	160
5.7	(a) Règles de colorisation en fonction de la distance à l'aéronef : vert pour acceptable, orange pour danger potentiel et rouge pour critique, (b)(d) Colorisation des objets détectés comme obstacles potentiels dans la carte d'obstacles $IPM_{Obst}$ en fonction de leur proximité à l'aéronef, (c)(e) Application de la colorisation des obstacles potentiels dans l'image d'origine augmentée $IM_{Obst}$ . . . . .	162
5.8	(a-c-f) Colorisation de plusieurs cartes d'obstacles et (b-d-f) résultats de colorisation de l'image pour plusieurs estimations de déplacement de l'aéronef entre les instants $t - 1$ et $t$ . . . . .	163
5.9	Illustration de la méthode multi-plans (inspirée de [90]). . . . .	164
5.10	(a) Image à l'instant $t$ . Exemple de deux cartes d'obstacles : (b) sur le plan du sol et (c) sur un plan virtuel situé à 25cm du sol. (d) et (e) sont des zooms de (b) et (c). . . . .	167
5.11	Zoom sur les cartes d'obstacle résultant de la fusion d'information de six cartes dont l'espacement global est (a-b) de 25cm (a) en se basant sur l'intersection et (b) en se basant sur la moyenne des cartes, (c-d) de 250cm (c) en se basant sur l'intersection et (d) en se basant sur la moyenne des cartes. On présente le résultats de coloration (e) de la carte d'obstacles dont on a tiré (a) et (f) la projection sur l'image. . . . .	169
6.1	(a) Impact modéré du brouillard (image provenant de [46]), (b) Faible impact du brouillard, (c) Impact simulé du brouillard (image provenant du dataset FROSI <sup>1</sup> ) et (d) Fort impact du brouillard (image provenant du dataset Dense Haze du challenge NTIRE19 <sup>2</sup> ). . . . .	172
6.2	Illustration du modèle de Koschmieder (extrait de [16]). . . . .	175
6.3	Distance maximale de visibilité fixée par le simulateur à (a) 100m, (b) 200m, (c) 300m et (d) 400m. . . . .	180
6.4	(a) Référence, (b) Image avec brouillard, (c) Image après CLAHE, (d) Image après Fattal, (e) Image après Aod-Net et (f) Image après Cycle-Gan. . . . .	182
6.5	Exemples d'images associées simulées (a) en infrarouge et traitée par CLAHE et (b) dans le domaine du visible. . . . .	188
6.6	Principe de la transformée en ondelettes discrète. . . . .	189
6.7	Principe du transfert de couleur. Application de la méthode proposée dans l'article [162]. . . . .	190

6.8	Résultat de la fusion des images dans le domaine du visible et infrarouge, présentées ci-dessus, par (a) la transformée en ondelettes discrète et (b) le transfert de couleur. . . . .	190
6.9	(a) Exemple de transfert de couleur 4 classes, (b) Exemple de transfert de couleur 10 classes, (c) Répartition des 4 classes et (d) Répartition des 10 classes. . . . .	191
6.10	(a) Image impactée par le brouillard et (b) Image en sortie de l'algorithme RetineX. . . . .	195
A.1	Schéma explicatif des Figures de l'Annexe A. Répartition des résultats de la différence de couleur entre les couleurs d'une classe et la couleur de référence. La totalité des différences de couleur est notée en bleu et la répartition pour une classe donnée est notée en rouge. . . . .	217
A.2	(a) Distance euclidienne et (b) Distance euclidienne pondérée sur les canaux R, G et B. . . . .	218
A.3	Distance euclidienne sur (a) les trois canaux H, S et V, (b) les canaux H et S, (c) le canal H, (d) le canal S. . . . .	218
A.4	Distance euclidienne sur (a) les trois canaux H, S et L, (b) les canaux H et S, (c) le canal H, (d) le canal S. . . . .	219
A.5	Distance euclidienne sur (a) les trois canaux Y, Cb et Cr, (b) les canaux Cb et Cr, (c) le canal Cb, (d) le canal Cr. . . . .	220
A.6	Distance euclidienne sur (a) les trois canaux Y, I et Q, (b) les canaux I et Q, (c) le canal I, (d) le canal Q. . . . .	221
A.7	Distance euclidienne sur (a) les trois canaux de L, $a^*$ et $b^*$ , (b) les canaux $a^*$ et $b^*$ , (c) sur le canal $a^*$ , (d) sur le canal $b^*$ . (e) Distance CIE94 sur les canaux L, $a^*$ et $b^*$ et (f) Distance CIEDE2000 sur les canaux L, $a^*$ et $b^*$ . . . . .	222
A.8	Distance euclidienne sur (a) les trois canaux de L, C et H, (b) les canaux C et H, (c) sur le canal C, (d) le canal H. (e) Distance CIE94 sur les canaux L, $a^*$ et $b^*$ et (f) Distance CIEDE2000 sur les canaux L, $a^*$ et $b^*$ . . . . .	223
A.9	Distance euclidienne sur (a) les trois canaux X, Y et Z, (b) les canaux X et Y, (c) sur les canaux X et Z, (d) les canaux Y et Z, (e) le canal X, (f) le canal Y, (g) le canal Z. . . . .	224

# Liste des tableaux

3.1	Liste des espaces de couleur et des fonctions de distance sélectionnées pour l'étude d'adaptation de la couleur de référence à partir d'une image. Les notations couramment utilisées sont indiquées entre parenthèses. . . . .	69
3.2	Tableau comparatif des performances de quelques espaces de couleur et fonctions de distance. . . . .	74
3.3	Tableau comparatif de plusieurs espaces de couleur et fonctions de distance sur les images de tests (non-utilisées pour la base de données). . . . .	79
3.4	Résultats de performances d'algorithmes de détection de lignes avec et sans affinage adaptatif de la couleur de référence. . . . .	83
3.5	Scores fournis par l'algorithme de logique floue pour la classe 'Jaune' et la classe 'Non Jaune'. . . . .	87
3.6	Scores fournis par l'algorithme de logique floue pour les neuf classes de la base de données de couleurs. . . . .	89
3.7	Résultats de précision de la labellisation par les arbres de décisions lors de la phase d'entraînement. . . . .	89
4.1	Tableau de résultat pour des images réelles et simulées sans pré-traitement IPM ou conditions météorologiques dégradées. . . . .	121
4.2	Tableau de résultats des images ayant reçues une transformation IPM en pré-traitement. . . . .	123
4.3	Tableau de résultats pour des conditions météorologiques dégradées. . . . .	125
5.1	Tableau de résultats préliminaires pour l'estimation du déplacement d'avion dans le plan du sol. . . . .	155
6.1	Résultats de l'amélioration de l'image par débrumage. . . . .	181
6.2	Résultats de la détection de lignes. . . . .	184



# Chapitre 1

## Introduction

### Sommaire

---

1.1	Problématiques . . . . .	2
1.2	Objectifs et contributions . . . . .	7
1.3	Plan du manuscrit . . . . .	8
1.4	Publications . . . . .	9

---

La vision par ordinateur était un problème majeur au milieu du 20<sup>ième</sup> siècle. Après plus de 50 ans de recherches dans le domaine, les progrès théoriques ont permis d’obtenir de très bons résultats en pratique. De nos jours, la vision par ordinateur est de plus en plus utilisée pour des applications embarquées notamment en robotique, et dans les transports intelligents, mais aussi dans des applications où le capteur est fixe comme la sécurité (détection de personnes par vidéosurveillance), le contrôle non destructif (détection de défauts sur des pièces dans l’industrie), etc.

Depuis les années 80, d’autres capteurs sont apparus, notamment pour les applications embarquées SONAR, RADAR et LIDAR, et plus récemment les caméras à temps de vol, fournissent des données sensorielles 3D, complémentaires des données d’apparence 2D fournies par les caméras dans diverses bandes spectrales, visibles et infrarouges (proche : NIR, court : SWIR, long : LWIR). De nouvelles approches de détection d’objets ont été proposées pour tirer parti de ces données 3D sur la scène observée, données non disponibles directement en vision.

Avec l’essor des nouvelles technologies, le développement de la puissance de calculs et les ambitions humaines vers plus de progrès, ces dernières années ont vu apparaître de nombreux sujets d’étude sur l’utilisation de la vision par ordinateur dans le but de développer la navigation autonome des moyens de transport, appuyée par cette diversité de capteurs.

Dans la suite de cette introduction sont présentées d’abord les problématiques abordées dans cette thèse et liées à la navigation autonome d’un aéronef au sol puis les objectifs de ces travaux et les contributions auxquelles ils ont conduit et enfin le plan de ce manuscrit.

## 1.1 Problématiques

Cette thèse se place dans le domaine de l’aéronautique, qui est un domaine déjà fortement automatisé. En effet, l’existence de systèmes de pilotage automatique remonte au début du 20<sup>ième</sup> siècle. De nos jours, les phases de vol de l’avion sont supervisées par un système de gestion de vol qui a pour but d’optimiser le trajet de l’avion, secondé par un système de pilotage automatique ainsi que de nombreux algorithmes de gestion d’urgences. Dès 1937, le domaine de l’aviation s’intéresse aux décollages et atterrissages automatiques. Plus récemment, on peut noter la réussite du premier atterrissage automatique assisté par vision (en 2019), réalisé grâce aux recherches de deux universités allemandes et basé notamment sur des informations provenant de caméras infrarouges. Fin 2019 également Airbus réussit le premier décollage entièrement automatique d’un avion de ligne, dans le cadre du projet ATTOL (Autonomous Taxi, Take-Off and Landing)<sup>1</sup> qui vise l’automatisation des phases de roulage, décollage et atterrissage de l’avion. Ce projet de démonstrateur s’est conclu en Juin 2020. AIRBUS poursuit ces travaux sur l’augmentation des capacités autonomes de ses aéronefs via un nouveau projet de démonstrateur appelé DragonFly.

---

<sup>1</sup><https://www.air-cosmos.com/article/conclusion-du-programme-dessais-en-vol-attol-chez-airbus-23304>

Il reste cependant une phase du trajet qui n'est à ce jour pas automatisée, la phase au sol, ou roulage, qui précède le décollage et suit l'atterrissage. Elle-même peut être décomposée en deux phases distinctes pour lesquels les enjeux sont différents : la phase de roulage pour rejoindre ou quitter la piste via des taxiways (on parle aussi de taxiage) et la phase de roulage aux abords des portes d'embarquement. Dans ces deux cas, des marquages au sol indiquent les trajectoires que doivent suivre les aéronefs. Dans le cadre de la navigation aux abords des bâtiments, il convient d'ajouter au suivi de lignes une attention plus importante à la détection d'obstacles potentiels.

L'automatisation de la phase au sol nécessite donc des algorithmes d'analyse de l'environnement, d'estimation de position et de déplacement de l'aéronef ou encore de détection d'obstacles. Ce sont des problématiques que l'on retrouve par exemple dans le domaine de l'automobile.

Plusieurs raisons peuvent être avancées pour expliquer pourquoi l'automatisation du roulage se fait tardivement dans l'aéronautique, alors que les technologies (capteurs, algorithmes) sont développées depuis près de 20 ans pour équiper les véhicules intelligents dans l'automobile :

- La réglementation qui encadre la navigation des aéronefs est très sévère;
- Paradoxalement, vu la différence de taille et les problèmes de structure, il est plus délicat d'intégrer des capteurs sur un aéronef que sur un véhicule;
- Les problèmes liés à la certification limitent l'introduction de nouvelles technologies à grande échelle, aussi bien du matériel (composants embarqués : CPU, architectures dédiées à base de FPGA ou autres) que du logiciel (difficulté de faire certifier des algorithmes de traitement d'images surtout s'ils ne sont pas déterministes ou s'ils font appel à de l'apprentissage à partir de bases de données);
- Enfin, le marché est contrôlé jusqu'à maintenant par seulement deux compétiteurs Boeing et Airbus, seuls à décider des innovations dans les aéronefs qu'ils produisent.

Notons que les pilotes des compagnies aériennes sont demandeurs, depuis longtemps, d'une aide pour le suivi de piste sur les aéroports, en particulier pour le roulage par toutes conditions météorologiques entre la zone de parking et la piste de décollage. Une des raisons essentielles est que 30% des accidents (et aussi, 30% des victimes) en aéronautique surviennent pendant les phases de roulage et décollage. Cette proportion était en augmentation du fait de l'encombrement des zones aéroportuaires (Figure 1.1) et de l'augmentation du trafic aérien (sans prendre en compte la forte perturbation liée à la crise depuis Mars 2020).

Pour un aéronef, les raisons principales de ces accidents sont :

- L'incursion sur piste d'éléments non prévus : présence sur la piste ou le taxiway d'un autre aéronef, d'un véhicule, d'une personne ou d'un objet quelconque (pièce tombée d'un autre avion);
- La collision avec un obstacle : autre aéronef, mais aussi, véhicule au sol, personne, éléments fixes de l'aéroport (pylônes, bâtiments...);



Figure 1.1: Exemple de trafic dense sur un aéroport.

- L'excursion de piste : les roues de l'aéronef sortent du tarmac, typiquement vont dans l'herbe. Même si cela se produit à faible vitesse, un tel événement nécessite une immobilisation pour réaliser une inspection de la structure de l'aéronef.

Les pilotes et les compagnies aériennes sont en attente d'innovations pour éviter ces accidents ou ces pertes d'exploitation. Mais le segment sol est, jusqu'à maintenant, une phase opérationnelle manuelle qui n'a pas été considérée comme une étape clé pour les études technologiques de même niveau que le segment vol ou que l'atterrissage tout temps.

De ce fait, les caméras n'ont été exploitées pour la navigation au sol pour l'aviation commerciale qu'à partir de la fin des années 90, lorsque sont apparues des visualisations à cristaux liquides capables de recevoir des signaux vidéos. Pour offrir une capacité vidéo, il existait auparavant la technologie HUD (Head-Up Display), issue de l'aviation militaire. Cet affichage HUD a été introduit dans l'aviation commerciale, pour développer l'atterrissage tout temps à l'aide de la technologie EVS (Enhanced Vision System).

Le système EVS a été développé à la NASA [71]. Un tel système est couplé à un affichage HUD pour projeter l'image devant le pilote, superposée à la scène réelle (Figure 1.2). Notons que les caméras EVS pourraient aussi être exploitées dans les phases de taxiage, mais leur champ de vision est trop restreint (généralement  $40^\circ$ ) pour fournir de bonnes capacités de détection.

Nous montrons en Figure 1.3 un boîtier EVS et, en Figure 1.4, les images renvoyées par les capteurs présents dans le boîtier.

Bien que, à partir de 2000, les avancées technologiques aient permis une diminution du coût des matériels en vision, l'utilisation dans le domaine commercial des systèmes EVS est restée figée du fait de la réglementation. Ils n'ont pu être exploités que dans des niches d'activités commerciales, comme les avions d'affaires (Dassault, Gulfstream...) et le transport de fret, le but étant d'obtenir des baisses des limites minimales pour pouvoir opérer (ex : distance de visibilité minimale pour pouvoir décoller).

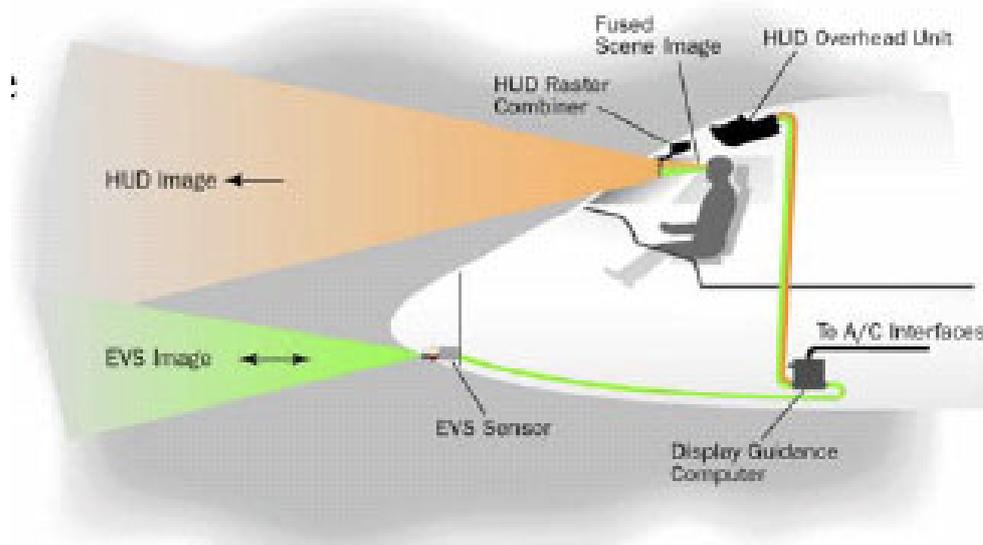


Figure 1.2: Principe d'un système EVS, combiné avec un affichage HUD.

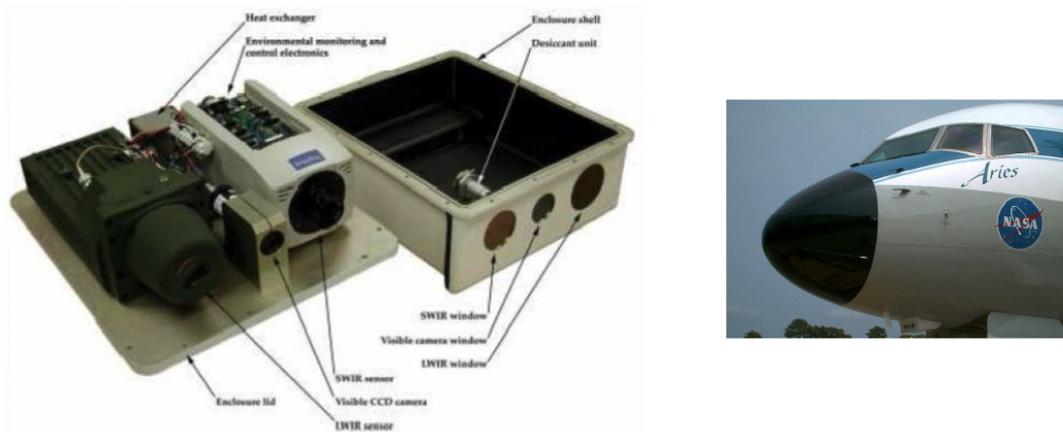


Figure 1.3: À gauche, un boîtier EVS, contenant des caméras VIS, SWIR (1,1 à  $2,5\mu\text{m}$ ) et LWIR (8 à  $14\mu\text{m}$ ). À droite, le boîtier monté sous le nez d'un aéronef (extrait de [71]).



Figure 1.4: Images acquises par le boîtier EVS (extrait de [71]).

Les premiers projets d'aide au roulage ont donc été développés en utilisant des caméras dans le visible à la fin des années 90. Bien que ces systèmes restent inefficaces en condition de météo défavorable (brouillard, pluie intense...), vu les enjeux commerciaux dans la lutte des avionneurs Boeing–Airbus, ils ont connu

un essor rapide. En 1995 Boeing avait lancé le B777-300 avec un système GMCS (Ground Maneuver Camera System) en analogique alors qu'en 1998 Airbus introduit le dispositif eTACS (External and Taxi Aid Camera System) en fibre optique, sur l'A340-500/600. Depuis lors, le retour des compagnies aériennes a confirmé l'attractivité des systèmes vidéo pour la navigation au sol.

En Figure 1.5, nous montrons les affichages produits grâce à deux dispositifs installés sur plusieurs avions d'AIRBUS (notamment tous les A380) :

- Le système eTACS exploite deux caméras dans le visible, une caméra montée dans la dérive (FIN CAM), l'autre montée sous le ventre de l'avion (BELLY CAM), derrière la roulette. Les deux images sont transmises à un système qui, en temps réel, les synchronise et les aligne afin d'afficher la vue composite montrée à gauche de la Figure 1.5. Sont rajoutés aussi dans des positions fixes apprises au préalable, les points et traits bleus, afin d'aider le pilote à situer l'aéronef sur le taxiway.
- Le système OANS (On-Board Airport Navigation System) exploite la description de l'aéroport et la localisation de l'aéronef fournie par fusion GPS/IMU (Unité de Mesure Inertielle), afin d'afficher la position et l'orientation de l'aéronef par rapport aux taxiways et aux marquages.



Figure 1.5: Affichage dans le cockpit pendant la phase de taxiage : à gauche le dispositif eTACS, et à droite le dispositif OANS (copyright AIRBUS).

Cette illustration a été prise sur une vidéo enregistrée sur l'aéroport de Toulouse-Blagnac en 2016, fournie par AIRBUS dans le cadre de projets collaboratifs auxquels le LAAS participe. On peut noter la complexité des marquages dans cette zone.

Le LAAS participe, depuis plus de 10 ans, à des projets collaboratifs (SART, TALVIS, COCOTIER, présentés en Chapitre 2) qui visent à développer un système d'aide au roulage permettant à un aéronef roulant à très faible allure de se mouvoir en conditions tous temps sur les bretelles de piste jusqu'à son point fixe avant son

décollage, ou de rejoindre sa zone de parking après son atterrissage. Ce système doit permettre un gain substantiel en termes de sûreté du transport aérien par l'identification visuelle par tout temps de la piste, des bretelles d'accès, des bâtiments et des zones parking ainsi qu'une amélioration de sécurité en particulier sur les plateformes de correspondance aéroportuaires (hubs) où la circulation des véhicules et des personnes est intense.

Vers 2010, ces projets visaient à développer un système capable d'afficher plus d'informations au pilote, afin de l'aider à mieux comprendre la situation. Ces informations sont produites par l'analyse des images acquises depuis les caméras montées sur l'aéronef. À partir de 2016 environ, la stratégie a changé : le but est d'accroître les capacités autonomes de l'aéronef, donc rendre automatique les fonctions qui ne le sont pas encore (décollage, atterrissage, roulage), cela à terme, pour réduire le nombre de pilotes, passer de deux à un seul pilote dans le poste de pilotage.

Cette thèse rentre dans le cadre de projets collaboratifs, qui visent donc à s'inspirer des méthodes existantes sur les systèmes autonomes en Robotique et en Véhicule Intelligent, pour proposer de nouvelles technologies adaptées au contexte aéronautique ; l'évaluation se fait sur des données simulées ou des données réelles enregistrées au préalable. Nous n'avons pas accès aux données acquises par les projets ATTOL ou DragonFly : ce sont des projets internes à AIRBUS, qui visent à évaluer des technologies permettant d'accroître l'autonomie de l'aéronef, via des démonstrations en vol.

## 1.2 Objectifs et contributions

Cette thèse se place dans ce contexte dynamique de recherche d'automatisation des phases de roulage d'un aéronef dans les zones aéroportuaires. Nous souhaitons apporter une contribution au développement de systèmes basés vision pour le développement de cette technologie dans le domaine de l'aéronautique.

Au cours de cette thèse, nous nous concentrons sur l'amélioration de l'aide au pilotage d'aéronefs. Pour cela, nous nous intéressons aux domaines de la reconnaissance de scène (ou environnement dans lequel gravite l'aéronef) et à la détection d'obstacles présents dans cette scène, comme par exemple d'autres avions, le personnel et les infrastructures aéroportuaires ou encore des objets imprévisibles. Dans un premier temps, l'objectif est de remonter ces informations aux pilotes afin de les aider dans la prise de décisions. À long terme, ces technologies ont pour but de réaliser une phase de roulage entièrement autonome.

Lorsque les conditions de vision sont dégradées (nuit ou mauvaises conditions météorologiques), la tâche de roulage est rendue plus difficile aux pilotes qui, de surcroît, voient leur charge de travail augmenter. Cela peut alors induire une mauvaise compréhension de l'environnement dans lequel se déplace l'avion, une augmentation des dangers que peut rencontrer le pilote et un accroissement de la fatigue pour ce dernier. D'autre part, il est fréquent que des conditions climatiques fortement dégradées retardent les vols voire mènent à leur annulation et paralysent le trafic aérien.

C'est pourquoi nous recherchons une solution d'automatisation qui puisse aider le pilote lorsque ce phénomène apparaît, en améliorant la perception de l'environnement

et en facilitant la prise de décision. Pour cela, nous questionnons la pertinence de l'utilisation de données basées vision provenant de différents systèmes d'acquisitions (solution multi-spectrale) pour améliorer la remontée d'informations au pilote.

Dans cette thèse, nous proposons plusieurs contributions dans le domaine de l'automatisation de la phase de roulage d'un aéronef. Dans un premier temps, nous proposons de travailler sur des pré-traitements des données acquises par des caméras opérant dans le domaine du visible et de l'infrarouge afin de fusionner ces informations. Nous menons aussi une étude sur l'information de couleur, élément clé de notre solution de détection de motifs d'intérêt à remonter au pilote. Dans un second temps, nous présentons un algorithme de détection de motifs, principalement les marquages au sol, appliqué aux zones aéroportuaires. Dans un troisième temps, nous utilisons les motifs détectés pour proposer un algorithme de détection d'obstacles autour de l'aéronef afin de fournir des informations au pilote pour l'aide à la navigation. Pour finir, nous comparons des méthodes de débrumage couplées à l'utilisation des données infrarouges afin d'améliorer la visibilité du pilote lors d'épisodes de brouillard.

### 1.3 Plan du manuscrit

Le Chapitre 2 approfondit le contexte de cette thèse et souligne les spécificités du domaine de l'aéronautique. Y sont présentées les différentes briques de la solution globale proposée pour répondre à la problématique de cette thèse.

Le Chapitre 3 énumère les différents pré-traitements appliqués aux données utilisées au cours de cette thèse. Cette phase est souvent nécessaire en vision par ordinateur et a pour but de mettre en valeur les zones d'intérêt dans les images utilisées.

Le Chapitre 4 détaille la première brique de la solution proposée pour répondre à la problématique de nos travaux : l'analyse de scène. Dans ce chapitre, nous proposons une méthode basée sur un algorithme de filtre particulaire afin d'isoler des motifs d'intérêt dans une image. Cette méthode permet en particulier d'extraire les marquages au sol : elle est comparée à deux autres approches proposées dans la littérature pour résoudre ce problème classique, notamment pour les véhicules intelligents.

Le Chapitre 5 utilise les résultats du Chapitre 4 pour alimenter la deuxième brique de la solution : la détection d'obstacles. Ce chapitre propose une utilisation des motifs d'intérêt dans le but d'estimer le déplacement de l'aéronef et la position de potentiels obstacles dans son environnement.

Dans le Chapitre 6, nous cherchons à étendre l'utilisation de notre solution dans le cadre de conditions climatiques amenant à une vision dégradée comme par exemple la tombée de la nuit ou encore les épisodes de brouillard. Nous proposons d'utiliser la fusion de données de deux capteurs opérant dans des bandes spectrales différentes pour améliorer la récupération d'informations à fournir au pilote.

Ces cinq chapitres sont suivis d'une conclusion qui reprend les différentes contributions apportées dans les chapitres précédents et offre des pistes de recherches pour des travaux futurs.

## 1.4 Publications

Cette thèse a mené aux publications suivantes :

- Claire Meymandi-Nejad, Salwa El Kaddaoui, Michel Devy, Ariane Herbulot. Lane detection and scene interpretation by particle filter in airport areas. *14th International Conference on Computer Vision Theory and Applications (VISAPP 2019)*, Feb 2019, Prague, Czech Republic.
- Claire Meymandi-Nejad, Ariane Herbulot, Michel Devy. Débrumage d'images pour la détection de lignes dans le domaine aéronautique. *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, Jun 2020, Vannes, France.
- Claire Meymandi-Nejad, Esteban Perrotin, Ariane Herbulot, and Michel Devy. Aircraft Navigation on Taxiways: Evaluation of Line Detection Algorithms Proposed for Automotive Applications. In *Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control (ISCSIC)*, Nov 2020.
- Claire Meymandi-Nejad, Esteban Perrotin, Ariane Herbulot, and Michel Devy. Colorimetric space study: Application for line detection on airport areas. *7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, Apr 2021.
- Esteban Perrotin, Claire Meymandi-Nejad, Ariane Herbulot, Michel Devy. Détection des lignes aéroportuaires par méthode de filtrage particulière: Évaluation de fonctions d'observations. *18ème édition d'ORASIS*, Sept. 2021, Saint Ferréol, France.



# Chapitre 2

## Contexte général

### Sommaire

---

<b>2.1</b>	<b>Présentation des projets . . . . .</b>	<b>12</b>
<b>2.2</b>	<b>Normes et recommandations aéronautiques . . . . .</b>	<b>16</b>
2.2.1	Certification des algorithmes . . . . .	16
2.2.2	Couleurs des éléments de la zone aéroportuaire . . . . .	16
<b>2.3</b>	<b>État de l’art général sur la vision embarquée . . . . .</b>	<b>18</b>
<b>2.4</b>	<b>Solution proposée au cours de cette thèse . . . . .</b>	<b>20</b>
2.4.1	Capteurs disponibles pour la perception en Robotique et Transports Intelligents . . . . .	20
2.4.2	Présentation des briques algorithmiques . . . . .	25
<b>2.5</b>	<b>Conclusion . . . . .</b>	<b>30</b>

---

L'assistance à la navigation dans le domaine aéronautique est fortement présente dans les phases de décollage, d'atterrissage et de vol. Il reste cependant à automatiser la phase au sol. C'est dans ce contexte que se placent les travaux de cette thèse.

Dans ce chapitre, nous commencerons par détailler les projets dans lesquels s'inscrivent nos travaux, menés grâce à une collaboration entre des industriels et des laboratoires. Ces projets font suite à des précédents travaux dont nous décrirons les objectifs et les conclusions. Le domaine d'application de cette thèse, l'aéronautique, est empreint de normes et recommandations dont nous préciserons ensuite certains points impactant fortement nos travaux. Nous présenterons un état de l'art général sur la vision embarquée avant de définir la solution technique choisie pour répondre à la problématique posée par cette thèse.

## 2.1 Présentation des projets

Le travail effectué dans cette thèse a été réalisé au sein de l'équipe RAP (Robotique, Action et Perception) du LAAS-CNRS. Il se place dans le cadre des projets TALVIS (Taxi Aid by Low VISibility, 2016-2017) et COCOTIER (COnccept de COckpit et Technologies Intégrées En Rupture, 2019-2022), tous les deux financés par la DGAC.

Le projet TALVIS impliquait deux avionneurs AIRBUS et DASSAULT AVIATION, deux PME fournisseurs de technologies OKTAL SE (simulation) et VODEA (électronique, systèmes embarqués) et deux laboratoires (LAAS-CNRS et ONERA via le DOTA, son département spécialisé en Optique). Il était coordonné par LATECOERE, qui a par ailleurs une coopération internationale avec l'IATM (Institute of Aerospace Technologies at the University of Malta). LATECOERE est un équipementier qui réalise des sous-ensembles pour les avionneurs : notamment il fournit à AIRBUS, le système eTACS présenté en Introduction (Figure 1.5).

Le but du projet TALVIS était de concevoir et développer un système d'assistance au pilotage d'aéronefs dans les zones aéroportuaires, en particulier sur les taxiways : il s'agit de proposer un ADAS (Advanced Driver Assistance System) pour pilote d'avion, dans la phase de navigation au sol. L'objectif était de réaliser un système plus avancé que le système eTACS, avec deux objectifs majeurs : compléter la vision dans le visible par une vision dans l'infrarouge, et intégrer des traitements des images (fusion des images, détection des marquages et des obstacles...) pour afficher plus d'informations au pilote. La validation se fait essentiellement sur des images simulées par les outils développés par OKTAL SE.

Ce projet TALVIS faisait suite au projet FUI SART (Système d'Aide au Roulage Tout temps, 2009-2012), coordonné aussi par LATECOERE, avec les mêmes objectifs que TALVIS, à savoir, développer un ADAS plus performant que le système eTACS existant. Participaient à ce projet déjà l'équipe RAP du LAAS-CNRS, l'entreprise THALES, en charge du système d'affichage à intégrer dans le cockpit de l'aéronef, et la PME CEDIP (racheté en 2008 par l'entreprise US FLIR), qui a développé une caméra LWIR 1M pixels répondant aux besoins du projet. Deux thèses ont été réalisées dans le cadre de ce projet, dédiées l'une au développement d'une architecture parallèle à base de FPGAs, dans laquelle étaient exécutés des traitements bas niveau sur les images (fusion, détection et suivi de points

d'intérêt...)(thèse de D.Botero [17]) et l'autre aux traitements de plus haut niveau (SLAM par fusion vision monoculaire/IMU) exécutés sur des CPU connectés aux FPGAs (thèse de A.Gonzalez [58]).

Dans ce cadre, un système multi-spectral dans les bandes (VIS) et infrarouge thermique (LWIR) avait été développé, avec deux objectifs : la fusion des deux images pour produire une vue fusionnée composite et la localisation de l'aéronef dans le repère aéroport à partir de la vision.

Dans SART, la focalisation du travail était sur l'architecture du système embarqué sur l'aéronef, plus que sur les algorithmes. C'était l'inverse dans TALVIS. La Figure 2.1 illustre des travaux accomplis dans SART. Dans TALVIS, le focus n'était plus de développer une architecture matérielle intégrable sur un aéronef; les travaux concernaient plus le développement des algorithmes et leur évaluation à l'aide du simulateur d'OKTAL SE.



Figure 2.1: Intégration matérielle réalisée dans le cadre du projet SART : à gauche la campagne de mesures réalisée sur l'aéroport de Blagnac avec les capteurs montés sur un camion et à droite le boîtier des caméras VIS/LWIR (long wavelength infrared) et kits de développement sur FPGAs.

Le projet TALVIS s'est achevé de manière prématurée en Juin 2018, car ses objectifs ne correspondaient plus aux priorités de AIRBUS. Les travaux ont repris avec des objectifs plus ambitieux, dans le cadre du nouveau projet COCOTIER, coordonné par AIRBUS, avec un grand nombre de partenaires. Le but du projet COCOTIER est de se diriger vers une automatisation complète de l'avion, notamment sur la phase de roulage, de décollage et d'atterrissage, par le biais de différentes technologies dont la vision. Cette thèse rentre maintenant dans le cadre d'un lot de travail de COCOTIER, lot auquel sont associés OKTAL SE, VODEA, AIRBUS, le LAAS-CNRS et l'ONERA. Le but de ce lot est de sélectionner différents types de capteurs (sont envisagés la vision dans différentes bandes spectrales, plusieurs types de LIDAR ou de RADAR analysés par l'ONERA), de les caractériser et d'étudier les positions possibles de ces capteurs ainsi que les algorithmes et traitements disponibles pour répondre aux nouvelles fonctions de navigation et de surveillance basées vision. Le projet COCOTIER a démarré en Mai 2019.

Les travaux du LAAS-CNRS ne concernent que la vision multi-spectrale : nous sommes repartis des acquis algorithmiques obtenus dans les projets SART et TALVIS, dans le but d'analyser les images dans le visible ou l'infrarouge, pour détecter des éléments d'intérêt : objets présents sur les taxiways (autres aéronefs, véhicules, piétons...) et signalétiques horizontales et verticales (balises, bords de piste, marquages, barres de stop, numéro des taxiways...). Pour l'instant, comme dans SART, les résultats des algorithmes de détection sont exploités pour construire une vue fusionnée augmentée par ces éléments significatifs, vue affichée pour le pilote. Ces algorithmes seront intégrés à terme, dans des fonctions plus évoluées, qui répondront à des besoins opérationnels, comme la détection d'obstacles, la génération d'alertes s'il existe un risque et le suivi de lignes pour l'aide au guidage de l'aéronef sur les taxiways.

Dans le cadre de la thèse nous nous sommes d'abord concentré sur la détection de motifs puis d'obstacles et la génération d'une image augmentée, en incluant la dimension multi-spectrale du système. À court terme, cette détection doit pouvoir informer le pilote sur les risques mentionnés dans l'introduction : excursion en dehors du taxiway, collision avec des obstacles statiques ou mobiles, et incursion non attendue d'objets sur les taxiways. Ces alertes sont notamment essentielles dans des conditions de visibilité dégradées (nuit, brouillard...). À long terme, la solution proposée devra permettre de répondre à une demande d'automatisation générale de la navigation de l'aéronef au sol.

Il s'agit donc de développer des fonctionnalités qui prennent en compte des conditions dégradées de visibilité d'une part, et des contraintes avioniques de sécurité et d'intégration critiques : puissance de calcul limitée, système embarqué. Notons que notre équipe ne s'intéresse plus à l'architecture du système à concevoir pour exécuter les algorithmes que nous proposons. La PME VODEA participe au projet pour évaluer comment embarquer certains algorithmes dans l'état actuel de la technologie; vu les évolutions très rapides du matériel depuis la fin du projet SART, VODEA étudie des architectures à base de GPU, plutôt que de FPGA.

Vu le nombre important d'algorithmes à considérer, les travaux de cette thèse sont focalisés sur deux types de détection : d'abord la détection de la signalisation horizontale (marquages au sol et balises), puis la détection des obstacles. L'objectif sera de détecter différents objets pouvant être des obstacles à la circulation de l'avion tels que des piétons, des voitures, des bus, d'autres avions ou même des bâtiments. Pour anticiper le temps de manœuvre d'un avion, il faudra être capable de détecter un objet à une distance pouvant s'élever jusqu'à 300 mètres.

Une fois qu'une signalétique ou qu'un obstacle est détecté dans une image à un instant, il conviendra de valider sa détection dans les images suivantes, puis de le suivre d'image en image afin d'évaluer le risque. La thèse a essentiellement porté sur la détection : nous proposons que les éléments détectés à l'instant  $t$  servent de prédiction pour initialiser la détection à l'instant  $t+1$ . Si besoin, vu le faible nombre de signalétiques ou obstacles détectés dans chaque image, le suivi temporel pourrait se faire aisément par des approches de type Tracking by Detection (appariement des objets détectés sur les images successives).

Pour mener à bien ces travaux, nous disposons d'images obtenues grâce au simulateur fourni par l'entreprise OKTAL SE ainsi que de quelques images fournies par AIRBUS, extraites de vidéos acquises par des caméras réelles, généralement installées dans la dérive des avions (FIN CAM). Ce ne sont que des images dans le visible, de qualité médiocre vu qu'elles sont issues d'une décompression. En 2020, AIRBUS ne disposait pas encore de bases d'images brutes (issues directement des caméras) provenant d'avions en phases d'atterrissage, de décollage ou de roulage dans des zones aéroportuaires. Cette situation évolue et de telles bases sont en cours de construction.

Un des objectifs de l'utilisation du simulateur est d'analyser la possibilité de porter des traitements réalisés sur des images simulées à des images obtenues en conditions réelles. Si cette possibilité est avérée, l'utilisation du simulateur permettra de créer une base de données de grande dimension (on parle de plusieurs milliers, voire millions d'images), annotée automatiquement grâce à la connaissance des scénarii précis donnés en entrée de la simulation. Cette base, complétée par un nombre limité d'images réelles annotées manuellement, facilitera notamment le développement de méthodes basées réseau de neurones.

## 2.2 Normes et recommandations aéronautiques

Les travaux réalisés au cours de cette thèse se placent dans le domaine particulier de l'aéronautique. C'est un domaine dans lequel les normes et recommandations sont fortement présentes. Cela présente des avantages et inconvénients dans le développement de solutions technologiques. Ci-dessous, nous décrivons des réglementations importantes relatives à nos travaux : la certification d'algorithmes et les normes que doivent respecter l'instrumentation aéroportuaire, notamment les normes colorimétriques.

### 2.2.1 Certification des algorithmes

Dans le domaine aéronautique, chaque partie d'un avion est soumise à de fortes contraintes en termes de certification, que ce soit les parties matérielles ou logicielles.

La difficulté de certification des algorithmes est fortement présente. En effet, il a par exemple fallu de nombreuses années pour présenter une certification sur l'utilisation de technologies "multi-cœurs". Pour les méthodes basées réseaux de neurones, la certification est encore plus complexe et est toujours aujourd'hui un défi à relever, ce critère étant d'une grande importance dans le domaine de l'aéronautique.

En parallèle de cette thèse, le LAAS-CNRS et AIRBUS ont démarré en Juin 2019 un autre projet de recherche pour accompagner un doctorant CIFRE. Nous avons collaboré sur différents sujets avec ce doctorant, E. Perrotin. Son sujet concerne en particulier, les possibilités de certification d'algorithmes de détection de marquages et d'obstacles dans le domaine aéronautique. Au cours de ces travaux, il est prévu qu'il étudie la capacité de certification d'une partie de l'algorithme proposé dans cette thèse pour répondre à la problématique de détection de signalétiques aéroportuaires.

À plus grande échelle, l'Institut interdisciplinaire d'intelligence artificielle de Toulouse (ANITI) contient la chaire "Nouvelles approches de certification des systèmes basés sur l'intelligence artificielle (IA) pour l'aéronautique civile", dont est titulaire Mme Claire Pagetti, spécifiquement dédiée à cette problématique. De la même façon, le projet DEEL (DEpendable and Explainable Learning) lancé en 2018 et conduit par l'institut de recherche technologique Saint-Exupéry de Toulouse vise à développer des algorithmes d'intelligence artificielle certifiables.

### 2.2.2 Couleurs des éléments de la zone aéroportuaire

Au cours de cette thèse, nous cherchons à faire ressortir des motifs pour remonter des informations au pilote afin d'aider à la prise de décision. Parmi ces motifs, on peut notamment citer les marquages au sol, qui permettent au pilote de positionner correctement l'aéronef à l'instant courant et de suivre des trajectoires établies afin de respecter une bonne circulation dans la zone aéroportuaire. En pratique, dans la plupart des cas, l'avion suit un marquage : la roulette avant doit rester sur ce marquage, sauf lors des virages pour changer de taxiways (phénomène dit de "oversteering").

Dans l'aéronautique, la signalétique de la zone aéroportuaire est longuement définie dans les recommandations de l'OACI (Organisation de l'Aviation Civile Internationale) [78]. De part ces recommandations, les différentes couleurs des signalétiques (marquages, balises...) sont définies par des sous-ensembles de l'espace de couleur XYZ. Cela nous permet donc d'utiliser l'information de couleur comme hypothèse forte notamment pour des algorithmes de détection de marquages au sol.

Avec ces définitions, il est possible de sélectionner des valeurs seuil pour extraire la plupart des marquages d'une image. Cependant, les images exploitées dans cette thèse pour valider les algorithmes, qu'elles soient créées via le simulateur ou acquises via des caméras réelles dans le visible, sont définies dans l'espace de couleur sRGB (un sous-espace de l'ensemble XYZ, alors que la définition de la couleur de référence des marquages, fournie par la norme OACI, est elle donnée dans l'espace XYZ complet, pour des conditions de visibilité correspondant à un temps clair et respectant l'étalon colorimétrique D65<sup>1</sup>.

Au cours de cette thèse, nous nous concentrons plus particulièrement sur les marquages au sol de couleur jaune qui représentent la ligne centrale et les bordures du tarmac aux abords de la piste de décollage. Nous constatons que les bornes définies pour la couleur de référence dans l'espace XYZ ne sont pas comprises dans les plages de couleurs accessibles dans les images que nous possédons.

Pour utiliser cette couleur de référence dans notre implémentation, nous pouvons lui appliquer une modification empirique afin de redéfinir les bornes en fonction de l'espace de couleur sRGB. La Figure 2.2 montre le sous-espace de couleur jaune initial défini par les recommandations OACI (en noir), le sous-espace corrigé de manière empirique (en rouge) ainsi que le domaine sRGB (en bleu) dans l'espace de couleur XYZ. La correction empirique vise aussi à étendre les conditions de visibilité pour lesquelles nous pourrions exploiter cette norme.

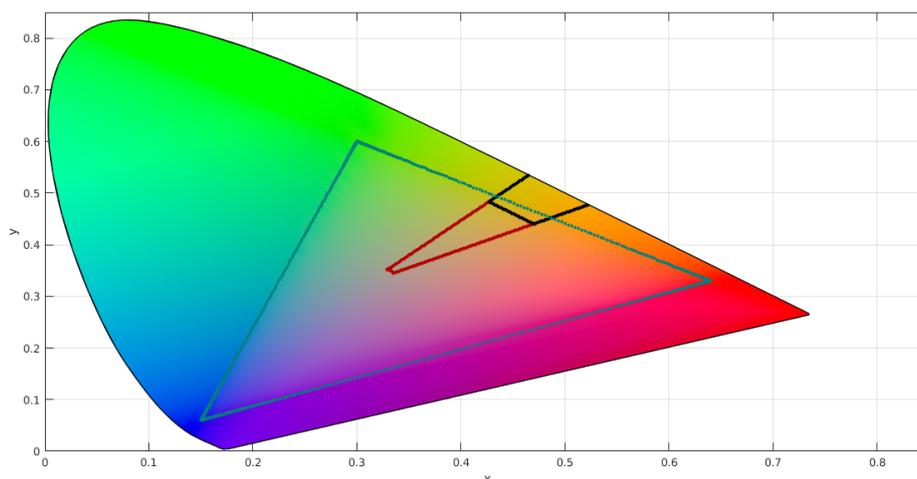


Figure 2.2: Représentation de l'espace de couleur XYZ avec le triangle sRGB (en bleu), les bornes du jaune de référence données par la norme OACI (en noir) et la correction empirique (en rouge).

<sup>1</sup><https://web.archive.org/web/20171204115107/http://www.cie.co.at/publ/abst/s005.html>

## 2.3 État de l'art général sur la vision embarquée

La vision embarquée est une thématique majeure de nos jours. Elle permet de tirer partie de l'information renvoyée par des capteurs (caméras, radars, lidars, sonars...) afin d'aider à l'automatisation de tâches à réaliser par un système. L'essor des calculateurs de plus en plus puissants et de moins en moins coûteux ainsi que l'augmentation des capacités de mémoire des systèmes embarqués de vision a aussi entraîné le développement de nombreuses applications basées vision.

Cette discipline peut avoir plusieurs buts, notamment en robotique ou dans les transports intelligents où elle peut être présente sous les formes suivantes :

- ADAS (Advanced Driver-Assistance Systems ou aide à la conduite automobile) sur les voitures récentes, où le but est d'alerter le conducteur pour qu'il soit conscient des risques à venir (situation awareness, en anglais),
- Systèmes de contrôle d'un véhicule ou d'un robot autonome, où le but consiste à générer des trajectoires ou des lois de commandes soit pour suivre une trajectoire (asservissement exploitant la localisation dans l'environnement), soit pour atteindre un but défini dans une image (asservissement visuel), soit pour éviter de potentielles collisions.

Dans le domaine automobile, on peut citer un grand nombre d'utilisations de la vision embarquée pour :

- La détection d'éléments de signalisation (feux, panneaux, passages piétons)
- La détection de marquages et voies de navigation
- La détection de changement de voies (par exemple les entrées et sorties d'autoroute)
- L'aide au maintien dans la voie de circulation
- La détection et reconnaissance d'obstacles potentiels (piétons, vélos, voitures...)
- L'estimation de la distance à l'obstacle pour le freinage intelligent ou le suivi de véhicule
- L'aide au stationnement
- L'estimation de la distance de visibilité ou aide à la vision nocturne
- L'estimation de dépassement de limite de vitesse

D'autre part, de nombreux projets visent à passer d'une aide à la navigation à une navigation autonome du système de transport. On peut citer par exemple le cas des navettes autonomes qui fonctionnent en environnement contrôlé comme la navette EZ10 développée par les entreprises Ligier et Easymile. Plus axé vers la voiture autonome, on peut citer le challenge VIAC (VisLab Intercontinental Autonomous Challenge) qui, en 2010, a mené quatre véhicules à parcourir 16 000 km de manière autonome entre l'Italie et la Chine.

Au niveau industriel, Waymo a proposé un service de taxi autonome à Phoenix en 2020 alors que Uber a dû ralentir ses tests de voitures autonomes après plusieurs accidents graves. De nombreux projets de voitures autonomes sont expérimentés de nos jours et les constructeurs comme Tesla, BMW, Mercedes-Benz ou encore Toyota prévoient de commercialiser des voitures à haut niveau d'automatisation dans les prochaines années.

Dans le domaine de l'aéronautique, la vision embarquée au service de l'automatisation de la navigation des aéronefs est un objectif à court terme. Le projet ATTOL financé en interne par AIRBUS et cité en introduction a par exemple démontré qu'il est possible d'effectuer, avec un avion commercial, des atterrissages de manière autonome, en exploitant uniquement des technologies embarquées (vision, IMU...), sans aide d'infrastructures externes (en particulier, radio-navigation par ILS pour Instrument Landing System, qui permet des atterrissages automatiques depuis plus de 50 ans). Au niveau recherche, plusieurs brevets ont été déposés en ce sens, par exemple [164] en 2017, dont le but est de détecter un élément spécifique de la signalétique de l'aéroport (barre stop) et la remonter au pilote. L'IATM, qui a coopéré au projet TALVIS, a aussi proposé des études dans le domaine de la navigation assistée par vision (détection de marquages et d'obstacles) [160], [161] et [159]. Enfin, on peut noter les travaux des Universités Techniques de Munich et Braunschweig [177], [6] et [88], qui se concentrent sur l'atterrissage assisté par vision.

## 2.4 Solution proposée au cours de cette thèse

Cette section a pour but de présenter la solution proposée pour répondre à la question de l'aide à la navigation d'aéronefs dans les zones aéroportuaires. Comme indiqué dans la présentation du projet, le choix de différentes technologies de capteurs est envisagé pour répondre à cette problématique. Il convient donc de revenir dans un premier temps sur les technologies disponibles. Ensuite, nous présenterons les différentes briques algorithmiques distinctes mises en place au cours de cette thèse.

### 2.4.1 Capteurs disponibles pour la perception en Robotique et Transports Intelligents

Il existe plusieurs technologies qui permettent l'acquisition de la donnée brute, c'est à dire la réplique numérique de l'environnement étudié. On retrouve bien évidemment les capteurs de types caméra (visible, infrarouge), qui offrent une grande quantité d'informations. En effet, outre des informations d'apparence (luminance, couleur, texture), elles permettent de mettre en lumière des formes géométriques vectrices d'informations importantes, et ce très facilement avec un faible coût.

Seulement, la notion de profondeur pose un problème qui n'est pas trivial en vision même s'il a été résolu par plusieurs techniques. Face à cela, on peut alors opposer d'autres types de capteurs tels que les capteurs laser (Lidar), radar ou encore ultrasons qui permettent d'acquérir directement des informations 3D.

#### 2.4.1.a Ultrasons

De nombreuses recherches ont été effectuées en se basant sur la technique capteurs à ultrasons. En effet, le bas coût des capteurs et leurs bonnes portées de détection en ont fait une technologie intéressante pour le développement de systèmes de détection d'obstacles et c'est donc une des techniques très utilisées en robotique. Le principe du capteur à ultrasons est basé sur la télémétrie. Une onde est propagée au devant de l'objet et la réflexion de cette onde sur de potentiels obstacles est mesurée. En tenant compte du temps écoulé entre l'émission de l'onde et la réception de sa réflexion, on peut déterminer la distance à l'objet. Pour fusionner les mesures successives ou les mesures obtenues par plusieurs capteurs ultrasons disposés autour d'un robot ou véhicule, cette distance peut ensuite être reportée dans une carte d'occupation ou dans des Vector Histogram Fields.

Cependant cette technique a plusieurs inconvénients. Tout d'abord, elle est très sensible à la température et à l'humidité, ce qui la rend peu fiable pour des utilisations en milieu extérieur. De plus, la détection de l'objet se fait dans un cône de détection. Même si la mesure de distance est assez précise, l'information de position exacte dans le cône de détection est bruitée. Enfin, la portée maximale des capteurs ultrasons est limitée (de l'ordre de trois à cinq mètres), ce qui est souvent insuffisant pour les applications de détection d'obstacles dans le contexte automobile sur autoroute par exemple. Néanmoins tous les véhicules modernes sont équipés de tels capteurs pour assister le conducteur dans les manœuvres de parking.

### 2.4.1.b Radar

D'une même façon, les technologies Radar (RADio Detection And Ranging) détectent la réflexion d'ondes radio millimétriques par les potentiels obstacles. Cette technologie est bien plus présente dans l'industrie automobile. Dans [146], trois radars de courte portée sont comparés pour étudier les possibilités de parking d'une voiture. De nombreux véhicules sont équipés de régulateurs de distance à base de radar longue portée (ACC : Automatic Cruise Control).

Un des inconvénients de la technologie Radar est sa faible précision car la donnée perçue est assez bruitée. De plus les capteurs radar ne possèdent pas une grande ouverture, ce qui les rendent peu précis lors de la création de cartes d'occupation. Néanmoins, un radar a une portée importante (supérieure aux autres capteurs). De ce fait, un obstacle peut être détecté et suivi avant d'être perçu et localisé précisément par un LIDAR ou une caméra.

### 2.4.1.c Lidar

La technologie Lidar (LIght Detection And Ranging) est aussi beaucoup utilisée en robotique pour la reconstruction de l'environnement et la détection d'obstacles. D'une façon analogue au sonar et au radar, le lidar permet de reconstruire un système en se basant sur l'étude de la lumière réfléchi par un objet éclairé. La distance est calculée en exploitant soit le temps de vol séparant l'émission d'un pulse lumineux de la réception de la lumière réfléchi par l'objet, soit le déphasage entre le signal lumineux modulé émis, et celui réfléchi par un obstacle.

Contrairement aux caméras, les capteurs Lidar sont peu impactés par la luminosité ambiante et offrent une bien meilleure précision que le radar ou le sonar, autant en distance (télémétrie) qu'en orientation (faible divergence du faisceau laser). Il existe plusieurs capteurs qui peuvent fournir des informations aussi bien en 2D (laser à balayage dans un plan, très utilisé en Robotique depuis les années 80) qu'en 3D (laser scanner à balayage, rendu populaire depuis 2005 environ, par le fournisseur Velodyne).

C'est cependant une technologie très coûteuse, ce qui la rend peu adaptée aux applications bas coût (robotique grand public, automobile). Les informations recueillis par cette technologie, notamment en 3D et lorsque la distance est grande, peuvent être éparées et ne pas suffire à décrire un objet.

Enfin, tandis que les surfaces réfléchissantes ont tendance à saturer le capteur lors de la réception de l'onde réfléchi, les surfaces très mates ou avec un fort angle d'incidence avec le faisceau laser peuvent ne rien réfléchir, créant une absence de mesures. Malgré ces défauts, cette technologie est exploitée sur de nombreuses voitures autonomes en cours de développement (ex: Google car, devenu Waymo).

### 2.4.1.d Lumière structurée

Pour être exhaustif sur les capteurs de profondeur qui exploitent un éclairage laser, citons les capteurs constitués d'un illuminateur laser, qui projettent sur une scène un motif lumineux, et d'une caméra, qui observe la déformation de ce motif par le relief de la scène. Les capteurs RGBD de type Kinect sont de ce type. Un cas particulier est la projection d'un plan de lumière : on parle de nappe laser. Ce plan illumine la scène et la caméra récupère la lumière reflétée par les objets présents.

Ce plan peut être vertical, horizontal ou oblique en fonction du positionnement du capteur et des résultats voulus.

Ces capteurs présentent plusieurs problèmes : (1) pour la Kinect, la portée est faible, (2) en extérieur, l'éclairage apporté par le laser est faussé par l'éclairage naturel du soleil, (3) pour la nappe laser, une immense partie de la scène n'est pas étudiée. Ce sont donc des inconvénients en automobile par exemple, où les obstacles peuvent être nombreux, répartis dans tout l'espace et de toutes tailles donc difficiles à cibler avec un seul plan de détection.

#### 2.4.1.e Caméras

Parmi les caméras, on peut différencier plusieurs spectres :

- **Les caméras RGB (Red, Green, Blue) ou en nuance de gris**

Dans la plupart des cas d'études, les caméras utilisées exploitent les longueurs d'onde de la lumière visible (de 400 à 780 nm environ). On retrouve donc un bon nombre d'applications basées sur des caméras RGB ou GreyScale. Les caméras RGB sont dotées d'un filtre Bayer, monté devant la matrice de pixels. Ce filtre permet d'observer, dans des pixels séparés, les spectres associés aux longueurs d'onde R, G et B. Une opération dite de démosaïcing ou de débayerisation, applique une interpolation afin d'associer à chaque pixel les trois informations R, G et B.

Ces caméras RGB apportent plus d'informations pour comprendre une scène, notamment pour les applications où les attributs de couleur et texture des surfaces sont exploitées (exemple : détection des marquages) et où les informations contextuelles sont recherchées (exemple : segmentation de la route, détection des carrefours...).

Un problème important, qui sera abordé en chapitre 3, concerne la variation de la couleur en fonction de l'éclairage ambiant. Ce problème, désigné par "Color consistancy", fait qu'exploiter la couleur en Vision n'est pas si simple. Certains auteurs préfèrent même s'en passer : par exemple, les points d'intérêt très utilisés en Vision depuis 1990 environ (détection ou suivi d'objets, odométrie optique, SLAM visuel...) sont définis uniquement en luminance.

- **Les caméras infrarouges**

L'utilisation de caméras infrarouges est présente dans diverses applications (vision nocturne, détection de mouvements ou d'intrusions, domaine de la surveillance et de la sécurité, domaine militaire, domaine spatial...) qui utilisent, selon le cas, différentes bandes spectrales (présentées en Figure 2.3). Ainsi on retrouve :

- l'infrarouge proche (near infrared, noté NIR, et short-wavelength infrared, noté SWIR), entre 780 nm et 3  $\mu\text{m}$ , utilisé pour la télédétection
- l'infrarouge moyen (middle wavelength infrared, noté MWIR), entre 3  $\mu\text{m}$  et 5  $\mu\text{m}$ , utilisé dans les applications militaires et spatiales
- l'infrarouge lointain (long wavelength infrared, noté LWIR), entre 8  $\mu\text{m}$  et 14  $\mu\text{m}$  (capteurs à micro-bolomètres en particulier) et très lointain (very long wavelength infrared, noté VLWIR) entre 14  $\mu\text{m}$  et 1 mm, utilisé pour l'imagerie thermique et la détection de personnes.

Même si la diffusion de ces technologies tend à faire baisser les prix, ces capteurs ont encore des coûts prohibitifs pour être déployés massivement sur tout véhicule. On retrouve tout de même l'utilisation de caméras LWIR dans le domaine de la robotique ou sur des véhicules haut de gamme, pour la détection d'obstacles de nuit.

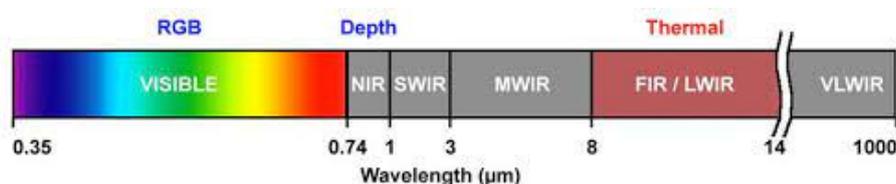


Figure 2.3: Le spectre électromagnétique (extrait de [91]).

#### 2.4.1.f Combinaisons de technologies

Il peut être utile de mélanger les types de technologies pour varier les informations et enrichir le modèle de perception de l'environnement. Cela a même été fait dans de nombreux travaux, notamment pour les véhicules intelligents, sur lesquels on trouve généralement un ou plusieurs radars, lidars et caméras (voir par exemple la Figure 2.4). On retrouve sur ce point plusieurs types de fusions de données, dont on donnera quelques exemples non exhaustifs ci-dessous.

- **Caméras infrarouge et visible**

De nombreuses études portent sur la fusion de données issues d'une caméra couleur et d'une caméra infrarouge. La caméra infrarouge permet d'apporter des informations importantes de nuit ou lorsque les conditions de visibilité ne sont pas adaptées à la caméra RGB dans le domaine du visible, comme en cas de brouillard par exemple. En contrepartie, la trace thermique déposée par l'objet reste encore un certain temps visible en infrarouge après le passage d'un objet, ce qui amène du bruit. La fusion des images issues des deux types de caméras permet donc d'utiliser les propriétés des deux modèles en réduisant leurs inconvénients.

- **Vision 2D et Vision 3D**

Dans [132], les auteurs proposent de fusionner des données de caméra 2D avec des données de capteur 3D (LIDAR principalement) pour réaliser des acquisitions et calculer une odométrie optique en environnement extérieur. Ils présentent certains projets utilisant les caméras 3D pour la cartographie, couplées aux caméras 2D pour la localisation. L'utilisation de la caméra 2D permet aussi de pallier les problèmes pouvant être occasionnés par l'utilisation d'une caméra 3D en extérieur, notamment l'acquisition de données avec une faible densité de points 3D lorsque l'on souhaite reconstruire un environnement extérieur.

- **Stéréo-vision**

Elle permet, en fusionnant les informations en recouvrement dans les images acquises par deux caméras au même instant  $t$ , de déterminer plus précisément la forme et/ou la position 3D d'un objet.

Ces caméras sont rigidement couplées, dans une configuration telle qu'une observe plus vers la droite, l'autre plus vers la gauche. La reconstruction 3D requiert la construction d'une carte de disparité, qui donne, pour chaque pixel de l'image gauche, la différence de position avec le pixel de l'image droite, correspondant à la projection du même point 3D de la scène.

Dans le modèle stéréo le plus courant, les deux caméras sont positionnées pour que leurs plans image soient, au mieux, coplanaires et alignés, que leurs axes optiques soient parallèles, et que leurs centres optiques soient séparés d'une distance appelée "baseline". Ce positionnement parfait étant impossible à obtenir de manière mécanique, on applique aux images d'origine, une opération dite de rectification, qui construit deux images rectifiées, celles qui seraient acquises avec un banc stéréo parfaitement aligné. Généralement les opérations de correction des distorsions et de rectification des images acquises par un banc stéréo se font simultanément, à partir de tables calibrées au préalable.

Cette technique est très utilisée dans le domaine automobile, notamment pour la détection d'obstacles. On peut par exemple citer les travaux de [37], appliqués à la navigation autonome d'un robot en environnement dynamique.

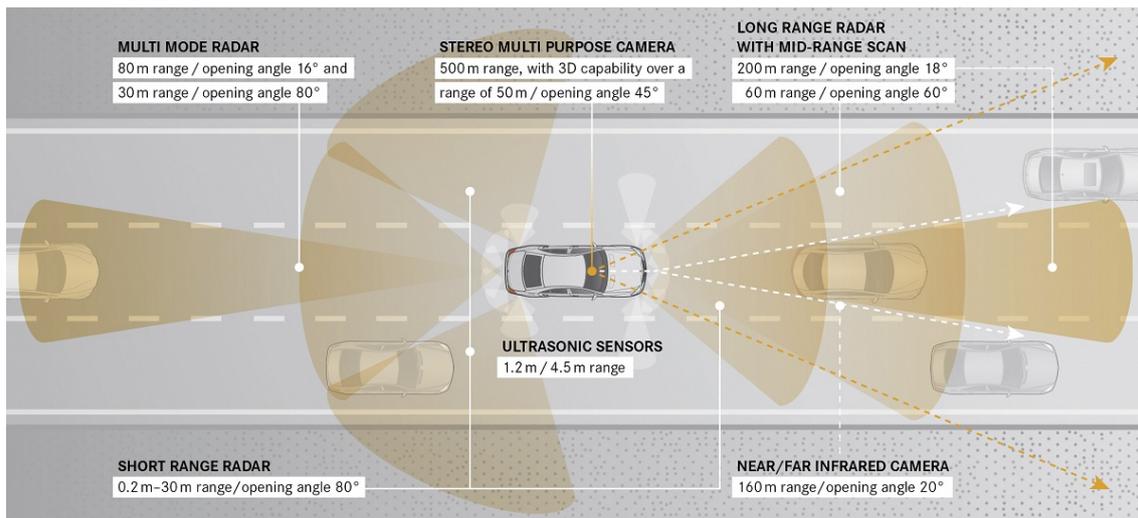


Figure 2.4: Exemple de combinaison de technologies (caméras, radars, ultrasons) pour l'aide à la conduite d'un véhicule (Mercedes Classe S<sup>2</sup>).

#### 2.4.1.g Choix des capteurs pour la thèse

En ce qui concerne les techniques d'acquisition, nous avons présenté ci-dessus les technologies de RADAR, SONAR et LIDAR, ainsi que les caméras visibles et infrarouge. Le contexte du projet nous amène donc à devoir confronter ces technologies afin de remplir notre cahier des charges : détecter la signalétique, et détecter un objet à une distance pouvant aller jusqu'à 300 mètres, avec un champ de vue permettant d'avoir une information sur la distance entre l'obstacle potentiel et les ailes de l'avion, dans n'importe quelles conditions météorologiques (jour, nuit, brouillard...).

<sup>2</sup><https://www.auto-innovations.com/site/brevetech/mercedesSadas.html>

Concernant la signalétique, seule la vision est pertinente, en particulier la vision couleur (donc dans le visible) car comme nous l'avons vu, chaque élément de signalétique (marquage, balise, bande...) a une couleur définie dans la norme OACI. Par contre, les marquages au sol ne sont pas perçus dans des images acquises par des caméras IR (en particulier, LWIR); les balises sont perçues comme points chauds, mais on perd l'information colorimétrique. Notons que faute de données, nous n'avons pas pu tester nos algorithmes dans des conditions de faible visibilité, typiquement de nuit, avec éclairage par les phares avion.

Concernant les obstacles, d'ores et déjà, les techniques d'ultrasons sont déconseillées pour leur sensibilité à la température et à l'humidité ainsi que par leur courte portée. La technologie radar est quant à elle assez bruitée et ne possède pas de grande ouverture donc le champ est limité, ce qui ne convient pas à notre cahier des charges. Par ailleurs, même s'il existe des travaux dans le cadre de COCOTIER impliquant THALES et AIRBUS, il n'existe pas à ce jour de RADAR exploitable de manière pertinente sur un aéronef.

Les technologies Lidar font partie d'un champ de recherche actif et ici aussi, l'ONERA DOTA évalue plusieurs technologies LIDAR dans le cadre de COCOTIER, mais les capteurs de type "Focal Array" existants ont encore une résolution trop faible pour être exploitables dans nos applications. Par ailleurs, les capteurs Velodyne existants sont trop encombrants pour être montés sur un aéronef. Pour le projet TALVIS, on préférera des technologies plus matures comme les caméras RGB et infrarouges.

Pour ce qui est du projet COCOTIER, un intérêt est porté sur la diversification des types de capteurs et leur positionnement autour de l'avion. Dans le cadre de cette thèse, nous réfléchissons à la multiplication du positionnement du boîtier caméra visible / caméra infrarouge à plusieurs endroits de l'avion : sous et sur le cockpit, dans la dérive, à l'avant des ailes de l'avion ou encore derrière la roulette de nez du train d'atterrissage tricycle. Le projet ATTOL n'exploitait qu'une seule caméra, placée dans le cockpit.

Concernant la stéréo-vision, nous l'écartons vu la difficulté de positionner deux caméras dans une configuration satisfaisante. Souvent dans une voiture, les deux caméras sont positionnées de part et d'autre de l'habitacle derrière le pare-brise; difficile à imaginer dans un aéronef tant le cockpit est encombré. Le maintien du calibrage étant essentiel en stéréo-vision, il semble illusoire de les positionner de part et d'autre du fuselage.

## 2.4.2 Présentation des briques algorithmiques

Dans cette partie, nous revenons sur la problématique et les objectifs de cette thèse et détaillons la solution proposée pour y répondre.

### 2.4.2.a Rappel de la problématique

L'objectif de cette thèse est de proposer une solution composée de plusieurs algorithmes en vue de l'automatisation de la phase de roulage pour permettre à long terme de développer un avion autonome. Comme nous avons commencé nos travaux dans le cadre du projet TALVIS centré sur l'assistance au pilote, nous nous concentrons dans un premier temps sur la remontée d'informations au pilote dans l'optique de fournir une aide à la décision.

Pour cela, nous nous sommes focalisés sur trois thématiques d'intérêt pour TALVIS ou COCOTIER. La solution que nous proposons est axée sur ces trois thématiques.

- La détection de motifs d'intérêt dans la zone aéroportuaire : marquages au sol, balises, séparation tarmac/herbe...;
- La détection d'obstacles potentiels et l'estimation de la distance à l'obstacle (un des points du projet TALVIS étant d'estimer si la détection d'un petit obstacle de type valise est réalisable avec les technologies actuelles) à grande distance de l'aéronef;
- L'aide à la navigation par conditions de vision dégradées (nuit, brouillard...)

#### 2.4.2.b Disposition du boîtier d'acquisition

Comme mentionné, le projet TALVIS se situe dans la lignée du projet SART, dont un des résultats a été le développement d'un boîtier, intégrable dans la dérive d'un A380, contenant des caméras RGB et LWIR. Même si ce boîtier n'a pas été réutilisé dans TALVIS, l'option première a été un système d'acquisition composé d'une caméra RGB et d'une caméra IR rigidement couplées (voir Figure 2.5). Ce système d'acquisition est placé, à l'origine, dans la dérive d'un avion, comme l'on peut voir sur la Figure 2.6. Le simulateur fourni par l'entreprise OKTAL SE nous permet de simuler les acquisitions de caméras placées à cet endroit sur le modèle d'avion A380, dans l'environnement simulé de l'aéroport de Toulouse-Blagnac. Dans le projet COCOTIER, nous gagnons en liberté sur le placement de ce boîtier. Cependant, vu l'abandon du développement de A380, la cible de nos études est maintenant l'A320, ce qui limite les places disponibles.

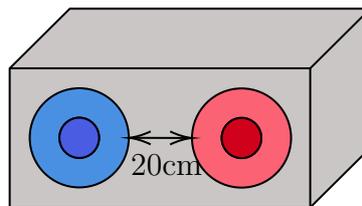


Figure 2.5: Positionnement de la caméra visible (en bleu) et infra-rouge (en rouge) dans le boîtier d'acquisition.

Nous cherchons à savoir si, dans nos conditions d'acquisition, il est possible de détecter de probables obstacles à 350 mètres en amont de l'avion et à 175 mètres de part et d'autre du nez de l'avion, ainsi que la taille de l'obstacle minimum détectable (par exemple un cube de 40 centimètres de côté simulant une valise, ou bien un piéton). Nous voulons aussi pouvoir détecter ces obstacles dans des conditions de vision dégradées.

Une étude réalisée par ONERA et LATECOERE en 2016 sur les produits disponibles sur étagère, a permis de sélectionner des caméras spécifiques, répondant au mieux au cahier des charges de l'application. La résolution de la caméra RGB est de 1280 pixels par 960 pixels, avec un angle de vue horizontal de 80 degrés. La résolution de la caméra IR est de 1024 pixels par 768 pixels, avec le même angle de

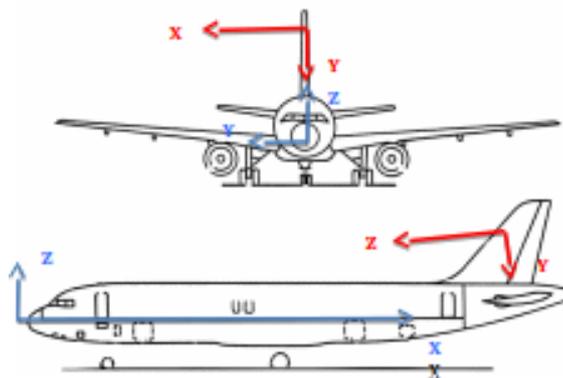


Figure 2.6: Représentation du repère caméra (en rouge) et du repère avion (en bleu).

vue. Les deux caméras sont placées à 19,5 mètres de hauteur par rapport au sol, séparées de 20 centimètres et inclinées d'un angle de 20 degrés.

Le projet COCOTIER permet une souplesse sur la position et la résolution des caméras puisque ces contraintes ne sont pas fixées, le but étant dans un premier temps de valider une solution algorithmique sur des images simulées avant de mener des tests sur des images réelles, en positionnant les caméras dans des positions jugées acceptables par AIRBUS.

Précisons que, grâce aux différentes coopérations avec AIRBUS, nous disposons de quelques images acquises dans le cadre du projet ATTOL. Comme ce projet visait à démontrer la faisabilité d'un atterrissage automatique, la caméra choisie a d'autres caractéristiques: champ de vue plus étroit, et résolution de 4M pixels.

#### 2.4.2.c Découpage de la solution en briques algorithmiques

Par la suite, nous noterons image VIS et image IR, les images acquises par un boîtier tel que celui réalisé dans SART (Figure 2.1) ou étudié dans TALVIS (Figure 2.5).

L'architecture logicielle choisie pour répondre à notre problématique peut être décomposée en trois blocs : un bloc de pré-traitements des images VIS et IR, un bloc d'analyse de scène et un bloc de détection d'obstacles, que nous avons liés les uns aux autres, par notre choix d'algorithmes, comme l'on voit en Figure 2.7.

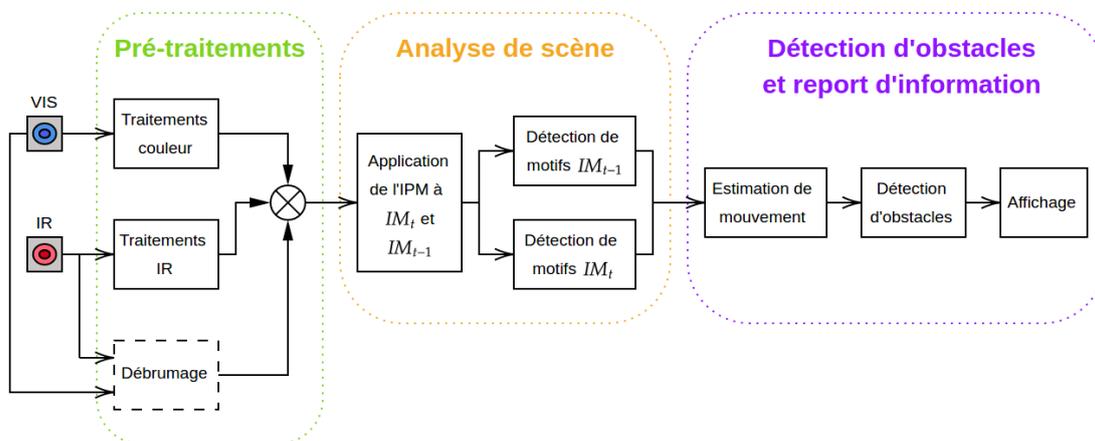


Figure 2.7: Briques fonctionnelles de la solution générale proposée pour répondre à la problématique de l'aide à la navigation d'aéronef.

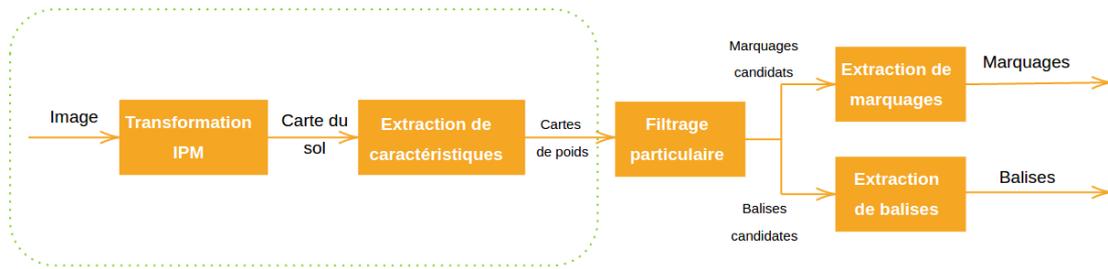


Figure 2.8: Étapes détaillées de la brique algorithmique "Analyse de scène" pour une image à un temps  $t$  donné.

Dans un premier temps, nous choisissons des pré-traitements de la donnée fournie par la caméra infrarouge ainsi que celle fournie par la caméra visible. Ces pré-traitements, qui seront décrits dans le Chapitre 3, nous permettent de faire ressortir des informations de contrastes et de couleurs dont nous aurons besoin pour la détection des motifs et des obstacles. Un pré-traitement dit de débrumage est aussi appliqué sur les images au besoin, si l'on constate un manque de visibilité dû à un épisode de brouillard, par exemple en utilisant la méthode proposée par [66], appropriée pour la mono-vision. Nous proposons une étude préliminaire sur le débrumage et sur la fusion d'images IR/VIS dans le Chapitre 6.

Le second bloc, "Analyse de scène", est détaillé en Figure 2.8 et sera présenté dans le Chapitre 4. La partie contenue dans l'encadré vert de cette Figure est une phase de pré-traitements, qui est spécifique à notre choix algorithmique d'utilisation du filtre particulaire et sera quant à elle décrite dans le Chapitre 3.

Nous avons choisi d'exploiter une version spécifique du filtre particulaire pour récupérer des motifs d'intérêt (marquages au sol, balises...) qui nous permettront ensuite d'estimer la matrice de mouvement de l'aéronef entre deux instants  $t-1$  et  $t$ . C'est une méthode de recherche de lignes appelée dans la littérature "Line tracking", au sens où on suit une ligne dans une image à partir d'un point initial (une "graine" de ligne ou "Line seed"). Dans notre implémentation de cette méthode, le suivi se fait du bas de l'image vers le haut, et nous exploitons le filtre particulaire pour faire du suivi spatial du motif dans l'image (et non temporel comme cela est d'usage). Une explication plus longue de l'implémentation est donnée dans la Section 4.3.2.

En amont du filtrage particulaire, nous avons décidé de nous baser sur une étude d'une image ayant subi une transformation de type Inverse Perspective Mapping (IPM). Une explication plus complète de la transformation IPM est donnée dans le Chapitre 3. Le principe de cette transformation est de projeter les pixels d'une image sur le plan du sol : on appelle carte du sol le résultat de cette transformation.

Une carte du sol est donc une image, dans lequel nous stockons l'apparence de chaque point 3D appartenant à ce plan. Cette carte a une résolution uniforme, par exemple une cellule correspond à une surface 20x20cm au sol. Dans cette carte, on corrige les déformations perspectives qui apparaissent dans une image du fait de la transformation projective, comme la convergence de lignes parallèles vers un point de fuite. Mais la transformation IPM introduit d'autres déformations : un pixel de l'image correspondant à un point 3D appartenant au sol est projeté au bon endroit sur la carte du sol, par contre, un pixel qui correspond à un point 3D situé au dessus du sol sera décalé.

Un exemple est donné en Figure 2.9. Les points 3D  $P'$  et  $P'''$  sont séparés d'une hauteur  $h$ . Ils se projettent sur les points 2D  $p'$  et  $p'''$  dans l'image. Lorsque l'on passe du plan de l'image au plan du sol en utilisant la transformation IPM, le point 3D  $P'$  se retrouve décrit par le point  $P''$  alors que le point 3D  $P'''$  ne change pas de position.

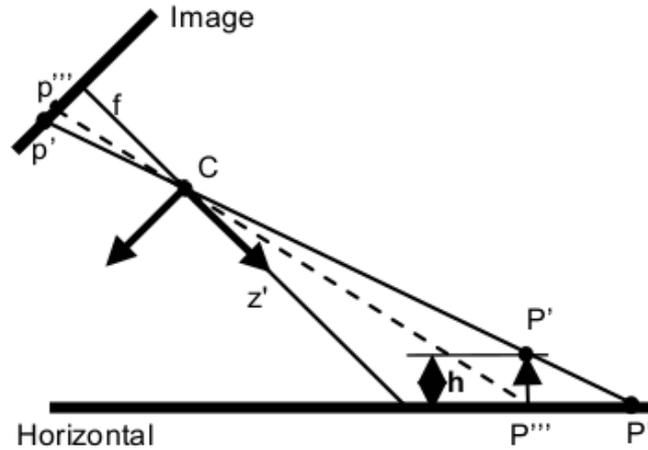


Figure 2.9: Projection sur le plan du sol (extrait de [90]).

Une particularité de cette transformation est qu'elle peut être utilisée pour fusionner des images différentes dans le plan du sol. Ce point est aussi détaillé dans le Chapitre 3.

Enfin, la dernière brique algorithmique de la solution, présentée dans le Chapitre 5, est axée sur la détection d'obstacles. Nous proposons, dans un premier temps, d'utiliser les motifs sélectionnés par la brique d'analyse de scène et de les apparier d'une image à l'autre afin d'estimer la matrice de mouvement de l'avion entre deux cartes du sol aux instants  $t - 1$  et  $t$ . Ces motifs d'intérêt sont ceux mis en valeur par le filtre particulière et correspondent à des objets que l'on souhaite remonter au pilote dans la vue augmentée. Une fois que l'on a estimé la matrice de mouvement, on peut corriger la carte du sol à l'instant  $t - 1$  en la ramenant dans le même espace temporel que la carte du sol à l'instant  $t$ .

On peut alors soustraire deux cartes du sol consécutives (créées à  $t - 1$  et  $t$ ) ramenées dans le même repère temporel afin de faire ressortir les obstacles potentiels. Les cellules de la carte du sol détectées comme "obstacle" peuvent alors être projetées dans l'image visible obtenue grâce aux caméras du boîtier d'acquisition afin de souligner les obstacles potentiels et faire remonter une information visuelle au pilote. Une explication plus complète de cette méthode est donnée dans le Chapitre 5.

## 2.5 Conclusion

Dans ce chapitre, nous avons détaillé les projets de collaboration entre industriels et laboratoires dans lesquels s'inscrivent les travaux réalisés au cours de cette thèse. Nous avons décrit quelques spécificités du domaine aéronautique relatives aux normes et recommandations que nous devons considérer dans nos travaux. Enfin, nous avons présenté la solution algorithmique que nous proposons pour répondre à la problématique d'assistance à la navigation d'un aéronef dans les zones aéroportuaires par vision multi-spectrale. Pour cela, nous sommes d'abord revenu sur les capteurs disponibles sur le marché, en précisant la possibilité de les intégrer sur l'aéronef. Puis nous avons décrit les trois blocs algorithmiques qui composent la solution technique envisagée : un bloc de pré-traitements, un bloc d'analyse de scène et un bloc de détection d'obstacles.

Dans le prochain chapitre, nous décrirons en détail les différents pré-traitements contenus dans le premier bloc.

# Chapitre 3

## Pré-traitements des données

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>32</b>
<b>3.2</b>	<b>Traitement des images infra-rouges</b>	<b>34</b>
3.2.1	Réduction de l'information contenue dans une image IR	34
3.2.2	Techniques de filtrage des artefacts	36
3.2.3	Conclusion sur le filtrage des images infrarouges	37
<b>3.3</b>	<b>Fusion de points de vues multi-caméras</b>	<b>39</b>
3.3.1	Fusion entre images issues d'un même boîtier de caméras	40
3.3.2	Fusion entre images acquises par plusieurs caméras disposées sur un avion	46
<b>3.4</b>	<b>Pré-traitements colorimétriques</b>	<b>51</b>
3.4.1	Méthode heuristique pour la création de la carte d'observation	52
3.4.2	Introduction du problème de la couleur dans la détection de marquages	58
3.4.3	Étude colorimétrique	63
3.4.4	Application à l'affinage adaptatif de la couleur de référence	76
3.4.5	Validation sur des images de test	78
3.4.6	Validation de l'intérêt de la méthode pour des algorithmes de détection de lignes	82
3.4.7	Conclusions sur l'étude colorimétrique	83
3.4.8	Pour aller plus loin : Application à la labellisation semi-automatique	84
<b>3.5</b>	<b>Conclusion</b>	<b>93</b>

---

## 3.1 Introduction

Comme présenté dans le Chapitre 2, il a été choisi d'effectuer une série de pré-traitements sur les images obtenues par le biais des différentes caméras placées sur un aéronef, avant d'étudier les deux gros blocs de l'algorithme global que sont l'analyse de scène (Chapitre 4) et la détection d'obstacles (Chapitre 5). Au cours de cette thèse, nous dissocions les pré-traitements des algorithmes destinés à l'amélioration des images sous conditions climatiques dégradées. Nos évaluations et réflexions sur ce sujet seront abordées dans le Chapitre 6.

Toute sorte de pré-traitements sont appliqués aux signaux reçus par les capteurs avant d'obtenir une image. On peut citer par exemple les algorithmes de dématricage ou de balance des blancs ou les traitements liés à la dynamique du capteur comme le contrôle de gain par exemple. Nous nous plaçons en aval de ces traitements, qui sont souvent intégrés dans les caméras. Nous traitons donc des images déjà modifiées, par des méthodes qu'on ne maîtrise pas forcément : l'impact de ces méthodes sur les résultats de nos algorithmes est évident (par exemple, impact du dématricage sur l'image VIS, impact de la correction des non uniformités (NUC) sur l'image IR...). Toutefois, il subsiste des traitements qui peuvent être nécessaires avant d'appliquer des briques algorithmiques plus complexes.

Dans le cadre de cette thèse, nous travaillons avec des images provenant d'un boîtier d'acquisition composé d'un capteur visible et d'un capteur infrarouge. Ces deux types de capteurs n'offrent pas des images ayant la même dynamique : quantification sur 8 bits pour la caméra dans le visible et sur 14 ou 16 bits pour la caméra infrarouge. Il convient donc d'appliquer des traitements supplémentaires à l'image issue du capteur infrarouge pour réduire la dynamique en conservant les contrastes dans l'image et filtrer le bruit potentiel du capteur. Ces pré-traitements sont abordés dans la Section 3.2.

Pour fusionner les deux images obtenues par le boîtier d'acquisition à l'étude dans le projet TALVIS, il convient d'appliquer des pré-traitements afin de les aligner dans une vue fusionnée, puis d'associer à chaque pixel de cette vue, les luminances enregistrées dans les images d'origine. De plus, cette thèse a pour objectif de réfléchir à l'utilisation de caméras placées à différents endroits de l'avion, pour remonter une information sur l'environnement dans lequel évolue l'aéronef. Pour cela, il faut être capable de fusionner les informations provenant de plusieurs systèmes d'acquisition. Ces problématiques feront l'objet de la Section 3.3.

Comme nous l'avons présenté dans le Chapitre 2, la transformation par Inverse Perspective Mapping (IPM) est un traitement récurrent des images dans la solution algorithmique présentée pour répondre à la problématique de cette thèse. Dans la Section 3.3.1.b, nous décrirons en détail cette transformation et présenterons le résultat associé, appelé carte du sol.

Enfin, la brique algorithmique traitant l'analyse de scène propose une détection de motifs spécifiques (balises, marquages au sol, délimitation tarmac-herbe...). La méthode proposée pour la détection de marquages exploite la couleur pour évaluer si un pixel est issu ou non d'un marquage. Nous avons donc effectué une étude colorimétrique préalable sur la manière d'aborder cette classification, au regard d'une couleur de référence, afin d'améliorer les résultats de l'analyse de scène. Cette étude est présentée dans la Section 3.4 et a fait l'objet d'une publication à VEHITS en 2021 [120].

## 3.2 Traitement des images infra-rouges

Dans le cadre de cette thèse, nous travaillons avec un système d'enregistrement visuel contenant deux caméras : une caméra classique fournissant des images visibles RGB et une caméra infrarouge (dans la bande 8-15  $\mu\text{m}$ , LWIR). Nous souhaitons, avec ce dispositif, étudier la capacité du système à assurer une aide à la navigation par tout temps en utilisant les informations du capteur infrarouge, moins impacté par les conditions météorologiques, en complément de la caméra visible.

Pour cela, nous devons adapter la dynamique de l'image obtenue par le capteur infrarouge à celle de l'image visible afin de pouvoir combiner les informations apportées par les deux capteurs. De plus, les capteurs IR possèdent des artefacts spécifiques et offrent une image qui peut être fortement bruitée. Dans la suite de cette section, nous présenterons quelques solutions proposées pour adresser ces problèmes.

### 3.2.1 Réduction de l'information contenue dans une image IR

Les capteurs IR non-refroidis (comme celui que nous utilisons dans le cadre du projet TALVIS) présentent souvent une dynamique importante par pixel (16-bit) pour pouvoir mesurer une large plage d'intensité de rayonnement infrarouge. Ce rayonnement infrarouge correspond à une mesure de température apparente. Dans la suite, l'intensité d'un pixel de l'image IR sera donc noté  $T$ . Afin de pouvoir afficher cette image, il est nécessaire d'extraire la dynamique intéressante pour l'affichage (8-bit) depuis cette représentation large dynamique.

Il ne suffit pas comme cela se fait généralement pour le visible, de prendre les 8 bits de poids forts, car les intensités sont regroupées généralement dans une plage qui n'est pas dans les hautes dynamiques. Cette adaptation de dynamique doit préserver le contraste dans les différentes régions de l'image, cela pour permettre l'interprétation visuelle de l'image. De plus, on peut noter que, dans le cadre de ce projet, l'information de l'image IR peut être encodée sur 16 bits mais n'utiliser que 14 bits pour exprimer l'information. Ci-dessous, nous proposons quelques méthodes de réduction d'informations.

#### 3.2.1.a Normalisation et Égalisation d'histogramme (HE)

La normalisation d'histogramme, aussi appelée expansion de dynamique, est une technique simple pour améliorer le contraste d'une image. Elle consiste à extraire grâce à un histogramme (qui représente la quantité de pixels pour chaque intensité dans l'image) la dynamique de l'image, puis à augmenter l'étendue de cette répartition en occupant les intensités inutilisées. On obtient alors une image à la dynamique plus étendue, permettant de mieux distinguer les détails dans des zones apparemment unies. Cependant, cette méthode aura tendance à amplifier le bruit (le grain) de l'image et à perdre les détails dans les zones très lumineuses ou très sombres.

L'égalisation d'histogramme quant à elle est une technique qui permet d'équilibrer la dynamique de l'image en étalant de manière uniforme les niveaux de gris de l'image sur tout l'intervalle de valeurs disponibles.

Idéalement elle permet d'obtenir un histogramme "plat" dont l'histogramme cumulé s'apparente à une droite. Elle permet aussi une augmentation globale du contraste dans l'image.

L'égalisation adaptative d'histogramme (AHE [134] ou adaptive histogram equalization) fonctionne de la même façon que l'égalisation d'histogramme mais permet d'augmenter localement le contraste. Pour cela, plusieurs histogrammes sont calculés au sein de l'image, chacun décrivant une zone distincte de celle-ci puis égalisés, ce qui a pour effet d'améliorer la définition des contours de ces zones. Cependant, cette méthode a tendance à amplifier le bruit présent dans les zones homogènes de l'image. Cet algorithme a été grandement amélioré par la suite par diverses équipes de recherche afin d'obtenir un traitement d'image améliorant la dynamique sans perte de détails et autres inconvénients.

### 3.2.1.b Contrast Limited Adaptive Histogram Equalisation (CLAHE)

L'algorithme CLAHE [191] est une extension de l'AHE. C'est une méthode d'égalisation d'histogramme qui, plutôt que de traiter l'image tout entière en une fois, procède en séparant l'image en blocs puis traite chacun de ces blocs séparément en y appliquant une égalisation d'histogramme. Ceci permet de préserver davantage les variations locales de contraste. L'image est ensuite reconstruite par interpolation de proche en proche des pixels de chaque bloc. Cet algorithme présente donc l'avantage de mieux conserver les détails de l'image par rapport à une égalisation d'histogramme classique mais peut amener à l'apparition d'effets de blocs et de bandes dans des images qui comportent des zones uniformes. Par exemple lorsque le ciel est présent dans l'image on peut voir l'apparition de bandes sombres dans le ciel en fonction du nombre de blocs choisis. Bien que le CLAHE permette une meilleure conservation des détails de l'image, cet algorithme peut tout de même perdre en détails du fait qu'il ne se base que sur l'histogramme.

Les paramètres de CLAHE dépendent généralement du type de scène observée, mais on peut s'attendre à une faible variation de l'environnement, ce qui permet une utilisation simple de cet algorithme. Quelques exemples peuvent être vus sur la Figure 3.1. Les images de gauche représentent une donnée infrarouge après normalisation d'histogramme et passage sur 8 bits alors que les images de droites ont été traitées grâce au CLAHE.

Dans le cadre du projet SART, D. Botero [17] a développé une fonction de correction de dynamique de l'image IR en utilisant l'algorithme CLAHE. Cette fonction s'exécute sur FPGA à la fréquence pixel.

### 3.2.1.c Bilateral Filter and Dynamic Range Partitioning (BF & DRP) amélioré

Cet algorithme [192] se fonde sur une toute autre technique puisqu'il se base sur la séparation de l'image en deux composantes via un filtre bilatéral : la base de l'image et ses détails. Ces deux parties sont ensuite traitées séparément puis refusionnées pour obtenir une image améliorée. Cependant, cet algorithme nécessitait un réglage très précis selon la scène observée et pouvait donner lieu à des artefacts indésirables tels qu'une inversion de gradient lorsque des zones avec une trop grande différence de température se chevauchent (comme ce serait le cas avec un avion aux moteurs chauds devant un ciel froid).



Figure 3.1: Exemple de deux images infrarouges converties par une normalisation d'histogramme globale (à gauche) et converties grâce au CLAHE (à droite).

Les chercheurs ayant proposé cette méthode ont donc amélioré l'algorithme : ils corrigent les artefacts en appliquant un filtre gaussien adaptatif sur la partie détails de l'image et ajoutent un contrôle du gain adaptatif. Ils traitent ensuite la base avec une projection d'histogramme à seuil. Ces modifications permettent d'obtenir une image plus nette, en étant moins dépendant des paramètres initiaux.

Dans l'article [102], les auteurs proposent de remplacer le filtre bilatéral par un filtre guidé (guided image filter, GIF) qui apporte des résultats similaires mais dont la complexité algorithmique est beaucoup plus faible. Cela permet de réduire le temps de calcul de l'algorithme. Le principe est d'utiliser une image guide pour adapter le filtre localement. Les informations du voisinage d'un pixel de l'image d'entrée sont combinées avec celles de l'image guide. Si l'image guide est la même que l'image d'entrée, la structure (les contours par exemple) du voisinage est gardée sinon la structure de l'image guide impacte le résultat de l'image de sortie. Contrairement au filtre bilatéral, dont l'usage peut provoquer des artefacts dus à l'inversion de gradient, l'image obtenue est cohérente avec la direction du gradient de l'image utilisée pour guider le filtre puisque le filtre est basé sur une combinaison linéaire. Les auteurs ont ensuite proposé une nouvelle amélioration de ces travaux dans [101]. Cette méthode a été développée et évaluée sur nos images mais n'a pas fourni de résultats satisfaisants.

### 3.2.2 Techniques de filtrage des artefacts

Lors du passage de 16 bits à 8 bits on peut tout de même constater un bruit plus marqué au sein de l'image infrarouge. Les algorithmes présentés ci-dessus incluent des méthodes de traitements pour essayer de filtrer localement le bruit sur l'image.

Il existe d'autres méthodes de filtrage d'images, dont les articles [123] et [33] offrent une vue d'ensemble fournie. On peut citer par exemple le filtre moyenneur, le filtre médian ou le filtrage adaptatif [171] qui sont les méthodes les plus communément utilisées.

Des méthodes plus complexes ont été proposées comme celle de l'article [19] où les auteurs comparent plusieurs fonctions de filtrage par rapport à leur méthode qui applique un filtrage non local mais basé sur un moyennage par voisinages : la valeur d'un point  $x$  est définie comme la moyenne des valeurs des points dont les voisinages sont similaires au voisinage de  $x$ . Le voisinage des points est pondéré par un noyau gaussien.

D'autres techniques comme la transformée en ondelettes [67] ou les réseaux de neurones convolutifs [45] ont aussi permis d'améliorer la réduction de bruit au sein des images ainsi que le temps de calcul.

### 3.2.3 Conclusion sur le filtrage des images infrarouges

Dans notre cas, nous avons décidé d'utiliser l'algorithme de CLAHE, déjà utilisé par l'équipe, sans apporter de filtrage supplémentaire à l'image. Cependant, on observe que l'image infrarouge obtenue possède de nombreux artefacts de textures, comme on peut le voir en Figure 3.2. Il pourrait être intéressant d'utiliser un filtre guidé avec l'image couleur pour uniformiser la texture de la ligne et du tarmac.

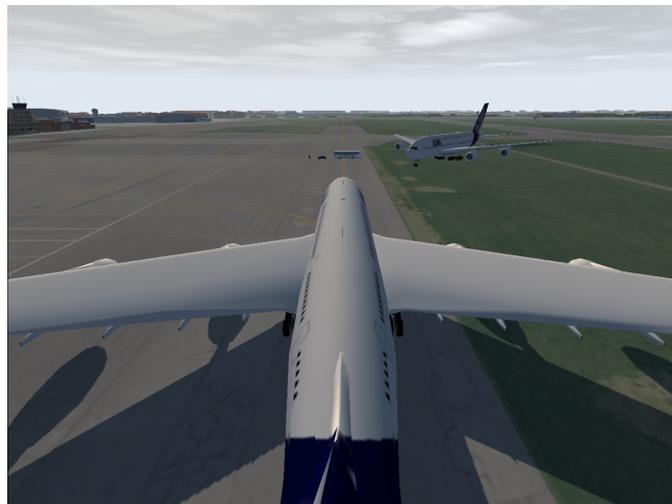
Nous ne nous sommes pas lancés plus avant dans ces traitements spécifiques aux images IR. D. Botero a pu exploiter de vraies images acquises par la caméra MegaPixels développée par FLIR dans le cadre du projet SART. De notre côté, nous n'avons malheureusement pas pu accéder à ce capteur et tous nos travaux sur les images IR se basent sur des images synthétiques générées par OKTAL SE.

OKTAL SE s'appuie sur des travaux communs avec l'ONERA DOTA pour caractériser les propriétés physiques des objets, sur la base de campagnes de mesure sur l'aéroport de Blagnac. On constate que la caméra IR détecte plus de texture sur le tarmac qui n'est pas homogène en matériaux et en température (on voit les traces des roues des derniers véhicules qui sont passés là). Malgré ces efforts, il apparaît souvent des artefacts sur les images IR, comme on peut le voir en Figure 3.2, et la texture sur le tarmac ne paraît pas réaliste vis-à-vis des images réelles acquises dans le cadre du projet SART.

Dans la section suivante, nous présentons une méthode de fusion de données provenant de multiples caméras par l'utilisation de la transformation par Inverse Perspective Mapping (IPM). C'est un pré-traitement important de notre solution algorithmique. Nous décrirons dans un premier temps la transformation IPM avant de présenter son utilisation pour la fusion multi-vues dans la Section 3.3.2.



(a)



(b)

Figure 3.2: (a) Image infrarouge après traitement par l'algorithme de CLAHE et (b) image visible.

### 3.3 Fusion de points de vues multi-caméras

Dans le cadre de cette thèse, nous devons travailler avec plusieurs types de capteurs (caméra visible et caméra infrarouge) et plusieurs positions de capteurs sur l'avion. D'une part, le boîtier permettant l'acquisition, développé dans le cadre des projets SART et TALVIS, est composé de deux caméras, suffisamment proches pour que l'on puisse les fusionner sans perdre trop d'informations mais assez éloignées pour que l'on ne puisse les superposer simplement. D'autre part, les positions des caméras autour de l'avion sont très espacées et demandent une autre forme de fusion de vues.

Il existe plusieurs niveaux de fusion d'informations provenant de caméras multiples [96]. En effet la fusion peut se faire au niveau du signal reçu par les caméras. On forme un seul signal combinant les différents signaux pour améliorer le rapport signal sur bruit (SNR) du signal obtenu par rapport aux différents signaux reçus à la sortie des capteurs.

Le deuxième niveau de fusion est la fusion au niveau des pixels des images reçues, où l'on crée une image combinée en fusionnant l'information provenant des pixels des différentes images acquises. Plusieurs approches existent : d'abord appliquée pixel à pixel, d'autres études ont proposé des algorithmes pour effectuer une fusion par zones de pixels, pour éviter des dégradations d'informations comme la perte de contraste par exemple. Cependant ces méthodes possèdent aussi des défauts puisqu'elles dépendent des paramètres des zones choisies et peuvent entraîner des effets de séparation en blocs de l'image fusionnée.

Une fusion à plus haut niveau est la fusion basée sur l'appariement de primitives sensorielles (points d'intérêt, contours, segments...). Chaque image est d'abord analysée pour faire ressortir des motifs d'intérêt qui sont ensuite recherchés dans les différentes images pour préciser les informations (position, apparence...) sur les éléments 3D de l'environnement qui se projettent en ces motifs. Enfin, la fusion au niveau le plus élevé est la fusion au niveau décisionnel où l'information est totalement analysée pour chaque image et les résultats décisionnels sont fusionnés entre les différentes vues.

De nombreuses études comparatives ont été effectuées dans le domaine de la fusion multi-caméras. Parmi elles, on peut citer [133] qui étudie la fusion d'images acquises à plusieurs résolutions en couvrant la fusion pixel à pixel et la fusion par régions alors que [97] se concentre sur la fusion au niveau du pixel. Dans [104], les auteurs présentent des techniques de fusion au niveau du pixel à base de techniques d'apprentissage profond. Dans l'étude [40] les auteurs présentent un grand nombre de méthodologie de fusion à plusieurs niveaux.

Dans cette section, nous présentons des méthodes d'alignement des deux caméras RGB et LWIR présentes dans le boîtier d'acquisition d'images ainsi que des méthodes de fusion des informations des caméras RGB intégrées en différentes positions sur l'aéronef.

### 3.3.1 Fusion entre images issues d'un même boîtier de caméras

Comment aligner des images acquises sur la scène depuis des points de vue différents ? Dans le cas des deux images RGB et IR fournies par le boîtier d'acquisition monté dans la dérive de l'avion, ces images ont des dynamiques et résolutions différentes, et sont légèrement décalées vu la baseline entre les centres optiques des deux caméras dans le boîtier (14cm pour le boîtier réellement développé dans le projet SART, 20cm dans nos simulations pour les projets TALVIS et COCOTIER). Grâce au CLAHE, la dynamique de l'image IR a été ramenée à 8bits.

Pour les résolutions (1280x960 pour RGB, 1024x764 pour IR), nous pourrions reconstruire l'image RGB pour la ramener dans la résolution de l'image IR, puis simplement, concaténer en chaque pixel dans la vue fusionnée, les luminances RGB et T, mais ce serait négliger la parallaxe.

Un simple calcul fait en considérant la baseline ( $B$ ) de 14cm, des matrices  $2/3''$  et des focales de 6mm ( $\alpha_v$ ) sur chaque caméra, montre que la disparité  $d$  entre les deux images, pour un point 3D à 10m ( $Z$ ) est :

$$d = \frac{B \alpha_v}{Z} \sim 10 \text{ pixels}$$

C'est à dire 1 pixel pour un point à 100m. Cela pourrait être négligé pour les FIN CAM (caméras placées dans la dérive) sur un A380, vu la longueur de l'avion, mais nous proposons plutôt une approche de fusion, en exploitant soit des appariements de primitives, soit une projection inverse sur une surface, ici le plan du sol.

#### 3.3.1.a Alignement basé analyse de la scène

L'alignement de deux images acquises de deux points de vue différents mais assez proche, pour maximiser la zone de recouvrement entre images, peut se faire par la détection de primitives que l'on peut apparier dans les deux images. Ces appariements donnent une information d'alignement local que l'on peut ensuite propager via une homographie. Le calcul de cette matrice d'homographie revient à minimiser l'erreur de re-projection des primitives d'une image dans l'autre. Ce type de méthodes est utilisée par les outils grand public dit de "stitching" permettant de construire un panoramique à partir de plusieurs photos. C'est suffisant pour obtenir un bon aspect visuel, mais on peut avoir des artefacts sur les objets proches.

Dans [23], les auteurs proposent d'aligner les éléments mobiles dans des flux vidéo infrarouge et visible. Les éléments mobiles sont d'abord isolés par une soustraction de l'arrière plan, puis décrits au travers de leurs contours. Les contours des deux vues sont ensuite alignés par la mise en correspondance de points de contours. Les points de contours sont décrits par la convexité locale du polygone et ses angles, qui sont des propriétés invariantes de la bande de perception (infrarouge ou visible). Cette méthode s'avère efficace pour la fusion de séquences acquises de positions fixes (typiquement, des caméras de vidéo-surveillance), sur lesquelles on peut détecter des éléments dynamiques entre lesquels on souhaite faire la fusion.

Dans [63], les auteurs proposent d'apparier des points et des lignes détectés dans une paire d'images visible et infrarouge, pour ensuite calculer une matrice d'homographie reliant les positions de ces primitives dans les images. Cette matrice est ensuite utilisée pour aligner l'image IR sur l'image VIS. Puis, à partir de l'appariement des primitives, une triangulation de Delaunay est appliquée sur

l'image IR d'origine, et chaque triangle est déformé (warping) pour être reporté dans l'image IR alignée.

La détection d'une ligne ne dépend pas de l'orientation du gradient, donc est stable entre images visible et infrarouge. Cela permet de s'affranchir du problème d'inversion de gradient entre les deux images, inversion qui peut impacter l'appariement de points. En effet, la plupart des descripteurs pour des points d'intérêt utilisent un codage du gradient discrétisé dans le voisinage du pixel d'intérêt. Le gradient pouvant varier fortement pour la description d'un élément perçu entre l'infrarouge et le visible, il devient très difficile d'apparier des points extraits de ces deux vues sur la base des descripteurs classiques comme montré dans [49].

Pour pallier ce manque de robustesse des méthodes d'appariement de points d'intérêt avec des descripteurs de type SIFT [149], les auteurs de [49] proposent donc des détecteurs de coins basés sur les détecteurs HARRIS ou LAPLACIAN multispectraux. Ces détecteurs sont donc capables de détecter les mêmes points dans une image infrarouge (NIR en ce cas) ou visible. Ils proposent ensuite une extension du descripteur SIFT pour l'invariance à la direction du gradient.

Toutes les approches décrites précédemment utilisent la mise en correspondance de primitives dans une paire d'images (infrarouge et visible), sans a priori sur la géométrie de la scène observée. Sans connaissance sur la géométrie de la scène, l'alignement des images se fait donc au mieux via des appariements de primitives, mais le résultat peut présenter des aberrations.

Il est possible que nous ne puissions pas, avec la position des caméras, identifier des points d'intérêt sous plusieurs angles différents contrairement à l'étude [179] où les auteurs peuvent reconstituer la carte du sol grâce à plusieurs vues d'un même objet, en s'appuyant sur un alignement dans le plan du sol. Les deux méthodes présentées par la suite définissent ce type d'alignement dans le plan du sol.

### 3.3.1.b Alignement dans le plan du sol

#### Alignement sur la base d'un étalonnage hors-ligne

Le modèle caméra projette un point 3D  $P$  de la scène vers un pixel  $p$  situé sur le plan 2D de l'imageur au travers d'une matrice de changement de repère  $R$  (monde vers caméra) et d'une matrice de projection  $K$ . Cette transformation peut être résumée par la matrice caméra  $C$ , issue du produit  $K \cdot R$ . Lors de cette projection, l'information 3D de la scène est perdue. Dans le cas d'un assemblage rigide de deux caméras visible (VIS) et infrarouge (IR), chaque point de la scène est projeté dans chaque plan image de chaque caméra au travers de la matrice caméra correspondante.

$$p_{VIS} = C_{VIS} \cdot P \quad \text{et} \quad p_{IR} = C_{IR} \cdot P$$

Dans le cas général, ces matrices  $C_{VIS}$  et  $C_{IR}$  sont de dimension  $3 \times 4$ , et donc ne sont pas inversibles : à partir d'un pixel  $p_{VIS}$  ou  $p_{IR}$ , on ne peut remonter au point 3D, noté  $P$ . Pour l'alignement des deux images visible (VIS) et infrarouge (IR), il est nécessaire pour un pixel donné  $p_{VIS}$ , projection d'un point  $P$  de la scène dans l'image VIS, de calculer via une transformation  $H$ , la position correspondante du pixel  $p_{IR}$ , projection du même point  $P$  dans l'image IR.

$$p_{IR} = C_{IR} \cdot P = H \cdot p_{VIS} = H \cdot C_{VIS} \cdot P$$

La matrice  $H$  est une matrice d'homographie 3x3, matrice inversible qui lie les deux plans image. Cette transformation  $H$  ne peut être calculée de manière exacte, que pour des points 3D,  $P$  appartenant à un plan donné  $\Pi$  de la scène [64]. Pour les points  $P$  de ce plan, les équations de projection peuvent en effet se réécrire comme suit.

$$p_{VIS} = H_{VIS} \cdot P \quad ; \quad p_{IR} = H_{IR} \cdot P$$

$H_{VIS}$  et  $H_{IR}$  étant elles-mêmes des homographies, donc des matrices 3x3 inversibles. De ce fait on peut écrire :

$$p_{IR} = H_{IR} \cdot H_{VIS}^{-1} \cdot p_{VIS}$$

$$H = H_{IR} \cdot H_{VIS}^{-1}$$

Pour calculer  $H$ , il est nécessaire d'étalonner la position relative des deux caméras par rapport au plan choisi  $\Pi$  en se basant sur une connaissance a priori du positionnement relatif des deux caméras. Pour le faire hors ligne, on dispose de deux solutions.

Soit on calcule les transformations  $H_{IR}$  et  $H_{VIS}$  en fonction du modèle de l'aéronef (position du boîtier d'acquisition sur l'aéronef), du positionnement relatif des deux caméras dans ce boîtier, et des paramètres intrinsèques de ces caméras. Sinon, on étalonne directement  $H_{IR}$  et  $H_{VIS}$ . Il suffit alors de mettre en correspondance quatre points  $P_i$  de coordonnées connues sur le plan de référence  $\Pi$  de la scène, avec les quatre pixels correspondants  $p_{i_{VIS}}$  et  $p_{i_{IR}}$  dans les images, pour estimer ces homographies à un coefficient près (on fixe généralement  $H_{33}$  à 1), par un système de Cramer à huit inconnues.

L'assemblage pouvant se dégrader ou bouger au cours du temps (dilatation thermique, flexibilités), et les deux caméras étant en mouvement, relativement au plan de référence, il peut être nécessaire de ré-étalonner l'assemblage au cours du temps.

### Alignement sur la base d'un étalonnage en ligne

Comme dit ci-dessus, l'alignement d'images par l'application d'une homographie nécessite de connaître le positionnement relatif des deux caméras ainsi que le positionnement des caméras relatif au plan d'alignement. Dans le cas d'un assemblage positionné sur un objet en mouvement, ce positionnement varie. Afin de fournir un alignement correct pour les points du plan de référence  $\Pi$ , il est donc nécessaire d'estimer en ligne les paramètres de positionnement de l'assemblage par rapport au plan  $\Pi$ . Cette estimation se base sur une connaissance a priori du positionnement, sur l'observation d'éléments de la scène et peut utiliser des données inertielles.

Dans le domaine automobile, les lignes de la route peuvent être utilisées pour extraire une partie des paramètres extrinsèques de la caméra [128]. En effet les lignes de la route étant parallèles, il est possible d'extraire de l'information à partir du calcul du point de fuite de ces lignes dans l'image.

De nombreuses méthodes dans l'automobile exploitent le plan du sol, supposé localement plat. La connaissance de l'homographie reliant le plan image d'une caméra montée sur le véhicule, avec le plan du sol, permet en particulier de générer :

- une image dite vue du ciel ou Bird Eye's View (BEV), image d'une caméra virtuelle située au dessus du véhicule.
- une carte du sol, qui est une projection des pixels de l'image dans le plan du sol.

Dans ces résultats, les effets de perspective qui apparaissent dans l'image acquise par la caméra embarquée sont supprimés.

Pour effectuer l'alignement des caméras VIS et IR présentes dans le boîtier d'acquisition, ainsi que la fusion des données provenant de différents boîtiers, nous avons décidé d'utiliser la transformation par Inverse Perspective Mapping (IPM) qui permet un alignement dans le plan du sol et la création d'une carte du sol. La Figure 3.3 présente les différentes homographies nécessaires à la création de la carte du sol. Ci-dessous, nous proposons une description globale de la transformation IPM.

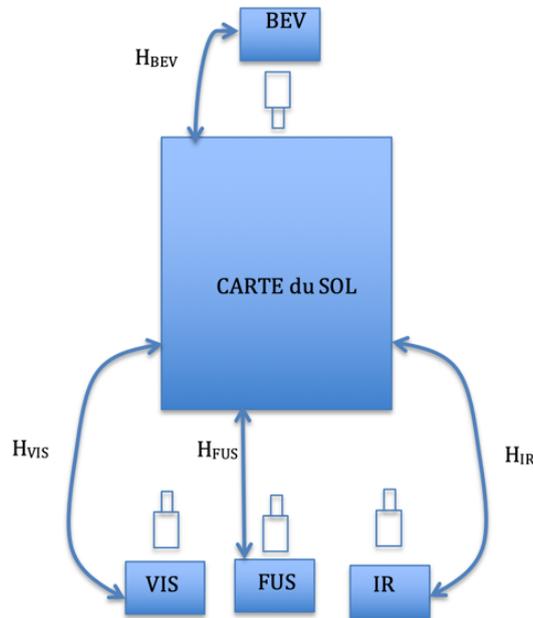


Figure 3.3: Homographies liant les images réelles, les vues construites (BEV et vue fusionnée) et la carte du sol.

### Transformation par Inverse Perspective Mapping (IPM)

Comme décrit dans [35], la scène que nous souhaitons analyser peut être représentée, dans l'espace euclidien, en fonction du repère de coordonnées monde ou dans celui de l'image. Dans notre cas, nous avons décidé de baser notre étude sur une carte du sol générée en utilisant une transformation appelée Inverse Perspective. Pour cela, nous utilisons les différents repères de coordonnées, comme illustré par la Figure 3.4 dans l'exemple d'une caméra placée dans la dérive de l'avion.

On définit un point  $P$  dans le système de coordonnées monde (respectivement dans le système de coordonnées de l'image) par les coordonnées  $P_W(X_W, Y_W, Z_W)$  (respectivement  $P_C(X_C, Y_C, Z_C)$ ). L'expression d'un point dans le repère caméra est alors définie par l'Équation 3.1.

$$P_C = [R_1 \ R_2 \ R_3].P_W + t \quad (3.1)$$

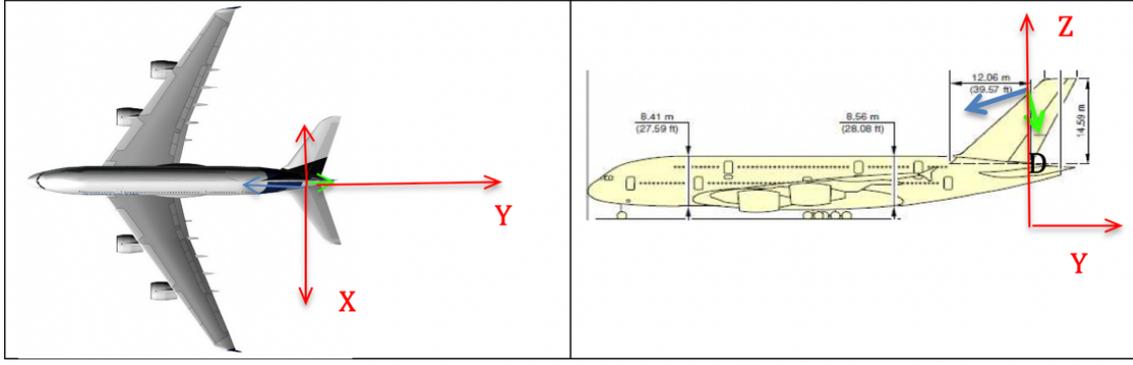


Figure 3.4: Représentation du repère caméra (en rouge/vert/bleu pour  $(X_C/Y_C/Z_C)$ ) et du repère de la carte du sol (en rouge pour  $(X,Y,Z)$ ).

$R_1$ ,  $R_2$  et  $R_3$  représentent les colonnes de la matrice de rotation entre les deux repères tandis que  $t$  est la composante en translation. On note  $X_{W/C}$ ,  $Y_{W/C}$  et  $Z_{W/C}$  les vecteurs  $X$ ,  $Y$  et  $Z$  du repère monde (indiqué  $W$ ) exprimés dans le repère caméra (indiqué  $C$ ) et  $O_{W/C}$  l'origine du repère monde dans le repère caméra. Cela nous permet d'explicitier dans l'Équation 3.2 la matrice de rotation  $R$  et le vecteur de translation  $t$  qui figurent dans l'Équation 3.1.

$$P_C = [X_{W/C} \ Y_{W/C} \ Z_{W/C}] \cdot P_W + O_{W/C} \quad (3.2)$$

L'homographie  $H$  qui permet de passer de la carte du sol à l'image est donnée par l'Équation 3.3. Le repère de la carte du sol est défini à partir de la projection du repère caméra au sol:  $O_W$  est  $O_C$  au sol,  $X_W$  est opposé à  $X_C$  et  $Y_W$  est opposé au  $Z_C$  projeté. Le repère du monde choisi ayant son plan  $XY$  confondu avec le plan du sol, un point du sol s'exprime donc comme  $P_W = (X_W, Y_W, 0)$ .

On peut redéfinir l'homographie  $H$  par les Équations 3.3 et 3.4, où  $K$  est la matrice intrinsèque de la caméra,  $D$  est la hauteur de la caméra par rapport au sol et  $\theta$  correspond à l'inclinaison de la caméra.

$$H = K \cdot [X_{W/C} \ Y_{W/C} \ 0_{W/C}] \quad (3.3)$$

$$H = K \cdot \begin{bmatrix} -1 & 0 & 0 \\ 0 & \sin(\theta) & D \cdot \cos(\theta) \\ 0 & -\cos(\theta) & D \cdot \sin(\theta) \end{bmatrix} \quad (3.4)$$

Pour remplir la carte du sol par transformation IPM, adaptée à nos spécifications, nous réalisons une discrétisation du sol où les dimensions d'une cellule sont définies par  $(\Delta X, \Delta Y)$ . Chaque cellule au sol est repérée par des coordonnées  $(i, j)$  dans la carte, et par ses coordonnées  $(X, Y)$  dans le repère cartésien lié au sol. Ce lien est décrit en Figure 3.5.

Dans notre application, il est nécessaire de percevoir des obstacles jusqu'à 350m vers l'avant. En ce sens, nous construisons une carte du sol de taille 1400x1400 en nombre de cellules, ce qui correspond à 350mx350m si chaque cellule est de résolution 25cmx25cm.

L'Équation 3.5 fait le lien entre les  $(X, Y)$ , coordonnées du centre d'une cellule exprimée dans le repère de la carte du sol, et  $(i, j)$ , ses coordonnées dans la carte du sol.  $D_i$  et  $D_j$  représentent la taille de la carte du sol.

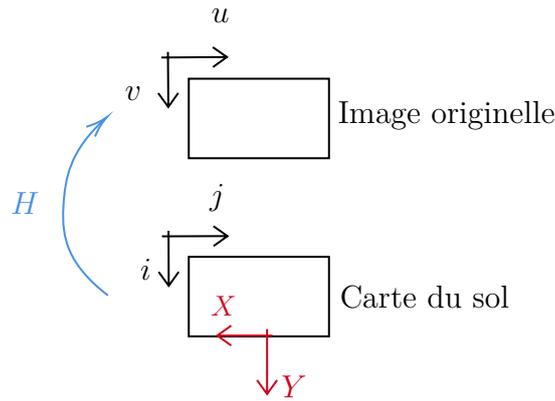


Figure 3.5: Création de la carte du sol par transformation IPM depuis l'image originelle (VIS ou IR).

L'intensité (RGB pour image visible, T pour image IR) reportée dans la cellule  $(i, j)$  est interpolée depuis l'intensité du pixel  $(u, v)$  dans l'image considérée  $IM$  (soit VIS, soit IR dans le cas des images du boîtier). Le lien entre les coordonnées  $(i, j)$  et  $(u, v)$  est donné par l'Équation 3.6.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -\Delta X & \Delta X.D_j/2 \\ \Delta Y & 0 & -\Delta Y.D_i \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad (3.5)$$

$$s. \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H. \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.6)$$

Les coordonnées  $(uv)$  issues de l'Équation 3.6 sont des coordonnées flottantes et donc tombent entre 4 pixels de l'image  $IM$ . Nous calculons l'intensité reportée en  $(i, j)$  de la carte du sol par interpolation bilinéaire. Nous n'avons pas choisi une interpolation par des splines (bi-cubiques ou bi-quintiques) pour ne pas atténuer les contrastes sur les marquages dans la carte du sol.

La Figure 3.5 présente le processus de transformation IPM et les résultats de cette transformation sont visibles en Figure 3.6. Dans la suite de la thèse, la plupart des algorithmes de détection de motifs et d'obstacles vont être appliqués à la carte du sol proposée en Figure 3.6(b).

L'utilisation de la transformation IPM nous permet de projeter dans la carte du sol les informations des différents boîtiers d'acquisition. De plus, chaque boîtier d'acquisition permet de fournir une vue fusionnée de l'image dans le domaine du visible (VIS) et de l'image infrarouge après traitements (IR). Dans le Chapitre 6, nous décrivons le principe de rétroprojection dans une caméra virtuelle située dans le boîtier d'acquisition, c'est à dire la construction d'une vue fusionnée FUS (voir Figure 3.3) de l'image VIS et de l'image IR sur laquelle on peut surimposer des informations relatives aux marquages, balises, obstacles...

Nous voulons apporter, dans un premier temps, une vue fusionnée des différentes caméras au pilote puis, à terme, une réponse décisionnelle. Pour l'instant, nous nous focalisons sur la fusion au niveau des pixels des images par la méthode d'alignement

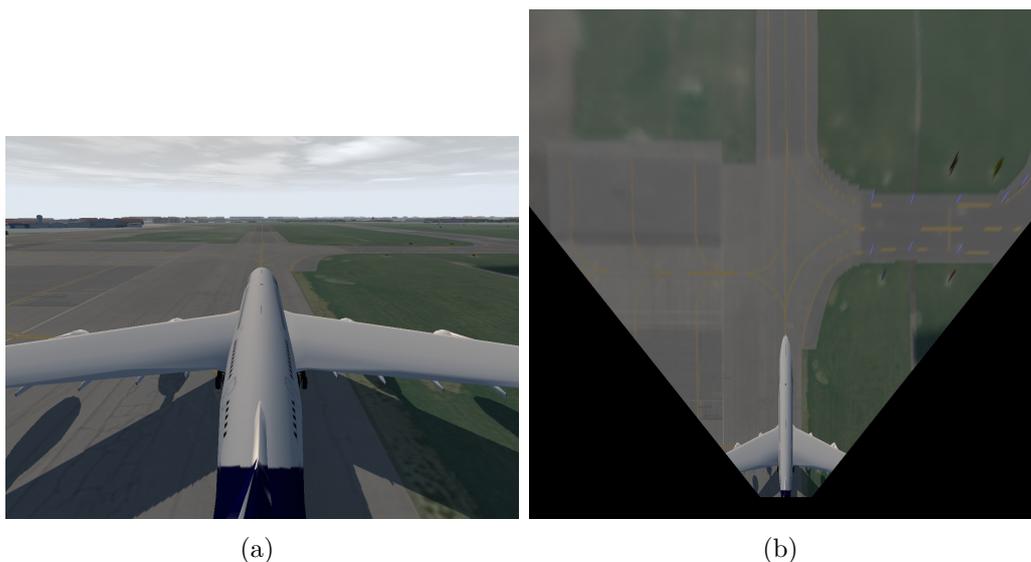


Figure 3.6: (a) Image originelle en RGB, (b) Résultat de la transformation IPM, appelé carte du sol.

par IPM, qui nous permet non seulement d'aligner les caméras au sein du boîtier d'acquisition mais aussi de fusionner les informations provenant de plusieurs caméras ou boîtiers d'acquisition, disposés en différentes positions autour de l'aéronef, dans une carte du sol comme nous le présentons ci-dessous.

### 3.3.2 Fusion entre images acquises par plusieurs caméras disposées sur un avion

Dans le projet COCOTIER, on ne se limite pas aux emplacements existants des capteurs; nous pouvons proposer d'autres emplacements, en particulier mieux adaptés pour détecter les obstacles. On peut s'inspirer de techniques de ceinture de micro-caméras, exploitées en Robotique au LAAS [38], ou dans des systèmes d'aide au parking : chaque image est projetée sur une carte du sol, la détection d'obstacles se fait dans cette carte par diverses méthodes.

Nous disposons de plusieurs emplacements pour les caméras autour de l'avion. Nous avons donc la possibilité de fusionner différentes vues pour obtenir une information plus précise de la détection d'obstacles ou de la détection d'objets. La Figure 3.7 présente un exemple des différentes images que l'on peut obtenir autour d'un avion.

Nous avons choisi d'utiliser la transformation IPM pour effectuer la fusion des différentes vues que nous obtenons avec les caméras placées sur l'avion. Nous pouvons donc définir une carte du sol avec des dimensions pré-définies qui concentre les informations de chacune des images obtenues.

Nous reprenons en Figure 3.8, la Figure 3.4, en intégrant les repères de 4 caméras : FIN CAM qui était déjà dans la Figure 3.4, mais aussi UPPER FRONT CAM, et deux WING CAMs, qui seraient intégrées à gauche et à droite du fuselage près de la jonction des ailes. Nous remettons le repère  $R_w = (O_w, X_w, Y_w, Z_w)$  (en rouge), repère au sol avec  $Y_w$  parallèle et opposé à l'axe avion,  $X_w$  vers la gauche de l'avion,  $Z_w$  verticale, qui passe par le centre optique de la FIN CAM.

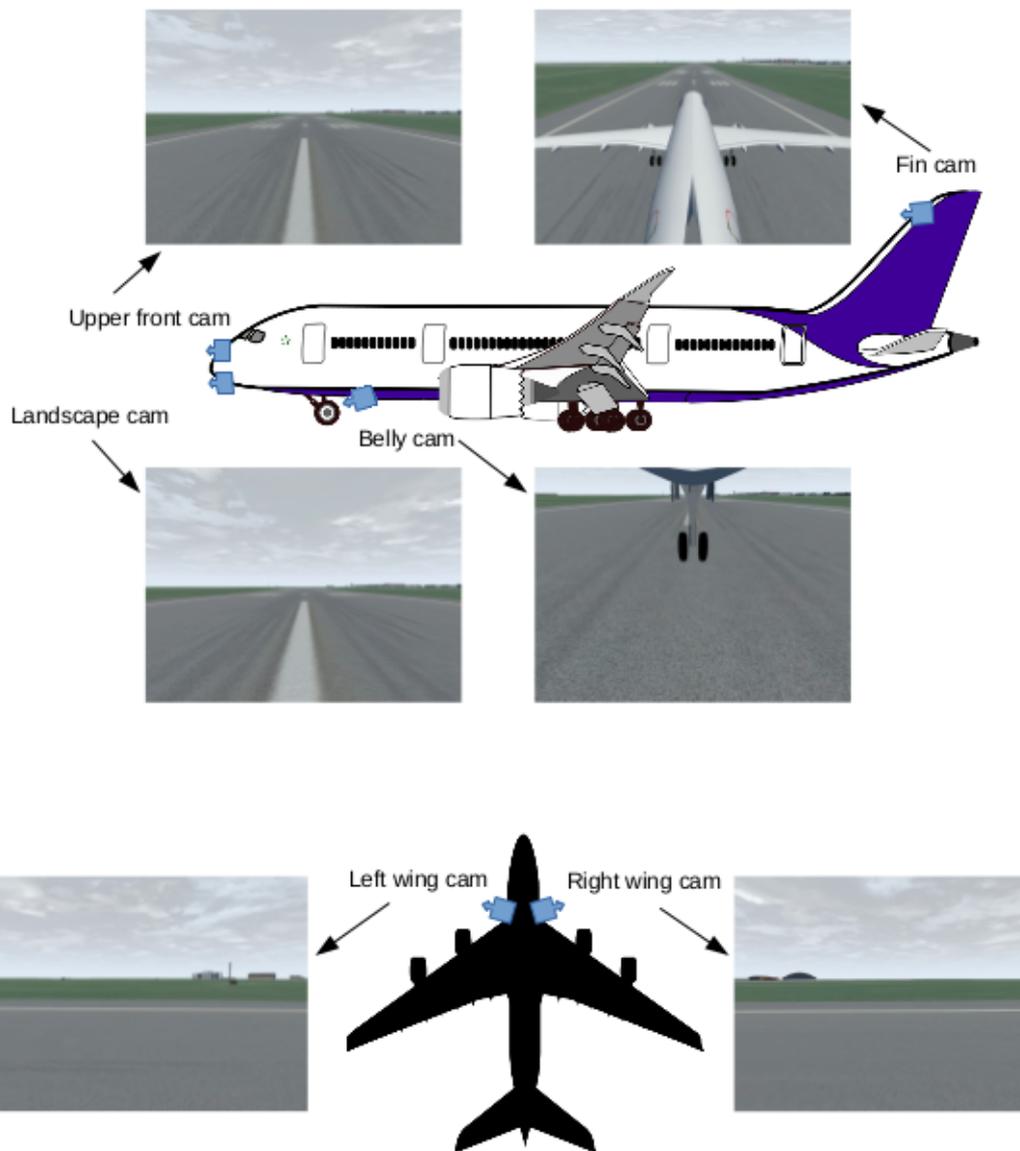


Figure 3.7: Représentation des différentes images obtenues grâce à la multiplicité des caméras embarquées sur un avion.

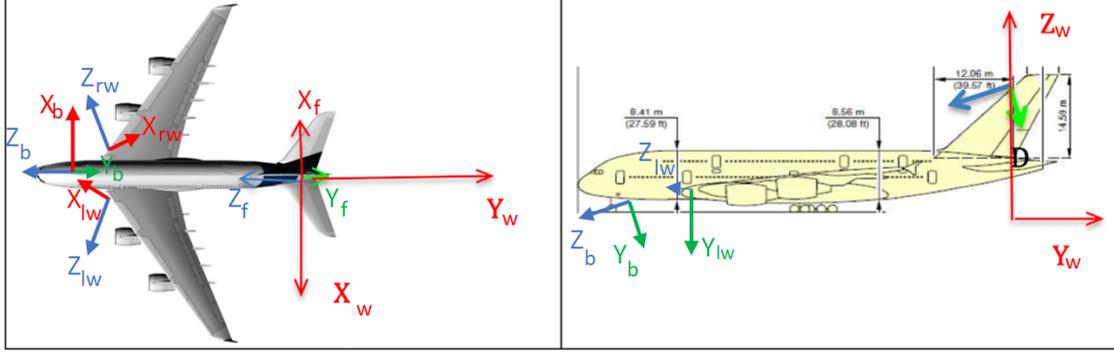


Figure 3.8: Représentation des repères caméras (en rouge/bleu/vert) et du repère de la carte du sol (en rouge et indicé  $w$ ).

On note :

- $\theta_f$ , l'angle de site de la FIN CAM (entre horizontale et axe optique),  $D_f$ , la hauteur du centre optique de la FIN CAM, coordonnée  $z$  du centre optique de la FIN CAM dans  $R_w$ ;
- $\theta_{uf}$ , l'angle de site de la UPPER FRONT CAM (entre horizontale et axe optique),  $D_{uf}$ , la hauteur du centre optique de la UPPER FRONT CAM, coordonnée  $z$  du centre optique de la UPPER FRONT CAM dans  $R_w$ ,  $T_{uf}$ , coordonnée  $y$  du centre optique de UPPER FRONT CAM dans  $R_w$ ;
- $D_w$ , les hauteurs des centres optiques des WING CAMs, coordonnée  $z$  des centres optiques des WING CAM dans  $R_w$ ,  $C_w$  et  $T_w$ , les coordonnées  $x$  et  $y$  des centres optiques des WING CAMs dans  $R_w$  et  $\alpha_w$ , l'angle entre axe avion et axes optiques des WING CAMs (angle vers la gauche, resp. droite, pour WING CAM gauche, resp. droite).

Les angles de site des WING CAMs (entre horizontale et axes optiques) sont supposés nuls. Moyennant ces notations, et les conventions pour les transformations entre repères rappelées dans l'Équation 3.2, nous donnons ci-dessous, pour FIN CAM (rappel par rapport à la section ci-dessus), UPPER FRONT CAM et les deux WING CAMs, les transformations entre les repères des caméras et le repère sol.

Donc en notant, comme en Équation 3.2, la transformation des coordonnées d'un point exprimé dans le repère sol, vers le repère caméra, par la relation  $P_C = [X_{W/C} \ Y_{W/C} \ Z_{W/C}] \cdot P_W + 0_{W/C}$ , nous avons :

- pour FIN CAM,

$$P_{C_f} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & \sin(\theta_f) & -\cos(\theta_f) \\ 0 & -\cos(\theta_f) & -\sin(\theta_f) \end{bmatrix} \cdot P_W + \begin{bmatrix} 0 \\ D_f \cdot \cos(\theta_f) \\ D_f \cdot \sin(\theta_f) \end{bmatrix} \quad (3.7)$$

- Pour UPPER FRONT CAM,

$$P_{C_{uf}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & \sin(\theta_{uf}) & -\cos(\theta_{uf}) \\ 0 & -\cos(\theta_{uf}) & -\sin(\theta_{uf}) \end{bmatrix} \cdot P_W + \begin{bmatrix} 0 \\ D_{uf} \cdot \cos(\theta_{uf}) + T_{uf} \cdot \sin(\theta_{uf}) \\ D_{uf} \cdot \sin(\theta_{uf}) - T_{uf} \cdot \cos(\theta_{uf}) \end{bmatrix} \quad (3.8)$$

- Pour WING CAM gauche,

$$P_{C_{lw}} = \begin{bmatrix} -\cos(\alpha_w) & -\sin(\alpha_w) & 0 \\ 0 & 0 & -1 \\ \sin(\alpha_w) & -\cos(\alpha_w) & 0 \end{bmatrix} \cdot P_W + \begin{bmatrix} C_w \cdot \cos(\alpha_w) - T_w \cdot \sin(\alpha_w) \\ D_w \\ -C_w \cdot \sin(\alpha_w) - T_w \cdot \cos(\alpha_w) \end{bmatrix} \quad (3.9)$$

- Pour WING CAM droite,

$$P_{C_{rw}} = \begin{bmatrix} -\cos(\alpha_w) & \sin(\alpha_w) & 0 \\ 0 & 0 & -1 \\ -\sin(\alpha_w) & -\cos(\alpha_w) & 0 \end{bmatrix} \cdot P_W + \begin{bmatrix} -C_w \cdot \cos(\alpha_w) + T_w \cdot \sin(\alpha_w) \\ D_w \\ -C_w \cdot \sin(\alpha_w) - T_w \cdot \cos(\alpha_w) \end{bmatrix} \quad (3.10)$$

Les homographies permettant de rechercher les coordonnées  $(u, v)$  dans les images, à partir d'un point du sol de coordonnées  $(X, Y)$ , centre d'une cellule de coordonnées  $(i, j)$  dans la carte du sol, sont construites à partir de ces transformations, comme dans l'Équation 3.4 pour FIN CAM.

L'homographie pour la caméra X CAM, est le produit entre la matrice intrinsèque de X CAM, et une matrice 3x3 obtenue en concaténant les deux premières colonnes de la rotation, et le vecteur de translation de la transformation liant le repère du sol avec le repère de X CAM. Le processus pour remplir la carte du sol à partir de plusieurs vues est alors le suivant :

---

**Algorithm 1** Génération de la carte du sol à partir de plusieurs vues.

---

**Pour** chaque  $(i, j)$  dans la carte du sol :

Calculer  $(X, Y)$  sur le plan du sol avec l'Équation 3.5

**Pour** chaque caméra :

Calculer  $(u, v)$  via l'homographie, comme dans l'Équation 3.6 pour FIN CAM

**Si**  $(u, v)$  tombe dans l'image **alors**

- Interpoler pour obtenir une intensité RGB (pour VIS) ou T (pour IR)
- Rappporter la valeur obtenue dans la carte, selon une stratégie de fusion à déterminer

**Fin Si**

**Fin Pour**

**Fin Pour**

---

Plusieurs stratégies sont possibles pour réaliser la fusion d'informations au niveau de la carte du sol :

- On peut fixer une importance égale à chaque caméra. Dans ce cas, les informations des pixels  $(u, v)$  des différentes caméras qui se recoupent dans une même cellule  $(i, j)$  selon les vues sont moyennées sans pondération;
- On peut fixer des degrés d'importance à chaque caméra. Cette fois-ci, on pondère l'information obtenue en  $(u, v)$  suivant la caméra pour définir la valeur de  $(i, j)$ ;
- On définit une heuristique suivant la partie de la carte du sol que l'on génère. Ainsi, on pourra fixer des zones de la carte du sol que l'on réserve pour une ou l'autre des caméras. Par exemple, on préférera la caméra placée dans le

cockpit à celle placée dans la dérive de l'avion pour recréer la carte du sol au niveau du nez de l'avion.

L'évaluation de la fusion multi-images est en cours. Pour le moment, nous fixons une importance égale à chaque caméra.

Nous avons présenté une méthode de fusion entre images provenant de différentes caméras qui s'appuie sur la transformation IPM. Dans la dernière section de ce chapitre, nous nous concentrons sur la notion de couleur au sein d'une image et aux pré-traitements que l'on peut appliquer afin de faire ressortir l'information de couleur. Cette donnée sera exploitée dans le Chapitre 4.

## 3.4 Pré-traitements colorimétriques pour la détection de motifs

La détection de lignes dans une image apporte une des principales informations nécessaire à l'ADAS (Advanced Driver-Assistance Systems) ou à la conduite de véhicules autonomes. C'est pour cela que l'on peut trouver de nombreux travaux effectués sur cette thématique dans le domaine automobile, voir [125].

Un nombre important de ces travaux utilisent des hypothèses ou informations sur la couleur et/ou la notion de contraste dans le but de détecter des lignes dans l'image. C'est pourquoi nous avons souhaité mener un travail préalable sur la notion de couleur avant d'effectuer notre détection de marquages.

Dans un premier temps, il convient d'expliquer précisément pourquoi nous nous intéressons à la couleur. Notre brique d'analyse de scène repose sur l'utilisation d'un filtre particulière. Le principe de ce filtrage est d'approcher la définition d'un motif recherché par des cycles de prédiction et de correction via des données d'observations.

Ici, nous utilisons le filtre particulière pour détecter, en premier lieu, des marquages mais aussi des balises, motifs pour lesquels nous disposons d'une information sur leur couleur grâce aux recommandations OACI. C'est pourquoi nous avons décidé de baser la fonction d'évaluation, qui permet la correction des prédictions par analyse des observations (ici couleur des pixels étudiés), sur une notion de proximité à la couleur de référence. Cela nous permet de générer, pour tous les pixels de l'image d'entrée du filtre particulière, une carte associant un poids à chacun d'eux, résultat de la fonction d'évaluation. La carte de poids peut-être binarisée pour servir de support à l'initialisation du filtre particulière, que l'on détaillera dans le Chapitre 4.

Dans la suite de cette section, nous détaillerons la méthode heuristique pour la création d'une carte d'observation (ou carte de poids), première méthode que nous avons utilisée pour présenter le filtre particulière dans l'article [118]. Ces travaux ont été initiés dans le cadre du projet TALVIS, en coopération avec S. Kaddaoui, qui a participé à ce projet en tant qu'ingénieur. Ensuite, nous présenterons les problématiques liées à l'utilisation de la couleur pour des applications spécifiques de détections de marquages et quelques exemples de solutions proposées dans la littérature. Puis nous abordons l'étude colorimétrique que nous avons menée afin de rendre plus robuste la génération de la carte de poids. Cela passe par une présentation des différents espaces définis pour représenter les couleurs et les méthodes de calcul de distance entre couleurs existantes. Nous menons une étude sur la représentation d'une couleur particulière (celle des marquages au sol) par rapport aux autres couleurs de l'image pour en tirer une méthode de différenciation des couleurs dans le but d'améliorer la détection des marquages au sol. Le même raisonnement vaudrait pour détecter la présence d'autres motifs (panneaux de signalisation, feux stop, balises milieu ou bord de pistes...).

### 3.4.1 Méthode heuristique pour la création de la carte d'observation

Dans un premier temps, nous nous basions sur la carte du sol obtenue après transformation IPM. Nous avons décidé d'utiliser les caractéristiques de couleur et de contours afin de créer plusieurs cartes de poids que l'on appelle de façon simplifiée carte de contours et carte de couleur. Ces cartes sont ensuite combinées afin de générer la carte de poids. Cette dernière est ensuite fournie au filtre particulière et sert à la correction de la prédiction du filtre particulière.

Le principe de ces cartes est de calculer un poids pour chaque pixel, qui définira sa probabilité d'appartenance à l'objet recherché. On calcule donc deux cartes donnant une information d'appartenance en relation avec les contours et la couleur de l'objet recherché. Les objets ou motifs que l'on souhaite détecter sont les marquages au sol (de couleur jaune), les balises (de couleur bleue) et la délimitation entre le tarmac et l'herbe. La couleur des marquages au sol et des balises est importante puisqu'on trouve plusieurs couleurs de marquages et balises dans les zones aéroportuaires dont la signification est différente.

Dans cette section on présentera d'abord les méthodes utilisées pour générer les cartes de contours et cartes de couleur, puis nous aborderons la fusion de ces cartes afin de créer la carte de poids (ou carte d'observation) et sa binarisation. Enfin, nous concluons sur ces méthodes et sur les raisons qui expliquent pourquoi nous avons entamé un travail spécifique sur la couleur.

#### 3.4.1.a Création des cartes de contours

Nous avons défini deux méthodes principales de génération de la carte de contours. L'une est utilisée pour détecter tout type de contours dans l'image tandis que la seconde est spécifique à la recherche des marquages au sol. Combinées, elles offrent de nombreuses informations.

##### **Carte de contours basée sur un filtre de Canny**

La première carte était créée en utilisant un filtre de Canny sur la carte du sol provenant de la transformation IPM et passée en niveaux de gris. L'intensité des pixels de la carte obtenue était ensuite soumise à une fonction exponentielle pour augmenter le contraste entre les niveaux de gris. La Figure 3.9 est un exemple de carte de contours obtenue à partir de la carte du sol présentée en Figure 3.6(b), après avoir appliqué en plus un filtre gaussien choisi en considérant un modèle de bruit gaussien dans la mesure.

##### **Carte de contours basée sur une modélisation du marquage**

La seconde carte était basée sur le modèle des marquages au sol dans les zones aéroportuaires, notamment la ligne centrale de guidage et n'était générée que dans le cadre de la recherche d'un marquage. Elle rejoint une méthode proposée par [15]. Dans l'aviation, les règles qui régissent les marquages sont nombreuses et communes à tous les aéroports. La largeur des marquages des lignes au sol est régulée et nous pouvons utiliser cette connaissance dans notre modèle de détection de lignes, en transformant cette largeur métrique en nombre minimum et maximum de pixels voisins qui appartiennent à un marquage dans l'image originelle ou dans la carte du sol.

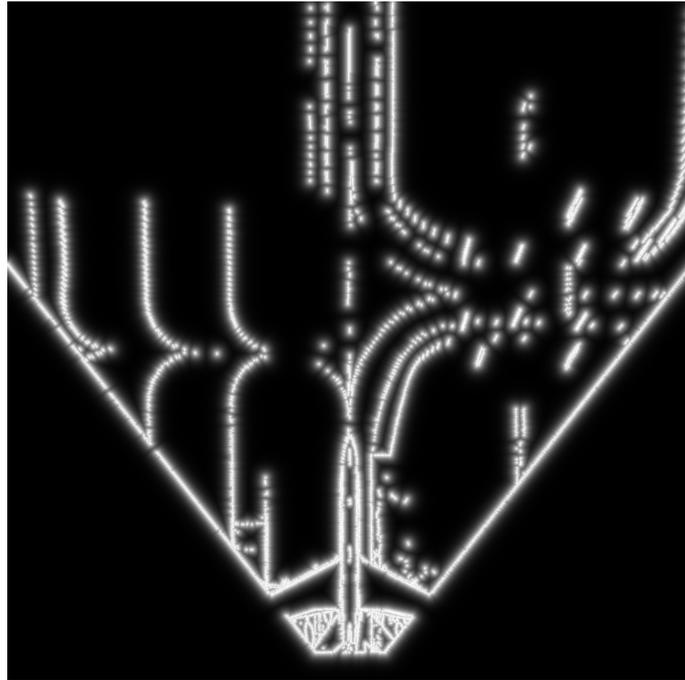


Figure 3.9: Carte de contours basée Canny.

Dans un premier temps, nous convertissons l'image depuis l'espace de couleur RGB dans l'espace de couleur HSV, où nous sélectionnons le canal de saturation. En se basant sur notre connaissance du modèle de l'aéroport, notre algorithme parcourt l'image résultante, dans le canal de saturation, dans le but de créer une version binaire de cette carte. Chaque pixel est comparé à ses  $m$ èmes voisins à droite et à gauche (fixé dans un premier temps à 5).

On cherche à déterminer si le pixel appartient à un marquage ou non par recherche de contraste entre l'intensité de celui-ci  $s(x, y)$  et celle de ses voisins ( $s(x, y + m)$  et  $s(x, y - m)$ ) en fonction d'un seuil  $thresh$ , selon les Équation 3.11 et 3.12. Le paramètre  $m$ , qui est le nombre de pixels représentant la demi largeur d'une ligne dans la carte du sol (résultat de la transformation IPM) a été fixé après étude d'images en simulation.

Une fois que nous obtenions cette carte binaire, nous appliquons une transformée de distance, basée sur l'algorithme présenté dans [47] et implémenté dans OpenCV. De la même façon que pour la première carte, on applique un filtre gaussien à la carte obtenue. Le résultat de ce traitement est visible en Figure 3.10.

$$s(x, y) = \begin{cases} 255, & \text{si } (d_{+m}(x, y) > 0) \ \& \ (d_{-m}(x, y) > 0) \ \& \ (d > thresh) \\ 0, & \text{sinon} \end{cases} \quad (3.11)$$

$$\text{où } \begin{cases} d = d_{+m}(x, y) + d_{-m}(x, y) \\ d_{+m}(x, y) = s(x, y) - s(x, y + m) \\ d_{-m}(x, y) = s(x, y) - s(x, y - m) \end{cases} \quad (3.12)$$

Ces deux cartes ont des propriétés intéressantes pour notre algorithme. Avec la première, nous pouvons détecter la délimitation entre le tarmac et l'herbe, ainsi que les balises (étirées sur le plan du sol par la transformation IPM) et les panneaux.

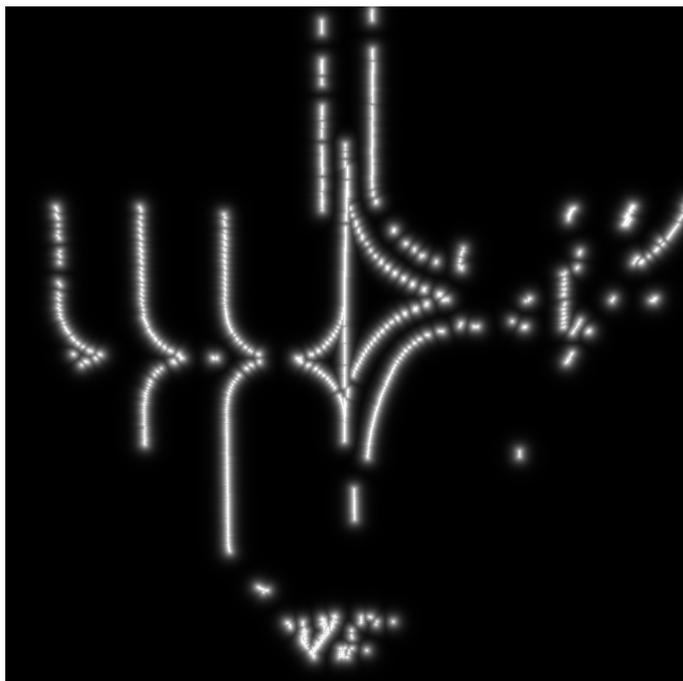


Figure 3.10: Carte de contours basée modèle.

La seconde carte est, elle, concentrée sur le modèle des marquages au sol et permet de réduire le nombre de zones de recherches pour le filtre particulaire.

Utiliser une combinaison des deux cartes (par exemple en appliquant un OU ou un ET logique entre les deux cartes) est aussi intéressant car certains marquages, notamment lorsque la ligne suit une courbe, peuvent être définis par moins de pixels et ne ressortent alors pas forcément dans la carte basée modèle. Cela s'applique aussi au cas où le marquage est partiellement masqué.

#### 3.4.1.b Création des cartes de couleur

Les couleurs des différents marquages et balises qui peuvent être rencontrés dans les zones aéroportuaires suivent des réglementations strictes au niveau des bornes de couleur acceptables. Nous pouvons utiliser ces spécificités pour notre carte de couleur. Cette carte est basée sur une étude de similarité entre une couleur de référence et les pixels d'une image (ou de la carte du sol si on travaille avec la transformation IPM).

Dans un premier temps, nous avons construit la carte de couleur en utilisant une distance euclidienne dans l'espace CIELAB. L'image ainsi que la couleur de référence sont donc convertie en CIELAB puis on applique la distance sur les canaux A et B afin de ne garder que les composantes de chrominance de la couleur.

Dans le cas de la délimitation entre le tarmac et l'herbe, nous avons défini des patches colorimétriques de référence issues d'images réelles et simulées représentant différentes orientations possibles de cette délimitation dans une fenêtre de pixels. Nous appliquons une convolution entre les patches de référence et l'image afin d'extraire la carte de couleur.

La Figure 3.11 donne un exemple de carte de couleur résultante dans le cas des marquages au sol. La Figure 3.12 représente la carte de couleur pour des balises bleues (celles positionnées entre les marquages de bords du taxiway, et la limite tarmac/herbe).



Figure 3.11: Carte de couleur pour les marquages au sol.



Figure 3.12: Carte de couleur pour les balises.

### 3.4.1.c Cartes d'observation globales

Pour chaque objet à identifier (marquages, balises, délimitations du tarmac), nous combinions les cartes de contours et de couleur obtenues ci-dessus. Une valeur était donc attribuée à chaque pixel et représentait une multiplication du poids de la carte de contours par celui de la carte de couleur. Nous avons testé d'autres méthodes de combinaisons mais celle-ci permet de réduire le bruit dans la carte de poids (et dans la carte binarisée qui en découle) de manière significative.

Par contre, on perd certains marquages dans la carte binarisée.

Cette carte d'observation est utilisée par le filtre particulaire pour affecter une confiance dans les particules et aide à l'estimation de la survie d'un cluster de particules. Une copie de cette carte globale est binarisée et utilisée lors de l'initialisation du filtre particulaire. Ces deux points sont expliqués plus en détail dans le Chapitre 4.

La Figure 3.13 présente le résultat de la carte d'observation pour les marquages, où l'on peut voir que le bruit présent dans la carte de contours est filtré par la combinaison de cette carte avec la carte de couleur. Cela améliore la précision de notre détection et réduit le temps de calcul du filtre particulaire puisque nous générons moins d'initialisation de clusters sur des zones de faible confiance (valeurs faibles dans la carte d'observation). En contrepartie, certains marquages risquent de ne pas être détectés.

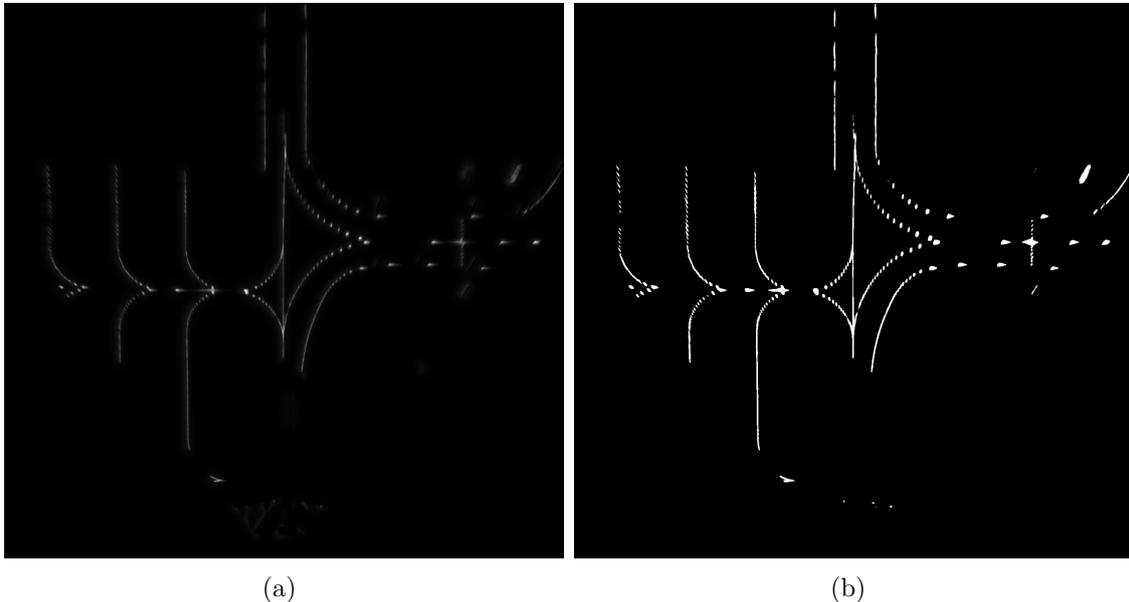


Figure 3.13: (a) Carte d'observation pour les marquages, (b) et carte binarisée associée.

#### 3.4.1.d Bilan sur la méthode heuristique de création de la carte d'observation

L'utilisation de cartes de contours et de cartes de couleur nous a permis de générer une carte de poids fournie au filtre particulaire. Cette méthode permet d'obtenir de bons résultats, que nous avons présenté dans l'article [118]. Cependant, elle possède quelques désavantages.

Dans un premier temps, la génération de la carte de contours basée sur la modélisation des marquages au sol est efficace dans la carte du sol puisque cette dernière corrige des effets de perspective. Cependant, elle peut présenter des difficultés lorsqu'elle est utilisée dans l'image acquise par le capteur ou sur une image dans laquelle les marquages sont en partie effacés. En effet, dans ces cas, la taille du marquage change suivant la distance au capteur ou l'état du tarmac et ne répond plus forcément aux paramètres fixés pour cette méthode.

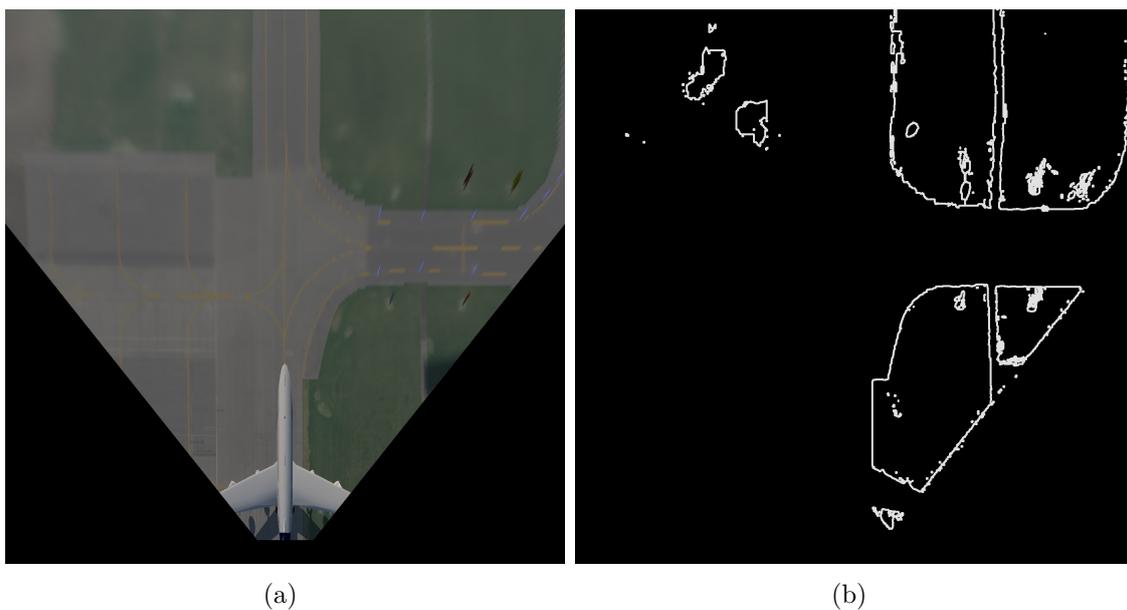


Figure 3.14: (a) Carte du sol et (b) carte d'observation binarisée de la délimitation tarmac/herbe.

De plus, la mise en valeur des contours est rendue plus difficile dans le haut de l'image puisque la qualité de la donnée diminue avec la distance au capteur. Par exemple, dans la Figure 3.14, on peut voir que la carte binarisée de la délimitation tarmac/herbe (Figure 3.14(b)) ne contient pas la bordure située en haut à droite de la carte du sol (Figure 3.14(a)).

Dans un second temps, la génération de la carte d'observation par combinaison des informations de couleur et de contours implique de définir de nombreux paramètres : valeurs de seuil, paramètres pour les filtres gaussiens... Nous avons constaté que cette méthode apporte des résultats satisfaisants pour des images simulées ainsi que pour des images acquises dans des bonnes conditions d'illumination. Cependant, le nombre de paramètres à régler ne rend pas cette méthode robuste aux variations que peut subir notre image d'entrée, notamment au niveau de la couleur.

C'est pourquoi nous avons décidé de nous intéresser à la problématique de la couleur, surtout au niveau des marquages au sol (jaunes) qui sont les motifs dont la détection est la plus sensible aux perturbations.

Dans la section suivante, nous introduisons le problème de la prise en compte de la couleur dans la détection de marquages. Puis nous réalisons une étude colorimétrique afin de proposer une nouvelle méthode de génération de carte de poids, par la suite appelée carte de distance puisqu'elle est uniquement construite sur une étude de distance entre une couleur de référence et la couleur des pixels de l'image, qui soit plus robuste aux perturbations que l'on peut rencontrer dans une application de vision appliquée à la robotique en extérieur.

### 3.4.2 Introduction du problème de la couleur dans la détection de marquages

Les travaux orientés couleur utilisent plusieurs espaces de couleur pour cette application, comme  $L^*a^*b^*$  (ou CIELAB) dans [84], HSI dans [154] ou HSV dans [100] et [113]. Dans le domaine automobile, la majorité des marquages aux sols (qui sont les lignes importantes à détecter dans l'image) sont de couleur blanche, même si cette couleur peut varier selon les types de routes ou les pays. Dans ce cas-là, comme présenté par [125], la plupart des applications de détection de marquages au sol préfèrent convertir l'image en nuance de gris et travailler avec le contraste entre le marquage et la route.

Dans notre cas d'application au domaine aéronautique, nous cherchons à détecter des marquages au sol pour alimenter plus tard des algorithmes de compréhension de scène ou de calcul de position et de déplacement de l'aéronef, nécessaires pour le développement de la conduite autonome. Dans le domaine aéroportuaire, ces marquages peuvent être de différentes couleurs (principalement bleus, rouges, jaune ou blancs) avec un sens propre à chaque couleur. Il nous faut donc pouvoir détecter facilement chaque type de marquage sans confusion de couleur.

Quelque soit la couleur à détecter, la perception de celle-ci est fortement dépendante de la luminosité ambiante, de la météo, de l'heure de la journée..., et donc très variable. Convertir une image depuis l'espace de couleur RGB en une image composée de nuances de gris peut entraîner des confusions, notamment au niveau du contraste entre le tarmac et certaines couleurs de marquage au sol. La couleur à détecter dans l'image peut non seulement être modifiée par l'optique du capteur ou la conversion informatique, mais aussi être impactée par les effets d'ombre et de variations de luminosité ambiante, les reflets ou les dégradations dues aux impuretés présentes dans l'air (poussière, brouillard, neige, pluie). Au cours de cette thèse, nous nous sommes concentrés sur les marquages de couleur 'Jaune' mais les travaux présentés ci-dessous peuvent être transposés aux autres couleurs de marquages.

Les marquages jaunes sont un peu plus compliqués à détecter que les marquages blancs, où la couleur blanche s'exprime facilement dans un bon nombre d'espaces de couleur. En effet, le jaune est plus propice à être confondu avec le vert, le rouge ou le marron clair par exemple, alors que ces couleurs sont présentes dans les zones aéroportuaires. Comme vu précédemment dans le Chapitre 2, le domaine aéroportuaire est fortement réglementé. Dans ce cadre, nous possédons une définition des différentes couleurs possibles pour les marquages, fournie par les recommandations OACI. Chaque couleur est contenue dans un intervalle de valeurs données dans l'espace de couleur XYZ, valide pour un niveau fixé de luminosité. Cependant, il est difficile de bien compartimenter les couleurs dans l'espace XYZ, tout comme dans l'espace RGB. Nous avons donc besoin d'une méthode pour décrire efficacement les différences et similarités entre les couleurs des pixels de notre image pour améliorer notre détection des marquages au sol. C'est pourquoi nous avons mené une étude préalable sur la couleur afin de développer une méthode plus robuste de création d'une carte d'observation pour la détection de marquages.

La détection de couleur dans les domaines automobiles et aéronautiques subit aussi les difficultés apportées avec le fait de travailler sur des images prises en extérieur et donc sensibles aux changements de conditions climatiques.

Ces perturbations peuvent modifier de façon importante la couleur perçue des marquages au sol et impacter négativement les résultats de leur détection. Il est donc important pour notre application de caractériser une définition flexible de la couleur de référence des marquages (donnée par l'OACI).

Le but de cette étude est d'améliorer la détection des marquages (pour l'instant de couleur jaune) pour pallier tous les effets (météo, luminosité, capteur, optique...) qui impactent la détection des marquages au sol. Pour cela, nous proposons une méthode, basée données, d'adaptation de la couleur de référence au sein d'une image. Cette méthode utilise une analyse statistique de la similarité entre des couleurs de l'image et une couleur de référence, en fonction de nombreux espaces de couleur et méthodes de calculs de distance que nous appelons paires (*EspaceCouleur, FonctionDistance*). Elle fournit, après adaptation de la couleur de référence, une information relative aux marquages au sol pour chaque pixel de l'image qui est ensuite utilisée par un algorithme de détection de marquages sous la forme d'une carte de distance dont on extrait une carte binarisée.

Des travaux [30][87] ont été menés sur les similarités et les invariances entre couleurs, avec de nombreux espaces de couleur définis au cours des années. Dans [112], l'auteur présente des résultats de définitions de plusieurs nuances de la couleur jaune dans différents espaces de couleur suivant plusieurs méthodes de calcul de distance entre couleurs. Un certain nombre de fonctions de calcul de distance ont été définies par le CIE (Commission Internationale de l'Éclairage), comme par exemple les distances CIE94 et CIEDE2000 dans l'espace de couleur CIELAB. De la même manière, le système CMC (Color Measurement Committee) a permis de définir le calcul de distance noté CMC l:c (en 1984) pour l'espace de couleur L\*C\*h. Dans le domaine automobile, les espaces de couleur les plus utilisés pour la détection de lignes sont les espaces HSV, HSI et CIELAB, mais dans notre contexte, nous avons rencontré des difficultés avec les valeurs de luminosité et de saturation de la couleur des marquages au sol.

Avec l'augmentation des ressources de calcul et des bases de données d'images ainsi que le développement de réseaux de neurones performants, les problèmes de détection de marquages ont été de plus en plus souvent traités avec des approches basées réseaux de neurones dans la littérature. Dans [83], l'auteur propose d'apprendre par réseaux de neurones une transformation de l'image dans un espace de couleur permettant d'améliorer les résultats de détection alors que d'autres travaux [29][126] se concentrent sur la détection des marquages (et voies de navigation) par réseaux de neurones. L'article [59] conclut que certains espaces de couleur (notamment RGB, CIELAB, et HSV), lorsqu'ils sont utilisés comme pré-traitement d'une image en entrée d'un réseau de neurones de modèle DenseNet [75], peuvent améliorer les résultats de classification de certaines classes d'objets, et que la combinaison de plusieurs de ces espaces pourrait amener à une meilleure précision.

Dans cette dernière étude, les classes définissent plutôt des objets à détecter dans l'image. Dans notre cas, les classes représentent des nuances de couleurs et l'objectif de nos travaux est de trouver quel espace de couleur nous permet de maximiser la labellisation des marquages dans la classe 'Jaune' en minimisant les mauvaises classifications, afin de définir une méthode permettant de fournir une carte des pixels représentant les marquages au sol la plus correcte possible.

Nous avons évalué les approches basées réseaux de neurones sur la base de la méthode proposée par [83] : cela n'a pas offert de résultats satisfaisants. De plus, les transformations non-linéaires basées modèles définies depuis plusieurs années dans le domaine de la colorimétrie ont abouti à des espaces de couleur complexes et nous ne pensons pas qu'une approche basée réseaux de neurones puisse produire une transformation parfaite pour répondre à notre problème. Enfin, la difficulté de certification des algorithmes est un inconvénient principal à l'utilisation de méthodes basées réseaux de neurones et toujours un défi à relever, ce critère étant d'une grande importance dans le domaine de l'aéronautique.

Une autre limitation majeure de ces méthodes est qu'elles nécessitent d'avoir des bases de données d'images labellisées conséquentes pour la phase d'entraînement. Nous ne disposons actuellement que de peu d'images annotées dans notre cadre d'application. De nombreuses bases de données existent dans le domaine de l'automobile mais ce n'est pas le cas dans le domaine aéronautique. C'est pourquoi nous n'utilisons pas d'approches basées réseaux de neurones dans cette étude menée sur la couleur.

### 3.4.2.a État de l'art des utilisations de la couleur pour la détection de marquages

Bien évidemment, la couleur n'est pas le seul indice visuel utilisé pour la détection de motifs au sein d'une image. On peut aussi penser à la texture, aux contrastes et contours, et en général à des descripteurs de l'objet. Dans notre cas, nous regardons spécifiquement le cas de la couleur, puisque nous possédons une hypothèse forte sur celle-ci. Il existe dans la littérature, plusieurs études se basant sur la couleur pour contraindre la détection, avec deux principales approches :

- **Le seuillage colorimétrique basé sur l'espace de couleur**

Comme nous l'avons dit en introduction de cette étude colorimétrique, de nombreux travaux menés sur la détection de lignes s'appuient sur des changements d'espaces colorimétriques pour singulariser les marquages. On peut par exemple citer l'utilisation de l'espace de couleur  $L^*a^*b^*$  dans [84], HSI dans [154] ou HSV dans [100] et [113]. Une fois l'image convertie dans le nouvel espace de couleur, plusieurs techniques de seuillages classiques (par valeur de seuil fixe ou par valeur trouvée par la méthode de Otsu par exemple) sont utilisées pour segmenter les marquages au sol.

- **La sélection d'une zone de l'image**

Partant du principe qu'une ligne se trouve sur la route, de nombreux travaux ont été menés au sein desquels les auteurs proposent de sélectionner une partie de l'image dans laquelle doit se trouver le motif recherché, à savoir un marquage au sol sur la route.

Dans l'article [3], les auteurs proposent de segmenter la route afin de réduire les fausses détections des marquages: pour cela ils partent d'une zone d'intérêt de confiance proche du véhicule, et la généralisent ensuite de proche en proche sur toute l'image. Dans leur étude, ils combinent plusieurs espaces de couleur et différents classifieurs pour assurer la détection de la route.

Dans le cadre automobile, la zone d'intérêt ou ROI (Region Of Interest) est la partie basse de l'image, la plus proche de la voiture (et donc de la caméra)

que l'on suppose être sur la route. C'est aussi majoritairement le cas dans l'aéronautique lorsque la caméra est placée à l'avant de l'avion, même si lorsque l'avion effectue un virage, la caméra peut ponctuellement se trouver au dessus d'une zone qui n'est plus le tarmac. Dans le cas d'une caméra placée dans la dérive de l'avion, le bas de l'image n'est plus une zone aussi privilégiée pour la détection du motif recherché. On peut aussi noter que la couleur et la texture varient suivant la distance à la caméra et rendent la détection de marquages lointains plus difficile si l'on se base seulement sur le bas de l'image pour sélectionner une référence.

Une combinaison des informations des images provenant de différentes caméras pourrait aussi permettre de définir un motif à rechercher, avec une attention particulière à apporter sur le fait que la caméra placée sous l'avion permet de visualiser les marquages au sol mais souffre de l'ombre de l'avion. Dans le cadre de ces travaux, nous n'avons pas poursuivi dans cette voie ni développer de méthode plus performante de segmentation du tarmac mais l'on pourrait imaginer d'utiliser ces informations pour aider à raffiner la couleur de référence que nous cherchons à détecter pour une image donnée.

#### 3.4.2.b Algorithme d'affinage adaptatif de la couleur de référence

Nous avons choisi d'effectuer la détection de marquages au sol en fonction de la couleur. Pour cela, nous avons besoin d'une couleur de référence. Depuis l'état de l'art, nous avons vu que la couleur de référence peut être choisie après avoir changé d'espace de couleur afin de l'isoler plus facilement des autres couleurs de l'image ou en sélectionnant une zone de couleur dans une région d'intérêt.

Nous avons décidé d'utiliser une méthode de seuillage de notre image, en fonction d'une couleur de référence. Ci-dessous, nous présentons un algorithme pour adapter la couleur de référence à un groupe d'images dans le but d'améliorer la segmentation colorimétrique et donc la détection des marquages dans l'image. En effet, puisqu'un grand nombre d'algorithmes de détection de lignes basés modèle utilisent des cartes binaires ou probabilistes (cartes de distance entre la référence de couleur et la couleur des pixels), dont le calcul est principalement basé sur une étude de la couleur, le choix de la couleur de référence dans l'image a un impact important sur le résultat final.

Dans notre cas d'application en aéronautique, nous possédons au départ une définition de la couleur des marquages dans les zones aéroportuaires fournie par les recommandations OACI, comme un sous-ensemble de l'espace XYZ. Il est donc théoriquement possible de définir un nombre d'équations et de valeurs seuils permettant d'extraire la plupart des marquages de l'image, en se basant sur ce sous-ensemble. Cependant, les images que nous récupérons des caméras sont définies dans l'espace de couleur sRGB (un sous-espace de XYZ), et la référence de couleur est définie dans l'espace global XYZ pour une certaine condition de luminosité.

Due aux perturbations de l'environnement, aux conditions météorologiques ou aux post-traitements des capteurs, la couleur des marquages au sol est propice aux variations par rapport à la référence de couleur. C'est pourquoi il nous semble nécessaire de développer une méthode de modification empirique de la définition de la référence de couleur, pour rendre la valeur de la couleur de référence plus flexible tout en la gardant proche de la couleur de référence donnée par les recommandations

OACI, qui subi déjà une déformation lors du passage de son expression dans l'espace XYZ à celle dans l'espace sRGB.

On note  $C^* = \{0.45, 0.45, 0.08\}_{XYZ}$  la couleur du marquage idéale. On peut alors appeler  $\hat{C}$  l'estimateur de  $C^*$  sur l'image obtenue par les capteurs. On propose dans notre solution de calculer  $\hat{C}$  en utilisant un algorithme de  $p$  plus proches voisins (ou  $p$ -Nearest Neighbors), présenté par l'Algorithme 2. Cet algorithme requiert trois paramètres:  $C^*$ ,  $p$  la taille du voisinage considéré, et *tolerance* qui est utilisé comme critère d'arrêt de l'algorithme. un quatrième paramètre *thresh*, peut être fourni et correspond à une valeur seuil utilisée pour convertir la carte de distance renvoyé par l'algorithme en carte binaire.

---

**Algorithm 2** Calcul de la carte de distance en utilisant un affinage adaptatif de la couleur de référence.

---

**Require:**  $p$ , *image*,  $C^*$ , *tolerance*, *thresh*

$i = 0$

$diff = inf$

$\hat{C}_0 = Convert2ColorSpace(C^*)$

$image = Convert2ColorSpace(image)$

**while**  $diff > tolerance$  :

$i = i + 1$

$dist = Distance(image, \hat{C}_{i-1})$

$\vec{x} = GetImageIndexes(sort(dist))$

$\hat{C}_i = \frac{1}{p} \sum_{k=1}^p (image(\vec{x}_k))$

$diff = Distance(\hat{C}_i, \hat{C}_{i-1})$

**end while**

$dist = Distance(image, \hat{C}_i)$

$bin = BinarizeDistanceMap(dist, thresh)$

**return**  $\hat{C}_i, dist, bin$

---

Avant d'utiliser cet algorithme, nous pouvons appliquer un algorithme de flou gaussien sur l'image pour retirer de possibles valeurs aberrantes. Cela permet aussi de renforcer la prise en compte de la contrainte spatiale entre les pixels et de réduire le bruit présent dans l'image. L'algorithme proposé est composé de différentes étapes. Tout d'abord, nous avons constaté que l'espace de couleur RGB n'est pas l'espace de couleur le plus efficace pour comparer des couleurs entre elles et n'est pas trop utilisé dans la littérature où on lui préfère les espaces HSV et CIELAB par exemple. C'est pourquoi la première phase de l'algorithme est de convertir l'image et la couleur de référence dans un autre espace de couleur avant d'appliquer une fonction de calcul de distance entre couleurs, comme par exemple la distance euclidienne.

Après l'application du calcul de distance, nous obtenons une image en nuance de gris, qui représente la carte de distance *dist*. Pour affiner la couleur de référence, nous sélectionnons les pixels contenant la plus petite valeur de distance à la couleur de référence. C'est pourquoi nous créons un vecteur de distances contenues dans la carte de distance et sélectionnons un faible nombre de ces pixels déterminé par  $p$ . La nouvelle couleur de référence est alors mise à jour en la remplaçant par la couleur moyenne des pixels sélectionnés.

Ce processus est répété jusqu'à ce que la distance entre l'ancienne et la nouvelle couleur de référence soit inférieure au paramètre *tolerance*. En sortie de l'algorithme, on peut obtenir une carte binaire qui est créée en appliquant une méthode de seuillage à la carte de distance.

Comme nous travaillons sur des images qui peuvent être impactées par les conditions météorologiques ou les ombrages, nous avons besoin que cet algorithme d'affinage de la couleur de référence soit flexible. Cela veut dire qu'il faut que l'on puisse sélectionner des paramètres adaptés à toute situation. L'algorithme que nous proposons a besoin de plusieurs paramètres qu'il faut fixer judicieusement.

C'est pourquoi nous avons mené une étude sur différentes fonctions de calculs de distance et espaces colorimétriques. La Section 3.4.3 présente une étude colorimétrique dans le but de sélectionner le meilleur couple (*EspaceCouleur*, *FonctionDistance*) et la Section 3.4.4 décrit la sélection des paramètres de l'algorithme défini ici, en se basant sur le résultat de l'étude colorimétrique.

### 3.4.3 Étude colorimétrique

Comme nous avons pu le voir dans l'introduction de la problématique, de nombreux travaux utilisent la couleur et/ou le contraste d'une image pour effectuer la détection de marquages. Les solutions qui utilisent la couleur ont besoin de différencier les couleurs entre elles. Pour cela, la méthode privilégiée est de définir une notion de distance entre couleurs et les grouper par proximité, relativement à ce calcul de distance. C'est pourquoi nous avons décidé de travailler sur les fonctions de distance proposées par les travaux effectués sur la notion de similarité entre couleurs dans la littérature. Le but étant de pouvoir fournir une carte d'observation binaire ou probabiliste sélectionnant les pixels appartenant à des marquages.

Dans cette partie, nous proposons de présenter les différents espaces de couleur existants ainsi que les méthodes de calcul de distance entre couleurs associées. Nous choisissons certains de ces espaces et calculs de distance pour, à l'aide d'une petite base de données, sélectionner un couple (espaces, distance) susceptible de faire ressortir la couleur des marquages au sol par rapport aux autres couleurs de l'image. On part de l'hypothèse que la couleur des marquages varie peu par rapport à la couleur de référence et on estime que l'on peut trouver un couple qui nous permettent de minimiser l'effet de cette variation de couleur sur la détection des marquages.

#### 3.4.3.a Espaces colorimétriques existants et distances associées

Il existe de nombreux espaces de couleur qui ont été définis pour décrire au mieux les couleurs perçues par l'œil humain. Ci-dessous, nous présentons les espaces les plus connus et les distances spécifiques à chaque espace.

- **L'espace de couleur sRGB [5]**

C'est l'espace le plus courant pour représenter les images dans le domaine informatique et il est plus communément appelé espace colorimétrique RGB. Chaque canal de cet espace est codé sur 8 bits avec le triplet (0,0,0) représentant le noir et le triplet (255, 255, 255) utilisé pour le blanc. Tout triplet pour lequel les valeurs des trois canaux sont égales représente une nuance de gris.

Il existe une corrélation forte entre les différents canaux de l'espace RGB, ce qui rend la séparation des couleurs par seuillage difficile. D'autres espaces ont été créés pour exprimer les images colorées. De plus, cet espace ne présentant qu'un sous nombre des couleurs perceptibles par la vision humaine, une priorité a été mise sur certaines couleurs. En revanche, les nuances entre le vert, le jaune et le bleu sont sous représentées, ce qui présente un inconvénient dans notre cas de détection des marquages jaunes par exemple.

Dans cet espace de couleur, la fonction de distance usuelle est la distance euclidienne :

$$distance = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \quad (3.13)$$

D'autres calculs de distances ont été envisagés pour se rapprocher de la vision humaine comme par exemple la distance  $D$  proposée dans l'Équation 3.14 ou l'approximation à faible coût de l'Équation 3.15, notée  $\Delta C$ <sup>1</sup>.

$$D = \begin{cases} \sqrt{2 * \Delta R^2 + 4 * \Delta G^2 + 3 * \Delta B^2} & \bar{R} < 128 \\ \sqrt{3 * \Delta R^2 + 4 * \Delta G^2 + 2 * \Delta B^2} & \text{autrement} \end{cases} \quad (3.14)$$

$$\text{avec } \Delta R = (R_2 - R_1), \Delta G = (G_2 - G_1) \text{ et } \Delta B = (B_2 - B_1)$$

$$\Delta C = \sqrt{\left(2 + \frac{\bar{r}}{256} * (R_2 - R_1)^2 + 4 * (G_2 - G_1)^2 + \left(2 - \frac{255 - \bar{r}}{256}\right) * (B_2 - B_1)^2\right)} \quad (3.15)$$

$$\text{où } \bar{r} = \frac{R_1 + R_2}{2}$$

Dans l'article [55], les auteurs proposent d'étudier plusieurs normes (distance de Manhattan, distance euclidienne, distance de Tchebychev) et fonctions de distance perceptive pour analyser la constance de couleurs (notamment dans les espaces RGB, XYZ, L\*a\*b\* et L\*u\*v\*). La distance euclidienne perceptive est notée :

$$PED(e_e, e_u) = \sqrt{w_R(R_e - R_u)^2 + w_G(G_e - G_u)^2 + w_B(B_e - B_u)^2} \quad (3.16)$$

$$\text{où } w_R + w_G + w_B = 1$$

Ils notent  $e_e$  la source de lumière estimée et  $e_u$  la vérité terrain, avec leurs canaux R, G et B associés, et proposent entre autre d'utiliser l'erreur angulaire pour calculer la constance entre couleurs, qui est définie comme suit.

$$d_{angle(e_e, e_u)} = \cos^{-1}\left(\frac{e_e \otimes e_u}{\|e_e\| \otimes \|e_u\|}\right) \quad (3.17)$$

- **L'espace de couleur CIE XYZ**

C'est l'espace de couleur principal en photométrie, les autres espaces de couleur sont exprimés comme des sous-espaces de l'espace CIE XYZ ou des différentes représentations de cet espace. Il représente l'ensemble des couleurs

<sup>1</sup><https://www.compuphase.com/cmtric.htm>

visibles pour un humain moyen. Le problème de cet espace est qu'il n'est pas perceptiblement uniforme, c'est à dire que la distance entre deux points ne suffit pas à décrire la similarité entre deux couleurs.

Les canaux X et Z représentent la chromaticité de la couleur alors que le canal Y représente la luminance. Ils sont définis pour représenter au mieux le système LMS (Long, medium, short) des cônes et bâtonnets de la vision humaine. La Figure 3.15 est une représentation du diagramme de chromaticité de l'espace XYZ (avec X en abscisse et Y en ordonnée).

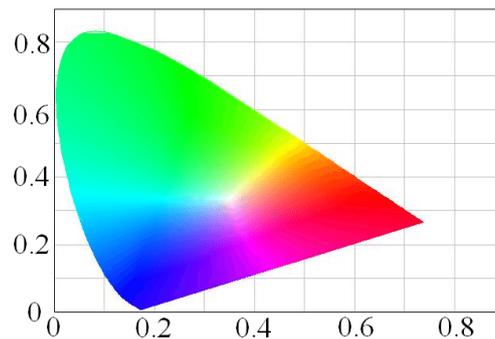


Figure 3.15: Représentation du diagramme de chromaticité de l'espace XYZ [39].

Cet espace a aussi donné lieu à la définition de l'espace CIE xyY où le canal Y est le même que celui de l'espace CIE XYZ et les composantes x et y sont les composantes X et Y normalisées par la somme des valeurs des trois canaux X, Y et Z.

Les espaces CIE XYZ et CIE RGB ont été définis en 1931. Plus tard, d'autres espaces seront proposés par le CIE pour améliorer la représentation des couleurs, comme CIE UVW en 1960, CIE U\*V\*W\*, CIELAB et CIELUV en 1976 ou les CIECAM (CIE Color Appearance Model) CIECAM97s en 1997 et CIECAM02 en 2002 qui essaient de décrire au mieux le système de vision des couleurs humain.

- **Les espaces de couleur CIELUV, CIELAB et LCH**

Les espaces de couleur CIELUV (ou  $L^*u^*v^*$ ) et CIELAB (ou  $L^*a^*b^*$ ) ont été créés pour être plus perceptiblement uniformes en comparaison avec l'espace CIE XYZ, c'est à dire que deux couleurs aux teintes proches devraient être séparées par une faible distance euclidienne. L'espace de couleur LCH (ou  $L^*C^*h$  pour Luminance-Chroma-Hue) est multiples fois défini dans la littérature. La plupart du temps, il est décrit comme une représentation cylindrique de l'espace de couleur CIELAB ou de l'espace de couleur CIELUV, où la composante C est une moyenne quadratique des composantes ( $u^*, v^*$ ) ou ( $a^*, b^*$ ) et la composante H est le résultat de la fonction arc tangente entre les deux composantes  $\text{atan2}(v^*, u^*)$  ou  $\text{atan2}(b^*, a^*)$ .

Bien que la distance euclidienne puisse être utilisée pour calculer la différence de couleur entre deux couleurs dans ces espaces, d'autres fonctions de distance ont été définies spécifiquement pour l'espace  $L^*C^*h$  provenant des coordonnées de l'espace CIELAB.

1. **Une amélioration de la distance euclidienne en LCH [143]**

On appellera ici ' $\Delta C_{LCH}$ ' une distance euclidienne pondérée de l'espace LCH définie par l'Équation 3.18, où  $\Delta L$  et  $\Delta C$  sont les différences entre, respectivement pour les canaux L et C, la couleur de référence (indiquée  $ref$ ) et la couleur d'un pixel (indiquée  $pix$ ).  $\delta H$  est une combinaison d'informations apportées par les canaux C et H des deux couleurs.

$$\Delta C_{LCH} = \sqrt{(\Delta L)^2 + (\Delta C)^2 + (\delta H)^2} \quad (3.18)$$

$$\text{où } \delta H = \sqrt{C_{ref} * C_{pix} * 2 * \sin\left(\frac{H_{pix} - H_{ref}}{2}\right)} \quad (3.19)$$

2. **La fonction de distance CMC l:c [32] ( $\Delta E^*_{CMC}$ )**

Cette fonction de distance a été proposée par le Colour Measurement Committee (CMC) en 1984. Cette métrique est notée CMC l:c où  $l$  et  $c$  sont deux paramètres ( $l$  pour luminosité et  $c$  pour chromaticité) qui permettent d'orienter le calcul suivant la préférence d'application. On retrouve majoritairement les ratios 2:1 et 1:1.

$$\Delta E^*_{CMC} = \sqrt{\left(\frac{\Delta L}{l * S_L}\right)^2 + \left(\frac{\Delta C^*_{ab}}{c * S_C}\right)^2 + \left(\frac{\Delta H^*_{ab}}{S_H}\right)^2} \quad (3.20)$$

$\Delta C^*_{ab}$  et  $\Delta H^*_{ab}$  sont calculés en fonction des composantes  $a^*$  et  $b^*$  des deux couleurs à différencier.

$$\Delta H^*_{ab} = \sqrt{\Delta a^*^2 + \Delta b^*^2 - \Delta C^*_{ab}^2}$$

$$\Delta C^*_{ab} = \sqrt{a^*_{ref}^2 + b^*_{ref}^2} - \sqrt{a^*_{pix}^2 + b^*_{pix}^2}$$

Un élément majeur à noter avec cette formule est que  $\Delta E^*_{CMC}(color, ref) \neq \Delta E^*_{CMC}(ref, color)$  puisque le paramètre  $S_L$  ne dépend que de la valeur L de la première couleur, de même pour les paramètres  $S_C$  et  $S_H$  qui dépendent des valeurs C et H de la première couleur. Le détail de ces paramètres peut être trouvé dans l'article [121].

3. **La fonction de distance CIE94 [115] ( $\Delta E^*_{94}$ )**

Définie en 1994, cette distance s'écrit comme suit.  $S_L$  est généralement équivalent à 1,  $S_C$  et  $S_H$  sont fonction de la couleur de référence.  $k_L$ ,  $k_C$  et  $k_H$  sont trois constantes dont les valeurs varient suivant l'application (arts graphiques ou textiles).

$$\Delta E^*_{94} = \sqrt{\left(\frac{\Delta L^*}{k_L * S_L}\right)^2 + \left(\frac{\Delta C^*_{ab}}{k_C * S_C}\right)^2 + \left(\frac{\Delta H^*_{ab}}{k_H * S_H}\right)^2} \quad (3.21)$$

4. **La fonction de distance CIEDE2000 [108] ( $\Delta E^*_{00}$ )**

Définie en 2000 pour améliorer l'uniformité perceptive, cette distance s'écrit comme suit.  $\Delta L'$  est équivalent à  $\Delta L^*$ ,  $\Delta C'$  et  $\Delta H'$  sont de nouvelles façons d'écrire une différence chromatique entre les deux couleurs à différencier et  $P$  est un terme de pondération fonction de  $R_T$  qui exprime une rotation de teinte (pour des soucis de lisibilité, on

ne détaillera pas les expressions de chaque membre de cette fonction de distance).

$$\Delta E^*_{00} = \sqrt{\left(\frac{\Delta L'}{k_L * S_L}\right)^2 + \left(\frac{\Delta C'}{k_C * S_C}\right)^2 + \left(\frac{\Delta H'}{k_H * S_H}\right)^2 + P} \quad (3.22)$$

$$P = R_T * \frac{\Delta C'}{k_C * S_C} * \frac{\Delta H'}{k_H * S_H} \quad (3.23)$$

- **L'espace de couleur HCL**

L'espace de couleur HCL (Hue-Chroma-Luminance) est présenté dans l'étude [112] avec une fonction de distance associée. Il est créé à partir d'une combinaison des valeurs des canaux R, G et B de l'espace de couleur RGB. L'étude propose aussi une nouvelle fonction de distance pour estimer la similarité entre deux couleurs. Ce nouvel espace est comparé aux espaces de couleur RGB, CIELAB, LCH, CIECAM02 et HSV en utilisant plusieurs fonctions de distance. Les résultats présentent une référence de couleur jaune ainsi que quarante sept couleurs considérées comme proches de la référence suivant les différents espaces de couleur et les différentes fonctions de distance appliquées.

- **Les espaces de couleur HSV/HSB, HSL et HSI**

Ces différents espaces de couleur sont souvent utilisés pour la sélection de couleur et sont assez proches conceptuellement. L'espace de couleur HSV (Hue-Saturation-Value) est équivalent à HSB (Hue-Saturation-Brightness) et diffère des espaces de couleur HSL (Hue-Saturation-Lightness) et HSI (Hue-Saturation-Intensity). Ces espaces de couleur se centrent autour des notions de teinte (ou couleur), de saturation (d'une couleur vive à délavée) et de luminosité (d'une couleur claire à sombre). Pour ces trois espaces, la teinte est calculée de la même façon, les canaux S, V, L et I ayant des spécificités au niveau de leur calcul par rapport aux canaux R,G et B. La Figure 3.16 présente une visualisation des espaces de couleur HSV et HSL. La représentation de l'espace HSI forme un bicone similaire à celui produit pour l'espace de couleur HSL.

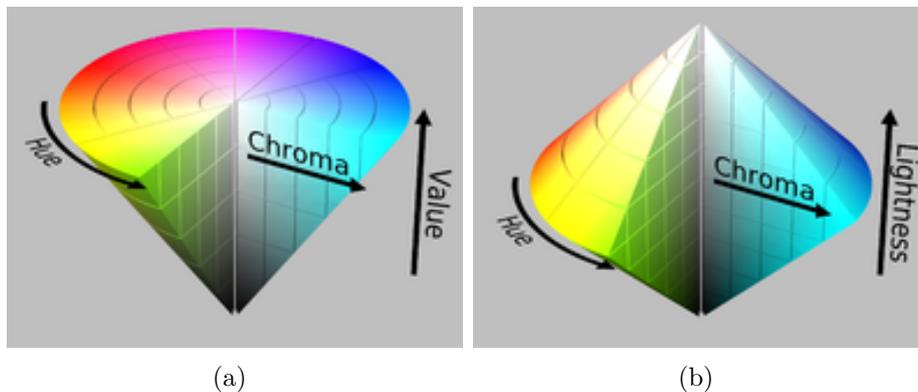


Figure 3.16: Représentation de l'espace de couleur (a) HSV et (b) HSL [173].

- **Les espaces de couleur utilisés pour la télévision**

Il existe de nombreux espaces de couleur adaptés à la visualisation d'images, notamment dans le domaine de la photographie ou de la télévision mais aussi de la compression et transmission. Parmi eux, on peut citer les espaces YUV, YCbCr, YPbPr et YDbDr où Y représente la luminosité, et les deux autres canaux donnent une information de chromaticité par rapport aux composantes bleu et rouge auxquelles on retire l'information de luminosité. L'espace de couleur YIQ est similaire à l'espace YUV, où le canal Y représente aussi la luminosité, mais les canaux I et Q forment un repère ayant un axe de rotation de  $33^\circ$  par rapport au repère formé par les canaux U et V. La Figure 3.17 montre une représentation de ces deux espaces.

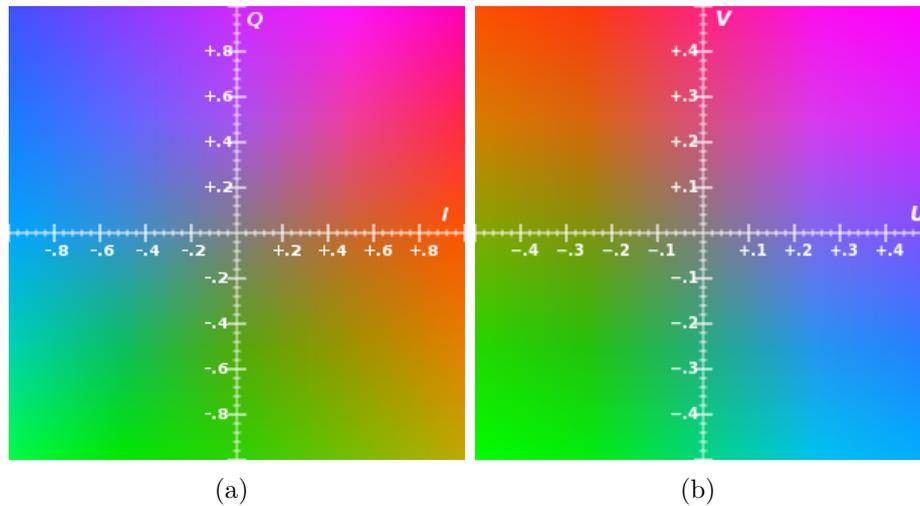


Figure 3.17: Représentation de l'espace de couleur (a) YIQ [174] et (b) YUV [175].

Il existe de nombreux autres systèmes de couleur que nous n'aborderons pas ici. Pour ce qui est de notre étude, nous avons décidé de réduire notre choix d'espaces à tester en fonction de l'intérêt qu'ils ont déjà suscité pour la détection de marquages et des fonctions de distance créées pour ces espaces.

### 3.4.3.b Présentation des espaces de couleur sélectionnés et des notations

Pour le reste de ce chapitre, dans un souci de lisibilité, nous utilisons une nomenclature pour toutes les méthodes de calcul de distance entre couleurs. Les nomenclatures habituelles sont référencées dans le Tableau 3.1. En complément, nous notons ' $\Delta E_{C_1 C_2 C_3}$ ' les distances euclidiennes calculées sur les trois différents canaux  $C_1$ ,  $C_2$  et  $C_3$  d'un espace de couleur donné et ' $\Delta E_{C_i}$ ', ou ' $\Delta E_{C_i C_j}$ ', les distances euclidiennes calculées sur un sous-ensemble des canaux. Par exemple, on notera  $\Delta E_{AB}$  la distance euclidienne calculée sur les canaux A et B de l'espace de couleur CIELAB, ou  $\Delta E_H$  celle calculée uniquement sur le canal H de l'espace de couleur HSV.

Pour sélectionner la ou les fonctions de distance qui discriminent le mieux la couleur des marquages au sol, nous étudions plusieurs couples (*EspaceCouleur*, *FonctionDistance*). On appelle 'différence de couleur' le résultat de la distance normalisée entre une couleur et la couleur de référence fournie par les

Espace de Couleur	Fonctions de Distance
HSV, HSL, XYZ, YCbCr, YIQ	Distance euclidienne sur les 3 canaux
	Distance euclidienne sur certains canaux
RGB	Distance euclidienne sur les 3 canaux
	Distance euclidienne pondérée sur les 3 canaux ( $\Delta C$ )
CIELAB	Distance euclidienne sur les 3 canaux ( $\Delta E^*_{ab}$ )
	Distance euclidienne sur certains canaux
L*C*h	Distance euclidienne sur les 3 canaux
	Distance euclidienne sur certains canaux
	CMC l:c 1984 ( $\Delta E^*_{CMC}$ )
	CIE 1994 ( $\Delta E^*_{94}$ )
	CIE 2000 ( $\Delta E^*_{00}$ )
	Distance euclidienne pondérée sur les 3 canaux ( $\Delta C_{LCH}$ )

Table 3.1: Liste des espaces de couleur et des fonctions de distance sélectionnées pour l'étude d'adaptation de la couleur de référence à partir d'une image. Les notations couramment utilisées sont indiquées entre parenthèses.

recommandations OACI pour un couple (*EspaceCouleur*, *FonctionDistance*) donné avec : *différence de couleur* =  $\|d_{i,E_j}(Ref, PixelColor)\|$ , où  $i$  est la fonction de calcul de distance et  $E_j$  est l'espace de couleur, sachant que certaines fonctions de distance ne sont définies que pour des espaces de couleur particuliers. Les distances obtenues sont normalisées en divisant la différence de couleur obtenue par la plus grande différence de couleur existante pour ce couple (*EspaceCouleur*, *FonctionDistance*). Pour certains couples, le couple est équivalent à la fonction de distance puisque celle-ci n'est définie que sur un espace de couleur précis, comme par exemple  $\Delta E^*_{CMC}$ .

### 3.4.3.c Construction de la base de données

Afin d'analyser les différents couples obtenus, nous avons créé une base de données de couleur constituée de triplets {R,G,B} uniques en utilisant des régions sélectionnées dans quelques images obtenues de caméras réelles et quelques images provenant du simulateur d'aéroport. Nous travaillons avec huit images de taille 1280pX960p, quatre images viennent de caméras placées sur un avion et quatre images viennent du simulateur. La moitié des images réelles et simulées viennent d'une caméra placée proche du cockpit de l'avion et l'autre moitié viennent d'une caméra placée dans la dérive d'un avion. Parmi ces huit images, trois représentent des conditions météorologiques dégradées (brouillard, tombée de la nuit, faible luminosité). Même si elles sont peu nombreuses, les images ont été choisies pour être représentatives de plusieurs cas : différentes zones du taxiway, différents motifs de marquages au sol, plus ou moins complexes et plus ou moins effacés.

Nous proposons dans un premier temps d'analyser les nuances de couleurs présentes dans les images sélectionnées. Les images peuvent être séparées en zones comme par exemple le taxiway (qui contient la plupart des pixels gris de l'image), le ciel (où l'on retrouve la majorité des pixels bleus ou de couleur claire) et l'herbe autour du taxiway. La plupart des couleurs rencontrées dans les images peuvent être simplifiées comme appartenant à un ensemble limité de classes : 'Bleu', 'Gris',

'Vert', 'Rouge', 'Noir' (ou couleurs sombres), 'Blanc' (ou couleurs claire) ou 'Jaune' (pour les marquages au sol notamment). Par la suite, on analysera la performance de chaque couple (*EspaceCouleur*, *FonctionDistance*) dans la discrimination de la couleur des marquages au sol par rapport aux autres classes.

Comme la zone d'herbe autour du tarmac est composée d'un grand nombre de teintes de couleurs différentes (notamment vertes, jaunes ou marrons), nous avons définies deux sous-classes de couleur dans notre base de données; appelées 'Vert' et 'Herbe'. La Figure 3.18 montre un exemple d'image utilisée pour créer la base de données et des masques correspondant à chaque classe de la base de données. Ces masques ont été annotés à la main.

La Figure 3.19 présente les classes qui ont été choisies pour l'étude ainsi que leur représentativité dans la base de données et dans les images utilisées pour construire cette dernière. Elles diffèrent puisque la base de données est composée de triplets uniques que l'on peut retrouver plusieurs fois dans une image. Les couleurs sont ordonnées par leurs valeurs en R puis G et B. La classe 'Non Défini' représente les pixels RGB qui ont été assignés à au moins 2 classes, suivant les images. On peut voir aux différentes nuances de teintes des différentes classes ainsi qu'à la classe 'Non Défini' qu'il est difficile d'obtenir une classe parfaitement épurée avec cette méthode mais considérons que dans l'ensemble la base de données est assez correcte pour obtenir des résultats significatifs.

#### 3.4.3.d Analyse statistique

Nous allons faire une analyse statistique des différences entre la couleur de référence OACI pour le jaune des marquages, avec toutes les couleurs enregistrées dans notre base, cela pour tous les couples (*EspaceCouleur*, *FonctionDistance*). Nous avons décidé d'utiliser plusieurs critères dans l'optique de sélectionner un sous-ensemble de couples candidats qui permettent une bonne discrimination de la classe 'Jaune' par rapport aux autres classes, ce qui devrait améliorer la détection de marquages. Ces différents critères sont :

- La précision pour un rappel de 100%, cela représente la proportion des différence de couleurs de la classe 'Jaune' par rapport à toutes les différences de couleurs inférieures ou égales à la différence de couleur maximale présente dans la classe 'Jaune'.
- La précision pour un rappel de 95%.
- Le rappel pour une précision de 95%.
- Le  $F_1$  score.
- La dispersion des différences de couleur de chaque classe, facilement représentée par des boites à moustaches.

avec

$$Précision = \frac{Vrais\ Positifs}{Vrais\ Positifs + FauxPositifs}.$$

$$Rappel = \frac{Vrais\ Positifs}{Vrais\ Positifs + FauxNégatifs}.$$

$$F_1score = 2 * \frac{Précision * Rappel}{Précision + Rappel}.$$

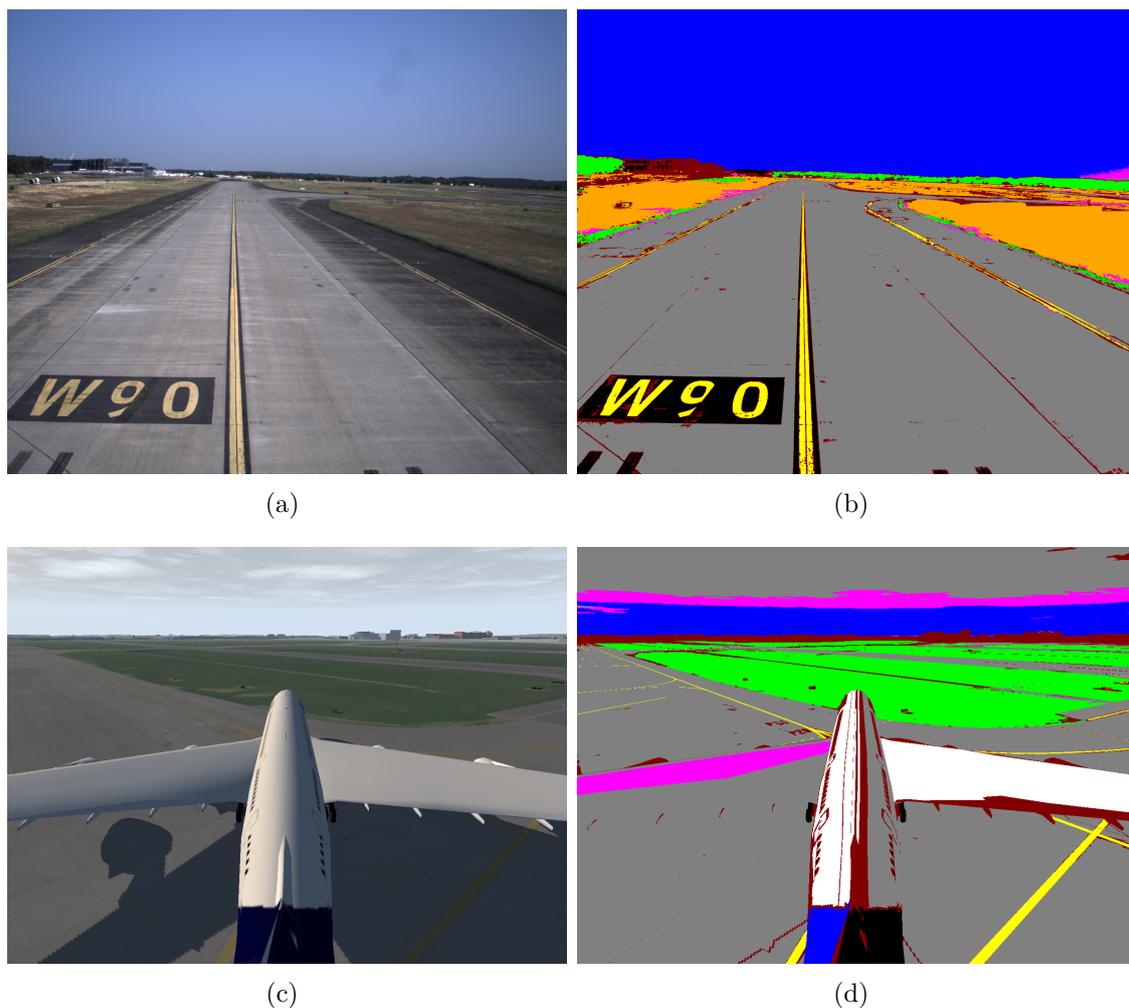


Figure 3.18: Exemple de création de la base de données depuis des images réelles et simulées. Chaque masque de couleur correspond à une classe, avec la couleur marron pour les pixels non labellisés, la couleur magenta pour les pixels de la classe 'Non Défini' et la couleur orange pour la classe 'Herbe'. Un exemple est donné pour (a) une image acquise en conditions réelles avec (b) le masque associé et pour (c) une image issue du simulateur avec (d) le masque associé.

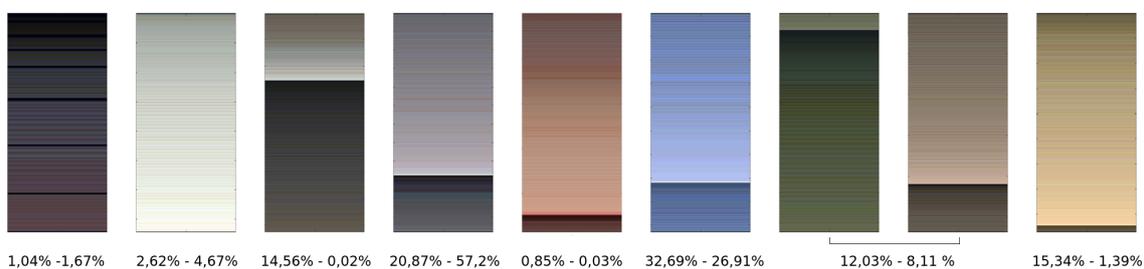


Figure 3.19: Classes 'Sombre', 'Clair', 'Non Défini', 'Gris', 'Rouge', 'Bleu', 'Vert', 'Herbe' et 'Jaune' de la base de données. En dessous, on retrouve les pourcentages de représentativité des classes sous la forme : représentativité dans la base de données - représentativité dans les images.

Les trois premiers critères informent sur la répartition des différences de couleur calculées pour un couple (*EspaceCouleur*, *FonctionDistance*). En étudiant les histogrammes de répartition des différences de couleur par classe, on peut déjà estimer visuellement quels couples peuvent ou non décrire une différence significative entre la classe 'Jaune' et les autres classes. La Figure 3.20 représente un exemple d'une répartition possible des différences de couleur obtenues pour un couple hypothétique. Quatre valeurs seuil sont définies pour schématiser les zones pour lesquelles on obtient une précision de 100%, une précision de 95%, un rappel de 95% et un rappel de 100%. On constate que pour qu'un couple soit discriminant de la classe 'Jaune', il est important que les valeurs seuil de précision à 100% et de rappel à 100% soient les plus proches possible, c'est à dire que l'on obtienne un  $F_1$  score le plus grand possible.

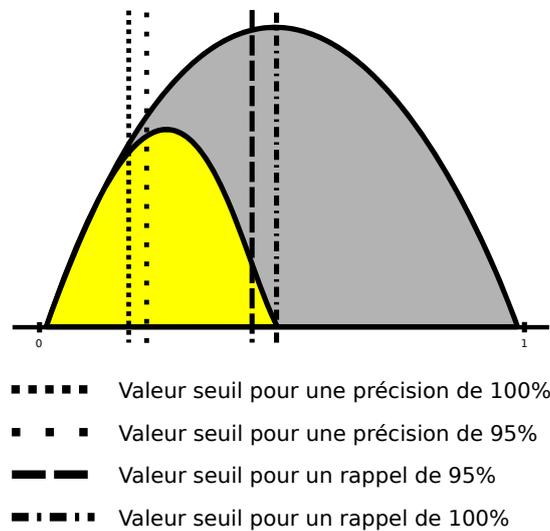


Figure 3.20: Exemple de répartition des différences de couleur d'un couple (*EspaceCouleur*, *FonctionDistance*) pour la classe 'Jaune' (en jaune) par rapport à l'ensemble des différences de couleur (en gris).

La Figure 3.21 présente les résultats de répartition des différences de couleur obtenues par étude de la base de données pour certains couples. Des résultats complémentaires sont disponibles en annexe, représentant la répartition des différences de couleur de chacune des classes pour les différents couples testés. On peut constater que les profils de répartition sont très variés. On peut voir en Figure 3.21 que les couples  $\Delta E^*_{CMC}$  et  $\Delta E_H$  (HSV/HSL) présentent des répartitions des différences de couleur localisées pour le 'Jaune' alors que pour les couples  $\Delta C$  et  $\Delta E_{XYZ}$  on observe un premier pic localisé puis une répartition étendue sur une grande partie du spectre de répartition des valeurs.

Le Tableau 3.2 présente les résultats de différents couples, basés sur ces critères. Les notations +, = et - sont définies par des valeurs obtenues pour ces critères dans les bornes respectivement :

- [100%; 75%], ]75%; 50%] et ]50%; 0%] pour les lignes *Précision pour Rap<sub>95%</sub>* et *Rappel pour Préc<sub>95%</sub>*;
- [100%; 25%], ]25%; 17%] et ]17%; 0%] pour la ligne *Précision pour Rap<sub>100%</sub>*;
- [100%; 80%], ]80%; 60%] et ]60%; 0%] pour la ligne *F<sub>1</sub> score*.

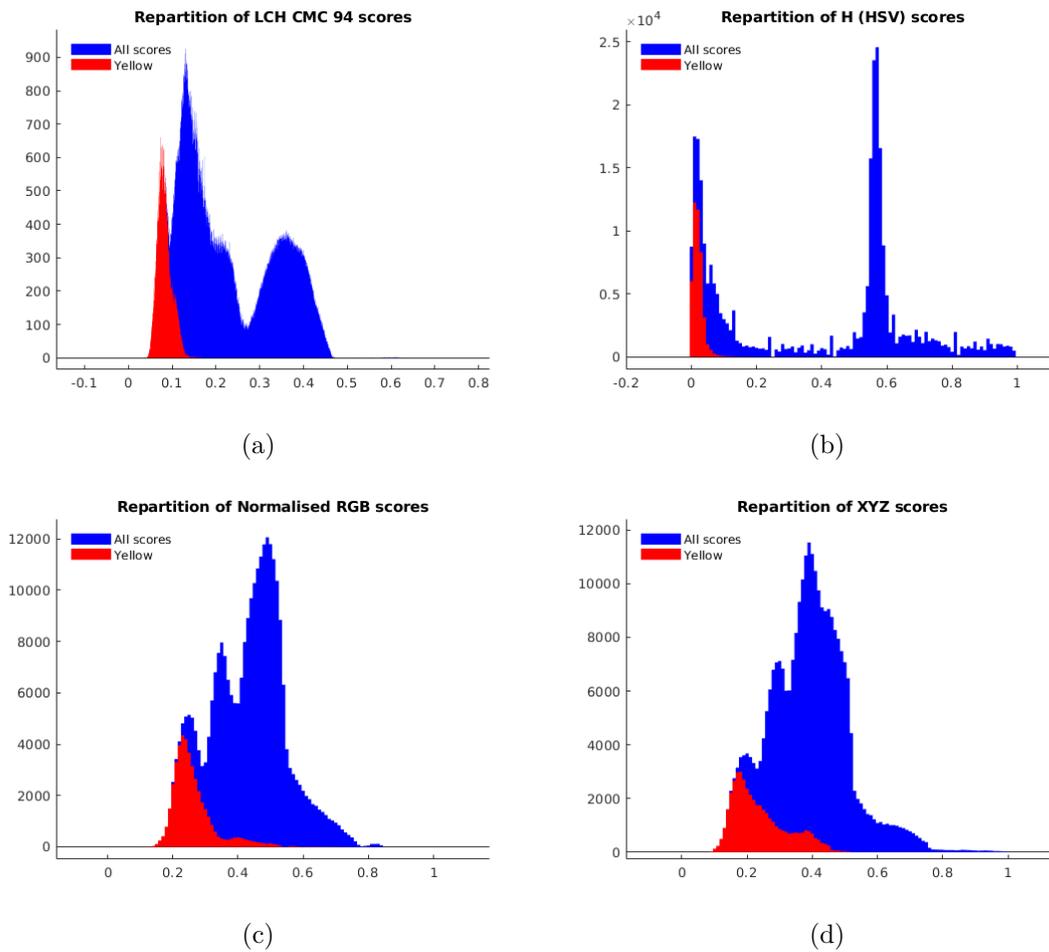


Figure 3.21: Représentation des différences de couleur d'un couple (*EspaceCouleur*, *FonctionDistance*) pour la classe 'Jaune' (en rouge) par rapport à l'ensemble des différences de couleur (en bleu) pour (a)  $\Delta E^*_{CMC}$ , (b)  $\Delta E_H$ , (c)  $\Delta C$  et (d)  $\Delta E_{XYZ}$ .

En suivant le Tableau 3.1, nous avons comparé trente sept paires mais n'en présentons que dix-huit dans le Tableau 3.2. Les couples présentés ont obtenu au moins un + à l'un des critères, à l'exception de  $\Delta E_H$  que nous avons conservé, car ce couple obtient de bons résultats sur un autre critère (boîtes à moustaches (voir Figure 3.22), et aussi parce qu'il est aussi souvent utilisé pour la détection de lignes dans la littérature.

En se basant sur les résultats du Tableau 3.2, nous pouvons observer trois groupes de méthodes : celles qui obtiennent en majorité des + aux critères (comme  $\Delta E_{AB}$  ou  $\Delta E^*_{CMC}$ ), celles qui obtiennent un mélange de + et de = (comme  $\Delta E_{IQ}$  ou  $\Delta E^*_{ab}$ ) et celles qui obtiennent peu de + mais plutôt des = ou des - comme  $\Delta E^*_{94}$  ou  $\Delta E_{HSL}$ ).

Dans le Tableau 3.2, nous avons rajouté une ligne *Dispersion* : la valeur + est attribuée aux couples qui assurent qu'au moins 75% des différences de couleur de la classe 'Jaune' ne peuvent pas être confondues avec une autre classe, la valeur = signifie que les paires pourraient entraîner une confusion entre la classe 'Jaune' et les classes 'Gris', 'Rouge', 'Vert' ou 'Herbe' et la valeur - est donnée aux couples qui ne peuvent pas être utilisés pour séparer la classe 'Jaune' des autres classes.

	$\Delta E^*_{ab}$	$\Delta E_{AB}$	$\Delta E_B$	$\Delta E^*_{94}$	$\Delta E^*_{00}$	$\Delta E_{LCH}$	$\Delta C_{LCH}$	$\Delta E_{CH}$	$\Delta E^*_{CMC}$
Précision pour $Rap_{100\%}$	=	+	+	+	-	-	+	-	+
Précision pour $Rap_{95\%}$	=	+	+	-	-	=	=	+	+
Rappel pour $Préc_{95\%}$	+	+	+	-	+	+	+	+	+
F <sub>1</sub> Score	+	+	+	=	+	+	+	+	+
Dispersion	+	+	+	=	+	=	=	=	+

	$\Delta E_{HSV}$	$\Delta E_{HSL}$	$\Delta E_H(\text{HSL/HSV})$	$\Delta E_{YCbCr}$	$\Delta E_{CbCr}$	$\Delta E_{Cb}$	$\Delta E_{YIQ}$	$\Delta E_{IQ}$	$\Delta E_I$
Précision pour $Rap_{100\%}$	-	-	-	-	+	+	-	+	-
Précision pour $Rap_{95\%}$	-	-	=	-	+	+	-	=	=
Rappel pour $Préc_{95\%}$	=	=	-	=	+	+	=	=	-
F <sub>1</sub> Score	+	+	=	+	+	+	+	+	+
Dispersion	-	-	=	-	=	+	-	=	-

Table 3.2: Tableau comparatif des performances de quelques espaces de couleur et fonctions de distance.

La Figure 3.22 offre un sous-ensemble des répartitions des différences de couleurs calculées pour les différentes classes en fonction des différents couples (*EspaceCouleur, FonctionDistance*). Le carré bleu correspond aux 25ième et 75ième centiles et la ligne rouge représente la médiane des différences de couleur. Les adjacents inférieurs et supérieurs (valeurs minimum et maximum non considérées comme des valeurs aberrantes) sont indiqués par les moustaches noires et les croix rouges sont les valeurs aberrantes, ayant des valeurs correspondant à plus de 1,5 fois la différence interquartile. Les lignes en tirets et en pointillées représentent les valeurs seuil atteintes pour l'adjacent supérieur et le 75ième centile de la classe 'Jaune'.

Nous pouvons voir dans la Figure 3.22 quelques couples (par exemple  $\Delta E_{XYZ}$  ou  $\Delta E_{RGB}$ ) qui ne peuvent pas être utilisés pour discriminer la classe 'Jaune' des autres classes. Il en est de même pour toutes les fonctions de distance appliquées à l'espace de couleur XYZ qui est l'espace dans lequel la référence de couleur pour les marquages au sol nous est fournie.

Cela appuie notre besoin de trouver une méthode qui permette de mieux discriminer la référence de couleur et les teintes de la classe 'Jaune' des autres couleurs de l'image. Certaines paires semblent permettre de séparer la classe 'Jaune' des autres classes, comme par exemple  $\Delta E_{CbCr}$ ,  $\Delta E^*_{ab}$  ou  $\Delta C_{LCH}$  puisque la plupart des classes obtiennent un grand nombre de différences de couleurs supérieurs à la valeur du 75ième centile de la classe 'Jaune'. Nous notons que l'amplitude des différences de couleurs varie grandement selon les paires étudiées, certaines ayant des valeurs très rapprochées pour l'ensemble de la classe 'Jaune'.

On note cependant que les valeurs des différences de couleurs des teintes de la classe 'Jaune' sont très souvent suivies par celles des classes 'Rouge' et 'Herbe'. Pour la détection de marquages, cela pourrait poser un problème puisque certains marquages sur le tarmac sont de couleur rouge et possèdent un sens complètement différent de celui des marquages jaunes.  $\Delta E_H$  (sur les espaces de couleur HSV ou HSL) possède une petite gamme pour les valeurs des différences de couleurs des classes 'Jaune', 'Rouge', 'Herbe' et 'Bleu' et peut permettre de séparer la classe 'Jaune' des autres classes. Cependant, on peut constater un grand nombre de valeurs aberrantes pour les différentes classes avec cette méthode. Ces résultats sont cohérents aux vues des nombreuses nuances de teintes de chaque classe, comme on peut le voir sur la Figure 3.19.

En se basant sur les résultats du Tableau 3.2 (nous sélectionnons les couples qui obtiennent majoritairement des + pour les différents critères) et de la Figure 3.22,

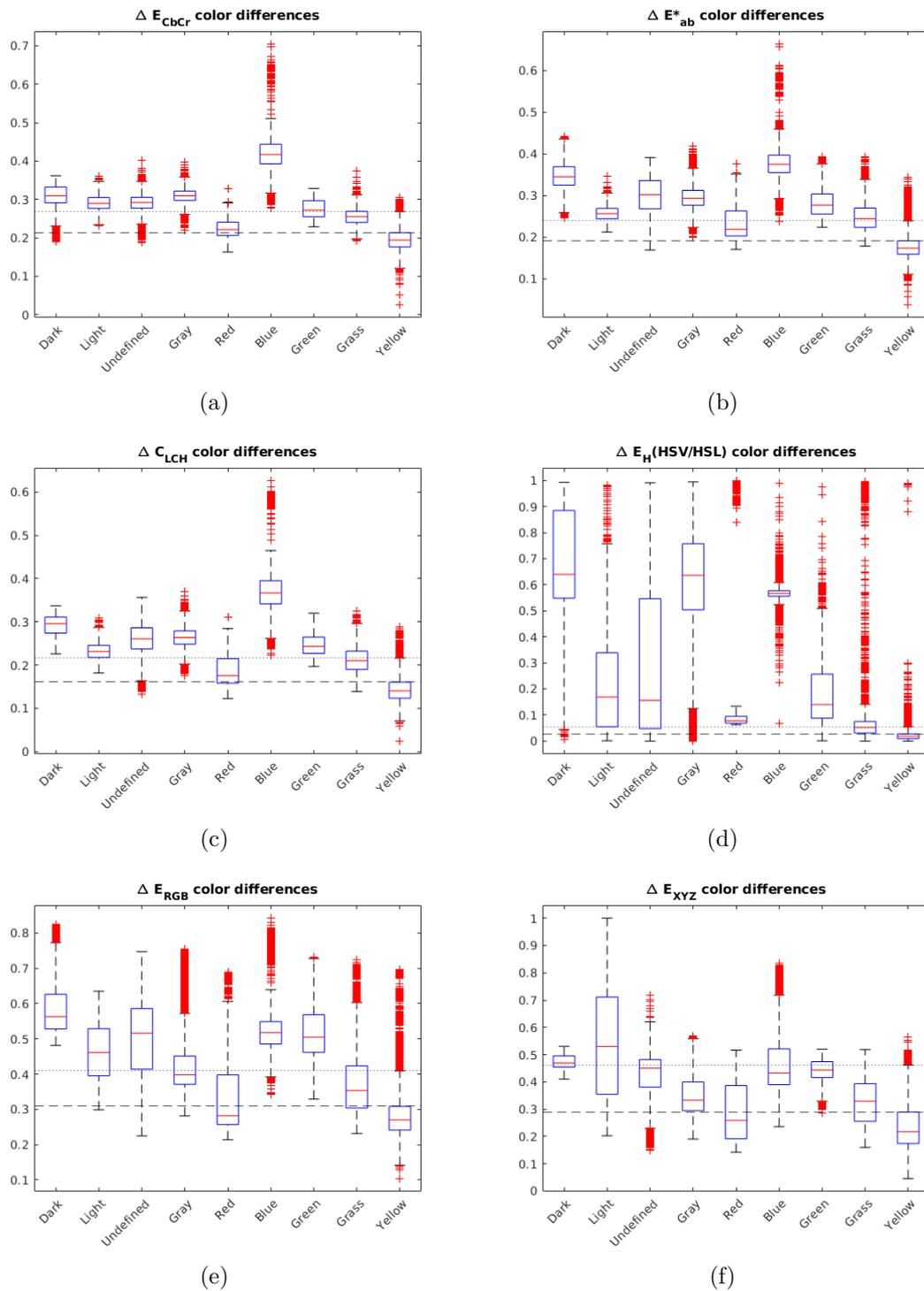


Figure 3.22: Boîtes à moustaches représentant la distribution des différences de couleur par classe pour quelques couples (*EspaceCouleur*, *FonctionDistance*).

nous avons réduit les nombreuses paires à un sous-ensemble de couples :

- $\Delta E^*_{ab}$ ,  $\Delta E_{AB}$ ,  $\Delta E_B$  et  $\Delta E^*_{00}$
- $\Delta E_{LCH}$ ,  $\Delta E_{CH}$ ,  $\Delta E^*_{CMC}$  et  $\Delta C_{LCH}$
- $\Delta E_{CbCr}$ ,  $\Delta E_{Cb}$ ,  $\Delta E_{IQ}$ , et  $\Delta E_H$  (appliquée indifféremment à l'espace de couleur HSV ou HSL)

Dans la section suivante nous décrivons comment cette étude colorimétrique nous permet de fixer les paramètres de l'algorithme présenté dans la Section 3.4.2.b puis nous validerons les performances de chaque couple à singulariser les pixels jaunes sur des images qui n'ont pas été utilisées pour créer la base de données dans la Section 3.4.5 afin de sélectionner le couple le plus adapté à notre application de détection de marquages jaunes dans les zones aéroportuaires.

### 3.4.4 Application à l'affinage adaptatif de la couleur de référence

#### 3.4.4.a Sélection des valeurs des paramètres de l'algorithme

L'Algorithme 2 possède trois paramètres à ajuster :  $p$ , *tolerance* et *thresh*. Nous proposons d'utiliser les résultats de l'analyse de couleur de la partie 3.4.3 pour fixer ces paramètres.

La Figure 3.19 montre que les couleurs des marquages au sol ne représentent qu'en moyenne 1,39% des pixels d'une image. Nous avons donc décidé de fixer  $p$  comme valant 0.1% de ce ratio moyen. Ce pourcentage nous permet de sélectionner assez de pixels pour calculer une nouvelle couleur de référence tout en minimisant la probabilité de sélectionner des valeurs aberrantes d'autres classes, pouvant se confondre avec le jaune et dégrader notre couleur de référence.

Le but du paramètre *thresh* est de constituer une carte binaire pour les algorithmes de détection de marquages. Pour choisir la valeur de ce paramètre, nous avons utilisé l'étude de couleur et notamment les critères de précision, rappel et de  $F_1$  score des couples (*EspaceCouleur*, *FonctionDistance*). Une première idée était d'utiliser la valeur seuil pour laquelle on obtient le meilleur  $F_1$  score dans l'analyse de la base de données. La meilleure paire devrait avoir un  $F_1$  score qui tend vers 1.

Bien que cette valeur seuil maximise le compromis entre précision et rappel, elle ajoute des valeurs aberrantes à la sélection. Dans ce contexte spécifique, nous avons décidé de plutôt utiliser la valeur seuil qui correspond à une perte de précision par rapport à la précision maximale. On choisit de prioriser la précision par rapport au rappel, pour minimiser le nombre de valeurs aberrantes sélectionnées. Toutefois, cela amène aussi à une augmentation potentielle de faux négatifs au niveau des pixels représentant des marquages au sol dans l'image.

La carte binaire obtenue en appliquant un seuillage à la carte de distance est aussi utilisée pour analyser le nombre de pixels représentant possiblement des marquages dans l'image. Si le nombre de pixels blancs présents dans la carte binaire (c'est à dire représentant de possible marquages) est inférieur à  $p$ , l'algorithme pourra difficilement converger vers une référence de couleur significative.

Comme nous utilisons la détection de ligne pour appliquer des algorithmes de compréhension de scène et d'estimation de position et déplacement de l'avion, nous avons besoin d'un nombre substantiel de pixels représentant des marquages au sol dans l'image pour obtenir des résultats significatifs. C'est pourquoi nous avons décidé de fixer un nombre minimum de pixels blancs dans la carte binaire en se basant sur une valeur de 5% du pourcentage de représentativité moyen de la classe 'Jaune' dans une image. Si nous obtenons moins de pixels blancs que cette valeur, nous rejetons l'image et effectuons les traitements sur la prochaine.

L'affinage de la couleur de référence est effectué tant que la distance entre l'ancienne et la nouvelle couleur de référence est supérieure au paramètre *tolerance*. La tolérance peut être définie pour chaque couple (*EspaceCouleur*, *FonctionDistance*). Dans l'espace de couleur XYZ, la tolérance est définie par les ellipses de MacAdam [110]. Autrement, une valeur seuil peut être utilisée. Par exemple, on peut utiliser la valeur représentant la plus petite valeur de différences de couleurs de la classe 'Jaune' dans notre base de données.

Ce paramètre permet d'effectuer plusieurs fois le processus d'affinage de la couleur de référence pour converger vers une couleur de référence spécifique à chaque image, tout en utilisant des paramètres choisis à la suite d'une étude sur un nombre d'images obtenues dans des conditions environnementales variées. Nous avons remarqué que dans notre cas d'application, il suffit de trois boucles algorithmiques avant d'obtenir une couleur de référence satisfaisante. Au dessus de cela, nous observons que l'algorithme a convergé : la valeur de la nouvelle couleur de référence n'évolue plus.

#### 3.4.4.b Extension: utilisation d'un masque du tarmac

En utilisant les résultats de la partie 3.4.3, on peut voir sur la Figure 3.22 que le calcul de distance sur le canal H dans les espaces de couleur HSV ou HSL n'est pas suffisant pour discriminer la couleur jaune de certaines valeurs aberrantes des classes 'Gris' et 'Non Défini' mais est assez efficace par rapport aux classes 'Bleu' et 'Vert'. Dans notre cas d'application, on sait que les marquages à détecter sont situés sur le taxiway qui est en majorité de couleur grise. On peut donc appliquer une transformation de notre image dans l'espace HSV et une méthode de seuillage pour calculer un masque sur l'image. On réduit ainsi la complexité de cette dernière en retirant des pixels appartenant à d'autres classes comme notamment le ciel ou l'herbe verte. On peut sélectionner la valeur seuil de la même manière que pour le paramètre *thresh*.

La Figure 3.23 donne un exemple d'application d'un masque sur une image réelle et sur une image simulée. Sur l'image simulée, on observe que le masque laisse passer des pixels de couleur gris clair ainsi que les pixels jaunes alors qu'il masque les autres zones de l'image. Pour les images réelles, où le tarmac est d'une couleur plus sombre, le masquage par le canal H offre une meilleure réduction de la complexité du bruit puisqu'il filtre aussi ces pixels. Notons que l'avion pourrait être facilement masqué dans les images FIN CAM.

Comme l'on utilise par la suite la transformation IPM, on pourrait être tenté de l'utiliser pour se concentrer sur les points au sol et donc segmenter le tarmac dans l'image tout en réduisant la probabilité de sélectionner des valeurs aberrantes au dessus du sol (par exemple la peinture d'un avion ou la lumière du soleil). Cependant, l'IPM apporte beaucoup de bruit à l'image et pollue un peu les résultats. Cette méthode est pourtant couramment utilisée dans la littérature pour segmenter la route dans le domaine automobile.

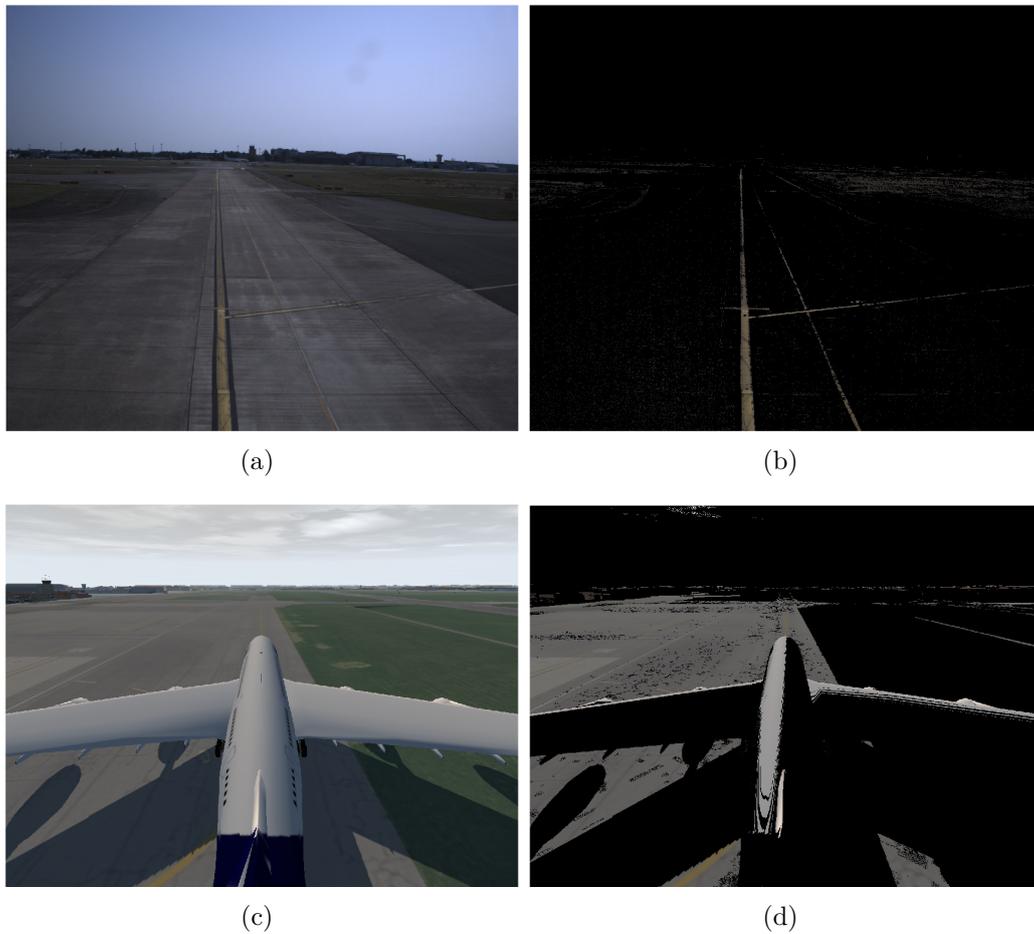


Figure 3.23: Résultats du masquage pour une image réelle (a-b) et pour une image simulée (c-d).

### 3.4.5 Validation sur des images de test

#### 3.4.5.a Étude quantitative des couples candidats

Grâce à l'analyse statistique de la base de données, nous avons pu pré-sélectionner quelques couples candidats pour l'espace de couleur et la méthode de calcul de distance utilisés dans l'Algorithme 2. Nous avons décidé de générer les cartes de distance et les cartes binaires pour chaque paire (*EspaceCouleur*, *FonctionDistance*) candidate sur des images réelles et simulées n'ayant pas été utilisées pour former la base de données, sous différentes conditions climatiques.

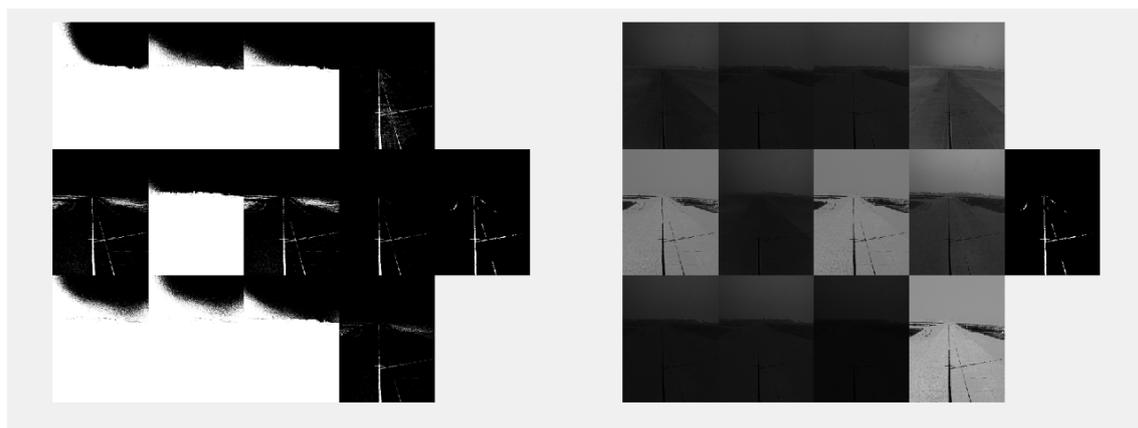
La comparaison des différents couples est effectuée au moyen de huit images, sélectionnées avec les mêmes proportions que pour les huit images utilisées précédemment pour la création de la base de données. Ce nombre d'images dans la base test est très faible, mais la validation nécessite une annotation des marquages, annotations faites "à la main", ce qui est assez fastidieux.

La Figure 3.24 présente les cartes binaires et les cartes de distance obtenues pour différents couples (*EspaceCouleur*, *FonctionDistance*) pour deux images ainsi qu'une comparaison avec la vérité terrain (dernière image de la seconde ligne). La carte binaire est calculée en effectuant un seuillage par le paramètre *thresh* de la carte de distance, comme expliqué dans la partie 3.4.4.

Les résultats obtenus pour  $\Delta E^*_{CMC}$ ,  $\Delta E^*_{00}$ ,  $\Delta E_{LCH}$ ,  $\Delta E^*_{CH}$  et  $\Delta E^*_H$  (sur HSV ou HSL) ressortent par rapport aux autres cartes. De manière générale, n'importe quelle fonction de distance entre couleurs effectuées dans l'espace de couleur L\*C\*h fournit des cartes binaires exploitables. Pour les autres cartes binaires, on observe que la valeur seuil obtenue par l'étude colorimétrique ne permet pas d'obtenir une carte satisfaisante.

	$\Delta E^*_{ab}$	$\Delta E_{AB}$	$\Delta E_B$	$\Delta E^*_{00}$	$\Delta E_{LCH}$	$\Delta C_{LCH}$	$\Delta E_{CH}$	$\Delta E^*_{CMC}$	$\Delta E_{CbCr}$	$\Delta E_{Cb}$	$\Delta E_{IQ}$	$\Delta E_H$
Précision	-	-	-	=	=	-	-	+	-	-	-	+
Rappel	=	+	+	-	=	=	=	=	+	+	+	-
F <sub>1</sub> score	-	-	-	+	+	=	=	+	-	-	-	-
Exactitude	=	-	-	+	+	=	+	+	-	-	-	+
Temps de calcul	=	=	=	=	=	=	=	-	+	+	+	+

Table 3.3: Tableau comparatif de plusieurs espaces de couleur et fonctions de distance sur les images de tests (non-utilisées pour la base de données).



(a)



(b)

Figure 3.24: Cartes binaires (gauche) et cartes de distance (droite) pour deux images réelles provenant du cockpit (a) et de la dérive (b) d'un avion. Première ligne :  $\Delta E^*_{ab}$ ,  $\Delta E_{AB}$ ,  $\Delta E_B$ ,  $\Delta E^*_{00}$ . Seconde ligne :  $\Delta E_{LCH}$ ,  $\Delta C_{LCH}$ ,  $\Delta E_{CH}$ ,  $\Delta E^*_{CMC}$ , vérité terrain. Troisième ligne :  $\Delta E_{CbCr}$ ,  $\Delta E_{Cb}$ ,  $\Delta E_{IQ}$ ,  $\Delta E_H$  (HSV/HSL). Cet ordre est conservé pour les quatre blocs.

Le Tableau 3.3 apporte des résultats quantitatifs sur la précision, le rappel, le  $F_1$  score et l'exactitude des cartes binaires comparées à la vérité terrain, avec une information donnée sur le temps de calcul nécessaire à l'obtention des cartes. La formule du calcul est la suivante, avec  $VP$  les vrais positifs,  $VN$  les vrais négatifs,  $FP$  les faux positifs et  $FN$  les faux négatifs :

$$Exactitude = \frac{VP + VN}{VP + VN + FP + FN}$$

Ces résultats représentent une valeur moyenne des résultats pour les huit images. Les scores +, = et - définissent des valeurs dans les intervalles respectivement  $[100\%; 75\%]$ ,  $]75\%; 50\%]$  et  $]50\%; 0\%]$  pour les lignes *Rappel* et *Exactitude* et  $[100\%; 25\%]$ ,  $]25\%; 10\%]$  et  $]10\%; 0\%]$  pour les lignes *Précision* et  $F_1$  score.

On peut former plusieurs groupes selon les résultats des couples dans le Tableau 3.3.  $\Delta E_{*CMC}$  et  $\Delta E_{LCH}$  montrent de bons résultats sur les différents critères d'évaluation.  $\Delta E_{*00}$ ,  $\Delta C_{LCH}$ ,  $\Delta E_{CH}$  et  $\Delta E_H$  obtiennent des résultats moyens alors que les résultats des autres méthodes ne sont pas satisfaisants. Cependant, le temps de calcul peut être un facteur important selon l'application. Dans notre cas, il est juste indicatif puisque, pour cette étude, les différentes fonctions présentes dans l'algorithme ne sont pas optimisées.

Comme attendu, les méthodes à faibles temps de calcul présentent dans l'ensemble de moins bons résultats que les méthodes plus complexes.  $\Delta E_{*CMC}$  s'impose comme le meilleur choix si le temps de calcul n'est pas une donnée limitante mais pour des applications temps réel,  $\Delta E_{LCH}$  est un bon substitut.  $\Delta E_H$  pourrait être sélectionnée pour son temps de calcul rapide même si ses résultats sont un peu moins probants. On peut aussi imaginer d'effectuer la ré-évaluation de la référence de couleur toutes les secondes et non toutes les images dans une application qui tournerait en temps réel (30 ou 60 images par seconde), dans le but de réduire l'impact du temps de calcul tout en conservant une référence de couleur correcte.

### 3.4.5.b Robustesse au bruit

Lors de l'analyse des différences de couleurs entre la couleur de référence et les couleurs de chaque classe définie dans la base de données, nous avons estimé la précision des couples (*EspaceCouleur*, *FonctionDistance*) pour discriminer la classe 'Jaune'. Cependant, notre base de données présente un biais de représentativité puisque la proportion de chaque classe dans la base est différente de sa proportion dans l'image.

Ceci est dû à notre construction de la base de données, qui est composée de triplets de couleur uniques. Cela veut dire que la couleur d'un pixel correspondant à une valeur aberrante de la classe 'Gris' peut être considérée comme appartenant à la classe 'Jaune', par exemple, sans avoir de grand impact sur les résultats de l'analyse statistique qui exploite cette base de données. Au contraire, ce triplet unique pourrait représenter une grande partie du taxiway dans l'image et modifier les performances entre l'étude de la base de données et les résultats des paires candidates présentés ci-dessus.

Cela explique pourquoi certains couples obtiennent de bons résultats lors de l'étude colorimétrique tandis que leurs résultats sur l'étude des cartes binaires et des cartes de distance ne sont pas satisfaisants. Nous pourrions ajouter un poids à chaque triplet dans la base de données pour corriger ce biais ou rompre l'unicité

des triplets. Cependant, nous avons décidé de garder ce biais de la base de données et de réaliser une étude de robustesse au bruit de chaque paire sur l'ensemble des images de test.

Cette étude est menée sur des images réelles uniquement. Pour chaque image, nous ajoutons un bruit blanc gaussien et utilisons la définition du rapport signal à bruit (SNR) suivante :

$$SNR = \frac{mean(image)}{\sigma_{noise}}.$$

Nous appliquons ensuite l'Algorithme 2 et obtenons les cartes de distance. Pour chaque carte de distance, nous calculons la meilleure carte binaire obtenue en effectuant un seuillage de la carte de distance avec toutes les valeurs seuil possibles. Nous sélectionnons la valeur seuil pour laquelle on maximise le  $F_1$  score entre chaque carte binaire et la vérité terrain. Enfin, nous comparons le meilleur  $F_1$  score avec le  $F_1$  score obtenu par seuillage de la carte binaire avec le paramètre *thresh*, sélectionné dans les parties 3.4.3 et 3.4.4, comme valeur seuil.

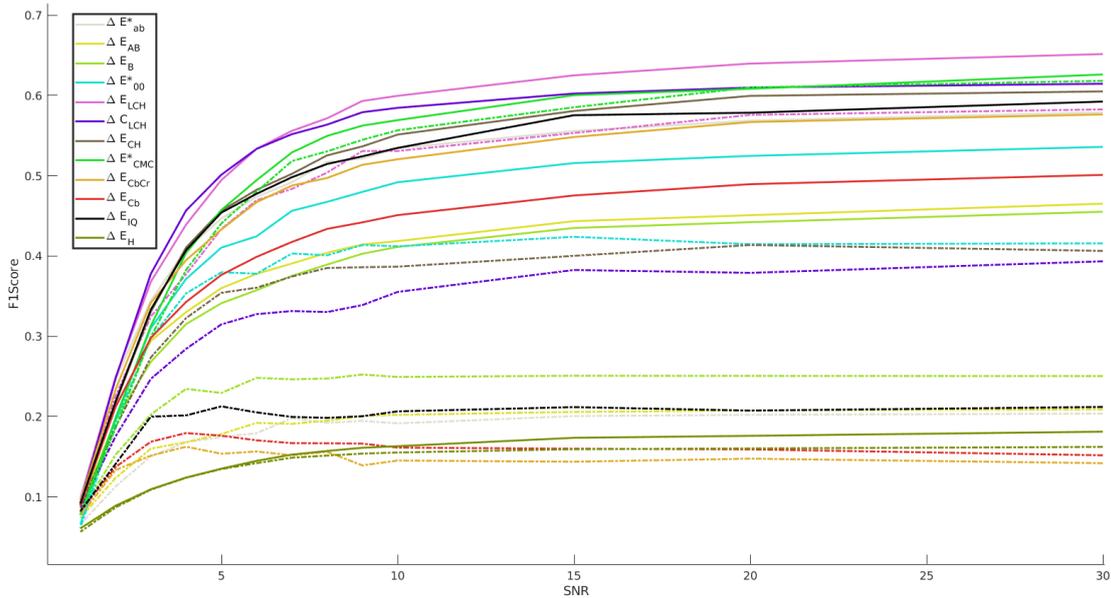


Figure 3.25: Impact du SNR sur le  $F_1$  score pour différentes paires candidates. Les lignes pleines représentent les résultats obtenus pour des valeurs seuil idéales tandis que les lignes en pointillés correspondent à ceux obtenus en utilisant la valeur seuil estimée par l'étude colorimétrique.

Les résultats de comparaison de ces valeurs moyennes de  $F_1$  score sur les images testées sont présentés dans la Figure 3.25. On remarque que tous les couples (*EspaceCouleur*, *FonctionDistance*) possèdent un profil similaire face à l'évolution du niveau de bruit dans l'image. Le SNR n'a pas d'impact direct sur le choix de la méthode retenue. Les courbes présentées par la Figure 3.25 montrent que les meilleurs résultats théoriques sont obtenus en utilisant  $\Delta E^*_{LCH}$ , suivi de  $\Delta C_{LCH}$ ,  $\Delta E^*_{CMC}$  et  $\Delta E^*_{ab}$ . En comparaison, les scores obtenus avec les valeurs seuil estimées (lignes en pointillés dans la Figure 3.25) sont bien plus faibles que les résultats théoriques, à l'exception de  $\Delta E^*_{CMC}$  et  $\Delta E^*_{LCH}$  où l'on peut considérer que l'étude colorimétrique suffit à trouver une valeur seuil qui, en moyenne, offre de bons résultats quelque soit l'image.

En s'appuyant sur les résultats de la Figure 3.25 et du Tableau 3.3, nous avons décidé de sélectionner la fonction de calcul de distance L\*a\*b\* CMC l:c 84 ( $\Delta E^*_{CMC}$ ) pour mesurer les différences de couleurs. On peut noter que le couple CIELAB CIEDE2000 ( $\Delta E^*_{00}$ ) offre aussi de bons résultats pour l'obtention de la carte binaire. Il permet aussi d'obtenir une carte de distance plus contrastée, entre les pixels correspondant aux marquages au sol et les autres pixels de l'image, que la méthode L\*a\*b\* CMC l:c 84. Cependant, cette méthode est moins robuste sur nos différentes images et semble moins s'adapter aux différences d'environnement. Le contraste de la carte de distance peut, quant à lui, être augmenté en appliquant une fonction exponentielle à la carte de distance obtenue par l'algorithme d'affinage de la couleur de référence.

### 3.4.6 Validation de l'intérêt de la méthode pour des algorithmes de détection de lignes

Dans l'optique de valider l'intérêt d'une méthode pour des algorithmes de détection de lignes, nous avons décidé de comparer les résultats de deux algorithmes de détection de lignes que nous avons implémentés et comparés [119] et qui sont présentés dans le Chapitre 4. La première méthode de détection de ligne est basée sur la transformée de Hough comme présentée dans [73], que nous nommons HT dans le Tableau 3.4. La seconde repose sur l'utilisation d'un filtre particulier (noté PF) [118]. Cette comparaison est menée seulement sur des images réelles et utilisent toutes les deux une hypothèse forte sur la couleur. Pour analyser les résultats des algorithmes de détection de lignes suivant l'utilisation ou non de l'Algorithme 2, nous avons décidé d'utiliser trois critères :

- **La portée maximale de détection (en pixels)** : Une des difficultés, avec nos images, est que, pour celles provenant de la caméra placée dans la dérive de l'avion, les marquages importants se trouvent à au moins 50 mètres de la caméra. À cette distance, la couleur des marquages est sensible au bruit et aux saturations additionnelles, ce qui agrandit son écart à la couleur de référence.
- **La moyenne de la distance des pixels sélectionnés à la vérité terrain** : cela nous permet d'exprimer une mesure de la précision de la détection en calculant, pour chaque groupe de pixels associés à un marquage, la distance entre ces pixels et la vérité terrain. Les résultats de chaque groupe sont moyennés pour obtenir un critère de précision global de la détection.
- **Le rappel** : il exprime la proportion des éléments d'une classe correctement détectés parmi l'ensemble des éléments de cette classe.

Le Tableau 3.4 permet de constater que tous les critères sont améliorés, pour les deux méthodes HT ou PF, grâce à l'utilisation de la recherche adaptative d'une couleur de référence spécifique à l'image. En particulier, on note une augmentation significative de la portée maximale de détection, en comparaison avec la vérité terrain (GT), directement liée à une meilleure séparation entre les pixels correspondant aux marquages et ceux correspondant au tarmac, ce que l'on observe aussi dans les résultats de rappel.

L'écart est plus important pour les images provenant de la caméra placée dans la dérive de l'avion : le rappel passe de 34% à 58% pour la méthode basée sur

la transformée de Hough et de 61% à 86% pour la méthode basée sur le filtre particulière. Cela s'explique par le fait qu'à ces distances, les lignes sont plus floues et plus facilement prompt à être confondues avec le tarmac.

	Rappel		Dist. Moy.		Portée maximale		
	HT	PF	HT	PF	HT	PF	GT
cockpit	0.9	0.92	<b>0.14</b>	5.22	394	397	607
cockpit + adapt. ref.	0.94	<b>0.96</b>	3.07	0.71	536	<b>565</b>	607
dérive	0.34	0.61	11.73	59.02	326	357.7	810.56
dérive + adapt. ref.	0.58	<b>0.86</b>	6.95	<b>5.94</b>	<b>705</b>	618.5	810.56

Table 3.4: Résultats de performances d'algorithmes de détection de lignes avec et sans affinage adaptatif de la couleur de référence.

### 3.4.7 Conclusions sur l'étude colorimétrique

Lors de cette étude colorimétrique, nous avons évalué plusieurs fonctions de distance entre une couleur de référence et les pixels de l'image, sur plusieurs espaces de couleur. Nous avons défini des couples (*EspaceCouleur*, *FonctionDistance*) qui puissent permettre une meilleure classification des pixels appartenant aux marquages. Nous proposons un algorithme d'affinage adaptatif de la couleur de référence des marquages, la rendant spécifique à chaque image et amenant une information sur la couleur nécessaire à la détection des marquages au sol dans la zone aéroportuaire.

Nous avons apporté une méthode pour fixer les paramètres de cet algorithme, tout en essayant de conserver une flexibilité requise pour améliorer la robustesse des algorithmes de détection aux variations d'environnement. Notre algorithme nous permet d'obtenir une carte de distance entre une couleur de référence donnée et les couleurs de chaque pixel de l'image, ainsi qu'une carte binaire dont les pixels blancs représentent les pixels définis comme représentatifs d'un marquage au sol. Ces cartes peuvent ensuite être utilisées par des algorithmes de détection de lignes nécessitant une hypothèse sur la couleur. Nous avons observé une amélioration nette des résultats de détection grâce à l'utilisation d'affinage adaptatif de la couleur de référence.

Cette méthode pourrait être utilisée pour détecter d'autres marquages au sol de couleurs différentes ou bien des objets dont les couleurs sont réglementées par les normes aéronautiques comme par exemple les balises de couleur le long des piste ou bien les panneaux d'informations au sol. Pour cela, il suffit de relancer l'étude colorimétrique avec la nouvelle couleur de référence et ainsi obtenir le meilleur couple permettant de segmenter l'objet recherché. En admettant que de prochains travaux soient menés sur des méthodes basées réseau de neurones, il pourrait être intéressant de comparer les résultats de détection en changeant l'espace de couleur de l'image d'entrée du réseau de neurones, par exemple de RGB à L\*C\*h, pour analyser les résultats sur la détection. Pour cela, il faudra ré-entraîner le réseau de neurones sur la partie détection, pour qu'il soit spécifique à la détection dans l'espace de couleur fourni.

Lors de cette étude colorimétrique, nous avons travaillé avec des espaces de couleur connus, qui ne définissent pas de relation d'ordre entre les différents canaux

utilisés pour décrire la couleur. On pourrait imaginer d'adapter notre méthode à l'algorithme proposé par [89], pour redéfinir les bornes de valeur correspondant au jaune tout en ajoutant une relation d'ordre entre les différents canaux des espaces de couleur étudiés. Cette méthode devrait nous permettre d'obtenir un séparateur linéaire plus simple que pour les espaces de couleur actuels. L'utilisation de méthodes basées réseaux de neurones permet d'effectuer des classifications plus complexes sans avoir à définir théoriquement des séparateurs comme des séparateurs non-linéaires par exemple. Il reste que la certification de ces méthodes n'est pas facile.

Comme présenté dans la partie 3.4.3, nous possédons une classe 'Non Défini' qui contient des triplets  $\{R,G,B\}$  ayant été placés dans au moins deux classes différentes par la méthode d'annotation des images. Cela peut venir de changements des conditions météorologiques ou d'éclairage de la scène, qui impactent la couleur des pixels, mais aussi de la variation de la qualité d'une image à l'autre et de la difficulté de labelliser, suivant la couleur, une image pixel par pixel. On note souvent une confusion pour les pixels présents à la frontière entre deux objets distincts (par exemple les marquages et le taxiway) qui crée un biais dans la base de données. Notre méthode pourrait être améliorée en réduisant le biais, par exemple en pré-filtrant les images ou en définissant une manière plus robuste de labelliser.

### 3.4.8 Pour aller plus loin : Application à la labellisation semi-automatique

Dans l'étude colorimétrique que nous avons présentée, nous avons défini un algorithme utilisant une méthode de seuillage pour différencier la couleur des marquages au sol du reste de l'image. Fixer une valeur seuil est une problématique complexe, qui peut être appréhendée par plusieurs méthodes de seuillage, parmi celles-ci on peut citer par exemple la méthode de binarisation Otsu.

Ce faisant, notre approche revient à séparer notre image en deux zones qui sont 'marquages' et 'non marquages'. Intuitivement, on pourrait se dire que choisir une valeur seuil comme effectué précédemment est une manière simpliste de répondre au problème et qu'il existe des méthodes plus complexes de classification qui permettent d'ajouter une dimension non linéaire à cette classification.

D'autre part, nous avons mis en avant le manque d'images labellisées. Dans cette partie, nous avons décidé d'évaluer quelques méthodes de classification afin d'analyser la pertinence d'utiliser un outil un peu plus évolué et de voir si l'on peut utiliser des méthodes de classification pour effectuer des tâches de segmentation sémantique d'image et de labellisation semi-automatique. Nous avons utilisé notre base de données colorimétrique comme base pour cette étude.

#### 3.4.8.a Présentation de méthodes de classification

Les résultats de notre étude colorimétrique tels que présentés peuvent être utilisés par des méthodes de classification à des fins d'aide à la labellisation. Parmi les méthodes de classification les plus connues, on peut citer les méthodes ci-dessous.

- **Les arbres de décision**

Les arbres de décisions peuvent être représentés comme un ensemble de branches issues d'un nœud Racine et dont les terminaisons sont appelées

Feuilles. Les feuilles représentent les classes étudiées et chaque nœud correspond à un critère de séparation de la classification. Plusieurs techniques de classification découlent des arbres de décision comme par exemple l'algorithme Adaboost ou les forêts aléatoires (Random Forests).

Le principe de l'algorithme Adaboost est de combiner plusieurs classifieurs dits faibles, dont les principales caractéristiques sont la simplicité et la rapidité de calcul, tant que le résultat est plus satisfaisant que le hasard. Pour chaque itération, l'algorithme de classification qui minimise l'erreur de classification est sélectionné (s'il est assez rapide) et les poids des algorithmes sont reconfigurés. Le fonctionnement des forêts aléatoires est d'entraîner des arbres de décisions sur chaque échantillon et de sélectionner la classification par rapport au plus grand nombre d'avis donnés par la forêt aléatoire.

- **K-plus proches voisins**

C'est une technique classique de classification. L'algorithme va calculer la distance entre chaque point de la base d'apprentissage et l'élément à classer, garder les k plus proches voisins et attribuer à l'élément à classer la classe majoritaire chez ces k éléments sélectionnés.

- **Les machines à vecteurs de support ou séparateurs à vastes marges (support vector machine ou SVM)**

Le but de ces méthodes est de définir un hyperplan de façon à maximiser la distance entre l'hyperplan et les classes des données d'apprentissage [20]. Les SVMs peuvent être utilisés pour des applications multi-classes [74] en entraînant l'algorithme sur une classe et en la comparant aux autres (stratégie "un contre tous") ou en comparant les classes deux à deux (stratégie "un contre un"). Des améliorations ont été apportées pour permettre d'utiliser les SVMs dans des cas où les classes ne sont pas linéairement séparables.

- **Les réseaux de neurones**

Les réseaux de neurones sont rapidement devenus des éléments privilégiés pour la classification dans de nombreux domaines. Initialement utilisé sous la forme du perceptron, l'augmentation des capacités de calcul a mené à de nombreux travaux sur la complexification de leur architecture. Pour des applications en Vision, les CNN ou Réseaux de Neurones Convolutionnels permettent d'apprendre aussi les meilleures représentations qui permettent de détecter des objets dans des images.

- **La logique floue**

Le principe des classifieurs à base de logique floue est d'affecter à l'entrée un degré d'appartenance à chaque classe du problème. Pour cela, on compare la donnée d'entrée à des valeurs de référence en utilisant des fonctions d'appartenance pour chaque classe. Le degré obtenu est le reflet de la ressemblance de la donnée à la donnée type d'une classe. Comme il s'agit de logique floue, la somme des degrés n'est pas forcément égale à 1. Nous présenterons une application de cette méthode dans la section suivante.

Les classifieurs 'une classe' sont des méthodes consistant à apprendre seulement la classe recherchée lors de la phase d'apprentissage. Dans notre cas, par exemple, on pourrait n'utiliser que des couleurs appartenant à la classe 'Jaune' lors de l'apprentissage puis rechercher les pixels appartenant à cette classe sur une image pour la labellisation. Ce type de classifieur est intéressant lorsque la classe recherchée est spécifique et minoritaire par rapport à l'infinité des autres classes présentes dans

les données, ce qui est semblable à notre cas lorsque l'on regarde la représentativité des marquages dans une image. L'étude [3] compare l'utilisation de plusieurs classifieurs à une classe pour de la segmentation colorimétrique.

Plus spécifiquement, dans le domaine de la classification colorimétrique, on remarque plusieurs approches qui s'appuient ou non sur l'information spatiale. Dans l'article [111], les auteurs présentent plusieurs méthodes de classification spatio-colorimétrique où l'information spatiale est soit prise en compte conjointement ou prise en compte en addition de la classification colorimétrique.

Les auteurs de [129] proposent d'effectuer une classification colorimétrique basée sur l'étude d'histogrammes couleur compacts qui sont une représentation de l'histogramme de couleur d'une image sous forme de deux tableaux où le premier contient les triplets R, G et B unique de l'image et le second contient leur proportion dans l'image. La classification permet alors de faire de la segmentation d'image. Les articles [185], [153] et [53] offrent une bonne visualisation des méthodes de segmentation d'images.

### 3.4.8.b Résultats préalables par l'utilisation de LAMDA

Une des difficultés des approches basées données, outre le nombre de données nécessaire pour obtenir des résultats satisfaisants pour tout type de situation, est la précision de l'annotation. Au cours de cette thèse, nous avons travaillé avec un nombre réduit de données car nous disposons de peu de données labellisées. Comme la détection de marquages se base sur une forte hypothèse de couleur, il peut être intéressant de se pencher sur l'étude colorimétrique pour voir si elle peut être utilisée comme une aide à la labellisation.

Dans les parties précédentes, nous avons présenté les performances de différentes paires (*EspaceCouleur*, *FonctionDistance*) dans le but de trouver une méthode qui discrimine la couleur des marquages en fonction de la couleur de référence fournie par les normes OACI, tout en utilisant un algorithme de faible complexité. Cependant, cet algorithme repose sur une méthode de seuillage classique, qui est une opération binaire. On retrouve aussi cette binarisation lorsque l'on utilise un perceptron classique. Pour obtenir des classifications plus complexes, on peut augmenter le nombre de valeurs seuil utilisées pour définir des bornes d'un espace. C'est ce que font les arbres de décision et les réseaux de neurones plus complexes ou les supports vecteur machine (SVM). Une autre possibilité est d'utiliser des algorithmes de logique floue qui apportent une information plus linéaire.

Nous nous sommes donc demandé si nous pouvions, avec peu de données, utiliser certaines de ces méthodes pour combiner plusieurs paires (*EspaceCouleur*, *FonctionDistance*) et améliorer la précision de la classification colorimétrique des pixels. Pour cela, nous avons retenu les techniques de logique floue et celles d'arbres de décision. Nous avons créé une base de données inspirée de la base de données colorimétrique, qu'on appellera ici base de données de distance. Pour chaque couleur de la première base de données, on crée un vecteur de différence de couleur avec la référence OACI, calculée par les trente sept couples de l'étude colorimétrique auquel on ajoute une variable pour le label de classe. Chaque méthode de classification prend alors en entrée un vecteur de la base de données de distance.

Nous avons sélectionné l'algorithme LAMDA et la sélection d'attributs MEMBAS (Membership-margin based [69]) pour appliquer la logique floue à notre base

de données de couleur. Un vecteur d'entrée est formé en concaténant les résultats de chaque couple (*EspaceCouleur*, *FonctionDistance*), pour une couleur donnée. Ces variables d'entrée sont appelées attributs. L'algorithme calcule un score pour chaque attribut. Plus le score est élevé, plus la proportion de l'attribut à fournir de bons résultats de classification par rapport aux résultats des autres attributs est importante. En effet, la somme des carrés des scores est égale à 1 ce qui implique que le résultat d'un attribut est pondéré par celui de ses voisins. Cela ne veut pas forcément dire que la classification est bonne mais qu'elle est meilleure que les autres.

En utilisant cet algorithme sur une base de données de neuf classes, on obtiendra le score moyen de chaque attribut à classer les éléments des neuf classes. Autrement dit, si l'on veut être certains que les résultats soient centrés uniquement sur la capacité à classer les pixels jaunes, il faut transformer la base de données de distance à neuf classes en une base de données de distances à deux classes. La Figure 3.26 présente les résultats de la sélection d'attributs lors de l'entraînement pour deux classes (Figure 3.26(a)) et pour neuf classes (Figure 3.26(b)).

Les scores, présentés en Figure 3.26, pour l'espace de couleur XYZ corroborent les résultats de l'étude colorimétrique qui montrent que, même si cet espace est utilisé pour définir la couleur des marquages au sol dans les recommandations OACI, il n'est pas pertinent pour la classification. Le couple  $\Delta E^*_{H}$  des espaces de couleur HSV, HSL et L\*C\*h est noté comme un des meilleurs sélecteurs, suivi par les autres couples dépendants des espaces de couleur L\*C\*h et CIELAB (à l'exception de  $\Delta E^*_{A}$  et  $\Delta E^*_{C}$ ). Ces résultats sont cohérents avec nos premières conclusions.

Cependant, ce résultat ne nous donne que la qualité de la classification de chaque méthode prise individuellement par rapport aux autres méthodes. De plus, si nous l'utilisons sur la base de données de distances composée de toutes les classes, certains résultats peuvent être pondérés par une bonne capacité à discriminer entre elles d'autres couleurs. En effet, lorsque la base de donnée est composée de plus de deux classes, le score donné représente seulement la capacité de l'attribut à séparer la classe de l'élément labellisé de la classe la plus 'probable', et n'étant pas la classe labellisée, pour cet attribut.

	Non Jaune	Jaune
Sombre	0.589	0.302
Clair	0.571	0.446
Non Défini	0.605	0.443
Gris	0.633	0.335
Rouge	0.578	<b>0.617</b>
Bleu	0.611	0.312
Vert	0.603	0.506
Herbe	0.587	<b>0.602</b>
Jaune	0.531	0.685

Table 3.5: Scores fournis par l'algorithme de logique floue pour la classe 'Jaune' et la classe 'Non Jaune'.

L'algorithme LAMDA nous renvoie les résultats moyens présentés dans le Tableau 3.5 en termes de degrés d'appartenance généraux aux deux classes 'Jaune'

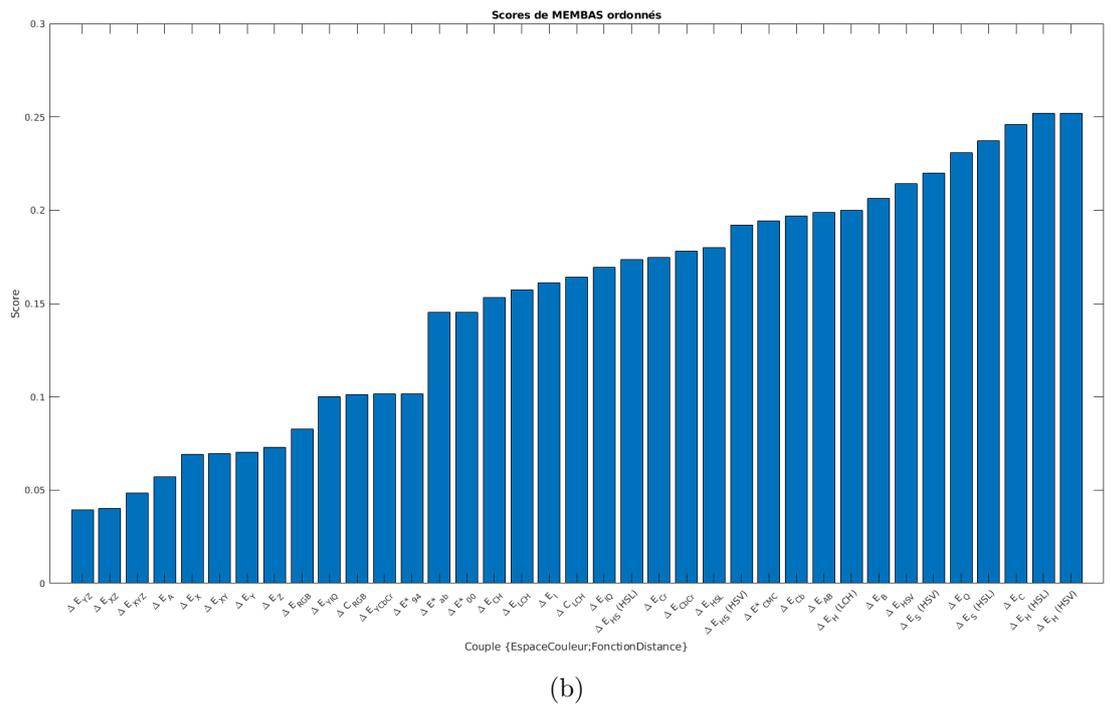
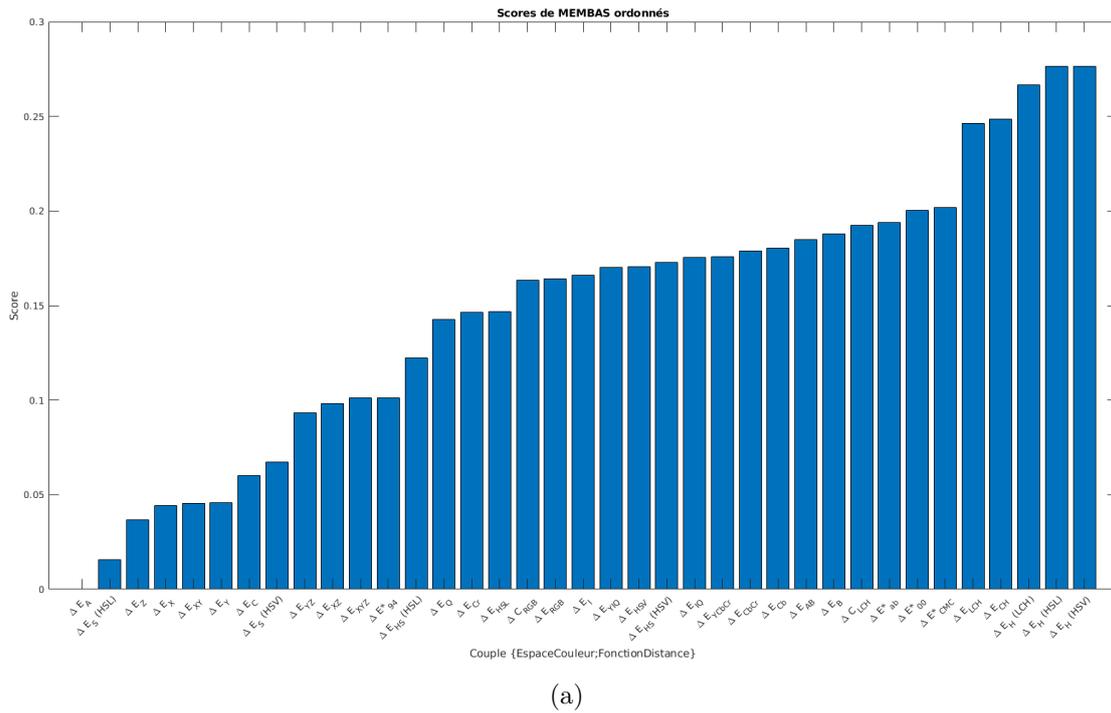


Figure 3.26: Pertinence des attributs fournie par la méthode LAMDA+MEMBAS pour une classification (a) à deux classes et (b) à neuf classes.

et 'Non Jaune' et le Tableau 3.6 propose ces résultats pour une étude sur les neuf classes. On remarque que la sélection des attributs intéressants pour la détection du jaune doit être privilégié sur une classification à deux classes si l'on veut être sûr que l'on sélectionne bien un couple séparateur pour le jaune. Cependant, on voit d'après le Tableau 3.5 que la labellisation à deux classes entraîne en moyenne des erreurs de labellisation sur les classes 'Rouge' et 'Herbe' que l'on ne retrouve pas

	Sombre	Clair	Non Défini	Gris	Rouge	Bleu	Vert	Herbe	Jaune
Sombre	<b>0.677</b>	0.433	0.582	0.587	0.408	0.473	0.471	0.38	0.302
Clair	0.352	<b>0.657</b>	0.444	0.503	0.444	0.384	0.364	0.446	0.446
Non Défini	0.57	0.504	<b>0.652</b>	0.583	0.543	0.398	0.584	0.564	0.443
Gris	0.533	0.517	0.605	<b>0.666</b>	0.436	0.484	0.468	0.418	0.335
Rouge	0.449	0.456	0.542	0.511	<b>0.667</b>	0.333	0.4797	0.604	0.617
Bleu	0.411	0.422	0.434	0.43	0.34	<b>0.682</b>	0.407	0.378	0.312
Vert	0.57	0.48	0.659	0.563	0.582	0.376	<b>0.729</b>	0.62	0.506
Herbe	0.479	0.53	0.642	0.536	0.667	0.328	0.58	<b>0.718</b>	0.602
Jaune	0.378	0.452	0.475	0.448	0.616	0.305	0.445	0.552	<b>0.685</b>

Table 3.6: Scores fournis par l'algorithme de logique floue pour les neuf classes de la base de données de couleurs.

lors d'une labellisation à neuf classes (même si les classes 'Rouge', 'Herbe' et 'Jaune' obtiennent de hauts degrés d'appartenances entre elles). Cela ne veut pas dire que la labellisation est meilleure à neuf classes mais qu'il faut être plus sélectif si l'on utilise la labellisation à deux classes. Ces résultats ne sont d'ailleurs pas étonnant puisque les couleurs des classes 'Rouge' et 'Herbe' sont plus proches de la classe 'Jaune' que d'un mélange des autres classes qui définit la classe 'Non Jaune'.

### 3.4.8.c Résultats préalables par l'utilisation d'un arbre de décision

De son côté, l'utilisation d'un arbre de décision permet de nous renseigner sur la pertinence d'utiliser un seul couple ou une combinaison de différents couples et d'avoir une meilleure information des combinaisons marquantes permettant de maximiser la classification puisque chaque branche de l'arbre est composée d'une combinaison de valeurs seuil sur les différentes variables du vecteur d'entrée. On peut faire varier la taille de l'arbre pour obtenir plus ou moins de précision. Le Tableau 3.7 donne les résultats de précision que l'on a obtenu pour une classification à deux classes et pour une classification à neuf classes, en donnant les attributs en entrée de l'arbre ou en effectuant une analyse en composante principale (PCA), qui nous informe aussi sur l'utilité d'un attribut plutôt qu'un autre. On donne ces résultats pour des arbres profonds (100 nœuds), moyens (20 nœuds) et simples (4 nœuds).

	Deux Classes			Neuf Classes		
	Profond	Moyen	Simple	Profond	Moyen	Simple
Avec PCA	95.5	93.8	92.9	75.9	66.2	40.3
Sans PCA	97.8	97.2	95.2	81.8	59.8	40.5

Table 3.7: Résultats de précision de la labellisation par les arbres de décisions lors de la phase d'entraînement.

On peut noter que les résultats obtenus lors de l'entraînement de l'arbre sont meilleurs lorsque l'on utilise pas de PCA. Les résultats sont aussi meilleurs lorsque l'on ne regarde que deux classes, 'Jaune' et 'Non Jaune'. Cependant, on ajoute en complexité quand on commence à séparer la classe 'Non Jaune' en plusieurs classes et notamment les classes 'Rouge' et 'Herbe' ainsi que les classes 'Gris', 'Non Défini', 'Sombre' et 'Clair' sont proches les unes des autres et expliquent cette baisse de précision.

La Figure 3.27 donne un aperçu des arbres de profondeur moyenne obtenus pour deux classes et pour neuf classes. La lecture de l'arbre se fait de haut en bas. Pour chaque nœud de l'arbre, l'attribut est comparé à une valeur seuil. Le résultat de classification est donné en fin de branche.

On remarque que la classe 'Jaune' est représentée sur une seule partie de l'arbre, à discriminer avec les classes 'Herbe' et 'Rouge', pour la classification à neuf classes. On retrouve les couples  $\Delta E^*_{CMC}$  et  $\Delta E_B$  qui permettent entre autre la spécification de la classe 'Jaune' ce qui rappelle les résultats obtenus dans l'étude colorimétrique. On remarque aussi l'utilisation de couples peu éclairés par l'étude colorimétrique et l'algorithme MEMBAS dans les nœuds du bas de l'arbre, qui permettent une sélection plus fine de la classe.

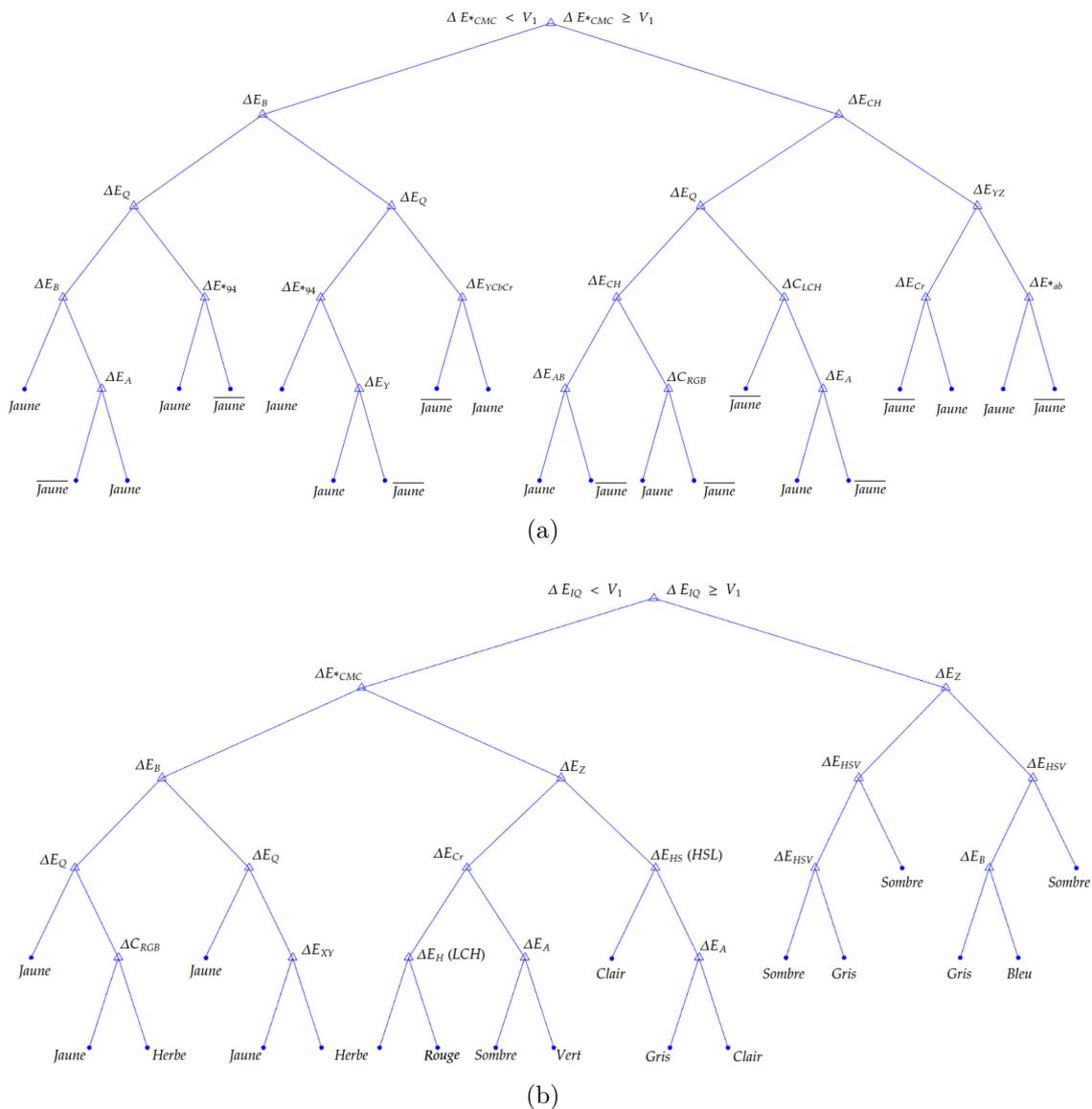


Figure 3.27: Représentation des arbres de profondeur moyenne pour une classification (a) à deux classes et (b) à neuf classes. Le premier nœud explique la façon dont est créé l'arbre.

#### 3.4.8.d Comparaison des deux méthodes

La Figure 3.28 contient les résultats de labellisation de la méthode de logique floue et de l'arbre de décision. On remarque que la labellisation par logique floue est plus complète que la labellisation par arbre de décision. Pour autant, elle ne suffit pas à récupérer les marquages au sol présents au fond de l'image qui, dû à leur distance à la caméra, se confondent plus facilement avec le tarmac. On remarque d'ailleurs que la labellisation par logique floue n'est pas aussi efficace quant au filtrage de faux positifs que la méthode par arbre de décision. On constate plus de fausses labellisations dans la Figure 3.28(d).

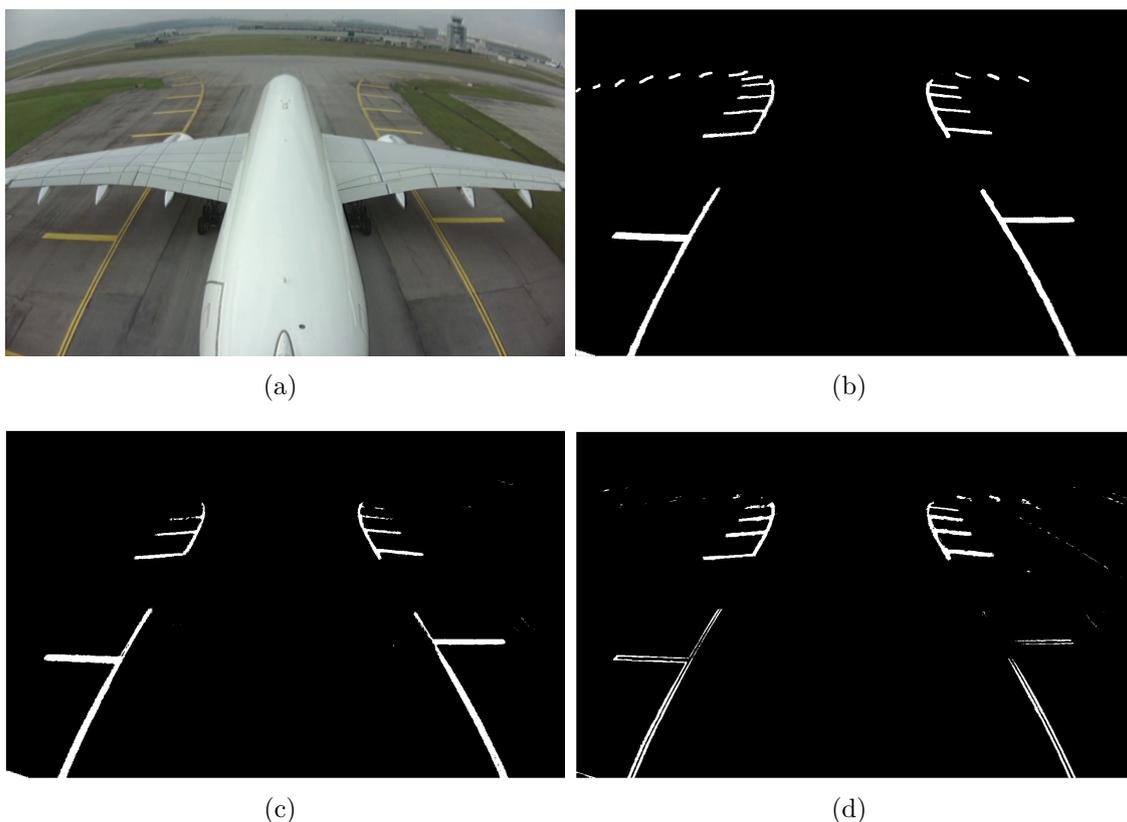


Figure 3.28: (a) Image provenant d'une caméra placée dans la dérive d'un avion et (b) labellisation manuelle. (c) Labellisation par un arbre de décision. (d) Labellisation par logique floue.

#### 3.4.8.e Conclusions et améliorations possibles

Nous avons proposé d'utiliser l'étude colorimétrique pour mettre en place une labellisation semi-automatique des marquages dans une image, en utilisant différentes méthodes de classifications en utilisant les résultats des paires (*EspaceCouleur*, *FonctionDistance*).

Ces résultats sont encourageants mais ne peuvent servir à une classification totalement automatique cependant ils peuvent permettre de réduire le travail de labellisation de l'utilisateur. La méthode de logique floue peut être un peu plus contrainte puisque la valeur seuil permettant la labellisation peut être fixée à un

pourcentage plus ou moins strict de la valeur du degré d'appartenance d'un pixel à une classe.

La classification actuelle se base sur une étude colorimétrique de chaque pixel. Or, on connaît plus d'informations sur la ligne en tant qu'utilisateur, comme par exemple la forme qu'elle possède ou le fait qu'elle se trouve sur le tarmac. De plus, cela rendrait la classification plus correcte puisque la classe ne serait plus 'Jaune' mais 'Marquage' et la base de données ne serait pas pénalisés par des pixels objectivement jaunes mais appartenant à la classe 'Herbe' ou des pixels objectivement gris mais appartenant à la classe 'Jaune'. Il manque donc au moins à cette classification une notion de spatialité. De manière assez rapide, on pourrait envisager de renforcer la labellisation en faisant une cascade de classifieurs avec ceux que l'on possède actuellement.

Il existe de nombreuses méthodes de classifications spatiale basées sur la couleur, notamment au niveau des réseaux de neurones. On peut aussi penser à des algorithmes comme le partitionnement en K-moyennes (ou k-means) où l'on pourrait fixer à minima la classe jaune. On peut aussi penser à l'algorithme de superpixels [1] pour diviser l'image par zone. Dans notre cas, nous avons déjà essayé cet algorithme mais nous avons peu de pixels jaunes et les marquages au sol ont tendance à disparaître si la complexité de l'algorithme n'est pas assez grande.

Une méthode rapide d'ajouter de la spatialité avec nos données actuelles serait de découper les images que nous possédons en petites vignettes labellisées et de définir des motifs représentatifs d'une ligne (probablement des vignettes représentant uniquement des pixels jaunes ou des représentations bicolores de pixels jaunes et de pixels gris). En combinant un des deux classifieurs que nous avons déjà entraînés avec cette méthode de labellisation, on devrait ajouter une notion de spatialité à notre méthode de labellisation. La Figure 3.29 schématise ce processus.

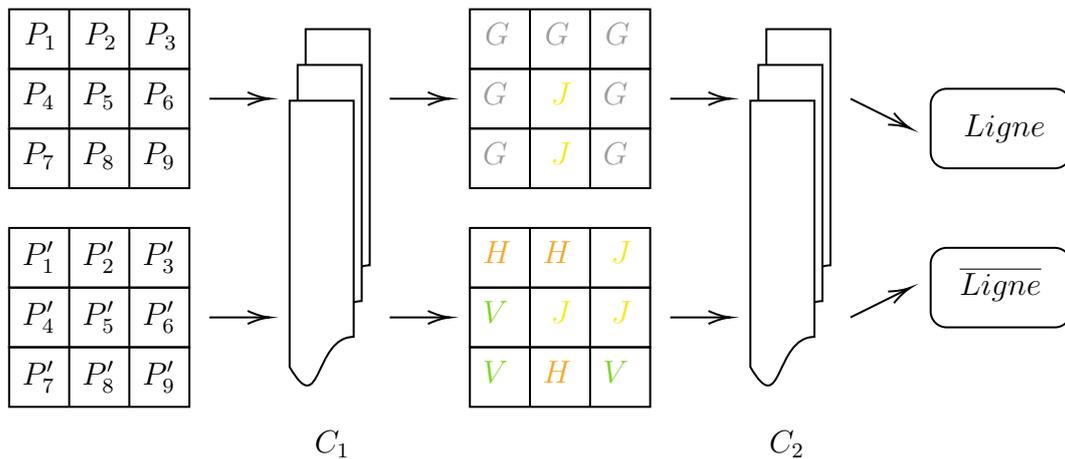


Figure 3.29: Schéma en cascade de classifieurs utilisés pour la labellisation semi-automatique d'images réelles. Le premier classifieur ( $C_1$ ) associe une classe à chaque pixel et le second classifieur ( $C_2$ ) détecte une ligne suivant le motif de couleur fourni en entrée.

## 3.5 Conclusion

Dans ce chapitre, nous avons mis en lumière plusieurs pré-traitements nécessaires à l'exploitation des images reçues par les capteurs intégrés dans les boîtiers d'acquisition : des algorithmes d'adaptation de dynamique et de réduction de bruit, des techniques de fusion d'images provenant de différentes sources ainsi qu'une étude sur l'exploitation d'attributs colorimétriques pour la segmentation de zones d'intérêt (marquages, balises...).

Pour le reste de la thèse, nous avons décidé d'utiliser l'algorithme de CLAHE afin d'adapter la dynamique des images infrarouges. Durant ces travaux, nous n'avons eu accès qu'à des images provenant du simulateur fourni par OKTAL-SE et ces images sont d'assez bonne qualité, ce qui implique que nous n'avons pas forcément besoin d'ajouter des algorithmes de réduction de bruits secondaires. Cependant, il est possible que les images provenant d'un vrai capteur aient besoin de traitements supplémentaires, comme ceux évoqués dans cet état de l'art.

Pour corriger les effets de perspective, ce qui permet de construire une image dans laquelle les marquages au sol restent parallèles, mais aussi d'améliorer la vision à distance lointaine du capteur, nous avons décidé d'utiliser la transformation IPM pour former une carte du sol. Cette transformation nous permet aussi d'aligner les deux capteurs du boîtier d'acquisition, ainsi que de proposer une vue combinée qui fusionne les images acquises par les différentes caméras placées autour de l'avion. C'est une transformation qui est souvent utilisée dans le domaine automobile et qui semble s'adapter facilement au contexte aéronautique vu que l'hypothèse de sol plat est généralement vérifiée dans un aéroport.

Nous avons mené une étude sur la manière d'exploiter des couleurs de référence pour la détection de signalétiques dans le domaine aéronautique (marquage, balise, panneaux, feux...). Notre étude a porté sur les marquages : leur couleur est définie par des recommandations de l'OACI mais la couleur perçue par le capteur et retranscrite dans le domaine RGB diffère de la couleur initiale de référence, dû aux conditions météorologiques, au capteur et aux traitements du signal. Nous avons donc cherché une façon d'exprimer la différence entre cette couleur de référence et la couleur de chaque pixel, robuste aux différentes variations de la couleur des marquages. En parallèle, nous avons étudié la possibilité de se servir de ces résultats pour développer un algorithme de labellisation d'image. Nous avons conclu que le calcul de la différence de couleur au moyen de  $\Delta E^*_{CMC}$  était le plus pertinent et qu'une labellisation, même seulement semi-automatique, pouvait en résulter.

Dans le prochain chapitre, nous allons présenter la brique fonctionnelle "Analyse de scène" dans laquelle nous définirons l'algorithme que nous avons choisi pour effectuer la détection de motifs (marquages au sol, balises, délimitations tarmac-herbe...). Nous proposerons aussi une comparaison avec d'autres méthodes de détection de marquages au sol.



# Chapitre 4

## Détection de motifs dans les zones aéroportuaires

### Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>96</b>
<b>4.2</b>	<b>État de l'art sur la détection de marquages</b>	<b>98</b>
4.2.1	Présentation des algorithmes à base de méthodes géométriques	98
4.2.2	Présentation des algorithmes à base de réseaux de neurones	101
4.2.3	Algorithme sélectionné pour la détection de marquages	102
<b>4.3</b>	<b>Détection par filtrage particulaire</b>	<b>103</b>
4.3.1	Principe du Filtre Particulaire	104
4.3.2	Détail des différentes phases du filtre particulaire	106
4.3.3	Cas des marquages apparaissant comme "horizontaux" dans le plan du sol	109
4.3.4	Post-traitement des clusters de particules	110
4.3.5	Résultat du filtre particulaire	111
<b>4.4</b>	<b>Amélioration du filtre particulaire</b>	<b>114</b>
4.4.1	Redéfinition du modèle dynamique $h$ de propagation des particules	114
4.4.2	Participation à l'élaboration d'un filtre particulaire fenêtré	116
<b>4.5</b>	<b>Comparaison des performances</b>	<b>118</b>
4.5.1	Présentation de la comparaison	119
4.5.2	Résultats de la comparaison	120
4.5.3	Conclusion de la comparaison des trois méthodes	129
<b>4.6</b>	<b>Conclusion</b>	<b>132</b>

---

## 4.1 Introduction

Dans le cadre de cette thèse, nous nous intéressons à la navigation d'un aéronef sur le taxiway et autour des portes d'embarquement. Avant d'étudier la détection de possibles obstacles présents autour de l'avion, il est nécessaire de récupérer des informations en rapport avec l'environnement afin de le situer au sein de celui-ci. C'est pourquoi l'analyse de scène est une brique importante pour l'assistance à navigation ou la conduite autonome. Elle nous permet de comprendre où se trouve l'aéronef et les obstacles potentiels vis-à-vis de l'environnement (typiquement, sur le taxiway ou non).

Dans ce chapitre, nous nous consacrons à cette première brique de l'aide à la navigation. Dans notre contexte, les motifs de la scène susceptibles de nous intéresser sont multiples, la Figure 4.1 en présente quelques uns. On peut noter par exemple les différentes signalisations autour et sur les voies de navigation : horizontales (marquages, balises, nom des taxiways peints au sol...) ou verticales (panneaux...). La séparation entre les voies de navigation et le reste de l'environnement (que l'on appellera par la suite séparation tarmac/herbe) est aussi un élément que nous pourrions souhaiter détecter.

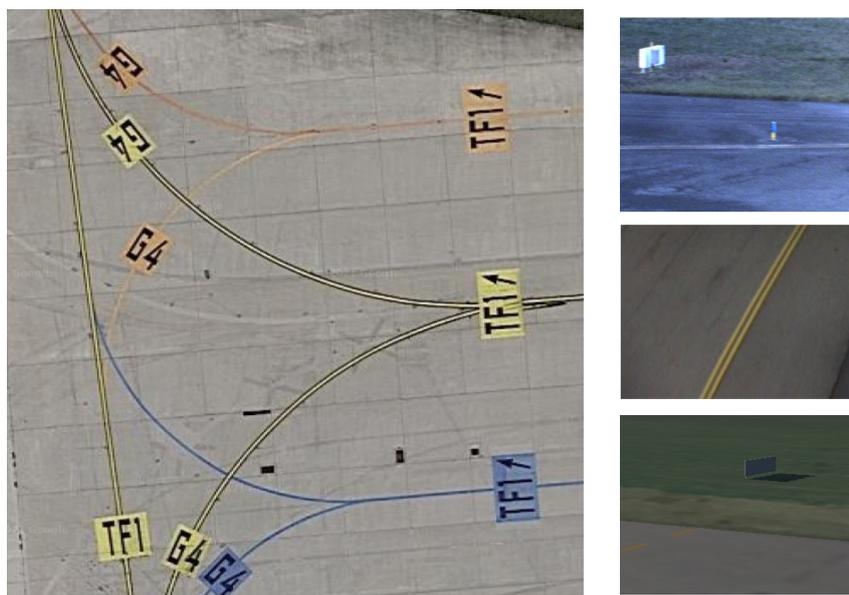


Figure 4.1: Exemple de motifs d'intérêt des zones aéroportuaires : motifs de marquages au sol complexes, balises de bord de pistes, lignes doubles, panneaux...

Dans un premier temps, nous présenterons différentes méthodes d'analyse de scène proposées dans la littérature et particulièrement les méthodes géométriques ainsi que les méthodes basées réseaux de neurones de détection de lignes. En effet, les motifs que nous souhaitons détecter en priorité sont les marquages au sol dont la détection ainsi que celle de la frontière tarmac/herbe peuvent s'interpréter comme une détection de lignes dans l'image. On montrera dans la Section 4.3.4 que l'utilisation de la transformation IPM nous permet d'exprimer la détection des balises comme étant aussi une détection de ligne dans l'image.

Après quoi, nous présenterons, dans ce chapitre, une implémentation de l'algorithme de filtre particulaire que nous avons choisi pour détecter des marquages au sol ainsi que des balises et montrerons les résultats sur les motifs étudiés. Cette partie a fait l'objet d'une publication à VISAPP en 2019 [118]. Nous avons apporté de nouvelles améliorations depuis cet article, que nous détaillerons en suivant.

Bien que l'approche par filtre particulaire soit aussi utilisé dans le cadre d'un suivi temporel, nous ne l'utilisons que dans le but d'une détection au sein de l'image à un instant  $t$ . De même, on retrouve souvent un lien très fort entre les problématiques de détection et de suivi dans les méthodes proposées dans la littérature. Dans ce chapitre, nous ne nous intéressons qu'à la détection des motifs dans une image et non à son suivi dans une suite d'images.

Pour finir, ce chapitre présentera une comparaison entre plusieurs méthodes dédiées à la détection de marquages au sol. L'algorithme de filtre particulaire sera comparé à une autre méthode géométrique ainsi qu'à une méthode basée réseaux de neurones. Cette comparaison permettra aussi de s'interroger sur la pertinence de la transformation IPM dans la détection de marquages ainsi que sur la robustesse des différentes méthodes aux changements de conditions climatiques. Cette partie a fait l'objet d'une publication à ISCSIC en 2020 [119].

## 4.2 État de l'art sur la détection de marquages

Bien que le sujet ait été profondément discuté dans le domaine automobile et que des solutions matures sont aujourd'hui embarquées sur des véhicules dits intelligents pour alimenter différentes fonctions (ex: Lane Keeping Assist qui prévient le conducteur si son véhicule sort de sa voie), peu de travaux ont été dédiés à l'analyse de scène dans le domaine aéronautique. On peut néanmoins citer [160] qui se focalise sur l'extraction de la ligne de marquage centrale du taxiway et l'estimation du positionnement de l'aéronef par rapport à celle-ci ou [164], uniquement axé sur la détection de la "stop bar" qui est un élément parmi d'autres de signalétique dans les zones aéroportuaires. Le modèle de route et les connaissances a priori (présence de plusieurs voies, largeur de celles-ci, motifs des marquages...) diffèrent entre les deux domaines. Néanmoins, nous pensons que les méthodes utilisées dans le domaine automobile peuvent être adaptées pour le domaine aéroportuaire puisque la problématique reste commune : détection des "lignes", au sens d'un marquage continu, plein ou en pointillé, constitué de une ou deux bandes, et d'une couleur déterminée selon le contexte.

De nombreuses méthodes ont été proposées dans la littérature scientifique pour la détection de lignes. On peut séparer ces travaux en deux grandes familles : les algorithmes à base de méthodes géométriques et les algorithmes à base de réseaux de neurones.

### 4.2.1 Présentation des algorithmes à base de méthodes géométriques

On peut trouver deux familles d'approches principales dans les méthodes géométriques : les approches basées sur les modèles géométriques de la scène et les approches basées sur les caractéristiques et l'apparence de la scène (couleurs, textures).

Les approches 2D et 3D basées modèles géométriques se fondent sur des définitions fortement contraintes de la scène pour exécuter la détection. Pour ces approches, dans le cadre de la détection de voies de navigations par exemple, les paramètres de la caméra, la largeur de la route et sa courbure, le nombre de lignes présentes et la position du véhicule par rapport aux éléments observés sont des éléments nécessaires pour les recherches présentées dans [9], [35], [105] et [25]. Ces méthodes présentent une bonne robustesse mais demandent beaucoup de ressources de calcul et de fortes hypothèses sur la géométrie de la scène.

Les approches basées sur l'apparence et les caractéristiques de la scène utilisent une combinaison de caractéristiques de bas niveau telles que les modèles de couleur de la route, les contrastes entre les différents objets, leurs formes ou les orientations de groupes de pixels par exemple. Elles sont moins coûteuses en termes de complexité de calculs ce qui les rend plus attractives pour des applications temps réel, comme pour les recherches de [154], [100] et [73]. Cependant ces techniques sont moins robustes aux problèmes d'ombrage ou d'occultation. Ces approches demandent souvent des pré-traitements comme par exemple :

- **la définition d'une zone de recherche privilégiée (ou région d'intérêt, ROI)**  
Elle permet de réduire la zone de recherche de l'image et donc la complexité et le temps de calcul.
- **le calcul du point de fuite**  
Il permet de définir la ligne d'horizon. En effet, sur des portions droites et plates de la route, cela permet de discriminer les contours de l'image qui appartiennent aux différents marquages et, éventuellement, au bord de la route, puisque toutes ces lignes sont parallèles dans la scène (donc convergentes vers le point de fuite dans l'image). La connaissance de la ligne d'horizon permet de réduire les zones de l'image sur lesquelles on applique les algorithmes de détection de lignes.
- **l'application de la transformée IPM** que nous avons présentée dans le chapitre précédent, qui se substitue au calcul du point de fuite puisque, dans la carte du sol, les lignes des marquages et des bords de route sont parallèles.
- **le débruitage et l'amélioration des contrastes**  
Les motifs, et particulièrement les marquages au sol, présentent des différences de contrastes et de texture. Il peut être intéressant d'effectuer des pré-traitements de lissage de l'image pour faire ressortir les discontinuités dans une image et améliorer la détection de motifs.
- **l'extraction des contours**  
Elle se fait principalement en calculant des gradients dans l'image comme avec les méthodes de Sobel, Canny ou la détection de contours localement rectilignes [168] (LSD) par exemple.
- **la conversion dans un autre espace de couleur pour l'extraction de la couleur ou la transformation de l'image visible couleur en niveau de gris**  
Comme nous l'avons présenté dans le Chapitre 3, l'espace RGB n'est pas forcément l'espace privilégié pour la détection de lignes ou de motifs. Comme les composantes de la scène peuvent être soumises à des variations d'illumination, de nombreux travaux préfèrent étudier la scène dans des espaces définis spécifiquement pour prendre en compte la luminosité de la scène. Dans le domaine automobile, les marquages à détecter sont principalement blancs, aussi l'information de couleur est moins utile pour leur détection et le passage en niveaux de gris simplifie le coût de détection.

Ci-dessous, nous revenons sur les approches géométriques les plus utilisées dans la détection de lignes.

#### 4.2.1.a Transformée de Hough

La transformée de Hough est une approche générale permettant de regrouper des pixels qui appartiennent à des courbes de forme donnée (dans la version initiale, des droites). Dans une première étape, on doit extraire les pixels pouvant appartenir à ces courbes; ce sont généralement les points de fort gradient. Puis une méthode de

vote dans un espace paramétrique adapté, permet de trouver les courbes recherchées. Concernant la première étape, pour la détection des marquages au sol, on peut exploiter les informations sur la couleur et les contours d'une image.

On retrouve dans la littérature scientifique plusieurs définitions de la transformée de Hough : l'algorithme classique, la transformée généralisée de Hough (qui permet de détecter des objets dont la forme géométrique ne possède pas de représentation analytique simple) et la transformée de Hough probabiliste (qui permet de diminuer le temps de calcul en appliquant la transformée de Hough à un sous-ensemble sélectionné aléatoirement).

L'extraction de l'information de couleur dépend de la définition de la couleur de référence, qui est plus difficile à exprimer que la définition des contours. Quelques auteurs, comme [113] et [84], utilisent la transformée de Hough sur des images converties en HSV, Lab ou YCbCr. D'autres auteurs (comme [142] et [86]) utilisent des seuils locaux adaptatifs dans le but d'être robuste à la variation de luminosité.

Ce seuillage produit une image binaire sur laquelle on peut par exemple appliquer un gradient morphologique (qui peut aussi se faire sur des images en niveaux de gris) ou bien des méthodes communes dans la littérature comme Sobel [181] ou Canny [152]. Il existe cependant des applications de la transformée de Hough pour des images en nuances de gris comme proposé par [85].

Le regroupement en clusters du résultat peut être amélioré par l'utilisation d'un algorithme de regroupement hiérarchique sur les clusters déjà fournis par la transformée de Hough, comme présenté dans [73]. Cependant, la transformation de Hough reste un algorithme à force brute qui peut, dans certains cas, s'avérer coûteux.

Les deux méthodes présentées par la suite sont couramment utilisées pour la détection mais aussi le suivi de motifs. On peut aussi différencier la transformée de Hough des ces deux autres méthodes par leur aspect global/local. La transformée de Hough passe par une analyse globale de l'image pour trouver des clusters de pixels appartenant à des courbes (de manière générale dans la détection de marquages au sol). Les méthodes de filtrage par filtre particulière ou par Kalman sont locales : les clusters sont formés progressivement en "suivant" les courbes à partir d'une initialisation (des "line seeds").

Dans ce chapitre, on ne s'intéresse qu'à la détection des motifs au sein d'une image. Certains auteurs parlent de "line tracking" (suivi des pixels appartenant à une ligne dans une image), à ne pas confondre avec le suivi de lignes dans une séquence d'images que nous n'aborderons pas dans cette thèse.

#### 4.2.1.b Filtre particulière

On peut noter l'utilisation du filtre particulière dans les travaux de [148], [34] ou [92]. Le filtre particulière est un filtre bayésien récursif qui estime l'état d'un système par un ensemble de particules. Ces particules sont mises à jour selon un cycle de prédiction et de correction. Dans le cas de la détection de motifs, les particules vont estimer la position du motif recherché dans l'image et une fonction d'évaluation va estimer la probabilité d'une particule de représenter le motif. Dans notre cas spécifique de la détection de ligne au sein de l'image (ou de la carte du sol si l'on a appliqué la transformation IPM), chaque pixel de l'image représente une observation qui permet de mettre à jour les différentes particules.

Cette méthode permet de faire une approximation des fonctions complexes définissant les motifs recherchés, par exemple les équations représentant les marquages au sol. Cela permet d'extraire des motifs dont la définition géométrique est difficile à modéliser. Cependant, l'algorithme de filtre particulaire est une méthode qui est très consommatrice en terme de temps de calcul suivant le nombre de particules utilisées par le filtre ainsi que le nombre d'itérations du cycle prédiction-correction. Nous avons cependant choisi d'utiliser cette méthode pour effectuer la détection de marquages dans nos images. Cette approche sera décrite en détail dans la Section 4.3.

#### 4.2.1.c Filtre de Kalman

L'utilisation du filtre de Kalman a été proposée par de nombreux travaux ([9], [24]), pour la détection de marquages au sol. La méthode proposée dans [24] exploite une technique d'estimation par filtrage de Kalman étendu (EKF), afin de mettre à jour une représentation de la route, sous la forme de deux courbes quadriques définies dans le plan du sol, correspondant soit aux marquages limitant la voie dans laquelle circule le véhicule, soit aux bords de la route s'il n'y a pas de marquage. À l'état initial, ces courbes sont connues avec une grande incertitude : le modèle est ensuite mis à jour itérativement par EKF à partir des observations successives des marquages ou des bords de route, représentées par des courbes quadriques dans le plan de l'image. Notons que, dans le cas du suivi de ligne, le déplacement de la caméra doit être estimé afin de prédire la position des marquages dans l'image à l'instant  $t+1$ , à partir de l'état de la route estimé à l'instant  $t$ .

Cette méthode est plus rapide que le filtre particulaire cependant elle demande d'avoir une meilleure connaissance du modèle de la route. En effet, même s'il est incertain, le modèle doit être choisi. Par exemple dans [24], le modèle choisi pour représenter les marquages au sol sont des quadriques. Cela rend plus complexe, par rapport au filtre particulaire, la détection de motifs dont la modélisation mathématique n'est pas chose aisée.

#### 4.2.2 Présentation des algorithmes à base de réseaux de neurones

Les réseaux de neurones convolutifs (CNN) fournissent actuellement les meilleures performances pour des applications de traitement d'image et de vision par ordinateur. Dans le domaine automobile, ils obtiennent de très bons résultats. Cependant, la nature des zones aéroportuaires est un peu différente du domaine automobile et il n'existe pas encore de réseaux de neurones entraînés pour ce cas d'étude. Mais cela pourrait évoluer très vite, car AIRBUS s'appuie sur sa filiale ACubed localisée dans la Silicon Valley, pour mettre en œuvre des méthodes de Machine Learning adaptées aux zones aéroportuaires. De premiers algorithmes sont à l'étude, en particulier pour la détection de pistes dans la phase d'atterrissage.

Plusieurs structures de réseaux de neurones ont été proposées pour la détection de marquages au sol dans le domaine automobile.

Une des structures classiques est le réseau de neurones sous forme d'encodeur-décodeur. C'est une structure composée de deux réseaux de neurones. Le réseau encodeur est utilisé pour la sélection de points d'intérêt et le réseau décodeur utilise

ses informations pour faire ressortir le motif à détecter. Parmi les travaux proposés utilisant cette méthode, on peut citer [126]. C'est un CNN dont l'architecture est composée de trois réseaux, un encodeur et deux décodeurs. Le premier décodeur est utilisé pour fournir une image binaire des pixels contenant une information probable sur les marquages. Le second décodeur crée une carte de pixels qui va être utilisée pour regrouper les pixels de l'image binaire en plusieurs voies de navigation. Comme notre extraction de motifs est basée sur les pixels contenant de l'information sur les marquages au sol, nous pourrions nous contenter du résultat fourni par le premier décodeur. Dans [93], les auteurs proposent d'ajouter la détection du point de fuite à celle des voies de navigations et des marquages au sol par le biais d'un réseau de neurones.

Une autre structure couramment utilisée est la structure GAN (pour Generative Adversarial Network). C'est aussi une structure composée de deux réseaux de neurones. Le premier réseau de neurones va créer une image en sortie en se basant sur l'image d'entrée. Le second réseau compare la prédiction du premier réseau avec une vérité terrain fournie ou apprise par ailleurs et associe un score qui permet d'affiner la prédiction du premier réseau. Dans [54], les auteurs proposent l'utilisation d'un GAN pour détecter des marquages. Cependant, ces réseaux sont plus complexes à entraîner que les réseaux de type encodeur-décodeur.

Comme pour les méthodes géométriques, d'autres structures ont été proposées pour associer détection et suivi de lignes. On peut citer par exemple les réseaux de neurones récurrents (RNN) ou encore leur évolution, les LSTM (long-short time memory).

### 4.2.3 Algorithme sélectionné pour la détection de marquages

Parmi les différentes méthodes de détection de lignes, nous avons choisi d'utiliser le filtre particulière pour notre cas d'application. Son utilisation dans le cadre de "line tracking" semble appropriée et ne ferme pas la porte à une amélioration future d'association entre détection au sein d'une image et suivi d'une image à l'autre. De plus, la configuration des marquages aéroportuaires (successions de segments de droites et d'arcs de cercle, convergence et divergence des marquages lors des croisements de taxiways) est difficile à modéliser et se prête bien à l'utilisation du filtre particulière.

Nous proposons ainsi par la suite d'utiliser une méthode d'interprétation de scène basée sur des modèles faibles (qui permettent d'évaluer la similarité d'une particule par rapport au motif recherché) dans le but de détecter des motifs aéroportuaires (marquages, balises, délimitations tarmac/herbe). Nous utilisons une combinaison d'observations et de connaissances a priori de la scène dans le but d'estimer des fonctions complexes décrivant par exemple la forme des marquages aux sols. Les travaux de [147] et [81] s'inscrivent dans le même procédé.

Cette méthode peut être coûteuse en temps de calcul mais peut être parallélisée. Le but à terme étant d'implémenter les algorithmes retenus sur des architectures parallèles (à base de FPGA ou GPU), nous avons décidé de le sélectionner malgré son coût.

## 4.3 Détection des marquages et balises par filtrage particulaire

Nous avons abordé la transformation IPM comme pré-traitement, ce qui permet de supprimer les effets perspective (les droites parallèles dans la scène le sont aussi dans la carte du sol). Un exemple est donné en Figure 4.2. L'algorithme de filtre particulaire peut être effectué sur l'image visible d'origine ou bien sur la carte du sol. Dans cette partie, nous présentons les résultats du filtre particulaire sur la carte du sol. Dans la Section 4.5, nous proposerons d'étudier l'impact de cette transformation sur les performances de l'algorithme au niveau de la détection de marquages au sol.

Nous nous limitons à la segmentation des pixels appartenant aux marquages dans l'image, en exploitant les observations fusionnées dans la carte de poids dont la génération est décrite dans le Chapitre 3. Plusieurs traitements complémentaires classiques permettront de fournir la position des lignes sur le plan du sol, cela pour alimenter des besoins opérationnels pour la navigation (par exemple le calcul de la position de l'aéronef par rapport à la ligne centrale du taxiway, notamment dans les virages pour contrôler le nécessaire overshoot de l'avion et éviter de franchir la limite tarmac/herbe).

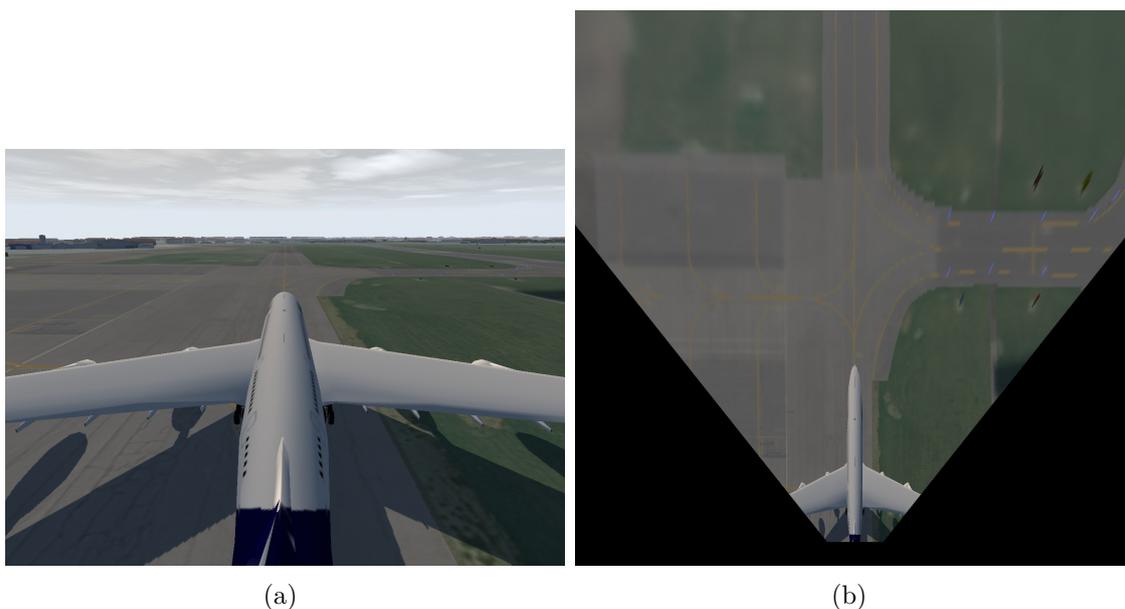


Figure 4.2: Présentation d'images d'entrée du filtre particulaire : (a) image  $IM$  originale et (b) carte du sol, résultat de la transformation IPM.

Dans cette section, nous abordons le principe de l'algorithme de filtre particulaire que nous avons mis en place. Nous détaillons ensuite les phases d'initialisation, de prédiction et de correction du filtre. Enfin, nous présentons les motifs pouvant être extraits par cet algorithme de filtre particulaire et les résultats présentés à VISAPP en 2019 [118].

### 4.3.1 Principe du Filtre Particulaire

Le filtre particulaire est un filtre bayésien récursif qui estime l'état d'un système par un ensemble de particules. Ces particules sont mises à jour selon un cycle de prédiction et de correction. On peut généraliser cette méthode par le formalisme suivant.

On note  $X_t$  un vecteur d'état à l'instant  $t$ . La fonction  $f$  relie  $X_t$  au vecteur d'état  $X_{t-1}$  à l'instant  $t-1$  et représente le modèle dynamique du système (Équation 4.1) qui est utilisé pour le cycle de prédiction. Comme nous ne connaissons pas parfaitement  $f$ , même si on l'estime, il existe un bruit  $U_t$  qui pondère ce lien. Le choix du modèle dynamique pour notre application est présenté plus tard en Section 4.3.2.b.

$$X_t = f(X_{t-1}) + U_t \quad (4.1)$$

On note  $Y_t$  le résultat de l'observation. La fonction  $g$ , appelée fonction d'évaluation, permet alors de lier le motif théorique au résultat de l'observation, tout en admettant une perturbation  $V_t$  (Équation 4.2).

Pour la fonction d'évaluation, nous avons fait le choix de nous baser sur une carte de distance dont la génération a été expliquée dans le Chapitre 3. Dans notre cas, l'observation est l'intensité du pixel décrit par la particule. Le résultat de la fonction d'évaluation est le poids associé à cette particule par la carte de poids. Comme présenté plus tôt, nous avons fait le choix de nous baser, pour la fonction d'évaluation, sur une étude colorimétrique et d'utiliser le calcul de distance entre la couleur d'un pixel et une couleur de référence. Dans la Section 4.4.2, on aborde la possibilité de complexifier cette fonction d'évaluation en la remplaçant par exemple par un classifieur.

$$Y_t = g(X_t) + V_t \quad (4.2)$$

On peut définir plusieurs phases distinctes du filtre particulaire, qui seront explicitées, pour notre cas, en Section 4.3.2 :

- une phase d'initialisation du filtre;
- une phase de prédiction dans laquelle on estime la position du motif recherché;
- une phase de mise à jour des particules dans laquelle on pondère cette estimation par l'observation de celle-ci.

Nous utilisons un filtre particulaire pour la détection d'objets particuliers tels que les marquages au sol, la délimitation du tarmac ou encore les balises dans les zones aéroportuaires. Pour cela, nous implémentons un échantillonnage avec ré-échantillonnage par importance (SIR) dont la Figure 4.3 présente le principe.

Le filtre particulaire approxime la distribution de filtrage  $p(x_t | y_0, \dots, y_t)$  de l'état  $x_t$  via un ensemble d'échantillons, appelées particules et notées  $x_t^{(N)}$ . On note  $y_t$  l'observation à l'instant  $t$  et  $N$  le nombre de particules utilisées. On appelle  $q(x_t | x_{0:t-1}, y_{0:t})$  la distribution proposée pour la prédiction, qui correspond à un bruit près à la fonction  $f$ .

Les particules sont pondérées par un poids de correction de la prédiction noté  $w_t^{(N)}$  qui est une valeur pondérée de l'approximation de la probabilité a posteriori

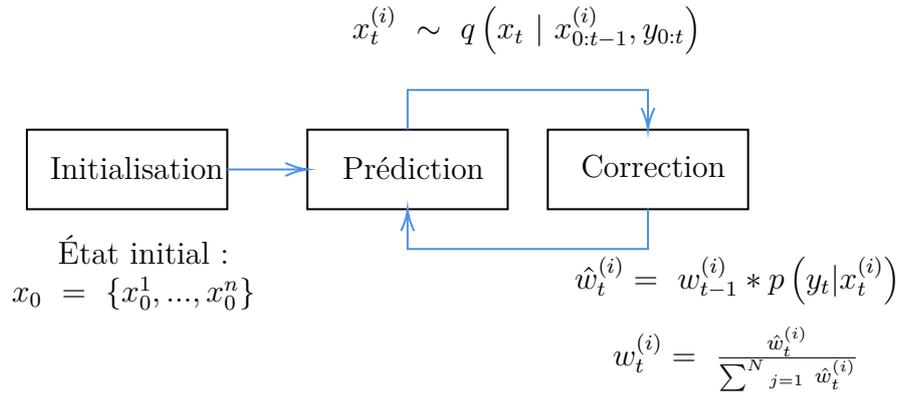


Figure 4.3: Principe du filtre particulaire.

de la particule  $\hat{w}_t^{(N)}$ . La somme des poids associés aux particules d'un cluster est pondérée et égale à 1. Les particules sont mises à jour par une série de prédictions basées sur la connaissance a priori du système et la correction des prédictions au regard de l'observation des particules. Le paramètre  $p(y_t | x_t^{(i)})$  correspond à un bruit près à la fonction  $g$  que l'on multiplie par le poids associé à la particule à l'instant  $t - 1$ .

On peut définir une estimation de l'état courant en fonction des particules qui le décrivent, qu'on note  $\hat{x}_t$  est donnée par l'application de la fonction argmax aux particules.

Le filtre particulaire, comme le filtre de Kalman, est principalement utilisé dans le cadre de la détection et le suivi d'un motif de manière spatio-temporelle. Cependant, dans notre application, nous nous concentrons sur la détection d'un motif au sein de l'image (détection spatiale) plutôt qu'au cours d'une suite d'images (détection temporelle).

En pratique, le filtre suit les mêmes équations, mais la variable  $t$  est redéfinie. Elle ne représente plus une notion de progression temporelle entre deux images mais une notion de progression au sein de l'image. Dans notre utilisation du filtre particulaire, elle représente alors un pas d'incrément du filtre dans la hauteur de l'image.

Les marquages et autres motifs d'intérêt sont plus définis dans le bas de l'image, puisque ces pixels représentent des zones plus proches de la caméra. Aussi, nous initialisons les clusters de particule dans le bas de l'image et la progression s'effectue de bas en haut.

Comme nous effectuons une détection spatiale, par le filtre particulaire, des motifs au sein de l'image, l'information qui va être observée par le filtre est déjà connue avant l'initialisation. On peut donc en amont générer une carte d'observation, comme présenté dans le Chapitre 3. Binarisée, elle est aussi utilisée pour la phase d'initialisation du filtre particulaire. Dans la section suivante, on décrit plus précisément les étapes du filtre particulaire mis en place pour la détection de motifs.

### 4.3.2 Détail des différentes phases du filtre particulaire

#### 4.3.2.a Initialisation du filtre particulaire

L'initialisation du filtre particulaire se fait grâce à la carte de poids binarisée dont la création est détaillée dans le Chapitre 3. Pour l'initialiser le filtre, cette carte est parcourue ligne par ligne, en partant du bas de l'image. Chaque ligne est parcourue colonne par colonne. Lorsque le premier pixel blanc est détecté dans la carte binarisée, l'indice de sa position dans la hauteur de la carte est choisi comme indice d'initialisation  $t_0$ . Les itérations du filtre se font à partir de cette hauteur de l'image (ou de la carte du sol).

Cette ligne est parcourue colonne par colonne et des segments sont formés par regroupement de pixels blancs connexes. Chaque segment est relié à un cluster de particules dont les positions initiales sur le segment sont tirées aléatoirement de manière uniforme. Le nombre de particules  $n$  par cluster est un paramètre essentiel de l'algorithme. Il doit être fixé en prenant en compte la taille du motif que l'on souhaite détecter ainsi que la puissance de calcul dont dispose l'algorithme. Dans notre cas, nous avons fixé  $n$  à 10.

À chaque particule est associé un poids donné par la carte globale de poids. Les poids des différentes particules sont normalisés pour que le poids global du groupement de particules soit égal à 1.

La méthode d'initialisation est aussi effectuée durant l'itération du filtre particulaire, comme expliqué en Section 4.3.2.c. Cela permet, entre autre, de pouvoir suivre des marquages dont le tracé est en pointillés. La Figure 4.4 illustre le processus d'initialisation par le biais d'une carte binarisée sur laquelle les lignes rouges représentent les hauteurs de l'image dans lesquelles sont détectées des nouveaux départs de filtre et les flèches bleues indiquent les segments initiaux.

#### 4.3.2.b Prédiction et correction

Le filtre particulaire implémenté dans notre algorithme scanne l'image d'entrée toutes les  $l$  lignes, en partant du bas de l'image, où  $l$  est défini de manière à être assez petit pour ne pas manquer de départs de détection potentiels tout au long de l'examen de l'image d'entrée, mais assez grand pour réduire le temps de calcul.

Les poids, qui représentent le résultat de l'observation, sont utilisés pour tout  $t > t_0$  pour donner une plus grande importance aux particules les plus probables de représenter l'objet recherché, ce qui améliore la prédiction des particules futures lors du ré-échantillonnage multinomial.

Lors de l'étape de prédiction, les poids des particules d'un groupement sont cumulés pour donner une échelle de poids, visible en Figure 4.5 sur un exemple de 5 particules. Chaque intervalle de cette échelle est associé à une particule (partie "Échelle des poids cumulative" de la Figure 4.5). On tire aléatoirement un nombre défini (égal au nombre de particules du groupement) de valeurs entre 0 et 1, que l'on représente par des croix rouges sur l'échelle de poids.

Ces associations vont faire ressortir les particules au temps  $t$  dont les positions serviront de départ pour la prédiction des particules au temps  $t + 1$ .

La position des particules  $P_{(t+1)}^{0:k}$  est donnée par le modèle dynamique. Dans un premier temps, nous avons défini l'évolution des motifs recherchés comme une propagation rectiligne du bas vers le haut de l'image. On décrit l'état  $X_t$  par une

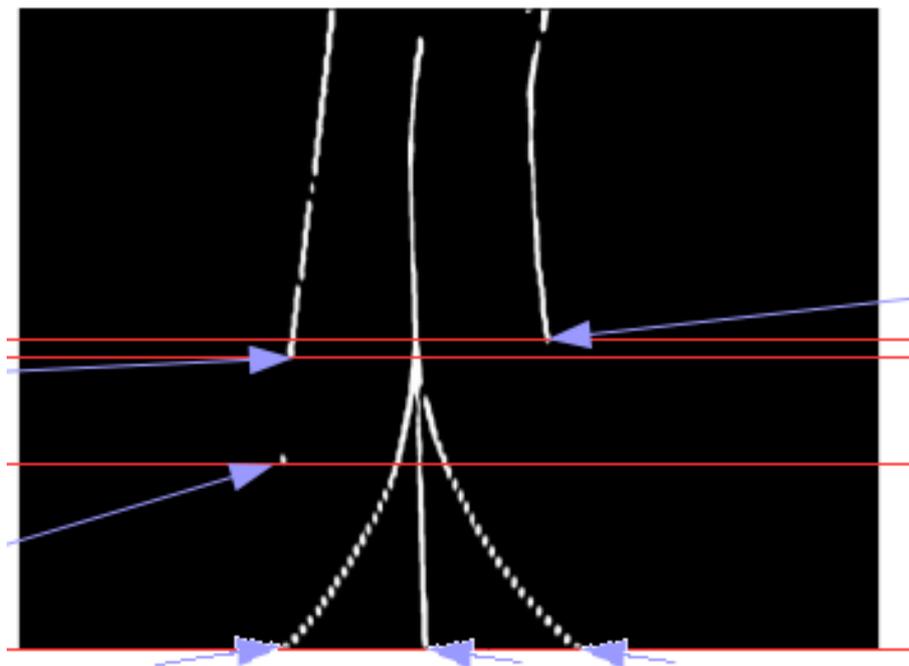


Figure 4.4: Exemple de carte binarisée. Les segments initiaux servant à l'initialisation d'un cluster de particules sont indiqués par les flèches bleues et les lignes rouges indiquent les hauteurs de la carte dans lesquelles on détecte de nouveaux départs de clusters.

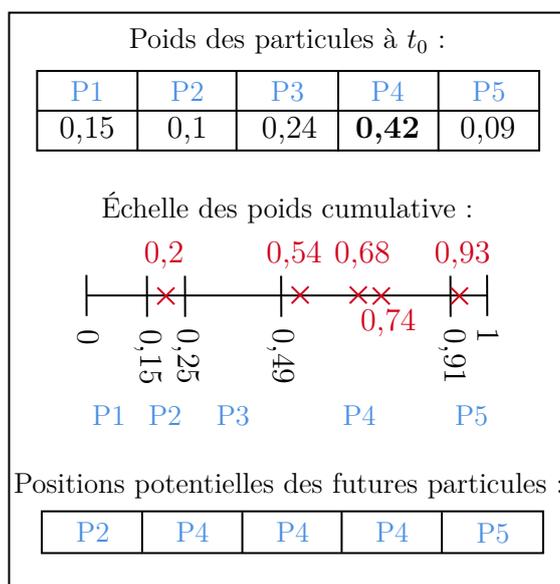


Figure 4.5: Politique de ré-échantillonnage pour 5 particules.

composante en largeur et une composante en hauteur dans l'image. Le modèle dynamique peut donc s'écrire, dans Équation 4.3, comme reliant l'état  $X_{t+1}$  à l'état  $X_t$  par une action :

- d'une distribution gaussienne centrée autour de particules sélectionnées, sur la composante  $u$  (largeur) des particules, notée  $U_{t_u}$ .
- du pas d'échantillonnage  $l$  sur la composante  $v$  (hauteur) des particules;

$$X_{t+1} = X_t + \begin{bmatrix} 0 \\ l \end{bmatrix} + \begin{bmatrix} U_{t_u} \\ 0 \end{bmatrix} \quad (4.3)$$

La Figure 4.6 présente le principe de distribution gaussienne centrée autour de particules sélectionnées par le tirage aléatoire effectué ci-dessus. La position en  $u$  des particules futures est estimée par application d'une gaussienne centrée autour de la position des particules potentielles. Nous fixons l'écart-type de la gaussienne à 10.

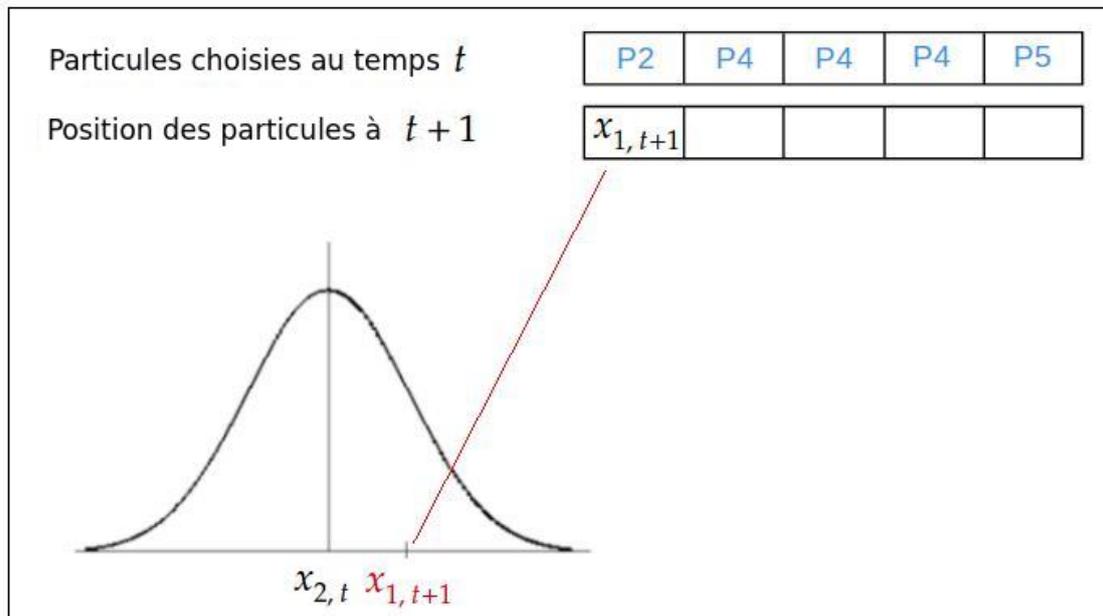


Figure 4.6: Sélection des futures particules.

Notre filtre particulaire comporte plusieurs clusters de particules qui évoluent de manière synchrone au sein de l'image. Ces derniers peuvent se rejoindre spatialement, démarrer ou s'arrêter à des instants distincts. C'est pourquoi, avant d'effectuer une nouvelle itération du filtre particulaire, nous devons appliquer des traitements complémentaires décrits ci-dessous.

#### 4.3.2.c Traitements complémentaires des groupements de particules avant itération

Comme nous l'avons dit plus tôt, les marquages dans les zones aéroportuaires sont difficile à modéliser puisque, contrairement à ceux que l'on retrouve sur le réseau routier, le nombre de lignes est variant et son motif est complexe. Lors de son exécution, notre implémentation du filtre particulaire parcourt l'image toutes les  $l$  lignes (fixé par le modèle dynamique) et de nouveaux objets peuvent alors apparaître. Pour gérer ces différents cas, une nouvelle phase d'initialisation de groupements de particules est toujours effectuée avant de passer à l'itération de ligne suivante.

D'autres peuvent au contraire disparaître. D'autre part, les phénomènes d'occultations ou de non continuité, du marquage par exemple, peuvent déclencher de nouvelles détections d'objets alors que ces pixels blancs appartiennent au même objet recherché. Pour gérer ces cas, nous utilisons plusieurs valeurs de seuil. Une

probabilité de survie est calculée pour chaque particule, en fonction de son poids. La probabilité de survie du groupement entier est alors calculée à partir de toutes les probabilités de survie des particules qu'il contient suivant l'Équation 4.4 où  $w_i$  est le poids d'une particule et  $p_i$  est la probabilité de survie de la particule. Lorsque la probabilité de survie  $p_C = \sum p_i$  du groupement de particules devient trop faible et descend en dessous d'un seuil d'acceptabilité, le groupement est éliminé. Dans le cas contraire, on définit deux états possibles : une probabilité de survie forte (représentée en vert sur la Figure 4.7(a)) et une probabilité de survie faible (représentée en rouge).

$$p_i = \begin{cases} 1, & \text{si } w_i \geq t_{SL} \\ w_i, & \text{sinon} \end{cases} \quad (4.4)$$

Dans le cas de fusion ou de séparation de groupements de particules, nous avons décidé d'étudier le barycentre de chaque groupement. En se basant sur leurs proximités, nous pouvons décider de pratiquer une fusion ou une séparation. Une illustration de ces cas est présentée en Figure 4.7.

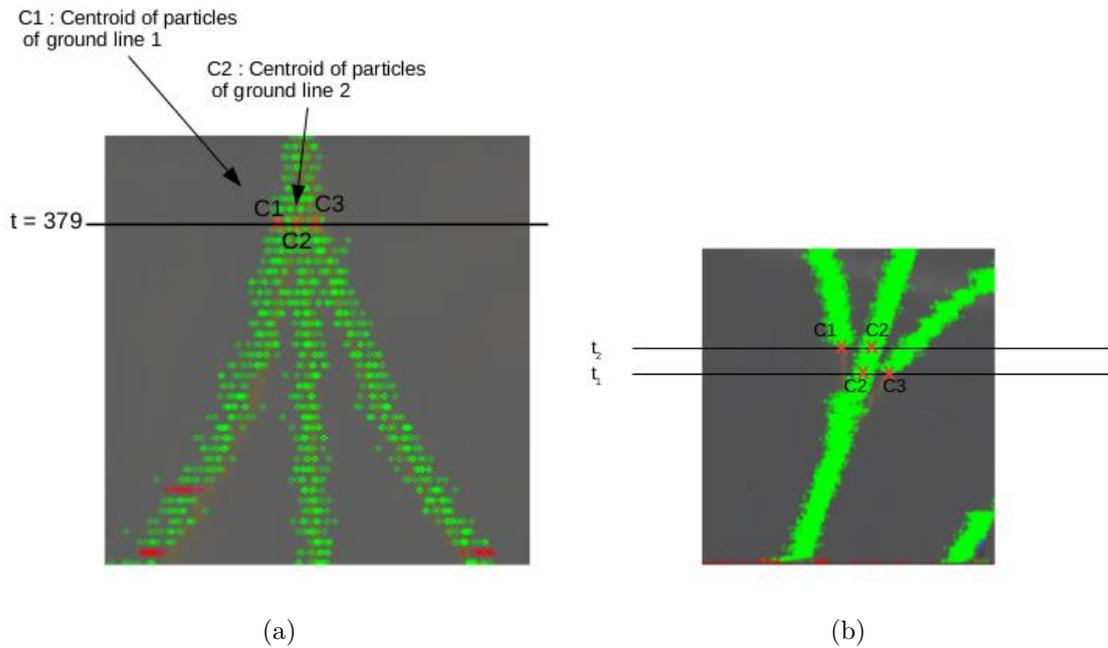


Figure 4.7: Cas de la détection de ligne : (a) Fusion de groupement de particules, (b) Détection de nouveaux débuts de groupements de particules.

### 4.3.3 Cas des marquages apparaissant comme "horizontaux" dans le plan du sol

Certains marquages peuvent, lors de la projection au sol par la transformation IPM, apparaître comme "horizontaux", c'est à dire que les pixels la représentant se trouvent tous sur une même ligne de la carte du sol, perpendiculaire à la direction de propagation que nous avons choisi.

Dans ce cas, notre choix de parcourir la carte du sol de bas en haut, ligne par ligne, ne va pas nous permettre de détecter ces marquages. Une solution simple est d'effectuer un deuxième passage du filtre particulaire sur la carte du sol transposée.

Cela permet la détection des marquages précédemment détectés comme perpendiculaires à la direction de propagation. S'il existe d'autres marquages courbes ou diagonaux dans le plan du sol par exemple, ils seront à nouveau détectés par le filtre particulière. Après cela, nous combinons les résultats des deux détections dans la carte du sol classique.

A la suite de la combinaison des deux applications du filtre particulière, nous obtenons une liste de clusters de particules. Nous appliquons quelques traitements aux clusters obtenus afin de récupérer les motifs recherchés, ce que nous présentons dans la section suivante.

#### 4.3.4 Post-traitement des clusters de particules

Une fois que le filtre particulière a parcouru l'image, la sortie du bloc algorithmique est une liste des positions des particules de chacun des clusters pour chaque itération du filtre. Pour chaque itération, on possède aussi l'information de la particule la plus probable de représenter l'objet recherché à cet instant.

S'agissant du filtre particulière dans le cadre de la détection spatiale dans l'image, une itération du filtre nous donne simplement un objet en cours de construction. Ce qui est intéressant, ce sont les points de départ et de fin de ces objets : la ligne sur laquelle il a été initialisée, celle sur laquelle il s'arrête car il n'est plus détecté, les pixels qui le composent. Il nous faut alors extraire les motifs d'intérêt (délimitations du tarmac, marquages au sol et balises) de ces sorties.

##### 4.3.4.a Extraction des marquages et des délimitations tarmac/herbe

La transformation *IPM* nous permet de travailler avec une perspective corrigée. De ce fait, les marquages au sol représentant des lignes de guidages ainsi que les bordures du tarmac peuvent être assimilés à des droites dont on approxime l'équation en sélectionnant, pour chaque itération du filtre, les particules les plus probables d'appartenir à un motif recherché. Ces particules représentent des points potentiels de la droite affine dont on cherche l'équation que l'on peut trouver en appliquant un algorithme d'ajustement de droite (par des méthodes d'approximation linéaires ou quadratiques par exemple). Ces droites peuvent être horizontales, obliques ou verticales, puisque leur équation est exprimée dans le repère de coordonnées des cellules de la carte du sol.

Il est cependant possible de trouver des marquages représentés par des courbes. Pour l'instant, nous nous focalisons sur les marquages assimilables à des droites affines en utilisant une approximation polygonale afin de redécouper ces zones courbes en de multiples segments.

Lors d'une occultation, d'une mauvaise précision due à la transformation *IPM* ou lorsque le marquage est effacé ou en pointillé par exemple, il arrive qu'un objet unique soit représenté par plusieurs groupements de particules, et donc plusieurs segments. Nous ajoutons alors une phase de fusion des segments pour corriger cet effet, comme dans l'exemple en Figure 4.8.

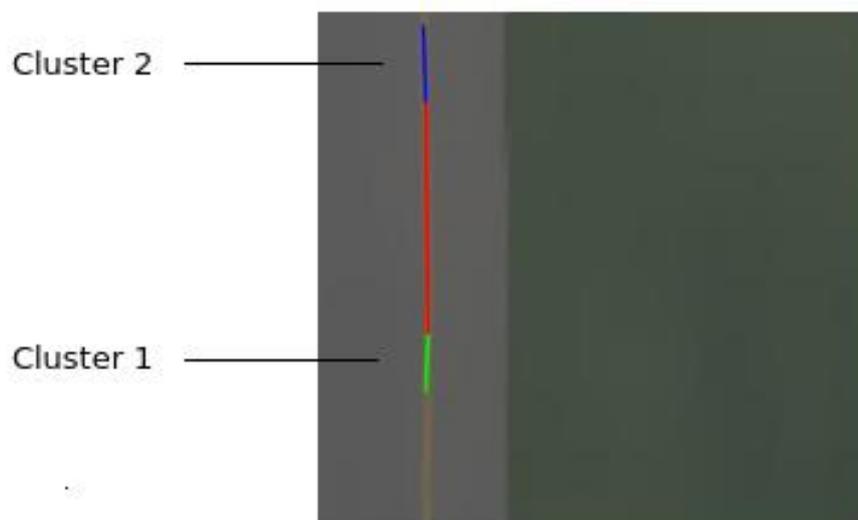


Figure 4.8: Groupements de particules à fusionner en bleu et en vert, occultation simulée en rouge.

#### 4.3.4.b Extraction des balises

Lorsque l'on applique la transformation IPM à l'image originelle, les pixels représentant les balises, qui se trouvent un peu au dessus du sol, se retrouvent déformés dans la carte du sol, comme on peut le voir en Figure 3.12. En utilisant le filtre particulaire de la même façon que pour les marquages et les bordures du tarmac, on peut extraire des cartes de poids significatives pour la détection des balises. De même, la sortie du filtre particulaire va être composée de groupements de particules issues de la détection des balises. Dans ce cas précis, nous estimons que la particule la plus probable à l'état  $t_0$  de chaque groupement de particules correspond à la position probable optimale de la balise.

Dans la section suivante, nous présentons les résultats du filtre particulaire pour la détection des marquages au sol ainsi que des balises, qui sont les informations principales à remonter au pilote. Dans la suite du chapitre, nous effectuons une étude comparative du filtre particulaire uniquement dans le cadre de la détection des marquages au sol, la méthode basée réseaux de neurones choisie ne permettant pas de comparer la détection des balises et de la séparation tarmac/herbe.

#### 4.3.5 Résultat du filtre particulaire

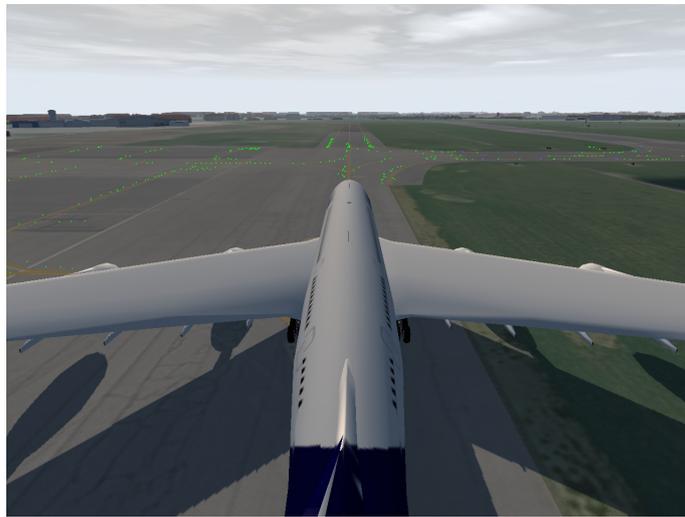
Pour illustrer les résultats de l'algorithme, nous générons une image synthétique à l'aide du simulateur de OKTAL-SE, pour un cas d'intersection de marquages au sol, comme l'on peut voir en Figure 4.2(b), qui est un cas commun dans les zones aéroportuaires.

La Figure 4.10 donne une vue globale des résultats de l'algorithme. Dans la Figure 4.10(a) les pixels verts représentent une estimation de la position des balises. La Figure 4.10(b) montre les résultats du filtre particulaire pour les droites "verticales", et les points rouges représentent les groupements de particules de faible probabilité. Les estimations des droites représentant les marquages au sol sont soulignées en vert dans la Figure 4.10(d). Les Figures 4.10(c) et 4.10(e) donnent les

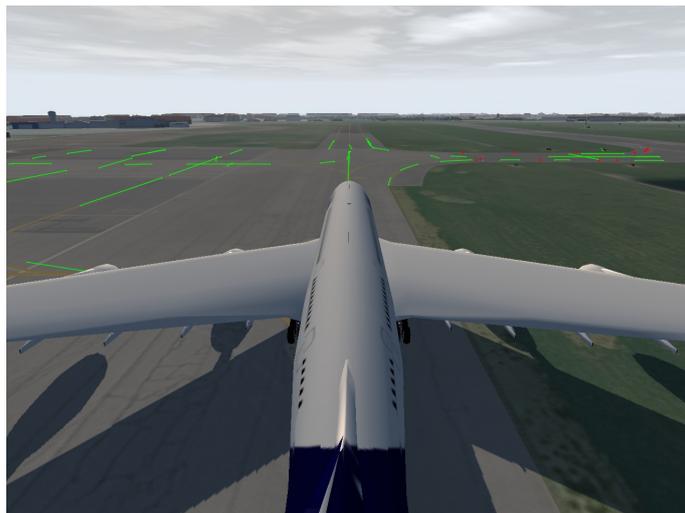
résultats du filtre appliquée à la carte du sol transposée.

Pour produire une image utile au pilote, nous devons renvoyer les résultats dans l'image d'origine en appliquant une homographie inverse qui permet alors de surligner les marquages (en vert) dans l'image originelle comme l'on peut voir en Figure 4.9, où la Figure 4.9(a) montre une fusion des détections de marquages des Figures 4.10(b) et 4.10(c). Les balises sont représentées en rouge dans la Figure 4.9(b).

Il est clair que les résultats bruts de l'algorithme devraient être traités afin de n'afficher que les informations pertinentes pour le pilote, par exemple uniquement l'axe du taxiway sur lequel se trouve l'avion. Dans ce cas, un post-traitement potentiel pour extraire cet axe serait de fusionner les informations données par les marquages, les limites tarmac/herbe et les balises.



(a)



(b)

Figure 4.9: (a) Positions de particules de différents groupements après passage du filtre particulaire sur l'intégralité de l'image, rapporté à l'image originelle, (b) Combinaison de l'estimation des marquages (vert) et des balises (rouge).

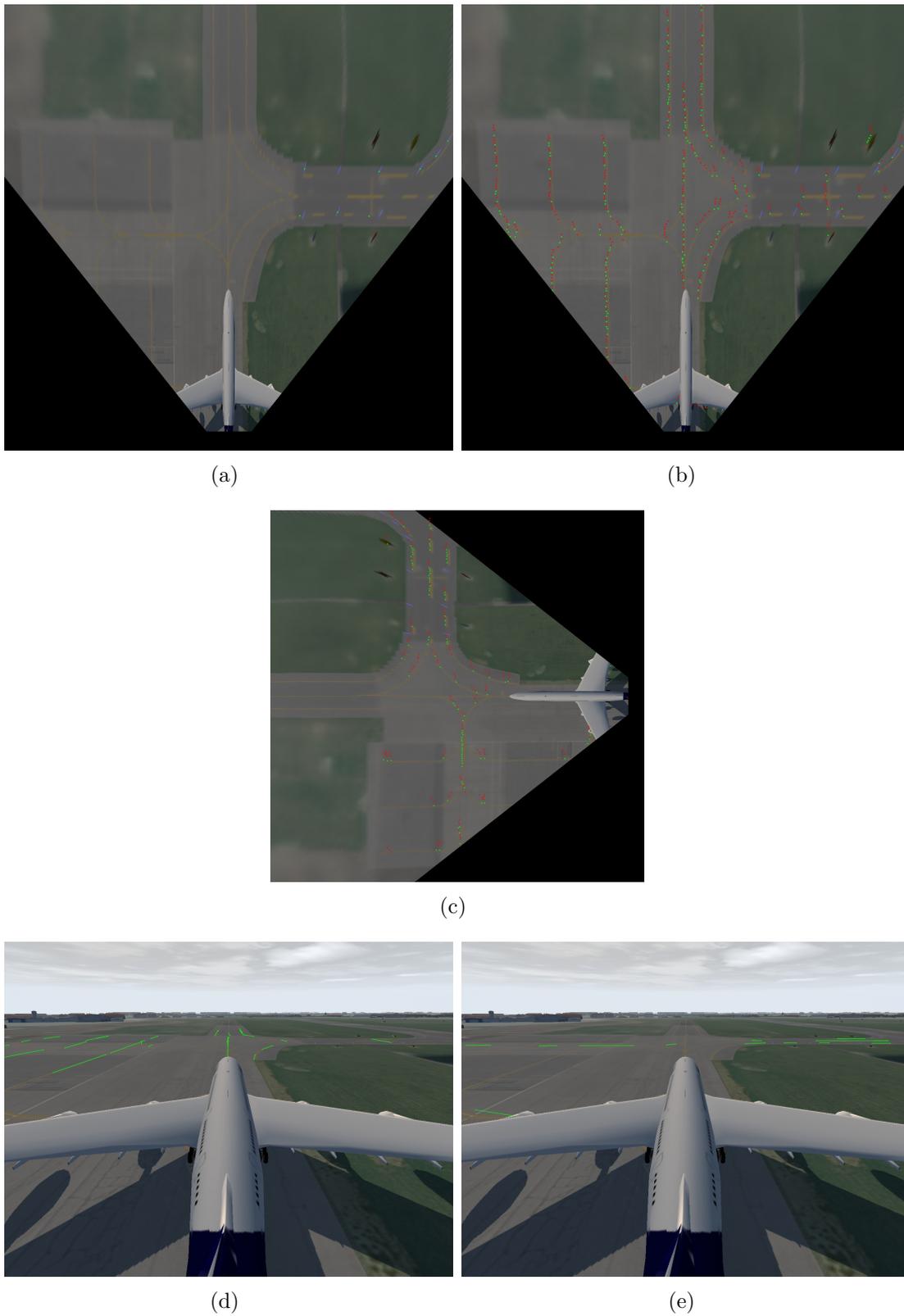


Figure 4.10: (a) Détection de balises, (b) Positions de particules de différents groupements après passage du filtre particulaire sur l'intégralité de la carte du sol, (c) Positions de particules de différents groupements après passage du filtre particulaire sur l'intégralité de la carte du sol transposée, (d) Détection de marquages rapportée à l'image originelle, (e) Détection de marquages par la carte du sol transposée, rapportée à l'image originelle.

## 4.4 Amélioration du filtre particulaire

### 4.4.1 Redéfinition du modèle dynamique $h$ de propagation des particules

Notre première implémentation du filtre effectuait une propagation des particules seulement selon l'axe vertical de l'image, comme défini par l'Équation 4.3. Ce modèle dynamique est adapté dans le cas de suivi de marquages parallèles à la direction de propagation du filtre particulaire (dans notre cas ce sont les marquages verticaux qui sont parallèles au sens de propagation du bas vers le haut de l'image).

C'est un modèle dynamique qui est satisfaisant dans le cas où nous appliquons le filtre particulaire à la carte du sol et que l'avion suit un taxiway rectiligne. Cependant, les marquages présents dans les zones aéroportuaires sont plus complexes (combinaisons de parties rectilignes et courbes), comme on peut le voir en Figure 4.11 et ce modèle dynamique n'est alors plus suffisant.



Figure 4.11: Exemple de zones aéroportuaires.

Nous avons travaillé à l'amélioration de la propagation des particules en proposant deux nouvelles équations de propagation. L'Équation 4.5 donne l'évolution de la particule en changeant l'état de notre filtre particulaire. Le nouvel état du filtre n'est plus seulement dépendant de  $X_t$  mais aussi de  $X_{t-1}$ . Cela nous permet de garder les informations de la position précédente de la particule ainsi que celles de la position courante.

$$X_{t+1} = X_t + \begin{bmatrix} \delta \\ l \end{bmatrix} + \begin{bmatrix} U_{t_u} \\ 0 \end{bmatrix} \quad \text{avec } \delta = X_{t_u} - X_{t-1_u} \quad (4.5)$$

Nous pouvons alors définir un axe de propagation qui viendra corriger la position de la prochaine particule par rapport à une propagation verticale basique afin d'améliorer la détection des marquages courbes. La composante  $X_{t+1_u}$  (largeur)

de l'état est donc fonction d'une part de la distribution gaussienne mais aussi des composantes  $X_{t_u}$  et  $X_{t-1_u}$  des états  $X_t$  et  $X_{t-1}$ .

Nous remarquons que cette méthode permet de continuer à suivre une ligne lorsqu'elle est oblique et non plus seulement verticale ou horizontale. Cependant, dans les deux premières équations, le pas est fixe. Lorsque nous essayons de suivre des courbes, il se peut que la variation de position en abscisse soit trop grande pour être corrigée. Dans ce cas-là, nous préférons réduire le pas de propagation, comme proposé par l'Équation 4.6.

$$X_{t+1} = X_t + \left[ l * \left( 2 - \frac{\delta}{(1+e^{-|\delta|})} \right) \right] + \begin{bmatrix} U_{t_u} \\ 0 \end{bmatrix} \quad (4.6)$$

La Figure 4.12 présente les résultats pour les trois méthodes de propagation : initiale, compte tenu de l'état précédent et à pas variable. Ici l'entrée est une image de synthèse créée sous Matlab, avec un seul marquage non rectiligne. La propagation des particules s'effectue du haut vers le bas de l'image.

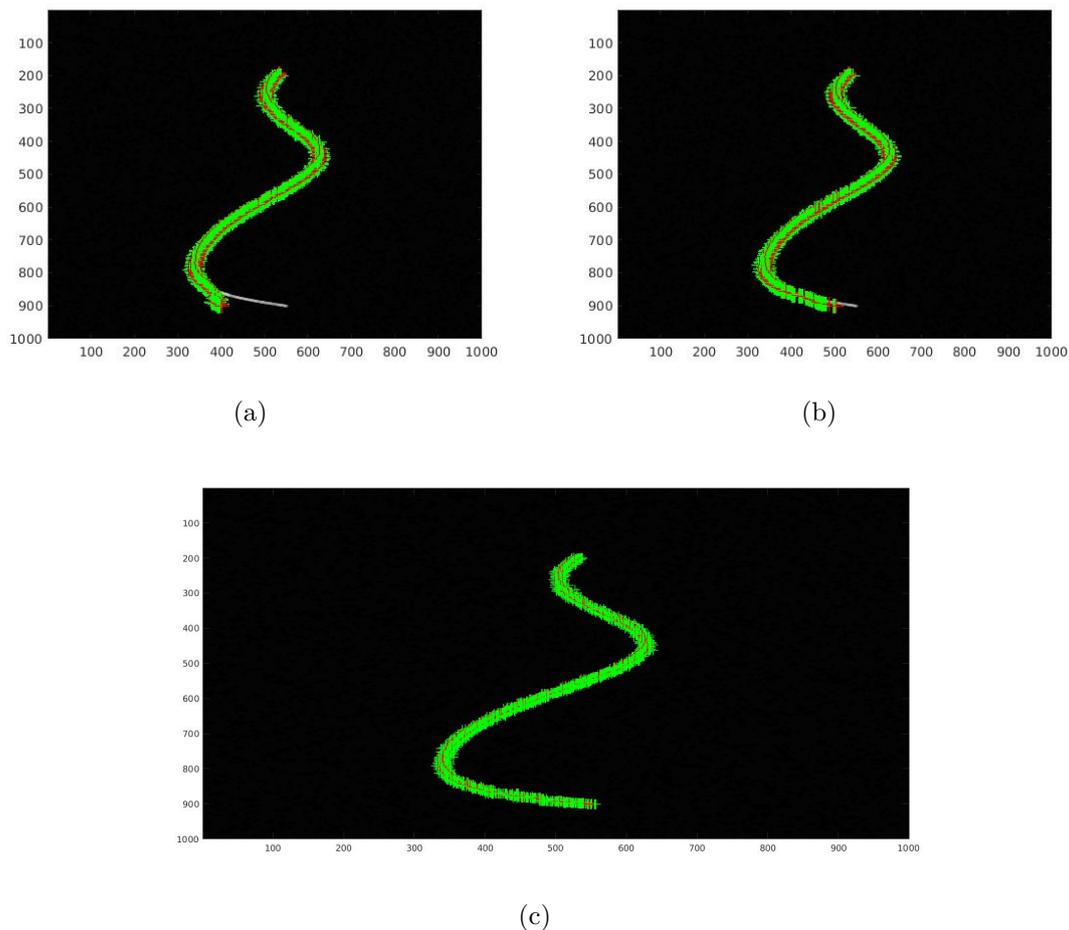


Figure 4.12: Résultats d'une propagation du haut vers le bas de l'image pour les méthodes de propagation (a) initiale, (b) compte tenu de l'état précédent et (c) à pas variable.

#### 4.4.2 Participation à l'élaboration d'un filtre particulière fenêtré

Une autre thèse est effectuée en parallèle dans l'équipe par E. Perrotin, débutée en 2019. Ensemble, nous avons travaillé à définir une nouvelle version du filtre particulière basée sur une étude présentée dans [166].

Le principe est de représenter l'état comme une fenêtre de pixels dont la taille peut varier pour s'adapter à la profondeur dans l'image et donc à la distance du motif à détecter à la caméra. Cette méthode est donc plus adaptée au suivi de marquages dans l'image d'origine, sans application de la transformée IPM. Combinée à une meilleure appréciation de la couleur de référence des marquages, cela devrait réduire le nombre de fausses détections et améliorer la précision.

Comme pour le filtre particulière précédent, il reste donc à définir le vecteur d'état  $X_t$ , le modèle dynamique (la fonction  $f$  de l'Équation 4.1), la fonction d'évaluation (la fonction  $g$  de l'Équation 4.2), la méthode de ré-échantillonnage (qui dépend des distributions a priori de  $U_t$  et  $V_t$  les bruits de prédiction et d'observation) et les critères d'initialisation, de fusion/séparation des clusters de particules et d'arrêt du filtre.

Nous proposons le vecteur d'état  $X_t = (x, y, w, h, \theta)_t$  avec  $(x, y)$  les coordonnées du centre de la particule,  $w$  et  $h$  la largeur et la hauteur de la fenêtre et  $\theta$  la rotation du motif. Le but est de suivre des fenêtres rectangulaires qui potentiellement, peuvent changer d'orientation quand on parcourt l'image.

Le modèle dynamique peut être équivalent à celui de la dernière version proposée du filtre particulière. Cependant ici, le pas de propagation entre  $t$  et  $t+1$  est calculé en ligne et colonne, en fonction de l'état courant  $X_t$ , donc de largeur, hauteur et orientation de la fenêtre, dépendant des courbures maximales définies par les recommandations OACI pour  $\theta$  et fonction de la profondeur dans l'image pour  $w$  et  $h$ . De même pour la définition de la méthode de ré-échantillonnage.

Les possibilités pour la fonction d'évaluation sont plus importantes que dans le cas précédent. On peut penser à plusieurs méthodes d'évaluation du motif observé d'une particule par rapport aux observations. Parmi elles, on peut citer l'utilisation de métriques standard (PSNR/SSIM) en comparant avec une banque pré-établie de motifs, l'utilisation de descripteurs ou d'hypothèses sur la couleur, les contrastes ou la structure du motif (filtres de Gabor) ou le recours à des méthodes de classification.

La Figure 4.13 donne un exemple d'évolution des motifs observables  $P_t$  par un filtre particulière fenêtré. Ma contribution à ce nouvel algorithme a été purement théorique : c'est E. Perrotin qui est en charge de son implémentation logicielle. Cette adaptation du filtre particulière a mené à une publication commune à ORASIS [?].

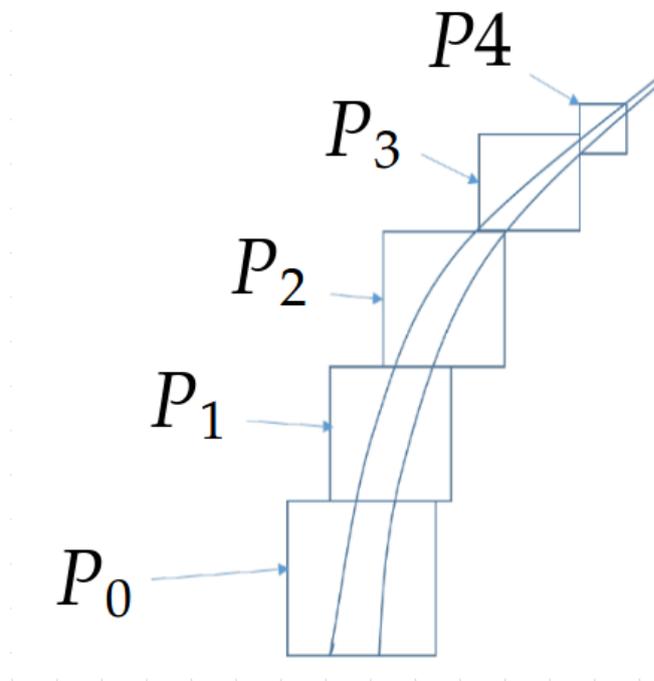


Figure 4.13: Schématisation de motifs observables par un filtre particulaire fenêtré pour cinq particules avec une rotation fixée comme nulle de la fenêtre.

## 4.5 Comparaison des performances du filtre particulaire par rapport à deux autres méthodes

Nous avons décidé de comparer la méthode du filtre particulaire avec deux autres méthodes : la transformation de Hough et une approche basée sur des réseaux de neurones. La Figure 4.14 met en lumière les similarités et les différences entre ces méthodes, en précisant les entrées et sorties de chaque méthode.

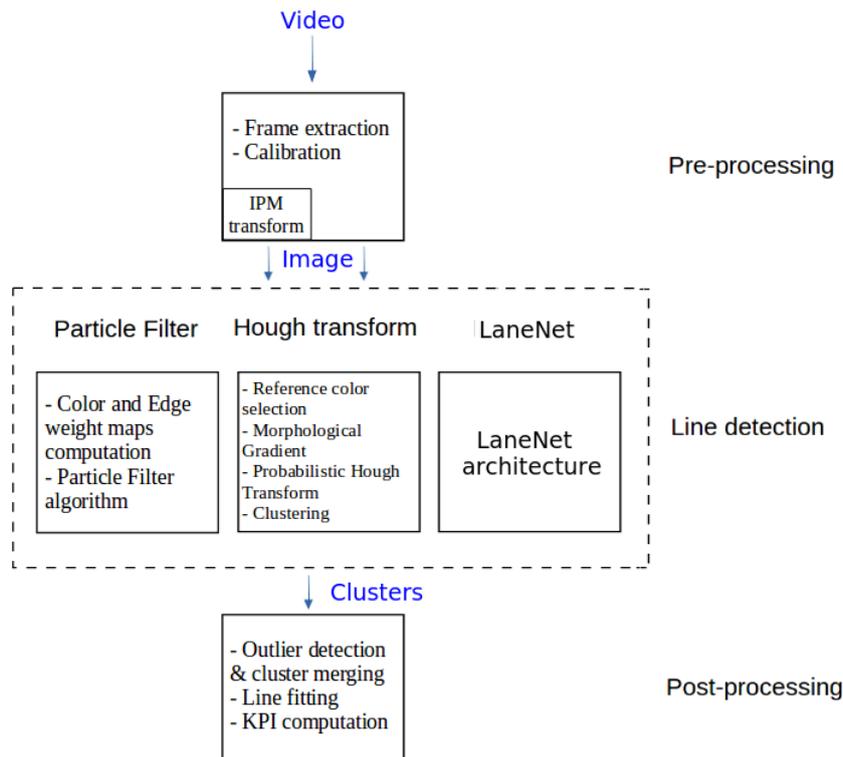


Figure 4.14: Définition des trois méthodes utilisées.

Avant d'appliquer une des trois méthodes, nous appliquons plusieurs algorithmes de pré-traitement, comme de la correction de distorsion pour les images réelles. Nous avons décidé, pour notre comparaison, de séparer l'étape de transformation IPM du pré-traitement global. Nous passons donc chacune des méthodes sur des images avec et sans la transformation IPM.

Les méthodes de filtre particulaire et de transformation de Hough utilisent un algorithme basé sur la couleur pour effectuer la détection de ligne. Nous avons décidé de différencier ces algorithmes pour les deux méthodes et de ne pas les regrouper dans un pré-traitement commun. L'algorithme de détection de couleur utilisé par le filtre particulaire se base sur les résultats de l'étude colorimétrique présentée dans le Chapitre 3 (on retrouve couramment dans la littérature l'utilisation de HSV, comme dans [113] ou de CIELAB comme dans [84]) alors que l'algorithme utilisé pour la transformée de Hough est basé sur une définition du jaune dans l'espace de couleur XYZ.

La transformée de Hough prend comme entrée une image binaire obtenue grâce à une extraction de caractéristiques. Comme nous voulons utiliser la couleur comme

caractéristique, nous définissons un sous ensemble de l'espace de couleur et un seuil pour obtenir l'image binaire. L'implémentation de la méthode par Transformée de Hough, a été faite à partir de OpenCV, par E. Perrotin.

En revanche, le filtre particulaire est alimenté par une carte de poids qui représente la probabilité d'un pixel d'appartenir à une ligne et qui est une combinaison d'une carte de couleur et d'une carte de distance.

Les trois méthodes possèdent la même sortie, qui correspond à une liste de groupes, dits clusters, de pixels. Ces clusters représentent un ensemble de pixels considérés comme définissant un marquage ou une partie de marquage dans l'image. Ils peuvent être composés de pixels consécutifs (dans le cadre du filtre particulaire ou de l'approche par réseau de neurones) ou comme un ensemble de pixels représentant un segment (dans le cadre de la transformée de Hough).

## 4.5.1 Présentation de la comparaison

### 4.5.1.a But de la comparaison

Cette comparaison possède trois buts. Le premier est de comparer les résultats des trois méthodes sur diverses images avec une vérité terrain, ici extraite interactivement sur chaque image (dans une étude à venir, cette vérité-terrain pourrait être donnée pour les images de synthèse, par le simulateur de OKTAL SE). Le second but est de questionner la pertinence de l'utilisation de la transformation IPM comme algorithme de pré-traitement et sa contribution à l'amélioration des performances des algorithmes. Le dernier but est de tester la robustesse des algorithmes face à des images en conditions météorologiques dégradées. La comparaison s'effectue sur un ensemble d'images réelles et simulées, choisies de façon à faire varier les conditions d'acquisition et la qualité. Nous disposons ainsi d'images acquises par des caméras placées (ou simulées) dans le cockpit et dans la dérive de l'avion, de manière à évaluer la détection des marquages en fonction de la position du système d'acquisition sur l'aéronef. Le simulateur nous permet d'ajouter, à cet ensemble de comparaison, des images impactées par des conditions météorologiques dégradées par le brouillard ou la baisse de luminosité ambiante.

### 4.5.1.b Critères utilisés

Pour comparer ces trois algorithmes, nous avons décidé de baser notre analyse sur plusieurs critères définis ci-dessous.

1. Critère basé sur la portée de détection

- **La zone maximum de détection**

Un de nos objectifs est d'avoir une large zone de détection puisque la vitesse de l'aéronef implique que les décisions soient prises aussi tôt que possible. Cette valeur représente la hauteur maximum de détection dans l'image.

2. Critères basés sur la précision de la détection

- **Moyenne et écart type** de la distance de groupes de pixels à la vérité terrain

Ce sont les critères principaux pour quantifier la précision des algorithmes. Ils calculent pour chaque groupe de pixels, la distance entre un pixel proposé et la vérité terrain.

- **Rappel**

Nous avons décidé d'utiliser ce critère pour analyser la pertinence de l'algorithme en divisant le nombre de vrais positifs par le nombre total de pixels renvoyés par la méthode.

Le résultat des différents critères est donné dans la Section 4.5.2. Ils sont calculés sur différentes tranches de l'image ainsi que sur l'image globale. Ces tranches correspondent, avec une hypothèse de sol plat, à une distance approximative de la représentation d'un point de l'espace (dans le plan du sol) à la caméra. Pour la caméra placée dans la dérive de l'avion, avec une résolution de 1280pX960p, les tranches sont définies comme suit : la tranche "0m-100m" qui correspond au bas de l'image et est composée des lignes 0 à 553, la tranche "100m-250m" qui correspond au milieu de l'image et est composée des lignes 554 à 694 et la tranche "250m+" qui correspond au haut de l'image et est composée des lignes 695 à 959. Pour la caméra placée dans le cockpit, avec une résolution de 1280pX960p, les tranches sont respectivement composées des lignes 0 à 430, 431 à 465 et 466 à 959.

## 4.5.2 Résultats de la comparaison

### 4.5.2.a Performance générale des méthodes

Le Tableau 4.1 présente les différents critères calculés pour des images réelles et simulées, sans pré-traitement par la transformation IPM et par temps clair. Entre parenthèses, nous indiquons le nombre d'images utilisées pour obtenir les résultats. Leur nombre est faible mais nous prévoyons d'augmenter la base de données pour valider ces résultats sur un grand nombre de données. La Figure 4.15 fournit une représentation graphique des résultats sur l'image globale.

Nous pouvons remarquer que la plupart des pixels détectés se trouvent dans les tranches basses et moyennes de l'image, ce à quoi on s'attend vu que la dernière tranche possède très peu de pixels distinguables puisqu'à plus de 250m des caméras. Le rappel baisse donc de manière significative dans cette dernière tranche.

En général, le rappel du filtre particulaire est supérieur à celui des deux autres méthodes. Cependant, il faut relativiser les performances du rappel au regard de la précision de l'algorithme. Le filtre particulaire a une moins bonne précision que la méthode à base de transformée de Hough. C'est en partie parce qu'il retourne une grande partie de points pour chaque cluster. Cela améliore le rappel lorsque les clusters convergent vers les motifs à détecter mais le grand nombre de particules impacte la précision. Au contraire, la transformée de Hough renvoie peu de points mais qui sont en moyenne plus précis. Cependant, quelques fausses détections peuvent dégrader rapidement le rappel de la méthode.

Les performances de LaneNet sont quantitativement mauvaises. Cependant, ces performances sont liées à notre système d'annotation de la vérité terrain ainsi qu'à son entraînement. Le réseau de neurones essaie d'extrapoler des marquages pour mieux décrire les voies de circulation. Notre but est de détecter les lignes présentes dans la vérité terrain. Ainsi l'extrapolation de la ligne sous les ailes de l'avion, par

exemple, impacte le rappel et la précision quantitativement alors que qualitativement elle améliore la robustesse de la détection.

Camera - Type (nb images)		Portée maximale (en pixels)				Rappel		
		HT	PF	CNN	GT	HT	PF	CNN
Cockpit - Réelle (2)	Global	394	408.5	<b>563</b>	607	<b>0.89</b>	0.88	0.14
	0m-100m					0.89	0.95	0.16
	100m-250m					0	0.1	0.02
	250m+					0	0	0
Dérive - Réelle (10)	Global	407.5	564.75	<b>721.13</b>	810.56	0.42	<b>0.61</b>	0.45
	0m-100m					0.44	0.63	0.55
	100m-250m					0.03	0.32	0.14
	250m+					0	0.05	0.31
Cockpit - Simu (13)	Global	<b>454</b>	452.2	295.3	477.54	0.86	<b>0.94</b>	0.42
	0m-100m					0.91	0.94	0.45
	100m-250m					0.52	0.87	0.04
	250m+					0.43	0	0
Dérive - Simu (13)	Global	<b>719.92</b>	690.69	563.77	745.15	0.71	<b>0.79</b>	0.09
	0m-100m					0.94	0.83	0.1
	100m-250m					0.73	0.75	0.03
	250m+					0.38	0.29	0.01

Camera - Type (nb images)		Distance Moyenne			Écart-Type		
		HT	PF	CNN	HT	PF	CNN
Cockpit - Réelle (2)	Global	<b>0.17</b>	9.21	55.92	0.4	26.73	77.81
	0m-100m	0.17	0.52	66.57	0.4	3.99	82.54
	100m-250m	0	109.88	11.67	0	63.45	10.03
	250m+	0	84.24	10.71	0	65.39	10.03
Dérive - Réelle (10)	Global	14.68	<b>12.77</b>	37.7	20.54	23.28	64.02
	0m-100m	4.49	5.77	43.51	8.22	11.79	76.34
	100m-250m	29.1	7.55	44.6	16.81	22.72	28.26
	250m+	50.9	41.84	29.92	2.26	11.38	15
Cockpit - Sim (13)	Global	<b>1.67</b>	1.83	9.99	3.79	4.47	21.64
	0m-100m	1.61	2.61	19.04	3.54	8.22	41.01
	100m-250m	0.95	15.05	18.16	1.13	37.52	7.72
	250m+	0.1	1.4	0.77	0.11	0	0.42
Dérive - Simu (13)	Global	<b>0.62</b>	15.38	113.9	1.27	33.98	107.9
	0m-100m	0.03	41.28	162.5	0.13	48.11	120
	100m-250m	0.59	2.88	38.84	1.28	10.63	33.54
	250m+	1.42	19.85	2.58	1.03	7.25	1.46

Table 4.1: Tableau de résultat pour des images réelles et simulées sans pré-traitement IPM ou conditions météorologiques dégradées.

#### 4.5.2.b Impact de la transformation IPM

La transformation IPM demande une bonne estimation de la position de la caméra sur l'avion, que nous n'avons pas eue avec les images réelles. Nous évaluons principalement l'intérêt de la transformation IPM sur les images synthétiques, acquises par des caméras positionnées dans la dérive et à l'avant de l'avion, avec une résolution de 1280pX960p et un champ de vue horizontal de 80°. Nous fournissons aussi les résultats pour les images réelles avec une position approximative des caméras mais notons que la précision de ces approximations peut grandement impacter les résultats des algorithmes.

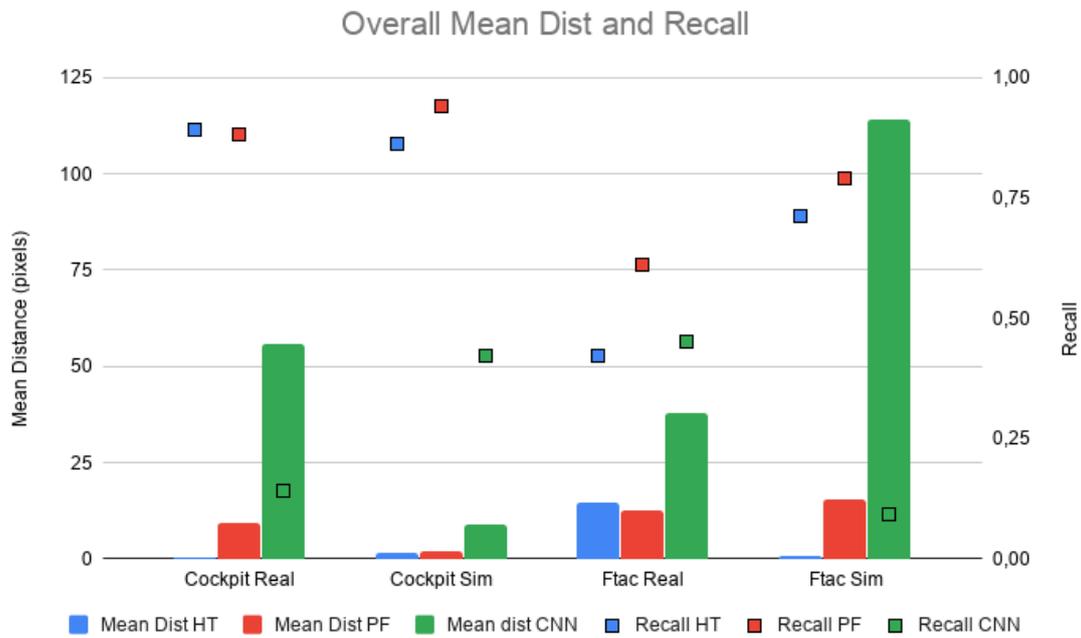


Figure 4.15: Performance des trois méthodes sur l'image globale.

Les lignes 1, 2, 3 et 5 du Tableau 4.2 sont à mettre en relation avec les lignes 1, 2, 3 et 4 du Tableau 4.1 et donnent une idée de l'impact de la transformation IPM sur les performances des méthodes. Les lignes 4 et 6 du Tableau 4.2 sont utilisées à titre comparatif avec les lignes 3 et 5 du même tableau, puisqu'elles indiquent les performances des algorithmes en conditions météorologiques dégradées. La Figure 4.16 reprend les résultats des tableaux pour une étude comparative graphique.

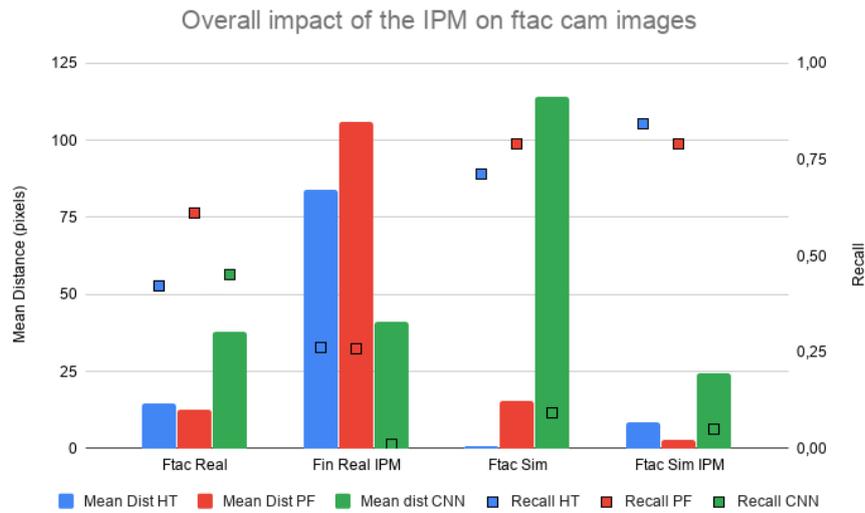
L'utilisation de la transformation IPM n'améliore pas drastiquement le rappel des trois méthodes. Pour LaneNet, les résultats sont dégradés, probablement à cause de la nature de son entraînement. Ces résultats sont prévisibles. La transformation IPM ajoute du bruit à l'image puisque l'on essaie de reconstruire une information qui n'est pas forcément présente dans l'image initiale. Ce pré-traitement améliore la portée de détection, comme on peut le voir sur la Figure 4.17 mais réduit la précision et le rappel de chacune des méthodes. Ceci est dû aux interpolations utilisées pour passer de la vue initiale à la vue du dessus. Un pixel choisi dans l'image ayant subi la transformation IPM est estimé au moyen de deux interpolations. Une fois comparé à la vérité terrain de la vue initiale, il est normal que la précision de la détection soit réduite.

D'après les résultats, on note aussi que la transformation IPM améliore les performances lorsque la position de la caméra est bien connue, comme pour les images simulées. Cependant, quand elle est appliquée aux images réelles, avec une approximation de la position de la caméra, nous remarquons que la transformation IPM a des chances de dégrader les performances. Enfin, on note que l'impact de la transformation IPM pour les images simulées n'est pas le même en fonction de la position de la caméra. De manière générale, les résultats de la tranche lointaine de l'image sont améliorés par la transformation IPM.

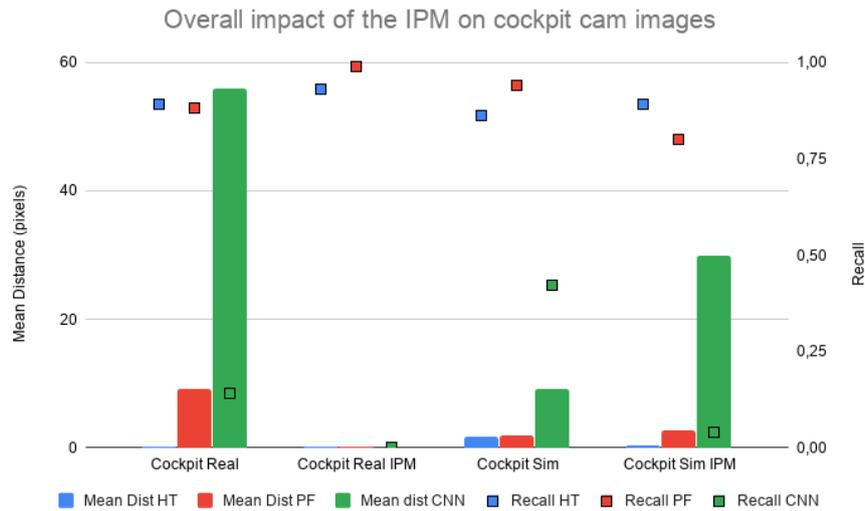
Camera - Type - Qualité (nb images)		Portée maximale (en pixels)				Rappel		
		HT	PF	CNN	GT	HT	PF	CNN
Cockpit - Réelle (2)	Global	<b>125.5</b>	122.5	0	607	0.93	<b>0.99</b>	0
	0m-100m					0.93	0.99	0
	100m-250m					0	0	0
	250m+					0	0	0
Dérive - Réelle (10)	Global	270.75	<b>616.25</b>	174.63	810.56	<b>0.26</b>	<b>0.26</b>	0.01
	0m-100m					0.29	0.52	0.01
	100m-250m					0	0.26	0
	250m+					0	0.18	0.01
Cockpit - Simu (13)	Global	462.8	<b>467.4</b>	274.7	477.54	<b>0.89</b>	0.8	0.04
	0m-100m					0.8	0.77	0.03
	100m-250m					0.89	0.79	0.07
	250m+					0.5	0.75	0
Cockpit - Simu - Dégradée (13)	Global	449,9	<b>455.1</b>	179.1	477.54	<b>0.84</b>	0.56	0.02
	0m-100m					0.83	0.72	0.01
	100m-250m					0.85	0.47	0.02
	250m+					0	0.07	0
Dérive - Simu (13)	Global	703.62	<b>705.85</b>	460.62	745.15	<b>0.84</b>	0.79	0.05
	0m-100m					0.57	0.77	0.01
	100m-250m					0.85	0.76	0.06
	250m+					0.83	0.87	0.03
Dérive - Simu - Dégradée (13)	Global	655.92	<b>675.85</b>	411.38	745.15	<b>0.73</b>	0.60	0.02
	0m-100m					0.45	0.54	0.01
	100m-250m					0.84	0.56	0.02
	250m+					0.1	0.19	0.03

Camera - Type - Qualité (nb images)		Distance Moyenne			Écart-Type		
		HT	PF	CNN	HT	PF	CNN
Cockpit - Réelle (2)	Global	<b>0.19</b>	0.25	0	0.89	2.51	0
	0m-100m	0.19	0.25	0	0.89	2.51	0
	100m-250m	0	0	0	0	0	0
	250m+	0	0	0	0	0	0
Dérive - Réelle (10)	Global	84.01	105.8	<b>40.97</b>	55.93	64.89	35.33
	0m-100m	14.25	18.77	63.6	19.68	20.48	48.49
	100m-250m	114.07	140.89	18.43	16.22	54	7.16
	250m+	116,32	171.93	40.22	25.37	14.92	29.8
Cockpit - Simu (13)	Global	<b>0.33</b>	2.61	29.89	4.24	9.49	16.39
	0m-100m	0.78	28.13	138.4	1.86	65.33	31.27
	100m-250m	0.28	6.49	38.36	0.79	24.96	19.21
	250m+	0	0.33	11.25	0	1.31	0.68
Cockpit - Simu - Dégradée (13)	Global	<b>1.94</b>	10.57	19.58	3.09	15.54	9.24
	0m-100m	12.88	59.12	78.8	31	71.74	28.57
	100m-250m	1.55	19.71	24.57	0.69	36.91	37.23
	250m+	6.24	5.73	6.24	1.01	4.71	1.4
Dérive - Simu (13)	Global	8.57	<b>2.70</b>	24.39	3.54	11.72	34.23
	0m-100m	119.3	0.59	93.85	167	1.38	62.16
	100m-250m	0.22	4.38	6.99	0.64	18.23	5.72
	250m+	0.02	0.26	1.05	0.06	0.85	0.25
Dérive - Simu - Dégradée (13)	Global	49.21	<b>18.61</b>	31.36	128.7	27.64	22.27
	0m-100m	210.8	1.14	88.41	182.9	2.47	32.25
	100m-250m	0.47	29.18	12.07	1.02	41.46	13.8
	250m+	0.25	0.46	8.88	0.23	0.67	1.09

Table 4.2: Tableau de résultats des images ayant reçues une transformation IPM en pré-traitement.



(a)



(b)

Figure 4.16: Impact de la transformation IPM sur les performances d'images provenant de caméras placées (a) dans la dérive et (b) à l'avant de l'avion.

#### 4.5.2.c Impact des conditions météorologiques

La Figure 4.18 présente les différences de détection entre deux conditions météorologiques et les résultats des critères sont visibles dans le Tableau 4.3. Ce tableau est à comparer avec les lignes 3 et 4 du Tableau 4.1. On peut observer une baisse de la distance de détection maximale pour les méthodes de filtre particulaire et de transformée de Hough. Le résultat est prévisible puisque les deux méthodes reposent sur des hypothèses de couleur et de contrastes qui sont fortement impactées par le brouillard. Au contraire, la méthode LaneNet semble peu impactée, en termes de résultats seulement. Qualitativement parlant, l'algorithme détecte moins de lignes ce qui améliore virtuellement sa précision (voir Figure 4.19).

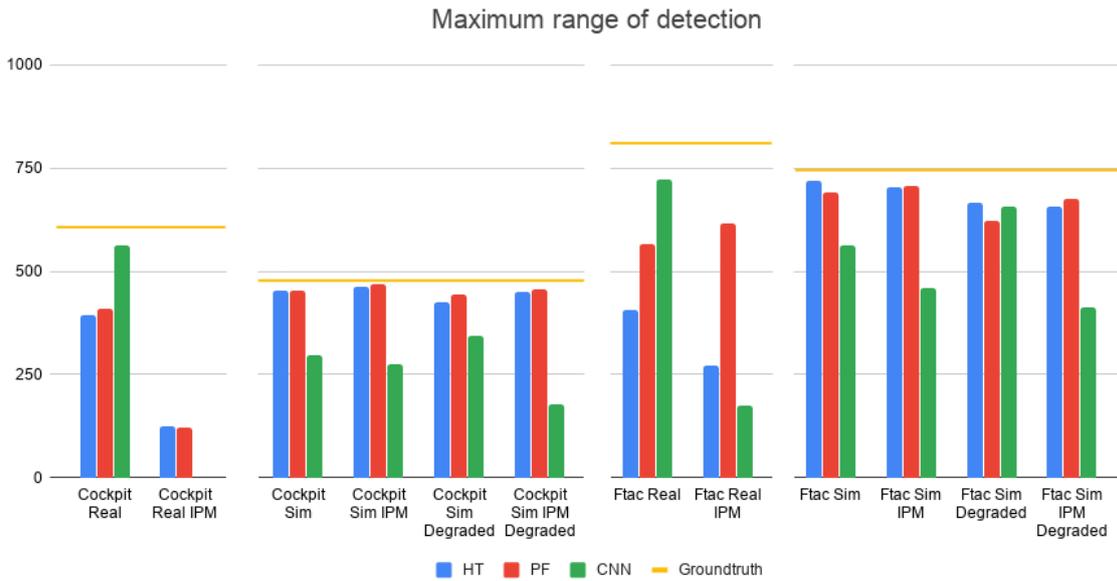


Figure 4.17: Performances des trois méthodes en fonction de la portée de détection.

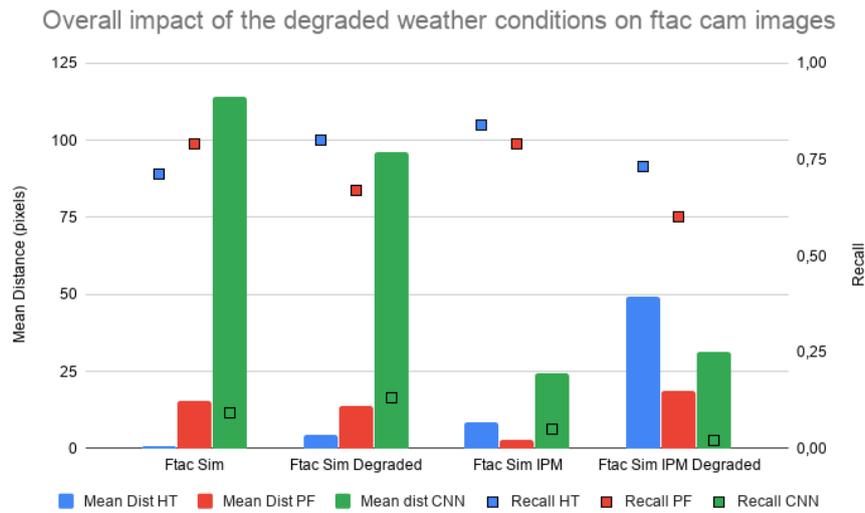
Camera - Type (nb images)		Portée maximale (en pixels)				Rappel		
		HT	PF	CNN	GT	HT	PF	CNN
Cockpit - Simu (13)	Global	423.4	<b>442.5</b>	342.9	477.54	<b>0.9</b>	0.86	0.36
	0m-100m					0.9	0.95	0.41
	100m-250m					0.69	0.63	0.05
	250m+					0	0	0.01
Dérive - Simu (13)	Global	<b>665.15</b>	622	656.92	745.15	<b>0.8</b>	0.67	0.13
	0m-100m					0.87	0.77	0.15
	100m-250m					0.75	0.57	0.02
	250m+					0.1	0.05	0.04

Camera - Type (nb images)		Distance Moyenne			Écart-Type		
		HT	PF	CNN	HT	PF	CNN
Cockpit - Simu (13)	Global	<b>1.5</b>	4.15	17.53	9.2	9.95	34.2
	0m-100m	1.63	13.97	25.44	3.52	39.08	44.34
	100m-250m	0.46	28.63	30.07	0.57	40.16	23.95
	250m+	0	17.52	12.86	0	7.95	6.73
Dérive - Simu (13)	Global	<b>4.35</b>	13.93	96.33	3.54	37.84	92.1
	0m-100m	6.31	51.38	152.2	9.82	58.4	107.6
	100m-250m	0.59	17.65	55.88	1.31	32.8	47.45
	250m+	0.75	11.71	9.47	0.44	9.6	4.82

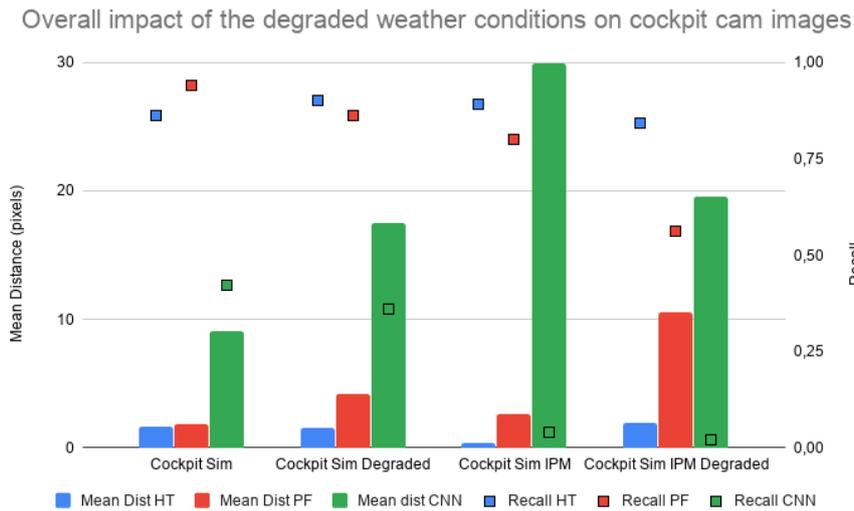
Table 4.3: Tableau de résultats pour des conditions météorologiques dégradées.

#### 4.5.2.d Comparaison qualitative

La Figure 4.19 offre une comparaison de la détection de lignes sur une même scène avec deux conditions météorologiques différentes où la Figure 4.19(a) est une version de la scène dégradée par la tombée de la nuit et la présence de brouillard. La Figure 4.19(b) représente la détection par temps clair. Nous pouvons voir que le filtre particulaire et la transformée de Hough détectent des lignes de manière homogène dans l'image. LaneNet rencontre des difficultés dues au motif particulier des marquages, spécifiques aux zones aéroportuaires, non présents dans le domaine



(a)



(b)

Figure 4.18: Impact des conditions météorologiques dégradées sur (a) les images de la caméra placée dans la dérive et (b) les images de la caméra placée à l'avant de l'avion.

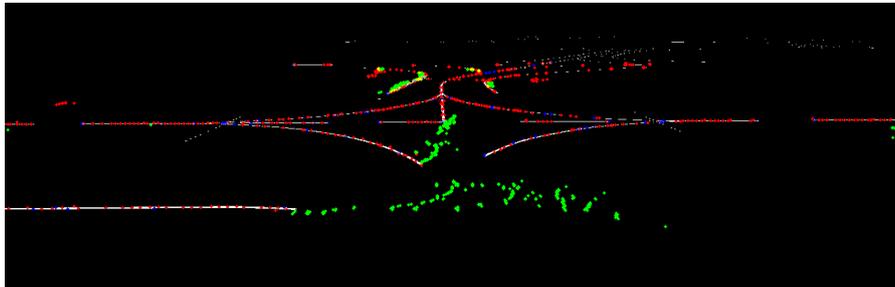
automobile. Les performances de LaneNet sont bonnes lorsque l'on peut trouver des lignes parallèles dans l'image, une configuration récurrente dans les bases de données automobiles. La détection de marquages dans la Figure 4.19(c) est éparse. Elle est réduite du fait de la difficulté de définir un contraste fort entre la ligne et le tarmac.

Dans les Figures 4.20, 4.21 et 4.22, nous avons sélectionné trois images, issues de caméras placées à l'avant de l'avion (pour 4.20 et 4.22) et dans la dérive de l'avion (pour 4.21) pour présenter une comparaison qualitative des algorithmes où les résultats du filtre particulaire sont représentés en rouge, ceux de la transformation de Hough sont représentés en bleu et ceux de LaneNet en vert sur la vérité terrain.

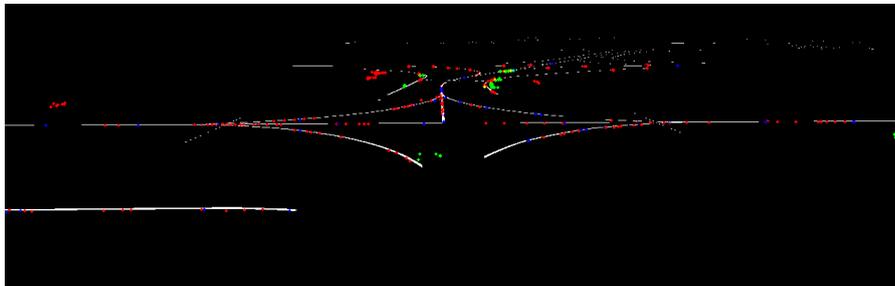
Dans cette étude, nous ne comparons pas la capacité des trois méthodes à fournir des équations de droites mais leur capacité à définir le motif recherché, c'est à dire les marquages au sol. Les méthodes de transformée de Hough et LaneNet nous



(a)



(b)



(c)

Figure 4.19: (a) Image en conditions météorologiques dégradées et zoom (zone jaune) sur les résultats (bleu pour la transformée de Hough, rouge pour le filtre particulaire et vert pour LaneNet) par (b) temps clair et (c) conditions météorologiques dégradées.

donnent des informations sur des segments de droites que nous avons discrétisés afin de pouvoir les comparer avec les résultats du filtre particulaire. c'est pourquoi les résultats des trois méthodes sont représentés par des points.

La Figure 4.22 montre l'impact de la transformation IPM sur les résultats obtenus. La détection peut paraître moins bonne dans les premières lignes de l'image ayant subi la transformation IPM en pré-traitement mais c'est un effet de la transformation. La précision est similaire, cependant une cellule de la carte du sol représente plusieurs pixels de l'image réelle dans la première tranche de la carte. C'est pourquoi, lors de la re-projection on voit moins de résultats de détections dans

les images auxquelles on a appliqué la transformation IPM, puisque ces derniers sont plus espacés.

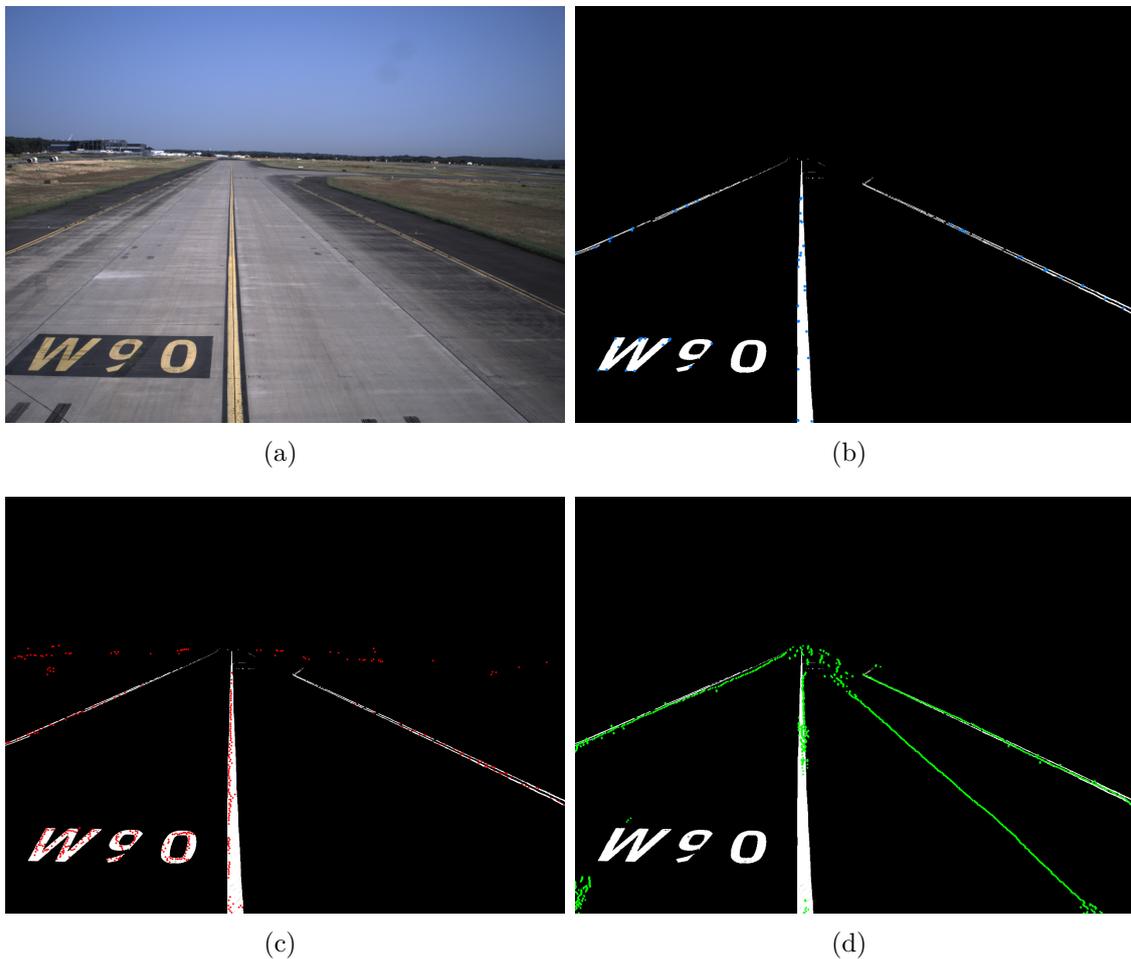


Figure 4.20: (a) Image réelle d'une caméra placée à l'avant de l'avion. Vérité terrain et résultats (b) pour la transformée de Hough, (c) pour le filtre particulaire et (d) pour LaneNet.

On remarque que les trois méthodes possèdent des performances encourageantes sur les images réelles. Cependant, le filtre particulaire est sujet à des fausses détections sur les pixels éloignés de la caméra, dû à sa carte de probabilité et à sa sélectivité. La transformée de Hough n'est pas aussi performante que les deux autres méthodes en termes de détection longue distance sur les images issues de caméras placées dans la dérive de l'avion. LaneNet a tendance à recréer des lignes, de part son précédent entraînement. Cela permet à l'algorithme d'être résistant aux occultations occasionnelles mais dégrade virtuellement la précision.

Pour les images simulées, la portée maximum de détection et la précision de la détection sont améliorées pour les images ayant bénéficié du pré-traitement à base de transformation IPM. La détection des lignes proches est sensiblement dégradée puisque, dû à la transformation IPM, un pixel dans la carte du sol représentant plusieurs lignes de pixels dans l'image réelle. Cependant, les pixels lointains dans l'image réelle sont amplifiés et rendus plus facile à détecter, malgré l'ajout d'un bruit certain.

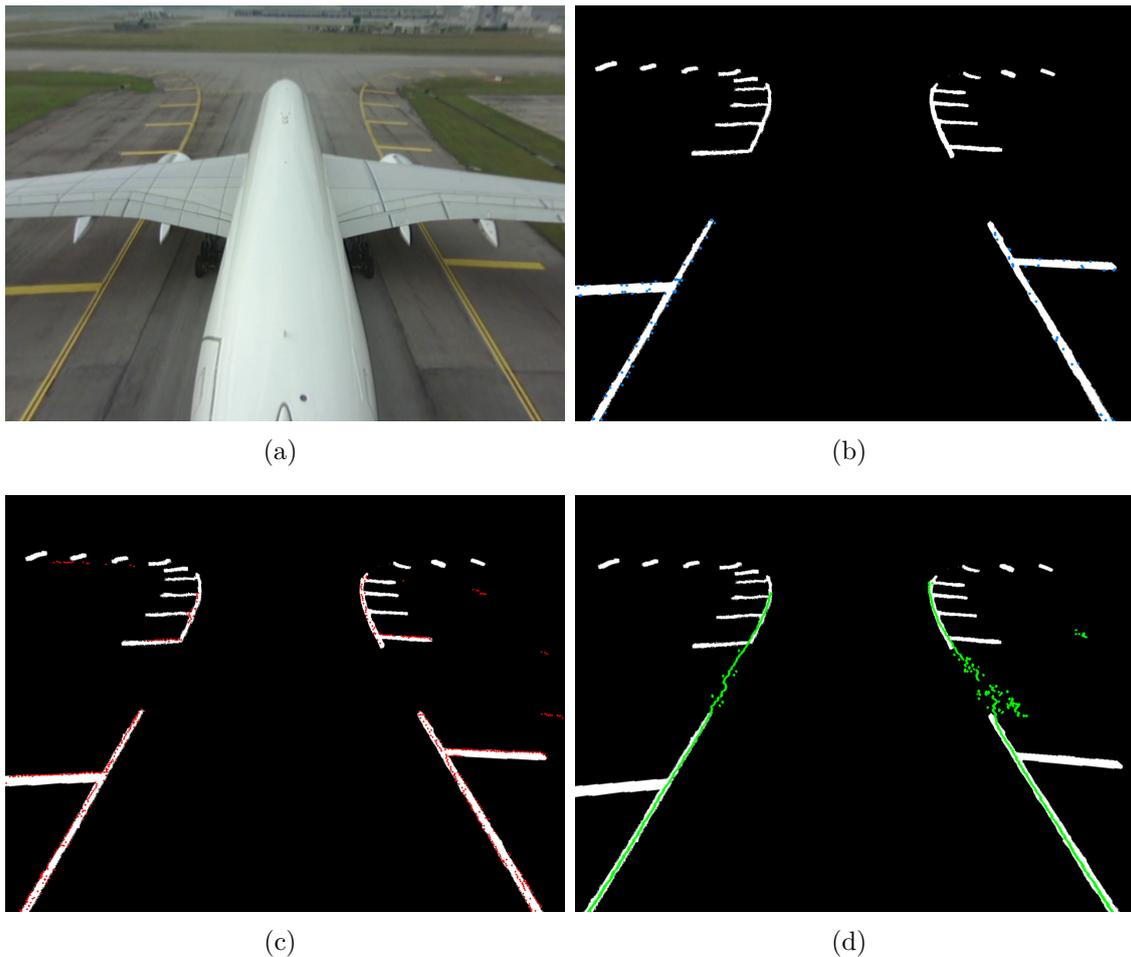


Figure 4.21: (a) Image réelle d'une caméra placée dans la dérive de l'avion. Vérité terrain et résultats (b) pour la transformée de Hough, (c) pour le filtre particulaire et (d) pour LaneNet.

### 4.5.3 Conclusion de la comparaison des trois méthodes

Nous avons comparé les performances du filtre particulaire, de la transformée de Hough et de LaneNet sur des images réelles et simulées, obtenues en plaçant des caméras à différentes positions sur l'avion. Nous avons aussi étudié l'impact de la transformation IPM en pré-traitement et l'impact de conditions météorologiques dégradées. Les résultats présentés ci-dessus ont été commentés en se basant sur les valeurs moyennes des résultats des algorithmes de détection sur un ensemble d'images.

Pour le filtre particulaire, on observe que les résultats sur les images simulées sont satisfaisants mais la transposition sur les images réelles donne des résultats mitigés. Un certain nombre de valeurs seuils doivent être fixées dans cette méthode ce qui implique des difficultés d'adaptation pour les images réelles ou pour des conditions météorologiques dégradées.

La transformée de Hough est basée sur une forte hypothèse de couleur, facile à fixer en simulation mais plus dure pour des images réelles même si elle se base sur une couleur de référence fournie par les normes aéronautiques. Il faudrait appliquer pour HT, la méthode d'adaptation de la couleur de référence présentée au Chapitre 3.

LaneNet produit des résultats qui peuvent paraître insatisfaisants mais cela était

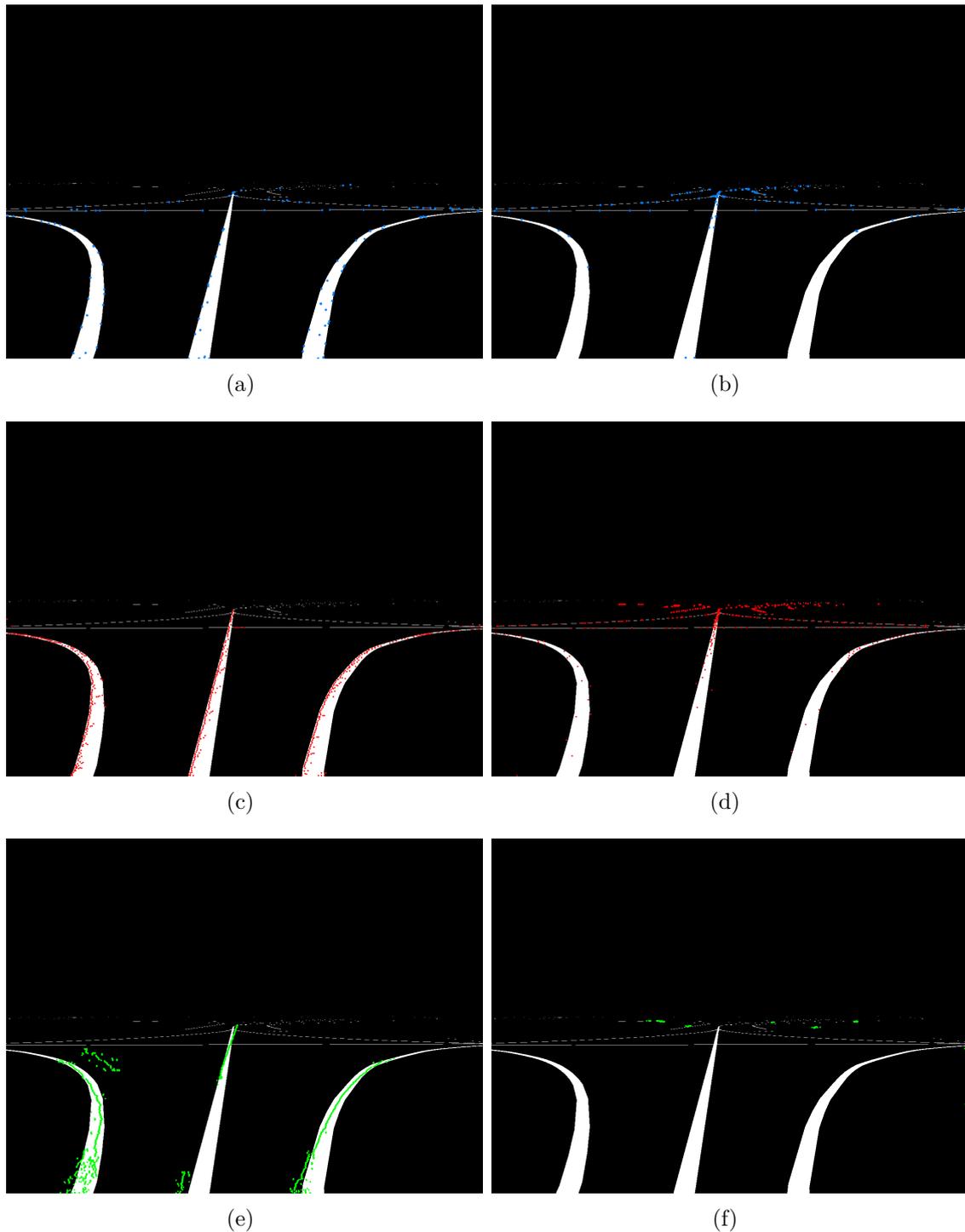


Figure 4.22: Résultats sans puis avec IPM d'une image simulée provenant d'une caméra placée à l'avant de l'avion (respectivement (a) et (b)) pour la transformée de Hough, (respectivement (c) et (d)) pour le filtre particulaire et (respectivement (e) et (f)) pour LaneNet.

prévu. LaneNet produit un masque pour le résultat de la détection de marquages. Nous avons dû passer par une discrétisation de ce masque afin de pouvoir le comparer avec les deux autres méthodes. Lors de la création du masque, une interpolation de la courbure du marquage est effectuée, ce qui explique pourquoi le masque continue sur les zones où le marquage est couvert par les ailes de l'avion. Cette estimation

de la présence du marquage pénalise la méthode par rapport aux critères que nous avons choisi pour la comparaison. De plus, le passage du domaine automobile à l'aéronautique déplace le réseau de neurone dans un certain nombre de situations pour lesquelles il n'a pas été entraîné. Ceci ajouté au fait qu'il essaie de reproduire des lignes parallèles impacte ses performances, notamment en termes de précision. Un ré-entraînement du réseau devrait permettre d'améliorer ces résultats.

La transformation IPM requiert d'avoir une grande précision sur la position de la caméra. Elle améliore les résultats de portée de détection sur les images simulées où la position de la caméra est parfaitement contrôlée. Cependant, même une faible modification de la position de la caméra peut dégrader les résultats sur des images réelles où la position de la caméra varie pendant la phase de taxi et entre le décollage et l'atterrissage (pour la caméra placée dans la dérive). Enfin, les méthodes donnent de bons résultats pour la détection proche en conditions météorologiques dégradées alors que l'impact sur les détections à portée moyenne ou lointaine est fort.

A la suite de cette thèse, nous prévoyons d'agrandir la base de données d'images, au moins simulées dans un premier temps puis par acquisitions réelles, pour pouvoir ré-entraîner le réseau de neurones et améliorer ses performances en l'adaptant aux motifs particuliers des zones aéroportuaires.

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté la première brique algorithmique nécessaire pour l'aide à la navigation de véhicules : l'analyse de scène. Nous avons présenté les méthodes les plus récurrentes dans le domaine automobile. Parmi elles, on peut faire une distinction entre les méthodes géométriques et les méthodes basées réseaux de neurones. De nombreuses approches ont été définies dans ces deux familles de méthodes. Les méthodes géométriques demandent souvent une connaissance du modèle du motif recherché et peuvent être sensibles au bruit de cette modélisation ou aux variations sur les hypothèses utilisées (variation d'illumination dans le cas où la couleur est une hypothèse forte du motif par exemple). Certaines approches géométriques peuvent être coûteuses en temps de calcul. Les méthodes basées réseaux de neurones s'imposent de nos jours et présentent des résultats satisfaisants mais demandent d'avoir une base de données d'entraînement qui n'existe pas dans le domaine aéronautique.

Nous avons présenté une application du filtre particulière à la détection de motifs (marquages au sol, balises, délimitation tarmac/herbe...) dans une image aéroportuaire. Cet algorithme s'appuie sur la définition préalable de cartes d'observations basées sur des informations de couleur et de contours. C'est pourquoi l'étude préalable du Chapitre 3 est nécessaire. Une nouvelle version de ce filtre particulière est à l'étude dans laquelle la particule représentera une fenêtre de pixels plutôt qu'un pixel isolé. De nouvelles problématiques se présentent, notamment sur l'initialisation et l'interruption d'un tel filtre. De plus, une perspective d'amélioration du filtre pourrait être de prendre en compte le positionnement des motifs détectés pour une image et de s'en servir pour l'initialisation de la détection sur l'image suivante.

Nous avons comparé le filtre particulière avec une méthode basée sur la transformée de Hough et une méthode basée réseaux de neurones. Lors de cette étude, nous avons discuté de la pertinence de la transformation IPM et de la robustesse des algorithmes face à des conditions climatiques dégradées. Comme nous l'avions prévu, il n'est pas possible d'utiliser directement un CNN pour la détection de marquages au sol dans la zone aéroportuaire sans l'avoir entraîné à l'avance, puisque les motifs sont assez différents. Néanmoins, on peut constater que, sans avoir été ré-entraîné, les résultats sont déjà prometteurs. L'étude de cette comparaison montre, pour les trois méthodes, que l'utilisation de l'IPM permet d'améliorer la portée de détection mais son usage est déconseillé lorsque les conditions météorologiques sont dégradées.

Dans le prochain chapitre, nous allons présenter la brique fonctionnelle "Détection d'obstacles". Nous montrerons l'utilisation des motifs extraits grâce à la méthode du filtre particulière pour l'estimation du déplacement de l'avion et la détection d'obstacles.

# Chapitre 5

## Détection d'obstacles dans les zones aéroportuaires

### Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>134</b>
<b>5.2</b>	<b>Techniques de base</b>	<b>136</b>
5.2.1	Estimation du mouvement d'un véhicule depuis des données acquises par des capteurs embarqués	136
5.2.2	Création d'une grille d'occupation	137
<b>5.3</b>	<b>État de l'art sur la détection d'obstacles</b>	<b>138</b>
5.3.1	Détection d'obstacles par méthodes géométriques	138
5.3.2	Détection d'obstacles par apprentissage automatique	142
<b>5.4</b>	<b>Estimation de mouvement</b>	<b>147</b>
5.4.1	Motifs d'intérêt	148
5.4.2	Matrices de coût	150
5.4.3	Estimation des matrices de rotation et de translation	153
5.4.4	Précision de l'estimation des matrices de rotation et translation	154
<b>5.5</b>	<b>Méthode géométrique - Carte d'obstacles</b>	<b>156</b>
<b>5.6</b>	<b>IPM multi-plans</b>	<b>164</b>
<b>5.7</b>	<b>Conclusion</b>	<b>170</b>

---

## 5.1 Introduction

Les premières méthodes de détection (et par la suite de suivi) d'objets ont été proposées à des fins militaires avant d'être massivement étudiées pour les multiples applications de la vision par ordinateur telles que dans les domaines de la vidéo-surveillance, de l'automobile ou de la robotique. C'est pourquoi de nombreuses techniques de détection d'objets ont été et sont encore proposées de nos jours, surtout grâce aux progrès des algorithmes basés réseaux de neurones. Dans le contexte de systèmes embarqués en robotique, automobile ou aéronautique, un objet détecté devient obstacle, s'il peut se trouver sur la trajectoire du porteur. Un obstacle peut être fixe (bâtiment, pylône...) ou mobile (piéton, voiture...).

En robotique, la détection d'obstacles temps réel dans les systèmes embarqués doit :

1. être réactive : c'est-à-dire qu'elle doit s'exécuter à une fréquence suffisante pour permettre une réaction du robot. Cette fréquence dépend des vitesses respectives du robot et de l'obstacle, du temps de réponse du système, de la distance de perception... Par exemple dans le cas d'un véhicule conduit par un humain, sur autoroute, la fréquence doit être de 100Hz pour limiter les conséquences d'un accident entre deux véhicules roulant à 120km/h, avec une portée maximale des capteurs de 50m, et un temps de réaction du conducteur de 1s;
2. ne pas faire d'erreurs, ni générer de fausses alarmes (trop de faux positifs vont stopper ou ralentir la progression d'un robot en mode autonome, ou bien vont diminuer la confiance du conducteur d'un véhicule), ni rater un obstacle (de faux négatifs peuvent avoir des conséquences graves);
3. localiser l'obstacle et estimer sa dynamique vis-à-vis du robot ou du véhicule qui porte les capteurs. Dans le cas d'un robot terrestre, c'est-à-dire se déplaçant sur le sol, position et vitesse peuvent être exprimées dans la surface du sol, surface qui peut être plane (position et vitesse en 2D) ou non (position et vitesse en 3D);
4. être robuste aux changements qui peuvent survenir dans l'environnement : conditions climatiques, conditions d'illumination... Pour ce faire, on peut combiner plusieurs capteurs, en particulier pour la vision, ou exploiter des caméras sensibles dans plusieurs bandes spectrales par exemple;
5. respecter certaines contraintes matérielles. Selon l'application, les contraintes de coût, d'encombrement du sous-système de détection d'obstacles (capteurs, unités de traitement), d'intégration dans le système global, peuvent être plus ou moins importantes.

Pour satisfaire les contraintes 2 à 4, il convient de traiter les données sensorielles avec des algorithmes complexes, d'autant plus si le cahier de charges est sévère (par exemple, détecter un objet au sol comme une valise, à une distance de plus de 200m). De ce fait, pour satisfaire la contrainte 1, il convient de porter ces algorithmes sur des architectures dédiées permettant de tirer parti du parallélisme, typiquement à

base de FPGA, DSP, GPU ou de multiples CPU. Comme il a été dit au Chapitre 2, cette problématique a été traitée au LAAS dans le cadre du projet SART, mais elle n'est plus notre priorité dans les projets TALVIS et COCOTIER et n'est pas prise en compte dans le choix de l'algorithme utilisé pour la détection d'obstacles.

Dans le cadre de cette thèse, nous nous intéressons à l'aide à la navigation d'un aéronef sur le tarmac et aux abords des portes d'embarquement. Ces situations peuvent amener à la présence d'obstacles autour de l'aéronef, comme par exemple d'autres aéronefs, des véhicules ou des piétons. C'est dans le but de détecter ces présences potentielles et d'en informer le pilote que nous nous sommes intéressés à la brique algorithmique de détection d'obstacles. Notons que dans le contexte aéroportuaire, un objet peut devenir un potentiel obstacle du fait des recommandations aéroportuaires. Par exemple, il n'est pas autorisé que les ailes d'un aéronef en mouvement passent au-dessus d'un véhicule, même s'il ne peut pas y avoir de collision. De même, il existe des distances à respecter entre le bout des ailes et tout objet fixe (pylône, bâtiment). Ces recommandations doivent être prises en compte lors de la remontée d'informations au pilote et plus tard pour l'automatisation de la détection.

Dans la première partie de ce chapitre, nous exposons quelques techniques de base nécessaires à la représentation et la détection d'objets avant de présenter différentes méthodes de détection d'objets et d'obstacles proposées dans la littérature. Nous reviendrons sur certaines méthodes géométriques ainsi que des méthodes basées réseaux de neurones.

L'estimation du déplacement d'un système est fréquemment basée sur la détection et le suivi de motifs d'intérêt. Parmi elles, l'estimation du flot optique et de l'odométrie optique sont souvent présentes dans la littérature scientifique. Dans un second temps, nous présenterons l'utilisation que nous faisons des motifs extraits grâce à la méthode du filtre particulaire dans le Chapitre 4 afin de déduire le déplacement 2D de l'aéronef sur le plan du sol (en rotation et en translation) entre deux images.

Enfin, nous utiliserons l'information de déplacement pour notre méthode de détection d'obstacles. Nous combinons l'estimation de déplacement à l'utilisation de la transformation IPM dans le but de créer une carte d'obstacles entre deux instants  $t - 1$  et  $t$ . Nous proposons aussi d'utiliser l'information de cette carte d'obstacles pour indiquer au pilote de l'aéronef, sur une image en réalité augmentée, la position d'un obstacle potentiel et son degré de criticité.

## 5.2 Techniques de base pour la représentation et la détection d'objets

Outre les traitements classiques de rectifications et de filtrages appliqués à l'image, il peut être intéressant de passer par plusieurs phases d'analyse de la scène étudiée et de ses différentes composantes avant de se lancer dans la détection d'objet. Dans le Chapitre 4, nous avons proposé une méthode d'extraction de motifs d'intérêt. Cette phase permet de faire ressortir des informations qui, traitées, profitent aux phases de détection, suivi et classification des obstacles.

Plusieurs méthodes existent pour tirer profit de la mise en relation de points d'intérêt [165][138][130] afin de caractériser les objets présents dans la scène. Cependant, nous n'avons pas utilisé ces fonctionnalités dans notre brique algorithmique de détection d'obstacles.

Ci-dessous, nous présentons deux techniques de base pour la représentation et la détection d'objets dans une scène : l'estimation du mouvement du porteur du système d'acquisition et la construction d'une grille d'occupation.

### 5.2.1 Estimation du mouvement d'un véhicule depuis des données acquises par des capteurs embarqués

Pour plusieurs fonctionnalités et/ou algorithmes, il est essentiel, dans le domaine de la détection d'obstacles, d'avoir une idée de la vitesse des différents acteurs. Ainsi, certaines méthodes, telles que l'étude du flot optique ou la construction et la mise à jour de cartes d'occupation que nous présenterons ci-dessous, vont nécessiter d'avoir des informations sur le mouvement du porteur du dispositif d'acquisition. On peut aussi avoir besoin de la vitesse du porteur pour prédire le mouvement et la vitesse d'un objet perçu comme un obstacle ou un danger.

Il est possible, dans certaines applications robotiques ou automobiles, de récupérer ces informations en se basant sur les capteurs proprioceptifs (odométrie, IMU, tachymétrie...) prévus à cet effet au sein du système porteur du dispositif d'acquisition. Se baser sur ces informations induit tout de même une dépendance à ces capteurs, ainsi que des problèmes d'architecture logicielle, pour que ces données sur l'état du système soient accessibles aux modules de perception. Ainsi, si l'information devient faussée ou s'accompagne d'un bruit, les algorithmes de détection en seront affectés.

On peut alors préférer que le module de perception soit autonome pour estimer le mouvement du porteur. Concernant l'IMU par exemple, dans le projet SART qui avait seulement pour but de développer un boîtier de capteurs à intégrer dans la dérive, une centrale inertielle Xsense avait été rajoutée dans le boîtier. Comme on dispose de caméras, on peut exploiter les techniques bien connues d'odométrie optique, en exploitant la fusion avec les données inertielles pour estimer le facteur d'échelle (ce qui est un problème majeur dans le cas de la vision monoculaire). Ce type d'odométrie par vision monoculaire couplée à une IMU, est très utilisée sur des drones [50].

En robotique ou sur des véhicules intelligents, on peut fusionner l'odométrie optique de manière serrée ou lâche, avec d'autres capteurs, typiquement des LIDAR 2D ou 3D [46], généralement en conservant les primitives sensorielles exploitées

dans l'estimation pendant quelques itérations. On ne parle pas ici de SLAM et de reconstruction incrémentale de l'environnement, hors sujet dans cette thèse.

Dans notre cas, l'estimation du mouvement du porteur des systèmes d'acquisition se fait par appariement des motifs d'intérêt extraits par le filtre particulaire aux instants  $t - 1$  et  $t$  (voir Section 5.4) et permet de générer une carte d'obstacles (description en Section 5.5).

### 5.2.2 Création d'une grille d'occupation

Cette technique est généralement basée sur une exploitation de données de profondeur. Elle a été initialement introduite en Robotique [43] pour fusionner de manière probabiliste des données ultrasons lorsqu'un robot se déplace en milieu intérieur. C'est donc une représentation, plus qu'une méthode, très utilisée en Robotique ou sur les Véhicules : c'est un support de fusion spatio-temporelle de données issues de plusieurs capteurs ou d'un même capteur qui se déplace.

Cette représentation n'est possible que si le robot ou véhicule se déplace sur un plan, ou sur une surface qu'on peut considérer comme localement plane. De même que pour la construction de la carte du sol décrite dans le Chapitre 3, le principe est de discrétiser le plan du sol en petites cellules sous forme d'une grille, que l'on appelle grille d'occupation. On attribue ensuite à chaque cellule une probabilité qu'elle soit occupée.

Dans la section suivante, nous proposons un état de l'art des méthodes de détection d'obstacles par approches géométriques se servant des techniques présentées ci-dessus ou par réseaux de neurones.

## 5.3 État de l'art sur la détection d'obstacles

Dans cette partie nous proposons un état de l'art de différentes approches de détection d'obstacles qu'il conviendra de repérer dans une scène et dont il faudra ensuite suivre l'évolution. Il faudra donc être capable de détecter ces objets, c'est à dire dans un premier temps, d'arriver à faire le lien entre (1) un ensemble constitué d'apparence (couleur, texture...) et de formes géométriques (contours, coins...) dans le domaine de l'image et (2) des connaissances a priori sur le monde réel : structuration de l'environnement (par exemple, sur une route, tout ce qui est perçu sur la route est un potentiel obstacle), ou apparence et modèle 3D appris au préalable, des objets susceptibles d'être des obstacles. Pour ce faire, le raisonnement doit se baser sur des modèles mathématiques complexes.

Pour une utilisation en temps réel de cette détection d'objets, ces méthodes doivent être particulièrement attentives aux temps des calculs engendrés par les algorithmes mis en œuvre. On peut différencier ces méthodes selon le type de données qu'elles prennent en entrée (images, nuages de points 3D, coupes laser...) et selon les types d'approches. Nous étudierons dans cet état de l'art uniquement des approches par vision monoculaire, puisque c'est le seul capteur qu'il est envisagé d'intégrer sur les aéronefs à court terme, les technologies LIDAR et RADAR pour l'aéronautique n'étant pas prêtes. Parmi ces approches on retrouve :

- Les approches géométriques classiques
- Les approches d'apprentissage automatique dites Machine Learning à base de descripteurs fournis par l'analyse des données, et d'un classifieur (SVM, AdaBoost, Random Forest)
- Les approches d'apprentissage profond dites Deep Learning, qui apprennent aussi les descripteurs les plus adaptés pour détecter des objets présents dans la scène.

Ce qui fait la grande différence entre les méthodes géométriques et les autres, est que dans le premier cas, nous ne disposons pas de connaissance a priori sur la nature de l'obstacle : l'obstacle est détecté de par sa position géométrique (utilisation de la transformation IPM par exemple s'il est sur la route) ou de par son mouvement (utilisation de flot optique par exemple). Dans le deuxième cas, les obstacles sont les objets qui ont été appris. Si le système n'apprend pas qu'un piéton est un obstacle, il ne le détectera pas.

### 5.3.1 Détection d'obstacles par méthodes géométriques

Cette partie est dédiée aux différentes techniques utilisées dans le domaine de la vision monoculaire. Nous avons vu, dans le Chapitre 2, qu'il n'était pas réaliste de considérer la vision stéréoscopique (cockpit très encombré, difficulté de maintenir le calibrage...). Même si nous avons au LAAS, une expérience sur les méthodes exploitant les u et v-disparité [94] qui seraient pertinentes dans notre contexte vu que le sol est plat, nous ne les prenons pas en compte dans cet état de l'art. De même, nous ne considérons pas les méthodes de détection d'objets mobiles, qui exploitent la soustraction d'une image de fond, méthodes non applicables quand la caméra est embarquée sur un porteur mobile.

Plusieurs techniques peuvent être retrouvées dans la littérature et sont présentées ci-dessous. Ces méthodes se basent sur des postulats différents.

### 5.3.1.a Flot optique

La mesure du flot optique est une technique populaire utilisée pour la détection et le suivi d'objets, l'extraction des surfaces planes, la détection de mouvement, l'aide à la navigation et l'odométrie optique (estimation de la vitesse de déplacement du système) [12]. Dans [10], la détection d'objets dynamiques se fait avec cette technique, par l'utilisation d'odométrie visuelle.

Il existe plusieurs méthodes de calcul de flot optique. Une des méthodes principales de calcul du flot optique en image 2D est la méthode différentielle de Lucas-Kanade [107]. Cette méthode se base sur l'hypothèse que le déplacement d'un pixel d'une image à une autre est faible et constant dans son voisinage.

L'étude du flot optique peut s'effectuer par utilisation des points d'intérêt comme décrits dans [72][70]. Dans ce cas, on parlera de flot optique épars. Il est moins coûteux en terme de calcul ou de manière analytique. Cependant il est basé sur l'hypothèse que les points d'intérêt sont stables dans le temps ce qui n'est pas toujours exact en infrarouge ou en 3D lorsque la variation de résolution du nuage de points est rapide et importante. En effet un objet lointain sera composé d'un plus faible nombre de points. S'il n'est pas assez grand, il pourrait ne pas être identifié.

Il est aussi possible de faire une étude de flot optique sur l'intégralité de l'image, ce que l'on appelle flot optique dense, mais cela est plus coûteux en temps de calcul [37]. Il est tout de même important de noter que le flot optique permet de caractériser les objets mobiles. Dans un contexte où l'on peut porter un intérêt fort à des objets statiques dans une scène, la technique de flot optique pourrait ne pas être adapté.

### 5.3.1.b Méthode plan+parallaxe

Le but de ce type de méthode est de segmenter une scène pour récupérer une région statique et des régions en mouvement qui représenteront les objets présents dans cette scène. Cette méthode se base sur l'utilisation d'homographies et des contraintes de géométrie épipolaire. Dans cet article [184], les auteurs proposent d'ajouter une nouvelle contrainte de "consistance de la structure", qui décompose l'image en un plan statique de référence et des structures n'appartenant pas à ce plan. Elle est géométriquement basée sur la contrainte trilinéaire puisqu'elle propose une étude de ces zones sous trois angles de vues.

Si l'image est fournie par une caméra en mouvement, une première étape consiste à étudier le mouvement propre au système. Une fois connu, les pixels cohérents avec cette vitesse seront labellisés comme étant au second plan et faisant partie de l'environnement.

Les pixels incohérents sont labellisés comme des régions dont le mouvement est indépendant de celui de la caméra donc certainement des objets à détecter. Les homographies sont donc utilisées pour corriger, au niveau des pixels, le mouvement inhérent à la caméra, d'une image sur l'autre. Les pixels résiduels de l'homographie (c'est à dire les pixels cohérents) correspondent à la partie statique de la scène. Les autres pixels peuvent correspondre à des objets mobiles ou à des larges structures 3D possédant une grande variance des données de profondeur.

La contrainte épipolaire est souvent utilisée pour détecter un mouvement entre deux vues différentes d'une image. La méthode se base sur des points d'intérêt mis en correspondance d'une image à la suivante. Si un point d'intérêt dans la première image ne correspond pas à la contrainte épipolaire posée par son appariement avec sa représentation dans la deuxième vue, le point 3D, qui s'est projeté sur ces pixels dans ces deux vues, est considéré comme étant en mouvement.

Cependant, si le mouvement du pixel s'effectue dans le plan défini par la contrainte épipolaire, la détection du mouvement échouera. Ce cas peut se produire lorsque la caméra et l'objet se déplacent dans la même direction.

### 5.3.1.c Inverse Perspective Mapping

Dans le cas de la détection d'objet sur une surface considérée essentiellement plane (par exemple une route [22]), la méthode de transformation de l'image appelée Inverse Perspective Mapping (IPM) que nous avons présentée au Chapitre 3, peut être utilisée. Rappelons que, si la position de la caméra par rapport au sol est connue, l'algorithme IPM permet d'obtenir une projection de l'image sur le plan du sol, grâce à une homographie. L'IPM construit une carte discrétisée du sol (bird's eye view en anglais), dans laquelle les effets perspective sont retirés, mais cette carte est déformée par les objets de l'environnement qui ne sont pas sur le sol. Une très bonne explication de l'utilisation de l'IPM pour les caméras de recul dans le domaine de l'automobile est présentée dans [99].

L'utilisation de l'IPM pour la création d'une carte d'obstacles exploite la projection des images acquises aux instants  $t - 1$  et  $t$  sur le plan du sol (supposé plan) ainsi que la connaissance du mouvement 2D. On peut alors créer une carte d'obstacles dans laquelle toutes les cellules correspondant à la projection de points du sol sont mises à 0. Les autres cellules seront donc des projections d'obstacles posés (ou non) au sol. Nous détaillerons ce principe dans la Section 5.5. Un inconvénient de cette utilisation réside dans le fait que les sols ne sont généralement pas parfaitement plats, ce qui peut induire du bruit dans la carte d'obstacles.

Dans [15][176], cette méthode est exploitée en stéréo-vision. Dans ce cas, les deux images droite et gauche sont projetées au sol, et la différence permet de mettre en évidence les points qui ne sont pas au sol, donc qui sont de potentiels obstacles. L'intérêt de l'IPM en stéréo-vision est qu'on peut se passer d'estimation du mouvement.

Signalons aussi que l'approche IPM est parallélisable. D. Botero [17] a, dans sa thèse, proposé plusieurs architectures pour porter cet algorithme sur FPGA, que cela soit en monoculaire (en exploitant une estimée du mouvement fournie par ailleurs) ou en stéréo-vision.

### 5.3.1.d Utilisation de la grille d'occupation pour la détection d'obstacles

Une grille d'occupation probabiliste permet de modéliser l'évolution de l'environnement autour du porteur lors de son déplacement. En contrepartie, l'utilisation d'une grille demande une forte disponibilité d'espace de stockage et produit une discrétisation de l'espace (nous avons choisi des cellules 25x25cm dans nos cartes du sol), qui peuvent induire des biais non souhaités ou des limitations quant à la résolution ou la précision de détection.

Ces problématiques de place mémoire et de résolution ont été considérées de diverses manières. En Robotique d'intérieur, les espaces sont de taille finie. Souvent la grille couvre tout l'espace dans lequel le robot évolue. Il existe des approches SLAM qui exploitent la grille pour la localisation du robot et pour déterminer si, entre deux acquisitions, une partie de la carte s'est déplacée. Si l'on peut retrouver des cellules en mouvement avec cette technique, un processus de regroupement de cellules "mobiles" adjacentes est nécessaire pour identifier les objets mobiles individuellement [169][170].

Pour un véhicule ou un robot d'extérieur, c'est plus complexe. On ne peut pas représenter tout l'environnement par une grille : généralement le robot gère une grille glissante d'une taille raisonnable avec les capacités mémoire du système embarqué. Deux stratégies sont exploitables qui, chacune, utilisent l'estimation du mouvement 2D du robot :

- Soit la grille glisse à chaque mise à jour : le robot reste en position fixe dans la grille, en bas et au milieu, comme notre aéronef dans les cartes du sol produites dans le Chapitre 3. Le robot "oublie" de ce fait l'espace qui est derrière lui : la mise à jour nécessite, à chaque itération, une rotation et une translation de la grille en  $t - 1$  pour construire la grille en  $t$ . Nous ferons de même ci-dessous dans notre stratégie de détection d'obstacles.
- soit la grille glisse avant que le robot en sorte, donc la fréquence de "glissement" dépend de la taille choisie et de la vitesse du robot. Entre deux glissements, le robot se déplace dedans en exploitant toujours l'estimation de son mouvement 2D. À chaque itération, la grille est mise à jour à partir des données issues des capteurs. Quand le robot se rapproche d'un des bords de la grille, alors elle est "glissée" pour le remettre par exemple en bas et au milieu. On peut dans ce cas éviter les rotations et garder une orientation constante de la grille glissante.

Les auteurs de [158] proposent d'utiliser une méthode appelée Bayesian Occupancy Filter, qui permet d'estimer, à chaque instant, l'état de la grille d'occupation en combinant une étape de prédiction (en fonction des instants précédents et d'un modèle dynamique) et une étape d'estimation (qui incorpore à la prédiction le résultat des nouvelles observations) comme pour un filtrage bayésien. Cette méthode permet notamment la fusion d'informations provenant de plusieurs capteurs. Elle explicite la notion d'incertitude de présence, évite la problématique d'association des objets entre deux images consécutives (ce qui permet de contourner les difficultés rencontrées face aux apparitions, disparitions ou occultations d'obstacles d'une image à l'autre) et ne nécessite pas d'avoir une modélisation de l'obstacle. Cette méthode est hautement parallélisable ce qui rend possible son utilisation par un système embarqué.

Dans sa thèse [122], J. Moras propose l'utilisation de plusieurs types de grilles de perception (grille d'accumulation, grille d'occupation probabiliste et grille d'occupation évidentielle) pour la navigation robotique en milieu urbain. Il conclut que les grilles d'occupation évidentielle offrent un résultat analogue aux grilles d'occupation probabilistes tout en utilisant moins d'hypothèses a priori sur la confiance dans la mesure. Cependant, la complexité calculatoire de cette méthode est importante.

Dans notre cas, la carte du sol obtenue par la transformation IPM, n'est qu'un support pour générer la vue fusionnée RGB/IR ou pour fusionner toutes les images acquises par les caméras montées sur l'aéronef. Une perspective, pour le futur, serait de la gérer comme une grille d'occupation glissante, en associant à chaque cellule, en plus de l'apparence RGBT, une information probabiliste indiquant si elle est libre ou occupée.

### 5.3.1.e Approche Vision Infrarouge

Plusieurs constructeurs (Daimler Benz) ou équipementiers (VALEO) ont embarqué une caméra infrarouge sur leurs Démo Cars (démonstrateurs des dernières innovations), mais ils proposent peu de traitements dédiés à ce type d'images. Signalons que les propriétés radiométriques dans plusieurs bandes spectrales ont été exploitées pour identifier la nature du terrain (terre, goudron, herbe...) sur des robots évoluant en terrain naturel.

### 5.3.1.f Approche par segmentation de l'espace libre

Cette approche est courante en Robotique. Elle est généralement couplée à une représentation discrétisée de l'environnement par Grille d'Occupation en terrain plat, ou par Carte d'Élévation en terrain accidenté. Quand on dispose de capteurs ultrasons ou de LIDAR 2D, l'approche est très simple puisqu'on sait que l'espace est libre entre le capteur et la distance qu'il renvoie. En vision, c'est plus complexe.

Il est nécessaire de segmenter le sol grâce à des connaissances a priori sur son apparence, souvent par des techniques de propagation à partir de l'espace immédiat autour du robot. À l'initialisation, on fait l'hypothèse que le véhicule (robot, voiture) est dans l'espace navigable (sol plat, route) : on exploite les images pour analyser si les abords du robot sont classifiés comme appartenant au "sol" ou non, puis on propage dans les cellules plus éloignées, jusqu'à arriver à la limite de l'espace navigable (par exemple : murs pour un robot d'intérieur ou bords de la route pour un véhicule), aux obstacles (objet posé au sol) ou à une distance maximale de perception.

En Robotique, une telle approche a été appliquée à partir de la vision omnidirectionnelle [116] et, dans notre équipe, à partir d'une ceinture de micro-caméras disposées autour du robot [77]. Dans ce cas, pour traiter le flot vidéo de  $N$  caméras en temps réel, il a été nécessaire de développer une architecture hybride, dans laquelle les traitements parallélisables (fusion des images, corrections des distorsions...) étaient exécutés sur FPGA.

## 5.3.2 Détection d'obstacles par apprentissage automatique

Plus récemment, les recherches en matière de vision assistée par ordinateur se sont tournées vers les technologies de type apprentissage automatique [28]. La reconnaissance d'objets peut aussi se faire par l'utilisation d'algorithmes créés par le biais des outils de l'apprentissage automatique que sont par exemple les classifieurs classiques ou les réseaux de neurones. Dans cet état de l'art nous nous intéresserons à ces méthodes de classification par apprentissage de plus en plus étudiées de nos jours et qui apportent de très bons résultats.

Pour ce qui est des réseaux de neurones artificiels, ils sont inspirés du fonctionnement du cerveau humain à base de neurones interconnectés. Chaque couche exécute des opérations (logiques, statistiques, linéaires...) sur l'ensemble des données de la couche précédente. Les premiers travaux sur le sujet sont de McCulloch et Pitts [114].

La détection d'objet par apprentissage automatique va être sélective. Ne seront reconnus que les objets de certaines classes qui ont été apprises au préalable via une base de données (voitures, bus, piétons, arbre dans le domaine de l'automobile par exemple). La classification peut être :

1. supervisée, c'est à dire que l'on dispose d'une base de données d'images annotées (appelées vérité terrain) qui donne une classification exacte et sur laquelle se base l'algorithme de classification pour apprendre,
2. non supervisée où la classification se fait sans vérité terrain,
3. semi-supervisée où la vérité terrain est partielle.

Suivant les méthodes, la classification peut se faire par zones d'intérêt ou par segmentation sémantique. Dans le premier cas, l'image est parcourue par des fenêtres glissantes de taille variable qui sont ensuite fournies à l'algorithme. Dans le deuxième cas, chaque pixel est affecté à une classe (dans notre cas, herbe, tarmac, obstacle...). Nous avons d'ailleurs abordé ce point dans le Chapitre 3 en proposant des algorithmes de détection de marquages au sol par segmentation sémantique dans un premier temps puis en proposant, en perspective, une approche fenêtrée.

### 5.3.2.a Détection d'obstacles par classifieurs

Dans le cas de l'apprentissage automatique l'étape de classification intervient souvent après une première étude de détection de zones d'intérêt par le biais des méthodes géométriques de détection d'objets comme celles que nous avons vu précédemment. Pour réaliser la classification, les réseaux de neurones pondèrent leurs entrées, en se basant par exemple sur les données de flot optique, les données géométriques, les données de profondeur, couleur et texture ou sur les descripteurs.

Le choix des descripteurs va entre autre dépendre du type d'objet et de sa distance au système d'acquisition. Par exemple en automobile, si un véhicule est proche et de face, le logo pourra permettre de détecter la marque, s'il est loin, sa forme permettra de l'identifier comme une voiture.

Le descripteur HOG (Histogram of Oriented Gradients) donne de très bons résultats dans la détection de véhicule et de piétons. Il est utilisé dans plusieurs travaux [27], [155] et associé à différents algorithmes de classification comme les SVMs ou AdaBoost.

Le descripteur HOG est semblable en certains points au descripteur SIFT [106]. Le principe du descripteur est de décrire l'image par un ensemble de vecteurs, chacun calculé sur une partie d'image, représentant des distributions d'orientations de gradients de couleur (ou des directions de contours). Ce descripteur met en lumière les contours et l'intensité du gradient. Ces vecteurs sont normalisés par zone pour être plus robustes face aux variations d'illuminations. Plusieurs déclinaisons de ce descripteur existent comme Sparse-HOG [26].

Une fois qu'une région de l'image est décrite par un descripteur, plusieurs méthodes sont exploitables pour le classifier. Leur principe a été détaillé dans le Chapitre 3.

### **Les machines à vecteurs de support (Support Vector Machine - SVM)**

Dans leurs travaux [131], les auteurs proposent d'utiliser des ondelettes de Haar pour entraîner leurs SVMs à détecter des piétons. Les ondelettes de Haar calculent une différence d'intensité de pixels dans des zones rectangulaires représentant des morceaux d'images. Cela forme des masques qui sont efficaces pour décrire des contours. Les auteurs de [141] cherchent à séparer le plus efficacement les pixels relatifs à un véhicule des pixels n'appartenant pas à ces véhicules, en se basant sur les couleurs.

Les SVMs ont un coût élevé en terme de temps de calcul. Dans la thèse [65], une méthode combinant les descripteurs HOG et BoW (Bag of Word) est proposée pour réduire le coût algorithmique de l'algorithme de classification.

### **L'algorithme de classification AdaBoost**

Les auteurs de [190] et [28] proposent une utilisation conjointe de HOG et de l'algorithme de classification AdaBoost [51] afin d'effectuer une détection de piétons et voitures avec un temps de calcul réduit par rapport aux utilisations de SVMs.

### **Les forêts d'arbres décisionnels ou forêts aléatoires (Random Forest)**

Dans ses travaux [18], l'auteur propose une technique d'apprentissage automatique basée sur des arbres de décision. Cet algorithme est repris dans l'article [11] sous forme de cascade, c'est à dire, une structure à étage dont la complexité augmente avec le temps. Cette technique permet entre autre de réduire les fausses détections et d'augmenter la vitesse de calcul. Les auteurs étudient ici trois formes de votes entre les arbres décisionnels :

1. en pondérant les étages par rapport à leurs performances,
2. en permettant aux étages de rejeter une image avant la fin de l'algorithme, si elle ne contient pas l'objet recherché,
3. en tenant compte du vote émis par les étages précédents.

Leur approche est utilisée pour des applications de détection de piétons, de voitures ou de visages.

#### **5.3.2.b Détection d'obstacles par apprentissage profond**

L'apprentissage profond est similaire à l'apprentissage automatique mais s'exprime à un plus haut niveau d'abstraction. Les techniques d'apprentissage profond sont très à l'étude depuis la dernière dizaine d'années. Ils ont apporté de très grosses améliorations aussi bien dans les domaines de la vision par ordinateur que dans la robotique, la bio-informatique, la sécurité, la santé et l'intelligence artificielle.

La principale différence de l'apprentissage profond est qu'il se base sur un réseau de neurones multicouches. Chaque couche applique une transformation (linéaire ou non) à sa donnée d'entrée, avec une abstraction croissante. Cette multiplicité des couches permet de se passer de la recherche de descripteurs, par exemple, puisque certaines couches du réseau de neurones seront dédiées à l'apprentissage de ces derniers en amplifiant les détails importants au sein de l'image et en supprimant

les variations hors sujet.

L'apprentissage profond se base sur une architecture de réseau de neurones particulière appelée réseau de neurones convolutifs (CNN). Ce type de réseaux est très adapté pour classifier des données représentées sous formes de tableaux multiples, comme les tableaux de données de pixels pour chaque canal R, G, B dans une image couleur par exemple. C'est pour cela que les réseaux convolutifs sont largement utilisés dans le domaine de la vision par ordinateur, en particulier depuis qu'ils ont offert des résultats spectaculaires lors du concours ImageNet en 2012.

Les réseaux convolutifs sont composés de différents types de couches de neurones. D'une part les neurones convolutifs, qui traitent une partie de l'image en appliquant des fonctions de convolution. C'est dans ces couches que l'on va chercher à reconnaître des motifs, des formes géométriques par exemple. D'autre part les neurones dits de pooling, qui récupèrent les informations du premier type de neurones et forment une couche de sortie. Le modèle des réseaux de neurones convolutifs superpose ces types de couches de neurones avec plusieurs paquets de couches de neurones convolutifs séparés par une ou plusieurs couches de neurones de pooling.

Il existe de nombreuses architectures de réseaux de neurones convolutifs. Dans l'article [76], les auteurs proposent une analyse de plusieurs de ces architectures afin de définir les architectures à favoriser suivant le type d'application et les contraintes de temps de calcul. Parmi les différentes architectures, on peut citer entre autre R-CNN (Regions with CNN features) [57], Fast R-CNN [56], SSD (Single Shot Multibox Detector) [103] ou SqueezeDet [178].

La méthode MS-CNN (Multi-Scale CNN) [21] propose de détecter des objets de tailles multiples et différentes en utilisant le même réseau et sans avoir à modifier la taille de l'image d'entrée (et donc de l'objet) pour apprendre au réseau les différentes tailles possibles. L'algorithme est composé de deux sous-réseaux : le premier sélectionne des zones candidates dans l'image et le deuxième s'occupe de la classification. C'est l'architecture du premier réseau, en branches, qui fournit des zones candidates de différentes tailles pour une architecture de réseau fixe.

En 2016 est apparu la méthode YOLO (You Only Look Once) [137], mise à jour en 2020 avec YOLO V4. L'intérêt de cette méthode est de prédire les boîtes délimitant l'objet géographiquement dans une scène (bounding boxes) et les probabilités d'appartenance à une classe, en une seule évaluation du réseau de neurones, ce qui rend cette méthode beaucoup plus rapide que d'autres architectures d'apprentissage profond.

Les techniques d'apprentissage automatique et profond donnent de très bons résultats en terme de détection d'objets. Cependant ces techniques sont très gourmandes en terme de temps de calcul. Il est possible de le réduire en réduisant le nombre de couches, dans le cas de l'apprentissage profond. Le temps de calcul peut aussi être réduit en présélectionnant des zones d'intérêt sur une image, où l'on sait que la probabilité de détecter un objet/obstacle est plus grande, par exemple en réduisant l'image en effectuant au préalable une détection de la zone correspondant au tarmac et aux alentours proches.

Les performances d'une méthode d'apprentissage profond dépendent largement

de la base d'apprentissage. Par ailleurs, quand un réseau est entraîné sur une base d'apprentissage spécifiquement pour lui faire détecter un objet, on ne sait pas en fait ce qu'il apprend vraiment (l'objet, ce qui l'entoure...). Un défi majeur est d'expliquer les résultats fournis par un réseau, on parle de Explainable AI. Notons aussi dans notre contexte de l'aéronautique, le défi majeur de la certification, auquel AIRBUS s'attaque à travers le projet DEEL déjà évoqué au Chapitre 2.

Dans cette section, nous avons présenté plusieurs méthodes de détection d'objets qui peuvent par la suite devenir des obstacles s'ils sont dans l'environnement direct de l'avion. La solution algorithmique que nous avons choisie se base sur l'utilisation de la transformée *IPM*, déjà exploitée par le filtre particulaire. Pour cela nous avons besoin d'estimer le déplacement de l'aéronef entre deux instants  $t - 1$  et  $t$ . La prochaine section détaille l'utilisation du résultat du filtre particulaire pour estimer ce déplacement.

## 5.4 Utilisation du résultat du filtre particulaire pour l'estimation de mouvement

Dans un premier temps, avant de nous intéresser à la détection d'obstacles, nous avons besoin de travailler la donnée fournie par le filtre particulaire afin d'en récupérer une information sur le déplacement de l'avion. Dans cette brique de l'algorithme, nous nous intéressons à l'estimation du mouvement de l'avion. Les blocs algorithmiques précédents nous ont permis de récupérer, en sortie du filtre particulaire, une liste de positions potentielles des balises lumineuses dans notre image, ainsi qu'une liste des marquages au sol et des délimitations tarmac/herbe, dont nous connaissons les positions des cellules de la carte du sol associées aux marquages et dont nous avons estimé les équations des droites.

Dans le cas présenté en Figure 5.1(b), nous traitons les marquages pointillés comme représentant un seul grand segment. Pour cela, nous récupérons les informations des différents marquages fournies par le filtre particulaire et estimons une nouvelle équation de type  $0 = ax + by + c$  qui regroupent tous les marquages pointillés appartenant à un même marquage global, avec  $(x, y)$  qui définissent les coordonnées d'une cellule dans la carte du sol.

Les algorithmes que nous avons vu précédemment sont appliqués aux cartes du sol *IPM* aux temps  $t - 1$  et  $t$ . Pour les prochaines étapes, nous récupérons ces vecteurs d'informations aux temps  $t - 1$  et  $t$  et travaillons à les faire correspondre pour en déduire une matrice de mouvement en rotation et translation de l'avion entre les temps  $t - 1$  et  $t$ .

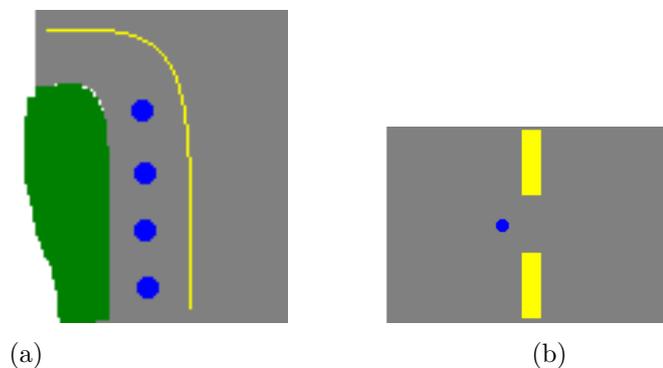


Figure 5.1: Schéma des interactions entre délimitations du tarmac (frontière vert-gris), balises (en bleu) et marquages au sol (en jaune). Les éléments peuvent être utilisés seuls pour former un motif ou en combinaison (par exemple marquage + liste de balises et leur position en fonction du marquage) pour créer des motifs plus complexes qui seront appariés pour estimer la matrice de mouvement.

Nous avons fait le choix de s'appuyer sur des éléments présents dans les zones aéroportuaires (délimitation tarmac/herbe, balises, marquages) plutôt que sur des primitives visuelles classiques (points d'intérêt, SVO [50], ORB-SLAM [124]...) vu :

1. l'uniformité des scènes aéroportuaires : points d'intérêt peu stables dans l'herbe ou sur tarmac. Nous avons, au LAAS, essayé une approche SLAM générique (C-SLAM, inspiré des travaux de J. Sola sur EKF-SLAM [150]) dans le projet SART [58]. Cependant, nous avons rencontré une difficulté pour valider la précision de telles approches sur des séquences réelles. Néanmoins,

les travaux de ONERA sur projet ATTOL ont montré qu'une approche SLAM par optimisation peut fonctionner dans ce contexte [48].

2. que le problème d'uniformité est encore plus fort sur les images de synthèse. Or, le but de COCOTIER est de valider le simulateur de OKTAL-SE et les algorithmes sur une base d'images simulées avant de les valider sur une base d'images réelles

Par ailleurs, l'extraction des motifs aéroportuaires (marquages, balises) sert à d'autres besoins opérationnels. Dans tous les cas, ces motifs seront extraits et suivis pendant des mouvements de l'aéronef au sol. Par exemple, nous avons la possibilité d'exploiter ces motifs (les marquages surtout), pour la localisation absolue dans l'aéroport, comme proposé dans le projet SART (thèse de A. Gonzalez [58]).

Nous ne sommes pas allés assez loin sur la représentation des marquages dans les virages : dans ce cas, l'approximation polygonale que nous utilisons (algorithme de la corde) ne suffit pas. Dans la nouvelle version du filtre particulaire sur laquelle nous travaillons avec E. Perrotin, ainsi que pour les besoins du suivi temporel des marquages, nous exploitons l'approximation par quadrique, via une optimisation robuste.

### 5.4.1 Motifs d'intérêt

Les informations sur les balises, marquages et délimitations du tarmac sont les données d'entrée de l'algorithme d'estimation de mouvement et vont nous permettre de composer des motifs d'intérêt. Les marquages et balises peuvent être utilisés seuls pour former ces derniers. Cependant, nous disposons aussi des spécifications aéronautiques qui nous permettent de contraindre notre modèle de positionnement des balises entre elles-mêmes, ainsi qu'entre elles et les marquages ou limites du tarmac. Les Figures 5.1(a) et 5.1(b) représentent un exemple d'interactions que nous pouvons retrouver dans les zones aéroportuaires entre ces différents objets. Les contraintes aéronautiques nous permettent aussi de rendre notre modèle plus robuste aux occultations. En effet, les balises possédant un certain espacement les unes par rapport aux autres, nous pouvons inférer la position d'une balise qui pourrait être masquée par une ombre ou un objet.

Ainsi les trois motifs retenus dans la suite de l'algorithme sont les balises, les marquages et des assemblages géométriques des motifs précédents. Ces derniers sont formés en mettant en relation les informations de positions des balises, marquages et délimitations tarmac/herbe (lorsque ces dernières sont disponibles) et leur disposition spatiale les uns par rapport aux autres. La Figure 5.1 présente par exemple deux cas où l'on peut former des assemblages géométriques.

Les trois types de motifs d'intérêt sont définis de manière automatique en suivant l'Algorithme 3, qui explicite en particulier la création des différents motifs d'intérêt. Pour cela, nous créons une classe mère *Motif* et trois classes filles *Balise*, *Ligne* et *AG*, comme représenté en Figure 5.2. Dans un second temps, on pourrait complexifier l'assemblage géométrique en sauvegardant la position spatiale de ses composants (balises, marquages, délimitations tarmac/herbe) les uns par rapport aux autres.

Pour chaque carte du sol (c'est à dire aux instants  $t - 1$  et  $t$ ) nous disposons de trois listes de motifs : une liste de balises, une liste de marquages et une liste

---

**Algorithm 3** Génération des trois vecteurs d'informations pour l'estimation de mouvement.

---

**Input :**  $vecBalises$ ,  $vecMarquages$ ,  $vecDelimTH$

**Pour** chaque  $Marquage$  de  $vecMarquages$  :

Créer un  $AG$  vide

**Pour** chaque  $Balise$  de  $vecBalises$  :

Calculer la distance de  $Balise$  à la droite de  $Marquage$

**Si** distance faible **alors**

Associer  $Balise$  à  $AG$

**Fin Si**

**Fin Pour**

**Pour** chaque  $DelimTH$  de  $vecDelimTH$  :

**Si**  $DelimTH$  est parallèle à  $Marquage$  **alors**

Calculer la distance de  $DelimTH$  à la droite de  $Marquage$

**Si** distance faible **alors**

Associer  $DelimTH$  à  $AG$

**Fin Si**

**Fin Si**

**Fin Pour**

**Si**  $AG$  n'est pas vide **alors**

Associer  $Marquage$  à  $AG$

Estimer position centrale de  $AG$

Ajouter  $AG$  à  $vecAG$

**Fin Si**

**Fin Pour**

**Output :**  $vecBalises$ ,  $vecMarquages$ ,  $vecAG$

---

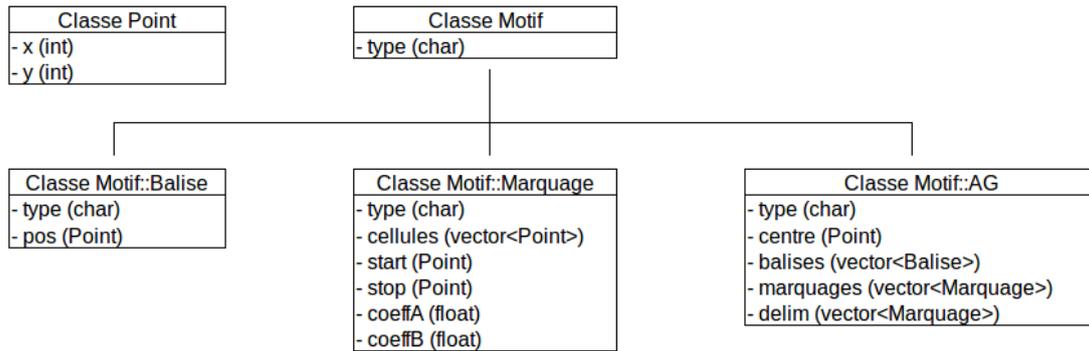


Figure 5.2: Représentation des classes utilisées pour définir les différents motifs d'intérêt utilisés pour l'estimation de mouvements.

d'assemblages géométriques. Ces motifs sont utilisés pour estimer le déplacement de l'aéronef en les associant d'une image à l'autre au moyen de matrices de coût.

### 5.4.2 Matrices de coût

Pour estimer le déplacement de notre avion, nous allons mettre en place un algorithme de mise en correspondance des motifs d'une image à l'autre. Pour cet algorithme, nous allons traiter les différents types de motifs (marquages, balises, et assemblages géométriques) séparément. Pour chaque type de motif, nous créons d'abord une matrice de coûts entre les motifs extraits des cartes du sol  $IPM_{t-1}$  et  $IPM_t$  par le filtre particulière, comme schématisé par la Figure 5.3. La fonction de coût utilisée pour la création de la matrice de coût va varier selon les motifs.

	$M_{1_t}$	...	$M_{n_t}$
$M_{1_{(t-1)}}$	<i>Coût</i>	...	...
...	...	...	...
$M_{m_{(t-1)}}$	...	...	...

Figure 5.3: Représentation d'une matrice de coût (calcul de similitude) entre des motifs de la carte du sol à l'instant  $t - 1$  (en bleu) et des motifs de la carte du sol à l'instant  $t$  (en rouge).

#### 5.4.2.a Matrice de coût des balises

Pour les balises, nous étudions la différence de position  $(x, y)$  entre les balises des cartes du sol à l'instant  $t$  et à l'instant  $t - 1$ .

$$\text{Coût}_{\text{balise}} = \sqrt{(x_{t-1} - x_t)^2 + (y_{t-1} - y_t)^2}$$

En effet, nous avons besoin de pouvoir estimer le mouvement de l'aéronef quand celui-ci se déplace à 60km/h (ou 16m/s) sur le tarmac. En échantillonnant à 10 images par secondes, la différence de position d'une balise entre les instants  $t - 1$  et  $t$  est inférieure à la distance entre deux balises ce qui nous permet de faire l'hypothèse que l'on peut apparier les balises d'un instant à l'autre sans risquer fortement de l'apparier avec une seconde balise de l'image.

#### 5.4.2.b Matrice de coût des marquages

On assimile les marquages à des droites dont on estime les paramètres de l'équation de la droite. La fonction de coût est composée de deux coûts qui représentent un calcul de similarité entre les paramètres des droites, exprimé aussi bien dans les coordonnées  $(x, y)$  de la carte du sol que par leurs coordonnées polaires  $(\rho(\phi), \phi)$ . Pour passer en coordonnées polaires, on fixe le point situé en bas et au centre de la carte du sol comme centre du repère et on exprime les droites en fonction de ce point.

De la même façon que pour les balises, on peut alors mesurer une similitude entre les coordonnées des marquages détectés dans les cartes du sol aux temps  $t - 1$  et  $t$ . L'Algorithme 4 décrit le calcul du coût pour les marquages. On prend ici en exemple  $M_i$  et  $M_j$  deux marquages aux temps  $t - 1$  et  $t$ .

#### 5.4.2.c Matrice de coût des assemblages géométriques

Pour les motifs d'assemblages géométriques, plusieurs facteurs entrent en compte dans la fonction de coût. On prend ici en exemple  $A_i$  et  $A_j$  deux assemblages géométriques aux temps  $t - 1$  et  $t$ . On peut alors écrire la fonction de coût ci-dessous.

Nous prenons en compte les nombre de composants (balises, marquages, délimitations) formant le motif  $(\Delta_{Nb_{balises}}, \Delta_{Nb_{marquages}}$  et  $\Delta_{Nb_{delimTH}}$ ), ainsi que leur organisation géométrique : position  $(Cost_{pos_{B/M}})$  et différence de distance moyenne  $(\Delta_{dist_{B/M}})$  des balises par rapport aux marquages au sol, position  $(Cost_{pos_{TH/M}})$  et différence de distance moyenne  $(\Delta_{dist_{TH/M}})$  des marquages par rapport aux délimitations tarmac/herbe, alignement  $(Cost_{alignB})$  et différence d'espacement  $(\Delta_{EspB})$  entre les balises des deux motifs.

Les différents éléments de cette fonction de coût sont calculés en fonction des composants des motifs. Certains ne sont calculés que si les deux motifs possèdent le même type de composant (par exemple une délimitation tarmac/herbe).

$$Coût_{AG} = \sqrt{(\Delta_{Nb_{balises}}(A_i, A_j))^2 + (\Delta_{Nb_{marquages}}(A_i, A_j))^2 + (\Delta_{Nb_{delimTH}}(A_i, A_j))^2 + (Cost_{pos_{B/M}}(A_i, A_j))^2 + (\Delta_{dist_{B/M}}(A_i, A_j))^2 + (Cost_{pos_{TH/M}}(A_i, A_j))^2 + (\Delta_{dist_{TH/M}}(A_i, A_j))^2 + (Cost_{alignB}(A_i, A_j))^2 + (\Delta_{EspB}(A_i, A_j))^2}$$

En se basant sur les recommandations aéronautiques, on peut définir une translation maximale de l'aéronef entre deux images, en fonction de la vitesse maximale de l'avion et de la fréquence d'échantillonnage. De la même façon, on peut définir une rotation maximale de l'aéronef, en fonction de la vitesse maximale de l'avion, de la fréquence d'échantillonnage et de la courbure maximale des virages.

Lorsque nous remplissons la matrice de coût pour les assemblages géométriques, nous pouvons appliquer une suite de rotations et translations au motif  $A_j$  afin de récupérer le coût minimal entre les deux motifs comme suit :

**Pour**  $t \in \{\text{translations possibles}\}$  :  
**Pour**  $r \in \{\text{rotations possibles}\}$  :  
 Appliquer  $t$  et  $r$  au motif  $A_j$   
 Calculer  $\text{Coût}_{AG}(t, r)$   
**Fin Pour**  
**Fin Pour**  
 $\text{Coût}_{AG} = \min(\text{Coût}_{AG}(t, r))$   
 Enregistrer  $t$  et  $r$  permettant d'obtenir le coût minimal

---

**Algorithm 4** Calcul du coût entre deux marquages.

---

**Input** :  $M_i, M_j$

*Comment* : Calcul du coût cartésien

**Si** Les droites représentant les marquages  $M_i$  et  $M_j$  sont verticales (reps. horizontales) dans l'image **alors**

*Comment* : On appelle *cross* la valeur de la colonne (resp. ligne) de la carte du sol qui représente le marquage

$$\text{Coût}_{\text{Cart}} = \sqrt{(\text{cross}_{M_i} - \text{cross}_{M_j})^2}$$

**sinon**

*Comment* : On note  $a$  le coefficient directeur et  $b$  l'ordonnée à l'origine de la droite représentant le marquage

**Si**  $a_{M_i}$  est de même signe que  $a_{M_j}$  **alors**

$$\text{Coût}_{\text{Cart}} = \sqrt{(a_{M_i} - a_{M_j})^2 + (b_{M_i} - b_{M_j})^2}$$

**sinon**

*Comment* : On note  $(x, y)$  les coordonnées du point central du segment représentant le marquage dans la carte du sol

$$\text{Coût}_{\text{Cart}} = \sqrt{(\|a_{M_i}\| - \|a_{M_j}\|)^2 + (x_{M_i} - x_{M_j})^2 + (y_{M_i} - y_{M_j})^2}$$

**Fin Si**

**Fin Si**

*Comment* : Calcul du coût polaire où  $\text{sim}()$  est une fonction de calcul de similitude entre les angles  $\phi_{M_i}$  et  $\phi_{M_j}$  en fonction de  $2\pi$

$$\text{Coût}_{\text{pol}} = \sqrt{(\rho_{M_i} - \rho_{M_j})^2 + \text{sim}(\phi_{M_i}, \phi_{M_j})}$$

*Comment* : Calcul du coût total

$$\text{Coût}_{\text{marquage}} = \text{Coût}_{\text{cart}} + \text{Coût}_{\text{pol}}$$

**Output** :  $\text{Coût}_{\text{marquage}}$

---

### 5.4.3 Estimation des matrices de rotation et de translation

Afin de pouvoir déduire une matrice de déplacement de l'aéronef dans le plan du sol, nous devons estimer la rotation et la translation effectuées par l'aéronef entre deux images. Pour cela, nous avons récupéré des motifs détectés dans deux cartes du sol aux instants  $t - 1$  et  $t$ , que nous cherchons à apparier afin d'obtenir des paires de motifs correspondants.

Nous avons créé trois matrices de coût entre les différents types de motifs détectés dans les deux cartes du sol. Pour apparier les motifs, nous pouvons appliquer des algorithmes de résolution de problèmes d'appariement bipartite et non bipartite, comme présenté dans [36] et [167], à nos matrices de coût.

Ces algorithmes nous permettent de former des paires entre les motifs extraits des cartes du sol  $IPM_{t-1}$  et  $IPM_t$  en minimisant le coût de chacune des paires par rapport aux autres. Pour chaque type de motif, nous sélectionnons deux paires de manière aléatoire, que nous nommons  $pA$  et  $pB$ . Ces paires de motifs sont alors utilisées pour l'estimation des matrices de rotation et translation. De la même façon que pour les fonctions de coût, nous utilisons trois approches différentes pour l'estimation.

#### 5.4.3.a Estimation grâce aux balises et aux marquages

Nous obtenons deux équations comme montré dans l'Équation 5.1 où  $T_x$  et  $T_y$  sont les composantes de la translation et  $\theta$  est l'angle de rotation.

$$\begin{cases} x_{t-1_{pA}} = x_{t_{pA}} \cdot \cos(\theta) - y_{t_{pA}} \cdot \sin(\theta) + T_x \\ y_{t-1_{pA}} = x_{t_{pA}} \cdot \sin(\theta) + y_{t_{pA}} \cdot \cos(\theta) + T_y \end{cases} \quad (5.1)$$

Nous définissons ensuite des segments  $segA$  et  $segB$  en reliant  $(x_{t-1}, y_{t-1})_{pA}$  à  $(x_{t-1}, y_{t-1})_{pB}$  et  $(x_t, y_t)_{pA}$  à  $(x_t, y_t)_{pB}$ ,  $(x, y)$  étant les coordonnées du pixel représentant la balise, définie par le pixel central de l'objet. Ces deux segments peuvent être exprimés par des vecteurs euclidiens de type  $(1, a)$  où  $a$  est le coefficient directeur du segment. Nous pouvons interpréter  $\theta$  comme l'angle entre ces deux segments en utilisant l'Équation 5.2.

Si un des coefficients directeur n'est pas calculable (i.e. les balises possèdent la même abscisse), nous nous référons aux coordonnées polaires du segment pour obtenir l'Équation 5.3.

$$\cos(\theta) = \frac{1 + a_{segA} \cdot a_{segB}}{\sqrt{1 + a_{segA}^2} \cdot \sqrt{1 + a_{segB}^2}} \quad (5.2)$$

$$\begin{cases} \rho_{segA} = \rho_{segB} - T_x \cdot \sin(\phi_{segB} + \theta) + T_y \cdot \cos(\phi_{segB} + \theta) \\ \phi_{segA} = \phi_{segB} + \theta \end{cases} \quad (5.3)$$

Ces équations sont utilisées pour obtenir les matrices de rotation et de translation dans le cas du motif *Balise*. Lorsque l'on travaille avec les paires de marquages, nous utilisons la même méthode puisque nous possédons l'équation des droites qui représentent les marquages, ainsi que la position d'au moins un point de cette droite.

Nous obtenons donc trois estimations de la rotation et de la translation de l'aéronef : la première provient de l'étude des deux paires  $pA_{Balise}$  et  $pB_{Balise}$  de

balises, la seconde et la troisième sont obtenues en utilisant la même méthode sur les droites des paires  $pA_{\text{Marquage}}$  et  $pB_{\text{Marquage}}$ .

#### 5.4.3.b Estimation grâce aux assembléments géométriques

Lorsque nous avons créé la matrice de coût dans le cas des motifs d'assembléments géométriques, nous avons enregistré la meilleure estimation de rotation et de translation entre un assemblément des cartes du sol  $IPM_{t-1}$  et  $IPM_t$ , c'est à dire celle qui minimise le coût de similitude lors de la superposition de deux motifs.

#### 5.4.3.c Choix de la matrice de déplacement de l'aéronef dans le plan du sol

On construit chaque matrice de déplacement de l'aéronef dans le plan du sol,  $Mov$ , définie par l'Équation 5.4, en utilisant les rotations et translations déduites des estimations précédentes. Ces matrices de mouvement seront ensuite mises en compétition pour améliorer la prédiction du mouvement.

$$Mov = \begin{bmatrix} \cos(\theta_{mov}) & -\sin(\theta_{mov}) & T_x \\ \sin(\theta_{mov}) & \cos(\theta_{mov}) & T_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

À ce niveau, puisque nous avons sélectionné à chaque fois deux paires de motifs, nous possédons cinq matrices d'estimation du mouvement de l'aéronef, une provenant des balises, deux des marquages au sol et deux des assembléments géométriques. Ces cinq matrices sont alors appliquées à toutes les paires de motifs, quelque soit le type de motif, et chaque paire vote pour une matrice de mouvement. Un seuil est fixé, basé sur le nombre des paires obtenues. Après le vote, nous sélectionnons la matrice de mouvement ayant obtenu le meilleur score. Si les paires lui ont accordé assez de votes pour dépasser le seuil de tolérance, la matrice est choisie comme estimée du mouvement. Autrement, le processus recommence avec de nouvelles paires aléatoirement sélectionnées. Si aucune matrice ne ressort après plusieurs itérations du processus, nous n'avons pas assez de motifs concordants pour estimer la matrice de mouvement. Dans ce cas, la détection d'obstacles n'est pas effectuée et le module passe à l'image suivante en appliquant la dernière matrice de mouvement calculée, puisque l'on part de l'hypothèse qu'entre deux images, la matrice de déplacement de l'aéronef ne varie pas significativement et que l'on possède une cohérence temporelle entre deux images.

Nous avons réfléchi à mettre en place une approche à base d'ICP (Iterative Closest Point) cependant nous disposons de peu de motifs à apparier donc l'utilité d'une telle méthode est relative.

#### 5.4.4 Précision de l'estimation des matrices de rotation et translation

Pour l'instant, nous disposons d'images issues du simulateur pour lesquelles nous n'avons pas l'information de déplacement de l'avion entre les instants  $t - 1$  et  $t$ . De même, nous n'avons que quelques images acquises en conditions réelles, sans informations sur le déplacement de l'avion. Cependant, le simulateur a évolué

depuis le début du projet et nous disposons depuis peu de la possibilité d'accéder à la position d'un objet du simulateur (avion, caméra ou obstacles) à l'instant  $t$ . Nous envisageons donc de mener une étude quantitative de grande ampleur sur la précision de l'estimation des matrices de rotation et de translation.

Les résultats préliminaires que nous possédons, effectués sur 92 images consécutives, sont consignés dans le Tableau 5.1 où *EC* et *EE* signifient respectivement "Estimation Correcte" et "Estimation Erronée" de la matrice de déplacement. Une estimation est considérée comme correcte lorsque la différence de rotation entre l'estimation et la vérité terrain ainsi que la différence de translation est égale à moins de 10% des rotations et translations maximales pouvant être effectuées pour un aéronef entre les instants  $t - 1$  et  $t$ .

Ligne droite		Virage		Global	
EC	EE	EC	EE	EC	EE
83.15%	16.85%	62.92%	37.08%	70.79%	29.21%

Table 5.1: Tableau de résultats préliminaires pour l'estimation du déplacement d'avion dans le plan du sol.

Les matrices de rotation et de translation estimées permettent dans l'ensemble de modéliser le déplacement de l'avion d'une image à l'autre de manière satisfaisante. On note cependant que l'estimation est rendue difficile quand la composante de rotation du mouvement est importante. La détection des motifs d'intérêt (balises, marquages...) peut être pénalisée dans le cas où l'avion entame un virage et réalise une manœuvre d'oversteering, puisque le nombre de ces motifs présents sur l'image est fortement réduit. Par conséquent, le résultat de l'estimation des matrices de rotation et de translation se dégrade. L'utilisation de caméras placées sur les côtés de l'aéronef en simulation devrait permettre de rendre plus robuste la détection de motifs. Cependant, il n'existe pas encore de caméras montées à ces endroits de l'aéronef qui nous permettraient de faire une validation sur des images acquises en conditions réelles.

Enfin, les mauvaises estimations du déplacement de l'avion sont espacées dans le temps alors que l'avion se déplace à une vitesse presque constante au cours de la séquence d'images. L'utilisation d'une heuristique de sélection de la matrice de déplacement en fonction de la matrice estimée à l'instant précédent nous permet, dans la plupart des cas, de remplacer la mauvaise estimation par l'estimation précédente.

## 5.5 Méthode géométrique pour la détection d'obstacles

Pour réaliser la détection d'obstacles, nous avons décidé d'utiliser, parmi les méthodes présentées dans l'état de l'art, la création d'une carte d'obstacles. Comme nous utilisons déjà la transformation IPM dans la détection de motifs, cela nous permet de mutualiser les traitements appliqués aux images.

Comme défini dans l'article [118] et présenté dans le Chapitre 3, nous appliquons une transformation IPM à nos images, en utilisant les Équations 5.5 à 5.8 pour passer du repère monde au système de coordonnées de l'image.

$$H = K \cdot [X_{W/C} \ Y_{W/C} \ 0_{W/C}] \quad (5.5)$$

$$H = K \cdot \begin{bmatrix} -1 & 0 & 0 \\ 0 & \sin(\theta) & D \cdot \cos(\theta) \\ 0 & -\cos(\theta) & D \cdot \sin(\theta) \end{bmatrix} \quad (5.6)$$

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -\Delta X & \Delta X \cdot D_j / 2 \\ \Delta Y & 0 & -\Delta Y \cdot D_i \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad (5.7)$$

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (5.8)$$

Pour rappel, on nomme  $H$  l'homographie qui permet de passer de la carte du sol à l'image. On note  $X_{W/C}$  et  $Y_{W/C}$  les vecteurs  $X$  et  $Y$  du repère monde (indiqué  $W$ ) exprimés dans le repère caméra (indiqué  $C$ ) et  $O_{W/C}$  l'origine du repère monde dans le repère caméra.  $K$  est la matrice intrinsèque de la caméra,  $D$  est la hauteur de la caméra par rapport au sol et  $\theta$  correspond à l'inclinaison de la caméra.  $D_i$  et  $D_j$  représentent la taille de la carte du sol et les dimensions d'une cellule de la carte sont définies par  $(\Delta X, \Delta Y)$ . Chaque cellule au sol est repérée par des coordonnées  $(i, j)$  dans la carte, et par ses coordonnées  $(X, Y)$  dans le repère cartésien lié au sol. Les coordonnées d'un pixel dans l'image sont données par  $(u, v)$ .

À ce point de l'algorithme, nous avons donc deux cartes du sol  $IPM$  aux temps  $t-1$  et  $t$  ainsi qu'une matrice estimée du mouvement de l'avion dans le plan du sol,  $Mov$ , composée d'une matrice de rotation et d'une matrice de translation. Nous pouvons donc appliquer la matrice de mouvement à la carte du sol créée au temps  $t-1$  pour exprimer cette dernière au temps  $t$ , créant ainsi la carte du sol corrigée notée  $IPM_{(t-1)/t}$ . Le processus est défini dans la Figure 5.4.

Dans le but de créer la carte du sol  $IPM_{(t-1)/t}$ , nous pouvons modifier l'Équation 5.7 pour lier les coordonnées  $(X_{(t-1)/t}, Y_{(t-1)/t})$  et  $(i_{(t-1)/t}, j_{(t-1)/t})$ , comme montré par l'Équation 5.9. La matrice estimée de mouvement est alors appliquée aux coordonnées des cellules de la carte du sol  $IPM_{(t-1)/t}$  en se servant de l'Équation 5.10 pour récupérer les coordonnées  $(X_{t-1}, Y_{t-1})$  correspondantes, où  $\theta_{mov}$  est l'angle de rotation entre deux images,  $T_x$  et  $T_y$  sont les composantes de la translation. En utilisant la relation entre les systèmes de coordonnées  $(X, Y)$  et  $(i, j)$  définis par les

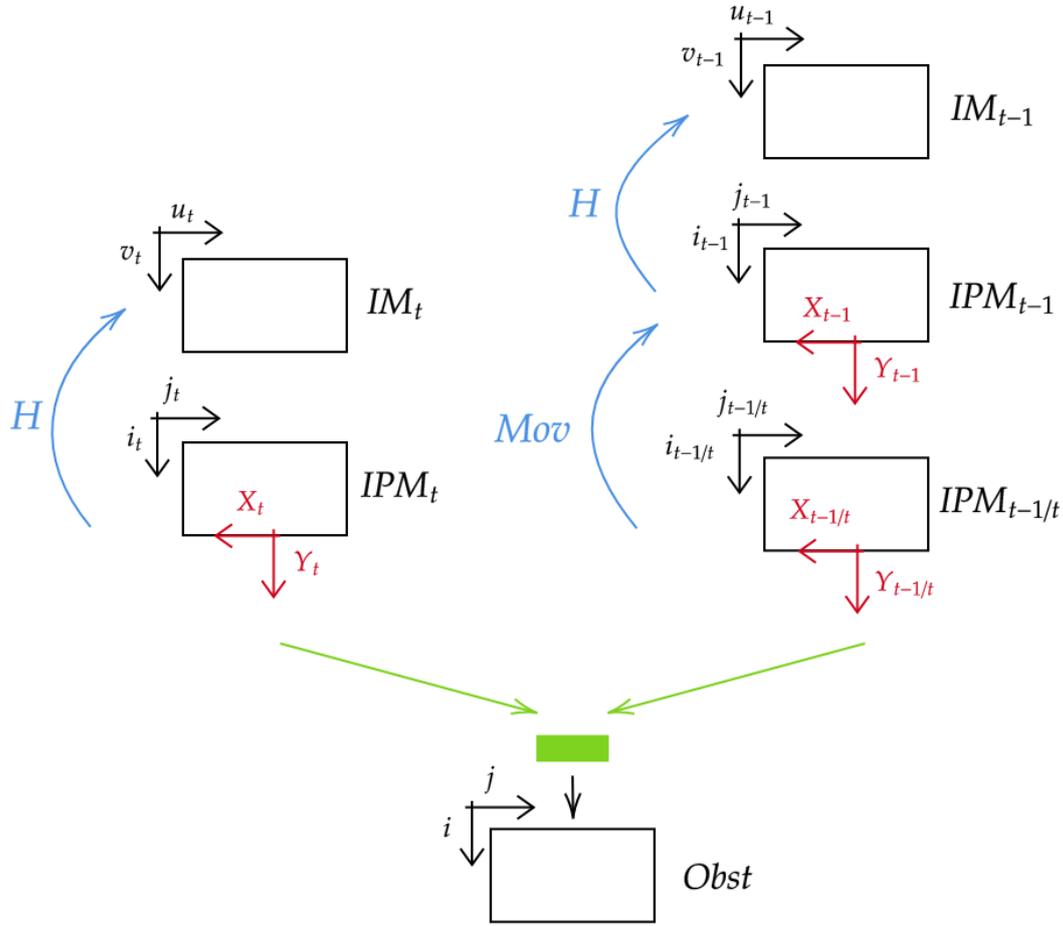


Figure 5.4: Création des cartes du sol *IPM* ainsi que de la carte d'obstacles *Obst* depuis les images originales *IM*.

Équations 5.7 et 5.9 on peut récupérer l'intensité de la cellule  $(i_{(t-1)/t}, j_{(t-1)/t})$  en utilisant les coordonnées  $(i_{(t-1)}, j_{(t-1)})$  et en appliquant une interpolation bilinéaire.

$$\begin{bmatrix} X_{(t-1)/t} \\ Y_{(t-1)/t} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -\Delta X & \frac{\Delta X \cdot D_j}{2} \\ \Delta Y & 0 & -\Delta Y \cdot D_i \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{(t-1)/t} \\ j_{(t-1)/t} \\ 1 \end{bmatrix} \quad (5.9)$$

$$\begin{bmatrix} X_{(t-1)} \\ Y_{(t-1)} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_{mov}) & -\sin(\theta_{mov}) & T_x \\ \sin(\theta_{mov}) & \cos(\theta_{mov}) & T_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{(t-1)/t} \\ Y_{(t-1)/t} \\ 1 \end{bmatrix} \quad (5.10)$$

En supposant que les obstacles sont des prismes droits posés au sol, l'exploitation des transformations IPM à deux instants  $t - 1$  et  $t$  permet de détecter de manière exacte, la trace de l'obstacle au sol, avec une déformation derrière l'obstacle par rapport à la caméra. Si l'obstacle n'est pas un prisme droit, comme c'est le cas pour un avion, alors, cette trace au sol est fautive : l'obstacle est détecté plus loin qu'il n'est en réalité.

Ces erreurs sont illustrées en Figure 5.5, dans laquelle nous représentons le plan image, le centre optique de la caméra  $C$ , le plan du sol, un obstacle  $UVWX$  posé sur le sol, la zone de projection  $U'W'$  de cet obstacle dans le plan image ( $V$  se projette

dans la zone  $U'W'$ ,  $X$  est occulté), et la projection inverse  $U''W''$  résultat de la transformation IPM sur le plan du sol.

En Figure 5.5(a), vu que le point  $U$  est sur le sol,  $U''$  se confond avec  $U$  : si on exploite la transformation IPM pour détecter la trace de l'obstacle au sol, la trace est exacte à l'avant, mais on commet une erreur  $E_{ARRIÈRE}$  (entre  $X$  et  $W''$ ) derrière l'obstacle, ce qui n'a pas de conséquence du point de vue de la navigation.

Par contre en Figure 5.5(b), l'obstacle n'est pas un prisme droit. Le point  $U$  n'est pas sur le sol (par exemple : le nez d'un aéronef est plus avancé que la roulette avant en contact avec le sol). On retrouve la même erreur  $E_{ARRIÈRE}$  derrière l'obstacle, mais il apparaît aussi une erreur  $E_{AVANT}$  (entre la projection de  $U$  au sol et  $U''$ ). Ceci est plus grave pour la navigation, car l'obstacle est détecté plus loin de la caméra qu'il n'est en réalité.

On peut obtenir la carte d'obstacles en soustrayant  $IPM_{(t-1)/t}$  à  $IPM_t$ . La transformation IPM permet de projeter tous les pixels au niveau du sol. Lorsque l'on effectue cette transformation, les pixels correspondant au sol ne sont pas distordus, mais tous les autres pixels n'appartenant pas au sol vont subir une déformation. Plus un objet est loin de la caméra, plus les pixels qui le définissent vont subir une déformation. Lorsque l'on récupère le résultat de la soustraction des deux cartes du sol  $IPM_t$  et  $IPM_{(t-1)/t}$ , les pixels représentant le sol vont disparaître et seule la différence induite par une mauvaise représentation des autres pixels (qui va se traduire par les seuls pixels blanc de la carte d'obstacles) va permettre de mettre en lumière des obstacles potentiels se trouvant au dessus du niveau du sol.

Nous pouvons aussi noter qu'un objet qui se déplacerait dans la même direction que l'aéronef et à la même vitesse, n'apparaîtrait pas sur la carte d'obstacles. Cela étant, cet objet ne serait pas un obstacle puisqu'il ne se rapproche pas de l'avion.

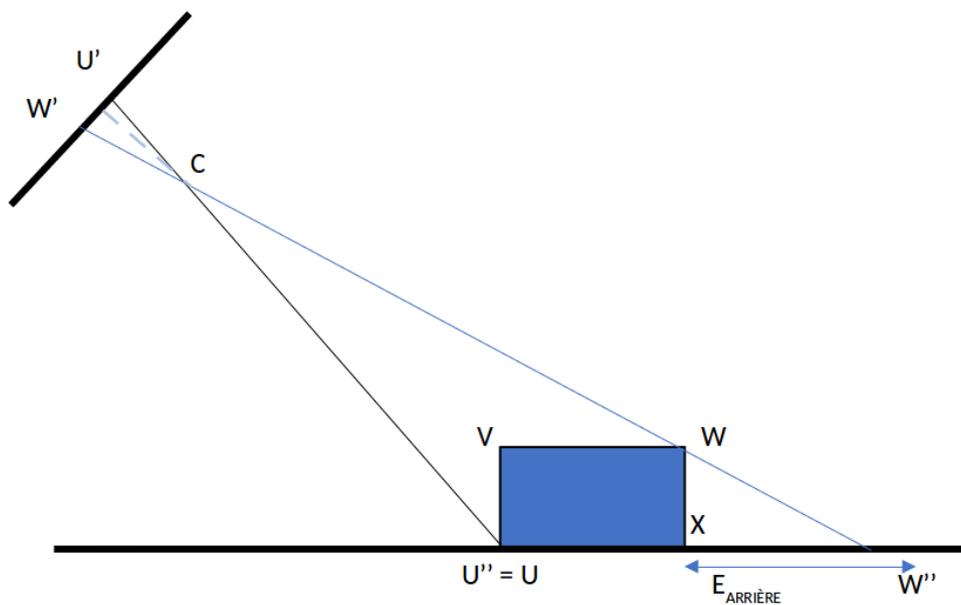
Utiliser une transformation IPM va augmenter le bruit dans notre image puisque la résolution des pixels se dégrade en relation avec leur distance à la caméra. De plus, cette vue n'est pas la meilleure pour remonter des informations au pilote. En utilisant l'Équation 5.11, on peut se passer de l'utilisation de la carte du sol créée par la transformation IPM et reconstruire  $IM_{(t-1)/t}$  directement. La représentation de la carte d'obstacles dans une image augmentée est alors donnée par la différence entre  $IM_t$  et  $IM_{(t-1)/t}$ . De manière analogue, nous utilisons l'intensité du pixel  $(u_{(t-1)}, v_{(t-1)})$  pour construire le pixel  $(u_{(t-1)/t}, v_{(t-1)/t})$ .

$$\begin{bmatrix} u_{(t-1)} \\ v_{(t-1)} \\ 1 \end{bmatrix} = M \cdot \begin{bmatrix} u_{(t-1)/t} \\ v_{(t-1)/t} \\ 1 \end{bmatrix} \quad (5.11)$$

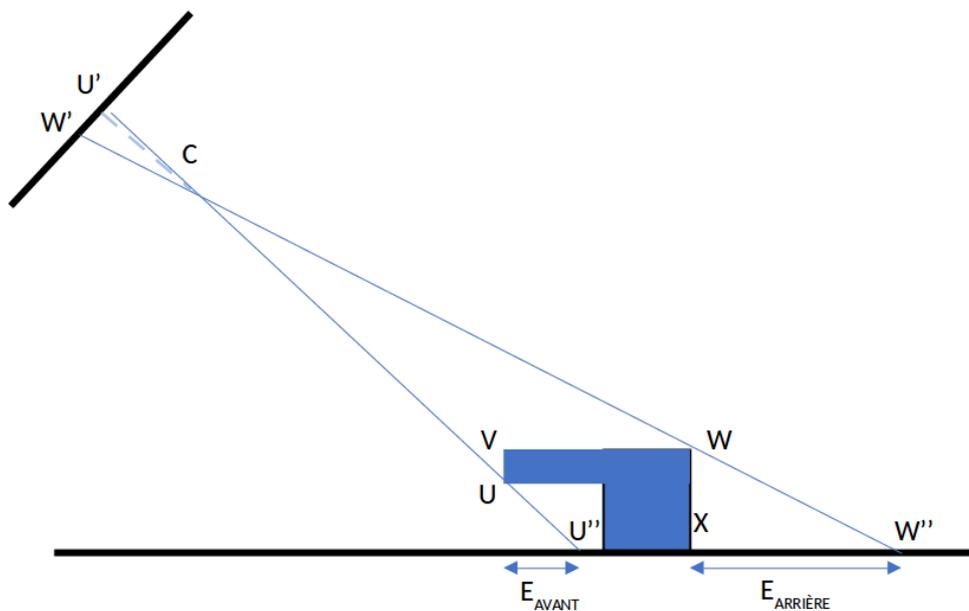
$$M = H \cdot \begin{bmatrix} \cos(\theta_{mov}) & -\sin(\theta_{mov}) & T_x \\ \sin(\theta_{mov}) & \cos(\theta_{mov}) & T_y \\ 0 & 0 & 1 \end{bmatrix} \cdot H^{-1} \quad (5.12)$$

Les résultats de ces transformations sont montrés en Figure 5.6. Dans cet exemple, le déplacement de l'avion correspond uniquement à une translation d'environ 50 mètres. Ce n'est pas un cas d'utilisation réel puisque l'aéronef ne se déplace pas aussi rapidement en 1/10ème de seconde mais cela illustre la distorsion appliquée à de larges obstacles.

Les Figures 5.6(b) et 5.6(a) représentent les versions en nuances de gris des images obtenues par la caméra. Les Figures 5.6(d) et 5.6(c) sont les cartes du sol IPM après



(a)



(b)

Figure 5.5: Représentation des erreurs induites par la transformation IPM lors de la projection sur le plan du sol pour (a) un prisme droit et (b) un obstacle pour lequel l'avant se trouve au dessus du sol.

la transformation appliquée à  $IM_{t-1}$  et  $IM_t$ . Les résultats de la compensation de mouvement sont donnés par la carte du sol  $IPM_{(t-1)/t}$  et l'image  $IM_{(t-1)/t}$  illustrées dans les Figures 5.6(e) et 5.6(f).

Les Figures 5.6(g) et 5.6(h) montrent les deux différentes méthodes de création de la carte d'obstacles probables (passage explicite par la carte du sol et utilisation de l'Équation 5.12), où la première est corrigée par les algorithmes suivants alors que la seconde non.

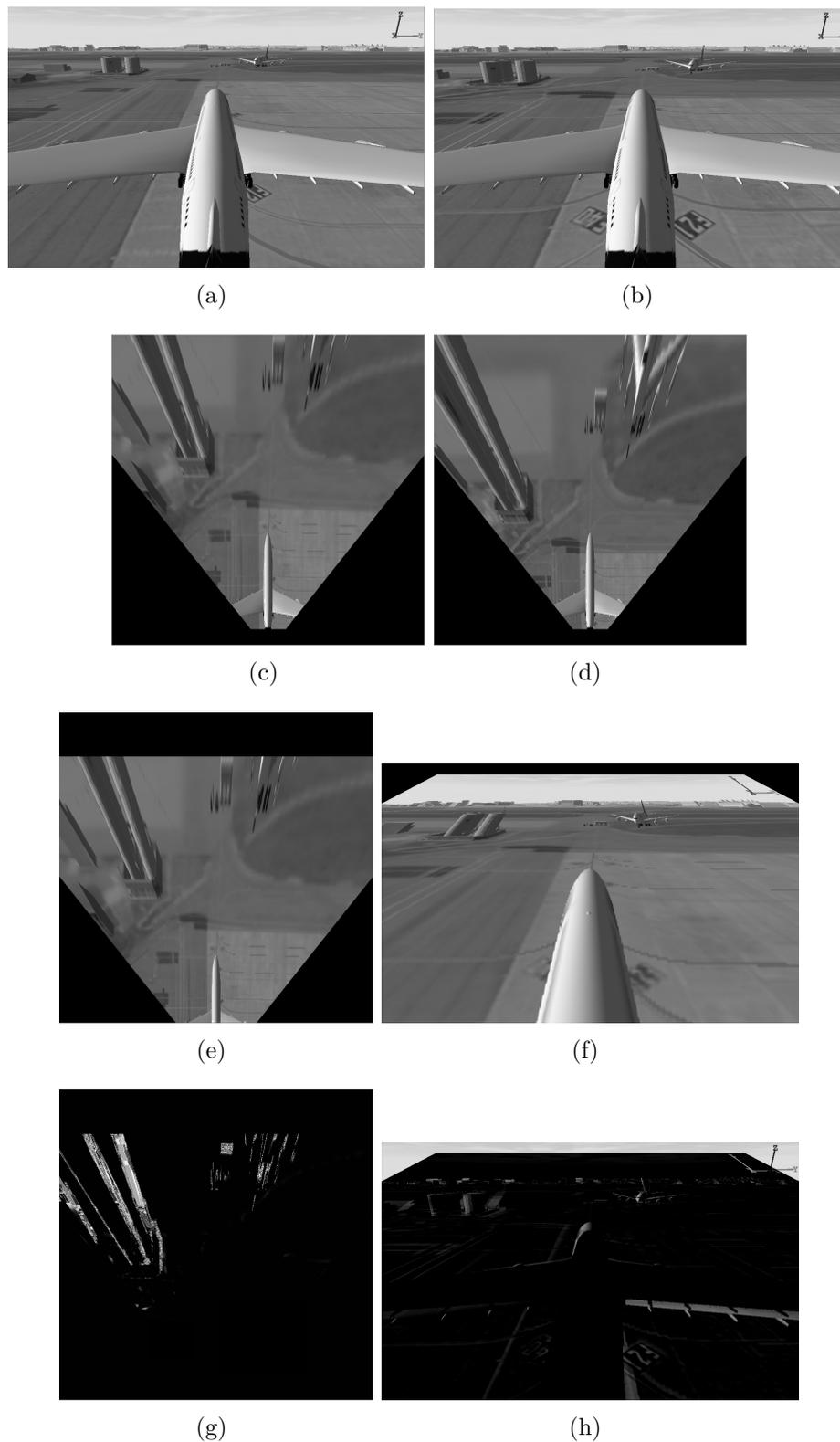


Figure 5.6: (a)  $IM_{t-1}$  l'image originelle en niveaux de gris à l'instant  $t-1$ , (b)  $IM_t$  l'image originelle en niveaux de gris à l'instant  $t$ , (c)  $IPM_{t-1}$  la carte du sol à l'instant  $t-1$ , (d)  $IPM_t$  la carte du sol à l'instant  $t$ , (e)  $IPM_{(t-1)/t}$  la carte du sol à l'instant  $t-1$  exprimée dans le repère de la carte du sol à l'instant  $t$ , (f)  $IM_{(t-1)/t}$  l'image reconstruite après projection inverse de  $IPM_{(t-1)/t}$ , (g)  $Obst$  la carte d'obstacles corrigée exprimée dans le repère du sol et (h)  $IM_{Obst}$  l'image non corrigée de la présence d'obstacles sans passage par la transformation IPM.

Des artefacts au niveau du sol peuvent être présents dans la carte d'obstacles. De plus, en soustrayant  $IM_{(t-1)/t}$  à  $IM_t$ , une part des pixels visibles au temps  $t$  ressortent puisqu'il n'étaient pas encore visibles au temps  $t - 1$ , comme nous pouvons voir sur la Figure 5.6(h). C'est pourquoi nous avons besoin d'appliquer des traitements à l'image obtenue avant de pouvoir l'utiliser.

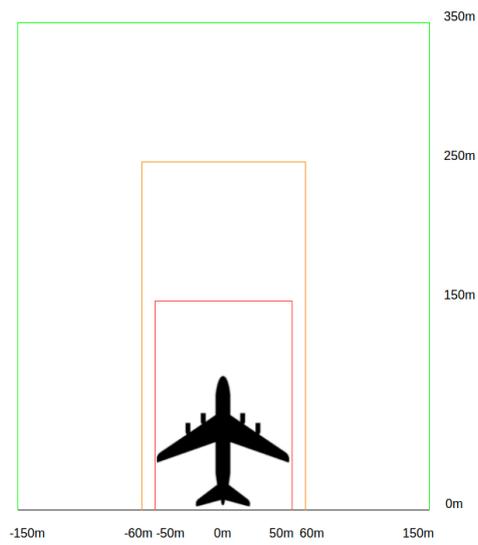
Dans un premier temps, nous retirons les pixels visibles seulement au temps  $t$ . Puis, nous améliorons le contraste entre les niveaux de gris pour faire ressortir les obstacles des artefacts en appliquant une fonction exponentielle aux intensités des pixels de l'image, suivant l'Équation 5.13 où  $\lambda$  est un paramètre à définir qui dans notre cas est fixé à 0.2 et permet d'étirer les niveaux de gris. Un filtre bilatéral est ensuite utilisé sur l'image pour séparer les niveaux de gris. Enfin, l'image est binarisée en se basant sur une valeur seuil définie par l'algorithme Otsu.

$$value_{(i,j)_{obst}} = \exp(\lambda \cdot value_{(i,j)_{t-(t-1)/t}}) \quad (5.13)$$

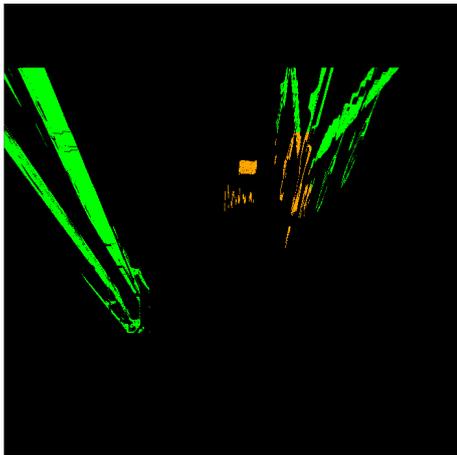
L'image obtenue après cette procédure de post-traitement est utilisée comme masque d'obstacles. Les pixels blancs de ce masque représentent alors de possibles obstacles qui vont être colorisés suivant une règle décrite en Figure 5.7(a). Cette règle définit 3 niveaux d'attention : acceptable (obstacle se situant à plus de 250m devant l'avion et à plus de 60m de chaque côté de l'avion), danger potentiel (obstacle se situant à une distance comprise entre 150 et 250m devant l'avion et entre 50 et 60m de chaque côté de l'avion) et critique (obstacle se situant à moins de 150m devant l'avion et à moins de 50m de chaque côté de l'avion). La colorisation de la carte d'obstacles  $Obst$  et de l'image  $IM_{Obst}$  est présentée dans les Figures 5.7(b) et 5.7(c), la dernière étant l'image à remonter au pilote. Comme dit précédemment, cet exemple souligne la distorsion appliquée aux larges objets comme les buildings. C'est pour cela que l'on observe un défaut dans la colorisation de la Figure 5.7(c).

La Figure 5.8 présente quant à elle un résultat de colorisation pour plusieurs cartes d'obstacles, en se basant sur l'estimation du déplacement de l'avion (matrice  $Mov$ ) et respecte les mêmes règles de colorisations que pour la Figure 5.7. On constate que des résultats de détection d'obstacles sont présents sur la Figure 5.8(f) en dessous du nez de l'avion et des réacteurs. Cela est dû à la transformation IPM d'obstacles ne se trouvant pas sur le plan du sol ( $E_{AVANT}$ ).

La localisation d'obstacles par transformation sur le plan du sol offre de meilleurs résultats lorsque l'obstacle est de faible hauteur et forme géométrique simple. Or l'aéronef déroge à ce modèle. Par la suite, nous avons proposé une solution pour améliorer la précision de la détection en effectuant une transformation IPM sur un plan à hauteur du fuselage de l'avion.



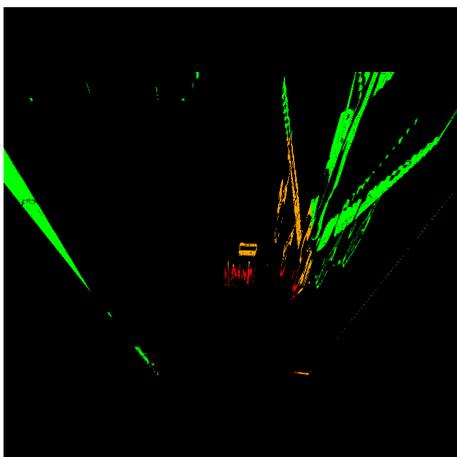
(a)



(b)



(c)



(d)



(e)

Figure 5.7: (a) Règles de colorisation en fonction de la distance à l'aéronef : vert pour acceptable, orange pour danger potentiel et rouge pour critique, (b)(d) Colorisation des objets détectés comme obstacles potentiels dans la carte d'obstacles  $IPM_{Obst}$  en fonction de leur proximité à l'aéronef, (c)(e) Application de la colorisation des obstacles potentiels dans l'image d'origine augmentée  $IM_{Obst}$ .

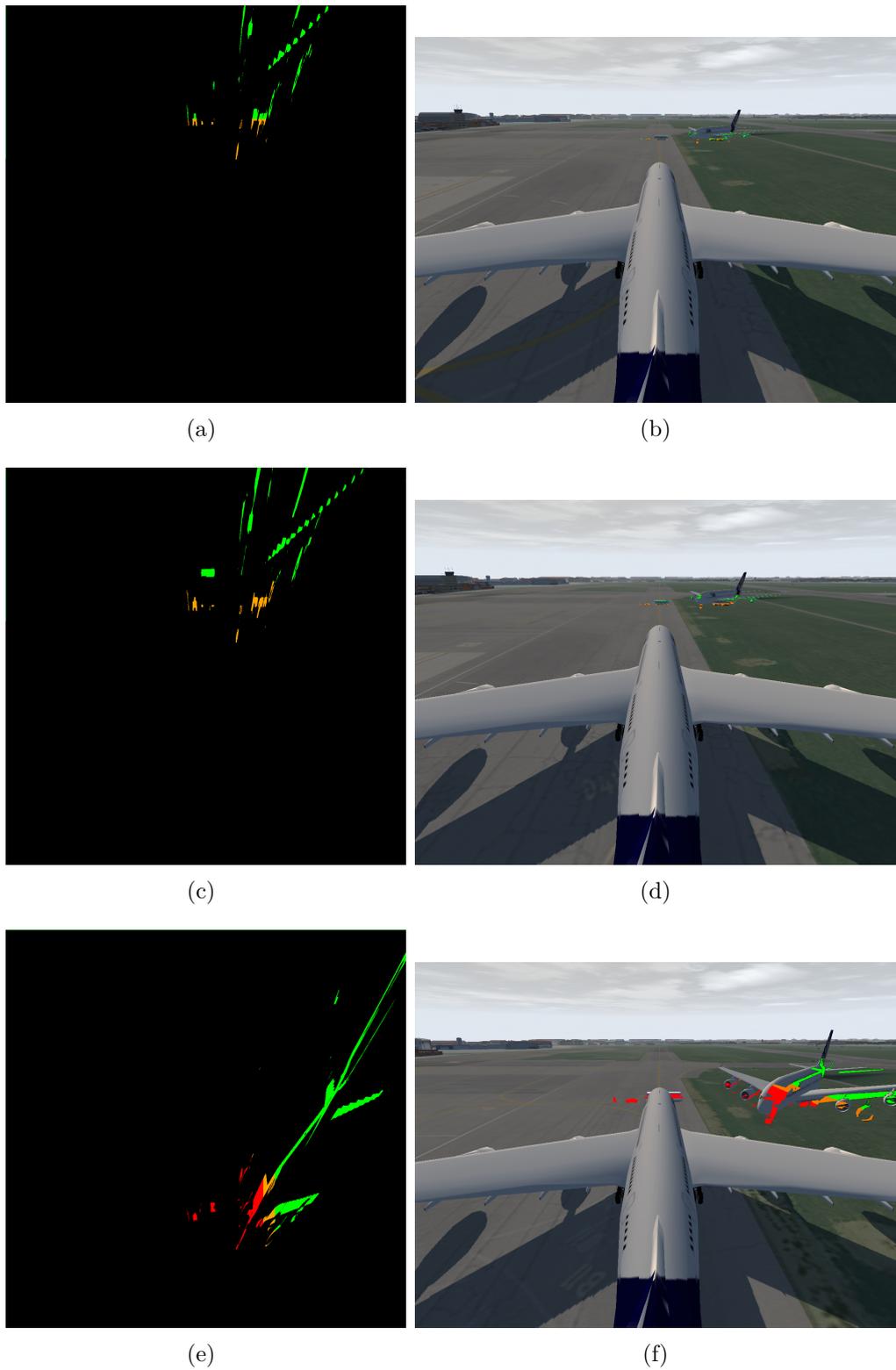


Figure 5.8: (a-c-f) Colorisation de plusieurs cartes d'obstacles et (b-d-f) résultats de colorisation de l'image pour plusieurs estimations de déplacement de l'aéronef entre les instants  $t - 1$  et  $t$ .

## 5.6 Utilisation de la transformation IPM multi-plans pour raffiner la détection

Pour pouvoir appliquer la transformation IPM aux images, on fait l'hypothèse d'un sol plat, ce qui permet de rendre inversible la projection. Or, dans le domaine de l'automobile comme dans les zones aéroportuaires, l'hypothèse de sol plat n'est pas strictement respectée et cela entraîne des distorsions dans le résultat de la transformation IPM.

Nous pouvons limiter ces erreurs en combinant les projections inverses sur des plans situés à plusieurs hauteurs. C'est le but d'une méthode IPM multi-plans. Dans les articles [41] et [90], les auteurs proposent d'utiliser une approche qui consiste à réduire ces distorsions en projetant l'image provenant de la caméra sur plusieurs plans situés à des distances variées au dessus du sol et en combinant ces informations.

Nous nous sommes intéressés à cette approche pour deux raisons. Dans un premier temps, nous possédons une incertitude sur la hauteur de la caméra et sur l'hypothèse de sol plat. Utiliser la transformation IPM sur plusieurs plans et fusionner les résultats obtenus pourrait permettre d'améliorer la détection de la position exacte de l'obstacle. Dans un second temps, l'aéronef est composé de plusieurs parties pouvant être considérées comme des obstacles, dont certains ne sont pas en contact avec le sol (ailes, réacteurs, nez de l'avion...). L'utilisation de la transformation IPM multi-plans pourrait être utilisée pour séparer les obstacles en contact avec le sol de ces derniers et affiner la détection pour ces deux cas. La Figure 5.9 illustre la transformation IPM multi-plans.

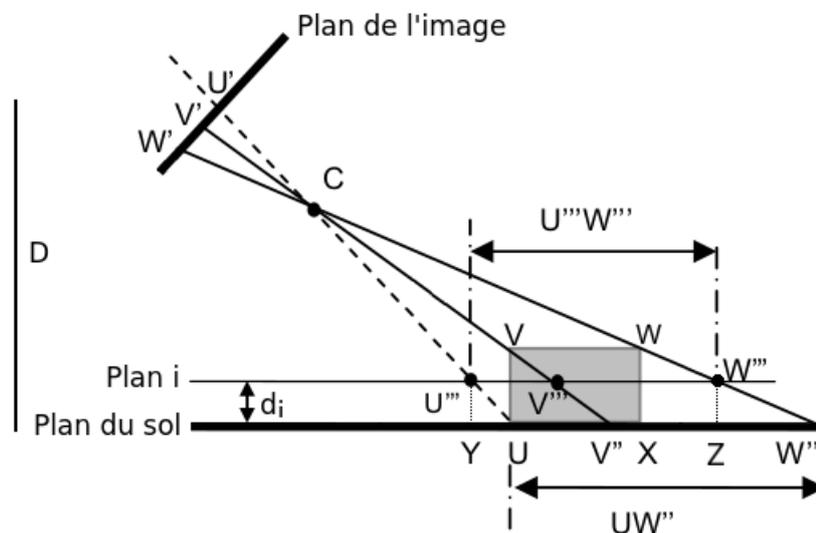


Figure 5.9: Illustration de la méthode multi-plans (inspirée de [90]).

On considère un obstacle dont les sommets sont représentés par  $U$ ,  $V$ ,  $W$  et  $X$  et une caméra placée à une distance  $D$  du sol. En utilisant la transformation IPM par rapport au sol, ces points sont projetés en  $U$ ,  $V''$ ,  $W''$  et  $X$ . La projection de l'obstacle  $UVWX$  dans le plan du sol est donc représentée par le segment  $UW''$ . On remarque qu'il existe une erreur  $XW''$  de projection dans le plan du sol. Cette distorsion dépend de la hauteur de l'obstacle et de sa position dans le champ de vue de la caméra.

Dans un second temps, on définit un plan  $d_i$ , se situant à une hauteur  $d_i$  du sol. On remarque que ces mêmes points sont alors projetés en  $U'''$ ,  $V'''$  et  $W'''$  (pour une question de lisibilité, nous ne représentons pas  $X'''$  sur le schéma mais il se trouve entre  $V'''$  et  $V''$  sur le plan  $i$ ). L'erreur de distorsion pour la projection du point  $W$  dans le plan du sol est alors réduite à  $Z$ , représentant  $W'''$  dans le plan du sol.

On introduit cependant une distorsion de la projection du point  $U$  en  $U'''$ . De la même façon, on note  $Y$  la représentation de  $U'''$  dans le plan du sol). La nouvelle représentation de l'obstacle est donc le segment  $U'''W'''$  dans le plan  $i$  ou encore  $YZ$  dans le plan du sol.

En combinant les informations obtenues par les deux plans, on peut ré-estimer la projection de l'obstacle  $UVWX$  sur le plan du sol par le segment  $UZ$  puisque l'on considère que l'obstacle est placé sur le plan du sol, ce qui permet de ne pas prendre en compte la distorsion représentée par le segment  $YU$ .

Nous reprenons les notations de la section ci-dessus pour appliquer la transformation IPM multi-plans à notre problème. L'idée est donc d'exprimer la carte du sol au temps  $t - 1$  à l'instant  $t$  et de les soustraire, tout en prenant en compte les différents plans  $n$  que l'on forme des deux cartes du sol. De manière analogue, on travaille seulement avec les images pour réduire le bruit dû aux interpolations.

Pour rappel, pour chaque image, nous obtenons une cartes d'obstacles en nuances de gris qui correspond à la probabilité de présence d'un obstacle et à laquelle nous appliquons divers traitements (filtrage des outliers, binarisation...). En appliquant la transformation IPM multi-plans à chaque image, on obtient des cartes d'obstacles pour des hauteurs de caméra différentes.

On peut dans un premier temps définir une nouvelle carte d'obstacles comme une moyenne des cartes d'obstacles à différentes hauteurs dans le but de raffiner la précision de la position de l'obstacle, comme proposé dans [90] et illustré par l'Équation 5.14, où l'on indice par  $i$  les cartes d'obstacles créées pour différents plans. On peut aussi utiliser d'autres méthodes comme par exemple en additionnant les valeurs des cartes d'obstacles obtenues, en donnant un degré d'importance à l'une ou l'autre des cartes ou encore en ne s'intéressant qu'à l'intersection valeurs des cartes d'obstacles.

$$Obst_{global} = \frac{1}{n} \sum_{i=0}^n (Obst_i) \quad (5.14)$$

Si l'on prend l'exemple pour deux plans, on peut définir deux cartes d'obstacles  $Obst_i$  et  $Obst_j$  qui représentent les cartes d'obstacles pour deux différences de hauteur de la caméra. On note  $IM_{(t-1)/t}^i$  et  $IM_{(t-1)/t}^j$  les images obtenues en exprimant  $IM_{t-1}$  à l'instant  $t$  pour les plans  $i$  et  $j$ . Ces dernières sont exploitées en nuance de gris. Les Équations 5.16 et 5.17 détaillent les composants  $Obst_i$  et  $Obst_j$  de l'Équation 5.15.

En reprenant l'Équation 5.14, on explicite dans l'Équation 5.18 les termes dépendant des hauteurs choisies pour les différentes hauteurs de caméra que l'on regroupe dans la variable  $d = [d_0 \dots d_n]$  dans l'Équation 5.19 où  $D$  est la hauteur théorique de la caméra par rapport au plan du sol et  $d_i$  est la hauteur du plan  $i$  par rapport au plan du sol.

$$Obst_i = (IM_t - IM_{(t-1)/t}^i) \quad \text{et} \quad Obst_j = (IM_t - IM_{(t-1)/t}^j) \quad (5.15)$$

$$\lambda \cdot \begin{bmatrix} u_{obst_i} \\ v_{obst_i} \\ 1 \end{bmatrix} = \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} - H_i^{-1} \cdot M^{-1} \cdot H_i \cdot \begin{bmatrix} u_{(t-1)} \\ v_{(t-1)} \\ 1 \end{bmatrix} \quad (5.16)$$

$$\lambda \cdot \begin{bmatrix} u_{obst_j} \\ v_{obst_j} \\ 1 \end{bmatrix} = \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} - H_j^{-1} \cdot M^{-1} \cdot H_j \cdot \begin{bmatrix} u_{(t-1)} \\ v_{(t-1)} \\ 1 \end{bmatrix} \quad (5.17)$$

$$2 \cdot \lambda \cdot \begin{bmatrix} u_{obst} \\ v_{obst} \\ 1 \end{bmatrix} = 2 \cdot \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} - [H_j^{-1} \cdot M^{-1} \cdot H_j + H_i^{-1} \cdot M^{-1} \cdot H_i] \cdot \begin{bmatrix} u_{(t-1)} \\ v_{(t-1)} \\ 1 \end{bmatrix} \quad (5.18)$$

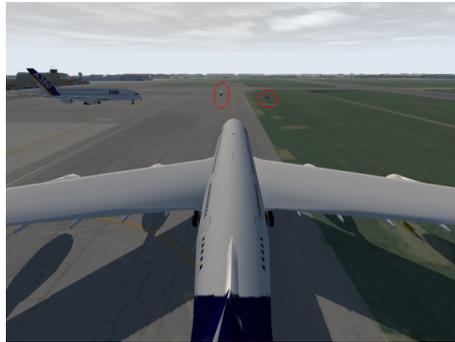
$$\text{Avec pour rappel } H_i = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin(\theta) & (D - d_i) \cdot \cos(\theta) \\ 0 & -\cos(\theta) & (D - d_i) \cdot \sin(\theta) \end{bmatrix} \quad (5.19)$$

Cette expression de l'image augmentée  $Obst_{global}$  nous permet de réduire le temps de calcul et la complexité apportée par l'utilisation de plusieurs plans de transformations IPM. Cependant, pour capitaliser sur la précision qu'apporte cette méthode, il est nécessaire d'effectuer les transformations IPM sur des plans très rapprochés (de l'ordre de quelques centimètres de différence de hauteur).

La Figure 5.10 présente un exemple de deux cartes d'obstacles dont l'une est calculée dans le plan du sol et l'autre dans un plan virtuel situé à 25 cm du sol. Les Figures 5.10(e) et 5.10(d) montrent des zooms des deux cartes d'obstacles. Les pixels gris et blancs définissent les obstacles. Sur chacune des figures, nous avons entouré en rouge les pixels correspondant au bus et la voiture présents sur la Figure 5.10(a). Les autres pixels blancs (ou gris) correspondent à l'avion présent au fond de l'image. On peut remarquer que la taille des obstacle est réduite sur la Figure 5.10(d), notamment au niveau du bus où la déformation est représentée par moins de pixels, ce qui signifie une réduction de l'erreur  $E_{ARRIÈRE}$ . D'autre part, les obstacles de cette scène se trouvant assez loin de l'avion et le déplacement en translation étant faible, l'erreur  $U'''U$  décrite en Figure 5.9 n'est presque pas représentée dans la carte d'obstacles.

La Figure 5.11 présente les cartes d'obstacles globales créées pour six plans dont la distance maximale est de 25cm (Figures 5.11(c) et 5.11(d)) ou de 250cm (Figures 5.11(a) et 5.11(b)) auxquelles nous avons appliqué un traitement d'augmentation du contraste, par soucis de lisibilité. Les figures de la première colonne représentent une fusion par intersection des cartes d'obstacles créées pour les six plans alors que celle de la seconde colonne représentent une fusion par moyennage.

On constate qu'une différence entre les plans de 250cm ne permet pas d'obtenir un résultat satisfaisant. En effet, si les plans sont trop éloignés les uns des autres, les résultats des cartes d'obstacles ne possèdent pas forcément de zone d'intersection. Auquel cas, on ne peut pas raffiner la position de l'objet. Par exemple, la Figure 5.11(c), qui représente l'intersection des plans, est presque vide (sauf morceau de l'avion dans le coin en haut à gauche de l'image). On voit apparaître les informations des différents plans en Figure 5.11(d), où l'on a une duplication de l'erreur de déformation des obstacles. Par exemple, pour le bus, on



(a)



(b)



(c)



(d)



(e)

Figure 5.10: (a) Image à l'instant  $t$ . Exemple de deux cartes d'obstacles : (b) sur le plan du sol et (c) sur un plan virtuel situé à 25cm du sol. (d) et (e) sont des zooms de (b) et (c).

peut visualiser des déformations distinctes, de plus en plus grandes, en terme de pixels.

La Figure 5.11(f) montre une projection de la colorisation des obstacles dans l'image. La voiture étant trop peu représentée par des pixels dans l'image, sa déformation est quasi-inexistante. On retrouve par contre une déformation du bus ainsi que de l'aéronef où l'on note notamment les erreurs  $E_{AVANT}$  présentes au niveau des réacteurs et du cockpit de l'aéronef.

Dans le domaine automobile, cette méthode peut être intéressante car le front d'une voiture ne dépasse pas le mètre. Pour un aéronef, les différences de hauteur entre les trains atterrissage, le nez de l'aéronef, les ailes et le haut de la dérive sont trop importantes. Pour tirer le meilleur partie de cette technique, il faudrait effectuer la transformation sur de nombreux plans et cela pose problème en terme de temps de calcul.

De plus, il nous reste à travailler sur un algorithme permettant d'occulter tout ce qui se trouve en dessous ou au dessus des plans étudiés. Nous avons présenté une carte d'obstacles qui moyenne les résultats des différents plans mais nous pourrions évaluer d'autres combinaisons des informations apportées par les différents plans (somme, pondération par importance, intersection...).

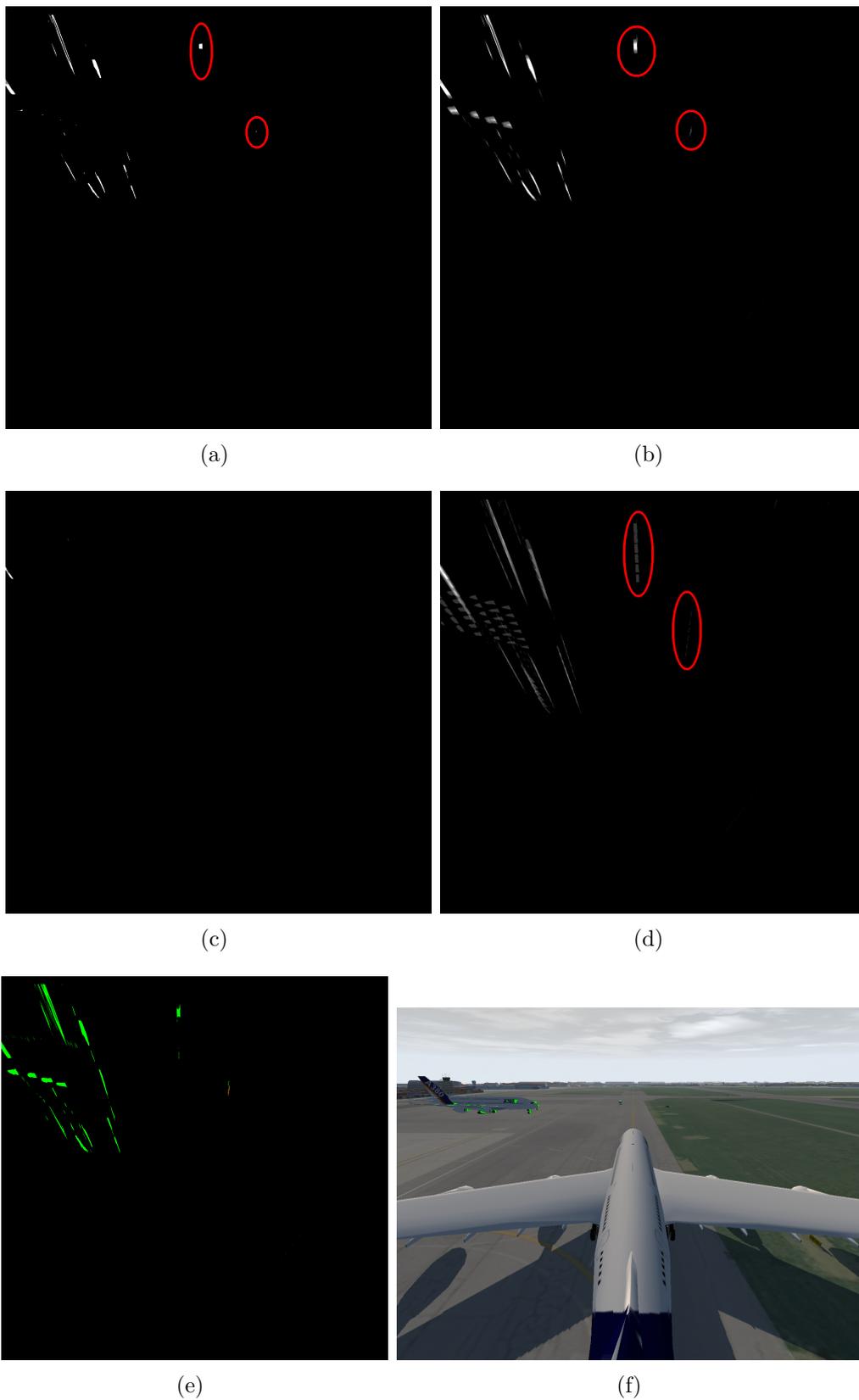


Figure 5.11: Zoom sur les cartes d'obstacle résultant de la fusion d'information de six cartes dont l'espacement global est (a-b) de 25cm (a) en se basant sur l'intersection et (b) en se basant sur la moyenne des cartes, (c-d) de 250cm (c) en se basant sur l'intersection et (d) en se basant sur la moyenne des cartes. On présente le résultats de coloration (e) de la carte d'obstacles dont on a tiré (a) et (f) la projection sur l'image.

## 5.7 Conclusion

Dans ce chapitre, nous avons présenté la seconde brique algorithmique nécessaire pour l'aide à la navigation de véhicules : la détection d'obstacles. Nous avons présenté les méthodes les plus récurrentes dans le domaine automobile. Parmi elles, on peut faire une distinction entre les méthodes géométriques et les méthodes basées réseaux de neurones.

Nous avons ensuite proposé de tirer parti de la détection de motifs apportée par le filtre particulaire afin d'estimer le déplacement de l'aéronef d'une image à l'autre. En combinant cette estimation du déplacement à l'utilisation de la transformation IPM, nous avons défini une méthode de création de carte d'obstacles. Cette carte d'obstacles nous permet de générer une image contenant une information de présence et de criticité au pilote de l'aéronef.

Pour l'instant notre algorithme, basé sur un filtre particulaire, dans le cas des marquages au sol et des bordures de tarmac, assimile les motifs à des droites. Dans l'optique de remonter cette information au pilote, il serait intéressant d'envisager d'utiliser des quadriques ou des courbes de Bézier, afin d'améliorer la précision de la détection des courbes. Cependant, nous n'utilisons la détection de ces motifs que pour estimer le déplacement de l'avion. C'est pourquoi l'estimation seule de droites affines suffit au fonctionnement de l'algorithme.

L'estimation de la matrice de déplacement de l'avion pourrait être améliorée en utilisant l'algorithme de Gauss-Newton ou un SVD, pour optimiser l'estimation de  $T_x$ ,  $T_y$  et  $\theta$ . La détection d'obstacles sur la carte d'obstacles se fait à partir des pixels blancs. Il faudrait implémenter un algorithme de clusterisation de ces pixels puisque seul le front de l'objet nous intéresse et donc les premières lignes de pixels blancs. Cela améliorerait la précision de l'estimation de la distance entre l'obstacle et l'avion, ainsi que la propreté du rendu au pilote. Dans [2], l'auteur proposait une approche similaire, pour détecter des obstacles depuis une micro-caméra montée sur un robot en milieu intérieur. Un algorithme y est proposé pour effacer dans la carte d'obstacles, les artefacts créés par IPM derrière un objet plus haut que le sol.

Ce chapitre a un but applicatif et se base sur les résultats de la méthode d'analyse de scène. La solution n'a pas pu être testée sur des images réelles, faute de données. Il serait intéressant de poursuivre cette étude sur de telles données afin de quantifier la précision des algorithmes proposés pour estimer le déplacement de l'avion et pour détecter la présence d'obstacles. Les méthodes de détection d'obstacles basées réseaux de neurones sont régulièrement améliorées. Une comparaison avec ces algorithmes serait alors intéressante après l'obtention de données réelles.

Au cours de cette thèse, nous avons peu abordé le problème des conditions de vision dégradées, qui est très important lors de la détection d'obstacles potentiels. En effet, les effets du brouillard peuvent rendre très difficiles l'appréciation des distances ou même la détection de la présence d'un obstacle. Dans le prochain chapitre, nous nous intéresserons plus particulièrement à ce cas de figure.

# Chapitre 6

## Adaptation aux conditions de vision dégradées par le brouillard

### Sommaire

---

<b>6.1</b>	<b>Introduction</b>	<b>172</b>
<b>6.2</b>	<b>État de l'art des méthodes de débrumage</b>	<b>174</b>
6.2.1	Modèle atmosphérique du brouillard	174
6.2.2	Méthodes de débrumage à base de rehaussement de contraste	175
6.2.3	Méthodes de débrumage basées sur des hypothèses et modèles mathématiques	176
6.2.4	Méthodes de débrumage à base de réseaux de neurones	178
6.2.5	Cas particulier de la polarimétrie	179
<b>6.3</b>	<b>Application d'algorithmes de débrumage au contexte aéronautique</b>	<b>180</b>
6.3.1	Retour sur les algorithmes testés	181
6.3.2	Résultats sur la qualité d'image	181
6.3.3	Application à la tâche de détection des lignes	183
6.3.4	Conclusion	184
<b>6.4</b>	<b>Prise en compte de l'infrarouge</b>	<b>185</b>
6.4.1	Méthodes de fusion de données	185
6.4.2	Application de la fusion de données au débrumage	192
<b>6.5</b>	<b>Proposition d'amélioration de méthodes</b>	<b>194</b>
6.5.1	Utilisation de l'IPM pour améliorer la carte de profondeur	194
6.5.2	Place de la fusion de données dans les approches basées réseau de neurones	194
6.5.3	Utilisation de l'algorithme Multi-Scale RetineX	195
6.5.4	Estimation de l'apport de l'infrarouge	195
<b>6.6</b>	<b>Conclusion</b>	<b>196</b>

---

## 6.1 Introduction

Comme tout traitement d'image en extérieur, notre système est très sensible aux variations des conditions météorologiques qui dégradent les résultats (baisse de luminosité et impact sur les couleurs de nuit, déformations dues aux gouttes de pluie, baisse de visibilité due au brouillard...).

Dans cette partie, nous comparons des méthodes qui permettent d'améliorer la qualité de l'image obtenue par temps de brouillard et nous analysons leurs capacités à limiter l'impact du brouillard sur la détection des marquages. Nous interrogeons aussi l'utilisation de données provenant de caméras infrarouges dans la capacité à améliorer la qualité de l'image obtenue pour un retour visuel au pilote de l'aéronef. La Figure 6.1 présente différents types d'images impactées par les effets du brouillard (réel ou simulé).

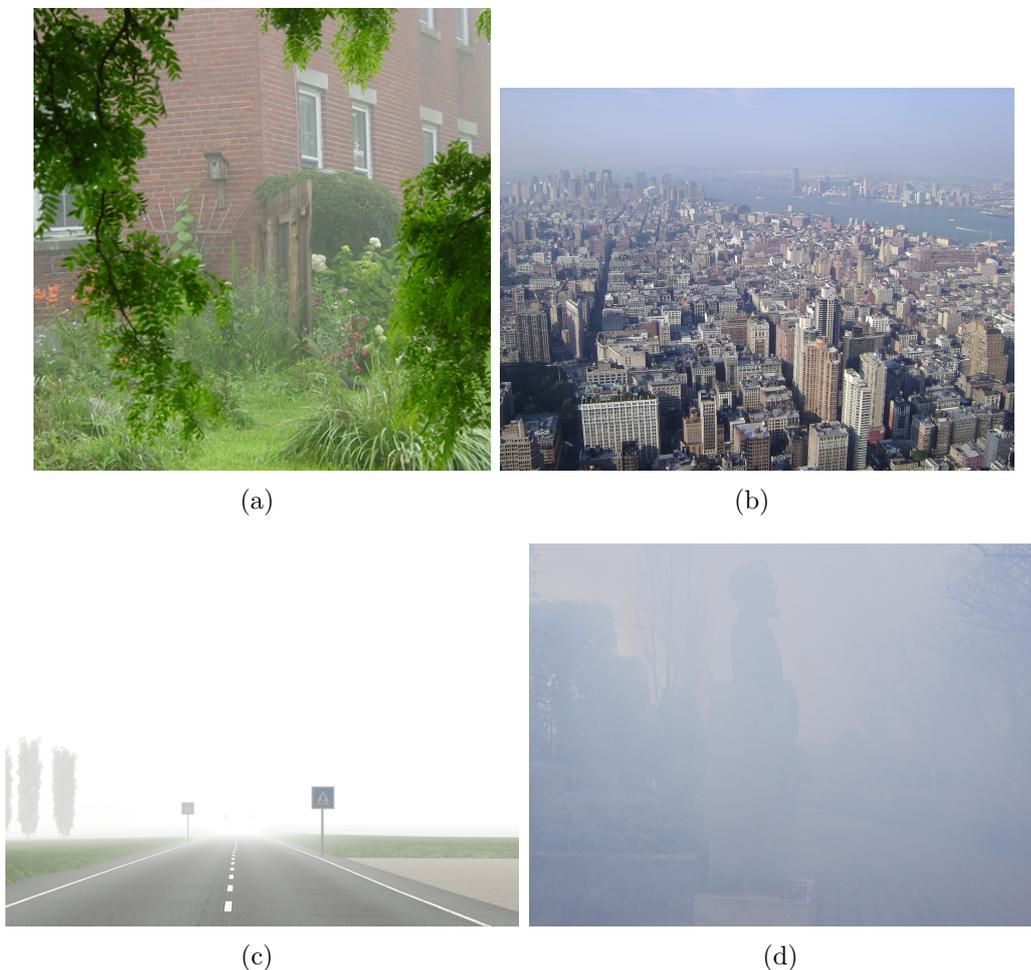


Figure 6.1: (a) Impact modéré du brouillard (image provenant de [46]), (b) Faible impact du brouillard, (c) Impact simulé du brouillard (image provenant du dataset FROSI<sup>1</sup>) et (d) Fort impact du brouillard (image provenant du dataset Dense Haze du challenge NTIRE19<sup>2</sup>).

<sup>1</sup><https://www.livic.ifsttar.fr/linstitut/cosys/laboratoires/livic-ifsttar/logiciels/bases-de-donnees/frosi/>

<sup>2</sup><https://data.vision.ee.ethz.ch/cvl/ntire19/>

Dans la première partie de ce chapitre, nous proposons une revue de différentes méthodes de débrumage proposées dans la littérature. Nous reviendrons sur les méthodes basées sur des modèles mathématiques de l'impact du brouillard sur l'acquisition de données des capteurs ainsi que des méthodes de débrumage basées réseaux de neurones présentées ces dernières années pour répondre à la problématique de la baisse de visibilité due au brouillard.

Une comparaison de différentes méthodes appliquées au domaine de l'aéronautique a été présentée dans une publication courte acceptée à RFIAP en 2020 [117]. Plusieurs méthodes sont comparées par rapport à l'amélioration de la qualité de l'image ainsi qu'à celle de la détection de marquages.

Dans le cadre de cette thèse, nous nous intéressons à deux types de capteurs : les caméras visibles ainsi que les caméras infrarouges. Les caméras infrarouges nous permettent d'accéder à une image de meilleure qualité de nuit ainsi que pendant des épisodes de conditions météorologiques dégradées. Dans ce chapitre, nous étudions donc la possibilité de coupler des informations de la caméra dans le domaine du visible avec celles de la caméra dans l'infrarouge lointain afin d'améliorer les tâches de détection de lignes et de détection d'obstacles. Nous revenons dans un premier temps sur les méthodes de fusion entre les données de caméras visibles et infrarouge puis sur les études proposant l'utilisation de l'infrarouge dans le débrumage avant de présenter des pistes d'utilisation de la fusion visible/infrarouge.

## 6.2 État de l'art des méthodes de débrumage

Il est possible de séparer les méthodes de débrumage en deux catégories : celles qui utilisent plusieurs images d'une même scène, observée sous différentes longueurs d'ondes ou différents degrés de polarisation et celles qui ne se basent que sur une seule image de la scène. Nos travaux se placent dans cette deuxième catégorie puisque les caméras choisies sont embarquées sur un aéronef en mouvement et n'offrent pas plusieurs interprétations d'une acquisition à un instant  $t$ .

Nous pouvons distinguer plusieurs branches de recherches. Tout d'abord, on peut citer les méthodes basées sur des techniques de filtrage, type égalisation d'histogrammes [8][62], filtre bilatéral [180] ou filtre médian [157]. Une autre partie des méthodes proposées se concentrent sur la capacité à approcher le modèle de Koschmieder, ou modèle Airlight, pour essayer de reconstruire l'effet du brouillard et le soustraire à l'image [46]. Enfin la question du débrumage est aussi traitée par les méthodes à base de réseaux de neurones comme [95] ou [44].

### 6.2.1 Modèle atmosphérique du brouillard

La lumière environnante se diffuse dans l'atmosphère à la rencontre de particules (poussières, gouttelettes d'eau...). L'atmosphère peut alors être considérée comme une source de lumière. Ce phénomène s'appelle le voile atmosphérique ou "Airlight". Il a été exprimé mathématiquement en 1924 par le modèle de Koschmieder, décrit par les Équations 6.1 à 6.3 où :

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (6.1)$$

$$t(x) = e^{-\beta d(x)} \quad (6.2)$$

$$\beta \propto \frac{1}{\lambda^\gamma} \quad (6.3)$$

- $I(x)$  est l'image impactée par le brouillard,
- $J(x)$  est l'image non impactée par le brouillard ou radiance de la scène,
- $t(x)$  est la carte de transmission,
- $A$  est le voile atmosphérique global pour chaque pixel,
- $\beta$  est le coefficient d'atténuation dans l'atmosphère ou coefficient d'extinction atmosphérique,
- $d(x)$  est la profondeur d'un pixel dans la scène,
- $\lambda$  est la longueur d'onde de la lumière incidente,
- $\gamma$  est un coefficient qui dépend de la taille des particules présentes dans l'environnement

Ce modèle est utilisé par certains algorithmes de débrumage pour travailler aux débrumages d'images acquises par temps de brouillard. Pour reconstruire l'image  $J(x)$ , les différents algorithmes proposés dans la littérature doivent estimer le voile atmosphérique  $A$  et/ou la carte de transmission  $t(x)$ . En posant différentes hypothèses, l'estimation d'un de ces paramètres amène à l'estimation du second et à la reconstruction de l'image  $J(x)$ . La Figure 6.2 illustre l'impact du brouillard sur l'image obtenue.

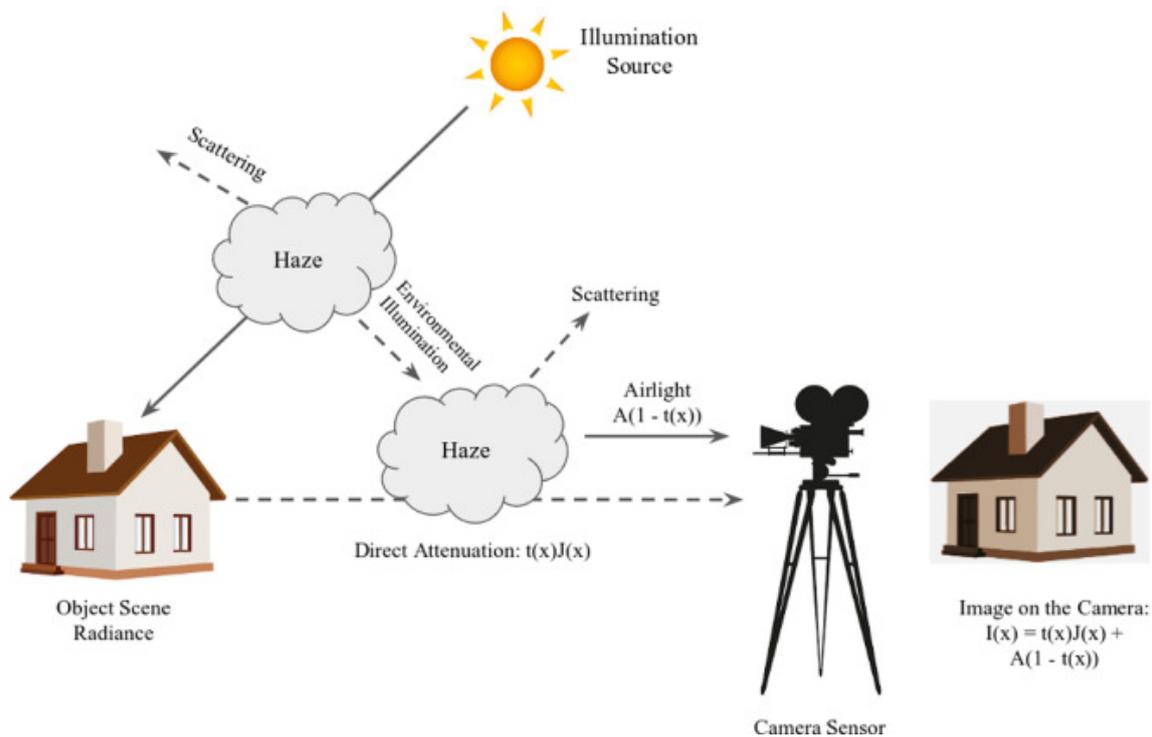


Figure 6.2: Illustration du modèle de Koschmieder (extrait de [16]).

## 6.2.2 Méthodes de débrumage à base de rehaussement de contraste

### 6.2.2.a Égalisation d'histogramme

L'impact du brouillard sur l'information perçue par le capteur pendant l'acquisition de l'image a pour effet d'atténuer fortement les contrastes et la saturation des couleurs des différentes zones de l'image. Comme présenté dans le Chapitre 3, l'égalisation d'histogrammes est un moyen de rehausser les contrastes dans une image. C'est pourquoi ces méthodes ont été proposées dans le but de restaurer des images impactées par le brouillard.

Dans l'article [8], les auteurs proposent d'utiliser l'algorithme CLAHE ainsi qu'une correction gamma adaptative afin de corriger l'effet du brouillard sur l'image capturée. Ils proposent d'effectuer le CLAHE sur l'espace de couleur  $L^*a^*b$  pour travailler notamment sur le canal de luminosité. La correction gamma appliquée à la suite du CLAHE permet de corriger de potentiels artefacts de couleur dus au CLAHE.

Les auteurs de [62] proposent également d'utiliser l'égalisation d'histogramme. Dans l'article, ils présentent des résultats sur des images simulées dans le contexte automobile, avec une modélisation de brouillard homogène et non-homogène. Leur méthode combine une approche basée sur le modèle de Koschmieder et une approche basée traitement de signal et égalisation d'histogramme locale où un algorithme de filtrage gaussien est appliqué après l'égalisation afin de supprimer les effets de blocs générés. L'amélioration de contraste due à l'égalisation permet alors de mieux estimer la carte de profondeur car elle fait apparaître des détails de l'image. La carte de profondeur permet d'estimer l'impact du brouillard sur les différents pixels de l'image et donc d'appliquer plus efficacement le modèle de Koschmieder.

### 6.2.2.b Filtrage

Dans la littérature, on retrouve aussi des méthodes qui étudient l'impact du brouillard sur l'image comme une perturbation bruitée du signal. Dans ce cas-ci, des méthodes de filtrage sont proposées afin de retirer l'effet indu par le brouillard. L'avantage de ces méthodes est qu'elles sont de faible complexité.

Dans [157], les auteurs proposent d'étudier le bruit apporté par le voile atmosphérique plutôt que d'estimer la carte de profondeur de l'image. Une balance des blancs est d'abord effectuée afin d'obtenir une zone uniforme correspondant à la plus forte intensité dans l'image (souvent le ciel) et que l'on considère être la représentation du brouillard. Cela peut être efficace sur un brouillard de simulation, puisque l'on fait l'hypothèse qu'il est blanc, mais peut induire une baisse de performance des algorithmes de débrumage sur des images réelles où le brouillard peut apparaître plus bleuté. L'impact du voile atmosphérique est ensuite estimé en analysant la luminosité de l'image dégradée par le brouillard, puisqu'il est dépendant de l'intensité du ciel (ou zone de plus forte intensité). Les auteurs utilisent ensuite un filtre médian sur l'impact déduit. Cela a pour but de préserver les contours et supprimer le bruit impulsionnel. Pour préserver les coins des contours, une modification du filtre médian est proposé dans cet article.

De la même façon, le filtre médian est aussi utilisé par les auteurs de [180] afin de lisser l'impact du voile atmosphérique. Cette étape est suivie d'une nouvelle étape de filtrage où les auteurs utilisent un filtre bilatéral guidé afin de lisser les zones de même texture et garder un fort contraste entre deux zones de textures différentes. Cela permet une meilleure estimation des zones de profondeur similaires dans l'image et donc une meilleure restauration de l'image.

## 6.2.3 Méthodes de débrumage basées sur des hypothèses et modèles mathématiques

Il existe plusieurs méthodes proposées dans la littérature qui exploitent le modèle de Koschmieder et proposent différentes méthodes pour estimer les paramètres de distance, de transmission ou d'extinction.

### 6.2.3.a Estimation de l'albédo d'une image

L'albédo est le pouvoir réfléchissant d'une surface, comparable à la réflectance. Fattal propose dans [46] une méthode de débrumage à base d'une seule image couleur, en décrivant plusieurs phases d'estimation des paramètres du modèle de Koschmieder en estimant l'albédo au sein d'une image. Dans un premier temps, l'auteur considère que l'image résultante du débrumage est fonction de la lumière réfléchie  $l(x)$  pour chaque pixel et d'un coefficient de réflectance  $R$  (supposé ou non constant) tel que  $J(x) = l(x)R(x)$ . Lorsque  $R$  est supposé constant, il peut être exprimé en deux composantes, l'une parallèle au vecteur représentant l'impact du voile atmosphérique  $A$  et l'autre  $R'$  orthogonale à celui-ci. En projetant  $I(x)$  en fonction de  $A$  et de  $R'$ , on peut réécrire le paramètre de transmission  $t(x)$  en fonction de ces projections et d'une constante qui peut être estimée en supposant que la covariance entre  $t$  et  $l$  est nulle.

Cette méthode présente cependant des limitations. Le modèle est sensible au bruit présent dans l'image. Ce bruit peut être pris en compte mais il nécessite qu'un

minimum d'information soit présent dans la majorité des pixels de l'image. Pour des cas de forts brouillards, les hypothèses du modèle ne sont plus forcément respectées. Enfin, cette méthode complexe peut prendre plusieurs secondes par image, ce qui la rend peut utilisable (sans architecture dédiée pour optimiser les calculs) dans un contexte temps réel.

### 6.2.3.b Hypothèse "Dark Channel Prior"

Cette méthode repose sur l'hypothèse que, pour une image RGB, il existe au moins un canal (R, G ou B) pour lequel des pixels de l'image tendent vers la valeur 0 et, au global, il existe probablement un pixel qui tend vers un pixel de couleur noire. En récupérant la valeur minimale des trois canaux, on devrait tendre vers 0. On considère que la valeur de ce pixel est celle d'une intensité sans brouillard ( $I_0 = 0$ ) et l'on soustrait alors sa valeur aux autres valeurs d'intensité des pixels de l'image pour estimer la transmittance  $t$  du modèle de Koschmieder.

La méthode a été proposée en 2009 dans l'article [68]. Une extension accompagnée d'un filtrage par méthode des moindres carrés pondérés a été proposée dans [7]. Elle peut être mise en comparaison avec des méthodes qui au contraire cherchent à estimer la zone de plus grande intensité de l'image [157]. Cependant, cette hypothèse est basée sur une étude statistique de la présence de pixels sombres dans une image et peut ne pas fonctionner pour certains types d'images où l'environnement est plus clair.

### 6.2.3.c Adaptation à un modèle chromatique

Les auteurs de [156] proposent de réécrire le modèle de Koschmieder comme un modèle chromatique, puisque leur méthode se base fortement sur les informations de couleur et de luminosité présentes dans l'image. Ils séparent donc le problème en deux sous-modèles, celui de la chromaticité de la lumière et celui des objets. Pour améliorer la visibilité des détails d'une image, ils proposent donc d'estimer la chromaticité de la source de lumière de l'image pour pouvoir la retirer. Cette méthode produit de bons résultats pour de faibles cas de brumes mais améliore la visibilité des détails tout en apportant beaucoup de grains aux images impactées par de forts brouillards.

### 6.2.3.d Hypothèse de monde plan

Dans [62], il est proposé d'utiliser une hypothèse de monde plan afin d'aider à l'estimation de la carte de profondeur. C'est une hypothèse que l'on retrouve souvent dans des applications automobile où l'on considère la route comme étant plane et avec peu d'objets en relief. En estimant  $d$  et le coefficient  $\beta$ , il est alors possible d'effectuer un débrumage en inversant le modèle de Koschmieder. La méthode est proposée pour une image en nuance de gris, et l'algorithme est appliqué sur chacun des trois canaux pour une image couleur.

### 6.2.3.e Haze-lines

Dans [14], les auteurs partent du principe que la plupart des images sont formées par un sous-ensemble de couleurs disponibles dans l'espace RGB et répétées dans l'image. Sous l'effet du brouillard, des pixels de même couleur répartis à plusieurs endroits de l'image se retrouvent impactés différemment en fonction de leur distance à la caméra. Les auteurs définissent donc des "lignes" dans l'espace 3 dimensions RGB sur lesquelles la valeur de la couleur se déplace d'un extrême à un autre (couleur non-impactée par le brouillard jusqu'au voile atmosphérique). Ces lignes se rejoignent au point représentant le voile atmosphérique.

Le processus consiste donc à classer chaque pixel comme appartenant à une des couleurs du sous-ensemble de RGB défini et à analyser les différentes teintes au sein de chaque classe afin d'estimer ces "lignes" et la valeur du voile atmosphérique. Une fois le voile atmosphérique estimé, en utilisant le modèle de Koschmieder, on peut calculer la carte de transmission de l'image et donc l'image non impactée par le brouillard. Le problème de cette méthode est qu'elle n'est pas adaptée lorsque le brouillard a un fort impact sur la couleur perçue dans l'image capturée puisque la classification des couleurs est beaucoup plus complexe.

## 6.2.4 Méthodes de débrumage à base de réseaux de neurones

Ces dernières années, de nombreuses méthodes à base de réseaux de neurones ont été proposées, face aux très bons résultats apportés dans les différentes applications de vision par ordinateur. Parmi celles-ci, on peut citer quelques architectures.

Dans [95], les auteurs proposent une méthode de débrumage par un réseau de neurones simple testé sur des images réelles et simulées. Pour cela, ils réécrivent l'image  $J(x)$  non impactée par le brouillard en fonction de l'image bruitée  $I(x)$ , d'un biais  $b$  de valeur 1 par défaut et d'un paramètre  $K(x)$  fonction des variables  $I(x)$ ,  $t(x)$  et  $A$  du modèle de Koschmieder. Une suite de cinq couches convolutives a pour but d'estimer  $K(x)$  pour en déduire  $J(x)$ .

En utilisant une structure plus complexe, les auteurs de [44] présentent une méthode de débrumage à base de GAN (Generative adversarial networks ou réseaux antagonistes génératifs) où le but est de créer deux réseaux générateurs d'images (un réseau pour passer d'une image impactée par le brouillard à une image corrigée et le réseau inverse). Chaque réseau est entraîné séparément grâce à deux réseaux discriminants puis sont mis l'un à la suite de l'autre. Les images d'entrée et de sortie sont comparées et ce résultat est pris en compte dans la fonction de coût. L'idée est de préserver l'information de texture et de structure ainsi que l'information de couleur. Un avantage de ce réseau est qu'il ne nécessite pas de paires d'images bruitées/non bruitées pour son entraînement.

L'article [186] propose une méthode d'estimation de la carte de transmission, du voile atmosphérique ainsi que du débrumage. La carte de transmission est estimée par un réseau basée sur une structure de type encodeur/décodeur aux couches densément connectées où la phase de pooling s'effectue par le biais de pyramides multi-couches. La fonction de coût définie pour ce réseau a pour but de préserver les contours afin d'optimiser l'estimation de la carte de transmission et donc le débrumage. Le voile atmosphérique est estimé par un réseau de type U-net. Ces

deux estimations sont utilisées afin de recréer l'image  $J(x)$  non bruitée. Une phase de vote est implémentée en sortie de cette restauration, par le biais d'un GAN afin de décider si l'image débrumée et la carte de transmission estimée semblent réelles. L'impact du voile atmosphérique est estimé constant sur l'image, ce qui est une hypothèse que l'on retrouve dans de nombreuses méthodes mais peut entraîner une baisse de performance sur certains types d'images.

### 6.2.5 Cas particulier de la polarimétrie

La lumière incidente est caractérisée par une certaine polarisation. L'utilisation de filtres polarisants permet de filtrer certaines directions du champ électrique de l'onde électromagnétique décrivant cette lumière. Ce filtre a pour résultat, en photographie par exemple, de faire disparaître les reflets sur un plan d'eau, d'améliorer les teintes ou encore de rehausser les contrastes d'images.

Cette technique est aussi utilisée pour le débrumage et présente de bons résultats sur l'atténuation du voile atmosphérique. Dans l'article [145], les auteurs proposent d'utiliser deux images acquises avec des filtres polarisants aux directions opposées (un filtre est orienté parallèlement et le second perpendiculairement au plan d'incidence). En se basant sur les informations issues des deux images, les auteurs proposent de découpler la transmission directe de la lumière du modèle Airlight afin de caractériser ce dernier. La variation due à la polarisation est définie comme étant une conséquence directe du voile atmosphérique.

Dans [187], les auteurs proposent une méthode de détection de voies de navigation par fort temps de brouillard en utilisant la polarimétrie comme méthode de débrumage. De la même façon, le voile atmosphérique est estimé grâce à une série d'images capturées par une caméra embarquée sur le véhicule, sous différentes polarisations. Le débrumage est suivi d'un algorithme de Canny afin de faire ressortir les contours dans l'image et une transformée de Hough pour détecter les marquages au sol.

Cette méthode présente de bons résultats lorsque les particules présentes dans l'atmosphère sont de petites tailles. Dans le cas de forts épisodes de brouillard, avec des gouttelettes de tailles importantes, l'impact est beaucoup moins prononcé. De plus, l'acquisition de plusieurs images de la même scène sous différents filtres est souvent requise, ce qui rend difficile l'utilisation dans notre cas d'application.

### 6.3 Application d'algorithmes de débrumage au contexte aéronautique

Pour notre étude, nous utilisons des images provenant d'un simulateur fourni par l'entreprise OKTAL-SE. Ce simulateur nous permet d'obtenir des images d'une même scène selon différentes conditions météorologiques. La Figure 6.3 présente par exemple plusieurs images de la même scène observée sous différents brouillards. Le simulateur nous permet de fixer, pour ceux-ci, la distance maximale de visibilité, permettant de varier l'impact du brouillard sur l'image. Cependant, il ne nous permet pas de simuler différents types de brouillards (brouillard de rayonnement, brouillard d'inversion ou brouillard d'advection), ces derniers ne sont pas différenciés par le simulateur.

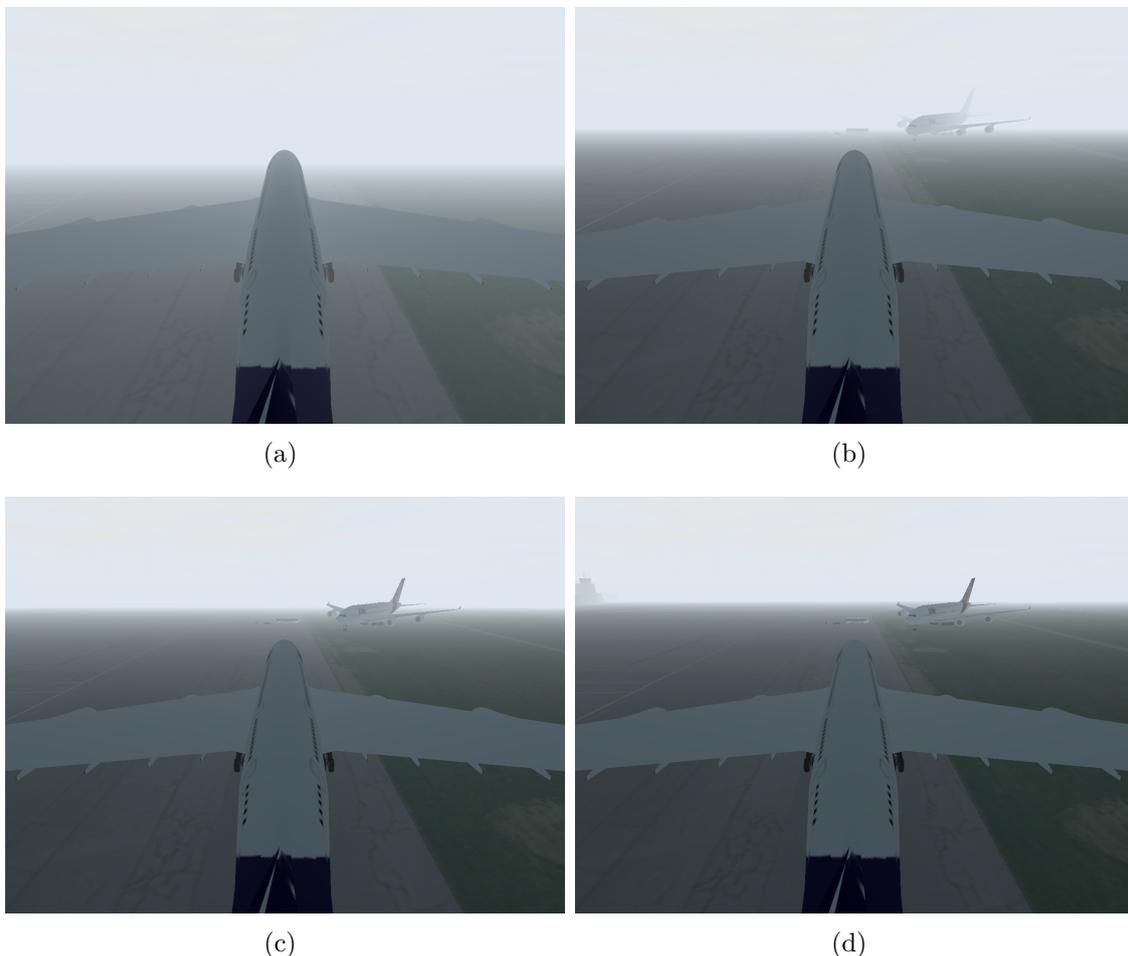


Figure 6.3: Distance maximale de visibilité fixée par le simulateur à (a) 100m, (b) 200m, (c) 300m et (d) 400m.

Dans le cadre du débrumage, nous avons décidé de tester plusieurs méthodes sur nos images et de comparer leurs résultats dans un premier temps en terme d'amélioration du signal puis dans un second temps dans un contexte de détection de lignes.

### 6.3.1 Retour sur les algorithmes testés

Nous comparons différentes méthodes de débrumage. Tout d'abord, nous nous intéressons à l'effet d'un algorithme de type CLAHE (Contrast Limited Adaptive Histogram Equalization) sur des images RGB bruitées, non basé sur l'apprentissage ni sur les modélisations du modèle du brouillard. Cette méthode est utilisée en combinaison avec d'autres traitements qui permettent de garder une fidélité dans les couleurs, comme par exemple dans les articles [62] ou [8]. De plus, nous nous en servons aussi pour le traitement des images provenant de notre capteur infrarouge. Ensuite, nous avons sélectionné une méthode souvent citée à titre comparatif [46], que l'on nomme ci-dessous "Fattal", du nom de son auteur, approche basée modèle et qui est fondée sur une approche basée modèle, qui cherche à déterminer les composantes d'impact colorimétrique du brouillard et de transmittance du modèle du brouillard. Enfin, nous avons choisi deux approches basées réseaux de neurones, d'une part "Aod-Net" [95] et d'autre part "Cycle-Dehaze" [44], qui utilisent respectivement une architecture en couches convolutives et une architecture de type Cycle-Gan.

### 6.3.2 Résultats sur la qualité d'image

Pour comparer les méthodes de débrumage hors contexte applicatif, nous avons décidé d'utiliser deux mesures quantitatives, exploitées pour évaluer les algorithmes de compression en comparant l'image compressée avec l'image originale :

1. Le Peak Signal to Noise Ratio (PSNR). Ce critère permet d'analyser la qualité de reconstruction de l'image par une analyse locale de chaque pixel.
2. La mesure de similarité entre deux images (SSIM). Contrairement au PSNR et SNR, le SSIM mesure la qualité visuelle de l'image "débrumée" par une méthode donnée, par rapport à l'image simulée sans brouillard.

Les résultats quantitatifs, calculés sur une image simulée dégradée par le brouillard sont apportés par le Tableau 6.1. Les résultats visuels de débrumage des différentes méthodes sont donnés en Figure 6.4.

Méthode	PSNR (dB)				SSIM			
	Overall	0-75	75-250	250+	Overall	0-75	75-250	250+
Image avec brouillard	14.7	19.06	13.8	11.28	0.69	0.71	0.46	0.78
Image après CLAHE	10.44	12.27	14.65	7.34	0.33	0.36	0.67	0.11
Image après Fattal	13.11	14.67	11.99	11.41	0.57	0.55	0.32	0.72
Image après Aod-Net	12.04	11.23	15.18	12.76	0.41	0.32	0.32	0.65
Image après Cycle-Dehaze	14.45	16.79	20.16	10.93	0.61	0.56	0.67	0.68

Table 6.1: Résultats de l'amélioration de l'image par débrumage.

Les critères sont calculées sur toute l'image d'une part, ainsi que par tranches de l'image, du bas vers le haut, ce qui correspond, avec une hypothèse de terrain plat, à une distance approximative à la caméra. L'image fait 1280px960p et les trois tranches ont ainsi été définies : la tranche "0-75m" qui correspond au bas de l'image, lignes 960 à 553, la tranche "75-250m" qui correspond au milieu de l'image, lignes 552 à 266, et enfin la tranche "250m et plus" qui correspond au haut de l'image,

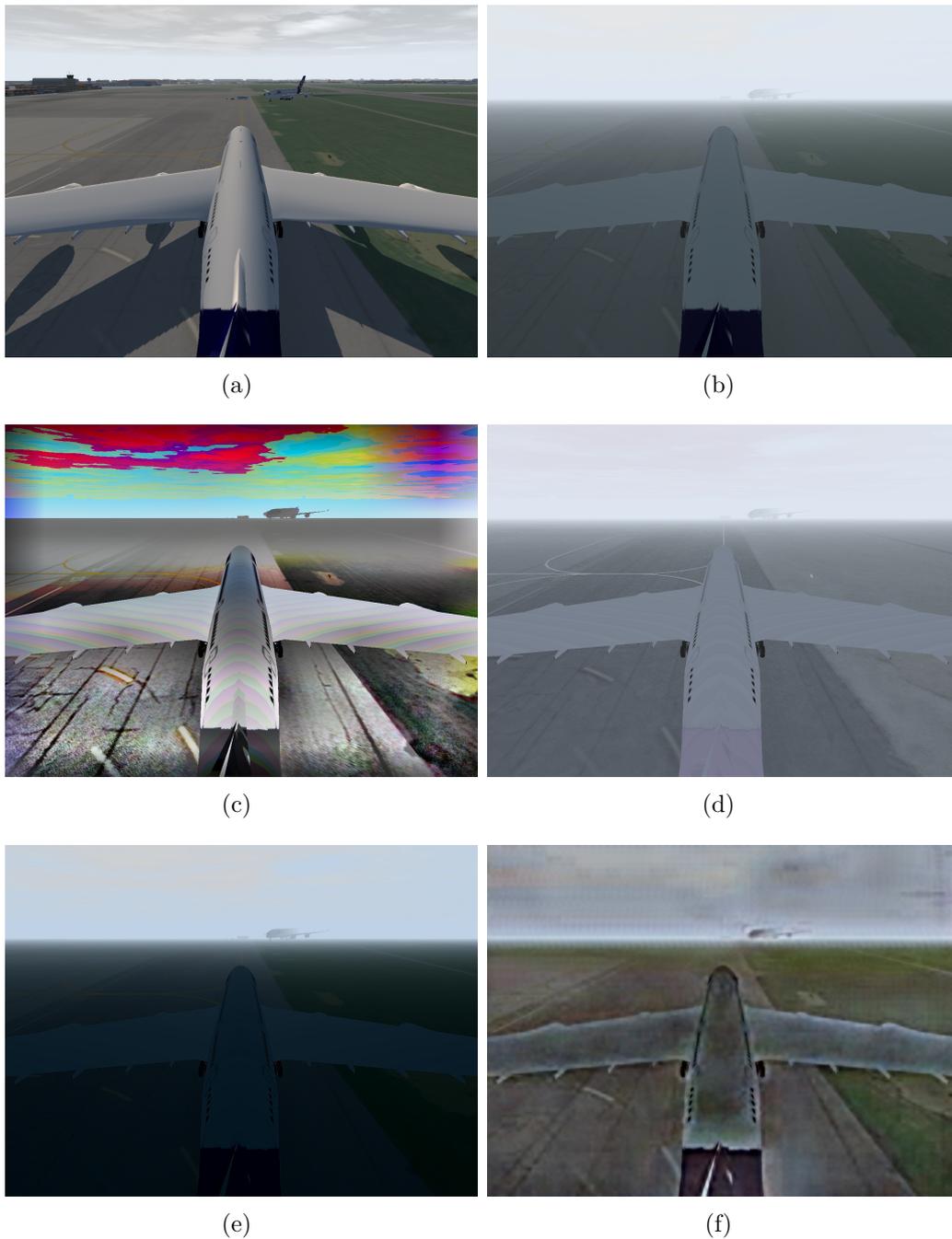


Figure 6.4: (a) Référence, (b) Image avec brouillard, (c) Image après CLAHE, (d) Image après Fattal, (e) Image après Aod-Net et (f) Image après Cycle-Gan.

lignes 265 à 0.

La dernière zone comporte majoritairement des pixels représentant le ciel. Gris-bleuté dans l'image sans brouillard, l'impact de ce dernier ne modifie pas drastiquement sa couleur. La moyenne du carré des différences entre le pixel de l'image de référence et des images débrumées est donc plus faible que dans les autres zones, cette zone étant plutôt uniforme, d'où les moins bons résultats au niveau du PSNR. Au global, les méthodes de débrumage n'améliorent pas le PSNR dans les tranches hautes et basses de l'image. Dans la tranche médiane par contre, on

observe une amélioration de la plupart des méthodes. Ce résultat est significatif car c'est la tranche qui nous intéresse.

Au niveau du SSIM, nous pouvons observer que les résultats de Fattal et Aod-Net sont sensiblement les mêmes sur la tranche médiane de l'image. Ces deux méthodes ont tendance à homogénéiser la luminance de l'image, ce qui détériore le SSIM puisqu'il est basé sur une analyse de la luminance, du contraste et de la structure de l'image. À l'inverse, les méthodes Cycle-Dehaze et CLAHE font plus ressortir les détails de l'image et obtiennent un meilleur SSIM, bien que le bruit de mise à l'échelle pénalise la première méthode tandis que les artefacts de couleur pénalisent la seconde. En effet, on peut noter le score exceptionnellement bas de la méthode CLAHE dans la zone supérieure de l'image.

Avec les images provenant du simulateur, nous obtenons des résultats moins bons que ceux cités dans les différents articles de référence. Enfin, l'ombre de l'avion présente dans l'image de référence ne peut pas être reconstruite depuis l'image impactée par le brouillard et ne participe pas à la bonne reconstruction de l'image.

D'un point de vue qualitatif pour l'œil humain, potentiellement le pilote de l'aéronef, nous pouvons comparer l'effet de ces traitements grâce à la Figure 6.4. La méthode CLAHE donne de très bons résultats sur des images en nuances de gris. Cependant, lorsqu'elle est utilisée pour des images RGB, la forte proportion de pixels représentant le ciel a tendance à ajouter des artefacts lors du débrumage. L'image résultante de l'application de la méthode basée modèle est impactée par l'a priori d'un effet colorimétrique du brouillard constant alors que l'impact est différent entre les zones de la scène observée. Cela a tendance à homogénéiser les couleurs de l'image débruitée. Le résultat de la méthode Aod-Net effectue un décalage de pics d'histogramme dans les trois canaux R, G et B. Cela a pour conséquence d'assombrir l'image. L'architecture Cycle-Gan offre de bons résultats mais la mise à l'échelle de l'image dans les deux sens par pyramide de Laplace ne suffit pas à réduire le bruit sur la couleur.

### 6.3.3 Application à la tâche de détection des lignes

Notre but étant d'utiliser le débrumage pour améliorer la détection des lignes par un algorithme, nous avons décidé de comparer les méthodes après détection de lignes. Pour cela, nous testons l'approche proposée dans le Chapitre 4, basée sur un filtre particulaire, sur les images obtenues après débrumage (voir Figure 6.4). La comparaison est faite selon plusieurs critères :

1. La moyenne de distance d'un pixel détecté comme appartenant à une ligne par rapport à la vérité terrain ainsi que l'écart type de cette mesure. Cela nous permet de comparer la précision de notre détection.
2. Le rappel, qui se calcule par le nombre de vraies détections positives divisée par la somme des vraies et fausses détections positives.
3. La détection maximale ou portée de la détection. Cette donnée est très intéressante pour nous puisque nous cherchons à avoir une détection la plus lointaine possible pour anticiper les décisions. On note, dans le Tableau 6.2, *FP* le résultat du filtre particulaire et *GT* la vérité terrain.

Méthode	Dist. Moy.	Ecart Type Dist.	Rappel	Détection Max.	
				FP	GT
Image sans brouillard	5,87	30.57	0.81	742	742
Image avec brouillard	0	0	0	0	742
Image après CLAHE	69.5	94.31	0.08	616	742
Image après Fattal	0	0	0	0	742
Image après Aod-Net	268.2	184.57	0.1	587	742
Image après Cycle-Dehaze	78.6	77.31	0.25	554	742

Table 6.2: Résultats de la détection de lignes.

Pour les lignes 2 et 4 du Tableau 6.2, les résultats des critères sont nuls puisque la méthode basée sur le filtre particulaire n'a détecté aucune lignes dans l'image. La plupart des méthodes de détection de lignes sont basées sur des modèles de la structure de la ligne, de sa couleur, de son contraste avec son entourage. Le bruit apporté par le brouillard a tendance à atténuer ces contrastes et les différentes méthodes de débrumage présentent des difficultés à reconstruire les contrastes et couleurs de l'image de référence. Dans notre cas, nous cherchons à détecter les marquages jaunes, porteurs d'informations différentes des marquages blancs. Cela explique les mauvais résultats de l'algorithme de détection de lignes, notamment sur l'image provenant de la méthode Fattal qui fait bien ressortir les lignes, à peine visibles dans l'image avec brouillard, mais pas leur couleur d'origine.

### 6.3.4 Conclusion

Les méthodes CLAHE et Cycle-Dehaze se détachent de cette étude. La version implémentée de Fattal se base sur l'hypothèse d'un impact colorimétrique du brouillard constant, ce que nous prévoyons de modifier comme proposé dans [46]. Les méthodes basées sur des réseaux de neurones ont été testées post-entraînement. Cependant, en entraînant ces réseaux sur des images spécifiques, nous pourrions améliorer les résultats. Il faudrait dans ce cas faire attention à ne pas introduire un biais en augmentant la base d'entraînement avec des images du simulateur. La méthode Cycle-Dehaze n'ayant pas forcément besoin de paires d'images pour l'entraînement, c'est un atout pour l'ajout à la base d'entraînement d'images acquises en conditions réelles pour lesquelles on ne disposera pas forcément de la vérité terrain.

Il faut noter que ces résultats préliminaires pourraient varier si l'on applique ces méthodes à des images acquises en conditions réelles. Par exemple, la méthode CLAHE est fortement impactée par le fait que les contrastes et contours sont beaucoup plus nombreux dans ce cas-là. Malheureusement, nous ne disposons pas pour l'instant de données acquises en conditions réelles par temps de brouillard. Il est envisagé, au cours de la thèse de E. Perrotin, d'obtenir ce genre de données.

## 6.4 Prise en compte de l'infrarouge

Comme une des modalités sensorielles étudiées est un boîtier intégrant une caméra LWIR et une caméra RGB, nous pouvons travailler sur de la fusion d'informations. Pour cela, nous présentons quelques méthodes de fusion d'informations proposées dans la littérature avant d'en appliquer certaines à notre cas d'application. Nous proposons aussi de réfléchir sur l'utilisation de l'infrarouge pour améliorer des méthodes de débrumage existantes.

### 6.4.1 Méthodes de fusion de données

Il existe plusieurs méthodes de fusion de données dans la littérature. Certaines se basent sur une fusion en se basant sur des propriétés statistiques alors que d'autres effectuent la fusion au niveau fréquentiel. Enfin, les méthodes de réseaux de neurones ont amené à plusieurs types de fusion ces dernières années. Les auteurs de [109] proposent une étude détaillée des techniques de fusion d'images visibles et infrarouges dont nous présentons quelques cas ci-dessous.

#### 6.4.1.a Transfert statistique de couleurs issues d'une image cible

Cette catégorie de techniques analyse la distribution de couleurs d'une image "cible" (aussi appelée image de référence) et transfère les propriétés statistiques (moyenne et écart-type) de cette distribution à une image "source" (donnée d'entrée de l'algorithme) dans un espace de couleurs intermédiaire comme YUV ou L-alpha-beta. L'image "cible" est donc sensée être proche de la scène observée (image "source"). Il existe de très nombreuses techniques visant à produire ce résultat.

Dans [163], [162] et [182] par exemple, les auteurs ont pour objectif de produire à partir d'une paire d'images visible/infrarouge nocturnes une image fusionnée proche d'une scène de jour en couleurs. Ils prennent pour image "source" une image dite en fausses couleurs, dont les 3 canaux sont une combinaison des niveaux de gris de l'image du spectre visible et de l'image infrarouge. L'utilisation des niveaux de gris de l'image du spectre visible au lieu de ses valeurs RGB est due au fait que le but de ces travaux est de fusionner des images visible et infrarouge prises pendant la nuit, où il y a peu d'informations de chromaticité. Les canaux R, G et B contiennent alors, selon ces travaux, les intensités en niveaux de gris des radiations du spectre visible, des radiations d'infrarouge proche ou des radiations infrarouges d'une caméra thermique.

Ce transfert de couleurs est qualifié de "global" car il calcule les statistiques de l'image en utilisant tous les pixels de celle-ci. L'algorithme est simple et peu coûteux en mémoire et en temps de calcul, mais il donne peu de contrôle sur le rendu final et exige qu'il y ait assez de similarités entre la composition de l'image source et la composition de l'image cible. En effet, si l'image "cible" et l'image "source" contiennent chacune des régions de teintes différentes, cela ne sera pas respecté dans le résultat final et par conséquent les couleurs réelles de la scène peuvent être lourdement modifiées. Ainsi, une LED froide pourrait ne pas apparaître dans sa couleur originelle car sa couleur sur l'image résultante sera corrélée avec sa température.

Conscients de ces inconvénients, d'autres travaux [183][135] ont proposé un transfert statistique "local" de couleurs. Dans [183] par exemple, les auteurs

transfèrent les statistiques d'une région de l'image "cible" à une région analogue de l'image "source". L'image "source" et l'image "cible" sont segmentées en utilisant un algorithme de clustering par mélange de gaussiennes puis les clusters de l'image "source" sont mis en correspondance avec ceux de l'image "cible". Pour que les transitions entre les régions de l'image "source" après le transfert de couleurs ne soient pas abruptes, les pixels situés à proximité des frontières sont colorisés en utilisant une somme pondérée par leurs probabilités d'appartenance aux régions en question.

Le transfert de couleurs utilisant la méthode de regroupement par modèle de mélange gaussien, appelée GMM (Gaussian Mixture Model), est plus sensible aux spécificités locales de l'image "source". Elle donne ainsi de meilleurs résultats, mais il exige que le nombre de clusters soit précisé au début, et ne s'adapte pas au cas où le nombre de clusters augmente en cours de l'exécution de l'algorithme.

#### 6.4.1.b Fusion des composantes entre images visible et infrarouge dans le domaine fréquentiel

Alors que les techniques décrites plus haut effectuent les transformations dans le domaine spatial, ces méthodes de fusion des composantes proposent d'effectuer les traitements dans le domaine fréquentiel. Ces approches suivent le même schéma général : l'image du spectre visible est convertie en un espace de couleurs séparant luminance et chrominance, le canal de luminance est ensuite fusionné avec l'intensité issue de l'image infrarouge et le résultat de cette fusion est ajouté aux deux canaux de chrominance extraits de l'image du spectre visible pour constituer l'image fusionnée finale, qui sera donc en fausses couleurs.

Pour fusionner le canal de luminance de l'image du spectre visible avec l'image infrarouge, on passe dans le domaine fréquentiel en utilisant une transformée fréquentielle multi-échelles : transformée en ondelettes [127][140], transformée en contourlets non sous-échantillonnés [60][80], ou encore transformée en curvelets [189].

La luminance de l'image fusionnée finale est reconstruite en appliquant la transformée inverse aux coefficients fusionnés. Généralement, la fusion des coefficients de basse fréquence revient à calculer la moyenne des coefficients basse fréquence de l'image infrarouge et du canal de luminance de l'image visible, alors que la fusion des coefficients passe-bande consiste à mesurer l'énergie ou la saillance des voisinages du pixel et de garder le coefficient (IR ou visible) dont le voisinage présente le plus d'énergie ou de saillance.

Suivant la même logique, il est proposé dans [82] une méthode de fusion visible/infrarouge qui consiste en la construction d'une pyramide de contraste pour chacune des deux images, puis en la fusion des coefficients des deux pyramides. Cet algorithme, bien plus avancé et complexe, est basé sur l'apprentissage automatique dans la fusion des coefficients, à l'encontre des modèles présentés plus haut, qui eux font appel à des règles de sélection figées. Il améliore le rendu de l'image fusionnée par une fonction de fitness qui sélectionne les meilleures fusions d'images avant d'itérer. Cette fonction d'évaluation permet d'améliorer automatiquement le choix des coefficients de fusion.

Les méthodes de fusion d'images présentées dans [182] et [172] proposent de fusionner les images visibles et infrarouges en passant l'image couleur dans le système HSV ou YUV plutôt que RGB et en incluant l'intensité infrarouge dans ces valeurs. On peut ainsi conserver les teintes de couleurs acquises dans le visible (en HS ou

en UV), tout en augmentant l'intensité (V ou Y) des objets chauds. Leur principal intérêt est la conservation en majorité des couleurs réelles de la scène. Cependant, la modification spécifique aux objets ne les rend pas forcément très démarqués du fond de l'image.

Afin d'améliorer leur résultats, certaines techniques se basent sur une combinaison entre l'insertion de l'infrarouge dans l'intensité de l'image couleur et l'exploitation d'une image « cible » mais avec pour conséquence d'éloigner l'image résultante des couleurs réelles.

#### 6.4.1.c Fusion de données par réseaux de neurones

Les travaux de colorisation d'images en niveaux de gris [188][31][79] et de colorisation d'images infrarouge [98] par réseaux de neurones à convolution sont envisageables dans la conception d'un algorithme de fusion de données infrarouge/visible par Deep Learning.

Dans [188] les auteurs présentent l'outil Deep Colorization; un réseau de neurones à convolution estime la chrominance d'une image en niveaux de gris dans l'espace CIELAB. Il prend en entrée le canal L d'une image en niveaux de gris et estime ses deux canaux chromatiques a et b. L'espace de couleurs CIELAB a été choisi car il est perceptuel, c'est-à-dire que la distance euclidienne entre deux couleurs dans l'espace CIELAB reflète la différence perçue par l'œil humain entre ces deux couleurs. Au lieu de traiter le problème comme un problème de régression de valeurs réelles pour a et b, le gamut ab est divisé en Q classes (intervalles) différentes et pour chaque pixel on estime les probabilités de son appartenance à chacune de ces classes. Ainsi, au lieu d'utiliser l'erreur quadratique moyenne, la fonction objectif se base sur le calcul de l'entropie croisée multinomiale avec un terme multiplicatif qui fait en sorte que les couleurs rares ne soient pas négligées. Enfin l'algorithme du recuit simulé est utilisé pour assigner chaque pixel à une classe finale. Les auteurs partent de la luminance de l'image en niveaux de gris pour estimer sa chrominance. Ceci n'est pas faisable pour la colorisation d'une image infrarouge où il faut estimer la luminance et la chrominance.

Les auteurs de [98] proposent une méthode de colorisation d'images infrarouges par apprentissage profond. Le réseau de neurones à convolution prend une image infrarouge en entrée et donne en sortie une image RGB correspondante. Afin de conserver les détails contenus dans l'image infrarouge, les propriétés spectrales de celle-ci sont transférées à l'image RGB estimée en utilisant une architecture multi-échelles : une phase de pré-traitements consiste en la construction d'une pyramide normalisée de l'image infrarouge, et le réseau est composé d'autant de branches que de niveaux de la pyramide, chaque branche étant responsable du traitement d'une image de la pyramide. Les branches ont la même structure mais utilisent des poids différents. La couche finale du réseau est entièrement connectée et a pour fonction de combiner la sortie des différentes branches. La dernière phase consiste en des post-traitements de l'image RGB estimée par le réseau de neurones : ses contours sont rendus plus nets par un filtre bilatéral joint qui prend l'image infrarouge comme référence, tandis que sa texture est améliorée en utilisant la composante haute fréquence de l'image d'entrée. Une des limites de cette méthode est que les objets n'émettant pas de radiations infrarouge ne seront pas présents dans l'image colorisée alors qu'ils le sont dans l'image du spectre visible.

Les méthodes de fusion infrarouge/vision par apprentissage profond sont capables d'atteindre de très bons résultats. Cependant, nous manquons d'ensemble de données d'entraînement pour considérer de ré-apprendre ces fonctions pour le cas particulier des zones aéroportuaires. De plus, la certification de ces méthodes est rendu difficile par leur complexité algorithmique.

#### 6.4.1.d Évaluation de la fusion d'images visibles et infrarouges

Nous considérons que nous possédons, comme présentée en Figure 6.5 une image dans le domaine visible et une image issue d'un capteur LWIR à laquelle on a déjà appliqué des pré-traitements type CLAHE.

Nous avons vu au Chapitre 3, comment construire une vue fusionnée à partir des deux images VIS et IR acquises par les caméras intégrées dans un boîtier qui pourrait être monté dans la dérive d'un aéronef (Figure 3.3). Les images sont donc fusionnées au sens géométrique, avec une résolution qui est celle de la vue fusionnée, choisie en fonction des utilisations prévues pour cette vue (simple affichage, détection de lignes, débrumage...).

Cette vue fusionnée est une image avec 4 canaux RGBT. La problématique ici, est de se ramener à une image RGB affichable au pilote, ou traitable par nos algorithmes, image RGB dans laquelle les informations de couleur et de réflectance infrarouge sont fusionnées au mieux.

Nous proposons deux algorithmes différents pour la colorisation d'images infrarouge, une approche basée sur la transformée en ondelettes discrète et une approche basée transfert de couleur. Ces travaux ont été initiés dans le cadre du projet TALVIS, en coopération avec S. Kaddaoui, qui a participé à ce projet en tant qu'ingénieur.

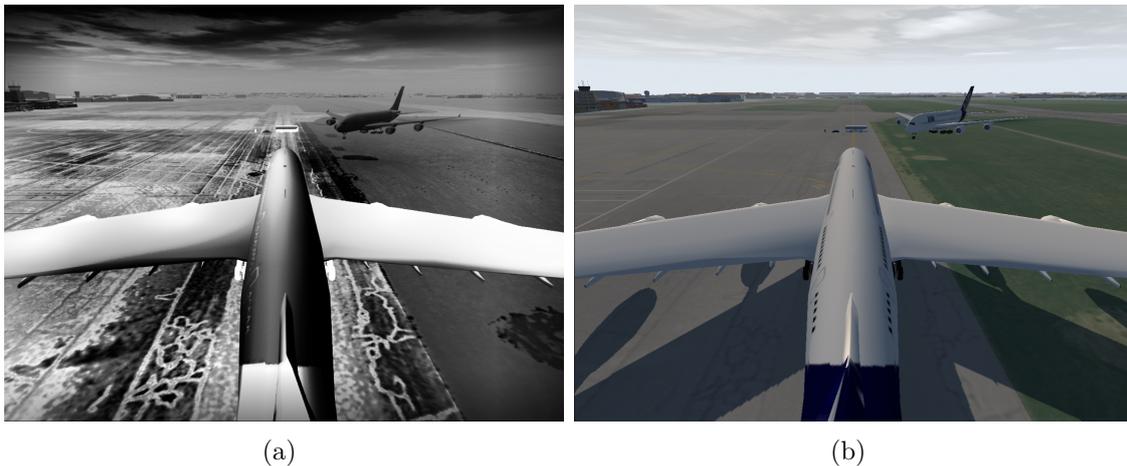


Figure 6.5: Exemples d'images associées simulées (a) en infrarouge et traitée par CLAHE et (b) dans le domaine du visible.

La Figure 6.6 présente le principe de la transformée en ondelettes discrète, basé sur l'article [127], et la Figure 6.8(a) son résultat, discuté ci-après. Dans un premier temps, on convertit l'image visible RGB dans l'espace de couleur CIELAB. On applique une transformée en ondelettes discrète ( $DWT$ ) au canal  $L$  de l'image visible convertie en CIELAB, ainsi qu'à l'image infrarouge obtenue après traitements pour obtenir les coefficients ( $A$  l'approximation de l'image,  $HD$  les détails horizontaux,

$VD$  les détails verticaux et  $DD$  les détails diagonaux) pour l'image infrarouge et l'image RGB (indités  $IR$  et  $VIS$ ). La fusion de ces coefficients est faite par une étude de similarité entre les énergies des coefficients  $IR$  et  $VIS$  pour créer un canal de luminance  $L$  en appliquant la transformée inverse ( $DWT^{-1}$ ). Ce canal est recombinaé avec les canaux  $A$  et  $B$  de l'image RGB dans le domaine CIELAB puis l'image est reconvertie en RGB.

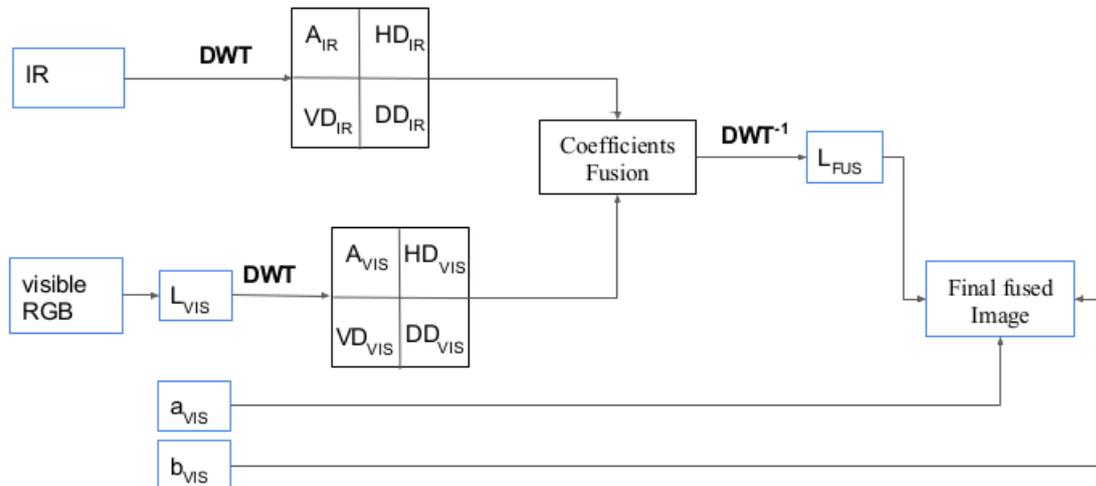


Figure 6.6: Principe de la transformée en ondelettes discrètes.

La Figure 6.7 présente le principe du transfert de couleur et la Figure 6.8(b) son résultat, discuté ci-après. L'image dans le domaine visible (Figure 6.5(b)) est convertie en niveaux de gris et fusionnée à l'image infrarouge traitée pour créer l'image "source", appelée False Color Fusion dans la Figure 6.7. Cette image est composée de trois canaux dont le premier comporte l'information de l'image dans le domaine visible convertie en niveaux de gris et les deux autres canaux sont des combinaisons linéaires des valeurs de l'image visible convertie en niveaux de gris et de l'image infrarouge après traitements. Une estimation des clusters de couleurs est effectuée pour l'image "source" et l'image "cible" puis les clusters sont analysés par une étude statistique (calcul de la moyenne et de l'écart-type) afin de modifier les histogrammes dans l'image en fausse couleur.

Nous avons mené quelques tests sur ces deux approches. La précision de l'algorithme de détection des motifs est un peu moins bonne dans les deux cas. La méthode de transfert de couleur peut être utilisée avec n'importe quelle image possédant la même représentativité en terme de couleurs alors que la méthode qui exploite le domaine fréquentiel impose d'avoir aussi l'image couleur correspondante. Il peut être intéressant de travailler sur la robustesse de l'algorithme ou l'amélioration du rendu de la Figure 6.8(b), en partie pénalisée par l'apparition d'artefacts dus à l'application du CLAHE à l'image infrarouge.

Un autre inconvénient de l'utilisation du transfert de couleur pour nos images est la forte sous-représentativité de la couleur jaune. L'algorithme se basant sur un système de colorisation par classe, le jaune des lignes a tendance à être englobé dans la classe représentant l'herbe. C'est pourquoi nous perdons pour certaines images l'information de la couleur de la ligne. On peut voir une illustration de ce problème dans la Figure 6.9.

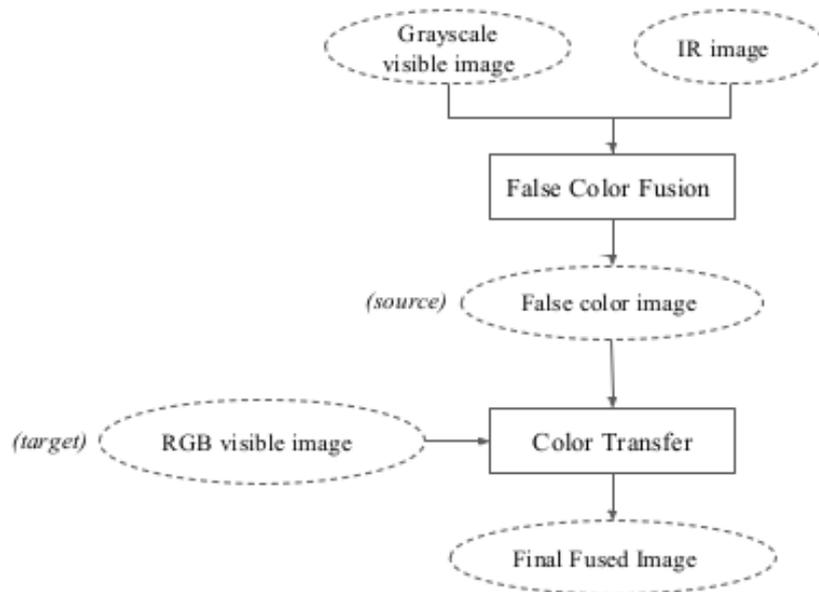


Figure 6.7: Principe du transfert de couleur. Application de la méthode proposée dans l'article [162].

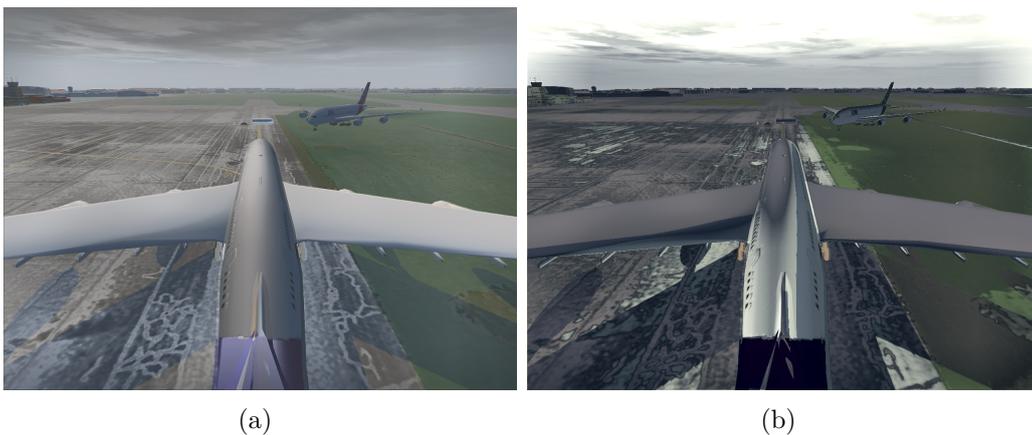
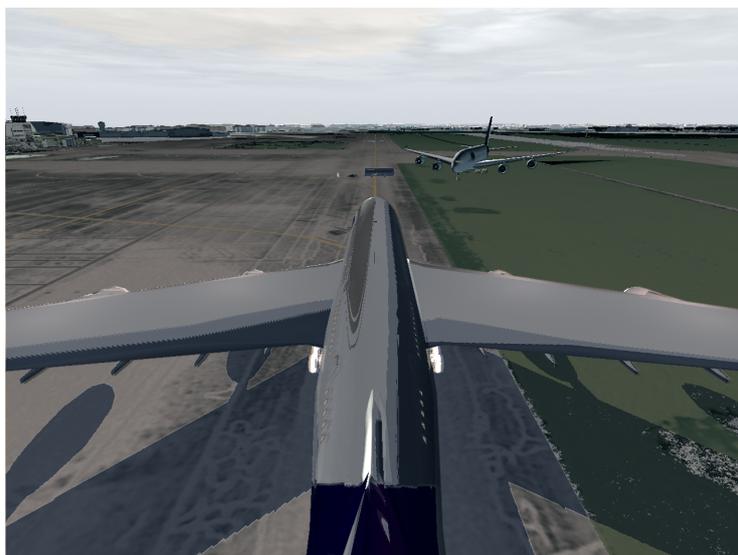


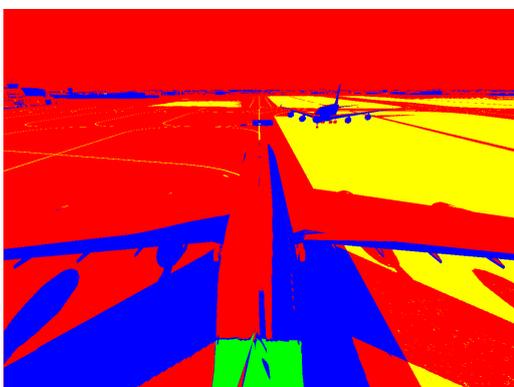
Figure 6.8: Résultat de la fusion des images dans le domaine du visible et infrarouge, présentées ci-dessus, par (a) la transformée en ondelettes discrète et (b) le transfert de couleur.



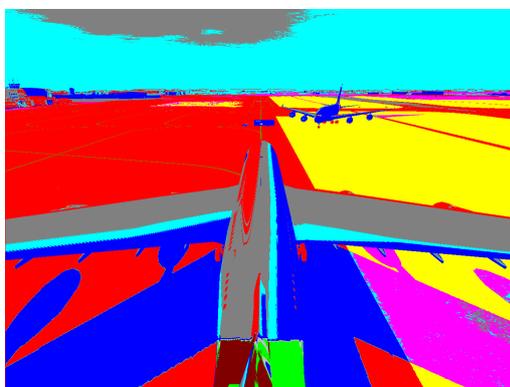
(a)



(b)



(c)



(d)

Figure 6.9: (a) Exemple de transfert de couleur 4 classes, (b) Exemple de transfert de couleur 10 classes, (c) Répartition des 4 classes et (d) Répartition des 10 classes.

### 6.4.2 Application de la fusion de données au débrumage

Ci-dessus, nous avons présenté quelques méthodes de fusion de données provenant de caméras infrarouges et visibles. Dans un premier temps, cela permet de faire de la colorisation d'images infrarouges et peut aussi être utile pour coloriser des images prises de nuit, avec une meilleure source d'informations sur les contrastes apportée par l'image infrarouge. Dans cette section, nous interrogeons l'utilisation de la fusion entre image infrarouge et image visible, non seulement dans le but de produire une colorisation de l'image infrarouge mais aussi de proposer une méthode de débrumage de l'image visible.

#### 6.4.2.a Impact du brouillard sur l'infrarouge

Le brouillard gêne la propagation des radiations électromagnétiques dans l'atmosphère par absorption et diffusion dues aux particules présentes dans celui-ci. Cet impact est dépendant en grande partie de la taille de ces particules appelées aérosols et l'impact du brouillard est augmenté avec la taille des particules.

Les normes OACI définissent le brouillard en trois catégories en fonction des conditions de visibilité notamment par rapport aux conditions de vision de la piste lors de la phase d'atterrissage. Dans l'étude [13], les auteurs proposent d'analyser l'impact du brouillard (modélisé à partir des paramètres de climat, température et type d'aérosols pour former des brouillards des trois catégories) sur la transmission spectrale depuis la borne basse de la plage spectrale du visible ( $0,4 \mu\text{m}$ ) jusqu'à  $15 \mu\text{m}$ . Les meilleurs résultats de transmission sont obtenus pour des longueurs d'ondes comprises entre  $8$  et  $12 \mu\text{m}$ . Cependant, pour des brouillards de catégorie III (c'est à dire à faible visibilité), le gain de visibilité est moins prononcé. Au global, le capteur infrarouge reste moins impacté que le visible lors de l'acquisition de l'image, ce qui explique pourquoi il a été proposé d'utiliser ce type de données pour des algorithmes de débrumage.

#### 6.4.2.b Exemples d'applications au débrumage

Pour la transformée en ondelettes, il n'est pas possible d'utiliser ce processus de fusion dans le cadre du débrumage puisqu'il nous faut une image similaire pour l'image visible et l'image infrarouge. Il faudrait donc disposer de l'image visible dans de bonnes conditions. Cela est techniquement possible avec une base de données d'images et une grande précision dans les méta-données de l'avion : on pourrait imaginer des informations extrêmement précises sur la position GPS et l'orientation de l'aéronef lors de la capture entre autre. En pratique, c'est peu réalisable puisque l'apparition d'un obstacle tel un autre avion ou un véhicule utilitaire (à l'approche des portes par exemple) n'est pas prévisible.

Dans le cadre du transfert de couleur, on peut penser que la méthode est plus simplement applicable. En effet, les teintes présentes dans les images que l'on peut enregistrer ne devraient pas fortement varier d'un aéroport à l'autre. On pourrait donc imaginer fournir en image d'entrée une image respectant les proportions des différentes couleurs présentes dans les zones aéroportuaires

Dans l'article [144], les auteurs proposent d'utiliser une combinaison d'une image visible et d'une image obtenue dans la bande proche infrarouge (NIR) comme méthode de débrumage. Leur méthode se base sur un filtrage par méthode des

moindres carrés pondérés des deux images et ne nécessite pas d'estimer ni le voile atmosphérique ni la carte de profondeur de la scène étudiée, ce qui la rend moins coûteuse en termes de ressources de calcul. L'image visible est convertie dans un espace couleur luminosité-chromaticité (par exemple  $L^*a^*b$ ) et seule l'information de luminosité est utilisée pour la fusion avec le proche infrarouge. La fusion se fait dans une représentation multi-résolution où l'on sépare l'image basse fréquence des détails (contrastes, structure) et seule l'information de détail est considérée pour la fusion en récupérant la valeur maximale pixel à pixel entre l'information provenant de l'image en proche infrarouge et celle provenant du canal de luminosité de l'image visible. La chromaticité est ajoutée par la suite lorsque l'image est reconstruite par la fusion entre l'information de contraste et l'information de base de l'image (contenant l'information de luminosité de basse fréquence). Pour des grandes variations de luminosité, cette méthode a tendance à fournir des couleurs non réalistes.

De leur côté, les auteurs de [151] présentent une méthode de fusion entre une image en proche infrarouge et une image visible par transfert de couleur dans le but d'effectuer un débrumage. Cependant, il existe des divergences de nuances entre les images en proche infrarouge et le canal de luminosité de l'image visible et les méthodes simples de fusion de ces canaux peuvent amener à des couleurs non réalistes ou à des artefacts. Dans cet article, les auteurs proposent une méthode de régularisation de la couleur afin de les faire disparaître en définissant une relation linéaire entre l'image en proche infrarouge et le canal de luminosité de l'image visible. Ainsi la fusion des deux images est intégrée dans l'équation de débrumage de l'image, basée sur le modèle de Koschmieder.

Enfin, on peut citer l'article [42] dans lequel les auteurs s'intéressent aux incohérences présentes dans les images en proche infrarouge dues aux différentes réflectances des matériaux. Plus précisément, ils se concentrent sur l'impact de l'absorption des infrarouges proches par la végétation. Ils proposent donc une méthode statistique en se basant sur l'hypothèse que le canal rouge de l'espace RGB est le plus robuste face à ces incohérences. En définissant une probabilité d'incohérence du pixel, ils l'appliquent au modèle de fusion utilisé dans [144] afin de minimiser l'obtention de couleur non réalistes.

Dans notre cas, nous pensons qu'utiliser un infrarouge lointain peut être pertinent puisqu'il est moins impacté par les effets du brouillard que le proche infrarouge ou encore le visible. En effet, en cas de fort brouillard, les techniques de débrumage ont souvent des difficultés à estimer les paramètres du modèle de Koschmieder. En réduisant l'impact du brouillard grâce à l'infrarouge lointain, nous pensons qu'une fusion de données avec l'image visible pourrait amener à de meilleurs résultats de débrumage dans ces conditions.

## 6.5 Proposition d'amélioration de méthodes

Nous n'avons malheureusement pas eu le temps de mener à bien tous nos travaux au cours de cette thèse, notamment car nous avons eu du mal à nous procurer des images infrarouges. Dans cette section, nous abordons les perspectives envisagées pour répondre au problème du débrumage.

### 6.5.1 Utilisation de l'IPM pour améliorer la carte de profondeur

Comme nous l'avons présenté au cours de cette thèse, nous avons décidé d'utiliser la transformation IPM comme pré-traitement de nos informations. L'étude de l'utilisation de cette transformation dans des conditions de vision dégradées dans le Chapitre 4 a montré cependant que cela pouvait ajouter du bruit sur des pixels lointains et perturber la détection de lignes.

Dans le cadre de l'utilisation de l'infrarouge pour des méthodes de débrumage, la donnée d'infrarouge est moins soumise au bruit que l'image visible dans des conditions dégradées. C'est pourquoi nous proposons d'utiliser la transformation IPM seulement sur l'image infrarouge. Cela revient à considérer l'hypothèse de monde plan présentée dans l'algorithme de [62]. Nous pensons donc qu'il est possible de reprendre cet algorithme en utilisant l'information de l'infrarouge et la transformation IPM afin d'améliorer l'estimation de la carte de profondeur, aussi bien celle de la route avec la transformation IPM que celle des objets avec les pré-traitements appliqués à l'image infrarouge pour rehausser les contrastes. Pour cela, on pourrait combiner cette méthode avec des méthodes de fusion d'images infrarouges et d'images visibles au niveau du canal de luminosité.

### 6.5.2 Place de la fusion de données dans les approches basées réseau de neurones

L'approche par réseau de neurones à base de GANs présentée dans [44] a la particularité de pouvoir être entraînée sans nécessiter de paires d'images sans brouillard et d'images avec brouillard, puisque les résultats des réseaux générateurs sont comparés avec des images issues des bases de données d'images présentant ou non un impact du brouillard. Elle offre aussi de bons résultats sur les bases de données d'images disponibles dans la littérature. D'après nos premiers essais sur nos images impactées par le brouillard, nous ne retrouvons cependant pas des niveaux de performances équivalents. Nous pensons donc qu'il nous faudrait ré-entraîner ce réseau avec nos données.

Tant qu'à faire, puisque nous devons ré-entraîner ce réseau, il paraît opportun de modifier la structure afin d'y ajouter l'information provenant du capteur infrarouge. Pour cela, nous avons plusieurs possibilités :

- Effectuer une fusion précoce de l'image infrarouge et de l'image visible avant de la fournir au réseau
- Modifier la structure du réseau afin de prendre en compte l'infrarouge comme un quatrième canal permettant de décrire la donnée.

Des études existent déjà dans le but de comparer la fusion précoce et tardive d'informations multimodales, comme par exemple dans [52] où il est question de fusionner une donnée RGB avec une information de profondeur. Pour leur application, la fusion précoce semble apporter de meilleurs résultats.

### 6.5.3 Utilisation de l'algorithme Multi-Scale RetineX

L'œil humain est capable d'adapter sa perception des couleurs en fonction de la luminosité ambiante bien plus facilement que les capteurs d'acquisition d'images dans le spectre du visible. L'algorithme de Multi-Scale RetineX [136] cherche à construire une image éclairée de manière constante par une source de lumière blanche et à améliorer la visibilité des couleurs d'une image en rapprochant le résultat de la perception humaine. Pour cela, la luminosité d'un pixel est caractérisée par son intensité et celle de ses voisins. Son utilisation peut être efficace pour faire ressortir des informations de contrastes dans des images où les écarts de luminosité sont importants ou bien dans les images impactées par le voile atmosphérique. Nous avons appliqué cet algorithme à une image bruitée et disposons pour l'instant de résultats préliminaires engageants (voir Figure 6.10). Cependant, cette méthode peut être coûteuse en temps de calculs et semble complexe à certifier.

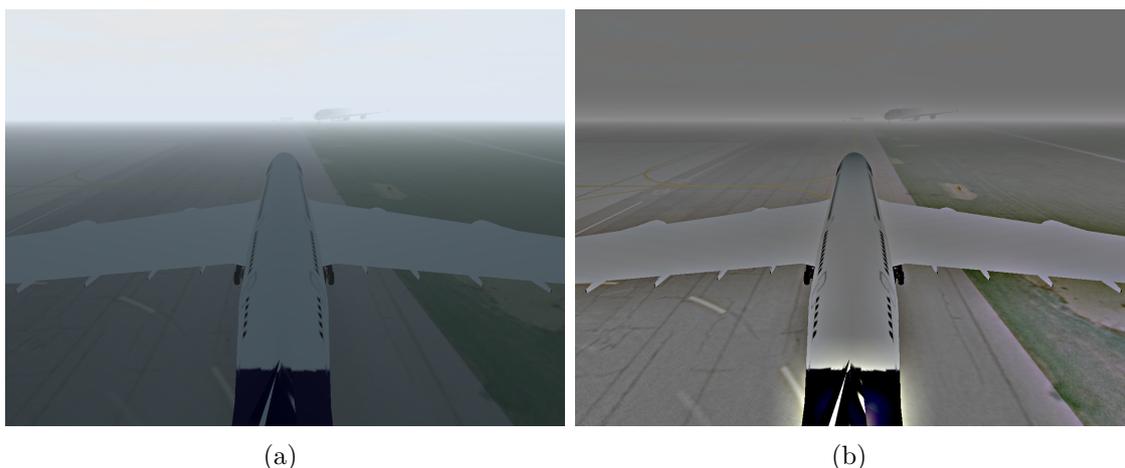


Figure 6.10: (a) Image impactée par le brouillard et (b) Image en sortie de l'algorithme RetineX.

### 6.5.4 Estimation de l'apport de l'infrarouge

Les résultats proposés par RetineX semblent permettre de réaliser une détection de marquages puisque la couleur des lignes jaunes est bien mieux restaurée avec cette méthode qu'avec les méthodes de débrumage implémentées plus tôt. Cependant, pour des épisodes de fort brouillard, nous supposons que l'infrarouge apporte tout de même une information importante sur les contrastes et objets dont la forme pourrait être masquée par le brouillard. C'est pourquoi il serait intéressant de comparer différentes techniques de débrumage avec et sans apport de l'infrarouge pour quantifier son intérêt par rapport à des approches basées seulement sur des images visibles.

## 6.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à un aspect particulier des applications de vision en extérieur : l'impact du brouillard sur les algorithmes d'aide à la navigation. Nous avons adressé ce problème en deux temps. Tout d'abord, une présentation de méthodes proposées dans la littérature dans le cadre du débrumage. Puis nous avons étudié la possibilité d'utiliser la spécificité de notre bloc d'acquisition (présence d'une caméra infrarouge et d'une caméra visible) afin d'améliorer les résultats fournis par les méthodes de débrumage.

De nombreuses méthodes de débrumage ont été proposées ces dernières années. Elles sont basées en majorité sur l'estimation de paramètre du modèle de Koschmieder afin d'atténuer l'impact du brouillard sur l'image résultante acquise par les capteurs. Dans ce chapitre, nous sommes revenus sur quelques unes de ces méthodes. Nous avons sélectionné parmi celles-ci des méthodes basées modèles ainsi que des méthodes à base de réseaux de neurones afin de comparer leurs performances en termes de PSNR et SSIM. Nous avons aussi appliqué l'algorithme de détection de marquages au sol proposé au Chapitre 4, aux images produites par ces différentes méthodes pour évaluer leur performance dans le cadre d'une application d'aide à la navigation.

Dans un second temps, nous avons introduit la possibilité d'exploiter deux images, acquises dans le visible et l'infrarouge par deux caméras intégrées dans un même boîtier. Nous avons présenté quelques méthodes de fusion de données infrarouge/visible et implémenté deux d'entre elles. Puis nous avons analysé quelques méthodes qui proposent d'utiliser cette fusion dans le cadre d'algorithmes de débrumage, surtout basées sur du proche infrarouge.

Enfin, nous avons présenté quelques perspectives à approfondir dans le domaine du débrumage. Nous prévoyons d'utiliser le simulateur afin de générer des images impactées par le brouillard pour créer une base de données grâce à laquelle nous pourrions tester plusieurs algorithmes. En ce sens, dans la thèse de [61], des méthodes de restauration d'images en cas de pluie et brouillard sont présentées, développées à base d'images simulées. Cependant, dans l'article [139], les auteurs se questionnent sur la pertinence d'utiliser des images simulées en entraînement pour améliorer les performances des algorithmes de débrumage. En effet, les modélisations optiques des brouillards simulées reposent sur des approximations qui ne sont pas toutes rencontrées sur des images réelles. Un apprentissage sur des images simulées seulement peut entraîner de grandes différences de résultats pour des algorithmes d'estimation du modèle de brouillard entre images réelles et images simulées. Pour finir, nous n'avons pas encore adressé le problème du débrumage de nuit, où les différentes sources de lumières peuvent mener à des estimations différentes du voile atmosphérique. L'étude très récente présentée dans [4] propose en ce sens une estimation locale du voile atmosphérique et une méthode de débrumage applicable de jour comme de nuit.

# Chapitre 7

## Conclusion

### 7.1 Synthèse

L'objectif de cette thèse était de proposer une solution d'aide à la navigation d'aéronefs dans les zones aéroportuaires. Cette aide passe par une capacité de détection de motifs d'intérêt dans l'environnement de l'appareil (marquages au sol, balises, panneaux, délimitations entre le tarmac et l'herbe...). Elle s'accompagne aussi d'une détection d'obstacles potentiels pouvant gêner ou bloquer le déplacement de l'aéronef. Ces informations, essentielles pour aider à la prise de décision, sont pour l'instant remontées au pilote. Dans un avenir proche, l'objectif devrait être de dépasser la remontée d'informations pour agir directement sur la planification et l'exécution de la trajectoire de l'avion lors de la navigation au sol, comme cela est fait en vol. Enfin, la problématique de l'amélioration des conditions de visibilité du pilote lors d'épisodes de vision dégradée a été abordée en proposant d'utiliser une fusion multi-spectrale de données.

Dans le Chapitre 3 nous avons présenté plusieurs pré-traitements appliqués aux images provenant des caméras visibles et infra-rouges. Nous avons aussi étudié la possibilité de travailler avec plusieurs caméras disposées à différents endroits de l'aéronef et discuté d'algorithmes de fusion de ces informations. Enfin, nous avons utilisé les résultats des travaux sur le pré-traitement de la couleur de l'image RGB pour présenter une méthode de labellisation semi-automatique des images simulées et réelles dont nous disposons. Ceci dans le but de simplifier la création d'une base de données d'images labellisées nécessaire pour le développement de solutions algorithmiques basées sur l'apprentissage profond afin de segmenter les marquages dans les images RGB.

De nombreux espaces et distances colorimétriques ont été testés dans l'étude de la couleur des marquages jaunes et les résultats de cette étude sur la détection de marquages ont été significativement positifs. Cependant, de récents travaux sur l'espace de couleur HSI offrent des résultats intéressants relativement à la couleur "Jaune". De plus, cette étude n'a pas encore été étendue aux autres types de couleurs que l'on peut trouver pour les marquages dans les zones aéroportuaires (blanc, rouge, bleu). Les premiers résultats des méthodes de labellisation sont encourageants mais ne peuvent se passer pour l'instant d'une vérification de l'utilisateur et ne mènent donc pas encore à une labellisation automatique.

Le Chapitre 4 décrit la brique algorithmique proposée pour répondre à la problématique de l'analyse de l'environnement dans lequel se déplace l'aéronef. Nous avons proposé une méthode basée sur l'utilisation d'un filtre particulaire, couplé à des cartes d'observation, qui encodent dans des poids, la probabilité de détecter différents motifs d'intérêt dans une image, comme les marquages au sol ou encore les balises. Nous avons comparé cette méthode à une détection de marquages par transformée de Hough et par réseau de neurones. Dans ce chapitre, nous nous sommes aussi intéressés à la pertinence de l'utilisation de la transformation IPM. La capacité à détecter les marquages dans des conditions de visibilité réduites a aussi été abordée.

La méthode de filtrage particulaire offre des résultats satisfaisants quant à la détection de balises et de marquages. Bien que nous ayons constaté que la transformation IPM permet d'améliorer la détection des marquages dans l'environnement lointain de la caméra, nous avons constaté qu'elle n'est pas productive lors de conditions météorologiques dégradées comme à la tombée de la nuit ou durant des épisodes de brouillard, car, dans ce cas, la qualité de la donnée visuelle est déjà réduite et le bruit ajouté par la transformation de l'image rend la détection encore plus difficile.

Le Chapitre 5 détaille la brique algorithmique dédiée à la détection d'obstacles. Nous avons décrit une utilisation des motifs (balises, marquages...) pour estimer le déplacement de l'aéronef d'une image à l'autre. Une fois que nous possédons la matrice de mouvement de l'aéronef, nous utilisons la transformation IPM sur deux images successives afin de former une carte d'obstacles présents autour de l'aéronef. Enfin, nous effectuons une remontée d'informations au pilote. L'avantage de cette approche est de détecter des obstacles sans informations a priori sur leur nature (aéronef, voiture...). Elle serait donc adaptée pour la problématique dite de "Foreign Object Detection", ou détection des objets perdus sur les pistes (pièces d'aéronefs, bagages...).

L'utilisation de différents motifs est nécessaire pour des cas de dépassement dans les virages par exemple, où les marquages ne sont pas forcément visibles par la caméra placée dans la dérive. L'utilisation de plusieurs caméras placées à différents endroits de l'aéronef permet ainsi de maximiser les points de vue d'un même objet pour augmenter la précision de la détection ainsi que de détecter des motifs masqués pour certaines caméras. La définition de l'équation des marquages pourrait être améliorée avec des méthodes plus pointues pour quantifier la précision de l'estimation du mouvement. Cependant, nous constatons que la remontée d'informations au pilote, pour des obstacles à moyenne et longue portée, est rendue possible par la solution proposée au cours de cette thèse. Il reste à mieux caractériser les performances de l'approche sur une base de données, qui est en cours d'acquisition sur le simulateur OKTAL SE.

Le Chapitre 6 s'intéresse à l'impact de conditions météorologiques entraînant une diminution de la visibilité, et plus spécifiquement lors des épisodes de brouillard. Nous avons présenté une comparaison de plusieurs méthodes de débrumage proposées dans la littérature et étudié leur intérêt sur la détection des marquages, dans un premier temps. Nous avons aussi discuté de l'utilisation de la donnée acquise par la caméra infra-rouge afin d'améliorer la visibilité dans ces conditions.

Nous remarquons que l'utilisation de l'infra-rouge permet un rehaussement des

contrastes. Les algorithmes de débrumage permettent une amélioration de la détection des marquages mais posent encore problème du fait des modifications qu'ils induisent sur la couleur de ces marquages. Cependant, le rehaussement des contrastes permet une meilleure visualisation des obstacles lors de la remontée d'informations au pilote. La fusion de l'information infra-rouge avec l'information de la caméra visible reste à améliorer.

## 7.2 Perspectives

L'étude réalisée sur la couleur des marquages dans le Chapitre 3, qui a mené à une présentation de méthode de semi-labellisation de marquages s'est concentré sur des méthodes classiques de calculs de distance entre couleurs. Dans un futur proche, la base d'images disponible dans le contexte aéroportuaire devrait être fortement augmentée. Il serait intéressant de mettre à jour les résultats de l'étude colorimétrique sur un plus grand set d'images. De plus, il serait intéressant de développer la recherche au niveau de réseaux de neurones dont le but serait de trouver une représentation de l'image pour laquelle les marquages ressortent par rapport au reste de la scène. Enfin, nous avons présenté, en conclusion de l'application à la labellisation, une méthode qui prendrait en compte l'information de spatialité dans le processus de labellisation. Il serait intéressant de développer cette piste dans le but d'améliorer la labellisation.

Dans le Chapitre 4, nous avons présenté l'ébauche d'un filtre particulière fenêtré afin d'améliorer la détection et possiblement le suivi des marquages. La phase d'initialisation ainsi que la fonction d'évaluation sont encore en cours de développement et devraient amener prochainement à une publication. Par ailleurs, la base de données n'étant pas assez grande pour le moment, la comparaison, dans ce chapitre, de l'algorithme de détection de marquages avec la méthode LaneNet est à approfondir après ré-entraînement de ce réseau. Nous avons ainsi constaté que le réseau ne peut pas être porté directement dans le domaine aéronautique, les motifs des marquages au sol et le positionnement de l'aéronef par rapport à ceux-ci étant assez différents du cas automobile pour nécessiter un ré-apprentissage.

La construction d'une base d'images labellisées automatiquement via le simulateur de OKTAL SE devrait, à court terme, permettre de ré-entraîner ce réseau de neurones qui, à première vue, présente un fort potentiel. Il serait aussi intéressant de comparer avec les résultats obtenus par une segmentation sémantique, via une architecture SegNet par exemple. AIRBUS a fait des premiers travaux dans ce domaine, qui semblent prometteurs.

Le Chapitre 5 présente des résultats qualitatifs de la méthode proposée. L'estimation de l'équation des marquages est à améliorer pour prendre en compte les parties courbes, même si l'utilisation des parties linéaires de la droite semble pour l'instant suffisante mais nous avons montré en Figure 1.5 que les courbes des marquages peuvent être assez complexes dans certaines zones des aéroports. Cela rendrait plus robuste l'appariement des motifs. Un ajout à court terme serait une étude comparative avec des méthodes géométriques existantes, autant dans l'appariement des motifs et l'estimation du déplacement de l'avion que dans la détection d'obstacles.

L'augmentation de la base de données devrait mener à une comparaison par rapport à des méthodes basées réseau de neurones. Elle serait spécifique aux objets et obstacles que nous souhaitons détecter (ainsi qu'aux signalétiques aéroportuaires) et composée de plus d'images de synthèse (annotées automatiquement) que d'images réelles (annotées manuellement). Plusieurs répartitions (90-10, 80-20...) devront être testées, en évaluant les performances sur des images réelles.

Dans le Chapitre 6, nous avons proposé plusieurs pistes d'amélioration d'algorithmes afin de maximiser la prise en compte de l'information de la caméra infra-rouge dans le débrumage de l'image visible. Dès lors que nous posséderons une base de données assez importante, il serait intéressant d'approfondir ces propositions. Il y a cependant un bémol à apporter. Nous travaillons avec des images de brouillard simulées et les algorithmes performant de l'état de l'art ne fournissent pas les résultats escomptés sur nos données. Il faudrait donc étudier la pertinence des images simulées en amont. L'acquisition de données réelles au cours d'épisodes de brouillard serait un plus.

Cette thèse a été réalisée dans un environnement fortement simulé. La détection de marquages au sol a pu être appliquée à quelques images obtenues en condition réelles mais elles n'ont pas permis de valider la détection de balises, panneaux ou obstacles. Il serait donc intéressant de pouvoir étudier ces scénarii sur des images réelles pour valider l'utilisation de la solution algorithmique proposée. De plus, les algorithmes ne sont actuellement pas optimisés pour une application embarquée et un travail est à effectuer en ce sens pour réduire les temps de calculs. Enfin, le problème de la certification de ces algorithmes se pose avant une potentielle mise sur le marché d'une telle technologie.

# Bibliographie

- [1] ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SÜSSTRUNK, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 11 (2012), 2274–2282. 92
- [2] ALHAMWI, A., VANDEPORTAELE, B., AND PIAT, J. Real time vision system for obstacle detection and localization on fpga. In *Computer Vision Systems* (Cham, 2015), L. Nalpantidis, V. Krüger, J.-O. Eklundh, and A. Gasteratos, Eds., Springer International Publishing, pp. 80–90. 170
- [3] ÁLVAREZ, J. M., GEVERS, T., AND LÓPEZ, A. M. Road detection by one-class color classification: Dataset and experiments. *ArXiv abs/1412.3506* (2014). 60, 86
- [4] ANCUTI, C., CODRUTA, A., VLEESCHOUWER, C., AND BOVIK, A. Day and night-time dehazing by local airlight estimation. *IEEE Transactions on Image Processing PP* (04 2020), 1–1. 196
- [5] ANDERSON, M., MOTTA, R., CHANDRASEKAR, S., AND STOKES, M. Proposal for a standard default color space for the internet - srgb. In *Color Imaging Conference* (1996). 63
- [6] ANGERMANN, M., WOLKOW, S., DEKIERT, A., BESTMANN, U., AND HECKER, P. Linear blend: Data fusion in the image domain for image-based aircraft positioning during landing approach. In *Proceedings of the ION 2019 Pacific PNT Meeting* (05 2019), pp. 752–766. 19
- [7] ANWAR, M. I., AND KHOSLA, A. Vision enhancement through single image fog removal. *Engineering science and technology, an international journal* 20, 3 (2017), 1075–1083. 177
- [8] ARORA, T., SINGH, G., AND KAUR, M. Evaluation of a new integrated fog removal algorithm idcp with airlight. *International Journal of Image, Graphics and Signal Processing* (2014). 174, 175, 181
- [9] ASIF, M., ARSHAD, M. R., ZIA, M. Y. I., AND YAHYA, A. I. An implementation of active contour and kalman filter for road tracking. *International Journal of Applied Mathematics* 37, 2 (2007), 71–77. 98, 101
- [10] BAK, A., BOUCHAFA, S., AND AUBERT, D. Dynamic objects detection through visual odometry and stereo-vision: a study of inaccuracy and improvement sources. *Machine vision and applications* 25, 3 (2014), 681–697. 139

- [11] BAUMANN, F., EHLERS, A., VOGT, K., AND ROSENHAHN, B. Cascaded random forest for fast object detection. In *Scandinavian Conference on Image Analysis* (2013), Springer, pp. 131–142. 144
- [12] BEAUCHEMIN, S. S., AND BARRON, J. L. The computation of optical flow. *ACM computing surveys (CSUR)* 27, 3 (1995), 433–466. 139
- [13] BEIER, K., AND GEMPERLEIN, H. Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation. *Aerospace science and technology* 8, 1 (2004), 63–71. 192
- [14] BERMAN, D., TREIBITZ, T., AND AVIDAN, S. Single image dehazing using haze-lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 3 (2020), 720–734. 178
- [15] BERTOZZI, M., AND BROGGI, A. Gold: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 7, 1 (1998), 62–81. 52, 140
- [16] BORKAR, K., AND MUKHERJEE, S. Single image dehazing by approximating and eliminating the additional airlight component. *Neurocomputing* 400 (2020), 294–308. ix, 175
- [17] BOTERO GALEANO, D. A. *Development of algorithms and architectures for driving assistance in adverse weather conditions using FPGAs*. Theses, INSA de Toulouse, Dec. 2012. 13, 35, 140
- [18] BREIMAN, L. Random forests. *Machine learning* 45, 1 (2001), 5–32. 144
- [19] BUADES, A., COLL, B., AND MOREL, J. . A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 2, pp. 60–65 vol. 2. 37
- [20] BURGESS, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2, 2 (1998), 121–167. 85
- [21] CAI, Z., FAN, Q., FERIS, R. S., AND VASCONCELOS, N. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision* (2016), Springer, pp. 354–370. 145
- [22] CERRI, P., AND GRISLERI, P. Free space detection on highways using time correlation between stabilized sub-pixel precision ipm images. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), IEEE, pp. 2223–2228. 140
- [23] CHAKRAVORTY, T., BILODEAU, G.-A., AND GRANGER, E. Automatic image registration in infrared-visible videos using polygon vertices, 2014. 40
- [24] CHAPUIS, R., AUFRERE, R., AND CHAUSSE, F. Accurate road following and reconstruction by computer vision. *IEEE Transactions on Intelligent Transportation Systems* 3, 4 (Dec. 2002), 261–270. 101

- [25] CHAPUIS, R., POTELLE, A., BRAME, J., AND CHAUSSE, F. Real-time vehicle trajectory supervision on the highway. *The International Journal of Robotics Research* 14, 6 (1995), 531–542. 98
- [26] CHAVEZ-GARCIA, R. O. *Multiple sensor fusion for detection, classification and tracking of moving objects in driving environments*. PhD thesis, Université de Grenoble, 2014. 143
- [27] CHAVEZ-GARCIA, R. O., BURLET, J., VU, T.-D., AND AYCARD, O. Frontal object perception using radar and mono-vision. In *2012 IEEE Intelligent Vehicles Symposium* (2012), IEEE, pp. 159–164. 143
- [28] CHAVEZ-GARCIA, R. O., VU, T.-D., AYCARD, O., AND TANGO, F. Fusion framework for moving-object classification. In *Proceedings of the 16th International Conference on Information Fusion* (2013), IEEE, pp. 1159–1166. 142, 144
- [29] CHEN, P., LO, S., HANG, H., CHAN, S., AND LIN, J. Efficient road lane marking detection with deep learning. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)* (2018). 59
- [30] CHENG, H., JIANG, X., SUN, Y., AND WANG, J. Color image segmentation: Advances and prospects. *Pattern Recognition* (2001). 59
- [31] CHENG, Z., YANG, Q., AND SHENG, B. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 415–423. 187
- [32] CLARKE, F. J. J., MCDONALD, R., AND RIGG, B. Modification to the jpc79 colour–difference formula. *Journal of the Society of Dyers and Colourists* 100, 4 (1984), 128–132. 66
- [33] COLL, B., AND MOREL, J.-M. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation* 4 (01 2005). 36
- [34] DANESCU, R., AND NEDEVSCHI, S. Probabilistic lane tracking in difficult road scenarios using stereovision. *Intelligent Transportation Systems, IEEE Transactions on* 10 (07 2009), 272 – 282. 100
- [35] DENG, J., KIM, J., SIN, H., AND HAN, Y. Fast lane detection based on the b-spline fitting. *International Journal of Research in Engineering and Technology* 2, 4 (2013), 134–137. 43, 98
- [36] DERIGS, U. Solving non-bipartite matching problems via shortest path techniques. *Annals of Operations Research* 13 (1988), 225–261. 153
- [37] DEROME, M. *Vision stéréoscopique temps-réel pour la navigation autonome d’un robot en environnement dynamique*. PhD thesis, Université Paris-Saclay, 2017. 24, 139

- [38] DEVY, M., MANZANO, M. I., BOIZARD, J.-L., LACROIX, P., FILALI, W., AND FOURNIOLS, J.-Y. Integrated subsystem for obstacle detection from a belt of micro-cameras. In *2009 International Conference on Advanced Robotics* (2009), IEEE, pp. 1–6. 46
- [39] DING, C., GU, J., MA, X., GAO, F., AND LI, Y. Target recognition in different color spaces. *2013 IEEE International Conference on Information and Automation, ICIA 2013* (08 2013), 151–155. vi, 65
- [40] DOGRA, A., GOYAL, B., AND AGRAWAL, S. From multi-scale decomposition to non-multi-scale decomposition methods: A comprehensive survey of image fusion techniques and its applications. *IEEE Access* 5 (2017), 16040–16067. 39
- [41] DOURET, J., AND BENOSMAN, R. A multi-cameras 3d volumetric method for outdoor scenes: a road traffic monitoring application. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR* (09 2004), vol. 3, pp. 334 – 337 Vol.3. 164
- [42] DÜMBGEN, F., HELOU, M. E., GUCEVSKA, N., AND SÜSTRUNK, S. Near-infrared fusion for photorealistic image dehazing. *Electronic Imaging 2018*, 16 (2018), 321–1. 193
- [43] ELFES, A. Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation* 3, 3 (1987), 249–265. 137
- [44] ENGIN, D., GENÇ, A., AND EKENEL, H. K. Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing. *arXiv:1805.05308 [cs]* (May 2018). arXiv: 1805.05308. 174, 178, 181, 194
- [45] FAN, Z., BI, D., XIONG, L., MA, S., HE, L., AND DING, W. Dim infrared image enhancement based on convolutional neural network. *Neurocomputing* 272 (2018), 396 – 404. 37
- [46] FATTAL, R. Single image dehazing. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 72:1–72:9. ix, 172, 174, 176, 181, 184
- [47] FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. Distance transforms of sampled functions. *Theory of Computing* 8 (2012), 415–428. 53
- [48] FERRERA, M., EUDES, A., MORAS, J., SANFOURCHE, M., AND LE BESNERAIS, G. Ov<sup>2</sup>slam: A fully online and versatile visual slam for real-time applications. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1399–1406. 148
- [49] FIRMENICHY, D., BROWN, M., AND SÜSTRUNK, S. Multispectral interest points for rgb-nir image registration. In *2011 18th IEEE International Conference on Image Processing* (2011), pp. 181–184. 41
- [50] FORSTER, C., ZHANG, Z., GASSNER, M., WERLBERGER, M., AND SCARAMUZZA, D. Svo: Semidirect visual odometry for monocular and

- multicamera systems. *IEEE Transactions on Robotics* 33, 2 (2016), 249–265. 136, 147
- [51] FRIEDMAN, J., HASTIE, T., TIBSHIRANI, R., ET AL. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of statistics* 28, 2 (2000), 337–407. 144
- [52] GADZICKI, K., KHAMSEHASHARI, R., AND ZETZSCHE, C. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)* (2020), pp. 1–6. 195
- [53] GARCÍA-LAMONT, F., CERVANTES, J., LOPEZ-CHAU, A., AND RODRÍGUEZ, L. Segmentation of images by color features: A survey. *Neurocomputing* 292 (03 2018). 86
- [54] GHAFOORIAN, M., NUGTEREN, C., BAKA, N., BOOIJ, O., AND HOFMANN, M. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *Computer Vision – ECCV 2018 Workshops* (Cham, 2019), L. Leal-Taixé and S. Roth, Eds., Springer International Publishing, pp. 256–272. 102
- [55] GIJSENIJ, A., GEVERS, T., AND LUCASSEN, M. A perceptual comparison of distance measures for color constancy algorithms. In *ECCV 2008, 10th European Conference on Computer Vision* (2008). 64
- [56] GIRSHICK, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1440–1448. 145
- [57] GIRSHICK, R., DONAHUE, J., DARRELL, T., AND MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 580–587. 145
- [58] GONZALEZ, A. *Localisation par vision multi-spectrale. Application aux systèmes embarqués*. PhD thesis, INSA de Toulouse, 2013. 13, 147, 148
- [59] GOWDA, S. N., AND YUAN, C. Colornet: Investigating the importance of color spaces for image classification. In *Computer Vision – ACCV 2018* (2019). 59
- [60] GUO, B., ZHANG, Q., AND HOU, Y. Region-based fusion of infrared and visible images using nonsubsampling contourlet transform. *Chinese Optics Letters* 6, 5 (2008), 338–341. 186
- [61] HALMAOUI, H. *Image Restoration During Foggy and Rainy Weather : Applications for Driver Assistance Systems*. Theses, Université d’Evry-Val d’Essonne, Nov. 2012. 196
- [62] HALMAOUI, H., CORD, A., AND HAUTIÈRE, N. Contrast restoration of road images taken in foggy weather. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops* (2011). 174, 175, 177, 181, 194

- [63] HAN, J., PAUWELS, E. J., AND DE ZEEUW, P. Visible and infrared image registration in man-made environments employing hybrid visual features. *Pattern Recognition Letters* 34, 1 (2013), 42 – 51. Extracting Semantics from Multi-Spectrum Video. 40
- [64] HARTLEY, R. I., AND ZISSERMAN, A. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN: 0521540518, 2004. 42
- [65] HARZALLAH, H. *Contribution à la détection et à la reconnaissance d'objets dans les images*. PhD thesis, Université de Grenoble, 2011. 144
- [66] HAUTIERE, N. *Détection des conditions de visibilité et estimation de la distance de visibilité par vision embarquée*. Theses, Université Jean Monnet - Saint-Etienne, Sept. 2005. 28
- [67] HE, H., LEE, W., CAO, Y., AND LUO, D. Insulator infrared image denoising method based on wavelet generic gaussian distribution and map estimation. *IEEE Transactions on Industry Applications* 53 (2017), 3279–3284. 37
- [68] HE, K., SUN, J., AND TANG, X. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (2009), 2341–2353. 177
- [69] HEDJAZI, L., AGUILAR-MARTIN, J., LE LANN, M.-V., AND KEMPOWSKY-HAMON, T. Membership-margin based feature selection for mixed type and high-dimensional data: Theory and applications. *Information Sciences* 322 (2015), 174–196. 86
- [70] HILDRETH, E. C. Computations underlying the measurement of visual motion. *Artificial Intelligence* 23, 3 (1984), 309–354. 139
- [71] HINES, G. D., UR RAHMAN, Z., JOBSON, D. J., WOODSELL, G. A., AND HARRAH, S. D. Real-time enhanced vision system. In *Enhanced and Synthetic Vision 2005* (2005), J. G. Verly, Ed., vol. 5802, International Society for Optics and Photonics, SPIE, pp. 127 – 134. v, 4, 5
- [72] HORN, B. K., AND SCHUNCK, B. G. Determining optical flow. *Artificial intelligence* 17, 1-3 (1981), 185–203. 139
- [73] HOTA, R. N., SYED, S., B, S., AND RADHAKRISHNA, P. A simple and efficient lane detection using clustering and weighted regression. In *15th International Conference on Management of Data, COMAD* (2009). 82, 98, 100
- [74] HSU, C.-W., AND LIN, C.-J. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks* 13, 2 (2002), 415–425. 85
- [75] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017). 59

- [76] HUANG, J., RATHOD, V., SUN, C., ZHU, M., KORATTIKARA, A., FATHI, A., FISCHER, I., WOJNA, Z., SONG, Y., GUADARRAMA, S., ET AL. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 7310–7311. 145
- [77] IBARRA MANZANO, M. *Vision multi-caméra pour la détection d'obstacles sur un robot de service: des algorithmes à un système intégré*. Theses, INSA de Toulouse, Jan. 2011. 142
- [78] ICAO. *AERODROMES: aerodromes design and operations*. International Civil Aviation Organization (ICAO), 2018. 17
- [79] IIZUKA, S., SIMO-SERRA, E., AND ISHIKAWA, H. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–11. 187
- [80] IKUTA, C., ZHANG, S., UWATE, Y., YANG, G., AND NISHIO, Y. A novel fusion algorithm for visible and infrared image using non-subsampled contourlet transform and pulse-coupled neural network. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)* (2014), vol. 1, IEEE, pp. 160–164. 186
- [81] JIANG, R., KLETTE, R., VAUDREY, T., AND WANG, S. Lane detection and tracking using a new lane model and distance transform. *Machine Vision and Applications* 22 (2010), 721–737. 102
- [82] JIN, H., AND WANG, Y. A fusion method for visible and infrared images based on contrast pyramid with teaching learning based optimization. *Infrared Physics & Technology* 64 (2014), 134–142. 186
- [83] KARARGYRIS, A. Color space transformation network. *arXiv* (2015). 59, 60
- [84] KAZEMI, M., AND BALEGHI, Y. L\*a\*b\* color model based road lane detection in autonomous vehicles. *Bangladesh Journal of Scientific and Industrial Research* 52 (12 2017), 273. 58, 60, 100, 118
- [85] KESIDIS, A. L., AND PAPAMARKOS, N. A gray-scale inverse hough transform algorithm. In *2000 IEEE International Symposium on Circuits and Systems (ISCAS)* (2000). 100
- [86] KHEYROLLAHI, A., AND BRECKON, T. P. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications* (2010). 100
- [87] KOSCHAN, A., AND ABIDI, M. *Digital color image processing*. John Wiley & Sons, 2008. 59
- [88] KÜGLER, M. E., MUMM, N. C., HOLZAPFEL, F., SCHWITHAL, A., AND ANGERMANN, M. *Vision-Augmented Automatic Landing of a General Aviation Fly-by-Wire Demonstrator*. American Institute of Aeronautics and Astronautics, 2019. 19

- [89] LAMBERT, P., AND CHANUSSOT, J. Extending mathematical morphology to color image processing. *Proc. CGIP* (2000). 84
- [90] LAMOSA, F., HU, Z., AND UCHIMURA, K. Vehicle detection using multi-level probability fusion maps generated by a multi-camera system. In *2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance* (2008), pp. 10–17. v, ix, 29, 164, 165
- [91] LARSON, E., COHN, G., GUPTA, S., REN, X., HARRISON, B., FOX, D., AND PATEL, S. Heatwave: Thermal imaging for surface user interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, Association for Computing Machinery, p. 2565–2574. v, 23
- [92] LEE, M., JANG, C., AND SUNWOO, M. Probabilistic lane detection and lane tracking for autonomous vehicles using a cascade particle filter. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 229 (02 2015). 100
- [93] LEE, S., KIM, J., YOON, J. S., SHIN, S., BAILO, O., KIM, N., LEE, T., HONG, H. S., HAN, S., AND KWEON, I. S. Vpnet: Vanishing point guided network for lane and road marking detection and recognition. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 1965–1973. 102
- [94] LEMONDE, V. *Stéréovision embarquée sur véhicule : de l'auto-calibrage à la détection d'obstacles*. PhD thesis, INSA Toulouse, 2005. Thèse de doctorat dirigée par Devy, Michel Systèmes informatiques Toulouse, INSA 2005. 138
- [95] LI, B., PENG, X., WANG, Z., XU, J., AND FENG, D. Aod-net: All-in-one dehazing network. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 4780–4788. 174, 178, 181
- [96] LI, H., CHAI, Y., YIN, H., AND LIU, G. Multifocus image fusion and denoising scheme based on homogeneity similarity. *Optics Communications* 285, 2 (2012), 91 – 100. 39
- [97] LI, S., KANG, X., FANG, L., HU, J., AND YIN, H. Pixel-level image fusion: A survey of the state of the art. *Information Fusion* 33 (2017), 100 – 112. 39
- [98] LIMMER, M., AND LENSCH, H. P. Infrared colorization using deep convolutional neural networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (2016), IEEE, pp. 61–68. 187
- [99] LIN, C.-C., AND WANG, M.-S. A vision based top-view transformation model for a vehicle parking assistant. *Sensors* 12, 4 (2012), 4431–4446. 140
- [100] LIPSKI, C., SCHOLZ, B., BERGER, K., LINZ, C., STICH, T., AND MAGNOR, M. A fast and robust approach to lane marking detection and lane tracking. *2008 IEEE Southwest Symposium on Image Analysis and Interpretation* (2008), 57–60. 58, 60, 98

- [101] LIU, N., AND CHEN, X. Infrared image detail enhancement approach based on improved joint bilateral filter. *Infrared Physics & Technology* 77 (2016), 405 – 413. 36
- [102] LIU, N., AND ZHAO, D. Detail enhancement for high-dynamic-range infrared images based on guided image filter. *Infrared Physics & Technology* 67 (2014), 138 – 147. 36
- [103] LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S., FU, C.-Y., AND BERG, A. C. Ssd: Single shot multibox detector. In *European conference on computer vision* (2016), Springer, pp. 21–37. 145
- [104] LIU, Y., CHEN, X., WANG, Z., WANG, Z. J., WARD, R. K., AND WANG, X. Deep learning for pixel-level image fusion: Recent advances and future prospects. *Information Fusion* 42 (2018), 158 – 173. 39
- [105] LOOSE, H., AND FRANKE, U. B-spline-based road model for 3d lane recognition. *13th International IEEE Conference on Intelligent Transportation Systems* (2010), 91–98. 98
- [106] LOWE, D. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999), vol. 2, pp. 1150–1157 vol.2. 143
- [107] LUCAS, B. D., KANADE, T., ET AL. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)* (1981), Vancouver, British Columbia. 139
- [108] LUO, M. R., CUI, G., AND RIGG, B. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application* 26, 5 (2001), 340–350. 66
- [109] MA, J., MA, Y., AND LI, C. Infrared and visible image fusion methods and applications: A survey. *Information Fusion* 45 (2019), 153–178. 185
- [110] MACADAM, D. L. Visual sensitivities to color differences in daylight\*. *J. Opt. Soc. Am.* (1942). 77
- [111] MACAIRE, L., VANDENBROUCKE, N., AND POSTAIRE, J.-G. Segmentation d'images par classification spatio-colorimétrique des pixels. *Traitement du Signal* (2004). 86
- [112] MADENDA, S. A new perceptually uniform color space with associated color similarity measure for content-based image and video retrieval. *Multimedia Information Retrieval Workshop, 28th Annual ACM SIGIR Conference* (2005). 59, 67
- [113] MAMMERI, A., BOUKERCHE, A., AND TANG, Z. A real-time lane marking localization, tracking and communication system. *Computer Communications* (2016). 58, 60, 100, 118

- [114] MCCULLOCH, W. S., AND PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5, 4 (1943), 115–133. 143
- [115] McDONALD, R., AND SMITH, K. Cie94-a new colour-difference formula. *Journal of the Society of Dyers and Colourists* 111, 12 (1995), 376–379. 66
- [116] MERVEILLEUX-ORZEKOWSKA, P. *Exploration et navigation de robots basées vision omnidirectionnelle*. PhD thesis, Université de Picardie, 2012. Thèse de doctorat dirigée par Mouaddib, El Mustapha et Labbani-Igbida, Ouiddad Sciences. Robotique Amiens 2012. 142
- [117] MEYMANDI-NEJAD, C., HERBULOT, A., AND DEVY, M. Débrumage d’images pour la détection de lignes dans le domaine aéronautique. In *RFIAP* (Vannes, France, June 2020). 173
- [118] MEYMANDI-NEJAD., C., KADDAOUI., S. E., DEVY., M., AND HERBULOT., A. Lane detection and scene interpretation by particle filter in airport areas. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, (2019)*, INSTICC, SciTePress, pp. 517–524. 51, 56, 82, 97, 103, 156
- [119] MEYMANDI-NEJAD, C., PERROTIN, E., HERBULOT, A., AND DEVY, M. Aircraft Navigation on Taxiways: Evaluation of Line Detection Algorithms Proposed for Automotive Applications. In *ISCSIC 2020: 2020 4th International Symposium on Computer Science and Intelligent Control* (Newcastle upon Tyne United Kingdom, United Kingdom, Nov. 2020), ACM, pp. 1–6. 82, 97
- [120] MEYMANDI-NEJAD, C., PERROTIN, E., HERBULOT, A., AND DEVY, M. Colorimetric space study: Application for line detection on airport areas. In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)* (Apr. 2021), SciTePress. 33
- [121] MOKRZYCKI, W., AND TATOL, M. Color difference delta e - a survey. *Machine Graphics and Vision* 20 (04 2011), 383–411. 66
- [122] MORAS, J. *Grilles de perception évidentielles pour la navigation robotique en milieu urbain*. Theses, Université de Technologie de Compiègne, Jan. 2013. 141
- [123] MOTWANI, M. C., GADIYA, M. C., MOTWANI, R. C., AND HARRIS, F. C. Survey of image denoising techniques. In *Proceedings of GSPX* (2004), vol. 27, pp. 27–30. 36
- [124] MUR-ARTAL, R., MONTIEL, J. M. M., AND TARDOS, J. D. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics* 31, 5 (2015), 1147–1163. 147
- [125] NAROTE, S. P., BHUJBAL, P. N., NAROTE, A. S., AND DHANE, D. M. A review of recent advances in lane detection and departure warning system. *Pattern Recognition* (2018). 51, 58

- [126] NEVEN, D., BRABANDERE, B. D., GEORGOULIS, S., PROESMANS, M., AND GOOL, L. V. Towards end-to-end lane detection: an instance segmentation approach. *IEEE Intelligent Vehicles Symposium* (2018). 59, 102
- [127] NIU, Y., XU, S., WU, L., AND HU, W. Airborne infrared and visible image fusion for target perception based on target region segmentation and discrete wavelet transform. *Mathematical Problems in Engineering 2012* (2012). 186, 188
- [128] OLIVEIRA, M., SANTOS, V., AND SAPPA, A. D. Multimodal inverse perspective mapping. *Information Fusion 24* (2015), 108 – 121. 42
- [129] OUATTARA, S., CLÉMENT, A., AND ROJAS-VARELA, J. Segmentation d’images couleur par analyse d’histogrammes couleur compacts. In *15ème colloque CNRIUT* (Lille, France, June 2009). 86
- [130] PALOMARES, J.-L. *Une nouvelle méthode d’appariement de points d’intérêt pour la mise en correspondance d’images*. Theses, Université Montpellier II - Sciences et Techniques du Languedoc, Oct. 2012. 136
- [131] PAPAGEORGIOU, C., AND POGGIO, T. A trainable system for object detection. *International journal of computer vision 38*, 1 (2000), 15–33. 144
- [132] PAUDEL, D. P., DEMONCEAUX, C., HABED, A., VASSEUR, P., AND KWEON, I. 2D-3D Camera Fusion for Visual Odometry in Outdoor Environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (United States, Sep 2014), pp. 1–6. 23
- [133] PIELLA, G. A general framework for multiresolution image fusion: from pixels to regions. *Information Fusion 4*, 4 (2003), 259 – 280. 39
- [134] PIZER, S. M., AMBURN, E. P., AUSTIN, J. D., CROMARTIE, R., GESELOWITZ, A., GREER, T., TER HAAR ROMENY, B., ZIMMERMAN, J. B., AND ZUIDERVELD, K. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing 39*, 3 (1987), 355 – 368. 35
- [135] POULI, T., AND REINHARD, E. Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics 35*, 1 (2011), 67–80. Extended Papers from Non-Photorealistic Animation and Rendering (NPAR) 2010. 185
- [136] RAHMAN, Z.-U., JOBSON, D. J., AND WOODDELL, G. A. Multi-scale retinex for color image enhancement. In *Proceedings of 3rd IEEE International Conference on Image Processing* (1996), vol. 3, IEEE, pp. 1003–1006. 195
- [137] REDMON, J., AND FARHADI, A. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 7263–7271. 145
- [138] RUSU, R. B., BLODOW, N., AND BEETZ, M. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation* (2009), pp. 3212–3217. 136

- [139] SAKARIDIS, C., DAI, D., AND VAN GOOL, L. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision* 126, 9 (2018), 973–992. 196
- [140] SAPPA, A. D., CARVAJAL, J. A., AGUILERA, C. A., OLIVEIRA, M., ROMERO, D., AND VINTIMILLA, B. X. Wavelet-based visible and infrared image fusion: a comparative study. *Sensors* 16, 6 (2016), 861. 186
- [141] SASIDHARAN, S. K. Vehicle detection in images using svm. In *International Journal of Advanced Electrical and Electronics Engineering (IJAEET)* (2013). 144
- [142] SAUDI, A., TEO, J., MOHD HANAFI AHMAD HIJAZI, AND SULAIMAN, J. Fast lane detection with Randomized Hough Transform. In *International Symposium on Information Technology* (2008), IEEE. 100
- [143] SCHANDA, J. *Colorimetry: understanding the CIE system*. John Wiley & Sons, 2007. 66
- [144] SCHAUL, L., FREDEMBACH, C., AND SÜSSTRUNK, S. Color image dehazing using the near-infrared. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (2009), IEEE, pp. 1629–1632. 192, 193
- [145] SCHECHNER, Y. Y., NARASIMHAN, S. G., AND NAYAR, S. K. Instant dehazing of images using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (2001), vol. 1, IEEE, pp. I–I. 179
- [146] SCHMID, M. R., ATEES, S., DICKMANN, J., VON HUNDELSHAUSEN, F., AND WUENSCH, H. . Parking space detection with hierarchical dynamic occupancy grids. In *2011 IEEE Intelligent Vehicles Symposium (IV)* (2011), pp. 254–259. 21
- [147] SEHESTEDT, S., KODAGODA, S., ALEMPIJEVIC, A., AND DISSANAYAKE, G. Robust lane detection in urban environments. *2007 IEEE/RSJ Int. Conference on Intelligent Robots and Systems* (2007), 123–128. 102
- [148] SHIN, B.-S., TAO, J., AND KLETTE, R. A superparticle filter for lane detection. *Pattern Recognition* 48, 11 (2015), 3333–3345. 100
- [149] SIMA, A., AND BUCKLEY, S. Optimizing sift for matching of short wave infrared and visible wavelength images. *Remote Sensing, vol. 5, issue 5, pp. 2037-2056* 5 (04 2013), 2037–2056. 41
- [150] SOLA, J., VIDAL-CALLEJA, T., CIVERA, J., AND MONTIEL, J. M. M. Impact of landmark parametrization on monocular ekf-slam with points and lines. *International journal of computer vision* 97, 3 (2012), 339–368. 147
- [151] SON, C., AND ZHANG, X. Near-infrared fusion via color regularization for haze and color distortion removals. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 11 (2018), 3111–3126. 193

- [152] SRIVASTAVA, S., LUMB, M., AND SINGAL, R. Improved lane detection using hybrid median filter and modified hough transform. *International Journal of Advanced Research in Computer Science and Software Engineering* (2014). 100
- [153] SULTANA, F., SUFIAN, A., AND DUTTA, P. Evolution of image segmentation using deep convolutional neural network: A survey. *Knowledge-Based Systems 201-202* (2020), 106062. 86
- [154] SUN, T.-Y., TSAI, S.-J., AND CHAN, V. Hsi color model based lane-marking detection. *2006 IEEE Intelligent Transportation Systems Conference* (2006), 1168–1172. 58, 60, 98
- [155] SUN, Z., BEBIS, G., AND MILLER, R. On-road vehicle detection using optical sensors: A review. In *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)* (2004), IEEE, pp. 585–590. 143
- [156] TAN, R. T., PETTERSSON, N., AND PETERSSON, L. Visibility enhancement for roads with foggy or hazy scenes. In *2007 IEEE Intelligent Vehicles Symposium* (2007), IEEE, pp. 19–24. 177
- [157] TAREL, J., AND HAUTIÈRE, N. Fast visibility restoration from a single color or gray level image. In *2009 IEEE 12th International Conference on Computer Vision* (2009). 174, 176, 177
- [158] TAY, M., MEKHNACHA, K., YGUEL, M., COUE, C., PRADALIER, C., LAUGIER, C., FRAICHARD, T., AND BESSIERE, P. The bayesian occupation filter. In *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*. Springer, 2008, pp. 77–98. 141
- [159] THEUMA, K., GAUCI, J., CHIRCOP, K., AND ZAMMIT-MANGION, D. *Fusion of Point Clouds for Obstacle Tracking during Airport Ground Operations*. American Institute of Aeronautics and Astronautics, 2020. 19
- [160] THEUMA, K., AND MANGION, D. Z. An image processing algorithm for ground navigation of aircraft. In *Advances in Aerospace Guidance, Navigation and Control* (Cham, 2015), J. Bordeneuve-Guibé, A. Drouin, and C. Roos, Eds., Springer International Publishing, pp. 381–399. 19, 98
- [161] THEUMA, K., ZAMMIT-MANGION, D., GAUCI, J., CHIRCOP, K., AND RIVIERE, N. *A Particle Filter for Ground Obstacle Tracking in the Airfield*. American Institute of Aeronautics and Astronautics, 2019. 19
- [162] TOET, A. Natural colour mapping for multiband nightvision imagery. *Information Fusion 4*, 3 (2003), 155–166. ix, 185, 190
- [163] TOET, A., AND FRANKEN, E. M. Perceptual evaluation of different image fusion schemes. *Displays 24*, 1 (2003), 25–37. 185
- [164] TOMAS, B., ONDREJ, K., AND ONDREJ, P. Methods and systems for providing taxiway stop bar information to an aircrew, 2017. 19, 98

- [165] TOMASI, C., AND KANADE, T. Detection and tracking of point features. Tech. rep., International Journal of Computer Vision, 1991. 136
- [166] TRAN, A., AND MANZANERA, A. A versatile object tracking algorithm combining Particle Filter and Generalised Hough Transform. In *International Conference on Image Processing Theory, Tools and Applications* (2015), IEEE. 116
- [167] VARADARAJAN, K. R., AND AGARWAL, P. K. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 1999), SODA '99, Society for Industrial and Applied Mathematics, pp. 805–814. 153
- [168] VON GIOI, R. G., JAKUBOWICZ, J., MOREL, J.-M., AND RANDALL, G. Lsd: a line segment detector. *Image Processing On Line 2* (2012), 35–55. 99
- [169] VU, T.-D. *Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects*. Theses, Institut National Polytechnique de Grenoble - INPG, Sept. 2009. 141
- [170] WANG, C.-C., AND THORPE, C. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)* (2002), vol. 3, IEEE, pp. 2918–2924. 141
- [171] WANG, J., AND HONG, J. A new self-adaptive weighted filter for removing noise in infrared images. In *2009 International Conference on Information Engineering and Computer Science* (Dec 2009), pp. 1–4. 36
- [172] WAXMAN, A. M., AGUILAR, M., FAY, D. A., IRELAND, D. B., AND RACAMATO, J. P. Solid-state color night vision: fusion of low-light visible and thermal infrared imagery. *Lincoln Laboratory Journal 11*, 1 (1998), 41–60. 186
- [173] WIKIPEDIA. Hsl and hsv. [https://en.wikipedia.org/wiki/HSL\\_and\\_HSV](https://en.wikipedia.org/wiki/HSL_and_HSV). Visualisé le 25-01-2021. vi, 67
- [174] WIKIPEDIA. Yiq. <https://en.wikipedia.org/wiki/YIQ>. Visualisé le 25-01-2021. vi, 68
- [175] WIKIPEDIA. Yuv. <https://en.wikipedia.org/wiki/YUV>. Visualisé le 25-01-2021. vi, 68
- [176] WILLIAMSON, T. A. *A High-Performance Stereo Vision System for Obstacle Detection*. PhD thesis, Carnegie Mellon University, 1998. 140
- [177] WOLKOW, S., ANGERMANN, M., DEKIERT, A., AND BESTMANN, U. Model-based threshold and centerline detection for aircraft positioning during landing approach. In *Proceedings of the ION 2019 Pacific PNT Meeting* (05 2019), pp. 767–776. 19

- [178] WU, B., IANDOLA, F., JIN, P. H., AND KEUTZER, K. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), pp. 129–137. 145
- [179] WU, H., ZHANG, X., STORY, B., AND RAJAN, D. Accurate vehicle detection using multi-camera data fusion and machine learning. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 3767–3771. 41
- [180] XIAO, C., AND GAN, J. Fast image dehazing using guided joint bilateral filter. *The Visual Computer* (2012). 174, 176
- [181] XU, H., AND LI, H. Study on a robust approach of lane departure warning algorithm. *International Conference on Signal Processing Systems* (2010). 100
- [182] YIN, S., CAO, L., LING, Y., AND JIN, G. One color contrast enhanced infrared and visible image fusion method. *Infrared Physics & Technology* 53, 2 (2010), 146–150. 185, 186
- [183] YU-WING TAI, JIAYA JIA, AND CHI-KEUNG TANG. Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 1, pp. 747–754 vol. 1. 185
- [184] YUAN, C., MEDIONI, G., KANG, J., AND COHEN, I. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *IEEE transactions on pattern analysis and machine intelligence* 29, 9 (2007), 1627–1641. 139
- [185] ZAITOUN, N., AND AQEL, M. Survey on image segmentation techniques. *Procedia Computer Science* 65 (12 2015), 797–806. 86
- [186] ZHANG, H., AND PATEL, V. M. Densely connected pyramid dehazing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 3194–3203. 178
- [187] ZHANG, L., YIN, Z., ZHAO, K., AND TIAN, H. Lane detection in dense fog using a polarimetric dehazing method. *Appl. Opt.* 59, 19 (Jul 2020), 5702–5707. 179
- [188] ZHANG, R., ISOLA, P., AND EFROS, A. A. Colorful image colorization. In *European conference on computer vision* (2016), Springer, pp. 649–666. 187
- [189] ZHENFENG, S., JUN, L., AND QIMIN, C. Fusion of infrared and visible images based on focus measure operators in the curvelet domain. *Applied optics* 51, 12 (2012), 1910–1921. 186
- [190] ZHU, Q., YEH, M.-C., CHENG, K.-T., AND AVIDAN, S. Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (2006), vol. 2, IEEE, pp. 1491–1498. 144

- [191] ZUIDERVELD, K. *Contrast Limited Adaptive Histogram Equalization*. Academic Press Professional, Inc., USA, 1994, p. 474–485. 35
- [192] ZUO, C., CHEN, Q., LIU, N., REN, J., AND SUI, X. Display and detail enhancement for high-dynamic-range infrared images. *Optical Engineering* 50, 12 (2011), 1 – 10. 35

# Annexe A

## Représentation des résultats des différences de couleur pour les couples (*EspaceCouleur*, *FonctionDistance*)

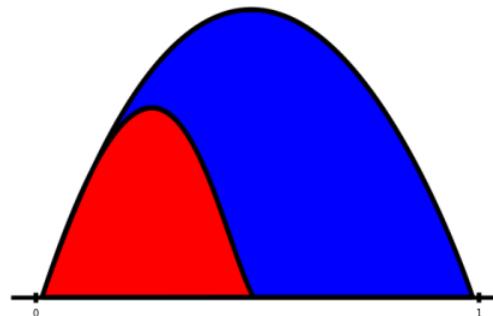


Figure A.1: Schéma explicatif des Figures de l'Annexe A. Répartition des résultats de la différence de couleur entre les couleurs d'une classe et la couleur de référence. La totalité des différences de couleur est notée en bleu et la répartition pour une classe donnée est notée en rouge.

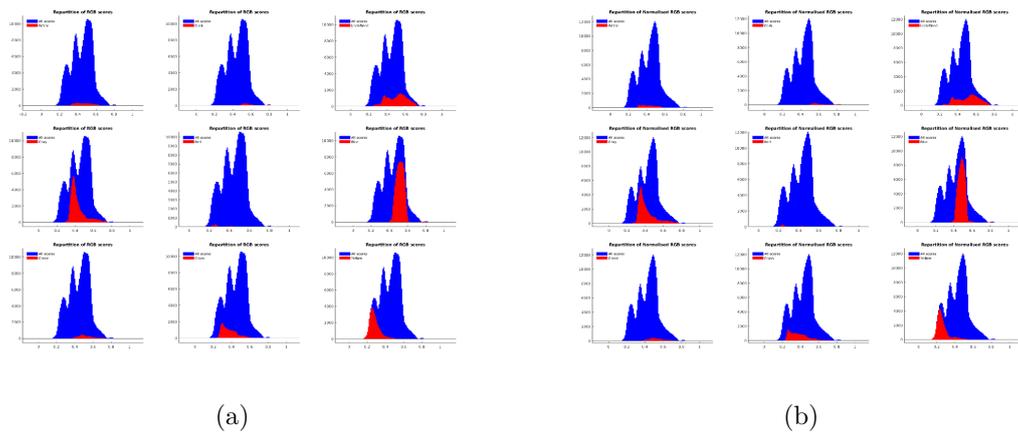


Figure A.2: (a) Distance euclidienne et (b) Distance euclidienne pondérée sur les canaux R, G et B.

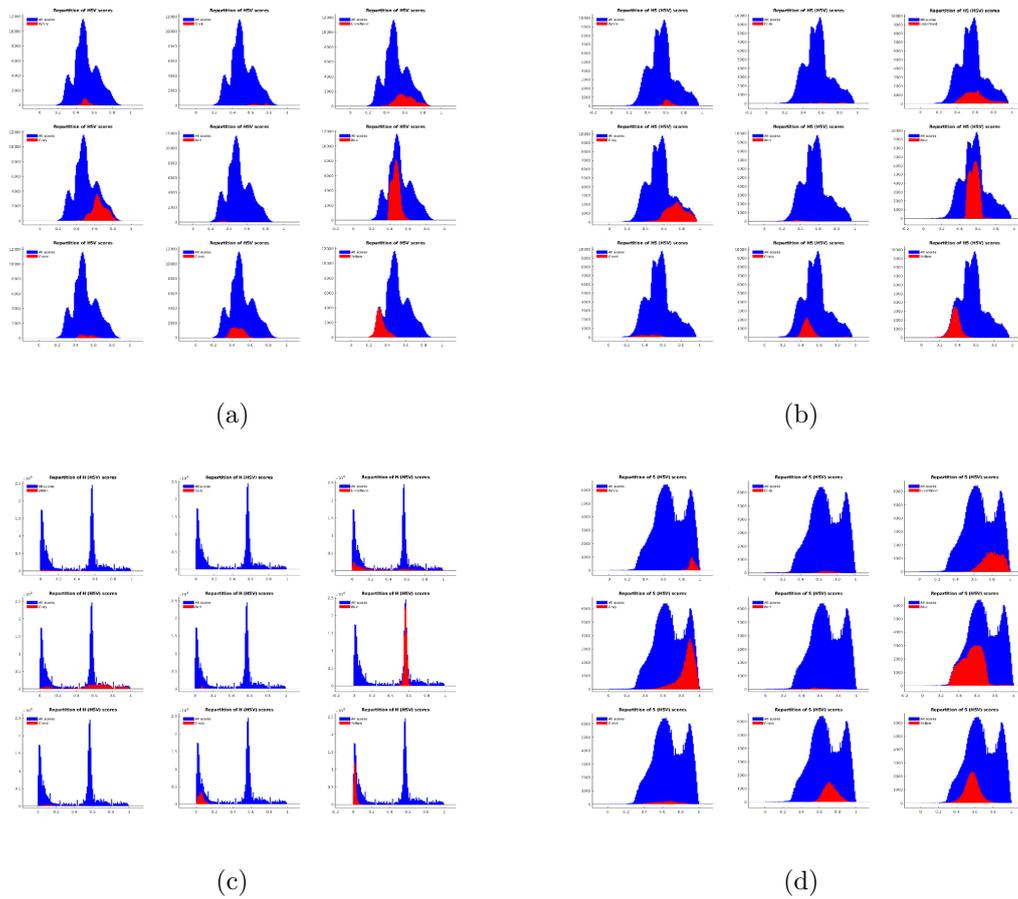


Figure A.3: Distance euclidienne sur (a) les trois canaux H, S et V, (b) les canaux H et S, (c) le canal H, (d) le canal S.

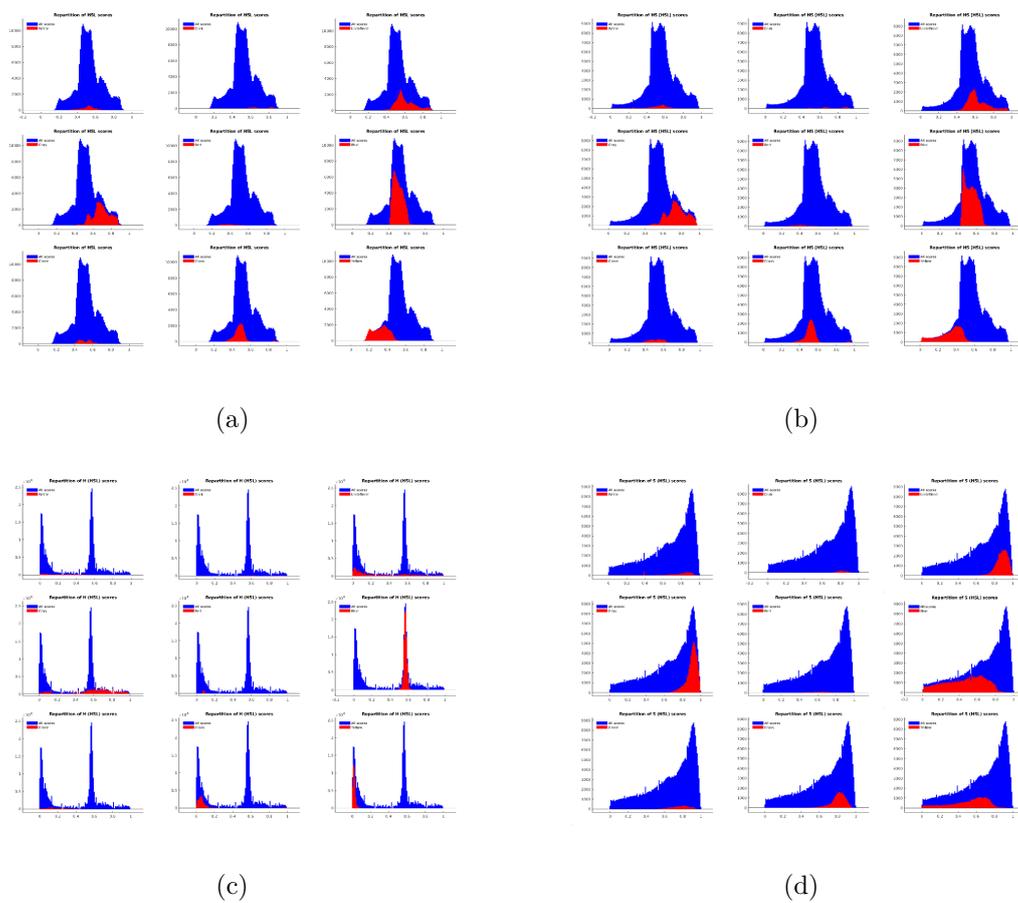


Figure A.4: Distance euclidienne sur (a) les trois canaux H, S et L, (b) les canaux H et S, (c) le canal H, (d) le canal S.

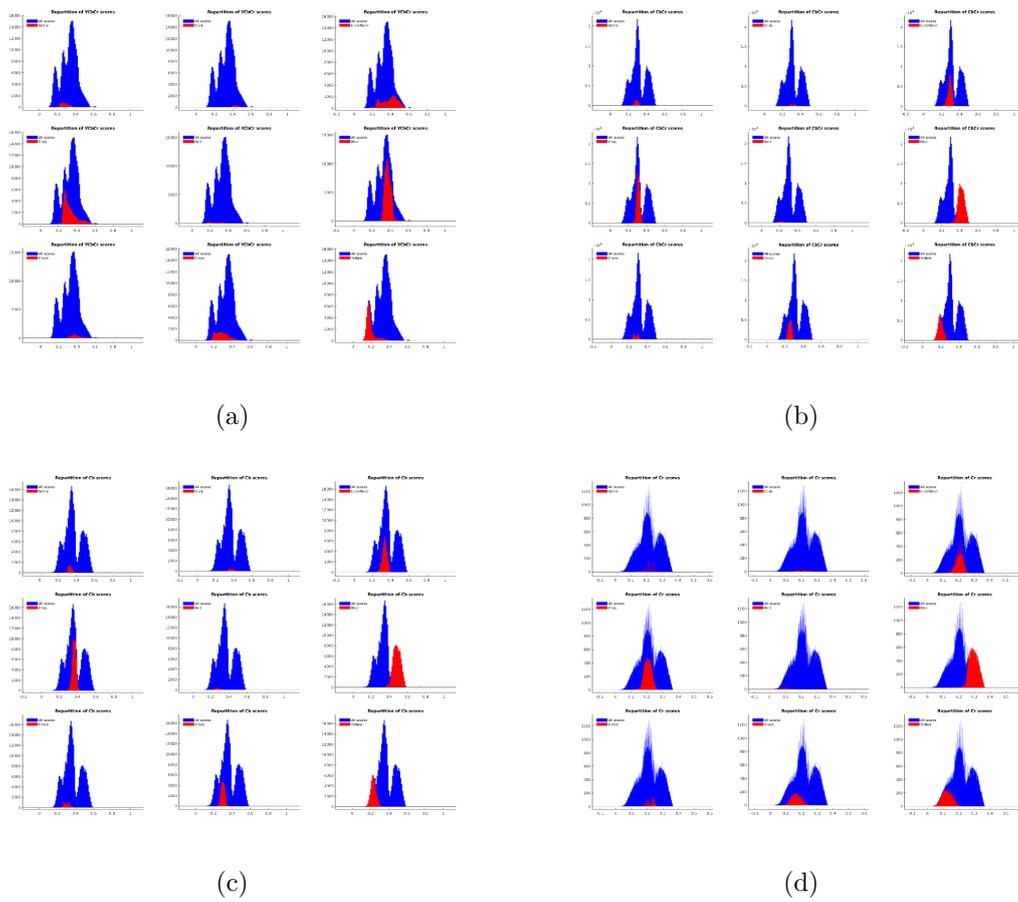


Figure A.5: Distance euclidienne sur (a) les trois canaux Y, Cb et Cr, (b) les canaux Cb et Cr, (c) le canal Cb, (d) le canal Cr.

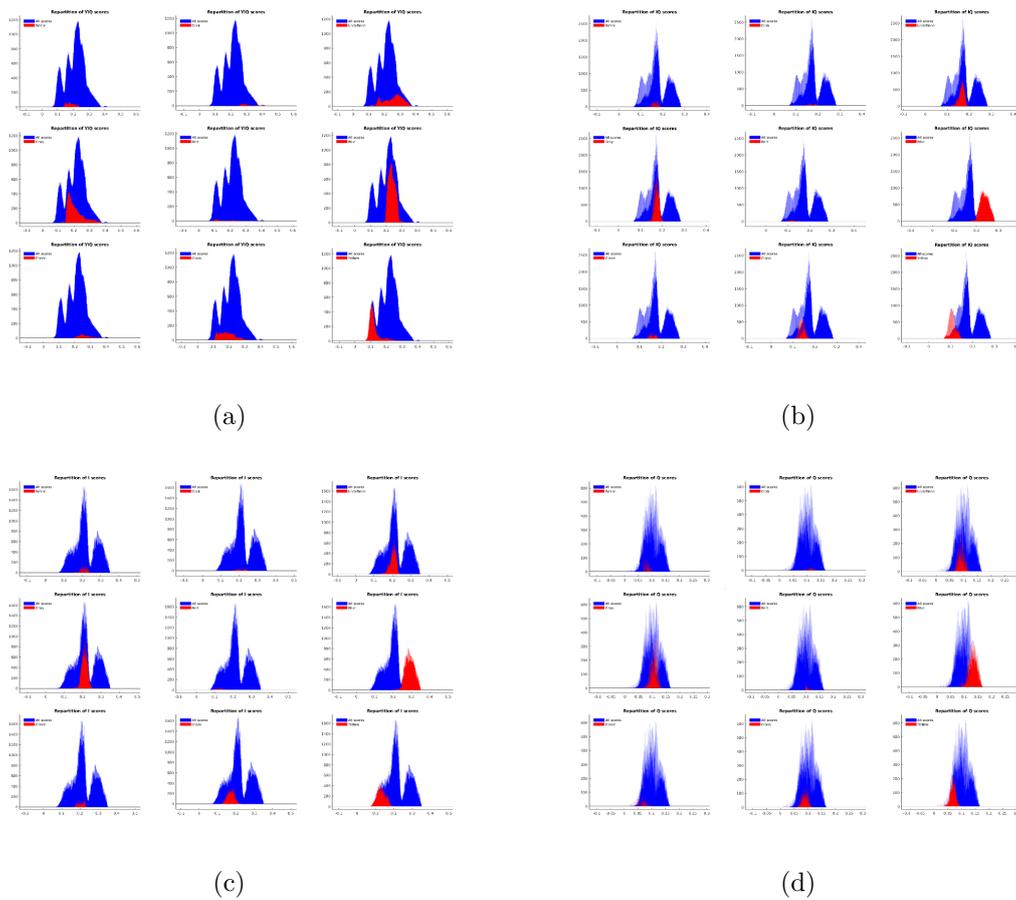


Figure A.6: Distance euclidienne sur (a) les trois canaux Y, I et Q, (b) les canaux I et Q, (c) le canal I, (d) le canal Q.

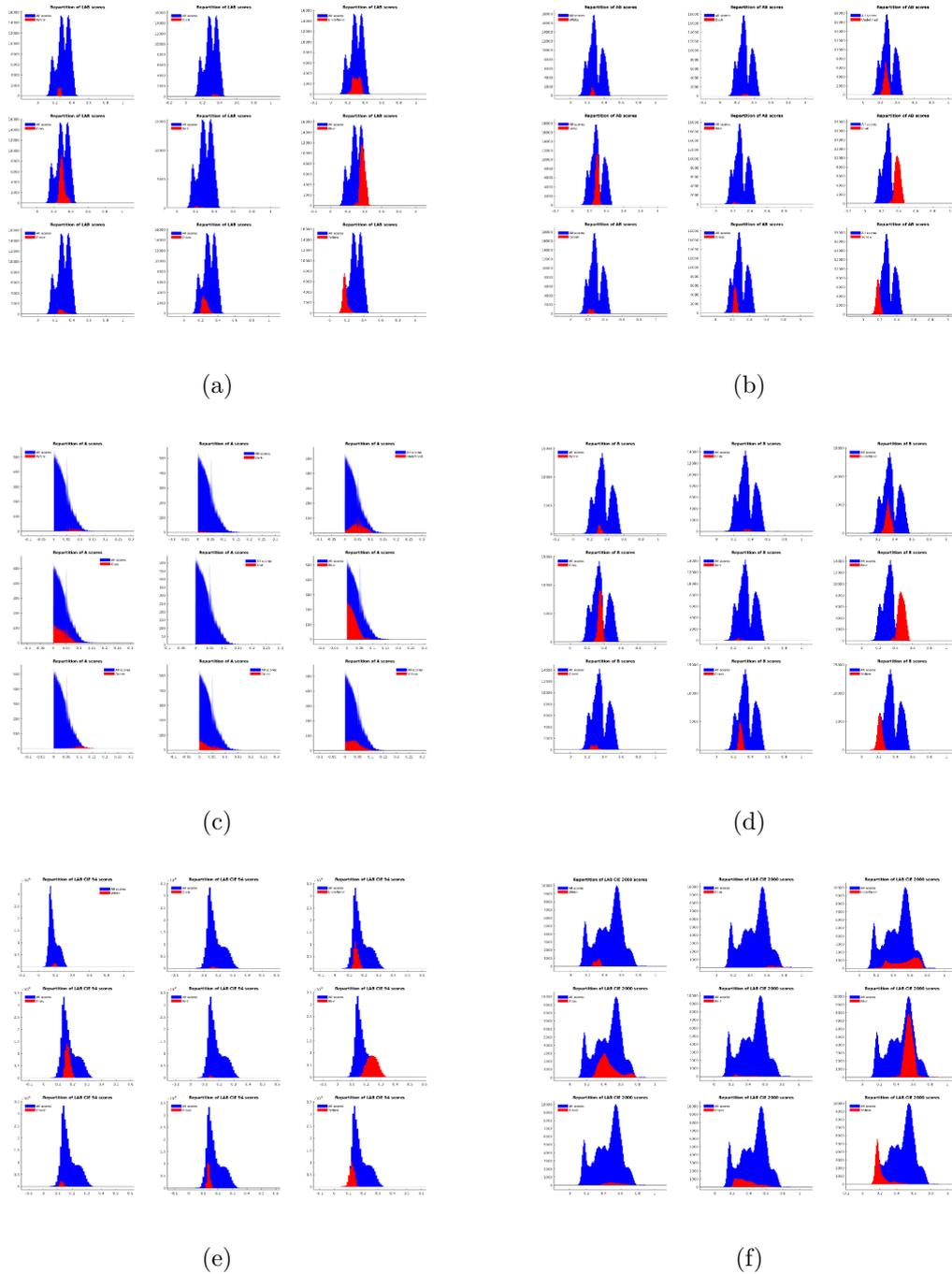


Figure A.7: Distance euclidienne sur (a) les trois canaux de L,  $a^*$  et  $b^*$ , (b) les canaux  $a^*$  et  $b^*$ , (c) sur le canal  $a^*$ , (d) sur le canal  $b^*$ . (e) Distance CIE94 sur les canaux L,  $a^*$  et  $b^*$  et (f) Distance CIE94 sur les canaux L,  $a^*$  et  $b^*$ .

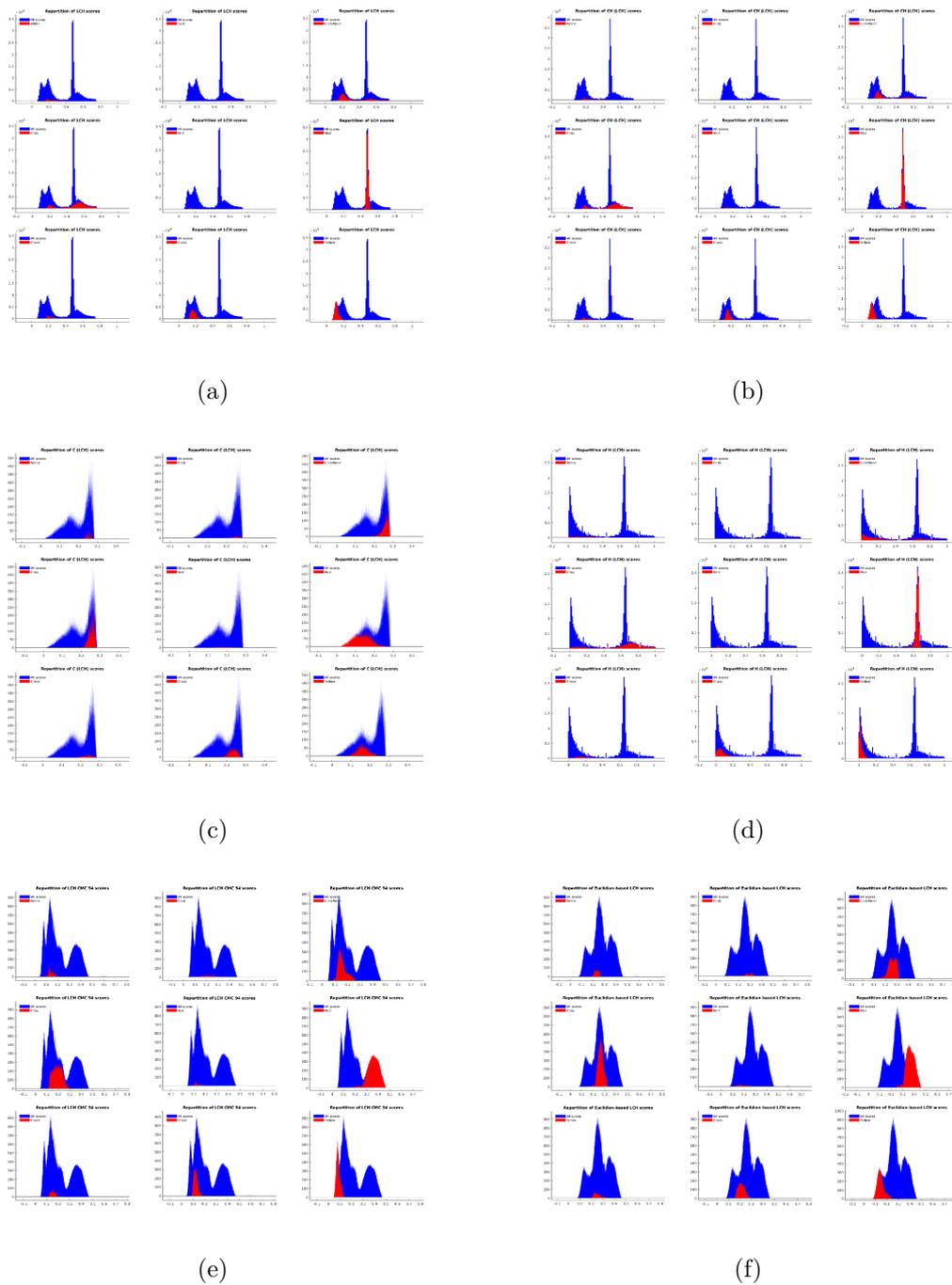


Figure A.8: Distance euclidienne sur (a) les trois canaux de L, C et H, (b) les canaux C et H, (c) sur le canal C, (d) le canal H. (e) Distance CIE94 sur les canaux L,  $a^*$  et  $b^*$  et (f) Distance CIEDE2000 sur les canaux L,  $a^*$  et  $b^*$ .

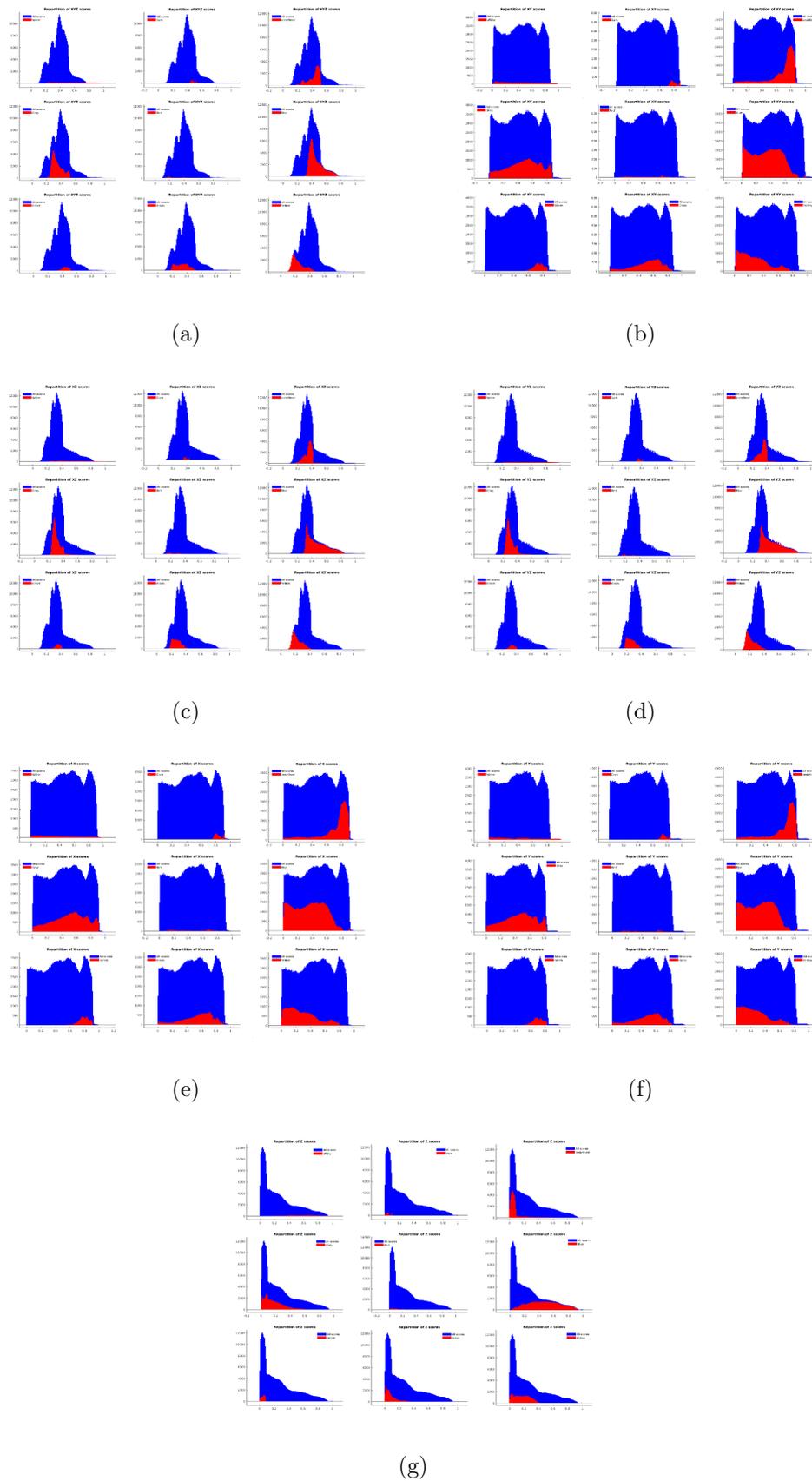


Figure A.9: Distance euclidienne sur (a) les trois canaux X, Y et Z, (b) les canaux X et Y, (c) sur les canaux X et Z, (d) les canaux Y et Z, (e) le canal X, (f) le canal Y, (g) le canal Z.

## Résumé - Abstract

L'imaginaire commun est emprunt de l'idée du véhicule autonome depuis de nombreuses années. On peut observer de multiples avancées dans le domaine automobile comme par exemple les navettes autonomes ou certains modèles de voitures, bien qu'encore en phase de test. De la même façon, les industriels de l'aéronautique cherchent à développer l'avion autonome de demain. Une grande partie des fonctionnalités sont déjà réalisées de manière automatique mais il reste des progrès à apporter lors des phases de roulage sur le tarmac.

Cette thèse est centrée sur l'utilisation d'informations provenant de différents types de capteurs dans le but d'apporter une aide au pilote pour la navigation de l'aéronef lors du roulage, pour tout type de conditions météorologiques. Pour cela, nous considérons deux fonctions essentielles : la détection de motifs sur une scène aéroportuaire et la détection d'obstacles. Nous utilisons deux types de caméras, dans le visible et dans l'infrarouge, montées à plusieurs endroits sur l'aéronef. Avant de traiter de ces fonctions, des pré-traitements sont nécessairement appliqués aux données fournies par les différents capteurs pour maximiser les performances des algorithmes d'aide à la navigation. Cette partie de la thèse aborde les méthodes de fusion de l'information, notamment au niveau de la dynamique de l'image, et propose une étude colorimétrique des images visibles en comparaison avec les normes aéronautiques. Une fois les pré-traitements effectués, la solution proposée repose, dans un premier temps, sur une méthode d'analyse de scène à base de filtre particulaire afin d'extraire des motifs d'intérêt (marquages au sol, balises, délimitations tarmac-herbe, ...). Cette méthode a été comparée à des approches que l'on retrouve classiquement dans le domaine automobile et pour lesquelles on peut s'interroger sur la difficulté de portage dans le domaine aéronautique. Par la suite, les motifs que l'on extrait sont utilisés pour décrire le déplacement de l'aéronef et estimer la présence d'obstacles potentiels. Cette information est, pour l'instant, remontée au pilote mais a vocation à nourrir un algorithme décisionnel dans le cas de la navigation d'un avion autonome. Au cours de cette thèse, une attention est portée sur la fusion d'informations provenant d'une caméra visible et d'une caméra infrarouge afin de supporter des cas de navigation plus complexes comme les phases nocturnes ou lors d'épisodes de brouillard.

---

The notion of autonomous vehicle has been deep-rooted in the collective imagination for a good number of years. We can observe many advances in the automotive field such as autonomous shuttles or current test phases on personal vehicles and taxis. Likewise, aeronautics manufacturers are seeking to develop the autonomous aircraft of the future. A large part of the functionalities are already carried out automatically, but there is still room for improvement during the taxiing phases on the tarmac.

This thesis is focused on the use of information coming from different types of sensors in order to provide an aid to the navigation of the aircraft during taxiing, for all types of weather conditions. For this, we consider two essential functions: the detection of patterns on an airport scene and the detection of obstacles. We use two types of cameras, visible and infrared, mounted at several locations on the aircraft. Before discussing these functions, pre-processing is necessarily applied to the data supplied by the various sensors in order to maximize the performance of the navigation aid algorithms. This part of the thesis deals with methods for the fusion of information, particularly in terms of image dynamics, and proposes a colorimetric study of visible images in comparison with aeronautical standards. Once the pre-treatments have been applied, the proposed solution performs, initially, a method of scene analysis based on a particle filter in order to extract patterns of interest (ground markings, beacons, tarmac-grass boundaries, ...). This method is compared to other algorithms that are conventionally found in the automotive field and for which one can question the difficulty of adaptation to the aeronautical field. Thereafter, the patterns that are extracted are used, by another stage of the proposed solution, to describe the movement of the aircraft and to estimate the presence of potential obstacles. This information is, for the moment, returned to the pilot but is intended to feed a decision-making algorithm in the case of autonomous airplane navigation. During this thesis, an attention is paid to the fusion of information coming from a visible camera and an infrared camera in order to support more complex navigation cases such as nocturnal phases or during episodes of fog.

