

CENTRE DES ETUDES DOCTORALES IBN ZOHR

Formation doctorale :

Sciences, Techniques et Ingénierie

Faculté des Sciences - Agadir

Laboratoire des systèmes informatiques et vision (LabSIV)

THESE

Présentée par

Abdenbi MAZOUL

Pour l'obtention de grade de

DOCTEUR de l'Université Ibn Zohr

Spécialité : **Informatique – vision par ordinateur**

Contribution à la stéréovision spatio-temporelle pour l'aide à la conduite

Directeur de thèse : **EL ANSARI Mohamed**

Soutenue le 26 juin 2013

Devant la commission d'examen composée de :

M. Karim Afdel	Faculté des sciences, Agadir	Président
M. Lhoussaine Masmoudi	Faculté des sciences, Agdal, Rabat	Rapporteur
M. Brahim El Khalil Chaouki	ENSA, Agadir	Rapporteur
M. Rachid Latif	ENSA, Agadir	Examineur
M. Ahmed Machmoum	Faculté des sciences, Agadir	Examineur
M. Mohamed El Ansari	Faculté des Sciences, Agadir	Directeur de thèse

Remerciements

Arrivé au bout de cette thèse ne m'aurait pas été possible sans l'inestimable collaboration de tant de personnes qui m'ont soutenu et encouragé même dans les moments les plus difficiles. Il m'est aujourd'hui difficile de n'en oublier aucune. Je m'excuse donc par avance des oublis éventuels.

En premier lieu, je tiens à remercier chaleureusement mon directeur de thèse, Monsieur Mohamed EL ANSARI, pour la confiance qu'il m'a accordée en me proposant ce sujet, et sans qui ce travail n'aurait pas vu le jour. Je lui suis très reconnaissant pour la qualité d'encadrement, ses encouragements, ses conseils précieux, ainsi que sa grande disponibilité dont il a fait preuve à mon égard. J'ai été extrêmement sensible à ses qualités scientifiques et humaines qui m'ont facilité énormément la réalisation de ce travail.

Je remercie sincèrement Monsieur Karim AFDEL, Monsieur Brahim El Khalil CHAOUKI, et Monsieur Lhoussaine MASMOUDI pour l'honneur qu'ils m'ont fait d'avoir accepté d'être rapporteurs de cette thèse et pour l'intérêt qu'ils ont porté à mon sujet de recherche. Mes sincères remerciements vont également à Mr. Rachid LATIF et à Mr. Ahmed MACHMOUM pour avoir accepté d'évaluer ce travail en tant qu'examineurs.

Je voudrais exprimer ma profonde gratitude à tous les professeurs du Département d'informatique pour m'avoir toujours encouragé à poursuivre mes études, ainsi que leur gentillesse, leur qualité humaine et la qualité pédagogique de leur enseignement.

Une très grand merci à mes parents et à toute ma famille pour m'avoir toujours soutenu et encouragé, pour tout ce qu'ils ont fait pour moi, pour leur patience, et leur soutien continu qui sans eux, je ne serais jamais arrivé là où je suis.

Merci à tous mes amis, j'espère ne pas oublier personne, qui m'ont accompagné et aidé avec leur esprit à la fois solidaire et critique. Je mentionne spécialement Mr. Nouh IZEM et Mr. Khalid ZEBARRA avec qui j'ai passé des moments inoubliables, merci pour leur soutien infini, leur amitié et pour les discussions sur plein de sujets.

Je tiens à remercier tous mes amis pour leur amitié, leur gentillesse et leur franchise, sans les citer un par un, qu'ils trouvent ici l'expression de la plus sincère amitié.

Table des matières

Notations et Acronymes.....	6
Table des figures	7
Liste des tableaux.....	10
Introduction générale	11
 Chapitre 1 Généralités sur la stéréovision passive : Etat de l'art.....	18
1.1 La stéréoscopie : Vision binoculaire	19
1.2 Capteur stéréoscopique	21
1.2.1 Le modèle de caméra	21
1.2.2 La géométrie épipolaire	24
1.3 Calibrage et rectification.....	26
1.3.1 Les paramètres d'un système stéréoscopique	26
1.3.2 Calibrage du stéréoscope.....	27
1.3.3 Rectification d'une paire d'images	29
1.3.4 Stéréo-triangulation.....	30
1.4 La mise en correspondance stéréoscopique	33
1.4.1 Principe et difficultés	33
1.4.2 La disparité.....	35
1.4.3 Les contraintes	36
1.4.4 Les primitives à mettre en correspondance.....	43
1.4.5 Les approches de mise en correspondance stéréoscopique	45
1.4.5.1 Les mesures de similarité	45
1.4.5.2 Les méthodes locales.....	48
1.4.5.3 Les méthodes globales	50
1.4.6 Méthodes temps-réel de mise en correspondance stéréoscopiques.....	56
1.4.7 Les approches de mise en correspondance spatio-temporelles	60
1.4.8 Les techniques de raffinement	61
1.5 Conclusion	63

Chapitre 2 L'approche spatio-temporelle pour l'estimation de la disparité	65
2.1 Introduction.....	66
2.2 Algorithme proposé pour la mise en correspondance stéréoscopique.....	67
2.2.1 Les primitives utilisées : points de contours	68
2.2.2 L'association entre les points de contours des images successives	69
2.2.3 La fonction de coût	71
2.2.4 La mise en correspondance spatio-temporelle des contours	72
2.2.4.1 La mise en correspondance temporelle	73
2.2.4.2 La mise en correspondance spatiale	73
2.2.5 Carte de disparité initiale	76
2.2.6 Estimation de la carte de disparité finale	77
2.2.6.1 Gamme de disparités	77
2.2.6.2 Programmation dynamique	80
2.2.7 Récapitulatif : Pseudo-code de l'algorithme proposé	83
2.3 Conclusion	84
 Chapitre 3 Résultats expérimentaux et statistiques	 86
3.1 Statistiques et comparaisons.....	87
3.1.1 Les résultats sur les images de synthèse	87
3.1.2 Les résultats sur les images réelles.....	89
3.2 Temps de calcul	96
3.3 L'effet de bruit : L'incertitude liée à la position des primitives.....	97
3.4 Conclusion	98
 Conclusion et perspectives	 100
 Bibliographie	 103

Notations et Acronymes

L	gauche (left)
R	droit (right)
k	L'instant d'acquisition des images.
I_k^L et I_k^R	Les images stéréoscopiques gauche et droite respectivement prises à l'instant k .
$f_k = (I_k^L, I_k^R)$	La trame (<i>frame</i>) : la paire d'images stéréoscopiques obtenue à l'instant k .
d_k	La carte de disparité calculée pour l'instant k .
d_{max} et d_{min}	La valeur de disparité maximale et minimale respectivement.
$C_k^{L,j}$	Un contour extrait de l'image gauche I_k^L .
S_k^L	L'ensemble des contours de l'image gauche I_k^L .
$Ass(.)$	L'ensemble de points associés aux points d'un contour
$Match(.)$	L'ensemble de points correspondant aux points d'un contour.
IDM	Carte de disparité initiale.
$MCEPs$	Les points de contour pour le contrôle d'appariement.
STM	(<i>spatio-temporelle Matching</i>) la nouvelle méthode que nous avons proposée.
TCM	(<i>Temporal Consistent Matching</i>) la méthode proposée dans [El-Ansari_10A].
$ADAS$	Les systèmes d'aide à la conduite (<i>Advanced Driver Assistance Systems</i>)
ITS	Les systèmes de transport intelligents

Table des figures

Figure 1.1	Les champs de visions dans un système de vision stéréoscopique.....	20
Figure 1.2	Les champs de visions stéréoscopiques sur les deux images stéréoscopiques. (a) le champ de vue de gauche, (b et c) le champ de vue commun et (c) le champ de vue de droite.....	21
Figure 1.3	Le modèle de sténopé.	22
Figure 1.4	La géométrie d'un modèle de la caméra sténopé : l'image d'un point P de l'espace est l'intersection entre la droite OP et le plan image (I) en un point p.....	23
Figure 1.5	La géométrie d'un système binoculaire.	24
Figure 1.6	La géométrie épipolaire	25
Figure 1.7	Différentes position d'une mire de calibrage [stereo_smp].	28
Figure 1.8	Rectification d'un couple stéréoscopique, chaque plan image est déformé de manière à ce que les points homologues p et p' aient la même ordonnée.	30
Figure 1.9	Relation entre la disparité et la profondeur.....	31
Figure 1.10	Les ambiguïtés d'appariement des pixels dans le cas une texture répétitive.	35
Figure 1.11	Violation de la règle d'unicité : le point p de l'image gauche correspond aux points p' , p'_1 et p'_2 de l'image droite.....	38
Figure 1.12	La contrainte d'ordre : pour un point P de l'espace donné, l'ordre de ses projections sur le plan image gauche ($p_2 < p_1 < p$) reste le même sur le plan image droit ($p'_2 < p'_1 < p'$).	39
Figure 1.13	Violation de la contrainte d'ordre : il s'agit d'un point P_2 appartenant à la zone interdite de P_1 . L'ordre de ses projections par rapport à celles de P_1 est inversé d'une image à l'autre.	40
Figure 1.14	Violation de la contrainte d'ordre, cas d'un objet transparent. L'ordre des projections de P_1 et P_2 sur les plus images est inversé d'une image à l'autre.....	40
Figure 1.15	La contrainte de la disparité maximale : une gamme de disparités $[d_{min}, d_{max}]$ est créée pour définir l'espace de recherche des appariements (la zone blanche).	42
Figure 1.16	La contrainte de visibilité, le point P_1 est visible par la caméra gauche et invisible pour la caméra droite, d'où l'absence de la projection du point P_1 sur l'image droite.	43

Figure 1.17	Transformation de Census d'un pixel sur une fenêtre de 5x5, à droite la valeur de Census obtenue pour le pixel central (le pixel ayant la valeur 23 sur la fenêtre de gauche).	48
Figure 1.18	Mise en correspondance par corrélation : Une fenêtre est définie autour du pixel de l'image droite à apparier et une autre fenêtre de même taille est décalée au long de l'intervalle de recherche sur la même ligne de l'image gauche, permettant d'évaluer le meilleur score de corrélation.	49
Figure 1.19	Exemple d'utilisation de la programmation dynamique. (A gauche) quatre points de la scène projetés sur les plans d'images gauche et droite. (A droite) Matrice de la programmation dynamique avec le meilleure chemin de correspondance.	52
Figure 1.20	Une représentation 3D d'un graphe coupé, (à gauche) les cercles représentent les nœuds et les lignes continues sont les arêtes entre les nœuds, la ligne pointillé indique une coupure, (à droite) la coupure minimale qui représente la surface de profondeur... ..	55
Figure 1.21	Le système de vision MSVM-III proposé par [Jia_04].....	58
Figure 2.1	Vue d'ensemble de l'algorithme d'appariement spatio-temporelle proposé.....	68
Figure 2.2	I_{k-1}^L et I_k^L représentent les images successives d'une séquence d'images. Le point QkL dans l'image I_k^L constitue le point associé du point P_{k-1}^L dans l'image I_{k-1}^L	69
Figure 2.3	La mise en correspondance spatio-temporelle des contours appartenant à deux paires d'images stéréoscopiques successives : (a) L'image gauche de la trame f_{k-1} , (b) l'image droite de la trame f_{k-1} , (c) l'image gauche de la trame f_k , (d) l'image droite de la trame f_k . Les contours $C_{k-1}^{L,j}$ et $C_{k-1}^{R,m}$ représentent respectivement les correspondants temporels de $C_k^{L,i}$ et $C_k^{L,n}$, avec $C_{k-1}^{R,m}$ le correspondant spatial de $C_{k-1}^{L,j}$. Le contour $C_k^{L,n}$ est déduit comme étant le correspondant spatial de $C_k^{L,i}$	74
Figure 2.4	La v-disparité, (a) exemple d'image de v-disparité calculée à partir d'une carte de disparité, (b) une représentation graphique d'une v-disparité contenant quatre objets (segments verticaux) et le profil de la route (droite oblique).....	78
Figure 2.5	Les grammes de disparité dans une v-disparité : (a) L'axe vertical réfère aux ordonnées des lignes image et l'axe horizontal représente les disparités. y_{max} désigne la hauteur de l'image. d_{max} la valeur de disparité maximale pour la partie des objets. Les disparités éventuelles doivent appartenir à la zone située entre les lignes (D1) et (D3) pour la partie supérieure et entre les lignes (D2) et (D3) pour la partie inférieure. (b) l'image v-disparité extraite de la carte de disparité initiale (IDM), les deux segments oblique et vertical (en rouge) représentent respectivement le profil de la route et l'objet le plus proche.....	79

Figure 2.6	L'espace de recherche bidimensionnelle subdivisé en plusieurs sous-espaces selon le nombre des points de contrôle MCEPs. L'axe horizontal représente la droite épipolaire gauche et l'axe vertical représente la ligne épipolaire droite correspondante. Les lignes verticales et horizontales correspondent aux positions de points de contour sur la ligne de l'image, et leur intersection représente l'appariement possible.	81
Figure 2.7	L'espace de recherche des appariements par programmation dynamique des points d'une paire de lignes conjuguées. La recherches des appariements s'effectue à l'intérieur de la zone blanche définie par la gramme de disparité (d_{min} et d_{max}) . Le chemin dans cette zone représente la solution optimale pour la mise en correspondance des points de deux lignes épipolaires. Les zones en gris sont des régions interdites dont les nœuds ne sont pas pris en considération par le processus d'appariement.	82
Figure 3.1	(à gauche) L'image gauche de la trame #293 de la séquence stéréoscopique virtuelle, (au milieu) l'image de contour correspondante obtenue par le détecteur de Canny et (à droite) la carte vérité terrain fournie dans [datasite_wvandermaak_08].	87
Figure 3.2	Les cartes de disparités de la paire d'images stéréoscopiques #293 de la séquence virtuelle calculées en utilisant la méthode (à gauche) STM, (au milieu) TCM et (à droite) TCM-Canny..	88
Figure 3.3	(à gauche) L'image gauche #4183 de la séquence stéréoscopique réelle et (à droite) l'image de contour correspondante obtenue en utilisant le détecteur de Canny.	90
Figure 3.4	La carte de disparités calculée en utilisant (à gauche) STM et (au milieu) TCM et (à droite) TCM-Canny..	90
Figure 3.5	Les sous-images correspondantes aux véhicules de gauche et de droite.	91
Figure 3.6	La carte de disparités pour le véhicule de gauche et de droite en utilisant STM.	91
Figure 3.7	La carte de disparités pour le véhicule de gauche et de droite en utilisant STM.	92
Figure 3.8	(A gauche) l'image gauche de la trame #1983 de la séquence stéréoscopique réelle #2 et (à droite) l'image de contour correspondante obtenue en utilisant le détecteur de Canny.	94
Figure 3.9	Les cartes de disparités obtenues en utilisant (à gauche) STM et (à droite) TCM.	94
Figure 3.10	L'image gauche de la trame #1983 avec des boîtes englobantes autour de (à gauche) la voiture de gauche (LC), et (à droite) la voiture de droite (RC).	95
Figure 3.11	Les cartes de disparités obtenues par STM pour les deux voitures les plus proches.	95
Figure 3.12	Les cartes de disparités obtenues par STM pour les deux voitures les plus proches.	95

Liste des tableaux

Tableau 1.1 Les méthodes temps-réel de mise en correspondance.....	59
Tableau 3.1 Résultats d'appariement obtenus avec STM et TCM.....	88
Tableau 3.2 Comparaison entre les résultats obtenus en utilisant SMT et TCM avec ceux obtenue par TCM-Canny.	89
Tableau 3.3 Résumé des résultats d'appariement obtenus avec STM et TCM au niveau de la voiture de droite.	92
Tableau 3.4 Résumé des résultats d'appariement obtenus avec STM et TCM pour la voiture de gauche.	93
Tableau 3.5 Résumé des résultats d'appariement obtenus par STM et TCM lorsqu'elles sont appliquées sur la voiture de gauche (la voiture la plus proche) de la séquence réelle #2.	93
Tableau 3.6 Résumé des résultats d'appariement obtenus en utilisant STM et TCM lorsqu'elles sont appliquées sur la voiture de droite (la voiture au milieu sur la voie gauche) de la séquence réelle #2.	93
Tableau 3.7 Les résultats de comparaison des trois méthodes TCM, TCM-Canny et STM appliquées sur les voitures de gauche (LC.) et de droite (RC.) des séquences réelles #1 et #2.....	96
Tableau 3.8 Le temps d'exécution pour chaque étape de l'algorithme STM en (ms).....	97
Tableau 3.9 Les résultats obtenus sur les paires d'images #293, #1983 et #4183, lorsqu'un bruit gaussien est ajouté aux séquences d'images. Les deux dernières lignes représentent les statistiques calculées au niveau du véhicule de gauche (LC) de la trame #1983 et du véhicule de droite (RC) de la trame #4183, respectivement.	98

Introduction générale

L'augmentation du nombre de voitures a entraîné une hausse du nombre d'accidents. Le manque d'expérience, l'inattention et la défaillance du jugement du conducteur ont été identifiés comme des facteurs contribuant aux accidents de la route [Ledesma_10]. Afin de diminuer le nombre d'accidents, nous avons besoin de développer un système efficace d'aide à la conduite capable d'avertir le conducteur et de réagir à temps pour éviter toute collision possible. L'objectif d'un tel système n'est pas nécessairement le pilotage automatique du véhicule, mais seulement une aide au conducteur. Dans ce cas, le système permet de surmonter les défauts du pilote comme le manque de vigilance, et de mauvaise appréciation des distances, etc, en l'avertissant quand il y a un danger potentiel. La mise en œuvre de ce système permettra de réduire le nombre d'accidents, améliorer la sécurité routière, et donc sauver des vies humaines. Pour atteindre cet objectif, différents types de capteurs peuvent être utilisés. Par exemple, les radars et les lidars émettent respectivement des ondes électromagnétiques et des faisceaux laser afin de déterminer la présence d'objets en analysant le temps de retour des ondes réfléchies. Tandis que, les sonars utilisent des ondes sonores pour mesurer la distance selon le même principe que les radars et les lidars. Ces capteurs sont classés comme actifs, parce qu'ils émettent un signal, puis analysent les signaux réfléchis par des objets de la scène.

La caméra est un autre type de capteurs, dits passifs, qui utilisent la lumière réfléchiée par des objets de la scène. Ces systèmes de vision ont pour objectif d'imiter le système visuel chez des êtres humains. La caméra permet de capturer des informations photométriques de la scène observée à partir d'un seul point de vue. Dans ce cas, le système doit être en mesure d'analyser le contenu de l'image afin de reconnaître, à un certain niveau de précision, des

objets présents dans la scène 3D. La plupart des systèmes basés sur des caméras pour la perception de l'environnement utilisent une technique similaire à celle utilisée par les êtres humains, à savoir un système de vision à deux caméras. L'analyse de la paire d'images gauche et droite (dites stéréoscopiques) prises simultanément par les deux capteurs permet, sous certaines conditions, de reconstruire l'information 3D de la scène observée.

Les systèmes avancés d'aide à la conduite

Les systèmes d'aide à la conduite ADAS (*Advanced Driver Assistance Systems*) sont classés parmi les systèmes de contrôle en temps réel dans le domaine du transport intelligent (ITS). Ces systèmes analysent une grande quantité de données fournies par divers capteurs embarqués à bord du véhicule. Ces données doivent être mises à jour afin que le système soit capable de contrôler l'état de l'environnement où il se trouve. Malgré les progrès réalisés dans ce sens, la conception d'un système ADAS est complexe. Cela est dû aux différentes contraintes (routières et de temps réel). Depuis quelques années, divers systèmes d'aide à la conduite (ADAS) ont été développés et mis en œuvre de plus en plus pour améliorer la sécurité des conducteurs sur la route. ADAS permet la surveillance de l'environnement ainsi que la détection des autres usagers de la route en informant le conducteur de la situation où il se trouve, ou agit de manière autonome, si nécessaire. Il s'agit d'un système de sécurité active, puisque l'action s'effectue en amont de l'accident. Les systèmes ADAS incluent diverses fonctionnalités telles que :

- **ACC (*Adaptive Cruise Control*)**: Des régulateurs de distance de sécurité permettant en permanence le contrôle de la distance de sécurité. Dans ACC, le conducteur sélectionne la vitesse de la voiture et sa distance par rapport au véhicule avant. L'ACC diminue automatiquement la vitesse du véhicule quand il détecte que le véhicule devant ralentit, et de même, la vitesse reprend le niveau sélectionné lorsque le trafic devient clair.
- **EBA (*Emergency Brake Assist*)** : C'est un système d'aide au freinage d'urgence qui réagit lorsque le conducteur ne se rend pas compte du danger.
- **BSD (*Blind Spot Detection*)** : Système de détection d'angles morts permettant d'avertir le conducteur quand il y a des véhicules dans l'angle mort du rétroviseur.
- **CW (*Collision Warning*)** : L'avertisseur de collision avertit le conducteur dans le cas d'une collision imminente.

- **LDW (*Lane Departure Warning*)** : Le système d'avertissement de sortie de voie conçu pour avertir le conducteur si le véhicule traverse les marquages de voie sans que le clignotant correspondant ait été activé. Ce système utilise la caméra pour détecter les voies en identifiant les marquages de voie.

Les systèmes de vision

Parallèlement aux systèmes de sécurité active, les systèmes de vision sont de plus en plus courants dans les véhicules routiers en raison des progrès de la technologie. Divers ADAS basés sur la vision sont déjà disponibles dans le commerce, mais ils présentent certaines limites concernant la robustesse et l'adaptabilité au changement d'éclairage / conditions visuelles dues par exemple aux intempéries. L'adaptabilité peut être considérée comme la capacité à réagir et à s'adapter aux différentes conditions visuelles en temps réel. Lorsque le véhicule entre dans un tunnel, par exemple, un changement important dans le profil d'illumination se produit, ce qui rend la perception de l'environnement assez difficile, même pour les yeux humains. D'autres situations problématiques visuelles possibles se produisent en raison de la pluie, etc. Malgré tous ces enjeux et défis, les systèmes de vision sont de plus en plus populaires dans le domaine des systèmes d'assistance à la conduite (ADAS).

Les systèmes basés sur la vision utilisent un ou plusieurs capteurs d'images pour observer la scène, puis différents algorithmes de traitement d'images sont utilisés pour analyser les images capturées afin d'en extraire les informations pertinentes. Un système simple de vision peut détecter la présence des objets dans la scène en identifiant, par exemple, leurs attributs visuels, telles que la forme, la taille, la couleur, etc. Ils peuvent également établir non seulement la présence d'un objet, mais aussi l'identifier (ou classer) selon les besoins d'une application. Dans les systèmes de transport intelligents (ITS), la technologie de vision est largement appliquée dans le but d'améliorer l'efficacité des réseaux routiers.

Parmi les avantages des systèmes de vision par rapport aux autres capteurs: Premièrement, ils sont peu coûteux et faciles à installer sur un véhicule ou dans un élément de l'infrastructure routière. Deuxièmement, ils peuvent capturer une énorme quantité d'informations visuelles sur un champ de vue très large. Grâce à des innovations continues sur les algorithmes de traitement d'images, ces données visuelles peuvent être exploitées pour avoir une bonne perception de l'environnement routier, et donc identifier ses objets.

La perception de la profondeur relative aux objets d'une scène est une étape clé pour la plupart des systèmes basés sur des caméras, et qui ont pour but la perception 3D de l'environnement. Parmi les nombreuses possibilités, la perception de la profondeur à partir de deux images prises de deux points de vue légèrement décalés est la méthode la plus utilisée. Ces images sont ensuite combinées afin de reconstituer l'information 3D (c'est le principe de la stéréoscopie).

Stéréovision

La stéréovision (ou la vision stéréoscopique) est la partie de la vision par ordinateur inspirée par la nature. Les deux yeux chez la plupart des êtres vivants permettent la perception et la navigation dans le monde tridimensionnel. Grâce aux progressions technologiques, et aux améliorations au niveau des caméras numériques, la stéréovision devient un domaine qui attire de plus en plus l'attention des chercheurs en vision artificielle. Cependant, le développement d'un algorithme efficace permettant la résolution du problème de l'extraction d'information 3D à partir de plusieurs vues reste toujours un défi spécialement dans les systèmes basés sur la vision stéréoscopique. La tâche principale de la stéréovision est de reconstituer l'information 3D du monde réel à partir des deux images (droite et gauche) acquises simultanément par les deux caméras qui observent la même scène. Ces systèmes ont donc pour but de reconstruire la troisième dimension qui a été perdue au cours de l'acquisition des images. Les projections d'un même point de la scène sur les deux plans images n'ont pas les mêmes coordonnées. Ainsi, la position exacte d'un tel point de la scène peut être récupérée si et seulement si les coordonnées de ses deux projections (appelées homologues) sont identifiées. Le problème de l'identification des pixels homologues dans les deux images est appelé le problème de mise en correspondance (ou appariement) stéréoscopique. La contrainte du temps et la précision des résultats sont souvent les deux critères qui caractérisent un algorithme de mise en correspondance.

Objectif de la thèse : Problématique et Contribution

Comme mentionné précédemment, la majorité des applications d'aide à la conduite (ADAS) nécessitent l'extraction en temps réel des informations 3D de l'environnement qui l'entoure, et que la stéréovision constitue la technique la plus utilisée pour l'obtention des informations de profondeur d'une scène à partir d'une paire d'images. De nombreux algorithmes ont été

proposés pour l'extraction de cette information. Cependant, la plupart de ces approches souffre encore d'un manque de précision et/ou de temps de calcul qui est souvent très élevé. L'information de la profondeur est essentielle pour la majorité des fonctionnalités d'un tel système basé sur la vision. Ainsi, la phase de la mise en correspondance constitue souvent une étape primordiale pour accomplir les autres tâches du système, à savoir la détection et la localisation des objets dans la scène, etc.

Les principales caractéristiques du développement d'une méthode de la stéréovision en temps réel sont la précision de la carte de disparités, contre le débit d'acquisitions des images. Ainsi, plus l'algorithme utilisé pour l'extraction de la profondeur est efficace et rapide, plus tout le système est performant. Il y a donc toujours une relation entre la précision des résultats obtenus et le temps de calcul, cette relation est appelée la fonction rapidité/précision (*speed-accuracy trade-off function*). Cela nous a motivé de nous concentrer sur le développement d'une approche permettant l'obtention des résultats efficaces en temps réel.

L'objectif de cette thèse est la mise en œuvre d'un algorithme capable de trouver le meilleur compromis entre la rapidité et la précision. La méthode proposée doit être robuste et rapide comme l'exige les applications routières (comme ADAS). Pour cela nous avons développé une approche qui consiste à apparier les images stéréoscopiques en intégrant la relation temporelle qui existe entre les images stéréoscopiques successives. Dans ce travail, nous nous sommes inspiré de l'approche proposée dans [El-Ansari_10A] afin de développer une méthode plus robuste répondant à cet objectif. Notre approche a donc pour but la mise en correspondance stéréoscopique en utilisant l'information temporelle. Cela signifie que la mise en correspondance d'une paire d'images courante se fait en exploitant les résultats d'appariement calculés à partir des images stéréoscopiques précédentes. Autrement dit, l'approche proposée exploite la cohérence spatio-temporelle entre les images consécutives d'une séquence stéréoscopique pour améliorer les résultats de l'appariement. Cela nous permet de calculer une carte de disparité initiale pour la trame courante qui est ensuite utilisée pour dériver ce que nous appelons *les points de contrôle de l'appariement* (MCEPs) ainsi que les gammes de disparités pour chaque ligne image. Ces points sont exploités pour guider le processus d'appariement par programmation dynamique pour la recherche d'une solution optimale. Ceci est réalisé en divisant pour chaque ligne l'espace de recherche dans la programmation dynamique en un nombre de sous-espaces en fonction du nombre des MCEPs

trouvés. L'approche proposée a été validée à la fois sur des séquences d'images stéréoscopiques virtuelles et réelles en montrant des performances satisfaisantes.

Organisation du mémoire

Cette thèse est organisée de la manière suivante :

- **Le chapitre 1** décrit les bases de la vision stéréoscopique ainsi que l'ensemble des processus qui y sont liés. Dans la première partie de ce chapitre, nous donnons une brève introduction à la géométrie épipolaire en utilisant le modèle sténopé d'une caméra. Cette géométrie est utilisée pour décrire l'ensemble des relations qui sont nécessaires pour définir complètement la transformation d'un point de l'espace 3D en un point de l'image bidimensionnelle sur les plans images des caméras. Ensuite, les différentes propriétés de la configuration stéréoscopique binoculaire sont présentées. Dans la deuxième partie, nous abordons le problème de la mise en correspondance stéréoscopique. Puis, nous expliquons les différents phénomènes qui le rendent compliqué en évoquant quelques hypothèses et contraintes fréquemment utilisées pour simplifier sa résolution. Ensuite, Nous décrivons les fonctions de similarité les plus couramment utilisées ainsi que les deux principales classes de méthodes d'appariement stéréoscopique: locales et globales. Enfin, avant de conclure le chapitre, nous avons proposé un état de l'art sur les méthodes d'appariement en temps-réel. La plupart de ces méthodes sont basées sur des matériels dédiés permettant l'accélération du processus d'appariement.
- **Le chapitre 2** présente en détail une approche robuste et efficace pour l'appariement stéréoscopique éparse en temps-réel. Nous débutons par une présentation générale de problématique en mettant le point sur l'objectif à atteindre, à savoir le développement d'une méthode efficace et rapide pour la mise en correspondance stéréoscopique. Puis, nous poursuivons par un état de l'art sur plusieurs techniques utilisant la cohérence temporelle pour obtenir des cartes de disparité plus précises à partir des séquences stéréoscopiques. Ensuite, nous expliquons en détail toutes les étapes de notre algorithme, de l'acquisition des images jusqu'à l'obtention de la carte de disparité finale.

- **Dans le chapitre 3**, nous validons notre méthode sur des séquences d'images stéréoscopiques virtuelles et réelles. Une étude comparative a été établie avec l'approche proposée dans [El-Ansari_10A] confirmant l'efficacité de la méthode proposée. Ce chapitre présente également des statistiques sur le temps de calcul des différentes étapes de l'algorithme, et d'autres concernant l'impact de l'incertitude liée à la détection des contours sur les résultats fournis par notre approche.
- Finalement, une conclusion générale récapitulant les principales contributions est présentée en faisant une synthèse générale de la présente recherche. La synthèse fait rappeler les points importants de l'approche proposée. Ensuite, nous dressons quelques perspectives possibles pour le prolongement de ce travail.

Chapitre 1

Généralités sur la stéréovision passive : Etat de l'art

Sommaire

1.1	La stéréoscopie : Vision binoculaire	19
1.2	Capteur stéréoscopique	21
1.2.1	Le modèle de caméra.....	21
1.2.2	La géométrie épipolaire.....	24
1.3	Calibrage et rectification	26
1.3.1	Les paramètres d'un système stéréoscopique.....	26
1.3.2	Calibrage du stéréoscope.....	27
1.3.3	Rectification d'une paire d'images	29
1.3.4	Stéréo-triangulation.....	30
1.4	La mise en correspondance stéréoscopique	33
1.4.1	Principe et difficultés	33
1.4.2	La disparité.....	35
1.4.3	Les contraintes.....	36
1.4.4	Les primitives à mettre en correspondance	43
1.4.5	Les approches de mise en correspondance stéréoscopique	45
1.4.5.1	Les mesures de similarité	45
1.4.5.2	Les méthodes locales.....	48
1.4.5.3	Les méthodes globales.....	50
1.4.6	Méthodes temps-réel de mise en correspondance stéréoscopiques.....	56
1.4.7	Les approches de mise en correspondance spatio-temporelles	60
1.4.8	Les techniques de raffinement.....	61
1.5	Conclusion.....	63

La vision stéréoscopique (ou la stéréovision) est un sujet très connu dans la communauté scientifique. La stéréoscopie a pour objectif la restitution de l'information 3D des objets d'un environnement. La perception de la profondeur basée sur des points de vue différents de deux champs optiques qui se chevauchent est la méthode la plus répandue et la plus fiable. Ce phénomène, connu sous le nom de la stéréoscopie, a été formellement discuté en 1838 dans un article publié par Charles Wheatstone [Wheatstone_38]. Il a indiqué que l'écart de position dans les images de deux yeux est causé par leur séparation horizontale, ce qui donne des informations de profondeurs. De la même manière, à partir d'une paire d'images 2D prises simultanément par une caméra stéréoscopique, il est possible d'extraire une grande quantité d'informations géométriques et photométriques de la scène observée.

La vision stéréoscopique est donc le processus de déduction d'information 3D à partir de deux ou plusieurs images 2D de la même scène. Les images doivent être obtenues à partir de deux points de vue différents, et l'emplacement relatif de chaque caméra capturant des images doit être connu à l'avance afin d'obtenir des mesures quantitatives. L'information 3D peut être obtenue par la recherche de la disparité (ou décalage) entre les points homologues de deux images stéréoscopiques. Ce chapitre décrit le concept de la vision stéréoscopique ainsi que l'ensemble de processus qui y sont liés.

Dans la première partie de ce chapitre, nous présentons quelques propriétés utilisées pour décrire le modèle sténopé de la caméra qui sont liées à la projection perspective. Ensuite, nous introduisons les différentes propriétés de la configuration binoculaire stéréoscopique, nécessaires pour définir complètement la transformation d'un point de la scène 3D en un point d'image 2D projeté sur les plans images des deux caméras. Dans la dernière partie, nous décrivons le problème de la mise en correspondance stéréoscopique, les hypothèses simplificatrices et les méthodes d'appariement stéréoscopique en temps réel.

1.1 La stéréoscopie : Vision binoculaire

L'utilisation des capteurs optiques est l'une des méthodes les plus importantes permettant l'obtention de la profondeur, qui représente la distance entre les objets composant la scène et les caméras. Les systèmes basés sur la vision binoculaires sont considérés comme des méthodes passives inspirées de la vision humaine, qui permet la reconstitution de

l'information 3D à partir de deux images 2D de la scène observée. La position d'un point de la scène peut être déduite à partir de la position de ses projections sur les deux images prises de deux points de vue légèrement décalés.

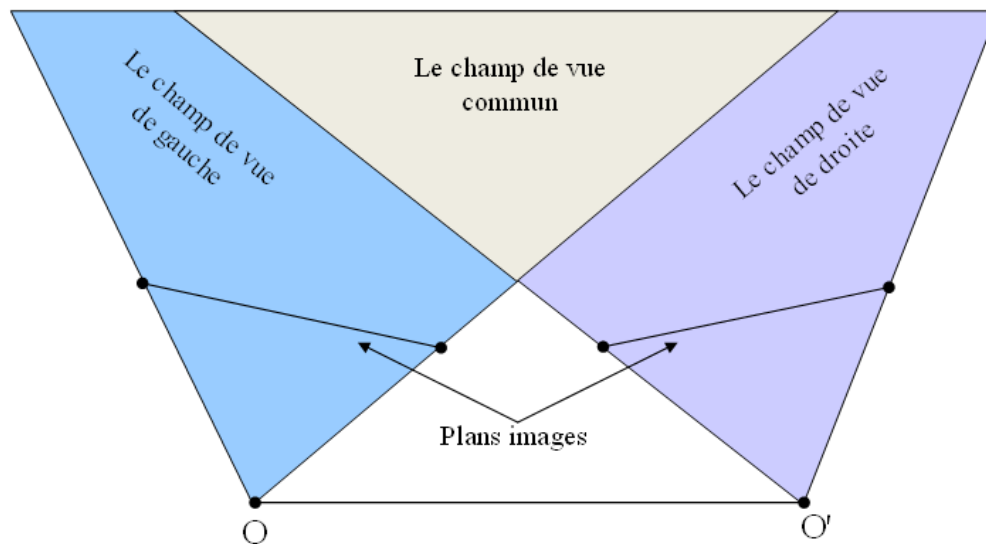


Figure 1.1 – Les champs de visions dans un système de vision stéréoscopique.

Dans cette thèse, nous nous focalisons notre attention sur la stéréovision binoculaire qui utilise deux caméras gauche et droite observant la scène depuis deux angles de vue différents, comme illustré dans la Figure 1.1. Cette figure montre une vue de dessus d'un système de stéréovision. Nous pouvons voir que chaque caméra possède son propre domaine de vision, divisant ainsi la scène en quatre champs de vue:

- **Le champ de vue commun** : contient les points de la scène qui peuvent être apparus sur les deux images gauche et droite.
- **Le champ de vue de gauche** : contient uniquement les points de la scène observée qui ne peuvent apparaître que sur l'image gauche.
- **Le champ de vue de droite** : contient uniquement les points de la scène qui ne peuvent apparaître que sur l'image droite.
- **Le champ invisible par les deux caméras.**

Etant données deux images d'une scène routière prises simultanément par deux caméras stéréoscopiques gauche et droite, comme représenté dans la Figure 1.2. L'image gauche (resp. droite) est décomposée en deux zones, l'une correspond à la projection des points qui appartiennent au champ de vision gauche (resp. droit), l'autre correspond aux points appartenant à la zone commune aux deux caméras. Un point P de la scène appartenant au champ de vision commun peut être représenté par un pixel dans l'image gauche et un autre dans l'image droite dont l'emplacement diffère d'une image à l'autre. La différence entre ces emplacements est utilisée pour estimer la distance entre le point P de la scène et les caméras. Ainsi, l'estimation de la profondeur de chaque objet de la scène 3D se fait en étendant ce principe à l'ensemble des points de la scène observés par les deux caméras.

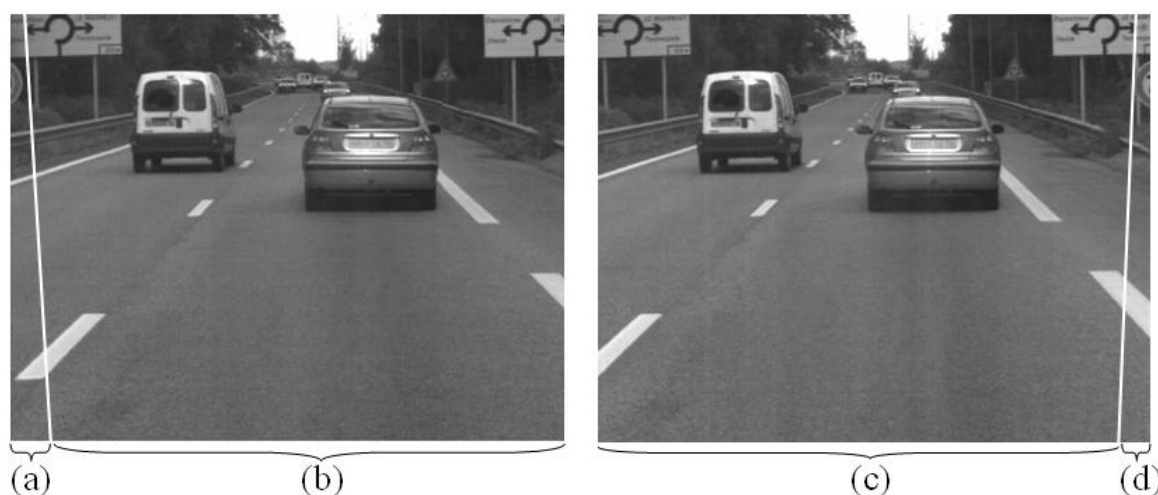


Figure 1.2 – Les champs de visions stéréoscopiques sur les deux images stéréoscopiques. (a) le champ de vue de gauche, (b et c) le champ de vue commun et (c) le champ de vue de droite.

1.2 Capteur stéréoscopique

1.2.1 Le modèle de caméra

La projection optique d'une scène tridimensionnelle sur le plan image a fait l'objet de nombreux travaux en traitement d'image. Lors de la prise de vue avec une caméra, l'environnement se trouve projeté sur un plan 2D. La majorité des caméras sont basées sur un modèle sténopé (dit aussi perspective), comme illustré sur la Figure 1.3 et Figure 1.4. Le

modèle sténopé est le plus simple et le plus utilisé pour décrire les propriétés géométriques des caméras équipées d'une lentille mince. Ces propriétés géométriques peuvent être expliquées par une projection perspective, tels que les rayons lumineux réfléchis par les objets de la scène passent à travers le centre optique, avant d'atteindre le plan image de la caméra. Afin d'éviter l'inversion de coordonnées produite par le sténopé (Figure 1.3), une configuration de projection directe est utilisée en plaçant le plan image en avant du centre optique au lieu de l'arrière. Avec cette dernière configuration, nous n'avons plus à faire inverser les images (Figure 1.4), ce qui facilite la présentation de la scène observée par la caméra.

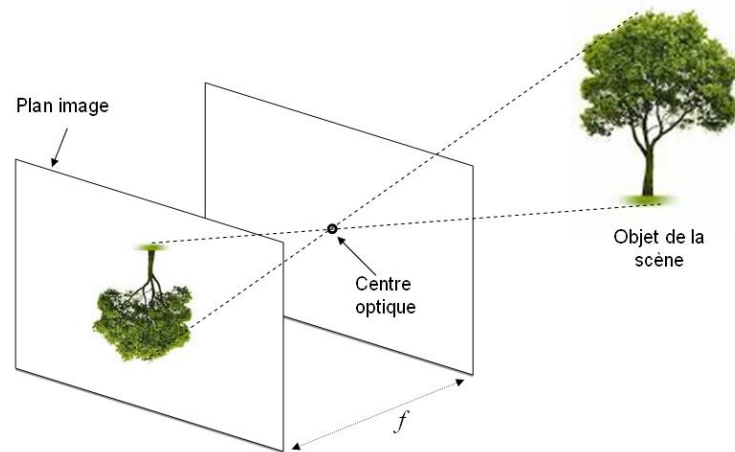


Figure 1.3 – Le modèle de sténopé.

La correspondance et la triangulation sont les bases de la reconstruction d'un modèle 3D à partir de plusieurs images 2D. Pour comprendre comment cela fonctionne, regardons d'abord le processus inverse, à savoir l'obtention des images 2D à partir d'une scène 3D. Supposons que nous avons une caméra à sténopé idéal (Figure 1.4) avec un centre de projection O (dit aussi centre optique) et un plan image (I) . P désigne le point physique de l'espace 3D et p sa projection dans le plan image (I) . La distance f entre le centre optique et le plan image est appelée *distance focale*. L'image p du point réel P de l'espace se trouve à l'intersection du segment de droite passant par P et O avec le plan image (I) . En revanche, la localisation du

point P le long de la droite Op à partir de sa projection sur le plan image et le centre optique reste indéterminée.

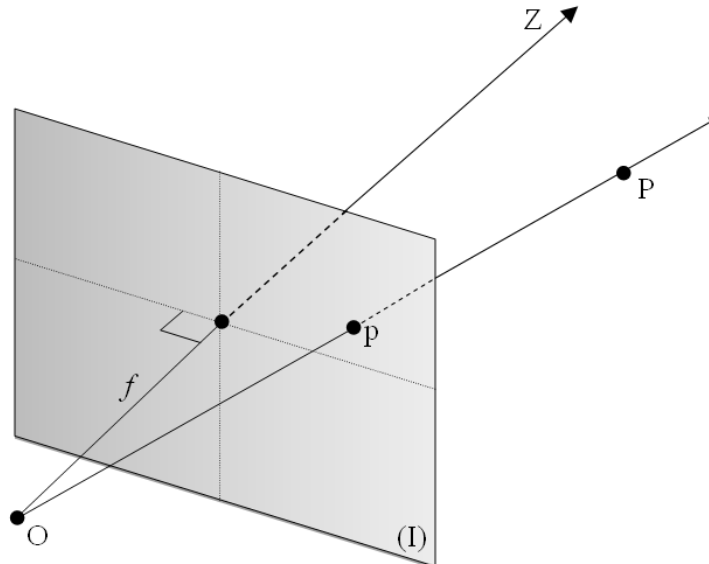


Figure 1.4 - La géométrie d'un modèle de la caméra sténopé : l'image d'un point P de l'espace est l'intersection entre la droite OP et le plan image (I) en un point p .

Cela montre que l'utilisation d'une seule image 2D ne suffit pas pour déterminer la position 3D du point P , d'où la nécessité d'ajouter une deuxième caméra avec un centre de projection O' et un plan image (II) : Soit p' l'image de P sur le plan image (II). P se trouve également sur la droite passant par O' et p' . Ainsi, nous pouvons déduire géométriquement que la position de P est complètement déterminée par l'intersection des deux droites Op et $O'p'$ (Figure 1.5). Ce processus appelé la *triangulation*, ou l'estimation de la position par l'intersection de deux droites de projection. Ce processus nous permet la localisation précise de la profondeur, et forme ainsi la base théorique de la stéréoscopie binoculaire. La phase de calibrage nous permettra ensuite de trouver la relation entre un point P de la scène exprimé dans le repère de référence et ses projections p et p' sur les plans images des captures gauche et droite.

Comme il est expliqué précédemment, les deux projections de P sont données par les intersections des lignes droites OP et $O'P$ avec les plans de projection (I) et (II). Contrairement au cas d'un système monoculaire (système à une seule caméra), l'emplacement

du point P réel peut être déduit à partir de ses projections p et p' sur les deux plans images, car P se trouve à l'intersection des lignes droites Op et $O'p'$.

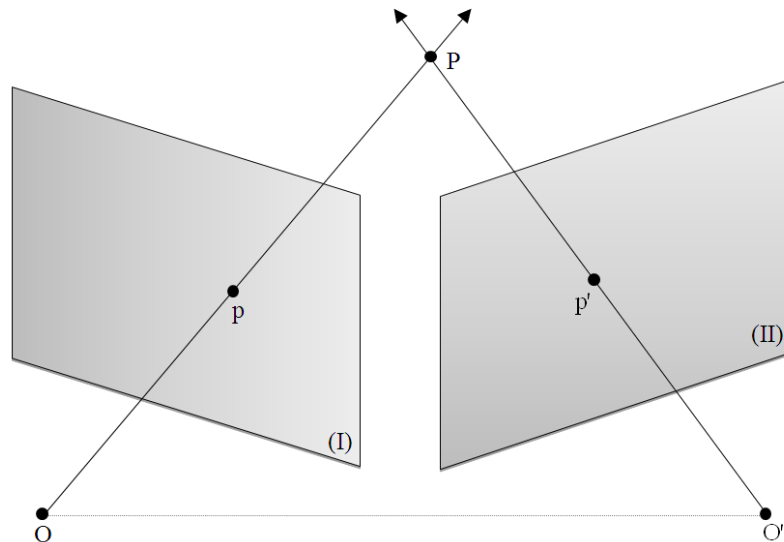


Figure 1.5 - La géométrie d'un système binoculaire.

Pour localiser donc la position du point P dans l'espace, un repère de référence doit être défini pour représenter la scène 3D. Toute la géométrie des deux capteurs ainsi que les points de l'espace seront définis dans ce repère.

1.2.2 La géométrie épipolaire

Lorsque deux caméras voient la scène 3D de deux points de vue différents, on définit une relation géométrique entre les points 3D de la scène et leurs projections sur les deux plans image 2D des caméras. Cette relation géométrique connue sous le nom de la *géométrie épipolaire*, qui décrit la relation entre les projections droite et gauche d'un point P de l'espace. La projection du point P sur les deux plans images donne naissance à deux points p et p' , dites points homologues. Combinés avec P forment un plan appelé *plan épipolaire*, qui passe également par les centres de projection O et O' des deux caméras (Figure 1.6). Dans un système qui utilise un modèle de caméra sténopé, et selon le principe d'acquisition illustré sur la Figure 1.6. Nous définissons plusieurs termes:

- La **ligne de base** est la ligne qui relie les centres optiques O et O' .
- Le **plan épipolaire** d'un point P de l'espace est un plan qui est déterminé par le point P et la ligne de base OO' .
- L'**épipôle** est le point d'intersection de la ligne de base OO' avec le plan de l'image. L'épipôle e (resp. e') correspond à la projection perspective dans l'image de gauche (resp. droite) du centre optique O' (resp. O).

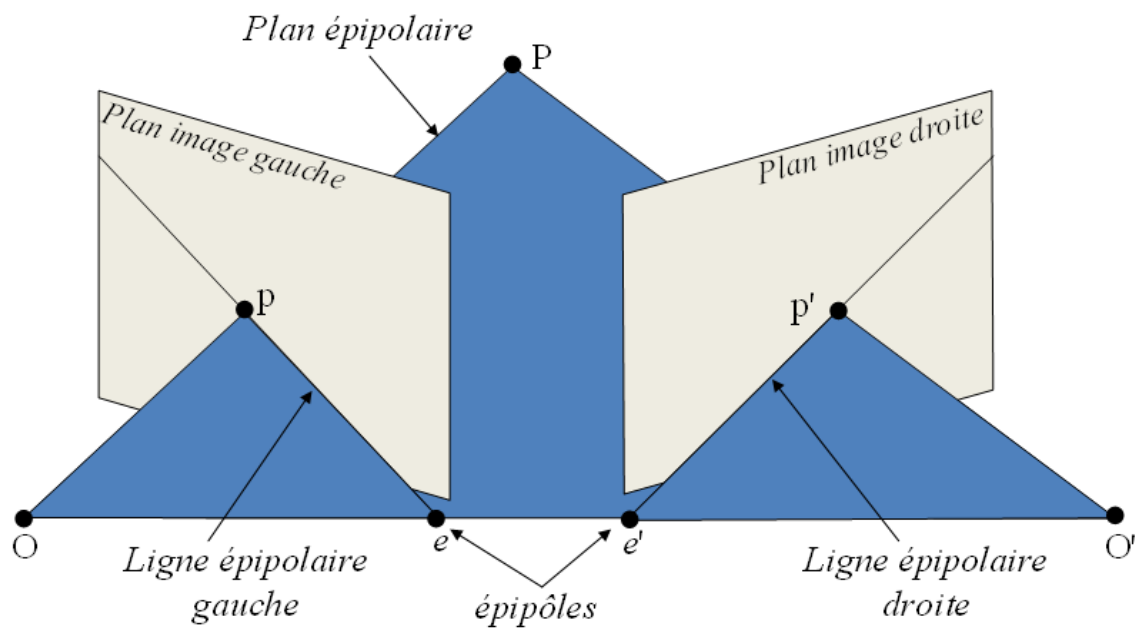


Figure 1.6 – La géométrie épipolaire

La contrainte épipolaire est l'une des plus importantes informations qui peut être exploitée dans la phase de la mise en correspondance (section 1.4). Elle permet de diminuer considérablement l'espace de recherche et par conséquent réduit le temps de mise en correspondance. Cette contrainte est bien appliquée pendant le processus de la rectification des images (section 1.3.3). Cependant, la rectification des images nécessite généralement que la procédure de calibrage soit effectuée à l'avance.

1.3 Calibrage et rectification

1.3.1 Les paramètres d'un système stéréoscopique

Souvent, on utilise $[x \ y \ 1]^T$ pour représenter la position d'un point 2D sur le plan image. $[X \ Y \ Z \ 1]^T$ est utilisé pour représenter la position d'un point de la scène 3D. En se référant au modèle sténopé, une matrice de caméra est également utilisée pour créer la relation entre les coordonnées 2D et les coordonnées 3D dans l'espace.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A[R \ T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.1)$$

Les paramètres employés dans ce modèle sont habituellement divisés en deux catégories : les paramètres intrinsèques qui sont internes à la caméra, et les paramètres extrinsèques qui peuvent varier suivant la position relative des deux caméras dans l'espace.

a) Paramètres intrinsèques :

Les paramètres intrinsèques sont les paramètres qui traduisent le passage du repère caméra au repère image. Ils se traduisent sous la forme d'une matrice appelée matrice caméra. Ce sont des paramètres fixes, inhérents à la conception de la caméra et de son optique (ils peuvent être modifiables dans le cas d'une caméra à focale réglable). Parmi les *paramètres intrinsèques* nous comptons:

$$A = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

- $\alpha_x = k_x \cdot f$ et $\alpha_y = k_y \cdot f$: les facteurs d'agrandissement de l'image, où k_x et k_y sont les facteurs d'échelles respectivement vertical et horizontal.
- f : la distance focale.
- x_0 et y_0 : les coordonnées de la projection du centre optique de la caméra sur le plan image.

b) Paramètres Extrinsèques :

Les paramètres extrinsèques sont les paramètres qui lient le repère caméra avec le repère de référence (donc le repère monde). Ces paramètres varient en fonction de la position du capteur dans l'espace. Ces paramètres sont au nombre de 12 et composent une matrice de rotation et un vecteur de translation permettant d'effectuer la transformation du repère caméra vers le repère monde :

$$[R \ T] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

- **R** : représente la matrice de rotation permettant de passer du repère lié à l'espace (repère monde) au repère lié à la caméra.
- **t_x , t_y et t_z** : sont les composantes du vecteur de translation permettant de passer du repère lié à la caméra au repère de référence.

1.3.2 Calibrage du stéréoscope

Le calibrage est un problème d'estimation des paramètres d'un modèle de capteur afin de déterminer la relation entre les objets de la scène et leurs images sur les plans images des deux caméras. Dans le contexte de la stéréovision, le calibrage est une tâche indispensable permettant de révéler la géométrie interne et les caractéristiques optiques de chaque caméra (les *paramètres intrinsèques*), ainsi que l'estimation de la position relatives entre les deux caméras (les *paramètres extrinsèques*), voir la section 1.3.1.

L'idée principale derrière le processus du calibrage des caméras est d'écrire l'ensemble d'équations liant les coordonnées d'un ensemble de points 3D et leurs projections sur les plans image gauche et droit. Dans un système calibré, il est donc possible d'estimer la position d'un point de la scène selon le repère de référence à partir des coordonnées de deux points homologues correspondants dans les repères des caméras.

Le processus de calibrage est généralement caractérisé par une matrice dite fondamentale. Cette matrice décrit le passage entre les différents repères du système stéréoscopiques. L'estimation de cette matrice se fait souvent grâce à un ensemble de points de la scène dont les positions de ses projections sont parfaitement connues dans les deux images. Différentes techniques ont été proposées pour effectuer cette étape [Papadimitriou_96, Fusiello_00], mais la plupart des méthodes utilisent des mires (Figure 1.7). Une mire est constituée d'une grille de carreaux blancs et noirs formant un damier, dont la taille et la position relative sont connues à l'avance. Le damier est placé à une certaine distance des capteurs. Il s'agit alors d'estimer les transformations géométriques inverses qui prennent en compte les distorsions géométriques permettant la rectification des lignes de l'image [Tsai_86]. Le processus de calibrage utilise plusieurs images d'une mire capturée dans différents positions et orientations (Figure 1.7).

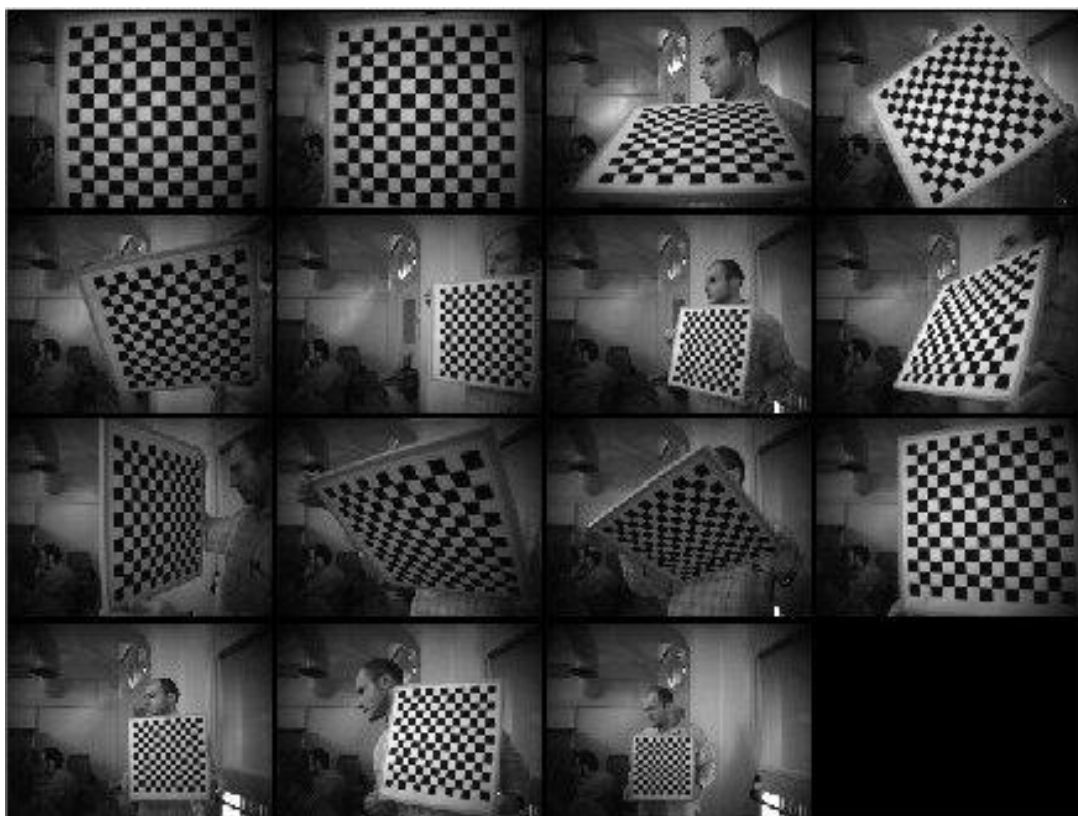


Figure 1.7 - Différentes position d'une mire de calibrage [stereo_smp].

Le calibrage des caméras a été l'objet de nombreux travaux depuis quelques dizaines d'années et continuer à être un sujet de recherche très actif dans le domaine de la vision par ordinateur.

Parmi les deux techniques les plus populaires du calibrage des caméras sont ceux de Tsai [Tsai_86] et Zhang [Zhang_00]. Tsai dans son modèle du calibrage suppose la connaissance de quelques paramètres d'une caméra afin de déduire une estimation initiale, cela nécessite la connaissance de plus de huit points de calibrage pour chaque image afin de résoudre le problème de calibrage à travers un certain nombre d'équations. Zhang exige pour faire le calibrage une mire dans au moins deux orientations différentes devant la caméra. Cette approche basée sur des points de calibrages permet donc de calculer les paramètres intrinsèques et extrinsèques du système stéréoscopique. D'autres techniques peuvent être utilisées pour faire le calibrage. Le lecteur pourra se référer à [Gruen_01] pour plus de détails sur ce sujet.

1.3.3 Rectification d'une paire d'images

Etant donnée une paire d'images stéréoscopiques, la rectification (appelée aussi configuration parallèle) est un processus permettant des déformations de chacun des plans images de manière à rendre les lignes épipolaires confondues et parallèles avec lignes images [Toulminet_06, Hautière_06], Autrement dit, un point p de l'image gauche et son homologue p' de l'image droite se trouve nécessairement sur les lignes de même ordonnées, ce qui simplifie considérablement l'étape de mise en correspondance, telle que la recherche d'un couple de points homologues s'effectue, dans cette configuration, le long des mêmes lignes horizontales des deux images. Dans ce cas, le système stéréoscopique est constitué de deux caméras projectives linéaires dont les plans image (I) et (II) sont confondus, comme représenté à la Figure 1.8, de telle manière à ce que les droite épipolaires soient parallèle et horizontales [Toulminet_06, Hautière_06].

L'utilisation de la géométrie épipolaire montre que la recherche des points homologues dans la phase de la mise en correspondance peut être réduite à une recherche 1D le long des lignes images (confondues avec les lignes épipolaires). Après rectification, deux point homologues ont donc le même ordonnée Y sur les deux plans images gauche et droite, ce qui simplifie énormément le problème de la mise en correspondance.

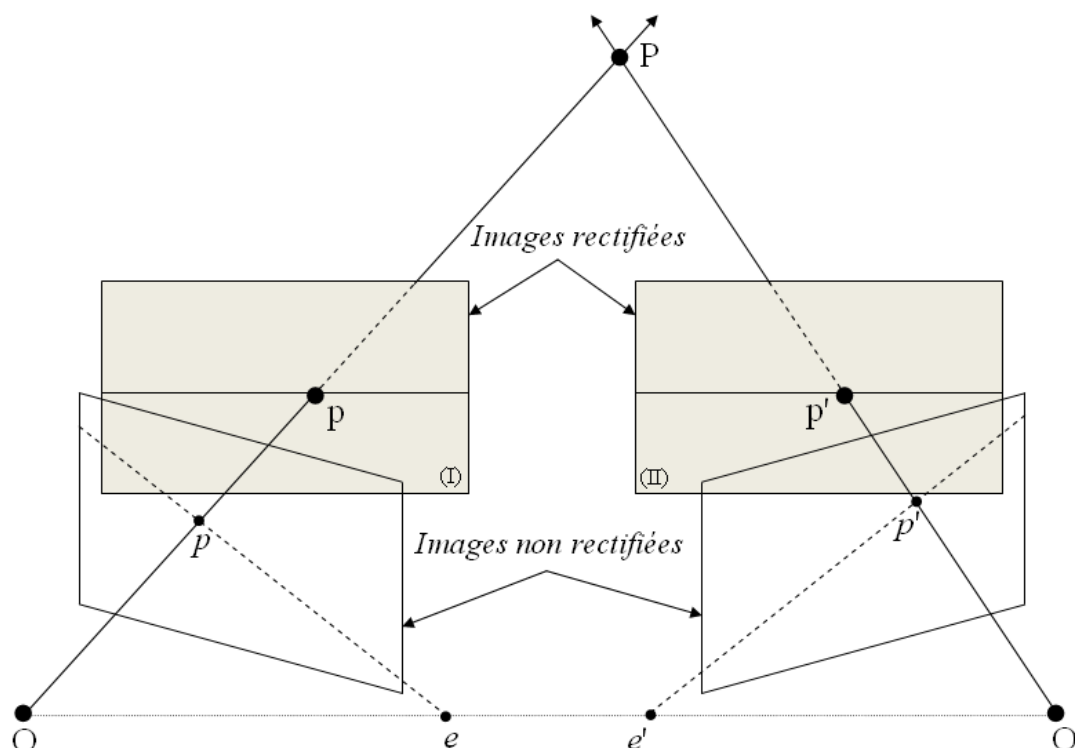


Figure 1.8 - Rectification d'un couple stéréoscopique, chaque plan image est déformé de manière à ce que les points homologues p et p' aient la même ordonnée.

Comme le montre la Figure 1.8, chaque pixel du plan image gauche est reprojété sur un autre plan image gauche correspondant. Ce nouveau pixel projeté se situe sur la ligne ayant la même ordonnée que son point homologue. Ce processus est connu sous le nom de rectification ou de rectification épipolaire dans [Faugeras_01]. Les distortions au moment de la rectification engendrent souvent des coordonnées non-entières, ce qui nécessite des méthodes d'interpolation afin d'estimer leur niveau de gris. La complexité algorithmique et la possibilité de faux résultats sont considérablement réduites dans [Zhang_96]. Diverses méthodes de rectification ont été proposées par [Fusiello_00, Hartley_99]. Tout au long de ce manuscrit, nous allons travailler avec des images rectifiées.

1.3.4 Stéréo-triangulation

La triangulation consiste à déterminer géométriquement les coordonnées tridimensionnelles d'un point de la scène à partir de ses projections sur les plans images des deux caméras

gauche et droite. Pour résoudre ce problème, il est nécessaire de connaître les fonctions reliant la position d'un point de la scène observée et les coordonnées 2D de ses projections sur les plans images et selon un repère de référence. Chaque point p d'une image correspond à une ligne dans l'espace 3D. Les points de cette ligne sont projetés en un point sur l'image. Autrement dit, un point P de l'espace se projette sur le plan image en p . Cette projection impose que les points O , p et P soient alignés. Malheureusement, cette transformation n'est pas inversible: un point de l'image ne représente pas un point de l'espace mais plutôt une droite dans l'espace.

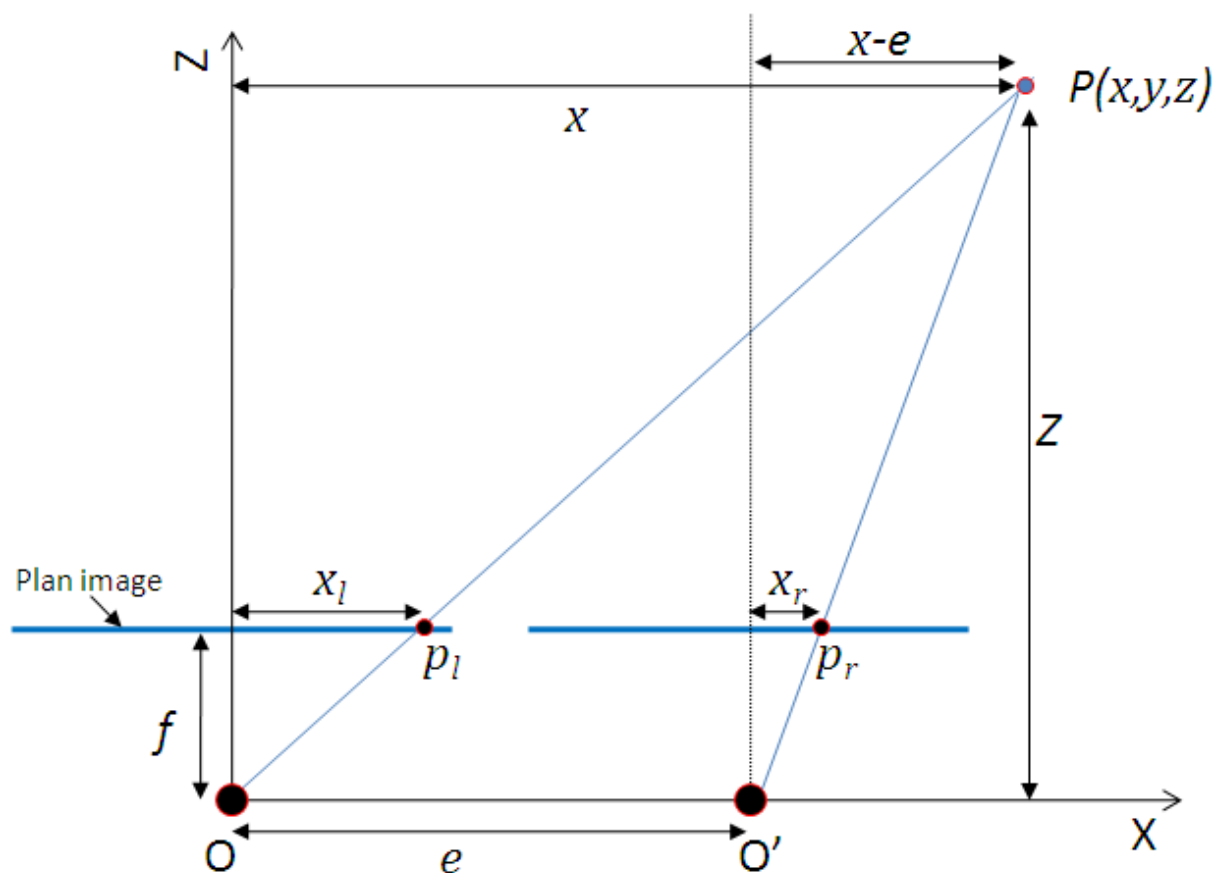


Figure 1.9 - Relation entre la disparité et la profondeur.

Il est donc impossible de retrouver la troisième dimension à partir d'une seule image. En revanche, si nous considérons un point observé dans deux images différentes, l'ambiguïté disparaît et il devient possible de trouver la position tridimensionnelle du point. Les équations des droites Op et $O'p'$ (exprimées dans le repère de référence) sont évaluées à l'aide des points p et p' et des paramètres obtenus lors de la phase du calibrage. Nous pouvons donc

calculer l'intersection des deux droites (Op et $O'p'$) pour obtenir les coordonnées du point P dans la scène 3D.

Considérons un système stéréoscopique particulièrement simple, telles que les images sont rectifiées de telle manière à ce que les lignes épipolaires soient toutes parallèles avec la ligne de base et confondues avec les lignes image. Ainsi, tous les pixels d'une ligne de la caméra gauche trouvent leurs homologues dans la ligne correspondante de la caméra droite. Dans ce cas simplifié, les abscisses x_l et x_r d'une paire de points homologues appartiennent à la même ligne épipolaire sur les deux images gauche et droite, ce qui nous permet de montrer que la profondeur est inversement proportionnelle à la valeur de la disparité entre les deux points homologues (Figure 1.9).

Dans un repère (O, X, Y, Z) considéré comme repère de référence. La Figure 1.9 représente une vue d'en haut d'un système stéréoscopique simplifié où P est un point de l'espace de coordonnées $(x, y, z)^T$. Les projections du point P dans les plans image gauche et droite sont notées respectivement p_l de coordonnées $(x_l, y_l)^T$ sur le plan image gauche, et p_r de coordonnées $(x_r, y_r)^T$ sur le plan image droit. En appliquant le théorème de Thalès dans les triangles de la Figure 1.9, nous obtenons l'égalité suivante:

$$\frac{x_l}{x} = \frac{x_r}{x - e} = \frac{f}{z} \quad (1.4)$$

Nous obtenons par un raisonnement identique suivant l'axe des ordonnées:

$$\frac{y_l}{y} = \frac{y_r}{y} = \frac{f}{z} \quad (1.5)$$

A partir de l'égalité (1.4), nous aurons:

$$\frac{x_l - x_r}{e} = \frac{f}{z} \quad (1.6)$$

Si l'on définit la disparité d comme étant la différence entre les deux abscisses x_l et x_r :

$$d = x_l - x_r \quad (1.7)$$

Nous pouvons écrire la composante z en fonction de d comme suit:

$$z = \frac{f \cdot e}{d} \quad (1.8)$$

En remplaçant dans l'équation (1.4) et (1.5), nous obtenons les expressions de l'abscisse x et de l'ordonnée y , comme suit :

$$x = \frac{x_l \cdot e}{d}, \quad y = \frac{y_l \cdot e}{d} = \frac{y_r \cdot e}{d} \quad (1.9)$$

Les équations (1.8) et (1.9) montrent que dans un système stéréoscopique dont nous connaissons bien la géométrie, il sera très facile d'en déduire les coordonnées d'un point réel à partir des abscisses de ses projections sur les plans images gauche et droite, la distance focale f et la distance e entre les caméras (ou *entraxe*). Dans un système où la distance focale f et l'entraxe e sont des constantes, les coordonnées du point P dans l'espace ne dépendront plus que des abscisses de ses projections sur le plan image de chaque caméra. Par conséquent, la reconstitution de l'information 3D de la scène à partir d'une paire d'images sera reformulée par la recherche des points homologues dans ces deux images gauche et droite, cela pour tous les points de la scène. Ce problème est connu sous le nom de la mise en correspondance (ou l'appariement) stéréoscopique, cette technique sera bien détaillée dans la section suivante.

1.4 La mise en correspondance stéréoscopique

1.4.1 Principe et difficultés

Le processus de mise en correspondance des images est une étape clé de la vision stéréoscopique, ce processus consiste à identifier entre deux images les points homologues, c'est-à-dire les points provenant d'un même point physique de l'espace. Le problème de la mise en correspondance est généralement difficile à résoudre, essentiellement à cause de son aspect combinatoire. Le développement d'un algorithme de la mise en correspondance nécessite la compréhension d'un certain nombre de problème qu'un chercheur peut rencontrer. Parmi les problèmes liés à l'appariement stéréoscopique citons les suivants :

- **Distorsions géométriques :** La projection d'un point de l'espace 3D sur un plan image 2D d'une caméra peut déformer les informations 3D représentant la scène. Les

distorsions géométriques sont les propriétés physiques et géométriques caractérisant la scène. Souvent, ces propriétés ne sont plus conservées dans le plan image. Cette déformation est d'ailleurs différente d'une image à l'autre à cause de la différence des points de vue de chaque caméra qui provoque la non-stabilité de la projection perspective d'une image à l'autre. Ainsi, les caractéristiques 3D des éléments de la scène n'apparaissent pas de la même manière dans les deux images. Par conséquent la géométrie locale d'une image est assez différente de celle de l'autre. Ce qui conduit à une première difficulté quant à l'identification de caractéristiques homologues dans les deux images.

- **Occultation** : parfois des points visibles dans une image sont absents dans l'autre image à cause du changement de point de vue. Dans le cas des zones occultées, la mise en correspondance s'avère être une tâche impossible. Certains auteurs proposent de gérer ce problème d'occultation dans leurs travaux [Hirschmüller_01, VanderMark_06, Lemonde_05].
- **Variation d'illumination** : il se peut qu'un point d'une image stéréoscopique puisse être apparié à un grand nombre de points ayant les mêmes propriétés photométriques dans l'autre image. Par ailleurs, l'appariement des deux images selon ce principe devient encore plus difficile, puisque les points homologues ne conservent pas les mêmes propriétés photométriques d'une image à l'autre. En effet, leur luminance peut être également assez différente à cause de conditions d'éclairage différentes dues des deux prises de vue, sans oublier du bruit intrinsèque à tous les problèmes de traitement d'images.
- **Texture répétitive** : lorsque les caractéristiques-image générées par ces structures sont assez proches par rapport à leur disparité. Il est probable que dans ce cas le processus de mise en correspondance confonde une caractéristique d'une structure dans une image avec des caractéristiques des parties voisines de la structure dans l'autre image (Figure 1.10).

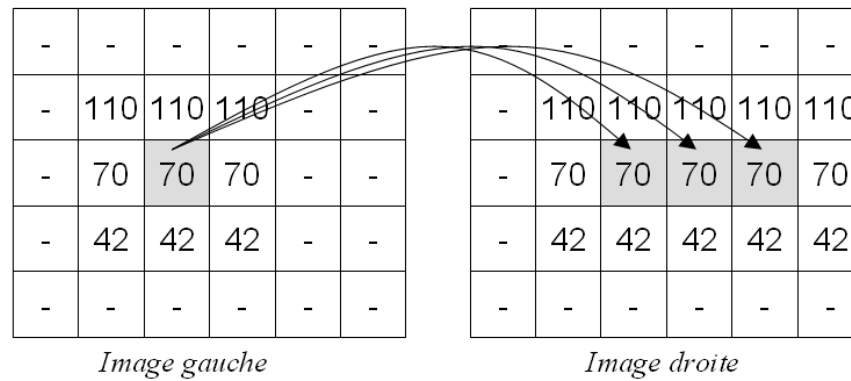


Figure 1.10 – Les ambiguïtés d'appariement des pixels dans le cas une texture répétitive.

- **Surfaces polies** : le problème qui perturbe souvent la mise en correspondance est le problème des reflets qui est provoqués par la présence dans la scène de surfaces polies, agissant comme un miroir. En réalité les deux caméras ne voient pas la surface polie, mais les images réfléchies par cette interface compte tenu de leur direction de vue. Sur les endroits occupés par ce type de surfaces, quasiment aucune correspondance n'est possible.

1.4.2 La disparité

Dans le domaine de la vision stéréoscopique le terme « *disparité* » décrit la différence entre les coordonnées des points (pixels) homologues sur les plans de la paire d'images stéréoscopique. D'après l'équation (1.8), la valeur de la disparité est inversement proportionnelle à la profondeur. Par conséquent, des grandes valeurs de la disparité correspondent aux points proches des caméras. Autrement dit, plus un point de la scène est loin du capture, plus la disparité entre ses points homologues est réduite, et vice versa.

Etant donné, un point P de l'espace dont $p(x_l, y_l)$ et $p'(x_r, y_r)$ ses projections perspectives sur les deux plans images gauche et droite, respectivement. La disparité du point P peut être définie comme étant une fonction d_l permettant l'attribution une valeur de disparité à chaque pixel de l'image gauche :

$$\begin{aligned}
d_l: \quad \mathbb{N}^2 &\rightarrow \mathbb{R}^2 \\
(x_l, y_l) &\rightarrow d_l(x_l, y_l) = (x_l - x_r, y_l - y_r)
\end{aligned} \tag{1.10}$$

Dans une configuration parallèle où les images gauche et droite sont rectifiées, les points homologues p et p' se trouvent sur les lignes de même ordonnées de l'image gauche et droite respectivement, c'est-à-dire $y_l = y_r$. La fonction (1.10) peut être simplifiée comme suit :

$$\begin{aligned}
d_l: \quad \mathbb{N}^2 &\rightarrow \mathbb{R} \\
(x_l, y_l) &\rightarrow d_l(x_l, y_l) = x_l - x_r
\end{aligned} \tag{1.11}$$

Lorsque l'image droite est considérée comme image de référence, la disparité à chaque pixel (x_r, y_r) de l'image droite s'écrit comme suit :

$$\begin{aligned}
d_r: \quad \mathbb{N}^2 &\rightarrow \mathbb{R} \\
(x_r, y_r) &\rightarrow d_r(x_r, y_r) = x_l - x_r
\end{aligned} \tag{1.12}$$

A partir des équations (1.11) et (1.12), nous pouvons déduire pour tout point (x_l, y_l) de l'image gauche, l'égalité suivante :

$$d_l(x_l, y_l) = d_r(x_l - d_l(x_l, y_l), y_l) \tag{1.13}$$

Le processus de la recherche de la disparité pour toutes les primitives extraites de l'image dans une image appelée *carte de disparité*, cette dernière est une image dont la valeur de chaque pixel correspond à la valeur de la disparité associée. La carte de disparité est utilisée afin de simplifier la visualisation des résultats de la mise en correspondance : plus le pixel est clair plus la disparité est importante, et vice versa.

1.4.3 Les contraintes

Le processus de la mise en correspondance est une procédure très difficile de la recherche. Afin de minimiser les faux appariements, un certain nombre de contraintes sont imposées. Ces contraintes sont généralement déduites des caractéristiques physique et géométrique des éléments composant la scène. Nous distinguons quatre catégories de contraintes de mise en correspondance.

- *Contraintes liées à la géométrie des captures* : l'exemple de contrainte le plus utilisé est la contrainte épipolaire.
- *Contraintes liées à la géométrie de la scène observée* : il s'agit des contraintes d'unicité, d'ordre et de gradient de disparité.
- *Contraintes physique dues aux propriétés d'illumination* : comme la contrainte d'invariance photométrique.
- *Contraintes liées aux techniques de la mise en correspondance* : les contraintes de continuité et de la disparité maximale.

Nous décrivons ci-après les contraintes les plus utilisées en vision stéréoscopique :

La contrainte épipolaire

La contrainte épipolaire, décrite à la section 1.2.2, est la contrainte la plus utile et qui est dérivée purement de la géométrie du capteur stéréoscopique. Elle ne s'applique que lorsque le système stéréoscopique est proprement calibré, c'est-à-dire quand les *paramètres intrinsèques* et *extrinsèques* sont parfaitement connus à l'avance. Cette contrainte permet d'apporter une réduction considérable au problème de la mise en correspondance. Elle permet la réduction du problème de l'appariement d'une recherche 2D à une recherche 1D. Sous cette contrainte, tout point d'une ligne épipolaire de la première image correspond à un point se trouvant sur la ligne épipolaire correspondante de la seconde image. Ainsi, dans le cas des images rectifiées, les points correspondants se trouvent sur les mêmes lignes image d'une paire d'images.

Contraint d'unicité

Elle est représentée par la règle : chaque caractéristique d'une image peut être appariée au plus à une caractéristique dans l'autre image [Marr_77]. Cette contrainte reflète la contrainte physique qu'un point d'une surface a une position unique dans l'espace à un moment donné. Sur les images, elle établit que chaque point sur le plan image provenant d'un seul point physique de l'espace ne peut correspondre qu'à un seul point dans l'autre image. Les quelques exceptions rares de cette règle se produisent lorsque deux points de l'espace se trouvent le long de la même ligne de vue par la première caméra, alors qu'ils sont séparés par la deuxième caméra (voir Figure 1.11). Cette règle exclue donc la

possibilité d'un alignement accidentel et évidemment elle n'est plus valable lorsque l'on traite des objets transparents.

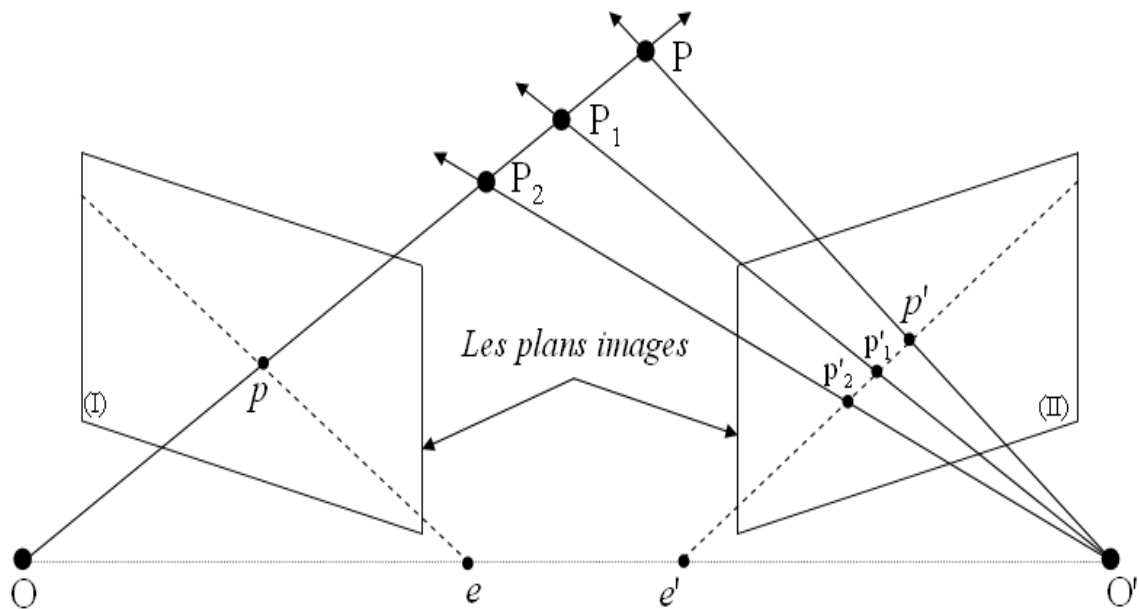


Figure 1.11 – Violation de la règle d'unicité : le point p de l'image gauche correspond aux points p' , p'_1 et p'_2 de l'image droite.

La contrainte d'ordre

La continuité du monde visuel impose une contrainte d'ordre le long des lignes épipolaires correspondantes: soit P_1 et P_2 deux points de la scène dont les projections sont (p_1 et p_2) sur l'image gauche et (p'_1 et p'_2) sur l'image droite, respectivement. Si p_1 se situe à gauche de p_2 dans l'image de gauche, alors cette même relation mutuelle existe généralement aussi entre leurs homologues (p'_1 et p'_2) dans l'image droite. La Figure 1.12 illustre la contrainte d'ordre.

Soit P_1 un point de l'espace observé par les deux caméras. En supposant que la scène est composée d'objets opaques, on peut associer au point P_1 une zone interdite, définie par le cône passant par P_1 et par la ligne de base OO' . La Figure 1.13 illustre un point P_2 appartenant à la zone interdite associée au point P_1 . Les projections de P_2 sur les deux

images viole la contrainte d'ordre par rapport aux deux projections de P_1 . Nous constatons que l'ordre de leurs projections est inversé. Cela arrive souvent lorsque les distances entre les objets, souvent de faibles dimensions, et le capteur stéréoscopique sont très différentes.

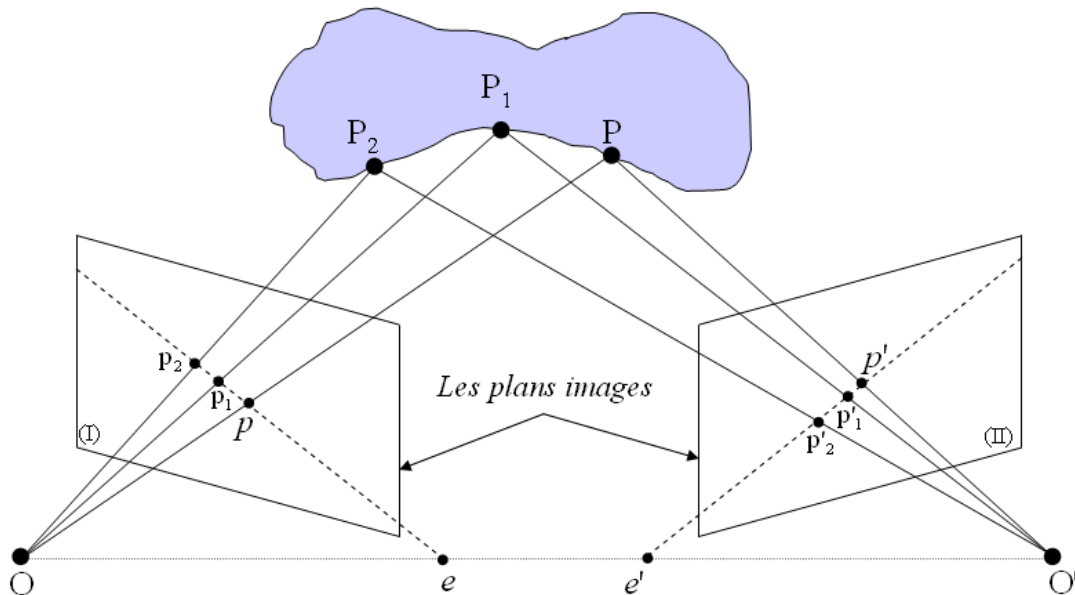


Figure 1.12 – La contrainte d'ordre : pour un point P de l'espace donné, l'ordre de ses projections sur le plan image gauche ($p_2 < p_1 < p$) reste le même sur le plan image doit ($p'_2 < p'_1 < p'$).

Supposons maintenant que P_1 constitue un point d'un objet transparent, et P_2 un point de même objet se trouvant à l'intérieur de la zone interdite associée à P_1 . Les projections de P_2 dans les deux images violent la contrainte d'ordre par rapport aux deux projections de P_1 , (Figure 1.14). Ainsi, la contrainte d'ordre ne peut pas être respectée dans le cas de surfaces transparentes.

Les algorithmes de mise en correspondance stéréoscopiques posent comme hypothèse que toutes (ou la majorité) les surfaces de la scène à traiter sont opaques. Sous cette hypothèse, la contrainte d'ordre n'est violée que très localement. Dans la pratique, la contrainte d'ordre est utilisée pour éliminer des paires d'appariements potentiels dont la position sur les lignes épipolaires correspondantes viole l'ordre par rapport à un appariement déjà validé se situant sur ces lignes épipolaires.

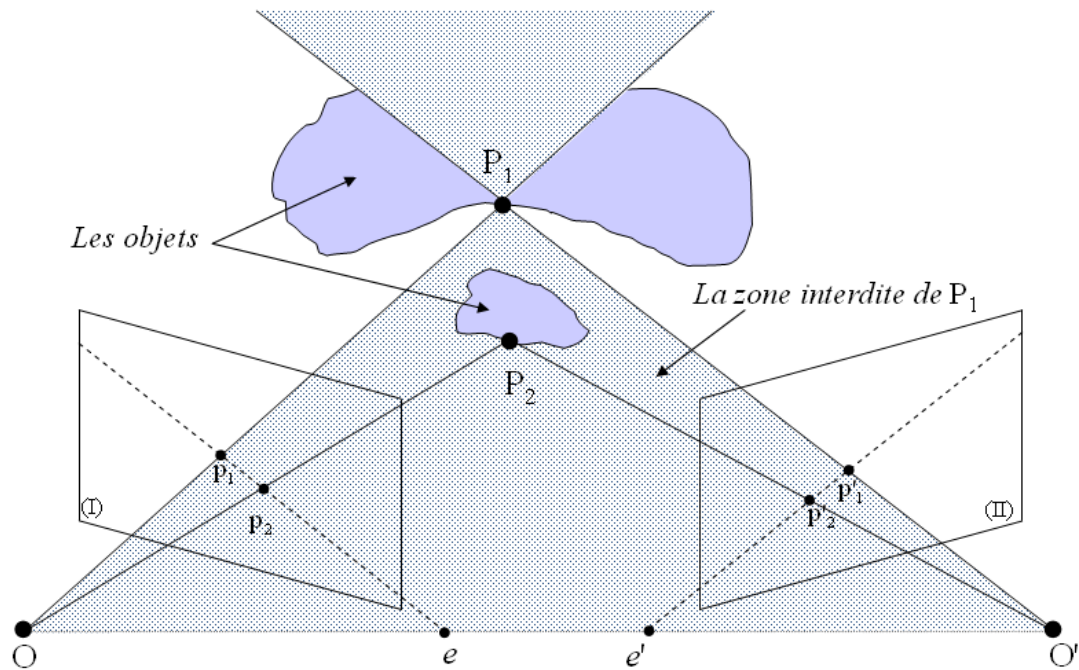


Figure 1.13 - Violation de la contrainte d'ordre : il s'agit d'un point P_2 appartenant à la zone interdite de P_1 . L'ordre de ses projections par rapport à celles de P_1 est inversé d'une image à l'autre.

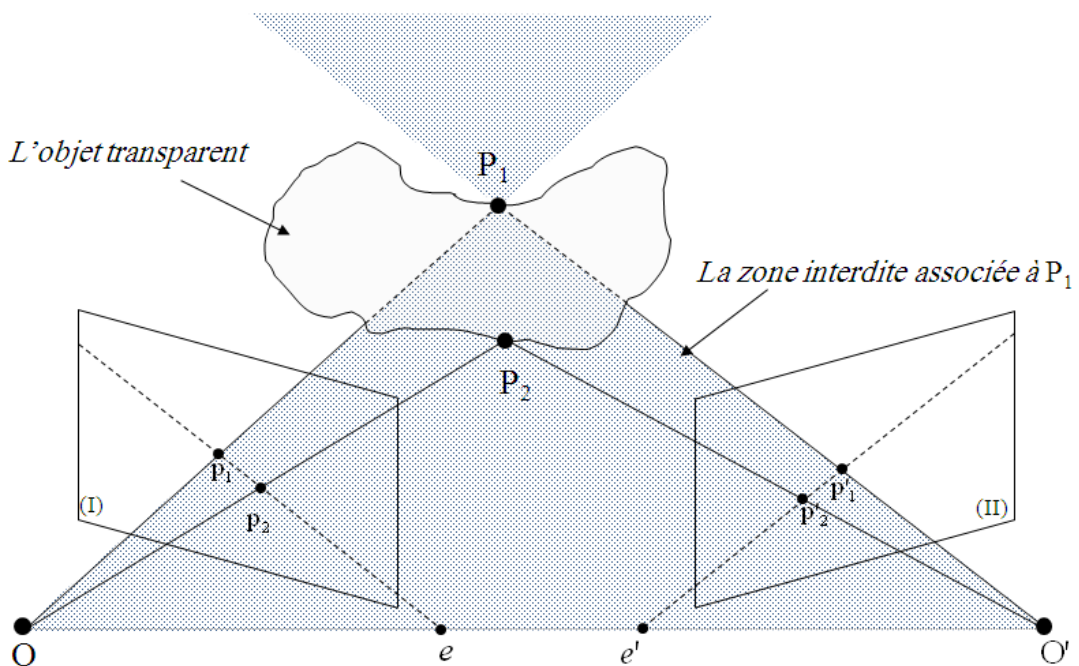


Figure 1.14 - Violation de la contrainte d'ordre, cas d'un objet transparent. L'ordre des projections de P_1 et P_2 sur les plans images est inversé d'une image à l'autre.

Une étude approfondie de la zone interdite, de son utilisation par rapport à la contrainte d'ordre et sa formulation de manière indépendante de la géométrie épipolaire a été effectuée par [Yuille_84]. Faugeras [Faugeras_93A] a défini une zone interdite définie à partir d'un point P_1 , tel que tout point appartenant à cette zone aura l'ordre de ses projections inversé sur les deux plans images. Ainsi, la zone interdite associée à un point physique peut être considérée comme l'ensemble de points qui enfreignent la contrainte d'ordre.

La contrainte de la continuité de la disparité

Elle s'appuie sur l'hypothèse suivante : les surfaces physiques sont localement continues et la variation de leur profondeur par rapport à l'observateur est généralement faible. Implicitement elle suppose que les changements discontinus sont très rares, puisqu'ils n'apparaissent qu'aux frontières physiques des surfaces [Marr_77]. Ceci implique dans les images qu'il n'y a pas de variation brutale de la disparité, et donc le champ de disparité varie presque partout d'une façon continue. Cette contrainte permet d'estimer à l'avance la disparité d'un point à partir des disparités d'autres points respectivement voisins. Autrement dit, des couples des appariements reçoivent leur confirmation par le vote (similitude de la disparité) des points voisins, ce qui permet ainsi dans certains cas de corriger des faux appariements. Une autre contrainte appelée *contrainte de continuité figurale* est également utilisée et qui impose la continuité de la disparité uniquement au long des contours de l'image [Horaud_93].

La contrainte de la disparité maximale

D'après l'équation (1.8) vue précédemment, nous avons la profondeur qui est inversement proportionnelle à la disparité. Selon la configuration du système stéréoscopique, il est possible de déterminer la profondeur minimale que le système peut détecter, et donc la disparité maximale d_{max} entre les paires de points homologues. D'après la même équation (1.8) qui montre que la disparité est inversement proportionnelle à la profondeur, nous pouvons déduire que la profondeur des objets situés devant les caméras stéréoscopiques est obligatoirement positive, ce qui implique des disparités positives ($0 \leq d_{min}$). Ceci impose que l'abscisse d'un point de l'image gauche x_l soit toujours supérieure à l'abscisse du point x_r correspondant dans l'image droite, ce qui vérifie la relation $x_l > x_r$. Cette

contrainte permet de réduire l'espace de recherche lors de la mise en correspondance, tels que seuls les pixels respectant cette contrainte sont traités.

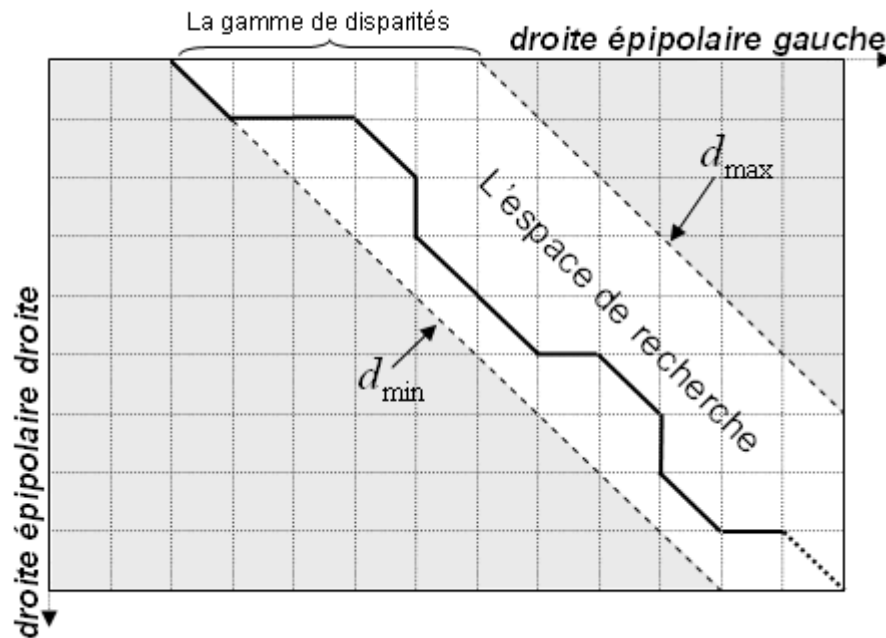


Figure 1.15 – La contrainte de la disparité maximale : une gamme de disparités $[d_{min}, d_{max}]$ est créée pour définir l'espace de recherche des appariements (la zone blanche).

Dans certaines applications, La disparité minimale d_{min} peut être également calculée en définissant ainsi une *gamme de disparités* $[d_{min}, d_{max}]$ (Figure 1.15), Ce qui permet de limiter l'intervalle des disparités à traiter [Sun_02], et donc réduire le taux des faux appariements et le temps de calcul de la mise en correspondance, ce qui est très utile pour les applications temps réel [Mazoul_12, El-Ansari_10A, El-Ansari_10A, Chaplin_98].

Contrainte de visibilité

Cette contrainte traite le problème d'occultation qui permet de pénaliser les zones occultées. Les zones occultées sont des régions dont les points visibles dans une image sont absents dans l'autre image à cause du changement de point de vue (Figure 1.16). La technique la plus utilisée pour faire face à ce problème est celle exploitant l'unicité des points homologues, tel qu'un point de l'image gauche ne correspond qu'un point dans l'image droite, et vice versa. Cette contrainte est introduite dans [Sun_05, Bleyer_05, Min_06, Merrell_07, Kolev_07A, Kolev_07B, Furukawa_07, Furukawa_08].

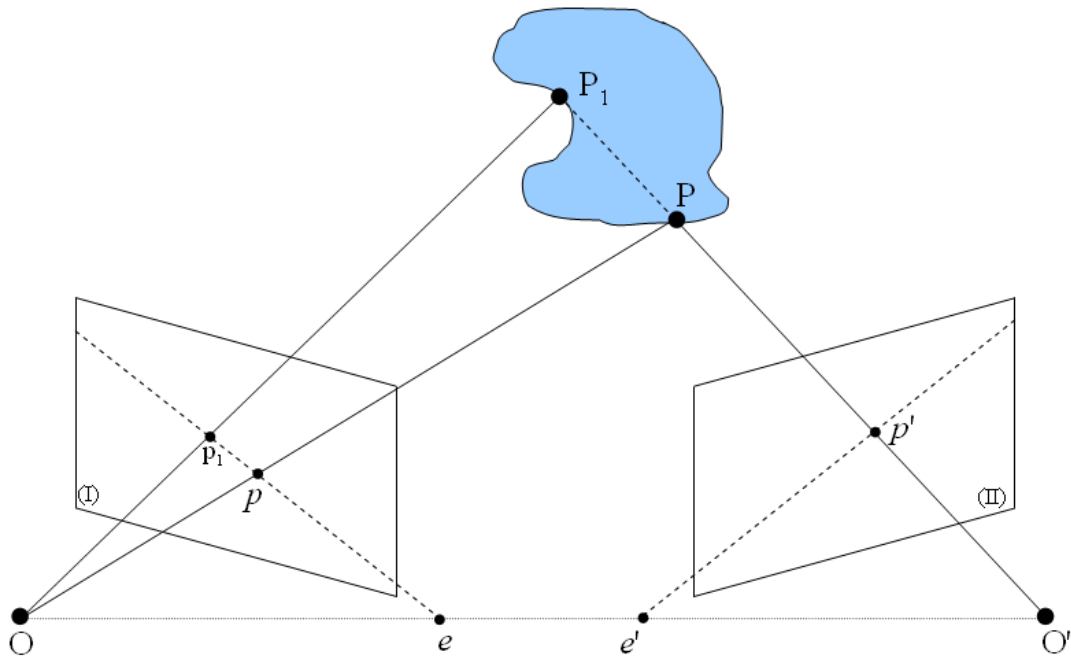


Figure 1.16 – La contrainte de visibilité, le point P_1 est visible par la caméra gauche et invisible pour la caméra droite, d'où l'absence de la projection du point P_1 sur l'image droite.

Contrainte d'invariance photométrique

Cette contrainte suppose que les projections d'un point de l'espace ont des intensités lumineuses similaires sur les deux plans image gauche et droite [Rabaud_04, Miled_09A]. Cela signifie que l'intensité lumineuse d'un point reste invariable au changement de vues. Dans le cas où cette contrainte n'est pas vérifiée, à cause de changement de point de vue, les déformations géométrique des projections, les occultations partielles, etc. Alors les mesures de corrélation sont perturbées, et provoquent ainsi des faux appariements. Nous citons quelques travaux récents intégrant la contrainte d'invariance photométrique [Furukawa_07, Sinha_07, Vogiatzis_07, Campbell_08, Furukawa_08, Hiep_09, Moses_09, Sinha_09, Smith_09, Kolev_10].

1.4.4 Les primitives à mettre en correspondance

Les primitives sont les indices visuels qui caractérisent un ou plusieurs pixels à appairer. Ces primitives peuvent être de différentes natures dont les plus simples sont les pixels de l'image, ou encore les point de contours. Ces primitives seront utilisées ensuite par le processus de la

mise en correspondance afin d'obtenir une carte de disparité dans laquelle on affecte à chaque pixel de la primitive une valeur de disparité.

Dans le cas où l'appariement concerne tous les pixels de l'image, la carte de disparité obtenue est appelée *dense*. Alors qu'on parle d'une carte de disparité *éparse* quand il s'agit seulement de quelques pixels de l'image, comme dans le cas où les primitives sont des points de contours. Quand la majorité, mais pas tous, des pixels de l'image sont appariés, la carte de disparité est appelée *semi-dense*.

Le problème des cartes de disparités dense est qu'elles sont très coûteuses en temps de calcul, et leur utilisation dans des applications qui nécessitent des résultats en temps réel est souvent impossible. L'appariement épars et semi-épars est très utile pour de nombreuses applications en temps réel. L'inconvénient de ces derniers types d'appariement (épars et semi-dense) réside dans l'impossibilité d'avoir une reconstruction 3D complète de la scène. Parmi d'autres primitives qui amènent à la construction des cartes de disparités éparées sont:

- **Points d'intérêts** : sont définis par des pixels ayant des propriétés plus discriminants par rapport à leurs voisins. ces points ont la particularité d'être invariants aux changements de point de vue. Comme type de points particuliers, nous pouvons citer les coins [Birchfield_99A] détectés par exemple par Harris [Harris_88], les contours, les segments, les jonctions, etc. Les points d'intérêts ont été utilisés dans de nombreuses méthodes de mise en correspondance [Konrad_00, Kim_05, Wei_04]. Les détecteurs les plus populaires sont celui de Canny [Canny_86], de Harris et Stephens [Harris_88].
- **Points de contour** : représentent les primitives les plus utilisées dans de nombreux travaux. Ce sont des points qui proviennent des changements brusques de niveau de gris dans l'image. Les détecteurs de points de contours les plus populaires sont Canny, Sobel, et d'autres peu populaires comme l'opérateur de déclivité [Miché_95], et ceux permettant l'extraction des courbes [Schmid_98, Fabbri_10].
- **Segments** : ces primitives ne sont pas beaucoup utilisées que dans les cas où la scène traitée contient des objets plats. Ces primitives sont des segments de droite souvent extraites par la transformation de Hough [Duda_72]. Les travaux de mise en correspondance qui ont utilisé des segments sont nombreux, nous citons à titre d'exemple les travaux de [Boukir_92 et Toulminet_06].

Dans le cadre de notre approche proposée dans ce manuscrit, les primitives que nous considérons pour la mise en correspondance sont les points de contours. L'algorithme génère ainsi des cartes de disparités éparses en temps réel avec une grande précision au niveau des résultats.

1.4.5 Les approches de mise en correspondance stéréoscopique

L'objectif maintenant est de trouver les points homologues dans le cas de deux images rectifiées gauche et droite. Le processus de la mise en correspondance consiste alors à comparer la similarité entre un pixel avec un sous-ensemble de pixels candidats de l'autre image. Les deux pixels qui maximise au minimise le score de similarité seront ensuite considérés comme des points homologues. Cependant, une simple comparaison des intensités lumineuses (niveaux de gris ou couleur) des pixels à apparier n'est pas toujours suffisante, ce qui nécessite d'appliquer un ensemble de contraintes stéréoscopiques vues précédemment (section 1.4.3). Certaines méthodes de mise en correspondance, dites *locales*, suggèrent une analyse des intensités lumineuses locale autour de chaque pixel dans les deux images, cela se fait dans une zone dite d'agrégation centrée ou non sur chaque pixel à apparier. Ces méthodes considèrent la contrainte de continuité des disparités dans le voisinage des points à apparier sans tenir compte de l'information contenue dans le reste de l'image. D'autres méthodes, dites *globales*, appliquent les contraintes stéréoscopiques sur la globalité de l'image pour l'estimation récursive de la disparité [Veksler_03]. D'autres méthodes, dites semi-globales, font intervenir dans son processus uniquement les pixels d'une [Forstmann_04] ou plusieurs lignes épipolaires adjacentes [Gong_05].

Le processus de la mise en correspondance a été largement étudié en vision par ordinateur. Un ensemble de méthodes de l'état de l'art peut être trouvé dans [Scharstein_02]. Traditionnellement, les algorithmes d'appariement stéréoscopique sont essentiellement classés en deux catégories: les méthodes locales et globales.

1.4.5.1 Les mesures de similarité

La mesure de similarité (dite aussi *mesure* ou *fonction de corrélation*) constitue le premier élément de tout algorithme de mise en correspondance stéréoscopique. Cette mesure permet la comparaison des valeurs des pixels à apparier afin d'évaluer leur degré de ressemblance. Dans

cette section, nous présentons brièvement les mesures de similarité les plus couramment utilisées et qui ont été développées spécifiquement pour l'appariement stéréoscopique.

Le principe d'une mesure de similarité, est de considérer que les deux pixels homologues et leur voisinage respectivement sont similaires d'un point de vue photométrique. Ce principe considère que la meilleure correspondance d'un pixel de l'image gauche avec un autre pixel dans l'image droite est établie en recherchant la meilleure ressemblance possible de mesure de similarité entre des fenêtres centrées aux deux pixels à apparier respectivement. L'utilisation des mesures de similarité fournissent de bons résultats et sont, en termes de traitement, de complexité variable. Les systèmes basés sur la corrélation souffrent de quelques limitations inhérentes à la scène observée, car ces mesures sont généralement sensibles : à l'absence de texture, à la présence de structures répétitives (Figure 1.10), aux occultations (Figure 1.16), aux distorsions de perspective et aux discontinuités des surfaces. Plusieurs mesures de similarité sont proposées dans la littérature [Aschwandten_92, Devernay_97]. Nous présentons les fonctions d'appariement les plus couramment utilisées pour mesurer la similarité entre deux fenêtres de corrélations de taille $(2w + 1) \times (2w + 1)$, une prise autour du pixel $I_g(x_l, y)$ de l'image gauche, et l'autre autour du pixel $I_d(x_l - d, y)$ candidat de la même ligne dans image droite:

- **Somme des écarts absolus (SAD) :**

$$SAD(x_l, y, d) = \sum_{u=-w}^w \sum_{v=-w}^w |I_g(x_l + u, y + v) - I_d(x_l + u - d, y + v)| \quad (1.14)$$

- **Somme des écarts absolus centrés (ZSAD) :**

$$ZSAD(x_l, y, d) = \sum_{u=-w}^w \sum_{v=-w}^w |(I_g(x_l + u, y + v) - \bar{I}_g) - (I_d(x_l + u - d, y + v) - \bar{I}_d)| \quad (1.15)$$

- **Somme des écarts quadratiques (SSD) :**

$$SSD(x_l, y, d) = \sum_{u=-w}^w \sum_{v=-w}^w (I_g(x_l + u, y + v) - I_d(x_l + u - d, y + v))^2 \quad (1.16)$$

- **Somme des écarts quadratiques centrés (ZSSD) :**

$$ZSSD(x_l, y, d) = \sum_{u=-w}^w \sum_{v=-w}^w \left((I_g(x_l + u, y + v) - \bar{I}_g) - (I_d(x_l + u - d, y + v) - \bar{I}_d) \right)^2 \quad (1.17)$$

- **Somme des écarts quadratiques centrés normalisés (NZSSD) :**

$$NZSSD(x_l, y, d) = \sum_{u=-w}^w \sum_{v=-w}^w \left(\frac{(I_g(x_l + u, y + v) - \bar{I}_g)}{Norm_w^g(x_l, y)} - \frac{(I_d(x_l + u - d, y + v) - \bar{I}_d)}{Norm_w^d(x_l - d, y)} \right)^2 \quad (1.18)$$

Où :

$$Norm_w^g(x_l, y) = \sum_{u=-w}^w \sum_{v=-w}^w \sqrt{(I_g(x_l + u, y + v) - \bar{I}_g)^2} \quad (1.19)$$

$$Norm_w^d(x_l - d, y) = \sum_{u=-w}^w \sum_{v=-w}^w \sqrt{(I_d(x_l + u - d, y + v) - \bar{I}_d)^2} \quad (1.20)$$

- **Corrélation croisée normalisée (NCC) :**

$$NCC(x_l, y, d) = \frac{\sum_{u=-w}^w \sum_{v=-w}^w I_g(x_l + u, y + v) \cdot I_d(x_l + u - d, y + v)}{\sqrt{\sum_{u=-w}^w \sum_{v=-w}^w (I_g(x_l + u, y + v))^2 \cdot (I_d(x_l + u - d, y + v))^2}} \quad (1.21)$$

- **Corrélation croisée normalisée centrée (ZNCC) :**

$$ZNCC(x_l, y, d) = \frac{\sum_{u=-w}^w \sum_{v=-w}^w (I_g(x_l + u, y + v) - \bar{I}_g) \cdot (I_d(x_l + u - d, y + v) - \bar{I}_d)}{\sqrt{\sum_{u=-w}^w \sum_{v=-w}^w (I_g(x_l + u, y + v) - \bar{I}_g)^2 \cdot (I_d(x_l + u - d, y + v) - \bar{I}_d)^2}} \quad (1.22)$$

Au lieu d'appliquer les mesures de similarités sur les niveaux de gris des pixels à appairer, la transformation de Census (ou de recensement) peut être utilisée comme étape prétraitement qui consiste à remplacer la valeur de chaque pixel par une chaîne binaires dont chaque bit représente un pixel voisin. Le bit associé à un pixel voisin est mis à « 1 » lorsque son niveau de gris est supérieur à celui du pixel central et à « 0 » sinon (Figure 1.17). Une fois l'image de Census est créée, le score d'appariement est effectué ensuite en comptant le nombre de bits qui diffèrent entre les deux chaînes binaire associées aux deux pixels à appairer. Pour cela, la distance de Hamming est souvent utilisée pour cette raison. Figure 1.17 illustre la transformation de Census appliquée sur un pixel dans une fenêtre de 5x5. Ainsi la chaîne binaire est obtenue de la façon suivante :

- le premier bit est 0 car $12 < 23$,
- le second 0 car $14 < 23$,
- le troisième 1 car $26 > 23$, ...

12	14	26	32	31	0	0	1	1	1
13	16	24	30	33	0	0	1	1	1
11	15	23	31	33	0	0	×	1	1
10	14	22	29	32	0	0	0	1	1
12	17	25	34	36	0	0	1	1	1

001110011100110001100111

Figure 1.17 - Transformation de Census d'un pixel sur une fenêtre de 5x5, à droite la valeur de Census obtenue pour le pixel central (le pixel ayant la valeur 23 sur la fenêtre de gauche).

Soient a et b deux chaînes binaires associées aux deux pixels à apparier, respectivement. La distance de Hamming est le nombre de bits de a qui diffèrent de ceux de b correspondants, et cela peut s'exprimer comme suit :

$$Hamming(a, b) = \sum_{i=1}^n (a_i \oplus b_i) \quad (1.23)$$

Où: $\oplus$ désigne le ou-exclusif.

1.4.5.2 Les méthodes locales

Les méthodes locales, aussi appelées mise en correspondance par corrélation, sont les plus populaires en stéréovision grâce à leurs efficacités et leurs simplicités. Ces méthodes se basent sur l'hypothèse que les intensités lumineuses (niveaux de gris ou couleur) au voisinage des pixels homologues sont similaires. On appelle une *fonction de coût* ou de *vraisemblance* (Section 1.4.5.1) toute fonction de corrélation permettant l'évaluation de degré de similarité (ou dissimilarité) entre deux ensembles de pixels. La fonction coût est définie dans une fenêtre de taille fixe (carrée ou rectangulaire) autour d'un pixel de l'image gauche et une autre fenêtre de même taille autour d'un pixel candidat sur la ligne épipolaire correspondante dans l'image droite. Dans notre cas où les images sont considérées rectifiées, les lignes épipolaires sont confondues avec les lignes images.

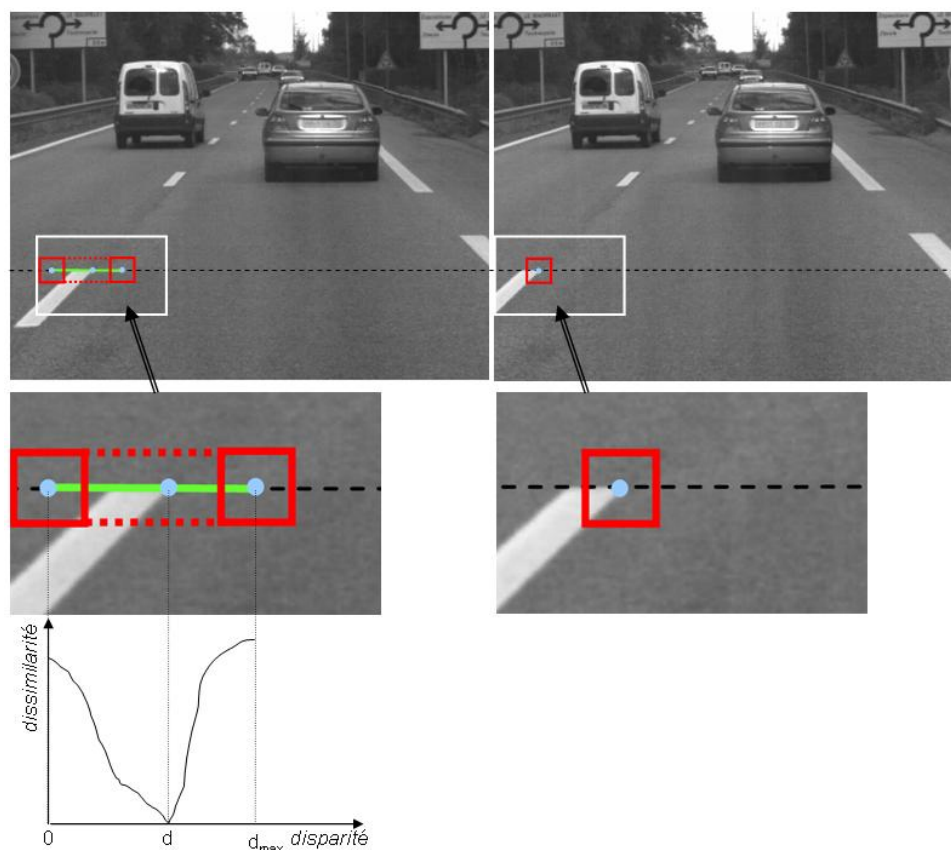


Figure 1.18 – Mise en correspondance par corrélation : Une fenêtre est définie autour du pixel de l'image droite à appairer et une autre fenêtre de même taille est décalée au long de l'intervalle de recherche sur la même ligne de l'image gauche, permettant d'évaluer le meilleur score de corrélation.

Le principe des méthodes locales basées sur la corrélation est illustré dans la Figure 1.18. Etant donné un pixel $I_r(x_r, y)$ dans l'image droite dont nous cherchons à calculer la correspondance. Une fenêtre est définie autour de $I_r(x_r, y)$ et une autre fenêtre de même taille est décalée au long de l'intervalle de recherche (ou gamme de disparité) sur la même ligne de l'image gauche. A chaque passage, une fonction de coût entre les deux fenêtres est calculée afin d'évaluer les scores de similarité. Le décalage à la fenêtre qui minimise (ou maximise) les scores de similarité sera considérée comme la valeur de disparité du pixel à appairer. Le score obtenu constitue ainsi *le coût de correspondance* (ou *matching cost*). Les approches d'appariement par corrélation exploitent la contrainte de continuité des objets sur les surfaces de la scène. Elle suppose que les pixels de la fenêtre de corrélation ont les mêmes valeurs de disparités, ce qui provoque certaines ambiguïtés au niveau de l'appariement.

Une difficulté majeure des méthodes locales basées sur la corrélation est la sélection de la taille et la forme de la fenêtre à utiliser. En effet, la majorité des méthodes utilisent une fenêtre rectangulaire de taille fixe. Toutefois, la limitation de cette technique est qu'elle échoue sur les zones où il y a des discontinuités de disparités. Pour surmonter ce problème, des approches plus sophistiquées ont été proposées en faisant varier la taille et la forme de la fenêtre en fonction de la variation d'intensité [Kanade_94, Woo_00, Veksler_01, Xing_09].

1.4.5.3 Les méthodes globales

Dans cette section, nous proposons un état de l'art sur quelques méthodes globales permettant l'appariement stéréoscopique. Dans ces approches, tous les pixels des images interviennent dans le processus d'optimisation [Veksler_03]. Une fonction de coût globale est évaluée entre les deux images gauche et droite. La carte de disparité calculée par la méthode globale est la carte qui minimise le coût global $E(d)_{global}$.

$$E(d)_{global} = E_{correspondance}(d) + E_{lissage}(d) \quad (1.24)$$

La fonction de coût globale E_{global} se décompose en deux fonctions:

- le coût de correspondance $E_{correspondance}$.
- le coût des contraintes $E_{lissage}$.

Le coût de correspondance $E_{correspondance}$ s'exprime, comme pour les méthodes locales, en fonction de la mesure de similarité et de la zone d'agrégation choisie. Généralement, les méthodes globales utilisent les coûts d'appariement d'un seul pixel courant. Alors que le coût des contraintes $E_{lissage}$, comme son nom l'indique, dépend d'une combinaison des contraintes de la mise en correspondance exposées précédemment dans la section 1.4.3, et décrit la contrainte du lissage permettant le contrôle de la convergence de l'algorithme utilisé. Une fois que le coût global a été défini, différentes méthodes d'optimisation peuvent être appliquées pour trouver un extremum.

La recherche la carte de disparité est obtenue en minimisant donc la fonction de coût global E_{global} . Plusieurs algorithmes de minimisation d'énergie ont été proposés pour résoudre l'équation (1.24). Les approches les plus courantes sont la programmation dynamique

[Sun_02, Kim_05, Veksler_05], les coupes de graphes (graph-cuts) [Boykov_01, Kolmogorov_01], et les méthodes variationnelles [Slesareva_05, Miled_06, Miled_09B, Kosov_09]. Comme indiqué plus haut, les méthodes globales font intervenir en général tous les pixels de l'image [Brown_03]. Cependant, certains auteurs proposent de réduire le calcul du coût à une seule ligne épipolaire [Forstmann_04] ou plusieurs lignes épipolaires adjacentes [Gong_05].

Nous décrivons ci-après quelques méthodes globales d'optimisation les plus utilisées dans les algorithmes de mise en correspondance stéréoscopique :

Programmation dynamique

La programmation dynamique (PD) est largement utilisée dans le cadre de l'optimisation de l'énergie dans le domaine de l'intelligence artificielle et la vision par ordinateur. Il permet de réduire le coût d'un problème d'optimisation en le divisant en plusieurs petits sous-problèmes, tel que chacun d'entre eux peuvent être facilement résolus. Les solutions des sous-problèmes sont ensuite combinées pour trouver la solution optimale du problème initial.

La technique de la programmation dynamique utilise la contrainte d'ordre pour optimiser le processus d'appariement entre deux lignes images conjuguées. Deux étapes peuvent être distinguées dans cette technique: une étape en avant (Algorithme 1.1) permettant de construire la matrice des coûts pour toutes les paires de pixels candidats possibles, et une étape en arrière (Algorithme 1.2) où les paires de pixels correspondants sont sélectionnées avec un chemin de coût minimal (ou maximal selon la fonction de coût utilisée). Autrement dit, la solution optimale correspond au chemin qui optimise le coût global dans un graphe dont chaque nœud représente le coût obtenu pour un couple de pixels candidats sur les deux droites épipolaires conjuguées (Figure 1.19) [Birchfield_98, Benschraier_96, Bobick_99]. Le coût calculé au niveau de chaque nœud est obtenu en utilisant une fonction de corrélation (SSD, SAD, etc). Le calcul d'une carte de disparité dense nécessite parfois une contrainte de lissage (ou continuité de disparités). Dans ce cas, la programmation dynamique extrait l'ensemble des disparités pour tous les pixels de l'image de référence, donnant ainsi la meilleure corrélation tout en respectant les contraintes de lissage et d'ordre.

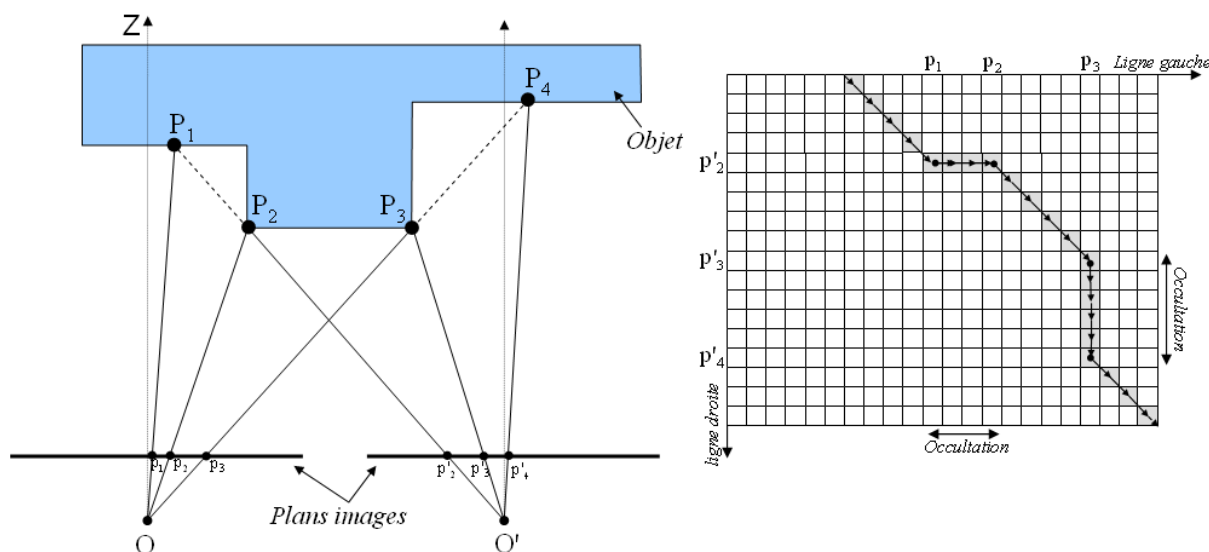


Figure 1.19 – Exemple d'utilisation de la programmation dynamique. (A gauche) quatre points de la scène projetés sur les plans d'images gauche et droite. (A droite) Matrice de la programmation dynamique avec le meilleur chemin de correspondance.

Bien que cette méthode ait l'avantage de respecter cette contrainte de lissage des disparités au long de la ligne épipolaire, il souffre du problème de la discontinuité des disparités provoqué principalement à la présence des objets ayant des profondeurs différentes sur la ligne ainsi que le problème de non-respect de la cohérence interlignes. Pour résoudre ces problèmes, une technique en deux phases de programmation dynamique a été développée dans [Kim_05, Otha_89].

Dans un algorithme basé sur la programmation dynamique, nous devons calculer la matrice de coût C , dans laquelle chaque élément C_{ij} correspond au coût de similarité entre le pixel i de la ligne de l'image droite avec un pixel j sur la ligne correspondante de l'image gauche. En balayant les lignes de la matrice C , nous analysons cellule par cellule en choisissant l'une des trois dernières cellules adjacentes qui forme un chemin du coût minimal avec la cellule courante. Un déplacement en horizontal ou vertical dans la matrice correspond à une occultation, alors qu'un déplacement en diagonale correspond à une éventuelle correspondance (Figure 1.19). En plus de la matrice des coûts, nous aurons une matrice M dans laquelle nous indiquons le chemin suivi pour arriver à chaque cellule (voir l'algorithme 1.1).

Algorithme 1.1 : *L'étape en avant de la programmation dynamique*

Occlusion = constante.

Début

Pour $i \leftarrow 1$ à i_{max}

$C(i, 0) = i * Occlusion$

FinPour

Pour $j \leftarrow 1$ à j_{max}

$C(0, j) = j * Occlusion$

FinPour

Pour $(i, j) \in \text{Liste des cellules de } C$

$min1 \leftarrow C(i - 1, j - 1) + C(i, j)$

$min2 \leftarrow C(i - 1, j) + Occlusion$

$min3 \leftarrow C(i, j - 1) + Occlusion$

$cmin \leftarrow \min(min1, min2, min3)$

$C(i, j) \leftarrow \min(min1, min2, min3)$

Faire suivant le cas

Cas $min1 = cmin$:

$M(i, j) \leftarrow 1$

Cas $min2 = cmin$:

$M(i, j) \leftarrow 2$

Cas $min3 = cmin$:

$M(i, j) \leftarrow 3$

FinCas

FinPour

Fin

Une fois la matrice M est créée (c'est-à-dire en allant de haut en bas), l'algorithme doit aller dans le sens inverse en suivant le chemin qui a produit le coût minimal (c'est-à-dire en allant de bas en haut). De cette façon, nous obtiendrons un chemin qui optimise le coût global de l'appariement entre les points de deux lignes conjuguées. Pour chaque correspondance sur le chemin, nous associons une valeur de disparité égale à la distance entre leurs abscisses. L'algorithme 1.2 décrit la phase de retour dans la programmation permettant extraction du chemin optimal, à savoir le chemin ayant le coût minimal.

La programmation dynamique peut être également utilisée pour l'estimation des cartes éparses, cela pour l'appariement des pixels de contour [Baker_81, Benard_84, Ohta_85, Sadeghi_08] ou des segments de droite [Lee_94]. [Ohta_85] recherche un chemin optimal non seulement en intra-ligne mais aussi en interlignes, ce qui consiste à faire une recherche optimale dans une région bidimensionnelle définie par des contours horizontaux et

verticaux. Cette dernière méthode prend bien en compte la cohérence interlignes, et donne ainsi une meilleure solution par rapport à la première. Le défaut majeur de cette approche est sa gourmandise en temps-machine.

Algorithme 1.2 : *L'étape de retour dans la programmation dynamique : extraction du chemin optimal*

Début

$p \leftarrow i_{max}$

$q \leftarrow j_{max}$

Tantque ($p \neq 0$ et $q \neq 0$) **faire**

Faire suivant le cas

Cas $M(p, q) = 1$:

$d(p) \leftarrow p - q$

$p \leftarrow p - 1$

$q \leftarrow q - 1$

Cas $M(p, q) = 2$:

"le pixel p est occulté"

$p \leftarrow p - 1$

case $M(p, q) = 3$:

"le pixel q est occulté"

$q \leftarrow q - 1$

FinCas

FinTanque

Fin

Coupages de graphe (graph-cuts)

La majorité des méthodes de mise en correspondance par programmation dynamique essaient d'apparier les pixels appartenant à la même ligne épipolaire sur les deux images sans tenir compte de la cohérence interlignes. Par conséquent, ils n'utilisent pas la nature bidimensionnelle du problème. Pour surmonter cet inconvénient, tenant en considération la contrainte de continuité bidimensionnel, une solution a été proposée en utilisant la théorie des graphes. Roy et Cox [Roy_98] étaient les premiers qui ont appliqué les graph-cuts pour résoudre le problème de la mise en correspondance stéréoscopique. Ce problème a été reformulé ensuite par [Veksler_99] dans lequel il considère le problème de l'appariement

comme une minimisation d'une fonction d'énergie. Ensuite, des algorithmes de graph-cuts itératifs ont été introduits dans [Kolmogorov_01, Kolmogorov_02A, Kolmogorov_02B].

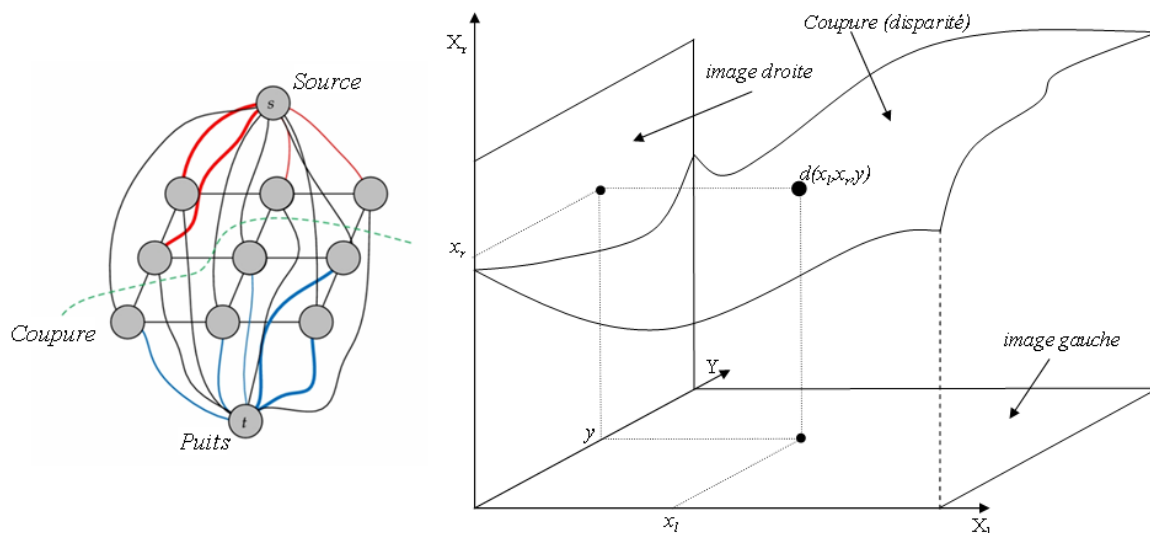


Figure 1.20 – Une représentation 3D d'un graphe coupé, (à gauche) les cercles représentent les nœuds et les lignes continues sont les arêtes entre les nœuds, la ligne pointillé indique une coupure, (à droite) la coupure minimale qui représente la surface de profondeur.

L'algorithme de *Coupure de graphe* (ou Graph-cuts) consiste à représenter le problème de la mise en correspondance sous forme d'un ensemble de nœuds et d'arêtes. Les nœuds du graphe représentent le coût local d'une paire de pixels candidats, alors que les arêtes reliant les nœuds assurent les contraintes de continuité et de lissage des disparités. Le graphe est composé d'une source s , un puits t et des points (x_l, y, d) , où (x_l, y) sont les coordonnées des pixels de l'image de référence (ici l'image gauche) (Figure 1.20), et d les disparités possibles pour ce pixel. A l'intérieur, chaque nœud est connecté avec ses six nœuds voisins, par quatre arêtes dans le même plan (appelé *n-link*) et deux arêtes avec deux nœuds dans les plans adjacents. Cette approche consiste donc à trouver une solution en coupant le graphe en plusieurs coupures. La coupure ayant un coût minimal sera considérée comme solution du problème. Les méthodes basées sur les graph-cuts permettent la détermination d'une solution au problème de minimisation du coût global donnant ainsi des cartes de disparités plus lisses par rapport aux approches locales. Les

méthodes récentes de coupures des graphes sont discutées dans [Paragios_06] (pages 79-95).

Malgré que les approches basées sur les graph-cuts donnent des résultats beaucoup plus lisses, elles sont, d'un point de vue expérimental, très coûteuses en temps de calcul et ne conviennent pas à des applications qui exigent des résultats en temps réel.

1.4.6 Méthodes temps-réel de mise en correspondance stéréoscopiques

Cette section donne un aperçu sur les plus importants systèmes stéréoscopiques en temps-réel basés sur des algorithmes proposés dans les deux dernières décennies. Nous étudions les systèmes basés sur du matériel dédié, ainsi que les systèmes basés sur des PC standards.

La contrainte de temps dans les méthodes de la mise en correspondance est très importante quand il s'agit des applications qui exigent l'exécution avec une cadence vidéo. Dans ce cas, le temps de calcul doit être inférieure à la cadence imposée par le système. Un nombre très important de recherches sur la stéréovision en temps-réel suivent deux axes principaux :

- Développement des algorithmes qui ne nécessitent pas de temps de calcul élevé, tout en fournissant de bons résultats.
- Utilisation de dispositifs matériels dédiés permettant de diminuer le temps de calcul.

Ces dispositifs comprennent le matériel à usage spécial, tels que les processus de signal numérique (Digital Signal Processor, ou DSP) ou Field Programmable Gate-Arrays (FPGA), et également des extensions vers les PC récents, tels que les instructions spécialisées d'extension multimédia (MMX) disponibles dans les processeurs récents [Forstmann_04, Hirschmuller_02] comme les unités de traitement graphique (dits aussi processeurs graphique ou GPU) [Gong_05, Marc_04].

Les processus de signal numérique DSP sont beaucoup utilisés dans les applications en temps réel. Les DSPs contiennent des instructions spécifiques permettant le traitement de plusieurs données simultanément en exploitant le parallélisme des algorithmes. A INRIA, Faugeras

[Faugeras_93B] a développé un système de vision trinoculaire basé sur quatre DSPs, et une corrélation croisée normalisée NCC (Normalised Cross Correlation) en tant que mesure de corrélation. Le système de Faugeras fournit des cartes de disparités de taille 64x64 avec une cadence de 400 cartes par seconde.

Un DSP est également utilisé par Kanade [kanade_96] afin de développer un système de vision à plusieurs caméras (jusqu'à 6 caméras) capable de fournir des carte de disparités avec une résolution de 256x256 à la cadence de 30 images par seconde. La somme des différences au carrée (SSD) a été utilisée comme mesure d'appariement local. L'algorithme d'appariement utilise également une valeur globale (SSSD) déduite à partir des résultats calculés pour les différents points de vue.

Un système basé sur une carte FPGA développé par Woodfill [Woodfill_97] et qui s'insère dans une carte PCI standard, produit 42 cartes de disparités par seconde à partir d'une séquence d'images stéréoscopique de taille 320x240. Ce système exploite une transformation de Census afin d'accomplir le processus d'appariement. Récemment, Tyzx [Site_Tyzx] a proposé un système stéréoscopique basé sur un algorithme similaire et une FPGA compatible à la carte PCI standard. Le système Tyzx fournit des cartes de profondeur de taille 512x512 à une cadence maximale de 200 cartes par seconde. Un autre système stéréoscopique basé sur FPGA et qui repose sur la transformation de Census a été développé par Corke [Corke_99]. Il permet d'exécuter jusqu'à 30 carte de disparité par seconde pour une séquences d'images stéréoscopiques de 256×256 de résolution. Darabiha [Darabiha_03] a développé un système de vision stéréoscopique temps-réel basé sur une carte FPGA capable de fournir jusqu'à 30 images par seconde avec des paires d'images stéréoscopique de taille 256×360 . Le système s'appuie sur un algorithme d'appariement stéréoscopique multi-résolutions et multi-orientations (MRMO) [Fleet_94].

Dans un système développé par [Ttofis_12], l'architecture met en œuvre un algorithme de mise en correspondance stéréoscopique. L'algorithme obtient des données à partir de la segmentation d'images afin d'augmenter la robustesse du processus d'appariement. Le système proposé intègre des techniques d'optimisation permettant à l'algorithme d'être adapté aux systèmes de vision embarqués. Un prototype de l'architecture a été mis en œuvre sur un FPGA, en réalisant ainsi une cadence de 30 fps pour des images de 640x480.

Dans [Jia_04], les auteurs ont proposé un système de vision (MSVM-III) avec trois caméras (Figure 1.21) permettant de générer des cartes de disparités à haute résolution et à la cadence vidéo. Le MSVM-III utilise une seule puce FPGA pour faire la rectification trinoculaire des images, un filtrage de type LOG et un algorithme d'appariement basé sur les régions. Leur système est capable de traiter plus de 30 cartes de disparité denses par seconde pour des images de 640×480 pixels de résolution et pour une gamme de disparités de 64 pixels, et 120 cartes/seconde pour les images de taille 320×240 pixels. En outre, le MSVM-III a un port IEEE 1394 permettant un débit à la cadence vidéo des données.

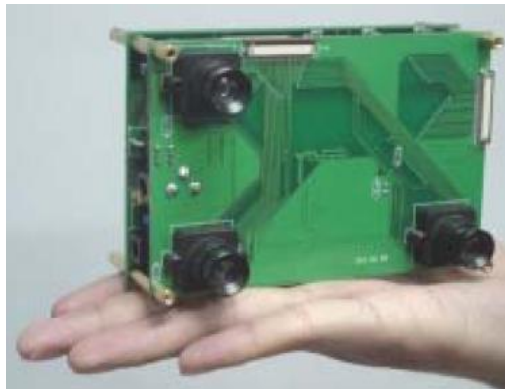


Figure 1.21 – Le système de vision MSVM-III proposé par [Jia_04]

Une puce de traitement vidéo [Van_01] a été développée par Sarnoff Corporation capable d'effectuer la mise en correspondance à la cadence vidéo pour des images de 320×240 pixels de résolution. Ce système utilise la somme des différences absolues (SAD) en tant que mesure de similarité, avec une vérification de cohérence gauche-droite des cartes calculées.

Sazan [Kimura_99] a développé un système de vision poly-oculaire (composé de 9 caméras) ayant deux cartes PCI. Une combinaison de plusieurs mesures SAD (SSAD) sont utilisées afin d'offrir une précision sous-pixel. Le système de Sazan calcule jusqu'à 20 cartes de profondeur par seconde de taille 200×200 avec une gamme de disparités de 25 pixels.

Il est intéressant de souligner que la bibliothèque OpenCV [OpenCV] inclut une implémentation en temps-réel de l'algorithme de Birchfield et Tomasi [Birchfield_99B] basé sur la programmation dynamique. Nous proposons aux lecteurs intéressés de se rapporter à [Sunyoto_04] pour une étude comparative de certains algorithmes de mise en correspondance stéréoscopique dense en temps-réel basés sur les PCs.

[Jin_10] présente une mise en œuvre de FPGA plus récente pour un système stéréoscopique en temps-réel. Ce système, dont l'algorithme est basé sur transformation de Census, produit des cartes de disparités denses. La mise en œuvre du matériel présenté dans [Ambrosch_10] permet l'exécution de 60 images par seconde pour une résolution de 750x400 pixels. L'algorithme utilise une version modifiée de la transformation de Census à la fois sur l'intensité et le gradient des images en combinaison avec la corrélation SAD.

Le tableau ci-dessous résume certains systèmes de mise en correspondance temps-réel implémentés sur différentes plateformes dédiées [Naoulou_06].

<i>Année</i>	<i>Auteur</i>	<i>Plateforme</i>	<i>Algorithme</i>	<i>Taille image</i>	<i>Images/s</i>
93	INRIA [Faugeras_93B]	DSP96002	-	64x64	2.5
96	Kanade [Kanade_96]	DSP C40	SSD	200x200	25
97	Woodfill [Woodfill_97]	FPGA	Census	320x240	42
97	SRI [Konolige_97]	DSP2181	SAD	160x120	8
99	Corke [Corke_99]	FPGA	Census	256x256	30
99	Gluckman [Gluckman_99]	Pentium II	SAD	600x60	7
99	SAZAN [Kimura_99]	PCI	SAD	320 x240	20
01	Areibi [Areibi_01]	FPGA XS40	P2P	-	4
03	Yang [Yang_03]	PC	SSD	-	8
03	Darabiha [Darabiha_03]	FPGA	MRMO	256x360	30
03	Jia [Jia_03]	FPGA	SSAD	640x480	30
04	Jia [Jia_04]	FPGA	SSAD	320x240	120
10	K. Ambrosch[Ambrosch_10]	FPGA	Census	750x400	60
12	Ttofis [Ttofis_12]	FPGA	segmentation	640x480	30

Tableau 1.1 – Les méthodes temps-réel de mise en correspondance.

1.4.7 Les approches de mise en correspondance spatio-temporelles

Ces dernières années, plusieurs techniques ont été proposées pour obtenir des cartes de disparité plus précises à partir des séquences stéréoscopiques en utilisant la cohérence temporelle [Davis_05, Gong_06, Tao_01, Zhang_03]. La plupart de ces méthodes utilisent soit l'algorithme du flot optique, soit une fenêtre spatio-temporelle dans leur processus d'appariement stéréoscopique. [Tao_01] propose une approche pour récupérer une profondeur dynamique à partir de plusieurs flux vidéo synchronisés, permettant ainsi d'obtenir progressivement une représentation 3D de la scène. La méthode de Tao fait appel à une segmentation couleur afin de modéliser chaque segment comme un plan 3D dont le mouvement est décrit en utilisant une vitesse constante. Une mesure de similarité et une contrainte du flot de la scène sont également utilisées dans le processus d'appariement [Vedula_91]. Le temps de calcul de cette méthode et la précision de ses résultats sont limitées par l'algorithme de segmentation utilisé.

Les auteurs dans [Vedula_91] ont présenté un algorithme linéaire permettant de calculer un flot de la scène stéréoscopique à partir du flot optique 2D. La reconstitution de l'information 3D est estimée ensuite à partir du flot de la scène stéréoscopique. Dans [Leung_04], la cohérence temporelle a été appliquée pour minimiser la différence entre les cartes de disparités des trames (frames) successives. Cette approche a été conçue seulement pour un traitement hors ligne (c'est-à-dire qu'il prend en entrée une séquence stéréoscopique et calcule ensuite les cartes de disparités correspondantes). [Gong_06] a développé un algorithme pour calculer à la fois des cartes de disparités et de flot de disparités d'une manière intégrée. La carte de disparité générée pour la trame courante est utilisée pour estimer la carte de disparité de la trame suivante. La carte de disparité estimée fournit des informations sur la correspondance spatiale de l'instant précédent afin de valider les cartes de flot de disparités estimées pour différents points de vue. Des cartes graphiques programmables ont été utilisées pour accélérer le processus du traitement.

Zhang [Zhang_03], a amélioré des méthodes traditionnelles en utilisant à la fois les deux informations, spatiale et temporelle. Une fenêtre spatiale a été typiquement utilisée pour calculer la somme des différences au carré (SSD) qui a été étendue à une fenêtre spatio-

temporelle pour calculer la somme de plusieurs SSD (SSSD). Les résultats de leur méthode sont beaucoup améliorés lorsqu'il s'agit d'une scène statique sous une lumière structurée. En revanche, la méthode ne fonctionne plus lorsqu'il s'agit de scènes dynamiques. Davis [Davis_05] a développé une approche similaire à celle de [Zhang_03]. Cependant, leur travail s'est focalisé sur des scènes ayant une géométrie statique et qui sont capturées sous un éclairage variable. Etant donné une séquence d'images stéréoscopique prise par une caméra en mouvement libre, Zhang [Zhang_09] a proposé une nouvelle approche pour construire pour chaque trame de la séquence une carte de profondeur dépendante de la vue. La méthode prend en entrée une séquence d'images stéréoscopique et fournit ensuite des cartes de profondeur correspondantes. Cette approche a également été conçue pour des traitements hors-ligne, et donc inapplicable pour des systèmes nécessitant des résultats en temps-réel (comme dans le cas des véhicules intelligents).

1.4.8 Les techniques de raffinement

Le raffinement des cartes de disparités est une étape post-traitement supplémentaire d'un algorithme de mise en correspondance permettant d'améliorer la qualité d'appariement. Une grande majorité de méthodes d'appariement stéréoscopique utilisent ces techniques dans leur algorithme. Cette étape ne s'agit pas de faire de mesures, mais bien d'une étape supplémentaire permettant l'amélioration des cartes de disparités estimées en corrigeant les faux appariements ou encore déduisant des valeurs de disparité pour les pixels occultés. Cette section présente quelques techniques de raffinement utilisées :

Contrôle de cohérence gauche-droite

La vérification de la cohérence gauche-droite [Egnal_02, Konolige_97, Little_90, Luo_95] est une technique couramment utilisée pour détecter les zones d'occultation sur les cartes de disparités. Dans le processus d'appariement, une image est choisie comme référence où pour chaque pixel nous recherchons son homologue dans l'autre image. Cette approche peut donner une carte de disparité estimée pour chaque image (gauche et droite). La première en basant sur l'image de gauche comme référence, et la seconde sur l'image de droite comme référence [Faugeras_93B].

Après le calcul des cartes de disparités gauche et droite, une vérification peut être appliquée ensuite pour déterminer les paires de pixels qui partagent les mêmes valeurs de

disparité dans ces deux cartes. Si les valeurs de disparité entre un pixel et son homologue estimé sont différentes, les deux pixels sont considérés comme incohérents. Par conséquent, Le contrôle de cohérence gauche-droite peut affirmer que deux pixels gauche et droite de coordonnées (x_l, y) et (x_r, y) sont homologues si la condition suivante est vérifiée:

$$d_l(x_l, y) = d_r(x_r, y) = d_r(x_l + d_l(x_l, y), y) \quad (1.25)$$

Si l'égalité (1.25) n'est pas vérifiée, alors le pixel est considéré appartenir à une zone d'occultation. Une solution simple pour associer une valeur de disparité à ces pixels est de les remplacer par la valeur de disparité significative des pixels voisins. Cette méthode suppose que les régions d'occultation partagent la même valeur de disparité avec les objets voisins. A partir de cette hypothèse, les pixels occultés sur la carte de disparité gauche peuvent recevoir leurs valeurs de leur pixel voisin non-occulté du côté gauche. De même, les valeurs des pixels occultés sur la carte de disparité droite seront remplacés par la valeur de disparité du pixel non-occulté de leur côté droit.

Propagation des disparités

L'idée générale de la propagation des disparités repose sur la contrainte de continuité des disparités. A partir de là, il y a de fortes chances que les pixels voisins d'un pixel, mis en correspondance de façon fiable, aient leurs correspondants au voisinage de ce pixel dans l'autre image. L'avantage de ces méthodes est qu'elles sont simple et moins coûteuses en temps de calcul mais sont très sensibles à la propagation des erreurs de départ.

Ces méthodes sont basées sur des techniques de croissance de régions à partir des pixels de départ. Le choix des disparités initiales à propager se fait par une sélection des pixels ayant de meilleurs scores de corrélation, c'est-à-dire, les couples de pixels homologues avec une plus grande fiabilité. Ensuite, la fonction de propagation commence par sélectionner les meilleurs couples afin de propager leurs valeurs aux pixels voisins. Le processus est répété jusqu'à ce qu'on obtienne une carte de disparité dense [Lhuillier_00, Lhuillier_02A]. Une technique d'interpolation linéaire ou encore de croissance de germes (*seeds*) [Lhuillier_02B] peuvent être utilisées.

Winner-take-all (WTA)

Winner-take-all (WTA) (ou gagnant prend tout) est une technique d'optimisation locale largement utilisée dans les algorithmes de mise en correspondance stéréoscopique. La WTA est souvent utilisée comme méthode de raffinement permettant de sélectionner la valeur de disparité redondante autour du pixel à apparier [Scharstein_02]. Ainsi, selon la contrainte de continuité des disparités, un pixel peut recevoir ou valider sa valeur de disparité à partir des valeurs des pixels voisins.

1.5 Conclusion

Dans ce chapitre, nous avons présenté les différents aspects des systèmes de vision stéréoscopique. Tout d'abord, nous avons introduit le modèle de caméra utilisée dans le processus de vision en se concentrant sur le processus d'acquisition des images stéréoscopiques à partir d'une paire de capteurs passifs. Puis, des explications géométriques d'un système stéréoscopique ont été présentées en mettant l'accent sur les relations entre les différentes composantes du système. Ensuite, le problème fondamental de la vision stéréoscopique, à savoir le problème de mise en correspondance, a été expliqué en décrivant un certain nombre de contraintes qui y sont liées, ainsi que les techniques les plus courantes visant à estimer la carte de disparité. Nous avons également démontré que la profondeur d'un objet ne dépend que de la valeur de sa disparité. Deux approches ont été distinguées: les approches locales, les approches globales. La contrainte du temps et de précision sont les deux paramètres qui peuvent classer les approches d'appariement stéréoscopiques. Par conséquent, le choix d'une approche ou d'une autre dépend essentiellement du système à développer. Finalement, nous avons donné un aperçu sur les différents systèmes de mise en correspondance temps-réel proposées dans la littérature dont la majorité sont basés sur des matériels spécifiques dédiés à accélérer les traitements des données en exploitant le parallélisme des algorithmes.

Malgré la diversité des méthodes d'appariement proposées, le besoin d'une méthode tenant compte à la fois de précision, robustesse et le temps de calcul, nous a motivé à réfléchir à une méthode rapide et robuste capable de donner de bons résultats sur un ordinateur PC simple. Nous proposons dans le chapitre suivant une nouvelle approche de mise en correspondance

stéréoscopique temps-réel, tirant profit simultanément des méthodes spatiales et temporelles. L'algorithme proposé tente ainsi d'assurer une bonne précision des résultats tout en diminuant le temps de traitement, un compromis au problème qui est toujours posé en stéréovision.

Chapitre 2

L'approche spatio-temporelle pour l'estimation de la disparité

Sommaire

2.1	Introduction	66
2.2	Algorithme proposé pour la mise en correspondance stéréoscopique	67
2.2.1	Les primitives utilisées : points de contours	68
2.2.2	L'association entre les points de contours des images successives	69
2.2.3	La fonction de coût.....	71
2.2.4	La mise en correspondance spatio-temporelle des contours	72
2.2.4.1	La mise en correspondance temporelle	73
2.2.4.2	La mise en correspondance spatiale	73
2.2.5	Carte de disparité initiale.....	76
2.2.6	Estimation de la carte de disparité finale	77
2.2.6.1	Gamme de disparités	77
2.2.6.2	Programmation dynamique	80
2.2.7	Récapitulatif : Pseudo-code de l'algorithme proposé	83
2.3	Conclusion.....	84

2.1 Introduction

Comme expliqué dans le chapitre précédent, la Stéréovision est une méthode bien connue pour obtenir une représentation précise [Bertozzi_00] et détaillée en 3D de l'environnement autour d'un véhicule intelligent (IV). Le problème clé de la vision stéréoscopique est de trouver la correspondance entre les pixels des images stéréoscopiques prises sous des angles de vue différents [Barnard_82]. Des études exhaustives sur des méthodes abordant le problème de mise en correspondance peuvent être trouvées dans [Brown_03, Dhond_89]. Une taxonomie des algorithmes de mise en correspondance stéréoscopique dense sont fournies par Scharstein et Szeliski [Scharstein_02]. Selon la taxonomie, les méthodes qui sont basées sur les coupures de graphe (Graph-cuts) [Boykov_01] semblent surpasser les autres méthodes. Toutefois, ces approches prennent beaucoup de temps ce qui les rend inadaptées pour les applications temps-réel, comme c'est le cas des systèmes avancés d'aide à la conduite (*Advanced Driving Assistance Systems ADAS*).

Bien qu'il y ait un fort soutien pour l'intégration de l'information temporelle dans le processus de mise en correspondance stéréoscopique qui permet l'obtention de meilleurs résultats [El-Ansari_08, Davis_05, Gong_06, Zhang_03], seule quelques travaux ont été consacrés à la reconstruction de scènes dynamiques à partir d'une séquence d'images stéréoscopiques. Nous croyons qu'en prenant en compte la cohérence temporelle entre les images consécutives, les résultats de l'appariement stéréoscopique peuvent être améliorés [El-Ansari_10A, El-Ansari_10B]. Ce chapitre présente notre approche pour la mise en correspondance stéréoscopique qui exploite la similarité entre les contours d'une paire d'images stéréoscopiques consécutives capturées par deux caméras montées à bord d'un véhicule intelligent. La méthode proposée présente une extension du travail développé dans [El-Ansari_10A]. En utilisant l'information temporelle, la question clé est : comment mettre en correspondance les images stéréoscopiques courantes en exploitant les résultats d'appariement calculés à partir des images stéréoscopiques précédentes? Pour répondre à cette question, nous devons d'abord penser à la façon dont nous allons déterminer un lien entre les paires d'images courantes et précédentes. Ici, nous utilisons l'approche détaillée dans [El-Ansari_10A].

Récemment, l'approche dite *association* a été présentée comme une technique pour trouver une relation entre les paires d'images stéréoscopiques successives [El-Ansari_10A]. En

utilisant la même technique, nous avons proposé une approche de mise en correspondance spatio-temporelle des contours extraits de deux paires d'images stéréoscopiques successives. Les points de contours appariés forment ensuite un ensemble de points de contrôle utilisés pour guider le processus d'appariement par programmation dynamique.

Plus précisément, une fois les contours significatifs sont sélectionnés à partir des paires d'images stéréoscopiques consécutives, la correspondance spatiale de ces contours des images précédentes est déduite des résultats d'appariement des points de contour calculés pour la même paire d'images. L'appariement temporel des contours des images gauches (resp. droites) consécutives est basé sur l'approche dite *association* [El-Ansari_10A]. Par conséquent, les correspondances spatiales entre les contours de la paire d'images courantes peuvent être facilement déduites. Les points appartenant aux contours dont l'appariement est connu, sont utilisés comme des points de contrôle pour conduire le processus de la programmation dynamique [Otha_89], ce qui permet de mettre en correspondance tous les points de contour de la paire d'images stéréoscopiques courante. La méthode proposée a été testée sur des séquences d'images de synthèse et réelles. Les résultats de l'approche proposée sont très satisfaisants en termes de temps de traitement et de qualité d'appariement. Dans ce qui suit, nous présentons la méthode proposée permettant la mise en correspondance spatio-temporelle.

2.2 Algorithme proposé pour la mise en correspondance stéréoscopique

Dans cette section, nous décrivons les étapes de la méthode proposée pour l'appariement des images stéréoscopiques prises par un capteur stéréoscopique monté à bord d'un véhicule intelligent. Il est à noter que toutes les images stéréoscopiques utilisées dans nos expériences sont rectifiées (c'est-à-dire les pixels homologues ayant la même ordonnée y). L'idée clé de l'approche proposée est d'exploiter le lien entre les paires d'images stéréoscopiques successives. Dans ce sens, les résultats de l'appariement de l'instant précédent sont utilisés pour améliorer ceux de l'instant courant. La notation suivante sera utilisée dans le reste de ce mémoire : I_{k-1}^L et I_{k-1}^R désignent respectivement les images stéréoscopiques gauche et droite de la trame f_{k-1} acquises à l'instant $(k - 1)$, et d_{k-1} la carte de disparité correspondante. De même, I_k^L et I_k^R représentent les images stéréoscopiques gauche et droite respectivement, de la

trame f_k acquises à l'instant k . Notons par la trame (frame) $f_k = (I_k^L, I_k^R)$ la paire d'images stéréoscopiques courante pour laquelle nous cherchons à calculer la carte de disparité d_k et par $f_{k-1} = (I_{k-1}^L, I_{k-1}^R)$ la paire d'images stéréoscopiques pour laquelle la carte de disparité d_{k-1} est disponible. Le problème de la mise en correspondance peut être formulé ainsi comme suit: le calcul de d_k se fait en tenant compte de la carte de disparité disponible d_{k-1} et la relation entre les images des deux trames (f_k et f_{k-1}). Une vue générale de l'algorithme proposé est fournie par la Figure 2.1.

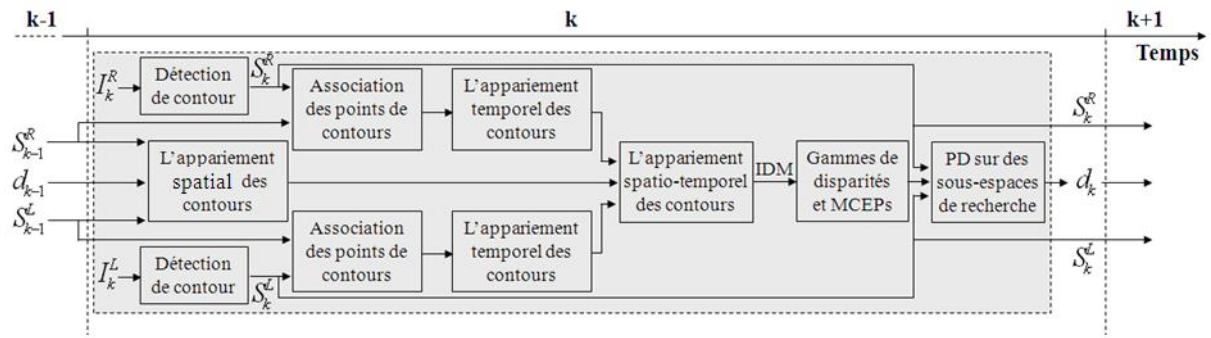


Figure 2.1– Vue d'ensemble de l'algorithme d'appariement spatio-temporelle proposé.

2.2.1 Les primitives utilisées : points de contours

La première étape de notre approche consiste à extraire dans des images stéréoscopiques les caractéristiques significatives à appairer. Ici, nous nous intéressons à la mise en correspondance des points de contours. Dans [El-Ansari_10A], l'opérateur déclivité [Miche_95] a été utilisé pour extraire les points de contour dans une séquence stéréoscopique. Cependant, cet opérateur n'est pas en mesure de détecter des contours horizontaux. Par conséquent, un nombre très important de points de contour est manqué, ce qui produit une carte de disparité moins riche en informations. Dans ce travail, nous utilisons pour la détection des contours de l'image l'opérateur de Canny [Canny_86]. Ce dernier a l'avantage de fournir des contours continus, qui sont indispensables à la méthode d'appariement proposée. L'utilisation du détecteur de Canny donne plus de points de contour, ce qui permet une reconstruction des informations 3D plus riches. La notation suivante est utilisée dans le reste du document: $S_f^m = \{C_f^{m,i}\}_{i=1, \dots, N_f^m}$ désigne l'ensemble des contours extraits de l'image I_f^m , où $f \in \{k-1, k\}$ représente l'indice de la trame, $m \in \{L, R\}$ indique l'indice de l'image

stéréoscopique (c'est-à-dire, L pour l'image de gauche et R pour l'image de droite) et N_f^m représente le nombre de contour dans l'image I_f^m .

2.2.2 L'association entre les points de contours des images successives

L'idée que nous avons eue à l'esprit est de savoir comment exploiter la disparité d_{k-1} calculée pour une trame f_{k-1} (la paire d'images stéréoscopique précédente) pour calculer la carte de disparité d_k de la trame courante f_k (la paire d'images stéréoscopique courante)? Pour répondre à cette question, il est nécessaire de trouver un lien entre les trames successives (f_{k-1} et f_k). En particulier, nous avons pensé à trouver des relations entre les points de contour appartenant aux mêmes lignes dans une paire d'images successives.

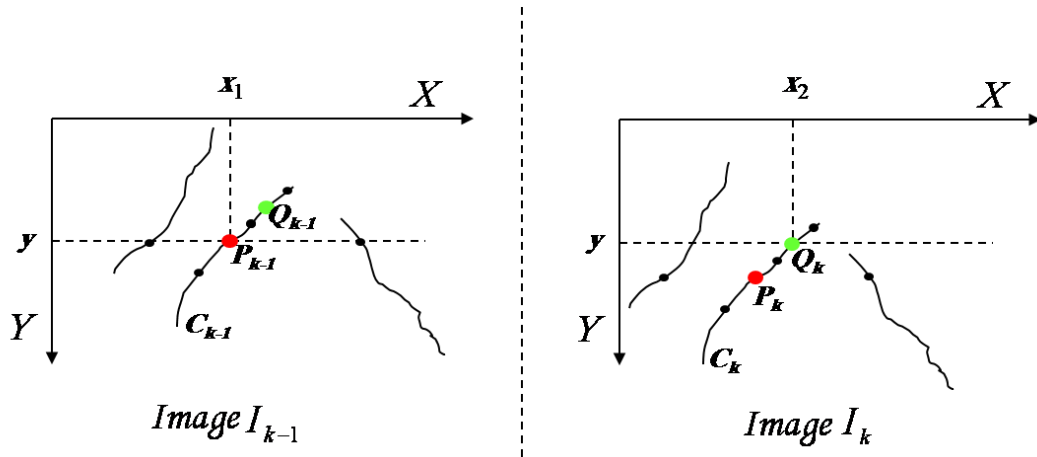


Figure 2.2 - I_{k-1}^L et I_k^L représentent les images successives d'une séquence d'images. Le point Q_k^L dans l'image I_k^L constitue le point associé du point P_{k-1}^L dans l'image I_{k-1}^L .

Soit I_{k-1}^L et I_k^L deux images gauches successives d'une séquence stéréoscopique. Notons par $Line_{k-1}^{y_0}$ et $Line_k^{y_0}$ deux lignes de I_{k-1}^L et I_k^L respectivement, ayant la même ordonnée y_0 dans les deux images. Supposons que les disparités au niveau des points de contour sur la ligne image $Line_{k-1}^{y_0}$ sont disponibles (calculées). Comment pouvons-nous donc utiliser ces disparités déjà calculées sur la ligne $Line_{k-1}^{y_0}$ à l'instant $(k-1)$ afin d'estimer les disparités des points de contour de la ligne $Line_k^{y_0}$? Pour y parvenir, nous avons besoin de trouver un

lien entre les points de $Line_{k-1}^{y0}$ et $Line_k^{y0}$. Pour cela, nous proposons de chercher les relations temporelles entre les points de contour d'une paire d'images successives prises par la même caméra. La technique dite *association*, utilisée pour la réalisation de cette tâche, a été proposée dans [El-Ansari_10A]. Dans cette section, nous passons en revue la méthode utilisée pour trouver l'association entre les points de contour d'une paire d'images successives (c'est-à-dire, l'association entre les points de contour des images I_{k-1}^L et I_k^L (resp. I_{k-1}^R et I_k^R)).

Supposons que nous voulons trouver l'association entre les points de contour d'une paire d'images gauches d'une séquence stéréoscopique. Pour cela, prenons deux points P_{k-1}^L et Q_{k-1}^L appartenant à un contour $C_{k-1}^{L,i}$ de I_{k-1}^L , tels que P_k^L et Q_k^L sont respectivement leurs points correspondants appartenant au contour $C_k^{L,i}$ de l'image I_k^L (voir la Figure 2.2). Le point associé au point P_{k-1}^L est défini comme un point de contour $C_k^{L,i}$ ayant la même ordonnée y que P_{k-1}^L [El-Ansari_10A, El-Ansari_10B]. D'après la Figure 2.2, nous pouvons noter que le point Q_k^L représente le point associé à P_{k-1}^L (Notez que l'association ne signifie pas la correspondance). Deux points associés sont deux points appartenant à deux contours correspondants de deux images successives de la même séquence, et ayant la même ordonnée y . Dans la pratique, due à la vitesse de prise de vue très élevée, nous supposons que le mouvement des objets d'une image à l'autre est très faible. Par conséquent, si x_1 et x_2 représentent les abscisses de P_{k-1}^L et Q_k^L , respectivement. x_2 devrait appartenir à l'intervalle $[x_1 - \Delta x, x_1 + \Delta x]$, où Δx est une constante fixée pour définir l'intervalle dans lequel les points associés sont sélectionnés, ce qui permet de contrôler le nombre de points candidats à associer. Le processus de l'association utilise la fonction de coût (équation (2.1)) sur l'intervalle limité par Δx afin de sélectionner le candidat associé à x_1 . Comme on le voit sur la Figure 2.2, le point P_k^L représente le correspondant du point P_{k-1}^L . Cependant, le point Q_k^L constitue l'associé du point P_{k-1}^L . On remarque que les points P_k^L et Q_k^L sont différents en raison du mouvement du point P_{k-1}^L dans l'image I_k^L .

Il est évident que nous ne pouvons pas trouver l'association pour tous les points de contour en raison du changement de point de vue et du mouvement des objets de la scène. Cela ressemble aux algorithmes de mise en correspondance où une partie est visible dans une image et occultée dans l'autre. La mise en correspondance temporelle de deux contours dépend du nombre de points associés qu'ils partagent : Plus le nombre de points associés est important,

plus les résultats d'appariement sont précis, et plus la carte de disparité pré-estimé (ou initiale) est dense (section 2.2.5), ce qui aide à obtenir des gammes de disparités plus précises au niveau de chaque ligne image. Le nombre de point associés trouvés dépend de deux paramètres:

- La valeur de Δx qui détermine la zone 1-D de recherche dans laquelle les points de contours à associer seront sélectionnés. Le choix de Δx dépend du type de la scène à traiter et le nombre d'images acquises par seconde. Si le mouvement des objets de la scène est important et la fréquence d'acquisition des images est petite, alors, nous devrions attribuer une grande valeur à Δx . Toutefois, Δx peut prendre une petite valeur lorsque le mouvement des objets est faible et la fréquence d'acquisition des images est grande (dans notre travail la valeur de Δx est fixée à 16). Ceci a une influence sur le temps d'exécution globale de l'algorithme.
- Les points associés doivent appartenir au même contour d'un objet avec un coût d'appariement (équation (2.1)) très petit. La fonction de coût (équation (2.1)) utilisée est basée sur l'amplitude des gradients, l'orientation ainsi que la valeur de niveau de gris des points de contour à associer. Nous ne retenons que les paires de points dont le coût est minimal et inférieure à un seuil T . Ainsi, le nombre de couples de points de contour associés dépendent de la valeur de T . Une valeur très grande de T fournit plus de points associés avec un nombre d'associations erronées. La plus petite valeur de T donne des associations très précises, mais aussi le nombre de points associés très réduit. Ce qui explique le rapport existant entre la précision et le nombre d'association. Dans notre travail, la technique Winner-Take-All (WTA) est utilisée pour réduire le nombre d'associations erronées. La WTA est appliquée sur l'ensemble des points appartenant aux paires de contours partageant le maximum de points associés (section 2.2.4.1).

2.2.3 La fonction de coût

Comme critère de similarité entre les points de contour correspondants (ou associés), nous utilisons une fonction de coût C (équation (2.1)) basée sur l'amplitude des gradients, l'orientation ainsi que les niveaux de gris des points de contour à apparier (ou à associer). Soit e_L et e_R deux points de contour de I_k^L et I_k^R , respectivement. Nous désignons par m_L et m_R

(resp. θ_L et θ_R) les amplitudes de gradient (resp. orientations) des points e_L et e_R , respectivement. Il est évident que les points homologues dans une paire d'images doivent avoir les mêmes amplitudes de gradient ou proche, ainsi que les mêmes valeurs de niveau de gris et orientations ou presque. Par conséquent, nous définissons la fonction de coût comme suit:

$$C(e_L, e_R) = \sqrt{\left(I_k^L(x_L, y_L) - I_k^R(x_R, y_R)\right)^2 + (m_L)^2 + (m_R)^2 - 2 * m_L * m_R * \cos(\theta_L - \theta_R)} \quad (2.1)$$

Où (x_L, y_L) et (x_R, y_R) sont les coordonnées des points de contour e_L et e_R , respectivement. Les amplitudes de gradient et les orientations de e_L et e_R sont définies comme suit :

$$m_L = \sqrt{\left(I_k^L(x_L + 1, y_L) - I_k^L(x_L - 1, y_L)\right)^2 + \left(I_k^L(x_L, y_L - 1) - I_k^L(x_L, y_L + 1)\right)^2} \quad (2.2)$$

$$m_R = \sqrt{\left(I_k^R(x_R + 1, y_R) - I_k^R(x_R - 1, y_R)\right)^2 + \left(I_k^R(x_R, y_R - 1) - I_k^R(x_R, y_R + 1)\right)^2} \quad (2.3)$$

$$\theta_L = \tan^{-1} \left(\frac{I_k^L(x_L, y_L - 1) - I_k^L(x_L, y_L + 1)}{I_k^L(x_L + 1, y_L) - I_k^L(x_L - 1, y_L)} \right) \quad (2.4)$$

$$\theta_R = \tan^{-1} \left(\frac{I_k^R(x_R, y_R - 1) - I_k^R(x_R, y_R + 1)}{I_k^R(x_R + 1, y_R) - I_k^R(x_R - 1, y_R)} \right) \quad (2.5)$$

La même fonction de coût (équation (2.1)) est utilisée pour calculer la similarité entre les points de contour associés pour une paire d'images stéréoscopiques successives.

2.2.4 La mise en correspondance spatio-temporelle des contours

Dans cette section, nous illustrons la procédure pour déterminer les contours correspondants dans deux trames successives f_k et f_{k-1} en utilisant l'approche d'association (section 2.2.2) et en exploitant la carte de disparité d_{k-1} déjà calculée à l'instant $(k - 1)$ pour la trame f_{k-1} .

2.2.4.1 La mise en correspondance temporelle

Il s'agit de l'appariement des contours des images I_{k-1}^L et I_k^L (resp. I_{k-1}^R et I_k^R). Dans ce qui suit, nous illustrons l'approche de la mise en correspondance temporelle proposée en utilisant juste les images I_{k-1}^L et I_k^L . Le même processus est utilisé pour faire l'appariement des contours de I_{k-1}^R et I_k^R .

La mise en correspondance temporelle consiste à trouver pour chaque contour $C_{k-1}^{L,j}$ dans l'ensemble S_{k-1}^L un contour correspondant $C_k^{L,i}$ dans S_k^L , si elle existe. Désignons par $Ass(C_{k-1}^{L,j}) = \{ae_n\}_{n=1, \dots, N_i}$ l'ensemble des points de contour ae_n appartenant à l'image I_k^L représentant les points associés aux points de $C_{k-1}^{L,j}$, avec N_i le nombre d'associations (points associés) trouvé pour le contour $C_{k-1}^{L,j}$. Si M_j représente le nombre de points du contour $C_{k-1}^{L,j}$, alors $N_i \leq M_j$, car il se peut qu'il y a des points de contour dans l'image I_{k-1}^L qui n'ont pas de points associés dans l'image I_k^L . Dans le cas idéal où les résultats sont parfaits pendant le processus d'association, tous les points de contours de l'ensemble $Ass(C_{k-1}^{L,j})$ doivent trouver leurs associations sur le même contour $C_k^{L,i}$, qui est le contour correspondant. Malheureusement, il peut y avoir quelques fausses associations dues au mouvement des objets dans la scène, changement de point de vue d'un instant à l'autre, etc. Par conséquent, les points de contour ae_n peuvent appartenir à différentes contours de S_k^L . Nous calculons ainsi le correspondant de $C_{k-1}^{L,j}$ par la recherche de d'un contour $C_k^{L,i}$ ayant le maximum de points dans $Ass(C_{k-1}^{L,j})$. La même méthode est appliquée pour faire la mise en correspondance de tous les contours de S_{k-1}^L afin de trouver leurs correspondants dans S_k^L .

2.2.4.2 La mise en correspondance spatiale

Cette étape consiste à faire correspondre les contours des images stéréoscopiques I_{k-1}^L et I_{k-1}^R en utilisant la carte de disparité d_{k-1} . Le même principe adopté pour l'établissement de la correspondance temporelle est utilisé pour calculer la correspondance spatiale.

Soit $Match(C_{k-1}^{L,j}) = \{me_n\}_{n=1, \dots, N_m}$ l'ensemble des points de contour de l'image I_{k-1}^R correspondant aux points de $C_{k-1}^{L,j}$, N_m le nombre de points de contour correspondant aux

points de $C_{k-1}^{L,j}$. Si M_j représente le nombre de points de contour composant $C_{k-1}^{L,j}$, alors $N_m \leq M_j$, cela revient au fait qu'un certain nombre de points de contour dans l'image I_{k-1}^L n'ayant pas de correspondants dans l'image I_{k-1}^R (à cause des problèmes de la mise en correspondance stéréoscopique). Dans le cas idéal où le processus d'appariement est parfait, tous les points de contour appartenant à l'ensemble $Match(C_{k-1}^{L,j})$ doivent appartenir au même contour homologue de $C_{k-1}^{L,j}$. Malheureusement, plusieurs problèmes liés à la structure stéréoscopique amène aux faux appariements. Par conséquent, les points de contour me_n peuvent appartenir au différents contour de S_{k-1}^R . Le contour $C_{k-1}^{L,j}$ a donc comme correspondant un contour $C_{k-1}^{R,m}$ ayant le maximum de points dans $Match(C_{k-1}^{L,i})$.

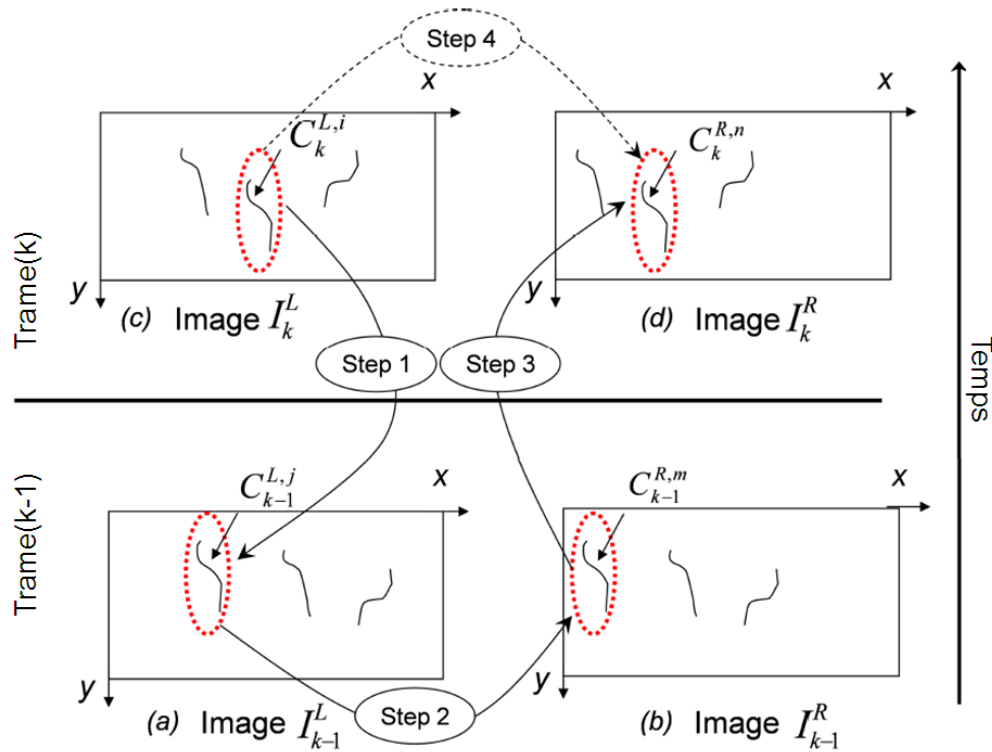


Figure 2.3- La mise en correspondance spatio-temporelle des contours appartenant à deux paires d'images stéréoscopiques successives : (a) L'image gauche de la trame f_{k-1} , (b) l'image droite de la trame f_{k-1} , (c) l'image gauche de la trame f_k , (d) l'image droite de la trame f_k . Les contours $C_{k-1}^{L,j}$ et $C_{k-1}^{R,m}$ représentent respectivement les correspondants temporels de $C_k^{L,i}$ et $C_k^{L,n}$, avec $C_{k-1}^{R,m}$ le correspondant spatial de $C_{k-1}^{L,j}$. Le contour $C_k^{L,n}$ est déduit comme étant le correspondant spatial de $C_k^{L,i}$.

À ce stade, trois étapes de mise en correspondance des contours sont établies, une spatiale pour les images stéréoscopiques (I_{k-1}^L et I_{k-1}^R) (étape 2 de la figure 2.3), et deux autres temporelles pour les images (I_{k-1}^L et I_k^L) (étape 1 de la figure 2.3) et les images (I_{k-1}^R et I_k^R) (étape 3 de la figure 2.3). Compte tenu de ces informations, nous pouvons en déduire facilement la correspondance spatiale des contours de I_k^L et I_k^R , comme le montre l'étape 4 de la figure 2.3. Autrement dit, soit $C_k^{L,i}$ un contour dans I_k^L . L'appariement spatiale de $C_k^{L,i}$ se fait en quatre étapes (figure 2.3). Tout d'abord, nous cherchons dans I_{k-1}^L un contour correspondant de $C_k^{L,i}$ que nous appelons $C_{k-1}^{L,j}$. Deuxièmement, le contour homologue $C_{k-1}^{R,m}$ de $C_{k-1}^{L,j}$ est trouvé dans l'image I_{k-1}^R . Troisièmement, nous cherchons le correspondant temporelle $C_k^{R,n}$ de $C_{k-1}^{R,m}$ dans l'image droite I_k^R . Enfin, nous pouvons déduire que $C_k^{R,n}$ représente le correspondant de $C_k^{L,i}$. Nous répétons le même processus pour tous les contours de l'image I_k^L afin de trouver leurs correspondants dans l'image I_k^R , s'ils existent.

Algorithme 2.1 : La mise en correspondance spatio-temporelle des contours de la paire d'images stéréoscopiques I_k^L et I_k^R .

```

Pour  $i=1$  à  $\text{sizeOf}(S_k^L)$  faire
  Si  $\text{TemporalMatchOf}(C_k^{L,i})$  existe alors
     $C_{k-1}^{L,j} = \text{TemporalMatchOf}(C_k^{L,i})$ 
    Si  $\text{SpacialMatchOf}(C_{k-1}^{L,j})$  existe alors
       $C_{k-1}^{R,m} = \text{SpacialMatchOf}(C_{k-1}^{L,j})$ 
      Si  $\text{TemporalMatchOf}(C_{k-1}^{R,m})$  existe alors
         $C_k^{R,n} = \text{TemporalMatchOf}(C_{k-1}^{R,m})$ 
         $\text{SpacialMatchOf}(C_k^{L,i}) = C_k^{R,n}$ 
      finSi
    finSi
  finSi
finPour

```

Le pseudo-code de l'algorithme (2.1) résume les principales étapes du processus de la mise en correspondance spatio-temporelle permettant l'appariement des contours d'une paire d'images stéréoscopiques I_k^L et I_k^R . Le processus prend l'image de gauche comme image de référence. Cela signifie que pour chaque contour de l'ensemble S_k^L de I_k^L , nous cherchons un correspondant dans l'ensemble des contours S_k^R de l'image I_k^R .

2.2.5 Carte de disparité initiale

Ici, nous illustrons comment obtenir une carte de disparité initiale pour une paire d'images stéréoscopique de la trame f_k à partir des contours correspondants déduits par l'approche de l'appariement spatio-temporel décrite dans (section 2.2.4). Soit $C_k^{L,i}$ et $C_k^{R,n}$ deux contours homologues appartenant aux images I_k^L et I_k^R respectivement (Figure 2.3), et e_m^L un point de contour appartenant à $C_k^{L,i}$. Étant données que les images stéréoscopiques sont rectifiées. Le correspondant du point e_m^L , s'il existe, doit appartenir à $C_k^{R,n}$ et ayant la même ordonnée y que e_m^L . Nous répétons ce processus pour tous les points de $C_k^{L,i}$ afin de trouver leurs correspondants sur $C_k^{R,n}$. La même méthode est appliquée sur toutes les paires de contours appariées dans la trame f_k . Ainsi, les paires de points homologues obtenus seront utilisés pour générer une carte de disparité initiale (IDM) pour la trame courante f_k .

L'IDM utilisée dans notre travail est beaucoup plus précise par rapport à la carte de disparité pré-estimée calculée dans [El-Ansari_10A]. Car dans notre travail, l'IDM est calculée à partir d'un appariement temporel des contours (section 2.2.4.1). L'utilisation des contours nous a permis d'éviter les problèmes inhérents à la technique d'association, et donc d'améliorer la qualité de l'IDM. Cependant, dans [El-Ansari_10A], l'IDM a été calculée directement à partir des associations relevées entre les points de contour des images consécutives. Dans ce cas, les fausses associations affectent directement la qualité de l'IDM obtenue. Avoir une bonne IDM est donc cruciale pour le succès de l'approche proposée et celle de [El-Ansari_10A]. Dans les deux approches, l'IDM est utilisée afin d'estimer une gamme de disparités pour chaque ligne image. De plus, dans notre approche, nous avons utilisé l'IDM afin d'obtenir les *points de contour pour le contrôle d'appariement* (MCEPs) permettant de guider le processus de la programmation dynamique pour faire correspondre les points de contour restant dans la trame courante (section 2.2.6.2).

L'algorithme (2.2) illustre l'implémentation permettant la génération de la carte de disparité initiale (IDM) : pour chaque contour $C_k^{L,i}$ de l'image I_k^L , nous cherchons les points e_m^L du contour $C_k^{L,i}$ ayant un correspondant e_n^R sur le contour homologue $C_k^{R,n}$. Puisque les images utilisées sont considérées rectifiées, alors le point homologue e_n^R de e_m^L se trouve sur la

même ligne image. Une carte de disparité initiale notée pd_k est calculée à partir de toutes les paires de points appartenant aux contours homologues.

Algorithme 2.2 : Estimation de la carte de disparité initiale (IDM)

```

Pour  $i=0$  à  $sizeOf(S_k^L)$  faire
     $C_k^{R,n} = SpacialMatchOf(C_k^{L,i})$ 
    Pour tous les points  $e_m^L$  de  $C_k^{L,i}$  faire
         $e_n^R = MatchOf(e_m^L)$ 
        Si  $e_n^R$  existe dans  $C_k^{R,n}$  then
             $pd_k(e_m^L) = e_n^R$ 
        FinSi
    FinPour
FinPour

```

2.2.6 Estimation de la carte de disparité finale

Dans la section 2.2.4, nous avons décrit comment faire correspondre les points appartenant aux contours d'une paire d'images stéréoscopique de la trame f_k . Afin de compléter la mise en correspondance de tous les points de contour de la trame f_k , nous présentons dans cette section notre méthode d'appariement basée sur la programmation dynamique en tenant compte de l'IDM obtenue. Les points dont l'appariement est déjà calculé dans l'IDM seront utilisés comme des points de contrôle pendant le processus d'appariement de la carte de disparité finale.

2.2.6.1 Gamme de disparités

Dans cette section nous détaillons la méthode utilisée pour calculer les gammes de disparités de la trame f_k . Il est clair que les gammes de disparités représentent un outil primordial pour l'obtention des disparités précises. Plus la gamme est petite, plus les résultats d'appariement sont précis. Nous avons proposé une nouvelle technique permettant de calculer les gammes de disparités, non seulement pour chaque ligne, mais également entre les points d'une même ligne basant sur les valeurs de l'IDM. Cela veut dire que l'appariement d'un point de l'image sera contrôlé par les valeurs des deux points (gauche et droit) de l'IDM respectivement voisins. Cette technique est expliquée en détail dans la section suivante.

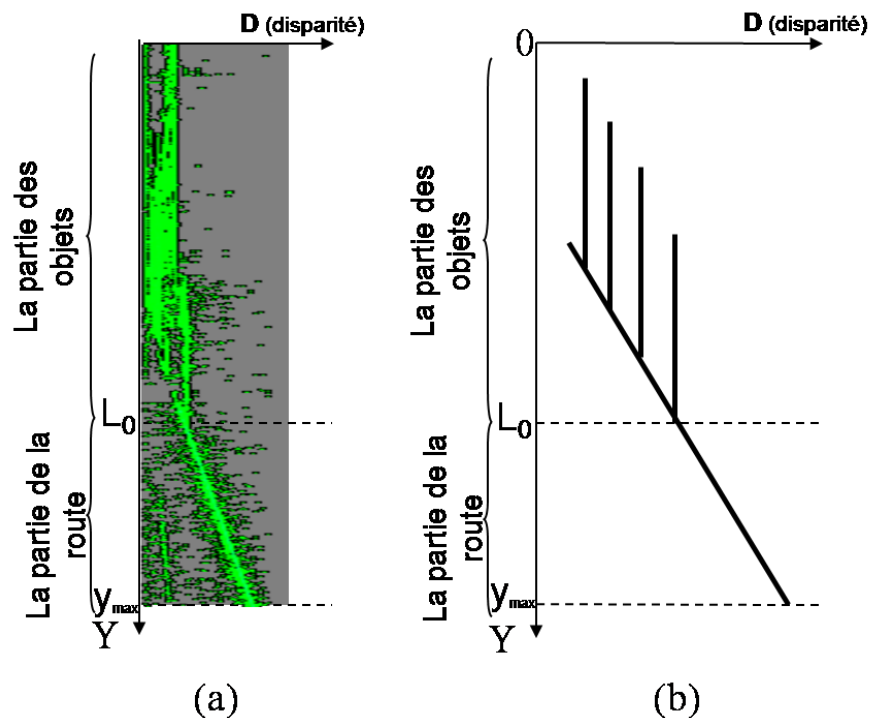


Figure 2.4 – La v-disparité, (a) exemple d’image de v-disparité calculée à partir d’une carte de disparité, (b) une représentation graphique d’une v-disparité contenant quatre objets (segments verticaux) et le profil de la route (droite oblique).

Désignons par pd_k la carte de disparité pré-estimée (ou initiale IDM) déjà calculée pour la trame f_k (section 2.2.4). H une fonction appliquée sur l’ensemble des points de pd_k , telle que $H(pd_k) = vpd_k$, où vpd_k est appelée l’image de v-disparité (ou tout simplement « v-disparité ») [Labayrade_02]. La fonction H cumule les points ayant les mêmes valeurs de disparité pour chaque ligne de l’IDM. La façon de construire une v-disparité est détaillée dans [Labayrade_02]. La v-disparité constitue un moyen simple permettant d’avoir une bonne approximation de la surface de la route. Il a été démontré dans [Labayrade_02] que les obstacles et le profil de la route peuvent être représentés par des lignes verticales et obliques respectivement. Dans une scène routière contenant quatre objets, la v-disparité correspondante doit être comme illustré dans la Figure 2.4 (b). Notons que nous avons quatre lignes verticales représentant les quatre obstacles, et une ligne oblique représentant la surface de la route. Pour le calcul de la gamme de disparités, nous avons divisé la v-disparité en deux parties: la partie supérieure contenant les objets et une partie inférieure contenant uniquement le profil de la route. Les deux parties sont séparées par la ligne $y = L_0$. Nous proposons de trouver la gamme de disparités indépendamment pour chacune des deux parties.

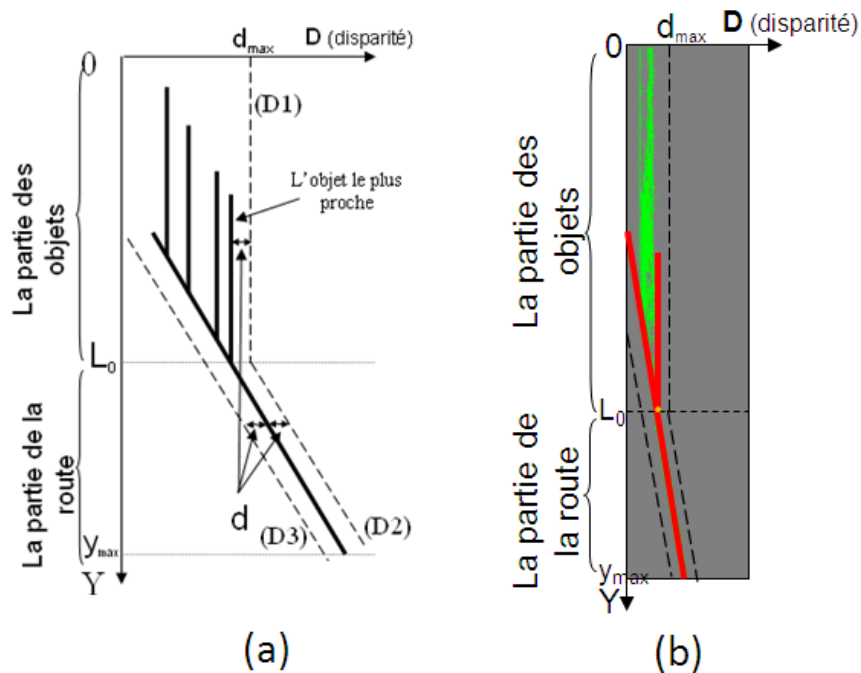


Figure 2.5 – Les grammes de disparité dans une v-disparité : (a) L'axe vertical réfère aux ordonnées des lignes image et l'axe horizontal représente les disparités. y_{max} désigne la hauteur de l'image. d_{max} la valeur de disparité maximale pour la partie des objets. Les disparités éventuelles doivent appartenir à la zone située entre les lignes (D1) et (D3) pour la partie supérieure et entre les lignes (D2) et (D3) pour la partie inférieure. (b) l'image v-disparité extraite de la carte de disparité initiale (IDM), les deux segments oblique et vertical (en rouge) représentent respectivement le profil de la route et l'objet le plus proche.

Dans la partie inférieure (Figure 2.4 et Figure 2.5), la surface de la route est représentée par une ligne oblique. Dans le cas idéal une seule valeur de disparité est possible pour chaque ligne en formant ainsi une droite oblique. La valeur de disparité pour la ligne d'ordonnée y_i s'écrit de la forme $a * y_i + b$, où a et b sont les paramètres de l'équation de la droite représentant la route. Afin de tenir compte de l'incertitude inhérente au calcul, les disparités possibles sur la ligne $\{y = y_i\}$ doivent appartenir à l'intervalle $[a * y_i + b - d \text{ et } a * y_i + b + d]$, où d est la gamme de disparités sur la ligne y_i . Dans notre travail la valeur de d est fixée arbitrairement à 3. Les gammes de disparités possibles pour la partie inférieure (Figure 2.5) sont représentées par une zone située entre deux droites (D2) et (D3), et dans la partie supérieure par (D1) et (D3). Autrement dit, la disparité maximale au niveau de la ligne L_0

constitue la disparité maximale pour toutes les lignes de la partie supérieure (la partie des objets) dont la disparité maximale et minimale peut s'écrire:

$$d_{max} = a * L_0 + b + d \quad (2.6)$$

$$d_{min} = \text{Max}(0, a * y_i + b - d) \quad (2.7)$$

Le gamme de disparités pour toute ligne y_i de l'image peut être s'exprimée comme suit :

$$\begin{cases} [a * y_i + b - d, a * y_i + b + d] & \text{si } y_i > L_0 \\ [\text{Max}(0, a * y_i + b - d), a * L_0 + b + d] & \text{si } y_i \leq L_0 \end{cases} \quad (2.8)$$

2.2.6.2 Programmation dynamique

Soit $\{e_{i,sl}^L\}_{i=1,\dots,N_{sl}^L}$ (resp. $\{e_{i,sl}^R\}_{i=1,\dots,N_{sl}^R}$) l'ensemble des points de contour de la ligne sl de l'image I_k^L (resp. I_k^R). Nous supposons que ces points sont ordonnés selon leurs abscisses. N_{sl}^L (resp. N_{sl}^R) est le nombre de points de contour sur la ligne sl . Nous allons montrer comment faire correspondre ces points de contour en utilisant la programmation dynamique. La même technique est utilisée pour toutes les lignes de la paire d'images stéréoscopiques I_k^L et I_k^R .

La programmation dynamique est l'une des techniques efficaces pour résoudre le problème de la mise en correspondance stéréoscopique. Elle permet d'apparier les points de contour d'une paire de ligne épipolaires conjuguées. Cette approche repose sur la recherche d'un chemin optimal dans un graphe 2D qui minimise une fonction de coût, tout en utilisant la contrainte d'ordre des primitives [Baker_81, Benard_84, Otha_89]. Nous proposons de subdiviser l'espace de recherche en plusieurs sous-espaces, selon le nombre de MCEPs trouvés sur la ligne sl de l'IDM.

La figure 2.6 illustre un exemple de l'espace de recherche 2D subdivisé en plusieurs sous-espaces de recherche. Les lignes verticales indiquent les positions des points de contour sur la ligne épipolaire (ou ligne image) gauche, tandis que les lignes horizontales correspondent à ceux de la ligne épipolaire droite. Les intersections de ces lignes sont appelées « les nœuds » qui représentent les candidats d'appariement. Chaque nœud correspond à une étape de la programmation dynamique, où une décision doit être prise afin de sélectionner un chemin optimal vers ce nœud. Le résultat d'appariement final est obtenu en cherchant un chemin qui

correspond à une minimisation globale de la fonction de coût. À l'intérieur de chaque sous-espace de recherche, le chemin doit aller du coin supérieur gauche au coin inférieur droit (en raison de la contrainte de l'ordre).

Comme représenté sur la Figure 2.6, chaque sous-espace de recherche consiste à appairer les points de contour entre les MCEPs consécutives. À titre d'exemple, le sous-espace de recherche (2) est utilisé pour faire correspondre les points de contour entre e_1^L et e_2^L de la ligne épipolaire gauche et ceux situés entre e_1^R et e_2^R de la ligne épipolaire droite. Les nœuds (e_1^L, e_2^L) et (e_1^R, e_2^R) représentent deux paires de points de contour déjà appariés par l'approche de l'appariement spatio-temporel décrite à la section 2.2.4. Ils constituent les deux points de contrôles MCEPs déterminant le sous-espace de recherche (2).

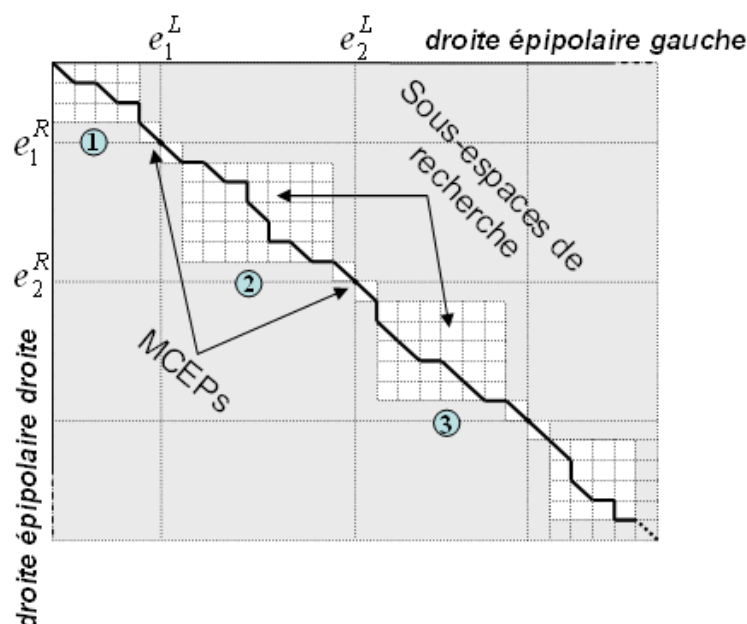


Figure 2.6– L'espace de recherche bidimensionnel subdivisé en plusieurs sous-espaces selon le nombre des points de contrôle MCEPs. L'axe horizontal représente la droite épipolaire gauche et l'axe vertical représente la ligne épipolaire droite correspondante. Les lignes verticales et horizontales correspondent aux positions de points de contour sur la ligne de l'image, et leur intersection représente l'appariement possible.

Tout d'abord, la gamme de disparités permet de sélectionner les nœuds validés dans un sous-espace de recherche. Deuxièmement, la fonction de coût (équation (2.1)) est utilisée ensuite pour calculer la similarité entre toutes les paires de points possible à appairer. Le chemin optimal trouvé nous permettra de déterminer les paires de points de contour entre e_1^L et e_2^L de

la ligne épipolaire gauche correspondants à ceux entre e_1^R et e_2^R de la ligne épipolaire droite. Le même processus est répété pour tous les sous-espaces de recherche et pour toutes les autres lignes de l'image en effectuant ainsi l'appariement de tous les points de contour de l'image.

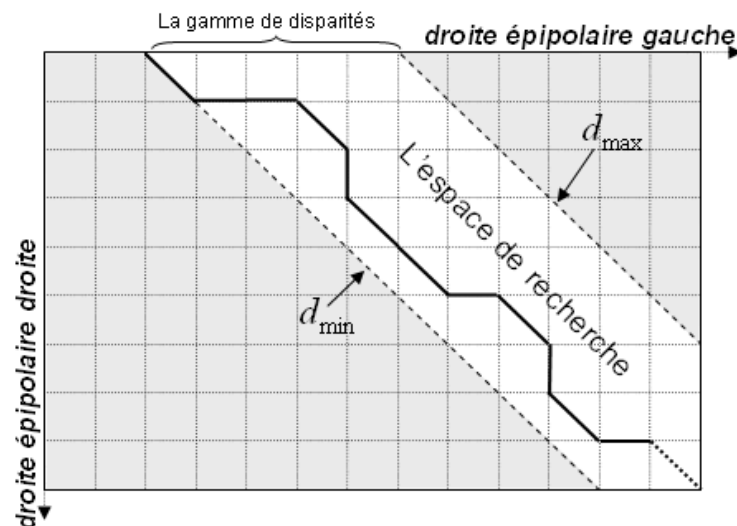


Figure 2.7 – L'espace de recherche des appariements par programmation dynamique des points d'une paire de lignes conjuguées. La recherche des appariements s'effectue à l'intérieur de la zone blanche définie par la gamme de disparité (d_{min} et d_{max}). Le chemin dans cette zone représente la solution optimale pour la mise en correspondance des points de deux lignes épipolaires. Les zones en gris sont des régions interdites dont les nœuds ne sont pas pris en considération par le processus d'appariement.

Algorithme 2.3 : *L'appariement par la programmation dynamique dans les sous-espaces de recherche*

MCEP{sl} : l'ensemble des points de contour pour le contrôle de l'appariement sur la ligne *sl*.

Pour *sl* = 1 **à** HeightOf(I_k^L) **faire**

Pour *i*=1 **à** sizeOf(MCEP{*sl*}) – 1 **faire**

$e_i = \text{MCEP}\{sl\}(i)$

$e_{i+1} = \text{MCEP}\{sl\}(i + 1)$

 - Application de la programmation dynamique sur les points de contour appartenant au sous-espace de recherche défini par e_i et e_{i+1} .

FinPour

FinPour

2.2.7 Récapitulatif : Pseudo-code de l'algorithme proposé

Cette section présente les différentes étapes de notre approche pour l'appariement des points de contour d'une paire d'images d'une séquence stéréoscopique. L'algorithme commence par une étape d'initialisation dans laquelle une carte de disparité est calculée en utilisant une méthode spatiale simple de mise en correspondance. Ensuite, l'algorithme répète les étapes de (2) à (11) pour chaque paire d'images stéréoscopiques de la séquence.

Algorithme 2.4 : *Les différentes étapes de l'approche proposée*

- **Etape 1 :** *initialisation de l'algorithme*
 - **Etape 1.1 :** *Détection de contours de I_0^L et I_0^R en utilisant l'opérateur de Canny*
 - **Etape 1.2 :** *Calcul de la carte de disparité d_0 par une méthode simple*
 - **Etape 1.3 :** $k=1$
- **Etape 2 :** *Détection de contours de I_k^L et I_k^R en utilisant l'opérateur de Canny*
- **Etape 3 :** *la mise en correspondance spatiale des contours d'une paire d'images stéréos I_{k-1}^L et I_{k-1}^R , pour chaque contour $C_{k-1}^{L,i}$ de S_{k-1}^L , nous recherchons son correspondant dans S_{k-1}^R .*
- **Etape 4 :** *la mise en correspondance temporelle des contours d'une paire d'images I_{k-1}^L et I_k^L successives.*
 - **Etape 4.1 :** *l'association entre les points de contour en utilisant la méthode proposée dans [El-Ansari_10A]. Pour chaque point de contour de l'image I_k^L nous cherchons son point associé dans l'image I_{k-1}^L .*
 - **Etape 4.2 :** *Pour chaque contour $C_k^{L,i}$ de S_k^L , nous cherchons son correspondant dans S_{k-1}^L .*
- **Etape 5 :** *Les mêmes étapes de (Etape 4) sont répétées pour calculer la mise en correspondance temporelle entre les contours d'une paire d'images I_{k-1}^R et I_k^R successives.*
- **Etape 6 :** *La mise en correspondance spatio-temporelle des contours de la paire d'images stéréos I_k^L et I_k^R (Algorithme 2.1).*
- **Etape 7 :** *Estimation de la carte de disparité initiale (IDM) (Algorithme 2.2).*
- **Etape 8 :** *Calcul de la gamme de disparités*
La même technique proposée dans [El-Ansari_10A] est utilisée pour déterminer la gamme de disparités possibles pour chaque ligne de $f_k = \{I_k^L, I_k^R\}$ en analysant l'image de v-disparité vpd_k calculée à partir de la carte de disparité initiale pd_k .
- **Etape 9 :** *Programmation dynamique appliquée sur les sous-espaces de recherche (Algorithme 2.3).*

- **Etape 10 :** Estimation de la carte de disparité finale d_k à partir des paires de points de contour composant le chemin optimal calculé par la programmation dynamique.
- **Etape 11 :** $k = k + 1$
- **Etape 12 :** Répéter les étapes (2) à (11) pour la paire d'image suivante.

2.3 Conclusion

Nous avons présenté dans ce chapitre une nouvelle approche d'appariement stéréoscopique combinant à la fois les avantages des méthodes spatiales et temporelles. La qualité de l'appariement et le temps de traitement sont les deux critères pris en compte lors de la conception de cette approche. Notre approche consiste à exploiter les résultats de l'appariement d'une paire d'images stéréoscopiques précédentes afin d'améliorer les résultats d'appariement de la trame courante. La mise en correspondance temporelle entre les contours d'une paire d'images gauches (resp. droites) consécutives s'effectue en utilisant l'approche de l'association proposée dans [El-Ansari_10A] qui nous a permis d'apparier les contours d'une paire d'images successives. L'algorithme de mise en correspondance spatio-temporelle permet de déduire à partir de la carte de disparité précédente l'appariement de quelques contours de la trame courante, ce qui nous permettra ensuite de générer une carte de disparité initiale (IDM). La carte de disparité IDM ainsi obtenu permet d'améliorer les résultats de l'appariement courant de la façon suivante :

- Extraction des gammes de disparités correspondantes pour chaque ligne image, et cela à travers une image v-disparité pré-estimé calculée à partir des informations fournies par l'IDM.
- Les points appariés dans l'IDM sont utilisés comme des points de contrôle permettant de guider le processus de la recherche des appariements par programmation dynamique.

L'approche proposée permet de traiter itérativement les images d'une séquence stéréoscopique. A chaque itération, une carte de disparité est calculée en utilisant la carte de disparité de l'itération précédente. Ainsi, de la même façon, la carte de disparité obtenue sera utilisée pour la prochaine itération, et ainsi de suite. Le chapitre suivant présente les résultats

expérimentaux obtenus avec la méthode proposée dans ce chapitre. Notre méthode est testée sur des images réelles et d'autres de synthèse. Les résultats obtenus sont comparés à ceux calculés par la méthode proposée dans [El-Ansari_10A]. Finalement, des statistiques sur le temps de traitement ainsi que l'effet de bruit sur les résultats sont fournis.

Chapitre 3

Résultats expérimentaux et statistiques

Sommaire

3.1	Statistiques et comparaisons.....	87
3.1.1	Les résultats sur les images de synthèse.....	87
3.1.2	Les résultats sur les images réelles.....	89
3.2	Temps de calcul.....	96
3.3	L'effet de bruit : L'incertitude liée à la position des primitives.....	97
3.4	Conclusion.....	98

Dans ce chapitre nous présentons les résultats de l'approche abordée au chapitre précédent. Les expériences sont réalisées sur des séquences d'images stéréoscopiques réelles et une de synthèse. Ce chapitre s'articule sur quatre parties : dans les deux premières parties, la méthode est appliquée sur des images de synthèse puis sur des images réelles. Les résultats ainsi obtenus ont été validés par des comparaisons statistiques multiples (quantitatives et qualitatives) avec ceux obtenus par la méthode proposée dans [El-Ansari_10A]. Dans la troisième partie, nous avons donné un détail sur le temps de calcul pour les différentes étapes de notre algorithme. La dernière partie analyse l'impact du bruit sur les résultats d'appariement. Pour cela, un bruit gaussien est ajouté sur les images avant d'en appliquer notre approche.

Désignons par STM (*spatio-temporelle Matching*) la nouvelle méthode que nous avons proposée, et TCM (*Temporal Consistent Matching*) la méthode proposée dans [El-Ansari_10A]. Les résultats de la comparaison entre les deux méthodes sont présentés pour montrer les améliorations apportées par la STM par rapport TCM.

3.1 Statistiques et comparaisons

3.1.1 Les résultats sur les images de synthèse

Tout d'abord, nous avons utilisé les images stéréoscopiques d'une séquence virtuelle de MARS/PRESCAN disponibles dans [datasite_wvandermaak_08]. La taille des images est de 512 x 512. La Figure 3.1 illustre l'image gauche de la trame #293 de la séquence. L'image de contour obtenue par le détecteur de Canny est représentée au milieu de la Figure 3.1. Les cartes de disparités correspondantes calculées par les deux méthodes STM et TCM sont présentées sur la Figure 3.2. Pour plus de clarté, nous utilisons des fausses couleurs pour représenter les différentes valeurs de la carte de disparité.

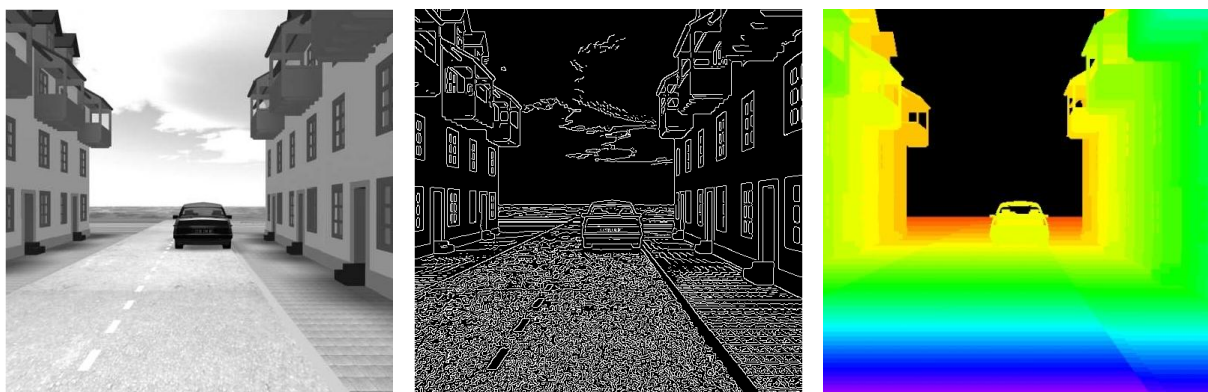


Figure 3.1– (à gauche) L'image gauche de la trame #293 de la séquence stéréoscopique virtuelle, (au milieu) l'image de contour correspondante obtenue par le détecteur de Canny et (à droite) la carte vérité-terrain fournie dans [datasite_wvandermaak_08].

Le tableau 3.1 résume les résultats de la correspondance obtenus par les deux méthodes (STM et TCM). Le tableau indique le nombre de points de contour appariés (NME), le nombre d'appariements corrects (NCM), ainsi que le pourcentage d'appariements corrects (PCM) obtenus pour la trame #293. Il est clair que l'approche STM fournit plus d'appariements

corrects par rapport à ceux de TCM (environ deux fois plus nombreux). Bien sûr, cela est dû à l'utilisation du détecteur de Canny qui fournit des contours très riches. D'autre part, le pourcentage de faux appariements en utilisant STM est beaucoup plus faible qu'avec TCM. Nous avons obtenu des performances similaires pour les autres trames de la séquence virtuelle.

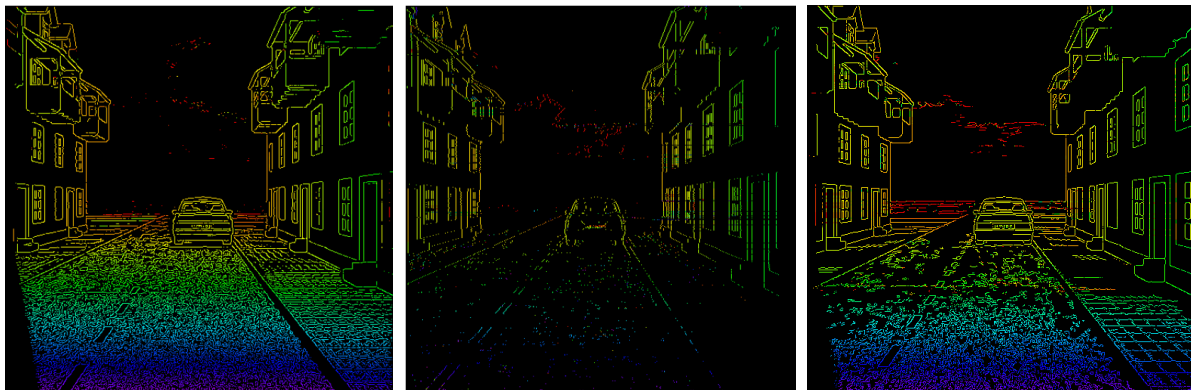


Figure 3.2- Les cartes de disparités de la paire d'images stéréoscopiques #293 de la séquence virtuelle calculées en utilisant la méthode (à gauche) STM, (au milieu) TCM et (à droite) TCM-Canny.

Méthode	NME	NCM	PCM (%)
TCM	15 934	14 021	88.03
STM	37 237	35 002	94.43

Tableau 3.1– Résultats d'appariement obtenus avec STM et TCM.

Les améliorations inhérentes à la nouvelle méthode, par rapport à celle de [El-Ansari_10A], provient de deux facteurs: (1) l'opérateur de Canny utilisé pour la détection des points de contour, (2) l'approche de la mise en correspondance spatio-temporelle proposée pour l'appariement des contours des paires d'images stéréoscopiques consécutives. Pour voir l'impact de ces deux facteurs sur la méthode proposée, nous proposons de comparer notre méthode STM avec la méthode TCM appliquée sur les points de contour extraits par l'opérateur de Canny (TCM-Canny) au lieu TCM avec déclivité [Miche_95].

Le tableau 3.2 résume les résultats obtenus pour de la trame #293 comparés avec ceux obtenus par les deux méthodes (TCM et TCM-Canny). Nous pouvons remarquer que le nombre d'appariements corrects (NCM) est augmenté de 9758 et 20981 lorsque nous utilisons respectivement TCM-Canny et STM. La méthode STM (resp. TCM-Canny) réussit d'avoir une augmentation de 149% (resp. 38%) de paires de points de contour appariés. Les pourcentages d'appariements corrects fournis par STM et TCM-Canny sont 94,43% et 89,13%, respectivement. La méthode STM augmente à la fois le nombre de points de contour appariés (NME) et le pourcentage d'appariements corrects obtenus (PCM). Alors que la méthode TCM-Canny ne fait qu'accroître le NME. Par conséquent, les améliorations apportées par notre approche (STM) proviennent de l'utilisation de la mise en correspondance spatio-temporelle des contours.

Méthode	NME	NCM	PCM (%)
TCM	15 934	14 021	88.03
TCM-Canny	28 321	25 244	89.13
STM	37 237	35 002	94.43

Tableau 3.2– Comparaison entre les résultats obtenus en utilisant SMT et TCM avec ceux obtenue par TCM-Canny.

3.1.2 Les résultats sur les images réelles

La méthode proposée a été également testée sur deux séquences d'images réelles acquises par un capteur de vision stéréoscopique embarqué à bord d'une voiture en mouvement. La vitesse du véhicule est de 90 km par heure. Le capteur de vision stéréoscopique utilisé fournit 10 images par seconde.

La figure 3.3 montre l'image de gauche #4183 de la séquence réelle #1 utilisée pour tester la méthode proposée, ainsi que son image de contour correspondante. Les cartes de disparité calculées par les méthodes STM et TCM sont présentées sur la figure 3.4.

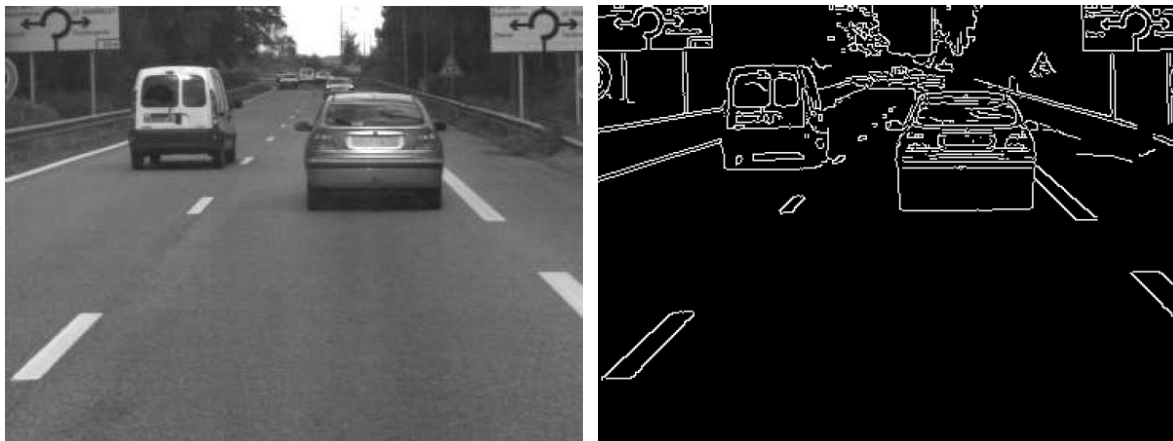


Figure 3.3 - (à gauche) L'image gauche #4183 de la séquence stéréoscopique réelle et (à droite) l'image de contour correspondante obtenue en utilisant le détecteur de Canny.

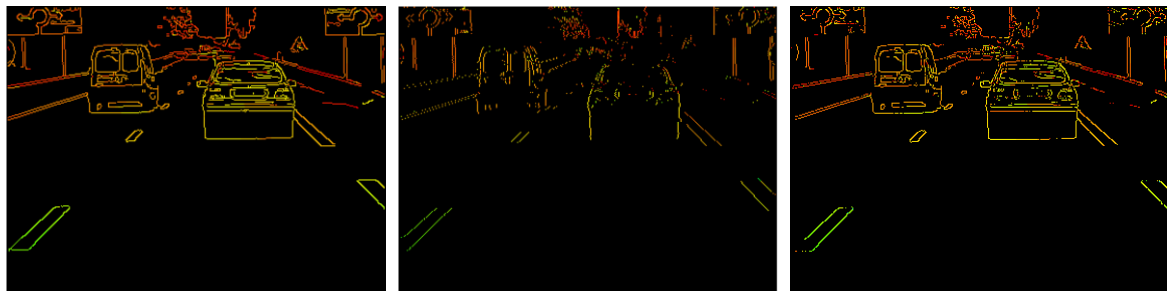


Figure 3.4 – La carte de disparités calculée en utilisant (à gauche) STM et (au milieu) TCM et (à droite) TCM-Canny.

Selon la contrainte de continuité de la disparité, les points de contour appartenant au même contour (c'est-à-dire les contours d'un même objet) doivent avoir des valeurs de disparité similaires ou très proche (de l'ordre de ± 1). Nous utilisons cette contrainte comme une référence pour évaluer la performance de la méthode proposée. Nous focalisons notre attention sur les disparités calculées au niveau des deux voitures (gauche et droite) comme le montre la figure 3.5. Les figures 3.6 et 3.7 montrent les cartes de disparités calculées par STM et TCM. Il est clair que les disparités calculées par la STM sont plus lisses que celles calculées par TCM. Pour une comparaison quantitative, nous avons analysé manuellement les disparités calculées par les deux méthodes.



Figure 3.5 - Les sous-images correspondantes aux véhicules de gauche et de droite.

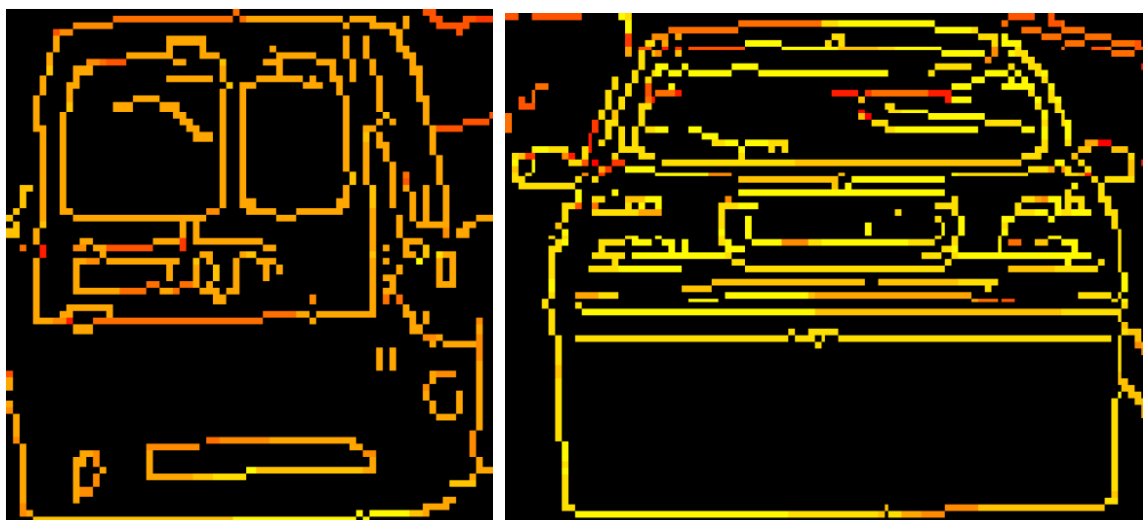


Figure 3.6– La carte de disparités pour le véhicule de gauche et de droite en utilisant STM.

Les tableaux 3.3 et 3.4 présentent à peu près le nombre de points de contour appariés (NME), le nombre d'appariements corrects (NCM) et le pourcentage d'appariements corrects (PCM) en utilisant les deux méthodes STM et TCM. Encore une fois, la méthode STM que nous avons proposée a plus de succès à mettre en correspondance plus de points de contour avec un pourcentage plus élevé d'appariements corrects. Le pourcentage d'appariements corrects trouvés pour la voiture de droite (tableau 3.3) avec la STM et TCM sont respectivement 89,79% et 71,31%, ainsi que le nombre d'appariements corrects sont 1108 et 179 respectivement. Une comparaison similaire a été faite pour la voiture de gauche (voir tableau

3.4). Dans l'ensemble, la STM est plus performante que TCM, et cela à la fois en termes de nombre d'appariements corrects et le pourcentage de fausses appariements.

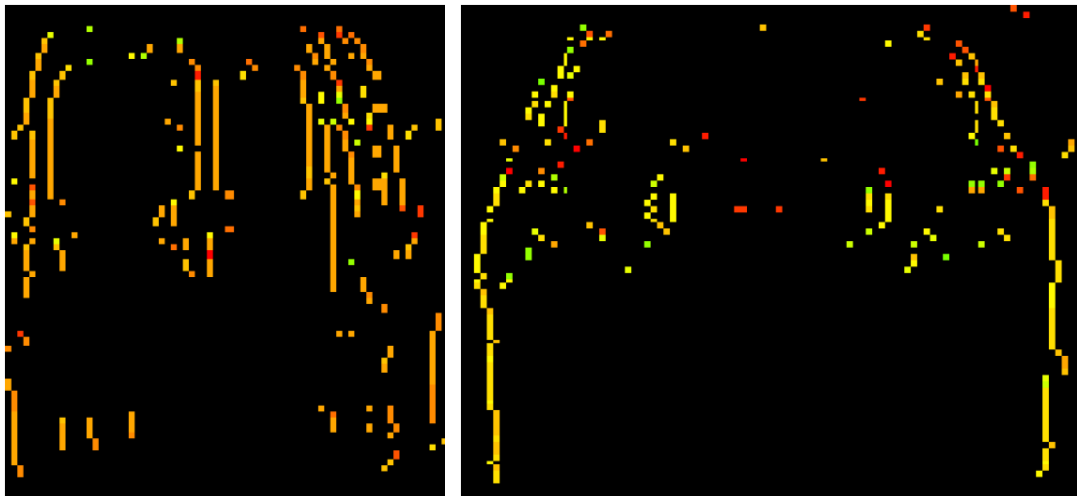


Figure 3.7– La carte de disparités pour le véhicule de gauche et de droite en utilisant TCM.

Méthode	NME	NCM	PCM (%)
STM	1 234	1 108	89.79
TCM	251	179	71.31

Tableau 3.3 – Résumé des résultats d'appariement obtenus avec STM et TCM au niveau de la voiture de droite.

Ensuite, la méthode STM et TCM ont été également appliquées sur la séquence réelle #2. La figure 3.8 illustre l'image gauche de la trame #1983 et son image de contour correspondante. Les cartes de disparités calculées par STM et TCM sont présentées sur la figure 3.9. Notre analyse est focalisée sur les disparités calculées au niveau des voitures les plus proches dans la séquence. Les figures 3.11 et 3.12 montrent les cartes de disparité calculées par STM et TCM. Les disparités ont été calculées en utilisant les mêmes principes que pour la séquence #1. Nous remarquons à partir des figures 3.11 et 3.12 que les disparités calculées par la méthode TCM sont beaucoup plus bruitées et peu nombreuses. Alors que la carte de disparités obtenue par STM est plus lisse et plus dense au niveau des contours. Comme nous pouvons le

constater dans les tableaux 3.5 et 3.6, la méthode STM a réussi à trouver plus d'appariements correctes par rapport à TCM.

Méthode	NME	NCM	PCM (%)
STM	710	621	87.46
TCM	348	296	85.06

Tableau 3.4 – Résumé des résultats d'appariement obtenus avec STM et TCM pour la voiture de gauche.

Méthode	NME	NCM	PCM (%)
STM	2 148	1 322	61.54
TCM	1 476	734	49.72

Tableau 3.5– Résumé des résultats d'appariement obtenus par STM et TCM lorsqu'elles sont appliquées sur la voiture de gauche (la voiture la plus proche) de la séquence réelle #2.

Méthode	NME	NCM	PCM (%)
STM	682	441	64.66
TCM	395	132	33.41

Tableau 3.6– Résumé des résultats d'appariement obtenus en utilisant STM et TCM lorsqu'elles sont appliquées sur la voiture de droite (la voiture au milieu sur la voie gauche) de la séquence réelle #2.



Figure 3.8 – (A gauche) l'image gauche de la trame #1983 de la séquence stéréoscopique réelle #2 et (à droite) l'image de contour correspondante obtenue en utilisant le détecteur de Canny.

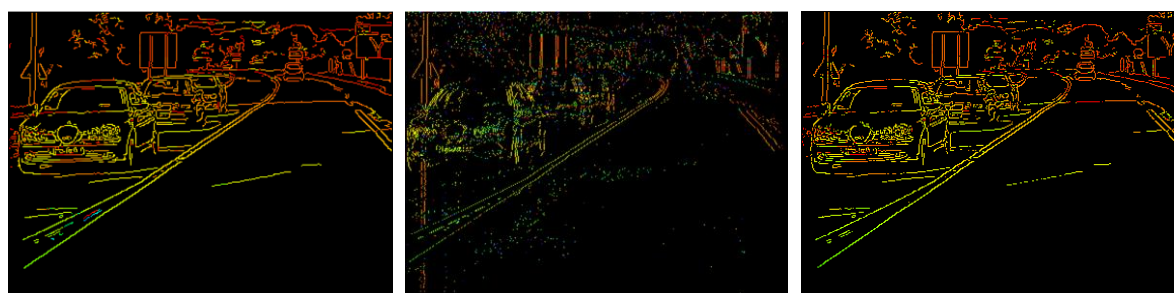


Figure 3.9 – Les cartes de disparités obtenues en utilisant (à gauche) STM et (à droite) TCM.

Pour voir l'impact de l'opérateur de Canny et la mise en correspondance spatio-temporelle que nous avons proposée sur les résultats obtenus, nous avons comparé les résultats obtenus par les deux méthodes STM et TCM-Canny. Ces dernières utilisent, toutes les deux, les contours détectés par l'opérateur de Canny. Le tableau 3.7 montre les résultats obtenus lorsque nous appliquons les méthodes TCM, TCM-Canny et STM sur les images des séquences réelles #1 et #2. Comme dans la séquence de synthèse, nous pouvons conclure que le nombre d'appariements corrects (NME) et le pourcentage d'appariement corrects (PCM) fournis par la méthode proposée (STM) sont supérieurs à ceux obtenus par TCM-Canny. Par conséquent, la grande partie de l'amélioration de la méthode STM est due par le facteur de la correspondance spatio-temporelle.

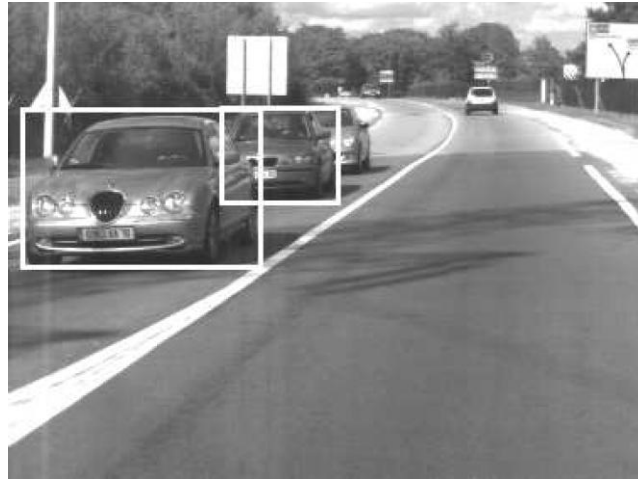


Figure 3.10 – L'image gauche de la trame #1983 avec des boîtes englobantes autour de (à gauche) la voiture de gauche (LC), et (à droite) la voiture de droite (RC).

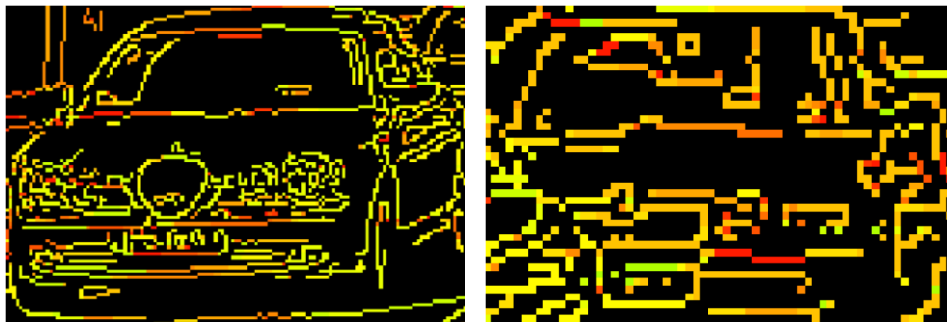


Figure 3.11 – Les cartes de disparités obtenues par STM pour les deux voitures les plus proches.

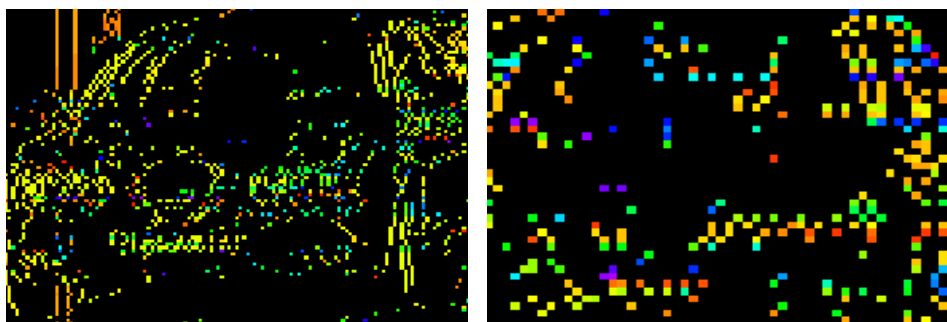


Figure 3.12 – Les cartes de disparités obtenues par TCM pour les deux voitures les plus proches.

	NME			NCM			PCM		
	TCM	TCM- Canny	STM	TCM	TCM- Canny	STM	TCM	TCM- Canny	STM
LC. #1	348	752	710	296	617	621	85.06	82.04	87.46
RC. #1	251	1 053	1 234	179	792	1 108	71.31	75.21	89.79
LC. #2	1 476	1957	2 148	734	992	1 322	49.72	50.69	61.54
RC. #2	395	645	682	132	312	441	33.41	48.37	64.66

Tableau 3.7 – Les résultats de comparaison des trois méthodes TCM, TCM-Canny et STM appliquées sur les voitures de gauche (LC.) et de droite (RC.) des séquences réelles #1 et #2.

3.2 Temps de calcul

Le matériel utilisé dans nos expériences est un HP Intel (R) Core (TM) 2 Duo CPU 2.09 GHZ fonctionnant sous Windows XP. Le tableau 3.8 décrit le temps d'exécution des différentes étapes de l'approche d'appariement proposée. L'appariement d'une paire d'images stéréoscopiques virtuelles, réelles #1 et réelles #2 prend 188.4554 ms, 123.7622 ms et 103.9962 ms, respectivement. Cela signifie que la STM peut apparier entre 8 et 9,6 images par seconde dans le cas des séquences réelles. Le temps consommé par le détecteur de Canny est d'environ 6% de la totalité du temps de calcul quand la méthode STM est appliquée à des séquences réelles.

Les séquences réelles utilisées dans ce travail sont acquises par un capteur stéréoscopique fournissant 10 images par seconde. Nous avons utilisé la séquence acquise hors ligne pour tester notre algorithme. Pour que notre algorithme puisse s'exécuter en temps réel, le temps d'exécution doit être réduit de 3-23 ms, ce qui permettra à la méthode proposée d'apparier 10 images par seconde. Pour cela, nous proposons comme perspective de ce travail d'améliorer le temps d'exécution en exploitant un certain parallélisme de l'algorithme. Étant donné que le

processus d'appariement est effectué indépendamment pour chaque ligne d'image, le temps de calcul peut être donc réduit en utilisant plusieurs processeurs.

	Virtuelle (ms)	Réelle #1 (ms)	Réelle #2 (ms)
Détection de contour	15.6508	7.0194	6.8880
L'association temporelle entre les points de contour	16.8200	7.9992	7.2114
L'appariement temporel des contours	3.5608	1.2002	0.9592
Génération de la carte de disparité initiale (IDM)	0.0208	0.0118	0.0132
Calcul de la gamme de disparités	10.3600	7.2340	8.1362
Estimation de la carte de disparités finale d_k	142.0430	100.2976	80.7882
Total	188.4554	123.7622	103.9962

Tableau 3.8 – Le temps d'exécution pour chaque étape de l'algorithme STM en (ms).

3.3 L'effet de bruit : L'incertitude liée à la position des primitives

Ici, nous étudions l'incertitude de la localisation des contours sur les résultats de la mise en correspondance. Nous proposons d'ajouter un bruit gaussien de moyenne nulle et de variance σ aux séquences d'images avant d'effectuer la détection des contours. Ce qui affecte la précision des positions des points de contour. Le bruit modifie les positions de certains points, ce qui génère des faux points de contour.

Le tableau 3.9 montre les résultats obtenus lorsque les séquences sont bruitées avec un bruit gaussien pour les différentes valeurs de σ . Pour chaque valeur de σ nous présentons le pourcentage de points de contour inchangé (PUEP), ainsi que le pourcentage d'appariements corrects (PCM) correspondant. Les résultats présentés dans le tableau 3.9 concernent uniquement la trame #293, la voiture de gauche (LC) et la voiture de droite (RC) des séquences virtuelle, réelles #1 et #2, respectivement. Nous pouvons remarquer que:

- Le détecteur de Canny n'est pas très vulnérable au bruit. Par exemple, pour $\sigma = 3$ le pourcentage de points de contour qui ont gardé leurs positions (PUEP) est de 99,53% sur RC de la trame #4183. Nous avons besoin d'ajouter un bruit gaussien avec une valeur importante de σ afin d'influencer la précision de la position pour un grand nombre de points de contour.
- Le taux d'appariement correct (PCM) est réduit lentement, même si la valeur de σ augmente fortement.

Les mêmes remarques sont valables pour les autres trames des différentes séquences. La deuxième remarque signifie que le nombre de faux appariements augmente lentement à mesure que la valeur de σ diminue fortement. Les faux appariements sont dus aux points de contours qui sont mal créés à cause du bruit ajouté aux images stéréoscopiques. Cependant, les points de contour mal localisés ont peu d'effet sur le processus d'appariement.

	$\sigma = 3$		$\sigma = 5$		$\sigma = 10$		$\sigma = 15$	
	PUEP	PCM	PUEP	PCM	PUEP	PCM	PUEP	PCM
#293	99.04	93.19	98.41	91.12	96.52	88.79	93.39	87.90
#1983 (LC)	98.89	60.94	98.10	59.74	96.24	57.35	92.98	53.45
#4183 (RC)	99.53	88.57	99.15	87.34	98.28	83.82	95.19	79.79

Tableau 3.9 – Les résultats obtenus sur les paires d'images #293, #1983 et #4183, lorsqu'un bruit gaussien est ajouté aux séquences d'images. Les deux dernières lignes représentent les statistiques calculées au niveau du véhicule de gauche (LC) de la trame #1983 et du véhicule de droite (RC) de la trame #4183, respectivement.

3.4 Conclusion

Nous avons vu dans ce chapitre les résultats expérimentaux obtenus en utilisant notre méthode d'appariement spatio-temporel proposée (STM). Les différents résultats présentés sont obtenus à partir des séquences d'images stéréoscopiques réelles et de synthèses. Les résultats obtenus sont comparés avec une méthode similaire qui utilise également l'approche

d'association (TCM). Les améliorations apportées par STM peuvent être justifiées par les raisons suivantes:

- Nous avons remplacé l'opérateur de déclivité, utilisé dans [El-Ansari_10A], par le détecteur de Canny. Ceci permet d'obtenir plus de points de contour que l'opérateur de déclivité qui n'est pas en mesure de détecter les contours horizontaux. Par conséquent, la méthode STM peut fournir plus de points de contour appariés.
- La carte de disparité initiale (IDM) calculée par STM est un élément crucial pour les deux méthodes STM et TCM, car les gammes de disparités pour ces deux méthodes sont dérivées directement de l'IDM. Dans [El-Ansari_10A], l'IDM a été calculée à partir de l'association entre les points de contour des images consécutives, ainsi que les correspondances entre les points de contour de la paire d'images stéréoscopiques précédentes. Cependant, dans l'approche proposée, l'IDM est calculée à partir de la mise en correspondance spatio-temporelle des contours des deux trames consécutives. Par conséquent, l'IDM calculé par la STM est plus précise que celle calculée par la méthode TCM.
- En TCM, le même espace de recherche est utilisé pour trouver des correspondances entre les points de contour sur les lignes d'images (ou épipolaire) correspondantes. Dans l'approche proposée, MCEPs sont utilisés pour diviser l'espace de recherche en sous-espaces. Cela force le chemin de recherche dans le graphe de la programmation dynamique afin de traverser les paires de points de contour à appairer dans l'IDM. Par conséquent, les résultats trouvés par la STM sont beaucoup améliorées.

Conclusion et perspectives

L'objectif de tout système d'aide à la conduite (ADAS) est d'être en mesure de prévenir le conducteur dans des situations dangereuses risquant d'aboutir à un accident. Le problème majeur des systèmes ADAS est la perception de l'environnement 3D autour du véhicule intelligent. Cette perception constitue souvent une étape primordiale pour les autres fonctionnalités du système, à savoir la détection et la localisation des obstacles dans la scène, détection de la pluie, reconnaissance des feux de croisement et des panneaux, etc.

La stéréovision constitue un outil très populaire pour la perception de la profondeur à partir de deux images acquises sous des angles de vues différents, en tentant d'imiter le système de vision humaine. Le problème délicat de tout algorithme basé sur la stéréovision consiste à trouver la mise en correspondance entre les images stéréoscopiques. Cette tâche est l'une des problèmes les plus difficiles en vision par ordinateur due aux différentes contraintes géométriques et photométriques liées à la fois à la scène et aux caméras. A chaque instant (selon la fréquence d'acquisition des caméras) les caméras fournissent deux images stéréoscopiques au système qui se chargera de les traiter afin de résoudre les problèmes cités ci-dessus, à savoir la mise en correspondance et la reconstruction 3D.

Dans cette thèse, nous avons présenté une méthode de mise en correspondance spatio-temporelle rapide qui est très utile pour les applications d'aide à la conduite ADAS. L'approche met en œuvre la relation temporelle entre des trames consécutives. Ceci permet d'utiliser les résultats de l'appariement obtenus sur la trame précédente afin d'améliorer les résultats de la mise en correspondance pour la trame courante. Une carte de disparité initiale IDM est générée dont ses points sont utilisée comme des points de contrôle. Ces points permettent ensuite de guider le processus de la recherche par programmation dynamique afin de trouver des correspondances entre les points de contour restants dans la trame courante. Ceci est réalisé pour chaque ligne en divisant l'espace de recherche de la programmation

dynamique en un nombre de sous-espaces en fonction du nombre de points de contrôle appariés.

La validation de notre approche a été établie par des expérimentations sur des images stéréoscopiques de synthèses et réelles. Ces dernières nous ont permis de tester notre algorithme sur des images capturées par une paire de caméras embarquées à bord d'un véhicule intelligent. Tous les résultats obtenus par notre méthode sont comparés avec ceux obtenus avec une méthode proposée dans [El-Ansari_10A] qui utilise une approche similaire. Les statistiques sur le temps de calcul et les incertitudes liées à la détection des contours ont validé la robustesse et l'efficacité de la proche proposée.

Ce travail offre plusieurs perspectives intéressantes :

Une première perspective de l'approche présentée est de l'appliquer afin de calculer des cartes de disparités denses. Cependant, la contrainte du temps reste un défi pour aboutir à un tel objectif. Pour cela, nous souhaitons exploiter le parallélisme de certaines étapes de notre algorithme afin que nous puissions faire une implémentation adaptée aux processeurs parallèles ou aux cartes numériques multiprocesseurs. Dans son implémentation actuelle, la méthode proposée ne pouvait pas être appliquée en temps-réel. Notons que l'association temporelle et le calcul de la carte de disparité finale dans l'algorithme proposé sont exécutés indépendamment pour chaque ligne de l'image. Ces deux étapes consomment environ 84,61% de l'ensemble du temps d'exécution. L'utilisation d'une architecture à processeurs parallèles permettra de réduire le temps d'exécution, telle que la durée de traitement peut être réduite en utilisant un certain nombre de processeurs égal au nombre de lignes d'image. A titre d'exemple, le nombre de lignes de la séquence réelle #1 est 286. En utilisant 286 processeurs, le temps d'exécution des deux étapes ci-dessus peut être divisé par 286. Ainsi, l'appariement d'une trame de la séquence # 1 prendra seulement 16,30 ms au lieu de 103.99 ms. Dans ce cas, notre approche peut correspondre plus de 60 images par seconde des séquences d'images réelles.

Dans une seconde perspective, nous sommes convaincus que le point fort de notre approche est la carte de disparité initiale (IDM) générée par une méthode de mise en correspondance spatio-temporelle. Plus que la carte IDM est précise et riche en information, plus que la carte de disparité finale est fiable. La technique d'association

utilisée, malgré sa simplicité, elle souffre des limitations inhérentes à la scène traitée, à savoir la scène routière. Ainsi, nous souhaitons améliorer la approche actuelle en utilisant des fonctionnalités plus puissantes, telle que la transformation de caractéristiques visuelles invariante à l'échelle (ou SIFT) [Lowe_04]. Toujours dans la même idée, nous pensons de développer des extensions spatio-temporelles aux approches d'appariement des régions proposées dans [El_Ansari_00 et El_Ansari_07] en utilisant la même méthode proposée dans cette thèse.

Une autre perspective de ce travail est d'exploiter les résultats 3D fournis par notre méthode afin de développer des techniques de détection dans l'environnement du véhicule, plus particulièrement, la détection des véhicules, piétons, panneaux de signalisation routière, segmentation de la route, etc.

Bibliographie

- [Ambrosch_10] K. Ambrosch, W. Kubinger: Accurate hardware-based stereo vision, *Computer Vision and Image Understanding*, pp. 1303-1316, 2010.
- [Areibi_01] S. Areibiet J. Zelek: A Smart Reconfigurable Visual System for the Blind, *Tunisian-German Conference on: Smart Systems and Devices*, pp: 628-633, Hamamat, Tunis, Mars 2001.
- [Aschwanden_92] P. Aschwanden et W. Guggenbül: Experimental results from a comparative study on correlation type registration algorithms. *Robust computer vision : Quality of Vision Algorithms*, pages 268-282, 1992. 24.
- [Baker_81] H. H. Baker and T. O. Binford: Depth from edge and intensity based stereo, *inProc. 7th Int. Joint Conf. AI*, pp. 631-636, 1981
- [Baker_81] H.H. Baker et T.O. Binford : Depth from edge and intensity based stereo. In *In : Proc. 7th Inter. Joint Conf. on Artificial Intelligence*, pages 631–636, 1981.
- [Barnard_82] Barnard S, Fisher M (1982) Computational stereo. *ACM ComputSurv* 14:553–572.
- [Benard_84] M. Benard Automatic: A method based on feature detection and dynamic programming, *Photogrammetria*, vol. 39, pp.169 -181 1984.
- [Benard_84] M. Benard : Automatic stereo photogrammetry a method bared on feature detection and dynamic programming. *Pattern Recognition in Photogrammetry*, 39:169– 181, 1984.
- [Bensrhair_96] A. Bensrhair, P. Miche et R. Debie : Fast and automatic stereo vision matching algorithm based on dynamic programming method. in *Pattern Recognition Letters*, 17:457–466, 1996.
- [Bertozzi_00] Bertozzi M, Broggi A, Fascioli A (2000): Vision-based intelligent vehicles: state of the art and perspectives. *Rob AutonSyst* 32:1–16
- [Birchfield_98] S. Birchfield et C. Tomasi : Depth discontinuities by pixel-to-pixel stereo. In *Proc. of the IEEE Inter. Conf. on Computer Vision*, 1998.
- [Birchfield_99A] S. Birchfield et C. Tomasi : Multiway cut fro stereo and motion with slanted surfaces. In *Inter. Joint Conf. on Artificial Intelligence*, volume 1, pages 489–495, 1999.

- [Birchfield_99B] Birchfield S., Tomasi C.: Depth Discontinuities by Pixel-to-Pixel Stereo, *Int. Journal of Computer Vision*, 35(3), 1999, 269–293.
- [Bleyer_05] M. Bleyer et M. Gelautz: A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry & Remote Sensing*, 59:128–150, 2005.
- [Bobick_99] A.F. Bobick et S.S. Intille : Large occlusion stereo. *IJCV*, 33(3):181–200, 1999.
- [Boukir_92] Boukir, S., Bouthemy, P., Chaumette, F., Juvin, D. (1992): Mise en correspondance de segments dans une séquence d'images par une approche locale. Rapport de recherche 1792, Institut National de Recherche en Informatique et en Automatique, INRIA.
- [Boykov_01] Boykov Y, Veksler O, Zabih R (2001): Fast approximate energy minimization via graph cuts. *IEEE Trans Pattern Anal Mach Intell* 23(11): 1222–1239.
- [Boykov_01] Boykov, Y., Veksler, O., Zabih., R. (2001): Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222-1239.
- [Brown_03] Brown MZ, Burschka D, Hager GD (2003): Advances in computational stereo. *IEEE Trans Pattern Anal Mach Intell* 25(8): 993–1008
- [Brown_03] M.Z. Brown, D. Burschka et G.D. Hager : Advances in computational stereo. *IEEE Trans. on PAMI*, 25:993–1008, Aug 2003.
- [Campbell_08] N. Campbell, G. Vogiatzis, C. Hernandez et R. Cipolla : Using multiple hypotheses to improve depth-maps for multiview stereo. In *ECCV*, pages 766–779, 2008.
- [Canny_86] Canny, J. (1986): A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI, 8(6):679-698.
- [Chaplin_98] B-A. Chaplin and M-A: Chapman.1998-A procedure for 3D motion estimation from stereo image sequences for a mobile mapping system. In: *The International Archives of Photogrammetry and Remote Sensing Volume 32 (3W1)*, 1998.
- [Corke_99] Corke P., Dunn P., Banks J.: Frame-rate stereopsis using non-parametric transforms and programmable logic, *IEEE Int. Conf. on Robotics and Automation*, Detroit, USA, 1999, 1928–1933.
- [Cyganek_03] Cyganek B, Borgosz J (2003): An improved variogram analysis of the maximum expected disparity in stereo images. In: Bigun J, Gustavsson T (eds), *SCIA 2003*, LNCS 2749, pp 640–645.

- [Darabiha_03] Darabiha A., Rose J., Maclean J.W: Video-rate stereo depth measurement on programmable hardware. Int. Conf. on Computer Vision and Pattern Recognition, Madison, Wisconsin, Vol. 1, 203–210, 2003.
- [datasite_wvandermaak_08] Stereo data for algorithms evaluation (2008): <http://stereodatasets.wvandermaak.com/>
- [Davis_05] Davis J, Nehab D, Ramamoorthi R, Rusinkiewicz S (2005): Spacetime stereo: a unifying framework for depth from triangulation. *IEEE Trans Pattern Anal Mach Intell* 27(2):1–7
- [Devernay_97] F. Devernay : Vision stéréoscopique et propriétés différentielles des surfaces. Thèse de doctorat, Ecole Polytechnique, France, janvier 1997. 24.
- [Dhond_89] Dhond UR, Aggarwal JK (1989) : Structure from stereo a review. *IEEE Trans Syst Man Cybern* 19:1489–1510.
- [Duda_72] Duda, R. O. and P. E. Hart: Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Comm. ACM*, Vol. 15, pp. 11–15 (January, 1972).
- [El_Ansari_00] M. El Ansari, L. Masmoudi, L. Radouane: A new region matching method for stereoscopic images. *Pattern Recognition Letters* 21(4): 283-294 (2000).
- [El_Ansari_07] M El Ansari, L Masmoudi, A Bensrhair: A new regions matching for color stereo images. *Pattern recognition letters* 28 (13), 1679-1687 (2007).
- [El-Ansari_08] El-Ansari M, Mousset S, Bensrhair A (2008): A new stereo matching approach for real-time road obstacle detection for situations with deteriorated visibility. In: *Proceedings of the IEEE intelligent vehicle symposium*. Eindhoven University of Technology, Eindhoven.
- [El-Ansari_10A] El-Ansari M, Mousset S, Bensrhair A (2010): Temporal consistent real-time stereo for intelligent vehicles. *Pattern RecognLett* 31(11):1226–1238
- [El-Ansari_10B] El-Ansari M, Mousset S, Bensrhair A, Bebis G (2010): Temporal consistent fast stereo matching for advanced driver assistance systems (ADAS) . *Proceedings of the IEEE intelligent vehicles symposium*. San Diego, pp 825–831
- [Fabbri_10] R. Fabbri et B. Kimia : 3d curve sketch - flexible curve-based stereo reconstruction and calibration. In *CVPR*, 2010.
- [Faugeras_01] O. Faugeras and Q.T. Luong: *The geometry of multiple images*. MIT Press, Cambridge, 2001.
- [Faugeras_93A] O. Faugeras: *Three-Dimensional Computer Vision, a Geometric Viewpoint*. MIT Press, 1993.

- [Faugeras_93B] O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Moll, G. Berry, J. Vuillemin, P. Bertin et C. Proy: Real time correlationbased stereo: algorithm, implementations and applications, Rapport de recherche n 2013 (INRIA), août 1993.
- [Fleet_94] Fleet D.J. : Disparity from local weighted phase-correlation. Int. Conf. on Systems, Man, and Cybernetics, San Antonio, USA, Vol. 1, 48–54.
- [Forstmann_04] S. Forstmann, Y. Kanou, J. Ohya, S. Thuerling et A. Schmitt : Real-time stereo by using dynamic programming. Workshop on real-time 3D sensors and their use, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 3, pages 29–36, Washington, DC, USA, juin 2004.
- [Furukawa_07] Y. Furukawa et J. Ponce : Accurate, dense, and robust multi-view stereopsis. In CVPR, 2007.
- [Furukawa_08] Y. Furukawa et J. Ponce : Accurate, dense, and robust multi-view stereopsis. IEEE Trans. on PAMI, 1(1):2008, August 2008.
- [Fusiello_00] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. Machine Vision and Applications, 12(1):16–22, 2000.
- [Gluckman_99] J. Gluckman, S.K. Nayar et K.J. Thorek, Real-Time Omnidirectional and Panoramic Stereo, DARPA Image Understanding Workshop, 299–303, Monterey, USA, 1999.
- [Gong_05] M. Gong and Y.H. Yang: Near real-time reliable stereo matching using programmable graphics hardware. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 924–931, San Diego, USA, June 2005.
- [Gong_06] Gong M (2006): Enforcing temporal consistency in real-time stereo estimation. In: Proceedings of the European conference on computer vision, Graz, pp III–564–577
- [Gruen_01] A. Gruen and T. S. Huang: Calibration and Orientation of Cameras in Computer Vision. Secaucus, NJ: Springer-Verlag New York, Inc., 2001.
- [Harris_88] C. Harris et M. Stephens : A combined corner and edge detector. Proceedings of The Fourth Alvey Vision Conference, pages 147–151, Manchester, UK, janvier 1988. 22.
- [Hartley_99] R.I. Hartley: Theory and practice of projective rectification. International Journal of Computer Vision, 35(2):115–127, November/December 1999.
- [Hautière_06] N. Hautière, R. Labayrade, M. Perrollaz et D. Aubert : Road scene analysis by stereovision : a robust and quasi-dense approach. Proceedings of the IEEE

- International Conference control, automation, robotics and vision , pages 1453-1458, Singapore, décembre 2006. 106, 114.
- [Hernandez_07] C. Hernandez, G. Vogiatzis et R. Cipolla : Probabilistic visibility for multiview stereo. In CVPR, pages 1–8, 2007.
- [Hiep_09] V.H. Hiep, R. Keriven, P. Labatut et J.-P. PONS : Towards high-resolution large-scale multi-view stereo. In CVPR, pages 1430–1437, 2009.
- [Hirschmüller_01] H. Hirschmüller: Improvements in real-time correlation-based stereo vision. IEEE Workshop on Stereo and Multi-Baseline Vision, Kauai, HI, USA, décembre 2001. 106, 107
- [Hirschmuller_02] H. Hirschmuller, P.R. Innocent, and J. Garibaldi: Real-time correlation-based stereo vision with reduced border errors. International Journal of Computer Vision, 47:229–246, April-June 2002.
- [Horaud_93] Horaud, R., Monga, O. (1993): Vision par ordinateur, outils fondamentaux. Traité des nouvelles technologies, série Informatique. Hermès, Paris, France.
- [Jia_03] Y. Jia, Y. Xu, W. Liu, C. Yang, Y. Zhu, X. Zhang et L. An: A Miniature Stereo Vision Machine for Real-Time Dense DepthMapping, 3rd International Conference Computer Vision Systems, 268–277, Graz, Austria, 2003.
- [Jia_04] Y. Jia, Y. Xu, W. Liu, C. Yang, Y. Zhu, X. Zhang et L. An: A Miniature Stereo Vision Machine (MSVM-III) for Dense Disparity Mapping, 17th International conference on Pattern Recognition, vol. 1, 728–731, Cambridge, UK, 23-26 août 2004.
- [Jin_10] S. Jin, J. Cho, X. D. Pham, K. M. Lee, S-K. Park, M. Kim, J. W. Jeon: FPGA Design and Implementation of a Real-Time Stereo Vision System, IEEE Trans. Circuits Syst. Video Technol., pp. 15 - 26, Jan. 2010.
- [Kanade_94] Kanade, T., Okutomi, M. (1994): A stereo matching algorithm with an adaptive window : Theory and experiment. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(9):920-932.
- [Kanade_96] T.Kanade, A.Yoshida, K.Oda, H.Kano et M.Tanaka: A Stereo Machine for Video-Rate Dense Depth Mapping and Its New Applications, Proceeding of IEEE International conference on Computer Vision and Pattern Recognition, pp.196-202, Juin 1996.
- [Kim_05] Kim, C., Lee, K., Choi, B., Lee, S. (2005): A dense stereo matching using two-pass dynamic programming with generalized ground control points. In IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR, vol. 2, pages 1075-1082, San Diego, Etats-Unis.

- [Kimura_99] Kimura S., Shinbo T., Yamaguchi H., Kawamura E., Naka K: A Convolver-Based Real-Time Stereo Machine (SAZAN). Conf. on Computer Vision and Pattern Recognition, 457–463.
- [Kolev_07A] K. Kolev, M. Klodt et T. Brox : Continuous global optimization in multiview 3d reconstruction. IJCV, 84(1):80–96, 2007.
- [Kolev_07B] K. Kolev, M. Klodt, T. BROX, S. ESEDOGLU et D. CREMERS : Continuous global optimization in multiview 3d reconstruction. In CVPR, 2007.
- [Kolev_10] K. Kolev, T. Pock et D. Cremers: Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In ECCV, 2010.
- [Kolmogorov_01] Kolmogorov, V. & Zabih, R. (2001): Computing visual correspondence with occlusions using graph cuts. In Proceedings of the International Conference on Computer Vision (ICCV), pp. 508-515.
- [Kolmogorov_02A] Kolmogorov, V. & Zabih, R. (2002): Multi-camera scene reconstruction via graph cuts. In Proceedings of the 7th European Conference on Computer Vision (ECCV), pp. 82-96. Springer-Verlag.
- [Kolmogorov_02B] Kolmogorov, V. & Zabih, R. (2002): What energy functions can be minimized via graph cuts? In Proceedings of the 7th European Conference on Computer Vision (ECCV), pp. 65- 81. Springer-Verlag.
- [Konolige_97] K. Konolige, Small Vision Systems: Hardware and Implementation, 8th International Symposium on Robotics Research, 111–116, Hayama, Japan, 1997.
- [Konrad_00] Konrad, J., Lan, Z. D. (2000): Dense disparity estimation from feature correspondences. In Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems, vol. 3957, pages 90-101, San Jose, CA, USA.
- [Kosov_09] Kosov, S., Thormahlen, T., Seidel, H.-P. (2009): Accurate real-time disparity estimation with variational methods:. In International Symposium on Visual Computing, vol. 5875, pages 796-807, Las Vegas, United States.
- [Labayrade_02] Labayrade R, Aubert D, Tarel JP (2002): Real time obstacle detection in stereo vision on non flat road geometry through v-disparity representation. In: Proceedings IEEE intelligent vehicle symposium, Versailles.
- [Labayrade_02] R. Labayrade, D. Aubert, and J. P. Tarel: Real time obstacle detection in stereo vision on non flat road geometry through v-disparity representation. In Proc. IEEE Intelligent Vehicle Symposium, Versailles, France, June 2002.
- [Ledesma_10] Ledesma, R., Montes, S., Po?, F. & Lopez-Ramon, M. (2010): Individual differences in driver inattention: The attention-related driving errors scale, Traffic Injury Prevention 11(2): 142–150.

- [Lee_94] J.H. Lee et N.L. Ricker: Extended kalman filter based nonlinear model predictive control. In *Industrial and Engineering Chemistry Research*, volume 33, pages 1530–1541, 1994.
- [Lemonde_05] V. Lemonde: Stéréovision embarquée sur véhicule : De l'auto-calibrage à la détection d'obstacles. Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, 2005. 2, 3, 33, 46, 110, 115
- [Leung_04] Leung C, Appleton B, Lovell BC, Sun C (2004): An energy minimisation approach to stereo-temporal dense reconstruction. In: *Proceedings of the IEEE international conference on pattern recognition*, Cambridge, pp 72–75.
- [Lowe_04] Lowe DG (2004): Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 60(2):91–110.
- [Marc_04] R.Y. Marc, M. Pollefeys, and S. Li: Improved real-time stereo on commodity graphics hardware. In *Proceedings of IEEE Workshop on Computer Vision and Pattern Recognition*, pages 36–36, Washington, DC, USA, 2004.
- [Marr_77] D. Marr and T.Poggio: A Theory of Human Stereo Vision. Technical Report AI Memo 451, MIT Artificial Intelligence Laboratory, November 1977.
- [Mazoul_12] A. Mazoul, M. El Ansari, K. Zebbara, G. Bebis: Fast spatio-temporal stereo for intelligent transportation systems, *Pattern Analysis and Applications*, ISSN 1433-7541, DOI 10.1007/s10044-012-0310-x, 2012.
- [Merrell_07] P. Merrell, A. Akbarzadeh, L. WANG, P. Mordohai, J.-M. Frahm, R. YANG, D. NISTER et M. POLLEFEYS: Real-time visibility-based fusion of depth maps. In *ICCV*, 2007.
- [Miche_95] Miché P, Debie R (1995): Fast and self-adaptive image segmentation using extended declivity. *Ann Telecommun* 50(3–4): 401–410.
- [Miled_06] Miled, W., Pesquet, J. C., Parent, M. (2006): Disparity map estimation using a total variation bound. In the *3rd Canadian Conference on Computer and Robot Vision*, pages 48-55.
- [Miled_09A] W. Miled, J.C. Pesquet et M. Parent: A convex optimization approach for depth estimation under illumination variation. *IEEE Trans. on Image Processing*, 18(4):813–830, Avril 2009.
- [Miled_09B] Miled, W., Pesquet-Popescu, B., Cherif., W. (2009): A variational framework for simultaneous motion and disparity estimation in a sequence of stereo images. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 741-744, Taipei, Taiwan.

- [Min_06] D.Min, S. Yoon et K. Sohn: Segment-based stereo matching using energy-based regularization. In *Multimedia Content Representation, Classification and Security*, volume 4105, pages 761–768, 2006.
- [Moses_09] Y. Moses et I. Shimshoni: 3d shape recovery of smooth surfaces dropping the fixed viewpoint assumption. *IEEE Trans. on PAMI*, 31:1310 – 1324, July 2009.
- [Naoulou_06] A. Naoulou: Architectures pour la stereovision passive dense temps reel : application a la stereo-endoscopie, rapport de thèse, 05.09.2006.
- [Ohta_85] Y. Ohta et T. Kanade: Stereo by two-level dynamic programming. In *Proc. of the 9th inter. joint conference on Artificial intelligence*, volume 2, pages 1120– 1126, 1985.
- [OpenCV] Open Source Computer Vision Library:
<http://www.intel.com/research/mrl/research/opencv/>
- [Otha_89] Otha Y, Kanade T (1989): Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans Pattern Anal Mach Intell* 7(2):139–154.
- [Papadimitriou_96] D.V. PAPADIMITRIOU et T.J. DENNIS : Epipolar line estimation and rectification for stereo image pairs. *IEEE Trans. on Image Processing*, 5:672–676, 1996.
- [Paragios_06] N. Paragios, Y. Chen et O. Faugeras: *Handbook of Mathematical Models in Computer Vision*. Springer, 2006.
- [Rabaud_04] C. Rabaud, O. Strauss et F. Comby: Fuzzy dense stereoscopic matching. In *Rencontres Francophones sur la Logique Floue et ses Application (LFA)*, 2004.
- [Roy_98] Roy, S., Cox, I.: A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem, *International Conference on Computer Vision*, (1998).
- [Sadeghi_08] H. Sadeghi, P. Moallem et S.A. Monadjemi: Feature based dense stereo matching using dynamic programming and color. *International Journal of Engineering and Mathematical Sciences*, 3(4):179–186, 2008.
- [Scharstein_02] Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:1.
- [Schmid_98] C. Schmid et A. Zisserman: The geometry and matching of curves in multiple views. In *in Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 394–409, Jun 1998.
- [Sinha_07] S.N. Sinha, P. Mordohai et M. Pollefeys : Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *ICCV*, 2007.

- [Sinha_09] S.N. Sinha, D. Steedly et R. Szeliski: Piecewise planar stereo for imagebased rendering. In ICCV, 2009.
- [Site_Tyzz] Tyzz : <http://www.tyzz.com>.
- [Slesareva_05] Slesareva, N., Bruhn, A., Weickert, J. (2005): Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps. In the 27th DAGM Symposium, vol. 3663, pages 33-40, Vienna, Austria.
- [Smith_09] B.M. Smith, L. Zhang et H. Jin: Stereo matching with nonparametric smoothness priors in feature space. In IEEE Conf. on CVPR, pages 1–8, 2009.
- [stereo_smp] http://www.vision.deis.unibo.it/smatt/stereo_smp.html.
- [Sun_02] C. Sun: Fast stereo matching using rectangular subregioning and 3D maximum surface techniques. International Journal of Computer Vision, 47(1-3):99-117, 2002. 33, 34, 77.
- [Sun_05] Y. J. Sun, LI, S.B. Kang et H.-Y. Shum : Symmetric stereo matching for occlusion handling. In CVPR, 2005.
- [Sunyoto_04] Sunyoto H., van der Mark W., Gavrilă D. M: A Comparative Study of Fast Dense Stereo Vision Algorithms, IEEE Intelligent Vehicles Symposium, Parma, Italy, 252–259.
- [Tao_01] Tao H, Sawhney HS, Kumar R (2001): Dynamic depth recovery from multiple synchronized video streams. In: Proceedings of IEEE international conference on computer vision and pattern recognition. Kauai.
- [Toulminet_06] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair et A. Broggi: Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. IEEE Transactions on Image Processing, 15(8):2364-2375, août 2006. 106
- [Tsai_86] R.Y. TSAI: An efficient and accurate camera calibration technique for 3d machine vision. In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pages 364–374, Miami Beach, FL, 1986.
- [Ttofis_12] C. Ttofis and T. Theodoridis: Towards Accurate Hardware Stereo Correspondence: A Real-Time FPGA Implementation of a Segmentation-Based Adaptive Support Weight Algorithm, to appear in the Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE12), Dresden, Germany, 12-16 March 2012.
- [Van_01] Van der Val G., Hansen M. and Piacentino M: The ACADIA Vision Processor, 5th Int. Workshop on Computer Architecture for Machine Perception, Padova, Italy, 2001, 31–40.

- [VanderMark_06] W. van der Mark et D.M. Gavrilu: Real-time dense stereo for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):38-50, mars 2006. 9, 107, 108
- [Vedula_91] Vedula S, Baker S, Rander P, Collins R, Kanade T (1991): Threedimensional scene flow. In: *Proceedings of IEEE international conference on computer vision and pattern recognition*, pp II-722-729.
- [Veksler_01] Veksler, O. (2001): Stereo matching by compact windows via minimum ratio cycle. In *IEEE Conference Proceedings of International Conference on Computer Vision*, vol. 1, pages 540-547, Vancouver, BC, Canada.
- [Veksler_03] O. Veksler: Fast variable window for stereo correspondence using integral images. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 565-570, Madison, WI, USA, juin 2003. 30.
- [Veksler_05] Veksler, O. (2005): Stereo correspondence by dynamic programming on a tree. In *IEEE Conference Proceedings of Computer Vision and Pattern Recognition*, vol. 2, pages 384-390, San Diego, United States.
- [Veksler_99] Veksler, O. (1999): Efficient graph-based energy minimization methods in computer vision. Thèse de doctorat, Université de Cornell, Ithaca, Etats-Unis.
- [Vogiatzis_07] G. Vogiatzis, C. Hernández, P. H. S. Torr et R. Cipolla: Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. on PAMI*, 29:2241-2246, 2007.
- [Wei_04] Wei, Y., Quan, L. (2004): Region-based progressive stereo matching. In *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, vol. 1, pages 106-113, Washington, Etats-Unis.
- [Wheatstone_38] [SK 50] C. Wheatstone: On some remarkable, and hitherto unobserved, Phenomena of Binocular Vision, *Philosophical Transactions of the Royal Society of London*, vol. 128, pp. 371-394, June 1838.
- [Woo_00] Woo, O., Ortega, A. (2000): Overlapped block disparity compensation with adaptive windows for stereo image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(2):194-200.
- [Woodfill_97] J. Woodfill et B. Von Herzen: Real-Time Stereo Vision on the PARTS Reconfigurable Computer, *5th IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 201-210, Napa Valley, USA, avril 1997.
- [Xing_09] Xing, M., Hexin, C., Yan, Z. (2009): Stereo image coding method using stereo matching with difference based adaptive searching windows. In *IEEE International Workshop on Imaging Systems and Techniques*, pages 373-376, Shenzhen.

- [Yang_03] R. Yang et M. Pollefeys: Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware, IEEE Computer Vision and Pattern Recognition 2003, pp. (I-211 - I-217), 18-20 June 2003.
- [Yuille_84] A. L. Yuille and T. Poggio: A generalized ordering constraint for stereo correspondence, MIT Artificial Intell. Lab., Tech. Rep. AI Memo 777, May 1984
- [Zhang_00] Z. Zhang: A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 1330–1334, 2000.
- [Zhang_03] Zhang L, Curless B, Seitz SM (2003): Spacetime stereo: shape recovery for dynamic scenes. In: Proceedings of IEEE international conference on computer vision and pattern recognition, Madison, pp 367–374.
- [Zhang_09] Zhang G, Jia J, Wong T, Bao H (2009): Consistent depth maps recovery from a video sequence. IEEE Trans Pattern Anal Mach Intell 31(6):974–988.
- [Zhang_96] Z. Zhang: Determining the epipolar geometry and its uncertainty, A review. Technical report, INRIA Technical Report 2927, 1996.