

RAPPORT DE STAGE

3<sup>ème</sup> ANNEE 2006/2007

VISION 3D STEREOSCOPIQUE ET CAMERAS 3D.

Vincent MALLERON

STMicroelectronics - AST

Tuteur Universitaire : Pr. Christian DUFAZA

Tuteur Industriel : Lionel MARTIN





## Remerciements

Je tiens tout d'abord à remercier mon tuteur industriel de stage, M. Lionel Martin, manager des projets recherche et innovation au sein d'AST Rousset, pour les conseils et l'expérience qu'il m'a apporté, pour la confiance qu'il m'a témoigné ainsi que pour m'avoir permis d'effectuer ce stage au sein d'une entreprise de dimension internationale.

Je tiens également à remercier M. Christian Dufaza, professeur au sein de l'école polytechnique universitaire de Marseille et membre du laboratoire L.2.M.P pour sa présence et ses conseils tout au long du stage.

Je remercie également conjointement messieurs L.Martin et C.Dufaza pour le sujet intéressant sur lequel ils m'ont permis de travailler durant une année au cours de deux stages et d'un projet de fin d'étude.

Je remercie également à remercier l'ensemble des membres de l'équipe d'AST Rousset et particulièrement M. Bernard KASSER, manager d'AST Rousset pour m'avoir accueilli parmi eux.

Enfin, je tiens à remercier l'ensemble des stagiaires et doctorants présents lors de la période de mon stage. Je remercie particulièrement M. Yannick Del-Confetto ainsi que M.Simon Quéré pour leur aide et leurs conseils.

---

## Sommaire

Introduction .....	1
<b>I Présentation de STMicroelectronics.....</b>	<b>2</b>
I.1 STMicroelectronics dans le monde .....	2
I.2 STMicroelectronics et le site de Rousset .....	2
I.3 La division Advanced System Technology (A.S.T) .....	3
I.3.1 Présentation d'A.S.T.....	4
I.3.2 Le Service A.S.T Rousset.....	4
<b>II La Vision Stéréoscopique.....</b>	<b>5</b>
II.1 Principe .....	5
II.2 Systèmes Existants.....	7
II.3 Calibration & Géométrie épipolaire .....	8
II.3.1 Calibration d'une Caméra .....	8
II.3.2 Calibration Stéréoscopique .....	10
II.4 Principe de Rectification épipolaire.....	11
II.4.1 Principe .....	11
II.4.2 Méthodes de Rectification .....	12
II.5 Les algorithmes d'appariement.....	14
II.5.1 Mesures de Similitudes.....	15
II.5.2 Contraintes d'appariement .....	19
<b>III Programme de Vision Stéréoscopique développé.....</b>	<b>23</b>
III.1 Cahier des charges .....	23
III.2 Fonctionnalités proposées .....	23
III.2.1 Ouverture et synchronisation des flux vidéo .....	24
III.2.2 Calibration des caméras .....	24
III.2.3 Correction de la distorsion.....	25
III.2.4 Estimation de la géométrie épipolaire.....	25
III.2.5 Rectification Epipolaire .....	26
III.2.6 Appariement .....	26
III.2.7 Estimation de la distance des points d'une scène .....	27

---

IV Notre Algorithme de Corrélation .....	30
IV.1 Principe de l'algorithme .....	30
IV.2 Mesures de Similitudes implémentées .....	31
IV.3 Exemples & résultats.....	32
IV.3.1 Jeu d'images stéréoscopiques : Castle.....	32
IV.3.2 Jeu d'images stéréoscopiques : Plan Incliné .....	33
IV.3.3 Jeu d'images stéréoscopiques : Parcmètres.....	34
IV.3.4 Jeu d'images stéréoscopiques :Tsukuba .....	35
IV.3.5 Jeu d'images stéréoscopiques : L.2.M.P.....	36
V Evolutions planifiées .....	37
Conclusion.....	38
Bibliographie.....	39
Sommaire des Annexes .....	40

---

## Table des Figures

Figure 1 – Synoptique général de la vision par ordinateur.....	6
Figure 2 - Modèle géométrique d'une caméra .....	8
Figure 3 - Objet de Calibrage .....	9
Figure 4 - Modèle géométrique d'un capteur stéréoscopique .....	10
Figure 5 - Principe de rectification épipolaire .....	12
Figure 6 - Passage du repère caméra au repère caméra rectifiée.....	13
Figure 7 - Jeu d'image Stéréoscopique avant et après rectification.....	13
Figure 8 - Géométrie épipolaire estimée .....	15
Figure 9 - Cartes de Disparités obtenues avec différentes mesures de similitude.....	16
Figure 10 - Disparity map after Rank Transform + SAD.....	17
Figure 11 - Principe de la transformée de Census .....	18
Figure 12 - Principe de la distance de Hamming.....	19
Figure 13 - Carte de disparité après Census+Hamming .....	19
Figure 14 - Contrainte Épipoilaire.....	20
Figure 15 - Contrainte d'unicité.....	21
Figure 16 - Contrainte d'ordre .....	21
Figure 17 - Correction de la distorsion optique.....	25
Figure 18 - Géométrie épipolaire estimée .....	26
Figure 19 - Carte de Disparité en temps réel .....	27
Figure 20 - Système stéréoscopique et Triangulation .....	28
Figure 21 - Recherche de Correspondant .....	30
Figure 22 - Carte de Fiabilité.....	31
Figure 23 - Castle, images gauche, droite, carte de disparité .....	32
Figure 24 - Disparités Réelles .....	33
Figure 25 - Set d'image Plan Incliné Gauche, Droite et Carte de disparité associée .....	33
Figure 26 - Carte de disparité réelle .....	33
Figure 27 - Parcètres : (a) Gauche (b) Droite (c) SSD (d) Census (e) SAD.....	34
Figure 28 - Tsukuba : (a) Gauche (b) Droite (c) SAD (d) Census (e) SSD.....	35
Figure 29 - L2MP : (a) Gauche (b) Droite (c) Census 3x3 (d) 5x5 (e) 7x7 .....	36

## Introduction

Dans le cadre de la validation de mon diplôme d'ingénieur ainsi que du Master Minelec (Aix-Marseille I) j'ai choisi de réaliser un stage au sein de la société STMicroelectronics et plus particulièrement de la division A.S.T (Advanced System Technology).

Ce choix à été réalisé de façon à poursuivre un travail débuté lors de mon stage de seconde année sur le sujet de la vision 3D, il m'a également permis d'intégrer une grande entreprise et à ce titre de confirmer mon orientation professionnelle future.

Ce stage entre dans le cadre d'un projet de collaboration PACA (PS3-16) entre STMicroelectronics et le laboratoire L.2.M.P. L'objectif est de concevoir un modèle de système 3D fonctionnel sur support informatique afin d'évaluer les possibilités d'intégration on chip d'un système de vision 3D. Ce stage fait suite à une étude bibliographique réalisée sur le sujet au cours de mon stage de seconde année ainsi qu'à un projet de fin d'étude de 100h sur le sujet.

Ce rapport contient une présentation de l'entreprise, de la division ainsi que du service dans lequel ce travail à été réalisé. On poursuivra par une présentation des primitives nécessaires à la compréhension de la vision stéréoscopique. On exposera ensuite les deux phases principales de ce travail :

- dans un premier temps la conception d'un programme de vision 3D fonctionnel
- dans un second temps la réalisation et la comparaison de différents algorithmes pour la vision 3D.

On terminera par les évolutions prévues pour la suite du stage.

## **I Présentation de STMicroelectronics**

### **I.1 STMicroelectronics dans le monde**

STMicroelectronics, société franco-italienne de renommée mondiale dans le domaine de la microélectronique est née de la fusion en 1987 de SGS-Microelettronica (groupe italien) et THOMSON Semi-conducteurs (groupe français). La société devint alors SGS-THOMSON. C'est en 1998 à l'issue du départ du groupe THOMSON du groupement d'actionnaire que la société a prit le nom de STMicroelectronics.

Pour citer quelques chiffres, STMicroelectronics c'est environ 50000 salariés dans le monde, 15 sites de productions principaux, 16 centres de R&D avancés, 39 centres de conception et d'applications, 78 bureaux de ventes directes dans 36 pays et un chiffre d'affaires de 9.85 Milliards de dollars en 2006.

La mission principale de ce grand groupe tient en une phrase :

*« Offrir l'indépendance stratégique à l'ensemble de ses partenaires dans le monde en étant un fournisseur de semi-conducteur profitable, viable et disposant d'une large gamme de produits. »*

En effet avec plus de 380 Milliards de composants vendus en 2004, STMicroelectronics produit des composants de haute technologie servant dans de multiples secteurs d'activité comme l'imagerie, les cartes bancaires, les loisirs, l'automobile, l'industrie, la recherche, la téléphonie, le multimédia pour ne citer qu'eux.

### **I.2 STMicroelectronics et le site de Rousset**

Au pied de la Sainte Victoire, à deux pas d'Aix-en-Provence, le site de Rousset occupe une place stratégique au sein du Groupe STMicroelectronics.

ST Rousset est l'acteur majeur de la microélectronique en Europe, au cœur de l'un des premiers pôles de microélectronique français et européen, avec 40% de la production française.

## Présentation de STMicroelectronics

ST Rousset c'est :

- Un centre de production de 2 milliards de puces par an, soit 15% de la production mondiale de STMicroelectronics.
- Un atelier de test de plaquette.
- Un centre de Recherche & Développement à l'origine de nombreux partenariats auprès d'équipementiers, sous-traitants, laboratoires publics, acteurs de l'enseignement supérieur local pour mettre en réseau leurs compétences et savoir-faire.
- Des équipes de développement et d'industrialisation de produits, qui génèrent un chiffre d'affaires de l'ordre de 700 millions d'Euros par an.
- 1,7 milliard d'euros investis depuis 12 ans dont 85 Millions en 2005.
- Près de 2850 salariés, dont plus de 1000 ingénieurs.
- 100 millions d'euros de dépenses annuelles en R&D.
- 700 brevets déposés sur les dix dernières années qui font de Rousset un des principaux contributeurs de STMicroelectronics.
- Les technologies de fabrication de semi-conducteur les plus avancées de 0,35 à 0,09 microns sont utilisées sur le site.
- Plus de 40 millions d'euros d'achats sont réalisés en région PACA chaque année.
- Une activité qui induit directement près de 9 000 emplois (fournisseurs, sous-traitants, prestataires de services...) et indirectement près de 15 000 emplois dans les services à la population.
- Un salarié sur deux est originaire de la région.
- Le site accueille plus de 20 000 visiteurs par an.

Le Groupe STMicroelectronics réalise à Rousset une part importante de ses investissements, pour permettre au site d'assurer son développement et de demeurer compétitif dans un marché mondial très concurrentiel.

### **I.3 La division Advanced System Technology (AST)**

#### **I.3.1 Présentation d'A.S.T**

A.S.T. (Advanced System Technology) est un service de recherche et de développement au niveau système de STMicroelectronics. A l'image du reste de la société, ce service est multinational et multiculturel : il est implanté à Genève et Lugano (Suisse), à San José, San Diego, Portland et Boston (USA), à Singapour (Asie), à Agrate, Catania et Lecce (Italie) et enfin Grenoble et Rousset (France). Cette position géographique permet de confronter différentes compétences, différents points de vue et différentes méthodes de travail dans la réalisation de projet.

Le service dépend directement de la direction générale. La mission du département AST est de fournir à la société des solutions techniques innovantes en prévision des futurs besoins des divisions de l'entreprise ainsi que d'explorer de futures opportunités de marché. AST doit pour cela avoir une vision du marché à 4 ou 5 ans. Il a pour but le développement d'architectures avancées qui, à long terme, permettront à l'entreprise de commercialiser des produits qui stratégiquement l'amèneront à maintenir un rôle de leader mondial dans les domaines concernés. Les recherches portent sur les produits numériques, les communications sans fil (wireless), les cartes à puce, la domotique, l'électronique automobile, l'imagerie.

A.S.T. apporte des compétences et des connaissances dans tous ces domaines et transmet son savoir-faire aux divisions afin qu'elles puissent l'intégrer dans leurs nouveaux produits.

#### **I.3.2 Le Service A.S.T Rousset**

L'équipe implantée à Rousset est composée de 13 personnes, en grande majorité des ingénieurs. Cette équipe travaille sur le thème de la sécurité : une partie est spécialisée dans la conception plateformes sécurisées, la transmission numérique sécurisée (cryptographie) et l'autre partie travaille sur biométrie, le traitement de l'image (reconnaissance d'individu, interface homme machine) en restant tournée vers l'innovation.

## II La Vision Stéréoscopique

### II.1 Principe

La Vision Stéréoscopique vise à reconstruire la structure 3D d'une scène. Afin de réaliser cette reconstruction il est nécessaire de connaître les coordonnées 3D de tous les points d'une scène. La scène est représentée par un couple d'images stéréoscopique. Ces images sont deux représentations de la scène prises sous des angles différents. Chaque point de la scène est projeté dans les images de la paire stéréoscopique à des positions différentes. En vision stéréoscopique pour disposer des coordonnées 3D d'un point il faut disposer des coordonnées de ses deux projections, respectivement dans les images gauches et droites.

L'obtention des coordonnées de ces deux projections d'un même point 3D est réalisée grâce à une phase d'appariement qui vise à trouver, pour un point donné dans une image, son point correspondant dans l'autre image. Ces deux points sont les projections respectives du même point 3D dans les images droite et gauche.

Les algorithmes d'appariement utilisent diverses techniques qui sont soit locales et basées sur la comparaison de fenêtre, dites « area-based methods », soit globales. Dans ce cas elles reposent sur la minimisation d'une fonction de coût globale. Ces dernières permettent d'obtenir des appariements denses et de bonne qualité mais ont un coût de calcul élevé et ne sont donc pas adaptées à une application en temps réel. A l'inverse les méthodes locales sont relativement rapides et permettent d'obtenir une bonne fiabilité dans l'appariement.

Par ailleurs il faut remarquer qu'il existe un certain nombre de points de l'image qui n'ont pas de correspondants. Ceci est dû au phénomène d'occultation : un point peut être situé hors du champ de vision d'une des deux caméras ou être masqué par un objet dans une image et non dans l'autre.

Le principe de la vision stéréoscopique est résumé dans le schéma synoptique de la figure 1. Il est à noter qu'en amont des étapes de la figure 1 il est nécessaire de réaliser la calibration du système stéréoscopique de façon à disposer des paramètres intrinsèques et extrinsèques du système. Ces paramètres seront essentiels dans les phases de rectification et de calcul des positions 3D des points.

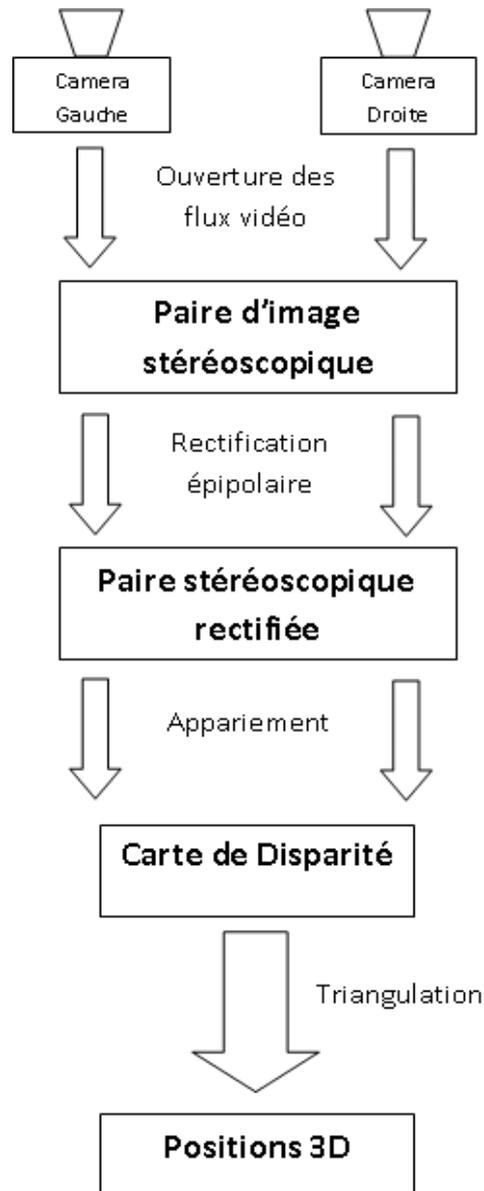


Figure 1 – Synoptique général de la vision par ordinateur

Enfin, une fois les positions 3D des points de la scène connues on peut réaliser une reconstruction 3D de la scène. En théorie cette reconstruction pourrait être dense mais en pratique on dispose principalement d'un certain nombre de surfaces positionnées à des distances différentes. Cette information de distance est suffisante dans le cadre d'une détection d'obstacle, d'un suivi de personnes mais peut s'avérer limitée si l'on cherche à avoir une reconstruction dense. Il faudra donc s'orienter vers des algorithmes offrant les meilleures performances possibles en termes de détection des disparités.

## II.2 Systèmes existants – État de l’Art

Un certain nombre de systèmes utilisant la vision stéréoscopique sont actuellement commercialisés. [1][5][7] On peut par exemple citer les sociétés TYZX ou Point Grey qui commercialisent toutes deux des modules de vision 3D stéréoscopique comprenant un module de vision (2 caméras fixées et pré-calibrées) ainsi qu’une interface avec un PC. Ces modules sont pilotés grâce à des API propriétaires. Il existe par ailleurs un certain nombre d’applications utilisant ces modules. Ces applications réalisent des fonctions telles que la détection et le suivi de personnes, l’évaluation de distance ou la navigation virtuelle dans l’environnement 3D.

L’offre existante est donc relativement fournie mais les systèmes existants ont des coûts élevés (de 3000\$ à 10000 \$). Il n’existe par ailleurs que très peu de systèmes intégrés sur un module unique offrant de bonnes performances. Le but de cette étude est donc d’étudier la faisabilité de la conception d’un système de vision 3D embarqué sur Silicium.

On note enfin que les systèmes existants sont de deux types principaux : systèmes binoculaires ou trinoculaires (en ligne ou en triangle). L’ajout d’une troisième caméra permet de supprimer certaines occlusions et donc de déterminer une carte de disparité plus dense.

## II.3 Calibration et Géométrie épipolaire

### II.3.1 Calibration d’une Caméra

#### II.3.1.1 *Modèle géométrique d’une caméra*

Afin de définir le modèle géométrique de notre caméra on va considérer deux transformations : une projection perspective qui transforme le point 3D de l’espace en un point 2D dans le repère caméra et une transformation caméra image qui va permettre d’établir le passage entre le repère caméra et le repère image. On peut schématiser une caméra comme sur la figure 2.

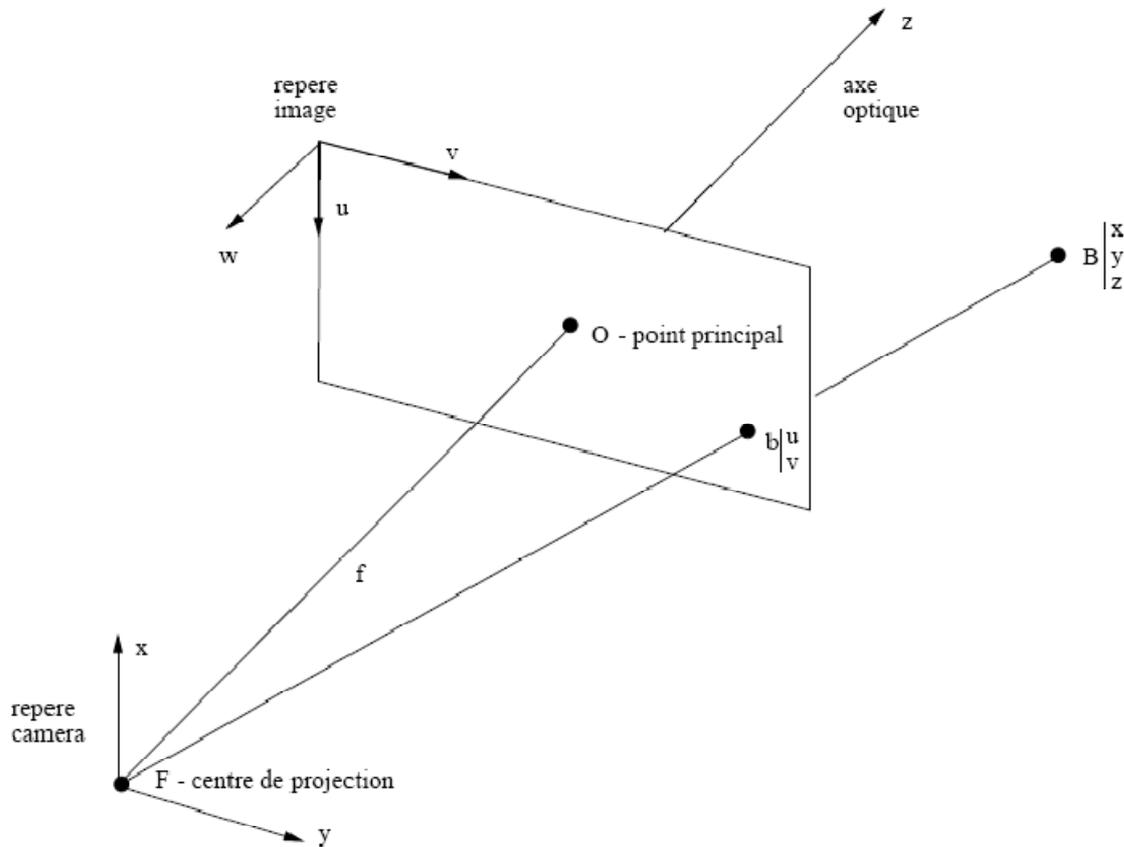


Figure 2 - Modèle géométrique d'une caméra

Sur cette figure, le point  $F$  représente le centre de projection. Le point  $B$  est le point 3D et  $b$  sa projection 2D dans le repère image. Le point principal  $O$  correspond à la projection du centre de projection  $F$  du repère caméra vers le repère image. La distance  $f$  est la distance focale de la caméra et représente la distance entre le repère image et le repère caméra. La connaissance de ces distances et points caractéristiques permet d'estimer la géométrie d'un capteur. Le but de la calibration est donc d'estimer ces paramètres afin de disposer de la connaissance de cette géométrie [13].

On distingue deux types de paramètres principaux : les paramètres intrinsèques à la caméra et ses paramètres extrinsèques [4].

### II.3.1.2 Paramètres Intrinsèques

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

$$C = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

$(u_0, v_0)$  sont les coordonnées  $(x, y)$  du point principal dans le repère image. On a par ailleurs  $\alpha_u = -k_u \cdot f$  et  $\alpha_v = k_v \cdot f$  où  $k_u$  et  $k_v$  sont les facteurs d'échelles respectivement vertical et horizontal. On note que les coordonnées sont exprimées en pixels et les coefficients d'échelle en pixels/mm<sup>2</sup>.

Afin de déterminer ces éléments caractéristiques le principe de la calibration est le suivant : on observe une mire ( figure 3 ) dont les propriétés sont connues (en général une grille où le nombre de carrés et leur espacement est connu). On détecte un certain nombre de points caractéristiques (intersections entre les carrés noirs et les carrés blancs) et on peut déduire de leur position dans les images capturées les différents paramètres de calibration.

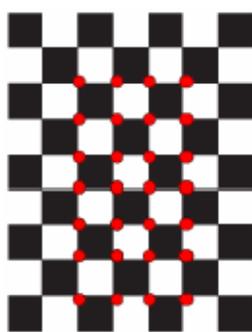


Figure 3 - Objet de Calibration

Afin d'avoir une bonne précision sur les valeurs des paramètres de calibration un certain nombre de prises de vues de l'objet de calibration sont nécessaires. (Voir étude comparative en annexe 4).

### II.3.1.3 Paramètres Extrinsèques

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

$$A = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## II.3.2 Calibration Stéréoscopique

### II.3.2.1 Modèle géométrique du capteur stéréoscopique

Le modèle géométrique du capteur stéréo peut être représenté de la façon suivante :

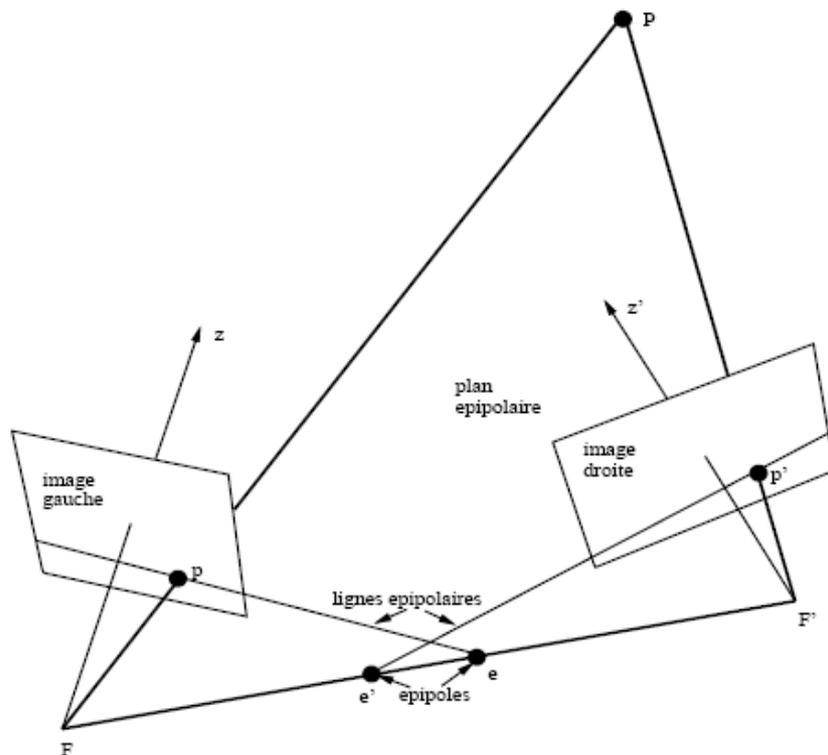


Figure 4 - Modèle géométrique d'un capteur stéréoscopique

Sur ce schéma on considère le couple de caméra stéréoscopique et on peut introduire la notion de géométrie épipolaire. En effet, on peut démontrer que pour chaque point de

l'image gauche (resp. droite) si ce point possède un correspondant dans l'image de droite (resp. gauche) celui-ci se trouve sur une droite appelée droite épipolaire. Toutes les droites épipolaires sont concourantes en un point appelé épipôle. Il existe donc un épipôle droit et un épipôle gauche. Grâce à la calibration stéréoscopique on pourra disposer de tous les paramètres nécessaires au calcul des équations des droites épipolaires. Ces équations pourront être déterminées à partir de la matrice fondamentale que l'on pourra elle-même calculer ou estimer. (Voir Annexe 1). La matrice fondamentale décrit la géométrie du capteur stéréoscopique, géométrie que l'on appelle géométrie épipolaire.

### II.3.2.2 Calibration

Le but de la calibration stéréo est de déterminer la matrice de passage du repère image gauche au repère image droite. Dans notre cas on va déterminer la matrice de passage du repère de la caméra gauche au repère de la caméra droite. On pourra alors aisément repasser du repère caméra au repère image en utilisant la matrice caméra C déterminée en amont pour chaque caméra. La matrice de passage du repère caméra gauche au repère caméra droite est la matrice stéréo  $A_s$ .

$$A_s = \begin{pmatrix} r_{11} & r_{12} & r_{13} & b_x \\ r_{21} & r_{22} & r_{23} & b_y \\ r_{31} & r_{32} & r_{33} & b_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$(b_x, b_y, b_z)$  correspond au vecteur de translation  $[FF']$ , et la matrice  $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$  la matrice de rotation entre le repère caméra de gauche et le repère caméra de droite.

## II.4 Principe de rectification épipolaire

### II.4.1 Principe

La rectification épipolaire se base sur une modification de la géométrie épipolaire présentée au paragraphe précédent (voir figure 4). Le but recherché est la parallélisation des lignes épipolaires de façon à simplifier la recherche de correspondants entre deux images. En effet, après rectification chaque point d'une image trouve son correspondant sur la même ligne que lui dans l'image correspondante [10],[11].

Il est nettement plus simple de mettre en œuvre cette recherche ligne par ligne plutôt que d'utiliser les équations de ligne épipolaire. La rectification devient donc un préalable à l'appariement de deux images.

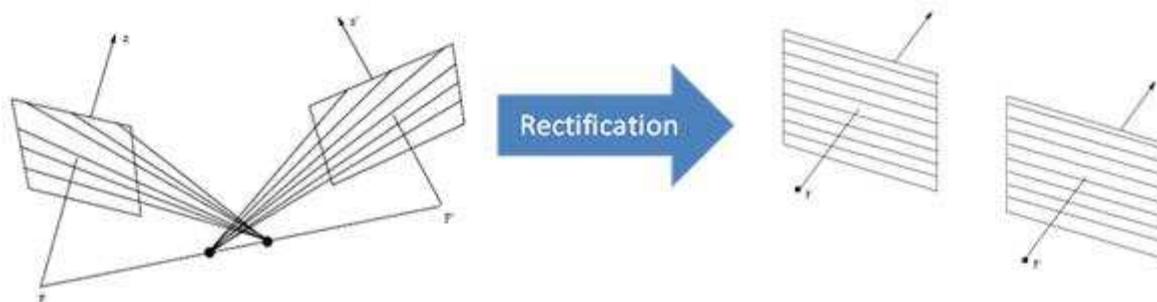


Figure 5 - Principe de rectification épipolaire

La figure 5 ci-dessus présente un ensemble de droites épipolaires avant et après rectification. On constate bien que les droites épipolaires sont parallèles après rectification : la recherche de correspondants peut se faire ligne par ligne. On peut par ailleurs noter que les épipôles sont rejetés à l'infini.

### II.4.2 Méthodes de rectification

Il existe un nombre important de méthodes de rectification, qui possèdent chacune leurs propriétés spécifiques, induisant plus ou moins de distorsion etc. Quelle que soit la méthode utilisée la rectification est une rotation spatiale qui vise à ramener deux images dans un repère coplanaire. Dans notre cas on a implémenté et utilisé un algorithme exposé par A. Fusiello [10]. Le principe est de définir un nouveau repère (repère image rectifié) ainsi qu'un nouveau jeu de matrices de projection perspective qui permettent, en ayant la connaissance des matrices de projection perspective initiales de passer, pour chaque caméra, du repère caméra initial au repère caméra rectifiée (figure 6). Une fois l'image connue dans ce repère caméra rectifiée on peut aisément passer dans le repère image rectifié en utilisant les paramètres intrinsèques. Les matrices de projection perspectives sont fixes une fois la calibration réalisée : le calcul des matrices de rectification (matrices permettant le passage du repère image au repère image rectifiée) est donc à réaliser une seule fois après chaque calibration. Ces matrices pourront ensuite être utilisées afin de rectifier un jeu d'images stéréoscopiques (figure 7).

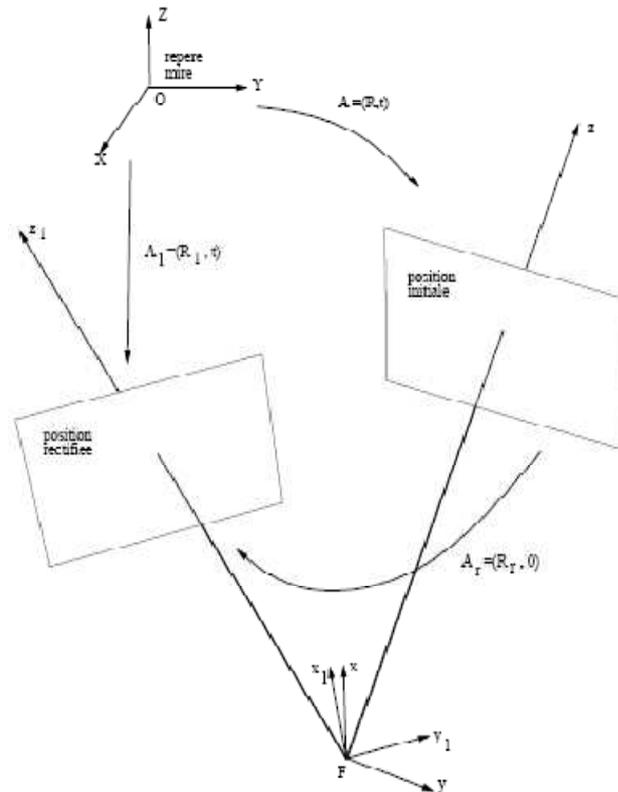


Figure 6 - Passage du repère caméra au repère caméra rectifiée

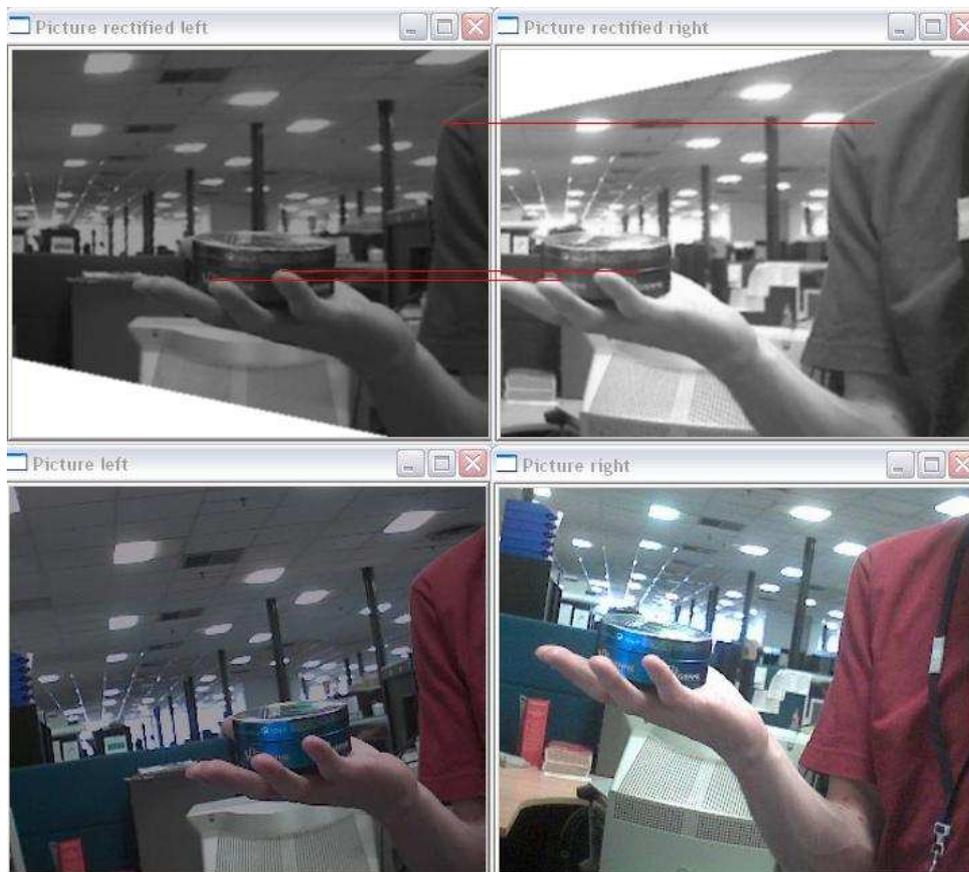


Figure 7 - Jeu d'image Stéréoscopique avant et après rectification

## II.5 Les algorithmes d'appariement

Les algorithmes d'appariement sont au cœur du processus de vision 3D. Ils consistent simplement en une association des points correspondant entre deux images. Ces points représentent les projections gauches et droites d'un même point 3D. Le résultat de cette phase est une image de disparité relative entre les images gauche et droite. La disparité correspond à la distance sur la ligne entre les coordonnées en abscisse de chacune des projections, elle est inversement proportionnelle à la distance réelle. Cette étape correspond, en vision humaine à l'association par le cerveau des images perçues par l'œil gauche et l'œil droit. Les algorithmes existants sont nombreux et varient selon divers paramètres : mesure de similitude entre les pixels utilisée, différentes contraintes implémentées, transformations et filtrages préalables appliqués.

### II.5.1 Mesures de similitude

La mesure de similitude constitue le cœur de l'algorithme d'appariement. Ce sont ces mesures qui vont permettre de déterminer les paires de points. Ce sont des mesures algébriques qui permettent de calculer la ressemblance entre deux pixels. En pratique cette mesure de ressemblance est réalisée sur une fenêtre et non pas sur un seul et unique pixel. On parle donc de fenêtre de corrélation. Pour être appariés, deux points devront être similaires et avoir un voisinage similaire. [2]

Il existe deux types principaux de mesures de similitude :

- Les mesures directes, qui travaillent directement sur l'image obtenue grâce à des comparaisons d'intensité.
- Les mesures dites non-paramétriques qui déterminent les correspondances après passage de l'image dans une fonction de transformation non paramétrique (de type transformée du Rang ou transformée de Census) Ces transformations seront explicitées au paragraphe II.5.1.2

#### II.5.1.1 Mesures directes

Les mesures de similitude directes sont des mesures qui travaillent sur la ressemblance directe en intensité entre les pixels des images à appairier. Afin de mesurer la similitude entre deux voisinages de pixels on va réaliser la somme des similitudes sur un voisinage du pixel à appairier. Ce voisinage est la fenêtre de corrélation.

Différentes mesures de similitude sont présentées dans le tableau ci-dessous :

Sum of Absolute Differences	SAD	$\sum_{(u,v) \in W}  I_1(u,v) - I_2(x+u, y+v) $
Zero mean Sum of Absolute Differences	ZSAD	$\sum_{(u,v) \in W}  (I_1(u,v) - \overline{I_1}) - (I_2(x+u, y+v) - \overline{I_2}) $
Sum of Squared Differences	SSD	$\sum_{(u,v) \in W} (I_1(u,v) - I_2(x+u, y+v))^2$
Zero mean Sum of Squared Differences	ZSSD	$\sum_{(u,v) \in W} ((I_1(u,v) - \overline{I_1}) - (I_2(x+u, y+v) - \overline{I_2}))^2$
Normalised Cross Correlation	NCC	$\frac{\sum_{(u,v) \in W} I_1(u,v) \cdot I_2(x+u, y+v)}{\sqrt{\sum_{(u,v) \in W} I_1^2(u,v) \cdot \sum_{(u,v) \in W} I_2^2(x+u, y+v)}}$
Zero mean Normalised Cross Correlation	ZNCC	$\frac{\sum_{(u,v) \in W} (I_1(u,v) - \overline{I_1}) \cdot (I_2(x+u, y+v) - \overline{I_2})}{\sqrt{\sum_{(u,v) \in W} (I_1(u,v) - \overline{I_1})^2 \cdot \sum_{(u,v) \in W} (I_2(x+u, y+v) - \overline{I_2(x,y)})^2}}$

Figure 8 : Mesures de Similitude

Les techniques de type SAD, SSD, NCC ont un inconvénient majeur : elles sont extrêmement sensibles aux différences de gain et de luminosité entre les deux images de la paire stéréoscopique. Pour palier à ce problème on a développé les techniques de type ZSAD, ZSSD, ZNCC qui normalisent l'intensité par rapport à la valeur de l'intensité moyenne sur la fenêtre étudiée. Cela permet de s'affranchir des problèmes dus aux différences de luminosité entre les images.

Par ailleurs d'un point de vue coût, les techniques SAD et SSD sont les plus simples et donc les moins coûteuses en temps de calcul.

Cette simplicité est la contrepartie aux inconvénients de ces techniques (illumination constante, pas de facteur de gain entre les images)

La figure 9 ci dessous présente un couple d'image stéréoscopique, avec une luminosité différente entre les deux images ainsi que les résultats obtenus grâce à un algorithme utilisant diverses mesures de similitude :

- (a) SAD : la différence d'illumination entre les deux images entraine une mauvaise carte de disparité
- (b) ZSAD : On corrige les erreurs dues à la distorsion radiométrique
- (c) SSD : idem SAD
- (d) ZSSD : idem ZSAD

- (e) NCC : Correction additionnelle des différences de gain
- (f) ZNCC : idem ZSSD + correction du gain

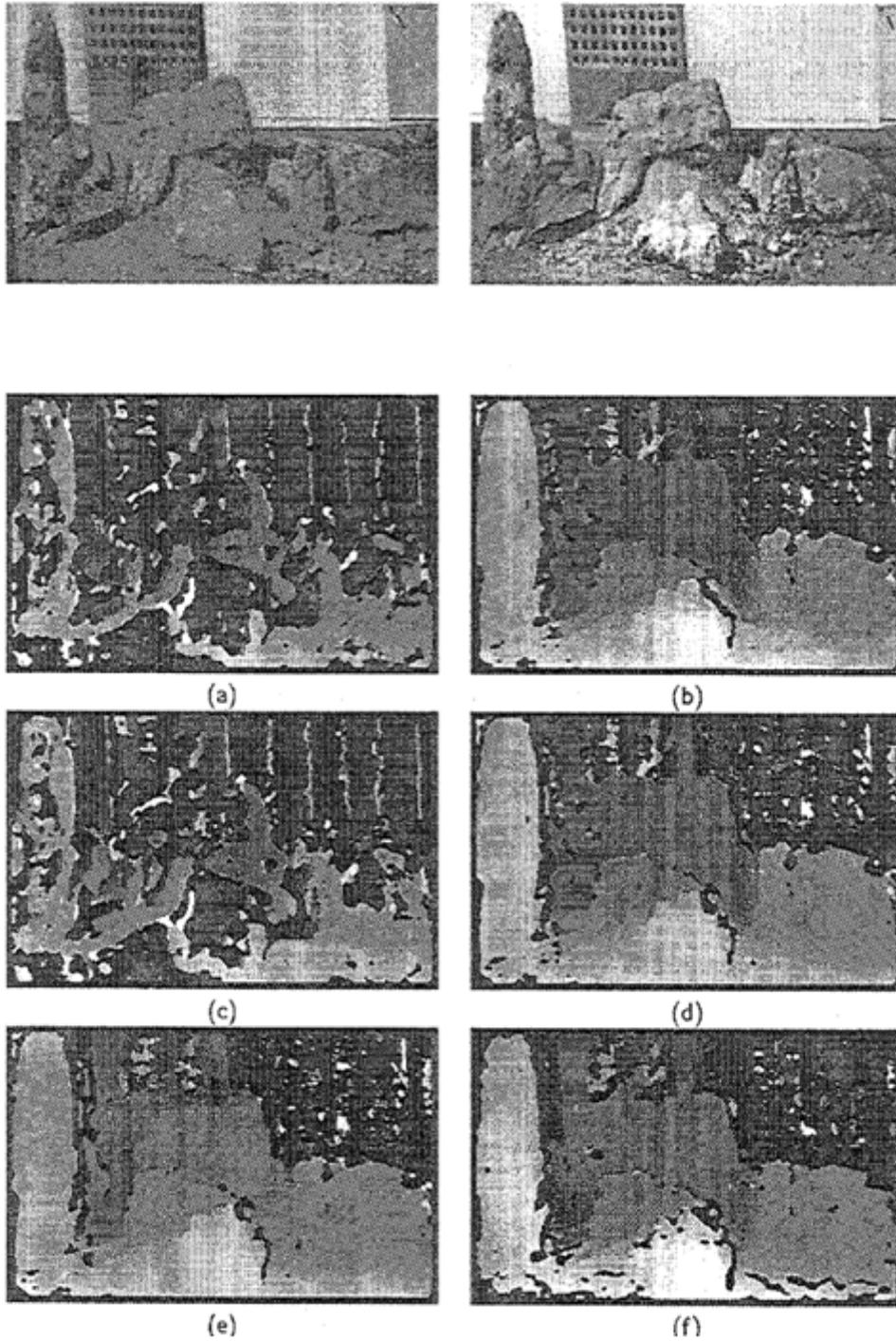


Figure 9 – Cartes de Disparités obtenues avec différentes mesures de similitude

Source des figures 9,10 : J.Banks, M.Bennamoun and P.Corke. *Non Parametric Techniques for fast and robust stereo matching*. IAIF'97. Nov 97 [15]

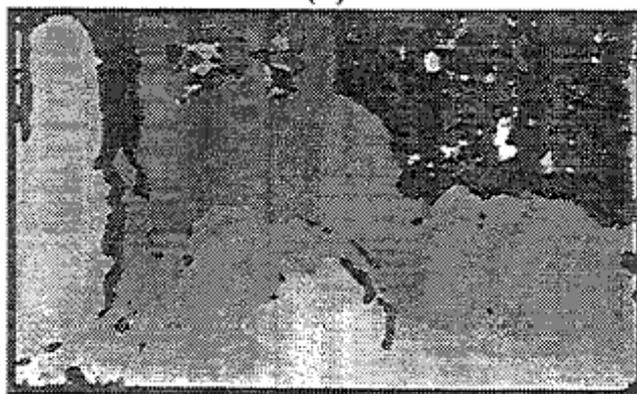
### ***II.5.1.2 Mesures non paramétriques***

Les mesures non paramétriques sont réalisées non plus sur les images réelles mais sur leur versions transformées par une transformée non paramétrique. Les transformations non paramétriques sont des transformations basées sur l'ordre relatif de l'intensité des pixels dans une fenêtre et non pas sur les intensités directes. On distingue deux transformées non paramétriques particulièrement intéressantes : la transformée du rang et la transformée de Census.

#### ***II.5.1.2.1 Transformée du rang***

La transformée du rang se définit de la façon suivante : la valeur de cette transformation correspond au nombre de pixels dont la valeur d'intensité est inférieure à la valeur du pixel d'intérêt. Le pixel d'intérêt est généralement le pixel central d'une fenêtre correspondant à l'ouverture de la transformation. Une fois la transformée appliquée on apparie les deux images grâce à une mesure de similarité. On choisira en général le SSD ou le SAD pour leur coût de calcul raisonnable [9].

Un des avantages de la transformée du rang, hormis la résistance aux distorsions radiométriques est de nécessiter un faible nombre de bit pour coder une fenêtre complète. En effet si on se place dans une fenêtre 7x7 on peut représenter cette fenêtre grâce à 7 bits alors dans l'image traditionnelle il est nécessaire d'avoir 8 bits par pixel.



**Figure 10 - Disparity map after Rank Transform + SAD**

II.5.1.2.2 Transformée de Census

La transformée de Census [1] est également une transformée non paramétrique. Elle est définie de la façon suivante : pour un pixel donné on va étudier le voisinage de ce pixel (une fenêtre de taille choisie). On réalise une comparaison entre les valeurs d'intensité du pixel et les valeurs d'intensité de ses voisins. On définit un nouveau codage sur un vecteur de la façon suivante : si la valeur du voisin est supérieure à la valeur du pixel d'intérêt on code par un 1, sinon on code 0.

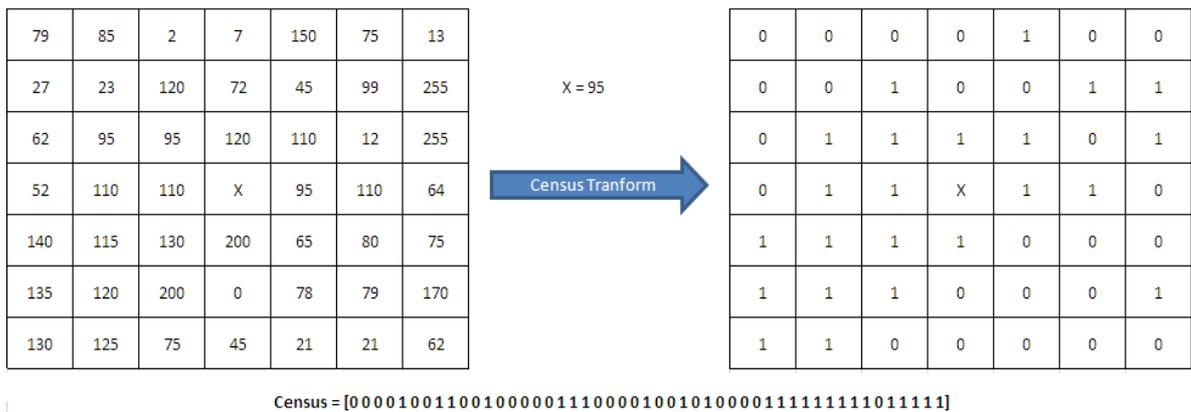


Figure 11 - Principe de la transformée de Census

On code donc un pixel et son voisinage (fenêtre de taille n) sur un vecteur de n\*n-1 bits. Ce codage est résistant aux variations d'intensité et de gain puisqu'il repose sur la notion d'intensité relative.

Une fois les images codées par la transformée de Census l'appariement est réalisé grâce à une mesure de similitude particulière : la distance de Hamming. La distance de Hamming est une distance au sens mathématique du terme qui permet de réaliser la comparaison entre deux chaînes (caractères, bits...). Elle traduit en fait, pour deux chaînes de bits, le nombre de bits dont la valeur diffère. La figure 12 explicite le principe de la distance de Hamming.

Deux points similaires au sens de la transformée de Census auront une distance de Hamming faible.

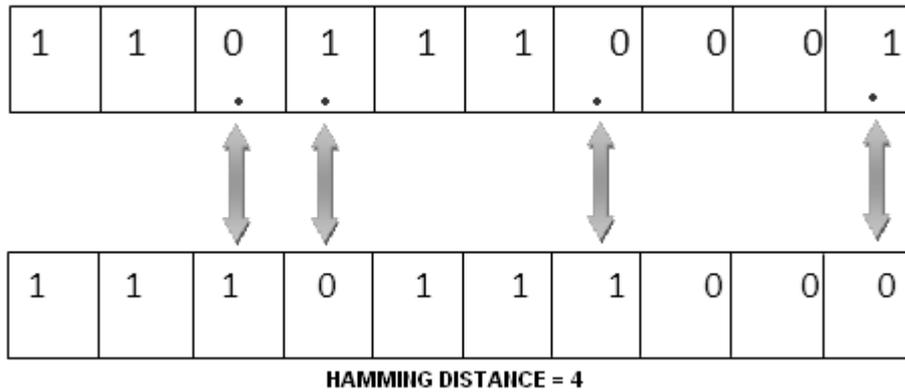


Figure 12 - Principe de la distance de Hamming

On peut par ailleurs modifier la transformée de Census de façon à la rendre plus robuste aux discontinuités. Ceci a été réalisé en remplaçant la valeur du pixel central de la fenêtre de transformation par la valeur moyenne des pixels dans la fenêtre.



Figure 13 – Carte de disparité après Census+Hamming

La transformée de Census dispose d'un certain nombre d'avantages. Tout d'abord, elle est résistante aux problèmes de différences d'illumination et de différence de gain, du fait de l'utilisation d'un codage en intensité relative. Par ailleurs, la simplicité de la mesure de similitude en fait un excellent candidat à une implémentation hardware [1]. En effet il s'agit d'une simple comparaison bit à bit, ce qui peut être effectué rapidement.

### II.5.2 Les contraintes pour l'appariement

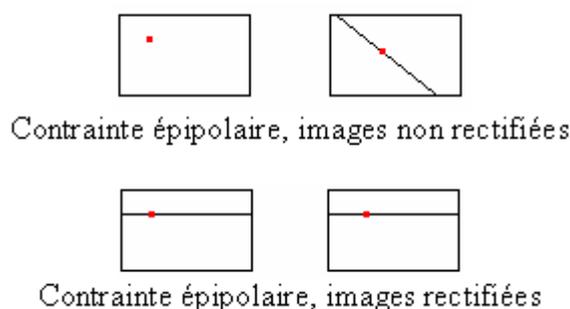
Afin de réaliser un appariement global correct on doit appliquer un certain nombre de contraintes à l'algorithme de façon à pouvoir sélectionner le vrai correspondant d'un point et non pas un autre point lui étant semblable. On va exposer ici les contraintes les plus

courantes. Certaines de ces contraintes vont par ailleurs permettre de diminuer le coût de l'algorithme en diminuant l'espace de recherche de correspondants.

### ***II.5.2.1 Contrainte Épipoilaire***

La contrainte épipoilaire est une contrainte forte due à la géométrie des caméras. Cette géométrie est considérée comme connue après la calibration. L'expression brute de la contrainte est la suivante : tout point de l'image de gauche trouve son correspondant sur une ligne de l'image de droite appelée ligne épipoilaire et réciproquement.

Après la phase de rectification épipoilaire les lignes épipoilaires sont parallèles : un point trouve son correspondant dans l'autre image sur la ligne de même ordonnée.



**Figure 14 - Contrainte Épipoilaire**

La contrainte épipoilaire est la seule contrainte forte dont l'on dispose. Elle dépend de la géométrie du système et permet de réduire le problème de l'appariement à la recherche sur une ligne. La rectification permet de simplifier grandement cette recherche d'un point de vue algorithmique.

### ***II.5.2.2 Contrainte d'Unicité***

La contrainte d'unicité [3] est une contrainte qui est quasiment toujours véridique. Elle peut s'énoncer simplement de la façon suivante : un pixel d'une image ne peut correspondre qu'à un seul et unique pixel de son image correspondante. (Noter qu'il peut également ne correspondre à aucun pixel de l'image correspondante). Il existe un certain nombre d'algorithmes qui reposent en grande partie sur cette contrainte [1].

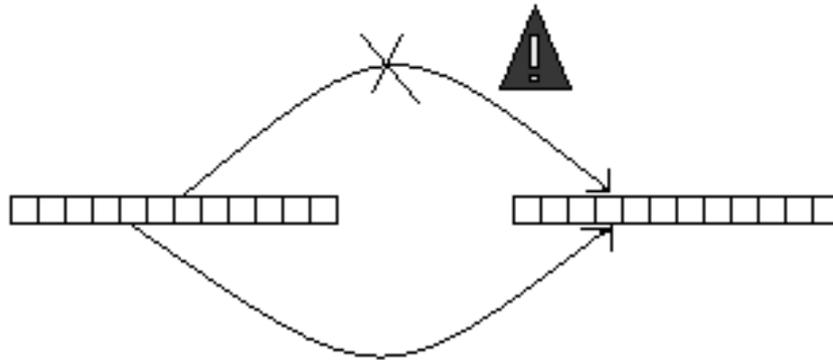


Figure 15 - Contrainte d'unicité

Si la contrainte d'unicité n'est pas implémentée de façon correcte le problème se traduit principalement sur les surfaces homogènes : en effet, les points seront tous appariés avec le même point (le premier rencontré par l'algorithme).

Il est donc nécessaire d'utiliser un algorithme itératif permettant de sélectionner les meilleurs candidats à l'appariement pour chaque point.

### II.5.2.3 Contrainte d'Ordre

La contrainte d'ordre [4] est une contrainte simple mais qui ne sera valable que dans le cas où la scène ne contient pas d'objets transparents fortement inclinés (l'ordre des projections est inversé dans ce cas).

Le principe de la contrainte d'ordre est le suivant : si un point  $m(x,y)$  est situé à la gauche d'un point  $n(u,y)$  alors son correspondant  $m'(x',y')$  sera également situé à la gauche du point  $n'(u',y')$  correspondant de  $n$ .

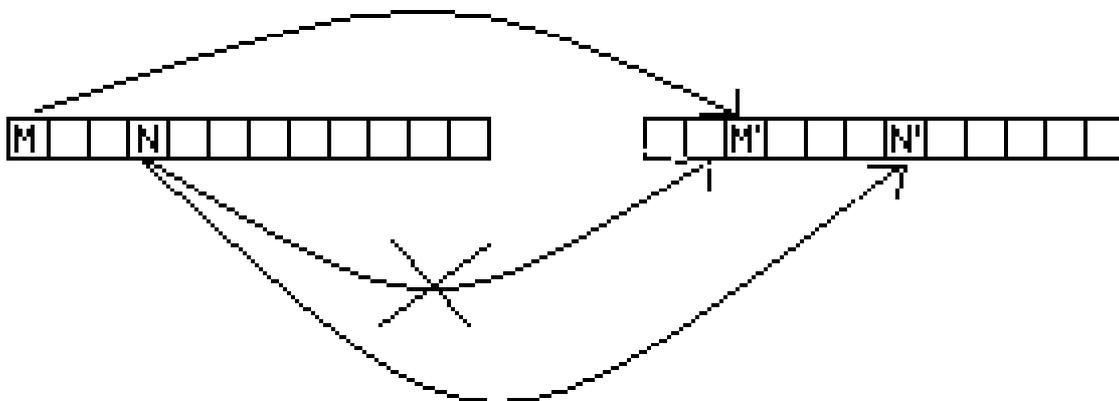


Figure 16 - Contrainte d'ordre

#### ***II.5.2.4 Contrainte de Continuité***

Cette contrainte introduit le fait que la répartition des disparités entre les deux images varie selon un gradient. [4] Cette variation est donc continue. Cette contrainte est relativement délicate à mettre en œuvre car elle doit être satisfaite sur des parties spécifiques de l'image qui correspondent bien à des zones exemptes de toute discontinuité. Elle se traduit par le choix d'une fenêtre de corrélation suffisamment petite pour ne pas contenir de discontinuités.

Il existe un certain nombre d'algorithmes de corrélation se basant sur cette contrainte. On peut citer l'algorithme PMF (Pollard-Mayhew-Frisby) [4] chapitre 6 pages 219-222 qui permet d'obtenir des résultats intéressants.

#### ***II.5.2.5 Contrainte de consistance Gauche/Droite***

Cette contrainte énonce le fait qu'une paire de points  $(m, m')$  obtenue grâce à une recherche de correspondants gauche/droite doit être vérifiée par une recherche droite/gauche (c'est-à-dire trouver le correspondant du point  $m'$  dans l'image de gauche. Si ce point est  $m$  la consistance gauche/droite est vérifiée et la mise en correspondance est validée. Si ce n'est pas le cas la paire est invalidée.

Cette vérification permet de détecter efficacement les occlusions entre les images gauches et droites. Ceci à une contre partie en temps de calcul puisqu'il sera nécessaire de calculer deux cartes de disparité au lieu d'une seule : le coup en temps de calcul est donc globalement multiplié par deux puisque la phase de recherche de correspondance est la plus longue phase du processus de vision stéréoscopique.

La conception d'un algorithme d'appariement se fait donc par le choix d'une part de la mesure de similitude utilisée et d'autre part par des diverses contraintes à utiliser dans l'algorithme. Selon les choix réalisés les propriétés de l'algorithme seront différentes en termes de rapidité, du type d'images qui pourront être utilisées ou de la précision finale obtenue. Nos choix seront exposés au paragraphe IV traitant de l'algorithme développé dans le cadre de ce stage.

### **III Programme de Vision Stéréoscopique développé**

Les objectifs de ce stage se divisent en deux points principaux. Dans un premier temps réaliser le développement d'un programme de vision stéréoscopique nous permettant de disposer d'un modèle fonctionnel de système 3D et dans un second temps l'étude et l'implémentation de différents algorithmes et mesures de similitude. Le second point sera traité dans la partie suivante. On présentera ici les différentes possibilités offertes par notre programme ainsi qu'un certain nombre d'exemple. Cette partie à été réalisée en collaboration avec M. Jérôme ROUCHETON, stagiaire au L2MP sous la direction du Pr. Christian DUFAZA.

#### **III.1 Cahier des charges**

Le but recherché lors du début de la conception de ce programme de vision par ordinateur était l'obtention d'une solution complète, couvrant l'ensemble des étapes de la vision 3D : de l'acquisition vidéo au calcul de positions 3D en passant par la calibration des capteurs.

Ce modèle pourra par la suite être utilisé afin de caractériser un algorithme ou une configuration géométrique des capteurs.

Afin de réaliser ce modèle on dispose de deux caméras STV0676 et d'une station de travail DELL. On utilise le langage C++ ainsi que la librairie de traitement de l'image OpenCv (voir Annexe 2).

On dispose également comme primitive du travail réalisé au cours de notre projet de fin d'études ainsi que d'une classe de gestion d'une caméra réalisée par Mr. Simon CONSEIL PhD au sein de l'équipe d'AST ROUSSET.

#### **III.2 Fonctionnalités proposées**

Les fonctionnalités implémentées dans notre programme de vision stéréoscopiques sont multiples. Les fonctions ajoutées suivent le schéma synoptique présenté figure 1 et permettent de réaliser l'ensemble des fonctions nécessaires à la vision par ordinateur :

- Ouverture de Flux
- Calibration des caméras
- Correction de la distorsion
- Estimation de la géométrie épipolaire
- Rectification Épipoilaire



- Appariement
- Estimation globale des distances d'une scène

### **III.2.1 Ouverture et synchronisation des flux vidéo**

La première étape de notre processus de vision consiste à obtenir les flux vidéo depuis nos caméras. Ces flux doivent être le mieux synchronisés possible de manière à éviter les écarts temporels entre les images gauche et droite. En effet, si les deux images sont légèrement déphasées l'une par rapport à l'autre on cherchera alors à apparier deux images représentant la même scène sous des angles différents et à des temps différents. Par ailleurs, un dispositif de sauvegarde de vidéos a également été mis au point de façon à pouvoir conserver un état donné du système stéréoscopique. L'étude d'un certain nombre de vidéos test permet d'observer les évolutions du programme de vision mais également de détecter aisément les problèmes qui peuvent apparaître au cours de la conception.

### **III.2.2 Calibration des caméras**

La seconde étape du processus de vision est l'étape de calibration des caméras. Cette étape va nous permettre de récupérer les différents paramètres caractérisant la caméra (voir section II.3). Ces paramètres seront utiles par la suite pour les étapes de rectification et de récupération des coordonnées 3D.

La calibration des deux capteurs est réalisée de façon simultanée de manière à avoir des jeux de paramètres extrinsèques décrivant la même configuration de l'espace 3D.

Dans notre cas la calibration est réalisée en deux étapes : dans un premier temps un opérateur présente un pattern de calibration (figure 3) au système stéréoscopique, on utilise un détecteur de coins de façon à détecter la position des intersections entre les carrés dans l'image. Les positions des intersections sont relevées dans les deux images : cette étape permet d'établir un certain nombre de correspondances, correspondances qui seront utilisées par la fonction de calibration afin de déterminer les différents paramètres caractéristiques des caméras. Cette fonction de calibration est basée sur l'algorithme de Zhang [14].

Une fois la calibration achevée on dispose sous forme de matrices des paramètres intrinsèques et extrinsèques des deux caméras mais également d'un certain nombre de paramètres de calibration ainsi que d'une évaluation de l'erreur de reprojection. Si cette erreur est trop élevée la calibration ne sera pas suffisamment précise et devra être

réalisée à nouveau (voir Annexe 3 pour une étude sur l'influence d'un certain nombre de paramètres sur la qualité de la calibration).

### III.2.3 Correction de la distorsion

Une fois la calibration achevée on dispose des paramètres de distorsion qui vont nous permettre de corriger la distorsion des images. Cette distorsion est due à l'optique des caméras et ne peut donc être corrigée que par un traitement de l'image à posteriori.



Figure 17 - Correction de la distorsion optique

Cette phase est indispensable de façon à travailler sur des images le plus proche du réel possible. De plus la distorsion engendrée par l'optique des caméras peut être différente entre la caméra gauche et droite, il devient nécessaire dans ce cas de corriger la distorsion pour réaliser un appariement correct.

### III.2.4 Estimation de la géométrie épipolaire

Notre programme permet également de réaliser une évaluation de la géométrie épipolaire d'une scène. Connaître la géométrie épipolaire, revient à connaître les équations de toutes les droites épipolaires. Ces droites sont définies par la matrice fondamentale qui permet d'associer à chaque point de l'image gauche une droite dans l'image de droite et réciproquement. Cette estimation de la matrice fondamentale et donc de la géométrie peut être réalisée grâce à la connaissance d'un certain nombre de paires de points (au minimum 8). La matrice fondamentale peut être calculée grâce à l'algorithme de RANSAC (voir annexe 4). Une fois la matrice fondamentale estimée on

peut déterminer les équations des lignes épipolaires. La figure 18 ci-dessous présente deux faisceaux de lignes épipolaires. Pour la première image l'épipôle est situé à l'intérieur de l'image alors qu'il est situé à l'extérieur pour la seconde.



Figure 18 - Géométrie épipolaire estimée

### III.2.5 Rectification Epipolaire

Cette fonctionnalité de notre programme permet de réaliser la rectification des images comme exposé au paragraphe II.4. Un exemple a par ailleurs été présenté figure 7. On rappelle que le but de la rectification est d'appliquer une transformation à la paire stéréoscopique de façon à ce que les points correspondants soient situés sur la même ligne. Cette étape permet de simplifier grandement l'étape d'appariement.

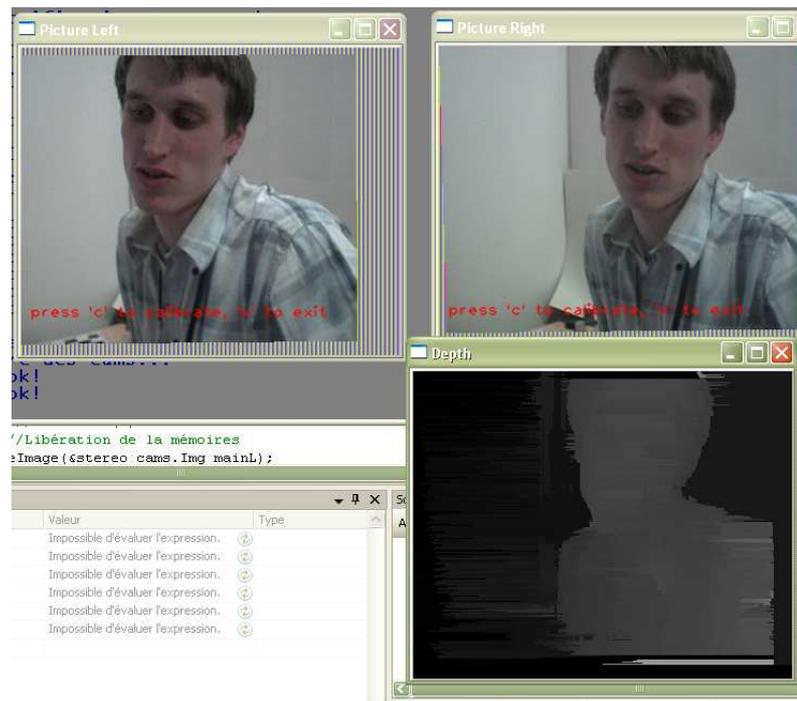
On a implémenté dans notre programme 2 algorithmes de rectification aux propriétés différentes, l'algorithme proposé par A. Fusiello [10] et l'algorithme implémenté sous Matlab par J.Y. Bouguet [13]. Les deux rectifications sont fonctionnelles et permettent de rectifier en temps réel une paire d'images dans un flux vidéo.

### III.2.6 Appariement

Une fois l'étape de rectification complétée avec succès on peut réaliser l'appariement de nos images. L'appariement des images consiste en une recherche du point correspondant à un point d'une image de la paire stéréoscopique sur la ligne

correspondante dans l'autre image. Une fois le correspondant établi, on dispose de l'information de disparité qui représente la différence entre les coordonnées du point et de son correspondant sur la ligne. Le résultat global de l'appariement est donc une carte de disparité qui donne la disparité pour chaque point de l'image que l'on a apparié.

Dans le cadre de notre programme de vision3D temps réel on utilise l'algorithme de Birchfield [8] implémenté directement dans OpenCv. Cet algorithme nous permet d'obtenir une carte de disparité globalement correcte et donc d'avoir une bonne estimation de la distance des points d'une scène.



**Figure 19 – Carte de Disparité en temps réel**

On note que la carte de disparité est construite sur une échelle n/b 8bits la plus grande disparité étant le point le plus proche et donc le plus clair sur la carte. A l'inverse le point de plus faible disparité est le point le plus éloigné et donc le plus noir sur la carte de disparité.

### III.2.7 Estimation de la distance des points d'une scène

Notre programme permet, à partir d'une carte de disparité de calculer et de récupérer l'ensemble des coordonnées 3D des points de la scène. Ce calcul est une primitive à la reconstruction 3D d'une scène et est réalisé grâce à la triangulation.

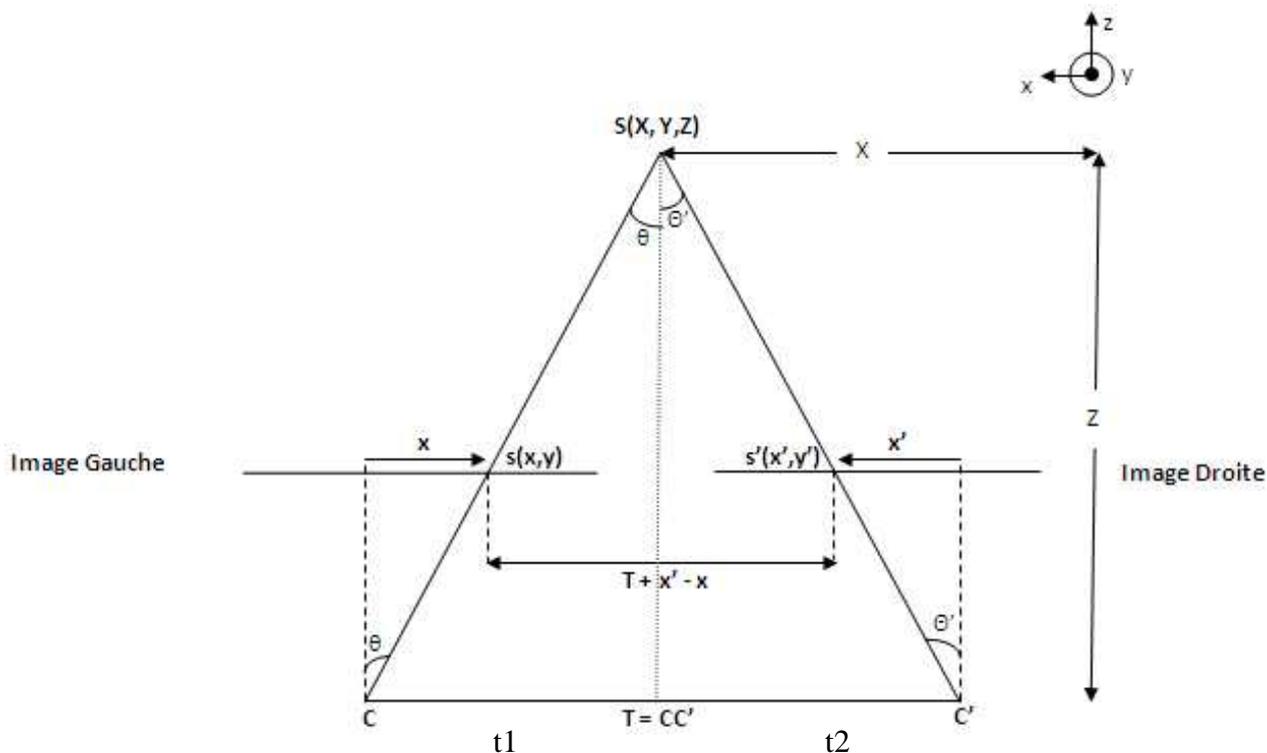


Figure 20 - Système stéréoscopique et Triangulation

On note d la disparité. A partir des distances connues sur le schéma de la figure 20 on veut déterminer les coordonnées 3D du point S (X,Y,Z). Pour cela on va tout d'abord exprimer les valeurs de  $\theta'$  et de  $\theta$ .

$$\tan(\theta) = \frac{-x}{f} , \quad \tan(\theta') = \frac{x'}{f}. \quad (1)$$

De la même façon on a :

$$\tan\left(\frac{\pi}{2} - \theta\right) = \frac{Z}{t1} , \quad \tan\left(\frac{\pi}{2} - \theta'\right) = \frac{Z}{t2}.$$

On rappelle que :

$$\tan\left(\frac{\pi}{2} - \theta\right) = \frac{1}{\tan(\theta)} .$$

D'où :

$$\frac{1}{\tan(\theta)} = \frac{Z}{t1} , \quad \frac{1}{\tan(\theta')} = \frac{Z}{t2}.$$

Et :

$$\tan(\theta) = \frac{t1}{Z}, \quad \tan(\theta') = \frac{t2}{Z}.$$

De la on tire :

$$\tan(\theta) + \tan(\theta') = \frac{t1 + t2}{Z} = \frac{T}{Z}.$$

De (1) viens :

$$\tan(\theta) + \tan(\theta') = \frac{x' - x}{f} = \frac{d}{f}.$$

On a donc :

$$\frac{T}{Z} = \frac{d}{f}.$$

Et :

$$Z = \frac{f * T}{d}. \quad (2)$$

On a donc déterminé l'expression de la coordonnée Z de S. Il reste à déterminer X et Y. On utilise un raisonnement similaire à celui réalisé ci-dessus et on démontre que :

$$X = \frac{Z * x}{f} \quad \text{et} \quad Y = \frac{Z * y}{f}. \quad (3)$$

Dans notre programme on calcule les coordonnées (X,Y,Z) de chaque point d'une scène grâce à la carte de disparité et on stocke les résultats dans un fichier texte : on pourra ultérieurement réaliser une reconstruction dense de la scène tridimensionnelle grâce à ces coordonnées.

## IV Notre Algorithme de Corrélation

Une fois la conception du modèle de vision stéréoscopique achevé nous nous sommes orientés vers la conception d'un algorithme de corrélation nous permettant de tester diverses mesures de similitudes ainsi que diverses contraintes pour la stéréoscopie.

### IV.1 Principe de l'algorithme

Notre algorithme se base principalement sur l'utilisation de deux contraintes : la contrainte épipolaire grâce à l'utilisation d'images rectifiées [10] et à la recherche de correspondants sur la ligne correspondante de l'image et la contrainte d'unicité [3] qui dit qu'un point ne peut être apparié qu'une seule et unique fois.

Le principe de l'algorithme est le suivant, on réalise un appariement ligne par ligne. On se positionne en un point de la ligne de l'image de gauche et on va parcourir la ligne de l'image de droite à partir du point de même coordonnées sur la ligne jusqu'au point de coordonnées distante de la valeur maximale de la disparité. (Figure 21)

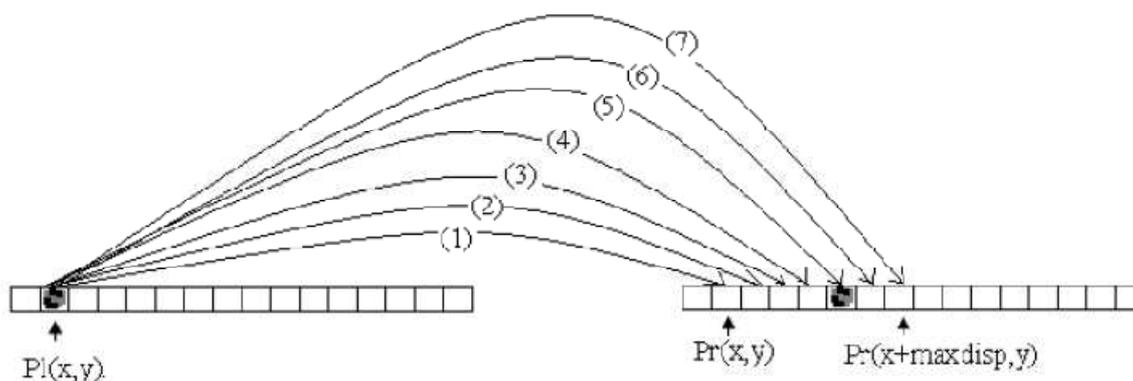


Figure 21 – Recherche de Correspondant

Pour chaque point rencontré on calcule la valeur de similitude et on sélectionne le point ayant la valeur de similitude la plus proche. On va ensuite utiliser la contrainte d'unicité : avant de définir une paire de point on vérifie si le point choisi comme meilleur correspondant n'est pas déjà apparié. Si tel est le cas on compare les valeurs de similitude entre les deux appariements possible de manière à choisir le meilleur des deux. L'appariement le moins ressemblant est rejeté. Il sera retraité grâce à un processus itératif : on réalise en effet la phase de traitement des points rejetés par l'unicité jusqu'à ce qu'ils aient tous été appariés où déclarés sans correspondants. Une fois cette étape terminée on dispose d'une carte de disparité. Cette carte de disparité pourra subir un traitement postérieur de façon à propager plus ou moins les zones définies comme fiables. En effet,

selon la valeur de similitude et la valeur de similitude des pixels de la zone on définit une carte de fiabilité. Cette carte de fiabilité sera utilisée afin de remplir les zones en occlusion (n'ayant pas pu être appariées) et donc obtenir une carte de disparité plus dense. Sur la figure 22 apparaissent en violet les points très fiables, en jaune les points de fiabilité moyenne et en vert les points en occlusion.



Figure 22 - Carte de Fiabilité

## IV.2 Mesures de Similitudes implémentées

Le but de la conception de notre algorithme de corrélation était entre autre de pouvoir comparer diverses mesures de similitudes. Notre algorithme en implémente actuellement 3 de manière fonctionnelle. La transformée de Census couplée à la distance de Hamming, le SAD (Sum of Absolute Differences) ainsi que le SSD (Sum of Squares Differences). Une mesure de similitude basée sur une corrélation NCC (Normalized Cross Correlation) est en cours de mise au point. La transformée du Rang ([1],[9]) est également implémentée mais inutilisable pour le moment car on ne dispose pas d'une mesure de similitude capable de tirer partie de cette transformation.

On dispose par ailleurs des versions du SAD, du SSD et de la transformée de Census qui fonctionnent sur une fenêtre de corrélation normalisée par rapport au pixel central de la fenêtre. Cette normalisation permet de s'affranchir des distorsions radiométriques.

Il est intéressant d'implémenter et de tester ces différentes mesures de similitude de façon à identifier leurs caractéristiques et leurs propriétés. Il faut garder à l'esprit que le but est d'étudier la possibilité d'implémentation hardware d'un algorithme de corrélation incluant une mesure de similitude. On présentera les résultats obtenus à la section suivante ainsi qu'une analyse comparée des mesures de similitude.

Il est à noter que la performance d'un algorithme dépend de façon égale des contraintes utilisées pour l'algorithme et de la mesure de similitude choisie.

### IV.3 Exemples & résultats

Afin d'évaluer notre algorithme nous avons utilisé un certain nombre de jeux d'images test appartenant à des bases de données classiques pour l'évaluation des algorithmes de stéréoscopie. Ces jeux d'images ont l'avantage de fournir une carte de disparité représentant la disparité réelle entre les pixels de deux images. Dans un souci de clarté on ne présentera ici qu'un nombre réduit d'images : Tsukuba, Castle, Pm, Plan Incliné et Lampe. On présentera également les résultats obtenus sur une paire d'images stéréoscopique réelle. La comparaison entre les différentes mesures de similitude sera réalisée grâce au jeu d'images Tsukuba.

L'ensemble des images utilisées sont extraites du Millsbury's Image Set [12] ou fournies par l'université de Tsukuba, Jpn.

#### IV.3.1 Jeu d'images stéréoscopiques : Castle

Ce jeu d'image est le premier que nous avons utilisé. Il ne fait pas parti d'un set d'évaluation mais possède des propriétés intéressantes : forte texturation, nombre connu de zones de disparité différentes, disparité de ces zones connues et présence d'occlusions bien localisées.

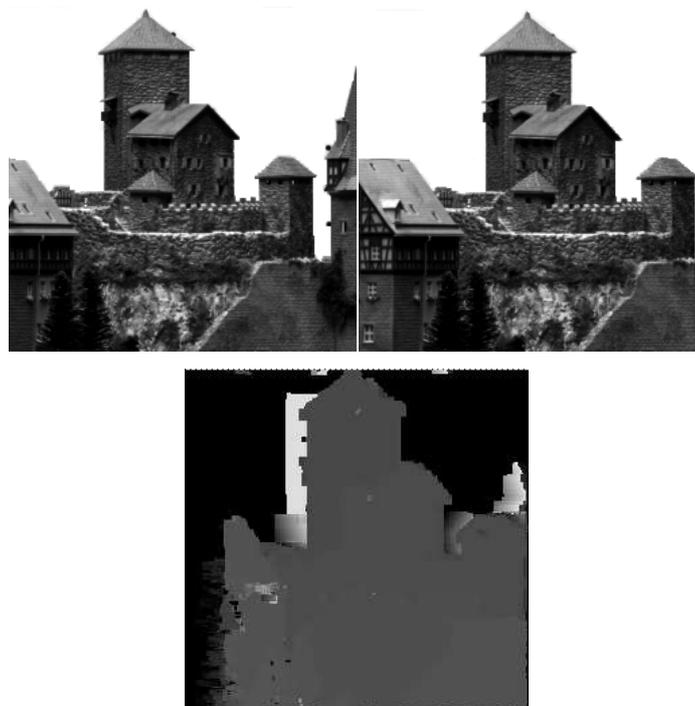


Figure 23 – Castle, images gauche, droite, Carte de disparité

La carte de disparité présentée est obtenue grâce à la transformée de Census. Sur ce jeu d'images stéréoscopique on dispose des informations suivantes :

- Disparité maximale = 28
- Arbres en bas à gauche, disparité= 28
- Mur en ruine en bas à droite, disparité = 27
- Maison Gauche : disparité = 26
- Mur Milieu : disparité = 25
- Maison Milieu : disparité = 24
- Maison arrière : disparité = 24

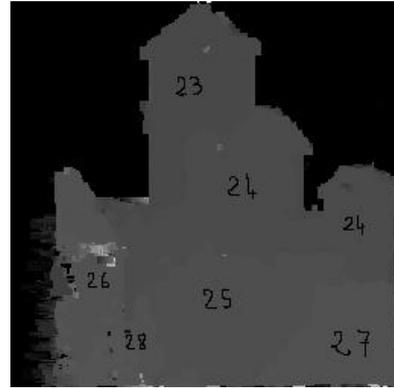


Figure 24 - Disparités Réelles

La carte de disparité obtenue est globalement correcte, on obtient environ 80% de points dont la disparité est concordante avec la disparité réelle. Les 20% restant sont les points situés aux discontinuités de disparité ou dans des parties en occlusion. On peut noter que la performance temporelle est moyenne puisque la carte de disparité est obtenue grâce à notre algorithme couplé à la transformée de Census en 2.5s. Ce temps pourra être par la suite assez aisément réduit, en effet l'étape couteuse est l'étape de calcul de la distance de Hamming entre deux points et cette étape est réalisée neuf fois pour chaque paire de points. Le temps pourra donc être réduit de façon conséquente en adaptant la technique de programmation de façon à conserver les résultats qui pourront être utiles par la suite.

#### IV.3.2 Jeu d'images stéréoscopiques : Plan Incliné

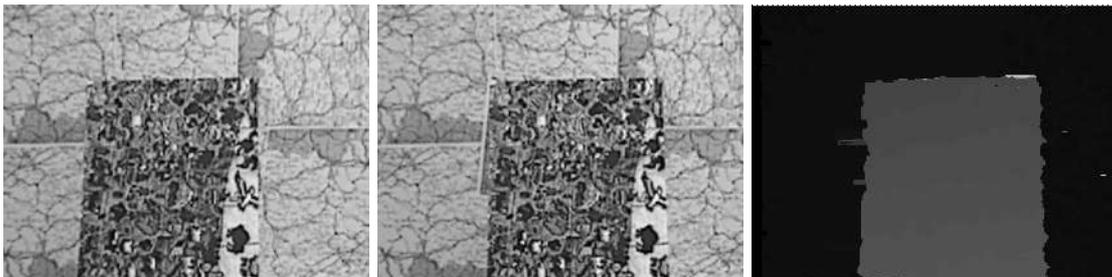


Figure 25 – Set d'image Plan Incliné Gauche, Droite et Carte de disparité associée



Figure 26 - Carte de disparité réelle

Ce jeu d'images permet d'étudier deux points principaux : tout d'abord il s'agit d'une paire stéréoscopique ou il n'existe que deux surfaces différentes. Par ailleurs l'image représente un plan incliné, on va donc pouvoir constater si le haut du plan est bien plus éloigné que le bas selon la carte de disparité. Enfin l'image est hautement texturée ce qui facilite l'appariement. La carte de disparité présentée à la figure 25 est obtenue grâce à notre algorithme et la transformée de Census couplée à la distance de Hamming. Le résultat obtenu est intéressant : si l'on compare la carte de disparité obtenue à celle correspondant à la disparité réelle (figure 26) après mise à l'échelle (échelonnage de la disparité de 0 à 255) on obtient un pourcentage de points appariés aux alentours de 90% : une certaine incertitude subsiste aux contours du plan mais l'inclinaison est rendue de façon fidèle par la carte de disparité.

#### IV.3.3 Jeu d'images stéréoscopiques : Parcmètres

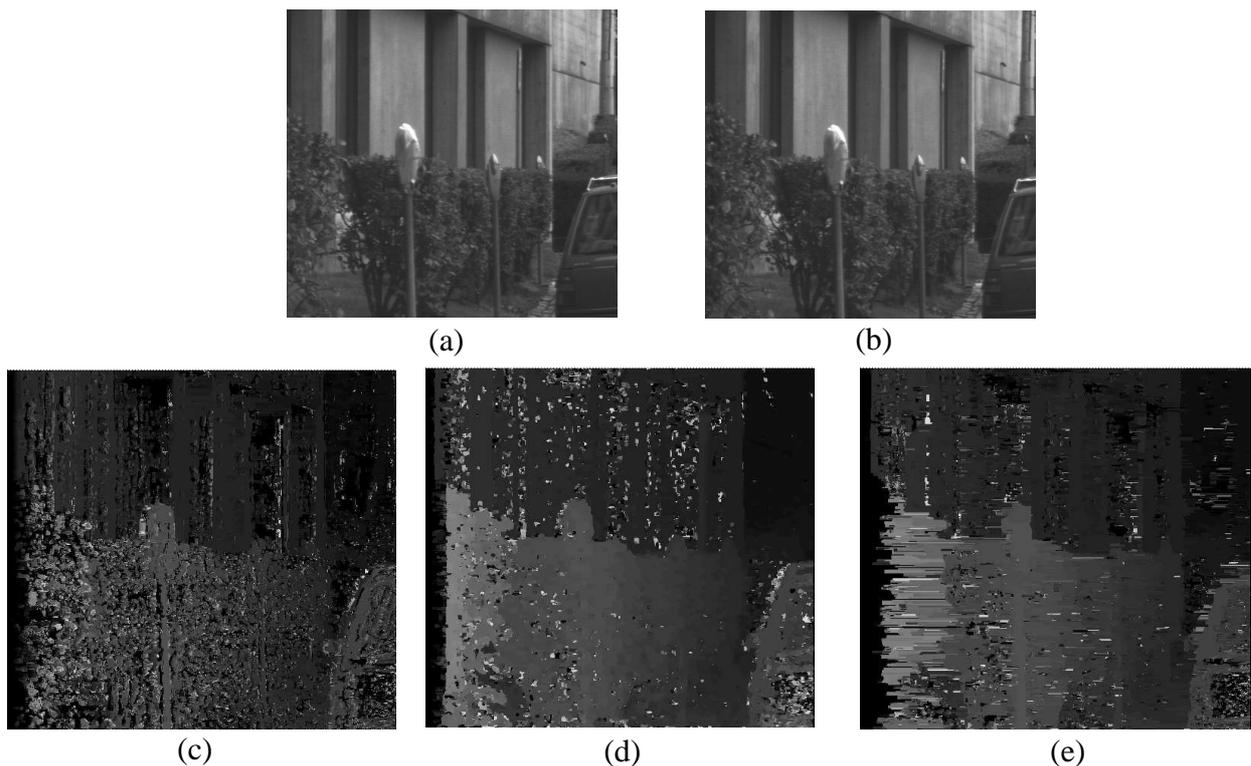


Figure 27 – Parcmètres : (a) Gauche (b) Droite (c) SSD Correlation (d) Census Correlation (e) SAD Correlation

Ce jeu d'images stéréoscopique nous permet d'étudier l'efficacité de notre algorithme sur une image en perspective, moyennement texturée. Par ailleurs pour ce jeu on réalise la comparaison entre les différentes mesures de similitude.

On réalise la comparaison entre les cartes de disparité obtenues et la carte de disparité réelle. On obtient pour le SAD un pourcentage de 55% de points correctement appariés, pour le SSD 70% et pour Census 77%. On constate donc que la corrélation basée sur la transformée de Census permet d'obtenir une carte de disparité plus précise. Cette précision se traduit notamment par un pourcentage d'erreurs plus faible au voisinage des discontinuités. Le SAD permet également d'obtenir une bonne estimation de la disparité, de façon plus rapide (une corrélation SAD prend environ dix fois moins longtemps qu'une corrélation avec Census). Enfin le SSD permet d'obtenir une carte de disparité de qualité moyenne, avec un grand pourcentage d'erreurs sur les surfaces uniformes ou de disparité constante.

#### IV.3.4 Jeu d'images stéréoscopiques : Tsukuba

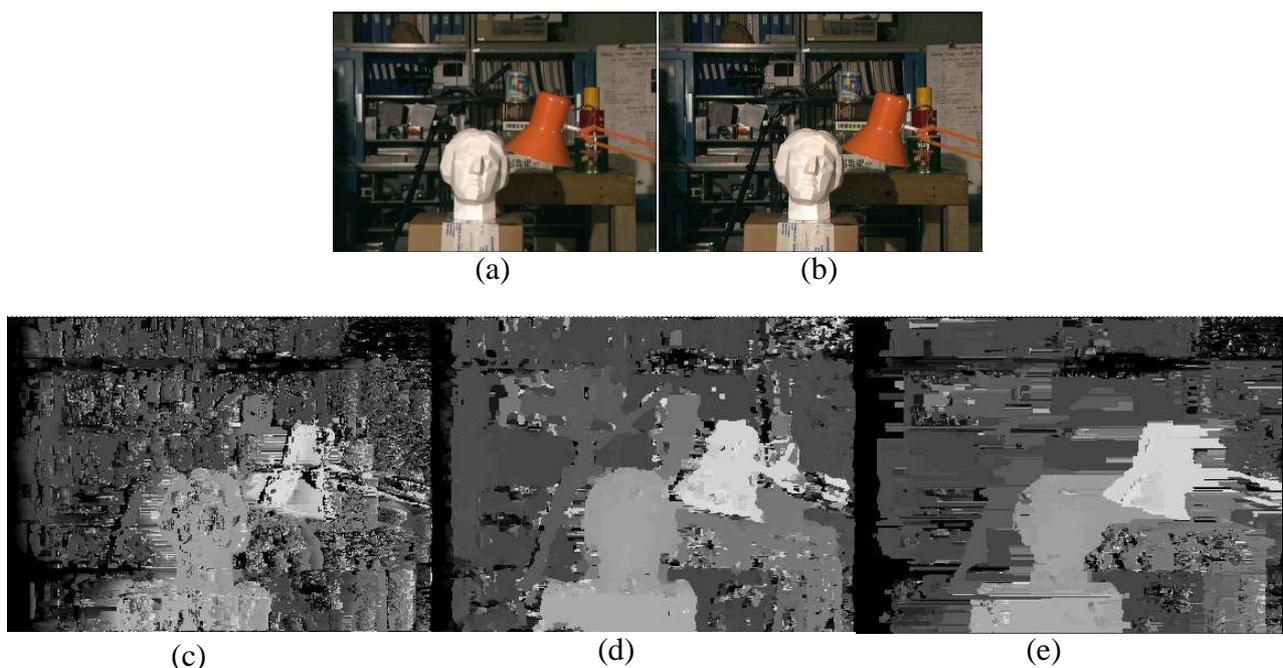


Figure 28 – Tsukuba : (a) Gauche (b) Droite (c) SAD Correlation (d) Census Correlation (e) SSD Correlation

Ce jeu d'images stéréoscopique représente une scène avec un arrière plan complexe, de nombreux objets à distinguer etc. Il est intéressant parce qu'il permet de déterminer si un algorithme permet bien de distinguer ou non un objet en particulier dans une scène. Par ailleurs, il permet également d'évaluer la possibilité pour un algorithme de détecter le détail. En effet si on observe le buste au premier plan on constate que la transformée de Census couplée à la distance de Hamming et la Corrélation SSD permettent de détecter les différences de niveau sur ce visage. Enfin on constate les mêmes différences entre les

algorithmes que pour les jeux précédents avec des chiffres moyens plus faibles : 45% pour le SAD, 61% pour le SSD et 67% pour la transformée de Census. Ces chiffres plus faibles s'expliquent par la complexité de la scène qui augmente le nombre de discontinuités dans la carte de disparité et donc les possibilités d'erreur.

#### IV.3.5 Jeu d'images stéréoscopiques : Jeu stéréo capturé par nos soins – L2MP

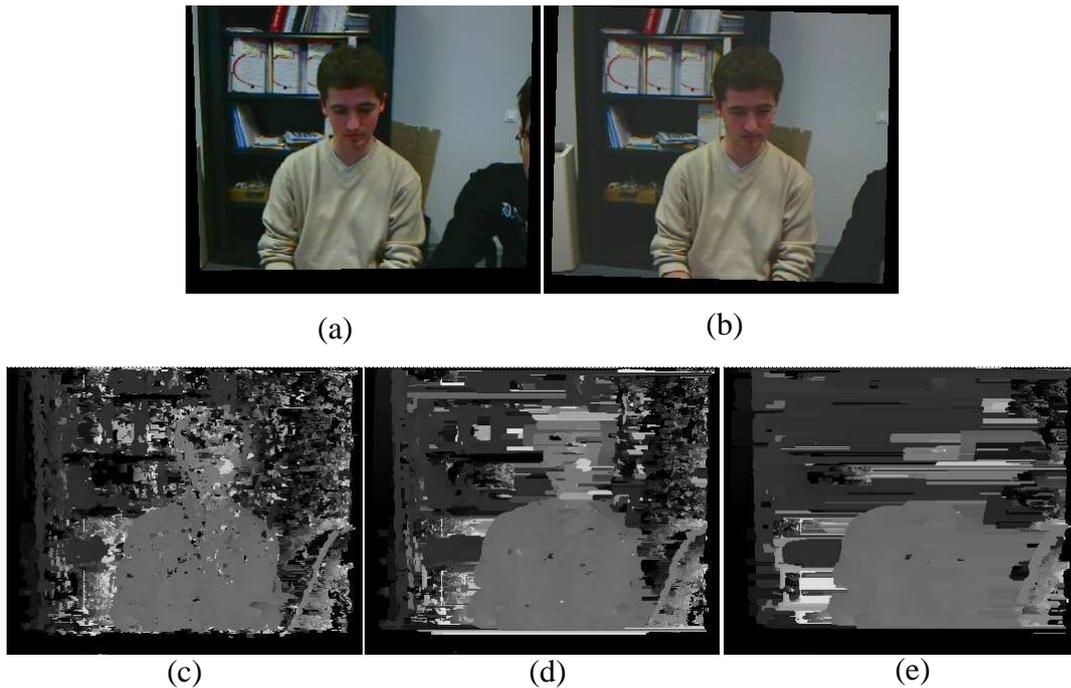


Figure 29 – L2MP : (a) Gauche (b) Droite (c) Census 3x3 (d) Census 5x5 (e) Census 7x7

Ce set d'image à été capturé dans les locaux du L2MP et rectifié grâce à notre programme de vision stéréoscopique (Section III). On applique à ce set d'image la transformée de Census avec des tailles différentes pour la fenêtre de corrélation. Cette étude nous permet de mettre en évidence la nécessité de s'orienter vers un algorithme capable d'adapter la taille de la fenêtre de corrélation en fonction de la texture de la scène à apparier. En effet, Une scène avec peu de zones homogènes devra être traitée avec une fenêtre de petite taille à l'inverse d'une scène comportant beaucoup de zones homogènes.

Par ailleurs on constate sur ce jeu d'images un comportement correct hormis sur certaines zones de couleur uniforme (visage) ou comportant de multiples discontinuités.

## V Évolutions Planifiées

Les objectifs de départ de ce stage étaient la conception d'un programme de vision 3D autonome permettant de disposer d'un modèle fonctionnel de système de vision ainsi que l'étude et l'implémentation d'un certain nombre d'algorithmes d'appariement et de mesures de similitude. D'ici la fin du stage il reste un certain nombre de tâches à réaliser, concernant principalement l'objectifs d'implémentation et de caractérisation des différents algorithmes de corrélation. Elles se divisent en deux directions principales : caractérisation et amélioration de l'algorithme.

Dans un premier temps il faudra poursuivre et approfondir la caractérisation des algorithmes et des mesures de similitudes implémentées de façon à disposer d'une bonne étude comparative des différentes techniques. Ceci permettra d'affiner les possibilités existantes pour une éventuelle implémentation silicium.

Dans un second temps il convient de poursuivre les améliorations de l'algorithme de corrélation implémenté, de façon à le rendre plus performant et plus polyvalent

Dans le cadre de l'amélioration de l'algorithme plusieurs points sont à réaliser : terminer l'implémentation de la mesure de similitude NCC (Normalized Cross-Correlation), ajouter la possibilité de normaliser les mesures SAD et SSD par rapport à la valeur moyenne des pixels sur la fenêtre de corrélation, améliorer l'efficacité temporelle de l'algorithme et enfin étudier la possibilité d'utiliser une fenêtre de corrélation de taille réglable automatiquement en fonction du type de surface.

Enfin il pourrait être intéressant de réussir à caractériser de façon précise le système stéréoscopique lui-même, grâce à notre algorithme (distances de fonctionnement, rapidité, influence de la qualité des caméras, etc.).

## Conclusion

Ce stage de fin d'études d'une durée de six mois (février 2007-juillet 2007) à été réalisé au sein de la division AST de STMicroelectronics et consacré à l'étude et à la conception d'un système stéréoscopique fonctionnel sur support informatique. Ce système va permettre de réaliser la caractérisation de différents algorithmes de corrélation afin d'évaluer l'intérêt d'une implémentation silicium d'un tel système.

La conception de ce système à été réalisée autour de deux axes principaux : dans un premier temps avec la réalisation en C++ d'un programme de vision 3D et dans un second temps par la conception d'un algorithme de corrélation permettant d'utiliser différentes mesures de similitudes.

Dans le cadre de ce stage j'ai pu prendre en main un outil puissant, le langage C++ et découvrir quelques règles de programmation essentielles. Par ailleurs, la conception d'un algorithme nécessitant une grande part de réflexion et de recherche cette étape à été pour moi particulièrement enrichissante. J'ai par ailleurs pu découvrir le travail d'ingénieur ainsi que la réalité et la vie quotidienne d'une grande entreprise.

Les systèmes de vision, d'estimation des distances sont de plus en plus nombreux mais aussi de plus en plus sophistiqués. La stéréoscopie est une technique encore minoritaire dans ce domaine mais qui possède un fort potentiel de croissance. En effet, elle permet d'obtenir des résultats globalement corrects avec un nombre de contraintes relativement faibles. Les applications immédiates sont multiples : suivi de personnes afin d'analyser le trajet dans un magasin par exemple mais aussi assistance à la conduite ou jeu en 3D immergée dans son salon sans autre accessoires qu'une paire de caméras...

## Bibliographie

- [1] J.Woodfill B.Herzen. Real-time stereo vision on the PARTS reconfigurable computer. *In IEEE Workshop in FPGAs for Custom Computing Machines*, pages 242-250, 1993.
- [2] O.Faugeras, B.Hotz, H.Mathieu, T.Viéville, Z.Zhang, P.Fua, E.Théron, L.Moll, G.Berry, J.Vuillemin, P.Bertin, and C.Proy. Real time correlation based stereo: algorithms, implementations and applications. *Technical Report 2013, INRIA, 1993*
- [3] L.D Stefano, M.Marchionni. A Fast Area-Based Stereo Matching Algorithm. *DEIS-ARCES, University of Bologna in Image and Vision Computing 22, pages 983-1005, 2004.*
- [4] Radu Horaud et Olivier Monga. Vision par ordinateur : outils fondamentaux. *Deuxième édition. Editions Hermès.*
- [5] Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano and Masaya Tanaka. A Stereo Machine for Video-rate dense depth mapping and it new applications, *The Robotics Institute, Carnegie Mellon University Pittsburgh, IEEE pages 196-202, 1996*
- [6] Yunde Jia. A Miniature Stereo Vision Machine (MSVM-III) for Dense Disparity Mapping, *17<sup>th</sup> International Conference on Pattern Recognition, 2004*
- [7] A. Darabiha, Video-Rate Stereo Vision on Reconfigurable Hardware, *Master's thesis, University of Toronto, 2003.*
- [8]S.Birchfield and C.Tomasi. Depth discontinuities by pixel-to-pixel stereo. *Technical Report STAN-CS-TR-96-1573. Stanford University. July 1996*
- [9] Kun Wang. Adaptive Stereo Matching Algorithm Based on Edge Detection. *International conference on Image Processing (ICIP). 2004*
- [10] A.Fusiello, E.Trucco, A.Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Application 2000. 12 : pages 16-22*
- [11] A.Fusiello. Epipolar Rectification. 2000. <http://www.sci.inivr.it/~fusiello>
- [12] Middlebury's Image data Set : Middlebury Stereo Vision Page. <http://cat.middlebury.edu/stereo/data.html>
- [13] J.Y Bouguet Calibration Toolbox : <http://www.vision.caltech.edu/bouguet/>
- [14] Z. Zhang. "A flexible new technique for camera calibration". *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.*
- [15]J.Banks, M.Bennamoun and P.Corke. Non Parametric Techniques for fast and robust stereo matching. IAIF'97. Nov 97

## Sommaire des Annexes

Annexe 1 : Lexique .....	i
Annexe 2 : Présentation d'OpenCv.....	ivv
Annexe 3 : Calcul de la matrice Fondamentale .....	iv
Annexe 4 : Calibration – Influences des différents paramètres .....	viv

## Annexe1 : Lexique

**Matrice de projection perspective** : Une matrice de projection perspective est une matrice qui permet d'assurer un changement de repère. Elle réalise en fait la projection de tous les points d'un repère vers un autre. Elle est composée d'une rotation spatiale et d'une translation.

**Disparité** : La disparité se définit comme la différence des abscisses de deux points correspondants. L'un appartenant à l'image gauche et l'autre à l'image droite la disparité représente la distance entre les deux points dans l'image. La disparité est inversement proportionnelle à la profondeur.

**Occlusion** : Un point est dit en occlusion lorsqu'il est visible dans une image de la paire stéréoscopique et qu'il ne l'est pas dans l'autre. Il peut être soit masqué par un objet soit hors du champ de vision d'une des deux caméras.

**Matrice Essentielle** : La matrice essentielle est la matrice qui lie dans le repère caméra, les coordonnées d'un point avec l'équation d'une droite correspondante.

**Matrice Fondamentale** : La matrice fondamentale est la matrice qui permet de lier les coordonnées dans le repère image référence d'un point avec l'équation d'une droite appelée droite épipolaire dans l'image correspondante. S'il existe le point correspondant se trouve sur cette droite.

**Épipolaire (géométrie)** : la géométrie épipolaire est la géométrie qui décrit la conformation d'un capteur stéréoscopique. On distingue particulièrement dans cette géométrie les épipôles et les lignes épipolaires.

**Ligne épipolaire** : Une ligne épipolaire est définie grâce à la matrice fondamentale. Elle est le fruit de la contrainte épipolaire qui dit que tout point d'une image trouve son correspondant sur une ligne dans l'autre image. Cette ligne se nomme la ligne épipolaire et passe par l'épipôle.

**Épipôle** : Toutes les lignes épipolaires d'une image sont concourantes en un point. Ce point est l'épipôle de cette image.

**RANSAC** : Random Sample Algorithm Consensus. Algorithme utilisé entre autre pour l'estimation de la matrice fondamentale

**Distorsion** : Le phénomène de distorsion peut se manifester de diverses façons mais se traduit toujours par une déformation des résultats. Dans ce cas, la distorsion se traduit par une déformation visuelle des images visualisées. On aborde également la notion de distorsion radiométrique qui se traduit par une déformation du spectre d'intensité lumineuse.

**Fenêtre de Corrélation** : La fenêtre de corrélation est utilisée par les mesures de similitudes. Il s'agit du voisinage sur lequel va être déterminée la ressemblance ou la non ressemblance entre deux zones. Cette fenêtre est carrée et peut être de taille variable ou fixe.

**Paramètres intrinsèques** : Ce sont des paramètres inhérents à la caméra qui permettent d'établir la transformation permettant de passer du repère caméra au repère image. On parle de matrice caméra.

**Paramètres extrinsèques** : Ce sont des paramètres qui permettent d'établir les équations de passage entre le repère caméra et le repère de calibration. Ils sont au nombre de 12 et représentent une rotation ainsi qu'une translation. Ces paramètres varient selon la position de la mire de calibration.

**Rectification** : La rectification est une étape du processus de vision 3D qui vise à simplifier la contrainte épipolaire en rendant les lignes épipolaires parallèles : après la rectification des images la recherche de correspondant se réduira à une simple recherche sur la ligne correspondante dans l'autre image au lieu de prendre place sur la ligne épipolaire.

**API** : Application Programming Interface : désigne un jeu de fonctions ou de méthodes qui sont utilisables pour accéder à certaines fonctionnalités.

**Centre de projection** : Le centre de projection est le centre du repère caméra.

**Point principal** : Le point principal est le centre du repère image, il est généralement situé au point milieu de l'image.

**Distance focale** : La distance focale est la distance qui sépare le point nodal d'émergence (ou centre optique) du capteur. Plus la focale est longue et plus l'angle de vision est faible.

**Corrélation** : La corrélation est l'action de mettre en correspondance les points similaires. On forme des paires de points. Les outils pour la corrélation sont les mesures de similitude et l'algorithme d'appariement.

## Annexe2 : Présentation d'Open Cv

Open Cv (Open Computer Vision) est une Librairie Open Source qui comporte un certain nombre d'algorithmes et d'exemples de code pour la vision par ordinateur.

Cette librairie vise à simplifier la conception d'applications dédiées à la vision en temps réel, la reconnaissance de contours, la segmentation etc... Le fait qu'elle soit Open Source signifie qu'elle est très évolutive puisque le développement de nouvelles fonctions, nouveaux algorithmes, sont réalisés par une vaste communauté.

Par ailleurs Open Cv est bâti sur IPL (Intel Image processing library), dont elle reprend le format d'images (IPL Image) ainsi qu'un certain nombre de fonctions de filtrage, etc. Elle implémente également un certains nombres d'algorithmes exclusifs. La librairie rajoute également un certain nombre de notions telles que la gestion des espaces couleurs RGB.

Open Cv est composée de **quatre grands blocs de fonctions**, comportant un total approximatif de 300 fonctions C.

- **CxCore** comporte les Fonctions de base d'Open Cv (fonctions de dessin, de gestion des points etc.)
- **Cv** comporte les Fonctions principales du traitement de l'image : un certain nombre de fonctions de filtrage (Sobel, Canny...) ainsi que les fonctions de calibration, appariement etc...
- **HighGui** comporte les fonctions de gestion des interfaces graphiques. Celle-ci sont simplifiée et permettent principalement la création de fenêtres graphiques dans lesquelles seront affichées des images ou des flux vidéo.
- **CvCam** contient un certain nombre de fonctions de gestions de Webcams, utiles pour ouvrir des flux vidéo.

Par ailleurs OpenCv dispose d'une communauté importante qui se manifeste sur un certain nombre de groupes de discussion. Nous avons pour notre part utilisé le groupe de discussion Yahoo ! qui est un des groupes les plus actifs.

Enfin, il existe un certain nombre de tutoriels, cours et documentation permettant de prendre en main la librairie. La documentation officielle peut être trouvée dans le répertoire /docs du chemin d'installation d'OpenCv.

Notons par ailleurs que la librairie fonctionne sous windows & Linux, et est adaptée pour les langages C++ et Python.

### Références :

[1] Programming With Intel Ipp (Integrated Performance Primitives) And Intel Opencv (Open Computer Vision) Under Gnu Linux. A Beginner's Tutorial, Jérôme Landré.

[2] Yahoo! OpenCv Discussion Group: <http://tech.groups.yahoo.com/group/OpenCV/>

### Annexe 3 : Calcul de la matrice fondamentale

La matrice fondamentale est la matrice qui permet de calculer pour chaque point d'une image l'équation de la ligne épipolaire associée. Cette matrice est de rang 2. Rappelons que la ligne épipolaire contient, s'il existe, le point correspondant dans l'autre image de la paire stéréoscopique. Il existe deux méthodes qui permettent d'obtenir cette matrice épipolaire et donc une estimation globale de la géométrie de l'espace.

La première méthode est un calcul direct à partir des valeurs obtenues à la calibration des capteurs. On calcule dans un premier temps la matrice Essentielle, qui lie un point du repère caméra à une ligne de ce repère caméra. Cette matrice, notée  $E$  se calcule de la façon suivante :

$$E = \begin{pmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{pmatrix} \times \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (1)$$

Où  $b_i$  et  $r_{jk}$  sont les coefficients de la matrice stéréo présentée au paragraphe II.3.2.2.

Une fois cette matrice essentielle déterminée on peut retrouver la matrice fondamentale par un simple changement de repère grâce à la matrice caméra.

$$\mathbf{F} = (\mathbf{C}'^{-1})^t \mathbf{E} \mathbf{C}^{-1} \quad (2)$$

La seconde méthode est une méthode d'approximation qui permet d'obtenir une estimation de la matrice fondamentale à partir d'un certain nombre de correspondances entre deux images. Cette estimation de la géométrie épipolaire, appelée calibration faible pourra permettre d'obtenir rapidement, sans besoin de calibration forte (i.e. calcul de tout les paramètres intrinsèques et extrinsèques) une information sur la configuration du système stéréoscopique et particulièrement de disposer de la contrainte épipolaire, notion essentielle pour l'appariement. (II.5.2.1)

Cette approximation peut être réalisée grâce à diverses méthodes :

- Méthode linéaire : en utilisant l'équation de la contrainte épipolaire :

$$m'^t \times F \times m = 0 \quad (3)$$

Avec :

$$m = \begin{pmatrix} u \\ v \end{pmatrix} \quad m' = \begin{pmatrix} u' \\ v' \end{pmatrix} \quad \text{et} \quad F = \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \quad (4)$$

On obtient en développant :

$$F_{11}u'u + F_{21}v'u + F_{31}u + F_{12}u'v + F_{22}v'v + F_{32}v + F_{13}u' + F_{23}v' + F_{33} = 0 \quad (5)$$

Si l'on dispose de neuf correspondances de points gauche/droite il deviendra possible de déterminer la matrice fondamentale en résolvant un système de 9 équations à 9 inconnues. Le nombre de correspondances nécessaires pourra être réduit à 8 dans le cas où on considère la contrainte  $||F|| = 1$ .

On note que la précision de cette méthode dépend entièrement de la qualité des appariements utilisés. La qualité de la matrice épipolaire obtenue sera donc relativement aléatoire.

- Méthode non linéaire : RANSAC

L'algorithme de RANSAC (RANdom Sample Algorithm Consensus) est un algorithme robuste qui utilise au minimum 8 paires de points. Le principe de cet algorithme est de sélectionner de façon aléatoire l'ensemble des points utilisés pour le calcul de la matrice fondamentale. Plusieurs matrices fondamentales sont calculées de cette manière et on choisit celle qui maximisera le nombre de points remplissant la contrainte épipolaire (3).

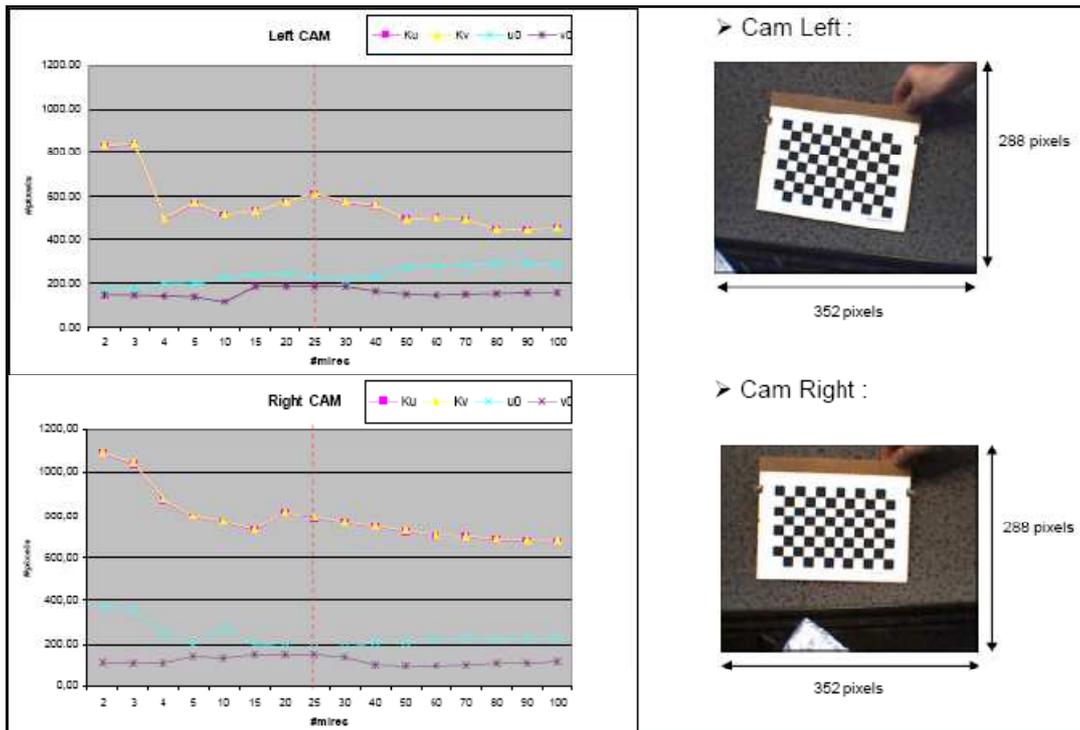
Si l'on compare les résultats obtenus entre les deux méthodes (à partir d'une calibration forte ou par estimation) on constate dans un premier temps que si la matrice fondamentale obtenue par le calcul à partir des paramètres de calibration forte est bien de rang deux, la matrice estimée est elle de rang 3. Par ailleurs lorsque l'on trace les lignes épipolaires dans une image on obtient en général plus de précision et plus de points correspondants avec une matrice fondamentale calculée.

## Annexe 4 : Calibration – Influence des différents paramètres

On présente ici une version simplifiée de l'étude de l'influence de trois paramètres sur les résultats de la calibration (paramètres intrinsèques). Les paramètres variant sont le nombre de prises de vues, la résolution et la distance de la mire aux caméras. Cette étude a été réalisée par J.Roucheton dans le cadre de son stage de fin d'étude.

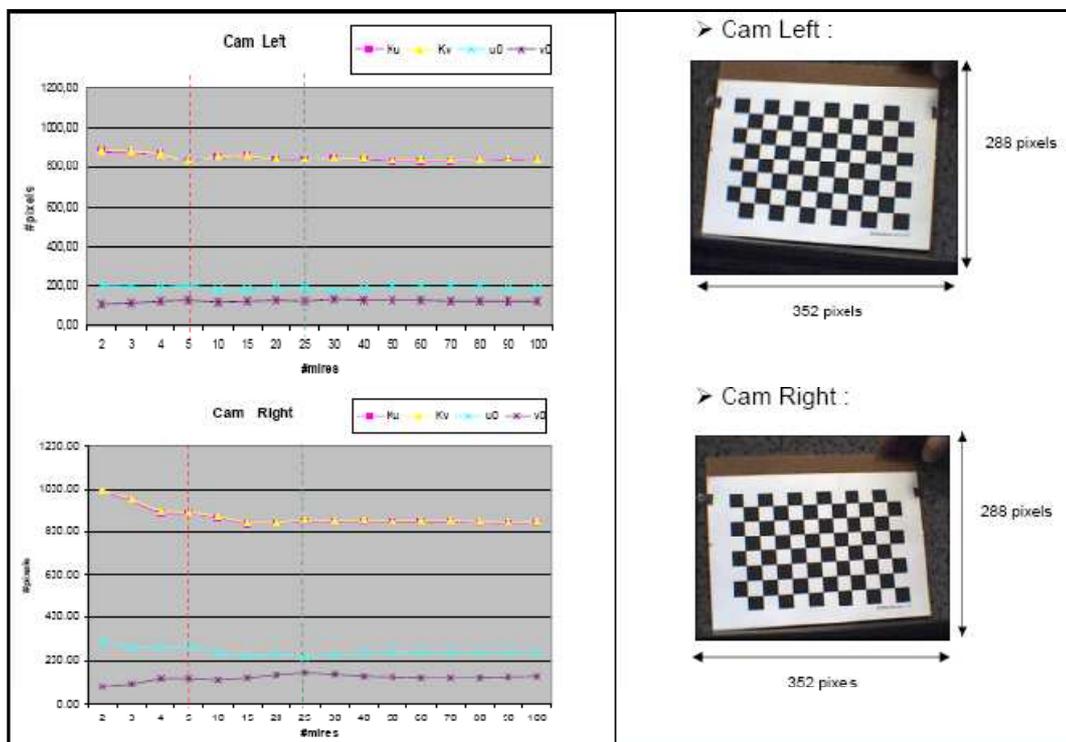
### 1<sup>er</sup> cas :

- Écartement des caméras : » **24 cm**
- Distance de la mire par rapport aux caméras : » **105 cm**
- Position des mires entre les prises de vues : **variable**
- Cam Left : **SN-510B**  
**352 x 288 (pixels)**
- Cam Right : **SN-510B**  
**352 x 288 (pixels)**



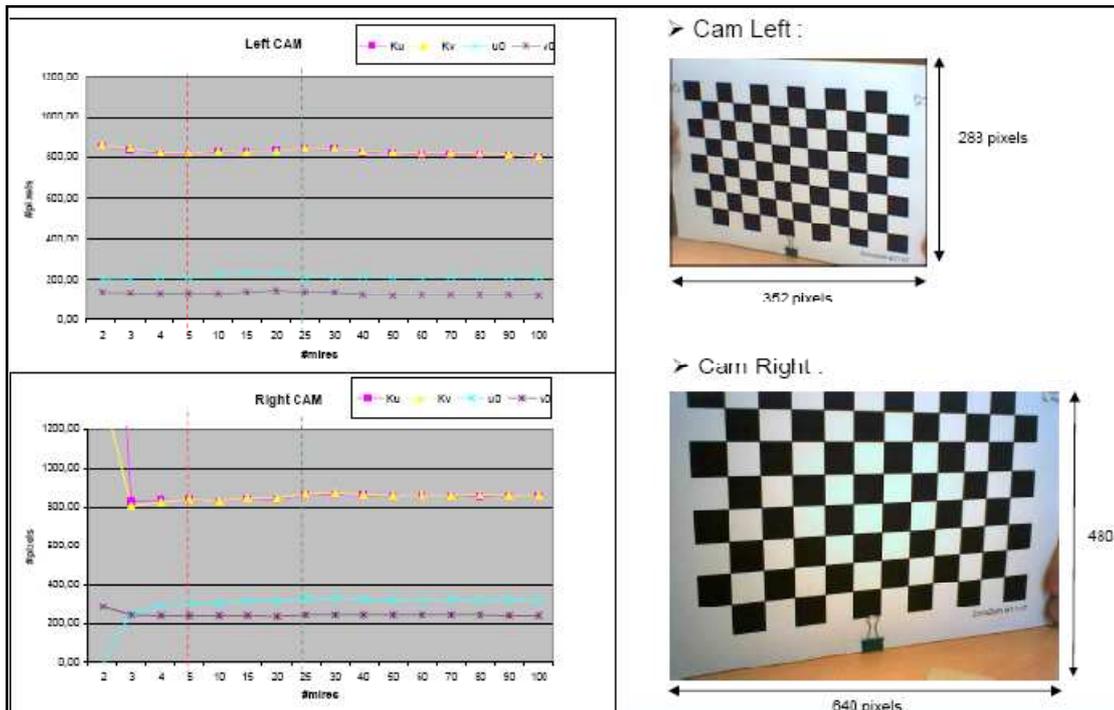
2<sup>ème</sup> cas :

- Écartement des caméras : » **24 cm**
- Distance de la mire par rapport aux caméras : » **78 cm**
- Position des mires entre les prises de vues : **variable**
- Cam Left : **SN-510B**  
**352 x 288 (pixels)**
- Cam Right : **SN-510B**  
**352 x 288 (pixels)**



3<sup>ème</sup> cas :

- Écartement des caméras : » **24 cm**
- Distance de la mire par rapport aux caméras : » **30 cm**
- Position des mires entre les prises de vues : **variable**
- Cam Left : **SN-510B**  
**352 x 288 (pixels)**
- Cam Right : **SPC 610NC**  
**640 x 480 (pixels)**



Observations :

- On constate bien que  $ku = kv$ . (pixel de type carré)
- Plus les mires sont éloignées des caméras, plus la plage de pixel utilisée pour ces dernières est restreinte et plus la variation des focales est importante.
- L'utilisation d'un nombre de prises de vues inférieur à 5 mires entraîne des résultats très médiocres.

**Conclusions :**

- Pour obtenir des résultats cohérents et stables :
- Utiliser la **plage maximale de pixel** pour les mires en plaçant ces dernières le plus près des caméras.
- Effectuer un minimum de **25 prises** de vues lors de la calibration

**Résumé :**

Ce stage à été réalisé au sein de la division AST de STMicroelectronics, en collaboration avec le laboratoire de recherche L.2.M.P (Laboratoire de Microélectronique et Matériaux de Provence) dans le cadre du projet de collaboration PACA PS3-16.

Les objectifs de ce stage étaient doubles : dans un premier temps réaliser la conception d'un programme de vision 3D pouvant servir de modèle et réaliser la caractérisation de ce modèle et dans un second temps étudier et implémenter un certain nombre d'algorithmes et de techniques d'appariement.

L'objectif final était d'étudier la possibilité de s'orienter vers une implémentation silicium d'un dispositif de vision 3D stéréoscopique. La vision stéréoscopique, technique simple de vision qui rejoint par le principe la vision humaine possède en effet un large champ d'applications potentielles, du jeu vidéo à l'aide à la conduite en passant par la robotique et la domotique. Ce champ est actuellement limité par le coût élevé des dispositifs.

Ce travail à été mené en collaboration avec un stagiaire du L.2.M.P ce qui a permis d'approfondir les recherches dans certains domaines tels que la rectification ou la calibration. On trouvera dans ce rapport un descriptif des différentes possibilités offertes par notre programme ainsi qu'une présentation et une étude détaillée des différentes mesures de similitudes implémentées.

---

**Absract:**

This internship has been performed in the AST division of STMicroelectronics, in collaboration with L2MP. The two main goals of this internship were the following: in a first time create a working 3D Vision program in order to establish a model and realize the characterization of this model and in a second time study and implement a set of correlation algorithms.

The final objective of this project is to study the opportunity of a silicon implementation of from videos to driver's help through robotics but this range is shortened by the high cost of the stereo systems.

This work has been performed in collaboration with an intern from L2MP which has permit to lead the research further in some domains such as rectification techniques or calibration.

In this report, you will find a description of the different possibilities offered by our 3D Vision program and a study of the different similitude measures implemented in our code.

Resumé – Abstract

---

Titre :	Vision 3D Stéréoscopique
Auteur :	MALLERON Vincent
Année :	2007
Langue :	Français
Mots Clés :	Stéréoscopie, Appariement, Corrélacion, Censur, SSD, SAD, NCC, Rang, OpenCv, Vision

Ce stage à été réalisé au sein de la division AST de STMicroelectronics, en collaboration avec le laboratoire de recherche L.2.M.P (Laboratoire de Microélectronique et Matériaux de Provence) dans le cadre du projet de collaboration PACA PS3-16.

Les objectifs de ce stage étaient doubles : dans un premier temps réaliser la conception d'un programme de vision 3D pouvant servir de modèle et réaliser la caractérisation de ce modèle et dans un second temps étudier et implémenter un certain nombre d'algorithmes et de techniques d'appariement.

L'objectif final était d'étudier la possibilité de s'orienter vers une implémentation silicium d'un dispositif de vision 3D stéréoscopique. La vision stéréoscopique, technique simple de vision qui rejoint par le principe la vision humaine possède en effet un large champ d'applications potentielles, du jeu vidéo à l'aide à la conduite en passant par la robotique et la domotique. Ce champ est actuellement limité par le coût élevé des dispositifs.

Ce travail à été mené en collaboration avec un stagiaire au L.2.M.P ce qui a permis d'approfondir les recherches dans certains domaines. On trouvera dans ce rapport un descriptif explicatif des différentes possibilités offertes par notre programme ainsi qu'une présentation et une étude détaillée des différentes mesures de similitudes implémentées.