

x 50376
1995
185

N° d'ordre : 1563

Année 1995

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Pour l'obtention du titre de

DOCTEUR

en Productique : Automatique et Informatique Industrielle

par

Jean-Christophe BURIE

MISE EN CORRESPONDANCE

D'IMAGES LINEAIRES STEREOSCOPIQUES

APPLICATION A LA DETECTION D'OBSTACLES

A L'AVANT DES VEHICULES ROUTIERS

Soutenue le 28 septembre 1995 devant le jury composé de :

MM.

P. VIDAL

B. DUBUISSON

P.-L. WENDEL

M. de SAINT BLANCARD

L. BARANES

J.-G. POSTAIRE

Président

Rapporteur

Rapporteur

Examineur

Examineur

Directeur de thèse

Professeur à l'U.S.T.L.

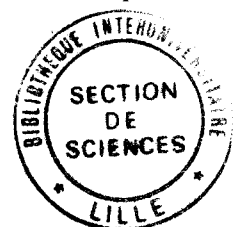
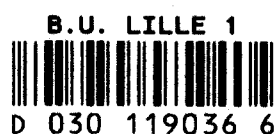
Professeur à l'U.T.C. de Compiègne

Professeur à l'E.S.P. de Strasbourg

Ingénieur chez P.S.A. Peugeot-Citroën

Directeur de l'I.N.R.E.T.S. Villeneuve d'Ascq

Professeur à l'U.S.T.L.



AVANT-PROPOS

Le travail présenté dans ce mémoire a été réalisé au Centre d'Automatique de l'Université des Sciences et Technologies de Lille 1 dans le cadre du programme européen Eurêka Prometheus. Celui-ci n'aurait pu aboutir sans l'aide de nombreuses personnes liées ou non à ce projet.

Je tiens à exprimer ma profonde reconnaissance à Monsieur Pierre Vidal, professeur à l'U.S.T.L., directeur du Centre d'Automatique de Lille, pour m'avoir accueilli au sein de son laboratoire et fait l'honneur de présider le jury de cette thèse.

Je tiens à exprimer toute ma gratitude à Monsieur Jack-Gérard Postaire, professeur à l'U.S.T.L., qui a été mon directeur de recherche pendant toutes ces années. Ses conseils, ses constants encouragements et sa grande disponibilité m'ont permis de mener à bien ce travail.

Je remercie Messieurs Bernard Dubuisson et Pierre-Louis Wendel, respectivement professeurs à l'Université de Technologie de Compiègne et à l'Ecole Supérieure de Physique de Strasbourg d'avoir accepté de juger mon travail.

J'adresse mes remerciements à Madame Marie de Saint Blancard, ingénieur chez P.S.A. Peugeot Citroën, pour la confiance qu'elle a témoigné à l'équipe du Centre d'Automatique de Lille, l'aide qu'elle nous a apportée durant les travaux effectués dans le cadre de Prometheus et, pour avoir accepté de faire partie du jury de cette thèse.

Je tiens également à remercier Monsieur Lionel Baranes, directeur délégué de l'I.N.R.E.T.S. - Centre de Villeneuve d'Ascq, qui me fait l'honneur de participer à ce jury.

Que Monsieur Yves David, ancien directeur de l'I.N.R.E.T.S. / C.R.E.S.T.A. soit remercié pour la confiance qu'il m'a accordée, l'aide et les encouragements qu'il m'a apportés durant mes travaux. Je remercie également Jean-Pierre Deparis, chargé de recherches à l'I.N.R.E.T.S. de Villeneuve d'Ascq pour ses conseils et son aide.

Mes remerciements vont également à Jean-Luc Bruyelle, chercheur au Centre d'Automatique de Lille, pour tout le travail effectué en commun, les figures du chapitre 2 extraites de sa thèse et les photographies de Prolab 2.

J'adresse également mes remerciements aux chercheurs de l'équipe Image et Décision ainsi qu'aux membres du Centre d'Automatique de Lille et tout particulièrement à Christophe Vieren, mon directeur sportif et entraîneur personnel, Luc Duvieubourg, Ludovic Macaire, Patrick Vannoorenberghe et François Cabestaing pour leur amitié et tous les kilomètres passés ensemble.

Je remercie également Michèle Rombaut, Omar Aboukhaled, Johan Amiard, les autres membres du laboratoire Heudiasyc (Heuristique et Diagnostic des Systèmes Complexes) de l'Université de Technologie de Compiègne ainsi que ceux des équipes ayant participé à Prolab 2 pour tout le travail fait en commun.

Je tiens enfin à remercier mon épouse Isabelle et ma mère pour leur disponibilité, leur patience et sans qui je n'aurais pu terminer ce mémoire à temps.

Ce travail a été financé par le projet Eurêka PROMETHEUS. Dans le cadre de ce projet, je tiens à remercier les constructeurs automobile P.S.A. Peugeot Citroën et Renault pour leur aide et leur soutien. Je remercie également l'I.N.R.E.T.S. et la Région Nord-Pas de Calais pour la bourse de recherche conjointe qu'ils m'ont accordée.

**TABLE
DES
MATIERES**

Chapitre 1 : INTRODUCTION	1
1-1. De la vision humaine à la vision artificielle	3
1-1.1. Qu'est-ce que la vision?	3
1-1.2. Le système binoculaire.....	4
1-1.3. La vision artificielle	4
1-2. La perception du relief	5
1-2.1. Restitution du relief.....	5
1-2.2. La vision artificielle tridimensionnelle.....	6
1-2.3. Applications de la stéréovision.....	7
1-2.3.1. Robotique mobile.....	7
1-2.3.2. La détection d'obstacles.....	7
1-3. Le programme de recherche Européen PROMETHEUS	7
1-3.1. Origine et objectif du programme	7
1-3.2. Axes de recherches.....	8
1-3.3. Le sous-programme ProArt.....	9
1-3.4. Les démonstrateurs communs européens.....	9
1-4. Organisation du mémoire	10
Chapitre 2 : STEREOSCOPIE LINEAIRE	13
2-1. Réflexions sur la nécessité de la détection d'obstacles sur route	15
2-1.1. Pourquoi tant d'intérêt pour la détection d'obstacles?	15
2-1.1.1. Importance de l'automobile	15
2-1.1.2. Les accidents de la route	15
2-1.1.3. Amélioration de la sécurité routière.....	16
2-1.2. Méthodes existantes de détection d'obstacles	16
2-1.2.1. Méthodes actives.....	16
2-1.2.2. Méthodes passives.....	18
2-1.3. Approche nouvelle	20
2-2. Conception d'un dispositif de prise de vue stéréoscopique linéaire.....	20
2-2.1. Les caméras linéaires	20
2-2.1.1. Constitution d'une caméra linéaire	20
2-2.1.2. Champ de visée d'une caméra linéaire.....	21
2-2.2. Principe du stéréoscope linéaire.....	22
2-2.2.1. Hypothèses.....	22
2-2.2.2. Installation des caméras sur le véhicule	22
2-2.2.3. Reconstitution de la profondeur et de la position.....	24
2-2.3. Réalisation électronique du stéréoscope linéaire.....	27
2-2.3.1. Caméras.....	27
2-2.3.2. Acquisition simultanée des images	27
2-2.4. Réalisation mécanique du stéréoscope linéaire embarqué	29
2-2.4.1. Contraintes	29
2-2.4.2. Monture d'une caméra	30
2-2.4.3. Monture commune	30
2-3. Calibrage du stéréoscope linéaire.....	31
2-3.1. Problèmes et contraintes.....	31
2-3.1.1. Problème à résoudre par le calibrage	31
2-3.1.2. Contrainte.....	31
2-3.2. Mire de calibrage.....	32
2-3.3. Procédure de calibrage	33
2-3.4. Précision du stéréoscope linéaire	33
2-4. Premiers essais du stéréoscope linéaire	36
2-4.1. Enregistrement de séquences stéréoscopiques linéaires.....	36
2-4.2. Caractéristiques des différents stéréoscopes linéaires	37
2-4.3. Exemples de séquences enregistrées	38
2-4.3.1. Exemple 1 : Piéton évoluant devant le stéréoscope	38
2-4.3.2. Exemple 2 : Route dégagée et passage d'une sortie d'autoroute	39
2-5. Conclusion.....	40

Chapitre 3 : PRINCIPES FONDAMENTAUX DE LA STEREOVISION	41
3-1. Approche générale.....	43
3-1.1. Le Paradigme de Marr.....	43
3-1.2. Organisation d'un système stéréoscopique	44
3-1.3. Mise en correspondance.....	44
3-2. Prise de vue	45
3-2.1. Les capteurs.....	45
3-2.2. Géométrie des capteurs	45
3-2.3. Caractéristiques supplémentaires d'un dispositif de prise de vue	47
3-3. Primitives stéréoscopiques	47
3-3.1. Notion de primitives.....	47
3-3.2. Type de primitives.....	48
3-3.2.1. Primitives de type point	49
3-3.2.1.1. Points de Moravec.....	49
3-3.2.1.2. Les points de courbure ou coins.....	49
3-3.2.2. Primitives de type contour	50
3-3.2.2.1. Points de contour	50
3-3.2.2.2. Modélisation du contour à partir des points de contour	50
3-3.2.3. Primitives de type région	51
3-3.2.3.1. Régions homogènes	51
3-3.2.3.2. Régions obtenues par segmentation couleur.....	51
3-3.2.4. Primitives de haut niveau : les contours actifs	52
3-3.3. Conclusion sur les primitives	52
3-4. Contraintes de mise en correspondance	53
3-4.1. Utilité des contraintes.....	53
3-4.2. Contrainte épipolaire.....	53
3-4.3. Contraintes locales	53
3-4.3.1. Contrainte d'orientation	54
3-4.3.2. Contrainte de continuité figurale.....	54
3-4.3.3. Contrainte de continuité de la disparité.....	55
3-4.4. Contraintes globales	55
3-4.4.1. Contrainte d'ordre	55
3-4.4.2. Contrainte d'unicité.....	57
3-4.5. Conclusion sur les contraintes.....	58
3-5. Approches originales pour percevoir le relief	59
3-5.1. La vision dynamique.....	59
3-5.1.1. Principe de la vision dynamique	59
3-5.1.2. Les différents types de méthodes	60
3-5.1.2.1. Approche par mise en correspondance	60
3-5.1.2.2. Méthodes de vision dynamique sans mise en correspondance	61
3-5.1.3. Conclusion sur la vision dynamique	61
3-5.2. La vision trinoculaire	62
3-5.2.1. Utilité d'une troisième caméra.....	62
3-5.2.2. Géométrie d'un dispositif trinoculaire	62
3-5.2.3. Algorithme de mise en correspondance	63
3-5.2.4. Conclusion	64
3-6. Mise en correspondance	65
3-6.1. Aspects fondamentaux	65
3-6.1.1. Formalisme de la fonction d'appariement.....	65
3-6.1.2. Combinatoire de la fonction d'appariement.....	66
3-6.1.3. Dilemme de la mise en correspondance.....	66
3-6.2. Critère de ressemblance	67
3-6.2.1. Pourquoi définir un critère de ressemblance?	67
3-6.2.2. Nature des critères.....	67
3-6.2.2.1. Les critères photométriques	68
3-6.2.2.2. Les critères géométriques.....	68
3-6.2.3. Exemples de critères	68

3-6.2.3.1. Exemples de critère de corrélation.....	68
3-6.2.3.2. "Pseudo-corrélation"	69
3-6.2.4. Conclusion	70
3-6.3. Algorithmes de mise en correspondance.....	70
3-6.3.1. Méthodes basées sur la corrélation.....	70
3-6.3.2. Méthodes basées sur la programmation dynamique	71
3-6.3.3. Méthodes basées sur la relaxation.....	72
3-6.3.4. Méthodes basées sur les réseaux de neurones.....	73
3-6.3.5. Méthodes hiérarchiques ou pyramidales.....	74
3-6.3.6. Méthodes de description structurelle (isomorphisme de graphe).....	75
3-6.3.7. Méthodes de prédiction et vérification d'hypothèses.....	76
3-6.3.8. Méthodes d'appariement sous contraintes	77
3-6.4. Conclusions sur les méthodes de mise en correspondance.....	77
3-7. Travaux existants en stéréovision linéaire.....	78
3-7.1. Système de guidage d'un robot mobile proposé par Inigo.....	78
3-7.1.1. Guidage du robot.....	78
3-7.1.2. Détection d'obstacle.....	78
3-7.1.3. Expérimentation et résultats.....	79
3-7.1.4. Conclusion sur la méthode proposée par Inigo	79
3-7.2. Système stéréoscopique de Colle destiné à la robotique mobile.....	80
3-7.2.1. Extraction de primitives stables	80
3-7.2.2. Mise en correspondance.....	80
3-7.2.3. Expérimentation et résultats.....	81
3-7.2.4. Conclusion sur l'approche de Colle	81
3-7.3. Approche réalisée par J.-L. Bruyelle.....	81
3-7.3.1. Traitements réalisés.....	81
3-7.3.2. Conclusion	82
3-8. Conclusion.....	82
Chapitre 4 : EXTRACTION DES PRIMITIVES.....	85
4-1. Choix de primitives en vision linéaire.....	87
4-1.1. Passage du matriciel au linéaire	87
4-1.1.1. Primitives de type point	87
4-1.1.2. Primitives de type région	88
4-1.1.3. Primitives de type contour	89
4-1.1.4. Primitives de haut niveau : les contours actifs	90
4-1.2. Primitives possibles en vision linéaire	90
4-1.3. Dualité Régions-Contours	91
4-1.4. Conclusion sur le choix du type de primitive.....	91
4-2. Choix du détecteur de points de contour.....	91
4-2.1. Contraintes à respecter par le détecteur de points de contour	92
4-2.1.1. Détection de tous les points de contour.....	92
4-2.1.2. Localisation précise de la position des points de contour	93
4-2.1.3. Ne pas créer de faux points de contour	97
4-2.1.4. Rapidité d'application	98
4-2.2. Réduction de la quantité de bruit contenu dans l'image.....	98
4-2.3. Organisation du détecteur de points de contour	98
4-3. Détection des points de contour	99
4-3.1. Les masques de convolution	100
4-3.1.1. Définitions.....	100
4-3.1.2. Pré-traitement : opération de lissage	101
4-3.1.3. Détecteurs de contour.....	101
4-3.1.4. Conclusion sur les masques de convolution.....	102
4-3.2. Détecteurs optimaux.....	103
4-3.2.1. Définition d'un détecteur optimal	103
4-3.2.2. Approche de Canny.....	103
4-3.2.3. Approche de Deriche	105
4-3.2.3.1. Détecteur optimal de contour.....	105

4-3.2.3.2. Opérateur de lissage	106
4-3.2.3.3. Influence du paramètre α	106
4-3.2.3.4. Conclusion	107
4-3.3. Implantation de la détection de contour	108
4-3.3.1. Implantation récursive d'un filtre	108
4-3.3.2. Lissage récursif	110
4-3.3.3. Calcul récursif du gradient	112
4-3.3.4. Conclusions sur le détecteur de contour	114
4-3.4. Détecteur optimal de Y. WAN	115
4-4. Extraction des points de contour	115
4-4.1. Exemples de méthodes de segmentation	115
4-4.1.1. Seuillage simple	115
4-4.1.2. Seuillage par hystérésis [Canny 83]	116
4-4.1.3. Méthode basée sur l'analyse d'histogramme	116
4-4.1.4. Autres méthodes de segmentation	117
4-4.2. Problèmes à résoudre lors du seuillage	117
4-4.2.1. Le problème du seuil	117
4-4.2.2. Les différents extremums locaux	117
4-4.2.3. Les extremums manquants	118
4-4.3. Extraction des points de contour par intervalle	118
4-4.3.1. Elimination des extremums générés par le bruit	118
4-4.3.2. Sélection des extremums significatifs par intervalle	118
4-4.3.3. Algorithme d'extraction des points de contour	122
4-4.4. Conclusion sur l'extraction des points de contour	122
4-5. Résultats de l'extraction des primitives	124
4-5.1.1. Piéton évoluant devant le stéréoscope	124
4-5.1.2. Route dégagée et passage d'une sortie d'autoroute	125
4-6. Attributs des primitives	126
4-7. Conclusion	126

Chapitre 5 : MISE EN CORRESPONDANCE D'IMAGES LINEAIRES STEREOSCOPIQUES

129

5-1. Introduction	131
5-2. Remarques préliminaires	131
5-2.1. Définitions	131
5-2.2. Représentation graphique	132
5-3. Contraintes de mise en correspondance	133
5-3.1. Contrainte d'unicité	134
5-3.2. Contrainte d'ordre	134
5-3.3. Contrainte épipolaire	135
5-3.4. Contrainte de signe du gradient	137
5-3.5. Conclusion	138
5-4. Algorithme séquentiel de mise en correspondance	139
5-4.1. Approche intuitive de la méthode	139
5-4.1.1. Exemple 1	139
5-4.1.2. Exemple 2	139
5-4.1.3. Exemple 3	140
5-4.1.4. Interprétations de cette approche	141
5-4.2. Elaboration de la procédure de mise en correspondance	141
5-4.2.1. Principes de base	141
5-4.2.2. Ebauche de la méthode	141
5-4.2.3. Hypothèse et principe fondamentaux	142
5-4.2.4. Réalisation de la procédure de mise en correspondance	142
5-4.3. Mesure de similarité	143
5-4.3.1. Normalisation des intervalles	143
5-4.3.1.1. Recentrage des intervalles	143
5-4.3.1.2. Echantillonnage spatial	144
5-4.3.1.3. Intervalles après normalisation	144

5-4.3.2. Mesure de similarité.....	144
5-4.4. Module de mise en correspondance élémentaire.....	145
5-4.4.1. Recherche du meilleur appariement.....	145
5-4.4.2. Tests d'appariement élémentaire.....	146
5-4.4.3. Règles de décision.....	150
5-4.4.4. Conclusion sur le module d'appariement élémentaire.....	151
5-4.5. Filtrage des paires incompatibles.....	151
5-4.5.1. Objectif du filtrage.....	151
5-4.5.2. Contrainte épipolaire.....	152
5-4.5.3. Contrainte de signe du gradient.....	153
5-4.5.3.1. Analyse du problème.....	153
5-4.5.3.2. Test de compatibilité sur les paires.....	154
5-4.5.3.3. Test de mise en correspondance.....	154
5-4.5.3.4. Interprétation des résultats.....	154
5-4.5.4. Conclusion sur le filtrage.....	156
5-4.6. Condition initiale et condition d'arrêt de l'algorithme.....	156
5-4.6.1. Condition initiale.....	156
5-4.6.2. Condition d'arrêt.....	156
5-4.7. Résultats de la mise en correspondance séquentielle.....	157
5-4.7.1. Piéton évoluant devant le stéréoscope.....	157
5-4.7.2. Route dégagée et passage d'une sortie d'autoroute.....	158
5-5. Mise en correspondance bidirectionnelle.....	159
5-5.1. Inconvénient de l'algorithme séquentiel de mise en correspondance.....	159
5-5.2. Analyse bidirectionnelle.....	159
5-5.3. Analyse rétrograde.....	160
5-5.3.1. Condition initiale.....	160
5-5.3.2. Condition d'arrêt.....	160
5-5.3.3. Contrainte épipolaire.....	160
5-5.4. Comparaison des résultats.....	161
5-5.5. Résultats de la mise en correspondance bidirectionnelle.....	161
5-5.5.1. Piéton évoluant devant le stéréoscope.....	162
5-5.5.2. Route dégagée et passage d'une sortie d'autoroute.....	162
5-5.6. Conclusion.....	163
5-6. Algorithme hiérarchique de mise en correspondance.....	163
5-6.1. Approche intuitive.....	163
5-6.1.1. Première considération.....	163
5-6.1.2. Deuxième considération.....	164
5-6.1.3. Interprétation.....	164
5-6.2. Mise au point de la procédure de mise en correspondance.....	165
5-6.2.1. Principe de base.....	165
5-6.2.2. Description de la méthode.....	165
5-6.3. Définitions supplémentaires.....	167
5-6.3.1. Définitions.....	167
5-6.3.2. Lien avec l'algorithme de mise en correspondance bidirectionnelle.....	168
5-6.4. Algorithme récursif de mise en correspondance hiérarchique.....	168
5-6.4.1. Principe général de notre algorithme.....	168
5-6.4.1.1. Définitions.....	169
5-6.4.1.2. Application à notre algorithme récursif.....	169
5-6.4.1.3. Exploration du graphe par l'algorithme.....	169
5-6.4.1.4. Prise en compte des résultats de l'algorithme récursif.....	171
5-6.4.2. Sélection des primitives de référence.....	172
5-6.4.3. Module d'appariement élémentaire hiérarchique.....	173
5-6.4.4. Conditions initiales.....	174
5-6.4.5. Conditions d'arrêt.....	175
5-6.4.5.1. Condition d'arrêt n° 1.....	175
5-6.4.5.2. Condition d'arrêt n° 2.....	175
5-6.5. Résultats obtenus lors de la mise en correspondance.....	176
5-6.5.1. Piéton évoluant devant le stéréoscope.....	176

5-6.5.2. Route dégagée et passage d'une sortie d'autoroute	177
5-6.6. Conclusion sur l'algorithme hiérarchique de mise en correspondance	177
5-7. Triangulation géométrique	177
5-8. Conclusion.....	179

Chapitre 6 : UTILISATION DE LA SIGNALISATION HORIZONTALE POUR LA MISE EN CORRESPONDANCE 181

6-1. Introduction	183
6-2. Définitions et notations	183
6-2.1. Repères de travail.....	183
6-2.2. "Liste stéréo"	184
6-2.3. "Liste objet".....	185
6-2.4. Notions de couple et de paire	185
6-3. Principe de la méthode de mise en correspondance bi-mode.....	186
6-3.1. Etude du contexte routier	186
6-3.2. Définition de la zone de danger.....	187
6-3.3. Principe de la méthode de mise en correspondance bi-mode.....	188
6-4. Détection des bandes blanches	189
6-4.1. Etude préliminaire.....	189
6-4.2. Paramètres caractéristiques d'une bande blanche	190
6-4.3. Reconstruction des objets ressemblant à une bande blanche	192
6-4.4. Reconnaissance des bandes blanches	193
6-4.4.1. L'amplitude A	193
6-4.4.2. L'épaisseur E	194
6-4.4.3. Les pentes P_1 et P_2	194
6-4.4.4. Les surfaces S_t et S_r	194
6-4.4.5. Règles de décision.....	195
6-4.5. Conclusion sur la détection des bandes blanches	196
6-5. Suivi des bandes blanches	197
6-5.1. Etude préliminaire.....	197
6-5.2. Recherche des bandes blanches dans une fenêtre.....	198
6-5.2.1. Choix de la taille de la fenêtre.....	199
6-5.2.2. Recherche d'un profil ressemblant à une bande blanche	200
6-5.2.3. Sélection du profil correspondant à une bande blanche.....	200
6-5.2.4. Conclusion sur la recherche des bandes blanches	201
6-5.3. Calcul des coordonnées dans le repère (O, X, Z)	202
6-5.4. Gestion partielle des occlusions	202
6-5.5. Disparition et réapparition des bandes blanches	204
6-6. Détermination de la zone de danger	205
6-6.1. Notion de zone d'intérêt.....	205
6-6.2. Précaution à prendre.....	206
6-6.3. Détermination de la zone de danger	206
6-6.3.1. Aucune bande blanche n'a été détectée	207
6-6.3.2. Une seule bande blanche a été détectée	207
6-6.3.3. Au moins deux bandes blanches ont été détectées.....	208
6-6.4. Détermination des zones d'intérêt.....	209
6-6.4.1. Notation utilisée pour les zones d'intérêt.....	209
6-6.4.2. Une seule bande blanche a été détectée	210
6-6.4.3. Au moins deux bandes blanches ont été détectées.....	212
6-7. Mise en correspondance bi-mode.....	214
6-7.1. Mode global	214
6-7.2. Mode local.....	214
6-7.3. Mise en correspondance bi-mode.....	216
6-7.4. Contrainte de cohérence spatiale.....	216
6-8. Conclusion sur la mise en correspondance bi-mode	218

Chapitre 7 : DETECTION D'OBSTACLES A L'AVANT DES VEHICULES	
ROUTIERS	219
7-1. Introduction	221
7-2. Reconstruction tridimensionnelle des objets	221
7-2.1. Difficulté de la reconstruction des objets	221
7-2.2. Sélection des points dans le repère (O, X, Z) avant reconstruction.....	223
7-2.3. Apparition de bandes blanches.....	225
7-2.4. Reconstruction des objets.....	226
7-2.4.1. Hypothèse	226
7-2.4.2. Fusion des points.....	227
7-2.4.3. Fusion des objets.....	228
7-2.5. Nature des objets	229
7-2.6. Sélection des obstacles potentiels.....	230
7-2.6.1. Détermination de la portée maximale Z_{max}	230
7-2.6.2. Sélection des obstacles potentiels	232
7-3. Suivi spatio-temporel	232
7-3.1. Objectif.....	232
7-3.2. Durée de vie	232
7-3.3. Suivi des bandes blanches	233
7-3.4. "Mise en correspondance spatio-temporelle" des objets	234
7-3.4.1. Notion de bande de recherche.....	234
7-3.4.2. Détermination de la bande de recherche	235
7-3.4.3. Recherche de l'objet correspondant dans la bande	237
7-3.4.4. Conclusion sur la mise en correspondance spatio-temporelle.....	237
7-3.5. Suivi temporel des objets	237
7-3.5.1. Apparition d'un objet.....	238
7-3.5.2. Disparition d'un objet	238
7-3.5.3. Confirmation de la présence d'un objet	238
7-3.5.3.1. Les types sont identiques	238
7-3.5.3.2. Les types sont différents	239
7-3.5.4. Conclusion	239
7-3.6. Sélection des obstacles.....	240
7-3.6.1. Elimination du "fond"	240
7-3.6.2. Elimination des objets récents.....	241
7-3.6.3. Localisation des obstacles	241
7-3.6.4. Transmission des obstacles	242
7-3.7. Conclusion sur le suivi spatio-temporel	242
7-4. Résultats expérimentaux.....	243
7-4.1. Détection et suivi des bandes blanches	244
7-4.1.1. Séquence "sortie d'autoroute"	244
7-4.1.2. Route de campagne dégagée	246
7-4.2. Détection des piétons	248
7-4.2.1. Séquence piéton	248
7-4.2.2. Piéton traversant la chaussée.....	249
7-4.2.3. Détection d'un piéton situé dans la zone de danger.....	250
7-4.3. Détection de véhicule	251
7-4.3.1. Suivi d'un véhicule	251
7-4.3.2. Dépassement par un autre véhicule.....	253
7-5. Conclusion.....	256
Chapitre 8 : LE VEHICULE EXPERIMENTAL PROLAB 2	257
8-1. Introduction	259
8-2. Le véhicule expérimental Prolab 2	259
8-2.1. Démonstrateur Prolab1.....	259
8-2.2. Le contexte scientifique	260
8-2.3. L'équipement de Prolab 2	261
8-2.3.1. Les capteurs proprioceptifs	261

8-2.3.2. Les capteurs extéroceptifs	262
8-2.3.3. L'interface homme-machine (IHM).....	262
8-2.3.4. Architecture informatique	264
8-2.4. Fonctionnalités de Prolab 2	264
8-2.5. Architecture globale	265
8-2.5.1. La perception	265
8-2.5.2. L'interface perception/décision.....	265
8-2.5.3. La décision	266
8-2.5.3.1. Analyse de situations	266
8-2.5.3.2. Supervision de manoeuvres	266
8-2.5.3.3. Gestion de la perception.....	266
8-2.5.4. L'information	267
8-2.6. Conclusion sur le véhicule expérimental Prolab 2	267
8-3. Détection des piétons par le véhicule Prolab 2	267
8-3.1. Intégration du stéréoscope linéaire.....	268
8-3.1.1. Stéréoscope linéaire	268
8-3.1.2. Architecture informatique	270
8-3.2. Implantation temps-réel des traitements.....	271
8-3.2.1. Rôle des processeurs	271
8-3.2.1.1. Rôles des processeurs i860	271
8-3.2.1.2. Rôle du processeur 486 du PC-AT	272
8-3.2.2. Parallélisation des tâches sur les i860	272
8-3.2.3. Relation Maître-Esclaves	273
8-3.2.4. Implantation logicielle réalisée pour Prolab 2.....	274
8-3.3. Protocole de communication.....	274
8-3.4. Disponibilité et rafraîchissement des données	275
8-4. Présentation au Board Member's Meeting '94 (BMM '94).....	276
8-4.1. Circuit de Morte-Fontaine	276
8-4.2. Scénario de démonstration	278
8-4.3. Démonstration	278
8-4.4. Commentaires sur les 3 jours de démonstration.....	281
8-5. Autres présentations de Prolab 2	283
8-5.1. Présentations statiques.....	283
8-5.2. Présentation clôturant le programme ProChip	283
8-6. Conclusion.....	285
CONCLUSION.....	287
BIBLIOGRAPHIE.....	293

Chapitre 1

INTRODUCTION

« Toute la conduite de notre vie dépend de nos sens, entre lesquels celui de la vue étant le plus universel et le plus noble. Il n'y a point de doute que les inventions qui servent à augmenter sa puissance ne soient des plus utiles qui puissent être » (Descartes, *La Dioptrique*, 1633). L'être humain s'est toujours efforcé d'améliorer ses conditions de vie. Pour cela, il n'a cessé de faire progresser la science afin de satisfaire ses intérêts. C'est ainsi que pour remédier aux troubles de la vue, il créa, dès le 11^{ème} siècle, de façon rudimentaire la loupe. Puis ses connaissances en optique lui permirent, à partir du 13^{ème} siècle, de mettre au point les lunettes. D'abord grossières, elles ne cessèrent de s'améliorer afin de s'adapter aux différents défauts de l'oeil : presbytie, myopie, etc. Mais l'homme, dans sa soif de connaissances, ne s'est pas arrêté à ses déficiences, il a voulu repousser les limites que lui imposait son système visuel. Ainsi, le monde de l'infiniment petit s'ouvrit à lui dès la fin du 16^{ème} siècle, avec l'invention du microscope par l'opticien hollandais H. Jansen. Galilée, quant à lui, repoussa les limites à l'infini en inventant, en 1609, la lunette astronomique. Les objets les plus lointains étaient alors à " portée d'oeil ".

Lorsque Descartes écrivit les quelques lignes énoncées au début de cette introduction, il pensait sans aucun doute à toutes ces inventions, mentionnées ci-dessus, qui permettent d'augmenter les capacités de notre sens le plus noble qu'est la vue. Cependant il était certainement loin d'imaginer que la science ne s'arrêterait pas là et qu'elle irait même jusqu'à recréer artificiellement le processus de la vision.

1-1. De la vision humaine à la vision artificielle

1-1.1. Qu'est-ce que la vision?

Pour répondre à cette question, nous sommes tentés de dire : la vision c'est voir ce qui nous entoure. En effet, l'Homme est doté d'un organe, l'oeil, chargé de percevoir les phénomènes lumineux afin de recréer une image du monde extérieur. Cette image se forme sur une membrane, la rétine, située dans le fond de l'oeil et constituée de centaines de milliers de cellules nerveuses. L'image projetée sur celle-ci n'est donc ni plus ni moins qu'un nombre impressionnant de points. Mais cette définition de la vision est incomplète, voir c'est également interpréter et comprendre. Comment à partir de cette multitude de points, sommes nous capables d'identifier une chaise ou une automobile?

Si la formation d'une image est une condition nécessaire à la vision, elle n'est pas suffisante. Il faut encore que l'organe récepteur soit solidaire d'un système nerveux suffisamment développé pour traiter l'information visuelle. Les cellules nerveuses composant la rétine ne se contentent pas de recevoir les stimuli lumineux. Celles-ci, directement reliées au cerveau, lui transmettent par l'intermédiaire des fibres du nerf optique les radiations lumineuses sous la forme d'un influx nerveux. Celui-ci interprète alors les sensations reçues, pour nous renseigner sur le monde extérieur.

L'oeil et le cerveau sont ainsi capables de percevoir les formes, les couleurs et les mouvements des objets qui nous entourent. Mais l'Homme possède également la capacité de percevoir les distances grâce à son système visuel binoculaire.

1-1.2. Le système binoculaire

L'homme vit et évolue dans un monde à trois dimensions. C'est sans doute pour cette raison que la nature l'a doté de deux yeux. En effet, la vision binoculaire lui permet de déterminer la géométrie spatiale de son environnement. Il peut ainsi se déplacer, attraper un objet, ou contourner un obstacle en évaluant la taille et la distance des éléments qui l'entourent.

Les images vues par l'oeil droit et par l'oeil gauche ne sont pas identiques. On dit qu'il y a diplopie lorsqu'elles donnent une impression de dédoublement. La diplopie est évitée, pour les images formées sur la partie centrale de la rétine, grâce à une opération cérébrale dite « fusionnement binoculaire » qui nous permet de ne percevoir qu'une seule image.

Pour reconstruire la troisième dimension, le cerveau utilise le décalage existant entre les deux images rétiniennes, appelé disparité stéréoscopique. Il s'agit alors pour celui-ci d'établir des correspondances entre les deux images d'un même objet.

Cependant, on comprend encore assez mal comment ces correspondances sont établies par le cerveau. D'ailleurs ce sujet de recherche constitue un domaine très actif en vision artificielle.

1-1.3. La vision artificielle

La vision artificielle, ou vision par ordinateur, qui s'est développée depuis les années 60, vise à recréer le processus de la vision.

La faculté de voir, c'est à dire reproduire une image du monde extérieur, est maintenant chose aisée. En effet, les progrès de l'électronique ont donné naissance à des systèmes de caméras

qui remplissent ce rôle. La plus grande difficulté de la vision par ordinateur consiste à interpréter ces images.

En effet, notre système visuel, nous permet quotidiennement de reconnaître sans difficulté des milliers d'objets. Chacun de nous est capable d'identifier l'image d'un chien ou d'un chat, même si elle ne nous est présentée que pendant une brève fraction de seconde.

S'il est aujourd'hui possible de concevoir des systèmes de vision capables de reconnaître un nombre limité d'objets dans des conditions bien contrôlées, comme sur une chaîne de production, ils sont toujours très loin de pouvoir effectuer des tâches qui semblent triviales pour chacun de nous, comme reconnaître un objet sous n'importe quel angle de vue.

Malgré ces difficultés, les chercheurs ont d'autres objectifs plus ambitieux : créer des systèmes intelligents dotés d'un système visuel semblable à celui de l'homme.

1-2. La perception du relief

La perception du relief est une fonction fondamentale qui permet de structurer le monde et l'espace environnant, d'évaluer les grandeurs et les distances. Mais cette perception est personnelle, rien ne nous permet de la partager. Le problème auquel furent confrontés, pendant des siècles, les artistes, les architectes ou les astronomes était de trouver le moyen de restituer fidèlement la réalité de l'espace.

1-2.1. Restitution du relief

Autrefois, l'homme ne disposait, pour représenter le monde qui l'entoure, que d'un support à deux dimensions. Mais restituer sur un seul plan ce que l'oeil perçoit en trois dimensions est un exercice difficile.

A l'époque de la Renaissance, les architectes Filippo Brunelleschi, Leon Battista Alberti et le peintre Piero della Francesca furent les premiers à s'intéresser à ce problème. De leurs études, est née la perspective. Celle-ci se propose de reconstituer la profondeur des scènes observées. L'expression du relief est alors donnée par une reconstruction de la réalité selon des règles géométriques parfaitement codifiées. Mais si la perspective permet de représenter fidèlement l'espace, elle ne donne qu'une illusion de la réalité.

La première tentative pour obtenir une image en relief fut réalisée par Charles Wheatstone en 1838. Celui-ci eut l'idée d'utiliser deux images et conçut le premier stéréoscope. Cet appareil

permet d'observer simultanément deux images d'un même objet, prises sous deux angles différents. Il restitue ainsi la profondeur et le relief.

En 1891, Louis Ducos de Hauron développa un nouveau système, les anaglyphes, qui consiste à superposer deux images différentes de la même vue après les avoir teintées l'une en bleu-violet, l'autre en rouge, puis à les regarder à l'aide de lunettes ayant respectivement des verres de mêmes couleurs. Chaque oeil n'aperçoit que l'image qui lui convient et le cerveau assure la fusion des deux photographies pour donner l'illusion d'une seule image en relief.

Si le stéréoscope et les anaglyphes sont toujours employés, l'avenir appartient aux techniques holographiques. Ces techniques utilisent les interférences produites par la superposition de deux faisceaux laser pour reproduire une photographie en relief et permettent ainsi de représenter fidèlement les objets en trois dimensions.

De nos jours, on ne se contente plus de vouloir restituer le relief. Les chercheurs s'intéressent également au moyen de percevoir le relief grâce à la vision artificielle.

1-2.2. La vision artificielle tridimensionnelle

Depuis quelques années, la vision par ordinateur a fait des progrès considérables. Le plus remarquable d'entre eux est la possibilité d'obtenir à partir d'images vidéo, des informations tridimensionnelles.

En effet, il n'est pas possible de déduire la géométrie spatiale d'une scène inconnue à partir d'une seule image de celle-ci. Par contre, dès que l'on dispose de plusieurs images prises sous des angles différents, il devient possible de calculer la position spatiale des points observés dans au moins deux images. On appelle *stéréoscopie* ou *stéréovision* le processus qui permet de combiner entre-elles plusieurs images d'une même scène pour en extraire des informations géométriques tridimensionnelles.

Nous avons vu au § 1-1.3 que la difficulté de la vision artificielle réside dans l'interprétation de l'image. Dans le cas de la stéréovision, celle-ci est encore accrue. En effet, la perception du relief n'est possible que si l'on réalise, comme le fait le cerveau, l'opération de fusionnement des deux images. En vision par ordinateur, l'étape de fusionnement porte le nom de *mise en correspondance* et fait l'objet de nombreuses recherches.

1-2.3. Applications de la stéréovision

Les applications de la stéréovision sont nombreuses. On peut citer par exemple la reconstitution de terrains à partir d'images aériennes, la modélisation de scènes d'intérieur ou d'extérieur ou encore l'analyse d'images biomédicales. Cependant, grâce à la stéréovision, les chercheurs espèrent développer des systèmes intelligents capables de percevoir leur environnement.

1-2.3.1. Robotique mobile

Longtemps, les robots ont été considérés comme des automates programmés une fois pour toutes pour l'exécution de la même tâche répétitive. Cependant un nouveau concept est apparu en robotique. Celui-ci vise à rendre les robots autonomes. Les robots sont donc devenus mobiles par l'adjonction de roues ou de pattes leur permettant de se déplacer. Mais leur véritable autonomie passe d'abord par la possibilité de s'adapter à leur environnement. Grâce à la vision tridimensionnelle, ceux-ci commencent à être capables de construire de manière dynamique des modèles de leur environnement, d'y détecter des changements et donc de réagir intelligemment de manière à pouvoir exécuter leur tâches.

1-2.3.2. La détection d'obstacles

Les déficiences de la vision humaine peuvent avoir de graves conséquences. Les services de la sécurité routière ont estimé qu'un tiers des accidents mortels de la circulation sont dus à des défauts ou à des gênes de la vision. De nombreuses recherches visent à équiper les véhicules de systèmes intelligents capables de détecter les obstacles afin de prévenir suffisamment tôt le conducteur et éviter ainsi les risques collisions. Les systèmes stéréoscopiques apparaissent tout naturellement comme des solutions envisageables puisque, à l'image du système visuel humain, ils ont pour objectif de percevoir et d'analyser leur environnement.

1-3. Le programme de recherche Européen PROMETHEUS

1-3.1. Origine et objectif du programme

Le programme Prometheus est né à Londres, le 20 juin 1986, lors de la conférence Eurêka. Les ministres européens des Affaires Etrangères, de l'Economie et de la Recherche avaient alors un objectif ambitieux en lançant Prometheus, le programme pour un trafic européen de la plus haute efficacité et d'une sécurité inégalée (littéralement « *PROgramM for a European Traffic with Highest Efficiency and Unprecedented Safety* »). Tous les grands constructeurs

automobiles européens (PSA, Renault, Jaguar, BMW, Daimler Benz, Porsche, Fiat, Volvo, etc.) leurs équipementiers (Valeo, Lucas, etc.), et leurs centres d'études se retrouvèrent dans cette communauté de recherche. Il faut également préciser que certains travaux de recherches de base furent confiés aux universitaires. Les objectifs étaient d'améliorer la sécurité, le respect de l'environnement et les performances de la circulation routière grâce à l'emploi des technologies de communication et de traitement de données les plus avancées.

1-3.2. Axes de recherches

Dans le cadre du programme Prometheus, différents axes de recherche ont été examinés dans le but d'améliorer tous les aspects du trafic routier européen. Les principaux axes étudiés sont les suivants :

- Conduite sûre :

Les recherches visent à améliorer la vision de nuit ou par brouillard à l'aide de caméras et d'indications données par affichage tête haute, à développer des systèmes d'aide pour prévenir les risques de collision ou maintenir automatiquement la trajectoire du véhicule. Ce secteur inclus également la mise au point de systèmes destinés à contrôler l'adhérence du véhicule.

- Harmonisation des flux de circulation :

Les scientifiques espèrent une réduction du nombre d'accidents, de la consommation et des émissions nocives. Cette harmonisation doit être atteinte par la communication des véhicules entre eux et par des systèmes intelligents qui veillent au maintien d'un écart de sécurité suffisant entre les véhicules. En cas d'accident, des systèmes automatiques doivent prévenir les secours plus précisément et plus rapidement.

- Gestion de la circulation et des transports :

Des cartes routières numérisées et des informations diffusées par les postes de contrôle du trafic permettent d'amener le conducteur au but par l'itinéraire le plus rapide. Des services d'informations spécialisés aident à choisir le moyen de transport le plus adapté. Les performances dans le domaine du transport des marchandises doivent être encore considérablement augmentées par un réseau informatif reliant le client au camion en passant par l'entreprise.

1-3.3. Le sous-programme ProArt

Le programme Prometheus, qui présente une structure fonctionnelle assez complexe, a été décomposé en un certain nombre de sous-programmes :

- ProCom concerne la communication inter-véhicule.
- ProNet concerne la communication intra-véhicule.
- ProChip concerne le développement de circuits et de matériels particuliers adaptés aux véhicules.

et celui auquel nous avons participé :

- ProArt, qui a eu pour but l'étude et le développement de méthodes issues du domaine de l'intelligence artificielle.

Les programmes ProArt et ProChip s'intègrent dans le groupe des recherches de base (basic researches). Ces sous-programmes ont fait appel à des laboratoires de recherche n'appartenant pas à l'industrie automobile.

Le programme ProArt a développé ses travaux dans deux directions : la vision et le copilottage, afin de concevoir des systèmes d'aide à la conduite.

Le groupe français du sous-programme ProArt est soutenu par le Groupement d'Intérêt Economique PSA-Renault qui a participé activement à l'élaboration de deux démonstrateurs Prolab 1 et Prolab 2. C'est dans ce dernier qu'a été intégré l'ensemble de notre travail.

1-3.4. Les démonstrateurs communs européens

Les différents démonstrateurs réalisés dans le cadre de Prometheus s'intègrent dans l'un des démonstrateurs communs européens (Common European Demonstrators : CEDs), c'est à dire un groupe concernant un domaine ou un objectif particulier. Les CEDs sont représentés dans le tableau 1.

Les démonstrateurs Prolab 1 et Prolab 2 développés par le groupe ProArt France font partie du CED 3. Les systèmes développés dans le cadre de ce CED assistent le conducteur et évitent l'accident grâce à l'évaluation des risques éventuels, permettant ainsi une mise en garde ou une intervention automatique.

Conduite sûre	
Amélioration de la vision	CED 1
Contrôle de l'adhérence et estimation de la marge de sécurité	CED 2-1
Système d'aide au suivi de ligne	CED 2-2
Adaptation de la conduite aux conditions de visibilité	CED 2-3
Contrôle de la vigilance du conducteur	CED 2-4
Système anti-collision	CED 3

Conduite coopérative	
Conduite coopérative	CED 4
Régularisation intelligente de la vitesse et de la distance	CED 5
Appel d'urgence	CED 6

Gestion du trafic et des flottes	
Gestion des flottes	CED 7
Guidage bimode	CED 9
Information de trafic / Guidage informé	CED 10

Tableau 1: Démonstrateurs communs européens (CED).

Dans le cas d'une commercialisation de ces systèmes, l'automatisation nécessitera la mise en place de nouvelles réglementations concernant les interventions automatiques et les problèmes de responsabilités en cas d'accidents. Mais la commercialisation n'est pas encore pour demain, beaucoup de recherches sont encore en cours pour développer notamment :

- Des interfaces homme-machine.
- Des capteurs et des actionneurs à bas prix sans lesquels les systèmes conçus seraient hors de prix.
- Des algorithmes robustes et appropriés pour l'identification des obstacles et l'intervention automatique.

1-4. Organisation du mémoire

Ce mémoire constitue la deuxième partie des recherches effectuées au Centre d'Automatique dans le cadre du programme Prometheus. La première partie, réalisée par Jean-Luc Bruyelle,

était axée sur la conception d'un capteur stéréoscopique, sa mise en oeuvre et sa validation. La présente thèse porte sur les algorithmes permettant d'analyser les images stéréoscopiques afin de développer un système de détection d'obstacles.

Le chapitre 2 présente les travaux réalisés par J.-L. Bruyelle dans le cadre de sa thèse [Bruyelle 94]. A partir d'une réflexion sur la nécessité de détecter les obstacles sur route et après avoir étudié les différentes méthodes de détection existantes, celui-ci a conçu un nouveau capteur stéréoscopique composé de deux caméras linéaires qu'il a baptisé *stéréoscope linéaire*. Ces travaux constituent la base de nos recherches dont l'objectif est de traiter les images fournies par ce capteur.

Les systèmes stéréoscopiques sont conçus dans le but de percevoir la structure tridimensionnelle de l'environnement dans lequel ils évoluent. Le chapitre 3 expose les démarches habituellement employées dans la littérature pour reconstruire les paramètres tridimensionnels d'une scène. Nous présentons, dans ce chapitre, les méthodes utilisées dans la littérature pour réaliser l'étape fondamentale d'un système stéréoscopique : la mise en correspondance.

Dans le chapitre 4, nous abordons le problème du choix de primitives adaptées aux images linéaires. Nous présentons les critères et les méthodes permettant la détection optimale et l'extraction de ces primitives.

La mise en correspondance constitue l'étape la plus importante d'un algorithme de stéréovision. A partir des ensembles de primitives trouvées dans deux images, elle consiste à déterminer les couples de primitives correspondant au même objet physique de la scène. Le chapitre 5 est consacré aux méthodes que nous avons développées pour apparier les primitives de deux images linéaires. Nous y détaillons également les calculs de triangulation géométrique.

Le chapitre 6 développe les techniques de reconstruction employées pour accélérer la procédure de mise en correspondance en fonction de la signalisation horizontale.

Le chapitre 7 est consacré à la reconstruction des obstacles présents devant le véhicule. Une analyse dynamique permet ensuite de fiabiliser la détection des obstacles. Quelques exemples de détection effectuée durant des essais en conditions réelles y sont également présentés.

L'objectif des recherches, faisant l'objet de ces deux thèses, est la réalisation de la fonction détection d'obstacles pour le véhicule expérimental Prolab 2 développé dans le cadre du programme Prometheus. Le chapitre 8 présente les caractéristiques du véhicule Prolab 2 ainsi que les modifications apportées afin d'y intégrer notre système de détection d'obstacles.

La dernière partie, conclusion de ce mémoire, présente les améliorations susceptibles d'être apportées à notre système de détection d'obstacles, ainsi que les perspectives futures.

Chapitre 2

STEREOSCOPIE LINEAIRE

Le travail effectué au Centre d'Automatique s'étant déroulé en deux temps, il nous a paru judicieux de rappeler, dans ce chapitre, les travaux réalisés par Jean-Luc Bruyelle lors de son doctorat. Nous nous sommes limités aux parties essentielles. Celles-ci permettront au lecteur d'avoir un aperçu général du travail effectué, celui-ci étant indispensable à la compréhension des recherches décrites dans les chapitres suivants. Pour des informations complémentaires sur le travail de Jean-Luc Bruyelle, le lecteur intéressé pourra se rapporter à son mémoire [BRUYELLE 94].

2-1. Réflexions sur la nécessité de la détection d'obstacles sur route

2-1.1. Pourquoi tant d'intérêt pour la détection d'obstacles?

2-1.1.1. Importance de l'automobile

L'automobile, encore réservée à certains privilégiés au début du siècle, s'est à tel point démocratisée, qu'elle est aujourd'hui présente dans pratiquement l'intégralité des ménages des pays industrialisés. Elle est devenue un élément essentiel de notre société. Mais l'importance de l'automobile dans la société moderne génère de nombreux inconvénients liés à la densité du trafic routier. La saturation des axes routiers aux abords des grandes villes est quotidiennement génératrice d'embouteillages. Le nombre croissant de véhicules pose des problèmes de stationnement, d'où la nécessité d'aménagements spécifiques à la fois coûteux et envahissants. Il ne faut pas oublier non plus les problèmes liés à l'environnement, longtemps négligés, comme le bruit et la pollution atmosphérique. Mais de tous ces inconvénients, le plus grave est la mortalité sur la route.

2-1.1.2. Les accidents de la route

Le nombre croissant de véhicules et l'amélioration constante de leurs performances ont eu pour conséquence l'augmentation du nombre d'accidents. Ceux-ci, de plus en plus graves, ont entraîné une hausse du nombre de blessés, mais surtout du nombre de personnes tuées.

Difficilement attribuables à une cause unique, les accidents sont souvent la conséquence d'une succession de circonstances fâcheuses. Les enquêtes de la sécurité routière ont montré que les facteurs liés au conducteur sont prépondérants dans les causes d'accidents [Gentile 94]. Nous pouvons citer par exemple : la vitesse, l'alcoolémie, la fatigue, l'inattention, etc.

2-1.1.3. Amélioration de la sécurité routière

En analysant les différentes causes d'accidents, on peut envisager de nombreux moyens pour y remédier : modifier les zones à risques du réseau routier, concevoir des véhicules plus sûrs. Mais le meilleur moyen serait sans doute d'éduquer et de former les conducteurs à ne plus prendre de risque au volant.

En attendant ce monde meilleur, une nouvelle approche de la sécurité routière est apparue depuis une dizaine d'année, faisant intervenir les technologies récemment développées (électronique, informatique) pour prévenir le conducteur de tout risque d'accident, et pour l'aider à l'éviter.

Le but est de créer une *voiture intelligente* capable de percevoir son environnement, mais aussi de tester son propre état (aide à la maintenance) et celui du conducteur (éthylisme, somnolence). Une telle voiture serait capable de donner des conseils à son conducteur sur la conduite à adopter, voire prendre des initiatives pour éviter un accident imminent.

Ce rêve a donné naissance à plusieurs projets de recherche, d'abord au sein des constructeurs, puis dans le cadre plus vaste de projet Eurêka en Europe, I.V.H.S. (Intelligent Vehicle - Highway System) aux Etats-Unis et V.I.C.S. (Vehicle Information and Communication System) au Japon. En Europe, le programme le plus ambitieux dans ce domaine est le projet Prometheus, dans lequel nous nous sommes impliqués.

2-1.2. Méthodes existantes de détection d'obstacles

La détection d'obstacles est un sujet sur lequel de nombreux chercheurs et constructeurs automobiles se sont penchés ces dernières années. Les résultats de ces recherches ont permis de développer deux types de méthodes.

2-1.2.1. Méthodes actives

Les méthodes actives de détection utilisent des détecteurs constitués de deux éléments distincts : un émetteur et un récepteur. La partie émettrice envoie, suivant la technologie employée, un faisceau d'onde radio (radar), d'onde sonore (sonar) ou de lumière (lidar). Ce sont les éventuelles réflexions renvoyées vers la partie réceptrice qui permettent de conclure sur la présence d'obstacles. Leur distance est alors connue en déterminant le temps mis par le faisceau émis pour revenir au récepteur (Cf. Figure 2.1).

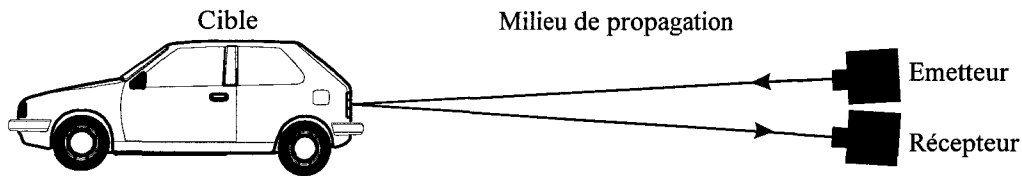


Figure 2.1 : Principe de fonctionnement d'un détecteur actif.

La portée et les performances des détecteurs actifs sont variables. Elles dépendent de nombreux paramètres :

- Les caractéristiques propres de l'émetteur (puissance, angle d'ouverture du faisceau) et du récepteur (sensibilité, sélectivité).
- L'absorption et la dispersion des faisceaux aller et retour par le milieu de propagation.
- Les propriétés réfléchissantes de la cible. Suivant la nature de la cible, le faisceau pourra être intégralement réfléchi, partiellement, voire pas du tout dans le cas de matériaux absorbants.

Les capteurs actifs peuvent être classés selon le type de technologie employée. On distingue :

- Les **radars** qui sont bien adaptés à la détection des véhicules dont la masse métallique importante réfléchit très bien les ondes radio. De plus, la précision de localisation est bonne, et la largeur du faisceau, qui définit les limites de la zone de détection, peut être déterminée par les caractéristiques de l'antenne. Un faisceau étroit associé à un balayage horizontal permet de donner une carte de profondeur de la scène, mais avec une certaine lenteur et une fiabilité amoindrie par l'utilisation d'un dispositif mécanique. Les être humains peuvent être détectés jusqu'à une distance de 80 mètres, mais les réflexions parasites causées par le fond de la scène (panneaux, arbres, ponts) créent de fausses alarmes fréquentes, ce qui diminue l'intérêt du dispositif pour ce type de détection. L'I.E.M.N. (Institut d'Electronique et Micro-électronique du Nord) en collaboration avec le L.E.O.S.T. (Laboratoire Electronique, Ondes et Signaux pour les Transports) a d'ailleurs développé un système anticollision basé sur un radar micro-onde à impulsions 60 Ghz [Deloof 94].
- Les **sonars** qui sont constitués d'un haut parleur qui émet des impulsions ultrasonores, dont les réflexions sont captées par un microphone. Les avantages sur

les radars sont un coût, une consommation électrique et un encombrement nettement inférieurs. Tous les obstacles acoustiquement réfléchissants peuvent être détectés par ultrasons, ce qui représente la majorité des objets présents dans la nature. Cependant sur route, ce type de capteurs est vite limité. A vitesse élevée, le récepteur (microphone) peut être saturé en raison du vent, ce qui limite la portée du dispositif. Les capteurs acoustiques sont, en revanche, souvent utilisés sur des robots mobiles comme capteurs de proximité [Wane 93], par exemple pour suivre un mur à distance constante [Briot 79].

- Les **lidars** qui fonctionnent sur le même principe que les radars, mais utilisent un faisceau lumineux — généralement issu d'un laser — au lieu d'un faisceau d'ondes radio. La détection est possible sur tous types d'objets, métalliques ou non, mais les objets sombres et mats posent évidemment des problèmes, de même que les surfaces transparentes qui sont traversées par le faisceau laser sans le réfléchir. Par temps clair, le lidar offre une bonne détection et, surtout, une localisation précise des obstacles. Cependant les mauvaises conditions atmosphériques (brouillard, pluie, neige) limitent la portée du système à cause d'une absorption et/ou d'une dispersion du faisceau lumineux. Dans le cadre du programme Prometheus, le professeur Gallice et son équipe ont conçu un capteur original constitué d'un télémètre laser couplé à une caméra vidéo [Gallice 91]. Ce capteur permet la détection des obstacles présents devant un véhicule automobile [Gallice 94a] et [Gallice 94b].

2-1.2.2. Méthodes passives

Dans le cas des méthodes passives, le détecteur n'est constitué que d'un seul élément : le récepteur. Il s'agit essentiellement d'un système visuel composé d'une ou plusieurs caméras (Cf. Figure 2.2). La démarche consiste à analyser les images provenant des caméras. Les différents avantages que procurent les méthodes de détection passives par rapport aux méthodes actives sont une meilleure résolution et une exploration plus rapide de la scène puisqu'aucun balayage n'est nécessaire pour percevoir la totalité des informations concernant l'environnement étudié. Il faut cependant préciser que l'interprétation des images exige des traitements informatiques généralement longs. Ces technologies présentent néanmoins des inconvénients. Le principe de fonctionnement de ces capteurs passifs suppose un contraste suffisant entre les différents obstacles et le fond pour pouvoir être vus et donc détectés.

Comme dans le cas des lidars, une mauvaise visibilité (forte pluie, brouillard, neige) rend souvent les images inexploitable ce qui compromet toute détection. L'utilisation de ce type de système est de même impensable la nuit sans utiliser de capteurs spécifiques (caméra infrarouge).

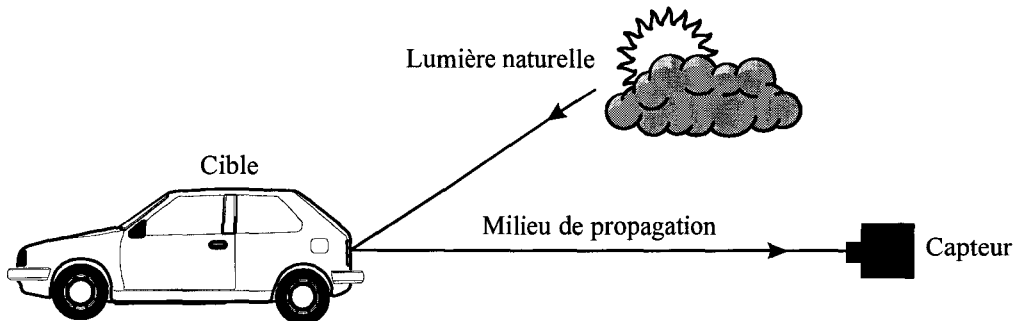


Figure 2.2 : Principe de fonctionnement d'un détecteur passif.

Il existe trois grandes approches pour la détection passive d'obstacles :

- La **segmentation d'images monoscopiques** permet de détecter les obstacles mais n'est pas très fiable car de nombreux éléments parasites, tels que les ombres, les marquages au sol dans le cas d'un contexte routier, sont assimilés à des obstacles. Cette approche bidimensionnelle est mal adaptée car elle ne permet pas d'obtenir les notions de relief et de mouvement relatif, paramètres essentiels dans le cadre de la détection d'obstacles. Le lecteur intéressé trouvera un exemple de cette approche dans [Graefe 90].
- L'**analyse du mouvement** permet de différencier les objets verticaux (obstacles et autres véhicules) des objets horizontaux (chaussée). Cette méthode donne de bons résultats mais possède un défaut majeur. Si elle détecte effectivement les obstacles, elle ne permet pas de les localiser. La démarche la plus répandue consiste à extraire le flot optique [Enkelmann 90], mais il existe des variantes à ce type de méthode [Carlsson 90].
- La **stéréovision** est l'approche la plus intéressante à notre avis. La connaissance de la position des différents objets dans l'espace permet de différencier les obstacles des autres objets. La stéréovision présente cependant l'inconvénient majeur d'impliquer une quantité de calculs importante, incompatible avec la détection d'obstacles en temps réel. L'étude approfondie de cette approche sera l'objet du chapitre 3.

2-1.3. Approche nouvelle

Si les méthodes actives de détection donnent de bons résultats, elles sont néanmoins lourdes et coûteuses à mettre en oeuvre. Il nous a donc paru plus judicieux de nous tourner vers les méthodes passives qui peuvent se contenter de capteurs simples et d'unités de traitement réduites facilement embarquables dans une voiture particulière.

Pour conserver les avantages d'un système de détection stéréoscopique (bonne localisation), sans en avoir les inconvénients (temps de traitement), nous avons abandonné les caméras matricielles employées habituellement et opté pour un système basé sur l'utilisation de caméras dites linéaires. Le reste de ce chapitre expose le principe de fonctionnement de ce nouveau capteur, développé dans le cadre du programme Prometheus, et que nous appelons **stéréoscope linéaire**. La conception de ce nouveau dispositif ayant fait l'essentiel de la thèse de Jean-Luc Bruyelle, nous nous contenterons ici de n'en reprendre que les principaux éléments.

2-2. Conception d'un dispositif de prise de vue stéréoscopique linéaire

2-2.1. Les caméras linéaires

2-2.1.1. Constitution d'une caméra linéaire

Une caméra linéaire est une caméra constituée :

- D'un **capteur linéaire**. C'est une version simplifiée des capteurs CCD matriciels utilisés dans les caméras vidéo standards. Il comporte essentiellement **une seule** rangée de photocapteurs, des registres à décalage, des amplificateurs de sortie et des circuits annexes destinés à la gestion des registres à décalage. Sa conception permet des acquisitions très rapides. Les capteurs linéaires présentent des caractéristiques intéressantes pour de nombreuses applications: grande résolution, faible quantité d'information à traiter, compatibilité avec les optiques photographiques courantes, souplesse des paramètres de prise de vue (fréquence d'acquisition et temps d'exposition adaptables selon les besoins et les conditions de prise de vue).
- D'un **objectif**. La grande largeur du capteur (20 à 28 mm) permet l'utilisation d'objectifs photographiques classiques dont les caractéristiques de résolution et de distorsion sont très supérieures à celles des objectifs vidéo généralement utilisés. Cette grande résolution des objectifs photographiques de qualité est indispensable pour tirer partie de celle des capteurs linéaires.

- De **composants électroniques**. L'électronique de synchronisation et de conditionnement du signal à la sortie du capteur pilote les principaux signaux, à savoir l'horloge de transmission des pixels, la commande de transmission de ces pixels, la commande d'exposition et la sortie vidéo.

Il est à noter qu'aucune caméra linéaire ne peut fonctionner sans un ordinateur équipé d'une carte spécialisée pour l'acquisition des images linéaires. Toutes les fonctions de la caméra, y compris la visualisation vidéo, sont accessibles uniquement par le biais de la carte d'acquisition et de son logiciel de gestion.

2-2.1.2. Champ de visée d'une caméra linéaire

Le champ de visée est réduit à un plan, nommé *plan de visée*, que nous supposons horizontal pour plus de clarté. De ce fait, la notion de verticale est absente.

Les caractéristiques géométriques du plan de visée sont les suivantes (Cf. Figure 2.3) :

- Ouverture angulaire du champ de visée :
$$\alpha = 2 \arctan \frac{l}{2 \cdot f}$$
- Largeur du champ à une distance D :
$$L = \frac{l \cdot D}{f}$$

où :

- l est la largeur du capteur CCD.
- f est la distance focale des objectifs.
- D est la distance entre la caméra et la ligne objet.

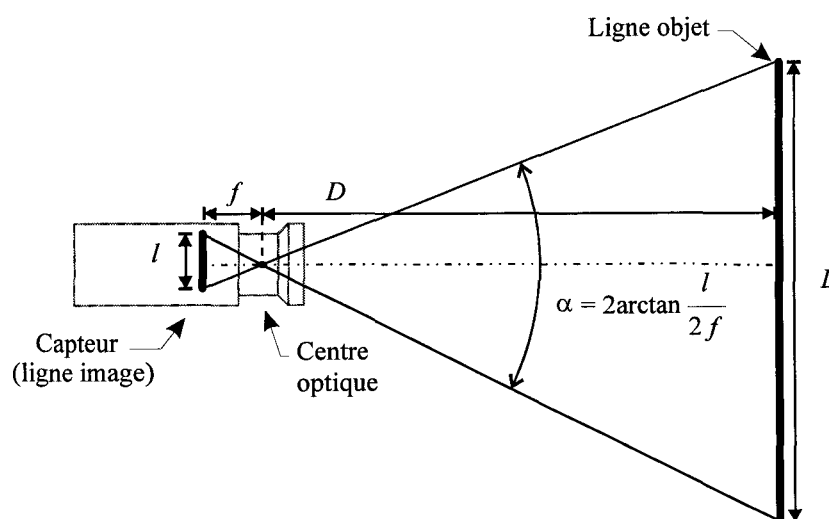


Figure 2.3 : Champ de visée d'une caméra linéaire.

2-2.2. Principe du stéréoscope linéaire

2-2.2.1. Hypothèses

Hypothèses sur les caméras

Le dispositif de prise de vue n'est pas différent dans son principe de tous les systèmes de prises de vue stéréoscopique à deux caméras. Cependant, la nature particulière des caméras linéaires, qui ne restituent pas l'information verticale, impose une configuration particulière sans laquelle les deux caméras du couple stéréoscopique ne verraient pas la même scène, ce qui rendrait impossible la reconstruction 3D :

1) Les plans de visée des deux caméras sont confondus.

Cette hypothèse signifie que :

- Les deux capteurs doivent être dans le prolongement l'un de l'autre.
- La ligne contenant les deux capteurs doit être parallèle à la ligne joignant les centres optiques des deux objectifs.

Dans la suite, le plan qui contient les capteurs et les centres optiques des deux objectifs sera appelé *plan de visée du stéréoscope linéaire*.

2) Les objectifs sont des sténopés.

L'objectif est assimilé à un sténopé (du grec *stenos*, étroit, et *opê*, trou), c'est à dire un trou infiniment petit, confondu avec le centre optique de l'objectif.

3) Les deux objectifs ont la même distance focale.

Hypothèse sur la scène

La plus classique des hypothèses, qui constitue la base de nombreuses méthodes de détection d'obstacles sur route, est que *la chaussée est plane* [Carlsson 90], [Enkelmann 90].

2-2.2.2. Installation des caméras sur le véhicule

Les caméras linéaires sont installées sur le véhicule, à un emplacement qui sera précisé au chapitre 8.

Leurs axes optiques sont parallèles et écartés d'une distance E (Cf. Figure 2.4). Le parallélisme des axes optiques n'est pas indispensable pour la reconstruction, mais cette configuration simplifie les équations et, de ce fait, réduit considérablement la quantité de calculs requis pour la reconstruction 3D, ce qui est essentiel pour un système devant fonctionner en temps réel.

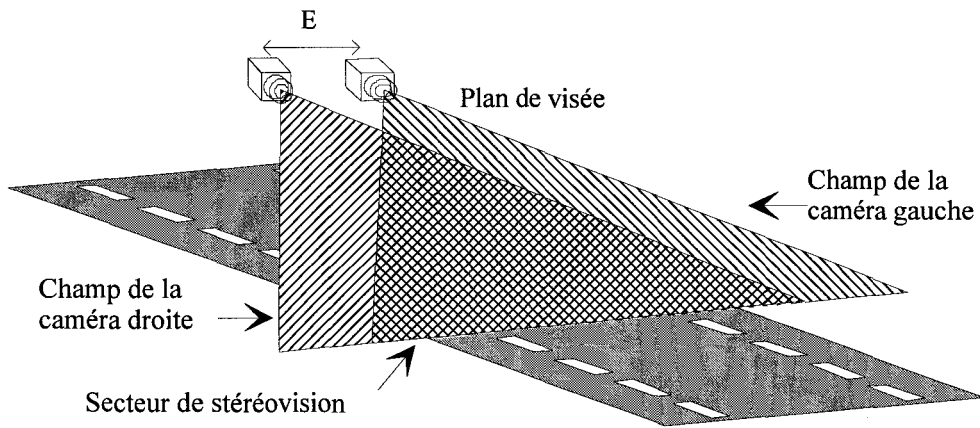


Figure 2.4 : Principe de la stéréoscopie par stéréovision linéaire.

Ces deux axes définissent le plan de visée qui est légèrement incliné par rapport à l'horizontale d'un angle θ , de manière à couper la chaussée à une distance D_{max} à l'avant du véhicule, qui définira la portée du dispositif (Cf. Figure 2.5).

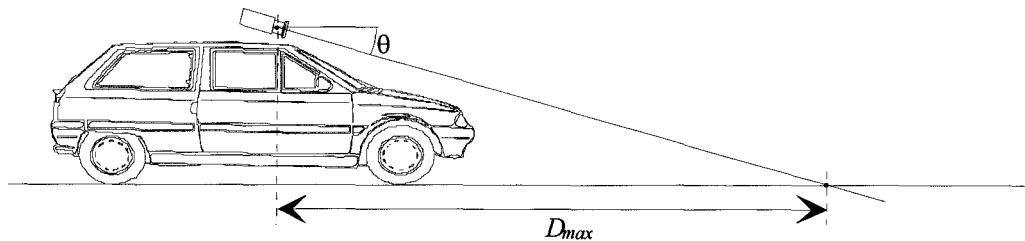


Figure 2.5 : Installation des caméras sur un véhicule.

La partie commune aux champs de visée des caméras gauche et droite est appelée *secteur de stéréovision* (Cf. Figure 2.4). Tout obstacle, même de faible hauteur, présent dans ce secteur est vu simultanément par les caméras de gauche et de droite, et peut ainsi être détecté et localisé par triangulation. Les caractéristiques géométriques du secteur de stéréovision sont précisées sur la figure 2.6.

- L'axe est confondu avec l'axe du stéréoscope.

- Distance minimale au centre optique :

$$D_{min} = \frac{E \cdot f}{l}$$

- Ouverture angulaire :

$$\alpha = 2 \arctan \frac{l}{2 \cdot f}$$

- Largeur couverte par un pixel à une distance D :

$$P = \frac{p \cdot D}{f}$$

où :

- l est la largeur du capteur CCD.
- N est le nombre de pixels du capteur.
- p est la largeur d'un photosite ($p = \frac{l}{N}$).
- f est la distance focale des objectifs.
- E est l'écartement des axes optiques des deux caméras.

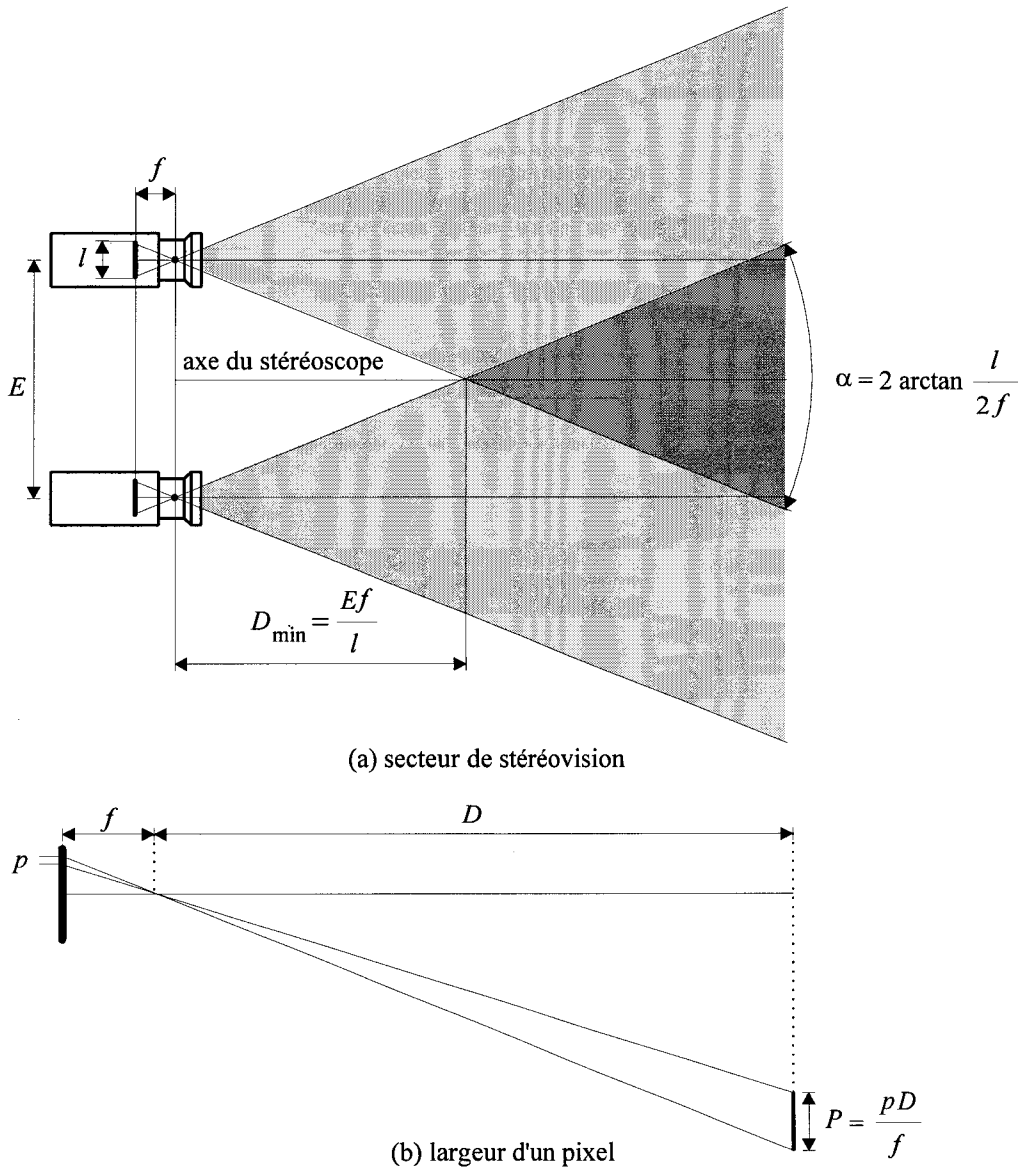


Figure 2.6 : Géométrie du stéréoscope linéaire.

2-2.2.3. Reconstitution de la profondeur et de la position

Considérons un point objet observé par le stéréoscope linéaire (Cf. Figure 2.7). Chaque capteur perçoit de ce point une image qui est sa projection à travers le centre optique de l'objectif associé.

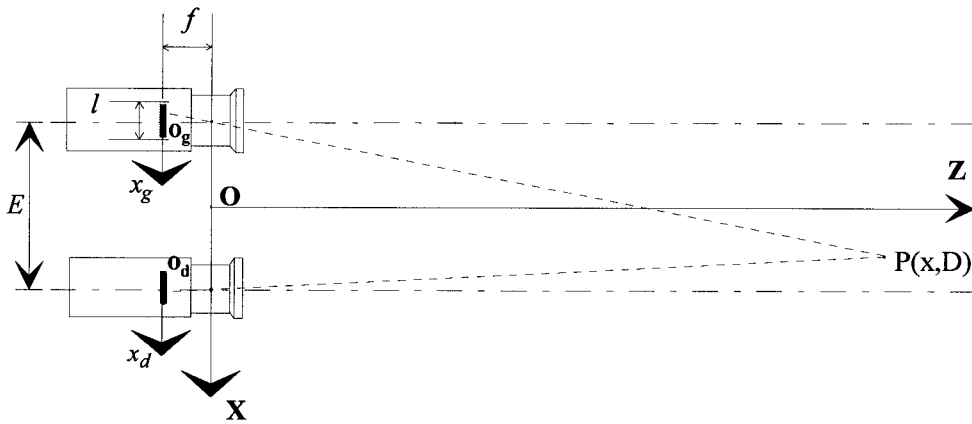


Figure 2.7 : Projection d'une image sur les capteurs du stéréoscope linéaire.

Définissons un repère lié au dispositif de prise de vue :

- L'origine O est au milieu du segment de droite joignant les centres optiques des deux objectifs.
- L'axe des X (abscisses) est la droite qui joint les centres optiques des deux objectifs.
- L'axe des Z (profondeurs) est contenu dans le plan optique, normal à l'axe des X , et est dirigé vers la scène. Il est confondu avec l'axe du stéréoscope.

Soit $P(X, D)$ la position du point objet P dans ce repère, où X est l'abscisse de P selon l'axe OX et D sa profondeur selon l'axe OZ .

Définissons, de plus, un repère lié à chaque capteur :

- L'origine est au centre du capteur. On la note o_g pour le capteur gauche, o_d pour le capteur droit.
- L'axe des abscisses, x_g ou x_d , est dirigé vers la droite du capteur quand on regarde vers la scène.

Soit x_g la position du point image sur le capteur gauche, et soit x_d sa position sur le capteur droit.

Les positions des deux points objets dans leurs repères locaux respectifs s'expriment sous la forme :

$$\begin{cases} x_g = \frac{f \cdot (\frac{E}{2} + X)}{D} \\ x_d = \frac{f \cdot (\frac{E}{2} - X)}{D} \end{cases} \quad \text{Equation 2.1}$$

Notion de disparité

La *disparité* du point P est la différence de position δ entre les images de P sur les deux capteurs (Cf. Figure 2.7) :

$$\delta = |x_g - x_d| \quad \text{Equation 2.2}$$

Dans le cas présent, elle prend la forme particulière :

$$\delta = \frac{E \cdot f}{D}, \quad \forall x \quad \text{Equation 2.3}$$

Ce résultat indique que la disparité est la même pour les images de tous les points situés à une même profondeur, indépendamment de leurs abscisses.

Calcul de la profondeur à partir de la disparité

La profondeur D se déduit directement de la disparité et des paramètres du stéréoscope à partir de la relation précédente :

$$D = \frac{E \cdot f}{\delta} \quad \text{Equation 2.4}$$

Calcul de la position à partir de la profondeur

Nous avons vu plus haut comment obtenir la position des points objets sur les capteurs gauche et droit (Cf. Equation 2.1). Nous pouvons utiliser ces équations pour déterminer la position X d'un point objet à partir de sa profondeur D :

$$\begin{cases} X = \frac{x_g \cdot D}{f} - \frac{E}{2} \\ X = \frac{x_d \cdot D}{f} + \frac{E}{2} \end{cases} \quad \text{Equation 2.5}$$

L'une ou l'autre de ces équations permet de calculer la position X du point objet à partir de sa profondeur D et de la position du point image correspondant sur le capteur gauche (x_g) ou sur le capteur droit (x_d).

2-2.3. Réalisation électronique du stéréoscope linéaire

Le stéréoscope linéaire est réalisé sous la forme d'un ensemble électronique dont le support est un ordinateur de type PC-AT.

2-2.3.1. Caméras

Nous avons choisi le modèle Fairchild CAM 1840N, dont les caractéristiques principales sont les suivantes :

- Dimensions (l x h x p, sans objectif) : 81,3 x 82,5 x 95,9 mm.
- Définition : 2592 points.
- Largeur du capteur : 25,9 mm.
- Dimensions des photosites : 10 mm x 10 mm.
- Obturateur électronique.
- Monture d'objectif compatible Nikon AI.
- 1 sortie vidéo (multiplexage des 2 sorties du capteur).
- Signaux de synchronisation.
- Alimentation : +5V, +15V, -15V. Ces tensions sont délivrées par deux alimentations à découpage – une par caméra – enfichées dans le bus ISA du ordinateur hôte.

La distance focale des objectifs de marque Nikon est de 35 mm.

2-2.3.2. Acquisition simultanée des images

La carte d'acquisition linéaire choisie est, quant à elle, une DT2856 de Data Translation.

La première difficulté est l'acquisition simultanée des images gauche et droite. La synchronisation précise des acquisitions de ces deux images est d'une importance capitale pour pouvoir les exploiter correctement par la suite. Si celles-ci étaient saisies à des instants différents, la mise en correspondance (Cf. chapitres 5 et 6) serait impossible puisque chaque image représenterait les objets à des positions différentes.

La première solution serait d'interconnecter deux cartes d'acquisition d'images linéaires afin de les synchroniser entre elles. Malheureusement, les cartes d'acquisition linéaire disponibles sur le marché ne sont pas conçues pour ce type d'utilisation. Pour éviter les modifications de ces cartes, le problème a été résolu en interfaçant les caméras avec une seule de ces cartes, par l'intermédiaire d'un multiplexeur développé spécifiquement pour garantir la cohérence des

séquences stéréoscopiques linéaires (Cf. Figure 2.8).

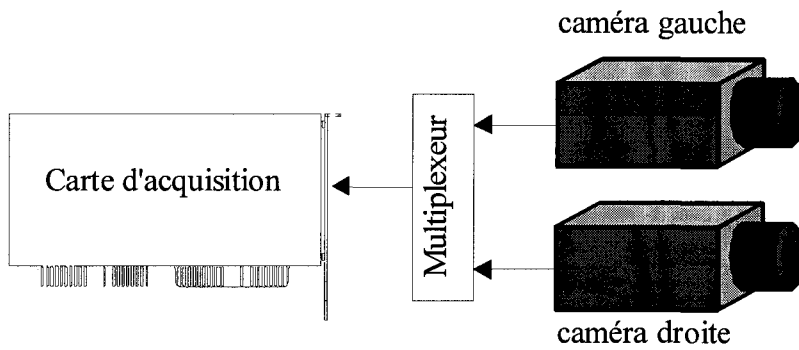


Figure 2.8 : Acquisition simultanée des images.

Ce multiplexeur possède deux ports d'entrée, un pour chaque caméra, et un port de sortie connecté à la carte d'acquisition linéaire où sont stockées les deux images. Sa principale fonction est la gestion des signaux de synchronisation et de commande entre les deux caméras et la carte d'acquisition, ainsi que le contrôle du transit des pixels vers la carte d'acquisition. Celle-ci possède deux entrées vidéo pour s'adapter aux caméras disposant de sorties différentes pour les pixels pairs et impairs. Ces deux entrées sont utilisées pour connecter les sorties des deux caméras via le multiplexeur. Ceci permet d'économiser le coût, mais aussi les interconnexions et la consommation électrique d'une deuxième carte d'acquisition.

Après chaque prise de vue, la mémoire interne de la carte d'acquisition contient un tableau de 2×2592 octets. Les éléments d'indices pairs correspondent aux pixels de la caméra gauche, tandis que les éléments d'indices impairs correspondent aux pixels de la caméra de droite (Cf. Figure 2.9).

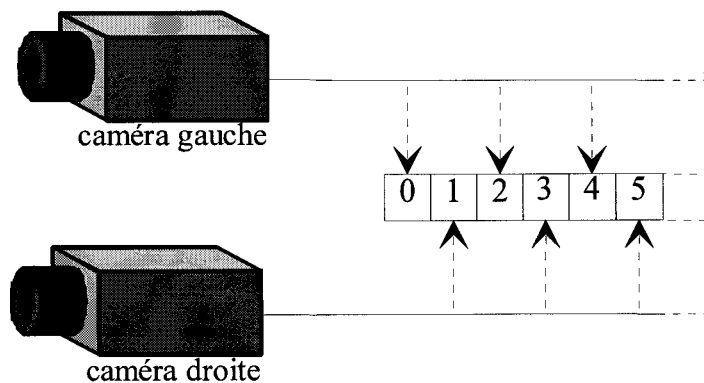


Figure 2.9 : Multiplexage des signaux vidéo.

2-2.4. Réalisation mécanique du stéréoscope linéaire embarqué

2-2.4.1. Contraintes

Comme nous l'avons vu au § 2-2.2.1, la stéréovision linéaire suppose que les plans de visée des deux caméras soient rigoureusement confondus. Cette contrainte est en particulier essentielle dans le cas où l'objet observé présente une faible extension verticale, ou s'il présente de fortes différences de niveau de gris dans la direction verticale (Cf. Figure 2.10).

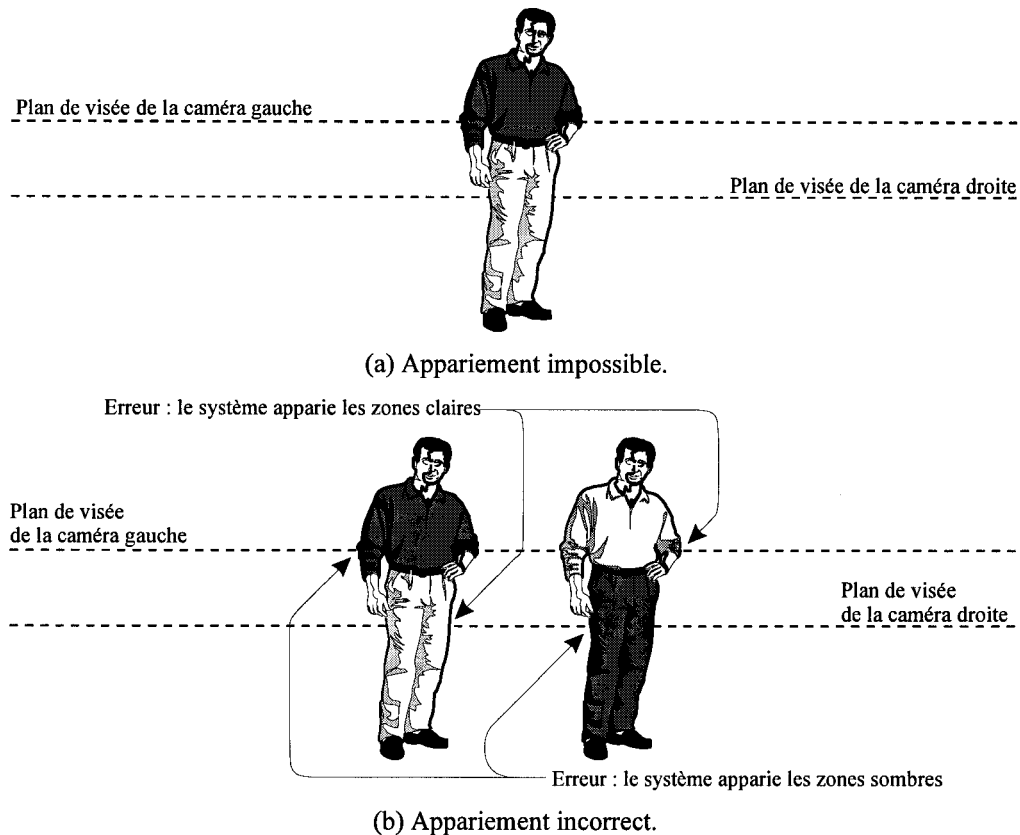


Figure 2.10 : Effet d'un décalibrage affectant la coplanéité des plans de visée.

En effet, dans ce cas, la mise en correspondance des différentes images serait impossible, si celles-ci ne correspondaient pas à la même « tranche » du sujet, c'est à dire à une même section par un plan unique.

Il est donc essentiel que le système mécanique de positionnement et d'orientation des caméras permette un réglage extrêmement précis de la coplanéité des champs de visée. Ceci implique un ajustement simultané et minutieux des angles d'inclinaison et de roulis, et de la hauteur relative des centres optiques des objectifs.

C'est pourquoi un support mécanique spécifique à la stéréovision linéaire a été conçu et réalisé, afin d'assurer la coplanéité des champs de visée. Il permet également de déterminer

les paramètres de prise de vue en ajustant les orientations des deux caméras.

2-2.4.2. Monture d'une caméra

Chaque caméra est montée sur un support indépendant (Cf. Figure 2.11), orientable avec une grande précision selon les angles de site, d'azimut et de roulis¹. La hauteur du support est également ajustable. Chaque monture est constituée de 3 platines métalliques visibles sur la figure 2.11 : la *platine inférieure* fixe, la *platine supérieure* orientable en site et en roulis et la *platine azimutale*.

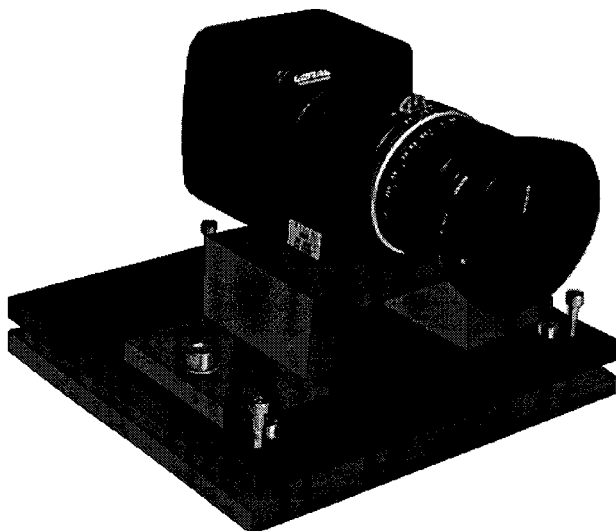


Figure 2.11 : Caméra linéaire sur sa monture ajustable.

2-2.4.3. Monture commune

Les deux montures de caméras décrites ci-dessus ne peuvent être montées directement sur le véhicule. En effet, la rigidité de celui-ci est insuffisante pour assurer le maintien du calibrage des caméras durant les prises de vues. Les deux caméras ont donc été montées, avec leurs supports, sur une monture commune très rigide qui repose sur le véhicule. La monture intégrée sur le véhicule se compose d'une poutre d'acier dont la forme et la masse (15 kg) assurent une inertie et une rigidité très importante au stéréoscope linéaire (Cf. Figure 2.12)

¹ L'orientation d'une caméra est exprimée en coordonnées polaires dans un repère centré sur le centre optique de son objectif. Les trois axes de rotation sont le *site*, l'*azimut* et le *roulis*.

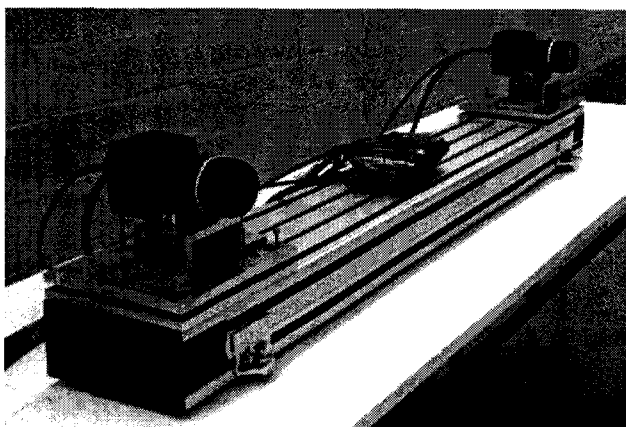


Figure 2.12 : Stéréoscope linéaire sur sa monture rigide.

2-3. Calibrage du stéréoscope linéaire

2-3.1. Problèmes et contraintes

2-3.1.1. Problème à résoudre par le calibrage

Le stéréoscope linéaire est composé de caméras linéaires, qui ne peuvent constituer un couple stéréoscopique que si leurs champs de visée sont situés dans le même plan, appelé *plan de visée* du stéréoscope (Cf. § 2-2.1.2).

En effet, le champ de visée vertical des caméras linéaires, limité à une seule ligne, ne permet pas la compensation passive d'un écart de position ou d'orientation de l'une des caméras par une procédure logicielle.

De plus, la portée du dispositif de détection est conditionnée par l'angle de site du stéréoscope (Cf. [Bruyelle 94] § 2.3.4). La simple mesure des paramètres de calibration permettrait de déterminer la portée, mais non de l'ajuster à la valeur voulue, ce qui est indispensable dans notre application.

Ces raisons font que nous devons utiliser une méthode active de calibrage, afin d'assurer la coplanéité des champs de visée des deux caméras et de mettre celles-ci dans une configuration géométrique prédéterminée.

2-3.1.2. Contrainte

Le champ vertical des caméras linéaires étant réduit à une ligne, un objet est soit vu par les caméras ou alors il ne l'est pas du tout. Dans ce cas, il est impossible de savoir où il se situe par rapport au plan de visée. La mire de calibration doit donc fournir le moyen de déterminer la position du plan de visée par rapport aux repères tracés sur la mire.

2-3.2. Mire de calibrage

La mire de calibrage est plane, réalisée en noir mat sur un fond blanc diffusant destiné à mieux « accrocher la lumière ». Elle est représentée sur la figure 2.13.

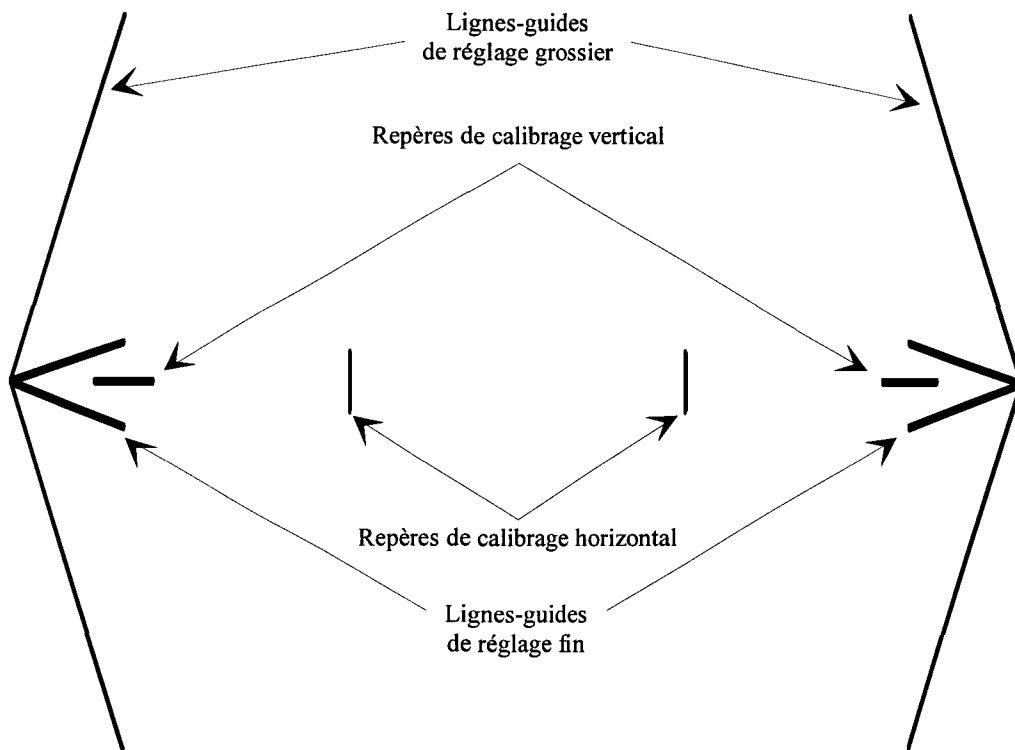


Figure 2.13 : Constitution de la mire de calibrage.

La mire est constituée d'un certain nombre de repères destinés à combler l'absence d'information verticale :

- **Repères de calibrage vertical**

Il s'agit de deux segments de droite horizontaux, placés dans le prolongement l'un de l'autre. Le plan de visée d'une caméra est correctement ajusté lorsque cette dernière « voit » ces deux repères en même temps. Les deux plans de visée sont coplanaires lorsqu'ils sont tous les deux parfaitement ajustés.

- **Repères de calibrage horizontal**

Il s'agit de deux segments de droite verticaux, symétriques par rapport au centre de la mire et placés de façon à couper la droite qui porte les repères de calibrage vertical. Ils sont écartés d'une distance égale à la distance qui sépare les centres optiques des objectifs des deux caméras. Les axes optiques des deux caméras sont parallèles lorsque l'image du repère gauche est au centre du capteur de la caméra gauche, et que l'image du repère droit est au centre du capteur de la caméra droite.

- **Lignes-guides**

Afin de savoir si le plan de visée se trouve au-dessus ou au-dessous des repères de calibrage vertical et horizontal, des lignes obliques, situées de part et d'autre des repères de calibrage, ont été dessinées sur la mire. Leur rôle est d'indiquer si la direction choisie pour l'ajustage est la bonne. En effet, sur l'écran, elles doivent s'écartier l'une de l'autre pour indiquer que l'ajustement va dans le bon sens.

Deux jeux de lignes sont prévus, l'un pour un réglage grossier, utile au début de la procédure de calibrage, l'autre pour un réglage fin, pour ajuster précisément le plan de visée lorsqu'il est proche des repères de calibrage.

Signalons que les dimensions de la mire ne sont pas arbitraires. Elles sont fixées en fonction des caractéristiques de la caméra et des dimensions de la scène à observer.

2-3.3. Procédure de calibrage

La procédure de calibrage est simple à effectuer, mais difficile à expliquer. Jean-Luc Bruyelle dans sa thèse l'a présentée sous la forme d'un guide pratique. La connaissance de cette procédure n'étant pas utile à la compréhension de la suite de mémoire, nous ne la détaillerons pas. Cependant le lecteur intéressé trouvera dans le chapitre 3 de [Bruyelle 94] toutes les explications nécessaires sur la procédure de calibrage et les justifications théoriques de celle-ci.

2-3.4. Précision du stéréoscope linéaire

Après le calibrage, le stéréoscope linéaire a été soumis à un étalonnage destiné à mesurer la précision de localisation des objets. Cette opération était nécessaire afin de déterminer le degré de fiabilité de notre système de détection. Pour cela, les positions déterminées avec le stéréoscope linéaire ont été comparées aux mesures de position faites au moyen de deux théodolites de chantier, présentant une précision théorique de 20 μm par mètre de distance. Cette précision est bien supérieure à celle du stéréoscope en raison de l'échantillonnage spatial opéré par les capteurs des caméras.

Les figures 2.14 et 2.15 montrent, respectivement, les résultats de l'étalonnage du stéréoscope, muni d'objectifs de 35 mm et de 20 mm. L'axe vertical représente la profondeur (distance le long de l'axe optique), l'axe horizontal représente l'abscisse (distance perpendiculaire à l'axe optique).

Chaque point muni d'un trait représente une mesure. Le gros point correspond à la position réelle mesurée avec les théodolites, tandis que l'extrémité du trait représente la position mesurée avec le stéréoscope. Les rectangles d'incertitude sur la position correspondant à l'effet d'échantillonnage du capteur sont également tracés, les distorsions et tout autre effet étant ignorés. Ceci permet de vérifier l'influence réelle de ces effets.

Les rectangles d'incertitude, correspondant à échantillonnage spatial par les capteurs CCD du stéréoscope, ont les dimensions suivantes :

$$\begin{cases} \Delta X = \frac{Ep}{\delta} \\ \Delta Z = \frac{Ef}{\delta} - \frac{Ef}{\delta - p} \end{cases}$$

où :

- E est l'écartement des axes optiques des deux caméras.
- f est la distance focale des objectifs.
- p est la largeur des photosites du capteur.
- δ est la disparité.

Notons également que les contraintes liées aux dimensions des salles dans lesquelles nous avons effectué ces procédures d'étalonnage nous ont empêché de réaliser certaines mesures. Il s'agit des bords des plans de visée, car avec les objectifs de 35 mm, la salle n'était pas assez large. Au-delà de 20 mètres, avec les objectifs de 20 mm, la salle n'était pas assez longue.

On constate que les erreurs sur la localisation dépassent notablement la seule imprécision de l'échantillonnage spatial, en particulier en bord de champ. Les écarts entre la distance réelle et la distance mesurée atteignent, avec les objectifs de 20 mm, jusqu'à 5% à l'extrême droite, et 15% à l'extrême gauche (Cf. Figure 2.15). Par ailleurs, l'asymétrie constatée entre les deux cotés du champ semble indiquer un désalignement de l'axe optique de l'une au moins des deux caméras. Les étalonnages réalisés avec une distance focale de 35 mm indiquent une précision de localisation meilleure que 4% sur la distance Z des objets détectés (Cf. Figure 2.14).

La précision atteinte étant suffisante pour le but fixé, les travaux sur le sujet n'ont pas été approfondis.

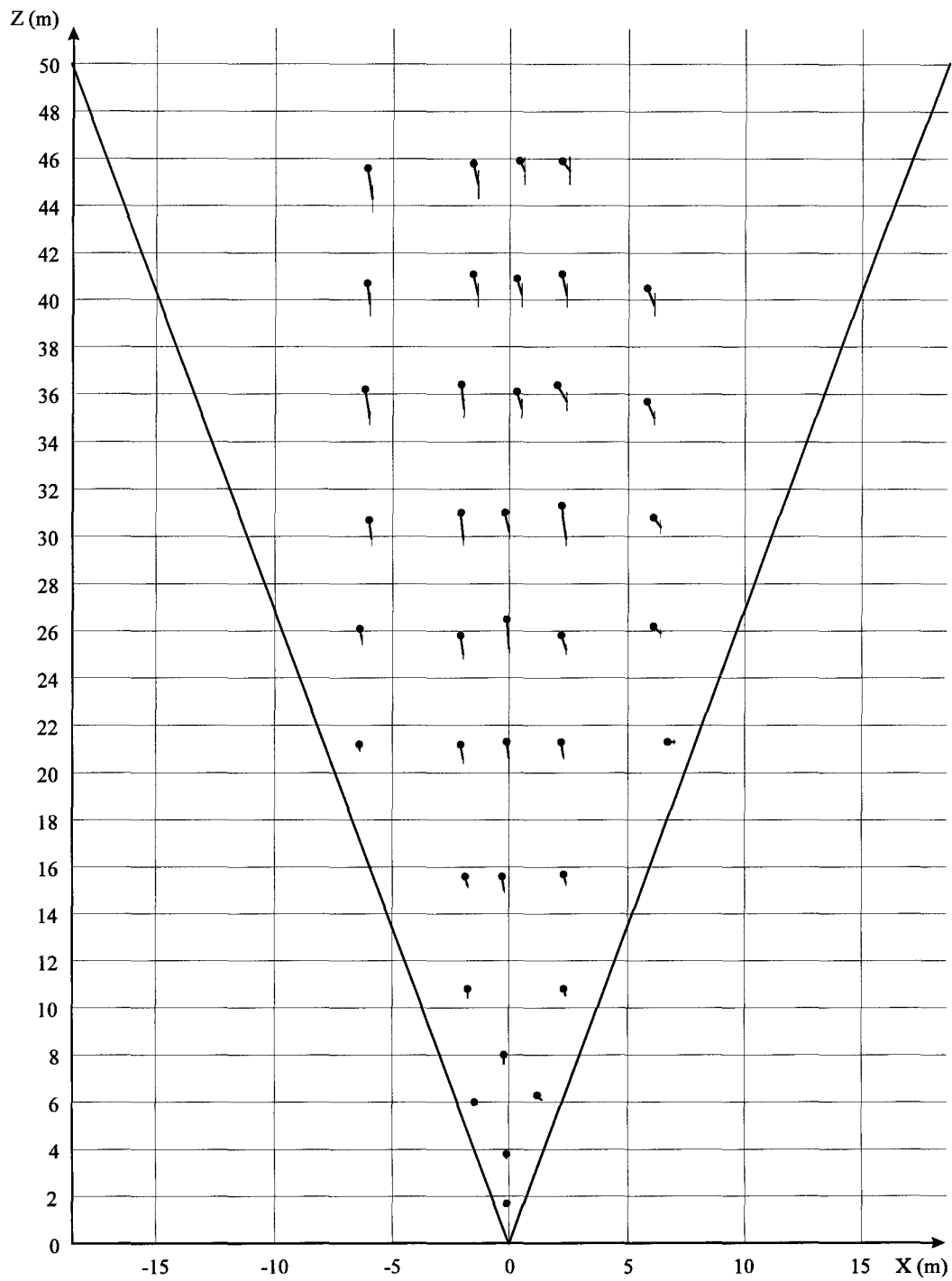


Figure 2.14 : Résultat de l'étalonnage avec des objectifs de 35 mm.

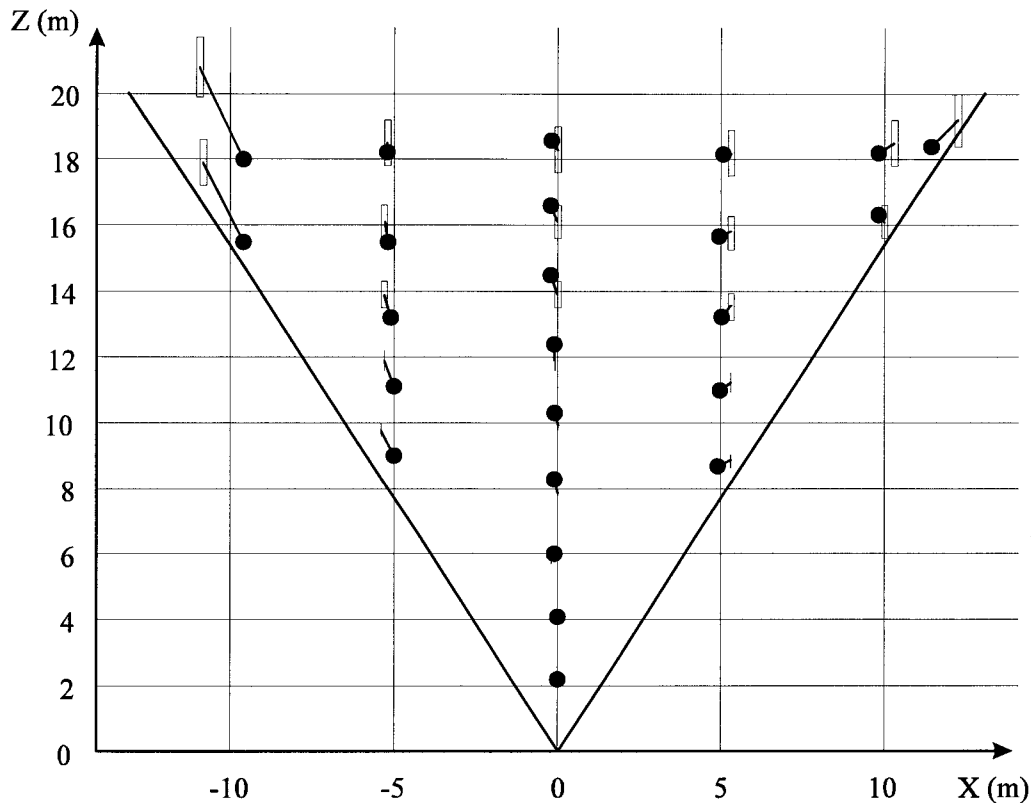


Figure 2.15 : Résultat de l'étalonnage avec des objectifs de 20 mm.

2-4. Premiers essais du stéréoscope linéaire

La description du stéréoscope linéaire lui-même est maintenant terminée. Mais avant de passer à l'analyse des images qu'il fournit, il est nécessaire de pouvoir enregistrer les séquences d'images, afin d'obtenir des données de référence.

2-4.1. Enregistrement de séquences stéréoscopiques linéaires

Pour améliorer la lisibilité, les images linéaires sont enregistrées sous forme de séquences pour les raisons suivantes :

- Une image linéaire seule est illisible. La visualisation d'une séquence, par contre, permet, avec une certaine expérience de ce type d'image, d'avoir une idée de la scène qui a été enregistrée, même si elle est parfois difficile à interpréter.
- Une séquence stockée sous la forme d'un fichier est plus facile à manipuler que plusieurs centaines de fichiers contenant chacun une seule image de la séquence enregistrée.

Il importe de remarquer que les images linéaires ne peuvent être enregistrées à l'aide d'un magnétoscope comme le sont d'habitude les images matricielles. En effet, les caméras

linéaires ne possèdent pas le même principe de fonctionnement que les caméras matricielles (Cf. 2-2.1.1) et les deux formats d'images ne sont pas compatibles.

Pour réaliser les enregistrements, nous utilisons le disque dur d'un PC comme un magnétoscope, et le stéréoscope comme une caméra vidéo. La visualisation est, quant à elle, réalisée grâce au moniteur VGA du PC hôte pour vérifier la qualité des enregistrements (Cf. Figure 2.16).

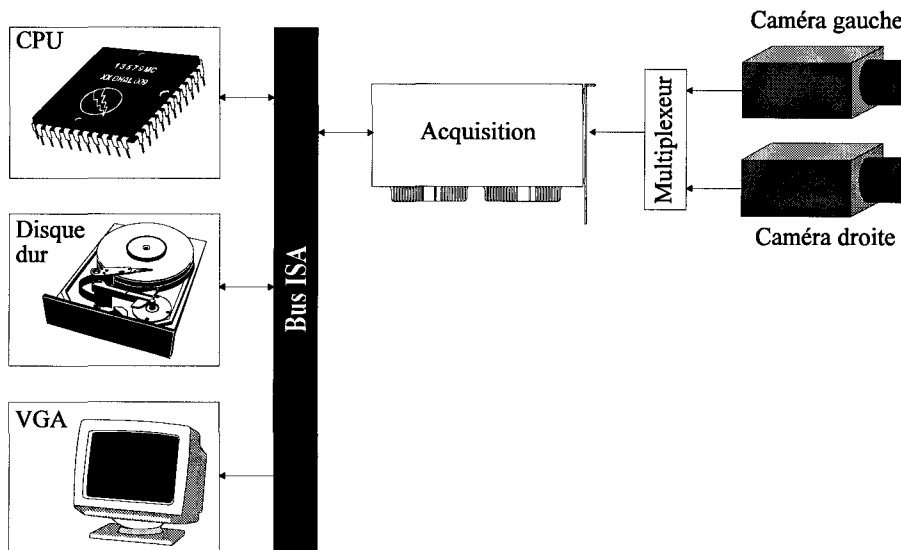


Figure 2.16 : Configuration matérielle pour l'enregistrement de séquences linéaires stéréoscopiques.

2-4.2. Caractéristiques des différents stéréoscopes linéaires

Les exemples de séquences, donnés ci après au § 2-4.3, ont été enregistrés avec le premier prototype de stéréoscope linéaire. Les caractéristiques de celui-ci étaient alors légèrement différentes de celles données précédemment, qui correspondent à la version finale du stéréoscope linéaire, implanté sur le véhicule expérimental Prolab 2 (Cf. chapitre 8).

Le tableau 2.1 indique les principales différences entre les deux versions du stéréoscope linéaire.

Caractéristiques		Premier stéréoscope	Second stéréoscope
Caméras	Définition	1728 pixels	2592 pixels
	Largeur du capteur	22,1 mm	25,92 mm
Objectifs		100 mm	35 mm
Implantation des caméras	Entre axe	1 m	0,465 m

Tableau 2.1 : Caractéristiques principales des deux stéréoscopes linéaires.

Par la suite, nous indiquerons avec quelle version du stéréoscope les séquences d'images ont été acquises.

2-4.3. Exemples de séquences enregistrées

Nous donnons ici des exemples de séquences enregistrées. Quelques précisions sont néanmoins nécessaires pour permettre leur interprétation.

Les séquences sont représentées sous la forme d'images bidimensionnelles où :

- L'axe horizontal représente le plan de visée du stéréoscope. Chaque ligne correspond donc à une image linéaire.
- L'axe vertical représente l'axe du temps, dirigé vers le bas de l'image.

Chaque séquence stéréoscopique est donc représentée à l'aide de deux images bidimensionnelles. La première image correspond à la séquence acquise par la caméra de gauche, la seconde à la séquence acquise par la caméra de droite.

Les séquences présentées sont accompagnées d'un texte descriptif afin d'aider le lecteur à les interpréter.

2-4.3.1. Exemple 1 : Piéton évoluant devant le stéréoscope

La figure 2.17 représente la première séquence enregistrée par le stéréoscope linéaire. Elle est devenue pour nous une image de référence pour la mise au point des différents algorithmes que nous détaillerons par la suite.

Le stéréoscope est monté sur un support fixe et vise la chaussée à une distance de 50 mètres. Les lignes blanches, clairement visibles, ont une largeur de 14 centimètres. Une voiture située hors du champ projette une ombre, visible sur la droite des images comme une zone entièrement noire.



Figure 2.17 : Evolution d'un piéton vu par le stéréoscope linéaire.

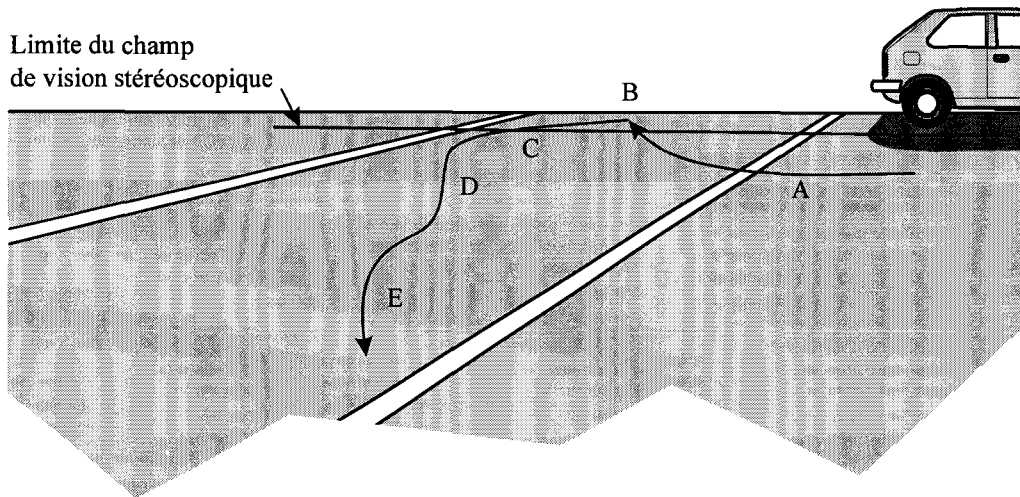


Figure 2.18 : Les déplacements du piéton de la figure 2.17.

La figure 2.18 décrit schématiquement les déplacements du piéton au cours du temps. On le voit d'abord se diriger vers son point de départ, situé un peu au-delà de l'intersection du plan de visée et de la chaussée (A). Ce faisant, il passe hors de portée du stéréoscope linéaire et, en conséquence, disparaît de l'image (B). Quelques secondes plus tard, il commence à se diriger vers le stéréoscope. Il réapparaît alors dans le champ de visée, en se dirigeant vers la caméra gauche (C), avant d'obliquer légèrement vers la caméra droite (D). Puis il part vers sa gauche et sort du champ de vision stéréoscopique (E).

2-4.3.2. Exemple 2 : Route dégagée et passage d'une sortie d'autoroute

Le stéréoscope linéaire est ici installé sur le toit d'un véhicule. La séquence a été prise par beau temps sur autoroute.

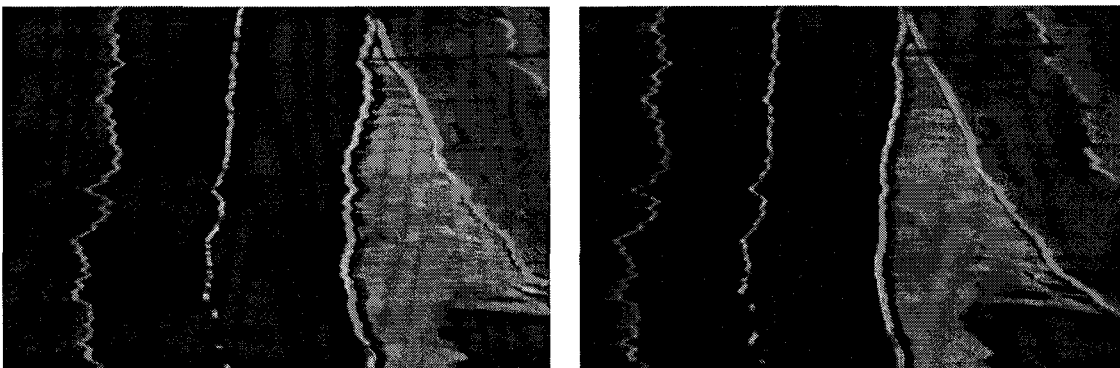


Figure 2.19 : Passage d'une sortie d'autoroute.

Cette séquence est représentative d'une chaussée sans le moindre obstacle à détecter (Cf. Figure 2.19). Nous longeons une sortie d'autoroute : le triangle gris que l'on voit sur la droite est le marquage au sol qui délimite cette sortie. Le ruban gris est l'image de la voie qui

s'éloigne de l'autoroute. Le contraste de l'image est important en raison de la présence de lignes blanches, mais aucun relief n'apparaît. Tout ce que l'on voit sur cette séquence est situé dans le plan de la chaussée.

2-5. Conclusion

La première partie du projet, réalisée par Jean-Luc Bruyelle, s'achève ici. A ce stade nous disposons des éléments suivants :

- Un dispositif de prise de vue stéréoscopique permettant d'obtenir les images de la scène étudiée : le *stéréoscope linéaire*.
- Des relations géométriques permettant de déterminer la position des objets à partir de leurs images sur les différents capteurs.
- Une méthode de calibrage permettant de fixer les paramètres de calibration, utilisés dans les relations géométriques.
- D'un dispositif permettant l'enregistrement de séquences qui serviront de base pour le développement de nos algorithmes.

La seconde partie du travail qui nous a été confié consiste à analyser les images issues du stéréoscope linéaire. Avant de nous lancer dans la description de nos travaux, il nous a paru judicieux de présenter une étude générale consacrée aux principes fondamentaux de la stéréovision. Cette étude fait l'objet du chapitre suivant.

Chapitre 3

PRINCIPES FONDAMENTAUX DE LA STEREOVISION

Les systèmes stéréoscopiques conçus par les scientifiques s'inspirent du système binoculaire humain. Si le stéréoscope linéaire que nous avons présenté au chapitre précédent est différent des systèmes stéréoscopiques courants, il a été créé dans le même but percevoir la structure tridimensionnelle de l'environnement dans lequel il évolue. Le problème principal auquel on se trouve confronté lorsqu'on développe un système stéréoscopique est le suivant : A partir de deux vues, au moins, d'une même scène et prises sous des angles différents, quelles opérations faut-il réaliser pour déterminer les coordonnées de tous les éléments présents dans la scène? Le but de ce chapitre est d'exposer en détail les démarches habituellement employées pour reconstruire, à partir d'un système stéréoscopique, les paramètres tridimensionnels d'une scène.

3-1. Approche générale

3-1.1. Le Paradigme de Marr

David Marr [Marr 82] a établi, en étudiant la vision humaine et animale, un concept connu sous le nom de *Paradigme de Marr*. Ce concept propose un modèle calculatoire pour le traitement et la représentation de l'information. Les principales lignes de ce concept, tirées de [Horaud 93], sont les suivantes :

- A partir d'une ou de plusieurs images un processus d'extraction de caractéristiques produit une description en termes d'attributs bidimensionnels. Ce niveau de représentation est appelé *première ébauche (primal sketch)*.
- La première ébauche constitue l'entrée d'un certain nombre de processus plus ou moins indépendants qui déterminent des propriétés tridimensionnelles locales relatives à la scène. Il s'agit d'une représentation centrée sur l'observateur, appelée *ébauche 2.5D*. Ces processus opèrent sur une séquence d'images telles qu'on les rencontre en analyse de scènes dynamiques, sur une paire d'images stéréoscopiques ou sur une seule image. Dans ce dernier cas, il s'agit de processus d'inférence qui utilisent des connaissances spécifiques. Dans le cas d'une analyse de contour, les connaissances sont uniquement géométriques. Elles sont à la fois géométriques et statistiques pour les processus d'analyse de textures. L'étude des ombrages et reflets utilise respectivement les connaissances photométriques et colorimétriques.
- L'ébauche 2.5D est mise en correspondance avec des connaissances 3D afin de construire une description de la scène en terme d'objets et de relations entre ces

derniers. Il s'agit maintenant d'une représentation centrée sur la scène : la description ne dépend plus de la position de l'observateur.

3-1.2. Organisation d'un système stéréoscopique

En pratique, le paradigme de Marr peut s'exprimer par les trois étapes de traitement suivantes : segmentation, reconstruction, reconnaissance. La structure de la majorité des systèmes stéréoscopiques s'en inspire et s'organise plus précisément de la façon suivante :

- Acquisition des images stéréoscopiques.
- Extraction des primitives dans chaque image.
- Mise en correspondance des primitives.
- Triangulation géométrique.
- Reconstruction tridimensionnelle de la scène.

Toutes ces étapes s'effectuent dans l'ordre et ont leur importance. Cependant la première de ces étapes concerne la prise de vue et ne fait pas exactement partie des traitements. Néanmoins le modèle géométrique du capteur est utilisé pour la reconstruction et la triangulation. Cet aspect n'a pas demandé de recherche spécifique puisque dans le cadre de notre application nous avons déjà un capteur stéréoscopique : le stéréoscope linéaire. Toutefois, nous le présenterons brièvement au § 3-2.

3-1.3. Mise en correspondance

Parmi toutes les étapes citées précédemment, la mise en correspondance des primitives reste le problème central de tous les algorithmes de stéréovision et notamment le thème principal de nos travaux. Sans cette étape clef, la perception du relief n'est pas réalisable et donc aucune reconstruction n'est envisageable. Les principaux aspects qui conditionnent le succès de la mise en correspondance sont :

- Le choix des primitives : certains types de primitives sont plus aisés à apparier que d'autres. De plus, ce choix détermine la représentation finale de la scène. La notion de primitive sera définie au paragraphe 3-3 au cours duquel nous exposerons les différentes primitives utilisées en stéréovision.
- Les conditions d'appariement : Un certain nombre de critères et de contraintes doivent être satisfaits afin de mesurer la qualité et la validité des appariements. Les contraintes employées dans la littérature pour faciliter la mise en correspondance seront détaillées au paragraphe 3-4.

- Les algorithmes : Ceux-ci sont très nombreux et utilisent des méthodes différentes. Le paragraphe 3-5 sera l'objet de la mise en correspondance au sens exact du terme. Il donnera notamment une liste détaillée des diverses méthodes généralement employées.

3-2. Prise de vue

3-2.1. Les capteurs

L'entrée d'un système stéréoscopique est constituée de capteurs : les caméras. L'approche la plus courante pour percevoir le relief consiste à utiliser deux vues différentes. C'est ce qu'on appelle la *vision binoculaire*. Cependant certains auteurs préconisent l'utilisation d'une troisième caméra, on parle alors de *vision trinoculaire* (Cf. § 3-5.1).

Il existe deux modèles de caméras possédant des caractéristiques géométriques différentes :

- Les caméras matricielles. Ce sont celles qui sont le plus fréquemment employées. Le capteur de ces caméras est constitué d'une matrice d'éléments photosensibles et est à la base de toutes les caméras vidéo standards que l'on trouve dans le commerce. Elles fournissent une image bidimensionnelle identique aux images télévisuelles.
- Les caméras linéaires. Ces caméras décrites au chapitre 2 (Cf. § 2-2.1) fournissent une image monodimensionnelle, c'est à dire une image vidéo constituée d'une seule ligne. Ce type de capteur est habituellement utilisé dans les milieux industriels, il est à la base de nombreux systèmes visuels d'inspection [Macaire 93]. Son usage dans un système stéréoscopique est cependant assez rare (Cf. § 3-7), ce qui rend notre travail original.

3-2.2. Géométrie des capteurs

La géométrie des capteurs est un élément important du dispositif de prise de vue car elle conditionne la suite des traitements stéréoscopiques.

Les calculs de triangulation géométrique permettent de déterminer les coordonnées d'un point de la scène dans un repère associé à cette scène. Ces calculs nécessitent la connaissance des coordonnées de l'image de ce point dans le repère lié à la caméra gauche mais aussi celles dans le repère lié à la caméra droite. Il est donc impératif de déterminer les relations géométriques permettant de passer du repère caméra gauche au repère caméra droite ainsi que celles permettant de passer des repères caméras au repère associé à la scène. La détermination

de ces relations s'effectue lors de la phase de calibrage. Cette phase particulièrement importante permet soit de déterminer les paramètres caractérisant la disposition des caméras, soit au contraire d'imposer une configuration précise au capteur en imposant les paramètres voulus.

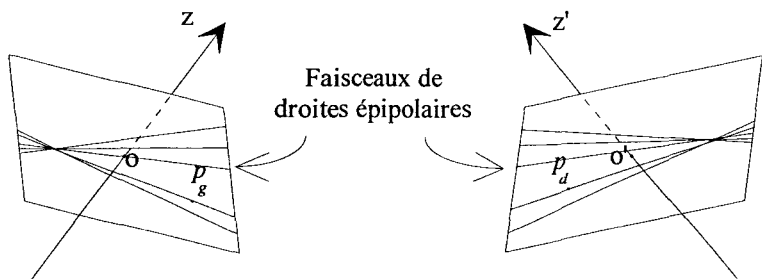


Figure 3.1 : Système stéréoscopique dont les capteurs ont une disposition et orientation quelconque.

Dans le premier cas, la disposition et l'orientation des caméras peuvent être quelconques (la condition minimum à respecter étant bien sur la convergence des axes optiques vers la scène observée) (Cf. Figure 3.1). Lorsque les paramètres sont connus, ceux-ci sont utilisés dans la recherche des couples à appairer. Ils font alors appel à la notion de droite épipolaire. Etant donné un point p_g dans l'image gauche, on cherche dans l'image droite son correspondant p_d . Le point p_d appartient nécessairement à une droite de l'image droite entièrement définie par la donnée de p_g . Cette droite est appelée **droite épipolaire** associée à p_g . Le problème étant symétrique, il existe une **droite épipolaire** associée à p_d , on l'appelle **droite épipolaire conjuguée**. Les équations de ces droites sont déterminées en fonction des paramètres évalués lors du calibrage du système stéréoscopique. L'ensemble de ces droites forme un faisceau d'épipolaires.

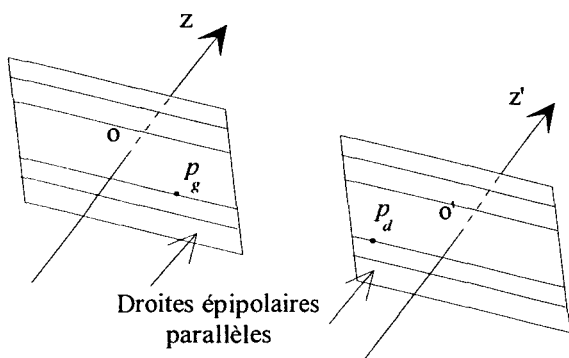


Figure 3.2 : Configuration particulière des capteurs.

Dans le second cas, on impose généralement la disposition avantageuse où les axes optiques sont parallèles, les capteurs sont coplanaires et leurs lignes de balayage parallèles (Cf. Figure 3.2). Les droites épipolaires prennent ici une configuration particulière

puisqu'elles constituent un ensemble de droites parallèles. Cette disposition simplifie considérablement les calculs et assure une meilleure précision dans la localisation des éléments de la scène tridimensionnelle. Cependant les méthodes d'ajustage utilisées pour aboutir à cette configuration sont parfois longues et difficiles. Pour ces raisons, certains auteurs préfèrent fixer une disposition quelconque à leurs capteurs et se ramener de façon analytique à une configuration de droites épipolaires parallèles en effectuant une *rectification épipolaire* qui consiste à appliquer aux images une transformation linéaire en coordonnées projectives. L'ensemble de ces modifications implique bien sûr un accroissement des temps de calculs.

3-2.3. Caractéristiques supplémentaires d'un dispositif de prise de vue

Outre le choix du type de capteur et de la géométrie des axes optiques : parallèles ou convergents, d'autres choix doivent être effectués. Parmi les plus importants nous pouvons citer :

- La distance entre les centres optiques (entre axe).
- L'angle d'ouverture des objectifs (distance focale).

Ces deux paramètres seront utilisés dans les relations mathématiques permettant de réaliser la triangulation géométrique (Cf. § 2-2.2.3), mais ils permettent de fixer d'autres caractéristiques :

- La distance minimale de prise de vue permettant d'appliquer le principe de stéréovision.
- La largeur de la scène observable à une distance donnée.

Il faut donc choisir judicieusement ces différents paramètres afin d'optimiser le système de vision en fonction du domaine d'application.

Remarque : Pour plus d'informations, le lecteur intéressé pourra consulter les ouvrages [Ayache 89] et [Horaud 93] qui traitent dans le détail les problèmes de calibrage, de géométrie des capteurs, des notions de droites épipolaires et de rectification épipolaire.

3-3. Primitives stéréoscopiques

3-3.1. Notion de primitives

La stéréovision nécessite l'appariement de primitives extraites de chacune des deux images. Ces primitives représentent des éléments visuels caractéristiques de l'image. En effet, il n'est

pas envisageable d'essayer de déterminer pour chaque pixel d'une image son correspondant dans l'autre image pour les raisons suivantes :

- Le temps de calculs : Si l'on considère les images linéaires de 2592 pixels utilisées dans cette étude, la mise en correspondance de chaque pixel nécessiterait alors 2592×2592 tests, soit 6,7 millions d'opérations. Même si ce chiffre est 10.000 fois plus faible que dans le cas d'images matricielles standards, cette approche n'est pas réaliste dans le cadre d'une application temps réel car elle exigerait beaucoup trop de calculs.
- Le faible contenu informationnel : Un pixel unique n'est caractérisé que par son niveau de gris, attribut insuffisant pour permettre un appariement fiable et sans ambiguïté. En effet, un pixel d'une des images serait apparié à tous les pixels de l'autre image qui présenteraient le même niveau de gris.

Il est donc essentiel de traiter préalablement les images afin d'en extraire un nombre limité de primitives, riches en contenu informationnel, afin de pouvoir faciliter la procédure d'appariement et de diminuer les temps de traitement.

D'autre part, le choix des primitives à appairer conditionne le type de reconstruction de la scène. Après la mise en correspondance, la phase de modélisation peut être immédiate si le résultat de la triangulation des primitives donne directement des modèles tridimensionnels utiles. Toutefois, des étapes intermédiaires de mise en forme des données sont parfois nécessaires suivant le niveau de représentativité des primitives appariées. Pour ces raisons, il est très important de choisir les primitives en fonction des critères suivants :

- Elles doivent être représentatives de la scène et permettre de reconstruire des points ou objets physiques observés.
- Elles doivent être faciles à extraire, et stables d'une image à l'autre.
- Leur appariement doit être aisé et le moins ambigu possible, ce qui dépend des attributs qu'on peut leur associer.

3-3.2. Type de primitives

Nous avons vu précédemment l'importance du choix de primitives adaptées. Ce paragraphe a pour objet de dresser une liste non exhaustive des principales primitives utilisées en analyse d'images matricielles. Nous verrons, par la suite, celles qui peuvent être appliquées aux images linéaires.

3-3.2.1. Primitives de type point

Malgré la remarque faite précédemment (Cf. § 3-3.1), certains pixels peuvent être considérés comme primitives si leurs voisinages répondent à certains critères. Ces points remarquables sont en général caractérisés par des attributs de nature photométrique.

3-3.2.1.1. Points de Moravec

Ces points [Moravec 77], [Moravec 81] sont ceux autour desquels est observée une variation forte de la fonction niveau de gris. Une mesure d'intérêt est définie comme la valeur minimale σ° de la variance σ du niveau de gris dans le voisinage du point, calculée dans quatre directions. On a alors $\sigma^\circ = \text{Min} (\sigma_h, \sigma_v, \sigma_{d1}, \sigma_{d2})$ où σ_h , σ_v , σ_{d1} , σ_{d2} représentent respectivement les variances dans les directions horizontale, verticale et dans les deux diagonales d1 et d2. On sélectionne alors les points dont la mesure d'intérêt est maximale. Cette procédure élimine les points appartenant à des régions trop homogènes ou faiblement texturées et les points se trouvant sur des contours lisses c'est à dire pour lesquels la variance est faible dans le sens du contour. On sélectionne ainsi en premier lieu des points à fort contenu informationnel, tels ceux appartenant à des coins.

3-3.2.1.2. Les points de courbure ou coins

Plusieurs approches pour obtenir ces points caractéristiques ont été présentées dans la littérature. Elles peuvent être divisées en deux groupes.

Le premier consiste à travailler directement sur le niveau de gris de l'image. Les détecteurs de coins sont basés sur une mesure de la courbure, définie comme le produit de l'amplitude du gradient et du taux de variation dans la direction de ce gradient. Les coins sont ensuite obtenus par seuillage. Parmi les détecteurs de coins les plus populaires, nous pouvons citer ceux proposés par Zuniga et Haralick [Zuniga 83], ainsi que Kitchen et Rosenfeld [Kitchen 82]. Une approche récente est celle développée par Noble [Noble 87].

Les approches du second groupe consistent d'abord à extraire les contours en tant que chaînes codées et ensuite à ne chercher parmi ces chaînes que les points présentant une courbure importante [Asada 86].

Une nouvelle approche proposée par Deriche [Deriche 90a] combine les deux types de concepts précédents en une seule procédure. Après extraction des points de contours, un algorithme de chaînage est utilisé afin d'obtenir des ensembles de points connectés. Pour

chaque point des courbes obtenues, la direction du contour, calculée à partir de la fonction niveau de gris de l'image, est utilisée pour déterminer la courbure. Les points de contours caractérisés par une forte courbure seront candidats à l'appariement.

3-3.2.2. Primitives de type contour

Les primitives de type contour sont certainement celles qui sont le plus utilisées en stéréovision pour les raisons suivantes :

- Leur représentativité : Le contour est la représentation, dans l'image, des frontières des objets et des variations spatiales des propriétés géométriques et physiques de leurs surfaces (orientation, matière, texture, couleur, etc.). De cette façon, les éléments significatifs d'une image peuvent être facilement mis en évidence.
- Facilité d'extraction : On trouve aujourd'hui dans la littérature de nombreux algorithmes de détection de contours dont certains seront cités dans le chapitre suivant (Cf. § 4-3).

Il faut cependant remarquer que la notion de contour est double. Elle englobe d'une part l'ensemble des points constituant le contour et d'autre part la modélisation des contours des objets à partir de ces points de contours.

3-3.2.2.1. Points de contour

Les points de contours s'obtiennent directement par l'application d'un opérateur de détection de contours. Cependant, l'image obtenue est souvent inexploitable. En effet, si elle contient effectivement un ensemble de points situés au niveau des contours des objets, cette description n'est jamais parfaite. Des points indésirables viennent la plupart du temps s'ajouter aux points de contour effectifs. D'autre part, certains points des contours ne sont pas détectés. Il faut également indiquer que, le nombre de points de contour obtenus est souvent très important, ce qui rend les algorithmes de mise en correspondance inopérants. Les primitives de type points de contour ne sont donc pas utilisables à l'état brut et il est nécessaire de chaîner ces points afin de les rendre exploitables sous la forme de contours.

3-3.2.2.2. Modélisation du contour à partir des points de contour

La première étape pour modéliser les contours consiste à relier les points de contours voisins de proche en proche. On passe ainsi à un niveau supérieur où les contours sont définis comme des chaînes de points connectés. Pour simplifier la description, les chaînes peuvent être

modélisées sous la forme de segments, d'arcs de cercle ou de polygones suivant les cas. De cette manière, on parvient à une représentation synthétique des objets par une modélisation de leur contour. Différents attributs de type photométrique et géométrique peuvent être associés aux contours ce qui en fait des primitives intéressantes pour l'appariement.

3-3.2.3. Primitives de type région

Le choix de primitives de type région permet de décrire une image avec un nombre très limité d'éléments, contrairement aux primitives de type point qui doivent être en quantité suffisante pour représenter correctement une scène. L'utilisation de ce type de primitive diminue les possibilités d'appariements et réduit donc les temps de traitement. De plus, le contenu informationnel de ces primitives est plus riche. Les attributs associés aux régions peuvent être de nature photométrique et géométrique (forme, surface). Les ambiguïtés d'appariement peuvent ainsi être aisément levées. Un inconvénient cependant est à noter. Les effets de perspective, dus aux images saisies sous des angles différents, entraînent des déformations de ces régions, ce qui rend parfois les appariements difficiles. Citons néanmoins quelques exemples de régions.

3-3.2.3.1. Régions homogènes

L'homogénéité peut se situer au niveau photométrique. Les régions formées sont alors constituées d'un ensemble de pixels connexes dont le niveau de gris est uniforme.

Des images fortement texturées peuvent être décrites par des régions de textures homogènes. Ce type de description est souvent plus synthétique que celui obtenu à partir de primitives de type contour qui sont dans ce cas très nombreuses [Lumia 83]. Il faut cependant préciser que l'analyse de texture est souvent très lente en raison de l'utilisation d'outils tels que la transformée de Fourier ou les matrices de co-occurrence.

3-3.2.3.2. Régions obtenues par segmentation couleur

Depuis quelques années, l'intérêt des chercheurs pour la couleur s'est considérablement accru. Les progrès technologiques, autant au niveau du matériel traitant la couleur qu'au niveau de la puissance de calcul des stations de travail, et la baisse des coûts du matériel permettent aujourd'hui d'envisager le traitement d'images couleur pour des applications où ce n'était pas encore pensable il y a quelques années.

De nombreux travaux ont permis de développer des outils permettant l'analyse d'images couleurs [Ultré 94], [Lee 91], [Claxton 87], [Di Zenso 86]. Ce type d'approche, bien que lourd en temps de calcul puisqu'il implique de travailler sur trois plans images, pourrait être d'un grand intérêt pour l'appariement de primitives. Il ne faut pas oublier que l'oeil humain qui perçoit les couleurs, les utilise certainement pour différencier les objets et percevoir le relief.

Dans le cadre de notre étude, nous n'envisagerons pas ce type de segmentation puisque nous utilisons des caméras dont les images sont constituées de 256 niveaux de gris.

3-3.2.4. Primitives de haut niveau : les contours actifs

Une autre approche consiste à utiliser comme primitives des contours actifs ou « *snakes* ». Introduits par Terzopoulos [Terzopoulos 87], les contours actifs peuvent être comparés à des élastiques. Ceux-ci sous l'action de forces externes et internes vont venir s'ajuster sur les contours des objets et permettre ainsi de les modéliser. Cette approche s'appuie sur les contours présents dans l'image. L'avantage procuré par leur utilisation est la résolution de nombreux problèmes de traitement d'images tels que la fermeture des contours et la modélisation de contours subjectifs.

Dans sa thèse, Selsis [Selsis 95] applique les contours actifs à des couples d'images stéréoscopiques sur lesquelles un prétraitement ne laisse subsister que les contours des objets en mouvement présents devant les caméras [Vieren 88], [Cabestaing 92]. Les contours actifs permettent d'abord de modéliser les objets mobiles présents dans la scène et servent ensuite de primitives pour l'appariement.

Ce type de primitive, que l'on peut qualifier de haut niveau, permet d'avoir une connaissance précise des objets étudiés avant leur appariement. Cette méthode évite des traitements de reconstruction, après la mise en correspondance, puisque les objets ont déjà été modélisés.

3-3.3. Conclusion sur les primitives

Nous venons de dresser un éventail des primitives généralement utilisées en vision matricielle. En vision linéaire, toutes ces primitives ne sont pas utilisables, nous verrons, dans le chapitre 4, comment elles apparaissent sur un capteur linéaire afin de déterminer celles qui sont réellement exploitables.

L'extraction des primitives constitue la première étape des traitements dans un système stéréoscopique. Cette première phase permet d'obtenir pour chaque image les listes de candidats qui alimenteront la procédure de mise en correspondance.

3-4. Contraintes de mise en correspondance

3-4.1. Utilité des contraintes

La mise en correspondance des primitives extraites des images gauche et droite nécessite de mettre en oeuvre des critères de ressemblance (Cf. § 3-6.2) pour pouvoir décider si les appariements sont corrects ou non. L'utilisation de contraintes permet souvent de lever les ambiguïtés. Il existe en effet des relations géométriques que doivent respecter les positions des primitives gauches et droites correspondant au même objet physique de la scène. Nous présentons, ci-dessous, une liste non exhaustive des principales contraintes utilisées dans la littérature.

3-4.2. Contrainte épipolaire

La contrainte épipolaire qui est la plus couramment employée est la seule contrainte géométrique intrinsèque aux capteurs stéréoscopiques. Elle est très souvent utilisée en stéréovision car elle simplifie la recherche des correspondants lors de l'appariement des primitives. Comme nous l'avons vu au § 3-2.2, deux primitives correspondantes se trouvent nécessairement sur des droites épipolaires conjuguées. L'application de la contrainte épipolaire permet donc une réduction de la recherche des primitives homologues aux droites épipolaires. La zone de recherche passe de deux à une seule dimension, diminuant le nombre de candidats possibles de façon significative. En employant cette contrainte avant l'évaluation du critère de ressemblance, on évite de nombreux calculs inutiles.

Cette contrainte s'applique donc aisément aux primitives de type point. Quant aux primitives de type contour, modélisées généralement par des segments de droite, cette contrainte est utilisée de la façon suivante : un couple de segments (S_g, S_d) vérifie la contrainte épipolaire si et seulement si le plan épipolaire passant par le milieu de S_g intersecte S_d . Une définition symétrique s'applique à un couple (S_d, S_g). De la même manière, on peut étendre cette définition aux primitives de type région.

3-4.3. Contraintes locales

Les contraintes locales sont déterminées en fonction des types d'attributs (photométriques ou géométriques) caractérisant les primitives. Elles sont dites locales car elles n'utilisent pas de relation faisant intervenir les primitives de la même image.

3-4.3.1. Contrainte d'orientation

La contrainte d'orientation est une contrainte sur les positions géométriques des primitives dans l'image. Elle impose aux primitives gauches et droites, que l'on tente d'apparier, de présenter des orientations compatibles.

Pour des segments de droite, les orientations sont déterminées en estimant les vecteurs directeurs de ces segments dans les repères images. Pour les primitives de type point de contour, l'information d'orientation est donnée par la tangente au contour pour le point considéré. Lorsqu'on apparie des régions, les directions des axes principaux et secondaires peuvent être utilisées. Il faut toutefois émettre une réserve, car les effets de perspectives, dus aux différents angles de prise de vue, entraînent parfois des déformations des régions pouvant rendre cette contrainte inapplicable.

Tout comme la contrainte épipolaire, il est préférable d'appliquer cette contrainte avant l'évaluation d'un critère de ressemblance. Si les orientations des primitives ne respectent pas la contrainte alors, ces primitives ne peuvent être les images d'un même objet de la scène.

Il faut noter que cette contrainte d'orientation n'a pas d'existence en vision linéaire car la notion d'orientation n'existe pas dans un signal monodimensionnel.

3-4.3.2. Contrainte de continuité figurale

La contrainte de continuité figurale est simple et efficace à mettre en oeuvre lorsque les primitives à mettre en correspondance se trouvent le long des contours des objets (points ou segments modélisant les contours). Si on suppose que la scène est constituée d'objets continus localement par morceaux, l'ensemble des primitives qui les modélisent doivent présenter une continuité identique. L'utilisation de cette contrainte impose donc un traitement régulier des lignes épipolaires. La contrainte de continuité permet alors de propager d'une ligne épipolaire à l'autre des hypothèses de mise en correspondance, c'est à dire que les primitives appariées sur une ligne épipolaire serviront à confirmer les appariements de la ligne épipolaire suivante. Il s'agit alors de privilégier les appariements qui prolongent un contour mis en évidence par les appariements réalisés en traitant les lignes épipolaires précédentes. Les nouveaux appariements sont donc validés en fonction des précédents ce qui permet d'assurer une certaine cohérence.

Mohan dans [Mohan 89] conseille de ne pas utiliser cette contrainte dans un raisonnement local, car elle peut être une source d'erreur. Il suggère de l'employer dans un contexte plus

global où l'on tient compte de l'ensemble des appariements de la droite épipolaire, de l'ensemble des contours déjà formés à partir des appariements réalisés en analysant les lignes épipolaires précédentes et d'autres contraintes plus générales comme la contrainte d'ordre par exemple (Cf. § 3-4.4.1).

3-4.3.3. Contrainte de continuité de la disparité

La contrainte de continuité de la disparité peut souvent être ajoutée à la contrainte précédente car elle s'appuie sur les mêmes hypothèses de continuité des surfaces. Cependant elle suppose également que les objets soient faiblement inclinés par rapport à l'axe optique du stéréoscope. De cette manière, les différentes surfaces présentent de faibles variations de leur profondeur. La profondeur étant directement liée à la disparité (Cf. § 2-2.2.3, Equation 2.4), il en résulte que les disparités doivent varier continûment le long des surfaces des objets.

Lorsque cette contrainte est associée à la précédente, un couple de primitives appariées est validé s'il prolonge bien un contour déjà existant et si sa disparité est compatible avec celle du couple précédent. On suppose, en effet, que la disparité le long d'un contour varie lentement et ne présente pas de discontinuité le long de ce contour.

Comme auparavant, cette contrainte peut être une source d'erreur si elle est appliquée localement. Dans un raisonnement global, l'ensemble des couples appariés doit donner naissance à un champ de disparité variant continûment. Ceci suppose bien sûr que la densité des primitives modélisant la scène est assez grande pour obtenir un champ exploitable.

3-4.4. Contraintes globales

Les contraintes locales sont parfois insuffisantes pour résoudre les ambiguïtés engendrées par la mise en correspondance. Il est donc nécessaire d'imposer des contraintes plus générales qui permettent de vérifier la compatibilité des appariements entre eux.

3-4.4.1. Contrainte d'ordre

Cette contrainte a été utilisée pour la première fois par Baker [Baker 81] pour éviter l'explosion combinatoire due aux multiples possibilités de mise en correspondance. Elle suppose que l'ordre des primitives le long des droites épipolaires conjuguées est le même dans chacune des deux images du couple stéréoscopique (Cf. Figure 3.3).

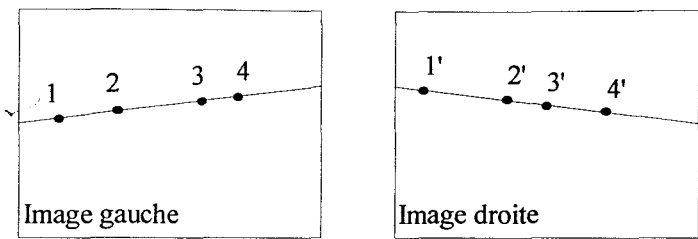
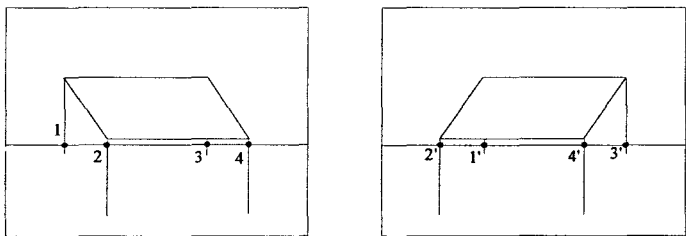
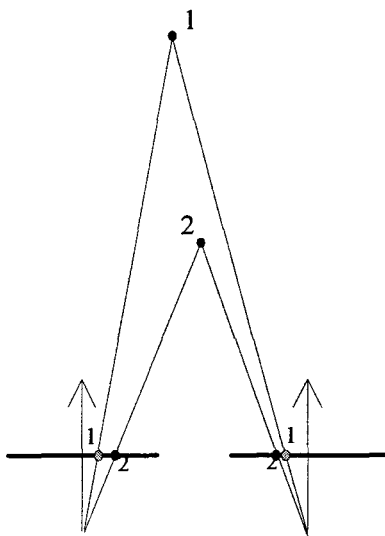


Figure 3.3 : La contrainte d'ordre.

Si celle-ci est toujours vérifiée lorsqu'il s'agit d'apparier des images aériennes, elle est parfois transgressée dans les scènes d'intérieur où l'on peut rencontrer des objets verticaux de type fil ou barre (Cf. Figure 3.4a). D'autre part dans les cas où il existe de grandes variations de disparité entre deux objets disjoints, des inversions peuvent se produire (Cf. Figure 3.4b).



a) pieds de table dans une scène d'intérieur.



b) objet proche et objet lointain.

Figure 3.4 : Exemples où la contrainte d'ordre n'est pas respectée.

Enfin la présence d'objets transparents dans une scène rend également cette contrainte inutilisable (Cf. Figure 3.5). Cependant il faut préciser que les cas d'objets transparents sont assez rares dans les scènes naturelles.

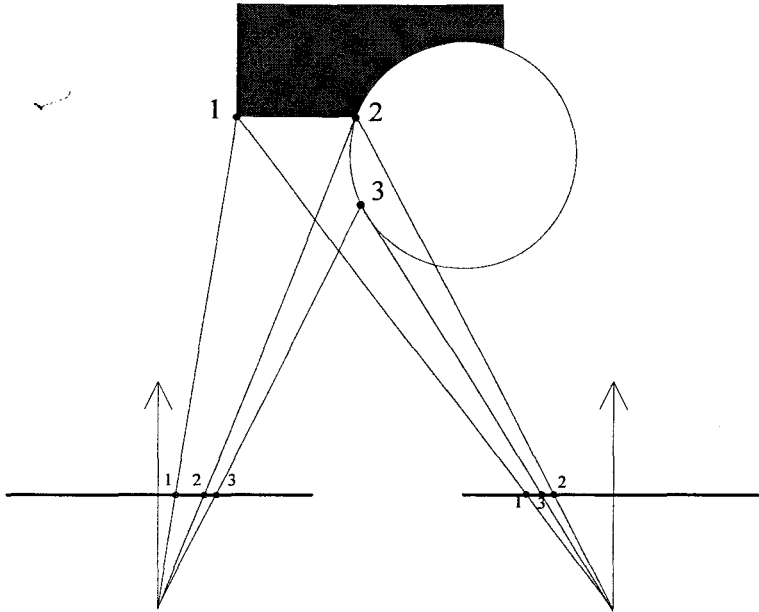


Figure 3.5 : Contrainte d'ordre non respectée dans le cas d'un objet transparent.

Puisqu'elle n'est pas toujours vérifiée, la contrainte d'ordre doit donc être utilisée avec beaucoup de précautions dans un algorithme de mise en correspondance. Il faut cependant remarquer que les algorithmes basés sur les méthodes de programmation dynamique (Cf. § 3-6.3.2) prennent en compte implicitement cette contrainte.

3-4.4.2. Contrainte d'unicité

La contrainte d'unicité suppose qu'un point image de l'image gauche p_g n'a qu'un seul correspondant p_d dans l'image droite (une hypothèse symétrique peut être appliquée aux points de l'image droite). Si cette hypothèse semble évidente, il faut cependant noter qu'il existe des cas limites où elle n'est pas toujours vérifiée. En effet, la figure 3.6 nous montre que la contrainte d'unicité n'est pas respectée lorsque deux points se projettent en un même point dans une image et en deux points différents dans l'autre image. Dans ce cas, le point p pourra être apparié avec p'_1 mais aussi avec p'_2 . Si l'on considère ces deux appariements comme compatibles, on augmente la difficulté de la mise en correspondance, puisque excepté pour l'alignement représenté sur la figure 3.6, un point d'une image devrait avoir un et un seul correspondant dans l'autre image. Pour des segments, on peut trouver des situations similaires où la contrainte d'unicité n'est pas respectée.

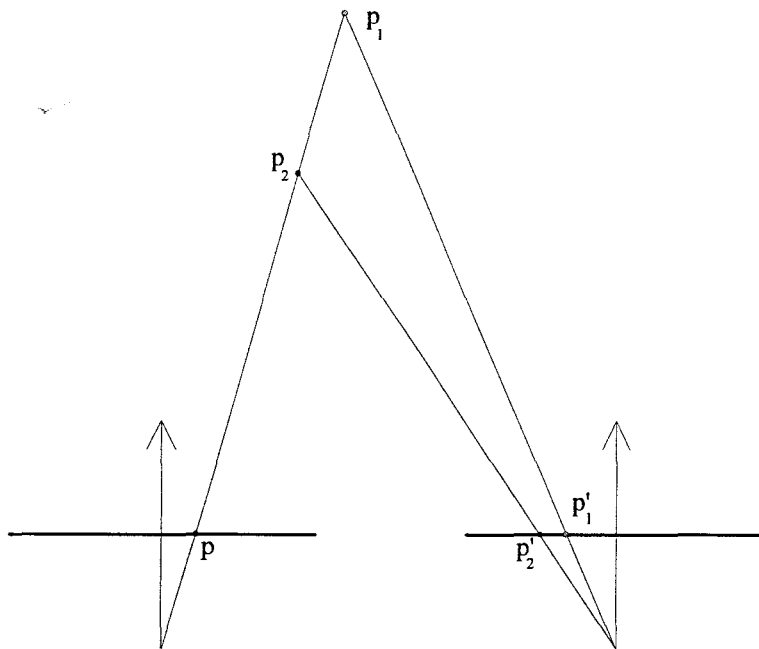


Figure 3.6 : La contrainte d'unicité n'est pas respectée.

En pratique, pour imposer la contrainte d'unicité qui facilite la mise en correspondance, on exclut ce type d'alignement accidentel. Si on se réfère à la figure 3.6, on considère généralement que l'objet p_2 , c'est à dire l'objet le plus proche, cache l'objet p_1 , ainsi le point p est l'image de l'objet p_2 . L'objet p_1 n'a pas d'image sur le capteur gauche, le point p'_1 n'a pas de correspondant, on parle alors de phénomène d'occlusion.

3-4.5. Conclusion sur les contraintes

Les mesures de ressemblance n'étant en général pas assez discriminantes, l'emploi de contraintes permet de pallier ce défaut. Pour chaque couple de primitives, les algorithmes de stéréovision opèrent habituellement de la manière suivante :

- Application de la contrainte épipolaire, à laquelle peut venir également s'ajouter toute contrainte permettant de juger la compatibilité des deux primitives à appairer comme la contrainte d'orientation par exemple. Cette phase permet d'économiser beaucoup de calculs puisque le traitement s'arrête pour le couple de primitive courant dès qu'une contrainte n'est pas respectée. Par contre, si toutes les contraintes sont vérifiées, on passe à la phase suivante.
- Calcul du critère de ressemblance pour la mise en correspondance. Seuls les couples de primitives pour lesquels le critère est satisfaisant sont considérés comme valides. Les autres appariements sont rejetés. Quand une primitive est appariée plusieurs fois, on parle d'appariements multiples, deux cas peuvent se produire suivant les

algorithmes :

- *Un seul couple* est conservé : celui présentant le critère de ressemblance jugé le meilleur.
- *Plusieurs couples* sont sélectionnés : le meilleur couple est alors déterminé ultérieurement.
- Vérification de la vraisemblance des appariements en appliquant des contraintes globales comme la contrainte d'ordre ou d'unicité. En outre, on peut conforter les résultats en comparant les appariements avec ceux obtenus dans les images précédentes lorsqu'il s'agit d'images dynamiques ou avec ceux des lignes épipolaires voisines en appliquant la contrainte de continuité figurale. De cette manière, on peut généralement lever la majorité des ambiguïtés dues aux appariements multiples.

Si la contrainte épipolaire est essentielle pour réduire le nombre de candidats dans la recherche des primitives homologues, il convient néanmoins de faire un usage modéré des autres contraintes. Celles-ci ne peuvent être utilisées systématiquement, elles doivent être sélectionnées en fonction du type d'images traitées. Par exemple, la contrainte d'ordre ne peut être utilisée pour des scènes d'intérieurs dans lesquelles on peut rencontrer des objets verticaux longs et fins comme des pieds de table.

3-5. Approches originales pour percevoir le relief

La stéréovision constitue l'approche la plus répandue pour reconstruire une scène tridimensionnelle. Cependant il existe d'autres méthodes originales que nous aurions pu envisager d'utiliser.

3-5.1. La vision dynamique

3-5.1.1. Principe de la vision dynamique

Une autre approche procède comme l'être humain pour qui le mouvement aide à la perception du relief. La méthode consiste alors à analyser une séquence d'images provenant d'une seule caméra. Les mouvements des indices visuels dans la scène permettent alors de reconstituer des scènes tridimensionnelles.

D'après Ullman [Ullman 79], l'extraction de l'information 3D à partir des scènes en mouvement peut être réalisée en deux étapes :

- La première étape mesure le champ de déplacement des objets.
- La seconde étape détermine le mouvement et la structure 3D.

Il faut cependant remarquer que les auteurs ne sont pas tous unanimes sur la séparation de ces deux étapes. Il existe des approches qui diffèrent de celle donnée ici, notamment dans [Negahdaripour 85].

Pour résoudre le problème on émet un certain nombre d'hypothèses simplificatrices. On suppose connus les mouvements de la caméra qui peut d'ailleurs être fixe. La fréquence d'acquisition est également un paramètre important à connaître. Si celle-ci est élevée, les déplacements des objets seront faibles d'une image à l'autre, l'estimation de leur mouvement en sera donc facilitée. Cependant les mesures ne seront pas très précises. Une fréquence d'acquisition plus faible permettra une estimation plus précise du mouvement des différents objets mais nécessitera des traitements plus complexes pour la mesurer.

3-5.1.2. Les différents types de méthodes

Il existe deux grands types de méthodes pour analyser une séquence et en extraire une mesure du champ de déplacement.

3-5.1.2.1. Approche par mise en correspondance

Cette approche consiste à apparier des primitives mais entre les images successives de la séquence. On retrouve donc toutes les méthodes classiques de stéréovision que nous détaillerons dans un prochain paragraphe (Cf. § 3-6.3). La mise en correspondance est cependant plus simple à réaliser car les images successives "*se ressemblent*" plus qu'en vision binoculaire. Il n'y a pas d'effets de perspective, moins d'occlusions et, l'éclairement des objets est le même d'une image à l'autre. Le suivi des primitives appariées dans les images successives permet d'avoir une estimation du champ de déplacement. Celui-ci sera d'autant plus précis que la densité des primitives est grande. Garnousset [Garnousset 86] effectue le suivi de points particuliers le long des contours à l'aide d'une méthode corrélative. Barnard [Barnard 80] travaille, quant à lui, avec des points de Moravec.

A partir de ces appariements, le calcul de la structure de la scène est donc possible si on connaît le déplacement de la caméra entre deux acquisitions successives.

3-5.1.2.2. Méthodes de vision dynamique sans mise en correspondance

Ces méthodes considèrent dès le départ la séquence d'images comme un signal à 3 dimensions : x, y, t , c'est à dire les deux dimensions spatiales de l'image auxquelles on adjoint l'aspect temporel.

3-5.1.2.2.1. Les méthodes différentielles

La détection du mouvement peut être obtenue par une simple soustraction entre deux images successives. Pour pouvoir être appliquées, ces méthodes font les hypothèses suivantes : la caméra est fixe, l'illumination de la scène est constante et un point est perçu toujours avec la même intensité malgré son déplacement. De cette manière, une différence d'image met en évidence les variations dues au mouvement. Ces méthodes sont néanmoins sensibles aux bruits et laissent souvent apparaître le fond de la scène. Vieren [Vieren 88] propose une méthode pour extraire les objets en mouvement sans faire apparaître la texture du fond.

Une analyse sur plusieurs images permet ensuite d'obtenir les trajectoires des objets en mouvement.

3-5.1.2.2.2. Calcul du flot optique

Le flot optique est défini comme le champ des variations temporelles de l'intensité. Comme pour les méthodes différentielles, on émet les mêmes hypothèses auxquelles on ajoute une contrainte d'harmonie des vitesses, c'est à dire que des points voisins dans l'image plane se déplacent de façon similaire [Horn 81]. On trouvera dans [Horn 81] et [Horn 86] des techniques pour calculer le flot optique.

Le flot optique permet une étude plus complète des différents mouvements de la scène. Il donne une idée précise des différentes translations et rotations à une distance constante de l'observateur ou dans la direction de l'observateur mais au prix de calculs plus longs.

3-5.1.3. Conclusion sur la vision dynamique

Les méthodes de vision dynamique permettent d'analyser une scène 3D par l'estimation du mouvement. L'avantage par rapport au système stéréoscopique est de n'utiliser qu'une caméra, on évite donc les longues procédures de calibrage. Mais contrairement à un système stéréoscopique, l'analyse 3D de la scène ne peut se faire que si les objets sont mobiles. Dans le cas contraire, on peut faire évoluer la caméra mais il faut alors connaître précisément les déplacements de celle-ci pour pouvoir ensuite en extraire des informations.

Cette dernière condition impérative, nous empêche d'envisager ce type de méthode dans le cadre de notre projet. En effet, le système de détection d'obstacles sera intégré sur un véhicule. Or les mouvements de celui-ci ne peuvent être parfaitement déterminés. En effet, les mouvements du véhicule seront sujets à de nombreuses variations dues aux accélérations, freinages et cahots de la route que nous ne pouvons prévoir. Pour ces raisons, les méthodes de vision dynamique n'ont pas été retenues.

On trouvera néanmoins dans [Bouthemy 88] une synthèse des différentes méthodes d'analyse d'une séquence d'images.

3-5.2. La vision trinoculaire

3-5.2.1. Utilité d'une troisième caméra

Nous avons vu au chapitre 2 que deux vues différentes d'une même scène suffisent pour reconstruire la troisième dimension. Cependant certains auteurs préconisent l'utilisation d'une troisième vue. Introduite par Yachida [Yachida 86], cette idée a rapidement séduit de nombreux chercheurs [Ito 86], [Pietikainen 86], [Shen 89], voire également [Ayache 89].

En stéréovision classique, c'est à dire binoculaire, on recherche pour simplifier les traitements les primitives homologues le long des droites épipolaires conjuguées (Cf. § 3-4.2). Pour une primitive donnée de l'image gauche, on peut cependant avoir plusieurs candidats possibles dans l'image droite. Il faut pour lever les ambiguïtés d'appariement disposer d'algorithmes complexes et souvent très lents.

L'utilisation d'une troisième caméra permet une simplification importante de la mise en correspondance. Lorsqu'on apparie deux primitives des images 1 et 2, on teste immédiatement la validité de l'appariement en vérifiant sur l'image 3, la présence de la primitive correspondante.

3-5.2.2. Géométrie d'un dispositif trinoculaire

Dans un système trinoculaire, un point p de la scène se projette en trois points p_1 , p_2 et p_3 . Nous avons vu au § 3-2.2 que dans un système stéréoscopique binoculaire, à chaque couple de primitives pouvait être associée une paire de droites épipolaires. De telles paires peuvent être définies en vision trinoculaire. A partir des trois images du point p nous pouvons constituer 3 couples à appairier différents : (p_1, p_2) , (p_1, p_3) et (p_2, p_3) . Nous définissons alors 3 paires d'épipolaires conjuguées, soit 6 droites épipolaires.

La figure 3.7 illustre la géométrie d'un capteur stéréoscopique doté de 3 caméras. Géométriquement, lorsque p_1, p_2 et p_3 forment un appariement correct le point p_i ($i \in \{1,2,3\}$) se trouve nécessairement à l'intersection des épipolaires D_{ij} et D_{ik} (Cf. Figure 3.7)

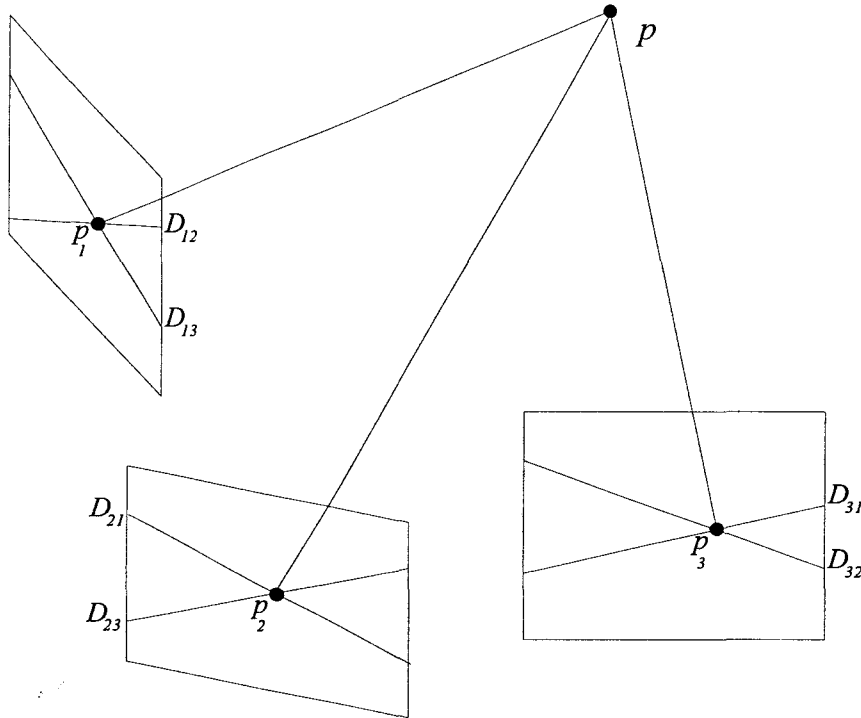


Figure 3.7 : Géométrie de la vision trinoculaire.

3-5.2.3. Algorithme de mise en correspondance

A partir de la géométrie du dispositif trinoculaire et des propriétés des droites épipolaires, on peut développer un algorithme simple de mise en correspondance basé sur la géométrie épipolaire. Cet algorithme comprend 3 grandes phases.

Pour chaque primitive p_1 de l'image 1 :

- **Sélection des couples** : Sélectionner dans l'image 2, l'ensemble des primitives $\{p_2\}$ se trouvant sur la droite épipolaire déterminée en fonction de p_1 .
- **Appariement** : Parmi tous couples possibles entre p_1 et l'ensemble $\{p_2\}$, déterminer celui qui présente le meilleur critère de ressemblance. Soit (p_1, p_2) le couple retenu.
- **Validation** : Si l'appariement entre p_1 et p_2 est correct, alors le point homologue p_3 de l'image 3 se trouve précisément à l'intersection des deux droites épipolaires déterminées en fonction de p_1 et de p_2 . Si le point p_3 est effectivement présent,

l'appariement est bon, dans le cas contraire le couple est rejeté. La phase d'appariement doit recommencer en éliminant la primitive p_2 de la liste $\{p_2\}$.

L'algorithme est relativement simple par rapport à ceux généralement utilisés en vision binoculaire (Cf. § 3-6.3). Grâce à la troisième image, l'algorithme peut vérifier immédiatement et simplement la validité des couples mis en correspondance. Ceci permet de diminuer le nombre d'erreur d'où une fiabilité accrue dans la reconstruction de la scène.

Il faut également noter qu'un tel système est symétrique c'est à dire que l'algorithme peut être appliqué de la façon suivante : appariement entre les images 2 et 3, vérification dans l'image 1 ou tout autre combinaison obtenue par permutation circulaire.

3-5.2.4. Conclusion

Nous avons vu brièvement que la vision trinoculaire apporte des améliorations par rapport à la vision binoculaire. Par rapport aux algorithmes de vision binoculaire, l'algorithme trinoculaire est :

- Plus **simple**. Sa mise en oeuvre est donc moins coûteuse en puissance de calcul et en temps d'exécution.
- Plus **fiable**. La seule contrainte utilisée est la contrainte épipolaire. Il n'est pas nécessaire, comme en vision binoculaire, de faire des hypothèses heuristiques qui ne sont pas toujours vérifiées, par exemple la contrainte d'ordre.
- Plus **précis**. L'utilisation de trois points de mesure au lieu de deux permet de réduire l'influence du bruit. Il en résulte une meilleure localisation des points 3D reconstruits.

Cependant, il faut noter quelques inconvénients mineurs :

- Le champ de vision commun à trois caméras est plus petit que celui défini par deux caméras. De ce fait, la taille de la scène pouvant être reconstruite est moins importante.
- La présence de trois caméras implique trois pré-traitements pour extraire les primitives de chaque image.

Mais dans le cadre de notre projet, un système trinoculaire présente d'autres inconvénients majeurs :

- L'intégration de 3 caméras sur un véhicule est très complexe. En effet, la

configuration du système trinoculaire doit d'une part permettre de disposer d'un champ commun de taille suffisante pour pouvoir effectuer efficacement les détections, et d'autre part rester fixe afin de conserver les paramètres de calibration. Comme un véhicule est un objet déformable, l'intégration d'un tel système demanderait beaucoup de recherche.

- Les trois pré-traitements nécessaires pour extraire les primitives augmenteraient les temps de calculs, ce qui serait pénalisant pour un système de détection d'obstacles. Il ne faut pas oublier que le matériel informatique embarqué doit être de taille réduite. On ne peut donc envisager une parallélisation des traitements telle que peuvent le faire les gros systèmes informatiques.

La vision trinoculaire n'a donc pas été retenue pour notre système embarqué de détection d'obstacles.

3-6. Mise en correspondance

3-6.1. Aspects fondamentaux

3-6.1.1. Formalisme de la fonction d'appariement

Pour définir mathématiquement la fonction d'appariement, on serait tenté de l'assimiler à une application bijective de l'ensemble G des primitives de l'image gauche, dans l'ensemble D des primitives de l'image droite. Cette application ne devrait faire correspondre qu'un seul élément de G à un seul élément de D et réciproquement. Or ce raisonnement est faux pour les raisons suivantes :

- **Le champ de vision commun** : Dans un système stéréoscopique la disposition des caméras est telle, qu'elles ne peuvent observer totalement la même partie de la scène. Il existe seulement un champ commun appelé secteur de stéréovision. Ainsi sur la figure 3.8a, l'objet "*rond*" se situe dans le secteur de stéréovision, représenté en gris, il est donc vu simultanément par les deux caméras. Par contre l'objet "*carré*" se trouve hors de la zone commune, il n'est vu que par la caméra gauche.
- **Les occlusions** : Les images étant prises sous des angles différents, il peut y avoir dans le secteur de stéréovision des objets ou parties d'objets qui ne sont pas vus simultanément par les deux caméras. Par exemple sur la figure 3.8b, l'objet "*rond*" est vu par la caméra gauche, mais non par celle de droite car il est occulté par l'objet "*carré*".

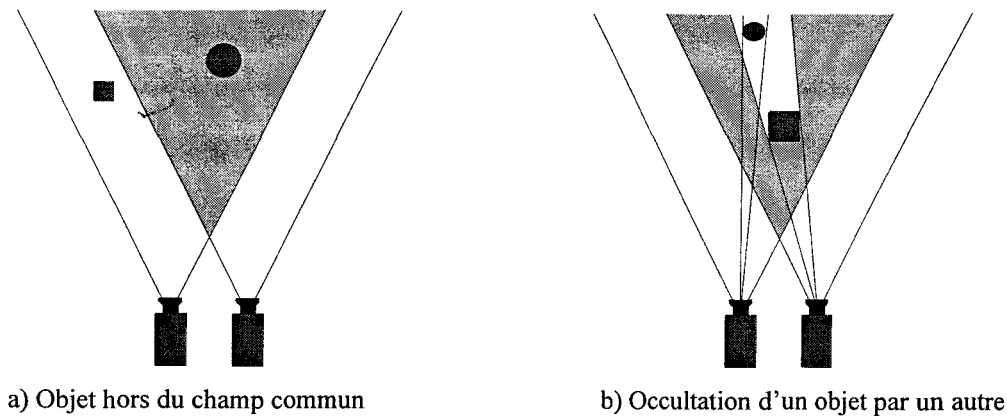


Figure 3.8 : Cas de primitives n'ayant pas de correspondant.

Dans des situations comme celles-ci, il y aura dans l'ensemble G des primitives qui n'auront pas de correspondants dans l'ensemble D et réciproquement.

Il n'y a donc pas de correspondance biunivoque entre les ensembles G et D , chaque primitive a **au plus un** correspondant. Ceci explique en partie la difficulté de la mise en correspondance, les algorithmes doivent éliminer les primitives n'ayant pas d'homologues dans l'autre image.

3-6.1.2. Combinatoire de la fonction d'appariement

Si les ensembles G et D sont respectivement composés de N_g et de N_d primitives, le nombre d'appariements possibles est alors de $N_g \times N_d$. Parmi ces couples, seules N paires seront retenues avec si l'on tient compte des remarques du paragraphe précédent, $N \leq \text{Min}(N_g, N_d)$. Il est donc fondamental de réduire le plus rapidement possible la combinatoire d'une part pour réduire le nombre d'erreurs possibles mais également pour diminuer les temps de traitements. La combinatoire importante justifie l'utilisation des contraintes, notamment la contrainte épipolaire (Cf. § 3-4.2) qui réduit le nombre d'appariements possibles de façon significative en vision matricielle, également en linéaire mais dans des proportions moindres.

3-6.1.3. Dilemme de la mise en correspondance

La mise en correspondance permet d'apparier les primitives de deux images stéréoscopiques. Les couples ainsi formés permettent, grâce aux calculs de triangulation, de localiser précisément les objets dans la scène réelle. Or, mise en correspondance et précision ne sont pas compatibles.

Nous avons vu au § 3-2.3 que l'un des paramètres importants dans la conception d'un système stéréoscopique était l'entre-axe des caméras, c'est à dire la distance entre les axes optiques. Ce

paramètre intervient directement dans les calculs de triangulation (Cf. § 2-2.2.3). Plus l'entre-axe est important, plus les points de vue sont différents ce qui donne une meilleure précision de localisation. On serait donc tenté d'écarter les caméras au maximum pour obtenir une reconstruction très précise de la scène. Cependant, plus les caméras sont éloignées l'une de l'autre, plus les images stéréoscopiques sont dissemblables : le champ de vision commun est réduit, les déformations géométriques sont plus importantes et les risques d'occlusion plus grands. La mise en correspondance entre les deux images est alors plus difficile à réaliser.

Le dilemme de la mise en correspondance consiste donc à établir un compromis entre précision et facilité de mise en correspondance ou plutôt entre précision et rapidité. Dans le cas d'images complexes, on peut toujours entreprendre une procédure d'appariement mais les algorithmes les plus performants pouvant les réaliser risquent d'être très lents.

3-6.2. Critère de ressemblance

3-6.2.1. Pourquoi définir un critère de ressemblance?

Nous verrons dans le paragraphe suivant que la plupart des algorithmes de mise en correspondance font appel à des critères de ressemblance. Lors de l'appariement de deux primitives, il faut pouvoir déterminer si elles correspondent bien aux images d'un même point de la scène. Cependant la *ressemblance* de deux primitives est une notion abstraite pour un algorithme informatique. Comment peut-il décider que la primitive *A* ressemble plus à la primitive *B* que la primitive *C*? Il faut donc développer un critère de ressemblance, également appelé *fonction de coût*, qui permet de la quantifier. Le critère choisi doit tendre vers une valeur limite pour une ressemblance parfaite, ce qui facilite ainsi la prise de décision. Dans la pratique, cette valeur n'est jamais atteinte car deux primitives correspondantes ne sont jamais totalement identiques. Le principe consiste alors pour un ensemble d'appariements possibles à sélectionner celui qui se rapproche le plus de la valeur limite du critère. En d'autres termes, le meilleur couple est celui qui maximise ou minimise, selon le cas, le critère choisi.

3-6.2.2. Nature des critères

L'hypothèse sous-jacente à la notion de ressemblance de deux primitives est que les attributs qui leur sont attachés doivent être similaires si ces deux primitives se correspondent. La nature des attributs et donc du critère exploité peut être de différents ordres : photométriques et géométriques.

3-6.2.2.1. Les critères photométriques

La ressemblance est dans ce cas fondée sur le niveau de gris des primitives.

Pour les primitives de type point, l'information unique de leur niveau de gris n'est pas suffisante car trop sensible au bruit. On prend donc en compte l'allure de la fonction image autour du point. Pour cela, il suffit de définir une fenêtre centrée sur chaque point que l'on tente d'apparier et de la comparer avec des fenêtres similaires dans la seconde image stéréoscopique.

Pour des primitives d'autres types, on peut envisager de définir une fenêtre qui englobe chaque segment ou région. Mais une telle méthode est difficile à mettre en oeuvre car ces primitives ont généralement des dimensions différentes dans les deux images. Les fenêtres choisies devant être de taille identique, elles peuvent alors englober des éléments parasites qui viendraient fausser l'estimation de la ressemblance.

3-6.2.2.2. Les critères géométriques

Ils s'appliquent généralement aux primitives de niveau supérieur comme les contours ou les régions. Pour ces primitives, il est possible de déterminer différents critères prenant en compte leurs caractéristiques géométriques comme la longueur pour les segments, la surface et le périmètre pour les régions, etc. Pour mesurer la compatibilité au sens géométrique, les critères admettent généralement des marges d'erreurs sur les mesures de paramètres.

3-6.2.3. Exemples de critères

Les critères géométriques étant souvent établis de façon empirique, nous donnons quelques exemples de critères photométriques utilisés. Ceux-ci sont surtout basés sur les techniques de corrélations et peuvent être définis sous différentes formes.

3-6.2.3.1. Exemples de critère de corrélation

Un moyen de mesurer la similarité de deux régions carrées centrées par exemple sur des points de contours est le coefficient de corrélation défini par :

$$C(g, d) = \sum_{i=-M}^{i=M} \sum_{j=-N}^{j=N} [I(x-i, y-j) - I'(x'-i, y'-j)]^2$$

où g est un point d'une image de coordonnées (x, y) et d un point de l'autre image de coordonnées (x', y') . $I(x, y)$ est la valeur du niveau de gris au pixel de coordonnées x et y . La mesure est effectuée sur deux fenêtres de taille $2M \times 2N$ centrées sur ces points.

La corrélation est maximale pour une ressemblance parfaite, c'est à dire quand $C(g, d)$ est nulle. Lorsque $C(g, d)$ prend des valeurs strictement positives, les régions ne sont pas corrélées.

On peut également calculer la corrélation de deux régions centrées aux points g et d grâce à la mesure suivante :

$$C(g, d) = \sum_{i=-M}^{i=M} \sum_{j=-N}^{j=N} [I(x-i, y-j) \cdot I'(x'-i, y'-j)]$$

L'interprétation est ici différente, la ressemblance est parfaite lorsque la valeur de $C(g, d)$ est maximale.

Nous venons de présenter deux exemples de critères basés sur la corrélation, mais ce ne sont pas les seuls, il en existe d'autres dans la littérature.

3-6.2.3.2. "Pseudo-corrélation"

Une des limitations de la corrélation réside dans la nécessité de travailler sur des fenêtres de taille identique. Or dans le cas d'images stéréoscopiques, les effets de perspectives conduisent deux primitives homologues à être de taille différente. Pour remédier à ce problème Ohta [Ohta 85] propose une "pseudo-corrélation" qui prend en compte ces différences en utilisant un critère qui s'applique sur des signaux mono-dimensionnels.

Soient g_1, \dots, g_k (respectivement d_1, \dots, d_l) les niveaux de gris des pixels dans la fenêtre de largeur k (resp. l) de l'image gauche (respectivement droite). Ohta définit la moyenne m et la variance σ^2 de la façon suivante :

$$m = \frac{1}{2} \left(\frac{1}{k} \sum_{n=1}^k g_n + \frac{1}{l} \sum_{m=1}^l d_m \right)$$

$$\sigma^2 = \frac{1}{2} \left(\frac{1}{k} \sum_{n=1}^k (g_n - m)^2 + \frac{1}{l} \sum_{m=1}^l (d_m - m)^2 \right)$$

De cette manière la contribution apportée par les deux fenêtres différentes est la même. Le coût s'exprime sous la forme :

$$C = \sigma^2 \sqrt{k^2 + l^2}$$

Le meilleur appariement est celui qui minimise la fonction C , c'est à dire celui qui correspond à la variance la plus faible.

3-6.2.4. Conclusion

Les critères de ressemblance fournissent une mesure de la qualité des appariements. Ce sont des critères *locaux* qui doivent être optimisés sur l'ensemble des appariements retenus. Les critères géométriques sont souvent insuffisants car ils laissent subsister des ambiguïtés. Aucune décision fiable ne peut être prise sur la base de tels critères car des appariements contradictoires peuvent présenter des indices de corrélation voisins. Il importe donc de faire appel à des contraintes plus globales qui permettent de lever les ambiguïtés.

3-6.3. Algorithmes de mise en correspondance

Il n'existe pas actuellement de méthode de mise en correspondance assez générale pour qu'elle puisse s'appliquer à une paire d'images stéréoscopiques indépendamment du contenu de ces images. Les algorithmes publiés dans la littérature utilisent différents principes que nous allons décrire ci-dessous.

3-6.3.1. Méthodes basées sur la corrélation

Les algorithmes fondés sur la corrélation utilisent principalement les attributs photométriques des images, c'est à dire les niveaux de gris pour juger la qualité des appariements. Ils sont simples et rapides à mettre en oeuvre. Ils consistent à considérer une fenêtre autour de la primitive que l'on veut apparier, puis à chercher dans l'autre image la primitive dont la fenêtre est la mieux corrélée. Afin de limiter le nombre de candidats, ces méthodes utilisent notamment les contraintes épipolaire, d'ordre et d'unicité. L'identité de la taille des fenêtres définies sur les deux images, limite cette approche à l'appariement de primitives de type point. Moravec [Moravec 77] a utilisé cette technique avec ses points d'intérêts (Cf. § 3-3.2.1.1), Castan [Castan 84] sur des points définis par les passages à zéro du laplacien.

Avec la mise en correspondance par corrélation, la taille de la fenêtre doit être suffisamment large pour inclure une sous-image représentative du voisinage de la primitive. La plupart des algorithmes existants fixent la taille de la fenêtre de façon empirique pour chaque application. Kanade dans [Kanade 94] propose, pour ajuster la taille de la fenêtre, un algorithme de mise en correspondance avec une fenêtre adaptative.

Les méthodes basées sur la corrélation furent les premières utilisées notamment sur les robots mobiles [Moravec 77] [Nevatia 76] en raison de leur simplicité. Cependant les méthodes corrélatives ont deux limites importantes : leur sensibilité aux déformations de perspectives et l'impossibilité à détecter les occlusions. Les effets de perspective font qu'une zone de taille

$x \times y$ sur l'image gauche aura sur l'image droite une taille $x' \times y'$ avec a priori $x \neq x'$ et $y \neq y'$. Pour être exacte, la corrélation devrait tenir compte de ces différences. Lors d'une occlusion, un point n'a pas de correspondant dans l'autre image. Avec une méthode corrélatrice, on trouvera toujours un "correspondant" qui présente la meilleure corrélation, même si elle est faible.

3-6.3.2. Méthodes basées sur la programmation dynamique

La programmation dynamique est une méthode d'optimisation basée sur le *principe de l'optimalité* [Bellmann 57], [Pontriagin 62], [Pontriagin 90] dont l'idée principale est que *toute partie d'un chemin optimal est elle-même optimale*.

Le problème de la mise en correspondance consiste donc à rechercher parmi tous les appariements possibles, un chemin qui sera considéré comme optimal s'il relie les appariements corrects entre eux. En pratique, on matérialise le problème en effectuant la recherche dans une matrice (ou un graphe) à deux dimensions. Chaque dimension est définie par les primitives de chaque image, ordonnées selon leur abscisse croissante et placées horizontalement de gauche à droite pour celles de gauche, verticalement de haut en bas pour celles de droite. Les éléments de la matrice (les sommets du graphe) représentent alors l'ensemble des appariements possibles auxquels on associe un coût, déterminé par une mesure de ressemblance, qui juge la qualité de la mise en correspondance. L'algorithme doit alors trouver le meilleur chemin qui permette d'aller d'une extrémité (supérieure gauche) à l'autre (inférieure droite) de la matrice (du graphe) en minimisant une fonction de coût. Le coût du chemin reliant les éléments de la matrice est défini en fonction du coût de chaque élément qui le constitue. Afin de trouver le chemin optimal, deux contraintes sont imposées par le principe de base de la programmation dynamique :

- L'ordre des primitives homologues doit être le même dans les deux images (contrainte d'ordre).
- Chaque primitive n'intervient qu'une seule fois (contrainte d'unicité).

Cette recherche ne pouvant être réalisée avec la totalité des primitives des images gauches et droites, elle utilise généralement la contrainte épipolaire et s'applique à l'ensemble des primitives se trouvant le long des épipolaires conjuguées.

Ce type de méthode permet également de traiter les cas d'occlusions. Il est en effet possible qu'une primitive d'une image n'ait pas de correspondante dans l'autre image. Dans ce cas le

coût associé à un couple contenant une telle primitive est très faible voire nul. Dans de telles situations, le chemin optimal est constitué de transitions horizontales et verticales entre les éléments de la matrice alors que les transitions entre deux couples appariés sont diagonales.

La plupart des travaux utilisant la programmation dynamique choisissent les points de contours comme primitives. Les plus connus sont [Baker 81] et [Ohta 85]. Récemment [Lee 94b] a cependant proposé de mettre en correspondance des segments de droite.

Les résultats obtenus avec une telle méthode sont généralement bons mais au prix d'un grand nombre de calculs qui rendent ces algorithmes très lents de plusieurs minutes à quelques dizaines de minutes suivant la complexité des images et la puissance de calcul disponible. Pour remédier à ce problème, Benschair [Benschair 92] a proposé un algorithme rapide de mise en correspondance ignorant la géométrie épipolaire.

Il faut également noter que l'utilisation impérative de la contrainte d'ordre, liée au principe de l'optimalité, limite son emploi à certaines catégories d'images comme les images aériennes par exemple où l'on est sûr que l'ordre des primitives est respecté.

3-6.3.3. Méthodes basées sur la relaxation

Les méthodes de relaxation s'appuient sur un étiquetage probabiliste itératif de l'ensemble des données à traiter. Chaque étiquette correspond en fait à l'une des classes à laquelle on désire assigner chaque élément étudié. On émet alors implicitement l'hypothèse que l'étiquetage d'un point est d'autant plus satisfaisant et fiable qu'il est cohérent avec l'étiquetage de ses voisins. Pour un aperçu sur les techniques de relaxation, leurs propriétés, on pourra consulter [Hancock 90] ou [Davis 81].

Dans le cas de la mise en correspondance, la relaxation s'applique de la façon suivante. Initialement, on réalise tous les appariements respectant certaines contraintes comme la contrainte épipolaire par exemple. Ensuite, on attribue à chacun de ces appariements, un poids ou indice de confiance qui permet de juger la qualité de l'appariement. Pendant la phase itérative, les appariements sont comparés à leurs voisins ce qui permet de faire évoluer les indices de confiances attribués et ainsi de réduire les ambiguïtés. A la fin de chaque itération, une mise à jour des appariements est effectuée en fonction des nouveaux indices. Le processus itératif s'arrête quand les indices n'évoluent plus.

De nombreux travaux ont proposé la relaxation comme méthode de mise en correspondance. Marr et Poggio [Marr 76] furent probablement les précurseurs dans l'utilisation de la relaxation pour la mise en correspondance. Leur algorithme se rapproche d'un processus neuronal puisque les indices évoluent en fonction d'un mécanisme inhibiteur si les appariements sont conflictuels et d'un mécanisme excitateur lorsque les appariements présentent des disparités voisines. Medioni [Medioni 85] et Long [Long 86] appariant des segments, provenant de l'approximation polygonale des contours, en minimisant un critère de variation de la disparité qui tient compte des appariements voisins. Pollard [Pollard 85] propose de mettre en correspondance des points de contour par formulation de la contrainte de continuité. Celle-ci fait intervenir un gradient de disparité limite entre deux couples de points appariés. Enfin, récemment Nasrabadi [Nasrabadi 92a] propose d'utiliser la relaxation pour mettre en correspondance des segments incurvés. Il détermine le centre de courbures de chaque segment. Il limite ainsi le nombre d'appariements possibles en appliquant la contrainte épipolaire aux centres de courbures. Les ambiguïtés d'appariement sont résolues ensuite par un algorithme de relaxation.

3-6.3.4. Méthodes basées sur les réseaux de neurones

Le problème de la mise en correspondance peut également être résolu par une approche neuronale.

Un réseau de neurones est un graphe *orienté et pondéré*. Les noeuds de ce graphe sont des automates simples appelés *neurones formels* et dotés d'un état interne, *l'activation*, par lequel ils influencent les autres neurones du réseau. Cette activité se propage dans le graphe le long d'arcs pondérés appelés *liens synaptiques*. Par extension, on appelle *poids synaptique* la pondération d'un lien synaptique. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ses pairs est appelée règle ou fonction d'activation. Le réseau est structuré en couches de neurones. Il comprend :

- Une *couche d'entrée* constituée d'un ensemble de neurones qui perçoit les données du problème.
- Une *couche de sortie* dont l'activation des neurones sera interprétée comme la réponse au problème.
- Une ou plusieurs *couches cachées*, voire aucune dans certains cas, composées d'un ensemble de *neurones cachés*.

On appelle neurones cachés ceux qui ne sont ni à l'entrée, ni à la sortie du réseau. Leur activation est donc réalisée par l'intermédiaire d'autres neurones. Pour une approche complète sur les réseaux de neurones, on pourra consulter [Jodouin 94].

Dans le cas de l'appariement, le problème est ramené à un problème d'optimisation. Une fonction d'énergie est construite pour représenter toutes les contraintes du problème. La mise en correspondance consiste alors à minimiser cette fonction d'énergie grâce à un réseau de neurones. En fait, les appariements corrects correspondent à des minimums locaux de la fonction d'énergie.

Nous pouvons citer les travaux de Lee [Lee 94a] ou de Nasrabadi [Nasrabadi 92b] qui utilisent des réseaux bidimensionnels de $N \times M$ neurones où N (resp. M) correspond aux nombres de points d'intérêts dans l'image gauche (resp. droite).

Dans le cadre de ce projet, Y. Ruichek [Ruichek 95a] et [Ruichek 95b] a utilisé un réseau de Hopfield pour apparier les images issues du stéréoscope linéaire. Les résultats obtenus sont encourageants mais des améliorations doivent encore être apportées pour envisager une implantation en temps-réel dans le véhicule expérimental.

3-6.3.5. Méthodes hiérarchiques ou pyramidales

Ces méthodes consistent à établir pour chaque image à mettre en correspondance une pyramide (ou hiérarchie). Chaque pyramide contient l'image représentée à différents niveaux de résolution : du niveau le plus "fin", correspondant à la résolution du capteur, au plus "grossier", la basse résolution. Pour construire simplement une pyramide il suffit de réduire une image en moyennant la valeur des pixels dans un carré $N \times N$. Ainsi pour $N = 2$, un carré 2×2 d'une image est remplacé par un seul pixel dans l'image de résolution inférieure, et ainsi de suite. Ainsi pour des images 512×512 , on construit les images 256×256 , 128×128 , etc, jusqu'à la résolution désirée par exemple 16×16 .

Après avoir extrait dans chaque image, les primitives intéressantes, généralement de type point, la mise en correspondance est effectuée au niveau le plus grossier. La taille des images étant réduite, l'application de la contrainte épipolaire permet de déterminer rapidement les points homologues à l'aide d'un critère de ressemblance qui peut être un indice de corrélation. L'étape suivante consiste à parcourir la pyramide de la basse résolution à celle du capteur, les résultats obtenus à un niveau servant à initialiser le niveau suivant. En reprenant notre

exemple, un point apparié dans l'image 16 x 16, correspond à une région de taille 2 x 2, dans l'image 32 x 32. L'intersection de cette région de taille 2 x 2 avec la droite épipolaire constitue la zone d'initialisation pour la recherche au niveau suivant. Cette stratégie est poursuivie pour chaque point apparié jusqu'au niveau de résolution le plus fin.

Cette méthode s'applique très bien aux images aériennes dont la résolution parfois importante (2048 x 2048) nécessiterait plusieurs dizaines de minutes de calculs avec d'autres méthodes comme la programmation dynamique.

Des exemples de méthodes pyramidales sont donnés dans [Heraud 93] et [Aubert 89]. Moravec [Moravec 81] avait également proposé une telle méthode pour aider un robot mobile à percevoir son environnement. Dans le cadre du programme Prometheus, un algorithme de mise en correspondance pyramidale a été développé [Meygret 90]. Celui-ci est fondé sur l'appariement de "*points de chaîne de contour*".

3-6.3.6. Méthodes de description structurelle (isomorphisme de graphe)

La mise en correspondance peut être vue comme un problème d'isomorphisme de graphe. Les images sont décrites sous forme d'un graphe relationnel où les noeuds correspondent aux primitives et les arcs aux relations entre ces primitives. Dans le cas où les primitives sont des segments de droite, les différentes relations entre ces segments sont : "*à gauche de*", "*à droite de*", "*colinéaire avec*", "*à une jonction avec*". La mise en correspondance consiste donc à chercher le meilleur appariement entre les noeuds et les arcs de deux graphes. En théorie, les deux graphes devraient être identiques, dans la pratique, ils sont différents. Les occlusions font que certaines primitives ainsi que les relations qui leur sont associées ne sont pas décrites dans l'une des images. Les effets de perspective conduisent à des relations différentes pour les primitives correspondant aux mêmes parties de la scène. Par exemple, des segments peuvent être colinéaires dans une image et se couper dans l'autre. Dans certains cas, la relation d'ordre n'étant pas respectée, elle entraîne des relations différentes entre deux primitives. Le problème d'appariement des deux graphes n'est donc pas simplement un problème d'isomorphisme de graphe mais plutôt un problème d'isomorphisme de sous-graphes maximal. La difficulté est donc de rechercher tous les isomorphismes entre sous-graphes d'un graphe et sous-graphes de l'autre graphe et de sélectionner ensuite l'isomorphisme qui met en correspondance le plus grand nombre de noeuds.

Cette approche a été proposée par Skordas [Skordas 88] lors de sa thèse, afin d'apparier des segments. Les contraintes épipolaire et d'orientation sont utilisées pour déterminer un certain nombre d'appariements potentiels. Ces appariements constituent les noeuds d'un "*graphe d'association*". On construit ensuite un arc entre deux noeuds si les appariements sont compatibles au sens de la contrainte d'ordre, d'unicité, figurale, etc. La meilleure correspondance entre les graphes des deux images est obtenue pour le graphe d'association présentant le plus grand nombre de noeuds mutuellement compatibles. Un ensemble de noeuds mutuellement compatibles forme une *clique*. Une clique est dite maximale si elle ne peut être incluse dans une autre clique. Une clique maximale correspond donc à un isomorphisme de sous-graphe et, la plus grande clique maximale correspond au plus grand nombre d'appariements possibles. On pourra également consulter [Heraud 89] pour plus de détails.

3-6.3.7. Méthodes de prédiction et vérification d'hypothèses

Les algorithmes basés sur ces méthodes procèdent en trois étapes indépendantes :

1. **Prédiction d'hypothèses** : Durant cette phase, on établit, à partir des primitives extraites de chaque image, un certain nombre d'appariements hypothétiques en fonction de critères de ressemblance géométriques ou photométriques.
2. **Propagation d'hypothèses** : On explore ensuite indépendamment chacune des hypothèses en la propageant aux voisins des primitives appariées. On utilise pour cela des contraintes de compatibilité entre appariements (Contrainte de continuité de la disparité, d'unicité).
3. **Vérification d'hypothèses** : On compare les appariements obtenus après la phase de propagation. Les hypothèses retenues sont celles ayant fourni le plus de couples de primitives compatibles dans la phase de propagation.

Parmi les travaux réalisés avec cette méthode, nous pouvons citer ceux de Robert [Robert 86] et de Ayache [Ayache 85] qui appariant des segments et dont le domaine d'application est la robotique mobile et plus récemment, les travaux de Krotkov [Krotkov 90] également sur des segments. Ce dernier travaille sur un système stéréoscopique actif, dont on peut commander de nombreux paramètres (mise au point, orientation des caméras, zoom,...), ce qui ne simplifie pas la mise en correspondance puisque les caractéristiques des images peuvent changer. D'une image à l'autre, il faut tenir compte des modifications des paramètres de calibration pour

déterminer la nouvelle géométrie épipolaire, mais également de l'aspect des primitives qui peut être fortement modifié. Wrobel [Wrobel 88] utilise cette méthode pour appairer les régions. Cependant celui-ci l'associe néanmoins à une technique de programmation dynamique qui guide la mise en correspondance des régions en appariant leurs contours.

3-6.3.8. Méthodes d'appariement sous contraintes

Il existe beaucoup d'autres algorithmes de mise en correspondance qui n'utilisent pas les principes évoqués ci-dessus. L'appariement est effectué par application de différentes contraintes. Le procédé est généralement le suivant. On établit un certain nombre d'appariements vérifiant une contrainte donnée comme la contrainte épipolaire par exemple. Ensuite on réduit l'ensemble des possibilités en appliquant successivement plusieurs contraintes jusqu'à l'obtention du meilleur appariement. Ces méthodes supposent bien entendu l'existence d'un nombre suffisant de contraintes pour aboutir à une solution.

Nous pouvons citer comme exemple [Deriche 90]. Les primitives appariées sont des points de courbures. L'auteur utilise la contrainte épipolaire pour définir des appariements potentiels. Deriche utilise ensuite pour valider ou infirmer ces appariements un certain nombre de contraintes dont notamment la contrainte de profondeur (Deriche suppose connues les distances minimales et maximales des objets dans la scène), de courbure, de direction du gradient et de continuité figurale.

Zagrouba [Zagrouba 92] propose d'appairer les régions en établissant un certain nombre de contraintes géométriques fondées sur des notions de compacité, moment d'inertie et en utilisant le rectangle circonscrit englobant la région compatible. Il ajoute également des critères de propagation des appariements pour les régions adjacentes.

3-6.4. Conclusions sur les méthodes de mise en correspondance

Nous venons de dresser une liste des méthodes couramment employées pour traiter le problème de la mise en correspondance. Cette liste n'est cependant pas exhaustive, il existe d'autres approches originales qui n'entrent dans aucune catégorie. Nous pouvons citer par exemple [Adjouadi 94] qui propose une technique de mise en correspondance exploitant la transformation de Walsh.

A quelques exceptions près, toutes les méthodes présentées ont en commun l'utilisation de contraintes, c'est à dire d'hypothèses sur la scène, et d'un critère de ressemblance parfois appelé fonction de coût qui juge la qualité des appariements. Ces méthodes n'étant pas assez

générales pour traiter tous les types d'images, il arrive également que deux techniques leur soient associées. Par exemple, Wrobel [Wrobel 88] associe la programmation dynamique et la méthode de prédiction/vérification. Il faut également noter que bon nombre de méthodes font appel aux techniques de corrélation qui constituent le meilleur moyen pour comparer deux grandeurs.

3-7. Travaux existants en stéréovision linéaire

L'utilisation de caméras linéaires dans un système stéréoscopique n'est pas très fréquente. La perte d'information due à la nature monodimensionnelle du signal en est certainement responsable. Il existe cependant à notre connaissance deux précédents.

3-7.1. Système de guidage d'un robot mobile proposé par Inigo

Inigo [Inigo 85], [Inigo 87] propose d'utiliser un système stéréoscopique basé sur deux caméras linéaires pour guider un robot mobile dont la géométrie est proche de celle décrite au chapitre 2. Pour détecter d'éventuels obstacles, les caméras fixent le sol quelques mètres à l'avant du robot.

3-7.1.1. Guidage du robot

La fonction de guidage est réalisée à l'aide d'une seule caméra chargée de détecter une ligne noire peinte sur le sol. Pour analyser la scène, on suppose que la ligne est représentée sur l'image par un créneau de largeur connue l . La détection s'effectue à l'aide de deux filtres de type médian f_1 et f_2 de taille respective l_1 et l_2 avec $l_1 < l < l_2$. Les deux images obtenues après l'application des filtres de type médian servent à déterminer une nouvelle image. Celle-ci est calculée par soustraction pixel à pixel des deux images filtrées et permet de mettre en évidence la ligne de guidage.

3-7.1.2. Détection d'obstacle

Inigo détecte d'abord les discontinuités dans les deux images. L'information donnée par les niveaux de gris n'étant pas suffisante, il calcule deux nouvelles images où le niveau de gris de chaque pixel est remplacé par la variance locale. Aux quatre images (niveaux de gris et variance locale, gauche et droite), on applique un seuillage (avec plusieurs seuils déterminés à partir d'histogrammes). Les seuils sont ajustés automatiquement aux creux des histogrammes des images. Grâce à un seuillage multiple, on extrait alors les points qui seront candidats à un algorithme d'appariement par corrélation.

Pour la mise en correspondance, les points sont représentés par un vecteur à deux dimensions dont les éléments sont le niveau de gris du point et sa variance locale. Pour chaque point caractéristique de l'image de gauche, on détermine un modèle à l'aide d'une fenêtre centrée sur ce point. Ce modèle prend en compte les niveaux de gris mais également les variances sur l'ensemble de la fenêtre. Chaque modèle est ensuite appliqué à chaque point de l'image de droite. On calcule alors un coefficient de corrélation prenant en compte la variance et le niveau de gris. Pour chaque modèle, parmi l'ensemble des possibilités, le meilleur appariement est celui qui donne le coefficient de corrélation le plus faible. Une carte des profondeurs est ensuite déterminée par triangulation, permettant de détecter les obstacles présents devant le robot.

3-7.1.3. Expérimentation et résultats

Le système stéréoscopique est constitué de deux caméras de 256 pixels chacune. L'ensemble des traitements effectués sur un processeur 68000 est réalisé en 250 ms soit 4 couples d'images stéréoscopiques pour des scènes présentant des obstacles. La cadence de 10 images par seconde est atteinte lorsque rien n'entrave les déplacements du robot. Les résultats obtenus sur plusieurs scènes permettent d'envisager un guidage en temps-réel du robot mobile.

3-7.1.4. Conclusion sur la méthode proposée par Inigo

La méthode proposée par Inigo pour le guidage du robot est simple mais il ne tire pas avantage du capteur stéréoscopique puisqu'elle n'utilise qu'une seule caméra. Son inconvénient est de devoir connaître a priori la taille de la ligne de guidage dans l'image.

Cette approche détecte facilement les obstacles. En effet, les caméras étant dirigées vers le sol, la variance des niveaux de gris est supposée constante. L'apparition d'un objet aura pour conséquence une modification locale de la variance. Une segmentation sur un tel critère permet donc une localisation aisée des obstacles. Si un objet détecté se trouve à une distance inférieure à la portée maximale du banc stéréoscopique alors c'est un obstacle réel, sinon c'est un objet se trouvant sur le sol (ombre, tache).

La disposition inclinée du système stéréoscopique limite l'horizon d'observation mais permet, en revanche, d'avoir un fond uniforme qui simplifie les images. Le nombre de contours et d'objets sont donc limités ce qui justifie la méthode de mise en correspondance choisie. Cette méthode d'appariement, par un simple calcul de nombreux coefficients de corrélation, ne

serait pas valable dans le cas d'images plus complexes telles que celles obtenues avec des caméras dont le champ serait horizontal. Dans ce cas, beaucoup d'objets, de variances et niveaux de gris quelconques pourraient alors être observés. La procédure de mise en correspondance utilisée serait alors confrontée à de nouveaux problèmes qu'elle serait incapable de résoudre. Il s'agit principalement des occlusions et des ambiguïtés d'appariements dues aux ressemblances de primitives.

3-7.2. Système stéréoscopique de Colle destiné à la robotique mobile.

Dans sa thèse, Colle [Colle 90] a développé un système stéréoscopique semblable à celui d'Inigo. Les caméras sont fixées sur un support rigide afin de conserver un champ de vision commun. L'ensemble caméras et support est posé sur des rails permettant ainsi le déplacement du système stéréoscopique de façon à simuler les mouvements d'un robot.

3-7.2.1. Extraction de primitives stables

Les primitives retenues par Colle sont les points de contours. Mais il considère que l'appariement de ces points peut conduire à des détections instables du point de vue temporel. Il propose donc de prendre en compte une information supplémentaire au niveau monoculaire, afin d'assurer la stabilité des primitives avant de tenter les appariements. Il observe donc pour chaque caméra, l'évolution temporelle des points où le gradient présente des maxima locaux. Cette étude dynamique lui permet d'effectuer un filtrage temporel et d'extraire les contours stables, c'est à dire ceux présents sur plusieurs images successives. Il élimine ainsi les points de contours épars dus au bruit. Cette approche se ramène à une méthode de prédiction/décision. Lorsqu'un point est "connecté", son évolution dans le temps fait apparaître une courbe dont la forme dépend de la position du point objet et de son mouvement par rapport aux caméras. Ces courbes sont qualifiées de contours spatio-temporels.

3-7.2.2. Mise en correspondance

Puisque à chaque instant, l'auteur dispose des points de contours et de leur évolution au cours du temps, il utilise cette information dynamique pour exprimer une contrainte d'appariement. En supposant la scène constituée d'objets statiques, Colle impose aux paires de primitives de respecter une contrainte de *rigidité*, autrement dit, puisque la scène est fixe, les mouvements des points reconstruits dans la scène sont des translations identiques. La solution est obtenue lorsque toutes les paires appariées donnent des translations identiques. L'algorithme d'appariement utilise les contraintes de cohérence géométrique et d'unicité. Initialement, on

établit une solution constituée des appariements vérifiant la contrainte d'unicité. Dans une phase itérative, on modifie ces appariements afin de minimiser un critère global calculé à partir des translations obtenues. La procédure s'arrête quand les modifications n'entraînent plus de diminution du critère.

3-7.2.3. *Expérimentation et résultats*

Le système stéréoscopique est constitué de caméras de 1024 pixels et d'objectifs de 25 mm. L'ensemble des traitements est réalisé "hors-ligne" sur un PC-AT doté d'un processeur 286. Les axes optiques des caméras sont parallèles entre eux et horizontaux par rapport au sol contrairement au dispositif d'Inigo. L'étude a été faite sur des scènes simulées avec et sans bruit, puis sur des scènes d'intérieur réelles, constituées de nombreux objets présentant des arêtes verticales (principalement des cartons) en déplaçant le système stéréoscopique sur des rails. Les résultats obtenus sont satisfaisants sauf en présence de bruit.

3-7.2.4. *Conclusion sur l'approche de Colle*

L'approche de Colle est originale car elle associe la vision binoculaire et la vision dynamique, cette dernière étant généralement dédiée à la vision monoculaire. (Cf. § 3-5.1). Cette association se révèle efficace et adaptée aux contraintes de temps-réel grâce à l'utilisation de caméras linéaires qui réduisent la quantité d'information à traiter ce qui serait difficilement réalisable avec des images matricielles. L'un des aspects originaux de la méthode réside dans la recherche des primitives stables au cours du temps. Cette étude dynamique permet d'éliminer les points parasites dus au bruit, facilitant énormément par la suite la mise en correspondance. L'inconvénient de la méthode proposée par Colle est qu'elle est réservée aux scènes statiques. La présence d'objets mobiles irait à l'encontre de la contrainte de rigidité de la scène, car les mouvements des points le long des contours spatio-temporels ne correspondraient plus à des translations identiques.

3-7.3. *Approche réalisée par J.-L. Bruyelle*

Nous tenons à mentionner ici l'approche simple faite par J.L. Bruyelle [Bruyelle 94] à seule fin de valider son stéréoscope linéaire dont nous avons décrit le principe au chapitre 2.

3-7.3.1. *Traitements réalisés*

Les primitives utilisées sont également les points de contours. Ceux-ci sont extraits grâce au détecteur de Canny après lissage de l'image par un filtre médian afin d'éliminer le bruit.

Le principe de mise en correspondance est basé sur la corrélation. Il examine les niveaux de gris sur deux fenêtres de même largeur, centrées sur les points de contour qu'il tente d'apparier. Afin d'obtenir une estimation de la ressemblance, il utilise un indice de corrélation. Comme ce seul critère lui semble insuffisant, il met en jeu différentes contraintes : ordre, contrainte du signe du gradient (Cf. § 5-3.4). Pour chaque point de l'image de gauche, il cherche tous les appariements possibles, c'est à dire ceux qui respectent les différentes contraintes imposées et présentent un indice de corrélation supérieur à un seuil donné. Il autorise ainsi les appariements multiples, laissant le soin de la recherche du meilleur appariement à une autre procédure, notamment en imposant des contraintes spatio-temporelles.

3-7.3.2. Conclusion

La validation des travaux de J.-L. Bruyelle, notamment sur des scènes simples mais réalistes nous a poussés à améliorer les procédures d'analyse des images stéréoscopiques afin de reconstruire des environnements 3D plus complexes et surtout plus rapidement. Nous avons donc décidé de reprendre l'ensemble des traitements à leur point de départ et de développer notre propre méthode qui fait l'objet des chapitres suivants.

3-8. Conclusion

La perception du relief est une fonction complexe que l'être humain effectue sans même y songer. Lorsqu'il s'agit de reproduire cette fonction à l'aide d'une ou plusieurs caméras et d'un système informatique, un nombre considérable de problèmes surgissent alors. Une analyse des méthodes existantes nous a fait découvrir une approche commune à la plupart des auteurs. Basée sur le principe de la vision humaine, la vision binoculaire également appelée stéréovision constitue une approche très répandue mais qui soulève encore de nombreuses difficultés dans le cas de scènes réelles et complexes.

La conception d'un système stéréoscopique passe par une phase importante : la création du banc stéréoscopique. La phase indispensable du calibrage permet de déterminer les relations géométriques entre les caméras qui serviront au cours des traitements. De ces relations, dépend également la phase de triangulation qui permettra la reconstruction 3D. Mais la partie la plus difficile reste la mise en correspondance. Cette étape obligatoire a pour but d'apparier des éléments caractéristiques de chaque image, appelés primitives. Celles-ci sont les images

des parties les plus représentatives des objets présents dans la scène observée. La reconstruction de la scène sera d'autant plus aisée et efficace que ces primitives seront plus représentatives des objets. L'extraction des primitives constitue la première tâche d'un algorithme de stéréovision, elle fera donc l'objet de notre prochain chapitre.

Chapitre 4

EXTRACTION DES PRIMITIVES

Le choix des primitives est une phase importante dans un algorithme de stéréovision. La nature de ces primitives et les différents attributs qu'on peut leur attribuer jouent un grand rôle lors de la phase d'appariement mais également dans la reconstruction de la scène. Nous avons vu, au chapitre précédent, les différentes primitives utilisées en vision matricielle. Dans le cas d'images linéaires, celles-ci se présentent sous des formes différentes. Il est donc nécessaire de choisir les primitives les mieux adaptées à ce type d'image (Cf. § 4-1). La détection des primitives devant répondre à certains critères (Cf. § 4-2), nous étudierons les différents détecteurs existants (Cf. § 4-3) afin de déterminer celui qui est le mieux adapté à nos besoins. Nous en arriverons enfin à la phase essentielle : l'extraction des primitives (Cf. § 4-4). Nous terminerons ce chapitre en présentant quelques résultats (Cf. § 4-5) et les attributs que l'on peut associer aux primitives retenues (Cf. § 4-6)

4-1. Choix de primitives en vision linéaire

Si le choix des primitives est important et diversifié en vision matricielle, celui-ci se trouve rapidement restreint en vision linéaire en raison de la nature monodimensionnelle du signal.

4-1.1. Passage du matriciel au linéaire

La transposition des primitives adaptées aux images matricielles n'est pas évidente lorsque l'on passe aux images linéaires. Nous allons donc étudier les transformations créées par la perte d'une dimension.

4-1.1.1. Primitives de type point

Dans le cas d'une image ligne, la recherche de points particuliers peut être envisagée, mais se trouve limitée puisque les notions de points de courbure ou de coins n'ont plus aucune existence dans un espace à une dimension, comme le montre la figure 4.1.

Il existe cependant la possibilité d'utiliser des points analogues aux points de Moravec (Cf. § 3-3.2.1.1). Toutefois, la mesure d'intérêt, c'est à dire la variance des niveaux de gris ne serait évaluée que sur des portions de signaux monodimensionnels, c'est à dire sur des fenêtres centrées autour des points étudiés. Les points sélectionnés seraient ceux présentant la plus forte variance. Mais un problème apparaîtrait dans les zones fortement texturées de l'image, où le nombre de points de variance locale élevée est important. Afin de limiter le nombre de points dans chaque zone, un seuillage serait nécessaire. Une telle approche ne semble pas envisageable car trop arbitraire au niveau de la sélection des points.

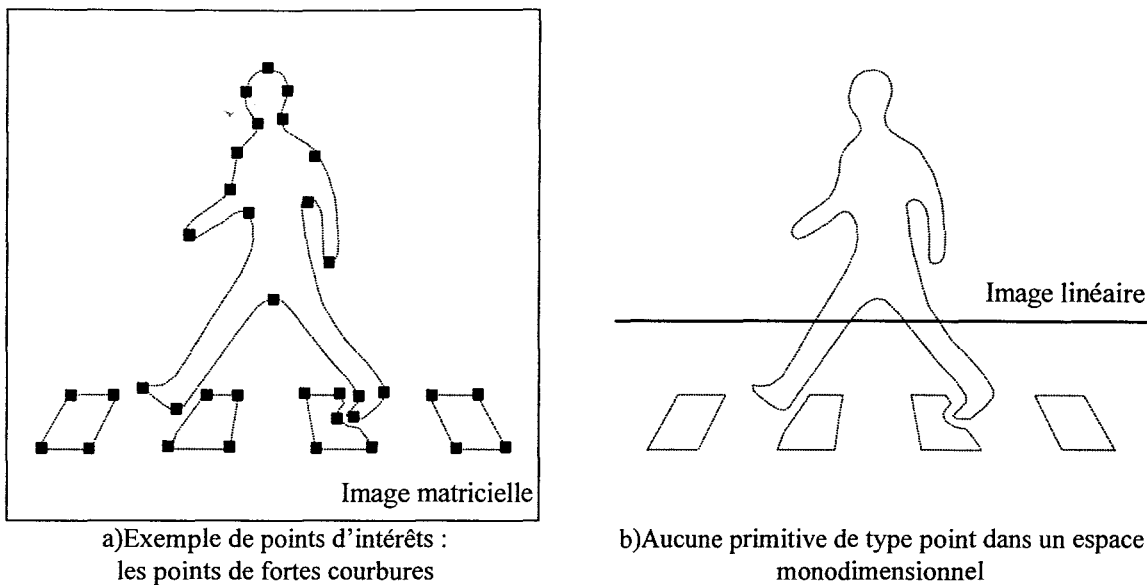


Figure 4.1 : Comparaison des primitives de type point.

4-1.1.2. Primitives de type région

Les régions, par essence à deux dimensions dans le cas d'images matricielles, se réduisent ici à de simples segments dans l'espace de travail constitué par l'image linéaire, comme le montre la figure 4.2.

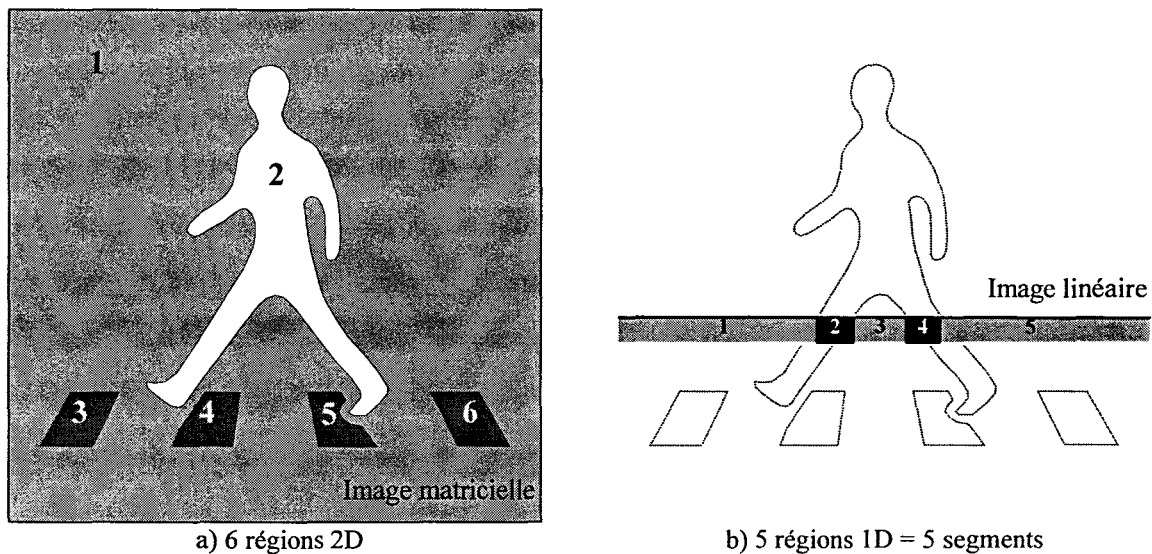


Figure 4.2 : Comparaison des primitives de type régions.

Les attributs de type photométrique peuvent être déterminés sur ces segments puisqu'il est toujours possible de calculer la valeur moyenne ou la variance du niveau de gris des pixels constituant chaque segment. Cependant ces attributs ne sont pas vraiment intéressants. L'intérêt majeur des primitives de type région provient de la richesse de leurs attributs, principalement de type géométrique, puisqu'ils décrivent aussi bien les dimensions que la

forme. Nous pouvons citer quelques exemples de tels attributs en mentionnant la surface, le périmètre, la compacité, les moments d'inertie, etc., mais cette liste est loin d'être complète. Dans le cas d'images linéaires, le seul paramètre géométrique pouvant être défini est la longueur (qui est également la surface) du segment qui forme la région. On constate donc que le nombre d'éléments pouvant décrire une primitive de type région en vision linéaire est restreint, ce qui ne facilite pas les appariements, surtout en cas d'ambiguïté.

D'autre part, les effets de perspective (Cf. § 3-3.2.3) peuvent entraîner l'apparition de deux régions correspondant au même objet physique, l'une sur l'image gauche, l'autre sur l'image droite, mais présentant deux longueurs différentes. Dans le cas d'images matricielles, cet inconvénient était gênant puisque deux régions pouvaient avoir des formes et dimensions différentes. Par contre, dans le cas d'images linéaires, cette limitation devient dramatique puisque le seul attribut géométrique n'est plus utilisable.

4-1.1.3. Primitives de type contour

La notion de contour présentée dans le cadre d'images matricielles (Cf. § 3-3.2.2) n'est vraiment utilisable que si l'objet dont on modélise le contour est intégralement présent (ou tout au moins en grande partie) dans l'image. La figure 4.3 montre que cette condition n'est pas réalisée dans le cas d'images linéaires.

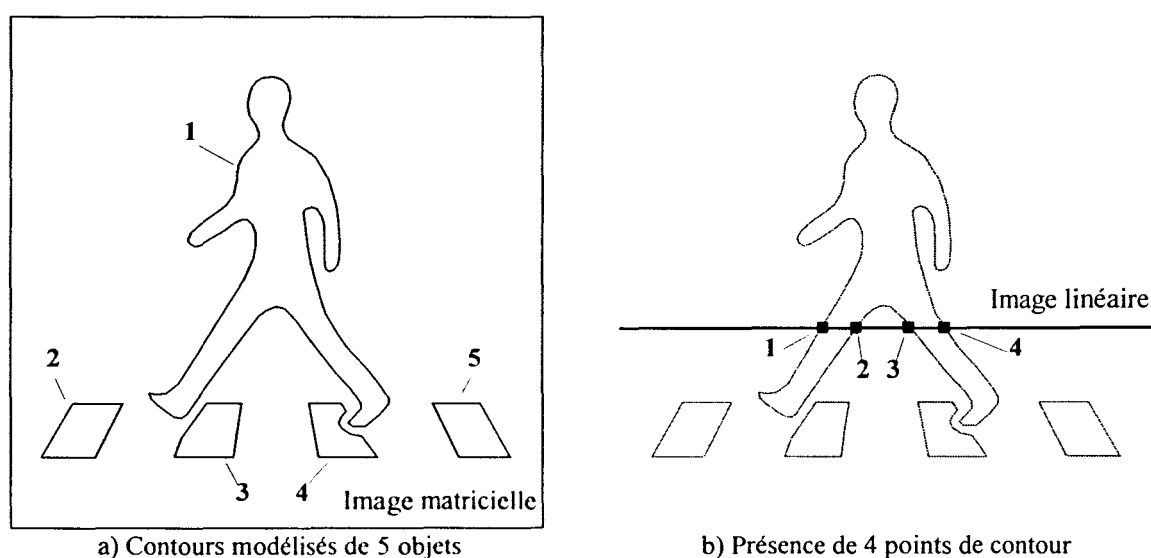


Figure 4.3 : Comparaison des primitives de type contour.

La perception d'un objet est réduite ici à une section plane de celui-ci. Le contour de l'objet, tel qu'il a été modélisé dans une image matricielle, n'a donc plus d'existence dans cet espace linéaire. La manifestation des contours dans une image linéaire provient donc de l'intersection

des contours de l'image matricielle avec une droite symbolisant l'image linéaire. Il en résulte un ensemble de points matérialisant les frontières de l'objet dans cette image linéaire. Ces points seront désormais appelés points de contour.

Contrairement aux images matricielles, il n'est pas trivial de passer à un niveau d'analyse supérieur. On pourrait envisager de regrouper les points de contour, associés à un même objet, en une seule primitive, celle-ci décrivant l'objet avec l'ensemble des points qui le constituent. Malheureusement, une analyse de connexité, comme celle utilisée en vision matricielle pour connecter ces points, n'est pas applicable en vision linéaire puisque ceux-ci ne sont pas voisins. De même, deux points de contour successifs dans une image linéaire n'appartiennent pas systématiquement au même objet.

4-1.1.4. Primitives de haut niveau : les contours actifs

Ce genre de primitives n'est pas transposable en vision linéaire, car les contours actifs n'ont pas d'existence possible dans les images lignes. La nature même du contour actif qui est une courbe fermée convergeant vers le contour bidimensionnel d'un objet empêche toute adaptation dans un monde monodimensionnel. Comment pourrait-on d'ailleurs représenter une courbe fermée dans un espace à une dimension?

4-1.2. Primitives possibles en vision linéaire

Il existe un grand nombre de primitives utilisables pour analyser les images matricielles, comme nous l'avons vu un peu plus haut (Cf. § 3-3.2). Cependant l'emploi d'images linéaires restreint considérablement les possibilités de choix. Seuls deux types de primitives ressortent de cette étude: les contours et les régions, mais avec une représentation différente pour chacun d'eux dans le cas d'images linéaires:

- Les contours deviennent des points caractéristiques que l'on appelle points de contour. La définition des points de contour en vision linéaire est différente de celle donnée en vision matricielle puisqu'un point de contour est considéré comme un point unique et non comme partie d'une courbe. De même les propriétés géométriques couramment attribuées au point de contour « matriciel », comme le vecteur direction du contour en ce point, et utilisées dans les techniques de mise en correspondance [Horaud 93], ne peuvent être définies en vision linéaire.
- Les régions sont, quant à elles, réduites à des segments de droites définis comme des portions de l'image linéaire où le niveau de gris est uniforme, d'où l'appellation de

régions homogènes. C'est le seul type de régions pouvant être défini dans le cas de l'analyse d'images linéaires. Cependant ces primitives possèdent beaucoup moins d'attributs que dans le cas d'images matricielles.

4-1.3. Dualité Régions-Contours

Si les points de contour et les régions homogènes sont des primitives de types différents, nous pouvons cependant montrer qu'elles sont dépendantes l'une de l'autre dans le cas d'images linéaires.

La région est définie comme une partie de l'image linéaire où le niveau de gris est uniforme. Deux régions seront distinctes s'il existe entre ces deux régions une différence de niveau de gris moyen. Le seul moyen pour les séparer sera donc de détecter une variation significative du niveau de gris dans l'image. Cette variation est plus communément appelée contour et la localisation de cette variation, point de contour. Toute région sera donc limitée par deux points de contour et chaque point de contour servira de frontière entre deux régions.

4-1.4. Conclusion sur le choix du type de primitive

Nous avons vu au cours de cette analyse que d'une part, deux types de primitive étaient utilisables en vision linéaire : les points de contour et les régions, et d'autre part, que ces deux types de primitive n'étaient pas indépendants. Nous pouvons donc conclure que, quelque soit le type de primitive utilisé, il nécessite la détection des points de contour présents dans l'image, c'est à dire des variations locales importantes de l'intensité de l'image. Dans les deux cas, les points de contour seront les éléments clefs de la procédure d'appariement. Nous pouvons de plus ajouter qu'il est logique de choisir les points de contour comme éléments caractéristiques, car ils correspondent aux frontières des différents objets présents dans l'image. Nous verrons, dans le chapitre 5, comment les points de contour seront utilisés.

4-2. Choix du détecteur de points de contour

Il existe aujourd'hui de nombreux algorithmes de détection de contours qui peuvent être aussi bien utilisés sur des images matricielles que sur des images linéaires. L'adaptation de ces algorithmes d'un espace à deux dimensions à un espace à une dimension ne pose généralement aucun problème. Il faut également préciser que toutes ces procédures de détection ne donnent pas les mêmes résultats. Certaines sont plus performantes, d'autres plus rapides. Devant tant d'outils disponibles, il est donc nécessaire de faire un choix, ou plutôt de

définir des critères de sélection permettant de réaliser ce choix.

4-2.1. Contraintes à respecter par le détecteur de points de contour

Le détecteur de points de contour le plus adapté doit se plier à certaines contraintes.

4-2.1.1. Détection de tous les points de contour

Traditionnellement, la notion de contour englobe souvent les points de discontinuité de la fonction image. Cependant, la discontinuité n'est pas toujours parfaitement nette. On peut généralement diviser les contours en trois types :

- ◆ Le contour parfait ou marche : Ce type de contour existe entre deux régions homogènes entre lesquelles la valeur du niveau de gris présente une forte variation sur très peu de points. Ce type de contour peut être dû au recouvrement d'un objet par un autre. Un point de contour de type marche correspond donc à un point de discontinuité dans la fonction niveau de gris. Son profil est représenté sur la figure 4.4(a). On inclut généralement dans cette catégorie les doubles marches dont il existe deux types : les crêtes et les escaliers (Cf. Figures 4.4 (c) et (d)).
- ◆ Le contour doux : Ce type de contour apparaît entre deux régions homogènes, quand la valeur du niveau de gris croît ou décroît régulièrement sur un intervalle correspondant à un nombre important de pixels (Cf. Figure 4.4(e)).
- ◆ Le contour bruité : Dans la pratique, les contours idéaux abrupts ou doux se trouvent généralement altérés par la présence de bruit, ce qui rend parfois leur détection difficile (Cf. Figure 4.4(b)).

On constate donc qu'il existe plusieurs types de points de contour. Le détecteur que nous choisirons doit donc être capable de tous les mettre en évidence. En effet, dans le cadre de notre application, nous ne pouvons nous permettre de négliger un obstacle dangereux, simplement parce que ses points de contour n'ont pas été détectés.

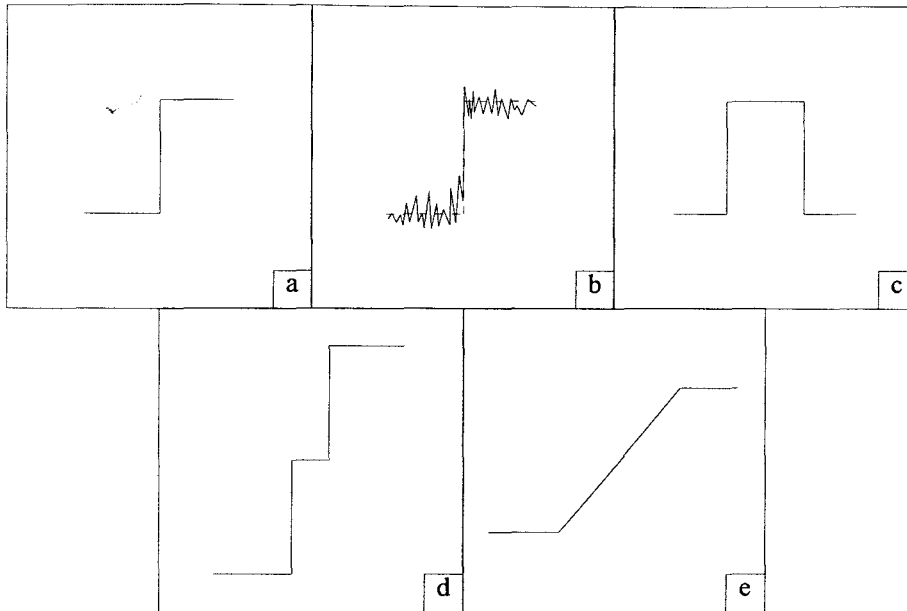


Figure 4.4 : Profil des différents types de contour.

4-2.1.2. Localisation précise de la position des points de contour

Au chapitre précédent, nous avons vu comment il était possible par simple triangulation de déterminer la distance et la position des objets grâce au stéréoscope linéaire (Cf. § 2-2.2.3). Appliquons les formules développées pour cette triangulation aux points de contour en introduisant une erreur de localisation.

Soit x_g (respectivement x_d) la position d'un point de contour dans l'image de gauche (resp. du point de contour correspondant dans l'image droite). Soit Δx_g (resp. Δx_d) l'erreur de localisation du point de contour dans l'image de gauche (resp. droite). Soit x'_g la position du contour mal localisé dans l'image de gauche (resp. x'_d pour l'image de droite) définie par :

$$x'_g = x_g + \Delta x_g \quad \text{Equation 4.1}$$

$$x'_d = x_d + \Delta x_d \quad \text{Equation 4.2}$$

Définissons alors la nouvelle disparité δ' :

$$\delta' = |x'_g - x'_d| \quad \text{Equation 4.3}$$

Nous verrons, au chapitre 5, que la contrainte épipolaire nous permet d'affirmer que x'_g (resp x_g) est toujours supérieur à x'_d (resp x_d) (Cf. § 5-3.3). L'équation 4.3 devient alors :

$$\delta' = x'_g - x'_d = \delta + \Delta\delta \quad \text{Equation 4.4}$$

où δ et $\Delta\delta$, sont respectivement la disparité et l'erreur commise sur cette disparité. Ces deux paramètres sont alors définis par :

$$\delta = x_g - x_d \quad \text{Equation 4.5}$$

$$\Delta\delta = \Delta x_g - \Delta x_d \quad \text{Equation 4.6}$$

En tenant compte des erreurs possibles, les équations donnant la distance et la position s'écrivent respectivement sous la forme :

$$D' = \frac{E \cdot f}{\delta'} \quad \text{Equation 4.7}$$

$$\begin{cases} X' = \frac{x'_g \cdot D'}{f} - \frac{E}{2} \\ X' = \frac{x'_d \cdot D'}{f} + \frac{E}{2} \end{cases} \quad \text{Equation 4.8}$$

Nouvelle expression de la distance D'

Si l'on exprime D' dans l'équation 4.7 en fonction de D , δ et de $\Delta\delta$, on obtient une nouvelle expression de D' :

$$D' = \frac{E \cdot f}{\delta + \Delta\delta} = \frac{E \cdot f}{\delta} \cdot \frac{1}{1 + \frac{\Delta\delta}{\delta}}$$

soit :

$$D' = D \cdot \frac{\delta}{\delta + \Delta\delta} \quad \text{Equation 4.9}$$

où :

- D est la distance qui serait obtenue sans erreur de localisation du contour (Cf. Equation 2.4).
- δ est la disparité qui serait obtenue sans erreur de localisation du contour.
- $\Delta\delta$ est l'erreur introduite dans la valeur de la disparité par une mauvaise localisation du contour.

Influence des erreurs de localisation

Soit $E_r(\Delta\delta)$ l'erreur relative commise sur le calcul de la distance, défini par :

$$E_r(\Delta\delta) = \frac{|D - D'|}{D} = \frac{\left| D - D \cdot \frac{\delta}{\delta + \Delta\delta} \right|}{D} = \frac{\left| 1 - \frac{\delta}{\delta + \Delta\delta} \right|}{1} = \frac{|\Delta\delta|}{|\delta + \Delta\delta|}$$

Pour éviter toute ambiguïté entre l'erreur relative E_r et l'erreur de localisation $\Delta\delta$, nous utiliserons le terme *écart* lorsque nous évoquerons l'erreur relative E_r .

La figure 4.5 représente l'évolution de l'écart E_r en fonction de la disparité δ pour différentes valeurs positives de $\Delta\delta$.

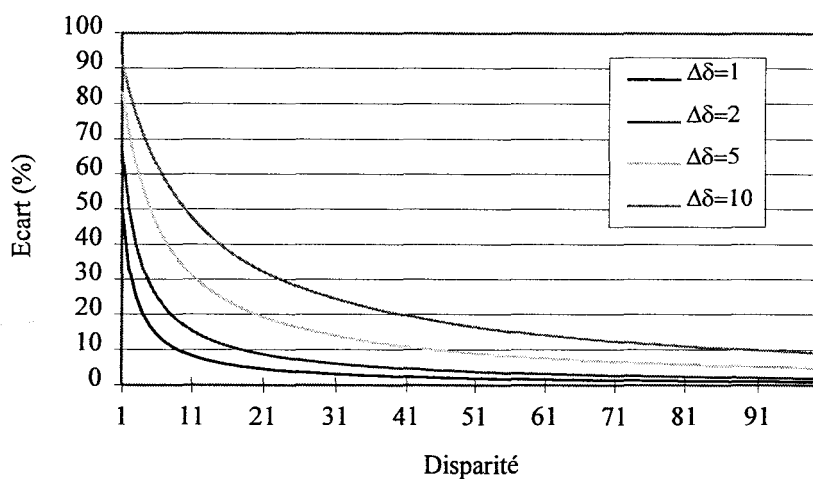


Figure 4.5 : Influence de l'erreur $\Delta\delta$ sur la mesure de distance ($\Delta\delta > 0$).

La figure 4.6 montre, quant à elle, la même évolution mais pour des valeurs négatives de $\Delta\delta$. Nous pouvons voir que lorsque l'erreur $\Delta\delta$ est proche de la disparité δ , l'écart peut être très important. Cependant, il s'agit de cas assez rares dans la pratique. Pour cette raison la figure 4.7 donne un agrandissement de la figure 4.6 pour des valeurs de l'écart inférieures à 100%.

De manière concrète, plus un objet est éloigné du stéréoscope, plus sa disparité est faible. Par contre, plus il est proche, plus sa disparité est grande. L'analyse que l'on peut tirer de ces graphiques est la suivante :

- Pour des objets proches du stéréoscope, donc avec une disparité relativement grande qui peut atteindre 90 pixels, l'écart E_r commis sur le calcul de la distance reste plutôt faible pour une erreur de localisation d'environ 1 à 2 pixels (inférieur à 2 %). Mais

l'écart peut atteindre 5, voire 10 %, si l'erreur de localisation est supérieure à 5 pixels. Il ne faut pas oublier que dans la pratique, les variations de niveaux de gris ne sont pas nettes, des erreurs de localisation relativement importantes sont donc envisageables.

- Pour les objets éloignés pour lesquels la disparité est relativement faible, on constate, pour des erreurs de localisation de 1 à 2 pixels, un écart E_r sur la mesure de distance pouvant aller de 3 à 15%. Pour des erreurs de localisation très importantes (5 à 10 pixels), l'écart peut aller de 15 à 50 %, voire plus pour des disparités vraiment très faibles.

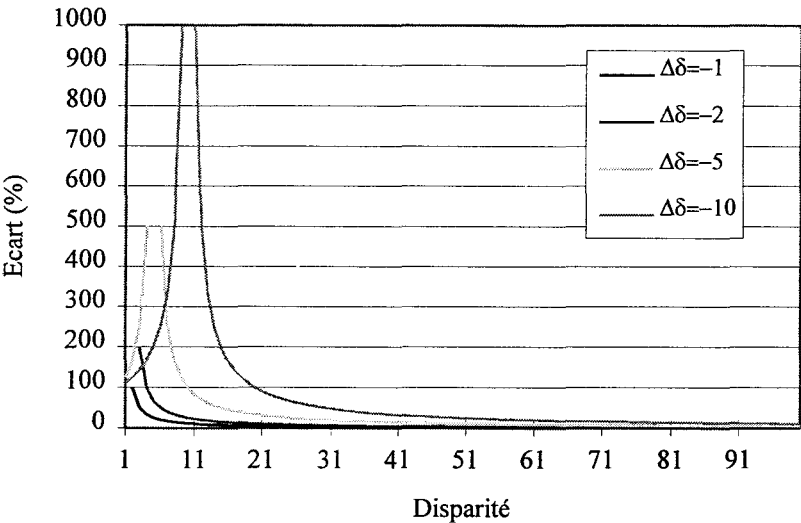


Figure 4.6 : Influence de l'erreur $\Delta\delta$ sur la mesure de distance ($\Delta\delta < 0$).

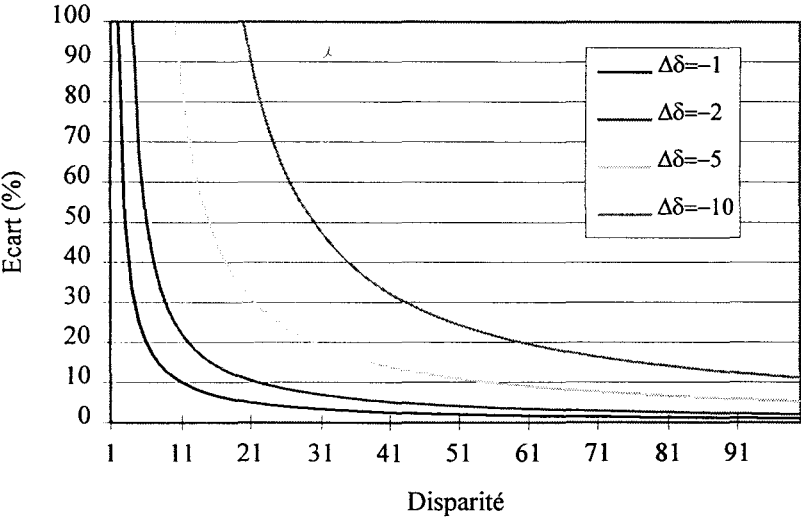


Figure 4.7 : Agrandissement de la figure 4.6.

La position d'un point de contour dans les images, c'est à dire la position des pixels qui reçoivent son image sur chaque capteur, étant directement liée à la distance, c'est à dire à la profondeur de l'objet par rapport aux capteurs (Cf. Equation 4.7), nous pouvons donc conclure que la moindre erreur sur le premier paramètre aura des conséquences importantes sur le second.

Cette analyse montre l'importance d'une localisation précise des contours. Tous les cas de figure pouvant se rencontrer dans des situations pratiques, ce type d'erreur conduirait à une mauvaise interprétation de l'environnement du stéréoscope. Dans le cas d'une application à la détection d'obstacles, les conséquences pourraient être graves. Toute erreur de localisation des contours pourrait être la cause d'une mauvaise estimation du danger.

Le détecteur utilisé devra donc localiser précisément les points de contour afin d'éviter des erreurs de localisation des objets dans la scène tridimensionnelle.

4-2.1.3. Ne pas créer de faux points de contour

Un détecteur de points de contour utilisant un opérateur de différentiation peut fournir au moins deux types de faux points de contour :

- La majorité des faux points de contour est engendrée par le bruit. Celui-ci peut être introduit par le dispositif de prise de vue lors de la formation de l'image sur le capteur. On parle de bruit « électronique ». Mais il peut aussi résulter de la nature de la scène observée, comme une scène texturée. On parle alors de bruit « sémantique ».
- Il existe un autre type de faux points de contour, moins connu. Ils sont engendrés par les doubles marches de type escalier (Cf. § 4-2.1.1). Ce type de contour peut induire pour un contour donné plusieurs réponses significatives de l'opérateur de différentiation. Il apparaît alors un, voire plusieurs, faux points de contour [Ziou 91].

Le détecteur devra donc être le moins possible sensible au bruit pour les raisons suivantes :

- Avec la présence de faux points de contour, la première conséquence pourrait être de trouver des objets là où il n'y en a pas et donc de déclencher des alarmes intempestivement, ce qui diminuerait la fiabilité du système.

- L'ajout de faux points à la liste des vrais points de contours augmenterait le nombre d'appariements possibles, d'où un accroissement du temps de traitement avec, bien sûr, une augmentation des possibilités d'erreur et de faux appariements.

4-2.1.4. Rapidité d'application

Le dernier point à considérer concerne le temps de traitement. Il faudra que l'extraction des points de contour puisse se faire le plus rapidement possible pour éviter de pénaliser les performances du système de détection d'obstacles.

4-2.2. Réduction de la quantité de bruit contenu dans l'image

La détection des contours fait appel à des opérateurs différentiels. Or, la différentiation numérique d'une image est une tâche délicate dans le sens où ses performances se détériorent rapidement en présence de bruit. En effet, nous avons vu au § 4-2.1.3 que le bruit peut générer des faux points de contours. De même, il peut être la cause d'une mauvaise localisation des contours puisqu'il peut les déplacer légèrement lorsqu'il s'ajoute au signal. Ces deux constatations vont à l'encontre des deux contraintes auxquelles doit se plier le détecteur de contour (Cf. § 4-2.1.2 et § 4-2.1.3). Il est donc impératif de réduire au maximum le bruit présent dans l'image.

D'après [Ziou 91], pour rendre la différentiation plus robuste, il est nécessaire d'utiliser la technique dite de régularisation. Il s'agit d'imposer de nouveaux critères au problème de différentiation pour restreindre l'espace des solutions possibles et de choisir celle qui minimise une fonction adéquate. Tikhonov [Tikhonov 77] (cité dans [Ziou 91]) montre que la régularisation numérique du problème de différentiation peut être effectuée simplement en lissant l'image.

Chaque image sera donc préalablement lissée afin de diminuer l'influence du bruit sur la détection des points de contour.

4-2.3. Organisation du détecteur de points de contour

En prenant en compte les observations précédentes, notre détecteur de contour inclura les opérations suivantes :

1. Le *lissage* : Il a pour objectif la réduction de la quantité de bruit contenu dans l'image. Il faudra néanmoins prendre certaines précautions. Cette opération a pour effet indésirable de modifier la structure de l'image par élimination et déplacement de certains contours et création de nouveaux contours. Il faudra établir un compromis entre l'élimination du bruit et la préservation des contours.
2. La *différentiation* ou *calcul du gradient* : c'est l'opération de calcul des dérivées nécessaire pour mettre en évidence les points de contour. L'opérateur utilisé devra, dans la mesure du possible, détecter tous les types de contours et les localiser le plus précisément possible.

La présentation de ces deux opérations fera l'objet du § 4-3, où nous détaillerons quelques opérateurs de lissage et de différentiation. Cependant l'exploitation des résultats obtenus avec un détecteur de contour nécessite une troisième étape : l'*extraction* des points de contour. Elle a pour objectif de déterminer parmi tous les maximums locaux de la fonction dérivée, ceux qui correspondent à des points de contour significatifs de l'image. L'*extraction* sera traitée au § 4-4.

4-3. Détection des points de contour

La détection de points de contour implique la recherche des variations locales et significatives de la fonction intensité de l'image. Ces variations sont mises en évidence en dérivant cette fonction. Le type et l'ordre des dérivées sont déterminés en général par le type du contour recherché et par leur utilisation ultérieure.

L'aspect monodimensionnel des signaux représentant les images linéaires nous amène à ne décrire les différents détecteurs que dans leur version monodimensionnelle. Le lecteur intéressé par une utilisation bidimensionnelle pourra trouver de plus amples informations dans les références données.

Dans le cas d'une image numérique, qui est une fonction discrète, l'estimation du gradient est donnée par des opérateurs discrets. Nous présentons maintenant quelques opérations de ce type.



4-3.1. Les masques de convolution

Les masques de convolution permettent de détecter les contours par approximation du gradient tout en réduisant le bruit. Plusieurs masques ou filtres sont utilisables dans un même but. Comme les performances des différentes méthodes dépendent de la nature des images, leur sélection se fait généralement par essais successifs.

4-3.1.1. Définitions

Dans le cas d'un signal monodimensionnel, le masque prend la forme d'un vecteur horizontal de coefficients. L'opération s'effectue en convoluant ce vecteur avec l'image de la manière suivante :

Soit V un vecteur de dimension n :

$$V = [v_1 \quad v_2 \quad \cdots \quad v_{n-1} \quad v_n]$$

où les v_i sont les coefficients du masque. On choisira n impair et $n \geq 3$.

Soit un ensemble de n pixels, appelé voisinage, centré sur le pixel considéré P de l'image présentant le niveau de gris p , et noté :

$$\left[P_{\frac{-n-1}{2}} \quad \cdots \quad P_{-1} \quad P \quad P_1 \quad \cdots \quad P_{\frac{n-1}{2}} \right]$$

où les p_i sont les niveaux de gris des pixels voisins.

Le filtrage consiste à remplacer le niveau de gris p de chaque pixel P de l'image par le niveau de gris p' avec :

$$p' = P_{\frac{-n-1}{2}} \cdot v_1 + \cdots + P_{-1} \cdot v_{\frac{n-1}{2}} + P \cdot v_{\frac{n+1}{2}} + P_1 \cdot v_{\frac{n+3}{2}} + \cdots + P_{\frac{n-1}{2}} \cdot v_n$$

On constate que le masque (vecteur) doit être de même taille que le voisinage. Afin que le filtre soit symétrique par rapport au pixel p considéré, le masque est constitué d'un nombre impair d'éléments. Généralement, le choix se porte sur des masques de taille assez petite d'environ 3 à 5 pixels.

Selon les valeurs des coefficients v_i , les masques de convolution permettent d'effectuer diverses opérations de filtrage.

4-3.1.2. Pré-traitement : opération de lissage

Afin de diminuer l'influence du bruit sur la détection des points de contour, les images sont préalablement traitées au moyen d'un opérateur de lissage. Nous présentons ci-dessous quelques opérateurs basés sur les masques de convolution.

Le filtre moyenneur :

Le filtre moyenneur est certainement le plus connu des filtres de convolution. Les coefficients du masque prennent les valeurs suivantes :

$$V = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad \text{où le coefficient } 1/3 \text{ sert à normaliser le résultat.}$$

Si ce filtre a pour effet de moyenniser le bruit sur plusieurs points, il atténue également les variations brusques de l'intensité, c'est à dire les contours qui de ce fait sont moins nets. L'effet produit se traduit visuellement par un effet de flou.

Des variantes de ce filtre sont possibles. On peut pondérer les coefficients de façon à donner plus d'importance au pixel central. On obtient par exemple :

$$V = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad \text{où le coefficient } 1/4 \text{ sert à normaliser le résultat.}$$

Le filtre médian :

Le filtre médian n'est pas une opération de convolution. Il s'applique également sur un voisinage donné mais c'est un filtre non linéaire.

Soit le voisinage de dimension n (avec n impair et $n \geq 3$) d'un pixel P . Les n pixels (le pixel P et ses $n-1$ voisins) sont ordonnés selon l'ordre croissant de leurs niveaux de gris. On affecte alors au pixel P la valeur médiane de l'ensemble des niveaux de gris ainsi ordonnés.

Le filtre médian produit un effet voisin du filtre moyenneur, mais il respecte beaucoup mieux les variations brutales, c'est à dire les contours.

Dans les deux cas, plus la taille de ces filtres est élevée (3, 5, 7, ...) plus leur effet de filtrage passe-bas contre le bruit est efficace.

4-3.1.3. Détecteurs de contour

Le détecteur le plus simple est de taille 3 et possède les coefficients suivants :

$$V = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad \text{Filtre A}$$

On peut étendre ce filtre sur 5 pixels. On obtient alors :

$$V = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Filtre B}$$

Si on utilise le filtre :

$$V = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 \end{bmatrix} \quad \text{Filtre C}$$

on introduit également un effet de moyennage.

Le filtre A donne d'excellents résultats pour des contours parfaits, c'est à dire des contours où la zone de transition entre deux niveaux de gris est étroite. Par contre si la largeur du contour dépasse la taille du masque, le filtre sera nettement moins efficace. On constate donc que ce type de filtre est très sélectif, et ne met en évidence qu'un certain type de contour. Il en est de même pour le filtre B dont l'efficacité est grande pour des contours de largeur inférieure à 5 pixels. Or, nous avons vu que dans les images naturelles, il en existe plusieurs types (Cf. § 4-2.1.1) et nous n'avons aucune connaissance a priori sur leurs caractéristiques.

Pour résoudre ce problème, Rosenfeld propose d'utiliser plusieurs filtres de tailles croissantes 3, 5, 7, ..., n chacun mettant en évidence des contours d'une certaine largeur. Il propose ensuite de retenir la valeur maximum entre les réponses des différents filtres utilisés [Rosenfeld 71] et [Rosenfeld 72].

Si les résultats de ce détecteur de contours sont excellents, la complexité de l'algorithme dépend du nombre de filtres utilisés. Si on en choisit un grand nombre, les contours de grande largeur seront mieux pris en considération, mais au prix de nombreux calculs.

4-3.1.4. Conclusion sur les masques de convolution

Les masques de convolution sont un moyen simple de lisser une image ou de calculer son gradient. Cependant, l'efficacité de ces filtres dépend essentiellement de la taille du masque utilisé.

Dans le cas d'une opération de lissage, plus le filtre est grand, plus il atténue le bruit. Mais ce résultat est obtenu au détriment des contours qui sont alors moins nets, et qui risquent de ne plus être détectés. Dans le cas du calcul du gradient, un second paramètre entre en jeu, à savoir le nombre de masques utilisés, puisque tous les contours ne peuvent être détectés avec un seul type de masque. Il faut également rappeler que plus la taille des masques est importante, plus le temps de calcul augmente, ce qui va à l'encontre du critère cité au § 4-2.1.4.

Le choix arbitraire de la taille des masques et les temps de calcul importants pouvant en résulter, nous ont conduits à ne pas retenir ce type de filtres.

4-3.2. Détecteurs optimaux

4-3.2.1. Définition d'un détecteur optimal

Le processus de lissage permettant de rendre la détection de contours moins sensible au bruit possède des effets indésirables qui peuvent être résumés par une perte d'informations (Cf. § 4-2.3). Plusieurs travaux ont été consacrés au problème de la recherche d'un filtre minimisant cette perte et différents critères ont été proposés. L'approche considérée consiste à déterminer le détecteur optimisant un critère donné. L'idée principale est de définir des critères de performance puis de mettre au point le détecteur optimal vis à vis de ces critères. Ce type d'approche semble le plus adéquat et connaît un succès croissant. Nous citerons les travaux de Canny et de Deriche.

4-3.2.2. Approche de Canny

Canny a étudié les propriétés souhaitées pour un détecteur de contours performant [Canny 83] (ou [Canny 86]). La méthode d'évaluation qu'il propose permet de définir des critères de performance dans le cas où l'opérateur de détection de contours est basé sur la convolution. Canny a retenu trois critères :

1. La qualité de la détection : le critère de détection permet de vérifier l'efficacité du détecteur de contours en tant que filtre appliqué sur un signal. La qualité de la détection est d'autant plus grande que le filtre est insensible au bruit présent dans l'image traitée.
2. La précision de la localisation : Le critère de localisation permet de quantifier la précision de la position du contour détecté dans une image bruitée par rapport à la position réelle du contour.
3. L'obtention d'une réponse unique pour un contour donné : Le critère de réponses multiples sert à déterminer si le filtre est susceptible de fournir plusieurs réponses pour un seul contour présent dans l'image.

Il faut noter que ces critères correspondent à ceux déjà mentionnés au § 4-2.1.

Canny propose une méthode pour déterminer un détecteur de contour optimal basé sur ces trois critères, en maximisant les critères de détection et de localisation. Le critère de réponses

multiples est introduit dans le processus de maximisation sous la forme d'une contrainte supplémentaire. Afin de simplifier le problème, Canny a imposé que son détecteur f soit à support fini, c'est à dire que f vérifie la contrainte :

$$f(x) = 0 \text{ si } x \notin [-W, W]$$

où W est une constante réelle positive marquant la limite de l'intervalle où f est défini.

Les critères sont évalués en appliquant le détecteur sur un signal monodimensionnel, décrit par une fonction continue d'une variable réelle x . Cette fonction est composée d'un contour idéal $G(x)$, centré en zéro, sur lequel est superposé un bruit $n(x)$ de variance n_o . La réponse $y(x)$ du filtre au signal est donnée en un point x_e par l'opération de convolution suivante :

$$y(x_e) = \int_{-\infty}^{+\infty} G(x_e - x) f(x) dx .$$

Le problème résolu par le biais des méthodes variationnelles aboutit au détecteur suivant :

$$f(x) = \begin{cases} \sin(\omega x)(a_1 e^{\alpha x} + a_3 e^{-\alpha x}) + \cos(\omega x)(a_2 e^{\alpha x} + a_4 e^{-\alpha x}) + c & \text{si } |x| \leq W \\ 0 & \text{sinon} \end{cases} \quad \text{Equation 4.10}$$

où α et ω sont des constantes réelles. Les coefficients a_1, a_2, a_3, a_4, c peuvent être déterminés grâce aux conditions limites suivantes :

$$f(0) = 0, f(W) = 0, f'(0) = S, f'(W) = 0 \quad \text{où } S \text{ est une constante.}$$

Pour faciliter son implantation, Canny propose d'utiliser comme détecteur la première dérivée d'une gaussienne, qui est une bonne approximation du détecteur de l'équation 4.10 :

$$f(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

Le signal est convolué avec la dérivée d'une gaussienne, dont le profil est représenté sur la figure 4.8. Le résultat est une approximation du gradient dont les extremums correspondent aux points de contour.

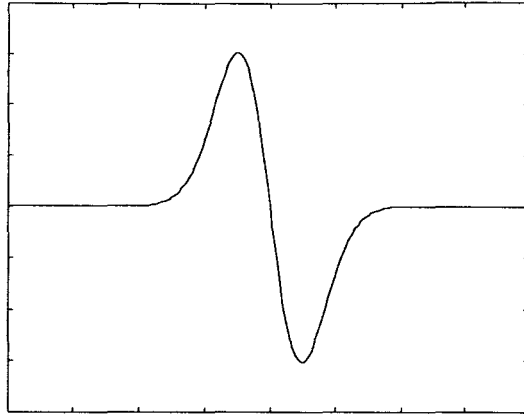


Figure 4.8 : Profil du détecteur optimal de Canny.

4-3.2.3. Approche de Deriche

4-3.2.3.1. Détecteur optimal de contour

Deriche a repris les 3 critères de performance de Canny [Deriche 87a] mais propose d'étendre le filtre à l'infini ($W \rightarrow \infty$). En utilisant les conditions limites données ci-dessous, avec W tendant vers l'infini, c'est à dire :

$$f(0) = 0, f(W) = 0, f'(0) = S, f'(\infty) = 0,$$

Deriche obtient le détecteur suivant :

$$f(x) = -ce^{-\alpha|x|} \sin(\omega x)$$

où c est une constante réelle et, α et ω des constantes positives.

Il montre que les critères de performance de Canny sont meilleurs pour ce détecteur que pour la première dérivée d'un gaussienne. Il remarque également qu'une meilleure localisation des points de contour est obtenue pour $\omega \approx 0$ c'est à dire : $\sin(\omega x) \approx \omega x$. Il obtient alors le détecteur optimal :

$$d(x) = -cxe^{-\alpha|x|} \tag{Equation 4.11}$$

où c est une constante réelle et α une constante positive.

La figure 4.9 représente le profil de l'opérateur de Deriche décrit par l'équation 4.11.

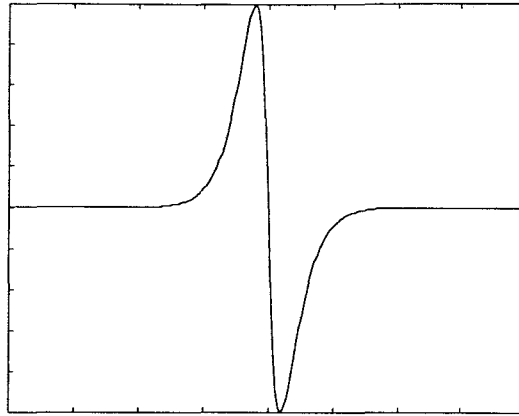


Figure 4.9 : Profil du détecteur optimal $d(x)$.

4-3.2.3.2. Opérateur de lissage

Il s'agit maintenant de déterminer un opérateur de lissage qui permettra d'éliminer efficacement le bruit. A partir du détecteur $d(x)$, Deriche détermine un opérateur de lissage $l(x)$ qui n'est autre que l'intégrale de l'opérateur de dérivation $d(x)$:

$$l(x) = s(\alpha|x| + 1)e^{-\alpha|x|} \tag{Equation 4.12}$$

où s est une constante réelle et α une constante positive.

La figure 4.10 montre l'allure de l'opérateur de lissage décrit par l'équation 4.12.

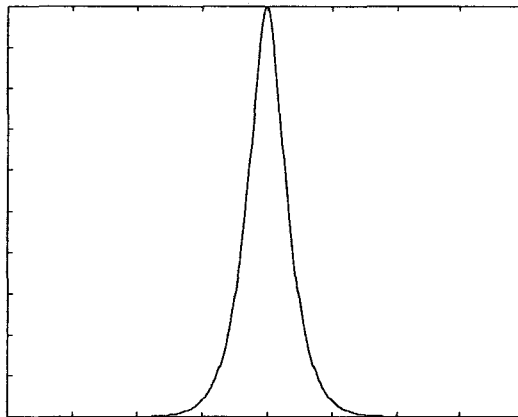


Figure 4.10 : Profil de l'opérateur de lissage $l(x)$ pour différentes valeurs de α .

4-3.2.3.3. Influence du paramètre α

La figure 4.11 illustre, pour les deux filtres $l(x)$ et $d(x)$, l'effet de l'ajustement du paramètre α qui permet de contrôler la taille de l'opérateur et donc de modifier le comportement du filtre. On constate que plus α est petit, plus l'étendue de l'opérateur est importante.

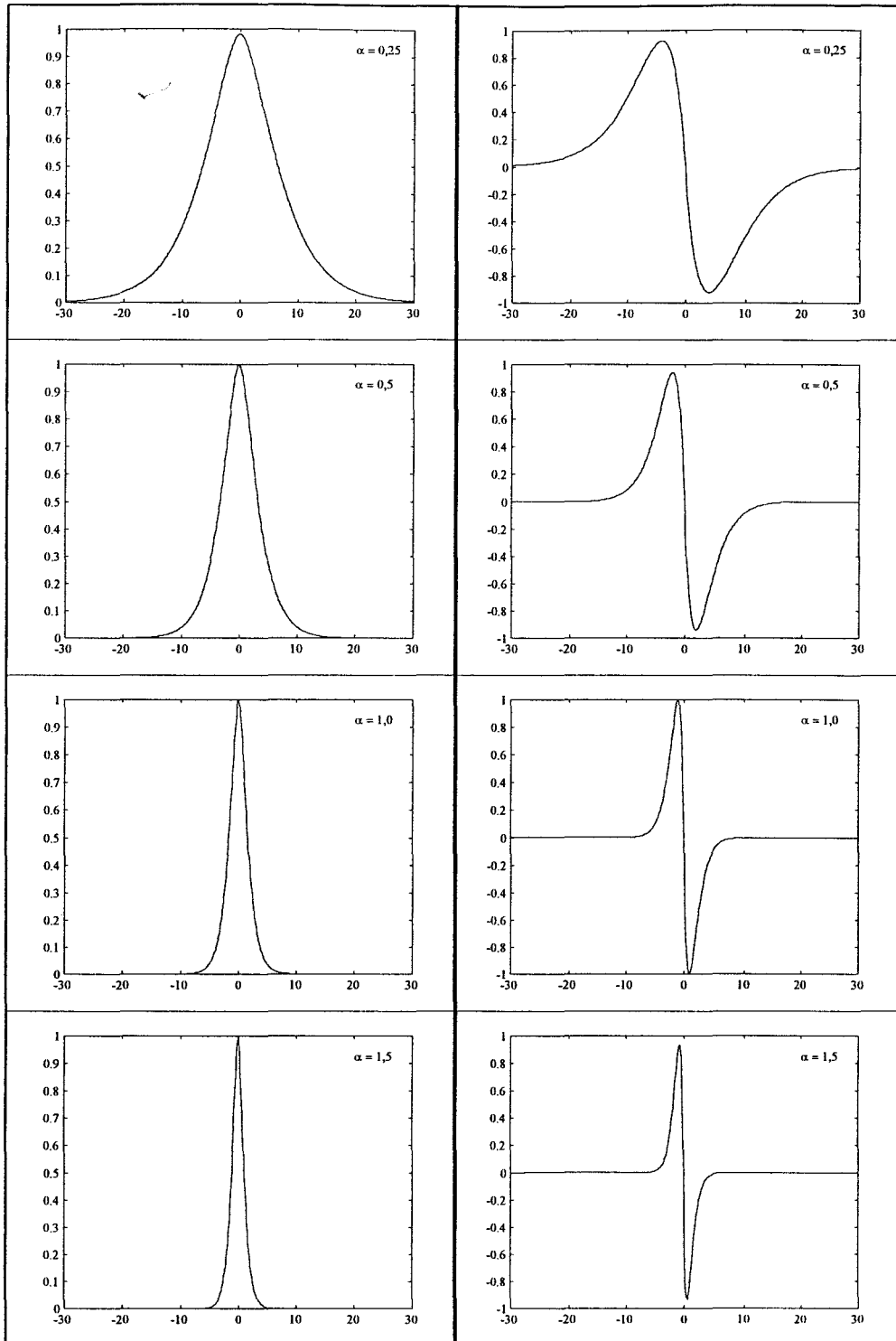


Figure 4.11 : Influence de α sur la taille des opérateurs $l(x)$ (à gauche) et $d(x)$ (à droite).

4-3.2.3.4. Conclusion

Nous avons choisi d'utiliser les filtres $l(x)$ et $d(x)$ de Deriche pour les raisons suivantes :

- Les filtres ont été développés en reprenant les critères de performances de Canny, à savoir :

1. La qualité des détections.
2. La précision de localisation.
3. L'obtention d'une réponse unique pour un contour donné.

Ce choix nous permet de satisfaire les critères mentionnés au § 4-2. Il faut noter que tout comme Canny, Deriche optimise les deux premiers critères en introduisant le troisième sous la forme d'une contrainte.

- Deriche a prouvé que son détecteur localise mieux les contours que celui Canny.

Nous avons donc cherché à implanter ces filtres d'une façon optimale afin que les opérations de lissage et de dérivation soient faites le plus rapidement possible.

4-3.3. Implantation de la détection de contour

Avant de mettre en oeuvre la détection, il reste cependant un point important à prendre en considération : le temps de calcul. L'opérateur bien que performant doit être également rapide pour les raisons citées au § 4-2.1.4.

4-3.3.1. Implantation récursive d'un filtre.

Deriche propose dans [Deriche 90b] et [Monga 91] une implantation récursive de ses opérateurs $l(x)$ et $d(x)$. Le principal avantage d'un filtre récursif réside dans la diminution du nombre total d'opérations élémentaires nécessaires à l'évaluation de la réponse, par rapport à un filtre non récursif de propriétés équivalentes.

L'implantation récursive est basée sur les constatations suivantes. En notant $x(i)$ la séquence d'entrée d'un filtre numérique non causal de réponse impulsionnelle infinie $h(k)$, la réponse $y(i)$ est donnée par l'expression suivante :

$$y(i) = \sum_{k=-\infty}^{+\infty} h(k)x(i-k) \quad \text{Equation 4.13}$$

Cette réponse $y(i)$ peut être décomposée en deux réponses, l'une $y_1(i)$ issue d'un filtre causal $h_1(k)$ appliqué à $x(i)$, l'autre $y_2(i)$ issue d'un filtre anticausal $h_2(k)$ appliqué à $x(i)$, selon l'expression :

$$y(i) = y_1(i) + y_2(i) \quad , \quad i = -\infty, \dots, 0, \dots, +\infty \quad \text{Equation 4.14}$$

$$y_1(i) = \sum_{k=-\infty}^{+\infty} h_1(k)x(i-k) \quad , \quad i = -\infty, \dots, 0, \dots, +\infty \quad \text{Equation 4.15}$$

$$y_2(i) = \sum_{k=-\infty}^{+\infty} h_2(k)x(i+k) \quad , \quad i = -\infty, \dots, 0, \dots, +\infty$$

Equation 4.16

$$h_1(k) = h(k), \text{ si } k \geq 0, 0 \text{ sinon}$$

$$h_2(k) = h(k), \text{ si } k > 0, 0 \text{ sinon}$$

Lorsqu'un filtre numérique causal $h_1(k)$ a une transformée en Z qui s'exprime sous la forme d'une fraction rationnelle $H_1(z)$ telle que :

$$H_1(z) = \frac{\sum_{k=0}^m b_k z^{-k}}{1 + \sum_{k=1}^n a_k z^{-k}} \quad 0 < m < n$$

Equation 4.17

la réponse $y(i)$ de ce filtre à une séquence d'entrée $x(i)$ est donnée par l'équation récurrente suivante :

$$y_1(i) = \sum_{k=0}^m b_k x(i-k) - \sum_{k=1}^n a_k y(i-k)$$

Equation 4.18

dans laquelle m et n désignent respectivement les degrés du numérateur et du dénominateur de la transformée en Z du filtre.

Si les transformées en Z des deux filtres $h_1(k)$ et $h_2(k)$ s'expriment sous la forme de fractions rationnelles $H_1(z)$ et $H_2(z)$, leurs réponses à une séquence d'entrée peuvent s'exprimer sous cette même forme récursive. Dans ce cas, le filtre complet $h(k)$ peut être implanté en utilisant deux équations de récurrence.

Il faut cependant noter que dans le cas d'une image numérique, l'étendue de la séquence d'entrée est limitée par la dimension M de l'image.

Le problème dans la conception d'un filtre récursif consiste à déterminer les coefficients a_k et b_k de la fonction de transfert (Cf. Equation 4.17). Deriche a présenté dans [Deriche 87b] une procédure pour déterminer ces coefficients pour les filtres les plus utilisés en détection de contour.

Cette méthode, appliquée aux filtres $l(x)$ et $d(x)$, amène aux équations récurrentes décrites dans les deux paragraphes suivants.

4-3.3.2. Lissage récursif

L'opérateur de lissage que nous avons retenu s'exprime sous la forme :

$$l(x) = s(\alpha|x|+1)e^{-\alpha|x|} \quad \text{Equation 4.19}$$

Deriche montre que le lissage d'un signal discret $x(n)$ avec $n = 1, \dots, M$ s'obtient à l'aide des équations récurrentes suivantes :

$$y_1(n) = k [x(n) + e^{-\alpha}(\alpha - 1)x(n-1)] + 2e^{-\alpha}y_1(n-1) - e^{-2\alpha}y_1(n-2) \quad \text{pour } n = 1, \dots, M$$

Conditions initiales : $x(0) = 0, y_1(0) = y_1(-1) = 0.$

$$y_2(n) = k [e^{-\alpha}(\alpha + 1)x(n+1) - e^{-2\alpha}x(n+2)] + 2e^{-\alpha}y_2(n+1) - e^{-2\alpha}y_2(n+2) \quad \text{pour } n = M, \dots, 1$$

Conditions initiales : $x(M+1) = x(M+2) = 0, y_2(M+1) = y_2(M+2) = 0.$

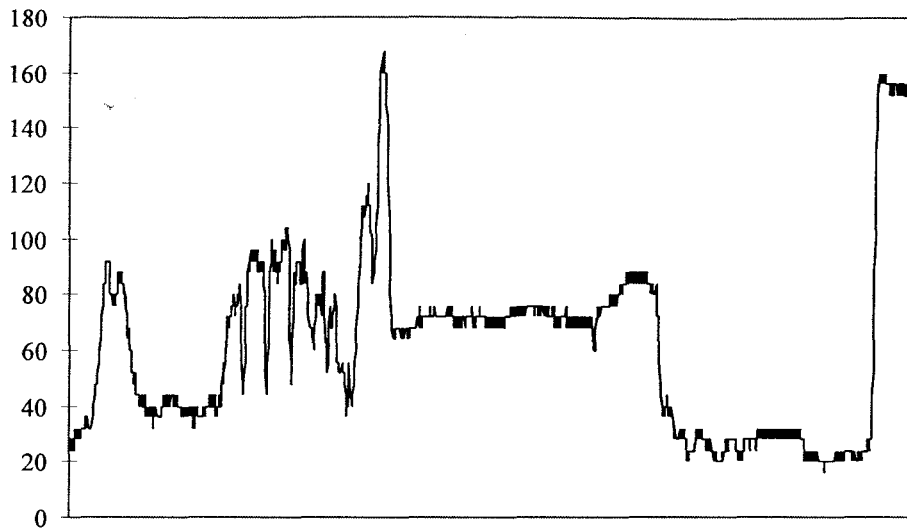
avec pour chaque équation : $k = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}.$

On calcule d'abord $y_1(n)$ pour n variant de 1 à M , puis $y_2(n)$ pour n variant de M à 1. Le signal lissé $y(n)$ s'obtient alors en prenant en compte les résultats de $y_1(n)$ et $y_2(n)$ de la manière suivante :

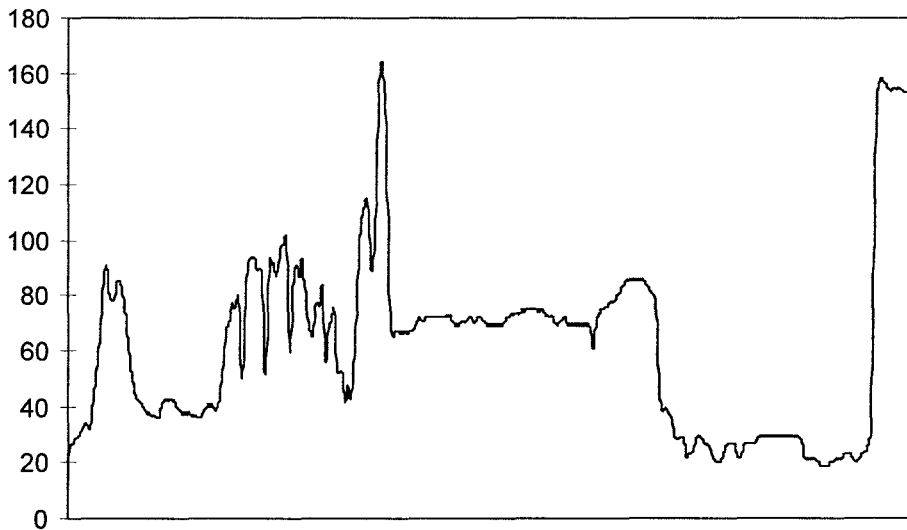
$$y(n) = y_1(n) + y_2(n) \quad \text{pour } n = 1, \dots, M$$

On remarque, d'après les équations ci-dessus, que nous pouvons modifier la valeur du paramètre α qui permet d'ajuster la taille de l'opérateur de lissage sans que cela augmente la quantité de calcul. En jouant sur le paramètre α , on pourra donc lisser l'image plus ou moins fortement sans que cela soit pénalisant en temps de calcul. Ceci montre l'avantage considérable qu'offre une implantation récursive du lissage.

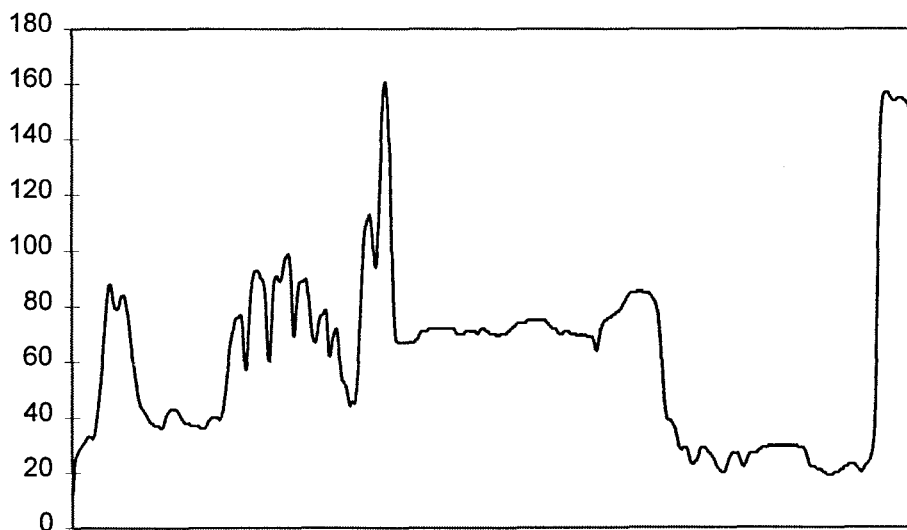
La figure 4.12 (de a à d) montre les résultats du lissage d'un signal monodimensionnel représentant une image linéaire et l'influence de α sur ce lissage.



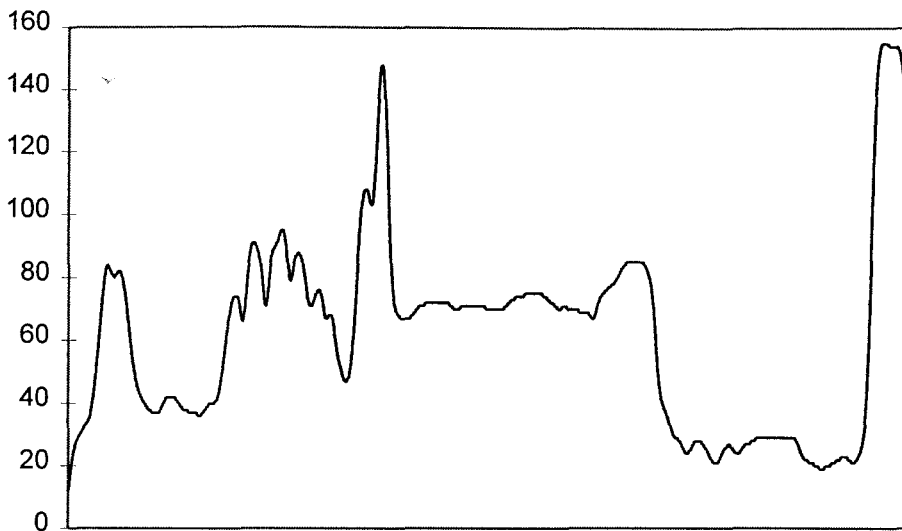
a) Image Originale.



b) Lissage pour $\alpha = 1,0$.



c) Lissage pour $\alpha = 0,5$.



d) Lissage pour $\alpha = 0,25$.

Figure 4.12 : Résultats du lissage d'une image linéaire pour différents α .

4-3.3.3. Calcul récursif du gradient

L'opérateur de dérivation que nous avons retenu s'exprime sous la forme :

$$d(x) = -cxe^{-\alpha|x|} \tag{Equation 4.20}$$

De la même manière que pour le lissage, le calcul du gradient d'un signal discret $x(n)$, avec $n = 1, \dots, M$, s'obtient à l'aide des équations récurrentes suivantes :

$$y_1(n) = x(n-1) + 2e^{-\alpha}y_1(n-1) - e^{-2\alpha}y_1(n-2), \quad \text{pour } n = 1, \dots, M.$$

$$\text{Conditions initiales : } x(0) = 0, y_1(0) = y_1(-1) = 0.$$

$$y_2(n) = x(n+1) + 2e^{-\alpha}y_2(n+1) - e^{-2\alpha}y_2(n+2), \quad \text{pour } n = M, \dots, 1.$$

$$\text{Conditions initiales : } x(M+1) = 0, y_2(M+1) = y_2(M+2) = 0.$$

avec pour chaque équation : $k = -\frac{(1 - e^{-\alpha})^2}{e^{-\alpha}}$

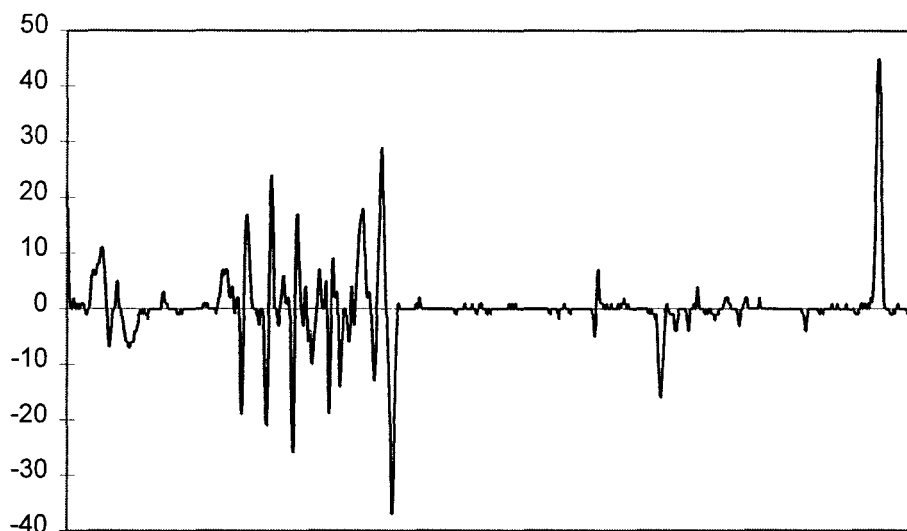
Le principe est le même que pour le lissage. Le gradient du signal s'obtient par l'équation :

$$y(n) = ke^{-\alpha} [y_1(n) - y_2(n)], \quad \text{pour } n = 1, \dots, M.$$

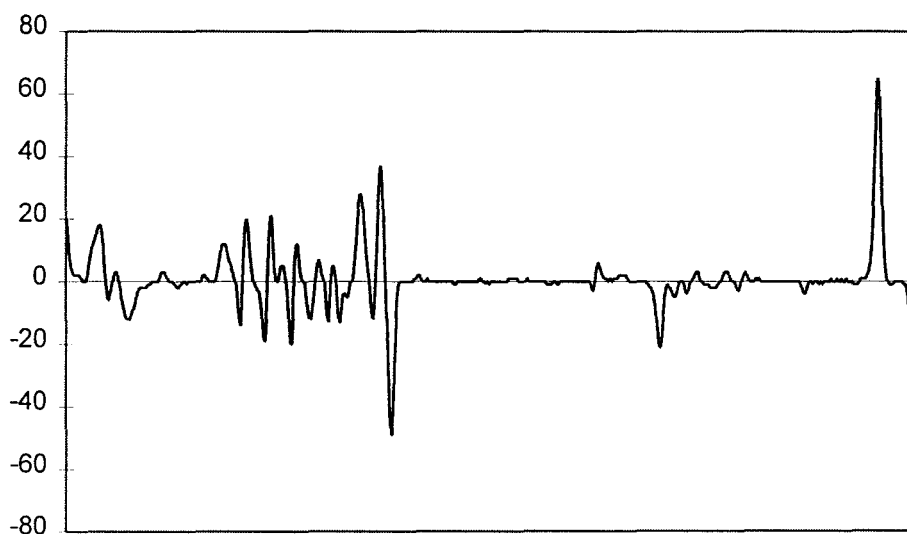
Nous pouvons faire la même remarque que pour le lissage. Le paramètre α , qui permet d'ajuster la largeur du filtre, ne joue pas sur le temps de calcul. Cependant, un compromis devra être trouvé. En effet, un opérateur étroit fera apparaître de nombreux contours locaux,

mais cachera la structure générale de la scène. En revanche, un opérateur large ne laissera apparaître que les contours les plus représentatifs au détriment des détails les plus fins.

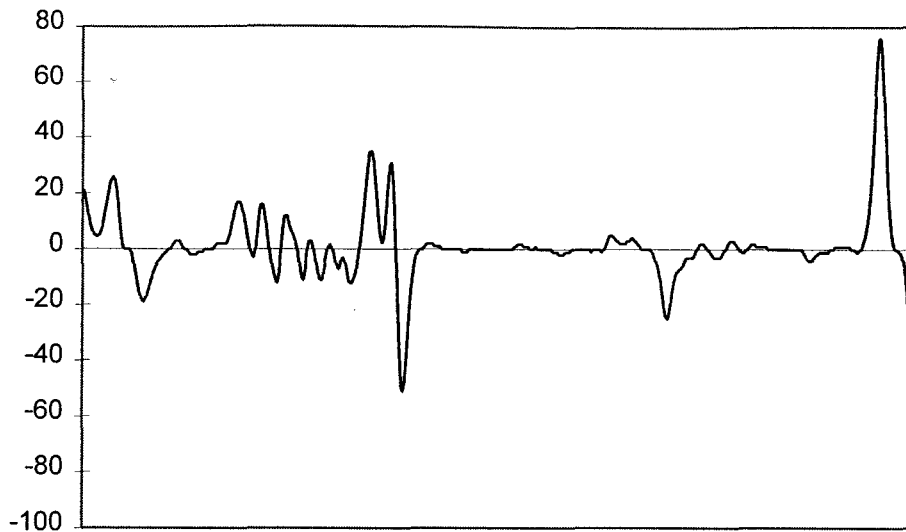
La figure 4.13 montre les résultats de la dérivation d'un signal monodimensionnel (Cf. Figure 4.12a) représentant une image linéaire et l'influence de α sur le calcul de la dérivée.



a) Dérivation pour $\alpha = 1,0$.



b) Dérivation pour $\alpha = 0,5$.



c) Dérivation pour $\alpha = 0,25$.

Figure 4.13 : Résultats de la dérivation d'une image pour différents α .

4-3.3.4. Conclusions sur le détecteur de contour

Nous avons présenté au § 4-2 un certain nombre de critères que devait posséder un bon détecteur de contour. Or il s'avère que les filtres de Deriche répondent parfaitement à ces critères, à savoir :

- Détection exhaustive des contours.
- Localisation précise des contours.
- Obtention d'une réponse unique pour un contour donné.
- Rapidité d'exécution grâce à l'implantation récursive quelque soit la taille de l'opérateur.

La détection des points de contour s'effectuera donc de la manière suivante :

1. Le lissage sera obtenu en appliquant les équations récursives du filtre $l(x)$ aux pixels de l'image linéaire $I(x)$. Le résultat obtenu sera l'image lissée $I_l(x)$.
2. Le calcul de l'image gradient sera ensuite réalisé en appliquant les équations récursives du filtre $d(x)$ aux pixels de l'image lissée $I_l(x)$.

D'après [Deriche 90b], le paramètre α doit être ajusté à des valeurs comprises entre 0,5 et 1,5 pour des images moyennement bruitées. Dans notre cas, l'expérience a montré que la valeur $\alpha = 1$ donne des résultats satisfaisants.

4-3.4. Détecteur optimal de Y. WAN

Nous tenons à mentionner, dans le cadre de notre projet, les travaux de Yuefeng Wan, chercheur au Centre d'Automatique de Lille. La détection des contours étant une phase importante dans un algorithme de traitement d'images et, particulièrement ici comme nous l'avons vu, il s'est donc penché sur le problème. Les résultats de ses travaux ont abouti à un nouvel opérateur de contour. Celui-ci est basé sur les critères de performance de Canny et a la forme d'une fonction sinusoïdale hyperbolique :

$$f(x) = e^{-\alpha|x|} \sinh(\beta x) = e^{-\alpha|x|} \left(\frac{e^{\beta x} - e^{-\beta x}}{2} \right)$$

Les performances de l'opérateur hyperbolique, comparées à celles des filtres optimaux de Canny et Deriche, sont globalement meilleures. Notamment, on constate que l'opérateur hyperbolique est moins sensible au bruit. Ce filtre sera donc, par la suite, implanté dans notre algorithme de détection d'obstacles afin d'en améliorer les performances.

Les travaux de Y. Wan seront développés dans sa thèse actuellement en cours de rédaction. Le lecteur intéressé peut néanmoins se référer pour le moment à [Wan 95].

4-4. Extraction des points de contour

La dernière étape de la détection des points de contour est la recherche des extremums locaux significatifs du gradient de l'image. Elle porte généralement le nom de segmentation. Cette étape est importante car on constate toujours la présence de faux points de contour et l'absence de certains points due aux détections manquées, et ceci malgré l'utilisation de détecteurs de plus en plus robustes.

4-4.1. Exemples de méthodes de segmentation

Nous présentons maintenant quelques méthodes utilisées pour segmenter les images.

4-4.1.1. Seuillage simple

C'est l'algorithme le plus connu, il consiste à supprimer tous les points dont l'amplitude du gradient est inférieure à un seuil donné. Cette méthode est la plus simple et a l'avantage d'être facile à implanter. Cependant elle donne de mauvais résultats dans la majorité des cas réels du fait de la non-uniformité des objets, du fond et de l'éclairage. De plus, le choix arbitraire du seuil peut avoir deux conséquences :

- Si le seuil est trop élevé, des points de contour significatifs peuvent être éliminés.
- Si le seuil est trop faible, tous les points de contour significatifs seront bien conservés, mais des points correspondants, soit à des points de contour non significatifs, soit à de faux points de contour, pourraient être pris en compte. On obtient alors un nombre important de points à traiter ensuite.

Cette méthode n'a pas été retenue, car elle ne permettait pas d'assurer l'extraction de tous les points de contour significatifs. Il faut également noter que cette méthode ne permet pas de reconstruire les points manquants.

4-4.1.2. Seuillage par hystérésis [Canny 83]

Canny propose un algorithme utilisant deux seuils, un seuil haut et un seuil bas. Le procédé employé est le suivant :

- Tous les points dont la valeur du gradient est supérieure au seuil haut sont conservés.
- Tous les points dont la valeur du gradient est inférieure au seuil bas sont éliminés.
- Les points, dont la valeur du gradient est comprise entre les deux seuils, sont conservés dans deux cas :
 1. Ils font partie d'une chaîne de points de contour dont un certain nombre de points ont une amplitude supérieure au seuil haut.
 2. Ils font partie d'une chaîne de points de contour dont tous les points ont une amplitude supérieure au seuil bas.

On constate donc que cette méthode fait appel à la notion de continuité du contour. Or, celle-ci n'est pas disponible dans le cas des images linéaires. Pour cette raison, le seuillage par hystérésis ne peut être utilisé dans notre cas.

4-4.1.3. Méthode basée sur l'analyse d'histogramme

L'histogramme d'une image permet de mettre en évidence la répartition statistique des niveaux de gris dans l'image. Si une image est composée d'un objet et d'un fond distincts par leurs niveaux de gris, l'histogramme traduira ces différences sous la forme d'une courbe dite *bimodale*, c'est à dire présentant deux maximums locaux : un pour le fond, un pour l'objet. Le meilleur seuil est celui qui permet de séparer la forme du fond. Pratiquement les résultats sont décevants lorsque l'image n'est pas binaire par nature. Les histogrammes des images réelles possèdent généralement de nombreux pics, on dit qu'ils sont multimodaux. La méthode

consiste alors à déterminer les seuils qui permettent de séparer ces pics. Mais très souvent, ces pics ne sont pas distincts ce qui rend difficile la détermination des seuils.

Il existe d'autres méthodes basées sur l'analyse d'histogramme. Nous pouvons citer les méthodes d'analyse de concavité d'histogramme [Rosenfeld 83], ou encore les méthodes de transformation d'histogramme [Weszka 79].

4-4.1.4. Autres méthodes de segmentation

On trouve dans la littérature d'autres méthodes plus complexes pour segmenter les images, notamment des méthodes basées sur l'entropie, la relaxation [Zucker 76] et des méthodes de seuillage multiple. Il faut remarquer que la plupart des techniques de segmentation de contours, comme la relaxation par exemple, ne sont pas applicables dans le cas des images linéaires puisque la notion de continuité des contours n'existe pas.

4-4.2. Problèmes à résoudre lors du seuillage

Les techniques de segmentation énoncées n'ont pas retenu notre attention pour deux raisons. Soit elles n'étaient pas applicables sur des images linéaires, soit les résultats obtenus sur nos images n'étaient pas satisfaisants. Nous avons donc développé notre propre méthode. Celle-ci est basée sur quelques réflexions personnelles.

4-4.2.1. Le problème du seuil

Lors d'une opération de seuillage, le choix du seuil pose généralement un problème délicat à résoudre. Il est difficile, voire impossible, de choisir le seuil adéquat permettant à la fois de sélectionner les contours significatifs et d'éliminer tous ceux qui ne le sont pas. De même, si l'on envisage l'utilisation de seuils adaptatifs, comment être sûr que tous les contours significatifs seront retenus? Nous avons vu que, dans le cadre de notre application, nous ne pouvions pas nous permettre d'en oublier (Cf. § 4-2.1.1). Nous avons donc décidé de rejeter les méthodes faisant intervenir un seuil pour sélectionner les extremums locaux.

4-4.2.2. Les différents extremums locaux

Tous les extremums locaux ne correspondent pas systématiquement à des contours significatifs. Même si ces nombreuses méthodes de détection sont robustes, on remarque la présence d'extremums générés par le bruit. Il arrive également que certains contours donnent naissance à une dérivée formée localement de deux extremums locaux. Il est donc nécessaire

d'éliminer ces informations qui augmenteraient les temps de traitement lors de la mise en correspondance et qui n'apporteraient rien à la compréhension de la scène.

4-4.2.3. Les extremums manquants

Il arrive que certains points d'un contour ne soient pas détectés. En vision matricielle, certaines méthodes de segmentation permettent de fermer des contours avec, par exemple, les techniques basées sur la relaxation [Davis 77], [Zucker 76]. En vision linéaire, l'information n'est pas suffisante pour utiliser de telles procédures, car la notion de continuité des contours n'est pas applicable. Nous ne pouvons donc pas prendre en compte cet aspect de la segmentation.

4-4.3. Extraction des points de contour par intervalle

La méthode que nous proposons s'applique à un signal monodimensionnel. Une éventuelle extension à un signal 2D est bien sûr envisageable, mais nous n'avons pas poussé nos investigations dans ce sens.

4-4.3.1. Elimination des extremums générés par le bruit

Si les extremums locaux dus au bruit sont toujours présents dans le gradient de l'image, on constate cependant deux choses. Premièrement, ils sont toujours grandement majoritaires dans l'image, ce qui explique notre volonté de les éliminer. Deuxièmement, la pratique nous a montré que l'amplitude de leur réponse à l'opérateur de différentiation était toujours très faible. La solution qui s'offre à nous est simple, elle consiste à n'utiliser qu'un seuil très faible.

Nous proposons donc d'utiliser un seuil bas S_b pour éliminer tous les extremums dont l'amplitude du gradient est comprise en S_b et $-S_b$. De cette manière, nous sommes sûrs de conserver tous les extremums correspondant à des contours significatifs. La figure 4.18 montre le résultat de cette première étape : le bruit et les faibles variations de niveaux de gris présents dans l'image sont éliminés.

L'expérience a montré que pour un seuil S_b fixé à 10, on éliminait tous les extremums locaux générés par le bruit.

4-4.3.2. Sélection des extremums significatifs par intervalle

A ce stade, nous pouvons émettre l'hypothèse que les extremums restants correspondent bien aux points de contour des objets présents dans l'image. Le problème réside dans l'existence

d'informations redondantes, c'est à dire qu'un contour peut donner naissance à plusieurs extremums locaux. Cette constatation est paradoxale quand on sait que le filtre optimal de Deriche a été déterminé en imposant la contrainte proposée par Canny qui est *l'obtention d'une réponse unique à un contour donné*.

Pour un contour parfait, avec une variation nette de la fonction niveau de gris, on obtient une réponse unique de l'opérateur de dérivation (Cf. Figure 4.14).

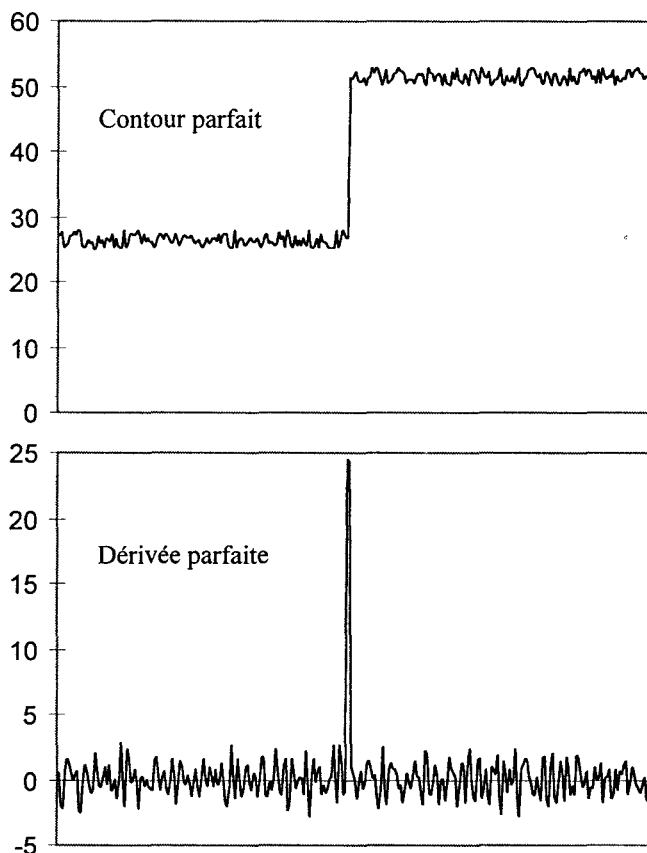


Figure 4.14 : Profil d'un contour parfait et de sa dérivée.

En pratique, les contours peuvent être plus ou moins larges, c'est à dire que la variation locale du niveau de gris de l'image peut être plus ou moins forte. De plus, cette variation n'étant pas forcément régulière, la dérivée peut présenter plusieurs extremums correspondants néanmoins à un seul contour (Cf. Figure 4.15).

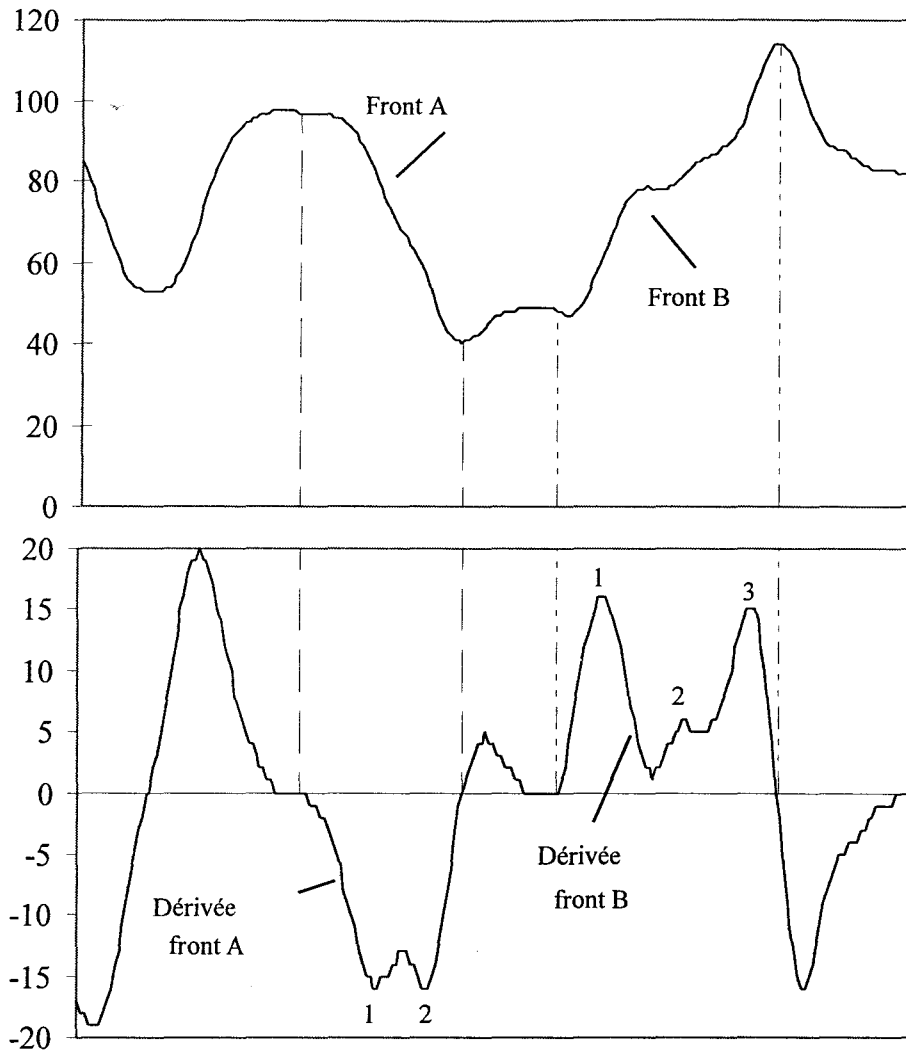


Figure 4.15 : Profil d'un contour engendrant une dérivée avec plusieurs extremums locaux.

Pour établir notre méthode de sélection des extremums locaux, nous avons posé certaines hypothèses.

Notre objectif est de conserver un seul extremum par contour. Il est donc d'abord nécessaire de pouvoir faire la distinction entre deux contours successifs pour éviter d'éliminer un vrai point de contour.

Hypothèses

- *Le signe de la dérivée est lié à la nature du contour* (Cf. Figure 4.16).

Il sera positif pour un contour caractérisé par un front montant.

Il sera négatif pour un contour caractérisé par un front descendant.

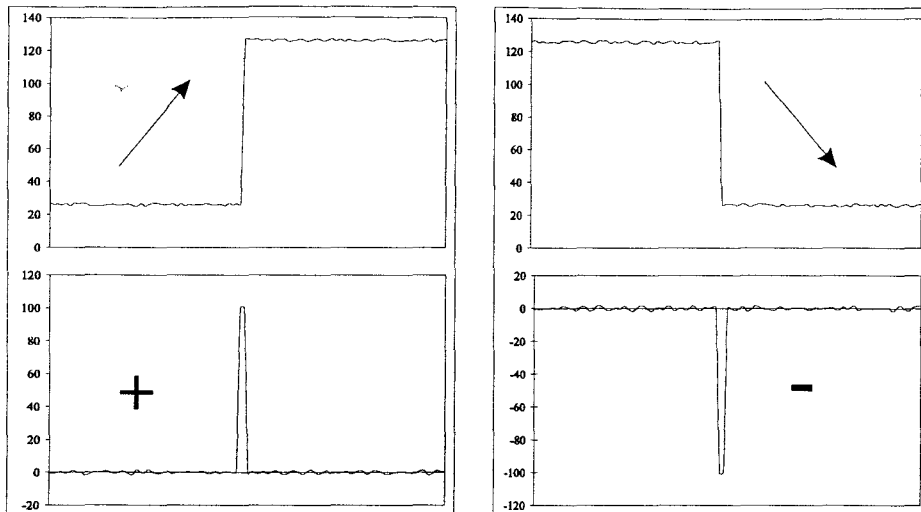


Figure 4.16 : Caractérisation du front par le signe de la dérivée.

- *Hypothèses de distinction des fronts* (Cf. Figure 4.17).

⇒ *Deux fronts successifs de même nature sont distincts si la dérivée effectue un passage par zéro.*

⇒ *Deux fronts successifs de natures différentes sont distincts si la dérivée change de signe.*

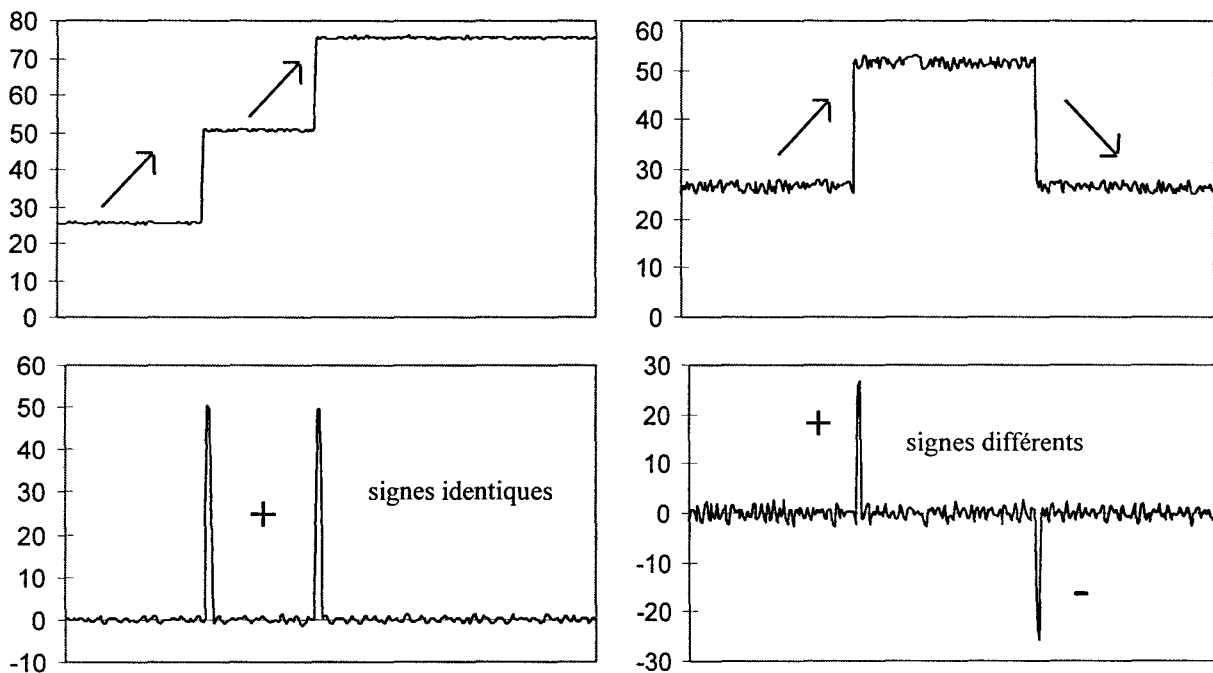


Figure 4.17 : Distinction des fronts.

Sélection des extremums

Nous nous proposons donc, pour caractériser chaque front, de diviser le signal dérivé en une suite d'intervalles adjacents. L'analyse est faite séquentiellement du début à la fin de l'image et un nouvel intervalle est créé chaque fois que la dérivée passe par zéro. Il ne reste plus ensuite qu'à sélectionner un seul extremum, quand il existe, dans chaque intervalle. Nous avons pour cela utilisé la méthode la plus simple : conserver l'extremum présentant l'amplitude la plus forte (Cf. Figure 4.18).

Notons que si, dans le cas d'un signal continu $f(x)$, un extremum local peut être défini simplement par :

$$\exists \alpha < 0, \exists \beta > 0 \quad \text{tel que} \quad |f(x)| > |f(x + \varepsilon)|, \forall \varepsilon \in]\alpha, \beta[,$$

dans le cas d'une image g , qui est un signal discret, les extremums se traduisent souvent par l'apparition de plateaux. La définition de l'extremum devient alors :

$$\exists k \in \mathbb{N} \text{ tel que } \begin{cases} |g_n| = |g_{n+i}|, \forall i \in]-k, k[\\ |g_n| > |g_{n+k}| \\ |g_n| > |g_{n-k}| \end{cases}$$

Dans ce cas, le point de contour est par convention localisé au centre du plateau.

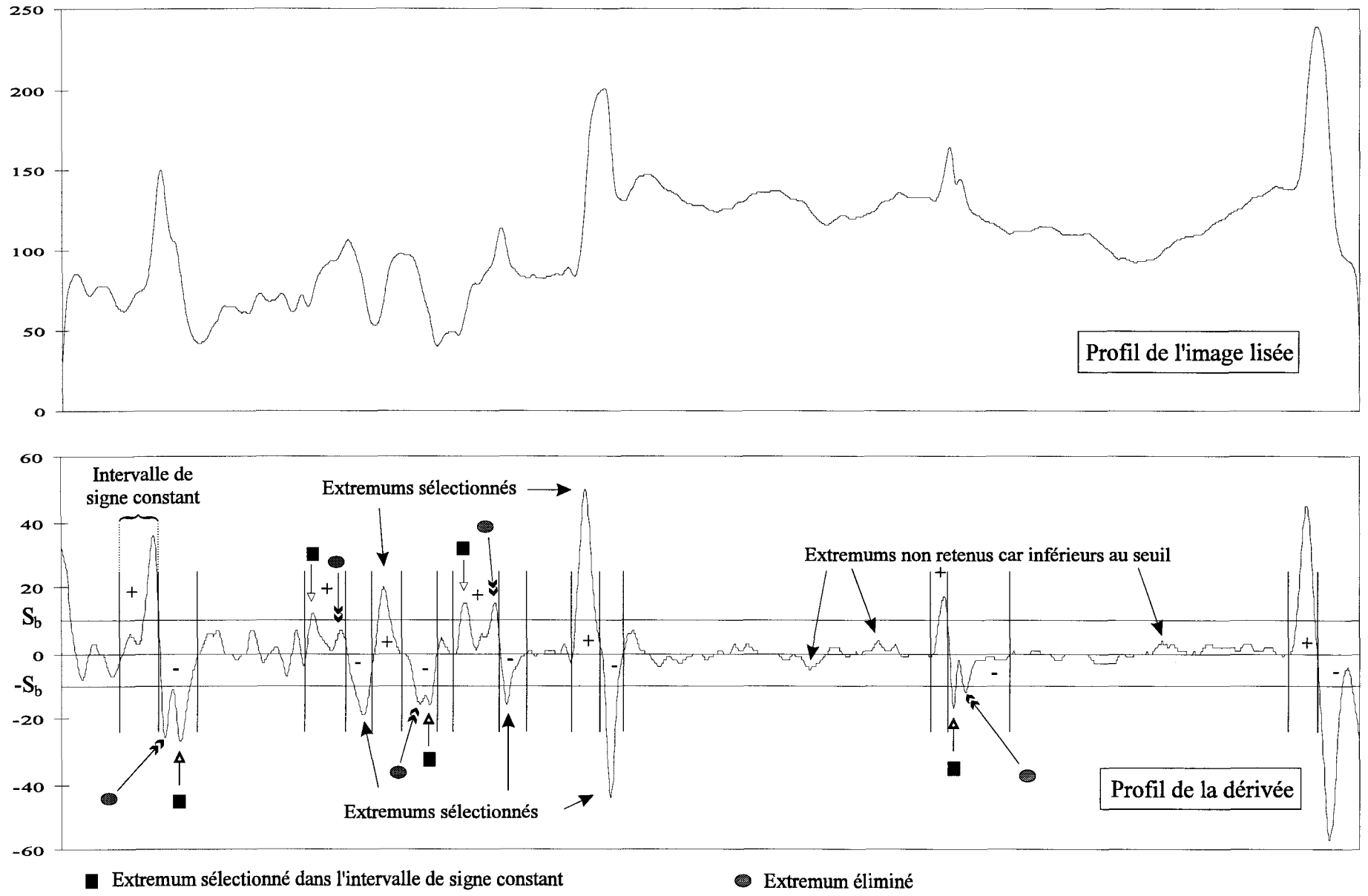
4-4.3.3. Algorithme d'extraction des points de contour

Afin d'extraire les points de contour de chaque image, l'algorithme d'extraction est appliqué successivement à la dérivée de l'image gauche puis à celle de l'image de droite. La procédure est implantée d'une manière optimale pour pouvoir être réalisée en une seule passe ce qui permet un gain de temps appréciable.

4-4.4. Conclusion sur l'extraction des points de contour

La méthode développée est originale dans le sens où la sélection des extremums locaux ne s'effectue pas en fonction d'un seuil. Cette technique permet ainsi d'éviter les inconvénients des procédés de seuillage classique qui sont sensibles à la non uniformité des objets, du fond et de l'éclairage. Le seuil utilisé, choisi délibérément petit, permet simplement d'éliminer les points de contour engendrés par le bruit et les faibles variations non significatives présentes dans l'image.

Figure 4.18 : Extraction des extremums par intervalles de signe constant.



Notre méthode de sélection des extremums locaux est efficace pour deux raisons :

- Elle permet de supprimer les extremums non significatifs provenant d'une double réponse du filtre à un contour.
- Elle conserve pour chaque objet présent dans l'image les contours les plus significatifs.

Le deuxième point est important, si l'on songe au cadre de notre application, car il nous assure de ne pas omettre d'obstacles.

4-5. Résultats de l'extraction des primitives

Nous présentons ici quelques résultats obtenus par le module d'extraction des primitives. Celui-ci est organisé en trois étapes successives :

- Lissage de l'image brut.
- Dérivation de l'image lissée.
- Extraction des extremums locaux.

Les résultats sont présentés sous forme d'images selon le même principe qu'au chapitre 2.

Pour obtenir une meilleure lisibilité, les images ci-après n'ont pas été rééchantillonnées afin d'éviter toutes pertes d'information. Dans un même but, chaque point de contour est représenté par un point noir, quelle que soit la valeur de l'amplitude du gradient.

Chaque point de contour sera utilisé comme primitive lors de la procédure de mise en correspondance.

4-5.1.1. Piéton évoluant devant le stéréoscope

Les figures 4.19 et 4.20 illustrent les résultats obtenus sur la séquence stéréoscopique "piéton", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.1).

On constate une très bonne détection des points de contour de l'image. Les bords des lignes blanches ainsi que la frontière entre la route et la zone d'ombre sont parfaitement détectés. De plus, aucune fausse détection n'apparaît au niveau de la chaussée malgré la présence de certaines irrégularités.

Nous pouvons dès à présent noter que le piéton engendre de nombreux points de contour. Si l'ensemble des points forme une structure cohérente au cours du temps. Les contours pris séparément ne respectent pas cette cohérence, c'est à dire qu'ils n'ont pas obligatoirement de

contour leur correspondant dans l'image suivante. Il faudra tenir compte de cette observation lors de la reconstruction tridimensionnelle. Nous reviendrons sur ce problème au chapitre 7.



Figure 4.19 : Extraction des primitives de l'image gauche (séquence stéréoscopique "piéton").

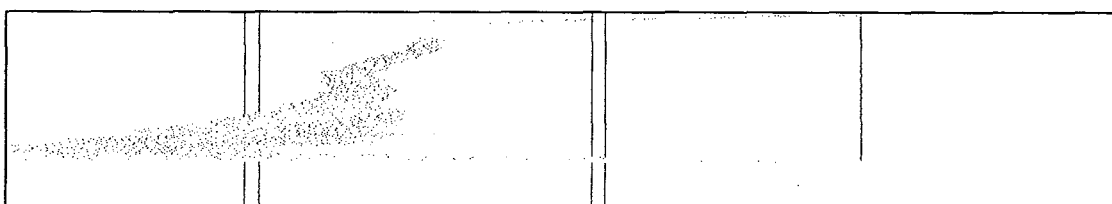


Figure 4.20 : Extraction des primitives de l'image droite (séquence stéréoscopique "piéton").

4-5.1.2. Route dégagée et passage d'une sortie d'autoroute

Les figures 4.21 et 4.22 illustrent les résultats obtenus sur la séquence stéréoscopique "sortie d'autoroute", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.2).

Comme dans l'exemple précédent, les lignes blanches sont très bien détectées et aucun point de contour n'apparaît sur les deux voies principales. Quelques points de contour, dus aux taches sombres situées sur la chaussée, apparaissent néanmoins au niveau de la sortie d'autoroute. On peut également remarquer que le bord gauche de la route engendre de nombreux points de contour qui ne correspondent pas à des obstacles et qu'il faudra éliminer.

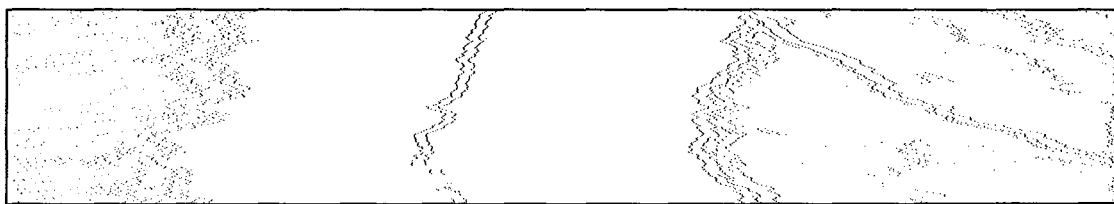


Figure 4.21 : Extraction des primitives de l'image gauche (séquence stéréoscopique "sortie d'autoroute").

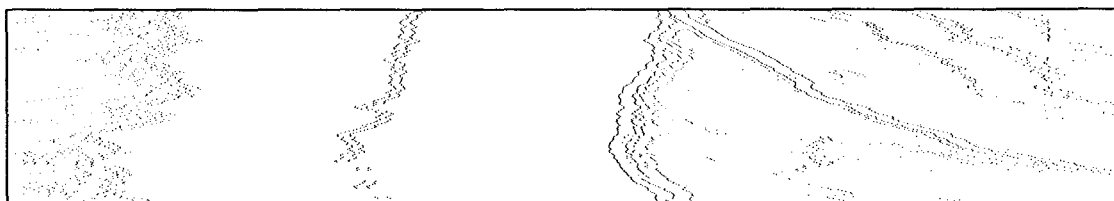


Figure 4.22 : Extraction des primitives de l'image droite (séquence stéréoscopique "sortie d'autoroute").

4-6. Attributs des primitives

Chaque primitive doit posséder des attributs qui seront utilisés, lors de la mise en correspondance, pour calculer les critères de ressemblance d'un couple de primitives gauche/droite et pour déterminer si les contraintes d'appariement sont satisfaites ou non.

Nous avons vu que les points de contour étaient les primitives les mieux adaptées aux images linéaires. Il s'agit donc, lors de la détection de ces points, de mémoriser pour chacun d'eux les attributs qui seront utilisés lors des appariements. Cette étape s'effectue en même temps que l'extraction des points de contour. Pour chacun des points, nous conservons en mémoire :

- Sa position dans l'image, c'est à dire son abscisse.
- La nature du point de contour (front montant ou descendant), ce qui revient à mémoriser le signe de la dérivée à la position qu'il occupe dans l'image.
- La valeur de l'amplitude du gradient.

Puisque nous ne connaissons pas le nombre de points de contour présents dans chaque image, les attributs de chaque point sont stockés dans une structure chaînée créée dynamiquement au fur et à mesure des besoins. A la fin de la procédure d'extraction des primitives nous disposons donc de deux listes (une par image) contenant les attributs de l'ensemble des points de contour. Ces deux listes seront transmises à la procédure de mise en correspondance. Notons que les deux images lissées (la gauche et la droite) seront également transmises à cette procédure.

4-7. Conclusion

Le choix des primitives est une phase importante dans un algorithme de stéréovision, car ce sont elles qui servent de base à l'appariement. Dans le cas des images linéaires, le faible contenu informationnel réduit considérablement les possibilités de choix. Nous avons montré que deux types de primitives étaient envisageables : les points de contour et les régions, mais que ces deux types étaient dépendants l'un de l'autre. Il en résulte que l'extraction des primitives passe par la détection des points de contour présents dans les images linéaires. Celle-ci est confiée à un détecteur de points de contour rapide et performant constitué de trois opérations distinctes et successives :

- Une opération de lissage destinée à éliminer le bruit présent dans l'image.
- Une opération de dérivation qui est appliquée à l'image lissée.
- Une opération d'extraction des extremums locaux significatifs correspondant aux contours pertinents de l'image.

Nous disposons donc à ce stade du traitement d'une liste de points de contour auxquels nous associons les caractéristiques suivantes :

- L'abscisse (position dans l'image).
- Le signe de la dérivée à l'abscisse du point (front montant ou descendant)
- La valeur de l'amplitude du gradient.

Cette liste de points, déterminée pour chaque couple d'images gauche et droite, servira à alimenter la procédure de mise en correspondance qui fait l'objet du chapitre suivant.

Chapitre 5

MISE EN CORRESPONDANCE D'IMAGES LINEAIRES STEREOSCOPIQUES

5-1. Introduction

Nous avons vu, dans le chapitre 3, les différentes méthodes existantes de mise en correspondance d'images stéréoscopiques (Cf. § 3-5.3). La plupart d'entre elles donnent d'excellents résultats, mais ont un inconvénient majeur : le temps des traitements est trop long pour envisager une application temps réel. De plus, certaines sont difficilement applicables dans le cas d'images linéaires. Dans le cadre de notre application, nous devons employer une procédure d'appariement rapide, mais également robuste. Afin d'optimiser les temps de traitement, nous utilisons des contraintes qui permettent de réduire la combinatoire du problème (Cf. § 5-3). Le module d'extraction des primitives, décrit au chapitre 4, nous fournit deux listes de points de contour. Ces deux listes constituent les variables d'entrée de notre méthode séquentielle de mise en correspondance (Cf. § 5-4). Pour permettre une amélioration des résultats, une mise en correspondance bidirectionnelle est réalisée (Cf. § 5-5). Afin de la rendre plus robuste, cette procédure a été modifiée afin d'apparier les primitives de façon hiérarchique, grâce à une implantation récursive (Cf. § 5-6). Lorsque les différentes primitives ont été appariées, les calculs de triangulation géométrique permettent de retrouver la structure tridimensionnelle de la scène (§ 5-7).

5-2. Remarques préliminaires

5-2.1. Définitions

Nous décrivons ici les différentes notations et termes que nous utiliserons au cours de notre exposé, ainsi que leurs significations.

Nous serons souvent amenés à mentionner les deux images stéréoscopiques. Nous employons les mots *gauche* et *droite* lorsque nous faisons référence aux images issues respectivement des capteurs gauche et droit du stéréoscope linéaire. Dans une image, les pixels sont repérés par leur abscisse dont l'axe est dirigé de la gauche vers la droite de l'image. Pour éviter de confondre la notion de gauche et droite d'une image avec celle d'image gauche et droite, nous employons le mot *début* pour évoquer la partie la plus à gauche de l'image et le mot *fin* pour la partie la plus à droite. Ainsi, pour une image de N pixels, le pixel d'abscisse zéro correspond au début de l'image et, le pixel d'abscisse N-1 à la fin.

L'extraction des primitives, détaillée au chapitre 4, fournit deux listes de points de contour ordonnés selon leurs abscisses croissantes :

- Soit $G(n)$ la liste des points de contour issus de l'image *gauche* avec $n = 0, 1, 2, \dots, N$.
- Soit $D(m)$ la liste des points de contour issus de l'image *droite* avec $n = 0, 1, 2, \dots, M$.

Pour des raisons que nous exposerons ultérieurement, nous considérons le premier et le dernier pixel de chaque image comme des points de contour. Ceux-ci portent respectivement les indices $n = 0$ et $n = N$ dans l'image de gauche et $m = 0$ et $m = M$ dans l'image de droite. De ce fait, la liste $G(n)$ contient donc réellement N_g points de contour, avec $N_g = N - 1$. De même la liste $D(n)$ en contient M_d , avec $M_d = M - 1$.

Lorsque nous tentons d'apparier le point $G(i)$ avec le point $D(j)$, ils constituent une *paire* que nous notons $\{G(i), D(j)\}$. Lorsqu'une paire $\{G(i), D(j)\}$ représente un couple de primitives appariées, nous l'appelons *paire valide*. Chaque paire possible est représentée graphiquement par un *noeud* (Cf. § 5-2.2)

La dernière définition concerne la notion d'*intervalle*. Celui-ci est défini comme l'ensemble des pixels d'une zone de l'image délimitée par deux points de contour d'une même image. Il est noté $[G(i); G(i+k)]$, avec $1 \leq k \leq N-i$ pour l'image gauche et $[D(j); D(j+k')]$ avec $1 \leq k' \leq M-j$ pour l'image droite. Quand $k = 1$ et $k' = 1$, les intervalles $[G(i); G(i+1)]$ et $[D(j); D(j+1)]$ représentent les zones délimitées par deux points de contour consécutifs. Ces zones, que nous avons déjà mentionnées au chapitre 4, sont en fait les *régions*. Nous retrouvons ici la dualité contours-régions que nous avons décrite au § 4-1.3 où nous avons montré que contours et régions sont étroitement liés.

Nous serons amenés à comparer les niveaux de gris des pixels contenus dans les intervalles de l'image gauche et de l'image droite. Pour limiter l'influence du bruit, ce sont les images lissées, obtenues par le filtre de Deriche (Cf. § 4-3.3.2), qui servent dans les calculs. Les images brutes ne sont pas utilisées lors de la mise en correspondance.

5-2.2. Représentation graphique

La procédure que nous avons développée est simple à mettre en oeuvre. Cependant il est difficile de l'expliquer de façon abstraite. Pour permettre une meilleure compréhension, nous nous aidons d'un graphique qui représente ce que nous appelons le « *plan de mise en correspondance* » (Cf. Figure 5.1). Cette représentation s'inspire de celle utilisée par Ohta dans [Ohta 85] pour expliquer son algorithme de mise en correspondance.

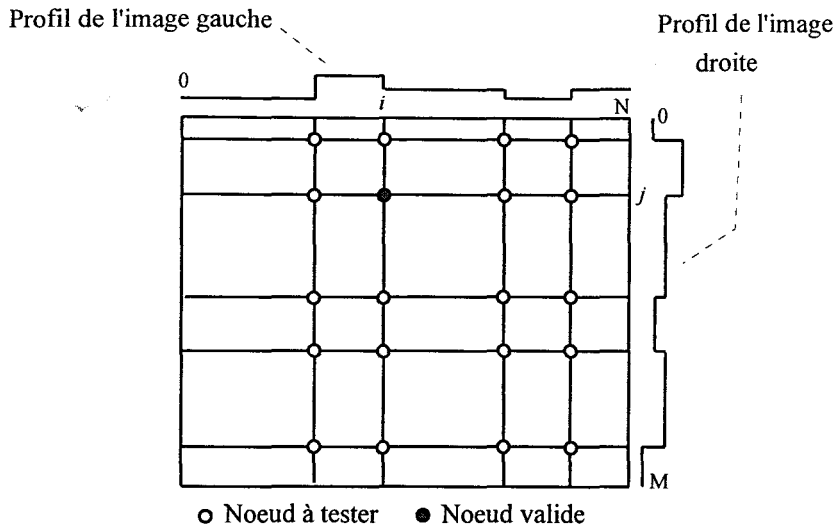


Figure 5.1 : Plan de mise en correspondance.

L'image gauche (respectivement droite) est représentée par son profil reproduit horizontalement (resp. verticalement) en haut (resp. à droite) du plan. Les lignes verticales (resp. horizontales) indiquent les positions des points de contour sélectionnés dans l'image de gauche (resp. droite). Chaque intersection, que nous appelons *noeud*, est définie par une paire $\{G(i), D(j)\}$ et symbolise un emplacement où un test d'appariement doit être effectué. Nous représentons graphiquement un noeud devant être testé par un disque blanc. Nous utilisons le terme de *noeud valide*, lorsque la paire $\{G(i), D(j)\}$ qui le définit a été appariée. Un noeud valide est représenté graphiquement par un disque gris.

Si l'on dispose de N_g points de contour dans l'image gauche et de M_d points dans l'image droite, $N_g \times M_d$ noeuds doivent être testés par la procédure de mise en correspondance. Dans le cas de la figure 5.1, où $N_g = 4$ et où $M_d = 5$, 20 noeuds sont donc examinés. Nous allons voir comment réduire le nombre de tests à effectuer en appliquant les contraintes de mise en correspondance.

5-3. Contraintes de mise en correspondance

Nous avons vu, au chapitre 3, que les critères de ressemblance locaux et purement photométriques ne sont en général pas suffisamment discriminants pour effectuer des mises en correspondance. Pour palier ce défaut, l'application d'un certain nombre de contraintes globales est donc nécessaire. Il faut toutefois noter que certaines contraintes utilisées en stéréovision matricielle ne sont pas applicables en stéréovision linéaire. Nous détaillons ci-dessous les différentes contraintes que nous utilisons dans notre procédure de mise en correspondance.

5-3.1. Contrainte d'unicité

Cette contrainte impose que tout point de contour de l'image linéaire gauche ait un et un seul correspondant dans l'image linéaire droite et réciproquement. Nous avons vu au § 3-4.4.2 que certains alignements accidentels de primitives entraînaient une transgression de cette contrainte. En vision linéaire, ce type de configuration est évidemment possible, mais nous excluons cette possibilité pour des raisons identiques à celles mentionnées au chapitre 3.

5-3.2. Contrainte d'ordre

Cette contrainte suppose que l'ordre des points de contour est le même dans chaque image linéaire. Si l'on considère des primitives de type point, comme nos points de contour, nous pouvons nous trouver assez souvent dans la configuration représentée sur la figure 5.2 (Cf. § 3-4.4.1).

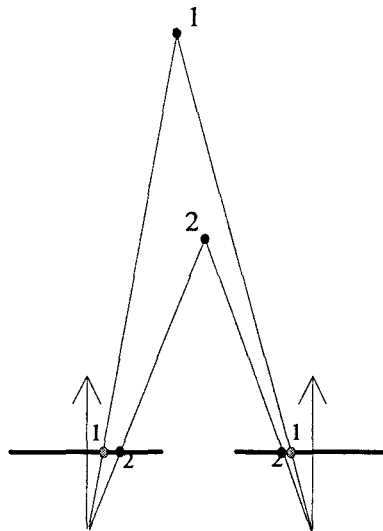


Figure 5.2 : Cas d'objets ponctuels où la contrainte d'ordre n'est pas respectée.

Il faut cependant noter que les objets ponctuels sont plutôt rares dans la nature. Les points de contour que nous utilisons en tant que primitives correspondent en fait aux frontières des objets. Nous devons donc tenir compte de la taille de ces objets. La figure 5.3 illustre ce qui se produit dans une configuration identique à celle de la figure 5.2, mais en considérant la présence des objets. L'objet à l'avant plan occulte celui qui se trouve à l'arrière plan qui n'a donc plus d'image sur le capteur droit. On constate donc que la contrainte d'ordre n'est plus transgressée. Il faut préciser que l'on exclut la présence d'objets transparents qui rendraient ce raisonnement inexact.

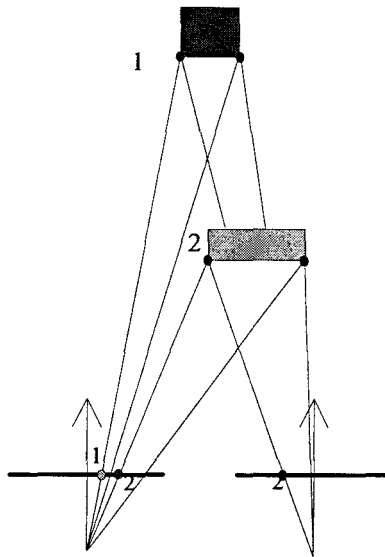


Figure 5.3 : La contrainte d'ordre n n'est pas transgressée.

On suppose donc que les situations avec des objets ponctuels pour lesquelles la contrainte d'ordre est transgressée ne peuvent se produire pour des objets courants. Cette hypothèse est tout à fait valable, dans le contexte routier qui est le cadre de notre application.

On peut également noter l'apparition du problème dû aux occlusions. Celui-ci sera traité dans le prochain chapitre (Cf. § 6-5.4).

Nous admettons donc que la contrainte d'ordre est toujours vérifiée dans les images linéaires.

5-3.3. Contrainte épipolaire

La contrainte épipolaire est la seule contrainte qui ne fait pas d'hypothèse sur la scène. Elle dépend uniquement de la géométrie du capteur. Nous avons vu au chapitre 3 qu'elle impose que deux primitives homologues se trouvent nécessairement sur des droites épipolaires conjuguées (Cf. § 3-4.2). Dans notre cas, cette contrainte est implicite à la géométrie du stéréoscope linéaire. En effet, la stéréovision linéaire n'est possible que si un point de la scène réelle est vu simultanément par les deux capteurs lignes, ce qui correspond à la contrainte épipolaire.

Nous pouvons cependant affiner la définition de cette contrainte. Dans le cas de caméras à axes optiques parallèles, nous avons vu au chapitre 2 (Cf. Equation 2.5) que la position d'un objet dans la scène réelle pouvait s'exprimer en fonction de l'abscisse x_g (resp. x_d) de son image sur le capteur gauche (resp. droit). Ces relations sont les suivantes :

$$\begin{cases} X = \frac{x_g \cdot D}{f} - \frac{E}{2} \\ X = \frac{x_d \cdot D}{f} + \frac{E}{2} \end{cases}$$

Si l'on exprime x_g en fonction de x_d , on obtient alors la relation :

$$x_g - x_d = \frac{E \cdot f}{D} \tag{Equation 5.1}$$

où les paramètres E , f et D représentent respectivement l'entre axe des caméras, la distance focale des objectifs et la distance d'un objet par rapport au stéréoscope linéaire. Ces paramètres étant strictement positifs, on obtient la contrainte suivante :

$$x_d < x_g \tag{Equation 5.2}$$

Nous appliquons donc la contrainte épipolaire en imposant la relation définie par l'équation 5.2 aux abscisses des primitives que l'on tente d'apparier. Tout couple de primitives ne respectant pas cette condition est considéré comme incompatible, ce qui permet de réduire le nombre d'appariements possibles.

Graphiquement, cette contrainte s'exprime sous la forme d'une diagonale coupant le plan de mise en correspondance en deux. L'ensemble des appariements possibles est représenté par les noeuds se trouvant dans la partie supérieure du plan de mise en correspondance (Cf. Figure 5.4). Dans notre exemple, nous réduisons à l'aide de la contrainte épipolaire le nombre de test d'appariement à réaliser de 20 à 12.

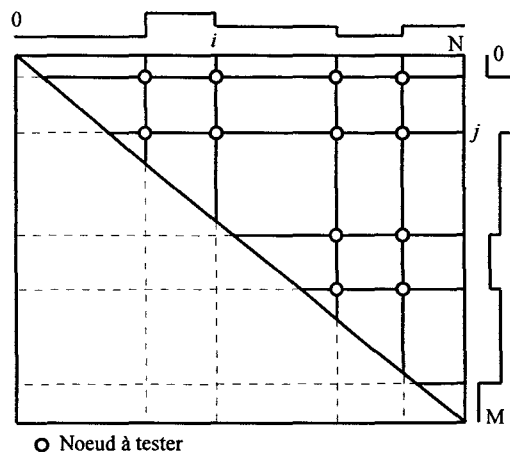


Figure 5.4 : Application de la contrainte épipolaire.

Si on compare l'équation 5.1 à l'équation 2.3, on constate que l'on peut simplifier l'équation 2.2, qui définit la disparité, en la remplaçant par la relation :

$$\delta = x_g - x_d \quad \text{Equation 5.3}$$

5-3.4. Contrainte de signe du gradient

Dans sa thèse, J.L. Bruyelle [Bruyelle 94] a proposé d'utiliser une contrainte photométrique qu'il a appelée la *contrainte de signe du gradient*. Il utilise l'attribut de signe du gradient que nous avons évoqué au chapitre 4 (Cf. § 4-6) en supposant que deux primitives gauche et droite engendrées par le même point présentent localement des dérivées de même signe.

Une dérivée positive correspond à un *front montant*, c'est à dire une variation du niveau de gris évoluant du sombre vers le clair pour un déplacement selon une abscisse croissante. Par contre, une dérivée négative indique un *front descendant*, c'est à dire une variation du niveau de gris du clair vers le sombre.

J.-L. Bruyelle a montré qu'un front montant ne peut pas être mis en correspondance avec un front descendant. Ceci implique que deux primitives correspondant au même point de la scène présentent des fronts de même nature.

Il faut cependant noter que cette affirmation peut être mise en défaut lorsqu'un objet transparent fortement incliné par rapport au plan des capteurs se situe entre les deux caméras, les primitives engendrées par cet objet peuvent présenter des dérivées de signes opposés, c'est à dire des fronts de nature différente. Ces cas étant extrêmement rares dans un contexte routier, nous excluons la possibilité de rencontrer des objets transparents.

La contrainte s'applique de la façon suivante. Tout couple de primitives ne respectant pas la contrainte de signe du gradient est considéré comme incompatible et on ne tente pas de les apparier.

Pour représenter graphiquement cette contrainte, il suffit de conserver les noeuds correspondant à des couples de primitives présentant des fronts compatibles. La figure 5.5 montre l'ensemble des appariements possibles après application de la contrainte épipolaire et de la contrainte du signe du gradient. Dans notre exemple, l'application de ces deux contraintes permet de réduire le nombre des appariements possibles à 6.

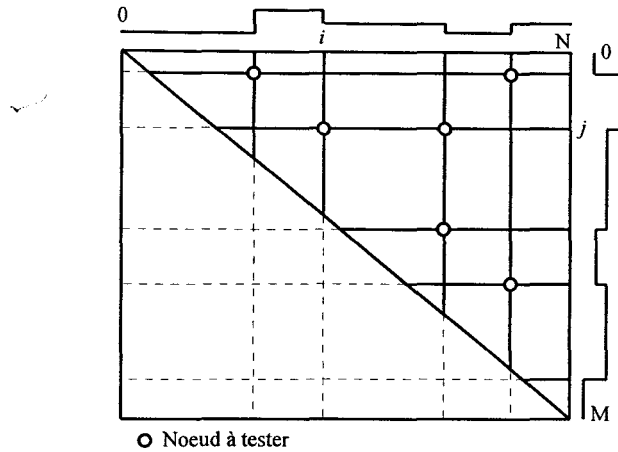


Figure 5.5 : Application des contraintes épipolaire et de signe du gradient.

5-3.5. Conclusion

Notre procédure de mise en correspondance utilise les quatre contraintes que nous venons de présenter, à savoir :

1. La contrainte épipolaire.
2. La contrainte de signe du gradient.
3. La contrainte d'unicité.
4. La contrainte d'ordre.

Les contraintes 1 et 2 sont des contraintes locales. Lorsque nous tentons d'apparier deux primitives, nous contrôlons d'abord qu'elles vérifient ces deux contraintes simultanément. Dans l'affirmative, la procédure de mise en correspondance se poursuit. Dans le cas contraire ce couple est immédiatement rejeté et la procédure de mise en correspondance passe au couple suivant. Ces deux contraintes permettent de réduire le nombre des possibilités d'appariements et donc de diminuer les temps de traitement.

Les contraintes 3 et 4 sont utilisées dans un raisonnement plus global qui sert à établir la procédure de mise en correspondance décrite dans le prochain paragraphe.

La contrainte d'unicité s'exprime graphiquement par la présence d'un seul noeud valide, au plus, sur chaque ligne du plan de mise en correspondance. Sur l'exemple de la figure 5.5, on constate que certaines lignes horizontales et verticales comprennent plusieurs noeuds, la procédure de mise en correspondance doit donc éliminer les noeuds incorrects.

5-4. Algorithme séquentiel de mise en correspondance

5-4.1. Approche intuitive de la méthode

Nous avons vu, au chapitre 3, que la mise en correspondance des primitives est un problème délicat. Pour mettre au point une nouvelle méthode de mise en correspondance, nous nous sommes basés sur les raisonnements que l'être humain emploierait pour résoudre certains problèmes complexes. Nous exposons notre approche à l'aide d'exemples. Ceux-ci sont basés sur des figures symbolisant des couples d'images stéréoscopiques. Ces exemples sont présentés sous la forme de « Question/Réponse » dans lesquels il faut mettre en correspondance des lignes blanches. Pour simplifier le problème, nous supposons dans ces exemples que les contraintes d'ordre et d'unicité sont respectées, qu'il n'y a pas d'occlusions, et que toutes les lignes sont dans le secteur de stéréovision, c'est à dire que chaque ligne a un correspondant dans l'autre figure.

5-4.1.1. Exemple 1

Question : D'après la figure 5.6, quelle est la ligne située dans la figure droite qui correspond à la ligne repérée par une flèche dans la figure gauche?

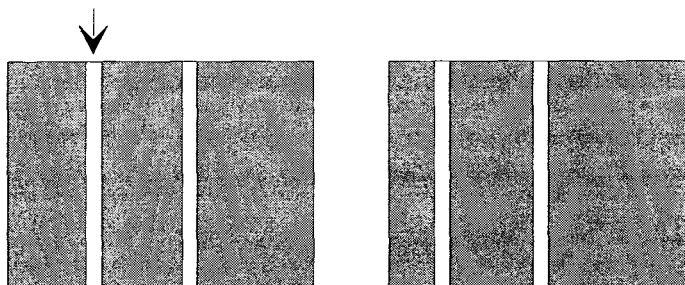


Figure 5.6 : Exemple 1.

Réponse : La première ligne en partant de la gauche de l'image.

Commentaire : Dans cet exemple, la réponse est immédiate pour la simple raison que la scène est simple à analyser. L'homme ayant une connaissance globale de la scène identifie rapidement la ligne correspondante. Cependant, il est difficile de savoir quelle stratégie a été employée pour parvenir à la solution.

5-4.1.2. Exemple 2

Question : D'après la figure 5.7, quelle est la ligne située dans la figure droite qui correspond à la ligne repérée par une flèche dans la figure gauche?

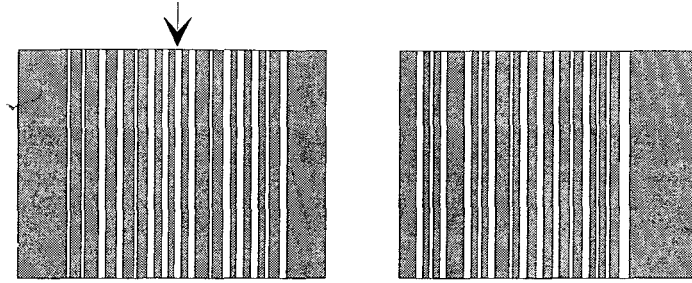


Figure 5.7 : Exemple 2.

Réponse : La 8^{ème} ligne en partant de la gauche de l'image.

Commentaire : Dans cet exemple, l'homme possède toujours une connaissance globale de la scène, mais ses techniques habituelles de déduction sont mises en échec par la complexité de la scène. Afin de répondre à la question, il doit adopter une stratégie différente. La méthode la plus simple consiste à compter les lignes, en partant du début de l'image gauche, pour déterminer la position relative de la ligne repérée par la flèche. La ligne correspondante se situera à la même position dans l'image de droite. Cette méthode revient en fait à appairer séquentiellement la première ligne de l'image gauche à la première ligne de l'image droite, puis les deuxièmes, les troisièmes, et ainsi de suite jusqu'à la ligne considérée.

5-4.1.3. Exemple 3

Question : D'après la figure 5.8, sachant que les lignes marquées d'un symbole ont été mises en correspondance, quelle est la ligne située dans la figure droite qui correspond à la ligne repérée par une flèche dans la figure gauche?

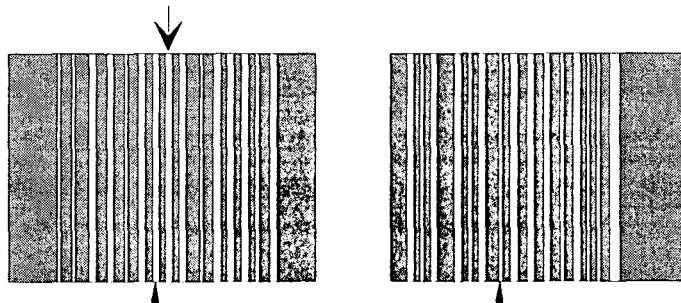


Figure 5.8 : Exemple 3.

Réponse : La ligne située juste à droite de la ligne marquée.

Commentaire : Dans cet exemple, la scène est aussi complexe que dans l'exemple 2. Cependant la réponse est aussi évidente que dans l'exemple 1. En fait la stratégie utilisée est différente de celle employée dans les deux cas précédents. Ici, la solution est déterminée en fonction de l'appariement précédent. On suppose que si les deux lignes ont été appariées, alors les lignes suivantes sont homologues.

5-4.1.4. Interprétations de cette approche

Les idées qui ressortent de ces trois exemples sont les suivantes :

1. L'analyse séquentielle des images permet de diminuer la difficulté de la mise en correspondance. En effet, puisqu'on suppose que les primitives respectent la contrainte d'ordre, les appariements peuvent être réalisés séquentiellement.
2. Un couple de primitives appariées permet de déterminer facilement le couple de primitives suivant. Lorsqu'un couple a été apparié, celui-ci peut servir de référence pour réaliser l'appariement suivant. On peut donc réaliser les appariements de proche en proche, ce qui est compatible avec la contrainte d'ordre. On sous-entend également la contrainte d'unicité. Aucune primitive ne peut être de nouveau appariée puisqu'on se déplace toujours dans le même sens.

5-4.2. Elaboration de la procédure de mise en correspondance

5-4.2.1. Principes de base

A partir des idées du paragraphe précédent, nous développons une procédure de mise en correspondance basée sur les deux principes suivants :

1. L'appariement des primitives s'effectue séquentiellement en explorant les images linéaires du début à la fin.
2. Lorsqu'un couple de primitives est apparié, il sert de base pour trouver l'appariement suivant.

5-4.2.2. Ebauche de la méthode

Pour expliquer la méthode, nous nous aidons du plan de mise en correspondance.

L'idée principale est d'explorer les images séquentiellement du début à la fin pour déterminer l'ensemble des appariements corrects. Pour être plus précis, il s'agit d'analyser les listes de primitives $G(i)$ et $D(j)$ en faisant croître les indices i et j . Graphiquement (Cf. Figure 5.9), cela consiste à aller d'une extrémité à l'autre du plan. La mise en correspondance des primitives peut donc être vue comme la recherche d'un chemin dont chaque étape est constituée d'un noeud. Initialement, la procédure débute dans le coin supérieur gauche du plan. L'objectif est d'atteindre le coin inférieur droit.

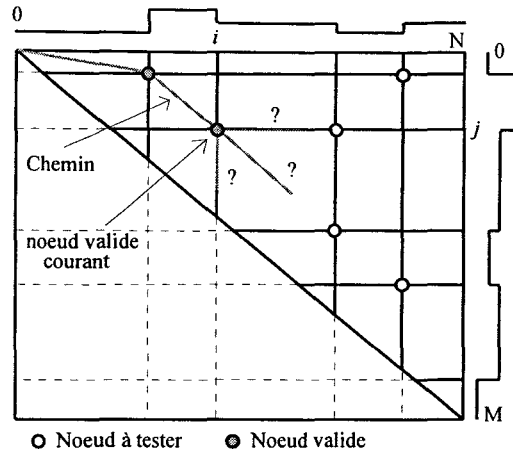


Figure 5.9 : Méthode de mise en correspondance.

Le second principe consiste à utiliser le dernier couple apparié qui forme la paire valide courante $\{G(i), D(j)\}$ pour trouver la paire valide suivante. Graphiquement, il s'agit, en partant du noeud valide courant, de déterminer le chemin qui aboutit au prochain noeud valide.

5-4.2.3. Hypothèse et principe fondamentaux

L'algorithme de mise en correspondance est construit à partir de l'hypothèse fondamentale suivante.

Hypothèse de validité

La paire valide courante $\{G(i), D(j)\}$ servant de référence correspond toujours à un appariement correct.

A partir de cette hypothèse et en utilisant la contrainte d'ordre (Cf. § 5-3.2), on peut donc en déduire le principe suivant.

Principe de mise en correspondance

A partir de la paire valide courante $\{G(i), D(j)\}$, on peut déterminer, s'il existe, le prochain appariement correct défini par la paire $\{G(i+k), D(j+k')\}$ avec $1 \leq k \leq N-i$ et $1 \leq k' \leq M-j$.

5-4.2.4. Réalisation de la procédure de mise en correspondance

La mise en oeuvre de la procédure nécessite deux éléments :

- Réaliser un **module de mise en correspondance élémentaire** basé sur le principe de mise en correspondance.
- Trouver une paire de primitives, satisfaisant **l'hypothèse de validité**, qui puisse servir de condition initiale au module de mise en correspondance élémentaire.

5-4.3. Mesure de similarité

Pour juger de la qualité des différents appariements possibles, nous devons utiliser une mesure de similarité. Nous pourrions, par exemple, calculer pour chaque appariement une mesure de corrélation entre deux fenêtres centrées sur les points de contour considérés. Cependant, nous avons vu au chapitre 3 (Cf. § 3-5.2.2.1) que le choix de fenêtres de taille identique pouvait générer des erreurs d'appréciation dans le cas d'images stéréoscopiques. Nous avons donc décidé d'utiliser notre propre critère de ressemblance.

Soit, par exemple, à évaluer la qualité de l'appariement du point de contour $G(i+k)$ avec le point de contour $D(j+k')$. On considère alors les intervalles définis par $[G(i);G(i+k+\varepsilon)]$ et $[D(j);D(j+k'+\varepsilon')]$, avec $\varepsilon \in \{0,1\}$, $\varepsilon' \in \{0,1\}$ et $\varepsilon+\varepsilon' \neq 0$. Nous verrons un peu plus tard la signification des variables ε et ε' .

5-4.3.1. Normalisation des intervalles

Les deux intervalles ainsi définis vont servir à définir notre mesure de similarité. Mais avant de pouvoir les utiliser, deux opérations doivent être réalisées.

5-4.3.1.1. Recentrage des intervalles

Les deux caméras dont sont issues les images possèdent des amplificateurs de sortie dont les gains sont différents. Il est en effet rare d'avoir deux caméras possédant exactement les mêmes gains de sortie. Cela se traduit par deux images dont les niveaux de gris moyens sont différents. La comparaison est donc délicate. Pour remédier à ce problème, les pixels de chaque intervalle sont ramenés à une valeur moyenne nulle de la façon suivante.

Soit un intervalle de longueur l . On note $h(w)$ les niveaux de gris de l'image correspondant à cet intervalle avec $w = 0, 1, \dots, l-1$. On calcule alors la moyenne m définie par :

$$m = \frac{1}{l} \sum_{z=0}^{z=l-1} h(z).$$

Les niveaux de gris des pixels de l'intervalle sont alors modifiés pour avoir une valeur moyenne nulle. Soit $h_m(w)$ les nouvelles valeurs des niveaux de gris de l'intervalle obtenues par la relation :

$$h_m(w) = h(w) - m \quad \text{avec } w = 0, 1, \dots, l-1.$$

Cette opération est réalisée dans chaque intervalle $[G(i);G(i+k+\varepsilon)]$ et $[D(j);D(j+k'+\varepsilon')]$

5-4.3.1.2. Echantillonnage spatial

La deuxième opération réalisée concerne la taille des intervalles. En effet, ceux-ci ont peu de chance d'être exactement de la même longueur. On ne peut donc pas effectuer de comparaison pixel à pixel. On procède alors de la façon suivante.

Soit l la longueur de l'intervalle le plus petit et L celle du plus grand. On note $h(w)$ les niveaux de gris de l'image correspondant à l'intervalle le plus petit avec $w = 0, 1, \dots, l-1$. De la même manière on note $H(W)$ avec $W = 0, 1, \dots, L-1$ les niveaux de gris de l'autre image, correspondant à l'intervalle le plus grand.

Les pixels de l'intervalle le plus petit sont conservés dans leur totalité. L'opération consiste à prélever l échantillons dans l'intervalle le plus grand. Afin de conserver une "image" représentative de l'intervalle le plus grand, ces échantillons sont choisis régulièrement espacés. On obtient alors l'intervalle échantillonné H_e de longueur l défini de la façon suivante :

$$H_e(w) = H\left(\text{int}\left(\frac{w \cdot L}{l}\right)\right) \quad \text{avec } w = 0, 1, \dots, l-1.$$

L'opérateur $\text{int}(x)$ conserve la partie entière de x afin d'obtenir une valeur discrète.

Grâce à cette opération nous pouvons donc comparer deux régions de tailles différentes, ce qui se produit souvent dans le cas d'images stéréoscopiques à cause des effets de perspectives.

5-4.3.1.3. Intervalles après normalisation

A la suite de ces deux opérations de normalisation, on dispose de deux ensembles de pixels, correspondant aux intervalles gauche et droit $[G(i);G(i+k+\epsilon)]$ et $[D(j);D(j+k'+\epsilon')]$, composés du même nombre de pixels Z et possédant une valeur moyenne nulle. On les note respectivement $H_g(z)$ et $H_d(z)$ avec $z = 1, 2, \dots, Z$.

5-4.3.2. Mesure de similarité

La mesure de similarité entre les pixels des deux intervalles normalisés associés aux deux candidats $G(i+k)$ et $D(j+k')$ est alors déterminée de la façon suivante :

$$S[G(i+k+\epsilon), D(j+k'+\epsilon')] = \frac{1}{Z} \sum_{z=1}^Z |H_g(z) - H_d(z)| \quad \text{Equation 5.4}$$

Comme nous sommes amenés à comparer des mesures faites sur des groupes d'intervalles différents, la mesure de similarité est normalisée par le coefficient Z qui correspond aux

nombres de pixels des intervalles normalisés.

Après cette mesure, la ressemblance entre les deux intervalles est d'autant plus grande que la mesure S est proche de zéro.

5-4.4. Module de mise en correspondance élémentaire

Afin de simplifier l'exposé, nous considérons dans un premier temps que tous les points de contour sont compatibles, c'est à dire qu'ils vérifient la contrainte épipolaire et la contrainte de signe du gradient.

5-4.4.1. Recherche du meilleur appariement

Le problème consiste à appairer le point de contour $G(i+k)$ avec le point de contour $D(j+k')$ en considérant les intervalles définis par $[G(i);G(i+k+\varepsilon)]$ et $[D(j);D(j+k'+\varepsilon')]$, avec $1 \leq k \leq N-i$, $1 \leq k' \leq M-j$, $\varepsilon \in \{0,1\}$, $\varepsilon' \in \{0,1\}$ et $\varepsilon+\varepsilon' \neq 0$.

Nous pourrions tenter de rechercher le meilleur appariement possible parmi toutes les possibilités offertes par l'étendue de l'ensemble des variables. Cette technique demanderait beaucoup de temps, ce qui n'est pas compatible avec notre objectif de détection d'obstacles en temps-réel.

En pratique, nous étudions simplement le premier appariement possible défini par les points de contour $G(i+1)$ et $D(j+1)$. On pourrait se contenter de déterminer si cet appariement est correct ou non. Dans l'affirmative, la paire $\{G(i+1), D(j+1)\}$ deviendrait la paire valide courante et servirait alors de référence pour rechercher l'appariement suivant, comme le stipule le principe de mise en correspondance que nous avons défini. Dans le cas contraire, que se passerait-il? Etant donné que nous n'avons pas réussi à déterminer une nouvelle paire valide, l'algorithme ne peut plus évoluer. Une solution serait alors de considérer la paire $\{G(i+2), D(j+2)\}$. Mais on pourrait se retrouver dans la même situation. A la fin de la procédure, plusieurs primitives de chaque image resteraient sans correspondant et ceci sans explication. Il faut essayer de comprendre pourquoi l'appariement a échoué.

La raison principale de cet échec est que l'une des primitives n'a pas de correspondant dans l'autre image. Celle-ci se trouve alors associée à une primitive ne lui correspondant pas réellement et le test d'appariement répond négativement. Pour éviter d'interrompre l'algorithme, il faut envisager cette possibilité.

5-4.4.2. Tests d'appariement élémentaire

En partant de la paire valide courante $\{G(i), D(j)\}$, on cherche à appairer le contour $G(i+1)$ avec le contour $D(j+1)$. Nous proposons alors de considérer les intervalles définis par $[G(i);G(i+1+\varepsilon)]$ et $[D(j);D(j+1+\varepsilon')]$, avec $\varepsilon \in \{0,1\}$, $\varepsilon' \in \{0,1\}$ et $\varepsilon+\varepsilon' \neq 0$ et d'effectuer une mesure de similarité telle que nous l'avons définie au § 5-4.3.2. Nous sommes donc amenés à calculer les trois mesures de similarité suivantes

- $S_{22} = S[G(i+2), D(j+2)]$
- $S_{12} = S[G(i+1), D(j+2)]$
- $S_{21} = S[G(i+2), D(j+1)]$

Afin de comprendre la signification de ces différentes mesures, nous allons nous aider des schémas de la figure 5.10.

La figure 5.10 (a) représente la partie du plan de mise en correspondance qui nous intéresse. Il s'agit du noeud valide courant $\{G(i), D(j)\}$ à partir duquel nous tentons de déterminer le chemin qui permet d'aboutir au prochain noeud valide. Le candidat possible est le noeud $\{G(i+1), D(j+1)\}$.

Graphiquement, nous représentons une mesure S_{uv} par un rectangle dont les côtés sont les intervalles $[G(i);G(i+u)]$ et $[D(j);D(j+v)]$ (Cf. Figure 5.10 (b) à (d)).

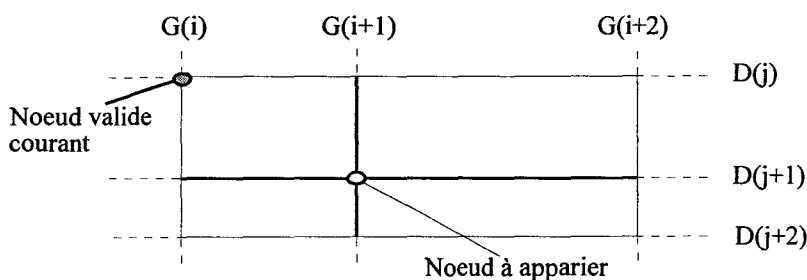


Figure 5.10 (a) : Partie du plan de mise en correspondance.

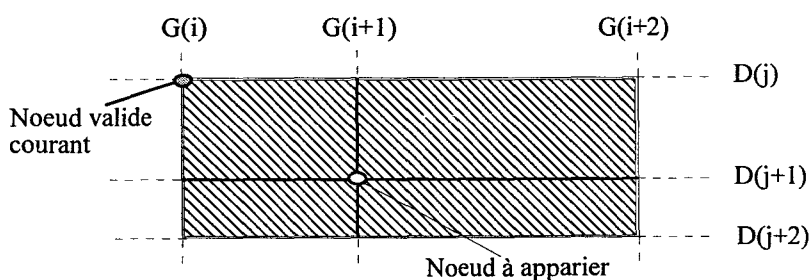


Figure 5.10 (b) : Calcul de S_{22} .

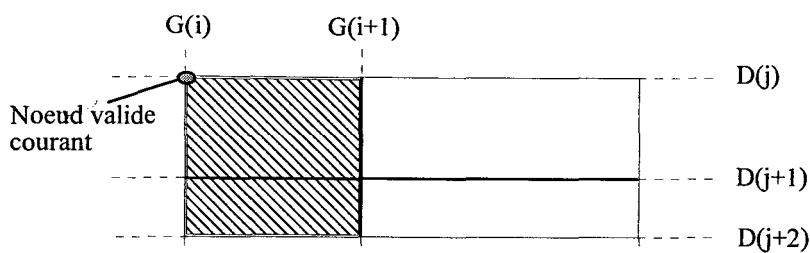
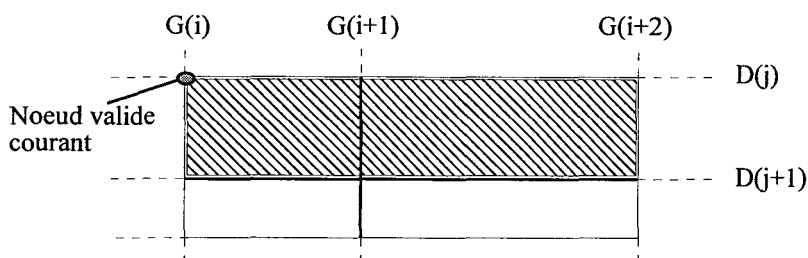

 Figure 5.10 (c) : Calcul de S_{12} .

 Figure 5.10 (d) : Calcul de S_{21} .

Figure 5.10 : Tests de mise en correspondance.

Interprétation de la mesure S_{22}

La mesure S_{22} sert à déterminer si l'appariement entre les points de contours $G(i+1)$ et $D(j+1)$ est correct. On évalue pour cela la mesure de similarité entre les intervalles normalisés définis par $[G(i);G(i+2)]$ et $[D(j);D(j+2)]$ (Cf. Figure 5.10(a)). Nous justifions le choix de ces intervalles de la façon suivante.

Nous avons vu, au chapitre 4 (Cf. § 4-1.3), qu'il existait une dualité entre les points de contour et les régions. Chaque point de contour délimite deux régions de l'image et chaque région est délimitée par deux points de contour. Ainsi le contour $G(i+1)$ délimite les régions $[G(i);G(i+1)]$ et $[G(i+1); G(i+2)]$ et la région $[G(i);G(i+1)]$ est délimitée par les contours $G(i)$ et $G(i+1)$. Puisque nous essayons d'apparier les points de contour, nous pouvons faire la supposition suivante.

Si, dans une image, un contour C délimite les régions $R1$ et $R2$, alors, dans l'autre image, le contour C' correspondant de C délimite les deux régions $R1'$ et $R2'$ telles que la région $R1'$ correspond à $R1$ et $R2'$ correspond à $R2$.

Si on considérait une seule région par image, nous pourrions être amenés à mesurer la ressemblance entre une région $R1$ et une région $R2'$ ne se correspondant pas mais dont la mesure de similarité indiquerait une forte ressemblance. Un mauvais appariement serait alors réalisé. Par contre la probabilité d'avoir deux régions successives $R1$ et $R2$ ressemblant aux

régions successives $R1'$ et $R2'$ mais ne se correspondant pas est beaucoup plus faible. La mesure de similarité sera donc plus fiable si elle s'effectue entre deux régions adjacentes.

Puisque la mesure S_{22} permet d'apparier le contour $G(i+1)$ avec le contour $D(j+1)$, quels résultat fournissent les mesures S_{12} et S_{21} ? La figure 5.11 montre un exemple où le point $G(i+1)$ est le correspondant $D(j+1)$. On constate que les mesures de similarité S_{12} et S_{21} s'effectuent sur des intervalles dissemblables. Les résultats de ces mesures reflètent ces différences. Ainsi, quand S_{22} est la mesure la plus faible, le contour $G(i+1)$ est apparié au contour $D(j+1)$.

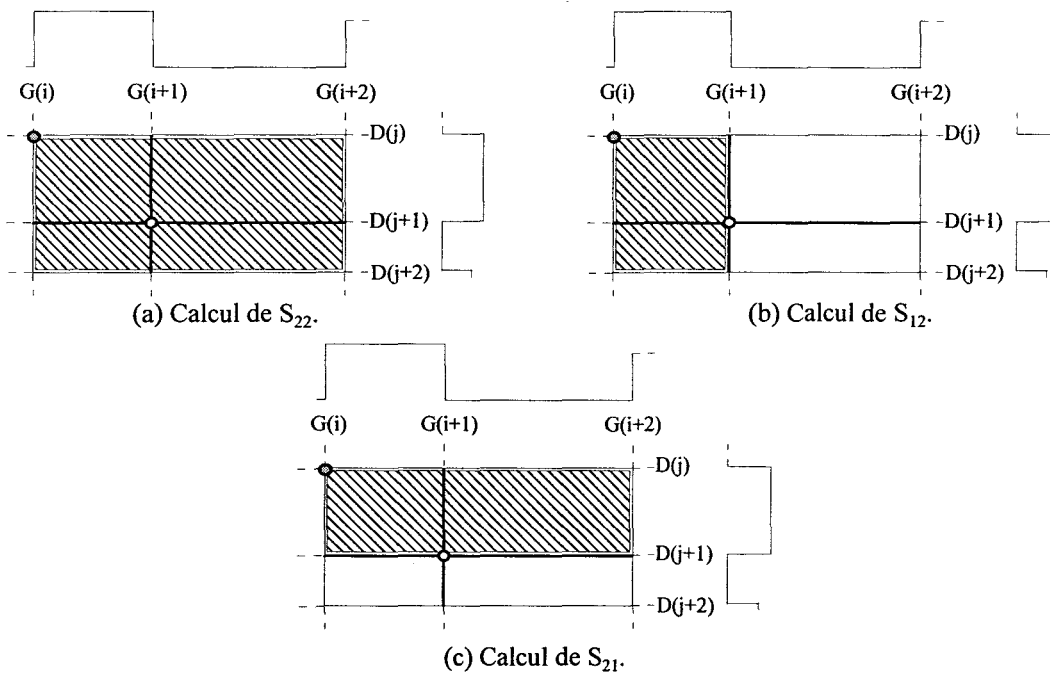


Figure 5.11 : Exemple de situation où $G(i+1)$ est le correspondant de $D(j+1)$.

Il faut apporter une petite précision en ce qui concerne les contours $G(i+2)$ et $D(j+2)$. Ceux-ci servent uniquement à définir la limite droite des intervalles. Comme nous ne tentons pas de les apparier, nous ne vérifions pas leur compatibilité au sens des contraintes épipolaire et de signe du gradient.

Interprétation des mesures S_{12} et S_{21}

Les mesures S_{12} et S_{21} ne correspondent pas à une tentative d'appariement. Elles sont destinées à prendre en compte la présence de primitives sans correspondant.

S_{12} évalue la mesure de similarité entre les intervalles normalisés $[G(i);G(i+1)]$ et $[D(j);D(j+2)]$ afin de juger si le contour $D(j+1)$ n'est pas sans correspondant. La mesure de S_{12} est représentée graphiquement sur la figure 5.10 (c).

S_{21} évalue la mesure de similarité entre les intervalles normalisés $[G(i);G(i+2)]$ et $[D(j);D(j+1)]$ afin de juger si le contour $G(i+1)$ n'est pas sans correspondant. La mesure de S_{21} est représentée graphiquement sur la figure 5.10 (d).

Nous expliquerons le principe de notre raisonnement à partir de la mesure S_{12} représentée sur la figure 5.12. Les hypothèses sont toujours les mêmes. Le contour $G(i)$ a été apparié au contour $D(j)$ et la contrainte d'ordre est respectée. L'hypothèse associée à la mesure S_{12} est que le contour $D(j+1)$ n'a pas de correspondant.

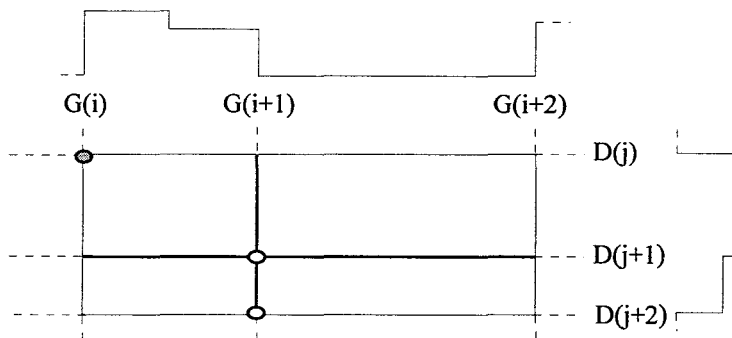


Figure 5.12 : Exemple de situation où $D(j+1)$ n'a pas de correspondant.

Après la paire $\{G(i), D(j)\}$ et la contrainte d'ordre, le contour $G(i+1)$ devrait avoir pour correspondant le contour $D(j+1)$. Or celui-ci n'a pas de correspondant. Cherchons alors le correspondant de $G(i+1)$. Toujours d'après la contrainte d'ordre, celui-ci devrait être $D(j+2)$. Supposons que $D(j+2)$ soit effectivement le correspondant de $G(i+1)$. Les intervalles $[G(i);G(i+1)]$ et $[D(j);D(j+2)]$ représentent alors la même région. La mesure S_{12} doit donc refléter la ressemblance de ces deux régions. Que deviennent alors les mesures S_{22} et S_{21} dans une telle situation? La figure 5.13 nous montre que ces deux mesures font intervenir la région définie par l'intervalle $[G(i+1);G(i+2)]$ qui n'a aucun rapport avec la région $[D(j);D(j+2)]$. Les mesures S_{22} et S_{21} reflètent les différences existant entre les intervalles.

Ainsi, lorsque la mesure S_{12} est plus faible que les mesures S_{22} et S_{21} , le contour $D(j+1)$ n'a pas de correspondant.

Un raisonnement identique peut être appliqué pour la mesure S_{21} . Ainsi, quand la mesure S_{21} est plus faible que les mesures S_{22} et S_{12} , le contour $G(i+1)$ n'a pas de correspondant.

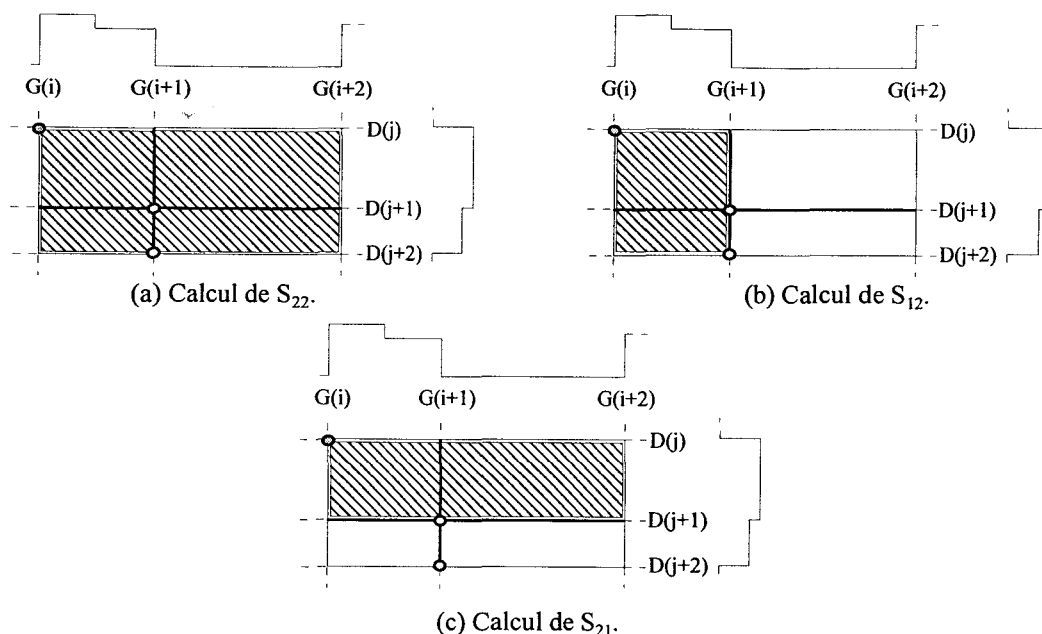


Figure 5.13 : Calcul de S_{12} , S_{21} et S_{22} dans la situation de la figure 5.12.

5-4.4.3. Règles de décision

Les trois mesures S_{22} , S_{12} et S_{21} sont comparées afin de déterminer la mesure la plus faible. En fonction du résultat obtenu, nous appliquons alors l'une des règles suivantes :

1. S_{22} est la mesure la plus faible : Le contour $G(i+1)$ est apparié au contour $D(j+1)$ et l'appariement est mémorisé. La paire valide $\{G(i+1), D(j+1)\}$ devient la paire valide courante. On forme alors la paire $\{G(i+2), D(j+2)\}$ qui devient la paire à tester.
2. S_{12} est la mesure la plus faible : Cela signifie que le point de contour $D(j+1)$ n'a pas de correspondant dans l'image gauche. Il est donc ignoré et remplacé par $D(j+2)$ dans la procédure d'appariement élémentaire. La paire $\{G(i), D(j)\}$ reste la paire valide courante et la paire $\{G(i+1), D(j+2)\}$ devient la paire à tester.
3. S_{21} est la mesure la plus faible : Cela signifie que le point de contour $G(i+1)$ n'a pas de correspondant dans l'image droite. Il est donc ignoré et remplacé par $G(i+2)$ dans la procédure d'appariement élémentaire. La paire $\{G(i), D(j)\}$ reste la paire valide courante et la paire $\{G(i+2), D(j+1)\}$ devient la paire à tester.

Graphiquement, chacune de ces règles correspond à un déplacement élémentaire dans le plan de mise en correspondance (Cf. Figure 5.14). La mesure S_{22} correspond à un déplacement diagonal qui permet d'atteindre le nouveau noeud apparié qui va devenir le nouveau noeud valide courant. Les deux autres mesures S_{12} et S_{21} correspondent respectivement à un

déplacement élémentaire vertical et horizontal. Ces déplacements ne permettent pas d'atteindre un autre noeud valide. Le point atteint constitue une étape intermédiaire à partir de laquelle on peut déterminer le nouveau noeud à tester en incrémentant les indices i ou j d'une unité, ce qui revient à chercher le nouveau noeud à l'extrémité de la diagonale issue de cette étape intermédiaire. Cette représentation permet de respecter la contrainte d'unicité qui s'exprime graphiquement par la présence d'un seul noeud valide au plus sur chaque ligne horizontale et chaque ligne verticale du plan de mise en correspondance.

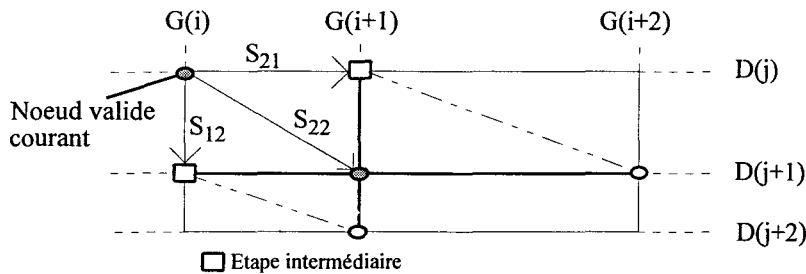


Figure 5.14 : Déplacements élémentaires dans le plan de mise en correspondance.

5-4.4.4. Conclusion sur le module d'appariement élémentaire

Le module d'appariement élémentaire que nous venons de détailler permet de mettre en oeuvre le principe de mise en correspondance que nous avons énoncé précédemment. A partir d'un couple déjà apparié, ce module détermine l'appariement suivant. Mais il ne se contente pas uniquement de vérifier la validité de la paire considérée. Il envisage également que les primitives de cette paire ne soient pas homologues. Quand le cas se présente, l'appariement ne peut être réalisé, le module met alors en jeu des règles d'évolution qui permettent à l'algorithme d'ignorer la primitive sans correspondant en proposant une nouvelle paire à appairier.

5-4.5. Filtrage des paires incompatibles

Pour exposer la méthode utilisée lors de l'appariement des contours nous avons supposé qu'ils étaient compatibles entre eux. Or nous avons vu aux paragraphes 5-3.3 et 5-3.4 que seuls les contours respectant la contrainte épipolaire et la contrainte de signe du gradient pouvaient être appariés.

5-4.5.1. Objectif du filtrage

Lorsque l'algorithme tente d'appairier deux primitives, il vérifie leur compatibilité. Si ces deux primitives le sont, le module de mise en correspondance élémentaire peut être utilisé pour valider l'appariement. Dans le cas contraire, bien que les deux primitives soient

incompatibles, l'algorithme doit continuer sa progression. Le filtrage des primitives permet donc d'éviter des traitements inutiles, mais également de relancer l'algorithme.

Comme précédemment, nous considérons la paire valide courante $\{G(i), D(j)\}$. Il s'agit alors de déterminer si les primitives $G(i+1)$ et $D(j+1)$ sont compatibles. Ce test de compatibilité se décompose en deux étapes successives.

5-4.5.2. Contrainte épipolaire

La contrainte épipolaire, telle que nous l'avons énoncée (Cf. § 5-3.3), impose que les abscisses des primitives homologues dans leurs images respectives vérifient la relation :

$$x_d < x_g.$$

Lors de l'extraction des primitives (Cf. chapitre 4) nous leur avons associé plusieurs attributs (Cf. § 4-6) parmi lesquels la position du point de contour dans l'image va servir à vérifier la contrainte épipolaire.

Quand nous comparons les abscisses x_g de $G(i+1)$ et x_d de $D(j+1)$, deux cas sont possibles.

1. La contrainte épipolaire est respectée. Dans ce cas, nous passons à la vérification de la contrainte de signe du gradient. (Cf. § 5-4.5.3).
2. La contrainte n'est pas respectée. Dans ce cas, les deux primitives ne sont pas compatibles. Il faut donc trouver une autre paire compatible pour permettre à l'algorithme de poursuivre les appariements. On peut, par exemple, éliminer l'une des deux primitives, mais le problème consiste alors à déterminer celle qui doit être supprimée.

Pour trouver une solution à ce problème, il suffit d'analyser la figure 5.15. Sur cette figure, les primitives gauche et droite sont représentées sur deux axes. Chaque axe symbolise un capteur et les positions des primitives sur cet axe correspondent à leurs abscisses sur le capteur.

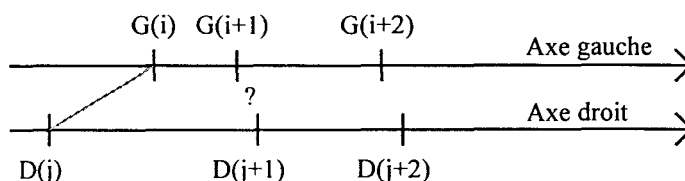


Figure 5.15 : Vérification de la contrainte épipolaire.

La paire valide courante a été appariée. On la représente symboliquement par un lien gris. Les primitives $G(i+1)$ et $D(j+1)$ ne respectant pas la contrainte épipolaire, nous allons étudier les

deux cas suivants.

1. Elimination de $G(i+1)$. On forme la paire de primitives $\{G(i+2), D(j+1)\}$. Leurs abscisses vérifiant la relation épipolaire $x_d < x_g$, nous pouvons tenter de les apparier.
2. Elimination de $D(j+1)$. On forme la paire $\{G(i+1), D(j+2)\}$. Mais l'abscisse de $D(j+2)$ étant supérieure à celle de $D(j+1)$, ces primitives ne vérifient pas la contrainte épipolaire.

Nous pouvons donc en déduire la règle suivante :

Si $G(i+1)$ et $D(j+1)$ ne vérifient pas la contrainte épipolaire, toutes les paires $\{G(i+1), D(j+k')\}$ avec $1 \leq k' \leq M-j$ ne vérifient pas la contrainte épipolaire.

La solution pour trouver la prochaine paire compatible consiste donc à ignorer la primitive $G(i+1)$. Plus généralement il suffit de chercher la première primitive $G(i+k)$ avec $2 \leq k \leq N-i$ vérifiant la contrainte avec la primitive $D(j+1)$. Cependant, si aucune primitive $G(i+k)$ avec $2 \leq k \leq N-i$ n'est trouvée, alors aucune des primitives $G(i+k)$ et $D(j+k')$, avec $1 \leq k \leq N-i$ et $1 \leq k' \leq M-j$, ne possède de correspondant.

5-4.5.3. Contrainte de signe du gradient

L'application de la contrainte de signe du gradient se fait juste après celle de la contrainte épipolaire. Il s'agit de vérifier que les primitives $G(i+1)$ et $D(j+1)$ présentent localement dans chaque image des dérivées de même signe. Cet attribut est mémorisé lors de l'extraction des primitives. Tout comme pour la contrainte épipolaire, deux cas peuvent se présenter :

1. Les dérivées sont de même signe. Dans ce cas, la paire $\{G(i+1), D(j+1)\}$ est compatible. On lance le module de mise en correspondance élémentaire (Cf. § 5-4.3).
2. Les dérivées sont de signes opposés. Dans ce cas, les deux primitives ne sont pas compatibles. Il faut donc trouver une autre paire compatible pour permettre à l'algorithme de poursuivre les appariements.

Ce cas est plus complexe que pour la contrainte épipolaire où une analyse géométrique est suffisante pour trouver la paire suivante. Dans le cas de la contrainte de signe du gradient, la méthode fera appel au module de mise en correspondance élémentaire.

5-4.5.3.1. Analyse du problème

Pour résoudre ce problème nous allons encore une fois nous rattacher au *principe de mise en correspondance* que nous avons énoncé. A partir de la paire valide courante $\{G(i), D(j)\}$, nous

pouvons déterminer la paire suivante. Puisque la paire suivante $\{G(i+1), D(j+1)\}$ n'est pas compatible, nous pouvons supposer que l'une des deux primitives ne possède pas de correspondant. Supposons que $G(i+1)$ (resp. $D(j+1)$) n'ait pas de correspondant. La primitive $D(j+1)$ (resp. $G(i+1)$) devrait alors avoir $G(i+2)$ (resp. $D(j+2)$) comme correspondant.

Le problème consiste alors à vérifier si l'une des paires $\{G(i+1), D(j+2)\}$ et $\{G(i+2), D(j+1)\}$ peut devenir la nouvelle paire valide courante. La première étape consiste à déterminer si ces nouvelles paires sont compatibles.

5-4.5.3.2. Test de compatibilité sur les paires

Si l'une des deux paires constitue un appariement correct, alors elle doit être compatible au sens de la contrainte épipolaire et de la contrainte de signe du gradient. Pour chaque paire $\{G(i+1), D(j+2)\}$ et $\{G(i+2), D(j+1)\}$, on contrôle donc que ces deux contraintes sont vérifiées simultanément.

5-4.5.3.3. Test de mise en correspondance

Pour les paires compatibles, on lance ensuite le module de mise en correspondance élémentaire comme nous l'avons décrit au § 5-4.3. Suivant le cas, celui-ci est utilisé une ou deux fois. Une analyse des résultats est donc nécessaire afin de déterminer les règles qui permettent de faire évoluer l'algorithme.

5-4.5.3.4. Interprétation des résultats

Quatre situations sont envisageables. L'appariement a été tenté :

1. Sur les deux paires $\{G(i+1), D(j+2)\}$ et $\{G(i+2), D(j+1)\}$.
2. Uniquement sur la paire $\{G(i+1), D(j+2)\}$.
3. Uniquement sur la paire $\{G(i+2), D(j+1)\}$.
4. Sur aucune des paires.

Situation 4 :

Ce cas se produit si les deux paires ne sont pas compatibles. On pourrait donc tenter d'élargir l'analyse aux primitives suivantes, mais cette méthode serait incertaine. En effet, le fait de ne pouvoir déterminer simplement la nouvelle paire valide courante indique une grande différence entre les deux images. Aucune primitive correspondante n'ayant été trouvée pour $G(i+1)$ et $D(j+1)$, ces deux primitives sont donc ignorées. La nouvelle paire à appairer devient alors $\{G(i+2), D(j+2)\}$.

Situation 2 et 3 :

Les situations 2 et 3 étant similaires, nous les traiterons en même temps.

Dans ces deux cas, une seule paire est compatible. Le module de mise en correspondance élémentaire est appliqué normalement, mais le résultat n'est pris en compte que si la mesure S_{22} est la plus faible, ce qui signifie que les primitives ont été appariées. Dans ce cas, la paire $\{G(i+1), D(j+2)\}$ (resp. $\{G(i+2), D(j+1)\}$) devient la nouvelle paire valide courante. La paire $\{G(i+2), D(j+3)\}$ (resp. $\{G(i+3), D(j+2)\}$) devient la nouvelle paire à appairer.

Dans le cas où S_{22} n'est pas la mesure la plus faible, aucune primitive correspondante n'ayant été trouvée pour $G(i+1)$ (resp. $D(j+1)$), cette primitive est donc ignorée. La nouvelle paire à appairer devient alors $\{G(i+2), D(j+2)\}$.

Situation 1 :

Comme pour les situations 2 et 3, seule la mesure de S_{22} est prise en compte. Trois cas sont ici envisageables :

A. S_{22} n'est pas la mesure la plus faible pour chaque tentative d'appariement effectuée sur les paires $\{G(i+1), D(j+2)\}$ et $\{G(i+2), D(j+1)\}$.

Aucun appariement n'a donc été réalisé, les primitives $G(i+1)$ et $D(j+1)$ sont donc ignorées. La nouvelle paire à traiter est $\{G(i+2), D(j+2)\}$.

B. S_{22} est la mesure plus faible pour une seule des deux paires.

Un appariement a été réalisé, il est donc pris en compte et devient la nouvelle paire valide courante. Suivant le cas, la paire $\{G(i+1), D(j+2)\}$ (resp. $\{G(i+2), D(j+1)\}$) devient la nouvelle paire valide courante. La paire $\{G(i+2), D(j+3)\}$ (resp. $\{G(i+3), D(j+2)\}$) devient la nouvelle paire à appairer.

C. S_{22} est la mesure la plus faible pour les deux tentatives d'appariement.

Cette situation est ambiguë. Elle indique que les deux paires ont été appariées, ce qui n'est pas logique. La sélection s'effectue encore en fonction de la mesure de similarité. La paire présentant la mesure la plus faible, c'est à dire la ressemblance la plus grande, est conservée. Les règles d'évolution sont les mêmes que dans le cas B.

Dans les différentes situations rencontrées, on peut constater que l'on se contente de vérifier la mesure S_{22} qui témoigne de la qualité de l'appariement des primitives. En effet, l'incompatibilité de la paire $\{G(i+1), D(j+1)\}$ prouve que les régions adjacentes aux

primitives $G(i+1)$ et $D(j+1)$ sont très différentes. Notre objectif est donc de retrouver le plus rapidement possible une nouvelle paire valide qui servira de référence fiable. Le fait de faire évoluer l'algorithme grâce aux mesures S_{12} et S_{21} pourrait poser quelques problèmes d'interprétation, surtout si S_{12} et S_{21} sont respectivement les mesures les plus faibles pour les paires $\{G(i+1), D(j+2)\}$ et $\{G(i+2), D(j+1)\}$. Nous préférons donc être plus prudents en ne considérant que la mesure S_{22} .

5-4.5.4. Conclusion sur le filtrage

Nous venons de voir que le filtrage des paires incompatibles n'élimine pas seulement les paires de primitives ne pouvant être appariées. Le filtrage met également en jeu des règles d'évolution qui permettent à l'algorithme de sélectionner une nouvelle paire compatible à tester, lorsque celui-ci se trouve dans une impasse. Si la contrainte épipolaire utilise uniquement une analyse géométrique, la contrainte de signe du gradient doit faire appel au module de mise en correspondance élémentaire afin de sélectionner la paire permettant de relancer l'algorithme.

5-4.6. Condition initiale et condition d'arrêt de l'algorithme

5-4.6.1. Condition initiale

Nous avons vu que notre algorithme utilise le principe de mise en correspondance défini au § 5-4.2.3. L'appariement est donc réalisé en fonction du dernier couple apparié. Cependant, pour débiter, l'algorithme a besoin d'un couple déjà apparié se trouvant de préférence le plus à gauche dans l'image, pour commencer l'exploration séquentielle des listes de primitives G et D . Or nous ne connaissons aucun couple satisfaisant ce critère et nous ne pouvons en déterminer aucun sans procédure de mise en correspondance. Nous avons donc choisi d'utiliser les premiers pixels de chaque image comme couple de base. Ces pixels, considérés comme des primitives, sont repérés par l'indice zéro. Les primitives $G(0)$ et $D(0)$ forment alors la première paire valide $\{G(0), D(0)\}$. Le principe de l'algorithme consiste à utiliser ce couple initial, qui n'a pas de réalité concrète, pour construire tous les appariements successifs.

5-4.6.2. Condition d'arrêt

La condition d'arrêt est très simple. Lorsque l'une des primitives $G(N)$ ou $D(M)$ fait partie de la nouvelle paire à appairer, la procédure de mise en correspondance s'arrête. Ces primitives correspondent en fait aux derniers pixels des images gauche et droite. En les considérant comme des points de contour et en les insérant dans les listes G et D , on sait quand toutes les

primitives ont été testées. Il faut noter que $G(N)$ et $D(M)$ permettent également de définir les intervalles $[G(N-2);G(N)]$ et $[D(M-2);D(M)]$ si l'algorithme tente d'apparier la paire $\{G(N-1), D(M-1)\}$ à partir de la paire valide $\{G(N-2), D(M-2)\}$.

5-4.7. Résultats de la mise en correspondance séquentielle

Cette procédure est appliquée successivement à chaque ligne de la séquence stéréoscopique. Elle fournit une liste constituée des paires de primitives mises en correspondance par l'algorithme.

5-4.7.1. Piéton évoluant devant le stéréoscope

La figure 5.16 montre le résultat obtenu sur la séquence stéréoscopique "piéton", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.1).

Chaque paire de primitives appariées permet, grâce aux formules de triangulation (Cf. § 5-7), de déterminer la distance et la position d'un point de la scène réelle. Les résultats représentés sur cette image s'interprètent de la manière suivante :

- L'axe horizontal représente l'abscisse dans la scène. Un point se trouvant sur le bord gauche de l'image est situé à 5 m à gauche de l'axe optique. Un point se trouvant sur le bord droit de l'image est situé à 5 m à droite de ce même axe.
- L'axe vertical représente le temps, celui-ci défilant du haut vers le bas de l'image.
- La distance est représentée par le niveau de gris. Celui-ci est d'autant plus clair que la distance par rapport au stéréoscope linéaire est grande.

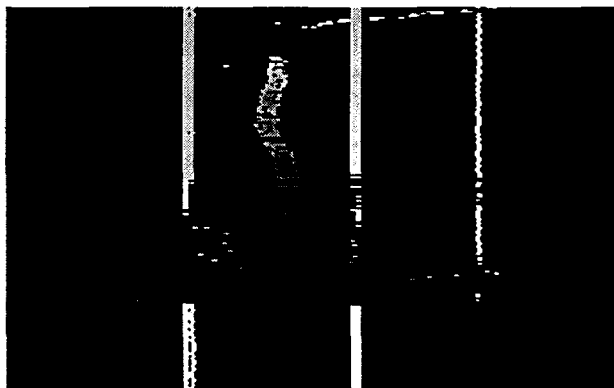


Figure 5.16 : Résultat obtenu sur la séquence "piéton".

On constate sur cette figure que les résultats obtenus sont très bons. Les points de contour des bandes blanches sont correctement appariés et leur détection est stable au cours du temps. En

effet, les positions et distances calculées sont constantes. Les niveaux de gris clairs indiquent que les bandes sont éloignées du stéréoscope. En ce qui concerne le piéton, les niveaux de gris variant du clair au sombre signalent bien que celui-ci se rapproche du stéréoscope. Le mouvement de l'obstacle peut donc être facilement défini. La transition entre la chaussée et la zone d'ombre à droite de la figure est également bien détectée.

On remarque cependant la présence de mauvais appariements lorsque des occlusions se produisent, c'est à dire lorsque le piéton cache l'une des bandes blanches à l'une des caméras. Ces appariements erronés sont provoqués par la mise en correspondance des points de contour d'une des bandes avec ceux du piéton. Mais ils sont peu nombreux ce qui montre l'efficacité de notre mesure de ressemblance.

La dernière observation que nous pouvons faire est l'absence de plusieurs appariements. Sur certaines images, le contour droit de la bande gauche n'a pas été apparié. Or, on remarque sur ces mêmes images que le piéton et la bande de gauche n'ont pas été détectés. Pourtant les points de contour des bandes et du piéton ont bien été extraits de ces images. Ces appariements manqués se produisent lorsque la mesure S_{22} n'est pas la mesure la plus faible. Dans ce cas le couple testé n'est pas retenu et l'algorithme cherche une nouvelle paire à appairier. Les couples suivants n'étant pas compatibles, l'algorithme ne réussit pas à les appairier, et laisse apparaître dans la scène reconstituée des "trous" correspondant à des appariements manqués. Ce type d'erreur se produit dans des zones complexes de l'image où les régions voisines sont très ressemblantes. On remarquera cependant que l'algorithme retrouve parfois des couples qu'il parvient à mettre en correspondance ce qui explique la présence de la partie droite de la bande gauche sur certaines images.

5-4.7.2. Route dégagée et passage d'une sortie d'autoroute

La figure 5.17 montre le résultat obtenu sur la séquence stéréoscopique "sortie d'autoroute", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.2).

Sur cette figure, un point se trouvant sur le bord gauche de l'image est situé à 9 m à gauche de l'axe optique. Un point se trouvant sur le bord droit de l'image est situé à 9 m à droite de ce même axe.

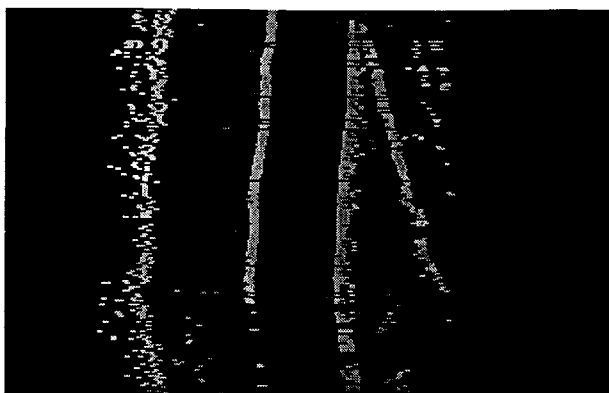


Figure 5.17 : Résultat obtenu sur la séquence "sortie d'autoroute".

Les bandes blanches sont les seuls éléments présents dans cette séquence. Elles sont correctement détectées. La bande centrale et celle de gauche présentent une bonne stabilité dans le temps, par contre la bande de gauche n'apparaît pas nettement. Les nombreux points présents sur sa gauche sont dus aux herbes situées sur le bord de la route. Il faut cependant noter que les herbes sont correctement détectées puisqu'elles se trouvent à la même distance que la ligne.

Quelques faux appariements apparaissent sur la voie de gauche à la fin de la séquence ainsi que sur la voie de droite. Ceux-ci pourront être facilement éliminés car ils sont détectés de façon irrégulière.

Comme dans l'exemple précédent, on peut observer de temps à autre des appariements manqués qui se manifestent sous la forme de discontinuités dans la bande centrale et la bande de droite.

5-5. Mise en correspondance bidirectionnelle

5-5.1. Inconvénient de l'algorithme séquentiel de mise en correspondance

La méthode que nous avons décrite au § 5-4 donne de bons résultats dans le cas de séquences stéréoscopiques simples. Lorsque les images deviennent complexes, des erreurs d'appariement ou des mises en correspondance manquées peuvent se produire. En cas d'erreurs, celles-ci peuvent se répercuter sur le reste de l'analyse.

5-5.2. Analyse bidirectionnelle

Si on se réfère de nouveau à l'exemple 2 (Cf. § 5-4.1), la résolution du problème peut se faire également en partant de la fin de l'image et dans certaines situations, un tel choix permet de contourner une difficulté. En effet, si l'analyse est trop complexe en partant du début,

pourquoi ne pas la tenter en partant de la fin de l'image? Donc pour limiter les risques d'erreurs, notre procédure de mise en correspondance séquentielle est exécutée deux fois. La première fois du début vers la fin de l'image, c'est ce que nous appelons *l'analyse progressive*, la seconde, de la fin au début, il s'agit de *l'analyse rétrograde*. Une comparaison des résultats permet ensuite d'éliminer certaines erreurs possibles.

5-5.3. Analyse rétrograde

Lors de cette analyse, le principe de la procédure de mise en correspondance est identique. Mais la différence essentielle vient du sens de parcours des listes de primitives G et D. Celles-ci sont explorées en sens inverse respectivement de l'indice N et M à l'indice zéro. Pour réaliser l'analyse rétrograde, trois modifications indispensables sont à apporter.

5-5.3.1. Condition initiale

Dans l'analyse rétrograde, le dernier pixel de chaque image, considéré comme un point de contour va servir à former la paire initiale. Celle-ci est définie par la paire $\{G(N), D(M)\}$ qui constitue le point de départ pour construire les appariements successifs.

5-5.3.2. Condition d'arrêt

L'analyse rétrograde s'arrête quand l'une des primitives $G(0)$ ou $D(0)$ fait partie de la nouvelle paire à appairer, c'est à dire quand les premiers pixels de l'image sont atteints.

5-5.3.3. Contrainte épipolaire

L'application de la contrainte épipolaire suit le même principe que celui de l'analyse progressive. La seule différence vient de la stratégie à suivre pour rechercher la paire compatible suivante. Un raisonnement similaire à celui donné au § 5-4.5.2, nous mène au résultat suivant.

Soient la paire valide courante $\{G(i), D(j)\}$ et la paire $\{G(i-1), D(j-1)\}$ ne respectant pas la contrainte épipolaire. Pour trouver la prochaine paire compatible, la solution consiste donc à ignorer la primitive $D(j-1)$. Plus généralement il suffit de chercher la première primitive $D(j-k')$ avec $2 \leq k' \leq j$ vérifiant la contrainte avec la primitive $G(i-1)$.

5-5.4. Comparaison des résultats

Chaque analyse fournit une liste de paires de primitives mises en correspondance. On les appelle liste progressive et liste rétrograde. Chaque paire est caractérisée par l'abscisse de la primitive gauche sur le capteur gauche, l'abscisse de la primitive droite sur le capteur droit et, la valeur de la mesure S_{22} obtenue lors de l'appariement. Les couples sont rangés dans l'ordre où ils ont été appariés. Les contours formant le premier couple de la liste progressive possèdent les abscisses les plus petites, tandis que ceux du dernier couple possèdent les abscisses les plus grandes. Pour la liste rétrograde, l'ordre est inversé. Pour optimiser les traitements, la liste progressive est parcourue du début à la fin tandis que la liste rétrograde est parcourue de la fin au début. Ainsi la comparaison des listes se fait séquentiellement sur des couples de primitives ordonnés selon leurs abscisses croissantes. Pour construire la liste définitive des appariements, les deux listes sont comparées et, ne sont conservés que les couples respectant les contraintes d'ordre et d'unicité. La comparaison nous amène à traiter trois types de possibilités :

1. Une paire $\{G(i), D(j)\}$ apparaît dans les deux listes.

La paire a été appariée deux fois, elle est donc rangée dans la liste définitive. On vérifie néanmoins que les abscisses des primitives gauche et droite sont bien supérieures à celles du couple précédemment rangé. L'expérience nous a montré que c'est toujours le cas.

2. Une primitive $G(i)$ (resp. $D(j)$) est appariée à une primitive différente dans chaque liste.

La contrainte d'unicité nous empêche de conserver ces deux appariements. Un choix doit donc être fait. La première étape consiste à vérifier que la contrainte d'ordre est respectée avec le couple précédemment stocké dans la liste définitive. Seul le couple respectant cette contrainte est conservé. Si deux couples vérifient la contrainte, c'est celui qui présente la mesure de similarité la plus faible qui est gardé.

3. Une paire $\{G(i), D(j)\}$ n'est présente que dans une seule liste.

Ce cas indique que la paire a été appariée dans une seule des analyses. Si la paire vérifie la contrainte d'ordre elle est rangée dans la liste définitive.

5-5.5. Résultats de la mise en correspondance bidirectionnelle

Cette procédure est appliquée successivement à chaque ligne de la séquence stéréoscopique. Comme précédemment, elle fournit une liste constituée des paires mises en correspondance pendant l'algorithme.

5-5.5.1. *Piéton évoluant devant le stéréoscope*

La figure 5.18 montre le résultat obtenu sur la séquence stéréoscopique "piéton", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.1).

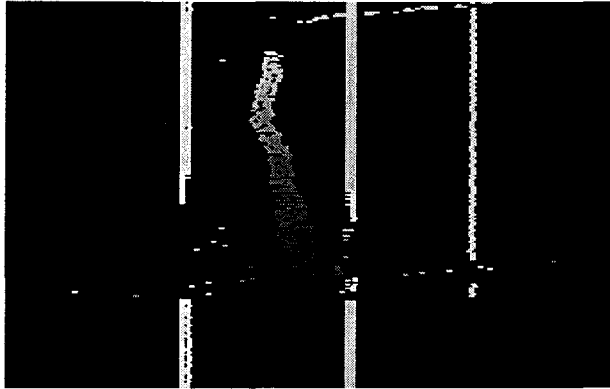


Figure 5.18 : Résultat obtenu sur la séquence "piéton".

Celui-ci est meilleur que le résultat obtenu par l'algorithme séquentiel de mise en correspondance (Cf. Figure 5.16). Les faux appariements qui apparaissent uniquement lorsque des occlusions se produisent sont moins nombreux. De même, on constate que l'analyse rétrograde a réussi là où l'analyse progressive a échoué. La bande droite et le piéton sont mieux détectés avec la méthode bidirectionnelle. Cependant on remarque que des "trous" apparaissent encore dans la scène reconstruite. En effet, le principe d'analyse séquentiel se heurte à une zone complexe constituée par le piéton et ne parvient pas à en appairier les points de contour. On peut cependant noter que l'algorithme n'engendre pas de faux appariements.

5-5.5.2. *Route dégagée et passage d'une sortie d'autoroute*

La figure 5.19 montre le résultat obtenu sur la séquence stéréoscopique "sortie d'autoroute", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.2).

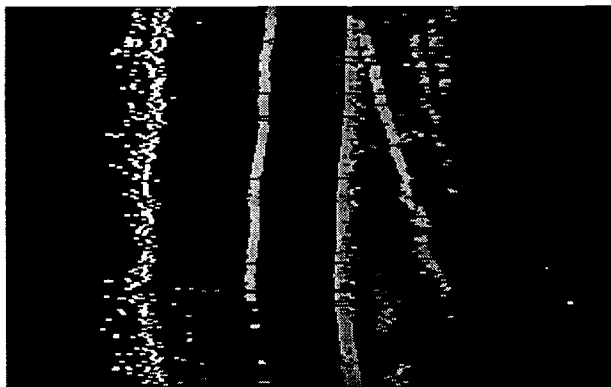


Figure 5.19 : Résultat obtenu sur la séquence "sortie d'autoroute".

On ne remarque pas de véritable amélioration sur cette figure. Il est vrai que cette séquence est très simple et l'analyse progressive suffit pour réaliser les appariements. On constate cependant la disparition de quelques "trous" dans la bande centrale et la bande de droite.

5-5.6. Conclusion

Cette méthode de mise en correspondance bidirectionnelle basée sur l'algorithme séquentiel de mise en correspondance décrit au § 5-4 est intéressante car elle permet une vérification des résultats obtenus. La mise en correspondance réalisée de deux points de vue différents donne ainsi la possibilité de confirmer certains appariements, mais également celle d'en éliminer d'autres plus ambigus.

Cependant, cette méthode ne nous a pas semblé assez robuste. Nous lui avons donc apporté certaines modifications que nous allons détailler dans le prochain paragraphe.

5-6. Algorithme hiérarchique de mise en correspondance

5-6.1. Approche intuitive

5-6.1.1. Première considération

Nous allons utiliser un exemple comme au § 5-4.1. Cependant, les images stéréoscopiques sont plus proches de la réalité. Les primitives n'ont pas obligatoirement de correspondant dans l'autre image, c'est à dire que certaines primitives peuvent se trouver en dehors du secteur de stéréovision. Elles peuvent donc être présentes au début de l'image gauche mais pas dans l'image droite ou à la fin de l'image droite mais pas dans l'image gauche. On considère par contre qu'il n'y a pas d'occlusions celles-ci sont traitées au chapitre suivant (Cf. § 6-5.4).

Question : D'après la figure 5.20, quelle est la ligne située dans la figure droite qui correspond à la ligne repérée par une flèche dans la figure gauche?

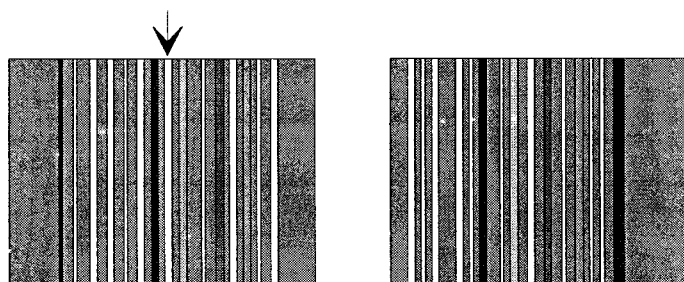


Figure 5.20 : Exemple tenant compte de la différence des primitives.

Réponse : La ligne située juste après la ligne gris foncée en partant de la gauche de l'image.

Commentaire : Les images étant différentes, la méthode consiste à chercher un point commun sur lequel on puisse se baser pour trouver la réponse. On peut par exemple prendre en compte les différences pouvant exister entre les lignes. Ici, les différences sont essentiellement dues aux niveaux de gris. Il suffit donc de trouver dans chaque image des lignes possédant des niveaux de gris communs pour obtenir des lignes de référence. A partir de ces lignes de référence, on pourra ensuite aisément en déduire les primitives correspondantes par une analyse de proche en proche comme dans l'exemple 2 (Cf. § 5-4.1.2).

5-6.1.2. Deuxième considération

La figure 5.21 nous montre la difficulté d'une analyse séquentielle quand le nombre de noeuds possibles est très important. Plus le nombre de possibilités augmente, plus la recherche d'un chemin dans le plan de mise en correspondance devient difficile. La première conséquence est que les probabilités d'erreurs sont plus importantes.

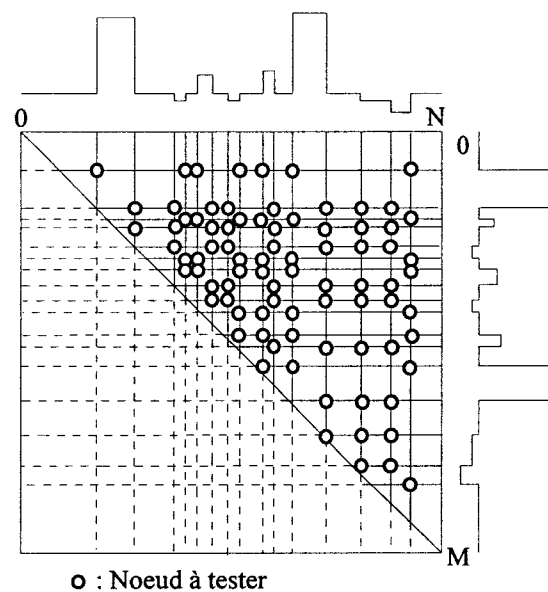


Figure 5.21 : Exemple de plan de mise en correspondance complexe.

L'idée de trouver des primitives de référence est intéressante car elles constituent des étapes clés dans le plan de mise en correspondance rendant ainsi la recherche du chemin moins aléatoire dans des cas complexes comme celui de la figure 5.21. A partir de ces étapes on peut de nouveau utiliser l'analyse séquentielle puisque les risques d'erreurs sont diminués.

5-6.1.3. Interprétation

Jusqu'à présent nous avons considéré des exemples où toutes les primitives sont présentes dans le secteur de stéréovision. En réalité, la géométrie du capteur fait que la partie de la scène présente sur le début de l'image gauche n'est pas visible sur l'image droite. Symétriquement,

la partie de la scène présente sur la fin de l'image droite n'est pas visible sur l'image gauche (Cf. § 3-5.1.1). Or, si ces parties contiennent des primitives, celles-ci ne possèdent pas de correspondants dans l'autre image. L'analyse bidirectionnelle ne tient pas compte de cette situation puisque les premiers et derniers pixels de chaque image servent de conditions initiales et peuvent entraîner des erreurs de mise en correspondance lors de l'analyse progressive mais également lors de l'analyse rétrograde. Or, dès qu'une erreur apparaît, elle risque d'affecter les appariements suivants. Les erreurs sont d'autant plus importantes que le plan de mise en correspondance est complexe à analyser.

L'idée est donc de trouver des primitives de référence afin de diminuer les risques d'erreurs d'appariements.

5-6.2. Mise au point de la procédure de mise en correspondance

5-6.2.1. Principe de base

L'idée qui ressort de l'exemple précédent est de déterminer des couples de primitives de référence à partir desquels on effectuera la mise en correspondance des autres primitives. Cette idée implique donc la recherche, dans chaque image, de primitives plus significatives que les autres. Or, la notion de primitive significative sous-entend le classement des primitives en fonction d'un critère qui permette ensuite de les sélectionner mais également que les classements faits dans chaque image soient compatibles entre eux, c'est à dire que les primitives homologues soient classées dans le même ordre. Si un critère fournit des classements compatibles pourquoi ne pas appairer les primitives hiérarchiquement c'est à dire en commençant par les plus significatives. De plus si on utilise l'idée du paragraphe 5-6.1.2 qui est celle du principe de mise en correspondance (Cf. § 5-4.2.3), les primitives significatives appariées servent de base pour rechercher les appariements de primitives un peu moins significatives.

5-6.2.2. Description de la méthode

Pour expliquer le principe de la méthode, nous ferons de nouveau appel au plan de mise en correspondance.

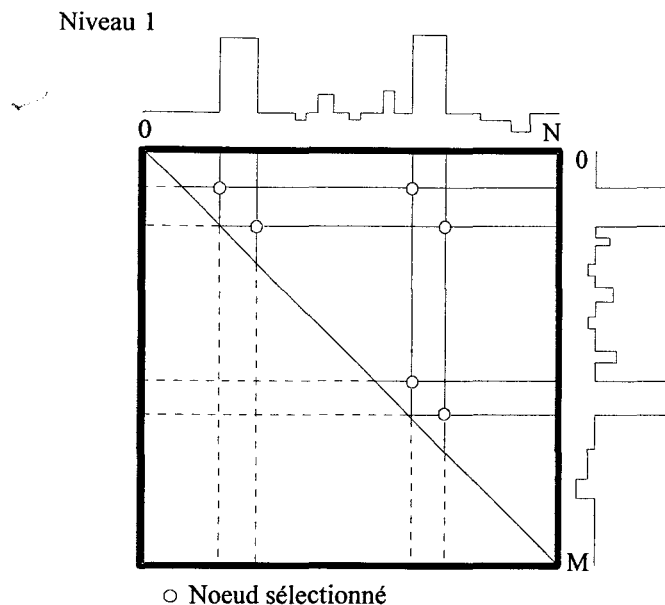


Figure 5.22 : Niveau 1 de la mise en correspondance hiérarchique.

L'idée principale est d'apparier hiérarchiquement les primitives. Nous utilisons pour cela la notion de *niveau de recherche*. Pour un niveau de recherche donné, on sélectionne dans chaque image un certain nombre de primitives. La figure 5.22 montre le niveau 1 de recherche où 4 primitives ont été sélectionnées dans chaque image. On constate que le plan à analyser est beaucoup plus simple que celui de la figure 5.21. Puisque le plan à analyser est plus simple, nous pouvons pour obtenir des couples de référence, appliquer la mise en correspondance bidirectionnelle dans, ce que nous appelons, le plan de recherche représenté graphiquement par un rectangle dont les limites sont dessinées avec un trait épais. On remarque qu'au niveau 1 le plan de recherche est confondu avec le plan de mise en correspondance.

Lorsque la mise en correspondance bidirectionnelle est terminée, elle fournit une liste de paires appariées. Cette liste sert de base pour mettre en correspondance les primitives au niveau suivant. La figure 5.23 montre le niveau 2 de recherche. Deux couples successifs appariés au niveau 1 servent à définir un nouveau plan de recherche. Dans notre exemple, nous pouvons voir 5 nouveaux plans de recherche définis à partir des 4 noeuds valides. Pour permettre une exploration complète, les extrémités du plan de mise de recherche au niveau 1 sont réutilisées au niveau 2 afin de définir aisément les plans de recherche situés aux extrémités du plan de mise en correspondance.

Niveau 2

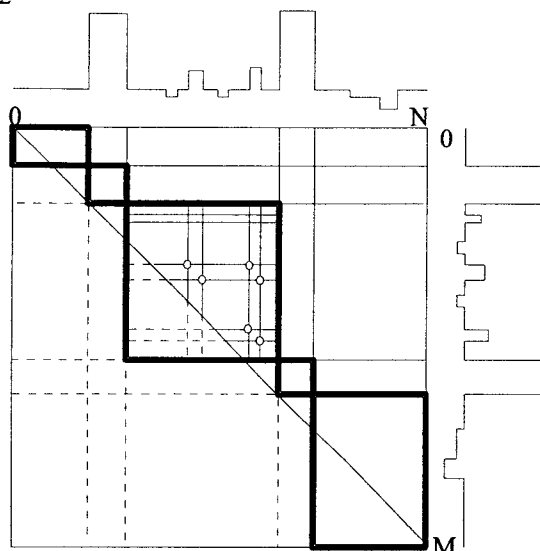


Figure 5.23 : Niveau 2 de mise en correspondance hiérarchique.

Au niveau 2 de recherche, la technique consiste à sélectionner dans chaque plan de recherche des primitives moins significatives à appairer. Dans le cas de la figure 5.23, un seul plan de recherche contient des noeuds à traiter. Dans un cas plus général, la mise en correspondance bidirectionnelle est alors appliquée dans chaque plan de recherche. Les différents résultats obtenus permettent alors de définir de nouveaux plans de recherche plus petits, dans lesquels la mise en correspondance bidirectionnelle est également appliquée. Pour l'optimiser, la procédure est implantée de façon récursive. Nous allons détailler cette implantation dans les paragraphes suivants.

5-6.3. Définitions supplémentaires

Pour exposer la façon dont a été implantée la méthode, nous allons tout d'abord donner quelques définitions et notations supplémentaires.

5-6.3.1. Définitions

Un plan de recherche est défini à partir de deux points, son extrémité supérieure gauche et son extrémité inférieure droite. Seules les paires de primitives appariées à un certain niveau de recherche peuvent devenir les extrémités d'un plan de recherche au niveau suivant. Une paire $\{G(i), D(j)\}$ formant une extrémité est notée $(G(i), D(j))$. Un plan de recherche est donc noté $PR[(O_g, O_d); (F_g, F_d)]$, où (O_g, O_d) et (F_g, F_d) sont respectivement l'extrémité supérieure gauche, appelée *origine*, et inférieure droite, appelée *fin*, du plan de recherche.

Le niveau n de mise en correspondance utilise les résultats du niveau précédent, c'est à dire la liste des paires appariées. Nous notons $\{ \{ C_G^n(k), C_D^n(k) \}, k = 0, 1, \dots, X^n \}$ la liste obtenue au

niveau n , où $C_G^n(k)$ est la primitive gauche appariée à la primitive droite $C_D^n(k)$ et, X^{n+1} représente le nombre de paires appariées. Cette liste sert à définir les plans de recherche du niveau $n+1$.

La liste de primitives gauches sélectionnées au niveau de recherche n est notée $G^n(i)$ avec $i = 0, 1, \dots, N^n$, où N^n est le nombre de primitives gauches sélectionnées au niveau n . De la même manière la liste de primitives droites est notée $D^n(j)$ avec $j = 0, 1, \dots, M^n$, où M^n est le nombre de primitives droites sélectionnées au niveau n .

5-6.3.2. Lien avec l'algorithme de mise en correspondance bidirectionnelle

Au paragraphe 5-5, nous avons décrit l'algorithme de mise en correspondance bidirectionnelle. Celui-ci était appliqué sur l'ensemble des primitives $G(i)$ et $D(j)$ dans le plan de mise en correspondance. Le point de départ était la paire $\{0,0\}$ et le point d'arrivée la paire $\{N,M\}$ pour l'analyse progressive et inversement pour l'analyse rétrograde. Les paires $\{0,0\}$ et $\{N,M\}$ étant considérées comme valides par l'algorithme, nous pouvons les assimiler aux extrémités et définir alors le plan de recherche $PR[(0, 0); (N, M)]$. L'algorithme de mise en correspondance bidirectionnelle peut donc être vu comme un algorithme hiérarchique particulier avec un seul niveau de recherche où toutes les primitives sont prises en compte. Celui-ci s'applique dans un plan de recherche particulier qui est le plan de mise en correspondance.

L'algorithme de mise en correspondance bidirectionnelle pourra donc être paramétré afin de s'appliquer uniquement dans un plan de recherche limité que nous notons $PR[(O_g, O_d); (F_g, F_d)]$. En choisissant précisément les extrémités (O_g, O_d) et (F_g, F_d) , celui-ci pourra être utilisé dans une zone particulière du plan de mise en correspondance avec un nombre limité de primitives.

5-6.4. Algorithme récursif de mise en correspondance hiérarchique

5-6.4.1. Principe général de notre algorithme

La mise en correspondance hiérarchique a été implantée selon un algorithme récursif afin de traiter tous les plans de recherche quelque soit leur nombre.

La stratégie employée pour rechercher les appariements dans le plan de mise en correspondance peut s'apparenter aux stratégies de recherche avec graphe utilisées en intelligence artificielle. Nous allons donc utiliser la terminologie associée au graphe.

5-6.4.1.1. Définitions

Le graphe est constitué d'un ensemble de *noeuds*. Certaines paires de noeuds sont connectées par des *arcs* qui sont orientés d'un membre d'une paire vers l'autre. Si un arc est orienté d'un noeud n_i vers un noeud n_j , on dit alors que le noeud n_j est un *successeur* du noeud n_i et le noeud n_i est un *parent* du noeud n_j . Un noeud ne possédant pas de parent est appelé *racine*. Un noeud n'ayant pas de successeur est appelé *feuille*.

Pour différencier le terme *noeud* associé au graphe et le terme *noeud* que nous avons défini comme la représentation graphique d'une paire de primitives, nous emploierons le mot anglais *node* lorsque nous ferons allusion au graphe.

5-6.4.1.2. Application à notre algorithme récursif

Un exemple de graphe expliquant le fonctionnement de notre algorithme récursif est représenté sur la figure 5.24. Les nodes sont les plans de recherche à l'intérieur desquels la procédure de mise en correspondance bidirectionnelle est appliquée aux primitives sélectionnées (Cf. § 5-6.4.2). Les résultats obtenus lors de la mise en correspondance servent ensuite à générer des nodes successeurs qui sont reliés au node parent par des arcs.

Dans notre exemple, au niveau 0 le node *a* confondu avec le plan de mise en correspondance est défini par deux extrémités que l'on a notées de façon synthétique $C^0(0)$ et $C^0(4)$ pour plus de clarté. La mise en correspondance appliquée dans le node *a* a permis d'apparier 3 couples de primitives notés $C^0(1)$, $C^0(2)$, $C^0(3)$. Pour permettre de générer les nodes successeurs, l'algorithme utilise les extrémités du node *a*. Ainsi les paires $C^0(0)$, $C^0(1)$, $C^0(2)$, $C^0(3)$, $C^0(4)$ permettent de créer les nodes *b*, *e*, *f*, *m*.

5-6.4.1.3. Exploration du graphe par l'algorithme

Pour trouver l'ensemble des appariements valides, l'algorithme va développer les nodes "*en profondeur d'abord*". Cette méthode consiste à traiter d'abord les nodes de niveaux supérieurs avant de traiter les nodes de même niveau.

Ainsi, lorsqu'un node du niveau *n* génère plusieurs successeurs au niveau *n+1*, l'algorithme cherche d'abord à étendre les recherches au niveau supérieur *n+2*. Pour cela nous donnons toujours priorité au node successeur le plus à gauche, c'est à dire au node successeur possédant la même *origine* que le node parent. Le traitement consiste alors à appliquer à ce node le module d'appariement élémentaire hiérarchique (Cf. § 5-6.4.3). Si ce module fournit

de nouvelles paires appariées, celles-ci généreront au niveau $n+2$ de nouveaux nodes successeurs et le processus recommence de la même manière. Dans le cas contraire on considère que le node de même niveau (ici $n+1$) possède une extrémité commune avec le node venant d'être traité. Lorsque tous les nodes d'un niveau ont été traités, on remonte au niveau n pour examiner les nodes existants de ce niveau. Le processus cesse d'engendrer de nouveaux nodes quand l'une des deux conditions d'arrêt est rencontrée (Cf. § 5-6.4.5).

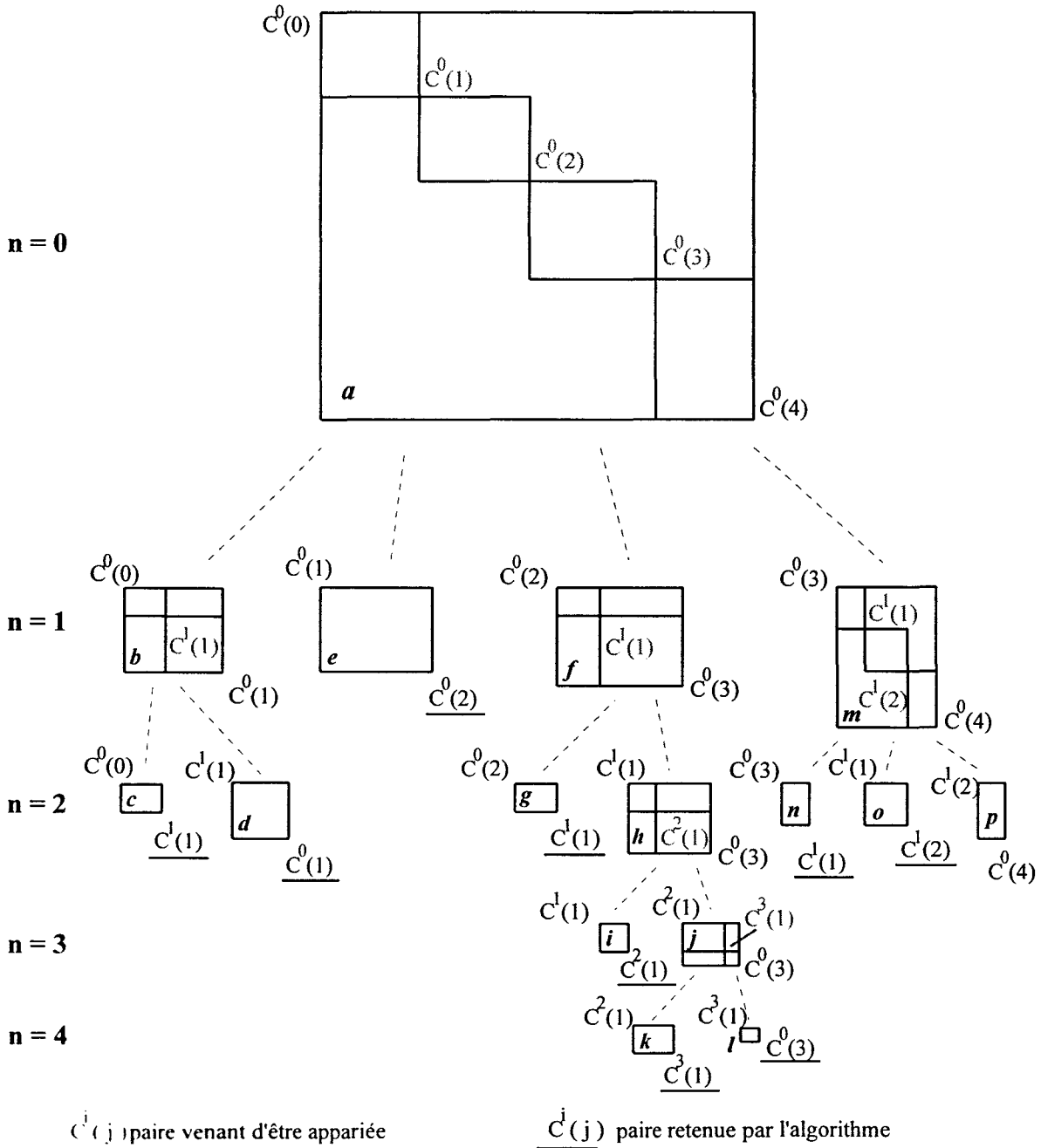


Figure 5.24 : Algorithme récursif de mise en correspondance hiérarchique.

Pour comprendre cette technique il suffit de regarder l'exemple de la figure 5.24. Dans ce graphe, le node a génère successivement les nodes b à p .

5-6.4.1.4. Prise en compte des résultats de l'algorithme récursif

La technique que nous employons s'apparente aux procédures aveugles de recherche avec graphe utilisées en intelligence artificielle [Nilsson 88]. Contrairement aux graphes utilisés en intelligence artificielle, la solution ne se trouve pas sur une feuille du graphe. En fait, chaque feuille possède une partie de la solution. Il faudra donc atteindre toutes les feuilles du graphe pour déterminer l'ensemble des solutions.

Lors de chaque appel à la procédure de mise en correspondance bidimensionnelle, les résultats, quand ils existent, servent à générer de nouveau plan de recherche. Nous pouvons mémoriser ces résultats au fur et à mesure, ce qui nous amène à sauvegarder dans notre exemple les résultats du node *a*, puis ceux du node *b* etc. Dans la liste finale, ainsi obtenue, les paires de primitives sont rangées dans l'ordre où elles sont appariées. Cependant pour faciliter l'utilisation ultérieure de cette liste, nous préférons conserver l'ordre que présentent les primitives dans les images.

Un appariement sera donc retenu quand un node ne pourra plus générer de node successeur. C'est à dire dès qu'une des conditions d'arrêt sera respectée (Cf. § 5-6.4.5). Pour cela, il suffira de conserver la paire définissant la *fin* de chaque feuille du graphe. Toutefois n'oublions pas qu'au niveau 0, le plan de recherche est initialisé avec les premiers et les derniers pixels de chaque image. Il ne faut donc pas les prendre en compte. Pour cela, il suffit simplement d'ignorer la *fin* du dernier node traité.

Dans le cas de notre exemple, après avoir traité les nodes *a* et *b*, l'algorithme aboutira au niveau 2 sur la feuille *c*. Il mémorisera donc la paire $C^1(1)$ qui aura été appariée au niveau 1. Ensuite il atteindra la feuille *d* toujours au niveau 2 mais cette fois-ci mémorisera la paire $C^0(1)$ qui aura été appariée au niveau 0. Finalement l'algorithme sera amené à mémoriser successivement les *fins* des feuilles *c*, *d*, *e*, *g*, *i*, *k*, *l*, *n* et *o*. Sur le graphe les paires retenues sont les extrémités soulignées. On constate que la paire $C^0(4)$ du node *p* n'est pas retenue puisqu'elle correspond au dernier node.

Afin de pouvoir implanter, l'algorithme de façon récursive, nous devons donc développer :

- Le module d'appariement élémentaire hiérarchique.
- La procédure permettant de sélectionner les primitives.

et définir :

- Les conditions initiales.
- Les conditions d'arrêt de l'algorithme.

5-6.4.2. Sélection des primitives de référence

Soit le niveau n de mise en correspondance, la procédure est appliquée au plan de recherche défini par $PR[(C_G^{n-1}(k), C_D^{n-1}(k)); (C_G^{n-1}(k+1), C_D^{n-1}(k+1))]$ avec $k=0,1,\dots,X^{n-1}-1$. Avant de lancer la procédure de mise en correspondance, il faut sélectionner des primitives dans chaque image.

L'idée retenue est que les contours qui délimitent des régions avec un fort contraste sont plus faciles à mettre en correspondance que les contours délimitant des régions présentant un faible contraste. En effet, lors de la mesure de ressemblance, les valeurs des différentes mesures S_{22} , S_{12} et S_{21} sont plus discriminantes si les régions présentent de grands contrastes. Lorsque les régions sont homologues la mesure S_{22} est faible, par contre lorsque celles-ci sont différentes cette mesure est d'autant plus grande que le contraste est important. Nous avons donc décidé de sélectionner les primitives en fonction de l'amplitude de leur dérivée.

Le premier critère de sélection impose que les primitives se trouvent dans le plan de recherche. Seules les primitives, dont les abscisses sont comprises entre les extrémités du plan de recherche, sont conservées. Pour les primitives gauches cela consiste à ne conserver que les primitives dont l'abscisse x vérifie la relation :

$$C_G^{n-1}(k) < x < C_G^{n-1}(k+1) \quad \text{Equation 5.5}$$

Pour les primitives droites, l'abscisse doit vérifier la relation :

$$C_D^{n-1}(k) < x < C_D^{n-1}(k+1) \quad \text{Equation 5.6}$$

Un second critère de sélection est appliqué aux listes établies en fonction du premier critère. Celui-ci est basé sur l'amplitude de la dérivée. Soit $d(x)$, l'amplitude de la dérivée associée à la primitive d'abscisse x . Le signal $d(x)$ étant signé, nous prenons également en compte le signe de la dérivée. L'algorithme ne conserve donc que les primitives gauches présentant une dérivée positive vérifiant la relation :

$$\frac{\text{Max}_g}{2 \cdot n} \leq d(x) \leq \text{Max}_g \quad \text{Equation 5.7}$$

et les primitives gauches présentant une dérivée négative vérifiant la relation :

$$\text{Min}_g \leq d(x) \leq \frac{\text{Min}_g}{2 \cdot n} \quad \text{Equation 5.8}$$

où n est le niveau de recherche et Max_g (resp. Min_g) l'amplitude maximale (resp. minimale) de la dérivée de l'image gauche. Cette valeur correspond à la réponse maximale (resp. la réponse minimale) de l'opérateur différentiel de Deriche (Cf. § 4-3.3.3).

Ainsi, au premier niveau de recherche, seules les primitives dont l'amplitude positive de la dérivée est comprise entre $\text{Max}_g/2$ et Max_g sont conservées. On remarque qu'au niveau suivant l'intervalle de sélection est compris entre $\text{Max}_g/4$ et Max_g . La raison est simple, si un contour n'a pas trouvé de correspondant lors de la mise en correspondance au niveau n , nous lui donnons une chance d'être apparié au niveau $n+1$. Le raisonnement est identique pour les primitives dont l'amplitude est négative avec la valeur Min_g .

Pour les primitives droites, des relations similaires sont appliquées :

$$\frac{\text{Max}_d}{2 \cdot n} \leq d(x) \leq \text{Max}_d \quad \text{Equation 5.9}$$

$$\text{Min}_d \leq d(x) \leq \frac{\text{Min}_d}{2 \cdot n} \quad \text{Equation 5.10}$$

où n est le niveau de recherche et Max_d (resp. Min_d) l'amplitude maximale (resp. minimale) de la dérivée de l'image droite.

Ainsi, au niveau n de recherche, la procédure de sélection des primitives fournit à partir des listes $G(i)$ et $D(j)$, les listes $G^n(i)$ et $D^n(j)$ qui sont respectivement la liste des primitives gauches vérifiant simultanément les relations données par les équations 5.5 et 5.7 ou 5.5 et 5.8 et la liste des primitives droites vérifiant simultanément les relations données par les équations 5.6 et 5.9 ou 5.6 et 5.10.

5-6.4.3. Module d'appariement élémentaire hiérarchique

Considérons le niveau de recherche n . La recherche au niveau $n-1$ a donné comme résultat la liste $\{ \{ C_G^{n-1}(k), C_D^{n-1}(k) \}, k = 0,1,\dots,X^{n-1} \}$ qui constitue l'ensemble des couples appariés à ce

niveau. La procédure est maintenant appliquée au plan de recherche défini par $PR[(C_G^{n-1}(k), C_D^{n-1}(k)); (C_G^{n-1}(k+1), C_D^{n-1}(k+1))]$ avec $k=0,1,\dots,X^{n-1}-1$.

La première étape consiste à déterminer à partir des listes de primitives $G(i)$ et $D(j)$, les listes $G^n(i)$ et $D^n(j)$ qui contiennent les primitives appartenant au plan de recherche considéré et dont l'amplitude est fonction du niveau de recherche (Cf. § 5-6.4.2).

La procédure de mise en correspondance bidirectionnelle est ensuite appliquée dans ce plan. Le résultat de celle-ci fournit une liste $\{ \{ C_G^n(p), C_D^n(p) \}, p = 1, \dots, P^n \}$ où P^n est le nombre de paires appariées à ce niveau.

Afin de pouvoir aisément créer les nouveaux plans de recherche au niveau $n+1$, on ajoute à la liste des résultats les paires définissant les extrémités du plan de recherche. On obtient alors la liste finale au niveau n définie $\{ \{ C_G^n(k), C_D^n(k) \}, k = 0, 1, \dots, X^n \}$ avec $X^n = P^n + 1$,

$$\{ C_G^n(0), C_D^n(0) \} = \{ C_G^{n-1}(k), C_D^{n-1}(k) \}$$

et
$$\{ C_G^n(P^n+1), C_D^n(P^n+1) \} = \{ C_G^{n-1}(k+1), C_D^{n-1}(k+1) \}.$$

La procédure continue ensuite au niveau $n+1$ en considérant le premier plan le plus à gauche c'est à dire $PR[(C_G^n(0), C_D^n(0)); (C_G^n(1), C_D^n(1))]$.

5-6.4.4. Conditions initiales

Comme tout algorithme récursif, notre algorithme hiérarchique nécessite une phase d'initialisation pour commencer à traiter les primitives. On considère donc la liste initiale suivante

$$\{ \{ C_G^0(p), C_D^0(p) \}, p = 0, 1 \}$$

avec
$$\{ C_G^0(0), C_D^0(0) \} = \{ 0, 0 \}$$

et
$$\{ C_G^0(1), C_D^0(1) \} = \{ N, M \}.$$

Au niveau 1, le plan de recherche $PR[(0,0), (N,M)]$ est confondu avec le plan de mise en correspondance (Cf. Figure 5.22).

5-6.4.5. Conditions d'arrêt

De même tout algorithme récursif nécessite une condition d'arrêt pour lui permettre de s'arrêter. Nous utilisons deux conditions d'arrêt. Lorsque l'une de ces conditions est satisfaite deux possibilités sont à envisager :

1. La paire $\{N, M\}$ définit l'extrémité inférieure droite du plan de recherche. Dans ce cas la procédure récursive est terminée, les images ont été entièrement traitées.
2. La paire $\{N, M\}$ ne définit pas l'extrémité inférieure droite du plan de recherche. On mémorise la paire $\{C_G^{n-1}(k+1), C_D^{n-1}(k+1)\}$, et on passe au plan de recherche suivant.

5-6.4.5.1. Condition d'arrêt n° 1

Soit le plan de recherche $PR[(C_G^{n-1}(k), C_D^{n-1}(k)); (C_G^{n-1}(k+1), C_D^{n-1}(k+1))]$ avec $k=0,1,\dots,X^{n-1}-1$ traité au niveau de recherche n . Si l'une des deux listes $G^n(i)$ et $D^n(j)$ est vide, aucun appariement n'est possible. L'algorithme ne poursuit pas les appariements dans le plan de recherche considéré. Ce cas se produit par exemple sur la figure 5.23 pour les deux premiers plans de recherche qui ne possèdent pas de primitives.

5-6.4.5.2. Condition d'arrêt n° 2

Soit le plan de recherche $PR[(C_G^{n-1}(k), C_D^{n-1}(k)); (C_G^{n-1}(k+1), C_D^{n-1}(k+1))]$ avec $k=0,1,\dots,X^{n-1}-1$. Les listes $G^n(i)$ et $D^n(j)$ ne sont pas vides mais le niveau de recherche a atteint la valeur n_{\max} . Ce cas se produit lorsque les primitives ne sont pas compatibles. Pour éviter à l'algorithme de tenter d'apparier indéfiniment ces primitives, on fixe un niveau limite afin que les recherches puissent s'arrêter dans le plan considéré. Ce critère d'arrêt est également utile lorsqu'on veut limiter la mise en correspondance aux primitives les plus significatives.

La valeur n_{\max} peut être fixée en fonction du critère de sélection des primitives et du seuil bas S_b utilisé lors de l'extraction des primitives (Cf. § 4-4). Puisque les primitives sont sélectionnées en fonction de l'amplitude de leur dérivée, il existe un niveau n_{\max} pour lequel :

$$\frac{\text{Max}_g}{2 \cdot n_{\max}} < S_b$$

n_{\max} est donc le plus petit nombre entier vérifiant la relation :

$$\frac{\text{Max}_g}{2 \cdot S_b} < n_{\max}$$

Des relations similaires peuvent être déterminées pour Max_d , Min_g et Min_d en prenant toutefois garde que pour les valeurs négatives Min_g et Min_d le seuil bas est $-S_b$.

Pratiquement, la valeur maximale de l'amplitude de la dérivée est 255 en valeur absolue, le seuil bas étant fixé à 10, ceci nous conduit à une valeur de n_{\max} égale à 13. Si nous fixons n_{\max} à 13 alors nous sommes sûrs de traiter toutes les primitives sélectionnées par le module d'extraction de contour. Dans la pratique, pour gagner du temps lors des traitements et éviter de traiter des détails très fins, on peut ajuster n_{\max} à des valeurs inférieures à 13.

5-6.5. Résultats obtenus lors de la mise en correspondance

Comme pour les procédures présentées dans les paragraphes précédents, l'algorithme hiérarchique de mise en correspondance est appliqué successivement sur chaque couple d'images stéréoscopiques. L'algorithme fournit pour chacun une liste constituée des appariements réalisés.

5-6.5.1. Piéton évoluant devant le stéréoscope

La figure 5.25 montre le résultat obtenu sur la séquence stéréoscopique "piéton", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.1).

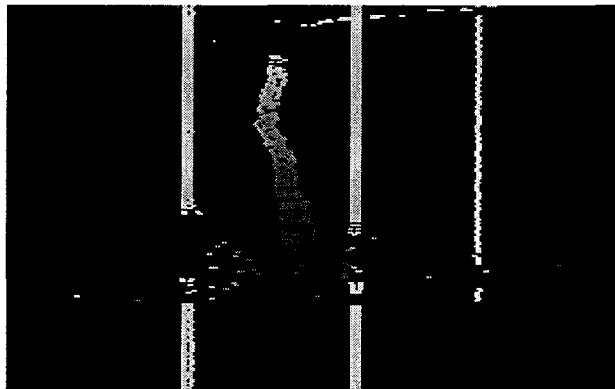


Figure 5.25 : Résultat obtenu sur la séquence "piéton".

Les résultats obtenus sont excellents pour les bandes blanches. A chaque fois qu'elles sont visibles simultanément par les deux caméras, leurs points de contour sont appariés. Néanmoins, on constate toujours la présence de faux appariements sur les images où se produisent des occlusions. Ceux-ci en nombre limités apparaissent de façon irrégulière et seront facilement éliminés par une analyse temporelle. Les points de contour du piéton sont également bien mis en correspondance. Même si on constate quelques "trous" sur certaines images, les variations de position et distance sont cohérentes d'une image à l'autre. Le résultat obtenu est suffisant pour définir son mouvement par un suivi temporel.

5-6.5.2. Route dégagée et passage d'une sortie d'autoroute

La figure 5.26 montre le résultat obtenu sur la séquence stéréoscopique "sortie d'autoroute", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.2).

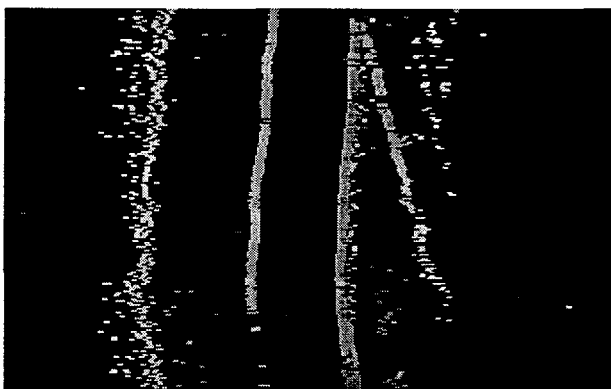


Figure 5.26 : Résultat obtenu sur la séquence "sortie d'autoroute".

Sur cette figure, on constate que les bandes blanches sont mieux détectées. La plupart des "trous" ont disparu. De même le nombre de faux appariements a diminué. Le résultat obtenu avec cette séquence permet d'envisager une reconnaissance précise de la route. En effet, la scène reconstituée présente une bonne stabilité dans le temps. Un suivi spatio-temporel des bandes blanches sera donc aisé à réaliser.

5-6.6. Conclusion sur l'algorithme hiérarchique de mise en correspondance

L'algorithme hiérarchique que nous venons de décrire est plus fiable que les méthodes décrites précédemment car il décompose le problème en une succession de sous problèmes plus simples. Il diminue ainsi la complexité du plan de mise en correspondance ce qui permet de déterminer aisément des noeuds de référence. Ces noeuds serviront à leur tour pour déterminer d'autres noeuds et ainsi de suite. L'avantage de cette technique est la suivante : si une erreur intervient, elle ne se répercute pas sur l'ensemble des résultats comme cela pouvait se produire pour l'analyse séquentielle ou bidirectionnelle. L'erreur reste localisée dans le plan de recherche où elle s'est produite.

5-7. Triangulation géométrique

Lorsque la mise en correspondance est terminée, la procédure de triangulation reçoit la liste des couples de primitives appariées. Son rôle est de déterminer pour chaque couple, la distance et la position du point de la scène en fonction :

- Des positions des deux primitives sur les capteurs gauche et droit.
- Des paramètres de calibration déterminés lors de la phase de calibrage du stéréoscope linéaire (Cf. chapitre 2).

A partir des équations définies au chapitre 2 (Cf. § 2-2.2.3), les formules de triangulation permettant de déterminer la distance Z et la position X sont :

$$\begin{cases} Z = \frac{f \cdot E \cdot N}{l \cdot (x_g - x_d)} \\ X = \frac{l \cdot Z \cdot (x_g - o_g)}{f \cdot N} - \frac{E}{2} \end{cases}$$

où :

- X est la position du point objet le long de l'axe Ox .
- Z est la distance du point objet le long de l'axe Oz .
- x_g et x_d sont respectivement la position du point image sur le capteur gauche et sur le capteur droit.
- o_g est la position du pixel central sur le capteur gauche c'est à dire le pixel sur lequel se projette un point situé le long de l'axe optique.
- f est la distance focale.
- E est l'entre axe du stéréoscope linéaire.
- N est la résolution du capteur c'est à dire le nombre de pixels dans l'image.
- l est la largeur du capteur.

Les résultats obtenus sont liés au repère du stéréoscope linéaire c'est à dire au repère (O, X, Z) (Cf. Figure 2-7).

On remarquera qu'on ne détermine pas la hauteur à laquelle le plan optique coupe les objets. Celle-ci n'est, en effet, pas utile car elle n'apporte aucune information supplémentaire sur les objets détectés.

La procédure de triangulation géométrique permet de déterminer, pour chaque couple de primitives, un point de la scène réelle auquel nous associons les caractéristiques suivantes :

- Abscisse de la primitive gauche sur le capteur gauche (position dans l'image gauche).
- Abscisse de la primitive droite sur le capteur droit (position dans l'image droite).
- La valeur de l'amplitude du gradient associée à la primitive gauche (Cf. § 4-6).

- La valeur de l'amplitude du gradient associée à la primitive droite.
- La nature de la pente : front montant ou descendant (Cf. § 4-6).
- Position du point dans la scène réelle.
- Distance du point dans la scène réelle.

Tous les points calculés forment une liste appelée *liste stéréo* dont les éléments sont ordonnés selon les positions croissantes. Cette liste sera utilisée pour la reconstruction tridimensionnelle et la détection d'obstacles.

5-8. Conclusion

En stéréovision matricielle, il existe de nombreuses méthodes de mise en correspondance. En vision linéaire, peu d'études ont été faites sur le sujet (Cf. 3-6). Ceci nous a donc amené à développer notre propre méthode de mise en correspondance. La méthode est originale car contrairement à la plupart des autres méthodes elle ne cherche pas, pour une primitive donnée, le meilleur appariement parmi tous les appariements possibles. Notre méthode est capable de déterminer immédiatement en exploitant au maximum les contraintes disponibles si un appariement formé est valide ou ne l'est pas. Cette technique évite ainsi beaucoup de calculs inutiles, ce qui procure un gain de temps appréciable. Le problème consiste maintenant à déterminer parmi les primitives appariées celles qui correspondent à des objets, d'analyser le comportement de ceux-ci au cours du temps afin de juger s'ils représentent un danger. Ces problèmes relatifs à la détection d'obstacles feront l'objet du prochain chapitre.

Chapitre 6

UTILISATION DE LA SIGNALISATION HORIZONTALE POUR LA MISE EN CORRESPONDANCE

6-1. Introduction

L'objectif de notre système stéréoscopique est de détecter les obstacles se trouvant à l'avant des véhicules routiers. Cependant une telle application doit demander des temps de réponse très courts si l'on veut que le système puisse alerter suffisamment tôt le conducteur de la présence d'un danger éventuel.

Nous avons vu, au chapitre 3, que la majorité des systèmes stéréoscopiques s'organise de la façon suivante :

- Acquisition des images stéréoscopiques.
- Extraction des primitives dans chaque image.
- Mise en correspondance des primitives.
- Triangulation géométrique.
- Reconstruction tridimensionnelle de la scène.

Les quatre premières étapes que nous avons décrites dans les chapitres précédents suivent fidèlement cette structure. La reconstruction tridimensionnelle de la scène apparaît donc logiquement comme la dernière étape à développer.

Cependant, pour améliorer les temps de traitement et simplifier la reconstruction tridimensionnelle, les trois dernières étapes, c'est à dire la mise en correspondance, la triangulation et une reconstruction partielle de la scène, ont été combinées. Ces trois étapes réunies permettent de définir la mise en correspondance bi-mode. Celle-ci est basée sur l'utilisation de la signalisation horizontale (Cf. § 6-3). La première étape consiste à détecter les bandes blanches présentes sur la chaussée (Cf. § 6-4), puis d'effectuer leur suivi dans le temps (Cf. § 6-5). Lorsque la position des bandes blanches est connue, elles sont utilisées pour définir une zone de danger (Cf. § 6-6). L'organisation judicieuse de ces diverses opérations permet de structurer la procédure de mise en correspondance bi-mode (Cf. § 6-7).

6-2. Définitions et notations

6-2.1. Repères de travail

Jusqu'à présent, les traitements étaient effectués dans les repères liés aux capteurs, c'est à dire les repères (o_g, x_g) pour l'image gauche et (o_d, x_d) pour l'image droite. Les points de contour de l'image gauche (resp. droite) sont d'ailleurs définis par leur abscisse dans le repère (o_g, x_g)

(resp. (o_d, x_d)). Or, après la mise en correspondance, la procédure de triangulation nous permet de déterminer les coordonnées dans la scène réelle des points ayant engendré ces points de contour. Celles-ci sont calculées dans le repère lié au stéréoscope linéaire, c'est à dire le repère (O, X, Z) (Cf. Figure 2.7).

Pour éviter toute confusion dans notre exposé, nous employons le terme *abscisse* pour faire référence aux repères images. Les termes *position* et *distance* sont utilisés pour situer les points dans la scène réelle.

Dans ce chapitre, les différents objets que nous reconstruisons et obstacles que nous analysons sont définis dans ces trois repères. En effet, les positions et distances s'expriment dans le repère (O, X, Z) , mais chaque objet possède également une image sur chaque capteur. Il peut donc être défini dans les repères (o_g, x_g) et (o_d, x_d) .

6-2.2. "Liste stéréo"

Nous sommes amenés, au cours de ce chapitre, à faire référence à la liste des paires de primitives appariées fournie par la procédure de mise en correspondance. A cette liste, appelée *liste stéréo*, a été associé un certain nombre de caractéristiques (Cf. § 5-7) que nous utilisons au cours des différentes procédures développées pour détecter les obstacles.

Soit $S(n)$ avec $n=1,2,\dots, N$, la liste stéréo où N est le nombre d'éléments contenus dans cette liste. Nous notons les différentes caractéristiques associées à la liste stéréo de la façon suivante :

Pour l'élément $S(n)$

$S(n).x_G$: Abscisse de la primitive gauche dans le repère image gauche (o_g, x_g) .

$S(n).x_D$: Abscisse de la primitive droite dans le repère image droite (o_d, x_d) .

$S(n).d(x_G)$: Amplitude de la dérivée associée à la primitive gauche d'abscisse x_G .

$S(n).d(x_D)$: Amplitude de la dérivée associée à la primitive droite d'abscisse x_D .

$S(n).pente$: Nature de la pente associée aux primitives.

$S(n).position$: Position du point dans la scène réelle, c'est à dire dans le repère (O, X, Z) .

$S(n).distance$: Distance du point dans la scène réelle, c'est à dire dans le repère (O, X, Z) .

Notons que les caractéristiques *position* et *distance* ne peuvent être utilisées qu'après l'application de la procédure de triangulation.

La contrainte d'ordre impose les trois relations suivantes :

- $S(n).x_G < S(n+1).x_G$
- $S(n).x_D < S(n+1).x_D$
- $S(n).position < S(n+1).position$

Les éléments de cette liste stéréo sont donc ordonnés selon leurs abscisses croissantes mais également en fonction des positions croissantes.

6-2.3. "Liste objet"

Les différents éléments de la liste stéréo nous permettent de reconstruire les objets. Ceux-ci sont rangés dans une liste appelée *liste objet*, notée $Obj(m)$ avec $m = 1, 2, \dots, M$, où M est le nombre d'objets présents dans la liste. Les objets sont ordonnés en fonction des positions croissantes. A chaque objet est associé un certain nombre d'attributs que nous énoncerons au cours de ce chapitre.

Puisque nous ne connaissons pas le nombre d'objets, les attributs de chaque objet sont stockés dans une structure chaînée créée dynamiquement au fur et à mesure des besoins.

6-2.4. Notions de couple et de paire

Dans le chapitre 5, nous avons utilisé indifféremment les termes de paires ou couples de primitives lorsque nous tentions d'apparier un point $G(i)$ avec un point $D(j)$. Une paire était notée $\{G(i), D(j)\}$.

Dans ce chapitre, la reconstruction tridimensionnelle nous oblige à rassembler les primitives dans les repères images et les points ayant engendré ces primitives dans le repère (O, X, Z) pour former des objets. Nous employons désormais le terme *couple* lorsque nous associons deux primitives de la même image, deux points de la liste stéréo ou deux objets pour former un nouvel objet. Selon le cas, le couple sera noté respectivement $\{G(i), G(i+k)\}$, $\{S(n), S(n+k')\}$ ou $\{Obj(m), Obj(m+k'')\}$.

Le terme *paire* conserve la signification et la notation données au chapitre 5.

6-3. Principe de la méthode de mise en correspondance bi-mode

La méthode de mise en correspondance que nous avons exposée dans le chapitre 5, ne prend en compte ni le contenu des images, ni les résultats obtenus en traitant les images précédentes. Dans ce chapitre, nous n'allons pas redéfinir une nouvelle procédure d'appariement mais plutôt utiliser les résultats qu'elle fournit afin de diminuer les temps de traitements. Pour réaliser cette opération, la procédure de mise en correspondance a été développée pour fonctionner selon *deux modes*, d'où le nom de mise en correspondance *bi-mode*. Le principe de la méthode est exposé dans ce paragraphe.

6-3.1. Etude du contexte routier

Une séquence acquise par l'une des deux caméras du stéréoscope linéaire installé à bord d'un véhicule est présentée sur la figure 6.1. On distingue parfaitement les traces blanches laissées par la signalisation horizontale. On constate qu'elles ne sont pas droites en raison des oscillations dues aux suspensions du véhicule lorsque celui-ci se déplace. De chaque côté de la chaussée, on remarque les traces laissées par les buissons et les arbres se trouvant sur le bord de la route. Les marques sombres sur la voie gauche de la chaussée correspondent à des bandes rugueuses.

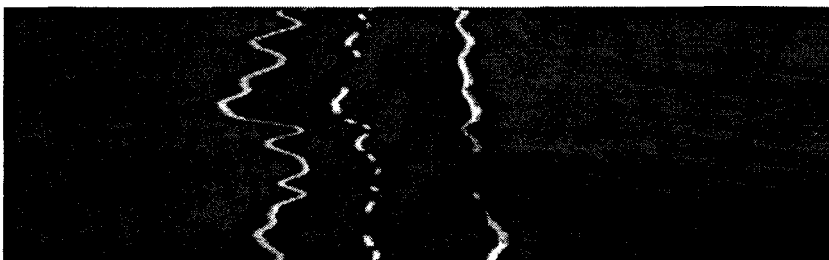


Figure 6.1 : Séquence d'images linéaires représentant une route.

Le stéréoscope linéaire est calibré de telle sorte que le plan optique rencontre la chaussée 30 m en avant du véhicule. Or, à cette distance, le stéréoscope linéaire couvre un champ de 20 m de large. Il est bien évident que les objets situés à 10 m à gauche ou à droite du véhicule ne présentent aucun danger direct pour celui-ci. La conclusion que nous pouvons donc tirer en regardant cette séquence est qu'une étude complète de chaque image linéaire est inutile. En effet, seuls les objets situés sur la route peuvent constituer un danger potentiel pour le véhicule. D'autre part, les bords de l'image qui correspondent aux bas-côtés de la route engendrent de nombreux points de contour que nous traitons inutilement puisque nous devons les éliminer par la suite. De plus, ces traitements augmentent les temps de calcul de façon importante.

6-3.2. Définition de la zone de danger

L'examen de différentes séquences acquises sur route nous a donné l'idée de restreindre la recherche des obstacles à une zone que nous avons appelée "*zone de danger*". Cette limitation a pour objectif de simplifier les traitements et surtout de diminuer les temps de calcul. Le problème consiste à définir une telle zone.

Généralement les routes sont délimitées par des bandes blanches situées sur le bord de la chaussée. L'idée que nous avons retenue est de repérer ces bandes et de les utiliser pour déterminer les limites de la zone de danger. Celle-ci est définie d'une part dans les repères (o_g, x_g) et (o_d, x_d) par les abscisses des contours des bandes. Les traitements peuvent donc être limités aux points de contour situés entre ces abscisses. D'autre part, après reconstruction tridimensionnelle, la zone de danger est également définie dans le repère (O, X, Z) par la position X_{min} de la bande gauche, la position X_{max} de la bande droite et la distance maximale Z_{max} qui correspond à la distance à laquelle le plan de visée rencontre la chaussée (Cf. Figure 6.2). Tous les objets se situant dans cette zone sont considérés comme des obstacles potentiels.

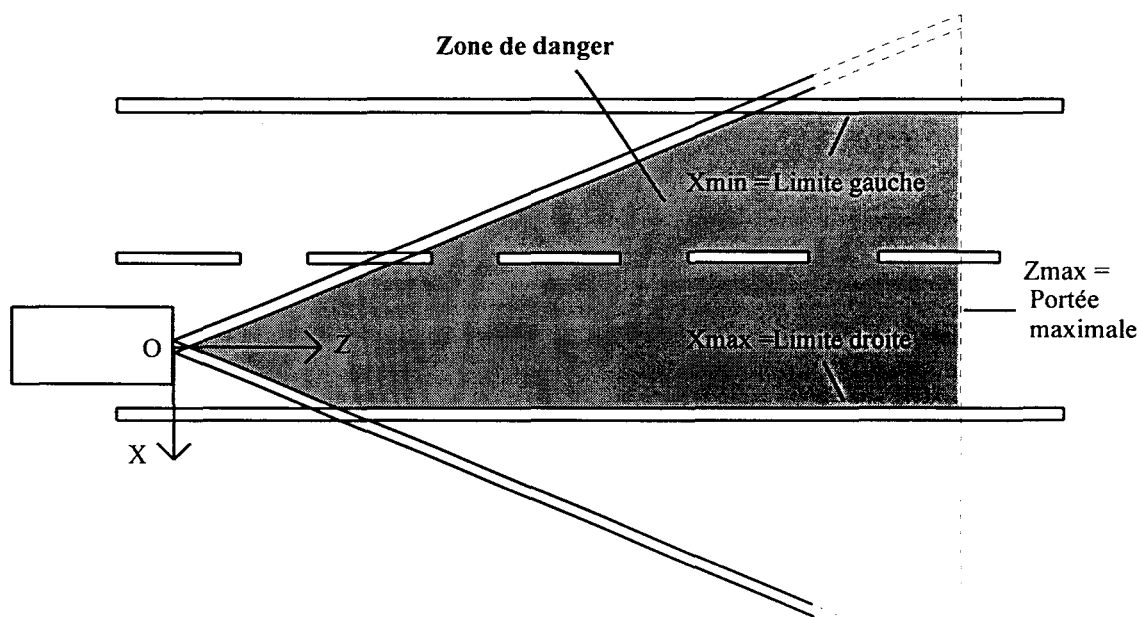


Figure 6.2 : Matérialisation de la zone de danger.

Cependant il est absurde de figer l'algorithme pour qu'il ne fonctionne qu'en présence de bandes blanches. En effet la signalisation horizontale est parfois inexistante ou s'est souvent effacée avec le temps. Dans ce genre de situation, aucune information ne permet de restreindre les traitements à une zone précise des images. Les images sont alors traitées intégralement

dans les repères images. Cependant, lorsque la mise en correspondance est terminée, des limites arbitraires sont fixées dans le repère (O, X, Z) afin de définir une zone de danger théorique pour limiter le nombre d'obstacles à étudier.

6-3.3. Principe de la méthode de mise en correspondance bi-mode

La limitation des traitements se fait principalement au niveau de la procédure de mise en correspondance qui est l'étape la plus gourmande en temps de calcul. En effet, celle-ci dépend essentiellement du nombre de points de contour présents dans l'image. Une réduction du nombre de points à traiter entraîne systématiquement une diminution des temps de traitement.

Les appariements des points de contour sont toujours réalisés par l'algorithme hiérarchique de mise en correspondance que nous avons présenté au § 5-6. Cependant la phase d'initialisation de cet algorithme dépend du contexte routier.

Quand aucune bande blanche n'est présente sur la chaussée, l'algorithme utilise le *mode global*. Ce mode correspond à l'analyse complète du plan de mise en correspondance comme nous l'avons décrit au chapitre 5. La procédure récursive est initialisée au niveau 1 dans le plan de recherche PR [(0,0); (N,M)] (Cf. Figure 6.3).

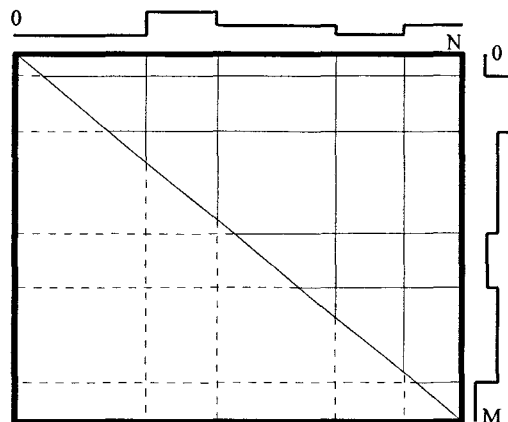


Figure 6.3 : Condition initiale de la procédure de mise en correspondance en l'absence de bande blanche.

En présence de bandes blanches, l'algorithme utilise le *mode local*. Ce mode correspond à une analyse partielle du plan de mise en correspondance. La procédure d'appariement est initialisée au niveau 1 dans le plan de recherche PR[(x_{o_g} , x_{o_d}); (x_{f_g} , x_{f_d})] (Cf. Figure 6.4) où x_{o_g} et x_{f_g} sont les abscisses, dans l'image gauche, des points de contour des bandes blanches définissant la zone de danger et x_{o_d} et x_{f_d} sont les abscisses des points homologues dans l'image droite.

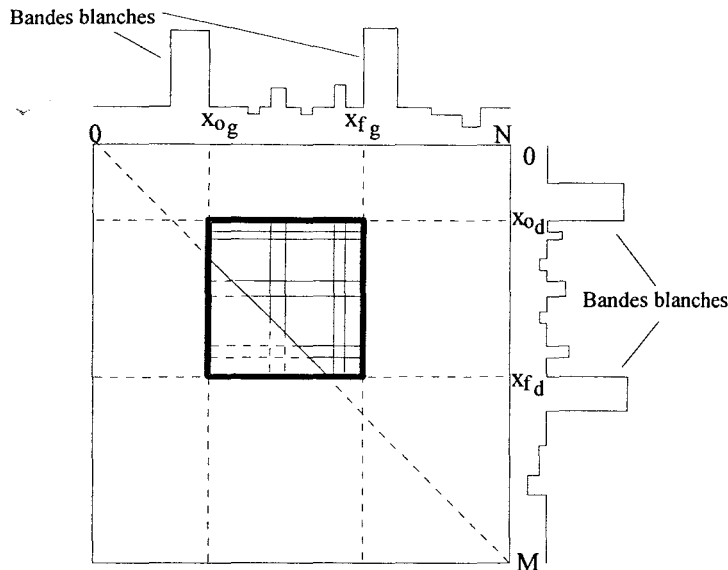


Figure 6.4 : Condition initiale de la procédure de mise en correspondance en présence de bandes blanches.

Comme nous ne détenons aucune information sur la nature de la route, l'algorithme commence par utiliser le mode global de mise en correspondance. Le passage du mode global au mode local se fait lorsque des bandes blanches apparaissent dans les images. Le passage du mode local au mode global se fait quand les bandes blanches disparaissent.

Pour mettre en oeuvre un tel processus de mise en correspondance, nous devons disposer d'une part d'une procédure détectant la présence de bandes blanches et d'autre part d'une procédure de suivi dynamique de ces bandes. La détection des bandes permet de définir la zone de danger et donc de passer en mode local. Tant que la procédure de suivi retrouve les bandes dans les images suivantes, le mode local reste activé. Par contre, en cas de disparition des bandes blanches, la procédure de suivi ordonne à l'algorithme de repasser en mode global.

6-4. Détection des bandes blanches

6-4.1. Etude préliminaire

Sur la figure 6.1, on constate que les bandes blanches se démarquent du reste de l'image car elles présentent des niveaux de gris très clairs. Cependant ce seul critère n'est pas suffisant pour détecter la présence d'une bande blanche. En effet les images peuvent présenter des zones dont les niveaux de gris sont très clairs comme des zones d'ensoleillement. La figure 6.5 montre le profil d'une bande blanche et d'une zone claire dans une image linéaire I où $I(x_i)$ est le niveau de gris au point d'abscisse x_i .

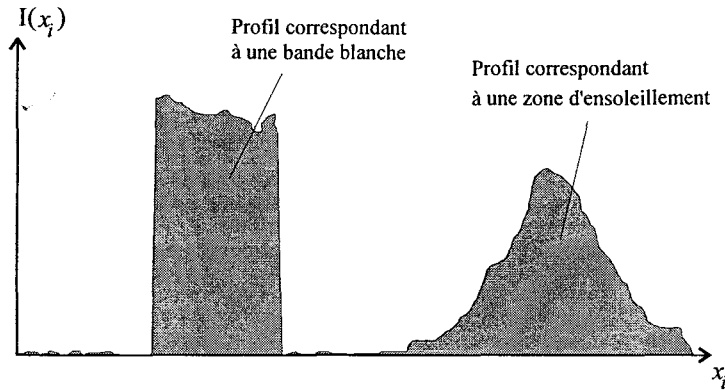


Figure 6.5 : Profil d'une bande blanche et d'une zone claire.

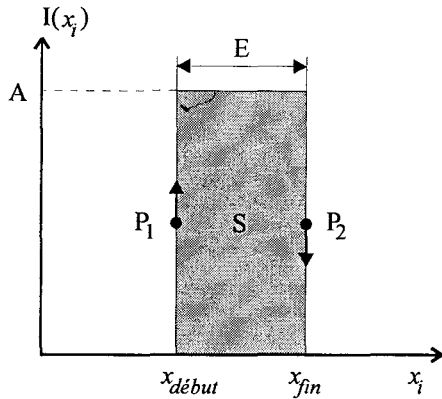
Si ces deux profils sont différents, ils peuvent parfois présenter un certain nombre de caractéristiques communes comme l'amplitude maximale des niveaux gris ou la largeur. Ces ressemblances conduisent alors à des confusions qui dans notre cas peuvent avoir des conséquences dramatiques. En effet, si la zone de danger est mal définie, certains obstacles risquent de ne pas être repérés. Une analyse plus poussée doit donc être effectuée pour permettre une détection fiable des bandes blanches.

Pour réaliser cette opération, nous disposons des images gauche et droite, mais également de la liste stéréo obtenue par le mode global de mise en correspondance. Cependant cette liste ne contient que des points dont les coordonnées dans le repère (O, X, Z) sont connues. La détection des bandes blanches nécessitant la reconnaissance de l'objet "**Bande Blanche**", une reconstruction partielle de la scène doit être effectuée à partir des points contenus dans la liste stéréo pour former les objets recherchés.

Pour détecter les bandes blanches, nous nous sommes inspirés des travaux de Lailier [Lailier 93] qui n'utilise qu'une seule caméra matricielle. Cependant, notre système stéréoscopique fournit des informations qui simplifient cette recherche, notamment la largeur réelle des bandes.

6-4.2. Paramètres caractéristiques d'une bande blanche

Pour déterminer la présence des bandes blanches dans une image linéaire, il est nécessaire de définir les paramètres qui caractérisent une bande blanche idéale dans le repère image. La figure 6.6 représente le profil d'une bande blanche idéale, où $I(x_i)$ est le niveau de gris au point d'abscisse x_i de l'image linéaire.



A : Amplitude des niveaux de gris de la bande blanche.
 E : Epaisseur du profil de la bande blanche.
 S : Surface engendrée par le profil de la bande blanche.
 P_1, P_2 : Pentés correspondant aux fronts de la bande blanche.

Figure 6.6 : Paramètres caractéristiques d'une bande blanche idéale.

- La valeur du paramètre A détermine la visibilité de la bande blanche, notamment pour savoir si la bande blanche se démarque suffisamment du fond.
- Le paramètre E correspond à l'épaisseur du profil de la bande blanche dans le repère image.
- Les valeurs des paramètres P_1 et P_2 donnent une information sur la netteté de la bande blanche. Ils permettent de différencier une bande blanche d'une zone d'ensoleillement. Dans le cas d'une bande blanche bien nette, ces paramètres ont des valeurs importantes.
- Pour le paramètre de surface S , nous allons considérer deux types de surface, l'une appelée surface théorique S_t qui correspond à celle d'une bande blanche idéale dans une image linéaire et l'autre appelée surface réelle S_r qui correspond à la surface réelle engendrée par la bande blanche dans l'image linéaire. Ces deux surfaces sont calculées de la façon suivante :

$$S_t = A \cdot E$$

$$S_r = \sum_{i=début}^{i=fin} I(x_i)$$

Ainsi, la différence entre la surface théorique S_t et la surface réelle S_r fournit une information sur la dégradation de la bande blanche qui correspond en général aux imperfections de celle-ci (Cf. Figure 6.7). Dans le cas d'une bande blanche idéale, la valeur de S_r est égale à la valeur de S_t .

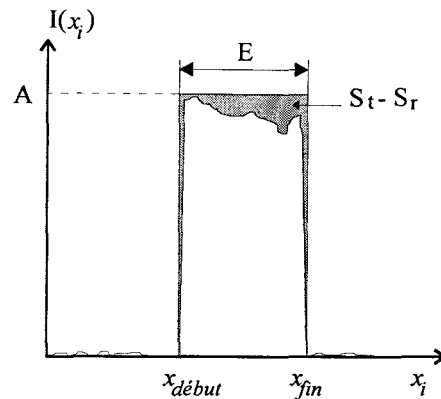


Figure 6.7 : Différence entre la surface théorique maximale S_t et la surface réelle S_r .

Ces paramètres sont utilisés pour différencier les bandes blanches des zones ensoleillées. Cependant la première étape consiste à trouver des objets ressemblant à des bandes blanches.

6-4.3. Reconstruction des objets ressemblant à une bande blanche

A partir de la liste stéréo, nous allons reconstruire des objets dont la forme est similaire à celle d'une bande blanche.

Une bande blanche peut être caractérisée par ses contours. La figure 6.6 montre que le profil d'une bande blanche présente un front montant suivi immédiatement d'un front descendant. Or, nous disposons dans la liste stéréo de l'information relative à la nature de la pente. Les points de la liste étant ordonnés en fonction des abscisses croissantes, nous pouvons donc considérer que le point $S(n)$ présentant un front montant suivi du point $S(n+1)$ présentant un front descendant est un indice de présence d'une bande blanche.

Cependant pour limiter le nombre d'objets pouvant ressembler à une bande blanche, nous utilisons un deuxième paramètre également caractéristique : la largeur réelle de la bande. La position, dans la scène réelle, des différents points de la liste stéréo étant connue, nous pouvons déterminer la distance entre deux de ces points. L'objet formé à partir des points $S(n)$ et $S(n+1)$ est retenu si sa largeur est compatible avec celles des bandes blanches normalisées.

La Communauté Urbaine de Lille - Direction de la Voirie et des Transports Urbains - Unité Territoriale de Signalisation fournit une documentation sur la signalisation horizontale. Une étude de cette documentation montre que, dans la pratique, la largeur des bandes varie de 6 cm pour les pistes cyclables à 50 cm pour les passages piétons. Pour nous assurer de détecter toutes les bandes blanches existantes, nous utilisons ces deux limites.

Enfin, les deux points $S(n)$ et $S(n+1)$ retenus correspondent aux contours d'une bande si les attributs de distance sont compatibles. En effet, les contours d'une bande doivent théoriquement se trouver à la même distance. Cependant, dans la pratique, les distances calculées ne sont pas toujours parfaitement égales. Pour prendre en compte les éventuelles erreurs de précision, nous imposons comme troisième et dernier critère de sélection, que la différence entre les deux paramètres $S(n).distance$ et $S(n+1).distance$ soit inférieure à 50 cm.

A partir de la liste stéréo, nous construisons donc des objets ressemblant à des bandes blanches en fonction des quatre critères suivants :

$$S(n).pente = \text{Front Montant.}$$

$$S(n+1).pente = \text{Front Descendant.}$$

$$\text{Largeur} = S(n+1).position - S(n).position$$

$$| S(n+1).distance - S(n).distance | < 50 \text{ cm}$$

avec

$$n = 1, 2, \dots, N-1$$

$$6 \text{ cm} \leq \text{Largeur} \leq 50 \text{ cm.}$$

L'étape suivante consiste à vérifier si les objets sélectionnés correspondent réellement à des bandes blanches.

6-4.4. Reconnaissance des bandes blanches

Afin de différencier les bandes blanches des zones très claires de l'image, il faut quantifier les différents paramètres donnés au § 6-4.2. Ils sont déterminés dans chaque image grâce aux caractéristiques associées à la liste stéréo.

Dans les quatre paragraphes suivants, nous omettons volontairement les indices G et D afin de rester dans un cas général. Pour appliquer ces calculs à l'image gauche (resp. droite) il suffit de remplacer la variable x par x_G (resp. x_D)

6-4.4.1. L'amplitude A

La valeur de l'amplitude A s'obtient par l'expression suivante :

$$A = \text{Max}(I(x_i)) \quad \text{avec } x_{début} \leq x_i \leq x_{fin}$$

où :

$$x_{début} = S(n).x$$

$$x_{fin} = S(n+1).x$$

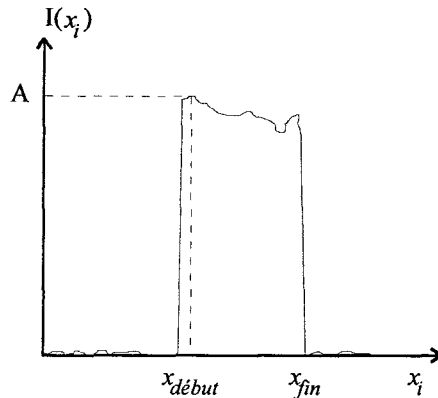


Figure 6.8 : Calcul de l'amplitude A.

6-4.4.2. L'épaisseur E

Le paramètre E indique l'épaisseur, exprimé en pixels, du profil. Il s'obtient par l'expression suivante :

$$E = x_{fin} - x_{début}$$

où :

$$x_{début} = S(n).x$$

$$x_{fin} = S(n+1).x$$

6-4.4.3. Les pentes P₁ et P₂

Les paramètres P₁ et P₂ indiquent l'amplitude de la dérivée, obtenue par l'opérateur de Deriche (Cf. Chapitre 4), aux points d'abscisses S(n).x et S(n+1).x. Il s'agit respectivement des caractéristiques S(n).d(x) et S(n+1).d(x).

6-4.4.4. Les surfaces S_t et S_r

Le principe retenu pour déterminer les valeurs de S_r et de S_t consiste à calculer ces deux surfaces entre les points de contour d'abscisses S(n).x et S(n+1).x. La différence entre ces deux surfaces permet de quantifier les imperfections d'une éventuelle bande blanche. Les surfaces S_t et S_r sont obtenues grâce aux expressions suivantes :

$$S_t = (A - I_m) \cdot E$$

$$S_r = \left(\sum_{i=début}^{i=fin} I(x_i) \right) - (I_m \cdot E)$$

avec :

A : l'amplitude

E : l'épaisseur du profil

$$x_{début} = S(n).x$$

$$x_{fin} = S(n+1).x$$

et
$$I_m = \frac{I(x_{début}) + I(x_{fin})}{2}$$

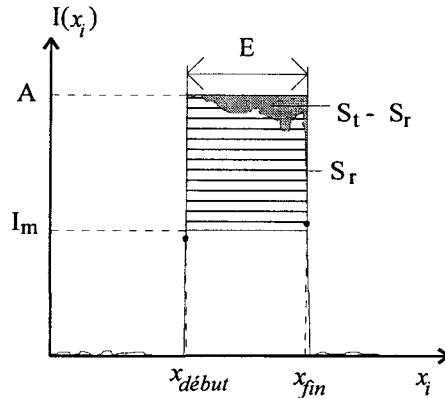


Figure 6.9 : Calcul des paramètres S_t et S_r .

6-4.4.5. Règles de décision

Pour déterminer si l'objet étudié correspond effectivement à une bande blanche, il suffit d'appliquer les règles suivantes :

1. $\frac{A}{A_{\max}} \geq \lambda_A$ avec $A_{\max} = I(x_i)_{\max} - I(x_i)_{\min} = 256$ pixels.
2. $\frac{|P_1|}{P_{\max}} \geq \lambda_p$ et $\frac{|P_2|}{P_{\max}} \geq \lambda_p$ avec $P_{\max} = \frac{d(x_i)_{\max}}{d(x_i)_{\min}} = \frac{256 \text{ pixels}}{1 \text{ pixel}} = 256$
3. $\frac{S_t - S_r}{S_t} \geq \lambda_s$

La première règle élimine les objets dont la luminosité ne présente pas un contraste suffisant avec le fond de l'image, en comparant l'amplitude normalisée du paramètre A à un seuil λ_A .

La deuxième élimine les objets dont les contours ne sont pas assez nets, en comparant les valeurs normalisées des pentes P_1 et P_2 à un seuil λ_p .

Enfin, la troisième règle permet d'éliminer les zones qui présentent des imperfections trop importantes par rapport à une bande blanche idéale. Ces imperfections sont quantifiées, en normalisant la différence entre la surface théorique S_t et la surface réelle S_r qui est ensuite comparée à un seuil λ_s .

Les seuils λ_A , λ_p et λ_s peuvent être ajustés en fonction de la sensibilité recherchée pour détecter les bandes blanches. Dans notre cas, nous sommes très tolérants sur les seuils λ_A et λ_p .

que nous fixons à 5 %. Par contre, le seuil λ_s qui permet d'éliminer les zones présentant des imperfections trop importantes par rapport à une bande blanche idéale est fixé à 50 %.

6-4.5. Conclusion sur la détection des bandes blanches

Grâce aux différents paramètres caractéristiques, il est possible de déterminer si les objets sélectionnés sont susceptibles de correspondre effectivement à une bande blanche. Pour chaque objet, la procédure de reconnaissance des bandes blanches est appliquée dans chaque image. Lorsque les 3 règles sont vérifiées simultanément dans l'image gauche et l'image droite, l'objet étudié est considéré comme une bande blanche.

Si l'objet formé à partir des points $S(n)$ et $S(n+1)$ de la liste stéréo est considéré comme une bande blanche, alors il constitue l'objet $Obj(m)$ de la liste objet et on lui associe les attributs suivants :

1. $Obj(m).type = \text{"bande blanche"}$
2. $Obj(m).x_G.début = S(n).x_G$
3. $Obj(m).x_G.fin = S(n+1).x_G$
4. $Obj(m).x_D.début = S(n).x_D$
5. $Obj(m).x_D.fin = S(n+1).x_D$
6. $Obj(m).position.début = S(n).position$
7. $Obj(m).position.fin = S(n+1).position$
8. $Obj(m).distance.début = \text{Min}(S(n).distance, S(n+1).distance)$
9. $Obj(m).distance.fin = \text{Max}(S(n).distance, S(n+1).distance)$

L'attribut 1 sert à identifier le type de l'objet. Ici, il s'agit d'une bande blanche. Les attributs 2 et 3 (resp. 4 et 5) permettent de définir la bande blanche dans le repère (o_g, x_g) (resp. le repère (o_d, x_d)). Les attributs 6 à 9 permettent de définir la bande blanche dans le repère (O, X, Z) .

On notera que, dans les repères images (o_g, x_g) et (o_d, x_d) , deux attributs suffisent pour localiser et définir un objet (Cf. Figure 6.10).

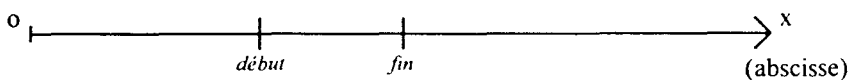


Figure 6.10 : Localisation d'une bande blanche dans un repère image.

Il en faut par contre 4 pour le situer parfaitement dans le repère (O, X, Z) . (Cf. Figure 6.11).

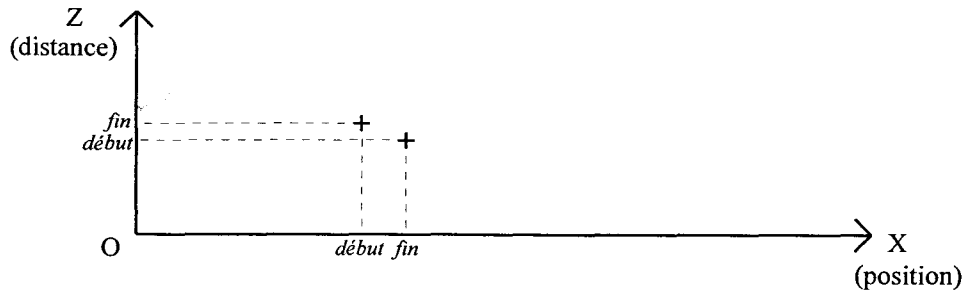


Figure 6.11 : Localisation d'une bande blanche dans le repère (O, X, Z).

En théorie, les contours d'une bande blanche se trouvent à la même distance. En pratique, il arrive parfois que des erreurs dues aux imprécisions de localisation du stéréoscope entraînent une légère différence de 1 à 2 %. C'est pour cette raison que nous utilisons les deux attributs *distance.début* et *distance.fin*. Nous verrons l'utilité de ces deux attributs lors de la reconstruction des autres objets (Cf. Chapitre 7).

6-5. Suivi des bandes blanches

La procédure de suivi a pour objectif de maintenir le mode local de mise en correspondance. Le problème consiste donc à déterminer les nouvelles positions et abscisses des bandes blanches car ces paramètres sont essentiels pour initialiser ce mode.

6-5.1. Etude préliminaire

L'opération de suivi nécessite d'abord que l'on choisisse un repère de travail. Si les coordonnées des bandes blanches dans le repère (O, X, Z) sont connues à l'instant $t-1$, seul le mode global de mise en correspondance permet de les calculer à l'instant t . Or nous voulons éviter d'avoir recours à ce mode lorsque des bandes blanches sont présentes sur la chaussée. La seule possibilité consiste donc à travailler dans les repères images (o_g, x_g) et (o_d, x_d).

La figure 6.1 montre que d'une image linéaire à l'autre, il existe une certaine continuité au niveau des bandes blanches. En vision matricielle on dirait que les régions représentant les bandes blanches sont connexes. En vision linéaire, la notion de connexité n'existe pas. Cependant nous emploierons les expressions de "*connexité spatiale*" ou "*spatialement connexe*" pour signaler que deux régions appartenant à deux images linéaires successives possèdent au moins un pixel situé à une abscisse commune (Cf. Figure 6.12).

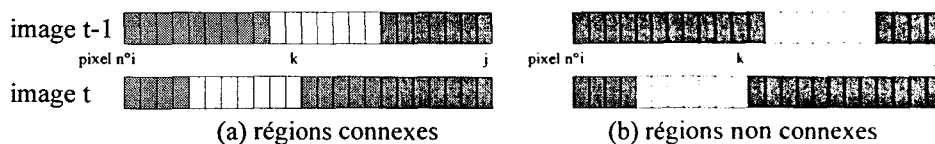


Figure 6.12 : Notion de connexité spatiale.

Si les bandes blanches sont spatialement connexes, on peut conclure que l'abscisse des points de contour définissant les bandes blanches varie peu d'une image à l'autre. La recherche des points de contour définissant la bande à l'instant t peut alors se faire dans une zone de recherche restreinte déterminée en fonction des abscisses de la bande à l'instant $t-1$.

Une bande blanche est un objet stable, c'est à dire que son image ne se déforme pas au cours de la progression du véhicule. On peut donc en déduire que si une bande blanche est présente sur une image aux instants $t-1$ et t , elle conserve un profil caractéristique identique. La procédure de reconnaissance des bandes blanches (Cf. § 6-4.4) peut donc facilement vérifier qu'un profil sélectionné dans la zone de recherche restreinte correspond bien à l'objet recherché.

La procédure de suivi se déroule donc en deux étapes. La première détermine une zone dans laquelle on cherche un profil ressemblant à celui d'une bande blanche. Dans un deuxième temps on vérifie que le profil trouvé correspond bien à une bande blanche.

Pour réaliser cette procédure, nous disposons, à l'instant t , des listes de primitives $G(i)$ et $D(j)$ fournies par la procédure d'extraction des primitives (Cf. Chapitre 4). Nous rappelons qu'à chaque élément de la liste $G(i)$ (resp. $D(j)$) sont associés les trois attributs suivants :

- $G(i).x$ (resp. $D(j).x$) : l'abscisse du point de contour dans l'image gauche (resp. droite).
- $G(i).pente$ (resp. $D(j).pente$) : la nature de la pente à l'abscisse du point (front montant ou descendant).
- $G(i).d(x)$ (resp. $D(j).d(x)$) : la valeur de l'amplitude du gradient.

Nous disposons également de la liste objet $Obj(m)$ qui contient les bandes blanches détectées à l'instant $t-1$. Comme nous serons amenés à comparer des listes obtenues à des instants différents, nous la notons $Obj(m, t-1)$ pour faire intervenir la notion de temps.

6-5.2. Recherche des bandes blanches dans une fenêtre

La première étape consiste à déterminer pour chaque bande de la liste $Obj(m, t-1)$ une fenêtre dans laquelle il existe une forte probabilité de la retrouver à l'instant t . Cette opération s'effectue dans chaque image. La notation employée ne nous permettant pas de rester dans un cas général, nous détaillons la procédure de suivi dans l'image gauche, c'est à dire dans le repère (o_g, x_g) . Cependant elle s'applique de la même manière dans l'image droite, il suffit de remplacer la lettre G par la lettre D .

6-5.2.1. Choix de la taille de la fenêtre

La figure 6.13 illustre la stratégie employée pour définir la fenêtre de recherche.

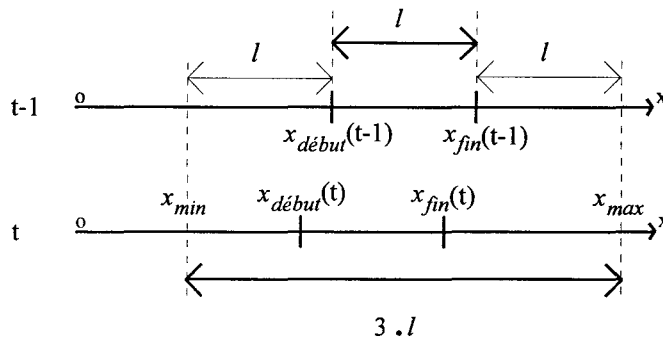


Figure 6.13 : Détermination de la fenêtre de recherche.

Soit l'objet **bande blanche** $\text{Obj}(m, t-1)$. Nous calculons d'abord l'épaisseur l de la bande dans le repère (o_g, x_g) . Elle s'obtient par l'expression suivante :

$$l = x_{fin}(t-1) - x_{début}(t-1)$$

avec :

$$x_{début}(t-1) = \text{Obj}(m, t-1).x_G.début$$

$$x_{fin}(t-1) = \text{Obj}(m, t-1).x_G.fin$$

On détermine ensuite les abscisses x_{min} et x_{max} grâce aux expressions :

$$x_{min} = x_{début}(t-1) - l$$

$$x_{max} = x_{fin}(t-1) + l$$

Ces abscisses permettent de définir une fenêtre de largeur $3.l$. La taille de celle-ci autorise ainsi dans le repère (o_g, x_g) un déplacement du profil de la bande blanche entre les instants **t-1** et **t** de l pixels vers la gauche ou vers la droite. Si on suppose que la l'épaisseur du profil de la bande reste constante dans le temps, seules les bandes spatialement connexes se situent dans la fenêtre ainsi définie et peuvent être retrouvées à l'instant **t**.

Si les bandes ne sont pas spatialement connexes, la fenêtre ne permet pas de les retrouver dans le repère (o_g, x_g) à l'instant **t**. Cependant une rupture de la connexité spatiale peut signifier un changement brutal soit dû au comportement du véhicule, soit à la configuration de la route. Dans les deux cas, il est préférable de repasser en mode global de mise en correspondance afin que l'algorithme puisse refaire le point sur la situation à l'avant du véhicule. C'est pour cette raison que la taille de la fenêtre de recherche a été limitée à $3.l$.

6-5.2.2. Recherche d'un profil ressemblant à une bande blanche

La fenêtre déterminée en fonction des abscisses de la bande dans l'image **t-1** est appliquée dans l'image **t**. On recherche alors dans cette fenêtre deux points de contour successifs $G(i)$ et $G(i+1)$ satisfaisant les conditions suivantes :

$$x_{min} \leq G(i).x \leq x_{max}$$

$$G(i).pente = \text{Front Montant}$$

$$x_{min} \leq G(i+1).x \leq x_{max}$$

$$G(i+1).pente = \text{Front Descendant}$$

On cherche en fait deux primitives qui peuvent correspondre aux contours d'une bande blanche comme nous l'avons fait au § 6-4.3 lors de la détection de bande. La seule différence est que nous ne connaissons pas la taille réelle de la bande blanche puisque nous travaillons dans le repère image. Cependant, la fenêtre étant de taille limitée, le nombre de profils formés avec les primitives $G(i)$ et $G(i+1)$ satisfaisant ces contraintes est restreint.

6-5.2.3. Sélection du profil correspondant à une bande blanche

Parmi tous les profils sélectionnés dans la fenêtre, un seul peut correspondre aux contours d'une bande blanche. Pour différencier ce dernier des autres formes, on applique le module de reconnaissance des bandes blanches décrit au § 6-4.4.

Si le module de reconnaissance indique que plusieurs profils correspondent à celui d'une bande blanche, nous utilisons un dernier critère de sélection. Ce critère suppose que d'une image à l'autre l'épaisseur du profil de la bande varie très peu, voire reste constante. On associe alors à chaque profil formé à partir des primitives $G(i)$ et $G(i+1)$, l'évolution de l'épaisseur dl définie par l'expression suivante :

$$dl = | l(t) - l(t-1) |$$

où :

$$l(t) = G(i+1).x - G(i).x$$

$$l(t-1) = \text{Obj}(m, t-1).x_G.\text{fin} - \text{Obj}(m, t-1).x_G.\text{début}$$

$l(t)$ correspond à l'épaisseur du profil analysé dans l'image à l'instant **t** et $l(t-1)$ à l'épaisseur du profil de l'objet **bande blanche** dans l'image à l'instant **t-1**. On notera que l'épaisseur $l(t-1)$ correspond à l'épaisseur l calculée au § 6-5.2.1 pour déterminer la taille de la fenêtre de recherche.

Le profil présentant la valeur d_l la plus faible est alors considéré comme le profil de la bande blanche dans le repère (o_g, x_g) à l'instant t .

6-5.2.4. Conclusion sur la recherche des bandes blanches

Pour chaque bande blanche $\text{Obj}(m, t-1)$, on recherche à l'instant t , dans l'image gauche et l'image droite, un profil correspondant à celui d'une bande blanche. Trois possibilités peuvent se présenter :

- Aucun profil n'est trouvé ni dans l'image gauche ni dans l'image droite. La bande blanche a donc disparu à l'instant t (Cf. § 6-5.5).

Si aucun couple des primitives n'a été trouvé, l'objet "**bande blanche**" $\text{Obj}(m, t-1)$ devient alors l'objet "**bande blanche disparue**" $\text{Obj}(m', t)$ en effectuant la mise à jour de l'attribut suivant :

1. $\text{Obj}(m', t).\text{type} = \text{"bande blanche disparue"}$

Le cas des bandes blanches disparues est traité au § 6-5.5.

- Un seul profil est trouvé dans l'une des deux images. Nous considérons qu'il s'agit d'un phénomène d'occlusion. Nous traitons ce cas au § 6-5.4.
- Un profil est trouvé dans chaque image. On considère alors que la bande blanche $\text{Obj}(m, t-1)$ détectée à l'instant $t-1$ a été retrouvée à l'instant t .

Soient les couples de primitives $\{G(i), G(i+1)\}$ et $\{D(j), D(j+1)\}$ définissant respectivement la bande blanche dans l'image gauche et l'image droite à l'instant t . Grâce à la procédure de suivi, les paires $\{G(i), D(j)\}$ et $\{G(i+1), D(j+1)\}$ sont considérées comme appariées et les abscisses de ces points de contour constituent les nouvelles coordonnées de la bande blanche dans les repères (o_g, x_g) et (o_d, x_d) . L'objet "**bande blanche**" $\text{Obj}(m, t-1)$ devient alors l'objet "**bande blanche**" $\text{Obj}(m', t)$ en effectuant une mise à jour des attributs suivants :

1. $\text{Obj}(m', t).\text{type} = \text{"bande blanche"}$
2. $\text{Obj}(m', t).x_G.\text{début} = G(i).x$
3. $\text{Obj}(m', t).x_G.\text{fin} = G(i+1).x$
4. $\text{Obj}(m', t).x_D.\text{début} = D(j).x$
5. $\text{Obj}(m', t).x_D.\text{fin} = D(j+1).x$

6-5.3. Calcul des coordonnées dans le repère (O, X, Z)

La recherche des bandes blanches dans une fenêtre détermine les nouvelles coordonnées des bandes blanches dans les repères (o_g, x_g) et (o_d, x_d) à l'instant t . L'étape suivante consiste à calculer les coordonnées de chaque bande retrouvée dans le repère (O, X, Z) .

La procédure de triangulation géométrique (Cf. § 5-7) nous permet de réaliser cette opération. Pour chaque objet "**bande blanche**" $\text{Obj}(m, t)$, les attributs $\text{Obj}(m, t).x_G.\text{début}$ et $\text{Obj}(m, t).x_D.\text{début}$ correspondant aux abscisses du front montant de la bande respectivement dans l'image gauche et l'image droite, servent à calculer les coordonnées position_1 et distance_1 du bord gauche de la bande blanche dans le repère (O, X, Z) . De même, nous déterminons les coordonnées position_2 et distance_2 du bord droit de la bande blanche à partir des attributs $\text{Obj}(m, t).x_G.\text{fin}$ et $\text{Obj}(m, t).x_D.\text{fin}$.

On réalise alors pour l'objet bande blanche $\text{Obj}(m, t)$ la mise à jour des attributs suivants :

6. $\text{Obj}(m, t).\text{position}.\text{début} = \text{position}_1$
7. $\text{Obj}(m, t).\text{position}.\text{fin} = \text{position}_2$
8. $\text{Obj}(m, t).\text{distance}.\text{début} = \text{Min}(\text{distance}_1, \text{distance}_2)$
9. $\text{Obj}(m, t).\text{distance}.\text{fin} = \text{Max}(\text{distance}_1, \text{distance}_2)$

Chaque bande blanche retrouvée à l'instant t est ainsi parfaitement définie dans les trois repères de travail (o_g, x_g) , (o_d, x_d) et (O, X, Z) .

Pour s'assurer que les objets retrouvés correspondent bien à une bande blanche, une dernière vérification des attributs de position et de distance est réalisée. En effet, la largeur des bandes doit être comprise entre 6 cm et 50 cm pour correspondre aux bandes normalisées. De même, les distances de chaque extrémité d'une bande doivent être cohérentes entre elles.

6-5.4. Gestion partielle des occlusions

La procédure de suivi nous permet également de gérer partiellement les occlusions.

Nous avons vu au § 3-6.1.1 que les occlusions se produisent quand un objet vient masquer un autre objet à l'une des caméras. L'objet masqué n'engendre alors des primitives que dans l'une des images. Toute tentative d'appariement de ces primitives est inutile puisqu'elles n'ont pas de correspondants dans l'autre image. Pour éviter toute erreur de mise en correspondance, il est préférable d'éliminer ces primitives de la liste des primitives à appairer.

Lorsque la procédure de suivi retrouve la bande blanche dans l'une des images mais pas dans l'autre (Cf. § 6-5.2.4), on peut supposer qu'un objet empêche la caméra de la voir et qu'il s'agit donc d'une occlusion. Pour résoudre ce problème, nous éliminons de la liste des primitives les points de contour qui définissent la bande présente. De cette manière aucune tentative d'appariement ne peut être effectuée.

Dans les explications ci-dessous nous considérons que l'occlusion se produit sur l'image droite. Cependant le raisonnement est identique lorsque celle-ci se produit sur l'image gauche. Il faut aussi noter que plusieurs phénomènes d'occlusion peuvent apparaître dans l'image gauche, dans celle de droite ou simultanément dans les deux images.

Soit le couple de primitives $\{G(i), G(i+1)\}$ correspondant au profil d'une bande blanche retrouvé dans l'image gauche à l'instant t . Aucun profil n'ayant été trouvé dans l'image droite, on considère qu'il s'agit d'un cas d'occlusion. On élimine alors les primitives $G(i)$ et $G(i+1)$ de la liste $G(i)$ avec $i = 0, 1, \dots, N$ qui devient la liste $G(i')$ avec $i' = 0, 1, \dots, N'$ avec $N' = N-2$. On effectue cette opération pour tous les cas d'occlusion rencontrés à l'instant t . Ainsi la procédure de mise en correspondance ne tente d'apparier que les primitives se trouvant dans les listes $G(i')$ et $D(j')$.

Le phénomène d'occlusion pouvant se reproduire dans plusieurs images successives, l'objet **"bande blanche"** $Obj(m, t-1)$ est considéré comme partiellement disparu. Il constitue alors l'objet $Obj(m', t)$ dont les attributs sont mis à jour selon le schéma suivant :

1. $Obj(m', t).type = \text{"bande blanche partiellement disparue"}$
2. $Obj(m', t).x_G.début = G(i).x$
3. $Obj(m', t).x_G.fin = G(i+1).x$

On conserve ainsi en mémoire la trace de bande blanche $Obj(m, t-1)$ dont les nouvelles coordonnées dans le repère image (o_g, x_g) sont connues. Comme les coordonnées dans le repère (o_d, x_d) n'ont pu être déterminées, on garde celles de la bande à l'instant $t-1$ qui sont associées à l'objet $Obj(m, t-1)$.

A l'instant $t+1$, on tente à nouveau de retrouver la bande blanche dans les deux images comme nous l'avons expliqué au § 6-5. Tant que le phénomène d'occlusion persiste, l'objet **"bande blanche partiellement disparue"** est gardé en mémoire. Dans le meilleur des cas, la bande blanche réapparaît dans les deux images quelques instants plus tard. L'objet redevient

alors un objet "**bande blanche**". Il arrive également que la bande blanche disparaisse des deux images. Ce cas est traité dans le paragraphe suivant.

6-5.5. Disparition et réapparition des bandes blanches

Si à l'instant t , toutes les bandes blanches disparaissent en même temps, l'algorithme passe en mode global de mise en correspondance. Cependant il arrive souvent qu'une bande disparaisse alors que les autres sont toujours présentes. En effet, la signalisation de certaines routes est composée de bandes discontinues, sur d'autres, la peinture est effacée. Dans de telles situations, il est impossible de suivre correctement toutes les bandes blanches. Cependant le mode local de mise en correspondance reste activé.

Lorsqu'une bande blanche n'est retrouvée à l'instant t dans aucune des images, l'objet "**bande blanche**" $\text{Obj}(m, t-1)$ est considéré comme disparu; il constitue alors l'objet $\text{Obj}(m', t)$ dont l'attribut de type devient :

1. $\text{Obj}(m', t).\text{type} = \text{"bande blanche disparue"}$

et dont les autres attributs sont ceux de l'objet $\text{Obj}(m, t-1)$.

Pour envisager les cas de disparitions temporaires, la procédure de suivi ne se contente pas à l'instant t de retrouver les objets "**bande blanche**" présents à l'instant $t-1$ et les objets "**bande blanche partiellement disparue**" lorsque des phénomènes d'occlusion se sont produits. La procédure prend également en compte les bandes blanches disparues pour déterminer si elles sont réapparues à l'instant t . Dans l'affirmative, l'objet "**bande blanche disparue**" redevient un objet "**bande blanche**". Dans le cas contraire, l'objet reste marqué en tant que "**bande blanche disparue**". L'algorithme essaiera alors de retrouver la bande dans les images acquises à l'instant suivant.

Les objets disparus peuvent, au bout d'un certain temps, devenir très nombreux. Pour éviter de conserver inutilement ces objets en mémoire, nous les dotons d'une durée de vie qui décroît à chaque fois que la procédure de suivi tente de les retrouver. Si, après un certain nombre de tentative, les objets "**bande blanche disparue**" ne sont pas réapparues, ils sont définitivement éliminés de la liste objet.

La notion de durée de vie des objets est traitée dans le § 7-3.2 qui est consacré au suivi des objets.

6-6. Détermination de la zone de danger

Lorsque les coordonnées des bandes blanches sont connues, elles servent à définir la zone de danger. Celle-ci doit être définie dans le repère (O, X, Z) pour limiter le nombre des obstacles à étudier, mais également dans les repères images (o_g, x_g) et (o_d, x_d) afin de réduire les traitements lorsque le mode local de mise en correspondance est activé.

6-6.1. Notion de zone d'intérêt

La zone de danger est parfaitement définie si les bandes matérialisant les bords de la chaussée ont été détectées. On pourrait donc se contenter de la définir dans les trois repères de travail en fonction de ces deux bandes. Cependant le nombre de bandes peintes sur la chaussée n'est pas limité à deux. Une route peut comporter deux, trois ou quatre voies avec le nombre de bandes blanches nécessaire pour les matérialiser.

En mode local de mise en correspondance, la procédure de suivi détermine les coordonnées de toutes ces bandes. Si on définit, dans les repères images, la zone de danger en fonction des abscisses des bandes délimitant la chaussée, la procédure de mise en correspondance appariera inutilement les points de contour des bandes situées dans cette zone puisque la procédure de suivi a déjà réalisé cette opération.

Pour éviter ces appariements inutiles, la zone de danger est divisée dans les repères images en *zones d'intérêt*. Comme le montre la figure 6.14, chaque zone d'intérêt est délimitée par deux bandes blanches. Ainsi si N bandes ont été détectées à un instant donné (avec $N > 1$), on définit $N-1$ zones d'intérêt ne comprenant aucune bande blanche. Par exemple, à l'instant t , deux bandes sont visibles. On ne peut donc définir qu'une seule zone d'intérêt qui d'ailleurs est identique à la zone de danger. Par contre, à l'instant $t+1$, trois bandes sont visibles simultanément. La zone de danger, qui est identique à celle définie à l'instant t , peut être divisée en deux zones d'intérêt.

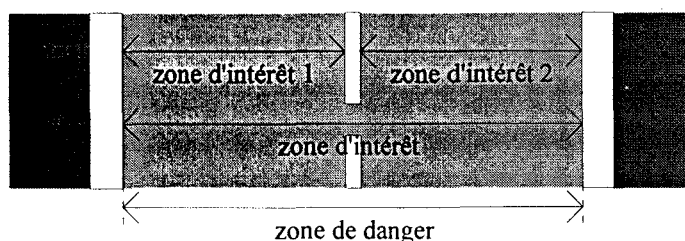


Figure 6.14 : Zone de danger et zones d'intérêt.

Il faut cependant préciser que même si elles sont liées, la zone de danger et la zone d'intérêt correspondent à deux notions différentes. La zone de danger est définie uniquement dans le repère (O, X, Z) et sert à restreindre le nombre d'obstacles à étudier. Les zones d'intérêt sont définies dans les trois repères de travail et permettent de limiter les traitements de mise en correspondance et de simplifier la reconstruction (Cf. § 7-2).

Soit la liste objet $\text{Obj}(m, t)$ composée des objets "**bande blanche**" détectés à l'instant t , avec $m = 1, \dots, M$. Nous détaillons, dans les paragraphes suivants, la technique utilisée pour définir les différentes zones à partir de cette liste.

6-6.2. Précaution à prendre

Une attention particulière doit être apportée au calcul des limites de la zone de danger. En effet, la sélection des objets que nous étudions dépend de cette zone. Si des erreurs sont commises lors de la détermination de cette zone, des obstacles dangereux peuvent être ignorés. Notre système de détection ne sera donc pas fiable.

Comme nous l'avons dit dans le paragraphe précédent, la zone de danger est parfaitement définie si les bandes matérialisant le bord de la chaussée ont été détectées. Cependant aucune information ne nous permet d'être sûr d'avoir effectivement détecté ces bandes.

Il ne faut pas négliger les détections manquées. De plus, l'algorithme peut ne trouver qu'une seule bande blanche, voire aucune.

Si nous voulons réduire le nombre d'objets à étudier, il faut impérativement définir cette zone de danger. Cependant elle ne doit être ni trop petite, ni trop grande. Dans le premier cas, des obstacles dangereux peuvent être ignorés, dans le second, le nombre d'objets à étudier peut être important ce qui augmente les temps de calculs. Nous devons donc étudier attentivement le nombre des bandes détectées ainsi que leur position par rapport au véhicule.

Des précautions identiques doivent être prise pour la définition des zones d'intérêt.

6-6.3. Détermination de la zone de danger

La zone de danger est définie dans le repère (O, X, Z) . Comme nous l'avons vu au § 6-3.2, la zone de danger est définie en fonction de trois paramètres X_{\min} , X_{\max} et Z_{\max} (Cf. Figure 6.2), mais à ce stade des traitements, seuls les paramètres X_{\min} et X_{\max} sont déterminés.

6-6.3.1. Aucune bande blanche n'a été détectée

Dans cette situation, nous définissons une zone de danger théorique en affectant aux paramètres X_{min} et X_{max} des valeurs par défaut suivantes :

$$X_{min} = - 5 \text{ m}$$

$$X_{max} = 5 \text{ m}$$

c'est à dire 5 m à gauche et 5 m à droite du véhicule (Cf. Figure 6.15). Nous surveillons ainsi une zone de 10 m de large. Cette zone peut paraître importante, mais elle nous assure de prendre en compte tous les obstacles pouvant présenter un danger pour le véhicule.

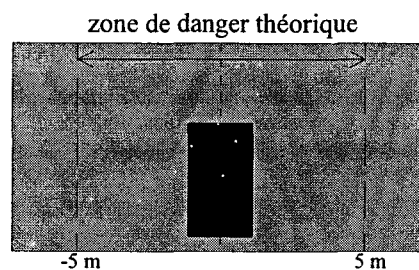


Figure 6.15 : Situation où aucune bande n'a été détectée.

6-6.3.2. Une seule bande blanche a été détectée

Cette situation indique que notre algorithme n'a pas réussi à détecter toutes les bandes présentes sur la chaussée. Les routes possédant une seule bande sont en effet très rares. Pour définir la zone de danger, plusieurs situations doivent être envisagées. Celles-ci sont représentées sur la figure 6.16.

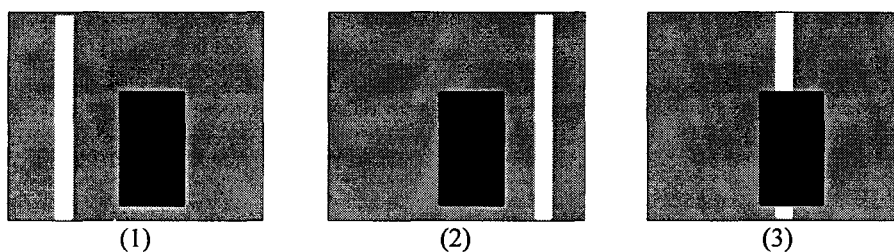


Figure 6.16 : Situations où une seule bande a été détectée.

Soit $Obj(1, t)$ la bande blanche détectée.

Situation 1: La bande se trouve sur la gauche du véhicule. La zone de danger est définie de la façon suivante :

$$X_{min} = Obj(1, t).position.fin$$

$$X_{max} = 5 \text{ m}$$

La valeur X_{min} est déterminée en fonction de la position réelle de la bande. Par contre, on affecte à la limite droite X_{max} la valeur par défaut de 5 m à droite du véhicule.

Situation 2: La bande se trouve sur la droite du véhicule. La zone de danger est alors définie de la façon suivante :

$$X_{min} = - 5 \text{ m}$$

$$X_{max} = \text{Obj}(1, t).\text{position.début}$$

Ce cas est traité comme le précédent mais en attribuant une limite par défaut de 5 m à gauche du véhicule.

Situation 3: Dans cette situation, on considère que les bandes délimitant la route n'ont pas été détectées et que la bande trouvée correspond à la bande centrale. On suppose donc que la voiture ne sort pas de la route. On affecte alors aux paramètres X_{min} et X_{max} les valeurs par défaut :

$$X_{min} = - 5 \text{ m}$$

$$X_{max} = 5 \text{ m}$$

6-6.3.3. Au moins deux bandes blanches ont été détectées

La figure 6.17 présente les différentes situations rencontrées pour une route à deux voies, mais elles peuvent être facilement étendues aux routes à trois ou quatre voies. Nous définissons d'ailleurs la zone de danger dans un cas général.

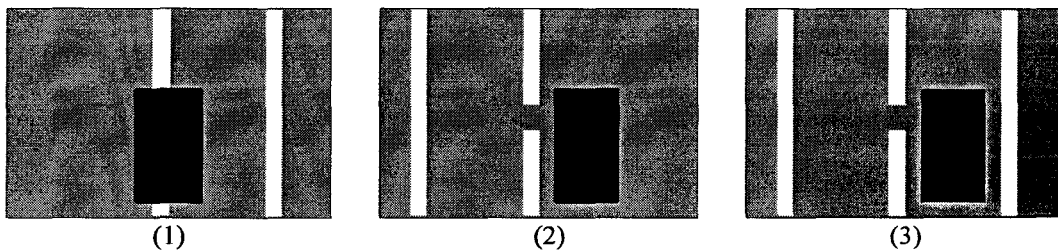


Figure 6.17 : Situations où au moins deux bandes ont été détectées.

Soit la liste objet $\text{Obj}(m, t)$ contenant les bandes blanches détectées avec $m = 1, \dots, M$. Nous rappelons que les objets sont rangés dans cette liste en fonction de leurs positions croissantes dans le repère (O, X, Z) .

Situation 1: Aucune bande blanche n'a été détectée à gauche du véhicule. Nous définissons alors la zone de danger de la façon suivante :

$$X_{min} = -5 \text{ m}$$

$$X_{max} = \text{Obj}(M, t).\text{position.début}$$

On affecte à la limite gauche X_{min} la valeur par défaut de 5 m à gauche du véhicule. La limite droite X_{max} est, quant à elle, définie par la bande située la plus à droite dans le repère (O, X, Z), c'est à dire la dernière de la liste objet.

Situation 2: Aucune bande blanche n'a été détectée à droite du véhicule. Nous définissons alors la zone de danger de la façon suivante :

$$X_{min} = \text{Obj}(1, t).\text{position.fin}$$

$$X_{max} = 5 \text{ m}$$

Dans cette situation, la limite gauche X_{min} dépend de la position de la bande située la plus à gauche dans le repère (O, X, Z), c'est à dire la première de la liste objet. Le paramètre X_{max} reçoit la valeur par défaut de 5 m à droite du véhicule.

Situation 3: Au moins une bande blanche a été détectée de chaque coté du véhicule. La zone de danger est donc entièrement définie par les positions de ces bandes.

$$X_{min} = \text{Obj}(1, t).\text{position.fin}$$

$$X_{max} = \text{Obj}(M, t).\text{position.début}$$

Les limites X_{min} et X_{max} correspondent respectivement à la première et la dernière bande de la liste objet.

Ce cas est le plus intéressant car il indique que l'algorithme a réussi à déterminer correctement la configuration de la route. Il permet ainsi de définir une zone de danger située exclusivement sur la chaussée. L'algorithme peut donc éliminer tous les objets sans danger situés sur les bas-côtés et se consacrer uniquement aux véritables obstacles situés sur la chaussée.

6-6.4. Détermination des zones d'intérêt

Ces zones ne sont définies que si des bandes blanches ont été détectées. Ces zones sont importantes car elles permettent de limiter les traitements de mise en correspondance et de simplifier la reconstruction tridimensionnelle.

6-6.4.1. Notation utilisée pour les zones d'intérêt

La mise en correspondance étant réalisée dans le repère image, chaque zone d'intérêt est définie dans les repères (o_g, x_g) et (o_d, x_d) . Elle permet ainsi d'initialiser l'algorithme récursif de mise en correspondance hiérarchique dans un plan de recherche précis (Cf. § 5-6). Comme

plusieurs zones d'intérêt peuvent être définies, on utilise l'expression d'*initialisation multi-zones*, chaque zone définissant un plan de recherche à explorer.

Soit $Zi(k)$ avec $k = 1, 2, \dots, K$, la liste des zones d'intérêt où K est le nombre de ces zones d'intérêt. La zone $Zi(k)$ est définie par les attributs suivants :

$Zi(k).x_{o_g}$: abscisse dans le repère (o_g, x_g) de l'extrémité supérieure gauche (*origine*).

$Zi(k).x_{f_g}$: abscisse dans le repère (o_g, x_g) de l'extrémité inférieure droite (*fin*).

$Zi(k).x_{o_d}$: abscisse dans le repère (o_d, x_d) de l'origine.

$Zi(k).x_{f_d}$: abscisse dans le repère (o_d, x_d) de la fin.

$Zi(k).position.origine$: position de l'extrémité origine dans le repère (O, X, Z) .

$Zi(k).position.fin$: position de l'extrémité fin dans le repère (O, X, Z) .

Ces attributs définissent ainsi au niveau 1 de mise en correspondance, l'ensemble des plans de recherche $PR[(Zi(k).x_{o_g}, Zi(k).x_{o_d}); (Zi(k).x_{f_g}, Zi(k).x_{f_d})]$ avec $k = 1, \dots, K$, où K représente le nombre de plans de recherche. Dans l'exemple de la figure 6.4, une seule zone d'intérêt a été définie.

L'utilisation des attributs $Zi(k).position.origine$ et $Zi(k).position.fin$ est précisée au § 6-7.4.

La notion de zone de danger étant liée à celle des zones d'intérêt, nous nous trouvons confrontés aux mêmes situations que précédemment pour les définir.

6-6.4.2. Une seule bande blanche a été détectée

Les situations traitées correspondent à celles représentées sur la figure 6.16.

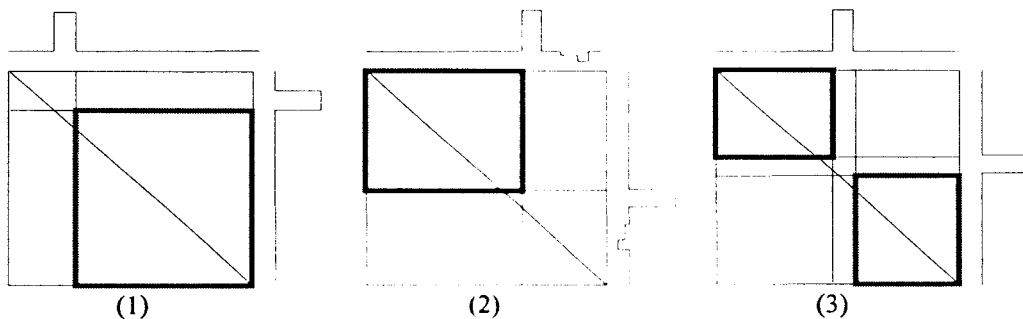


Figure 6.18 : Plans de recherche définis pour chaque situation de la figure 6.16.

Soit $Obj(1, t)$ la bande blanche détectée.

Situation 1: Une seule zone d'intérêt est définie de la façon suivante :

$$Zi(1).x_{o_g} = Obj(1, t).x_G.fin$$

$$Zi(1).x_{f_g} = G(N)$$

$$Zi(1).x_{o_d} = Obj(1, t).x_D.fin$$

$$Zi(1).x_{f_d} = D(M_D)$$

$$Zi(1).position.origine = Zmin = Obj(1, t).position.fin.$$

$$Zi(1).position.fin = Zmax = 5 \text{ m}$$

Pour éviter toute confusion entre les indices, le dernier pixel de l'image droite que nous notions D(M) est dorénavant noté D(M_D).

Cette zone d'intérêt définit le plan de recherche de la figure 6.18 (1). On note que l'extrémité inférieure droite du plan correspond au dernier pixel de chaque image, c'est à dire G(N) et D(M_D), et que les valeurs affectées aux paramètres de positions *origine* et *fin* correspondent à celles de la zone de danger (Cf. § 6-6.3.2 - Situation 1).

Situation 2: Une seule zone d'intérêt est définie de la façon suivante :

$$Zi(1).x_{o_g} = G(0)$$

$$Zi(1).x_{f_g} = Obj(1, t).x_G.début$$

$$Zi(1).x_{o_d} = D(0)$$

$$Zi(1).x_{f_d} = Obj(1, t).x_D.début$$

$$Zi(1).position.origine = Zmin = - 5 \text{ m}$$

$$Zi(1).position.fin = Zmax = Obj(1, t).position.début$$

Cette zone d'intérêt définit le plan de recherche de la figure 6.18 (2). On note que l'extrémité supérieure gauche du plan correspond au premier pixel de chaque image et que les valeurs affectées aux positions correspondent également à celles de la zone de danger. (Cf. § 6-6.3.2 - Situation 2).

Situation 3: Deux zones d'intérêt sont définies dans ce cas :

$$Zi(1).x_{o_g} = G(0)$$

$$Zi(1).x_{f_g} = Obj(1, t).x_G.début$$

$$Zi(1).x_{o_d} = D(0)$$

$$Zi(1).x_{f_d} = Obj(1, t).x_D.début$$

$$Zi(1).position.origine = Zmin = - 5 \text{ m}$$

$$Zi(1).position.fin = Obj(1, t).position.début$$

$$Zi(2).x_{o_g} = Obj(1, t).x_G.fin$$

$$Zi(2).x_{f_g} = G(N)$$

$$Zi(2).x_{o_d} = Obj(1, t).x_D.fin$$

$$Zi(2).x_{f_d} = D(M_D)$$

$$Zi(2).position.origine = Obj(1, t).position.fin$$

$$Zi(2).position.fin = Zmax = 5 \text{ m}$$

Ces zones d'intérêt définissent les plans de recherche de la figure 6.18 (3). La première zone est déterminée en fonction des premiers pixels de chaque image et les abscisses du bord gauche de la bande dans les repères images. La seconde zone est délimitée par les abscisses du bord droit de la bande et les dernier pixels de chaque image. Pour les paramètres de position, chaque zone est définie en partie par l'une des limites de la zone de danger et de la position d'un des bords de la bande dans le repère (O, X, Z).

6-6.4.3. Au moins deux bandes blanches ont été détectées

Les situations traitées correspondent à celles représentées sur la figure 6.17. Les schémas de la figure 6.19 traitent le cas d'une route à deux voies où les bandes blanches ne sont pas toujours toutes détectées (situations 1 et 2). Nous définissons cependant les zones d'intérêt dans un cas général.

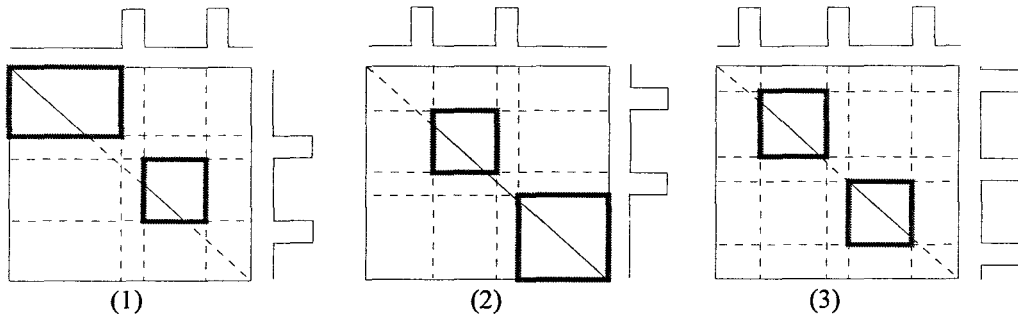


Figure 6.19 : Plans de recherche définis pour chaque situation de la figure 6.17.

Soit la liste objet $Obj(m, t)$ composée des bandes blanches détectées avec $m = 1, \dots, M$, où M , avec $M \geq 2$, est le nombre de bandes blanches. Nous rappelons que les objets sont rangés dans cette liste en fonction de leurs positions croissantes dans le repère (O, X, Z).

Situation 1: Aucune bande blanche n'a été détectée à gauche du véhicule. A partir des M bandes blanches, nous définissons K zones d'intérêt avec $K = M$ de la façon suivante :

Pour la première zone

- $Zi(1).x_{og} = G(0)$
- $Zi(1).x_{fg} = Obj(1, t).x_G.début$
- $Zi(1).x_{od} = D(0)$
- $Zi(1).x_{fd} = Obj(1, t).x_D.début$
- $Zi(1).position.origine = Zmin$
- $Zi(1).position.fin = Obj(1, t).position.début$

Pour les $K-1$ zones suivantes

- $Zi(k).x_{og} = Obj(m, t).x_G.fin$
 - $Zi(k).x_{fg} = Obj(m+1, t).x_G.début$
 - $Zi(k).x_{od} = Obj(m, t).x_D.fin$
 - $Zi(k).x_{fd} = Obj(m+1, t).x_D.début$
 - $Zi(k).position.origine = Obj(m, t).position.fin$
 - $Zi(k).position.fin = Obj(m+1, t).position.début$
- avec $m = 1, \dots, M-1$

Ces zones d'intérêt définissent les plans de recherche de la figure 6.19 (1).

Situation 2: Aucune bande blanche n'a été détectée à droite du véhicule. Comme précédemment, nous définissons K zones d'intérêt à partir des M bandes blanches, avec $K = M$:

Pour les K-1 premières zones

$$Zi(k).x_{o_g} = Obj(m, t).x_G.fin$$

$$Zi(k).x_{f_g} = Obj(m+1, t).x_G.début$$

$$Zi(k).x_{o_d} = Obj(m, t).x_D.fin$$

$$Zi(k).x_{f_d} = Obj(m+1, t).x_D.début$$

$$Zi(k).position.origine = Obj(m, t).position.fin$$

$$Zi(k).position.fin = Obj(m+1, t).position.début$$

avec $m = 1, \dots, M-1$

Pour la dernière zone

$$Zi(K).x_{o_g} = Obj(M, t).x_G.fin$$

$$Zi(K).x_{f_g} = G(N)$$

$$Zi(K).x_{o_d} = Obj(M, t).x_D.fin$$

$$Zi(K).x_{f_d} = D(M_D)$$

$$Zi(K).position.origine = Obj(M, t).position.fin$$

$$Zi(K).position.fin = Zmax$$

Ces zones d'intérêt définissent les plans de recherche de la figure 6.19 (2).

Situation 3: Au moins une bande blanche a été détectée de chaque côté du véhicule. Les zones d'intérêts peuvent être uniquement définies en fonction des bandes blanches. A partir des M bandes blanches, on définit K zones d'intérêt avec $K = M-1$. La zone k est définie de la façon suivante :

$$Zi(k).x_{o_g} = Obj(k, t).x_G.fin$$

$$Zi(k).x_{f_g} = Obj(k+1, t).x_G.début$$

$$Zi(k).x_{o_d} = Obj(k, t).x_D.fin$$

$$Zi(k).x_{f_d} = Obj(k+1, t).x_D.début$$

$$Zi(k).position.origine = Obj(k, t).position.fin$$

$$Zi(k).position.fin = Obj(k+1, t).position.début$$

avec $k = 1, \dots, M-1$

Ces zones d'intérêt définissent les plans de recherche de la figure 6.19 (3).

Comme pour la zone de danger, ce cas est le plus intéressant. Les nombreuses primitives provenant des objets situés sur les bas-côtés comme les herbes ou les arbres sont ici ignorées. La procédure de mise en correspondance ne tente d'apparier que les primitives engendrées par les objets présents sur la chaussée. Les temps de traitement sont ainsi considérablement réduits. Il en est de même des risques d'erreurs puisque le nombre de primitives à apparier est généralement très faible.

6-7. Mise en correspondance bi-mode

Le fonctionnement général de notre procédure de mise en correspondance bi-mode est décrit sous la forme de schémas.

6-7.1. Mode global

La figure 6.20 représente le mode global de mise en correspondance. Ce mode est utilisé quand l'algorithme ne dispose d'aucune information sur la nature de la chaussée. Celui-ci est composé de 3 modules.

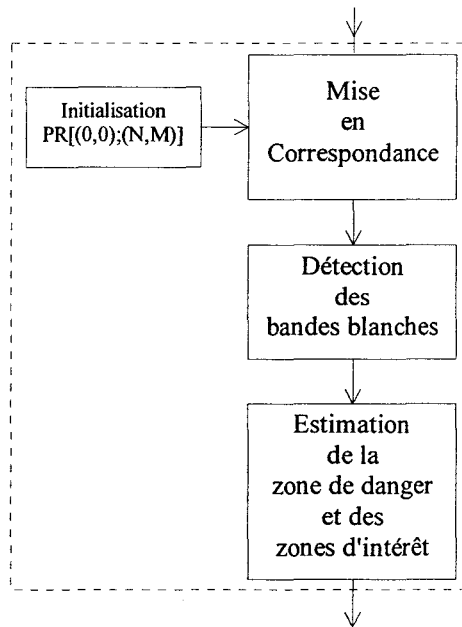


Figure 6.20 : Mode global de mise en correspondance.

1. Le premier module est chargé de réaliser la mise en correspondance de toutes les primitives extraites de l'image gauche et de l'image droite. La procédure récursive est initialisée au niveau 1 dans le plan de recherche $PR[(0,0); (N,M)]$.
2. A partir de la liste stéréo fournie par le premier module, le second recherche les bandes blanches présentes sur la chaussée.
3. On détermine ensuite la zone de danger et éventuellement les zones d'intérêt en fonction du résultat fourni par le second module.

6-7.2. Mode local

La figure 6.21 représente le mode local de mise en correspondance. Celui-ci est utilisé quand l'algorithme dispose d'informations concernant la signalisation horizontale peinte sur la chaussée. Ce mode comprend également 3 modules.

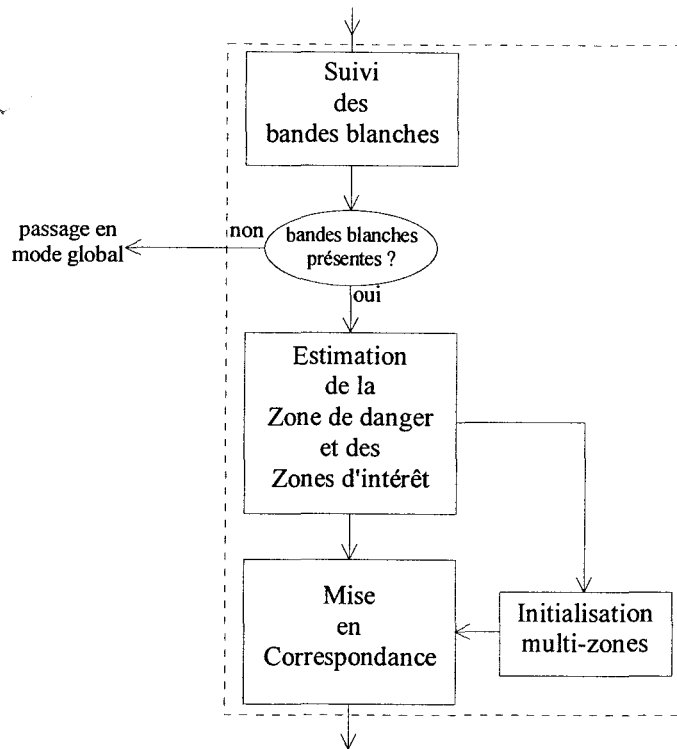


Figure 6.21 : Mode local de mise en correspondance.

1. Le module de suivi détermine les nouvelles coordonnées des bandes blanches dans les repères images et calcule ensuite leurs coordonnées dans le repère (O, X, Z). Celui-ci gère également les phénomènes d'occlusion lorsque ceux-ci se produisent.
2. La zone de danger et les zones d'intérêt sont réactualisées en fonction des nouvelles coordonnées des bandes blanches. Les zones d'intérêt constituent alors les paramètres d'initialisation du troisième module dédié à la procédure de la mise en correspondance.
3. Grâce à l'initialisation multi-zones, la procédure récursive de mise en correspondance est appliquée localement dans chaque zone d'intérêt. Ainsi, seules les primitives engendrées par les objets situés sur la chaussée sont traitées.

La figure 6.21 indique l'existence d'une liaison du mode local vers le mode global. L'algorithme utilise cette possibilité quand le module de suivi ne retrouve aucune bande blanche. Comme nous l'avons dit au § 6-5.2.1, une disparition brusque des bandes blanches indique un changement important dans le comportement du véhicule ou de la configuration de la route. Il est donc judicieux de ne plus se fier aux informations précédemment acquises et de repasser sans attendre en mode global pour refaire une analyse complète de la situation.

6-7.3. Mise en correspondance bi-mode

La figure 6.22 décrit schématiquement le processus complet de mise en correspondance bi-mode en indiquant les liaisons existantes entre les deux modes détaillés précédemment.

Le choix du mode de mise en correspondance se fait en fonction des détections faites à l'instant $t-1$. Si des bandes blanches sont présentes l'algorithme utilise le mode local. Dans le cas contraire, il fait appel au mode global.

Lors de la disparition complète des objets de référence, c'est à dire les bandes blanches, la liaison du mode local vers le mode global permet de refaire le point sur la situation et de prendre rapidement en compte les changements soudains intervenants à l'avant du véhicule.

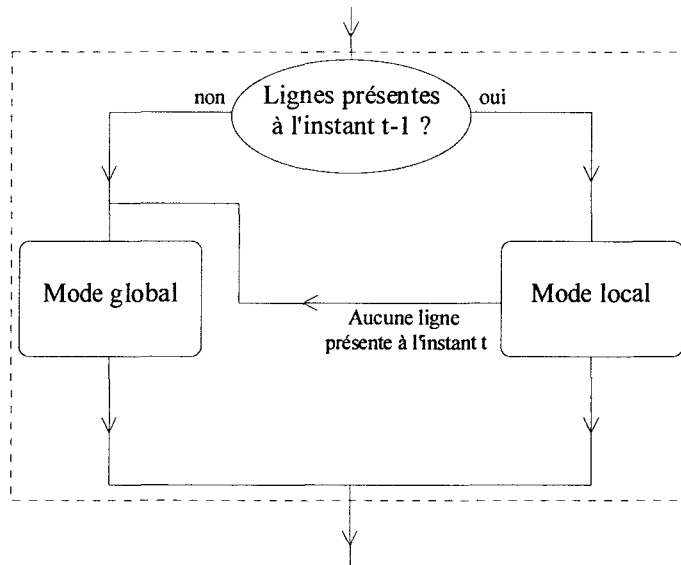


Figure 6.22 : Mise en correspondance bi-mode.

6-7.4. Contrainte de cohérence spatiale

Lorsque le mode local de mise en correspondance est activé, nous utilisons une contrainte supplémentaire pour vérifier les résultats obtenus par le module d'appariement élémentaire hiérarchique (Cf. § 5-6.4.3).

La contrainte de cohérence spatiale est basée sur la contrainte d'ordre (Cf. § 5-3.2) mais elle est appliquée dans le repère (O, X, Z) . En effet, si les primitives respectent la contrainte d'ordre dans l'image gauche et l'image droite, c'est à dire dans les repères images (o_g, x_g) et (o_d, x_d) , les points qui ont engendré ces primitives doivent obligatoirement respecter le même ordre selon l'axe (O, X) du repère (O, X, Z) . Dans le cas contraire, il s'agit d'une erreur d'appariement et la paire formée doit être éliminée.

Soit le niveau 1 de mise en correspondance dans une zone d'intérêt définie par le plan de recherche $PR[(Zi(k).x_{o_g}, Zi(k).x_{o_d}); (Zi(k).x_{f_g}, Zi(k).x_{f_d})]$. La procédure de mise en correspondance hiérarchique, appliquée dans ce plan, fournit une liste de paires $\{\{C_G^1(p), C_D^1(p)\}, p = 1, \dots, P^1\}$ où P^1 est le nombre de paires de primitives appariées. Grâce à la procédure de triangulation géométrique, on détermine ensuite pour chaque paire appariée $\{C_G^1(p), C_D^1(p)\}$, la position dans le repère (O, X, Z) du point ayant engendré ces primitives. Nous notons cette position $pos[\{C_G^1(p), C_D^1(p)\}]$.

Nous avons vu au § 6-6.4.1 que les attributs $Zi(k).position.origine$ et $Zi(k).position.fin$ ont été associés à chaque zone d'intérêt $Zi(k)$. Ces attributs permettent de situer les extrémités du plan de recherche dans le repère (O, X, Z).

L'application de la contrainte de cohérence spatiale consiste à contrôler que la position $pos[\{C_G^1(p), C_D^1(p)\}]$ calculée pour chaque paire de primitives appariées vérifie la relation :

$$Zi(k).position.origine \leq pos[\{C_G^1(p), C_D^1(p)\}] \leq Zi(k).position.fin$$

On forme ainsi une nouvelle liste $\{\{C_G^1(p), C_D^1(p)\}, p = 1, \dots, P^1\}$ où P^1 est le nombre de paires appariées vérifiant la contrainte de cohérence spatiale. Afin de pouvoir aisément créer les nouveaux plans de recherche au niveau suivant, on ajoute à la liste des résultats, les paires définissant les extrémités du plan de recherche comme nous l'avons décrit au § 5-6.4.3.

Dans un cas général, on obtient alors au niveau n de recherche la liste finale définie par $\{\{C_G^n(k), C_D^n(k)\}, k = 0, 1, \dots, X^n\}$ avec $X^n = P^{n-1} + 1$, où P^{n-1} est le nombre de paires appariées vérifiant la contrainte de cohérence spatiale et X^n le nombre de paires constituant la liste après l'ajout des extrémités du plan de recherche. On associe, de plus, à chaque paire $\{C_G^n(k), C_D^n(k)\}$ la position $pos[\{C_G^n(k), C_D^n(k)\}]$ déterminée par la procédure de triangulation.

Ainsi, au niveau $n+1$ de recherche, on peut appliquer dans chaque plan défini par $PR[(C_G^n(k), C_D^n(k)); (C_G^n(k+1), C_D^n(k+1))]$ avec $k = 0, 1, \dots, X^n - 1$ la contrainte de cohérence spatiale. Il suffit alors de conserver les paires $\{C_G^{n+1}(p), C_D^{n+1}(p)\}$ vérifiant la relation :

$$pos[\{C_G^n(k), C_D^n(k)\}] \leq pos[\{C_G^{n+1}(p), C_D^{n+1}(p)\}] \leq pos[\{C_G^n(k+1), C_D^n(k+1)\}]$$

Il faut préciser que les paires ne vérifiant pas cette contrainte sont éliminées de la liste résultat au niveau $n+1$. Cependant les primitives constituant ces paires ne sont pas éliminées des listes de primitives $G(i)$ et $D(j)$. Elles peuvent donc être appariées au niveau $n+2$.

6-8. Conclusion sur la mise en correspondance bi-mode

La conduite automobile est souvent facilitée grâce à la signalisation horizontale. En effet, les bandes blanches permettent au conducteur de se situer sur la route et d'estimer le danger associé à la présence d'obstacles sur la chaussée.

De la même manière, la mise en correspondance bi-mode utilise les bandes blanches. Grâce aux modules de détection et de suivi des bandes, l'algorithme définit des zones d'intérêt qui accélèrent les traitements en limitant le nombre de primitives à apparier.

La procédure est également conçue pour fonctionner en l'absence de bande. Dans ce cas, les traitements sont un peu plus longs car les images sont traitées dans leur intégralité. Cependant cette alternative permet à l'algorithme de détection d'être toujours opérationnel, quelle que soit la nature de la route.

Enfin, cette procédure détermine une zone de danger. Cette zone, déterminée dans le repère associé au stéréoscope linéaire, permettra de ne prendre en compte que les objets pouvant représenter un danger pour le véhicule. L'étape suivante consiste donc à reconstruire les objets à partir de la liste stéréo fournie par la procédure de mise en correspondance bi-mode et à déterminer ceux qui représentent réellement un danger. Ces deux aspects sont traités dans le chapitre suivant.

Chapitre 7

DETECTION D'OBSTACLES A L'AVANT DES VEHICULES ROUTIERS

7-1. Introduction

Nous avons vu, au chapitre 6, que la signalisation horizontale accélère de manière significative les traitements de mise en correspondance en limitant le nombre de points de contour à apparier. Cependant, si une reconstruction partielle de la scène permet de détecter la présence des bandes blanches, elle ne peut déterminer la nature des autres objets présents face au véhicule.

En effet, la correspondance bi-mode fournit une *liste stéréo* contenant un ensemble de points dont les coordonnées dans la scène réelle sont connues. Le problème consiste maintenant à fusionner ces points afin de reconstruire les objets et de définir leur type (Cf. § 7-2).

Lorsqu'un objet a été localisé devant le véhicule, on ne peut systématiquement déclencher une alarme. Un suivi spatio-temporel est réalisé pour déterminer si celui-ci représente un danger (Cf. § 7-3).

Enfin, différents exemples tirés de situations réelles sont présentés dans ce chapitre et montrent les possibilités qu'offre notre système de détection d'obstacles (Cf. § 7-4).

Nous employons dans ce chapitre les mêmes notations et définitions que celles décrites dans le chapitre précédent (Cf. § 6-2).

7-2. Reconstruction tridimensionnelle des objets

Nous avons vu au cours du paragraphe précédent que la reconstruction des bandes blanches simplifie et accélère la procédure de mise en correspondance. Le but de ce paragraphe est la reconstruction et l'identification des autres objets présents devant le véhicule.

7-2.1. Difficulté de la reconstruction des objets

La première idée que nous avons eue pour reconstruire les autres objets était d'utiliser une démarche identique à celle employée pour la reconnaissance des bandes blanches, c'est à dire définir un profil caractéristique pour chaque type d'objet habituellement rencontré dans le contexte routier.

La figure 7.1 montre le profil d'un piéton sur plusieurs images successives extraites de la séquence "piéton" (Cf. § 2-4.3.1). L'examen des différents profils permet de faire les remarques suivantes :

- En théorie, un objet est délimité par deux contours. En pratique, on constate qu'un piéton engendre plus de deux contours.
- Si l'allure générale du profil semble la même sur les images (a) et (b), elle change complètement sur les images (c) et (d).
- Il n'existe pas de caractéristiques communes à tous ces profils, comme l'amplitude maximale ou la largeur du profil. Celles-ci évoluent d'une image à l'autre de façon incohérente.

On peut faire les mêmes remarques pour les profils de voitures ou de camions.

La conclusion qui s'impose à la suite de cette étude préliminaire est que la reconstruction des objets en fonction de leur profil n'est pas envisageable, puisque aucune caractéristique nous permet d'identifier le type d'objet rencontré.

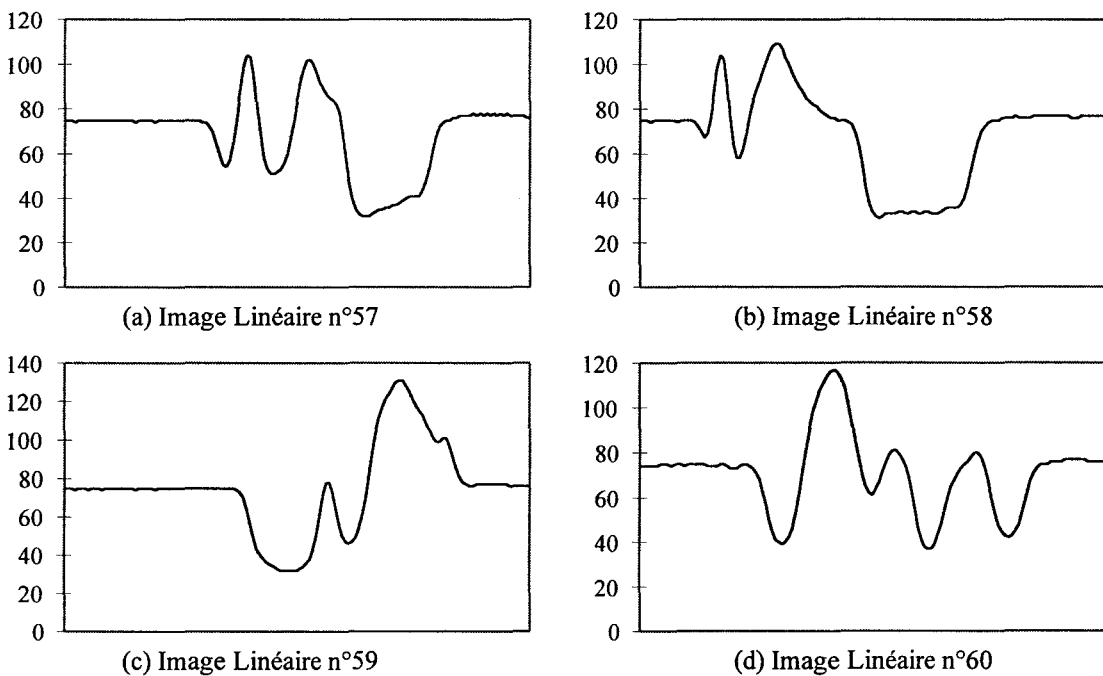


Figure 7.1 : Profil d'un piéton sur plusieurs images successives de la séquence "piéton".

La conséquence immédiate est qu'un suivi des points de contour dans les repères images, comme nous le faisons pour les bandes blanches, n'est pas réalisable. En effet, si les profils sont différents d'une image à l'autre, nous ne sommes pas sûrs de retrouver à l'instant t les points de contour formant l'objet à l'instant $t-1$. Dans le cadre de notre application, nous ne pouvons étudier l'évolution temporelle des points de contour avant leur reconstruction comme le fait Colle lors de l'analyse des scènes d'intérieur [Colle 90].

De plus, si deux objets différents sont "côte à côte", rien ne nous permet de les différencier. Par exemple, le profil de la figure 7.1 (d) peut très bien correspondre à deux objets dont les images engendrées sur le capteur sont situées à des abscisses voisines. Cependant aucun critère ne nous permet de les séparer. C'est pourquoi les phénomènes d'occlusion partielle ou totale ne peuvent être traités pour les objets autres que les bandes blanches.

Les informations dans les repères images n'étant pas suffisantes pour reconstituer les objets à l'instant t ou les suivre dans le temps, la reconstruction est donc réalisée dans le repère (O, X, Z) .

7-2.2. Sélection des points dans le repère (O, X, Z) avant reconstruction

Pour accomplir la reconstruction des objets, nous disposons de la liste stéréo $S(n)$ obtenue après la mise en correspondance bi-mode dont les éléments sont rangés selon les positions croissantes. La figure 7.2 montre un exemple de liste stéréo dont les points sont représentés dans le repère (O, X, Z) .

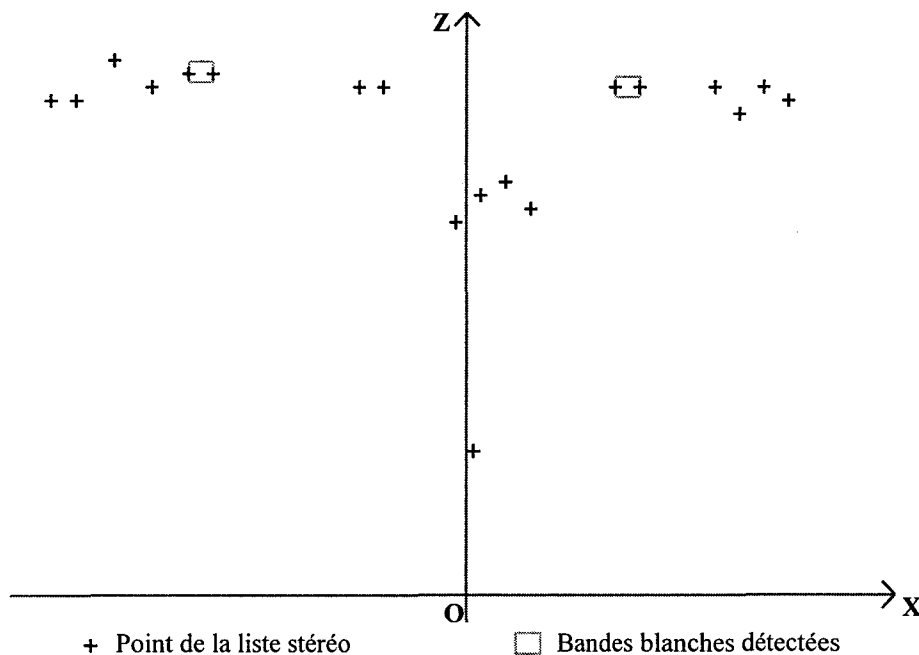


Figure 7.2 : Points de la liste stéréo dans le repère (O, X, Z) .

Il faut préciser que si des bandes blanches ont été détectées à l'instant t , les points constituant ces bandes sont éliminées de cette liste. Ainsi les points des deux bandes blanches représentées sur la figure 7.2 ne sont pas pris en compte lors de la reconstruction.

Pour simplifier la reconstruction, nous limitons le nombre de points à traiter. Cette sélection dépend du nombre de bandes blanches détectées :

- Aucune bande blanche n'a été détectée

Les points situés à 10 m à gauche ou 10 m à droite ne peuvent constituer des obstacles dangereux. Nous simplifions donc les opérations de reconstruction en ne considérant que les points situés dans la zone de danger. Nous conservons les points de la liste stéréo $S(n)$, avec $n = 1, \dots, N$, dont l'attribut de position vérifie la relation :

$$X_{\min} \leq S(n).position. \leq X_{\max}$$

On forme ainsi une liste stéréo réduite $S_1(n_1)$, avec $n_1 = 1, \dots, N_1 < N$, qui est utilisée pour la reconstruction.

Il faut noter que la sélection est faite en fonction de valeurs arbitraires. Si un objet se trouve à la limite de la zone de danger, une partie des points qui le constituent peut être éliminée. Cependant, quand, à l'instant t , un objet est mal identifié, tous les points qui le composent sont sélectionnés à l'instant $t+1$ si celui-ci est entré complètement dans la zone de danger. Dans le cas contraire, il s'agit d'un objet immobile qui se trouve à la limite de la zone de danger et qui par conséquent n'est pas dangereux.

- Au moins une bande blanche a été détectée

Lorsque des bandes blanches sont détectées, nous utilisons les zones d'intérêt pour simplifier la reconstruction. On détermine alors pour la zone d'intérêt $Z_i(k)$, la liste stéréo $S_k(n_k)$ avec $n_k = 1, \dots, N_k$, à partir des éléments de la liste $S(n)$ vérifiant la relation :

$$Z_i(k).position.origine \leq S(n).position. \leq Z_i(k).position.fin$$

La procédure de reconstruction est ensuite appliquée successivement à chaque liste stéréo $S_k(n_k)$.

Les zones d'intérêt permettent bien sûr d'éliminer les points se trouvant hors de ces zones. Cependant, le principal avantage est d'entamer la procédure de reconstruction des objets en effectuant une première classification des points de la liste stéréo $S(n)$ en plusieurs listes $S_k(n_k)$.

7-2.3. Apparition de bandes blanches

Ce paragraphe devrait logiquement se trouver dans le chapitre 6 où est traitée la détection des bandes blanches. Cependant, les problèmes relatifs à l'apparition de bandes blanches sont apparus lors de la reconstruction tridimensionnelle. Nous avons donc trouvé plus judicieux de placer ce paragraphe dans ce chapitre.

Lorsque le mode local de mise en correspondance est activé, la procédure de suivi permet de retrouver à l'instant $t+1$ les bandes présentes à l'instant t (Cf. § 6-5). Cependant une bande blanche qui n'a pas été détectée à l'instant t ne peut pas être retrouvée à l'instant suivant. La figure 7.3 montre deux exemples où une bande blanche apparaît brusquement. Ce genre de situation peut se produire lorsqu'une bande blanche n'apparaît que par intermittence, comme dans le cas d'une bande discontinue, ou lorsqu'une bande partiellement effacée n'a pu être détectée pendant un certain temps. Dans ces deux exemples, la procédure de suivi ne peut pas retrouver la bande blanche à l'instant $t+2$ puisqu'elle n'était pas présente à l'instant précédent. L'algorithme doit donc prévoir ce genre de situation lors de la reconstruction.

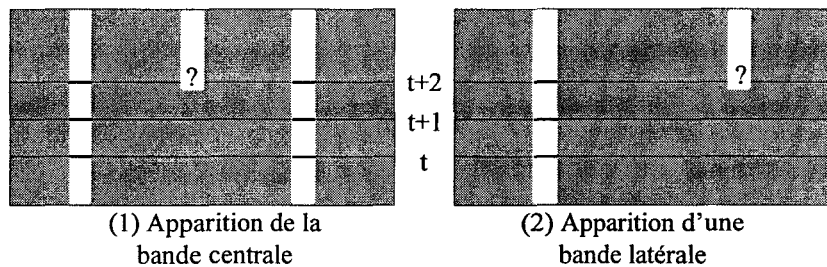


Figure 7.3 : Exemples d'apparition d'une bande blanche.

Pour détecter l'apparition d'éventuelles bandes blanches, la procédure de détection (Cf. § 6-4) est appliquée successivement à chaque liste stéréo $S_k(n_k)$. Dans l'exemple de la figure 7.4, l'examen de la zone d'intérêt permet de détecter une nouvelle bande blanche.

Lorsqu'une bande blanche est détectée, on procède à la mise à jour des éléments suivants :

- La liste objet $Obj(m, t)$. On ajoute la nouvelle bande à la liste.
- La liste des zones d'intérêt $Z_i(k)$. La zone où a été trouvée la nouvelle bande blanche est divisée en deux zones.
- La liste stéréo $S_k(n_k)$. Les points correspondant à la bande sont éliminés et la liste est éventuellement scindée en deux en fonction des attributs des deux nouvelles zones d'intérêt.

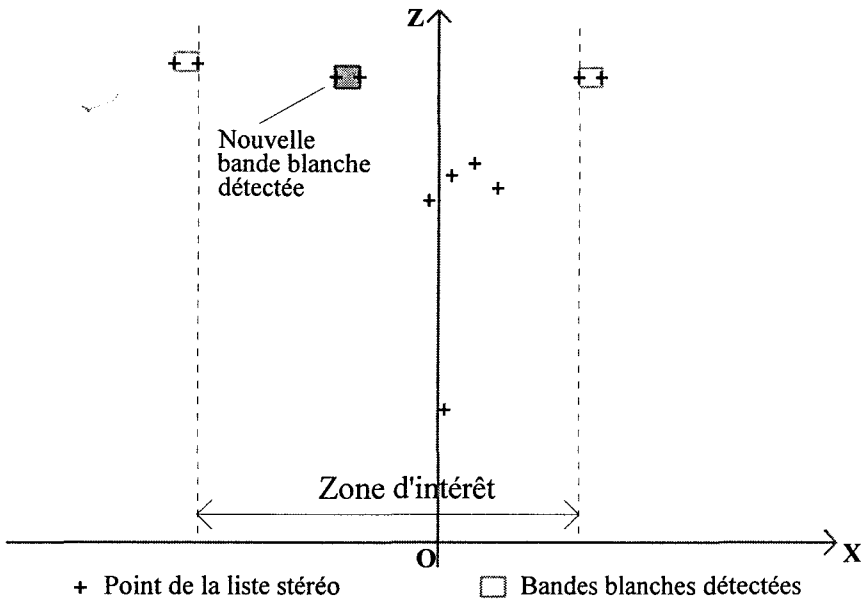


Figure 7.4 : Détection d'une nouvelle bande blanche.

7-2.4. Reconstruction des objets

7-2.4.1. Hypothèse

La procédure de reconstruction des objets utilise les attributs de position et de distance de chaque élément de la liste stéréo $S_k(n_k)$. On suppose que dans chaque zone d'intérêt le nombre d'objets est restreint et surtout que les points correspondant au même objet sont regroupés dans une même région de l'espace. Si cette hypothèse est vérifiée dans le cas d'un piéton ou d'une voiture vue de l'arrière, celle-ci ne l'est plus si la voiture est légèrement de travers (Cf. figure 7.5). Cependant dans le troisième cas, l'arrière du véhicule génère suffisamment de points pour former un objet significatif identifiable par la procédure de suivi.

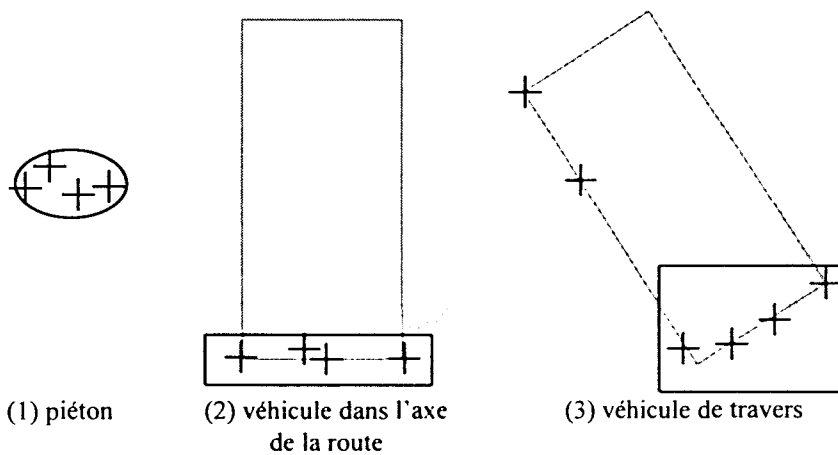


Figure 7.5 : Concentration des points dans le repère (O, X, Z).

7-2.4.2. Fusion des points

Pour réaliser la fusion des points, la liste stéréo $S_k(n_k)$ est analysée séquentiellement. Nous rappelons une fois encore que les éléments de cette liste sont rangés selon leurs positions croissantes. Pour éviter toute confusion avec la liste objet $Obj(m, t)$ utilisée pour les bandes blanches, les nouveaux objets sont rangés dans la liste objet $Objet(v, t)$.

Soit l'élément $S_k(n_k)$ appartenant à l'objet $Objet(v, t)$. Nous considérons que l'élément suivant $S_k(n_k+1)$ fait parti du même objet si ses attributs vérifient les deux relations suivantes :

$$S_k(n_k+1).position - S_k(n_k).position < \lambda_{position}$$

$$| S_k(n_k+1).distance - S_k(n_k).distance | < \lambda_{distance}$$

Dans l'affirmative, l'élément $S_k(n_k+1)$ appartient à l'objet $Objet(v, t)$ (Cf. Figure 7.6 (1)). On applique alors ces deux relations aux attributs des éléments $S_k(n_k+1)$ et $S_k(n_k+2)$. La fusion est réalisée ainsi de proche en proche.

Dans le cas contraire, l'élément $S_k(n_k+1)$ est trop éloigné de l'élément $S_k(n_k)$ pour appartenir au même objet (Cf. Figure 7.6 (2)). On crée alors un nouvel objet $Objet(v+1, t)$ dont le point $S_k(n_k+1)$ constitue le premier élément. La procédure de fusion continue en étudiant les coordonnées du point $S_k(n_k+2)$. L'objet $Objet(v, t)$ est, quant à lui, complètement défini, on effectue alors la mise à jour de ses attributs.

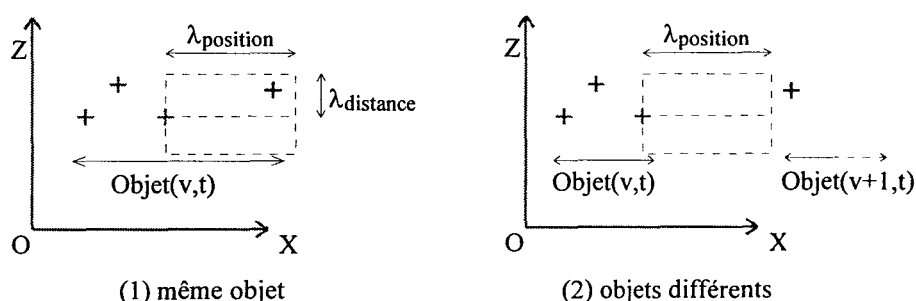


Figure 7.6 : Test de fusion de l'élément $S_k(n_k)$ avec l'élément $S_k(n_k+1)$.

Soit l'objet $Objet(v, t)$ construit à partir des éléments $S_k(n_k+\omega)$ de la liste stéréo $S_k(n_k)$ avec $\omega = 0, 1, \dots, \Omega$. On lui associe les attributs suivants :

2. $Objet(v, t).x_G.début = S_k(n_k).x_G$
3. $Objet(v, t).x_G.fin = S_k(n_k+\Omega).x_G$
4. $Objet(v, t).x_D.début = S_k(n_k).x_D$
5. $Objet(v, t).x_D.fin = S_k(n_k+\Omega).x_D$
6. $Objet(v, t).position.début = S_k(n_k).position$

7. $\text{Objet}(v, t).\text{position.fin} = S_k(n_k + \Omega).\text{position}$
8. $\text{Objet}(v, t).\text{distance.début} = \text{Min} (S_k(n_k + \omega))$ avec $\omega = 0, 1, \dots, \Omega$
9. $\text{Objet}(v, t).\text{distance.fin} = \text{Max} (S_k(n_k + \omega))$ avec $\omega = 0, 1, \dots, \Omega$

Ces attributs sont les mêmes que ceux attribués aux bandes blanches et sont définis de la même manière. La seule différence vient du nombre de points qui peut être supérieur à deux (Cf. Figure 7.7).

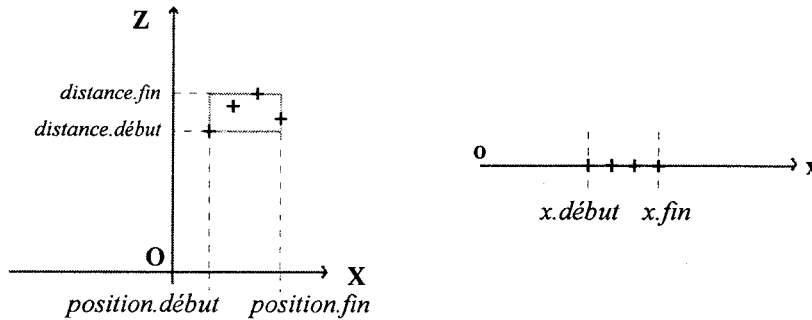


Figure 7.7 : Localisation des objets dans le repère (O, X, Z) et dans un repère image.

Comme pour les bandes blanches, les objets reconstruits sont parfaitement définis dans les repères images et le repère (O, X, Z).

Dans la pratique, nous fixons respectivement les seuils $\lambda_{\text{position}}$ et $\lambda_{\text{distance}}$ à 1,5 m et 2 m.

7-2.4.3. Fusion des objets

La figure 7.8 représente les objets reconstruits par le module de fusion des points. Cependant il arrive qu'un objet soit scindé en deux. Par exemple, sur cette figure, les objets 3 et 5 correspondent en fait au même objet. Mais l'objet 4 qui n'est en fait constitué que d'un point a contraint le module de fusion des points à créer deux objets. On rencontre ce genre de situation lorsqu'une erreur de mise en correspondance se produit. Un mauvais appariement conduit généralement à engendrer un point qui se démarque des objets réellement présent dans la scène, comme le point de l'objet 4.

Pour remédier à cette éventualité, nous tentons une fusion des objets. Selon le même principe pour la fusion des points, deux objets forment en réalité un seul objet s'ils sont proches l'un de l'autre.

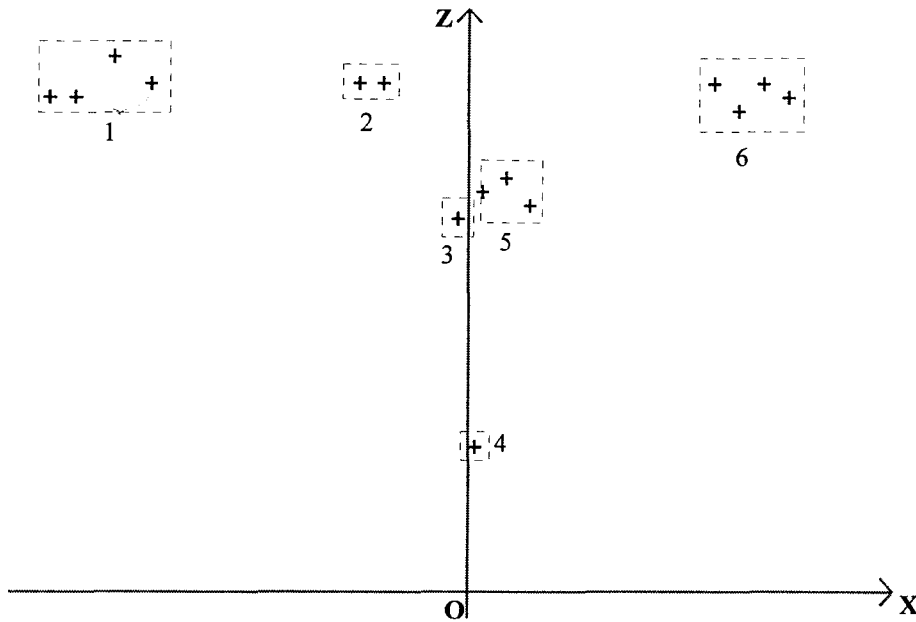


Figure 7.8 : Reconstruction des objets.

Les objets de la liste objet $\text{Objet}(v, t)$ étant également rangés selon les positions croissantes, la liste objet est analysée séquentiellement. Les objets $\text{Objet}(v, t)$ et $\text{Objet}(v+1, t)$ constituent un objet unique si leurs attributs vérifient les deux relations suivantes :

$$\begin{aligned} \text{Objet}(v+1, t).\text{position.fin} - \text{Objet}(v, t).\text{position.début} &< \lambda_{\text{position}} \\ | \text{Objet}(v+1, t).\text{distance.début} - \text{Objet}(v, t).\text{distance.fin} | &< \lambda_{\text{distance}} \end{aligned}$$

Dans l'affirmative, on fusionne les deux objets qui forment alors l'objet $\text{Objet}(v, t)$ dont on met à jour les attributs. L'objet $\text{Objet}(v+1, t)$ est, quant à lui, éliminé. On considère ensuite les objets $\text{Objet}(v, t)$ et $\text{Objet}(v+2, t)$. Dans le cas contraire, on passe aux objets $\text{Objet}(v+1, t)$ et $\text{Objet}(v+2, t)$.

Après l'exécution du module de fusion des objets, on dispose de la liste objet $\text{Objet}(v', t)$ avec $v' = 1, \dots, V'$ où V' est le nombre d'objets reconstruit à l'instant t .

7-2.5. Nature des objets

Nous avons vu au § 7-2.1, qu'excepté pour les bandes blanches, il est difficile d'identifier les objets dans les repères images. Dans le repère (O, X, Z) , l'identification des objets ne peut être envisagée qu'en étudiant leurs dimensions, c'est à dire leur largeur et leur épaisseur. Cependant, l'épaisseur n'est pas un paramètre significatif. En effet, il arrive qu'un piéton et une voiture présentent des épaisseurs similaires (Cf. Figure 7.5 (1-2)).

La nature des objets est donc déterminée en fonction d'un seul paramètre : la largeur. Pour l'objet $\text{Objet}(v, t)$, la largeur est calculée de la façon suivante :

$$\text{largeur} = \text{Objet}(v, t).\text{position.début} - \text{Objet}(v, t).\text{position.fin}$$

En fonction de la valeur obtenue, nous définissons 4 types d'objets (Cf. Tableau 7.1). On associe ainsi à chaque objet $\text{Objet}(v, t)$ un type que l'on mémorise dans l'attribut $\text{Objet}(v, t).\text{type}$.

Type	largeur
Piéton	$\lambda_{\text{piéton}_{\min}} < \text{largeur} \leq \lambda_{\text{piéton}_{\max}}$
Voiture	$\lambda_{\text{piéton}_{\max}} < \text{largeur} \leq \lambda_{\text{voiture}_{\max}}$
Camion	$\lambda_{\text{voiture}_{\max}} < \text{largeur} \leq \lambda_{\text{camion}_{\max}}$
Inconnu	Autres valeurs

Tableau 7.1 : Estimation du type des objets.

Il faut noter que la largeur des objets dépend du nombre de points de contour détectés dans chaque image. Puisque ce nombre varie d'une image à l'autre (Cf. § 7-2.1), elle peut changer brusquement et, par conséquent, le type également. La procédure de suivi étudie l'évolution de celui-ci afin de corriger les éventuelles erreurs d'identification.

En pratique, on fixe respectivement les seuils $\lambda_{\text{piéton}_{\min}}$, $\lambda_{\text{piéton}_{\max}}$, $\lambda_{\text{voiture}_{\max}}$ et $\lambda_{\text{camion}_{\max}}$ à 5 cm, 1 m, 2 m et 3 m.

7-2.6. Sélection des obstacles potentiels

Lorsque les objets ont été reconstruits et identifiés, l'étape suivante consiste à sélectionner ceux qui seront analysés par la procédure de suivi. Les paramètres X_{\min} et X_{\max} de la zone de danger ont déjà joué leur rôle lors du choix des points de la liste stéréo, le paramètre Z_{\max} est utilisé pour éliminer les objets trop éloignés du stéréoscope linéaire.

7-2.6.1. Détermination de la portée maximale Z_{\max}

Le paramètre Z_{\max} correspond à la portée maximale du stéréoscope linéaire, c'est à dire la distance à laquelle le plan de visée rencontre la chaussée. Si le stéréoscope est calibré pour avoir une portée de 30 m, celle-ci n'est que théorique. En pratique, les oscillations dues aux suspensions de la voiture la modifient constamment. Celle-ci doit donc être réévaluée à chaque nouvelle acquisition pour être utilisée.

Lorsque des bandes blanches sont détectées, l'évaluation de la portée maximale est simple. En effet, les bandes blanches peintes sur le sol se situent à l'intersection du plan de visée et de la chaussée. Le paramètre Z_{max} peut donc être facilement déterminé à partir des coordonnées bandes blanches.

Soit $Obj(m, t)$ la liste objet contenant la description de toutes les bandes blanches détectées à l'instant t . La portée maximale est définie par l'expression suivante :

$$Z_{max} = \text{Max} (Obj(m, t).distance.fin) \text{ avec } m = 1, \dots, M$$

On détermine également un autre paramètre que l'on note Z_{min} grâce à la relation suivante :

$$Z_{min} = \text{Min} (Obj(m, t).distance.début) \text{ avec } m = 1, \dots, M$$

En théorie, les paramètres Z_{max} et Z_{min} sont égaux. En pratique, il existe une légère différence due aux incertitudes de localisation (Cf. Figure 7.9). Ces deux paramètres permettent d'évaluer cette différence.

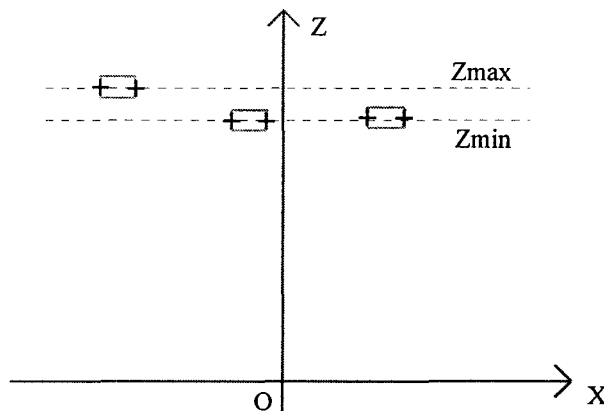


Figure 7.9 : Détermination des paramètres de portée.

Le paramètre Z_{min} est utilisé par la procédure de suivi (Cf. § 7-3.6.1).

Quand l'algorithme ne détecte aucune bande blanche, l'évaluation de la portée maximale n'est pas possible. En effet, l'objet le plus éloigné ne se situe pas nécessairement à la limite de portée de stéréoscope linéaire. De plus, il arrive que des erreurs de mise en correspondance engendrent des objets très éloignés. Les paramètres Z_{max} et Z_{min} sont alors fixés respectivement à 110 % et 90 % de la portée maximale théorique, c'est à dire la portée fixée lors du calibrage du stéréoscope linéaire.

7-2.6.2. Sélection des obstacles potentiels

Avant d'entamer la procédure de suivi, l'algorithme procède à une dernière sélection. Celle-ci s'effectue en conservant les objets situés à une distance inférieure à la portée maximale Z_{max} .

Soit la liste objet $Objet(v, t)$ composée des objets reconstruits. L'algorithme sélectionne les objets dont l'attribut de distance vérifie la relation :

$$Objet(v, t).distance.début < Z_{max}$$

Grâce à cette opération, l'algorithme élimine les objets situés au-delà de la portée maximale, c'est à dire les objets engendrés par des erreurs de mise en correspondance.

La procédure de reconstruction fournit finalement une liste objet dont les éléments se situent dans la zone de danger définie par X_{min} , X_{max} et Z_{max} . Chaque élément est considéré comme un obstacle potentiel et est étudié par la procédure de suivi. Nous notons la liste obtenue $Obst(u, t)$ avec $u = 1, \dots, U$ où U est le nombre d'obstacles potentiels sélectionnés.

7-3. Suivi spatio-temporel

7-3.1. Objectif

La mise en correspondance des primitives et la reconstruction tridimensionnelle ont permis d'obtenir la liste des objets se trouvant dans la zone de danger. Cependant, la décision de déclencher une alarme dès qu'un objet a été détecté à l'instant t serait prématurée. En effet, l'objet peut être un véhicule qui s'éloigne après nous avoir doublé ou encore un piéton traversant la chaussée, mais ne se trouvant pas directement sur la trajectoire du véhicule. Enfin il existe une autre possibilité qu'il ne faut pas négliger : des erreurs de mise en correspondance peuvent créer à un instant donné un objet qui semble dangereux. Les objets doivent être d'abord considérés comme des *obstacles potentiels*.

Le but du suivi est de localiser précisément les objets pendant une période suffisamment longue afin d'obtenir les caractéristiques précises de ceux-ci, notamment le sens de déplacement et la vitesse relative à partir desquels on peut évaluer le danger qu'ils représentent.

7-3.2. Durée de vie

Afin de déterminer à l'instant t si un objet est dangereux, il faut disposer d'informations sur son passé. En effet, lorsqu'un objet apparaît pour la première fois, on ne peut porter aucun jugement fiable sur celui-ci. Il peut effectivement correspondre à un objet dangereux, mais il

ne faut pas ignorer la possibilité d'une fausse détection. Par contre, si l'objet est détecté à plusieurs reprises, il est possible de mémoriser certaines de ses caractéristiques afin d'avoir à l'instant t une connaissance suffisante de son histoire pour porter un jugement sur celui-ci. Les caractéristiques les plus importantes sont ses coordonnées, son sens de déplacement et sa vitesse relative par rapport au véhicule. Cependant ces paramètres sont réévalués à l'instant t et ne témoignent pas du "vécu" de l'objet.

Nous associons donc à chaque objet une *durée de vie*. Celle-ci augmente à chaque nouvelle détection et constitue un critère important lors de la prise de décision. En effet, plus un objet est "vu" longtemps, plus il a de chance d'être réellement présent. Inversement, quand il disparaît brusquement sa durée de vie commence à décroître. De cette manière, on n'exclut pas la possibilité de le voir réapparaître dans les images suivantes. Si sa durée de vie devient nulle, l'objet est considéré comme ayant définitivement disparu et il n'est plus pris en compte. La durée de vie ne croît pas indéfiniment. En effet, lorsque celle-ci atteint une valeur suffisante pour estimer qu'un objet est réellement présent, il est inutile de l'augmenter davantage.

Dans la pratique, nous fixons la durée de vie maximale à 5 détections successives. Cependant nous considérons qu'une alarme doit être déclenchée si la durée de vie d'un objet dangereux est supérieure ou égale à 3 détections successives. En d'autres termes, lorsqu'un objet est détecté dans 5 images successives, nous estimons qu'il est réellement présent. Mais, si un objet est dangereux pour le véhicule, la détection dans 3 images successives est suffisante.

La durée de vie de l'objet $\text{Obst}(u, t)$ est mémorisée dans l'attribut $\text{Obst}(u, t).\text{vie}$.

7-3.3. Suivi des bandes blanches

Il ne faut pas oublier que le suivi se fait tout d'abord au niveau des bandes blanches (Cf. § 6-5). De cette manière, à chaque nouvelle acquisition, la position des bords de route est réactualisée (Cf. § 6-6) ainsi que la portée maximale du stéréoscope (§ 7-2.6.1) puisque les bandes blanches se situent en limite de portée. Ce suivi est nécessaire car, d'une part la voiture peut modifier sa position sur la chaussée, par exemple changer de voie, et d'autre part le tangage de la voiture modifie la portée du stéréoscope. Ainsi la zone de danger et les zones d'intérêt sont parfaitement redéfinies à chaque acquisition. Celles-ci permettent d'une part d'accélérer les temps de traitement et d'autre part d'éliminer les objets qui ne sont pas dangereux.

Il faut préciser que les objets "**bande blanche**" sont également dotés d'une durée de vie. Celle-ci augmente, sans dépasser la durée de vie maximale, tant que la bande blanche est détectée et diminue quand elle disparaît. La procédure de suivi des bandes blanches tente alors de la retrouver aux instants suivants. La durée de vie diminue à chaque tentative infructueuse. Quand celle-ci devient nulle, on considère que la bande a disparu. Elle est alors effacée de la liste objet.

7-3.4. "Mise en correspondance spatio-temporelle" des objets

Le but de cette opération est d'établir un lien entre les détections successives d'un même objet. En d'autres termes, pour chaque élément $\text{Obst}(u, t-1)$ on cherche l'élément $\text{Obst}(u', t)$ correspondant au même objet.

7-3.4.1. *Notion de bande de recherche*

Comme pour les bandes blanches, nous avons eu l'idée de déterminer une fenêtre autour de l'objet $\text{Obst}(u, t-1)$ et de chercher si un objet $\text{Obst}(u', t)$ est présent dans celle-ci. Comme nous travaillons dans le repère (O, X, Z) , deux paramètres sont nécessaires pour définir cette fenêtre : la largeur selon l'axe (O, X) et la profondeur selon l'axe (O, Z) . Cependant, il nous a semblé inutile de limiter la profondeur de la zone de recherche. En effet, la vitesse relative d'un objet par rapport au véhicule peut être définie dans le repère (O, X, Z) par deux composantes :

- Une composante longitudinale : donnée par le mouvement colinéaire au sens de déplacement du véhicule, c'est à dire selon l'axe (O, Z) .
- Une composante transversale : donnée par le mouvement perpendiculaire au sens de déplacement du véhicule, c'est à dire selon l'axe (O, X) .

Or, les vitesses transversales sont généralement beaucoup plus faibles que les vitesses longitudinales. En effet, les variations de position d'un véhicule changeant de voie ou d'un piéton traversant une route sont négligeables par rapport aux variations de distance d'une voiture fonçant sur un piéton ou un autre objet. Pour être sûr de retrouver un objet dont la vitesse longitudinale est très importante, il est donc préférable de définir une fenêtre dotée d'une grande profondeur. Une autre solution consiste à estimer la distance à laquelle il devrait se trouver à l'instant t en fonction de sa vitesse longitudinale déterminée à l'instant $t-1$. Mais cette méthode suppose d'avoir, au minimum, déjà détecté l'objet à deux reprises, et surtout d'avoir établi un lien entre ces deux détections afin de pouvoir évaluer cette vitesse.

Cependant, si un objet apparaît pour la première fois à l'instant $t-1$, sa vitesse longitudinale n'est pas connue. Si celle-ci est très importante, le seul moyen d'établir un lien entre l'objet détecté à l'instant $t-1$ et celui détecté à l'instant t est d'avoir défini une fenêtre de recherche dotée d'une grande profondeur. Si ce n'est pas le cas, aucun rapprochement ne peut être réalisé entre ces deux objets. L'objet détecté à l'instant $t-1$ est alors considéré comme une fausse détection, et celui à l'instant t comme nouveau. Un objet dangereux peut ainsi passer inaperçu si le même phénomène se reproduit aux instants suivants.

Pour éviter ce genre de problème, nous avons décidé d'utiliser une *bande de recherche*, qui n'impose aucune limite sur la profondeur (Cf. Figure 7.10).

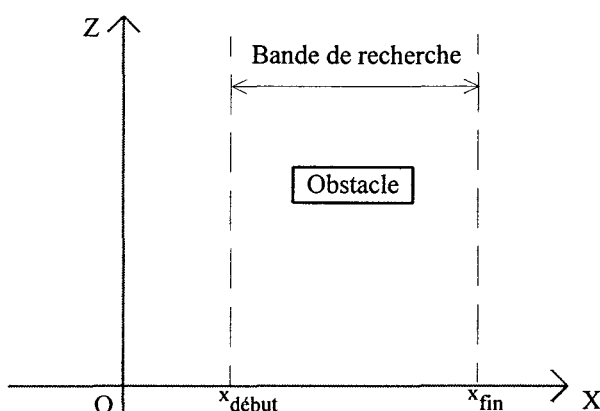


Figure 7.10 : Notion de bande de recherche.

7-3.4.2. Détermination de la bande de recherche

La largeur de la bande de recherche est calculée en fonction de la largeur de l'objet. Nous déterminons la largeur L de l'objet $\text{Obst}(u, t-1)$ à l'aide de l'expression suivante :

$$L = \text{Obst}(u, t-1).\text{position.fin} - \text{Obst}(u, t-1).\text{position.début}$$

Il faut cependant préciser que cette largeur peut être très faible, voire nulle si l'objet n'a été reconstruit qu'à partir d'un seul point de liste stéréo. Nous imposons donc que le paramètre L soit au minimum égal à 50 cm.

L'étape suivante consiste à évaluer la position $X(t-1)$ du centre de l'objet $\text{Obst}(u, t-1)$. Celle-ci s'obtient par l'expression :

$$X(t-1) = \frac{1}{2} (\text{Obst}(u, t-1).\text{position.début} + \text{Obst}(u, t-1).\text{position.fin})$$

En fonction de la position $X(t-1)$ et de la vitesse transversale de l'objet à l'instant $t-1$, nous estimons ensuite la position probable $X_{\text{théorique}}(t)$ du centre de l'objet à l'instant t de la façon suivante :

$$X_{\text{théorique}}(t) = X(t-1) + \text{Obst}(u, t-1).vitesse \times \delta t$$

où :

- $\text{Obst}(u, t-1).vitesse$ est la vitesse transversale de l'objet $\text{Obst}(u, t-1)$ à l'instant $t-1$.
- δt est l'intervalle de temps entre les instants t et $t-1$.

Nous déterminons enfin les limites de la bande de recherche avec les expressions :

$$x_{\text{début}} = X_{\text{théorique}}(t) - L$$

$$x_{\text{fin}} = X_{\text{théorique}}(t) + L$$

On obtient ainsi une bande de recherche de largeur $2L$ centrée sur la position prédite de l'objet à l'instant t (Cf. Figure 7.11).

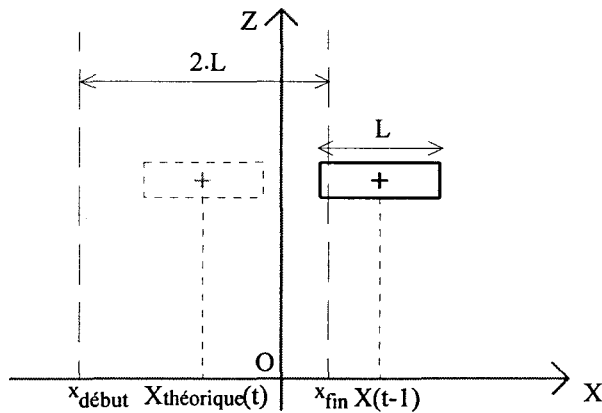


Figure 7.11 : Détermination de la bande de recherche.

La connaissance de la vitesse transversale de l'objet permet d'estimer sa position à l'instant t . De cette façon, on peut limiter la largeur de la bande de recherche. On diminue ainsi les probabilités d'avoir plusieurs objets dans la même fenêtre et par conséquent, les ambiguïtés de "mise en correspondance spatio-temporelle".

Quand la vitesse transversale de l'objet n'a pas encore été déterminée, l'attribut de vitesse étant initialisé à zéro, les positions $X_{\text{théorique}}(t)$ et $X(t-1)$ sont égales. Comme nous ignorons le sens de déplacement de l'objet, nous élargissons la bande de recherche à $3L$. Les limites de la bande sont alors :

$$x_{\text{début}} = X_{\text{théorique}}(t) - \frac{3}{2}L$$

$$x_{\text{fin}} = X_{\text{théorique}}(t) + \frac{3}{2}L$$

7-3.4.3. Recherche de l'objet correspondant dans la bande

Lorsque la bande est déterminée, l'algorithme recherche dans la liste objet $\text{Obst}(u', t)$ si un objet se situe dans celle-ci.

Si un objet $\text{Obst}(u', t)$ présente des attributs vérifiant l'une des deux relations :

$$x_{\text{début}} < \text{Obst}(u', t).\text{position.début} < x_{\text{fin}}$$

$$x_{\text{début}} < \text{Obst}(u', t).\text{position.fin} < x_{\text{fin}}$$

celui-ci est considéré comme le correspondant de l'objet $\text{Obst}(u, t-1)$. On établit alors un lien entre ces deux détections.

Dans le cadre de nos expérimentations, le nombre d'objets n'était pas suffisant pour avoir plus d'un objet dans la zone de recherche. Cependant, en cas d'ambiguïté, différents tests de comparaison peuvent être réalisés afin de sélectionner l'un des objets. Nous pouvons citer par exemple :

- Le type : si l'objet $\text{Obst}(u, t-1)$ correspond à l'objet $\text{Obst}(u', t)$, ils doivent présenter le même type.
- La distance et la position : on peut également conserver l'objet $\text{Obst}(u', t)$ dont les coordonnées sont les plus proches des coordonnées théoriques déterminées en fonction des attributs de l'objet $\text{Obst}(u, t-1)$. Ce test n'est bien sûr applicable que si les vitesses ont été déterminées.

7-3.4.4. Conclusion sur la mise en correspondance spatio-temporelle

La mise en correspondance spatio-temporelle est appliquée à chaque élément de la liste objet $\text{Obst}(u, t-1)$. Elle permet ainsi d'établir des liens entre les deux listes $\text{Obst}(u, t-1)$ et $\text{Obst}(u', t)$. Cependant chaque élément de la liste $\text{Obst}(u, t-1)$ ne possède pas nécessairement un correspondant dans la liste $\text{Obst}(u', t)$ et réciproquement. L'étape suivante consiste donc à analyser les différentes évolutions possibles pouvant se produire entre l'instant $t-1$ et l'instant t .

7-3.5. Suivi temporel des objets

Le but du suivi temporel des objets est d'analyser les différentes évolutions possibles pouvant se produire entre l'instant $t-1$ et l'instant t . Ce module permet ainsi de mettre à jour les attributs de chaque objet dont notamment la durée de vie, le sens de déplacement et la vitesse relative.

7-3.5.1. Apparition d'un objet

On considère qu'un objet $Obst(u', t)$ apparaît dans la zone de danger quand celui-ci ne possède pas de correspondant dans la liste $Obst(u, t-1)$. La durée de vie de l'objet est fixée à 1 pour indiquer que celui-ci est détecté pour la première fois. Aucune conclusion n'est possible à l'instant t car les seules caractéristiques connues, comme le type et les coordonnées, ne permettent pas de porter un jugement. Il faut attendre l'instant $t+1$ pour confirmer la détection et déterminer des caractéristiques plus significatives comme le sens de déplacement et la vitesse relative.

7-3.5.2. Disparition d'un objet

On considère qu'un objet $Obst(u, t-1)$ a disparu de la zone de danger quand celui-ci ne possède pas de correspondant dans la liste $Obst(u', t)$. On diminue alors sa durée de vie d'une unité.

Si celle-ci n'est pas nulle, on considère qu'il s'agit d'une détection manquée et que l'objet est toujours présent. On ajoute alors à la liste objet $Obst(u', t)$ l'objet $Obst(u'', t)$ présentant les mêmes attributs que ceux de l'objet $Obst(u, t-1)$. La procédure de mise en correspondance spatio-temporelle tentera de le retrouver à l'instant $t+1$.

Si la durée de vie est ramenée à zéro, on estime que l'objet $Obst(u, t-1)$ a définitivement disparu. Il n'est donc plus pris en compte. On élimine ainsi les objets ayant quitté la zone de danger, mais également les objets fugitifs engendrés par des erreurs de mise en correspondance.

7-3.5.3. Confirmation de la présence d'un objet

On considère que l'objet $Obst(u, t-1)$ est toujours présent si la procédure de mise en correspondance spatio-temporelle a établi un lien avec l'objet $Obst(u', t)$. Afin de vérifier la cohérence entre ces deux détections, on compare le type de l'objet $Obst(u, t-1)$ avec celui de l'objet $Obst(u', t)$. Deux situations sont possibles.

7-3.5.3.1. Les types sont identiques

Cette situation témoigne de la cohérence parfaite entre les détections. On procède alors à la mise à jour des attributs de l'objet $Obst(u', t)$. La première opération consiste à incrémenter la durée de vie d'une unité si la limite maximale n'a pas été atteinte. Comme l'objet a été détecté au moins deux fois, on peut évaluer son sens de déplacement et sa vitesse relative entre les instants $t-1$ et t . On procède de la façon suivante :

$$\text{Obst}(u', t).\text{vitesse} = \frac{1}{\delta t} (X(t) - X(t-1)).$$

$$\text{Obst}(u', t).\text{sens} = \text{SIGN}(\text{Obst}(u', t).\text{vitesse}).$$

où :

- $\text{SIGN}(z)$ est la fonction qui donne le signe de la variable z .
- $X(t)$ et $X(t-1)$ représentent respectivement les positions du centre de l'objet aux instants t et $t-1$. Celles-ci sont déterminées par les expressions :

$$X(t) = \frac{1}{2} (\text{Obst}(u', t).\text{position.début} + \text{Obst}(u', t).\text{position.fin})$$

$$X(t-1) = \frac{1}{2} (\text{Obst}(u, t-1).\text{position.début} + \text{Obst}(u, t-1).\text{position.fin})$$

- δt est l'intervalle de temps entre les instants t et $t-1$.

Lorsque ces différentes opérations sont terminées, les informations disponibles sont suffisantes pour déterminer si l'objet est dangereux pour le véhicule.

7-3.5.3.2. Les types sont différents

Cette situation indique qu'une erreur a été commise lors de l'identification des objets. Cependant, nous ignorons si cette erreur s'est produite à l'instant t ou à l'instant $t-1$. Pour lever cette ambiguïté, nous utilisons la durée de vie de l'objet $\text{Obst}(u, t-1)$.

Soient X le type de l'objet $\text{Obst}(u, t-1)$ et Y le type de l'objet $\text{Obst}(u', t)$.

Si l'objet $\text{Obst}(u, t-1)$ de type X présente une durée de vie supérieure ou égale à 2, on considère l'attribution du type Y à l'objet $\text{Obst}(u', t)$ comme une erreur et le type X lui est affecté. Par contre, si la durée de vie est égale à 1, on remet en cause l'attribution précédente, et le type Y est conservé. La prochaine acquisition permettra éventuellement de confirmer cette décision.

La mise à jour des autres attributs de l'objet $\text{Obst}(u', t)$ est ensuite réalisée comme dans le cas précédent (Cf. § 7-3.5.3.1).

7-3.5.4. Conclusion

La procédure de suivi temporel fournit la liste objet $\text{Obst}(u', t)$ composée des objets présents dans la zone de danger à l'instant t . Cependant ils ne sont pas tous dangereux. La dernière étape consiste donc à sélectionner les objets dangereux.

7-3.6. Sélection des obstacles

7-3.6.1. *Elimination du "fond"*

Vue du stéréoscope, la chaussée représente la partie la plus éloignée car elle se situe à la limite de portée que nous avons appelée Z_{max} . En théorie, elle constitue un fond idéal car elle présente des niveaux de gris uniformes. Elle n'engendre donc aucun point de contour et par conséquent aucun objet. En pratique, on rencontre à la surface de la chaussée des tâches ou des ombres qui donnent naissance à des objets dont la largeur est parfois identique à celle d'un piéton, voire d'une voiture. Mais les ombres et les tâches ne sont pas les seuls objets que nous pouvons trouver à cette distance Z_{max} . En effet, des objets peuvent également être détectés à cette distance. Pour éviter de déclencher une fausse alarme en prenant une ombre pour un obstacle, ou d'ignorer un obstacle en pensant qu'il s'agit d'une tâche sur le sol, il est essentiel de différencier ces deux catégories d'objets. Dans la pratique, il s'avère que les informations dont nous disposons sont insuffisantes pour réaliser une telle opération. La solution que nous avons donc retenue est d'ignorer à l'instant t tous les objets situés dans une zone que nous appelons le "fond". Celui-ci est délimité par les paramètres Z_{max} et Z_{min} définis au § 7-2.6.1. Nous pouvons nous permettre d'ignorer ces objets car ils ne représentent pas un danger immédiat pour le véhicule puisqu'ils sont encore éloignés de celui-ci.

L'avantage d'une telle opération est d'éliminer tous les objets engendrés par la chaussée. Il faut également préciser que, si aucune bande blanche n'a été détectée, tous les objets créés par les bas-côtés de la route et se trouvant dans la zone de danger sont également ignorés. L'inconvénient est de devoir ignorer les véritables obstacles situés dans cette zone. Cependant, ceux-ci sont toujours mémorisés dans la liste objet $Obst(u', t)$ avec tous leurs attributs. A l'instant $t+1$, la procédure de mise en correspondance spatio-temporelle tentera de les retrouver. Si l'un d'entre eux est réellement dangereux, il se rapprochera du véhicule. Ne faisant plus partie du "fond", il ne sera pas ignoré.

Il faut également noter que l'élimination du "fond" est réalisée après le suivi temporel des objets. L'analyse des objets se trouvant dans cette zone permet ainsi de définir leurs caractéristiques. De cette manière, s'ils en sortent, nous disposons d'informations suffisantes pour déterminer si une alarme doit être déclenchée.

Comme les objets situés dans cette zone doivent être ignorés et non supprimés, la liste $Obst(u, t)$ est conservée intégralement pour être utilisée à l'instant $t+1$ par la procédure de

mise en correspondance spatio-temporelle. On forme alors une nouvelle liste composée des objets n'appartenant pas au "fond". On effectue donc une première sélection en ne conservant que les objets dont l'attribut de distance vérifie la relation :

$$\text{Obst}(u, t).\text{distance.fin} < Z_{\text{min}}$$

Ces attributs sont mémorisés dans la liste objet $\text{OBST}(w, t)$ avec $w = 1, \dots, W$ où W est le nombre d'objets de la zone de danger n'appartenant pas au "fond".

7-3.6.2. *Elimination des objets récents*

Les éléments de la liste objet $\text{OBST}(w, t)$ ne peuvent pas encore être considérés comme des obstacles. En effet, certains présentent une durée de vie trop faible pour que leur détection soit jugée fiable. Ils sont donc ignorés.

Nous obtenons la liste des obstacles présents à l'instant t en conservant les objets dont la durée de vie vérifie l'expression :

$$\text{OBST}(w, t).\text{vie} \geq 3$$

On forme ainsi la liste $\text{OBSTACLE}(s, t)$ avec $s = 1, \dots, S$ où S est le nombre d'obstacles détectés à l'instant t .

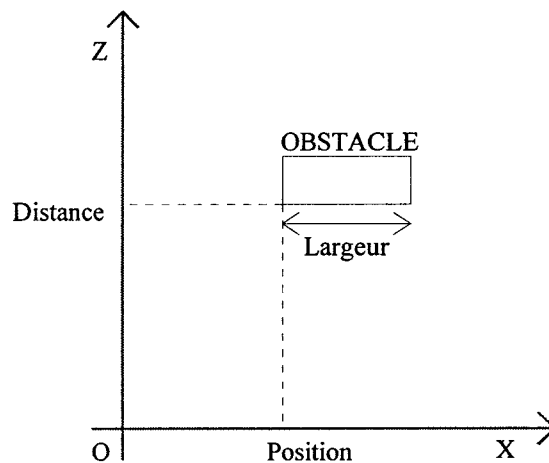


Figure 7.12 : Localisation d'un obstacle dans le repère (O, X, Z).

7-3.6.3. *Localisation des obstacles*

Pour localiser un obstacle dans le repère (O, X, Z) on associe à chaque élément de la liste $\text{OBSTACLE}(s, t)$ les paramètres suivants (Cf. Figure 7.12) :

$$\text{OBSTACLE}(s, t).\text{Position} = \text{OBST}(w, t).\text{position.début}$$

$$\text{OBSTACLE}(s, t).\text{Distance} = \text{OBST}(w, t).\text{distance.début}$$

$$\text{OBSTACLE}(s, t).\text{Largeur} = \text{OBST}(w, t).\text{position.fin} - \text{OBST}(w, t).\text{position.début}$$

$$\text{OBSTACLE}(s, t).\text{Vitesse} = \text{OBST}(w, t).\text{vitesse}$$

Il faut préciser que l'attribut vitesse correspond en fait à la composante transversale de la vitesse relative de l'obstacle par rapport au véhicule.

7-3.6.4. Transmission des obstacles

Dans le cadre du programme Prometheus, la liste OBSTACLE(s, t) est ensuite transmise au module de fusion de données dont le rôle est d'analyser les résultats fournis par les différents capteurs équipant le véhicule (Cf. chapitre 8). Si l'un des obstacles se trouve sur la trajectoire du véhicule, une alarme est alors déclenchée. Il faut cependant préciser que les obstacles de type inconnu, c'est à dire de faible largeur, ne sont pas transmis au module de fusion. En effet, afin d'alléger la tâche de celui-ci, les obstacles de faible largeur, détectés à l'instant t , qui correspondent généralement à des erreurs de mise en correspondance sont éliminés de la liste OBSTACLE(s, t). On simplifie ainsi le suivi de "haut niveau" effectué par le module de fusion en lui évitant d'étudier le comportement d'objets identifiés de façon incomplète. Cependant, notre procédure de suivi les conserve en mémoire. Si à l'instant suivant l'un d'entre eux est détecté comme PIETON ou VOITURE, il sera alors traité comme les autres obstacles et transmis au module de fusion.

7-3.7. Conclusion sur le suivi spatio-temporel

Le suivi spatio-temporel joue un double rôle dans notre système de détection d'obstacles. D'une part, le suivi des bandes blanches permet d'accélérer et de simplifier les traitements puisqu'elles servent à définir la zone de danger et les zones d'intérêt. D'autre part, il permet de retrouver, à chaque nouvelle acquisition, les objets présents en face du véhicule. Chaque objet est alors doté de certaines caractéristiques comme la vitesse ou la durée de vie. Celles-ci servent à déterminer s'ils représentent un véritable danger pour le véhicule.

Puisque plusieurs acquisitions sont nécessaires pour cerner de manière fiable les caractéristiques d'un objet, la vitesse de traitement apparaît comme un facteur important. Celle-ci doit être suffisante pour permettre de traiter plusieurs couples d'images afin de vérifier la cohérence de son comportement et prévenir le conducteur suffisamment tôt pour que celui-ci puisse éviter la collision.

7-4. Résultats expérimentaux

Nous présentons, dans ce paragraphe, différents exemples de détection réalisés par notre algorithme. A l'exception des images de la séquence "piéton" (Cf. Figure 2-17) et de celles de la séquence "sortie d'autoroute" (Cf. Figure 2-19), toutes les séquences ont été acquises avec le second prototype du stéréoscope linéaire (Cf. § 2-4.2). Les séquences présentées sont accompagnées d'un texte descriptif afin d'aider le lecteur à les interpréter.

Les résultats de notre algorithme de détection d'obstacles, c'est à dire la mise en correspondance bi-mode, la reconstruction et le suivi, sont représentés sous la forme d'une seule image. Celle-ci s'interprète de la manière suivante :

- L'axe vertical représente le temps, celui-ci défilant du haut vers le bas de l'image. Toutes les séquences ont été acquises à la cadence de 25 couples d'images par seconde (sauf les séquences "piéton" et "sortie d'autoroute" dont la cadence est de 10 couples d'images par seconde).
- L'axe horizontal représente la position dans la scène. Un objet se trouvant sur le bord gauche de l'image est situé à 10 m à gauche de l'axe optique. Un objet se trouvant sur le bord droit de l'image est situé à 10 m à droite de ce même axe. La largeur des objets est également représentée sur cette image.
- Les niveaux de gris correspondent au type de l'objet reconstruit. Par exemple, les bandes blanches sont affichées avec des niveaux de gris très clairs.

On notera que la distance n'est pas représentée sur ces images. La raison est simple. Comme nous ne disposions que d'un espace à deux dimensions afin de représenter pour chaque objet : sa position, sa distance, son type et l'instant auquel il a été détecté, nous avons dû trouver un compromis. Si le paramètre de distance était très important lors de l'élaboration de la procédure de mise en correspondance, il l'est beaucoup moins ici où nous cherchons d'abord à vérifier l'efficacité des procédures de reconstruction et de suivi des objets. Ce paramètre a donc été négligé dans les résultats présentés dans ce paragraphe.

Les expérimentations présentées dans ce chapitre ont été réalisées "hors ligne" sur un PC 486 DX 33. Les cadences de traitement atteintes sur ce type de machine sont d'environ 2 à 3 couples d'images par seconde.

7-4.1. Détection et suivi des bandes blanches

La première série d'exemples est consacrée à la détection et au suivi des bandes blanches.

7-4.1.1. Séquence "sortie d'autoroute"

La figure 7.13 montre les résultats obtenus sur la séquence stéréoscopique "sortie d'autoroute", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.2).

Les résultats obtenus sont très bons pour la bande centrale et celle de droite. Les objets "**bande blanche**", représentés par des niveaux de gris très clairs, ont été correctement détectés. Par contre, les résultats sont plutôt moyens en ce qui concerne la bande de gauche. D'une part, celle-ci n'est pas toujours détectée et d'autre part, sa largeur ne correspond pas à celle des autres bandes.

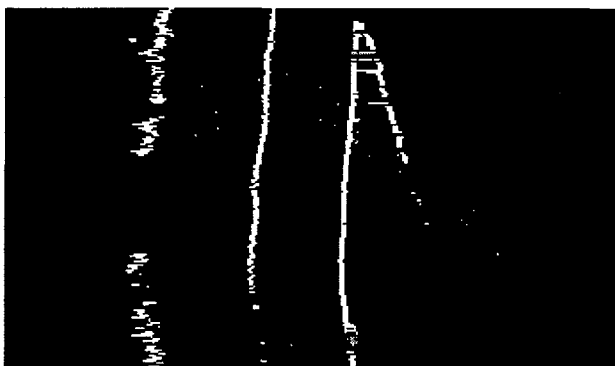


Figure 7.13 : Détection et suivi des bandes blanches sur la séquence "sortie d'autoroute".

L'examen de la séquence (Cf. Figure 2.19) montre que la bande gauche n'est pas très nette par endroits sur les images droites, ce qui explique les détections manquées. On constate qu'une bande dont les niveaux de gris sont uniformes et assez clairs est accolée à la bande blanche. La procédure de détection des bandes blanches a en fait considéré les deux bandes adjacentes comme une seule bande ce qui explique la largeur de la bande reconstruite. Cet exemple montre que les seuils utilisés pour reconnaître le profil d'une bande blanche ne sont pas assez sélectifs.

En comparant ce résultat avec celui obtenu par la procédure hiérarchique de mise en correspondance (Cf. Figure 5.26), on constate que les points engendrés par les bas-côtés ont été supprimés. On vérifie ainsi que la zone de danger a été correctement définie et qu'elle a permis d'éliminer les points se trouvant hors de cette zone.

On remarque également que la bande de droite s'interrompt à la fin de la séquence pour laisser la place à un objet dont le niveau de gris est foncé. Cette apparition s'explique en examinant

les images brutes de la séquence (Cf. Figure 2.19). Une dégradation de la bande due à une tâche empêche la procédure de suivi de retrouver la bande. Les contours engendrés par cette tâche sont ensuite fusionnés pour créer cet objet. Cependant celui-ci aurait dû être supprimé puisqu'il fait partie du "fond". En fait, le paramètre Z_{min} n'a été défini qu'en fonction de la bande gauche, ce qui ne permet pas d'estimer précisément la zone de "fond". Il faut au moins deux bandes situées de chaque côté du véhicule pour déterminer correctement le paramètre Z_{min} car la portée maximale du stéréoscope linéaire peut présenter une légère différence entre la gauche et la droite du champ.

7-4.1.2. Route de campagne dégagée

Pendant cette séquence, le véhicule roule sur une route de campagne représentée par un ruban gris sur lequel on distingue les bandes blanches. Aucun obstacle n'est présent. Les bas-cotés très clairs correspondent à de l'herbe. La route longe un bois dont les niveaux de gris très sombres sont visibles sur la droite de chaque image. On peut également remarquer l'ombre des arbres qui se dessine sur la chaussée au début de la séquence.

La chaussée n'est pas rectiligne en raison des mouvements latéraux du véhicule. En effet, celui-ci mord la bande blanche droite puis revient au centre de la voie droite à plusieurs reprises pendant cette séquence. Il faut également ajouter, de temps à autre, les oscillations du véhicule dues aux accélérations et aux légères déformations de la chaussée.

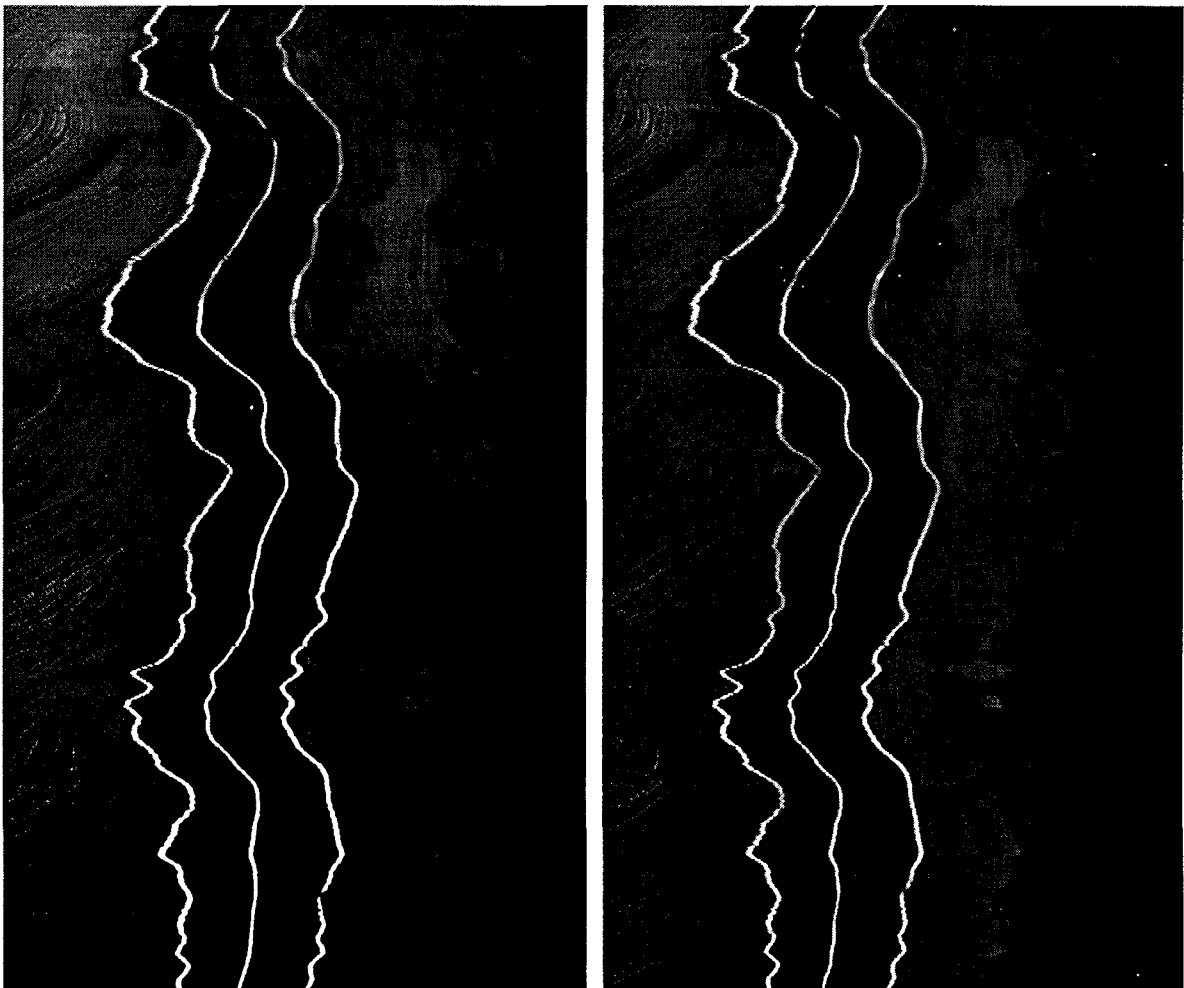


Figure 7.14 : Route de campagne dégagée.



Figure 7.15 : Scène reconstituée à partir de la séquence de la figure 7.14.

Comme dans l'exemple précédent, la bande centrale et la bande latérale droite sont parfaitement détectées et suivies pendant toute la séquence. La bande gauche, qui est pourtant parfaitement visible, n'est pas toujours détectée correctement. Nous n'avons trouvé aucune raison permettant d'expliquer ce phénomène. Les résultats obtenus sont néanmoins très satisfaisants. On constate également que les zones d'ombre n'ont pas engendré de faux obstacles, le "fond" a donc été parfaitement éliminé.

En améliorant encore la procédure de détection des bandes blanches, la configuration de la route, c'est à dire le nombre de voies et la largeur de la route, sont facilement identifiables. Puisque nous connaissons la position des bandes par rapport au véhicule, nous pouvons également le situer sur la chaussée. Enfin, en fiabilisant les détections, on pourrait même envisager une conduite automatique par un suivi des bandes blanches. Il faut toutefois signaler qu'une telle fonction n'est réalisable que sur des routes globalement rectilignes comme sur autoroute par exemple. En effet, lorsque le véhicule aborde un virage prononcé, le stéréoscope linéaire ne vise plus la route, les bandes blanches disparaissent alors des images.

7-4.2. Détection des piétons

7-4.2.1. Séquence piéton

La figure 7.16 montre le résultat obtenu sur la séquence stéréoscopique "piéton", dont les images brutes sont présentées au chapitre 2 (Cf. § 2-4.3.2). Cette séquence dure 20 secondes.



Figure 7.16 : Détection de piéton sur la séquence "piéton".

Deux types d'objets sont détectés dans cette séquence : les bandes blanches et le piéton. Les premières sont parfaitement détectées quand elles sont vues simultanément par les deux caméras. En ce qui concerne le piéton, son niveau de gris indique que le type qui lui a été assigné est bien le type PIETON, c'est à dire que sa largeur est comprise entre les seuils $\lambda_{\text{piéton}_{\min}}$ et $\lambda_{\text{piéton}_{\max}}$. Mais il faut surtout constater que les détections successives sont cohérentes, ce qui prouve l'efficacité des procédures de reconstruction et de suivi. Dans cette séquence, les différentes caractéristiques de l'obstacle, c'est à dire sa vitesse, son sens de déplacement et surtout la distance à laquelle il se trouve, sont parfaitement déterminées. Il est donc facile de déterminer si un risque de collision est possible.

En comparant ce résultat avec celui obtenu par la procédure hiérarchique de mise en correspondance (Cf. Figure 5.25), on peut faire plusieurs remarques :

- Les points isolés dus aux occlusions n'ont pas engendré d'objets. La gestion partielle des occlusions et le suivi temporel sont donc efficaces pour éliminer ce type de points.
- Au début de la séquence, le piéton entre dans le champ du stéréoscope linéaire puis disparaît. Le résultat de la figure 5.25 montre que le piéton a bien été détecté. Par contre, sur la figure 7.16, celui-ci n'est pas présent. En effet, le "fond" a été éliminé puisque nous ne pouvons faire la différence entre les véritables obstacles et les objets sans danger, comme les ombres par exemple.

7-4.2.2. Piéton traversant la chaussée

Notre véhicule roule maintenant sur une route dont la chaussée est délimitée par deux bandes blanches. Les deux courbes claires apparaissant de chaque côté de la route au début de la séquence correspondent à des bornes blanches de signalisation. Les deux rubans noirs entrelacés sont caractéristiques d'un piéton qui se déplace. Ils résultent de l'intersection du plan de visée avec les jambes du piéton. Lorsque celui-ci se déplace, le mouvement des jambes donne cette illusion d'entrelacement. Dans cette séquence, il traverse la route de la droite vers la gauche. A titre indicatif, la durée de cette séquence est de 7 secondes.

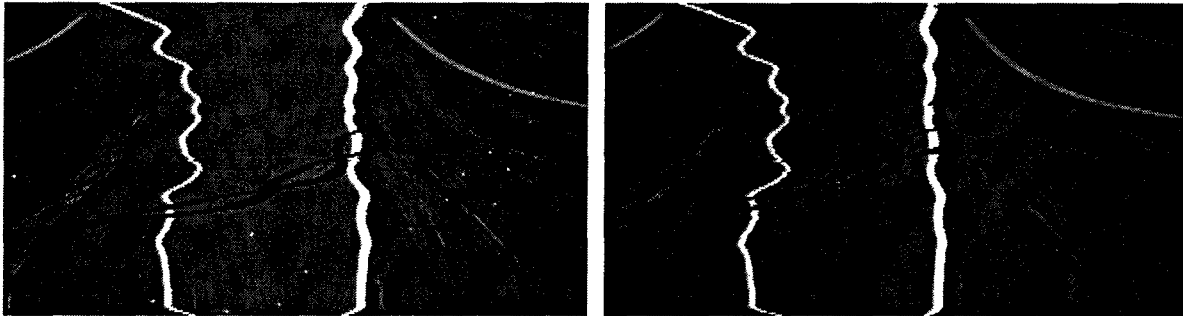


Figure 7.17 : Piéton traversant la chaussée.

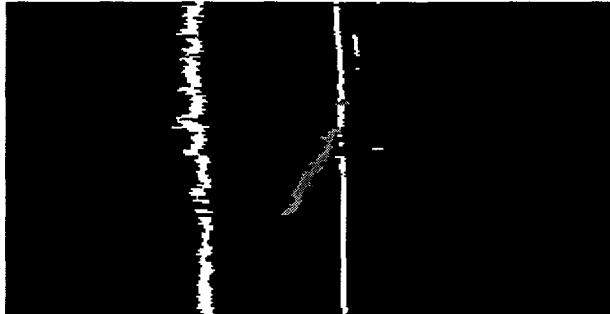


Figure 7.18 : Scène reconstruite à partir de la séquence de la figure 7.17.

Les bandes blanches sont parfaitement détectées sur cette séquence. On remarque néanmoins toujours le même problème avec la bande gauche. On distingue un trait blanc à côté de la bande droite. Il correspond à la borne qui a été prise pour une bande blanche. Il est vrai que les niveaux de gris très clairs de la borne se démarquent du fond.

Tant que le piéton se trouve en dehors de la route, il n'est pas détecté. En effet la zone de danger et les zones d'intérêt restreignent les traitements à la région située entre les bandes blanches. Dès qu'il commence à traverser, il est détecté en tant que PIETON. Il est ensuite suivi correctement puis disparaît soudain alors qu'il se trouve au milieu de la route. Ce phénomène s'explique en analysant la séquence de la figure 7.17.

Cette disparition correspond au moment où, dans les repères images, le piéton se retrouve à

gauche de la bande blanche gauche. Comme cette bande est toujours détectée, l'algorithme définit alors une zone de danger qui dans notre cas correspond à la zone délimitée par les bandes matérialisant la chaussée. Cependant dans les repères images, les contours engendrés par le piéton ne peuvent plus être pris en compte car ceux-ci se trouvent en dehors de la zone de danger, ou plus précisément, en dehors des zones d'intérêt. Le piéton n'est donc plus détecté. Toutefois la fiabilité de nos algorithmes n'est pas remise en cause. En effet, la disparition du piéton se produit alors que celui-ci se trouve sur la gauche du véhicule et à moins de 2 m de celui-ci. La proximité et les effets de perspective donnent alors l'illusion que le piéton est sorti de la route.

Il est important de noter qu'un l'algorithme qui détecterait un obstacle pour la première fois alors que celui-ci se trouverait très près du véhicule, ne permettrait pas d'éviter la collision. Il serait en effet trop tard pour déclencher une alarme. Dans notre cas, une alarme aurait été déclenchée à temps, puisque le piéton a été détecté au moment où il commençait à traverser.

7-4.2.3. *Détection d'un piéton situé dans la zone de danger*

La séquence est similaire à celle présentée sur la figure 7.17. On retrouve la chaussée, les bandes blanches, les courbes correspondant aux bornes de signalisation et un piéton. Celui-ci apparaît dans le champ du stéréoscope linéaire alors qu'il se trouve déjà sur la chaussée. Il se déplace ensuite vers la droite, quitte la route et fait quelques enjambées dans l'herbe. Il revient ensuite sur ses pas et traverse la route. A titre indicatif, la durée de cette séquence est de 12 secondes.

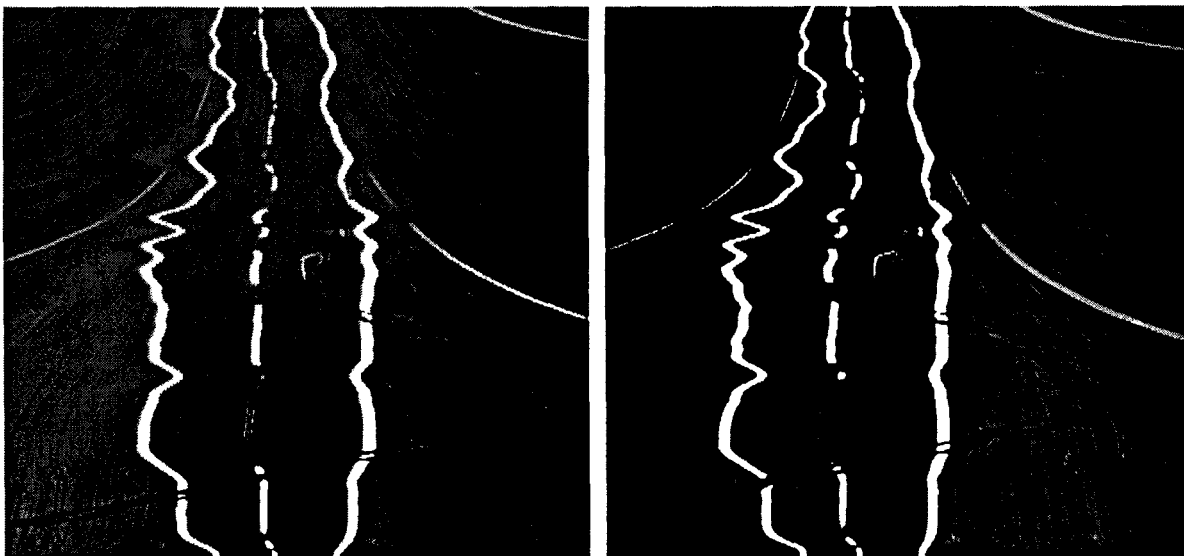


Figure 7.19 : Détection d'un piéton se trouvant dans la zone de danger.

Les résultats de cette séquence sont représentés sur la figure 7.20. Le piéton est détecté une première fois lorsqu'il entre dans le champ du stéréoscope linéaire. Quand celui-ci quitte la route, l'algorithme ne le détecte plus car il est sorti de la zone de danger. Par contre, il est de nouveau pris en compte, lorsqu'il revient sur ses pas pour traverser la chaussée. Il disparaît ensuite pour les mêmes raisons que dans l'exemple précédent.

Cet exemple met en évidence la restriction des traitements à la zone de danger. En effet le piéton n'est détecté que s'il se trouve sur la chaussée. Il faut toutefois préciser que la détection du piéton quand celui-ci est très près du véhicule n'est réalisable que sur une partie de la chaussée. Cependant, cette technique de restriction permet d'augmenter les cadences de traitement et donc d'évaluer plus rapidement si un risque de collision avec l'objet détecté est possible.

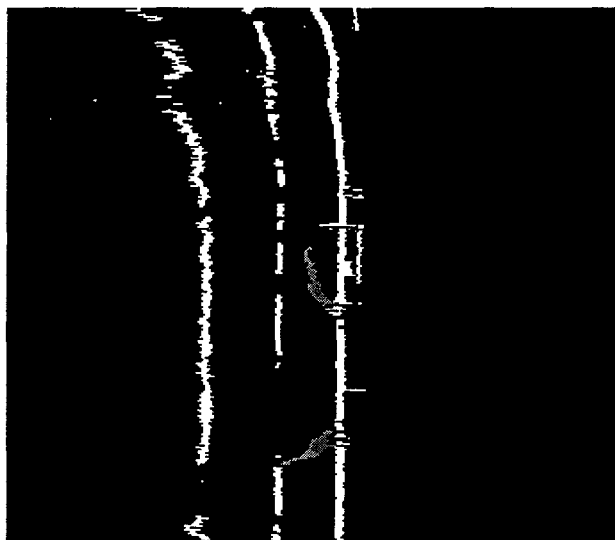


Figure 7.20 : Scène reconstituée à partir de la séquence de la figure 7.19.

7-4.3. Détection de véhicule

7-4.3.1. Suivi d'un véhicule

Nous roulons sur une route de campagne entourée d'herbes hautes. Nous nous rapprochons du véhicule qui nous précède et qui se manifeste sous la forme d'une large bande sombre.

L'ombre de la voiture est d'abord visible, puis les roues qui laissent entrevoir la chaussée. Puisque nous nous rapprochons toujours, les plans de visée de chaque caméra croisent ensuite le bas de caisse du véhicule. Nous nous laissons ensuite distancer avant d'accélérer de nouveau. Nous effectuons cette manoeuvre à plusieurs reprises pendant cette séquence. C'est pour cette raison qu'on distingue l'ombre, les roues et le bas de caisse dans cet ordre puis dans l'ordre inverse.

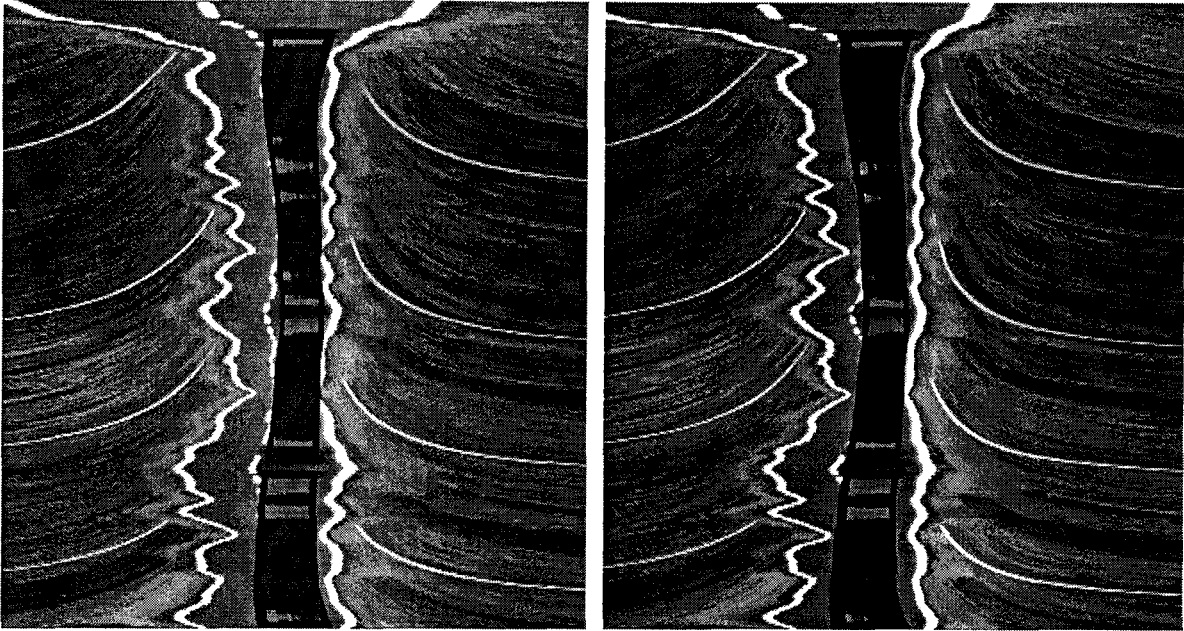


Figure 7.21 : Suivi d'un véhicule.

On remarque également les bornes blanches disposées régulièrement le long de la route qui se manifestent par des courbes blanches. A titre indicatif, la durée de cette séquence est de 14 secondes.

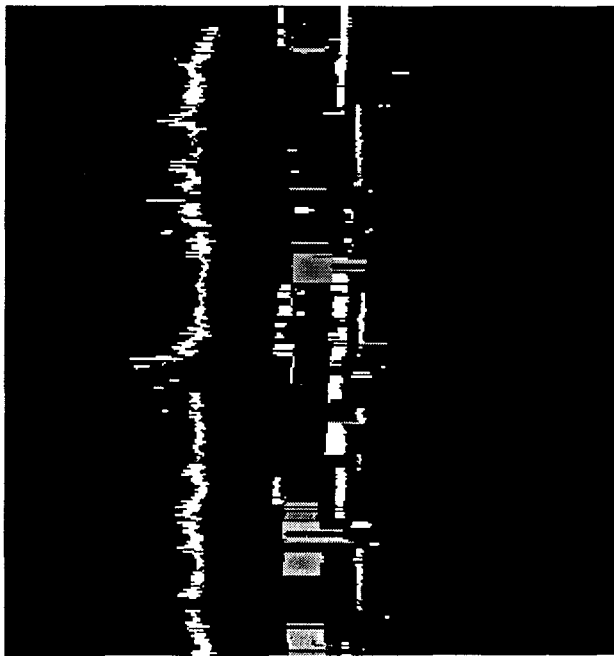


Figure 7.22 : Scène reconstituée à partir de la séquence de la figure 7.21.

La scène reconstituée montre nettement la bande blanche gauche. La bande droite est parfaitement reconstruite en début de séquence. Au milieu de la séquence, la bande centrale et celle de droite sont reconstruites, mais on constate qu'elles présentent une largeur trop importante. En effet, l'un des contours utilisé pour la reconstruction ne correspond pas à la

bande mais au véhicule. Cette confusion est due à la procédure d'extraction de contour. La conséquence principale est que le véhicule ne peut pas être détecté puisqu'un de ses contours a déjà été utilisé. C'est pour cette raison que le véhicule ne peut être reconstruit au milieu de la séquence. On distingue d'ailleurs sa forme entre les bandes blanches.

Pendant le premier et le dernier tiers de la séquence, le véhicule est correctement détecté et suivi par notre algorithme en tant que VOITURE. Cependant, une mauvaise estimation du paramètre Z_{min} conduit à sa suppression la plupart du temps. En effet, les bornes blanches situées sur le bord de la chaussée sont détectées en tant que "**bande blanche**" car elles sont très lumineuses. Elles interviennent donc dans la détermination de Z_{min} . Comme celles-ci sont fixes et que notre véhicule est mobile, leur distance par rapport à celui-ci diminue rapidement et le paramètre Z_{min} diminue en même temps. On arrive donc dans une situation où tous les objets situés à une distance supérieure à la borne sont éliminés même. On observe bien ce phénomène à la fin de la séquence. Lorsque la dernière borne apparaît, sa distance est supérieure à celle de l'obstacle, celui-ci est donc considéré comme un obstacle. Quelques instants plus tard, la situation s'inverse et celui-ci est supprimé.

Cet exemple montre qu'il faut être prudent lorsqu'on élimine les obstacles. En effet, l'ajustement des paramètres nécessaires à la sélection peut être mal effectué. Dans notre cas, on peut fixer une distance limite en dessous de laquelle le paramètre Z_{min} ne pourrait pas descendre. Mais la meilleure méthode serait d'améliorer notre procédure de détection des bandes blanches. On fiabiliserait ainsi la détermination du paramètre Z_{min} et on éviterait les confusions comme celles que nous avons rencontrées dans cet exemple.

7-4.3.2. Dépassement par un autre véhicule

Nous roulons sur une route bordée d'arbres et de buissons. Celle-ci est dépourvue de signalisation horizontale. Au tiers de la séquence, une voiture en train de nous doubler entre dans le champ du stéréoscope linéaire, puis se rabat sur la file de droite. Elle continue ensuite sur sa lancée puis disparaît. En fin de séquence, elle ralentit et entre de nouveau dans le champ du stéréoscope linéaire.

On constate que la route apparaît sous la forme d'un ruban gris très étroit. En effet, au cours de l'enregistrement de cette séquence, la portée du stéréoscope linéaire était d'environ 40 m. La largeur du champ correspondant était donc à peu près égale à 30 m. La partie intéressante de l'image que constitue la route était donc réduite. On remarque que la surface de la route n'est

pas uniforme. Elle présente en effet de nombreux raccord de bitume et de tâches.

A titre indicatif, la durée de cette séquence est de 12 secondes.

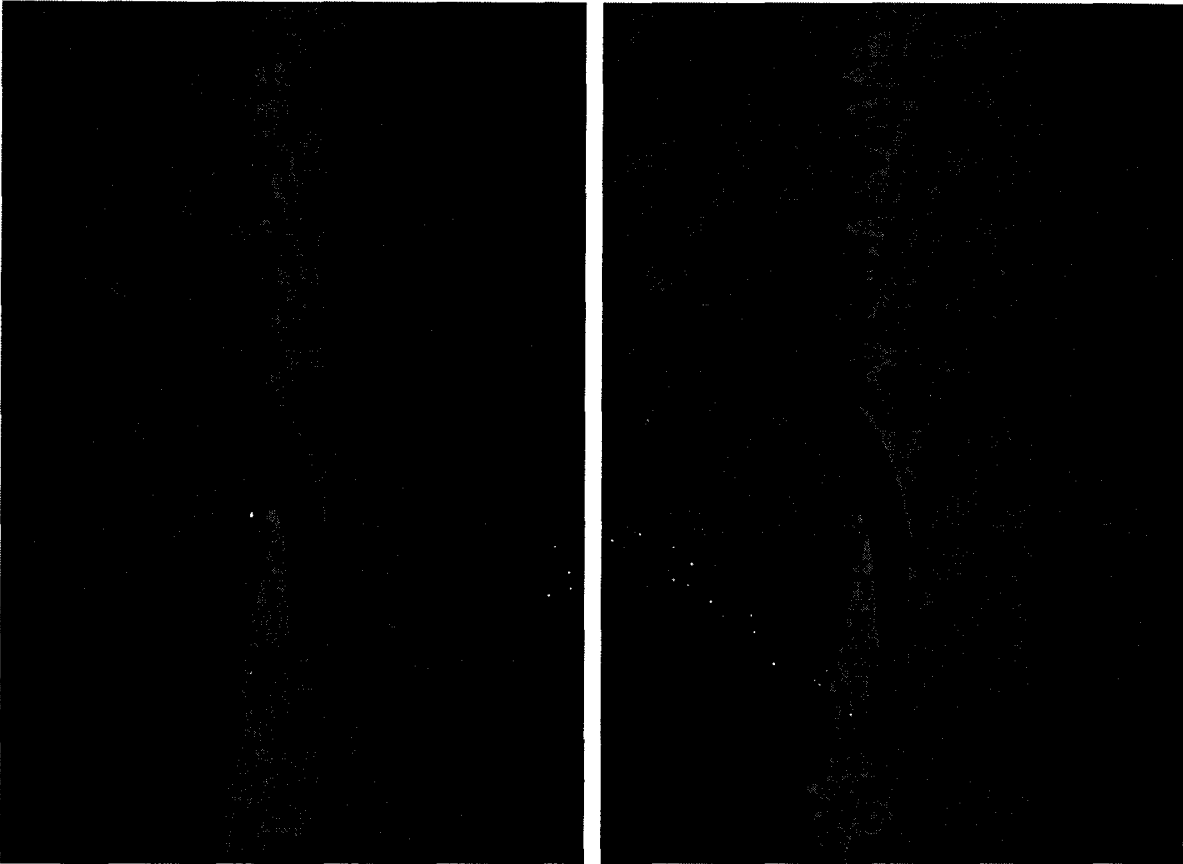


Figure 7.23 : Dépassement par un autre véhicule.

Lorsque notre algorithme est amené à traiter une telle séquence, il est obligé d'examiner les images dans leur intégralité. En effet, il ne dispose d'aucun point de référence lui permettant de définir des zones d'intérêts. Il doit se contenter de définir une zone de danger par défaut.

Dans la première moitié de la scène reconstituée (Cf. Figure 7.24), on constate l'apparition très brève de quelques objets. Ceux-ci sont engendrés par les herbes, les buissons ou encore les tâches présentes sur la chaussée. D'ailleurs, dans ce dernier cas, l'objet est détecté sur plusieurs images successives. Tous ces objets appartiennent à ce que nous appelons le "fond". En d'autres termes, ils se situent à la limite de portée du stéréoscope linéaire. Cette catégorie d'objets est éliminée par les paramètres Z_{min} et Z_{max} . Cependant, sur cette séquence, ils ne peuvent être définis précisément puisqu'il n'y a pas de bande blanche. Une valeur par défaut leur est donc affectée. S'ils permettent d'éliminer la majorité des objets appartenant au fond, il en reste néanmoins quelques uns. Il faut cependant préciser que leur durée de vie est très faible, ils ne seront donc pas considérés comme des obstacles.

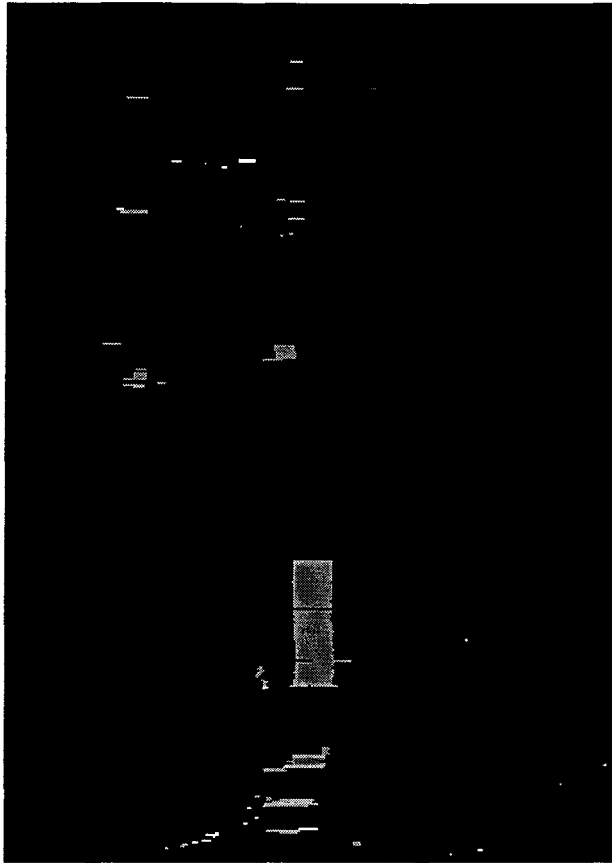


Figure 7.24 : Scène reconstituée à partir de la séquence de la figure 7.23.

En ce qui concerne le véhicule, celui-ci n'est pas détecté immédiatement. Il faut attendre la fin de la manoeuvre de dépassement pour que notre algorithme le repère. Lorsque le véhicule apparaît, seul son côté droit est visible. Un seul point de contour est donc détecté ce qui n'est pas suffisant pour reconstruire un objet. Par contre, lorsque le dépassement est terminé, le véhicule est néanmoins correctement détecté et ensuite suivi pendant plusieurs secondes. Le niveau de gris indique de plus qu'il a bien été identifié en tant que VOITURE. Alors qu'il s'éloigne, il n'est soudain plus détecté bien qu'il soit toujours présent sur les images. En fait, la distance à laquelle il se trouve, est comprise entre les valeurs Z_{min} et Z_{max} par défaut. Il est alors éliminé car il est considéré comme appartenant au fond. Il est cependant de nouveau détecté quand le véhicule ralentit.

Malgré les quelques détections manquées, notamment quand l'obstacle est légèrement sur le côté du véhicule, il faut noter que le bon fonctionnement de l'algorithme permet de détecter correctement les obstacles présents devant le véhicule. Cependant, il faut également préciser que notre algorithme ne conduit pas à des résultats aberrants. En effet, celui-ci élimine les obstacles incohérents et évite ainsi le déclenchement intempestif d'une alarme.

7-5. Conclusion

Un système de détection d'obstacles doit non seulement détecter les objets présents en face du véhicule mais doit également les identifier. En effet, pour déterminer si un objet est dangereux, il faut avant tout connaître son comportement. Les techniques présentées dans ce chapitre permettent la détection et le suivi des obstacles généralement rencontrés dans un contexte routier, c'est à dire les piétons et les véhicules.

Les procédures de reconstruction et de suivi terminent la description de notre système de détection d'obstacles. Celui-ci a été testé sur des séquences enregistrées tirées de situations réelles. Les résultats obtenus montrent que, dans la majorité des cas, notre système permet de détecter un obstacle se trouvant à l'avant du véhicule. Cependant les temps de calculs ne permettent pas de prévenir le conducteur suffisamment tôt pour qu'il puisse modifier le comportement de son véhicule en cas de danger. En effet, les cadences atteintes avec un PC 486 DX 33 ne sont que de 2 à 3 couples d'images par seconde. Nos algorithmes ont donc été implantés sur une carte de traitement autorisant des cadences de traitement importantes afin que notre système de détection d'obstacles soit intégré à un véhicule expérimental. Cette implantation fait l'objet du chapitre suivant.

Chapitre 8

LE VEHICULE EXPERIMENTAL

PROLAB 2

8-1. Introduction

L'ensemble des travaux décrits dans ce mémoire, ainsi que ceux réalisés préalablement par Jean-Luc Bruyelle lors de sa thèse, ont été accomplis dans un but précis : réaliser une fonction particulière du véhicule expérimental Prolab 2. Ce véhicule a été développé par le groupe ProArt France dans le cadre du programme Prometheus (Cf. § 8-2). La tâche qui nous avait été confiée dans le cadre de ce projet était la détection des piétons. Le système de détection d'obstacles que nous avons réalisé a donc été implanté dans le véhicule expérimental (Cf. § 8-3). Prolab 2 a été présenté aux pouvoirs publics et à la presse lors d'une démonstration clôturant le programme Prometheus (Cf. § 8-4) ainsi qu'à diverses reprises (Cf. § 8-5) avec un grand succès.

8-2. Le véhicule expérimental Prolab 2

Lors de la deuxième phase du programme Prometheus, qui s'est déroulée de 1991 à 1994, la décision de réaliser un deuxième démonstrateur Prolab 2 a été prise par les responsables du groupe ProArt France.

Avant de présenter le véhicule Prolab 2, nous évoquerons brièvement le véhicule Prolab 1 développé au cours de la première partie du programme Prometheus.

8-2.1. Démonstrateur Prolab1

Prolab 1 a été conçu par l'ensemble des laboratoires universitaires impliqués dans le groupe ProArt France pour la période 1988-1991.

Deux laboratoires étaient alors particulièrement concernés. L'équipe S.M.T.I du Laboratoire d'Electronique de Clermont-Ferrand (aujourd'hui LASMEA) était chargée de l'implantation matérielle et logicielle et surtout de l'ensemble de l'aspect perception à bord du véhicule. Le comportement dynamique du véhicule ainsi que le système expert ont été traités par le Laboratoire Heudiasyc de l'Université de Compiègne. L'interface homme-machine fut réalisée par la société Dialogics (aujourd'hui Dialexis).

Prolab 1 n'était pas un véhicule autonome. Le but de ce véhicule démonstrateur était de fournir des informations au conducteur sans agir sur le véhicule. Il était doté d'un certain nombre de modules :

- Le module "perception" permettait de connaître l'état intrinsèque du véhicule ainsi que l'environnement dans lequel il évoluait.
- Le module "cinématique" était chargé de calculer les distances de sécurité ainsi que les paramètres de dépassement en fonction de l'état intrinsèque du véhicule et de l'état dynamique des obstacles.
- Le module "décision", basé sur un système expert, centralisait l'ensemble des informations avant de présenter au conducteur celles qu'il avait jugées pertinentes. Il agissait également sur le mode de fonctionnement du capteur multisensoriel.
- L'interface homme-machine présentait les informations au conducteur sous forme de symboles.

L'ensemble de ces modules a été implanté dans un véhicule expérimental de type Renault 21, baptisé Prolab 1.

Prolab 1 a été présenté au public à plusieurs reprises et notamment au « Board Member Meeting Prometheus » qui s'est tenu à Turin en septembre 1991. Il faut noter que lors de cette manifestation, Prolab 1 était le seul véhicule routier expérimental français issu de la recherche. Les scénarii proposés étaient simples. Le conducteur était prévenu d'un éventuel danger en cas de changement de voie ou bien s'il abordait un obstacle à trop vive allure.

La suite de cette partie sera consacrée au véhicule expérimental Prolab 2.

8-2.2. Le contexte scientifique

Le véhicule Prolab 2 est le fruit de la collaboration des 9 laboratoires participant à ProArt France :

- LASMEA de Clermont-Ferrand URA CNRS 830.
- LIFIA Grenoble URA CNRS 394.
- HEUDIASYC Compiègne URA CNRS 817.
- INRIA Sophia Antipolis.
- IRISA Rennes.
- Ecole des Mines de Paris.
- Ecole des Mines de Fontainebleau.
- Ingénia Dialexis Toulouse.

et le

- Centre d'Automatique de Lille.

Chacun de ces laboratoires avait une tâche bien définie au sein du projet. Les différentes tâches à accomplir étaient les suivantes :

- La détection de véhicules à l'avant.
- La détection de véhicules à l'arrière.
- La détection de véhicules approchant sur les cotés (à l'arrêt au stop).
- La détection de la signalisation horizontale.
- La détection des piétons, des deux-roues et des animaux.
- La fusion des données transmises par les différents capteurs.
- Le copilottage.
- L'interface homme-machine.

La tâche confiée au Centre d'Automatique de Lille dans le cadre de ce projet était la détection des piétons.

Comme pour Prolab 1, l'objectif de Prolab 2 est d'aider le conducteur à détecter les dangers et les obstacles autour du véhicule.

8-2.3. L'équipement de Prolab 2

Pour pouvoir réaliser ces différentes fonctions, le démonstrateur, une Peugeot 605, a été doté d'un certain nombre d'éléments.

8-2.3.1. Les capteurs proprioceptifs

Les informations issues de ces capteurs permettent de connaître l'état dynamique du véhicule et les intentions du conducteur. Les différents capteurs placés dans le véhicule permettent d'évaluer :

- Les accélérations longitudinale et latérale.
- La vitesse de lacet du véhicule.
- Le freinage sur les roues.
- La vitesse de rotation des roues.
- L'angle du volant.
- La vitesse de rotation du moteur.
- L'état des clignotants.

Les données provenant de ces capteurs sont transmises au calculateur principal.

8-2.3.2. Les capteurs extéroceptifs

Ces capteurs sont utilisés pour percevoir les éléments extérieurs au véhicule. Dans Prolab 2, ce sont essentiellement des capteurs visuels :

- Une caméra à l'avant.
- Une caméra à l'arrière.
- 2 caméras latérales, une dans chaque rétroviseur extérieur.
- Un télémètre laser couplé à une caméra à l'avant du véhicule.
- Le stéréoscope linéaire composé de 2 caméras linéaires.

Le véhicule est également équipé d'un émetteur-récepteur infrarouge spécifique :

- Détecteur de balise active.

L'ensemble de ces capteurs est représenté sur la figure 8.1.

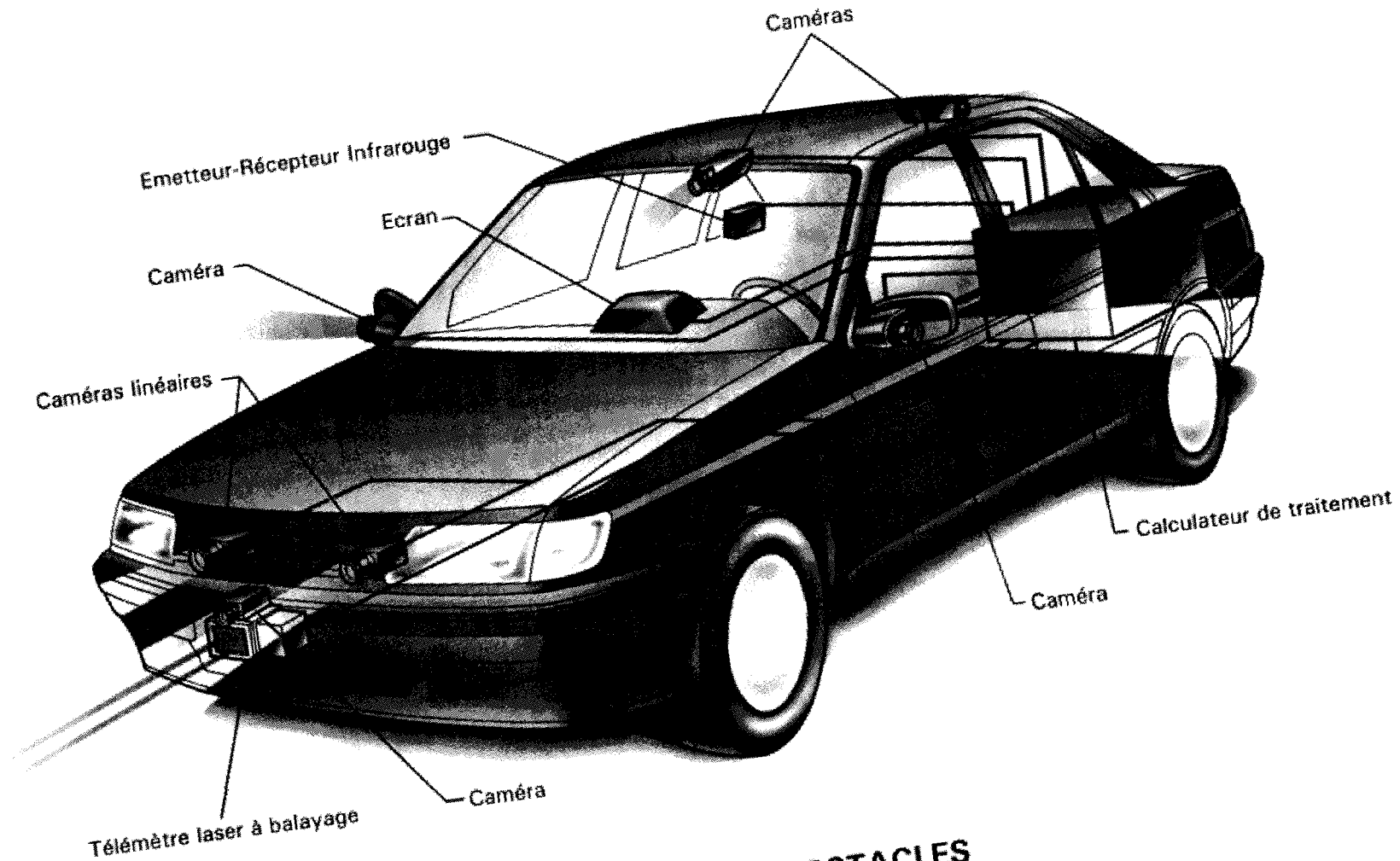
Ces capteurs ne sont pas suffisants pour couvrir toutes les zones entourant le véhicule. De ce fait, celui-ci possède des zones aveugles mais le système informatique les gère automatiquement, ce qui permet de garder en mémoire l'ensemble des obstacles présents autour du véhicule.

8-2.3.3. L'interface homme-machine (IHM)

Le véhicule est doté d'une interface multimodal disposant de trois niveaux d'aide : alarme, conseil et assistance. Son rôle est de présenter de façon synthétique les informations au conducteur. Il s'agit :

- D'une interface visuelle : un écran VGA couleur a été implanté dans le tableau de bord, à la droite du conducteur. Il est chargé d'assister, de conseiller ou d'alerter le conducteur grâce à des logos représentatifs de la situation dans laquelle il se trouve.
- D'une interface sonore : quatre haut-parleurs formant un système quadriphonique sont installés aux quatre coins de l'habitacle. A la plupart des logos sont associés des signaux sonores brefs et caractéristiques dont l'intensité dépend du message à faire passer au conducteur, conseil ou alarme.

Figure 8.1 : Le prototype expérimental Prolab 2.



Le véhicule expérimental PROLAB 2

DETECTION D'OBSTACLES

PEUGEOT 605 PROLAB 2



8-2.3.4. Architecture informatique

La partie informatique gérant l'ensemble du système a été installée dans le coffre du véhicule (Cf. Figure 8.1). Elle est constituée de différents éléments reliés entre eux, à savoir :

- Un rack VME comprenant deux systèmes de vision et le système de contrôle du télémètre.
- Deux PC comprenant chacun un module de perception directement relié à sa ou ses caméras.
- Un PC dédié au fonctionnement de l'interface homme-machine. Il a également la charge de lire les données provenant des capteurs proprioceptifs et celles venant des balises actives.
- Une carte passerelle : l'architecture hybride VME-PC nécessite l'utilisation d'une carte permettant la communication entre ces deux mondes différents. La liaison entre chaque PC et la carte passerelle se fait par l'intermédiaire d'une liaison série.

8-2.4. Fonctionnalités de Prolab 2

Prolab 2 est un système d'aide à la conduite. Son rôle est d'assister le conducteur lors de la réalisation de manoeuvres et de l'informer des dangers potentiels ou réels lorsqu'il est en situation de conduite. Les autres véhicules ne sont pas coopérants, c'est à dire qu'ils ne sont pas équipés de systèmes particuliers qui les rendraient différents des véhicules actuels. Les différentes fonctionnalités du démonstrateur sont les suivantes :

- Suivi de véhicule.
- Dépassement d'obstacle.
- Changement de voie.
- Détection de véhicules en carrefour.
- Détection et identification du marquage au sol.

et la fonctionnalité sur laquelle nous avons travaillé :

- Détection de piétons.

Pour réaliser ces fonctionnalités, le véhicule Prolab 2 n'utilise pas d'infrastructure différente de celle qui existe aujourd'hui. Une seule petite exception, l'utilisation de balises actives qui peuvent indiquer au véhicule Prolab 2 l'environnement statique, c'est à dire l'infrastructure de la route. Par exemple, lors d'une entrée sur autoroute la balise indique le nombre de voies, ou à l'approche d'un carrefour, elle peut indiquer le nombre de voies, la priorité des voies, etc.

Rien ne permet a priori de connaître l'environnement dynamique du véhicule, c'est à dire les obstacles et les piétons.

8-2.5. Architecture globale

L'architecture globale de Prolab 2 peut être divisée en trois modules :

- La perception.
- La décision.
- L'information.

Mais il faut également signaler un module important qui sert d'interface entre la perception et la décision et dont le rôle est de fusionner l'ensemble des données provenant des capteurs.

Nous présentons brièvement ces modules dans les paragraphes suivants.

8-2.5.1. La perception

Ce module comprend à la fois les capteurs proprioceptifs donnant les informations de l'état interne du véhicule et les capteurs extéroceptifs dont le rôle est de percevoir l'environnement du véhicule.

Les capteurs extéroceptifs, qui sont principalement des caméras, sont reliés à leur propre système informatique chargé des traitements "bas niveau". Chaque système permet donc d'analyser les données provenant de son capteur afin d'en extraire les informations utiles, soit sur l'environnement statique du véhicule, c'est à dire la signalisation horizontale (passage piéton, flèches de présélection, délimiteurs de voie, nombre de voies, etc.), soit sur l'environnement dynamique, c'est à dire la présence d'obstacles potentiels et leurs positions autour du véhicule.

Tous les capteurs ne fonctionnent pas continuellement. C'est le module de décision qui active sur requête le capteur approprié afin d'obtenir une information précise sur l'environnement du véhicule.

Les résultats de chaque système de perception sont envoyés ensuite vers le module interface perception/décision.

8-2.5.2. L'interface perception/décision

Ce module très important est chargé de la fusion et de la gestion dynamique des données provenant du module de perception.

Les capteurs du module perception travaillent de façon asynchrone et à des fréquences différentes. Ils fournissent des informations parfois redondantes, parfois complémentaires. Par contre, certaines données sont manquantes en raison du nombre insuffisant de capteurs. L'ensemble des données est donc géré dynamiquement pour obtenir l'image la plus complète et précise de l'état du système. Ce module est également chargé de la gestion des zones aveugles du véhicule, c'est à dire des angles morts. Pour cela, la cohérence temporelle des obstacles est maintenue afin d'avoir en permanence une estimation des dangers.

Les données sont ensuite transmises au module de décision.

8-2.5.3. La décision

Le module de décision, appelé copilote, est la partie centrale du système d'aide à la conduite. Son objectif est double. Le premier est de déterminer à chaque instant l'état du véhicule et sa situation par rapport à son environnement. Le second est d'assister le conducteur en lui signalant tout danger ou en lui suggérant les manoeuvres possibles en fonction de la situation. Pour réaliser cette aide à la conduite le copilote est décomposé en plusieurs sous-modules :

8-2.5.3.1. Analyse de situations

Ce sous-module constitué de plusieurs systèmes experts temps-réel a pour rôle essentiel d'effectuer un diagnostic pour détecter toute situation dangereuse. En fonction de celle-ci, il déclenche la fonction permettant la gestion d'une manoeuvre en activant le sous-module de *supervision de manoeuvres*. Il peut également demander au sous-module de *gestion de la perception* des informations complémentaires permettant d'effectuer et/ou d'améliorer le diagnostic de la situation.

8-2.5.3.2. Supervision de manoeuvres

Ce sous-module est activé par l'analyse de situation lorsque le conducteur a émis son intention d'effectuer une manoeuvre ou lorsque la situation impose d'effectuer une manoeuvre pour éviter une collision.

8-2.5.3.3. Gestion de la perception

Ce sous-module active ou désactive le fonctionnement des capteurs, sur requêtes du sous-module d'analyse de situation. En effet, le module de perception possède plusieurs modes de fonctionnement :

- Surveillance continue : on souhaite observer de façon continue l'évolution des données.
- Information ponctuelle : on souhaite, à un instant donné, une information qui n'est pas habituellement calculée.
- Détection d'événement : l'arrivée d'un événement particulier peut modifier le raisonnement.

8-2.5.4. L'information

Ce module, constitué de l'interface homme-machine, est destiné à transmettre au conducteur les messages importants du copilote. Les messages ont été étudiés pour être assimilables très rapidement et apparaître comme une aide et non une contrainte. Trois niveaux d'aides ont été définis :

- **Alarme** : le pilote est averti d'une situation dangereuse ou pouvant le devenir.
- **Conseil** : un conseil peut être donné au conducteur s'il le demande.
- **Aide** : un type de conduite est proposé au conducteur.

Les messages utilisés sont à la fois visuels et sonores.

8-2.6. Conclusion sur le véhicule expérimental Prolab 2

Nous avons présenté brièvement le véhicule expérimental Prolab 2 ainsi que ces fonctionnalités et son architecture générale. Comme il n'est pas de notre propos de détailler l'ensemble des modules implantés dans le véhicule, ce qui serait d'ailleurs très long, nous n'approfondirons pas ce sujet. Cependant, nous détaillerons la fonction de Prolab 2 qui nous a été confiée : la détection des piétons. Le lecteur intéressé trouvera de plus amples informations sur les travaux réalisés dans le cadre du groupe ProArt France dans le rapport rédigé par l'ensemble des laboratoires ayant participé au programme [ProArt 95].

8-3. Détection des piétons par le véhicule Prolab 2

La détection des piétons est une fonction originale de Prolab 2. En effet, peu de recherches sont faites dans ce domaine et d'ailleurs aucun prototype réalisé dans le cadre de Prometheus n'avait de fonction similaire.

Pour réaliser la détection des piétons, des modifications ont du être apportées d'une part au stéréoscope linéaire pour pouvoir l'intégrer au véhicule et, d'autre part aux algorithmes pour permettre une application temps-réel.

8-3.1. Intégration du stéréoscope linéaire

8-3.1.1. Stéréoscope linéaire

Le stéréoscope linéaire a été installé entre les phares de Prolab 2. Son intégration a nécessité la réalisation d'un deuxième prototype car le premier était trop encombrant et trop lourd pour pouvoir s'insérer à cet emplacement. Le second prototype devait donc être plus léger tout en conservant une bonne rigidité. En effet, toute déformation du stéréoscope linéaire due aux mouvements du véhicule nuirait au calibrage et pourrait altérer la fiabilité de la détection. Un nouveau support respectant ces contraintes a été conçu et réalisé par le Département Génie Mécanique et Productique de l'IUT A de Villeneuve d'Ascq.

Pour pouvoir s'insérer aisément entre les phares et s'adapter au milieu routier, les caractéristiques du stéréoscope linéaire ont été légèrement modifiées. L'entre-axe a été fixé à 465 mm. En ce qui concerne la distance focale des objectifs, nous avons opté pour des 35 mm, ce qui permet d'avoir un champ de vision de 40°. Enfin, le stéréoscope linéaire a été calibré pour que le plan optique rencontre la chaussée à environ 30 m à l'avant du véhicule. Il faut préciser que la procédure de calibrage est réalisée en plaçant les caméras sur un support rigide au lieu de les mettre sur le véhicule. En effet, ce dernier possède une suspension qui provoque une modification de l'assiette lorsqu'une force extérieure lui est appliquée. Malheureusement il en est de même lorsqu'on manipule les vis de la monture des caméras ce qui rend le calibrage du stéréoscope impossible.

La figure 8.2 montre le second prototype du stéréoscope linéaire placé entre les phares du véhicule lors de son installation. Pour permettre un accès facile au stéréoscope linéaire la calandre et le pare-chocs sont facilement démontables. On peut également voir sur cette figure, le multiplexeur (Cf. § 2-2.3.2) fixé entre les caméras. On remarquera la présence du télémètre laser sous le stéréoscope linéaire.

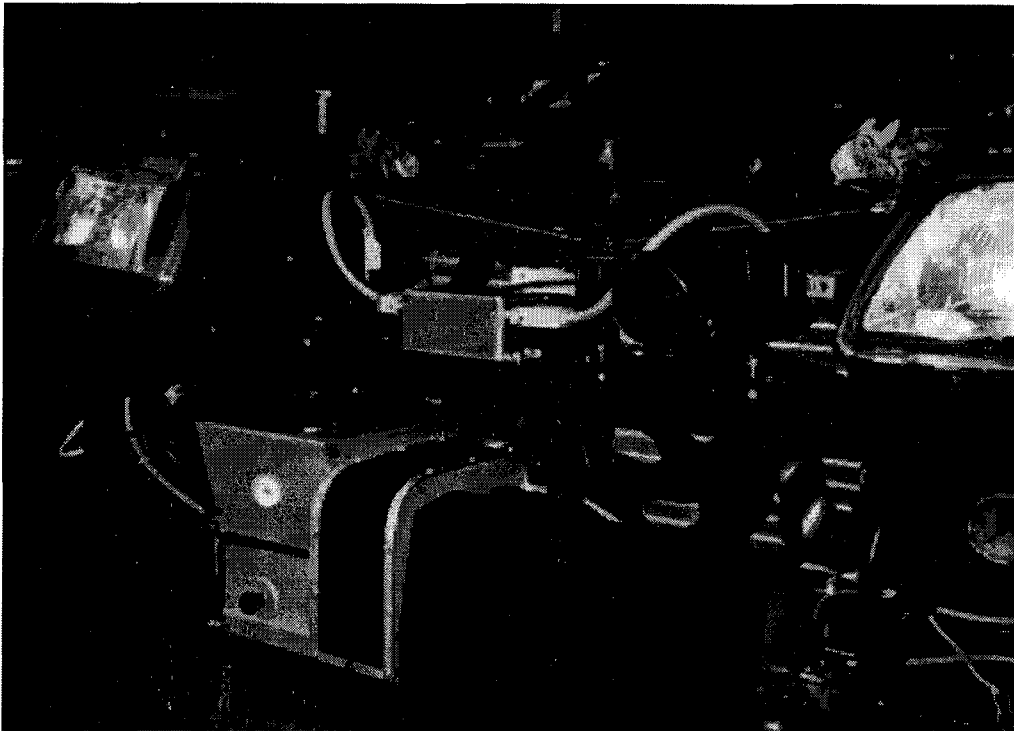


Figure 8.2 : Le second prototype du stéréoscope linéaire installé sur Prolab 2.

La figure 8.3, montre cette fois-ci le stéréoscope linéaire après sa complète intégration. Seuls les objectifs des caméras sont visibles. On remarquera également l'orifice présent dans le pare-chocs, laissant entrevoir le télémètre laser.



Figure 8.3 : Le stéréoscope linéaire après complète intégration.

L'alimentation des caméras est fournie par des cartes de type ISA insérées dans le PC-AT gérant le stéréoscope linéaire (Cf. § 8-3.1.2). L'acquisition des images est réalisée grâce à une

carte d'acquisition linéaire Data Translation DT2856 située également dans le PC-AT. Le stéréoscope linéaire, situé à l'avant du véhicule, est relié au PC-AT placé à l'arrière par des câbles passant sous le véhicule. Ces câbles permettent d'alimenter les caméras, de leur envoyer les signaux de commande et de récupérer les images linéaires.

8-3.1.2. Architecture informatique

L'architecture informatique du stéréoscope linéaire est réalisée sous la forme d'un ensemble électronique dont le support est un ordinateur de type PC-AT. Ce type de machine présente de nombreux avantages :

- Sa puissance de calcul et son coût peu élevé sont jugés réalistes pour le type d'application envisagée.
- Ses dimensions et sa consommation électrique sont modestes. L'adaptation à un véhicule automobile en est facilitée.
- Son bus ISA permet la réalisation simple de cartes d'extension pour répondre aux besoins particuliers de l'application. De plus, de nombreuses cartes existent dans le commerce, ce qui permet généralement de faire l'économie du développement de cartes spécifiques.

Malgré ces avantages, les systèmes de type PC-AT sont difficilement compatibles avec des applications temps-réel nécessitant des calculs importants. Pour résoudre ce problème, une carte spécialisée a été insérée dans le PC-AT. Cette carte dédiée aux traitements est de type Alacron AL860-XP et comporte :

- 2 processeurs RISC i860 cadencés à 50 MHz.
- 8 Mo de mémoire DRAM partagés par les deux processeurs.

La carte s'insère dans l'un des slots ISA de la carte mère du PC-AT. Elle permet ainsi d'obtenir une puissance de calcul très importante tout en conservant une architecture de type PC-AT.

Comme nous l'avons signalé au paragraphe précédent, le PC-AT contient également les cartes d'alimentation des caméras ainsi que la carte d'acquisition d'images linéaires.

Dans le cadre du projet, le PC-AT choisi est équipé d'un processeur très simple : un Intel 486SX cadencé à 33 MHz. Trois processeurs sont donc disponibles pour la détection des obstacles en temps-réel. Le PC-AT, ainsi que tous les systèmes de traitement sont situés dans le coffre arrière de Prolab 2 (Cf. Figure 8.4).

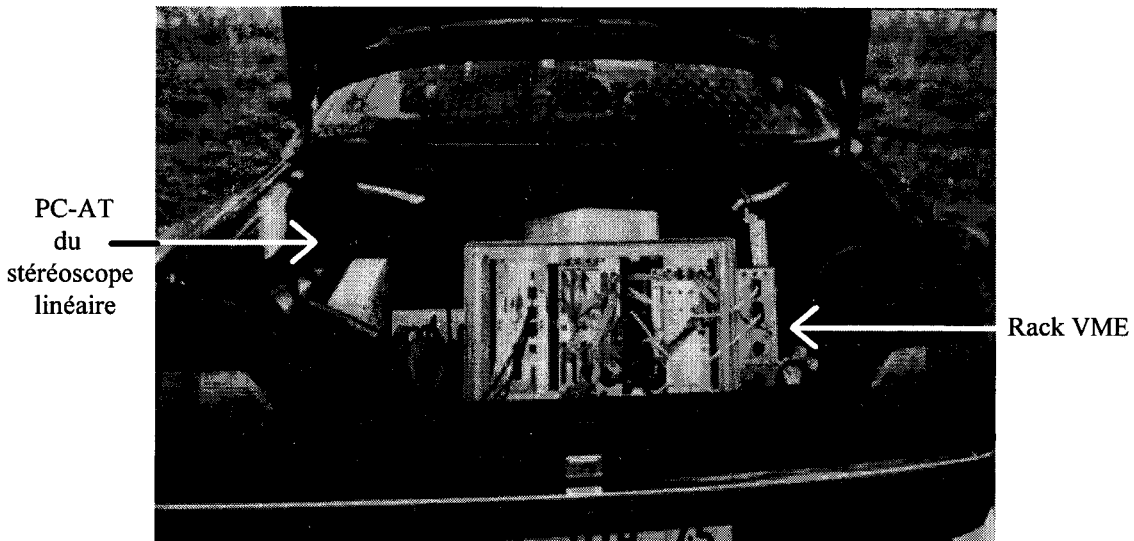


Figure 8.4 : Systèmes informatiques de Prolab 2.

8-3.2. Implantation temps-réel des traitements

Ce paragraphe présente la démarche que nous avons retenue pour implanter nos algorithmes. Nous y détaillons également l'étude que nous avons faite pour paralléliser les tâches sur la carte de traitement. Toutefois, il est apparu, au cours des développements, que les performances obtenues avec un seul processeur i860 étaient supérieures à celles imposées par les cadences de traitements. Pour des raisons d'efficacité, l'implantation sur les deux processeurs n'a donc pas été complètement réalisée, mais elle reste envisageable à court terme.

8-3.2.1. Rôle des processeurs

Pour permettre une implantation temps-réel des algorithmes, l'ensemble des opérations informatiques est partagé entre les 3 processeurs disponibles travaillant simultanément.

8-3.2.1.1. Rôles des processeurs i860

La détection des obstacles nécessitant d'importants calculs, ceux-ci sont confiés à la carte de traitement. La totalité des procédures exécutées sur cette carte est télé-chargée par le 486 du PC-AT à l'initialisation, via le bus ISA. Par la suite, les procédures de traitement sont accessibles depuis le 486 par un simple appel. A partir de l'instant où les images arrivent dans la mémoire de la carte, celle-ci va effectuer la totalité des traitements nécessaires à la détection, à savoir :

- l'extraction des primitives.
- la mise en correspondance.
- la reconstruction tridimensionnelle des obstacles.
- le suivi spatio-temporel.

8-3.2.1.2. Rôle du processeur 486 du PC-AT

La partie la plus gourmande en temps de calcul ayant été confiée à la carte de traitement, il ne reste plus au 486 qu'à gérer les différents éléments du système, c'est à dire :

- l'acquisition.
- le transfert des images de la carte d'acquisition vers la carte de traitement.
- le lancement des traitements.
- le transfert des résultats de la carte de traitement vers la mémoire du PC.
- les communications des résultats au copilote.

Bien que les tâches à effectuer par le 486 soit assez nombreuses, chacune d'entre elles ne nécessite que peu de temps d'occupation du processeur. Ceci explique l'utilisation d'un processeur SX33 qui n'est pas spécialement rapide.

8-3.2.2. Parallélisation des tâches sur les i860

La carte AL860-XP est équipée de deux processeurs pouvant travailler simultanément. Le parallélisme est géré par un mécanisme très simple basé sur l'utilisation de sémaphores afin d'éviter les conflits lors des différents accès à la mémoire commune ou au bus d'échanges.

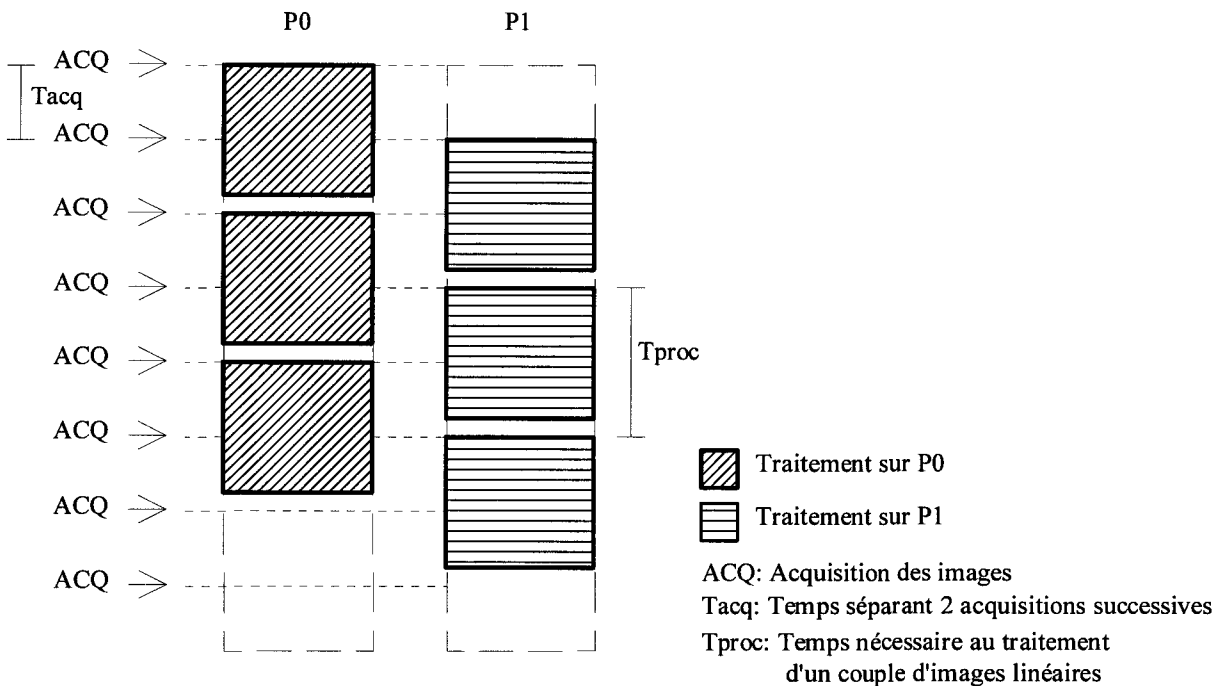


Figure 8.5 : Mode multiprocesseurs par division du temps de travail.

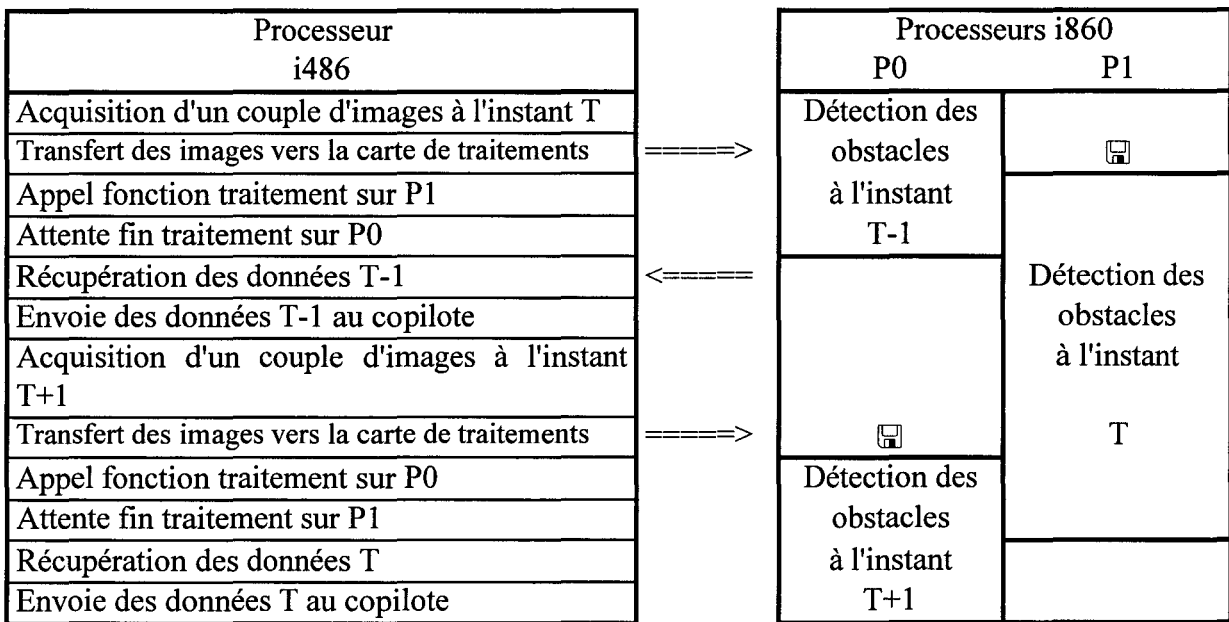
Le modèle général pour les algorithmes parallèles est de diviser le temps de travail pour accomplir une tâche donnée. Ceci s'effectue généralement de deux manières: la charge de travail est divisée soit en temps, soit en volume. Dans notre cas, la meilleure solution est la

division temporelle. En effet, les traitements appliqués à un couple d'images stéréoscopiques sont exécutés séquentiellement. Pour exploiter le parallélisme, les données images sont donc assignées alternativement au premier (P0) ou au second (P1) processeur i860 (Cf. Figure 8.5). De cette manière, on multiplie la cadence des traitements par deux.

8-3.2.3. Relation Maître-Esclaves

Le système étant constitué de trois processeurs distincts, une relation de type maître-esclaves doit être établie de façon à pouvoir les exploiter simultanément et optimiser les temps de traitement.

Le rôle du processeur maître est confié au 486 du PC-AT. Au démarrage du système, il initialise d'une part la carte d'acquisition en lui spécifiant les caractéristiques des caméras utilisées, et d'autre part la carte de traitement en téléchargeant l'ensemble des fonctions qui seront utilisées par la suite. Il établit également la communication avec le copilote. Les tâches s'effectuent ensuite selon le tableau suivant :



☐ : Réception des images sur la carte de traitement.

Cette implantation permet une utilisation optimale des processeurs i860, qui ont en charge la détection des obstacles. Sous le contrôle du i486, ils ne fonctionnent que par appels venant du processeur maître (Cf. Figure 8.6).

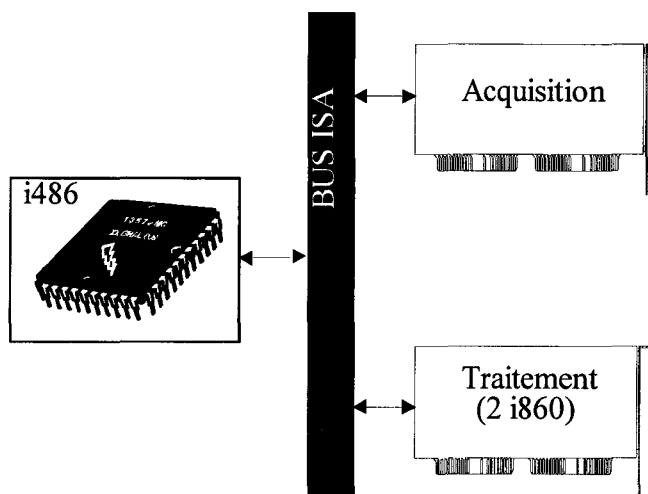


Figure 8.6 : Architecture multiprocesseurs.

8-3.2.4. Implantation logicielle réalisée pour Prolab 2

Tous les algorithmes développés ont été implantés sur la carte de traitement. Cependant il est apparu que la vitesse de traitement obtenue avec un seul processeur i860 était nettement suffisante pour transmettre au copilote les informations nécessaires sur les obstacles détectés par le stéréoscope linéaire. En effet, le copilote ne demandait la réactualisation de ces informations que toutes les 200 ms alors que le stéréoscope pouvait fournir des données rafraîchies toutes les 50 à 100 ms. Les performances de notre système stéréoscopique de détection d'obstacles étant supérieures à celles imposées par le copilote, nous avons implanté nos algorithmes sur un seul des processeurs de la carte de traitement. Nous avons cependant gardé la même répartition des tâches entre le processeur i486 et le processeur i860. De même, le cycle de traitement évoqué ci-dessus n'a pas été modifié. Seule la gestion du parallélisme n'a pas été implantée.

8-3.3. Protocole de communication

Le stéréoscope n'étant qu'un des nombreux systèmes de perception installés sur le véhicule, il est raccordé par l'intermédiaire d'une liaison série à l'interface perception/décision (Cf. § 8-2.5.2) qui fusionne l'ensemble des données provenant des différents capteurs. La détection des piétons n'étant utilisée que pour certains scénarii, le PC doit savoir à quel moment il doit être opérationnel et envoyer les données. Un protocole d'échanges du copilote vers le PC a donc été défini avec les messages suivants :

- MSG_INIT : fixe une date de référence commune à tous.
- MSG_BEGIN_LILLE : lance les traitements pour la détection des obstacles.

- MSG_TO_LILLE : début de l'envoi des données vers le copilote.
- MSG_FIN : arrêt des traitements.
- MSG_SYNCHRO : re-synchronisation.
- MSG_TEST : test de communication.

De la même manière, lorsque les données sont présentes, elles sont transférées du PC vers le copilote selon la structure :

Structure_Lille (Cas général)

Nombre d'obstacles	Nombre d'obstacles détectés.
Date	Date de la détection en millisecondes.
Tableau Structure_obstacle[]	Structure obstacle (maximum 5 obstacles).

Pour chaque obstacle détecté, on transmet les paramètres suivants :

Structure_obstacle

Vitesse	en décimètres / seconde (-127 à +127 dm/s).
Largeur	en centimètres (0 à 255 cm).
Position	en centimètres (-32768 à +32767 cm).
Profondeur	en centimètres (0 à 65536 cm).

La vitesse est signée ce qui permet de connaître le sens de déplacement de l'obstacle (Cf. § 7-3.5.3.1).

Quand aucun obstacle n'est détecté, la structure envoyée au copilote se simplifie sous cette forme :

Structure_Lille (Cas sans obstacle)

Nombre d'obstacle = 0	Aucun obstacle.
Date	Date du résultat.

De cette manière, le copilote dispose de données synthétiques sur l'état de l'environnement à l'avant du véhicule à une date donnée.

8-3.4. Disponibilité et rafraîchissement des données

Le temps nécessaire pour traiter un couple d'images n'est pas fixe. Celui-ci dépend de la complexité des images et augmente avec le nombre de contours détectés qui est une fonction croissante du nombre d'obstacles présents à l'avant du véhicule.

Lors des différents essais avec un seul processeur, les temps obtenus étaient compris dans un intervalle de 50 à 100 ms selon les cas étudiés. Le rafraîchissement des données ne peut donc pas être effectué à une fréquence fixe.

Cependant pour ne pas saturer les communications avec le copilote qui reçoit les données des différents capteurs présents sur le véhicule, la période de rafraîchissement des données a été fixée à 200 ms. Par conséquent, les données sont disponibles et transmissibles au copilote avec une périodicité fixe. Comme les traitements s'effectuent très rapidement, seuls les résultats les plus récents sont communiqués au copilote, les anciens étant ignorés. Cependant les résultats non transmis au copilote permettent de fiabiliser les détections par le suivi temporel des obstacles.

8-4. Présentation au Board Member's Meeting '94 (BMM '94)

A l'issue des huit années de recherche, les résultats du programme européen Prometheus ont été présentés dans la région parisienne. Pas moins de 88 véhicules étaient présents durant cette démonstration qui s'est déroulée du 18 au 20 octobre 1994. Les représentants des ministères de la recherche, de l'industrie et des transports des pays impliqués dans Prometheus, les responsables des constructeurs automobiles, des membres de la presse ainsi que des scientifiques ont été conviés à prendre place à bord des démonstrateurs afin de découvrir les résultats de ces recherches.

8-4.1. Circuit de Morte-Fontaine

La majorité des démonstrations a été réalisée au centre d'essais routier du CERAM à Morte-Fontaine près de Senlis, dans l'Oise. La figure 8.7 représente un plan global du circuit. Les pistes disponibles ont été partagées entre les différents types de démonstrations. Celles concernant l'évitement d'obstacles (CED 3 Collision Avoidance) dont nous faisons partie avaient lieu sur le circuit routier.

8-4.2. Scénario de démonstration

L'ensemble des fonctionnalités de Prolab 2 était présenté sous la forme d'un scénario. Celui-ci mettait en scène le véhicule Prolab 2, deux véhicules cibles et un piéton. Nous ne détaillons ci-après que les deux phases du scénario qui nous intéressent :

- Phase 2 : Sur la ligne droite de la clothoïde jusqu'à 50 m avant le carrefour du virage bleu, le démonstrateur repère la position des voies. Si la voiture mord la ligne de droite, le conducteur est informé.
- Phase 3 : En approche du virage bleu, le démonstrateur détecte les piétons en situation dangereuse. Quand Prolab 2 arrive, un piéton traverse. Un message de danger apparaît, Prolab doit s'arrêter devant le piéton. Quand le passage est libre, le démonstrateur démarre jusqu'au STOP.

Le passage d'une phase à l'autre était activé par des balises actives placées le long du parcours. De cette manière le copilote pouvait déclencher les algorithmes nécessaires à la réalisation de la phase en cours.

8-4.3. Démonstration

Nous ne détaillons ici que les différentes étapes concernant la détection de piétons.

La phase 2 du scénario débute au niveau de l'intersection *I* (Cf. Figure 8.7, la flèche indique le sens du parcours). A cet instant, Prolab 2 passe devant une balise active indiquant le début de la "partie autoroute". Cette partie du scénario permet de montrer la détection de mordure de ligne. C'est au début de cette phase que le message MSG_BEGIN_LILLE est envoyé au PC. Ce message indique au système informatique gérant le stéréoscope linéaire de commencer les traitements pour la détection d'obstacles.

La phase 3 débute environ 50 m avant *le virage bleu* (Cf. Figure 8.7) grâce à une balise annonçant l'arrivée au carrefour. Le pilote est également prévenu grâce à un affichage de l'IHM, comme le montre la figure 8.8.



Figure 8.8 : Approche carrefour signalée par l'interface Homme-machine.

Au même instant, le message MSG_TO_LILLE est envoyé au PC par le copilote, afin de déclencher l'envoi des données. Lorsqu'un piéton est présent sur la chaussée, comme le montre la figure 8.9, celui-ci est détecté par notre système.

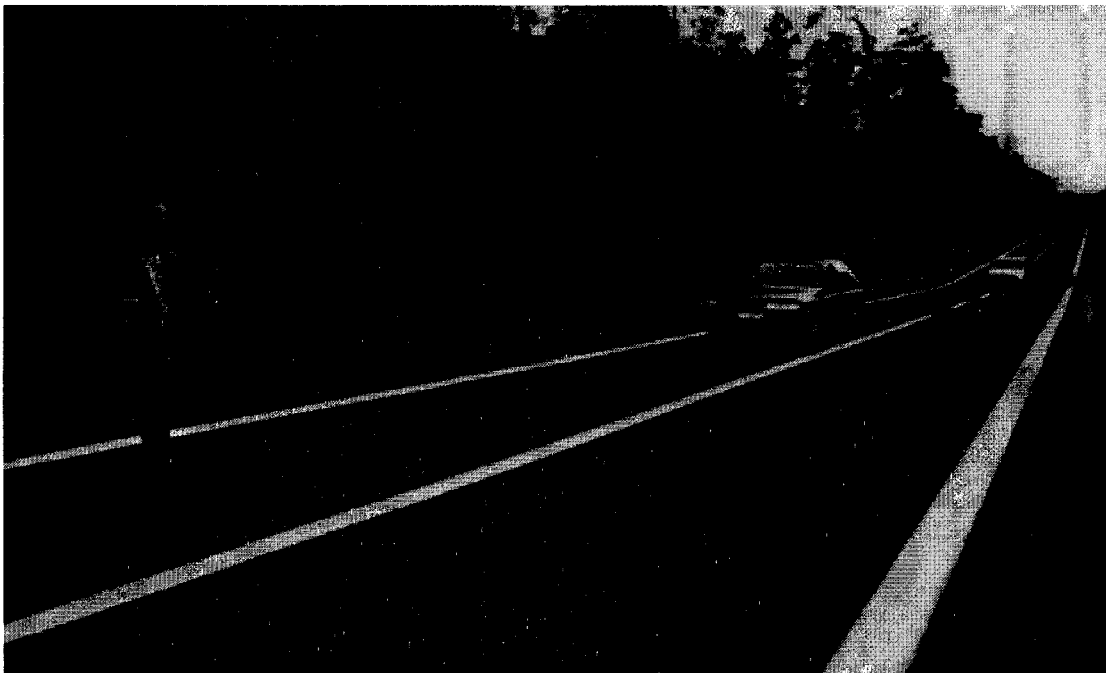


Figure 8.9 : Détection d'un piéton par Prolab 2.

Le copilote reçoit alors les caractéristiques de l'obstacle dont la date de la détection, sa position et sa distance par rapport au véhicule. Si le copilote estime que cet obstacle représente un danger pour le véhicule, il déclenche alors une alarme sur l'IHM. Ces alarmes sont à la fois visuelles et sonores. La figure 8.10 montre les deux signaux visuels pouvant être affichés par l'IHM lorsqu'un piéton est détecté. Le premier signale qu'un piéton arrive de la droite, le second prévient d'un piéton arrive de la gauche. Cette distinction est possible puisque le signe de la vitesse caractérise le sens de déplacement de l'obstacle.

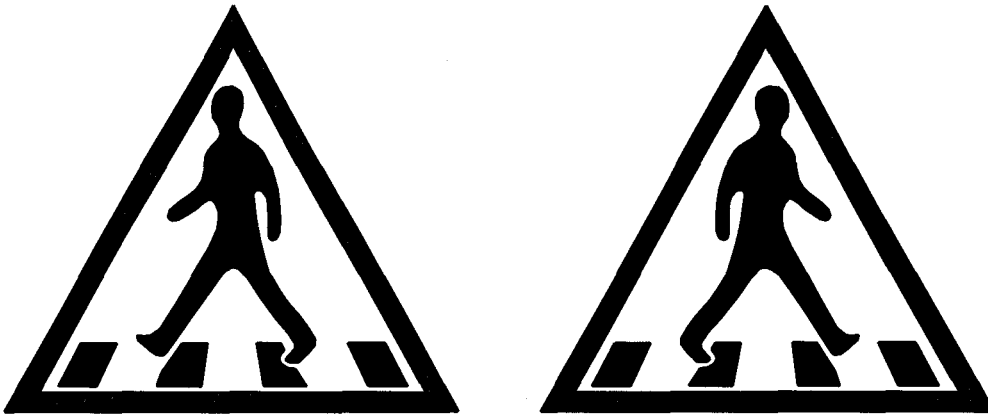


Figure 8.10 : Logos symbolisant la détection d'un piéton.

Lorsque le conducteur est averti du danger, il arrête le véhicule et laisse passer le piéton. La voiture redémarre ensuite pour s'arrêter de nouveau au STOP. Cet arrêt correspond à l'envoi du message MSG_FIN qui signale au PC l'arrêt des traitements. Celui-ci se met en attente jusqu'à la prochaine sollicitation du copilote.

Nous tenons à apporter quelques précisions supplémentaires concernant les trois messages envoyés par le copilote, c'est à dire MSG_BEGIN_LILLE, MSG_TO_LILLE et MSG_FIN. Initialement, il était prévu d'activer la détection des piétons au début de la phase 3, c'est à dire lors du passage devant la balise active informant le véhicule de l'arrivée au carrefour. Cependant, nous avons estimé que le temps qui nous était imparti était trop bref pour réaliser la détection dans de bonnes conditions. C'est pour cette raison que le message MSG_BEGIN_LILLE est envoyé au PC au début de la phase 2. Pendant cette phase, notre algorithme avait amplement le temps de repérer les lignes blanches délimitant la route et le suivi temporel de ces lignes permettait de fiabiliser leur détection. Les obstacles éventuellement détectés pendant cette phase n'étaient pas communiqués au copilote. Au début de la phase 3, le message MSG_TO_LILLE autorisait le PC à envoyer ses résultats. Comme l'algorithme avait déjà déterminé la zone de danger de façon précise et fiable grâce au suivi

des lignes blanches, la recherche d'éventuels obstacles ne se faisait que dans cette zone. L'algorithme travaillait donc à sa vitesse maximale dès le début de la phase 3, permettant ainsi d'effectuer une meilleure détection des obstacles par un suivi sur plusieurs images avant de communiquer les résultats au copilote. Nous avons constaté que l'algorithme atteignait des temps de traitements compris entre 50 et 60 ms par couple d'images linéaires lorsque la zone de danger était déterminée. L'envoi de données au copilote se faisant toutes les 200 ms, un obstacle pouvait être suivi sur 3 à 4 images avant qu'il ne soit communiqué au copilote. Ainsi la vitesse atteinte permettait de vérifier la validité des données avant de les envoyer. A la fin de la phase 3, le message MSG_FIN arrêta l'ensemble des traitements sur le PC, celui-ci se mettait alors en attente du prochain message MSG_BEGIN_LILLE.

8-4.4. Commentaires sur les 3 jours de démonstration

La présentation à BMM '94 a été globalement positive sur l'ensemble des 3 jours de présentation. Cependant, lors du premier jour, des problèmes sont apparus dans la chaîne de transmission de l'information du PC vers l'IHM. Des piétons détectés par notre système n'étaient pas toujours signalés sur l'IHM, rendant ainsi cette fonction inutile. Les deux jours suivants, ce problème fut pratiquement résolu, mais deux principales causes d'erreur subsistaient.

La première était relative à l'environnement dans lequel se déroulait le scénario. La ligne droite qui aboutissait au *virage bleu* se terminait par une côte en haut de laquelle se trouvait un passage piéton et un stop. L'intersection était en forme de T (Cf. Figure 8.7), et face à la route, se trouvaient une barrière de sécurité et un mur. Lorsque la voiture roulait sur la partie inclinée de la route, à partir d'un certain moment, le stéréoscope ne visait plus le sol, c'est-à-dire la route, mais en l'air, et scrutait la barrière et le mur. De ce fait, le temps pour détecter et suivre l'obstacle était très bref car l'image devenait rapidement trop complexe et aucune détection n'était alors possible. L'inconvénient majeur du stéréoscope linéaire est d'être inutilisable lorsque le plan de visée n'est plus dirigé vers le sol (Cf. Figure 8.11).

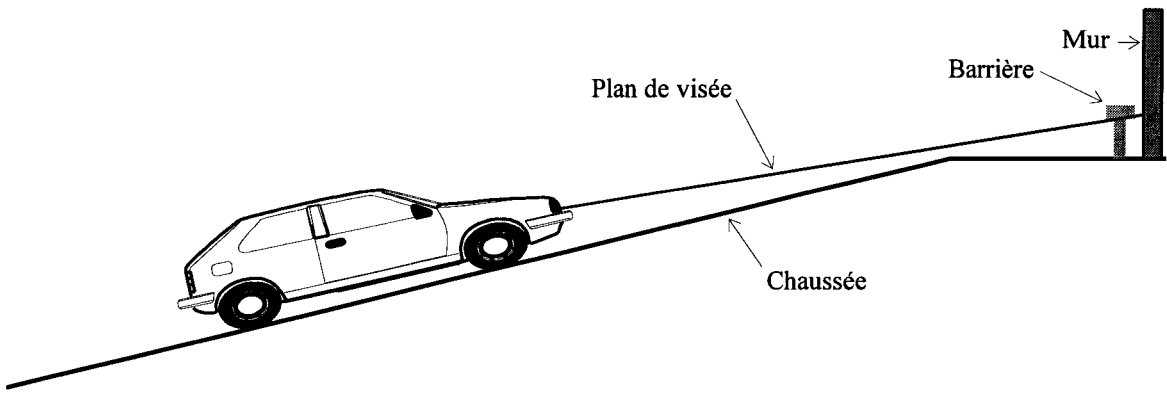


Figure 8.11 : Détection sur un plan incliné.

La figure 8.12 montre une séquence acquise lors de l'approche du STOP situé au virage bleu. On voit parfaitement les lignes blanches peintes sur la chaussée. La partie A permet de voir les effets produits lorsque l'extrémité du plan de visée passe de la zone plane à la zone inclinée. Ces effets disparaissent lorsque le véhicule se trouve complètement sur la partie inclinée (fin de la partie A). La partie C représente le passage piéton qui n'est détecté que brièvement puisque quelques images plus tard le plan de visée n'était plus dirigé vers le sol. On obtient alors les images de la partie D qui deviennent trop complexes pour pouvoir être traitées. Pour résoudre ce problème, la détection ne s'est pas faite au niveau du passage piéton qui se trouvait en haut de la côte, mais une quinzaine de mètres plus tôt, au milieu de la partie inclinée. Les traces sombres de la partie B correspondent à un piéton traversant la chaussée au milieu de la partie inclinée. On constate que dans cette partie les images obtenues sont beaucoup plus simples.

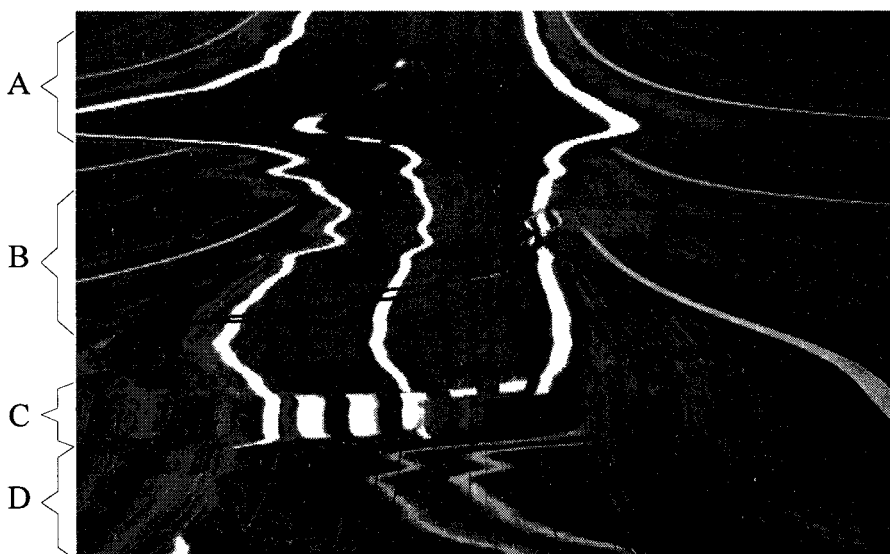


Figure 8.12 : Séquence acquise à l'approche du virage bleu.

Les deux jours suivants, la détection des piétons a été réalisée dans ce contexte, mais avec un taux de réussite se situant seulement autour de 75%. L'origine de ces 25% d'erreurs se situait au niveau des détections réellement manquées et dont les causes n'ont pas été entièrement établies. Elles pouvaient provenir d'un problème de communication, soit entre le PC et le copilote, soit entre le copilote et l'IHM, ou d'un problème d'interprétation des données de la partie haut niveau. Une autre cause possible est la vitesse de traitement insuffisante. Malgré un taux de traitement de 10 à 20 couples d'images stéréoscopiques par seconde, les déplacements relatifs entre le démonstrateur et les piétons restent encore trop importants pour effectuer un suivi correct. La solution est bien sûr d'augmenter la cadence des traitements en utilisant le second processeur de la carte de traitement. Malheureusement, ceci n'avait pu être mis en oeuvre par manque de temps.

8-5. Autres présentations de Prolab 2

Prolab 2 a fait l'objet de nombreuses présentations depuis octobre 1994 afin de faire connaître au public l'ensemble du travail réalisé par le sous-programme ProArt France dans le cadre du programme Prometheus.

8-5.1. Présentations statiques

Ces présentations furent essentiellement statiques. Parmi celles-ci nous pouvons citer :

- Le Prédit en février 95 à Paris.
- Le salon Applica à Lille en mars 95.
- Les journées CNRS à Clermont-Ferrand en mai 95.
- Il faut également signaler que plusieurs présentations ont également eu lieu à Compiègne.

Pendant ces présentations, il s'agissait essentiellement d'expliquer le travail fait par l'ensemble des laboratoires impliqués, de présenter les fonctionnalités du véhicule et également les moyens utilisés pour avertir le conducteur en cas de danger.

8-5.2. Présentation clôturant le programme ProChip

Le 9 juin 1995, le sous-groupe ProChip a présenté son démonstrateur à Labège près de Toulouse. Il s'agit d'une Safrane équipée d'un radar hyperfréquence pouvant détecter les obstacles situés entre 35 et 180 m et de deux lidars ayant les mêmes objectifs mais avec un champ d'action allant de 0 à 50 m.

Lors de cette présentation, les organisateurs ont invité les membres du groupe ProArt France à présenter leur démonstrateur Prolab 2. La presse ainsi que les responsables des constructeurs automobiles français P.S.A. Peugeot-Citroën et Renault étaient présents.

Les démonstrations ont eu lieu sur la piste d'athlétisme du stade Just-Fontaine de Labège. La piste était recouverte de cendrée et ne comportait aucune marque particulière. Trois fonctionnalités de Prolab 2 y ont été présentées :

- La détection de véhicules à l'avant par le télémètre laser.
- La détection de véhicules à l'arrière par la caméra arrière.
- La détection de piétons par le stéréoscope linéaire.

Le scénario proposé était simple puisqu'il s'agissait de suivre un véhicule cible sur la ligne droite de la piste. Après le virage, la caméra arrière entraînait en action pour détecter un véhicule cible voulant doubler Prolab 2. Le scénario se terminait avec la détection d'un piéton traversant la piste.

Les résultats obtenus lors de cette journée ont été excellents. La détection des piétons a atteint un taux de réussite approchant les 100 %. La seule petite erreur que l'on ne peut considérer comme telle a été causée par un photographe. Celui-ci voulant prendre une photographie de Prolab 2 s'est agenouillé sur la piste et a été détecté à la place du piéton "officiel". Il faut cependant remarquer que les algorithmes utilisés étaient les mêmes que ceux employés au BMM '94 et dans les mêmes conditions puisqu'un seul processeur i860 était chargé de la détection. Nous avons également constaté que l'algorithme n'était pas gêné par l'absence de ligne blanche. Malgré des temps de traitement légèrement plus longs, puisque les images étaient explorées intégralement à chaque nouvelle acquisition, la détection s'est faite sans aucun problème.

Cette démonstration nous a permis de montrer que notre système de détection d'obstacles par stéréovision linéaire était efficace et ne devait pas être remis en cause. Les problèmes rencontrés au BMM '94 étaient simplement dus à la configuration particulière des lieux qui obligeait notre système à travailler dans des conditions difficiles.

8-6. Conclusion

Dans le cadre du programme européen Prometheus, le véhicule expérimental Prolab 2 a été développé par le groupe ProArt France en coopération avec les constructeurs automobile PSA Peugeot-Citroën et Renault. L'objectif était de concevoir un système d'aide à la conduite afin d'assister le conducteur lors de la réalisation de manoeuvres et de l'informer des dangers potentiels ou réels.

Le stéréoscope linéaire a été implanté sur le véhicule expérimental Prolab 2. Le rôle de notre système de détection était de réaliser l'une des fonctions prévues sur le cahier des charges de Prolab 2 : la détection des piétons.

Les résultats obtenus au cours des différentes démonstrations ont montré que la stéréovision linéaire était une solution réaliste pour détecter les obstacles à l'avant des véhicules routiers. La détection d'obstacle s'est réduite, dans le cadre des scénarii retenus, à la détection des piétons. Il faut cependant indiquer que le principe de détection et les algorithmes utilisés sont les mêmes, quelque soit l'obstacle présent devant le véhicule, la reconnaissance du type d'obstacle se faisant simplement sur un critère de largeur.

Nous présentons ci-après quelques améliorations possibles afin d'augmenter les performances de notre système de détection.

Nous avons vu au chapitre 6 que la mise en correspondance bi-mode utilisait les lignes blanches délimitant la chaussée pour accélérer les traitements. Cependant de nombreuses routes ne possèdent pas ces lignes, l'algorithme est alors contraint d'explorer les images dans leur intégralité ce qui diminue ses performances. Une solution à ce problème serait de définir la zone de danger en fonction des limites de la chaussée quand les lignes blanches ne sont pas disponibles. Cette alternative permettrait ainsi d'atteindre les mêmes cadences de traitement. Une méthode possible serait d'étudier la densité des points de contour dans plusieurs zones des images linéaires. En effet, la chaussée constitue une région de l'image où le niveau de gris est uniforme. Peu de points de contour y sont donc détectés. Par contre, les bords de la route généralement constitués d'herbes et de buissons donnent naissance à des régions texturées présentant de nombreux points de contour. Une analyse de la densité des points de contour permettrait de déterminer les limites de la chaussée.

L'identification des obstacles se fait actuellement sur un critère de largeur. Une amélioration serait de prendre également en compte le mouvement relatif des objets par rapport à celui du démonstrateur. En effet, celui-ci peut être défini par deux composantes. La composante que nous définirons comme transversale, donnée par le mouvement perpendiculaire à celui du véhicule et la composante longitudinale donnée par le mouvement colinéaire à celui du démonstrateur. Une caractérisation des deux composantes pour chaque type d'objet permettrait ainsi d'affiner la reconnaissance du type.

CONCLUSION

L'objectif d'un grand nombre de projets de recherche est de concevoir des machines intelligentes dont le rôle est d'aider, de conseiller, voire à long terme de remplacer l'Homme. Dans de nombreux cas, les prises de décision de ces systèmes intelligents supposent une connaissance précise de leur environnement. C'est ainsi que les robots furent les premiers à être pourvus de capteurs visuels leur permettant de percevoir le monde extérieur. Dans un tout autre domaine, les constructeurs automobiles ont eu l'idée d'implanter de tels systèmes afin de rendre leurs véhicules plus sûrs. C'est ainsi qu'il y a 8 ans est né le programme européen Prometheus dont le but est d'améliorer le trafic et la sécurité dans les transports routiers. La conception de systèmes de détection d'obstacles fait partie des projets de recherche les plus ambitieux entrepris dans le cadre de ce programme. C'est dans le cadre du programme ProArt, sous-programme de Prometheus, que s'est inscrit notre travail qui consiste à développer un système de détection d'obstacles à base de capteurs visuels.

La contribution de Jean-Luc Bruyelle a permis la conception et la réalisation d'un nouveau dispositif stéréoscopique constitué de deux caméras linéaires qu'il a baptisé stéréoscope linéaire. L'utilisation de ce type de caméras à la place des caméras matricielles traditionnelles permet un gain de temps très substantiel au niveau de l'analyse des couples d'images stéréoscopiques. Notre travail fut axé sur l'analyse de ces couples d'images afin de développer les algorithmes de détection d'obstacles.

Une étude des travaux antérieurs effectués dans le domaine de la stéréovision a montré que la plupart des systèmes stéréoscopiques utilisent des caméras matricielles et que les différents traitements stéréoscopiques respectent un ordre bien défini. Une analyse des méthodes de mises en correspondance existantes a révélé qu'aucune d'entre-elles ne répondait à notre attente. Nous avons donc décidé de développer notre propre méthode d'appariement.

La première étape de l'analyse des couples d'images stéréoscopiques est l'extraction de primitives appropriées au type d'images et à la reconstruction souhaitée de la scène. Les points de contour sont apparus comme étant les primitives les mieux adaptées aux images linéaires car ils correspondent aux contours des objets présents dans les images. Les nombreux travaux réalisés sur le thème de la détection des contours nous ont permis de trouver un opérateur robuste et rapide. Afin de conserver le maximum d'information, nous avons développé notre propre technique d'extraction des points de contour détectés. Celle-ci nous assure de conserver le maximum de points de contour significatifs tout en éliminant les points sans intérêt.

La seconde phase constitue la base de l'algorithme de stéréovision. Elle consiste à apparier les primitives de l'image gauche avec celles de l'image droite. La méthode que nous avons développée est basée sur un principe d'analyse séquentielle des listes de primitives. Afin de contourner les difficultés rencontrées dans les images complexes, la méthode utilise un procédé de mise en correspondance hiérarchique et est implantée selon une technique récursive. Les résultats obtenus permettent de déterminer, grâce à la triangulation géométrique, les coordonnées dans la scène réelle des points ayant engendrés les primitives.

La détection d'obstacles nécessitant des temps de réponse très courts, la méthode de mise en correspondance a été modifiée afin de prendre en compte la configuration de la route. Celle-ci limite donc ses recherches à une zone de danger définie en fonction de la signalisation horizontale. La reconstruction tridimensionnelle permet de reconstituer les objets présents dans la scène. Ceux-ci sont ensuite suivis dans le temps afin de déterminer leur comportement face au véhicule. La procédure d'analyse de la cohérence spatio-temporelle permet ainsi d'identifier les obstacles dangereux pour le véhicule.

Le stéréoscope linéaire ainsi que les algorithmes développés ont été implantés dans le véhicule expérimental Prolab 2. Ce véhicule réalisé par le groupe ProArt France dans le cadre du programme Prometheus a bénéficié du soutien des constructeurs automobiles P.S.A. Peugeot-Citroën et Renault. Le rôle de notre système de détection dans ce véhicule est limité à la détection des piétons. Prolab 2 a été présenté aux pouvoirs publics et à la presse au cours de la démonstration clôturant le programme Prometheus ainsi qu'à de nombreuses reprises où il a remporté un grand succès. Les résultats obtenus dans des conditions réelles d'utilisation ont montré que la stéréovision linéaire était une solution réaliste pour détecter les obstacles à l'avant des véhicules routiers.

Si le système que nous avons développé a donné de bons résultats dans le cadre des expérimentations faites avec le véhicule Prolab 2, il faut tout de même préciser qu'il possède certaines limites d'utilisation. Le stéréoscope linéaire est parfaitement adapté à la détection d'obstacles sur les routes et autoroutes droites ou ne présentant que des virages de rayons de courbure relativement importants. Cependant, dans un virage relevé comme celui représenté sur la figure 1, le stéréoscope ne vise plus la route. Il peut donc détecter des objets qui ne sont pas des obstacles pour le véhicule et les considérer comme tels et déclencher de fausses

alarmes. La deuxième situation rencontrée est de ne pas voir suffisamment tôt un obstacle se trouvant pourtant très près mais se situant hors de la zone de danger.

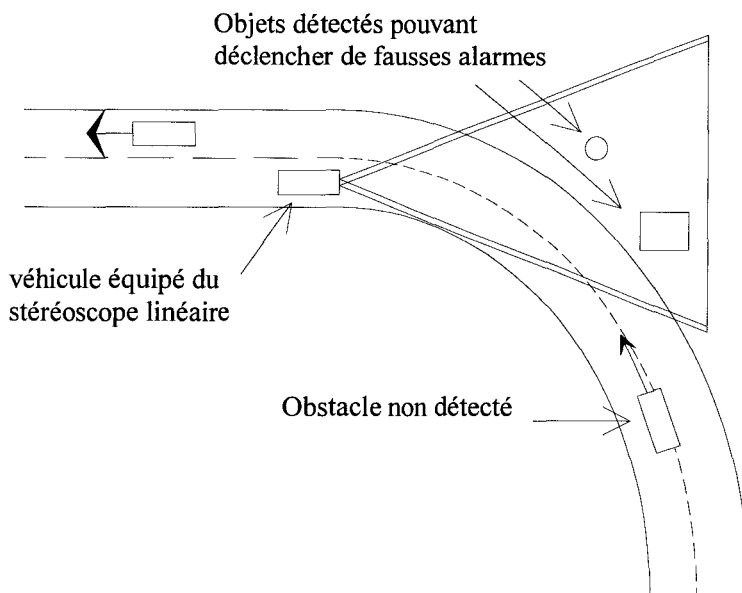


Figure 1 : Limitations du stéréoscope linéaire.

Le stéréoscope linéaire n'est évidemment pas limité à la seule détection d'obstacles sur route. Il existe une autre application potentielle du système. Celui-ci pourrait très bien être implanté sur les véhicules de transports guidés comme les trains, ou les métros. Le principal avantage avec ce type d'application est d'éliminer la quasi-totalité des problèmes rencontrés sur route. En effet, ces véhicules circulent sur des voies le plus souvent rectilignes. D'autre part les différentes courbes existantes présentent généralement des rayons de courbures importants en raison des vitesses atteintes. Les situations représentées sur la figure 1 sont, dans le cadre de cette application, extrêmement rares. Un tel projet est actuellement à l'étude et pourrait voir le jour dans un proche avenir.

La plupart des systèmes de détections d'obstacles actifs ou passifs possèdent certains inconvénients. D'une part ils ne sont pas fiables à 100 %, d'autre part ils ne couvrent pas une zone suffisante pour détecter tous les obstacles. Pour remédier à ce problème, une méthode consiste à utiliser plusieurs capteurs pouvant appartenir à des technologies différentes dans un même but. La fusion multisensorielle (ou multicapteurs) permet deux approches différentes. La première consiste à utiliser des capteurs couvrant des zones différentes afin de surveiller une zone plus importante. Les capteurs sont alors complémentaires. La seconde approche emploie des capteurs couvrant des zones identiques. Les données provenant de chaque capteur permettent alors de vérifier la validité des obstacles détectés. Ces deux approches peuvent

bien entendu être associées si les capteurs possèdent des champs d'actions couvrant une zone commune.

Jérôme Pontois, chercheur au Centre d'Automatique de Lille, va entreprendre des recherches sur la fusion multicapteurs. Il combinera les données provenant du stéréoscope linéaire avec celles d'un radar hyperfréquence développé par l'I.E.M.N. (Institut d'Electronique et Micro-Electronique du Nord) en collaboration avec le L.E.O.S.T. (Laboratoire Electronique, Ondes et Signaux pour les Transports). Le champ d'action du stéréoscope linéaire s'étend de 0 à 30 voire 50 m avec un angle de 40°. Celui du radar, par contre, varie de 20 à 150 m mais avec une ouverture limitée à 4°. La fusion pourra donc exploiter la complémentarité des capteurs tout en utilisant la zone commune pour vérifier la validité des détections.

Nous avons montré dans ce mémoire que la stéréovision linéaire était une solution originale pour analyser des scènes dynamiques tridimensionnelles. Cependant des améliorations peuvent être apportées aux algorithmes existants afin de rendre les détections plus fiables. Yassine Ruichek, chercheur au Centre d'Automatique de Lille, travaille actuellement sur l'implantation d'une procédure de mise en correspondance utilisant les réseaux neuronaux. Si les temps de traitement ne sont pas encore compatibles avec une application temps réel, les résultats obtenus sont encourageants car ils permettent de traiter des scènes plus complexes.

BIBLIOGRAPHIE

- [Adjouadi 94] M. Adjouadi et F. Candocia :
A stereo matching paradigm based on the Walsh transformation.
 IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 16,
 n° 12, p 1212-1218, Décembre 1994.
- [Asada 86] H. Asada et M. Brady :
The curvature primal sketch.
 IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8,
 n°1, p 2-14, Janvier 1986.
- [Aubert 89] D. Aubert :
Mise en correspondance d'indices d'images en résolutions multiples.
 Thèse de Doctorat, Institut National Polytechnique de Grenoble, France,
 Janvier 1989.
- [Ayache 85] N. Ayache et B. Faverjon :
***Un algorithme rapide de stéréoscopie passive utilisant la prédiction et
 la vérification récursive d'hypothèses.***
 5^{ème} congrès AFCET de Reconnaissance des Formes et Intelligence
 Artificielle, Grenoble, p 1217-1232, Novembre 1985.
- [Ayache 89] N. Ayache :
Vision stéréoscopique et perception multisensorielle.
 Ed. InterEditions, Paris 1989.
- [Baker 81] H. H. Baker et T.O. Binford :
Depth from edge and intensity based stereo.
 Proc. 7th Int. Joint Conference on Artificial Intelligence, p 631-636,
 Vancouver, Canada, Août 1981.
- [Barnard 80] S. T. Barnard et W.B. Thompson :
Disparity analysis of images.
 IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 2,
 n° 4, p 333-340, Juillet 1980.
- [Bellmann 57] R. Bellmann :
Dynamic programming.
 Princeton University Press, Princeton, New Jersey, Etats-Unis, 1957.
- [Benamrane 94] N. Benamrane :
***Contribution à la vision stéréoscopique par mise en correspondance de
 régions.***
 Thèse de Doctorat, Université de Valenciennes et du Hainaut Cambresis,
 France, 24 Janvier 1994.
- [Bensrhair 92] A. Bensrhair, P. Miché et R. Debrie :
New fast stereo matching for parallel processing.
 AMC'92, 2nd IEEE Int. WorkShop on Advanced Motion Control,
 Nagoya, Japon, 16-18 Mars 1992.

- [Bouthemy 88] P. Bouthemy :
Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images.
Technique et Sciences Informatiques, Vol 7, n° 6, p 527-546, 1988.
- [Briot 79] M. Briot, J.-C. Talou et G. Bauzil :
Le système de perception du robot mobile autonome Hilare.
Proc. 2^{ème} congrès AFCET/INRIA de Reconnaissance des Formes et Intelligence Artificielle, Toulouse, 12-14 Septembre 1979.
- [Bruyelle 94] J.-L. Bruyelle :
Conception et réalisation d'un dispositif de prise de vue stéréoscopique linéaire – Application à la détection d'obstacles à l'avant des véhicules routiers.
Thèse de Doctorat, Université des Sciences et Technologies de Lille, France, 2 décembre 1994.
- [Bruyelle 94a] J.-L. Bruyelle, J.-C. Burie et J.-G. Postaire :
A new vision system for obstacle detection.
2nd congrès Franco-Japonais de Mécatronique, Takamatsu, Japon, Vol 2, p 747-750, 1-3 Novembre 1994.
- [Burie 93a] J.-C. Burie et J.-G. Postaire :
A new edge matching procedure for obstacle detection by linear stereo vision.
IEEE Intelligent Vehicles Symposium '93, Tokyo, Japon, p 414-419, 14-16 Juillet 1993.
- [Burie 93b] J.-C. Burie, J.-L. Bruyelle et F. Cabestaing :
A special purpose architecture for onboard real time stereo vision.
26th International Symposium on Automotive Technology and Automation, Dedicated Conference on Advanced Transport Telematics (ATT) / Intelligent Vehicle - Highway Systems (IVHS) - Towards Development and Implementation, Aix-la-Chapelle, Allemagne, p 551-557, 13-17 Septembre 1993.
- [Burie 93c] J.-C. Burie, J.-L. Bruyelle et J.-G. Postaire :
Des caméras linéaires pour améliorer la sécurité des véhicules.
Colloque automatique pour les véhicules terrestres, Amiens, France, p 93-104, 22 Septembre 1993.
- [Burie 94a] J.-C. Burie et J.-G. Postaire :
A coarse to fine matching procedure for linear stereo vision.
International Federation of Automatic Control, 7th Symposium on Transportation Systems: Theory and Application of Advanced Technology, Tianjin, Chine, p 164-169, 24-26 Août 1994.
- [Burie 94b] J.-C. Burie et J.-G. Postaire :
Real time stereo vision with linear cameras: a multi-processor architecture.
IEEE Intelligent Vehicles Symposium '94, Paris, France, p 369-374, 24-26 Octobre 1994.

- [Burie 94c] [Bruyelle 94a]
- [Burie 95a] J.-C. Burie et J.-L. Bruyelle, J.-G. Postaire :
Detecting and localising obstacles in front of a moving vehicle using linear stereovision.
 Publié dans la revue "Mathematical and Computer Modelling", Special Issue of Computing technologies for Intelligent Vehicle Highway Systems, Pergamon Press, revue sous presse.
- [Burie 95b] [Rombaut 95]
- [Burie 95c] [Wan 95a]
- [Cabestaing 92] F. Cabestaing :
Détection de contours en mouvement dans une séquence d'images. Conception et réalisation d'un processeur câblé temps-réel.
 Thèse de Doctorat, Université des Sciences et Technologies de Lille, France, 31 Janvier 1992.
- [Canny 83] J.-F. Canny :
Finding edges and lines in images.
 Artificial Intelligence Laboratory, M.I.T., Cambridge, MA, USA, Tech. Rep. n° 720, Juin 1983.
- [Canny 86] J.-F. Canny :
A computational approach to edge detection.
 IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, N° 6, p 679-698, Novembre 1986.
- [Carlsson 90] S. Carlsson et J.-O. Eklundh
Object detection using model based prediction and motion parallax.
 Proc. Pro-Art Workshop on Vision, Sophia Antipolis, p 139-145, 19 et 20 avril 1990.
- [Castan 84] S. Castan et J. Shen :
A stereo algorithm taking into account the perspective distortions.
 Proc. 7th ICPR, p 441-443, Montréal, Canada, 1984.
- [Claxton 87] P.R. Claxton et E. K.-Y. Kwok :
The use colour to segment and label images.
 In. Proc. Alvey Vision Conf., p 295-302, 1987.
- [Colle 90] O. Colle :
Vision stéréoscopique à l'aide de deux caméras linéaires : Application à la robotique mobile.
 Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, France, 6 Novembre 1990.
- [Davis 77] L.S. Davis et A. Rosenfeld :
Curve segmentation by relaxation labeling.
 IEEE Transactions on Computers, Vol 26, p 1053-1057, 1977.

- [Davis 81] L.S. Davis et A. Rosenfeld :
Cooperating processes for low-level vision : A survey.
Artificial Intelligence, Vol 17, n° 1-3, p 245-263, Août 1981.
- [Deloof 94] P. Deloof, P.-A. Rolland et N. Haese :
La prévention des collisions routières par radar anticollision micro-onde.
Synthèse INRETS, n°25, Juillet 1994.
- [Deriche 87a] R. Deriche :
Using Canny's criteria to derive a recursively implemented optimal edge detector.
International Journal on Computer Vision, Vol 1, n° 2, p 167-187, Mai 1987.
- [Deriche 87b] R. Deriche :
Separable recursive filtering for efficient multi-scale edge détection.
Proc. Int. Workshop Machine Vision and Machine Intelligence, Tokyo, Japon, p 18-23, 2-5 Février 1987.
- [Deriche 90a] R. Deriche et O. Faugeras :
2-D curve matching using high curvature points : Application to stéréovision.
ISSN : CH2898-5/90/0000/0240\$01.00 © 1990 IEEE, p 240-242, 1990.
- [Deriche 90b] R. Deriche :
Fast algorithms for low-level vision.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, n° 1, p 78-87, Janvier 1990.
- [Di Zenso 86] S. Di Zenso :
A note on the gradient of multi-image.
Computer Vision, Graphics and Image Processing, n° 33, p 116-125, 1986.
- [Enkelmann 90] W. Enkelmann :
Obstacle detection by evaluation of optical flow fields.
Proceeding Pro-Art Workshop on Vision, Sophia Antipolis, p 146-155, 19 et 20 avril 1990.
- [Gallice 91] M. de Saint Blancard, J. Gallice et J. Alizon :
Procédé de détection d'obstacles présents devant un véhicule automobile et dispositif pour la mise en oeuvre d'un tel procédé.
Dépot français n° 9105628 du 7 mai 1991,
dépot européen n° 92400971.5 du 7 avril 1992
dépot USA n° 07/879281 du 7 mai 1992.
- [Gallice 94a] L. Trassoudaine, P. Checchin, J. Gallice, F. Collange et J. Alizon :
A multisensory system for road obstacle detection and tracking.
Proc. of IMACS Int. Symposium on Signal Processing Robotics and Neural Networks, p 7-10, Lille, France, avril 1994.

- [Gallice 94b] P. Martinet, J. Gallice, E. Montagne et J. Alizon :
Real time 3D location of a car from three characteristic points observed in a video image.
 International Federation of Automatic Control, Proc. 7th Symposium on Transportation Systems: Theory and Application of Advanced Technology, Tianjin, Chine, p 385-390, 24-26 Août 1994.
- [Garnousset 86] H. Garnousset :
Interprétation du mouvement 3D, une approche par vision dynamique.
 Thèse de 3^{ème} cycle en Automatique, LASS, Toulouse, 1986.
- [Gentile 94] F. Gentile :
La sécurité routière.
 Collection Que sais-je?, Presses Universitaires de France, Paris, 1994.
- [Graefe 90] V. Graefe, U.Solder et U. Regensburger :
Recognition of objects on highways.
 Proceeding Pro-Art Workshop on Vision, Sophia Antipolis, p 133-138, 19 et 20 avril 1990.
- [Hancock 90] E. R. Hancock et J. Kittler :
Discrete relaxation.
 Pattern Recognition, Vol 23, n° 7, p 711-733, 1990.
- [Horaud 89] R. Horaud et T. Skordas :
Stereo matching through feature grouping and maximal cliques.
 IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 11, p 1168-1180, Novembre 1989.
- [Horaud 93] R. Horaud et O. Monga :
Vision par ordinateur - Outils fondamentaux.
 Ed. Hermès, Paris, 1993.
- [Horn 81] B. K. P. Horn et B. Shunk :
Determining optical flow.
 Artificial Intelligence, vol 17, p 185-204, 1981.
- [Horn 86] B. K. P. Horn :
Robot Vision.
 MIT Press, Cambridge, Massachusetts, 1986.
- [Inigo 85] R. M. Inigo, T. Tkacik et E. S. McVey :
The application of linear image arrays to mobile robot guidance and navigation.
 Proc. 3rd Int. Conf. on Automated Guided Vehicle Systems, Stockholm, Suède, p 157-168, Octobre 1985.
- [Inigo 87] R. M. Inigo et T. Tkacik :
Mobile robot operation in real-time with linear image array based vision.
 Proc. IEEE Int. Symposium Intelligent Control, Philadelphie, Etats-Unis, p 228-233, Janvier 1987.



- [Ito 86] M. Ito et A. Ishii :
Three-view stereo analysis.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8,
p 524-531, Juillet 1986.
- [Joduin 94] J.-F. Jodouin :
Les réseaux de neurones : principes et définitions.
Collection Hermès, Paris, 1994.
- [Kanade 94] T. Kanade et M. Okutomi :
A stereo matching algorithm with an adaptative window : theory and experiment.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 16,
n° 9, Septembre 1994.
- [Kitchen 82] L. Kitchen et A. Rosenfeld :
Grey-level corner detection.
Pattern Recognition Letters, Vol 1, p 95-102, Décembre 1982.
- [Krotkov 90] E. Krotkov, K. Henriksen et R. Kories :
Stereo Ranging with verging cameras.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12,
n° 12, p 1200-1205, Décembre 1990.
- [Lailier 93] C. Lailier :
Guidage par vision artificielle d'un véhicule autonome sur route matérialisée par la signalisation horizontale. Application à la protection des chantiers mobiles d'entretien du réseau routier.
Thèse de Doctorat, Université des Sciences et Technologies de Lille,
France, 11 Juin 1993.
- [Lee 91] H.-C. Lee et D.R. Cok :
Detecting boudaries in a vector field.
IEEE Transactions on Signal Processing, Vol 39, n° 5, p 1181-1194,
Mai 1991.
- [Lee 94a] J. J. Lee, J. C. Shim et Y. H. Ha :
Stereo correspondance problem using the Hopfield neural network of a new energy function.
Pattern Recognition, Vol 27, n° 11, p 1513-1522, 1994.
- [Lee 94b] S.-H. Lee et J.-J. Leou :
A dynamic programming approach to line segment matching in stereo vision.
Pattern Recognition, Vol 27, n° 8, p 961-986, Août 1994.
- [Long 86] P. Long-Limozin :
Vision stéréoscopique appliquée à la robotique.
Thèse de Doctorat, Université de Nice, France, Octobre 1986.
- [Lumia 83] R. Lumia, R.M. Haralick, O. Zuniga, L. Shapiro, T.C. Pong et F.P Wang :
Texture analysis of aerial photographs.
Pattern Recognition, Vol 16, n°1, p 39-46, 1983.

- [Macaire 93] L. Macaire :
Inspection d'états de surfaces métalliques par vision linéaire. Application au contrôle qualité continu en sidérurgie.
Thèse de Doctorat, Université des Sciences et Technologies de Lille, France, 20 Janvier 1993.
- [Marr 76] D. Marr et T. Poggio :
Cooperative computation of stereo disparity.
Science, n° 194, p 283-287, 1976.
- [Marr 82] D. Marr :
Vision.
W.H. Freeman, San Francisco, 1982.
- [Medioni 85] G. Medioni et R. Nevatia :
Segment-based Stereo Matching.
Computer Vision, Graphics and Image Processing, n° 31, p 2-18, 1985.
- [Meygret 90] A. Meygret, M. Thonnat et M. Berthod :
A pyramidal stereovision algorithm based on contour chain points.
O.D. Faugeras, editor, E.C.C.V., number 427 in Lecture Notes in Computer Science, p 83-88, 1990.
- [Mohan 89] R. Mohan, G. Medioni et R. Nevatia :
Stereo error detection, correction and evaluation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 11, p 113-120, Février 1989.
- [Monga 91] O. Monga, R. Deriche et J.-M. Rocchisani :
3D edge detection using recursive filtering : Application to scanner images.
CVGIP : Image Understanding, Vol 53, n° 1, p 76-87, Janvier 1991.
- [Moravec 77] H.P. Moravec :
Toward automatic visual obstacle avoidance.
5th International Joint Conference on Artificial Intelligence, Cambridge, MA, USA, p 584, Août 1977.
- [Moravec 81] H.P. Moravec :
Rover visual obstacle avoidance.
Proc. 7th International Joint Conference on Artificial Intelligence, Vancouver, Canada, p 785-790, Août 1981.
- [Nasrabadi 92a] N. M. Nasrabadi :
A stereo vision technique using curve-segments and relaxation matching.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, n° 5, p 566-572, Mai 1992.
- [Nasrabadi 92b] N. M. Nasrabadi et C. Y. Choo :
Hopfield Network for stereo vision correspondance.
IEEE Transactions on Neural Network, Vol 3, n° 1, p 5-13, Janvier 1992.

- [Negahdaripour 85] S. Negahdaripour et B. K. P. Horn :
Determining 3D motion of planar objects from image brightness measurements.
Proc. 9th International Joint Conference on Artificial Intelligence, Vol 2, p 898-901, 1985.
- [Nevatia 76] R. Nevatia :
Depth measurement by motion stereo.
Computer Graphics and Image Processing, n° 5, p 203-214, Mai 1976.
- [Nilsson 88] N.-J. Nilsson :
Principes d'intelligence artificielle.
Collection Techniques Avancées de l'Informatique.
- [Noble 87] A. Noble :
Finding corners.
Proceeding Alvey Conference, Cambridge, Angleterre, 1987.
- [Ohta 85] Y. Ohta et T. Kanade :
Stereo by intra- and inter-scanline search using dynamic programming.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 7, n° 2, p 139-154, Mars 1985.
- [Pietikainen 86] M. Pietikainen et D. Harwood :
Depth from three-camera stereo.
Proc. IEEE Int. Conference on Computer Vision and Pattern Recognition, p 2-8, Miami Beach, Floride, Etats-Unis, 1986.
- [Pollard 85] S. B. Pollard, J. E. W. Mayhew et J. P. Frisby.
PMF : A stereo correspondance algorithm using a disparity gradient limit.
Perception, n° 14, p 449-470, 1985.
- [Pontriagin 62] L. S. Pontriagin :
The mathematical theory of optimal processes.
Interscience, New York, Etats-Unis, 1962.
- [Pontriagin 90] L. S. Pontriagin :
Optimal control and differential games : Collection of papers.
American Mathematical Society, Providence, RI, 1990.
- [ProArt 95] Laboratoires impliqués dans le groupe ProArt :
Rapport final des activités du groupe ProArt France.
En cours de rédaction.
- [Robert 86] A. Robert de Saint-Vincent :
Perception et modélisation de l'environnement d'un robot mobile : une approche par stéréovision.
Thèse de l'Université Paul Sabatier, Toulouse, 1986.

- [Rombaut 95] M. Rombaut, J.-C. Burie, J.-G. Postaire :
Prolab 2 : Un système d'assistance à la conduite.
Proc. Conf. Applica '95, Lille, France, 24 mars 1995.
- [Rosenfeld 71] A. Rosenfeld et M. Thurston :
Edge and curve detection for visual scene analysis.
IEEE Transactions on Computers, Vol C-20, n° 5, p 562-569, Mai 1971.
- [Rosenfeld 72] A. Rosenfeld, M. Thurston et Y. Lee :
Edge and curve detection : further experiments.
IEEE Transactions on Computers, Vol C-21, n° 7, p 677-691, Juillet 1972.
- [Rosenfeld 83] A. Rosenfeld et P. de la Torre :
Histogram concavity analysis as an aid in threshold selection.
IEEE Transactions on Systems, Man and Cybernetics, Vol 13, n° 3, p 231-235, 1983.
- [Ruichek 95a] Y. Ruichek et J.-G. Postaire :
Une approche neuronale pour la stéréovision linéaire.
15^{ème} colloque sur le traitement du signal et des images, GRETSI, Juan les Pins, 18-21 septembre 1995.
- [Ruichek 95b] Y. Ruichek et J.-G. Postaire :
Real-time neural vision for obstacle detection using linear cameras.
Proc. Intelligent Vehicles '95, IEEE, Detroit, Etats-Unis, 1995.
- [Selsis 95] M. Selsis :
Suivi et localisation 3D d'objets en mouvement à l'aide de modèles de contours actifs.
Thèse de Doctorat, Université des Sciences et Technologies de Lille, France, 1995.
- [Shen 89] J. Shen et J. Zhao :
A new trinocular stereo vision method.
Proc. 6^{ème} Scandinavian Conference on Image Analysis, Oulu, Finlande, Juin 1989.
- [Skordas 88] T. Skordas :
Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images.
Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, Octobre 1988.
- [Terzopoulos 87] M. Kass, A. Witkin et D. Terzopoulos :
Snakes: Active contour models.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 3, p 259-268, 1987.
- [Ullman 79] S. Ullman :
The interpretation of visual motion.
MIT Press, Cambridge, Massachusetts, 1979.

- [Ultré 94] V. Ultré, H. Deleplanque et J.-G. Postaire :
Une nouvelle technique d'étiquetage probabiliste itératif pour l'extraction des contours des images multispectrales.
AGI'94, Poitiers, Futuroscope, Juin 1994.
- [Vieren 88] C. Vieren :
Segmentation de scènes dynamiques en temps réel. Application au traitement de séquences d'images pour la surveillance de carrefours routiers.
Thèse de Doctorat, Université des Sciences et Technologies de Lille, France, 18 Avril 1988.
- [Wan 95] Y. Wan :
Un opérateur hyperbolique pour la détection des contours.
15^{ème} colloque sur le traitement du signal et des images, GRETSI, Juan les Pins, 18-21 septembre 1995.
- [Wan 95a] Y. F. Wan, J.-G. Postaire, J.-C. Burie et F. Cabestaing :
A new edge detector for obstacle detection with a linear stereo vision system.
IEEE Intelligent Vehicles Symposium '95, Detroit, Etats-Unis, 25-26 Septembre 1995.
- [Wane 93] S. Wane :
Contribution au pilotage des robots mobiles.
Thèse de Doctorat, Université de Technologie de Compiègne, 17 Mars 1993.
- [Weszka 79] J.S. Weszka et A. Rosenfeld :
Histogram modification for threshold selection.
IEEE Transactions on Systems, Man and Cybernetics, Vol 9, n° 1, p 38-52, 1979.
- [Wrobel 88] B. Wrobel-Dautcourt :
Perception de la distance par mise en correspondance de régions entre des images stéréoscopiques.
Thèse de Doctorat, Institut Polytechnique de Lorraine, Mars 1988.
- [Yachida 86] M. Yachida, Y. Kitamura et M. Kimachi :
Trinocular vision : new approach for correspondence problem.
Proc. Int. Conf. on Pattern Recognition, IEEE, p 1041-1044, Paris, Octobre 1986.
- [Zagrouba 92] E. Zagrouba, C. Krey et Z. Hamrouni :
Region matching by adjacency propagation in stereovision.
2nd Int. Conference on Automation, Robotics and Computer Vision, ICARCV '92, Vol 1, p CV8.5.1-CV8.5.5, Singapour, 16-18 Septembre 1992.

- [Ziou 91] D. Ziou :
La détection de contours dans les images à niveau de gris : mise en oeuvre et sélection de détecteurs.
Thèse de Doctorat, Institut National Polytechnique de Lorraine, France,
11 Octobre 1991.
- [Zucker 76] S.W. Zucker, R.A. Hummel et A. Rosenfeld :
An application of relaxation labeling to line and curve enhancement.
IEEE Transactions on Computer, Vol C-26, p 394-403, Avril 1976.
- [Zuniga 83] O. Zuniga et R.M. Haralick :
Corner detection using the facet model.
International Conference on Pattern Recognition and Image Processing,
p 30-37, 1983.

