



Université de Tébessa

Faculté des sciences exactes, sciences naturelles et de la vie
Département de mathématique et d'informatique

**Perception de l'environnement par stéréovision :
Application à la sécurité dans les systèmes de transports terrestres**

Mémoire présenté en vue de l'obtention de
Magistère en Informatique

Par
Kerboua Adlen

Encadré par
Pr. DJEDI NourEddine

Jury composé de :

Président :

Dr. LAOUAR Rida Maître de Conférences à l'Université de Tébessa.

Examineurs :

Dr. BABAHENINI Med Chaouki Maître de Conférences à l'Université de Biskra.

Dr. CHERIF Foudil Maître de Conférences à l'Université de Biskra.

Rapporteur :

Pr. DJEDI NourEddine Professeur de l'Université de Biskra.

Septembre 2011

Table des matières :

1. Introduction :

1.1. Introduction générale.....	08
1.2. La problématique.....	08
1.3. Organisation du mémoire.....	10
1.4. Notation générales.....	11
1.5. Les avantages de l'utilisation d'un système multiagents.....	12

2. Système d'aide à la conduite :

2.1. Introduction du chapitre.....	14
2.2. Définition.....	14
2.3. Les enjeux.....	14
2.4. Les prestations offertes par les ITS.....	15
2.5. Les exigences d'une application ITS.....	16
2.6. Les programme de recherche dans le domaine des ITS.....	16
2.7. Les systèmes actuellement sur le marché.....	17
2.8. Techniques pour la perception de l'environnement.....	18
2.8.1. Capteurs actifs.....	18
2.8.1.1. Le RADAR.....	18
2.8.1.2. Le LIDAR.....	19
2.8.2. Capteurs passifs.....	19
2.9. Avantage de l'utilisation de la vision.....	20
2.10. Les apports de la stéréovision.....	21
2.11. Détection des obstacles routiers par stéréovision.....	22
2.12. Conclusion du chapitre.....	23

3. Principes théorique de la stéréovision :

3.1. Introduction du chapitre.....	25
3.2. Géométrie du capteur stéréoscopique.....	25
3.2.1. Droites épipolaires.....	26
3.2.2. Configuration géométrique idéale.....	26
3.3. Calibrage.....	27
3.3.1. Paramètres extrinsèques.....	27
3.3.2. Paramètres intrinsèques.....	27
3.4. Mise en correspondance stéréoscopique.....	28

3.4.1. Méthodes à base de primitives.....	28
3.4.2. Les méthodes denses.....	28
3.4.2.1. Les méthodes locales.....	29
3.4.2.2. Les méthodes globales.....	30
3.4.2.2.1. Programmation dynamique.....	30
3.4.2.2.2. Techniques par réseaux de neurones.....	30
3.4.2.2.3. Techniques par algorithmes génétiques.....	31
3.4.2.3. Méthodes mixtes.....	31
3.4.3. Les contraintes.....	31
3.4.3.1. Les contraintes locales.....	32
3.4.3.1.1. La contrainte géométrique.....	32
3.4.3.1.2. La contrainte épipolaire.....	32
3.4.3.1.3. La contrainte d'intervalle de disparité [dmin,dmax]	32
3.4.3.2. Les contraintes globales.....	32
3.4.3.2.1. La contrainte d'unicité.....	32
3.4.3.2.2. La contrainte d'ordre.....	33
3.4.3.2.3. La contrainte de continuité de la disparité.....	33
3.5. Triangulation géométrique.....	33
3.6. Conclusion du chapitre.....	35
4. Mise en correspondance stéréoscopique :	
4.1. Introduction du chapitre.....	37
4.2. Principe.....	38
4.3. Une implantation optimisée.....	40
4.4. Application des prétraitements.....	41
4.4.1. Choix de la taille de la fenêtre de corrélation.....	41
4.4.2. Choix du critère de corrélation.....	42
4.4.3. Agrégation de la corrélation.....	42
4.4.4. La détection des zones peu texturées ou de textures uniformes.....	43
4.5. Calculs de disparités par système multiagents.....	44
4.5.1. Construction de la matrice de mise en correspondance.....	44
4.5.2. Seuillage des scores de corrélation.....	47
4.5.3. La détection des zones occultées.....	47
4.5.4. Initialisation des agents.....	47

4.5.5. Application des contraintes globales.....	48
4.5.5.1. Mise en œuvre.....	49
4.5.5.2. Recherche de l'optimum.....	49
4.5.5.3. Application de la contrainte d'unicité.....	50
4.5.5.4. Application de la contrainte d'ordre.....	50
4.5.5.5. La validation d'appariements.....	51
4.5.5.6. Etat final du système.....	51
4.6. Construction de l'image de disparité.....	52
4.7. Affinage des résultats.....	52
4.8. Tests et résultat.....	52
4.8.1 Evaluation de la robustesse de la méthode.....	53
4.8.2. Evaluation des performances en temps d'exécution.....	54
4.8.3. Comparaison de notre méthode avec celle de la programmation dynamique..	55
4.9. Conclusion du chapitre.....	56
5. Détection et suivi des obstacles pour la navigation :	
5.1. Introduction du chapitre	58
5.2. Travaux antérieurs.....	58
5.2.1. Rectification homographique.....	58
5.2.2. Mise en correspondance des images stéréoscopique.....	59
5.3. Méthode proposée.....	61
5.4. Volume tronqué de la route.....	63
5.5. Description détaillée du fonctionnement.....	64
5.5.1. Construction de la carte de navigation.....	64
5.5.2. Segmentation et détection d'obstacles.....	67
5.5.3. Appariement des obstacles.....	68
5.5.4. Prise en compte du roulis et du dévers.....	70
5.6. Résultats expérimentaux.....	70
5.7. Conclusion du chapitre.....	71
6. Conclusion :	
6.1. Conclusion générale.....	73
6.2. Perspectives.....	74
i. Bibliographie :	75
ii. Annexe : application et quelques résultats.....	79

Table des figures :

Figure 1.1 : Synoptique globale de notre système de vision stéréoscopique.....	9
Figure 2.1 : Le RADAR.....	18
Figure 2.2 : Le LIDAR.....	18
Figure 2.3 : Le projet UTA du groupe de recherche Daimler-Benz.....	20
Figure 3.1 : Configuration particulière du système de prise de vue.....	25
Figure 3.2 : Localisation tridimensionnelle dans le model stéréoscopique idéal.....	32
Figure 4.1 : Illustration de l'influence de la profondeur sur la disparité.	36
Figure 4.2 : Principe de mise en correspondance par corrélation.....	38
Figure 4.3 : Configuration idéale du système stéréoscopique (lignes épipolaires parallèles)	39
Figure 4.4 : Exemple d'une représentation graphique de la carte de disparité.....	40
Figure 4.5 : Synoptique globale de l'algorithme de mise en correspondance.....	40
Figure 4.6 : Transformation Census sur un pixel avec une fenêtre 5x5.....	42
Figure 4.7 : Principe du moyennage WxW (les zones en rouge sont ignorées)	42
Figure 4.8 : Détection et transformation Census des zones peu texturées.....	44
Figure 4.9 : La construction de la matrice de scores optimisée S.....	45
Figure 4.10 : L'impact des contraintes sur l'ensemble des appariements.....	49
Figure 4.11 : Comportement d'un agent (états et conditions)	51
Figure 4.12 : Exemple d'images de synthèse utilisées dans le test.....	53
Figure 4.13 : Cartes de disparités calculées (de 0 en blanc a -60 en noir).....	54
Figure 4.14 : Proportion des disparités correctement estimées.....	54
Figure 4.15 : l'influence de la taille de la fenêtre de corrélation.....	55
Figure 4.16 : Carte de disparité on peu aisément constaté la propagation des erreurs.....	55
Figure 5.1 : Limitation de l'hypothèse de la route plane.....	59
Figure 5.2 : L'approche de Labayrade [Labayrade 02].....	60
Figure 5.3 : Analyse de la fonction de cumul de disparité par ligne de la carte de disparité	62
Figure 5.4 : Adaptation la taille de la zone de détection en fonction de la disparité.....	63
Figure 5.5 : Architecture globale du système.....	64
Figure 5.6 : Etapes de construction de la carte de navigation CN.....	66
Figure 5.7 : Résultat du cumul 3D de disparité par portion de ligne.....	66
Figure 5.8 : Segmentation des obstacles par des cadres englobant.....	67
Figure 5.9 : Exemple d'appariement d'obstacles.....	68
Figure 5.10 : Comportement de l'agent AS (états et conditions).....	69

Figure 5.11 : Comportement de l'agent AD (états et conditions).....	69
Figure 5.12 : Comportement de l'agent AO (états et conditions).....	70
Figure 5.13 : Séquence d'images de test dans les conditions de navigation réelle.....	70

1. Introduction

1.1. Introduction générale :

L'outil informatique a envahi tous les domaines de la vie moderne, offrant d'énormes services à l'humanité le rendant indispensable, l'un des domaines où le développement de l'informatique a connu des avancés spectaculaires est le domaine de l'automobile, la présence, de plus en plus persistante de l'informatique (qui se développe de simples calculateurs jusqu'à devenir de véritables ordinateurs de bord) dans ce domaine a donné naissance au Intelligent Transportation System ou **ITS** qui couvre désormais un vaste terrain de recherche sur l'intégration de systèmes faisant appel à l'informatique dans le domaine de l'automobile, et regroupe toutes les techniques mises en œuvres pour offrir une aide aux conducteurs humains afin de prévenir au maximum l'avènement d'accidents graves.

On constate d'ailleurs que les équipements informatiques embarqués occupent de plus en plus de place dans la voiture moderne offrant une gamme de service qui ne cesse de s'élargir de jour en jour, allant de la sécurité au confort.

Cette tendance est motivée par la complexité croissante de l'environnement routier et le besoin d'amélioration de la sécurité routière explique un nombre important de travaux sur l'aide à la conduite, la voiture moderne devra aider le conducteur à mieux comprendre son environnement pour l'aider à anticiper ses décisions mais sans jamais le déresponsabiliser, la voiture devra donc être capable de percevoir mieux l'environnement que ne peut le faire l'être humain et pallier ses déficiences.

Le travail présenté dans cette thèse concerne l'aide à la conduite, nous nous sommes intéressés plus particulièrement à la perception de l'environnement à l'avant du véhicule par stéréovision embarquée afin d'extraire les informations nécessaires à la mise en œuvre d'un système d'évitement d'obstacles.

1.2. La problématique :

Parmi toutes les technologies de perception qui pourraient équiper les véhicules de demain, la stéréovision promet des caractéristiques très attractives, pouvant servir à plusieurs prestations, la stéréovision présente des caractéristiques techniques tout à fait comparables à d'autres capteurs pour un prix nettement inférieur, d'autre part, les techniques de vision permettent des prestations qu'aucun autre capteur actuel ne permet (comme différencier un piéton d'un véhicule).

Notre travail concerne principalement la mise au point d'un système robuste d'aide à la conduite par stéréovision embarquée, ce système est chargé de recouvrir la structure 3D de la scène, et d'extraire l'information nécessaire pour la détection d'éventuels obstacles routiers, et d'effectuer leurs suivi en temps réel.

Connaissant le modèle de projection de chacune des deux caméras, il s'agit de calculer les coordonnées 3D par triangulation d'un ensemble de points appartenant à la scène observée à partir de leurs projections sur les deux plans images, pour y parvenir, nous devons faire face aux problèmes suivants (figure 1.1) :

- 1) Estimer le modèle de projection de chacune des deux caméras, c'est à dire calibrer le capteur stéréoscopique.
- 2) Résoudre le problème de la mise en correspondance, ceci revient à associer un pixel dans une image donnée à sa contrepartie dans l'autre image, correspondant à la projection du même point de la scène réel afin de concevoir une carte de profondeur dense, ce processus est réalisé par la mise en œuvre d'un système multiagents à base de threads légers chargés de calculer la profondeur d'un maximum de pixels (disparité) pour ensuite estimer leurs coordonnées réelles par simple triangulation, l'introduction de cette technique permet notamment de répondre aux exigences nécessaires à ce genre d'application critique devant être à la fois robustes et répondre à des contraintes temporelles strictes.
- 3) Exploiter l'information existante sur l'image de profondeur « disparité » afin de détecter la chaussée praticable, et l'existence d'éventuels obstacles et estimer leurs distances par rapport au dispositif de prise de vue, cette tâche est réalisée pas une deuxième couche d'agents de nature plus évoluée que la précédente vu la complexité des décisions a prendre, cette couche d'agents servira d'intermédiaire entre la couche d'agents « de base » responsable de la construction de la carte de disparité et l'utilisateur humain en interprétant cette carte.

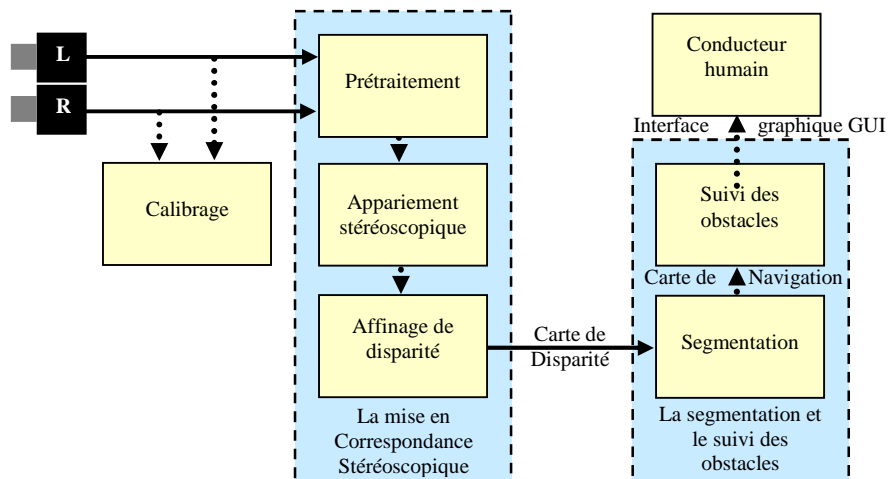


Figure 1.1 : Synoptique globale de notre système de vision stéréoscopique.

1.3. Organisation du mémoire :

Dans ce mémoire nous nous intéressons particulièrement à l'application de la stéréovision dans les systèmes d'aide à la conduite, on développe et on propose l'utilisation de la vision stéréoscopique embarquée afin de percevoir la profondeur de la scène observée par simple triangulation pour détecter et signaler la présence d'obstacles dans un contexte routier.

On commence tout d'abord dans le premier chapitre par exposer les différents systèmes d'aide à la conduite tout en mettant l'accent sur les apports des techniques basés sur la vision dans ces systèmes, on donnera les spécifications et les exigences des applications d'aide à la conduite auxquelles nous nous intéressons et les causes de l'engouement qu'a connu la communauté scientifique pendant la dernière décennie, pour finir par exposer les différents dispositifs de détection d'obstacles routiers par vision embarquée.

Au cours du deuxième chapitre, on expose les fondements théorique et géométrique de la vision stéréoscopique binoculaire, ce qui nous permet de reconstruire, partiellement ou entièrement une scène 3D à partir de deux images prises sous des angles légèrement différents, ensuite nous enchainons par l'étude du problème central de la stéréovision qui est celui de la mise en correspondance des pixels de l'image gauche avec leurs correspondants de l'image droite, la reconstruction de la scène 3D est ensuite assurée par l'exploitation des décalages ou « disparités » associés aux pixels appariés, le présent chapitre décrit aussi les traitements annexes à la stéréovision, particulièrement la calibration qui consiste à trouver la fonction qui défini la relation entre la projection d'un objet sur les deux plans image du stéréoscope en 2D et sa position dans un repère réel 3D et la rectification des images acquises par le stéréoscope.

Pour ensuite exposer la technique de mise en correspondance choisie, cette technique est basée sur le pixel, et utilise un critère de corrélation photométrique, on a également essayé d'expliquer les différentes contraintes utilisées pour réduire la complexité du problème.

Dans le troisième chapitre on construit notre propre algorithme de mise en correspondance stéréoscopique qui est la phase la plus importante pour deux raison, premièrement, c'est une étape qui nécessite des traitements très complexes au risque d'avoir une explosion combinatoire du problème, deuxièmement, cette étape est la base du processus de stéréovision et en dépendra la robustesse du système final.

Au cours de ce chapitre, on tente de répondre aux contraintes de temps réel et de robustesse spécifique à ce genre de système, on propose l'introduction d'une nouvelle technique générique d'appariement stéréoscopique, on propose l'utilisation d'un système multiagents composé de threads légers pour l'accomplissement de cette phase, cette approche nous permet de réduire le temps de calcul vu la possibilité de paralléliser les traitements sur une architecture multicœurs ou

éventuellement sur une architecture dédiée de type GPU, tout en augmentant la robustesse de la méthode en permettant l'émergence de la solution la plus optimale à partir de l'interaction des agents entre eux et avec leurs environnements.

Ensuite on présente une mise en test de notre algorithme selon un protocole de test standard qui consiste à détecter les paramètres de temps et de robustesse sur des images stéréoscopiques de synthèse téléchargées des pages web spécialisées qui présentent la particularité de pouvoir contrôler tous les paramètres de vérité de terrain.

Dans le quatrième chapitre, en amont de la mise en correspondance stéréoscopique qui fournit une carte de disparité dense de l'environnement routier, cette information nécessite d'être traitée pour identifier et suivre les obstacles éventuellement présents, ce traitement s'appelle l'extraction d'obstacles ou segmentation des obstacles, il permet de définir la surface de la route praticable devant le véhicule, pour ensuite essayer de définir tous les obstacles présents sur l'environnement et effectuer leurs suivi en un temps réel.

Pour réaliser cette tâche on exploite l'information de profondeur offerte par la carte de disparité dense calculée précédemment, ici, on implémente une deuxième couche d'agents plus évoluée cette fois-ci, qui sera intermédiaire entre la couche de mise en correspondance et l'utilisateur humain, la tâche de ce système consiste à calculer une nouvelle image qu'on appelle carte de navigation CN qu'on introduit pour faciliter l'interprétation de la carte de disparité dense, vu que cette carte de navigation CN ne comprendra qu'une information pertinente pour notre application, à savoir si une zone de cette carte est praticable ou non, cette nouvelle image est calculée en analysant les variations dans le cumul des disparités connues ligne par ligne, et d'en déduire la surface de la route praticable, ensuite détecter et suivre l'évolution de la distance de chaque obstacle sur la scène d'une façon robuste.

Une nouvelle phase de tests est réalisée sur des séquences d'images de navigation réelle d'un véhicule dans un environnement routier, afin de vérifier le degré de fiabilité de notre approche en matière de robustesse et de temps dans des conditions de navigation réelle.

En conclusion de ce mémoire, nous présentons dans le dernier chapitre les améliorations susceptibles d'être apportées à notre approche de vision stéréoscopique, ainsi que quelques perspectives futures.

1.4. Notations générales :

Pour le reste de ce manuscrit, et pour les images gauche et droite, de taille $N \times M$, les notations suivantes sont utilisées :

- 1) Un pixel de l'image gauche est noté $p_g(i,j)$ et respectivement $p_d(f,k)$ représente un pixel de

l'image droite.

- 2) Le niveau de gris associé au pixel de coordonnées $p(i,j)$ est $I(i,j)$.
- 3) On effectue la mise en correspondance de l'image gauche vers l'image de droite, les mêmes formules peuvent être utilisées dans le cas inverse.
- 4) La fonction qui associe à chaque pixel $p_g(i,j)$ de l'image gauche une disparité (l'écart de position) d est une fonction définie par : $d : N^2 \rightarrow Z^2$ telle que $p_g(i,j) \rightarrow d(i,j) = k-j : p_d(i,k)$.
- 5) La taille de la fenêtre de corrélation F est $W \times W$.
- 6) La fonction qui quantifie la ressemblance entre deux chaînes de bits, est la distance de

$$\text{Hamming est donnée par : } D_{\text{ham}}(f_g, f_d) = \sum_{K=0}^{N_f-1} \text{sgn} |f_g^k - f_d^k|$$

$$7) \text{ La fonction sgn est définie par : } \text{sgn}(x) = \begin{cases} -1 & \text{si } x \leq 0 \\ 0 & \text{si } x = 0 \\ 1 & \text{sin on} \end{cases}$$

1.5. Les avantages de l'utilisation de paradigme multiagents (approche parallèle) :

Nos algorithmes de vision stéréoscopique intègrent une solution basée sur une approche parallèle, cette solution se justifie par :

- 1) Une séparabilité nette entre les différents composants (détection, appariement, prise de décision ...etc.) d'un système de détection d'obstacles.
- 2) Meilleure évolutivité et un cycle de maintenance aisé (possibilité de remplacement d'un composant par un autre sans pour autant affecter tout le système).
- 3) Portabilité et possibilité d'exécution distribuée grâce à l'interopérabilité entre différents systèmes opératoires offerte par les plates de formes multiagents.

2. Système d'aide à la conduite

2.1. Introduction du chapitre :

Malgré les investissements croissants dans le domaine des transports publics (bus, tramway, train électrique, ...), le véhicule personnel reste le moyen de transport qui connaît le plus de croissance à travers le monde, toute fois, cette préférence n'est pas sans conséquence, en effet on enregistre un nombre très important d'accidents qui ne cesse de croître, afin de limiter le nombre et la gravité des accidents, une attention particulière est portée sur la sécurité routière avec le durcissement des sanctions à l'encontre des chauffards, surtout que les statistiques montrent que la défaillance humaine est responsable dans 90 à 95% des accidents.

Le constat des chiffres de morts déplorés chaque année montre clairement qu'il faut encore travailler à améliorer la sécurité routière, cette amélioration passe en partie par la conception de systèmes d'aide à la conduite ITS (Intelligent Transportation System) centré sur le conducteur humain, dont les défaillances de jugement sont la source principale des accidents largement plus que les défaillances liées au véhicule, la régression de ces chiffres passe obligatoirement par le développement et la généralisation de ce genre de système.

Dans les sections suivantes, nous allons présenter quelques aspects liés à l'aide de conduite, leurs importances, leurs caractéristiques les plus marquées, on effectue aussi un tour d'horizon sur les programmes de recherche qui sont d'actualité, l'accent est mis sur les techniques de perception de l'environnement, et l'importance des techniques basées sur la vision dans ce domaine.

2.2. Définition :

Vu la diversité des domaines dont touchent les ITS, il existe plusieurs définitions, de notre part, on prend la définition suivante :

On désigne par système d'aide à la conduite ou ITS tout dispositif capable de détecter et prévenir toute situation dangereuse, cette détection est basée sur la perception de l'état du véhicule et de l'environnement dans lequel il évolue par l'intermédiaire de capteurs, pour prévenir tout accident, les ITS peuvent agir ou bien à travers des actions directes sur le véhicule pour le redresser ou à travers le conducteur lui-même en lui signalant le danger par le biais d'interface homme-machine ce qui est le cas de la plus part des systèmes en développement qui sont « centrés conducteur ».

2.3. Les enjeux :

Les statistiques montrent que plus de 10 millions de personnes dans le monde sont chaque année impliquées dans des accidents de la route, 20 à 30% d'entre elles sont sévèrement blessées, et plus de 1 300 000 sont tuées [\[Wikipedia\]](#), ce phénomène qui prend de l'ampleur et devient une

vraie catastrophe nationale en Algérie avec plus de 4000 tués déplorés chaque année sur nos routes, ces accidents sont principalement causés par la défaillance du conducteur humain.

Devant cette hécatombe il est légitime de chercher toutes les techniques susceptibles de faire régresser ces chiffres terribles, la première action à entreprendre se dirige naturellement vers le conducteur du véhicule en lui proposant de plus en plus d'outils technologique d'aide à la conduite destiné à l'orienter à prendre les bonnes décisions en temps opportun.

Dans ce domaine, les constructeurs automobiles ont tout d'abord essayé de réduire la gravité d'une collision en améliorant la sécurité en aval, ainsi, l'introduction de la ceinture de sécurité, les airbags gonflables et autres mesures ont permis de réduire énormément la proportion et la gravité des accidents entre 1960 et 1990, cependant, la sécurité passive ne suffit plus pour baisser le nombre de victimes de la route d'une façon significative, désormais la plupart des chercheurs se concentrent sur l'amélioration de la sécurité en amont de l'accident, pour se faire, tous les efforts se dirigent vers le conducteur, il est informé en temps réel de l'éventuel présence d'un danger via des interfaces Homme-Machine sur écran ou autres dispositifs, vu que l'inattention et la défaillance du jugement du conducteur sont les principales sources d'accidents, cette classe de dispositifs d'aide à la conduite peut réduire significativement le nombre et la gravité des accidents [Itay 05].

Pour cette raison, et aussi pour des raisons de confort, les acteurs de l'industrie automobile sont de plus en plus demandeurs de systèmes intelligents, cet intérêt croissant se traduit par une recherche très active dans le domaine des ITS, regroupant des domaines très distincts, comme la localisation, la planification et la perception de l'environnement.

2.4. Les prestations offertes par les ITS :

Les ITS offrent une large gamme de prestations qui ne cessent de s'élargir, ces prestations ont plusieurs buts, comme la sécurité, le confort, ou même l'exploitation optimale du réseau routier, les différentes prestations offertes par cette catégorie de techniques sont reprises dans la liste ci-dessous qui est loin d'être exhaustive :

- 1) Suivre la route.
- 2) Se tenir dans la file correcte.
- 3) Se maintenir à une distance sûre des autres véhicules.
- 4) Détecter et signaler les obstacles.
- 5) Adapter la vitesse du véhicule en fonction des conditions de trafic et des caractéristiques de la route.
- 6) Trouver l'itinéraire le plus court et le moins encombré vers une destination donnée.
- 7) S'orienter, se déplacer et stationner dans l'environnement urbain.

1.5. Les exigences d'une application ITS :

Les systèmes d'aide à la conduite disposent de certaines spécificités qui sont directement liées à la nature de l'environnement routier, ajouter à cela des contraintes liées à la nature critique de leurs applications, qui sont la robustesse et le temps réel, pour [Bertozzi 00], n'importe quel système embarqué pour une application ITS doit satisfaire certaines exigences vitales :

- 1) Le système final doit être assez robuste pour s'adapter aux différents états et changements d'environnement, de route, de trafic, d'éclairage, ainsi qu'à des conditions météorologiques variables.
- 2) Les systèmes embarqués propres aux applications ITS sont des systèmes de sûreté critique pour lesquels un haut degré de fiabilité est exigé, par conséquent, le projet doit être approfondi et rigoureux pendant toutes ses phases, de la spécification des exigences à la conception et la mise en œuvre, une phase intensive de mise à l'épreuve et de validation est donc d'une importance capitale.
- 3) Afin de rendre la technologie ITS accessible au plus grand nombre, et par conséquent rendre cette dernière attractive pour les industriels du secteur, la conception d'un équipement ITS doit intégrer dès le départ des contraintes de coût stricts, les dépenses de tout ordre (comme la consommation électrique) doivent être tenues aussi basses que possible, puisque les performances du véhicule (qu'elles soient techniques ou économiques) ne doivent pas être affectées négativement par l'utilisation du système ITS.
- 4) Le matériel et les capteurs du système doivent être compacts et discrets, ils ne doivent changer le style ou le design du véhicule.
- 5) Enfin, puisque les systèmes ITS doivent être déclenchés et contrôlés par un opérateur humain, ils ont besoin d'une interface homme-machine conviviale, simple, réactive et efficace.

2.6. Les programmes de recherche dans le domaine des ITS :

Au cours des dernières décennies une grande importance a été donnée aux ITS afin d'améliorer les conditions de sécurité, d'optimiser l'exploitation des réseaux routiers, de diminuer la consommation d'énergie pour préserver l'environnement.

Les efforts consentis pour résoudre ces problèmes ont ouvert un nouveau domaine de recherche et d'application, cette tendance est démontrée par le nombre impressionnant de programmes et de projets de recherche qui ont été lancés dans le monde, avec d'importants budgets alloués, ce qui démontre clairement l'orientation de l'industrie automobile pour l'innovation et l'équipement de plus en plus de véhicules par des ITS.

Parmi une large liste de programmes lancés sur les cinq continents, on peut citer quelque'un des plus aboutis, comme le programme IVHS (Intelligent Vehicle and Highway System) aux Etats-Unis et VICS (Vehicle Information and Communication System) au Japon.

En Europe, on peu citer PROMETHEUS (Program for an European Traffic with Highest Efficiency and Unprecedented Safety) et DRIVE (Dedicated Infrastructure for Vehicle Safety in Europe) qui étaient les premiers dans ce domaine, d'autres programmes ont été lancés comme ROADSENSE (Road Awareness for Driving via a Strategy that Evaluates Numerous Systems), RADARNET (Multifunctional Automotive Radar Network), CARSENSE (Sensing of Car Environment at Low Speed Driving), PReVENT (Preventive and Active Safety), PAROTO (Projet Anticollision Radar et Optronique pour l'Automobile), STATUE (Système Télématique destiné à l'Accroissement de la sûreté des Transports guidés Urbains et à l'Emergence de nouvelles aides à l'exploitation), ARCOS (Action de Recherche pour une Conduite Sécurisée) et PREDIT (Programme pour la Recherche, le Développement et l'Innovation dans les Transports terrestres).

2.7. Les systèmes actuellement sur le marché :

Bien que les premiers systèmes ITS on été développés depuis plus de vingt ans, mais le plus souvent ils n'étaient réservés qu'aux voitures de haute gamme de grandes marques, tandis qu'aujourd'hui, les systèmes intelligents commencent réellement à investir le marché, la preuve est le nombre croissant de marques en équipant leurs véhicules, on présente ici quelques exemples les plus populaires :

- 1) La détection de franchissement de marquage routier, parfois appelée LKA (pour Lane Keeping Assist), a pour but de prévenir les changements de voie involontaires.
- 2) La régulation de distance est un système qui maintient le véhicule porteur à distance du véhicule précédent, le plus souvent couplé au régulateur de vitesse ACC « Adaptive Cruise Control ».
- 3) La prévention de collision (en cas de collision imminente), ce genre de système est utile lorsque le véhicule précédent freine ou qu'un obstacle surgit subitement, selon le constructeur on peut trouver : Presafe pour Daimler, Pre-crash Safety System pour Lexus et Collision Warning and Brake Support pour Volvo.

2.8. Techniques pour la perception de l'environnement :

La perception de l'environnement est l'élément le plus déterminant dans la chaîne de prise de décision dans un système d'aide à la conduite, les performances finales du dispositif dépendent fortement de cette étape où la précision, la robustesse et la rapidité doivent assurer une grande fiabilité.

Afin de pouvoir acquérir la structure de l'environnement, les systèmes d'aide à la conduite sont basés sur la perception de l'environnement par l'intermédiaire de divers capteurs, la plus part des capteurs utilisés actuellement dans le domaine des ITS sont issus de la robotique mobile adaptés aux exigences d'utilisation extérieure [Bertozi 00].

L'objectif de la perception de l'environnement d'un véhicule est de détecter, localiser et suivre l'évolution d'un grand nombre d'éléments évoluant autour du véhicule, on peut distinguer deux méthodes de perception de l'environnement selon la technologie du capteur utilisé : approche active ou passive.

2.8.1. Capteurs actifs :

Les capteurs actifs auxquels sont incorporés d'un émetteur produisent leur propre énergie pour « irradier » la scène dans la bande spectrale du récepteur, les plus utilisés sont les radars et les télémètres lasers et beaucoup plus rarement les ultrasons qui restent réservés aux obstacles proches pour un détecteur de recul par exemple le « Sonar ».

Cette catégorie de capteurs est caractérisée par sa mise en œuvre facile, les mesures obtenues sont robustes et pertinentes, mais également ce genre de capteurs présente plusieurs inconvénients notamment la pollution électromagnétique (interférence entre les capteurs de même type) et le coût relativement élevé de ce genre de dispositifs.

De nombreux systèmes actifs de détection d'obstacles sont proposés dans la littérature : RADAR [Tokoro 96], LIDAR [Wang 03], [Hancock 98] nous détaillerons les deux types de capteurs actifs les plus répandus dans l'industrie automobile.

2.8.1.1. Le RADAR :

Le RADAR (figure 2.1) est un capteur actif, comme ceux de cette catégorie, il fournit une excellente mesure directe de positionnement et de vitesse, lorsqu'il se déplace, il distingue aisément les cibles en mouvement du reste de l'environnement, par contre, à faible vitesse, cette segmentation est plus hasardeuse, ce qui le rend inadapté aux prestations fonctionnant à basse vitesse, son prix élevé en fait un capteur réservé aux véhicules hauts de gamme.



Figure 2.1 : Le RADAR.

2.8.1.2. Le LIDAR :

Le LIDAR ([figure 2.2](#)) est également un capteur actif acronyme de l'expression anglo-saxonne « Light Detection and Ranging », il permet de mesurer des distances ou des vitesses en se basant sur l'analyse des propriétés d'une lumière laser émise et réfléchiée vers l'émetteur, les LIDAR généralement employés dans les applications automobiles balayent la scène de 1 à 4 faisceaux.

Chacun de ces faisceaux permet d'estimer un plan de coupe de la scène, le LIDAR propose un bon compromis entre le coût et les performances, d'autre part, il convient aussi bien à des applications à haute vitesse qu'à basse vitesse, ce capteur est cependant limité par sa fréquence d'échantillonnage, ce qui en interdit l'utilisation pour les applications de sécurité de type pré-crash [[Widman 07](#)].



Figure 2.2 : Le LIDAR

2.8.2. Capteurs passifs :

Un capteur passif est un capteur qui capte une énergie émise naturellement, ces capteurs sont majoritairement des caméras, deux approches sont alors possibles : l'utilisation d'une seule caméra ou l'exploitation de deux points de vue avec deux caméras reliées de manière rigide sur le véhicule.

L'intérêt des caméras par rapport aux capteurs actifs tient surtout dans le champ de perception accessible et par la richesse et la capacité d'interprétation et d'analyse des scènes perçues mais aussi leurs coûts relativement bas.

La difficulté de l'utilisation de la vision artificielle est le coût des algorithmes puisque l'image en elle-même ne donne pas de mesures directes de distance comme c'est le cas pour un

capteur actif, il faut donc passer par des traitements complexes d'analyse d'image, dès lors les résultats obtenus ne sont pas issus d'une mesure physique mais sont le fruit d'une suite combinatoire de calculs qui peut être une source importante d'erreur d'évaluation.

Plusieurs techniques à base de traitement et analyse d'images ont été développées pour la perception de l'environnement d'un véhicule comme dans [Thorpe 98], [Williamson 99], [Bertozzi 00], [Labayrade 02], [Bücher 03] et [Chapuis 05], tandis que d'autres techniques combinent les systèmes actifs et passifs [Kato 02].

2.9. Avantages de l'utilisation de la vision :

Les capteurs actifs effectuent les mesures (comme celle du mouvement) d'une façon plus directe et exigeant moins de ressources en calcul car devant acquérir une quantité considérablement inférieure de données pour parvenir à l'estimation voulue que le font les capteurs passifs, notamment la vision qui prédomine cette catégorie.

Les systèmes basés sur la vision utilisent une ou plusieurs caméras pour capter l'environnement, la quantité d'information présent sur l'image est un avantage, mais rend le traitement extrêmement complexe, néanmoins, la richesse de l'information fourni par la vision (qui constitue plus de 90% des données exploitées par un conducteur dans sa tâche de pilotage) en font un des moyens les plus puissants pour percevoir l'environnement et de ce fait, était largement utilisée pour traiter un grand nombre de problèmes dans le domaine des ITS ([Steux 02], [Labayrade 02]).

Les avancées récentes en informatique et en technologie de cameras ont favorisé l'utilisation de la vision dans le domaine des véhicules intelligents, les progrès du matériel informatique, en particulier le développement d'architectures mobiles, permettent de proposer des machines plus puissantes, moins dispendieuses en énergie et surtout moins encombrantes et facilement embarquables (une simple tablette PC équipée d'un puissant processeur à faible consommation électrique), à des prix accessibles.

Les systèmes basés sur la vision semblent proposer une bonne alternative aux capteurs actifs, pour des prestations similaires, et un coût moindre, ajouter à cela le fait que certaines prestations ne sont accessibles qu'aux technologies basées sur la vision, comme la détection de marquage routier ou encore la distinction des différents types d'obstacles (piéton, véhicule).

Vu que les ITS sont plus exigeants que d'autres applications en terme des contraintes temporelles (temps-réel), en effet, le temps de réponse du système se doit d'être en accord avec la vitesse de déplacement du véhicule, qui peut être élevée, le défi est d'être capable de traiter une

grande quantité de données de façon fiable en temps acceptable, avec l'avènement de nouvelles générations d'architecture machine de plus en plus puissantes, tout le défi réside dans le développement d'algorithmes qui satisfassent cette contrainte, par conséquent, c'est un domaine de recherche très actif ([Cabestaing 99], [Labayrade 02], [Issa 04], [Perez 05], [Morat 08] et [Zhang 10]).

L'environnement dans lequel évolue l'automobile est caractérisé par une grande dynamique, les conditions d'éclairage de la scène observée où les conditions météorologiques influent directement sur les performances du capteur de vision, donc, les algorithmes de traitement développés doivent être assez robustes pour s'adapter à des conditions environnementales très différentes (le soleil, l'obscurité, la pluie, le brouillard).

2.10. Les apports de la stéréovision :

La stéréovision présente des avantages très intéressants (certains de ceux cités ci-dessous ne lui étant pas spécifiques, ils sont partagés par tous les capteurs à base de vision) :

- 1) Une quantité d'informations à la fois très importante et très riche.
- 2) La possibilité de pouvoir s'appliquer d'une façon générique (s'applique à tout objet).
- 3) La reconstruction 3D sans requérir d'informations très précises sur la scène observée.
- 4) Le suivi des véhicules dans les courbes, là où le RADAR échoue.
- 5) Une grande flexibilité, dans la mesure où un capteur de stéréovision permet le traitement des deux prises de vues en parallèle.
- 6) La nature non intrusive de ce capteur (capteur passif), lui épargne les problèmes d'interférences.
- 7) Le coût relativement bas d'un système de stéréovision le rend attractif en comparaison de capteurs actifs dotés de performances adaptées à l'application.



Figure 2.3 : Le projet UTA du groupe de recherche Daimler-Benz.

2.11. Détection des obstacles routiers par stéréovision :

L'utilisation d'un capteur stéréoscopique pour la perception de l'environnement exige la mise en correspondance des pixels des différentes images, l'avantage d'analyser les images stéréo se résume dans la possibilité de détecter directement la présence d'obstacles par l'analyse des disparités.

Le projet UTA du groupe de recherche Daimler-Benz (figure 2.3) visait à développer un système intelligent (Stop & go) à l'intérieur de la ville en adoptant la stéréovision pour extraire l'information 3D en temps réel, le prototype développé [Franke 98] est par ailleurs capable de reconnaître les signes du trafic, les balises de signalisation, les piétons, les passages protégés et les lignes d'arrêt.

Le groupe de recherche de MIT aux états unis, a développé un système de stéréovision pour la détection tridimensionnelle des routes et la visualisation du trafic, qui peut tout aussi bien être adapté à d'autres applications.

Dans le système GOLD, la suppression de l'effet de perspective dans les images stéréo [Broggi 99], permet d'obtenir deux images qui ne peuvent différer l'une de l'autre qu'en présence d'un ou plusieurs obstacles où l'hypothèse de planéité de la route n'est pas respectée, dans le cas contraire, il s'agit alors d'un système détectant l'espace libre devant le véhicule.

D'une façon similaire, l'unité de recherche de l'Université de Californie [Koller 95], a développé un algorithme qui rectifie l'image gauche en utilisant le point de vue de l'image droite, détectant ainsi les obstacles à partir de l'analyse des disparités, dans cette optique, un filtre de Kalman est utilisé pour suivre les obstacles.

Des algorithmes de détection d'obstacles plus récents ont été développés dernièrement, nous pouvons citer à titre d'exemple l'algorithme de détection d'obstacles avec un stéréoscope composé de deux caméras matricielles dans le cadre du projet ARCOS [Labayrade 02] ou encore la technique de détection rapide d'obstacles basée sur un stéréoscope composé de deux caméras linéaires ([Ruichek 05]), ces techniques ont le mérite d'exploiter uniquement les propriétés géométriques de la projection de la scène sur les caméras pour pouvoir détecter un obstacle.

Une étude plus détaillée des techniques de stéréovision est présentée dans les deux chapitres suivants, on passe en revue un état de l'art en la matière afin de pouvoir choisir la technique qu'on va adapter dans travail.

2.12. Conclusion du chapitre :

Les avantages énumérés au-dessus, font du capteur de stéréovision un outil de choix pour la perception de l'environnement du véhicule et justifie son emploi dans le cadre de notre projet, pour le développement d'un capteur passif répondant à des contraintes de coûts strictes en comparaison avec la richesse de l'information qu'il procure, celui-ci a pris la forme d'un capteur de vision stéréoscopique, se composant de deux caméras et d'une unité de traitement compact et économique, que ce soit financièrement et énergétiquement.

En outre, pour diminuer la complexité intrinsèque de la stéréovision, des contraintes spécifiques au domaine sont généralement adoptées, et seront amplement discutées au cours du chapitre suivant.

3. Principes théorique de la stéréovision

3.1. Introduction du chapitre :

La stéréoscopie ou vision en relief, est un procédé qui permet de calculer les coordonnées des points dans l'espace à l'aide de deux images observées à partir de points de vue différents, pour cela, on utilise généralement un modèle simplifié pour représenter une caméra qui est le modèle sténopé ou trou d'épingle, ce processus peut être décomposé en quatre étapes :

- 1) Détermination de paramètres intrinsèques et extrinsèques du capteur stéréoscopique pour cela on effectue un calibrage.
- 2) Appariement des primitives sur l'image gauche avec leurs homologues sur l'image droite.
- 3) Triangulation géométrique et reconstruction tridimensionnelle de la scène.

Dans le présent chapitre on commence d'abord par présenter quelques notions sur la géométrie de la vision stéréoscopique pour ensuite exposer les étapes de ce processus comme énuméré au-dessus.

3.2. Géométrie du capteur stéréoscopique :

La géométrie du système de prise de vue est un élément important dans un processus de stéréovision, pour que deux caméras constituent un capteur stéréoscopique, il faut que celles-ci observent la même scène, c'est à dire qu'elles soient disposées de telle sorte que leurs axes optiques convergent vers la scène observée, en utilisant le modèle sténopé, qui stipule que tous les rayons lumineux passent par un centre optique.

Alors un point P de la scène aura deux projections, p_g de coordonnées (i,j) sur le plan d'image gauche (correspondant à la surface photosensible CCD ou CMOS dans le cas d'une caméra numérique) et p_d de coordonnées (i,k) sur le plan d'image droite (p_g et p_d sont appelés des points homologues ou stereoscorrespondants), on peut ainsi déterminer les équations de la droite $O_g p_g$ passant par le centre optique O_g de la caméra gauche et le point-image p_g , et de la droite $O_d p_d$ passant par le centre optique O_d de la caméra droite et le point-image p_d , l'intersection de ces deux droites donne, par triangulation géométrique, la position du point P de la scène réelle.

Afin de pouvoir calculer cette intersection et donc de déterminer la position de P , il faut exprimer les deux équations des droites $O_g p_g$ et $O_d p_d$ dans un même repère, on choisit en général un repère associé à la scène, appelé repère de référence.

Il est donc nécessaire de connaître les relations permettant de passer du point P de l'espace 3D, exprimé dans le repère de référence, vers les points-image p_g et p_d , exprimés dans les repères 2D liés aux images gauche et droite (en pixels), la détermination de ces relations s'effectue lors de la phase de calibrage du système de prise de vue stéréoscopique.

3.2.1. Droites épipolaires :

La notion de droite épipolaire joue un rôle très important en stéréovision, elle permet de réduire l'espace des solutions lors de la recherche des appariements, une droite épipolaire décrit le lieu des points d'une image pouvant correspondre à un même point de l'autre image (figure 3.1), il y a donc, pour chaque point de l'image gauche, une droite épipolaire dans l'image droite et, réciproquement, pour chaque point de l'image droite, il y a une droite épipolaire dans l'image gauche, de plus, deux points d'une droite épipolaire d'une image ont leurs correspondants, s'ils existent, sur une droite épipolaire de l'autre image, ces deux droites sont appelées droites épipolaires conjuguées.

3.2.2. Configuration géométrique idéale :

Les droites épipolaires jouent un rôle fondamental en vision stéréoscopique comme exposé précédemment, en effet, lorsqu'on cherche pour un point d'une image un correspondant dans l'autre image, on peut limiter cette recherche le long de la droite épipolaire correspondante, il est donc important d'avoir les expressions mathématiques les plus simples pour ces droites épipolaires.

Il existe une configuration particulièrement intéressante permettant d'avoir des droites épipolaires parallèles avec les lignes des images, cette configuration, où les droites épipolaires conjuguées sont confondues, est obtenue quand les axes des deux caméras sont parallèles et quand la droite O_gO_d reliant les centres optiques est confondue avec les axes horizontaux des deux caméras (figure 3.1), dans ce cas, les deux épipôles se trouvent rejetés à l'infini.

Dans le reste de ce document, on ne prend en compte que le modèle idéal, l'utilisation de ce modèle particulier nous affranchira de l'étape de rectification coûteuse, et facilitera grandement notre algorithme de mise en correspondance.

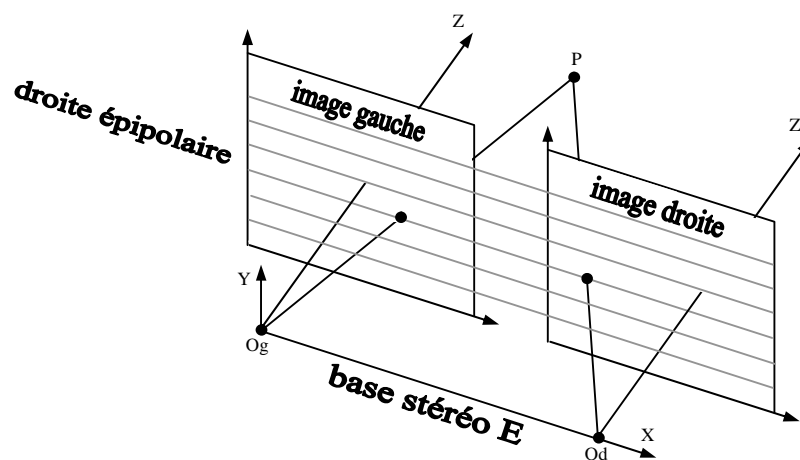


Figure 3.1 : Configuration particulière du système de prise de vue.

3.3. Calibrage :

Le calibrage d'un capteur stéréoscopique consiste à déterminer les transformations géométriques permettant de passer d'un point de l'espace 3D, exprimé dans un repère de référence lié à la scène observée, vers ses projections sur les images gauche et droite dans les repères 2D liés aux images (exprimé en pixel).

Pour cela, il faut disposer de paramètres intrinsèques et de paramètres extrinsèques, alors la transformation associée à chaque caméra est composée de deux matrices, la première, déduite des paramètres extrinsèques, permet le passage du repère de référence au repère lié à la caméra, la deuxième matrice, calculée à partir des paramètres intrinsèques, définit le passage d'un point de l'espace 3D, exprimé dans le repère lié à la caméra, vers sa projection sur l'image, exprimée dans le repère 2D lié à l'image, pour déterminer ces transformations, plusieurs techniques de calibrage ont été proposées dans la littérature par [Tsai 87] et [Zhang 00].

3.3.1. Paramètres extrinsèques :

Les paramètres extrinsèques définissent la géométrie du capteur stéréoscopique comme exposé en haut, ils comprennent en particulier la position et l'orientation exacte de chaque caméra par rapport à la scène, les paramètres extrinsèques du modèle expriment la transformation liée à la pose de la caméra par rapport au repère choisi.

3.3.2. Paramètres intrinsèques :

Outre les paramètres extrinsèques, d'autres critères doivent être définis précisément afin d'optimiser le système de vision en fonction de l'application destinataire, il s'agit des paramètres correspondant aux caractéristiques propres des caméras :

- 1) Taille et résolution des capteurs (généralement un matrice CCD ou CMOS).
- 2) Distance entre les centres optiques (base stéréo).
- 3) Distance focale des objectifs (spécifique a chaque camera).

Lorsque les paramètres extrinsèques et intrinsèques du capteur sont déterminés, ceux-ci sont utilisés dans la phase de mise en correspondance, et notamment pour le calcul de la position 3D de chaque pixel comme démontré dans la [section 3.5](#).

3.4. Mise en correspondance stéréoscopique :

Avant de pouvoir effectuer la reconstruction 3D de la scène à partir de la paire d'images stéréoscopiques, on doit d'abord être en mesure de mettre en relation les points de l'image gauche

avec leurs correspondants de l'image droite, l'étape clé du processus de stéréovision est donc la mise en correspondance des points des deux images.

Divers primitives images sont utilisables pour effectuer cette tâche, on peut classer les méthodes de mise en correspondance selon la nature des primitives et leurs attributs utilisés pour la mise en correspondance en deux grandes classes de méthodes :

3.4.1. Méthodes à base de primitives :

Les méthodes à base de primitives ou éparses, sont généralement utilisées pour localiser des objets dans une scène en utilisant des indices visuels de haut niveau tels que les points d'intérêt, les contours, les régions, etc. L'utilisation de telles primitives permet de réduire la combinatoire du problème de mise en correspondance, et ainsi le coût calculatoire de la recherche des appariements.

De plus, le caractère discriminant de ces primitives augmente les chances d'aboutir à de bons appariements grâce aux différents attributs qu'on peut leur associer comme le niveau de gris ou encore la forme géométrique, les méthodes à base de primitives fournissent des cartes de disparité éparses qui peuvent être complétées pour obtenir une carte de disparité dense en utilisant des techniques d'interpolation [Ohm 98].

3.4.2. Les méthodes denses :

Dans cette catégorie on utilise le pixel comme primitive de mise en correspondance des deux images stéréoscopiques et en résulte une carte de disparité dense, l'appariement est basé sur l'utilisation de critère de ressemblance photométrique (généralement le niveau de gris) pour effectuer la corrélation entre une fenêtre de pixels autour du pixel candidat à l'appariement, puis on cherche dans l'autre image le pixel dont la fenêtre, de même taille que la précédente satisfait le mieux au critère de ressemblance adapté [Chen 99] et [Zhang 00].

Cette méthode permet d'obtenir une carte de disparité dense et donc d'avoir une information riche de la profondeur de la scène observée, c'est d'ailleurs la méthode adaptée pour le reste de nos recherches.

Les objectifs de l'état de l'art que nous présentons sont de passer en revue et de caractériser les éléments théoriques et pratiques constituant la mise en correspondance stéréoscopie dense pour trouver celle qui se prête le mieux à l'intégration d'une méthode d'appariement stéréoscopique pour un système de détection d'obstacles sujet du chapitre suivant.

La difficulté réside dans le choix d'un algorithme qui permet d'obtenir de bons résultats notamment en ce qui concerne l'appariement des régions peu texturées ou occultées, tout en

essayant de minimiser les erreurs d'appariements tout en restant assez dynamique pour des traitements temps réel.

Scharstein et Szeliski [Szeliski 02] présentent un état de l'art assez exhaustif repris et enrichi dans [Chambon 05], un état de l'art qui compare les performances des différents algorithmes présents dans la littérature, ce qui nous a aidés dans le choix de la méthode à adopter.

Les auteurs constatent l'existence d'un grand nombre de méthodes d'appariement de pixels, qui peuvent être classifiées en deux grandes catégories: les méthodes locales et les méthodes globales, d'autres méthodes qui utilisent un mélange entre les deux en même temps appelées algorithmes mixtes ce qui est le cas de notre algorithme développé dans le chapitre suivant.

3.4.2.1. Les méthodes locales :

Différents critères de corrélation local (numériques pour la plus part du temps) ont été proposés comme critère de ressemblance entre deux fenêtres de pixels, une prise autour d'un pixel de l'image gauche et l'autre prise autour d'un pixel candidat à l'appariement pris sur l'image droite, ce traitement est par principe local car la meilleure disparité est recherchée indépendamment pour chaque pixel, voici quelques critères les plus utilisés :

- 1) **NCC** : Normalized Cross-Correlation, le critère le plus populaire, mais aussi le plus long à calculer, c'est la somme des produits entre les termes correspondants des fenêtres de corrélation, normalisée par le produit des moyennes quadratiques calculées pour chacune des fenêtres.
- 2) **ZNCC** : Zero mean Normalized Cross Correlation, la méthode ressemble à NCC mais on fait une corrélation croisée entre les écarts à la moyenne. C'est aussi un critère invariant aux variations uniformes de luminance dans une des images.
- 3) **SSD** : Sum of Squared Differences, ce qui correspond à la somme des différences quadratiques entre termes correspondants des deux fenêtres.
- 4) **ZSSD** : Zero mean Sum of Squared Differences, la méthode est identique à SSD sauf qu'au préalable, pour chaque fenêtre de corrélation, la moyenne des termes est d'abord calculée et soustraite à chaque terme : la différence est donc faite entre les écarts à la moyenne des termes, et non entre les termes eux-mêmes.
- 5) **SAD** : Sum of Absolute Differences, c'est la somme de la valeur absolue des différences entre termes correspondants des deux fenêtres.
- 6) **ZSAD** : Zero mean Sum of Absolute Differences, c'est une méthode semblable à SAD, mais on fait la différence entre écarts à la moyenne, c'est donc un critère invariant aux variations uniformes de luminance dans une des images.

3.4.2.2. Les méthodes globales :

Les méthodes globales utilisent des contraintes non-locales afin d'augmenter la robustesse dans les régions de l'image où l'appariement « local » n'est pas pertinent notamment dans des zones de texture uniforme, ou à cause des occlusions, par contre l'inconvénient majeur de ces méthodes est la plus grande complexité des calculs puisqu'il faut comparer le nombre d'appariements présents, généralement ces méthodes sont représentées sous forme de problème d'optimisation, cherchant à trouver les disparités qui minimisent un coût global de tous les appariements, appartiennent à ce groupe les méthodes de :

3.4.2.2.1. Programmation dynamique :

Le problème de l'appariement peut être vu comme un problème de recherche d'un chemin optimal dans un graphe dont les nœuds sont les couples de lignes candidates à l'appariement, on établit un graphe représentant tous les appariements possibles, l'objectif est donc de trouver dans le graphe un chemin de coût optimal qui respecte la contrainte d'ordre et d'unicité.

D'autres auteurs comme [\[Birchfield 98\]](#) et [\[Bobick 99\]](#) ont utilisé la programmation dynamique pour effectuer la mise en correspondance dense, c'est à dire en utilisant les pixels comme primitives d'appariement.

L'utilisation de la programmation dynamique fournit généralement de bons résultats, mais au prix d'un grand nombre de calculs qui rend les algorithmes plus ou moins lents suivant la complexité des images et la puissance du matériel disponible, aussi, cette technique présente le risque de propager les erreurs d'appariement car les chemins sont construits pas à pas en examinant les sommets d'une manière locale.

3.4.2.2.2. Techniques par réseaux de neurones :

De nombreux auteurs ont utilisé les réseaux de neurones pour traiter le problème de la mise en correspondance stéréoscopique, ici on considère le problème de mise en correspondance comme un problème d'optimisation, d'abord une fonction objective représentant les contraintes du problème est construite de telle façon que sa valeur minimale correspond aux bons appariements, un réseau de neurones est ensuite utilisé pour minimiser cette fonction objective, chaque neurone du réseau représente une possibilité d'appariement entre une ligne de l'image gauche et une ligne de l'image droite, la fonction objective est formulée à partir de contraintes globales.

Chaque neurone du réseau représente une possibilité d'appariement entre une primitive de l'image gauche et une primitive de l'image droite. Nasrabadi et Choo ont utilisé cette approche pour

appairer les points d'intérêt [Nasrabadi 92], la fonction objective est formulée à partir des contraintes d'unicité et de continuité de la disparité.

3.4.2.2.3. Techniques par algorithmes génétiques :

Holland [Holland 75] est le premier à avoir exposé des algorithmes génétiques fondés sur la sélection naturelle et les opérations génétiques du monde du vivant, les algorithmes génétiques ou évolutionnistes sont des techniques stochastiques d'optimisation globale d'une fonction objective qui est construite à partir des contraintes d'unicité, d'ordre et de continuité de la disparité.

Ils consistent à faire évoluer une population d'individus à l'aide d'opérateurs qui sont la sélection, le croisement et la mutation, ces opérateurs correspondent à des tirages aléatoires d'une façon itérative pour obtenir une population répondant au mieux à ces contraintes.

3.4.2.3. Méthodes mixtes :

Il existe aussi des méthodes qui correspondent à une combinaison d'une méthode locale et d'une méthode globale et que nous appelons méthodes mixtes, généralement on combine une méthode de corrélation de pixels avec une fonction d'optimisation sur une ligne entière.

L'un des exemples de cette approche est l'algorithme de Hong et Chen dans [Hong 04] dont la méthode a obtenu les meilleurs résultats parmi les méthodes mixtes dans l'évaluation de Scharstein et Szeliski [Szeliski 02].

3.4.3. Les contraintes :

Seule l'utilisation de critères de ressemblance photométrique n'est pas suffisante pour établir des appariements sans ambiguïté, vu qu'un pixel de l'image gauche peut ressembler à plusieurs pixels de l'image droite et réciproquement, pour lever les ambiguïtés d'appariement et réduire du même coup le nombre de tests d'appariements inutiles, on exploite un certain nombre de contraintes de type locale et globale reprises notamment dans [Mayer 03] et [Wei 04].

Les buts de l'utilisation de ces contraintes sont :

- 1) Limiter la combinatoire du problème de l'appariement.
- 2) Limiter le nombre de correspondants potentiels pour chaque pixel.
- 3) Choisir entre plusieurs correspondants potentiels.
- 4) Eliminer, le cas échéant, les fausses correspondances.

3.4.3.1. Les contraintes locales :

Les contraintes locales sont exprimées en fonction des propriétés locales des pixels appliquées à chaque couple de pixels individuellement sans faire intervenir des comparaisons avec d'autres couples, leur utilisation permet de réduire significativement la combinatoire du problème, dans notre procédure de mise en correspondance on utilise trois contraintes locales :

3.4.3.1.1. La contrainte géométrique :

Cette contrainte découle de la nature même du capteur stéréoscopique qui stipule que le correspondant d'un pixel de l'image gauche s'il existe, se trouve obligatoirement à une coordonnée plus petite, donc à gauche, sur l'image de droite.

3.4.3.1.2. La contrainte épipolaire :

Liée à la configuration géométrique du capteur stéréoscopique, les droites épipolaires conjuguées et parallèles (figure 1) se caractérisent par le fait que le correspondant d'un point s'il existe, se trouve exclusivement sur la même ligne dans l'autre image, ce qui permet de limiter la recherche à une droite au lieu de la totalité de l'image comme utilisé dans [Zhang 96].

3.4.3.1.3. La contrainte d'intervalle de disparité [dmin,dmax] :

Utilisée lorsque l'on connaît les limites maximales et minimales des distances utiles à l'application envisagée, la distance étant inversement proportionnelle à la disparité, la procédure de mise en correspondance ne considère alors que les couples de pixels ayant des disparités comprises entre une valeur minimale et une valeur maximale ([Koschan 96]).

3.4.3.2. Les contraintes globales :

Les contraintes globales s'appliquent à l'ensemble des appariements possibles obtenus après l'exploitation de contraintes locales réelles permettent de vérifier la compatibilité entre les couples afin de réduire les ambiguïtés de mise en correspondance ce qui augmente la robustesse de la méthode.

3.4.3.2.1. La contrainte d'unicité :

En stéréovision, un pixel de l'image gauche a au plus un correspondant dans l'image droite, et inversement, cette contrainte est utilisée dans tous les algorithmes globaux.

C'est une contrainte unaire, elle est définie par : Si $p_g(i,j_1) + d(i,j_1) = p_d(i,k)$ alors $\forall k \neq j$ $p_g(i,j_2) + d(i,j_2) \neq p_d(i,k)$.

3.4.3.2.2. La contrainte d'ordre :

La contrainte d'ordre impose que l'ordre des pixels de l'image gauche le long de la droite épipolaire doit être le même que celui de leurs correspondants de l'image droite, on peut trouver un exemple de l'application de cette contrainte dans [Szeliski 02].

C'est une contrainte binaire, elle est définie par : Si $p_g(i,j_1) + d(i,j_1) = p_d(i,k_1)$ et $p_g(i,j_2) + d(i,j_2) = p_d(i,k_2)$ alors $(j_1-j_2)(k_1-k_2) \geq 0$.

3.4.3.2.3. La contrainte de continuité de la disparité :

Cette contrainte suppose que la scène est formée d'un ensemble d'objets physiques dont la structure est représentée par des ensembles de pixels homogènes avec des niveaux de gris presque identiques, les points voisins dans l'espace ont des disparités voisines sur les images ce qui permet de déduire leurs disparités par propagation des disparités des pixels voisins déjà correctement appariés sans devoir les recalculer [Ohta 85] et [Mohan 89].

3.5. Triangulation géométrique :

Après la détermination de la disparité des pixels, on procède alors aux calculs de triangulations géométriques qui sont la finalité du processus de mise en correspondance, et permet de déterminer la position de chaque pixel dans l'espace.

Cette procédure utilise un modèle géométrique du système de prise de vue, décrivant la formation des images sur les capteurs, considérons la configuration particulière du stéréoscope présentée précédemment (figure 2.3), soient $p_g(i,j)$ et $p_d(i,k)$ les coordonnées images du point P sur les plans images gauche et droit, dans les repères relatifs aux caméras (en pixels).

Comme décrit précédemment les droites épipolaires sont parallèles, alors les ordonnées des deux pixels sont équivalentes, l'écart de position qui est égal à la différence des abscisses des deux pixels est appelé disparité entre les deux pixels homologues p_g et p_d il est notée : $d = j - k$.

Dans cette section, nous décrivons le modèle mathématique d'une caméra, à savoir la fonction de transfert entre les coordonnées pixel (i,j) d'un point du plan image et les coordonnées 3D (X,Y,Z) exprimées dans le repère de référence.

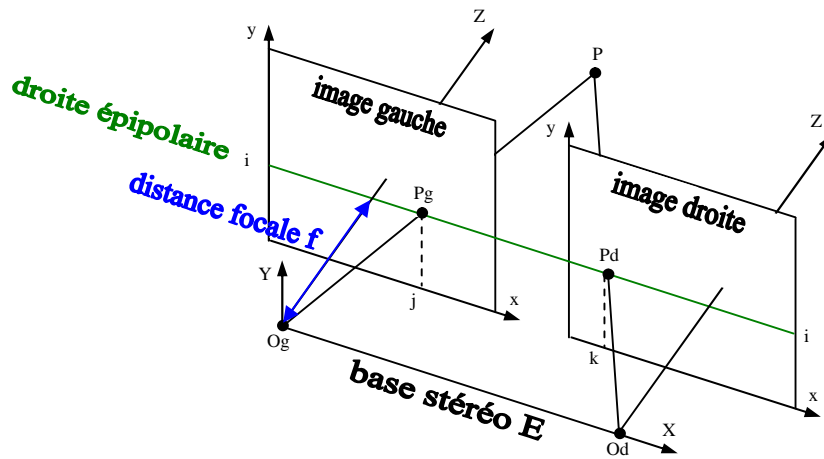


Figure 3.2 : Localisation tridimensionnelle dans le modèle stéréoscopique idéal.

Alors les coordonnées du point P dans l'espace 3D exprimé dans le repère de la scène (O_g, X, Y, Z) peuvent être déterminées par les équations suivantes :

$$X_p = Z_p \frac{x}{f}$$

$$Y_p = Z_p \frac{y}{f}$$

$$Z_p = \frac{f \cdot E}{d_x}$$

Pour effectuer ce calcul on utilise les paramètres intrinsèques des deux cameras, alors E est la base stéréo, c'est-à-dire la distance entre les centres optiques O_g et O_d et f est la distance focale des objectifs k_i et k_j en pixels/mm les facteurs de discrétisation en colonne et en ligne vu que sur le plan image, les coordonnées sont exprimées en unité pixel.

$$x = \frac{i}{k_i}$$

$$y = \frac{j}{k_j}$$

$$d_x = \frac{d}{k_i}$$

Dans notre application, on s'intéresse surtout à l'estimation de la distance (profondeur) d'un obstacle par rapport au dispositif de prise de vue qui correspond à :

$$Z_p = \frac{f \cdot E}{\frac{d}{k_i}} = \frac{k_i \cdot f \cdot E}{d}$$

3.6. Conclusion du chapitre :

La stéréovision est une technique très intéressante qui permet de recouvrir fidèlement la profondeur d'une scène à partir de deux images d'une façon très riche, cette technique exploite l'information de disparité pour calculer par simple triangulation la position 3D de tous les points de la scène observée par le stéréoscope, malgré les nombreuses techniques proposées dans la littérature, ce problème est toujours le centre d'intérêt de nombreux chercheurs dans le domaine de la vision artificielle.

4. Mise en correspondance stéréoscopique

4.1. Introduction du chapitre :

La technique fondamentale mise en œuvre en stéréovision est un processus qui, à partir de deux images d'une même scène prises de positions légèrement décalées, permet de retrouver la structure 3D de la scène observée, la reconstruction 3D d'une scène est assurée par l'exploitation conjointe des propriétés colorimétriques de la scène et des caractéristiques géométriques des caméras, cette tâche est connue sous le nom de la mise en correspondance stéréoscopique.

La mise en correspondance stéréoscopique, aussi appelée appariement, est le processus qui fourni, pour chaque point d'une image de la paire, son correspondant dans l'autre image, c'est une étape clé du processus de reconstruction 3-D (figure 4.1) ([Boguslaw 09]).

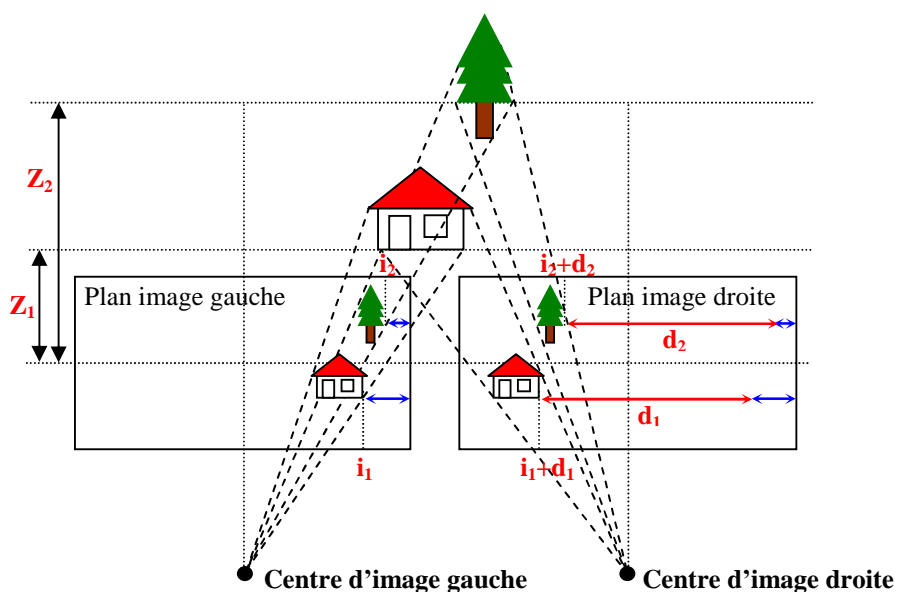


Figure 4.1 : Illustration de l'influence de la profondeur sur la disparité.

Comme exposé dans le chapitre précédent, différentes entités images sont utilisables pour la mise en correspondance entre deux images stéréo : le pixel, le segment, la chaîne de contour...ici, nous nous intéressons à l'appariement de pixels, car c'est la seule approche qui donne une image de disparité suffisamment dense et donc un maximum d'informations sur la scène à traiter.

L'appariement de ces images pour des applications de navigation et d'évitement d'obstacles peut rarement être effectués à des cadences suffisantes, cette limitation est essentiellement liée à la nature combinatoire du problème, mais dû aussi à l'utilisation de caméras capables d'acquérir des images de haute résolution, sans contraintes particulières, une mise en correspondance point à point des images n'est pas une solution optimale.

Dans ce chapitre, nous présentons une nouvelle méthode, robuste et rapide, pour résoudre le problème de la mise en correspondance stéréoscopique de pixels, notre méthode est basée sur

l'utilisation d'une approche multiagents afin de distribuer la résolution du problème sur un ensemble d'entités légères et autonomes.

L'idée à la base est l'implémentation d'un système multithreads dans le domaine de la vision, précisément pour l'appariement des images stéréoscopiques constituées d'un ensemble d'agents légers qui interagissent dans et au travers d'un environnement discret représenté par plusieurs matrices (l'image est une matrice de pixels) dans notre cas, afin de faire émerger de leurs comportements individuels simples, le meilleur appariement de pixels possible, où l'utilisation de cette approche permet d'assurer une exécution distribuer des traitements et profiter pleinement de la puissance offerte par les processeurs multicœurs ou une architecture dédié comme un GPU ce qui est intéressant pour tendre vers des applications temps réel.

La conception de cette méthode a donc été effectuée sur deux plans : le comportement individuel de chaque agent qui tend à apparier localement le maximum de pixels, en utilisant des paramètres de corrélation photométrique et des contraintes locales.

Le comportement collectif du système qui permet d'appliquer des contraintes globales en profitant de la possibilité d'interaction afin d'augmenter la robustesse là où l'appariement local n'est pas pertinent notamment dans des zones de texture uniforme, ou à cause des occlusions.

Afin d'être évaluée, cette méthode est appliquée sur des images de synthèses dont on connaît la structure de l'espace 3D observé « vérité de terrain », les expériences montrent que la méthode proposée donne de bons résultats tant au niveau de la robustesse qu'au niveau cadence de traitement.

4.2. Principe :

Un point P de la scène, visible dans les deux images se projette dans l'image gauche en p_g de coordonnées (i,j) et dans l'image droite en p_d de coordonnées (i,k) , on dit que les points p_g et p_d sont homologues, c'est-à-dire qu'ils sont stéréocorrespondent, car ils représentent le même point de la scène, alors la mise en correspondance consiste, à partir de deux images, gauche et droite, à retrouver les couples de points qui se correspondent, la différence de position de ces points est appelée disparité, elle est utilisée afin de calculer leurs position 3D par simple triangulation.

La mise en correspondance par corrélation représentent la majorité des travaux dans la littérature, reposent sur la ressemblance photométrique de deux pixels qui est quantifiée grâce à une mesure de corrélation, comme la seule mesure de corrélation entre deux pixels n'est pas assez discriminante, elle est généralement calculée sur une fenêtre support de pixels, entourant les dits points (figure 4.2).

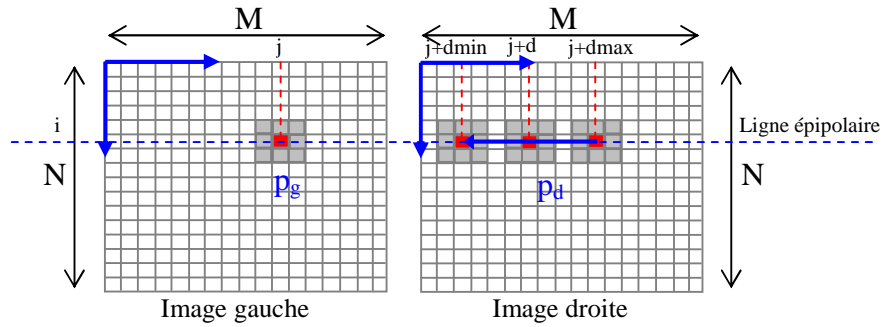


Figure 4.2 : Principe de mise en correspondance par corrélation.

Cette hypothèse est à la base des méthodes de mise en correspondance par corrélation qui sont au centre de nos travaux, bien qu'elle soit simple, leur efficacité a été plusieurs fois démontrée dans de nombreuses applications.

Beaucoup de mesures de corrélation ont été proposées dans la littérature comme présenté dans le chapitre précédent, pour prendre en compte les différentes difficultés de la mise en correspondance, comme les zones peu texturées ou les occultations, il s'avère donc très important de bien choisir le critère de mesure de corrélation adéquat.

Vu que les approches traditionnelles évaluent un à un tous les candidats p_d qui pourraient correspondre à un point p_g d'une façon séquentielle et retiennent le meilleur appariement ce qui constitue un vrai défi pour les applications temps réel, notre approche multiagents exploite l'exécution parallèle des traitements offerte par ce genre d'environnement.

On utilise plusieurs contraintes pour réduire la combinatoire du problème, les contraintes locales comme la contrainte géométrique, la contrainte épipolaire et la contrainte de disparité $[d_{min}, d_{max}]$, à la fin, on applique des contraintes globales (unicité, ordre, continuité de la disparité) pour lever certaines ambiguïtés.

On note que nous nous limitons dans la suite de ce chapitre au modèle de stéréoscope idéal dont les lignes épipolaires sont alignées et parallèles avec les lignes de balayage image, les distorsions peuvent être négligées, cela nous affranchi de l'étape de rectification (contrainte épipolaire respectée) (figure 4.3).

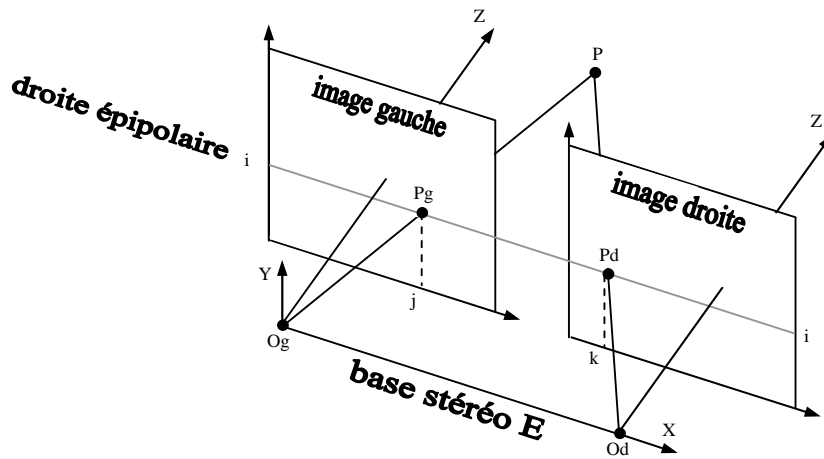


Figure 4.3 : Configuration idéale du système stéréoscopique (lignes épipolaires parallèles).

4.3. Une implantation optimisée :

Notre algorithme présente une manière de calculer la disparité pour tous les points (pixels) de l'image gauche $p_g(i,j)$ par rapport à leurs correspondants $p_d(i,k)$ de l'image droite, en résulte une carte de disparité, qui associe une disparité $d(i,j)$ à chaque pixel de l'image gauche.

Par convention, la carte de disparité est représentée par une image d'intensité de gris (figure 4.4), on associe les disparités faibles (environ -60) relatifs aux pixels proches à des intensités faibles (en claire), inversement, les disparités hautes ou éloignées (proche de 0) ont de fortes intensités (en sombre).



Figure 4.4 : Exemple d'une représentation graphique de la carte de disparité «vérité de terrain».

Donc, dans cette partie, nous détaillerons l'approche reprise par notre algorithme de mise en correspondance de pixels par corrélation qui est l'approche la plus simple présentée dans [Szeliski 02], avec quelques améliorations, dans cette implantation, l'algorithme est divisé en trois phases distinctes (figure 4.5) :

- 1) Appliquer les prétraitements : C'est une phase préparatoire qui effectuera les calculs qui seront utilisés pendant la phase multiagents.
- 2) Calculs des disparités par système multiagents : La deuxième étape est l'appariement multiagents proprement dit, il s'agit du cœur de notre méthode, les traitements de cette

phase sont de nature parallèle (multithreads), le résultat obtenu est une carte de disparité dense qui associe un ensemble de pixels de l'image gauche avec leurs correspondants dans l'image droite.

3) Application des traitements d'optimisations.

Dans la suite de cette partie, nous décrivons ces étapes plus en détail.

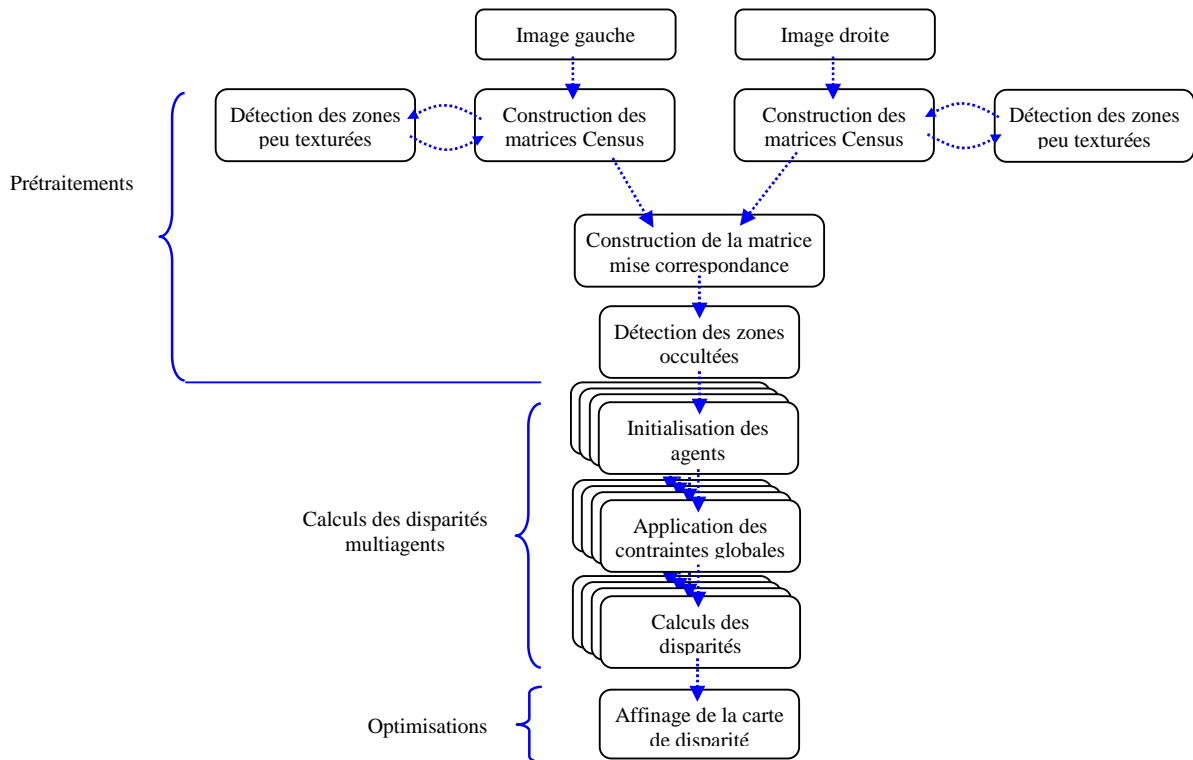


Figure 4.5 : Architecture globale de l'algorithme de mise en correspondance.

4.4. Application des prétraitements :

Certains calculs préalables sont réalisés séparément sur les deux images avant de procéder à la mise en correspondance (notamment l'application d'un filtre de lissage pour éliminer un éventuel bruit), le but de ces traitements est d'éviter les calculs redondants, avant d'entamer cette phase, on doit d'abord choisir la taille de fenêtre de corrélation et le critère de corrélation à adapter.

4.4.1. Choix de la taille de la fenêtre de corrélation :

Les méthodes utilisant des fenêtres de corrélation agrègent le coût de corrélation en le moyennant sur une fenêtre, la taille de cette fenêtre est un paramètre important de l'algorithme, il faut essayer de trouver le meilleur compromis entre les contraintes de temps de calcul et la qualité des résultats, puisqu'une fenêtre trop grande va augmenter la combinatoire, au contraire, une fenêtre trop petite ne sera pas assez discriminante, et augmentera le ratio de faux appariements, dans notre algorithme la taille de cette fenêtre carrée F de dimension W qui varie de 5 à 13.

4.4.2. Choix du critère de corrélation :

Le critère de corrélation est aussi un élément important de la méthode, les critères numériques présentés précédemment dans l'état de l'art, en particulier le critère ZNCC, qui malgré sa robustesse aux variations photométriques, est long à calculer, les critères SAD ou ZSAD sont par contre plus simples, mais généralement, ils s'avèrent peu robustes.

Pour des applications liées à la navigation temps réel, il est plus approprié d'utiliser un critère qui joint simplicité des calculs et robustesse, on a donc opté pour la transformation de rang proposé par Woodfill et Zabih [Zabih 94], un critère non numérique, plus qualitatif, appelé Census plus robuste au bruit et plus efficace sur des images peu texturées vu qu'il utilise la relation entre les données plutôt que les données elles mêmes.

4.4.3. Agrégation de la corrélation :

Au préalable, sur les images gauche et droite de dimension $N \times M$ à comparer, une transformation est appliquée afin de remplacer la luminance en chaque pixel, par une chaîne de bits, les bits de cette chaîne donnent les résultats des comparaisons de cette valeur (45) sur la figure avec les valeurs de luminances des pixels de la fenêtre, parcourue ligne à ligne, de gauche à droite, et de haut en bas (figure 4.6) : le bit de comparaison vaut 1 si 45 est plus grand ou égal à la valeur du pixel traité, 0 sinon.

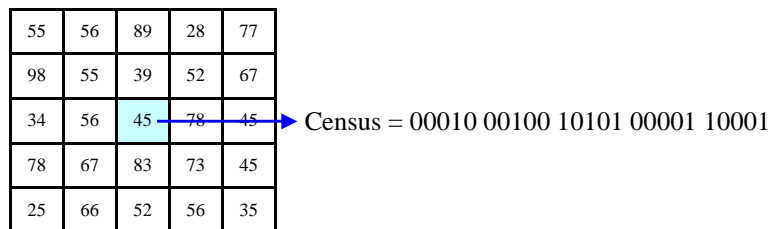


Figure 4.6 : Transformation Census sur un pixel avec une fenêtre 5x5.

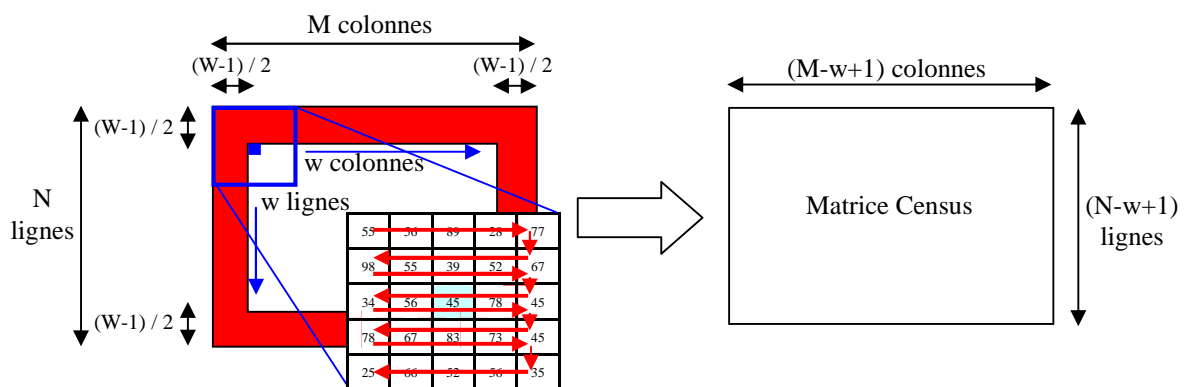


Figure 4.7 : Principe du moyennage $W \times W$ (les zones en rouge sont ignorées).

Après l'application de l'algorithme **Construction_Census_c** sur les deux images, on obtient deux matrices Census notées C_g et C_d de taille identique de $(M-w+1)$ colonnes et $(N-w+1)$ lignes où w est la taille de la fenêtre de corrélation (figure 4.7).

Construction_Census_c

Début

```

Pour c = {g,d} //le même traitement s'effectue sur l'image gauche et droite
  Lire Imgc[M,N] //chargement de l'image de taille MxN
  Pour i := (W-1) / 2 to M-(W-1) / 2 faire //parcours des lignes de l'image
    Pour j := (W-1) / 2 to N-(W-1) / 2 faire //parcours des colonnes de l'image
      Si (I(i,j) - I(i,j-1) ≥ Stexture) or (j = (W-1) / 2) alors //comparaison d'intensités de deux pixels voisins,
                                                                    On commence par la deuxième colonne
        Pour k := i - (W+1) / 2 to i + (W+1) / 2 faire //parcours des lignes de la fenêtre de corrélation W
          Pour f := j - (W+1) / 2 to j + (W+1) / 2 faire //parcours des lignes de la fenêtre de corrélation W
            Si I(i,j) ≥ I(k,f) alors Cc(i,j) := Cc(i,j) ⊕ 1;
            Sinon Cc(i,j) := Cc(i,j) ⊕ 0;
          Fin pour;
        Fin pour;
      Sinon Cc(i,j) := 0; //détection de zone peu texturée//
    Fin pour;
  Fin pour;
Fin pour;
Fin;

```

4.4.4. La détection des zones peu texturées ou de textures uniformes :

Ce sont des zones difficiles à apparier, c'est pourquoi de nombreux auteurs ont pris en compte le traitement de ce type de zones, le plus souvent, elles sont détectées en utilisant la variation d'intensité du niveau de gris [Szeliski 02].

Dans notre algorithme, au cours du processus de construction de la matrice Census, en traitant le pixel $p(i,j)$ on compare son niveau de gris avec celui de son prédécesseur direct sur la même ligne, le pixel $p(i,j-1)$ si la différence est inférieure à un seuil $S_{texture}$ choisi, on déduit alors qu'on est dans une zone de texture uniforme, on attribue une note de 0 à la transformée Census du pixel $p(i,j)$ (figure 4.8).

Cette optimisation permet ultérieurement de faire propager la même valeur de disparité aux pixels voisins appartenant à la même zone uniforme sans avoir à les recalculer (en application à la contrainte de continuité de disparité), l'avantage de cette technique est double, premièrement on réduit les faux appariements en évitant la recherche d'appariements dans les zones peu texturées (score Census notés par un 0) deuxièmement on réduit le temps d'exécution en évitant les calculs inutiles.

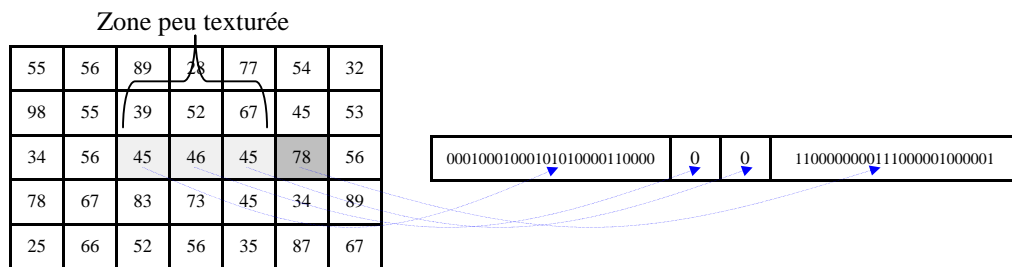


Figure 4.8 : Détection et transformation Census des zones peu texturées.

4.5. Calculs des disparités par système multiagents :

L'idée principale de l'approche multiagents est de distribuer la résolution d'un problème à un ensemble d'entités autonomes possédant une certaine "habilité sociale" par la mise en œuvre d'algorithmes distribués.

Dans notre projet d'appariement on développe un modèle basé sur un ensemble d'agents légers (light threads) collaborant afin de faire émerger de leur comportement individuel simple, la meilleure configuration d'appariement d'un couple d'images stéréoscopiques [Gechter 05].

Le choix d'implémenter des threads légers est motivé par la nature simple des décisions à prendre, ce genre d'entité possède un mécanisme de réaction aux évènements et une capacité d'interaction avec son environnement, ce qui permet d'assurer un coût d'exécution individuel assez faibles ce qui est intéressant pour les traitements temps réel.

La conception de cette méthode a donc été effectuée via une approche orientée agent, où le problème est abordé sur deux plans : le comportement individuel des agents et leur comportement collectif qui doit permettre de faire émerger la solution.

4.5.1. Construction de la matrice de mise en correspondance :

La mise en correspondance entre l'image de gauche et celle de droite revient alors à la mise en correspondance des deux matrices Census C_g et C_d pour se faire on construit d'abord une nouvelle matrice tridimensionnelle appelée la matrice de mise en correspondance S comme proposé dans [Ruichek 05], cette matrice contient tous les scores de corrélation résultat de la comparaison

par distance de Hamming des couples candidats potentiels à l'appariement appartenant aux deux matrices Censur gauche et droite.

L'utilisation de cette matrice est très importante afin d'optimiser certains calculs, cette matrice représente également un environnement pour l'évolution de nos agents, mais son apport le plus intéressant est la possibilité d'exclure les couples qui ne respectent pas les contraintes locales en amont du processus de mise en correspondance (les zones en gris sur la [figure 9](#)) ce qui réduira considérablement les calculs, le problème se ramène ensuite à la détermination des bons appariements qui représentent les meilleurs scores de corrélation et qui satisfassent au mieux les contraintes globales.

Cette matrice 3D de $(M-w-1)$ lignes qui correspond au nombre de colonnes que contiennent les deux matrices C_g ou C_d , et de $[dmin, dmax]$ colonnes vu qu'une cellule de la matrice C_g est comparée avec une partie de la ligne correspondante de la matrice C_d incluse dans l'intervalle $[dmin, dmax]$, cette optimisation est effectuée par un décalage vertical des lignes de la matrice S permet de réduire la taille de cette matrice et du même coup d'exclure les couples ne respectant pas les deux contraintes locales, la contrainte d'intervalle de disparité $[dmin, dmax]$ et la contrainte géométrique (pour réaliser ce décalage on introduit, alors la cellule $S(i,j,k)$ devient $S(i,j,k + (|dmax - dmin| - j))$).

Une attention particulière est accordée au traitement des cellules appartenant à C_g ou à C_d qui ont un score Censur de 0 et qui correspond à des zones peu texturées détectées lors de la phase précédente, les cellules S correspondantes sont notées par un score (-1) pour être ignorées pendant la phase prochaine.

La profondeur (le nombre de plans) de cette matrice représente le nombre de lignes que contiennent les deux matrices C_g ou C_d qui est de $(N-w-1)$ comme illustre la [figure 4.9](#).

La matrice S est calculée suivant la formule suivante :

$$S(i,j,k + (|dmax - dmin| - j)) = D_{ham}(C_g(i,j), C_d(i,k)), k \in \{j+dmax, j+dmin\}.$$

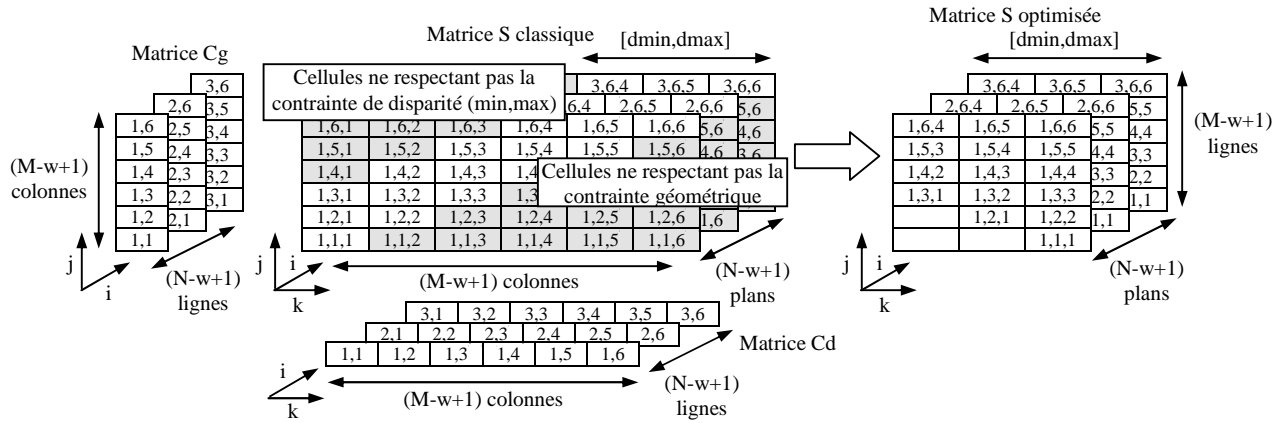


Figure 4.9 : La construction de la matrice de scores optimisée S.

Ici on présente une manière pour la construction du plan i de la matrice S optimisée (plan correspondant aux possibilités d'appariements entre la ligne i de la matrice C_g et la même ligne de la matrice C_d) :

Construction_ S_i

Début

```

Pour j de 1 a (N-w+1) faire //on parcourt toute la ligne i de la matrice Cg//
    Si Cg(i,j) <> 0 alors //si la cellule ne représente pas un pixel appartenant a une zone peu texturée//
        Pour k de j - | dmax-dmin | a j faire //on parcourt la ligne j de la matrice Cd dans l'intervalle de disparité//
            Si j - | dmax-dmin | >= 0 alors
                Si Cd(i,k) <> 0 alors //si la cellule ne représente pas un pixel appartenant a une zone peu texturée//
                    S(i,j,k + ( | dmax-dmin | -j)) = Dham(Cg(i,j),Cd(i,k)); //décalage induit pour optimisée la matrice S//
                Sinon
                    S(i,j,k + ( | dmax-dmin | -j)) = -1; //on note les zones peu texturée par -1//
            Fin si
        Fin si
    Fin pour
    Sinon
        S(i,j,k + ( | dmax-dmin | -j)) = -1; //on note les zones peu texturée par -1//
    Fin si
Fin pour

```

Fin

4.5.2. Seuillage des scores de corrélation :

Un seuillage sur les valeurs de corrélation permet de dégager trois seuils en divisant l'intervalle du score maximum de corrélation ($S_{\max} = W \times W$) en trois paliers égaux :

- 1) Le premier $S_{\text{occultation}}$ le plus bas de $[0, 1/3] \times S_{\max}$ détermine le score des pixels qui n'ont pas de correspondants, c'est le cas des pixels appartenant aux zones occultées.
- 2) Le deuxième de $[1/3, 2/3] \times S_{\max}$ représente les échecs de corrélation pour diverses raisons qu'il convient d'éliminer.
- 3) Le troisième $S_{\text{appariement}}$ de $[2/3, 1] \times S_{\max}$ représente les scores retenus.

4.5.3. La détection des zones occultées :

Un des problèmes les plus délicats en vision par ordinateur, et notamment dans le cadre de la mise en correspondance stéréoscopique, est le problème des occultations, c'est des zones pour lesquelles les pixels n'ont pas de correspondant, c'est-à-dire qu'il y'a des discontinuités de profondeur avec une partie visible dans une seule image, dans la littérature, de nombreux articles ont été proposés pour prendre en compte ce problème, les différentes techniques proposées reposent sur des méthodes avec seuillage des scores de corrélation comme exposé dans [Kong 03] et [Woetzel 04], les auteurs s'appuient sur l'hypothèse suivante : tous les pixels qui possèdent un score de corrélation en dessous d'un seuil sont considérés comme appartenant à une zone des occultations.

Dans notre approche, ce traitement est effectué simultanément avec la phase d'initialisation des agents comme exposé dans l'algorithme d'initialisation des agents, pour détecter si un pixel $C_g(i,j)$ appartient à une zone d'occultation, il faut que tous les scores de corrélation de ce pixel avec les pixels de $C_d(i,k)$ appartenant à l'intervalle de disparité soient inférieurs au seuil $S_{\text{occultation}}$ ce qui signifie que tous les scores des cellules de la ligne j de la matrice S sont inférieurs à ce seuil, alors toutes les cellules de cette ligne sont notées par (-2), ces zones seront ignorées pendant la phase suivante, ce traitement est illustré dans l'algorithme d'initialisation des agents.

4.5.4. Initialisation des agents :

En parcourant chaque plan S_i de la matrice S ligne par ligne, sur chaque cellule $S(i,j,k)$ qui contient un score supérieur à un seuil $S_{\text{appariement}}$ choisi, ce score représente la possibilité de mise en correspondance du couple de pixels $p_g(i,j)$ et $p_d(i,k)$, un agent $Ag_{i,j,k}$ est instancié.

L'algorithme qui réalisera l'initialisation d'agents sur le plan i de la matrice S , et détecter en même temps les pixels appartenant aux zones occultées, sera comme suite :

Initialisation_agents_plan_i

Début

```

  Occ boolean = true ;
  Pour j de 1 a (N-w+1) faire
  | Pour k de 1 a | dmax-dmin | faire
  | | Si S(i,j,k) ≥ Sappariement alors //détection d'un optimum//
  | | | New Agijk(Score,i,j,k); // Agijk extends thread
  | | | Agijk.Start();
  | | | Occ = false ;
  | | | Si S(i,j,k) > Soccultation alors //détection d'un pixel non occulté//
  | | | | Occ = false ;
  | | | Fin si ;
  | | Fin pour
  | Si Occ alors //détection d'un pixel occulté//
  | | Pour k de 1 a | dmax-dmin | faire
  | | | S(i,j,k) = -2 ;
  | | | Fin pour ;
  | | Fin si ;
  | Fin pour ;
Fin;

```

4.5.5. Application des contraintes globales :

L'application des contraintes globales est la phase la plus complexe de notre algorithme, la mise en œuvre d'un système concurrent permet de simplifier grandement l'algorithme et optimiser les traitements vu que la mise en correspondance s'effectue par émergence du à l'interaction au sein du système.

Après leurs initialisation, chaque agent Ag_{ijk} tend à évaluer le degré de respect de la cellule qu'il est chargé de validé $S(i,j,k)$ des contraintes globales (la contrainte de l'unicité et d'ordre) en interagissant et en défavorisant progressivement les autres scores des cellules qui mettent en échec ces contraintes par rapport à lui, quand le système atteint un état stable, tous les agents restants correspondent aux couples respectant au mieux les contraintes globales.

4.5.5.1. Mise en œuvre :

Le traitement des contraintes globales nécessite le test de tous les couples de points présents sur chaque plan i de la matrice S susceptible de violer les contraintes globales entre eux (chaque plan i représente la mise en correspondance entre une ligne de la matrice C_g et la partie incluse dans l'intervalle de disparité de la ligne correspondante de la matrice C_d).

Tous les agents sont initialisés avec un score de satisfaction (nombre entier), ce score est égal au score de corrélation de la cellule que l'agent est chargé d'évaluer, tous les agents ont leurs scores de satisfaction au maximum et leurs états courants en « ChercheMaximum ».

Sur chaque ligne, au début du traitement, on donne la priorité aux agents possédant le meilleur score de corrélation pour procéder au démarrage du système multiagents, [Ruichek 05] dans son algorithme de mise en correspondance par réseau de neurones de Hopfield ne prend pas en compte la pondération du score de corrélation, il ne traite que la pondération relative aux contraintes globales, alors que l'adaptation de cette priorité permet de favoriser les appariements possédant le meilleur score par rapport aux autres optimums éventuellement présents sur la même ligne ce qui offre d'avantage de robustesse.

4.5.5.2. Recherche de l'optimum :

Pour pouvoir détecter la cellule qui contient le score de corrélation le plus élevé pour chaque ligne parmi tous les optimum retenus, on procède comme suite, tous les agents du système qui sont en « ChercheMaximum » cet état permet à chaque agent Ag_{ijk} de comparer le score de sa cellule avec celui des autres cellules qui peuvent être présents sur la même ligne que lui, chaque agent constitue une liste qui contient toutes les cellules valides présents sur la même ligne, si l'agent arrive à parcourir toute cette liste sans qu'il trouve un score supérieur au sien, il passe alors à l'état « TesteUnicité », dans le cas contraire il demeure en attente actif en état de « ChercheMaximum ».

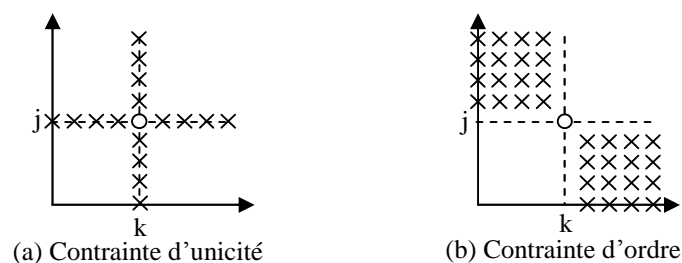


Figure 4.10 : L'impact des contraintes sur l'ensemble des appariements dans la matrice de scores.

4.5.5.3. Application de la contrainte d'unicité :

Ensuite chaque agent dont l'état actuel est en « TesteUnicité » procède de la même manière que l'étape précédente, il mis à jour une nouvelle liste contenant toutes les cellule de la matrice score susceptibles de ne pas respecter la contrainte d'unicité avec sa propre cellule (figure 4.10 (a)), cette liste comporte tous coordonnées des cellules sur la même ligne ou la même colonne que lui (on n'oublie pas de prendre en compte le décalage induit lors de la construction de la matrice S optimisée en diminuant le facteur $(|d_{max}-d_{min}|-j)$ de la valeur de k).

Dés qu'il détecte une cellule qui met en échec cette contrainte avec lui, il lui inflige une pénalité qui correspond à la pondération de la contrainte d'unicité, on prend (-2) pour la pénalité de la contrainte d'unicité violée afin de diminuer le score de corrélation de cette cellule et défavoriser ainsi le couple que cette cellule représente, se choix découle de la pondération supérieure allouée à cette contrainte parce que c'est la plus stable.

Ce traitement est répété jusqu'à ce que la liste de violation d'unicité soit vide, sauf dans le cas où l'agent détecte la présence dans cette liste d'une cellule dont l'agent responsable est déjà en état « Apparié », dans ce cas l'agent est immédiatement arrêté, cette approche de communication interprocessus par le biais de l'environnement (la matrice S dans notre cas) est adaptée pour éviter une communication par envoi de messages qui sera trop coûteuse en temps.

4.5.5.4. Application de la contrainte d'ordre :

Ensuite de la même manière, on procède à l'application de la contrainte d'ordre, les mêmes agents qui sont déjà en « TesteUnicité » et qui ont épuré leurs listes de violation de la contrainte d'unicité, procèdent à un nouveau changement d'état pour exécuté « TesteOrdre », ou chaque agent commence par constituer une nouvelle liste contenant les cellules présentes sur les deux zones qui violent cette contrainte par rapport à la position de l'agent (figure 4.10 (b)).

La cellule de l'agent Ag_{ifn} viol la contrainte d'ordre par rapport à celle de l'agent Ag_{ijk} si $(j-f)(k-n) < 0$, alors le score de la cellule_{ifn} est pénalisé par déduction du score (on prend -1 comme pondération de la contrainte d'ordre).

Utile de noter que chaque agent peut recevoir une pénalité qui débitera le score de corrélation de sa propre cellule à tout moment, quelque soit son état courant, infligé de la part d'autres agents dont il est considéré comme non respectant des contraintes globales par rapport à eux.

4.5.5.5. La validation d'appariements :

Chaque agent dont l'état est en « TesteOrdre » et qui a effectué toutes les comparaisons, procède à la comparaison du score de corrélation de sa cellule avec son score de satisfaction initial, si la cellule sauvegarde un score de corrélation intégral, l'agent change d'état pour se mettre en « Apparié », sinon il revient à l'état « ChercheMaximum » en mettant à jour son score de satisfaction pour qu'il soit égal au nouveau score de corrélation de la cellule, quand l'état de l'agent est « Apparié » ce qui signifie que la cellule correspond à un couple de pixels correctement estimé (score maximum + respect de la contrainte d'unicité + respect de la contrainte d'ordre), cet agent procède alors à la validation de l'appariement représenté par cette cellule.

4.5.5.6. Etat final du système :

Les agents dont l'état est en « ChercheMaximum » effectue des tests cycliques, dès qu'un agent devient le nouveau détenteur du score de corrélation maximum de sa ligne (le maximum précédent étant défavorisé par la réception de beaucoup de pénalités) il passe à l'état « TesteUnicité », ou l'agent procède à un nouveau cycle de tests comme décrit précédemment, si toute fois l'agent épuise complètement son score de corrélation, il est arrêté.

Ce traitement est répété (figure 4.11) jusqu'à ce que tous les agents restants seront en état « Apparié », cet état final du système représente la meilleure configuration d'appariement de pixels respectant au mieux les contraintes globales, chaque cellule alors représente un couple de pixel correctement apparié.

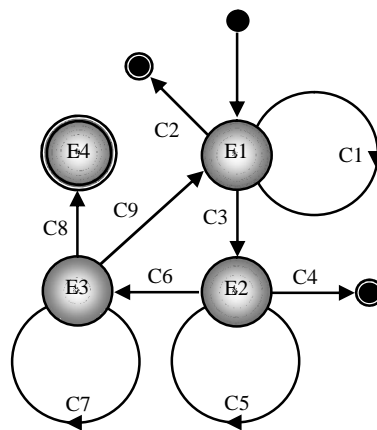


Figure 4.11 : Comportement d'un agent léger (états et conditions).

E1 : ChercheMaximum, E2 : Test Unicité, E3 : Test Ordre, E4 : Apparié.

C1 : Si le score de satisfaction de la cellule n'est pas l'optimum de sa ligne, C2 : Si le score de corrélation est épuisé, C3 : Si le score de corrélation est l'optimum, C4 : Si la contrainte d'unicité est violée par rapport à une cellule dont l'état est « Apparié », C5 : Si la liste de violation de l'unicité n'est pas vide, C6 : Si la liste de violation de l'unicité est vide, C7 : Si la liste de violation de l'ordre n'est pas vide, C8 : Si la liste de violation de l'ordre est vide et le score de corrélation est inchangé, C9 : Si la liste de violation de l'ordre est vide mais le score de corrélation a changé.

4.6. Construction de l'image de disparité :

Chaque agent Ag_{ijk} qui demeure présent sur la matrice S après que le système atteint son état final, représente un couple de pixels correctement apparié, il procède au calcul de la disparité d relative à cette cellule ($d=j-k$) qui sera traduite en image d'intensité de gris sauvegardée sur une nouvelle matrice bidimensionnelle D , cette image sert à visualiser le résultat d'appariement.

Pour construire cette matrice on distribue la variation d'intensité de gris allant de l'intensité la plus faible (blanc) à la plus forte (sombre) sur l'intervalle de disparité $[d_{min}, d_{max}]$ (on prend 256 niveau de gris).

Durant cette phase, un traitement spécial est réservé aux pixels appartenant aux deux zones particulières, les zones peu texturées et les zones occultées, pour ce faire, chaque agent Ag_{ijk} correctement apparié, teste la cellule $S(i, j+1, k)$ appartenant à la ligne voisine, si on détecte une cellule notée par un score égal à (-1) ce qui signifie qu'on a affaire à une cellule appartenant à une zone peu texturées, on procède alors à la propagation de la valeur de disparité à la cellule $(i, j+1)$ de l'image de disparité D , le test se poursuit successivement sur toutes les cellules voisines une par une jusqu'à ce que on trouve une cellule ayant un score différent de (-1) , où on s'arrête.

Si par contre on trouve un score de (-2) qui correspond à une cellule qui appartient à une zone occultée on affecte à la cellule relative sur l'image de disparité une valeur d'intensité de gris particulière pour qu'elle soit identifiable comme étant un pixel appartenant à une zone occultée.

4.7. Affinage des résultats :

Le filtrage de l'image de disparité qui contient pour chaque pixel de l'image gauche, la valeur trouvée pour la disparité (filtre médiane), les pixels non appariés ont une disparité nulle, le filtrage exploite des comparaisons simples : il permet d'invalider la disparité pour des pixels isolés (par exemple un pixel qui a une disparité nettement différente de celle des pixels voisins).

4.8. Tests et résultat :

Les critères d'évaluation les plus répandus dans la littérature sont la qualité de la carte de disparité et le temps d'exécution, pour pouvoir évaluer notre approche on essaye de quantifier ces deux paramètres importants.

On rappelle que les résultats que nous exposons ci-après correspondent à une implantation des deux caméras parfaitement parallèles (ne nécessitant aucune rectification), les tests sont effectués sur plusieurs couples d'images avec vérité de terrain de résolution 450×375 disponible sur le site de référence (Scharstein et Szeliski) :

<http://cat.middlebury.edu/stereo/data.html>

4.8.1 Evaluation de la robustesse de la méthode :

Pour pouvoir évaluer la robustesse de notre algorithme nous utilisons le protocole d'évaluation et de comparaison proposé par Scharstein et Szeliski [Szeliski 02], ce protocole est le plus utilisé, les auteurs évaluent trois catégories de pixels : les pixels dans les zones peu texturées, les pixels occultés et le reste des pixels.

Cette méthode consiste à compter les pixels correctement estimés appartenant à ces trois catégories, en comparant les disparités calculées par notre algorithme avec ceux des images de références dont on dispose de la carte de disparité de référence, la carte des zones occultées et des zones peu texturées (figure 4.12).

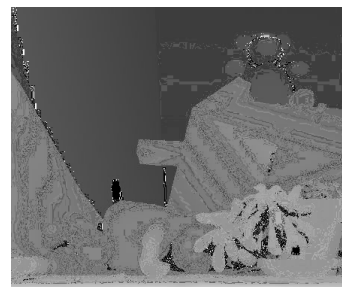


Image gauche de paire d'image de test « teddy »

Carte de disparité théorique

Figure 4.12 : Exemple d'images utilisées dans le test.

Nous notons $d_{\text{ref}}(i,j)$ la disparité de référence, et $d(i,j)$ la disparité calculée par notre algorithme, nous posons : $\text{Err}(i,j) = \|d(i,j) - d_{\text{ref}}(i,j)\|$, ici on s'intéresse au pourcentage d'appariements précis, un appariement est précis si $\text{Err}(i,j) \leq 1$ (un pixel).

Il est utile de préciser qu'on compare les disparités correctement estimées après l'application du seuil et pas le nombre total de pixels de l'image, ce critère est utilisé notamment dans [Faugeras 92] et [Morat 08].

La figure 4.13 correspond à le résultat de la carte de disparité calculée, les ratios de disparités correctes appartenant aux trois catégories sont calculées sur un ensemble paires d'images, les résultats sont repris par le graphique dans la figure 14, sur lequel nous pouvons remarquer que le ratio des estimations correctes évolue entre 68% et 52%, pour une moyenne de 60%, c'est un taux satisfaisant pour notre application vu le nombre d'appariement annulé par seuillage.

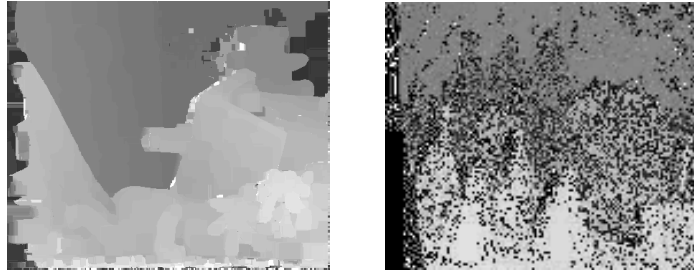


Figure 4.13 : Cartes de disparités calculées (de 0 en noir à -60 en blanc).

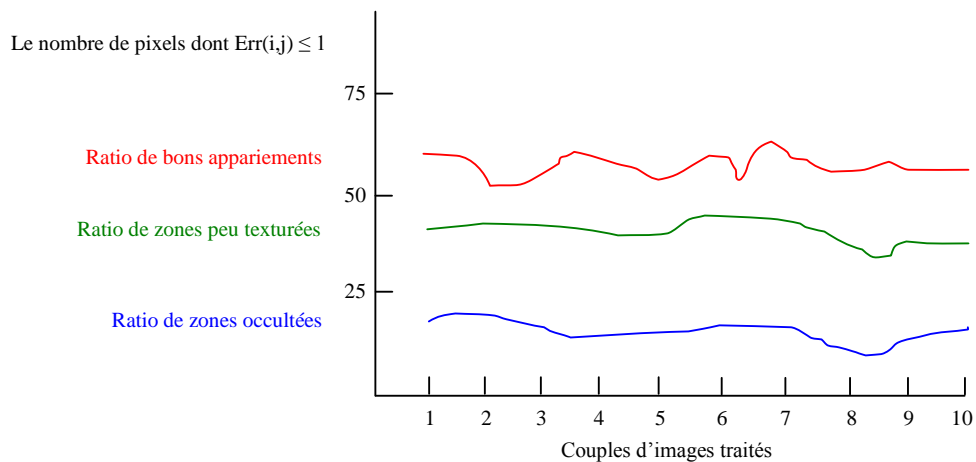


Figure 4.14 : Proportion des disparités correctement estimées des trois catégories de pixels considérées.

4.8.2. Evaluation des performances en temps d'exécution :

L'un des avantages majeurs de l'introduction de système multiagents dans la résolution du problème d'appariement des images stéréoscopique, autre que la notion de robustesse déjà exposée, est certainement l'aspect distribué des traitements offerts par ce genre de système ce qui permet un gain considérable en temps de calculs sur un processeur multicœurs ou sur architecture dédié GPU.

Pour essayer de quantifier les performances temporelles de notre algorithme on procède à évaluer le temps de traitement de l'ensemble des paires d'images par plusieurs tests pour plusieurs valeurs de la taille de la fenêtre de corrélation.

Sur le graphique suivant (figure 4.14), on compare la cadence de traitement avec celui du taux d'appariement correct, on constate que la taille de fenêtre de corrélation qui offre la meilleure performance concorde avec l'intersection des deux graphes.

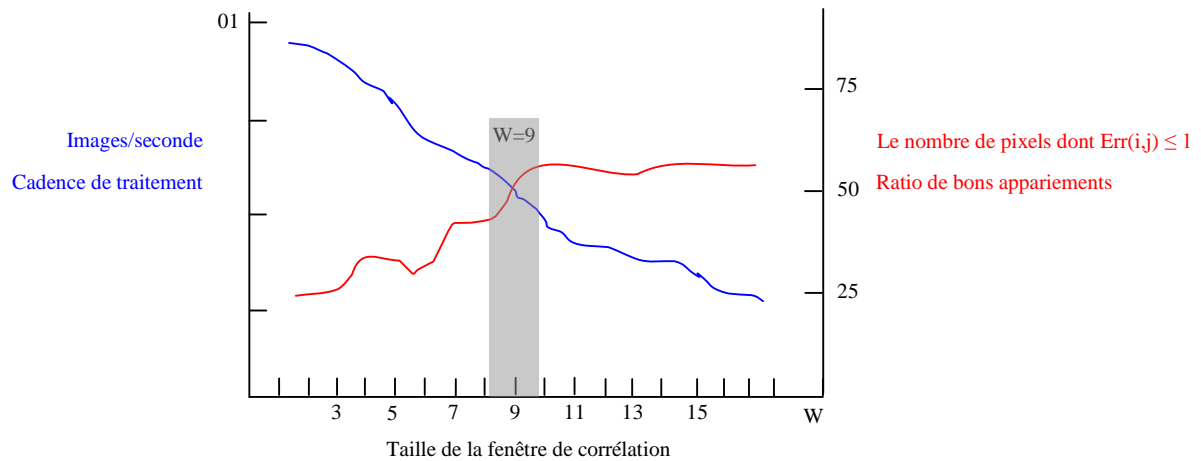


Figure 4.15 : l'influence de la taille de la fenêtre de corrélation sur les performances du système.

4.8.3. Comparaison de notre méthode avec celle de la programmation dynamique :

Comme vu précédemment dans l'état de l'art, la programmation dynamique est l'une des méthodes globales les plus utilisées dans la mise en correspondance stéréoscopique, essentiellement, elle consiste à minimiser une fonction d'énergie sur une ligne entière, sur le site de Scharstein et Szeliski on peut trouver une mise au test détaillée de cette méthode, globalement on peut constater :

- Taux de bons appariements : la méthode de la programmation dynamique donne de bons résultats avec un taux de bons appariements avoisinant les 80%, toutefois les cartes de disparité ainsi calculées ont la fâcheuse tendance de propager les erreurs d'une façon très prononcée (figure 4.16).
- Temps d'exécution : vu que cette méthode n'a jamais été proposée dans le cadre d'une application temps réel, vu qu'elle présente un temps relativement élevé pour ce genre de système (plus de 2 secondes par image).

Pour notre part, même à performance moindre, notre algorithme présente l'avantage crucial procuré par l'aspect parallèle des traitements, on peut donc réduire le temps d'exécution infiniment sur des architectures dédiées hautement multithreads.



Figure 4.16 : Sur cette carte de disparité on peut aisément constater la propagation des erreurs.

4.9. Conclusion du chapitre :

Dans ce chapitre, nous avons présenté un algorithme simple adapté à l'application envisagée à savoir la détection d'obstacle, ainsi que l'implantation optimisée d'un processus de mise en correspondance stéréoscopique, basé sur une approche par corrélations de critères photométriques.

Cette technique couplée à une approche multiagents offre d'énormes perspectives de recherche et de développement, et représente une méthode novatrice en montrant l'utilisabilité d'une telle approche à base d'agents pour aborder les problèmes de traitement d'images.

Les résultats obtenus sur des séquences synthétiques confirment les apports de cette approche, premièrement en ce qui concerne la robustesse de la méthode avec un score de bon appariement satisfaisant même pour les pixels appartenant aux zones peu texturées ou occultées, on constate aussi une cadence d'exécution qui peut atteindre 10 images/seconde en cas d'exécution de notre algorithme sur un matériel dédié GPU, cette exécution sur un matériel dédié ne nécessite qu'un changement minime vu la nature générique et parallèle de notre approche, cette possibilité fera prochainement l'objet de la rédaction d'un papier.

5. Détection et suivi des obstacles pour la navigation

5.1. Introduction du chapitre :

ITS couvre un vaste terrain de recherche sur l'intégration de systèmes, faisant appel à l'informatique et regroupe toutes les techniques mises en œuvres pour offrir une aide aux conducteurs humains afin de prévenir au maximum l'avènement d'accidents graves (se référé au chapitre 2).

La voiture moderne devra aider le conducteur à mieux comprendre son environnement pour l'aider dans ses décisions sans jamais le déresponsabiliser, elle devra donc être capable de percevoir mieux l'environnement que ne peut le faire l'être humain et pallier ses déficiences.

Nous présentons dans ce chapitre une méthode générique de détection et de suivi d'obstacles routiers en temps réel d'une façon robuste par stéréovision embarquée et cela en utilisant une approche multiagents, notre algorithme est capable de détecter tout type d'obstacles présents sur la scène en se basant sur les données offerts par un système de perception passif à base de stéréovision, notre approche s'affranchie de toute supposition sur l'état de la route (plan ou non) et sur la modélisation du mouvement du véhicule (roulis et dévers), elle est basée sur l'exploitation de l'information riche présente dans une carte de disparité dense, un système multiagents exploite cette carte pour effectuer une segmentation des variations du cumul de disparité (histogramme) pour chaque ligne de la carte de disparité pour pouvoir construire une nouvelle image CN (carte de navigation), le traitement s'effectue en comparant le cumul de disparité ligne par ligne du bas vers le haut, en partant de la première ligne visible à l'avant du véhicule, un obstacle est détecté si on constate un changement important dans l'histogramme de disparité d'une ligne par rapport à son prédécesseur, après la caractérisation de chaque obstacle, son évolution est suivi d'une façon autonome au fil de défilement des images à une cadence vidéo.

5.2. Travaux antérieurs :

Dans le domaine de l'aide à la conduite par stéréovision embarquée, plusieurs approches pour la détection d'obstacles dans un environnement routier sont présentées, dans leurs majorités, ces méthodes exploitent généralement l'information obtenue de la carte de disparité issue de la mise en correspondance des images stéréoscopique, les deux approches les plus répandues dans la littérature sont :

5.2.1. Rectification homographique :

Cette méthode exploite la rectification homographique sur une image de manière à ce que le plan de route soit identique dans les deux images, une simple différence entre les images permet alors de discriminer les pixels qui ne correspondent pas à ce plan, l'homographie est appliquée à

priori à l'aide d'une calibration préalable de la route, cette méthode suppose donc que la route est plane dont la position et l'orientation sont connues (par calibrage) et ne varient pas, la segmentation consiste alors à trouver les points qui correspondent au plan de la route.

Maraninchi et al. [Maraninchi 01] effectuent cette fonction directement à partir des images, à l'aide d'une rectification par le plan du sol, après avoir effectué cette opération, la route apparaît l'identique dans les deux images de la paire stéréoscopique, ainsi, les pixels qui représentent des obstacles et qui n'appartiennent pas à ce plan sont identifiés par simple différence des intensités des pixels.

Koller et al. [Koller 94] utilisent les mêmes images rectifiées, mais calculent en plus la disparité en chaque pixel, les pixels de disparité nulle (ou quasi nulle) sont des points du sol, alors que les autres représentent des obstacles, le calcul de la disparité permet également de retrouver leur position dans l'espace 3-D.

L'inconvénient majeur dont souffrent ces approches est qu'elles supposent que le sol est un plan 3-D figé et connu, mais lorsque le véhicule se déplace, les variations importantes de roulis et de tangage qu'il subit empêchent d'utiliser un tel modèle.

5.2.2. Mise en correspondance des images stéréoscopique :

Les méthodes par appariements représentent la majorité des travaux, se basent sur l'exploitation de la carte de disparité issue de l'appariement de pixels entre les deux vues du capteur, c'est à partir de ce nuage de points 3D que, l'extraction des obstacles, du profil de route et du point de fuite de l'image sont directement déduits.

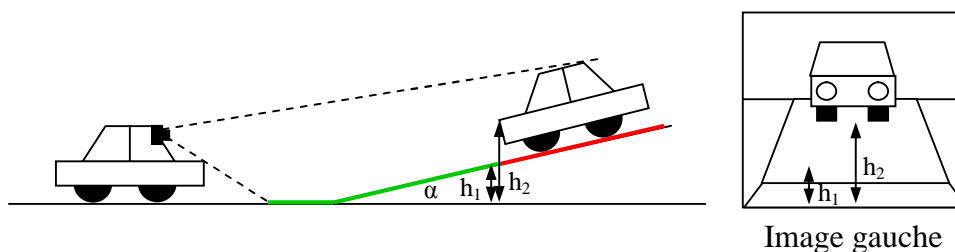


Figure 5.1 : Limitation de l'hypothèse de la route plane.

Dans la littérature, plusieurs hypothèses concernant l'environnement routier sont fréquemment utilisées de façon à faciliter le processus, la majorité des algorithmes existants supposent que la route est plane ([Williamson 98]), certaines techniques font l'hypothèse d'un plan constant ([Zhang 96]), d'autres estiment dynamiquement l'inclinaison du plan de la route et la hauteur relative du capteur stéréoscopique ([Koller 95] et [Hattori 98]).

Sauf que, si l'hypothèse d'une route plane est acceptable pour les premières mètres à l'avant du véhicule (figure 5.1), il ne l'est pas sur l'ensemble de la portion de route visible dans l'image, dans ce cas, la détection d'obstacles n'est fiable que sur la partie de la route localement plane et la distance de détection est limitée.

Alors viennent les techniques de détection et d'estimation des marquages routiers comme dans [Koller 95], où les marquages routiers sont extraits et mis en correspondance entre l'image gauche et l'image droite de la paire stéréo pour obtenir les paramètres du plan de la route, cependant, cette méthode est limitée par plusieurs contraintes liées à la mauvaise qualité du marquage, une autre possibilité consiste à modéliser les suspensions du véhicule par des équation complexes et pas suffisamment robustes.

Pour ces raisons, il est préférable d'estimer la position du plan de la route à la volée, ce processus passe généralement par la reconstruction complète de la scène à partir d'une carte de disparité dense qui peut être obtenue désormais à une cadence vidéo [Yang 04], le problème consiste alors à identifier, dans cette reconstruction, les parties de la scène qui forment le plan de la route, la diversité des scènes routières et les erreurs de reconstruction rendent la tâche complexe.



Figure 5.2 : L'approche de Labayrade [Labayrade 02] qui consiste à créer une v-disparité dans laquelle se dessine le profil de la route et des obstacles.

En réponse à tous ces problèmes, [Labayrade 02] a fourni une méthode robuste et rapide qui permet l'estimation dynamique du profil en long de la route, cette estimation est possible par la construction et l'analyse de l'image v-disparité pour estimer le profil longitudinal de la route ainsi que la distance qui sépare le stéréoscope des différents obstacles présents, la v-disparité (v axe vertical des images) (figure 5.2) permet d'identifier dans les plans images d'une paire stéréo calibrée et rectifiée les points de la scène qui présentent la même disparité (mouvement relatif en abscisse), le calcul de la v-disparité revient à accumuler les pixels d'une même ligne qui présentent une disparité commune, l'image de v-disparité présente pour chaque ligne de l'image la fonction de

répartition en niveaux de gris des disparités mesurées, reportées en abscisse, cette approche ne nécessite aucune extraction de structure cohérente dans la paire d'images stéréoscopiques (comme des marquages routiers).

Avec cette représentation, il est facile d'associer une droite au profil de la route (avec une transformée de Hough), une fois que les points 3-D appartenant à la route sont étiquetés, ceux restant sont étiquetés comme des obstacles, cette méthode fournit une bonne représentation géométrique de la scène routière, ce qui permet de prendre en compte les routes non planes.

La v-disparité permet de segmenter de manière fiable et rapide des surfaces planaires observées dans la scène, c'est de loin la méthode la plus robuste capable de fournir des résultats de segmentation en tenant compte d'un profil de route planaire par morceaux (jusqu'à 15 pour un même profil).

Pour tolérer des variations de la surface du sol, la littérature propose aussi des approches basées sur des modèles de surfaces, la méthode proposée par Lemonde dans sa thèse de doctorat [Lemonde 05] reprend les histogrammes de la v-disparité, mais réalise la segmentation indépendamment sur chaque ligne de la carte de v-disparité dans un espace 3D, le fait d'ignorer le profil longitudinal de la route permet d'étendre la méthode au cas des routes non-planes.

5.3. Méthode proposée :

Notre travail traite la détection d'obstacles dans un environnement routier d'une façon complètement autonome, ainsi que la possibilité d'évaluer leurs distances par rapport au dispositif de prise de vue stéréoscopique, ce traitement est le résultat d'une collaboration au sein d'un système multiagents collaborant pour l'accomplissement d'un but commun, ce système est chargé d'analyser et de segmenter la carte de disparité dense qui associe à un maximum de pixels leurs coordonnées dans l'espace 3D pour en extraire les informations nécessaires à la tâche de détection d'obstacles.

La technique que nous allons présenter dans ce chapitre adapte les mêmes bases que la v-disparité [Labayrade 02] qui a déjà montré sa robustesse grâce aux cumuls de disparité des lignes de la carte de disparité, avec quelques améliorations, notre approche mélange une approche purement v-disparité pour évaluer les lignes de la carte de disparité appartenant à la route, qui correspond aux lignes présentant une variation monotone dans l'histogramme de cumul des disparités, en cas où cet histogramme présente un pic important qui révèle la présence d'un obstacle, dès lors, on bascule vers une estimation 3D de ces lignes selon l'approche proposée par Lemonde [Lemonde 05] en exploitant la richesse de la carte de disparité dense.

Notre système multiagents comporte deux agents permanents, un agent de segmentation AS et un agent de détection AD, en plus d'un vecteur dynamique d'agents obstacle AO initialement

vide, alors, le premier agent permanent présent dans le système dès son initialisation, l'agent AS procède au traitement de la carte de disparité pour en déduire les informations nécessaires pour la détection des attribues de l'environnement routier, l'agent AS reporte ces informations sur une nouvelle image qu'on appelle carte de navigation CN.

Cette carte de navigation CN est spécialement introduite pour faciliter l'interprétation de la carte de disparité vu que cette nouvelle carte ne contiens que deux informations simple et pertinent pour notre objectif a savoir la détection et le suivi des obstacles routier, cette carte est réduite a indiquée si une zone de la carte de disparité est praticable ou non, cette carte nous affranchi aussi de toute estimation d'un modèle géométrique de la route.

Pour ce faire l'agent AS analyse le cumul de disparité des lignes de la partie tronquée de la carte de disparité (figure 5.4), successivement, ligne par ligne en partant du bas vers le haut, en supposant que les quelques premières lignes en partant du bas appartiennent à la route, alors chaque ligne qui ne présente pas de changement notable dans son cumul de disparité avec la dernière ligne qui appartient à la route est considéré comme appartenant à la route à son tour, en cas ou le cumul de disparité présente un changement brusque alors l'agent AS procède à une analyse plus affinée en basculant vers une estimation 3D qui nous permettra une caractérisation plus précise (dimensions, positions) des obstacles, ensuite, l'agent AD segmente cette carte de navigation CN et englobe les obstacles détectés par des cadres englobant, une fois les obstacles détectés, le suivi et l'évaluation de chaque obstacle sont assignés au agents obstacle AO qui le fera d'une façon autonome.

Dans cette approche, on procure à notre méthode la propriété de robustesse en effectuant une accumulation des estimations au fil du défilement des images à cadence vidéo des différents paramètres de l'environnement routier, ce qui nous affranchira notamment de la modélisation de l'effet du dévers et du roulis, ces paramètres sont calculés dynamiquement.

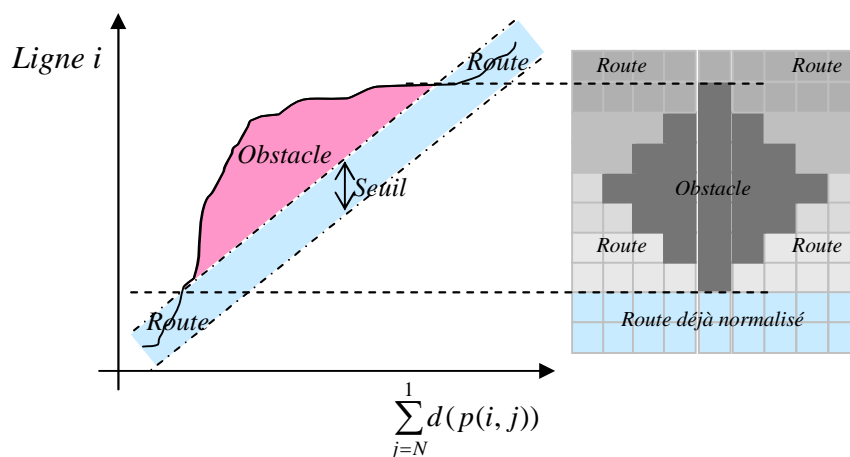


Figure 5.3 : Analyse de la fonction de cumul de disparité par ligne de la carte de disparité.

5.4. Volume tronqué de la route :

Pour les applications d'aide à la conduite, il est fondamental d'éviter les fausses détections, en particulier dues aux obstacles détectés en bord de route comme les trottoirs ou les arbres, pour ce faire, il faut limiter la détection des obstacles à la surface directe à l'avant du véhicule, ou encore, à la voie de circulation, et donc les points très en dessous ou au dessus de cette surface ne présentent pas d'intérêt pour les applications de détection d'obstacles

Lors du calcul de la carte de disparité, tous les pixels d'une image sont mis en correspondance pour déterminer leurs disparités, de cette manière, tout point 3-D couvert par le système peut être reconstruit, comme l'illustre la [figure 5.4\(a\)](#), la zone de couverture du stéréoscope est délimitée par l'intersection des deux champs de vision des deux caméras, dans notre application de détection d'obstacle pour la navigation ce volume s'avère plus grand que nécessaire, et donc engendre des calculs inutiles ([\[Nedevschi 05\]](#)).

Pour économiser du temps de calcul, on tronque ce volume (voir la [figure 5.4\(b\)](#)), le principe consiste à rogner l'image de disparité en fonction de la disparité pour obtenir une carte de disparité qui correspond juste au volume utile à la détection.

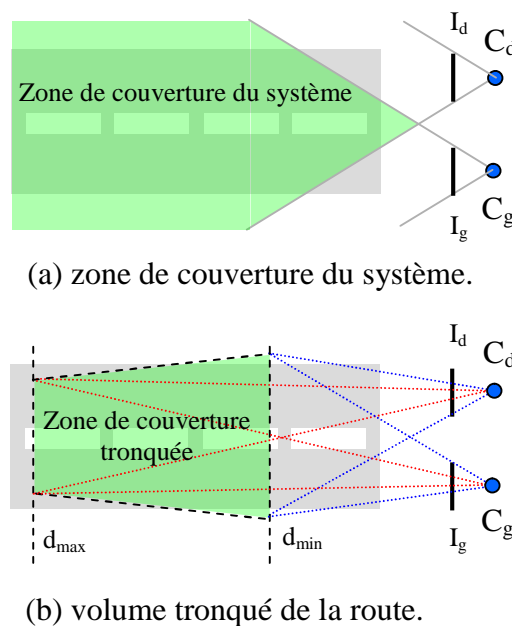


Figure 5.4 : Adaptation la taille de la zone de détection en fonction de la disparité.

5.5. Description détaillée du fonctionnement :

Au démarrage du système, les deux agents statiques sont instanciés : l'AS responsable de la création de l'image CN comme décrite précédemment, ensuite cette image est transmise à l'agent détecteur AD qui prend le relais (figure 5.5), il procède à une segmentation de celle-ci, vu que cette carte contient une représentation simple de l'environnement, sa segmentation revient plus facile que la segmentation de la carte de disparité d'origine.

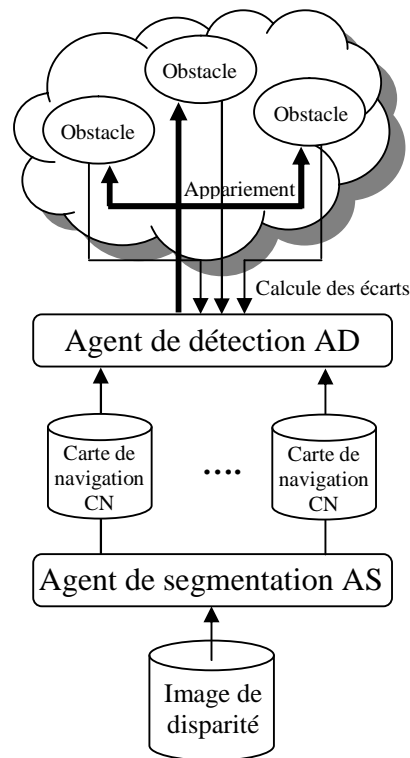


Figure 5.5 : Architecture globale du système.

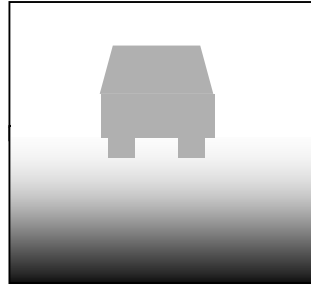
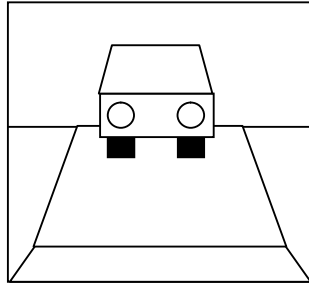
5.5.1. Construction de la carte de navigation :

Cette image est introduite spécialement à la place de l'image de v-disparité, pour faciliter le processus de segmentation de la carte de disparité dans un espace 3D (figure 5.6 (a) et (b)) , cette image est réduite à comporter que deux informations simples et déterminantes pour notre système, l'information relative à la chaussée praticable, et l'information relative aux obstacles présents (praticable ou non praticable), donc cette image sera dérivée directement de la carte de disparité dense, cette image qu'on appelle carte de navigation CN calculée par le premier agent permanent de notre système, l'agent de segmentation AS qui procède comme suite :

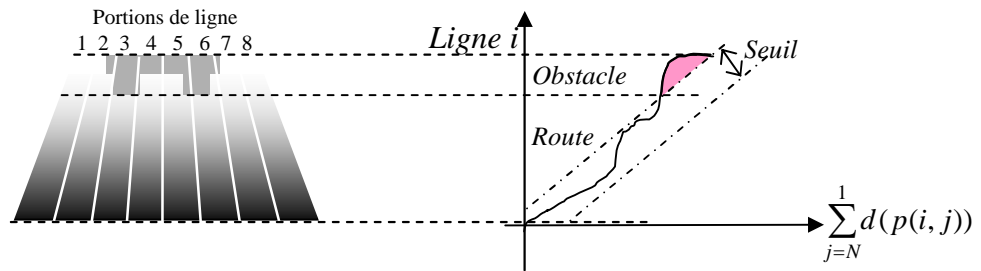
- 1) Calcul du cumul des disparités connus sur chaque ligne de la carte de disparité (v-disparité).
- 2) Création d'une nouvelle image CN (dimension équivalente au volume tronqué de la carte de disparité).

- 3) Normalisation comme praticable des premières lignes en partant du bas de la carte de disparité vu que c'est la partie la plus susceptible d'être en contact avec l'avant du véhicule, cette hypothèse simple, nous affranchi de la détection de la chaussée par la construction d'un modèle géométrique par la transformé de Haug.
- 4) Ensuite, l'agent AS procède à la comparaison de la dernière ligne déjà normalisée i avec la ligne suivante $i+1$, la ligne qui présente une variation continue de l'histogramme de cumul de disparité est normalisé à son tour comme appartenant à la chaussée praticable, donc on a :
si $\sum_{j=N}^1 d(p(i, j)) - \sum_{j=N}^1 d(p(i+1, j)) \leq \text{seuil}$, dans ce cas la ligne $i+1$ est normalisée à son tour comme appartenant à la chaussée praticable et sera utilisée comme référence pour le test de la ligne suivante.
- 5) Dans le cas où l'agent AS détecte un changement brusque ou une « cassure » dans le cumul de disparité d'une ligne par rapport à la ligne prédécesseur, l'agent procède alors à une segmentation plus affinée de cette ligne par rapport à la ligne précédente, en divisant la ligne en portions égales (figure 5.7), alors si la comparaison d'une portion de cette ligne avec la même portion de la ligne précédente présente une variation de disparité marquée alors cette portion est normalisée comme appartenant à un éventuel obstacle, tandis que reste des portions de cette ligne qui présentent une continuité de la fonction de cumul de disparité sont normalisés comme appartenant à la chaussée praticable (on effectue le même test précédent sur des portions égaux des lignes successives) (figure 5.6 (c), (d) et (e)).
- 6) Répéter ce traitement jusqu'à la dernière ligne du volume tronqué de la carte de disparité.

Le seuil de détection est choisi l'angle de la pente de la route utile a notre système, généralement on prend une pente de 10% (certains auteurs propose 15% comme valeur maximale), tandis que la longueur des portions de la ligne est relative à la largeur minimale des obstacles à détecter (ces dimensions sont relatives à la distance).

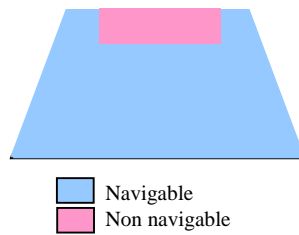


(b) Carte de disparité.



(c) Carte de disparité tronquée.

(d) cumul de disparité v-disparité.



(e) résultat final de la construction de la carte de navigation CN

Figure 5.6 : étapes de construction de la carte de navigation CN.

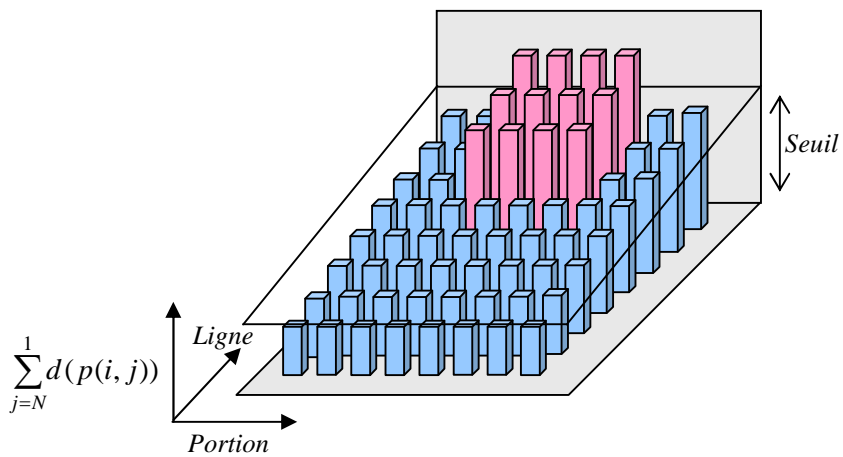


Figure 5.7 : Résultat du cumul 3D de disparité par portion de ligne.

Cette méthode nous affranchi complètement de reconstruire le profil longitudinal de la voie de circulation, vu que la navigation ne nécessite qu'une seule information pertinente sur la praticabilité de la chaussée ou non, la base de l'algorithme réside dans l'étude des lignes de l'image de disparité, une à une limitée à la voie de circulation.

5.5.2. Segmentation et détection d'obstacles :

Cette tâche est effectuée par l'agent AD, qui reçoit la carte de navigation CN de l'agent AS, l'agent AD effectue la segmentation de celle-ci ce qui revient beaucoup moins difficile que la segmentation directe de la carte de disparité dense, la carte CN ne contenant que deux types d'information relative à la praticabilité ou non de la chaussée, pour se faire l'AD procède comme suite :

- 1) On teste la hauteur de chaque zone marquée comme non praticable, si la dimension est supérieure aux limites minimums de dimensions d'obstacles détectables par notre système (ces dimensions sont relatives à la disparité allant du plus proche au plus loin), alors cette zone est reconnue comme étant un obstacle pertinent.
- 2) Les zones de petites tailles (changement de la fonction de cumul de disparité présent sur un petit nombre de lignes) sont ignorées et elles seront considérées comme résultantes d'erreurs d'appariement (même si ses zones représentent vraiment un obstacle, il ne sera pas assez volumineux pour empêcher le passage du véhicule).
- 3) Les zones ainsi identifiées sont caractérisées par les coordonnées d'un cadrant englobant (figure 5.8).

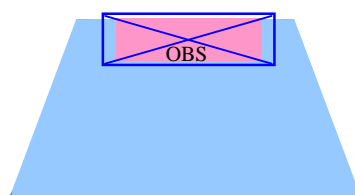


Figure 5.8 : Segmentation des obstacles par des cadres englobant.

Lors du traitement de la première image, l'agent AD instancie des agents obstacles AO au nombre des zones ainsi détectées et avec les coordonnées du cadrant correspondant, cette transmission de coordonnées est très importante, elle va servir pour le suivi de l'évolution de l'obstacle et sert aussi à la caractérisation des obstacles éventuels par leurs dimensions et faciliter ainsi leurs appariement au fil du défilement des images, à la fin l'AD procède au traitement de la prochaine CN calculée par l'agent AS.

5.5.3. Appariement des obstacles :

Les Agents Obstacle AO, sont des threads créés par l'AD, ils maintient un structure de données qui mémorisent les coordonnées des obstacle détecté dans l'image CN, et sert aussi à l'appariement des zones détectées par celui-ci avec les objets issus de l'itération précédente, dans le cas échéant, l'agent AD instancie à partir de la troisième classe d'agent de nouveaux agents AO correspondants aux zones non encore appariées (obstacles nouvellement entrés en scène).

Après avoir traité la première image CN et l'instanciation des agents AO au nombre des obstacles détectés, au cours des itérations suivantes, l'agent AS procède au calcul de l'image CN suivante et l'envoi à l'AD, l'agent AD effectue une segmentation en régions homogènes par l'algorithme décrite précédemment de cette image, pour chaque zone d'obstacles actuellement détectée, l'AD procède de la manière suivante :

- 1) Il met à jour une nouvelle liste d'obstacles.
- 2) Les agents objets AO existant testent cette liste en calculant la différence entre le centre de gravité du cadran ultérieur issu de l'image précédente avec le centre de gravité des obstacles de la nouvelle liste, et doivent identifier chaque obstacle qui offre un écart minimal, vu qu'on est dans un cadre de navigation en temps réel, on suppose que les écarts de position des obstacles entre deux images prises successivement sont faibles, donc, inférieur à un seuil de déplacement donné (relatif à la vitesse maximale du véhicule).
- 3) Les AO qui on pu identifier un obstacle procède a la mise a jour de leurs paramètre de position.

Alors l'agent AD, une fois toute la liste traitée, test les éléments non appariés, dans ce cas l'agent AD constate qu'il s'agit d'un nouveau obstacle qui entre dans la scène et instancie par conséquence un nouveau agent AO avec comme paramètres les coordonnées de la nouvelle zone.

Les agents AO qui n'arrivent pas a s'appariés sont considérés comme appartenant a des obstacles perdu (obstacles disparus de la scène), ils sont tout simplement arrêtés.

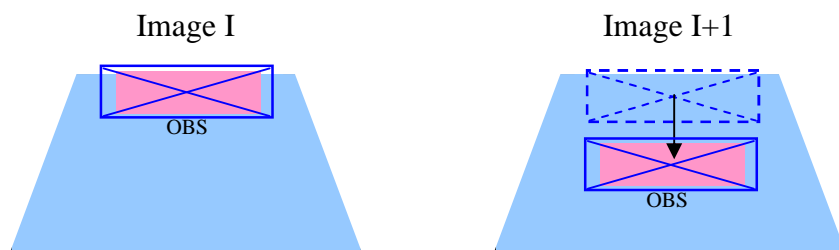


Figure 5.9 : Exemple d'appariement d'obstacles.

Ce comportement s'exécute parallèlement pour toutes les zones, après que toutes celles-ci soient traitées l'AD procède au traitement d'une nouvelle CN, ce traitement est répété jusqu'à ce que l'AS arrête le calcul de nouvelle CN.

Afin d'augmenter la robustesse de la méthode et diminuer la possibilité de toute fausse alerte, l'agent AO effectue le suivi de l'évolution de l'obstacle qui lui a été assigné pendant plusieurs images de la séquence (un obstacle pertinent pendant 5 images successives est considéré comme un obstacle réel) avant que l'obstacle soit encadré par un rectangle.

Le comportement de notre système est conçu comme un ensemble de trois machines à état fini (figure 5.10, figure 5.11 et figure 5.12), la synchronisation entre les trois agents s'effectue sur des données partagées.

Alors l'agent AS et l'agent AD sont synchronisés sur l'image CN comme ressource critique, selon un simple modèle de lecteur/rédacteur, un agent AD ne peut pas traiter une image CN si celle-ci n'a pas été créée, de même l'agent AS ne peut pas créer une nouvelle image CN si l'agent AD n'a pas encore libéré cette image.

Tandis que la synchronisation entre l'agent AD et les agents AO se fait sur la liste des obstacles, une exclusion mutuelle permet l'ordonnement de nos threads.

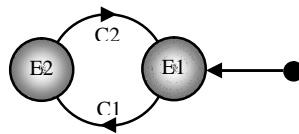


Figure 5.10 : Comportement de l'agent AS (états et conditions).

E1 : calcul de l'image CN entièrement, E2 : transmission de la carte CN.

C1 : si l'agent termine la construction de la CN, C2 : si l'agent transmet la CN entièrement vers l'agent AD.

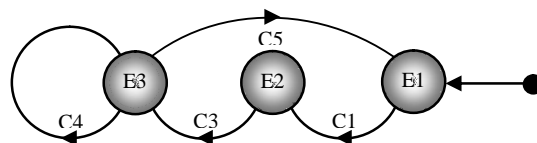


Figure 5.11 : Comportement de l'agent AD (états et conditions).

E1 : segmentation de l'image CN, E2 : mise à jour de la liste des obstacles, E3 : appariement des obstacles.

C1 : si l'agent termine la segmentation de la carte CN, C2 : si l'agent termine la mise à jour de la liste des obstacles, C3 : si l'agent reçoit la réponse des AO, C4 : si l'agent trouve des obstacles non identifiés sur la liste des obstacles, C5 : si toute la liste des obstacles est traitée.

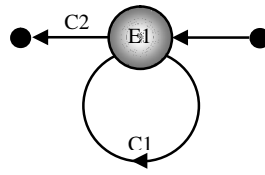


Figure 5.12 : Comportement de l'agent AO (états et conditions).

E1 : Apparié.

C1 : si l'agent réussit l'appariement d'obstacle, C2 : si l'agent ne réussit pas l'appariement d'obstacle.

5.5.4. Prise en compte du roulis et du dévers :

Chaque agent AO garde une structure de données qui traquent les coordonnées successives des emplacements occupés par chaque obstacle détecté, avec les estimations successives de sa position dans l'espace 3D, et vu que les phénomènes de roulis et de dévers ont une durée très courte dans le temps, alors au cours de chaque seconde l'agent AO calcule un facteur moyen de correction de la hauteur de l'obstacle afin de corriger l'effet d'un éventuel roulis ou dévers sur le véhicule.

5.6. Résultats Expérimentaux :

Le système qu'on a développé est testé sur une séquence réel téléchargée de sites spécialisés, représentant la navigation d'une voiture dans un environnement routier, notre système localise, caractérise et suit l'évolution de tous les obstacles présents par des cadrans englobant en un temps de réponse acceptable, notre système calcule aussi les distances relatives de ces obstacles par rapport au dispositif de prise de vue, les images (figure 5.13) suivantes extraites de la séquence:

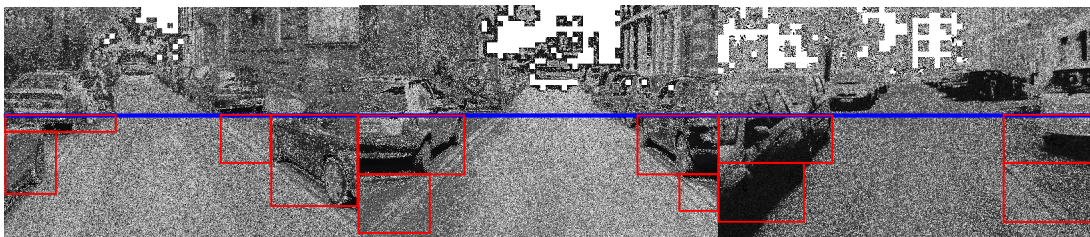


Figure 5.13 : Séquence d'images de test dans les conditions de navigation réelle (université de Karlsruhe).

On constate aussi une atténuation des effets du roulis et du dévers du véhicule, vu la technique de moyennage adaptée dans notre algorithme.

5.7. Conclusion du chapitre :

La flexibilité et l'autonomie offertes par les systèmes multiagents s'adaptent parfaitement à la nature de notre application de détection et de suivi des obstacles dans un environnement routier, les techniques multiagents prendront de plus en plus de places au sein du monde de la vision artificielle comme une nouvelle branche de vision intelligente.

Il est toute a fait naturel de tendre vers une spécialisation de notre agents détecteur AD pour lui procuré des propriétés cognitives qui lui permettrons d'effectué une classification des obstacles par famille, ce qui sera très intéressant pour l'adaptation d'une politique de sécurité correspondante au type d'obstacle, il est aussi intéressant de pouvoir détecter et de reconnaître les panneaux de signalisation afin de prévenir le conducteur en cas du non respect de la vitesse limite par exemple.

Cette spécialisation de nos agents ne nécessite pas un réaménagement total de notre système, mais seulement l'agent en question, cette modularité procure a l'approche multiagents toute sa puissance.

6. Conclusion et perspectives

6.1. Conclusion générale :

La popularité croissante des systèmes « intelligents » sur des véhicules de série bon marché actuellement commercialisés (comme ABS/EPS), démontre l'intérêt que porte l'industrie automobile à ce type de système, l'intégration de tels systèmes nécessite la maîtrise des technologies de perception de l'environnement du véhicule, mais la perception de l'état d'un véhicule et de son environnement reste un problème crucial dans le développement de l'automobile du futur, du fait de la complexité de ce problème, essentiellement lié à la complexité des situations observées, les constructeurs automobile tendent, depuis ces dernières décennies, à multiplier les études et les développements de dispositifs de perception et d'analyse, cette tendance est concrétisée par l'équipement de plus en plus de voitures de gamme assez populaire par des systèmes d'aide à la conduite.

L'objectif de cette thèse était d'étudier les possibilités d'application d'une approche parallèle pour implémenter une solution basée sur la stéréovision embarquée pour l'aide à la conduite, ce système procède par la perception de l'environnement d'un véhicule afin de détecter et de suivre les obstacles routiers.

La stéréovision est une technique très intéressante qui permet de recouvrir fidèlement la profondeur d'une scène à partir de deux images d'une façon très riche, malgré les nombreuses techniques proposées dans la littérature, ce problème est toujours le centre d'intérêt de nombreux chercheurs dans le domaine de la vision artificielle.

Nous avons présenté un nouveau algorithme rapide et robuste, ainsi que l'implantation optimisée d'un processus de mise en correspondance stéréoscopique, basé sur une approche par corrélations de critères photométriques, cette technique couplée à une approche multiagents offre d'énormes perspectives de recherche et de développement, et présente une méthode novatrice qui montre l'utilisabilité d'une telle approche parallèle pour aborder les problèmes de vision artificielle.

Les résultats obtenus sur des séquences synthétiques confirment les apports de cette approche, premièrement en ce qui concerne la robustesse de la méthode avec un score de bon appariement assez élevé même pour les pixels appartenant aux zones peu texturées ou occultées, on constate aussi une cadence d'exécution acceptable ce qui est primordial pour tendre vers des applications temps réel.

En ce qui concerne la détection d'obstacles en milieu routier, nous avons proposé une méthode basée sur la reconstruction 3D de l'environnement, le traitement de la carte de disparité dense permet au système multiagents la segmentation et la détection des obstacles routiers : pour ce faire, nous avons au préalable introduit une nouvelle image qui reprend les informations relatives à la segmentation de cette carte de disparité d'une façon semblable au concept v-disparité considérée

comme la plus performante à ce jour, cette nouvelle image qu'on a nommé carte de navigation CN se restreint à signaler les zones navigable ou non de la chaussée, ce qui nous permettra la détection de la voie de circulation et des obstacles, les résultats obtenus permettent de nous affranchir d'une modélisation très coûteuse et peu robuste du modèle géométrique de la route ainsi que les variations de pente et du dévers/roulis pour détecter les obstacles.

Cette approche est réalisée par le biais d'un système multiagents, cette nouvelle couche d'agents responsable de la construction de la carte de navigation et la localisation et le suivi de l'évolution des obstacles présents, sert d'intermédiaire et d'interprète entre la couche d'agents réactifs responsables de la mise en correspondance des paires d'images stéréoscopiques et l'utilisateur humain.

6.2. Perspectives :

La première orientation serai d'essayer d'exploité l'énorme possibilité d'exécution parallèle des calculs sur une unité dédié comme le GPU [Zhang 10] ce qui promet de réduire d'une façon spectaculaire le temps des calculs, cette orientation ne demande pas un réaménagement de notre algorithme qui est déjà de nature reparti, mais seulement d'effectuer quelques adaptations liée essentiellement a l'architecture matériel dédié (drivers matériel et binding).

Ensuite on doit travailler à augmenter la robustesse de notre algorithme de mise en correspondance stéréoscopique, et vu que la plupart des capteurs actuels permettent l'acquisition d'images en couleur qui apporte une information supplémentaire et plus discriminante, on peut proposer un nouveau formalisme pour utilisation de la couleur afin d'augmenter les performances dans le cadre de la mise en correspondance stéréoscopique de pixels.

Une autre extension envisageable de ce travail concerne la méthode de détection et de suivi des obstacles routiers, ici aussi, on peut exploiter la modularité et la flexibilité des systèmes multiagents pour proposer des méthodes d'identification plus sophistiquées (basées sur la reconnaissance de forme ou la distribution des couleurs par exemple) afin de pouvoir caractériser les différents obstacles par leurs natures (piéton, voiture,) pour différencier les types d'obstacles et donc d'appliquer une politique adaptée pour chacun d'entre eux, par exemple, on souhaitera protéger les plus vulnérables, comme les piétons ou les cyclistes.

Ces changement peuvent être inclus dans notre système par une simple spécialisation des agents sans opérer des changements conséquents, c'est l'apport principal de l'approche SMA par rapport à une approche classique qui nécessite un réaménagement radical pour pourvoir être réutilisé, l'autonomie des agents offre une meilleure flexibilité au système.

i. Bibliographie:

- [**Bertozzi 00**] M. Bertozzi, A. Broggi et A. Fascioli, "Vision-based Intelligent vehicles: state of the art and perspectives, *Journal of Robotics and Autonomous Systems*, 2000.
- [**Birchfield 98**] S. Birchfield et C. Tomasi, A pixel dissimilarity measure that is insensitive to image sampling, *IEEE Trans, On pattern analysis and machine intelligence*, 1998.
- [**Bobick 99**] A.F. Bobick et S.S. Intille, Large occlusion stereo, *International journal of computer vision*, 1999.
- [**Boguslav 09**] C. Boguslav, An introduction to 3D computer vision techniques and algorithms, livre de John Wiley & Sons, 2009.
- [**Broggi 99**] A. Broggi, M. Bertozzi et A. Fascioli, The 2000 km test of the ARGO vision-based autonomous vehicle, *IEEE Intelligent Systems*, 1999.
- [**Bücher 03**] T. Bücher, C.C. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage et W. Von Seelen, Image processing and behaviour planning for intelligent vehicles, *IEEE Trans. on Industrial Electronics*, 2003.
- [**Cabestaing 99**] F. Cabestaing, R. Yang, J.L. Bruelle et J.-G. Postaire, Real time processing of images sequences : application intrusion detection in secured areas, *Proceedings of International Conference On Quality Control by Artificial Vision*, Canada, 1999.
- [**Chambon 05**] S. Chambon, Mise en correspondance stéréoscopique d'images couleur en présence d'occultations, Thèse de doctorat, université Paul Sabatier, France, 2005.
- [**Chapuis 05**] J. Laneurit, R. Chapuis et F. Chausse. Accurate vehicle positioning on a numerical map, *International Journal of Control, Automation and Systems*, 2005.
- [**Chen 99**] Q. Chen et G. Medioni, Volumetric stereo matching method : Application to image-based modelling, In *Conf. on Computer Vision and Pattern Recognition*, Etats-Unis, 1999.
- [**Faugeras 92**] O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat et Z. Zhang, Quantitative and qualitative comparison of some area and feature-based stereo algorithms, *Allemagne*, 1992.
- [**Franke 98**] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paetzold et C. Wöhler, Autonomous driving goes downtown, in: *Proceedings of the IEEE Intelligent Vehicles Symposium '98*, Stuttgart, Germany, 1998.
- [**Gechter 05**] F. Gechter, O. Simonin, Conception de SMA réactifs pour la résolution de problèmes : Une approche basée sur l'environnement, *Soumission JFSMA05*, France, 2005.
- [**Hancock 98**] J. Hancock, M. Hebert et C. Thorpe, Laser intensity-based obstacle detection, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

- [**Hattori 98**] H. Hattori et A. Maki, Stereo matching with direct surface orientation recovery, In Ninth British Machine Vision Conference, 1998.
- [**Holland 75**] J. Holland, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and, artificial intelligence, Massachusetts Institute of Technology, MIT Press, 1975.
- [**Hong 04**] L. Hong et G. Chen, Segment-based stereo matching using graph cuts, Dans IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR, Etats-Unis, 2004.
- [**Issa 04**] H. Issa, Mise en correspondance stéréoscopique par algorithmes génétiques : nouveaux codages, Thèse de l'université des sciences et technologie de Lille, 2004.
- [**Itay 05**] G. Itay, B. Meny et S. Amnon, A monocular vision advance warning system for the automotive aftermarket, In Intelligent Vehicle Initiative of the world congress SAE. SAE, 2005.
- [**Jung 05**] B. Jung, Agent encapsulation in a cognitive vision MAS, Austrian Research Institute for Artificial Intelligence (OFAI), Australie, 2005.
- [**Kato 02**] T. Kato, Y. Ninomiya et I. Masaki, An obstacle detection method by fusion of radar and motion stereo, IEEE Trans. on Intelligent Transportation Systems, 2002.
- [**Koller 94**] D. Koller, Q.-T. Luong et J. Malik, Using binocular stereopsis for vision-based vehicle control, In Proceedings of the Intelligent Vehicles Symposium, 1994.
- [**Koller 95**] D. Koller, J. Malik, Q.-T. Luong et J. Weber, An integrated stereo-based approach to automatic vehicle guidance, in : Proceedings of the Fifth International Conference on Computer Vision, Boston, 1995.
- [**Kong 03**] D. Kong, H. Tao et H. H. Gonzalez-Banos, Sparse IBR using range space rendering, proceedings of british machine vision conference, Royaume-Uni, 2003.
- [**Koschan 96**] A. Koschan, Using perceptual attributes to obtain dense depth maps, Institute for Technical Informatics, Technical University of Berlin, Allemagne, 1996.
- [**Labayrade 02**] R. Labayrade, D. Aubert et J. Tarel, Real time obstacle detection in stereo vision on non flat road geometry through "V-Disparity" representation, in: Proceedings of the IEEE Intelligent Vehicle Symposium, France, 2002.
- [**Lemondé 05**] V. Lemondé, Stéréovision embarquée sur véhicule : de l'autocalibrage à la détection d'obstacles, thèse de doctorat, Institut National des Sciences Appliquées, France, 2005.
- [**Lou 05**] J. Lou, T.N. Tan, W.M. Hu, H. Yang et S.J. Maybank, 3-d model-based vehicle tracking. IP, 2005.
- [**Maraninchi 01**] F. Maraninchi et Y. Rémond, Argos : an automaton-based synchronous language. Computer Languages, 2001.

- [**Mayer 03**] H. Mayer, Analysis of means to improve cooperative disparity estimation. Proceedings of ISPRS conference on photogrammetric image analysis, Allemagne, 2003.
- [**Mohan 89**] R. Mohan, G. Medioni et R. Nevatia, Stereo error detection, correction, and evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 1989.
- [**Morat 08**] J. Morat, Vision stéréoscopique par ordinateur pour la détection et le suivi de cibles pour une application automobile, Thèse de doctorat, institut national polytechnique de Grenoble, France, 2008.
- [**Nasrabadi 92**] N. M. Nasrabadi et C. Y. Choo, Hopfield network for stereo vision correspondence, IEEE Transactions on pattern analysis and machine intelligence, 1992.
- [**Nedevschi 05**] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, Socol T, R. Schmidt et T. Graf, Driving environment perception using stereovision, in Proceedings of IEEE Intelligent Vehicles Symposium, 2005.
- [**Ohm 98**] J. Ohm, K. Gruneberg, E. Hendriks, E. Izquierdo, D. Kalivas, M. Karl et D. Papadimitos, A real-time hardware system for stereoscopic videoconferencing with viewpoint adaptation, Image Communication, 1998.
- [**Ohta 85**] Y. Ohta et T. Kanade, Stereo by intra and inter scanline search using dynamic programming, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985.
- [**Perez 05**] M. Perez, Stéréovision dense par traitement adaptatif temps réel : algorithmes et implantations », Thèse de l'Université des Sciences et de la Technologie de Lille, 2005.
- [**Ruichek 05**] Y. Ruichek, Perception de l'environnement par stereovision : Application à la sécurité dans les systèmes de transports terrestres, HDR, Université des sciences et technologies de Lille, France, 2005.
- [**Steux 02**] B. Steux, C. Laugeaux, L. Salesse et D. Wautier, Fade : A vehicle detection and tracking system featuring monocular color vision and radar data fusion, in: Proceedings of the IEEE Intelligent Vehicle Symposium, France, 2002.
- [**Szeliski 02**] D. Scharstein et R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, International journal of computer vision, 2002.
- [**Thorpe 98**] C. Thorpe, Vision and navigation, The Carnegie Mellon NavLab, Kluwer Academic Publishers, Etats-Unis, 1990.
- [**Tokoro 96**] S. Tokoro, Automotive application systems of a millimeter-wave radar, Proceedings of the IEEE International Intelligent Vehicles Symposium, Japon, 1996.
- [**Tsai 87**] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses, IEEE Trans, on Robotics and Automation, 1987.

[**Wang 03**] C. Wang, C. Thorpe et A. Suppe, Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds, Proceedings of the IEEE International Intelligent Vehicles Symposium, Etats-Unis, 2003.

[**Wei 04**] Y. Wei et L. Quan, Region-based progressive stereo matching, IEEE conference proceedings of computer vision and pattern recognition, Etats-Unis, 2004.

[**Widman 07**] G. Widmann, M. Daniels, L. Hamilton, L. Humm, B. Riley, J. Schiffmann, D. Schnelker et W. Wishon, Comparison of lidar-based and radar-based adaptive cruise control systems, In SAE transactions, 2007.

[**Wikipedia**] encyclopédie libre en ligne.

[**Williamson 98**] T. Williamson, A High-Performance stereo vision system for obstacle detection, thèse de doctorat, 1998.

[**Williamson 99**] R. Williamson et C. Thorpe, A trinocular stereo system for highway obstacle detection, Proceedings of the IEEE International Conference on Robotics and Automation, Etats-Unis, 1999.

[**Woetzel 04**] J. Woetzel et R. Koch, Multi-Camera real-time depth estimation with discontinuity handling on PC graphics hardware, IEEE conference proceedings of international conference on pattern recognition, Royaume-Uni, 2004.

[**Yang 04**] R. Yang, M. Pollefeys et S. Li, Improved real-time stereo on commodity graphics hardware, In Proceedings of Computer Vision and Pattern Recognition Workshop, 2004.

[**Zabih 94**] R. Zabih et J. Woodfill, Non-parametric local transforms for computing visual correspondence, Proceedings of european conference on computer vision, Suede, 1994.

[**Zhang 96**] Z. Zhang, Determining the epipolar geometry and its uncertainty : A review, Rapport de recherche, Institut national de recherche en informatique et en automatique, INRIA, 1996.

[**Zhang 00**] Z. Zhang, A flexible new technique for camera calibration. IEEE Trans, on Pattern Analysis and Machine Intelligence, 2000.

[**Zhang 10**] J. Zhang et Z. Xu, Parallel Computation for Stereovision Obstacle Detection of Autonomous Vehicles Using GPU, LSMS/ICSEE 2010, Part I, LNCS 6328, Allemagne, 2010.

Séquences d'image de tests téléchargées auprès de :

<http://cat.middlebury.edu/stereo/data.html>

<http://www.vision.deis.unibo.it/smatt/stereo.htm>

Université de Karlsruhe, Allemagne.

<http://www.rainsoft.de/software/datasets.html>

ii. Annexe : application et quelques résultats :

Nous avons développé notre application sous le langage Java, ce langage est parfaitement adapté au multithreading qui est la base de notre algorithme, ajouté a cela la modularité et la flexibilité indispensable offerte par Java.

Notre approche est basée sur le développement d'un nombre de classes indépendantes, ou chaque classe est chargée d'accomplir une tâche spécifique, allant du calcul de la matrice Census jusqu'à la détection et la segmentation d'obstacles en passant par la mise en correspondance stéréoscopique multithread.

Afin d'évaluer la pertinence de notre système final, on test le système ainsi développé en utilisant une data set stéréoscopique disponible sur internet sur le site de l'université de Karlsruhe (avec l'autorisation des propriétaires des images), ces séquences d'images stéréoscopique représente la navigation d'un véhicule en milieu routier, les images sont rectifiées au préalable.

L'image suivante représente une prise de vue stéréoscopique d'une scène routière classique, on constate une route plane au milieu en bas, et un ensemble d'obstacle sur les deux cotés de cette route, on constate que notre approche offre la possibilité de modifier un paramètre très important de notre approche qui est la taille de la fenêtre d'appariement qui est par défaut mise à 3.



Sur l'onglet gauche on constate deux partie, la première en haut représente un explorateur qui permet de spécifier les chemin exacte contenant notre data set, alors qu'en bas, on constate la division du traitement en plusieurs phase, cette division est opérée suivant l'enchaînement des

phases expliquées dans ce mémoire, alors on commence par les pré-calculs, ensuite on procède à la mise en correspondance (la phase multithreading), pour ensuite dessiner la carte de disparité qui représente la structure 3D de la scène observée, pour arriver enfin à la phase de détection et de segmentation des obstacles présents sur cette scène, après la fin de tous les traitements, on peut avoir quelque statistique (notamment en ce qui concerne le temps de chaque phase).

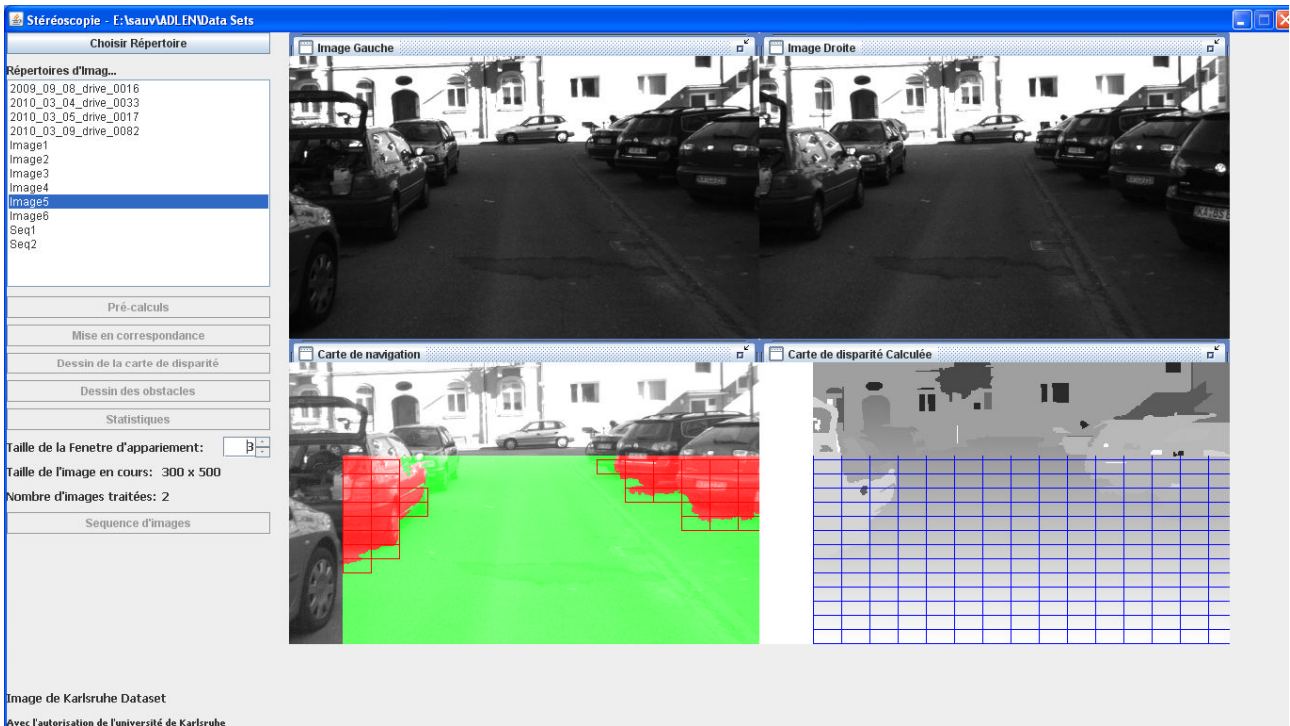


L'image au-dessus, montre l'exécution de l'ensemble des phases sur la paire d'image sélectionnée précédemment, on constate une parfaite détection de la route praticable (colorée en vert) et une segmentation sans faute des obstacles présents (colorée en rouge).

L'image suivante représente les statistiques du temps d'exécution de chaque phase de notre traitement en millisecondes, on constate que la phase du pré calculs est la phase la plus longue, elle comporte un calcul matriciel gourmand en temps processeur, cette phase peut être significativement réduite en utilisant une approche totalement parallèle sur architecture dédiée comme le GPU.



Sur l'image qui suit, on constate le choix d'une paire d'image contenant une perturbation de texture de la route juste devant le véhicule de test, cette différence de texture ne représente pas de déformation dans la chaussée et donc permet le passage du véhicule, on constate que le système ne donne aucune fausse alerte dans ce cas de figure.



Afin d'évaluer la cadence de traitement d'une façon plus fiable, on offre la possibilité d'exécuter le système sur des séquences d'images, l'image qui suit montre la détection des obstacles routiers sur la séquence d'image Seq2 comportant 42 paires d'images, dans l'ensemble, on constate avec l'évolution de la navigation une détection assez fidèle de la chaussée praticable et des obstacles présents, cela malgré les variations des conditions de luminosité entre les couples successive d'images.

Le temps globales du traitement de l'ensemble des paires d'images de cette séquence est de 13 657 ms, ce qui nous donne une cadence de traitement de 325 ms par paire (une vitesse de défilement de 3 paires/seconde), cette vitesse reste loin de la vitesse de prise de vue de la séquence Seq2 qui est de 10 paires/seconde (cette vitesse est précisée par les propriétaires de la séquence d'image), dans une future version de notre système, on propose de profiter de l'approche parallèle adaptée dans notre algorithme, et de la flexibilité offerte par le langage Java pour transférer l'exécution de l'ensemble des calculs sur une architecture graphique GPU massivement parallèle, cette technique nous permettra de fractionner le temps de calculs par un facteur de dix, et réduire ainsi significativement les deux premières phases gourmandes en temps de calculs (presque 80% du temps global), ce qui nous permettra d'atteindre facilement un défilement temps réel.

