

# Liste des abréviations

AFM : Atomic Force Microscopy.  
AVI : Audio Video Interleave.  
BMP : Bitmap.  
CCD : Charge-Coupled Device.  
CSP : Chip Scaled Packaging.  
DP : Dynamic Programming.  
DSI : Disparity Space Image.  
FPGA : Field Programmable Gate Array.  
FPS : Frame Per Second.  
GPS : Global Positioning System.  
I/O : Input/Output.  
IP : Internet Protocol.  
LCD : Liquid Crystal Display.  
PCIe : Peripheral Component Interconnect express.  
R&D : Research and Development.  
SAD : Sum of Absolute Differences.  
SAM : Scanning Acoustic Microscope.  
SEM : Scanning Electron Microscope .  
SMT : Surface Mount Technology.  
SNR : Signal to Noise Ratio.  
SSD : Sum of Squares Differences.  
TIFF : Tagged Image File Format.  
USB : Universal Serial Bus.

VDC : Voltage Direct Current.

WTA : Winner Take All.

ZNCC : Zero Normalized Cross Correlation.

# Liste des figures

|  |    |
|--|----|
| Figure 1 : Diagramme de Gantt du projet.....   | 22 |
| Figure 2 : Modèles de capteurs stéréoscopiques passifs .....   | 25 |
| Figure 3 : Exemple des images stéréos .....  | 26 |
| Figure 4 : Carte de disparité avec la méthode ponctuelle.....  | 27 |
| Figure 5 : Carte de disparité en utilisant la méthode locale .....   | 29 |
| Figure 6 : Etape de traitement pour la méthode semi-globale.....   | 31 |
| Figure 7 : Illustration d'un arbre pris par deux caméras .....   | 32 |
| Figure 8 : Schéma de la plateforme.....  | 33 |
| Figure 9 : Architecture de l'algorithme .....  | 36 |
| Figure 10 : Images Bayer .....   | 38 |
| Figure 11 : Images en niveau de gris.....  | 38 |
| Figure 12 : Image obtenue après soustraction du Background .....   | 39 |
| Figure 13 : Image après application du filtre unsharp.....   | 39 |
| Figure 14 : Image obtenue après application du filtre median .....   | 40 |
| Figure 15 : Image obtenue après Binarisation.....  | 41 |
| Figure 16 : Image après application des opérations morphologiques .....  | 41 |
| Figure 17 : Image obtenue après remplissage des trous.....   | 42 |
| Figure 18 : Algorithme de mesure de vitesse par stéréoscopie .....   | 43 |
| Figure 19 : Courbe 3D résultante d'une cross-corrélation avec la fonction 'normxcorr2' .....                             | 47 |
| Figure 20 : Courbe des disparités calculées pour 30 images.....  | 48 |
| Figure 21 : Courbe des distances calculées pour 30 images .....  | 49 |
| Figure 22 : Une coupe de la courbe du maximum (en bleu) et l'approximation polynomiale de cette dernière (en rouge)..... | 49 |
| Figure 23 : Les valeurs des disparités en utilisant une approximation polynômiale.....                                   | 50 |
| Figure 24 : Courbe des distances en utilisant une approximation polynomiale.....   | 50 |
| Figure 25: PC MXC-6000 .....   | 54 |
| Figure 26: Dalsa HC-1400 .....   | 56 |
| Figure 27: MicroEnable IV .....  | 58 |
| Figure 28: SISO ME IV-Trigger I/O.....   | 59 |

# Liste des tableaux

|                                       |    |
|---------------------------------------|----|
| Tableau 1 : Table des résultats ..... | 51 |
|---------------------------------------|----|

# Table des matières

|   |           |
|---|-----------|
| Dédicace .....  | 3         |
| Remerciements .....   | 4         |
| Résumé .....  | 5         |
| Abstract .....  | 6         |
| Liste des abréviations .....                                    | 7         |
| Liste des figures .....   | 9         |
| Liste des tableaux .....  | 10        |
| Table des matières .....  | 11        |
| Introduction générale.....                                      | 14        |
| <b>Chapitre 1.....</b>  | <b>15</b> |
| <b>Contexte général du projet .....</b>                         | <b>15</b> |
| 1 Contexte général du projet .....                              | 16        |
| 1.1 Organisme d'accueil.....                                    | 16        |
| 1.1.1 MAScIR .....  | 16        |
| 1.1.2 MAScIR MicroElectronics.....                              | 16        |
| 1.1.3 MAScIR Biotechnologie .....                               | 18        |
| 1.1.4 Nanomatériaux et Nanotechnologies .....                   | 19        |
| 1.2 Contexte et objectifs du projet .....                       | 19        |
| 1.2.1 Contexte du projet .....                                  | 19        |
| 1.2.2 Objectifs du projet.....                                  | 21        |
| 1.2.3 Conduite du projet.....                                   | 21        |
| 1.3 Conclusion.....   | 22        |
| <b>Chapitre 2 .....</b>   | <b>23</b> |
| <b>La stéréovision .....</b>                                    | <b>23</b> |
| 2 La stéréovision .....   | 24        |
| 2.1 Définition .....  | 24        |
| 2.1 Chaîne de traitement.....                                   | 25        |
| 2.2 La mise en correspondance .....                             | 26        |
| 2.2.1 Définition .....  | 26        |
| 2.2.2 Principales difficultés de la mise en correspondance..... | 26        |
| 2.3 Méthodes de mise en correspondance .....                    | 27        |
| 2.3.1 Méthode ponctuelle .....                                  | 27        |
| 2.3.2 Méthode locale .....                                      | 28        |

|   |   |           |
|---|---|-----------|
| 2.3.3                                       | Méthode globale .....   | 30        |
| 2.3.4                                       | Méthode semi-globale .....  | 31        |
| 2.4   | Formule d'estimation de distance par stéréoscopie.....                    | 32        |
| 2.5   | Architecture de la plateforme radar .....                                 | 33        |
| 2.5.1                                       | Présentation de la plateforme .....                                       | 33        |
| 2.5.2                                       | Vue globale .....   | 33        |
| 2.5.3                                       | Description du matériel .....   | 33        |
| 2.6   | Conclusion.....   | 34        |
| <b>Chapitre 3</b>                           | .....   | <b>35</b> |
| <b>Architecture globale de l'algorithme</b> | .....   | <b>35</b> |
| <b>3</b>                                    | <b>Architecture de l'algorithme</b> .....                                 | <b>36</b> |
| 3.1   | Architecture globale .....  | 36        |
| 3.2   | Description .....   | 37        |
| 3.2.1                                       | Images Bayer.....   | 37        |
| 3.2.2                                       | Détection et soustraction du Background : .....                           | 38        |
| 3.2.3                                       | Filtrage .....  | 39        |
| 3.2.4                                       | Binarisation .....  | 40        |
| 3.2.5                                       | Opérations morphologiques .....   | 41        |
| 3.2.6                                       | Détection de Blobs .....  | 42        |
| 3.2.7                                       | Calcul de la cross-corrélation.....                                       | 42        |
| 3.2.8                                       | Calcul de la distance de l'objet au système de vision stéréoscopique..... | 42        |
| 3.2.9                                       | Calcul de la distance sur la route.....                                   | 42        |
| 3.2.10                                      | Calcul de la vitesse .....  | 43        |
| 3.3   | Travail réalisé.....  | 43        |
| 3.4   | Conclusion.....   | 43        |
| <b>Chapitre 4</b>                           | .....   | <b>44</b> |
| <b>Tests et résultats</b>                   | .....   | <b>44</b> |
| <b>4</b>                                    | <b>Tests et résultats</b> .....   | <b>45</b> |
| 4.1   | Mode d'enregistrement .....   | 45        |
| 4.1.1                                       | Configuration .....   | 45        |
| 4.1.2                                       | Calibration.....  | 45        |
| 4.1.3                                       | Les vitesses enregistrées.....  | 45        |
| 4.2   | Réalisations .....  | 45        |
| 4.2.1                                       | Algorithme pour calcul de la disparité .....                              | 45        |
| 4.2.2                                       | Calcul de la distance sur route.....                                      | 47        |

---

|       |                                |    |
|-------|--------------------------------|----|
| 4.2.3 | Calcul de la vitesse .....     | 47 |
| 4.3   | Problèmes rencontrés .....     | 48 |
| 4.4   | Solutions.....                 | 49 |
| 4.5   | Résultats obtenus.....         | 51 |
| 4.6   | Conclusion.....                | 51 |
|       | Conclusion générale .....      | 52 |
|       | Bibliographies .....           | 53 |
|       | <b>Annexe A</b> .....          | 54 |
|       | <b>Liste du matériel</b> ..... | 54 |
|       | PC MXC-6000.....               | 54 |
|       | Caméra Dalsa HC-1400 .....     | 56 |
|       | FPGA MicroEnable IV.....       | 58 |
|       | SISO ME IV-TRIGGER I/O .....   | 59 |

## Introduction générale

Depuis toujours, la sécurité des moyens de transport a fait l'objet d'une attention particulière, notamment la voiture qui est statistiquement le moyen de transport le moins sûr. Selon de nombreuses études en accidentologie, un accident apparaît comme un dysfonctionnement du système composé du conducteur, du véhicule et de l'environnement. Dans la grande majorité des cas, l'homme est à l'origine de l'accident, soit à cause d'une perception dégradée de l'environnement routier, soit à cause d'un contrôle inadapté de la dynamique du véhicule.

L'excès de vitesse est l'un des facteurs primordiaux des accidents de la route, d'où la nécessité d'implémenter des systèmes de contrôle afin de pénaliser tout dépassement de la vitesse autorisé.

C'est dans ce sens que l'équipe systèmes embarqués de la fondation MAScIR a songé à développer un Radar se basant sur l'effet stéréoscopique obtenu par deux caméras, ce type de Radars utilise des capteurs passifs et vient remédier au problème de son antécédent (mono-caméras).

Ce rapport est organisé en quatre chapitres, dans le premier chapitre nous allons commencer par présenter le contexte général du projet, dans un deuxième chapitre présenter un état d'art sur la stéréovision et les méthodes de mise en correspondance disponibles, dans un troisième chapitre exposer l'architecture globale de l'algorithme et finalement dans un quatrième chapitre présenter le travail réalisé, les problèmes rencontrés, les solutions proposées et les résultats obtenus.

# Chapitre 1

---

## Contexte général du projet

*Dans ce chapitre nous allons présenter le cadre dans lequel s'inscrit ce projet et le cahier des charges à respecter pour la validation de ce travail.*

# 1 Contexte général du projet

## 1.1 Organisme d'accueil

Avant de présenter et de développer notre sujet en profondeur, il paraît, dans un premier temps, opportun de commencer par une présentation générale de la société, de survoler rapidement son historique, afin de mieux développer le sujet et la problématique traités lors de ce travail.

### 1.1.1 MAScIR

MAScIR (Moroccan foundation for Advanced Science, Innovation and Research) est un organisme de recherche à caractère scientifique et technologique. Il est voué à la recherche en nanotechnologie, en biotechnologie, en technologie numérique, en microélectronique, en énergie et en environnement, la fondation se veut présente là où les enjeux de la société l'exigent.

Rassemblant d'éminents chercheurs des quatre coins du monde, MAScIR regroupe des équipes scientifiques œuvrant dans des domaines innovants et complémentaires et met à leur disposition des instruments scientifiques de pointe.

Initialement fondée en 2007 par le Gouvernement Marocain en tant que fondation à but non lucratif.

**MAScIR MicroElectronics** : créé vers la fin de l'année 2008, a pour objectif de devenir un centre de Recherche et Développement dans le domaine de la microélectronique.

**MAScIR BioTechnology** : deuxième centre inscrit dans MAScIR œuvrant dans le domaine de la biotechnologie : recherche et développement des médicaments ou des biocides.

**Nano Technology** : qui a pour mission de mener des recherches appliquées, innovantes et à la fine pointe de la technologie pour créer de la propriété intellectuelle et des prototypes dans le domaine des nanomatériaux et des nanotechnologies. Ces recherches sont menées par une équipe internationale de haut calibre travaillant dans un environnement unique et utilisant une infrastructure de pointe.

### 1.1.2 MAScIR MicroElectronics

**MAScIR MicroElectronics** est un centre d'innovation et de développement des technologies dans le milieu microélectronique. Il se concentre sur le micro packaging, l'ingénierie, les tests

de simulation, le design, la qualification, le prototypage de produits micro-électroniques et les systèmes embarqués. Parmi les projets traités par MAScIR MicroElectronics, nous citons:

- Design et Micro packaging CSP et PILR,
- Tests de fiabilité sur les packages
- Circuits embarqués sur une application wafer level camera fabriqué au Maroc par Nemotek Technologie.

**MAScIR MicroElectronics** fait partie de la fondation à but non lucratif MAScIR dont les domaines de recherche sont la Nanotechnologie, la Microélectronique, les énergies, et l'environnement.

La mission de MAScIR est de promouvoir l'excellence dans la recherche et le développement des technologies au MAROC dans le but de générer de la valeur, des emplois à travers l'intégration dans le marché des technologies avancées dans le but de produire de la propriété intellectuelle et de créer des spin-offs.

**MAScIR MicroElectronics** offre ces services aux clients industriels et partenaires académiques dans bon nombre de domaines:

- Prototypage de Circuits Intégrés incluant le design de packages et la modélisation, l'assemblage et la qualification.
- Systèmes embarqués.
- Management de projets.

#### *a. Missions*

La mission du MAScIR MicroElectronics est de devenir un centre d'excellence en Microélectronique et en Miniaturisation incluant l'électronique Digitale et Analogique, le Logiciel, l'Optique et les Technologies de packaging.

MAScIR MicroElectronics fournit des services pour des clients industriels, mais elle développe aussi son propre business dans les domaines suivants :

- L'intégration et la miniaturisation des systèmes microélectroniques.
- L'analyse de fiabilité et défaillance des produits.
- Modélisation des systèmes complexes.

- Prototypage et industrialisation des produits innovants.
- Industrialisation des idées et résultats académiques.

#### ***b. Laboratoires***

MAScIR possède plusieurs laboratoires équipés de technologie avancée :

- Chambre blanche
- Laboratoire optique
- Laboratoire électronique

#### ***c. Équipements***

- Ligne CSP, Ligne SMT
- SEM, SAM
- X-Ray
- AFM

### **1.1.3 MAScIR Biotechnologie**

Le Centre de Biotechnologie de MAScIR combine une expertise scientifique de classe mondiale avec des laboratoires de recherche pour offrir des solutions efficaces innovantes et économiques afin de répondre à la demande du marché et aux besoins sociaux du Maroc et de sa région.

L'équipe de la biotechnologie médicale est principalement axée sur le développement de kits de diagnostics médicaux innovants ciblant les maladies les plus répandues au Maroc telles que le cancer et les maladies infectieuses. L'équipe s'intéresse également au co-développement des prototypes biogénériques en collaboration avec des partenaires industriels.

L'équipe de la biotechnologie verte se concentre sur des projets dans le domaine de la recherche appliquée des plantes, y compris l'exploitation, l'amélioration et les nouvelles technologies. Les principaux domaines de recherche sont:

- Bioénergie à base de micro-algues
- Crop Breeding
- Biotechnologie microbienne appliquée au développement agricole et alimentaire

### **1.1.4 Nanomatériaux et Nanotechnologies**

Des chercheurs de l'Institut de MAscIR des nanomatériaux et les nanotechnologies travaillent au niveau minuscule d'un nanomètre, où la matière montre des propriétés physiques, chimiques inhabituelles.

La mission du centre est de concevoir, d'inventer et d'encourager la recherche et le développement des applications de la nanotechnologie, favorisant ainsi le transfert de technologie à l'industrie afin de répondre à la demande du marché.

## **1.2 Contexte et objectifs du projet**

### **1.2.1 Contexte du projet**

#### **a. Définition du radar**

Un radar de contrôle routier est un instrument servant à mesurer la vitesse des véhicules circulant sur la voie publique à l'aide d'ondes radar. Ce type de cinémomètre est principalement utilisé afin d'identifier les contrevenants aux limites de vitesse.

#### **b. Principe de fonctionnement**

Ces radars utilisent le principe de l'effet Doppler-Fizeau pour mesurer la vitesse. Ils émettent une onde entretenue (ou continue) qui est réfléchiée par toute cible se trouvant dans la direction pointée. Par effet Doppler, cette onde réfléchiée possède une fréquence légèrement différente de celle émise : plus grande fréquence pour les véhicules s'approchant du radar et plus petite pour ceux s'éloignant. En mesurant la différence de fréquence entre l'onde émise et celle retournée, on peut calculer la vitesse de la cible. Ceci se fait en trouvant le battement entre les deux ondes.

En général, on utilise une longueur d'onde centimétrique des bandes X, K et Ka. La bande X est moins courante depuis quelques années, car elle est facilement perceptible par les récepteurs antiradars. De plus, elle est utilisée par d'autres appareils dont les ouvre-portes de garage. Une alternative au radar est celle du Lidar qui utilise le même principe que le radar Doppler pulsé ; c'est-à-dire qu'il mesure les temps d'aller-retour d'impulsions laser successives pour estimer la vitesse du véhicule.

#### **c. Angle de détection**

Les données ainsi obtenues ne montrent que la vitesse radiale entre le radar et le véhicule. Il faut donc que le sens de déplacement de la cible soit en ligne directe vers le radar. Si la cible

se déplace avec un angle par rapport à cette direction, la vitesse notée n'est que la projection sur la radiale au radar, soit la vitesse réelle multipliée par le cosinus de l'angle (facteur variant de 0 pour un déplacement perpendiculaire au radar à 1 pour celui vers le radar).

Comme la direction radiale pure demande de mettre le radar sur la route pour faire la mesure, les appareils sont généralement calibrés pour prendre les vitesses lorsque le radar est orienté suivant un angle de  $25^\circ$  par rapport à l'axe de circulation des véhicules contrôlés. Cette notion est extrêmement importante puisqu'un écart aussi minime soit-il à cette valeur entraîne des modifications significatives sur les valeurs détectées. Un radar mesurant des vitesses sous un angle de  $30^\circ$  par rapport à l'axe de circulation, un différentiel de seulement  $5^\circ$ , induit une minoration de la vitesse détectée de 4,5 %, soit une erreur de 4 km/h pour un véhicule roulant à 90 km/h et environ 6 km/h pour un véhicule se déplaçant à 130 km/h.

Le respect de cette contrainte est très important pour la validité du calcul compensatoire de l'effet Doppler-Fizeau. Un radar mobile posé rapidement sur le bord de la route ou monté sur un véhicule garé sans attention particulière à l'orientation par rapport à la voie de circulation peut ainsi être responsable de contraventions indues. La formation du personnel manipulant les radars de contrôle routier est donc elle aussi très importante puisque naturellement un angle plus faible que  $25^\circ$  avec l'axe de circulation sera défavorable à l'automobiliste, un angle plus grand que  $25^\circ$  sera favorable à l'automobiliste.

#### **d. Types de radars existants**

- Fixe : c'est un radar automatique fixe est composé de 3 éléments principaux : une cabine radar blindée, un cinémomètre et un dispositif de prise de vue.
- Mobile : peut-être utilisé en divers points du réseau routier, il peut soit être embarqué dans une voiture banalisée soit être installé sur un trépied en bord de route soit encore fonctionner dans une voiture en circulation.
- Tronçon : Contrôle la vitesse moyenne de circulation des véhicules sur une distance plus ou moins longue.
- Chantier : Le radar chantier est un radar semi-fixe destiné à assurer un contrôle des vitesses dans les zones de travaux. Ce radar d'un nouveau genre est un hybride entre le radar fixe et le radar mobile car il est conçu pour être déplaçable et mis en place en moins d'une heure, capable de fonctionner sans opérateur et sur batterie et bien sûr résistant au vandalisme.

### e. Technologies existantes

Il existe deux types de capteurs utilisés dans les systèmes radars :

- Capteurs actifs : mesure la distance en envoyant un signal à l'objet (laser, signal radio, ultra-son,...).
- Capteurs passifs : reçoivent les informations à propos de l'objet (souvent en utilisant la lumière).

Parmi les méthodes passives on trouve le radar mono-caméra déjà développé par l'équipe systèmes embarqués à MAScIR et le radar se basant sur une mesure stéréoscopique en utilisant deux caméras. Ce dernier fait l'objet de ce travail.

### 1.2.2 Objectifs du projet

Les méthodes se basant sur les capteurs actifs présentent de nombreux problèmes tels que :

- L'onde émise se reflète sur l'obstacle le plus proche.
- L'angle de détection doit être égal à  $25^\circ$ .
- Certaines fréquences sont facilement perceptibles par les antiradars.

Le radar mono-caméra présente moins d'inconvénients que ceux déjà cités, mais nécessite des repères sur route dont on connaît la valeur métrique afin d'en tirer par la suite une matrice de redressement nous permettant de savoir la valeur métrique correspondant au pixel sur l'image.

Le radar utilisant l'effet stéréoscopique viendra remédier à ces problèmes et permettra aussi d'avoir une bonne précision.

Ce projet entre dans le cadre des projets réalisés au sein de l'équipe système embarqués à MAScIR. Ce type de radar n'est encore pas développé ailleurs, c'est un brevet retenu par MAScIR.

### 1.2.3 Conduite du projet

#### a. Cahier des charges

Vu les inconvénients présentés par les systèmes radar existants, il s'est avéré nécessaire de développer un système permettant une précision de mesure de vitesse et remédiant aux problèmes des solutions existantes.

Le but de ce travail est de développer un algorithme permettant l'estimation de vitesse par stéréoscopie sur l'environnement Matlab.

Pour l'élaboration de cet algorithme, le travail a été réparti comme suit :

- Recherche bibliographique sur les méthodes existantes.
- Architecture de l'algorithme sur Matlab.
- Test et estimation de précision sur vidéothèque enregistrée avec la plateforme du projet RADAR.

### *b. Planification*

La planification est parmi les phases d'avant-projet. Elle consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de développement en se basant sur le diagramme de Gantt. Grâce aux réunions tenues avec mes encadrants, nous avons été éclairés sur les différentes étapes du projet ainsi que leur séquençement.

Le travail est divisé en trois étapes principales : la première étape consiste à la recherche des solutions existantes et des algorithmes permettant l'estimation de distance entre le système et l'objet en question, la deuxième consiste au calcul de disparité et la troisième étape est l'estimation de la vitesse et le test de précision de l'algorithme.

La figure ci-dessous présente le diagramme de Gantt du projet :

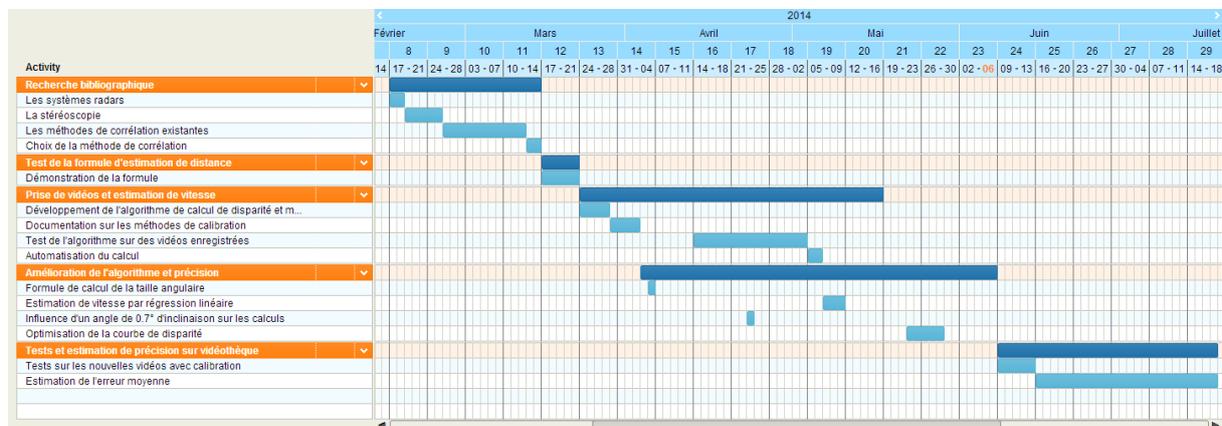


Figure 1 : Diagramme de Gantt du projet

## 1.3 Conclusion

Dans ce chapitre nous avons présenté le contexte général du projet commençant par une présentation de l'organisme d'accueil du stage, ensuite une présentation du cahier de charges du projet et vers la fin le déroulement du projet.

Nous allons présenter dans un prochain chapitre une vue générale sur la stéréoscopie, les algorithmes de mise en correspondance existants, les solutions choisies et finalement une présentation de la plateforme Radar.

# Chapitre 2

---

## La stéréovision

*Dans ce chapitre nous allons présenter une définition de la stéréovision, les algorithmes utilisés pour le calcul de la distance entre le système stéréoscopique et l'objet en question et une description du système Radar.*

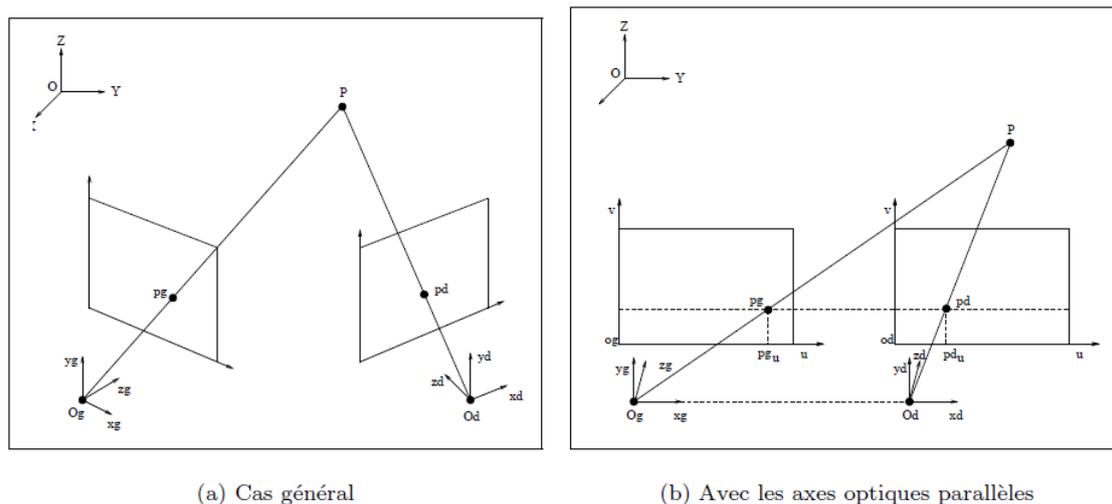
## 2 La stéréovision

### 2.1 Définition

La stéréovision binoculaire est une approche très répandue en vision artificielle. C'est un système qui est calqué sur le système visuel humain. Il possède deux capteurs passifs qui permettent d'acquérir des images de la scène depuis des points de vue proches. Généralement un capteur stéréoscopique est composé de deux caméras CCD dont les axes optiques, à l'image des yeux, ne sont pas parallèles (cf. figure 2 (a)). Chaque caméra est représentée par son modèle dit à sténopé : elle est définie par son centre optique  $O_g$  (pour la caméra gauche), son axe optique  $O_gz_g$  perpendiculaire au plan image et sa distance focale. De nombreuses applications positionnent aussi les caméras de telle sorte que leurs axes optiques soient parallèles (cf. figure 2 (b)). Cette solution offre l'avantage d'une mise en œuvre simplifiée de la contrainte épipolaire : deux points provenant de la projection du même point 3D se trouvent sur la même ligne d'image.

Les principaux challenges de la stéréovision sont :

- Performance temps-réel
- Bruit
- Sensibilité à la lumière
- Réflexion
- Distorsion de la perspective
- Manque de textures
- Mosaique
- Occlusion



(a) Cas général

(b) Avec les axes optiques parallèles

Figure 2 : Modèles de capteurs stéréoscopiques passifs

## 2.1 Chaîne de traitement

Un système de vision stéréoscopique comporte aussi une chaîne de traitements qui doit permettre de retrouver l'information tridimensionnelle (3D) à partir des images acquises. Ces traitements, indispensables à la reconstruction 3D, sont les suivants :

**Le calibrage :** le calibrage consiste à modéliser le capteur stéréoscopique et à le positionner dans le repère de la scène. Pour cela, on détermine la position et l'orientation de chaque caméra dans le repère de la scène. On modélise ensuite l'ensemble du processus de projection sur le plan image réalisée par la caméra.

**L'extraction des primitives d'image :** c'est une étape de segmentation d'image qui consiste à extraire des images (des matrices de pixels) des caractéristiques de plus haut niveau (segments, jonctions, régions, etc.). Ce sont ces primitives qui seront mises en correspondance puis reconstruites. Cette étape est donc très importante puisque la qualité de ses résultats déterminera en partie la qualité de la mise en correspondance donc de la reconstruction.

**La mise en correspondance :** elle consiste à identifier dans les deux images segmentées les primitives ( $pg$ ,  $pd$ ) provenant de la projection d'une même entité 3D,  $P$ . C'est une étape très importante de la stéréovision puisque c'est elle qui permet la reconstruction 3D.

**La reconstruction 3D :** la mise en correspondance fournit des couples de points d'image appariés ( $pg$ ,  $pd$ ). La position du point 3D  $P$  correspondant à cet appariement est déterminée par triangulation : c'est l'intersection des droites 3D  $Ogpg$  et  $Odpg$ .

Après l'étape de la calibration et de l'extraction des primitives, vient l'étape la plus importante et la plus difficile qui est celle de la mise en correspondance. Les différentes méthodes proposées se divisent en quatre catégories : les méthodes ponctuelles, les méthodes locales, globales et semi-globales. L'avantage de la première méthode c'est qu'elle est simple et nécessite un temps de calcul relativement faible mais elle est exposée souvent à des erreurs d'appariement. La deuxième méthode est plus précise et tente de palier aux problèmes de la première méthode mais à son tour présente un inconvénient au niveau du temps requis pour le calcul. La troisième méthode se base sur les deux méthodes déjà citées, elle est plus précise et nécessite un temps de calcul relativement faible mais elle est sensible aux erreurs.

## 2.2 La mise en correspondance

### 2.2.1 Définition

L'objectif est de retrouver sur les deux images les paires de pixels qui correspondent à la projection d'une même entité.

De nombreuses techniques ont été proposées pour tenter de résoudre les problèmes résultants de la mise en correspondance, sachant qu'il est difficile de les prendre tous en compte en même temps, mais qu'il est possible de trouver des compromis. La figure ci-dessous présente un exemple d'images stéréo et leur anaglyphe.

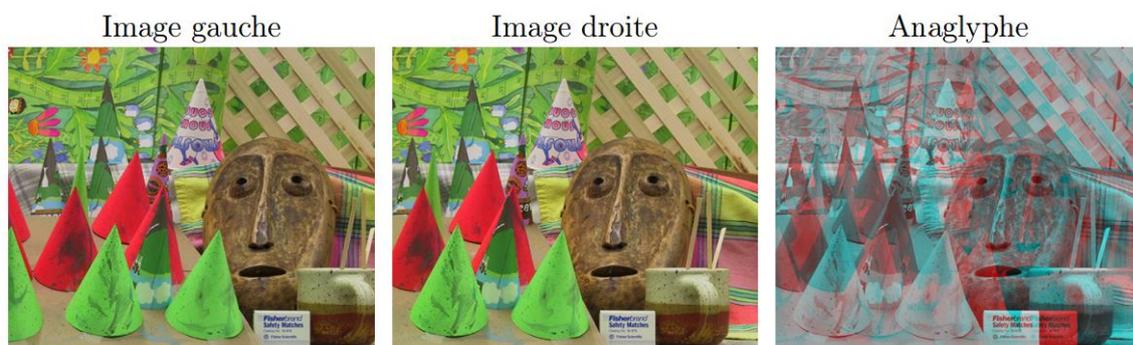


Figure 3 : Exemple des images stéréos

### 2.2.2 Principales difficultés de la mise en correspondance

Les méthodes de mise en correspondance de pixels s'appuient généralement sur l'hypothèse que les voisinages de deux correspondants se ressemblent. Les principales difficultés rencontrées lors d'une mise en correspondance sont dues essentiellement à deux raisons :

- *Information manquante* – L'information qui permettrait de réaliser une mise en correspondance de façon certaine n'est pas présente dans l'image. Il est donc difficile

de réaliser la mise en correspondance sans caractéristique discriminante complémentaire. C'est le cas notamment dans les zones occultées, les zones homogènes, les zones de discontinuité de profondeur et de raccourcissement.

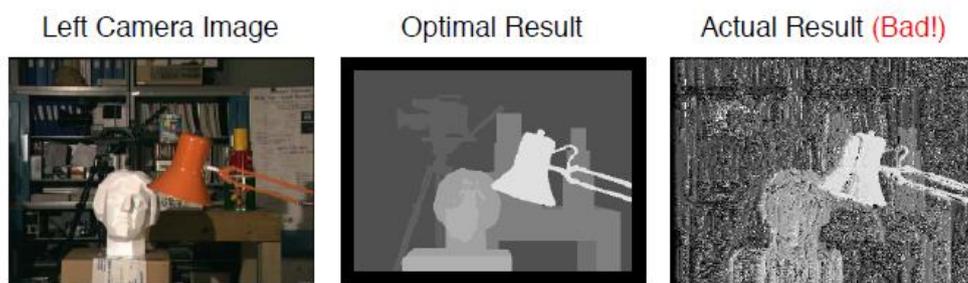
- *Zones occultées* – Il s'agit des zones dans lesquelles les éléments de la scène visibles sur une image n'apparaissent pas sur l'autre image car, vus depuis un autre angle, ces éléments sont cachés par d'autres.
- *Zones de discontinuité de profondeur* – Il s'agit des zones situées aux frontières des objets ayant des profondeurs différentes. Cela peut se traduire par des voisinages de deux correspondants qui ne se ressemblent pas nécessairement.
- *Zones de raccourcissement* – Différents points de la scène se projettent en plusieurs pixels d'une image mais sur un même pixel de l'autre image. Cela peut se produire lorsqu'une surface de la scène est très inclinée par rapport au plan image.

Parmi les principales contraintes de la mise en correspondance on trouve : épipolaires, unicité, ordre et seuil.

## 2.3 Méthodes de mise en correspondance

### 2.3.1 Méthode ponctuelle [3]

Avant l'apparition de la méthode locale, on utilisait la méthode ponctuelle qui consistait à comparer les pixels à part.



**Figure 4 : Carte de disparité avec la méthode ponctuelle**

D'après la figure 4, on constate que la carte de disparité est très bruitée, cela est dû à un faible rapport signal à bruit (SNR). Pour corriger ce problème, on utilise la méthode locale dont le principe est défini ci-après.

### 2.3.2 Méthode locale [3]

La méthode locale base son analyse sur une fenêtre de comparaison autour des pixels à mettre en correspondance. Dans la zone de recherche, un coût d'appariement est obtenu par corrélation entre les deux fenêtres. Les méthodes les plus couramment employées sont la somme des écarts quadratiques (SSD), la somme des écarts absolus (SAD) ou l'inter-corrélation normalisée centrée (ZNCC). Par la suite, la disparité pour laquelle on obtient le meilleur score d'appariement est retenue (approche WTA). Ces méthodes simples présentent l'avantage de nécessiter le moins de ressources et de générer des images de disparité dense, mais elles présentent un fort taux d'erreurs, notamment dans les zones d'occlusion et dans les zones peu texturées.

La méthode locale vient remédier au problème de la méthode ponctuelle en utilisant une fenêtre de correspondance autour du pixel d'intérêt.

Méthodes de calcul de la corrélation les plus courantes :

Somme de différence d'intensité absolue (SAD) :

$$SAD(\mathbf{f}_g, \mathbf{f}_d) = \|\mathbf{f}_g - \mathbf{f}_d\|_1 \quad (1)$$

Sommes de différence carrée (SSD) :

$$SSD(\mathbf{f}_g, \mathbf{f}_d) = \|\mathbf{f}_g - \mathbf{f}_d\|^2 \quad (2)$$

Corrélation croisée centrée normalisée (ZNCC) :

$$ZNCC(\mathbf{f}_g, \mathbf{f}_d) = \frac{(\mathbf{f}_g - \overline{\mathbf{f}_g}) \cdot (\mathbf{f}_d - \overline{\mathbf{f}_d})}{\|\mathbf{f}_g - \overline{\mathbf{f}_g}\| \|\mathbf{f}_d - \overline{\mathbf{f}_d}\|} \quad (3)$$

Avec :

$\mathbf{f}_g$  : image gauche.

$\mathbf{f}_d$  : image droite.

$\overline{\mathbf{f}_g}$  : moyenne de l'image gauche .

$\overline{\mathbf{f}_d}$  : moyenne de l'image droite.

L'algorithme de base de mise en correspondance locale de l'image de référence est donné ci-dessous.

**Fonction** mise en correspondance locale (gauche: Image, droite: Image) : Carte de disparité

**Pour** chaque pixel  $p^{ij}_g$  de gauche faire

**Pour** chaque pixel  $p^{kl}_d$  candidat de droite faire

[Calcul des scores de corrélation entre le pixel étudié et les correspondants possibles]

score  $\leftarrow$  score de corrélation calculé sur les voisinages de  $p^{ij}_g$  et de  $p^{kl}_d$

**Fin Pour**

Associer à  $p^{ij}_g$  le candidat  $\hat{p}^{kl}_d$  ayant obtenu le meilleur score ;

[Calcul de la disparité entre  $p^{ij}_g$  et le correspondant que l'on vient de trouver]

$(d_l \quad d_c)^T = \hat{p}^{kl}_d - p^{ij}_g$

**Fin Pour**

**Retourner** carte de disparité

**Fin**

L'application de cet algorithme sur des images stéréo, donne les résultats de la figure 5.



**Figure 5 : Carte de disparité en utilisant la méthode locale**

D'après la figure ci-dessus, on remarque que les résultats obtenus par application de la méthode locale donne de meilleurs résultats que la méthode ponctuelle. En effet la carte de disparité obtenue est plus dense.

Dans le but d'éliminer les erreurs de cette méthode, il est possible de faire une vérification bidirectionnelle qui correspond à l'utilisation de la contrainte de symétrie. On cherche les correspondants de l'image de gauche vers l'image de droite puis de l'image de droite vers l'image de gauche. Les couples qui apparaissent dans les deux résultats sont alors validés. Dans le cas contraire, où on ne retrouve pas le même correspondant, on considère que les pixels concernés se trouvent dans une zone occultée.

Parmi les difficultés rencontrées lors de la mise en correspondance par fenêtre de corrélation on cite :

- Discontinuité de surface
- Surfaces non-plane
- Structure répétitive
- Surface uniforme

Cependant, malgré les inconvénients de la méthode locale, la corrélation croisée est toujours utilisée en pratique car elle est simple, rapide et requiert une faible mémoire (car on a besoin que de la disparité de chaque pixel seulement).

### 2.3.3 Méthode globale [5]

La méthode globale prend en compte la totalité de l'image. Elle est basée sur le principe suivant : si la carte de disparité calculée est correcte et si on construit une image à partir de l'image de référence (par exemple celle de gauche) et la carte de disparité, alors l'image obtenue doit ressembler à l'image de droite. On cherche alors à trouver la carte de disparité qui maximise une fonction globale de similarité. Des optimisations ont été proposées est qui reposent sur la programmation dynamique (DP).

Cette méthode donne des images de disparité dense et comporte très peu d'erreurs. Elle est cependant complexe à mettre en place et nécessite des ressources de calcul et de mémorisation très importantes, incompatibles avec les contraintes matérielles de l'embarqué.

L'algorithme de mise en correspondance de la méthode globale est donné par :

**Fonction** mise en correspondance globale (gauche : **Image**, droite : **Image**) : **Carte de disparité**

La carte courante  $\leftarrow$  calcul de la disparité initiale () similarité  $\leftarrow$  droite – image déformée

**Tant que** (similarité ! seuil) **faire**

Déforme l'image de référence en fonction de la carte de disparité courante

similarité  $\leftarrow$  droite – image déformée

la carte courante  $\leftarrow$  mise à jour de la disparité

**Fait**

**Retourner** carte de disparité

**Fin**

Tant que la carte de disparité trouvée ne permet pas d'obtenir une similarité satisfaisante entre l'image déformée (à partir de l'image de gauche et de cette carte de disparité) et l'image de droite, on itère.

La méthode globale ajoute aussi une contrainte de lissage, notée  $E_{\text{lissage}}$ , qui montre à quel point la carte de disparité respecte certaines contraintes que l'on se donne (limite du gradient de disparité, contrainte d'unicité). Par conséquent, le problème peut-être normalisé de la manière suivante :

$$\hat{d} = \underset{d}{\operatorname{argmin}} E(d) = E_{\text{attache}}(d) + \lambda E_{\text{lissage}}(d) \quad (4)$$

où  $d$  est la fonction de disparité recherchée ;

$\lambda \in [0 ; 1]$  permet d'ajuster l'influence des deux termes :  $E_{\text{attache}}(d)$  et  $E_{\text{lissage}}(d)$ .

Avec :  $E_{\text{attache}}(d) = \sum_{i,j} C(i,j,d(i,j))$  où  $C(i,j,d(i,j))$  représente le coût de l'appariement du pixel  $(i,j)$  si on lui affecte la disparité  $d(i,j)$ .

### 2.3.4 Méthode semi-globale [5]

Cette méthode se base sur le même principe que la méthode globale mais sur des sous-parties de l'image (des lignes ou des blocs). Le découpage du problème d'optimisation en sous-problèmes permet de diminuer le besoin en ressources de calcul et en ressources mémoires par rapport aux méthodes globales, mais entraîne cependant l'apparition d'artefacts sur l'image de disparité. Une méthode de propagation de croyance appliquée à des blocs de pixels permet la mise en place d'un traitement en parallèle. L'optimisation de la disparité suivant les lignes épipolaires peut s'effectuer par programmation dynamique (DP). Dans cette méthode, l'objectif est de trouver le chemin qui minimise la fonction d'énergie à travers l'espace des disparités possibles sur une ligne (DSI). La programmation dynamique extrait l'ensemble des disparités donnant la meilleure corrélation tout en respectant la contrainte de continuité, ce qui permet l'obtention de cartes de disparité dense, homogènes, mais présentant des défauts sur les contours qui peuvent déformer certains objets.

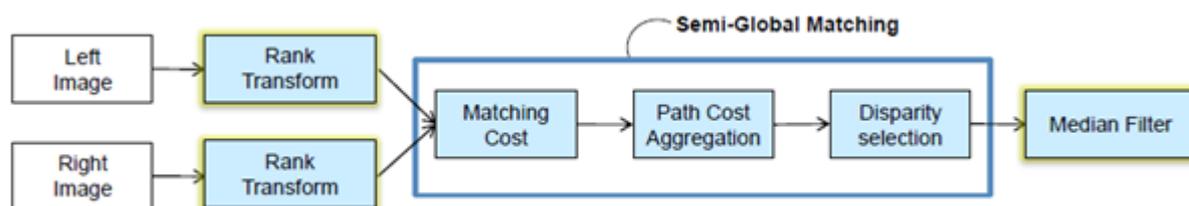


Figure 6 : Etape de traitement pour la méthode semi-globale

## 2.4 Formule d'estimation de distance par stéréoscopie [7]

Cette formule consiste à utiliser les paramètres intrinsèques de la caméra à savoir :

- La distance focale,
- L'ouverture des caméras.

La distance  $D$  peut être calculée à partir de la formule suivante comme suit :

$$D = \frac{Bx_0}{2 \tan\left(\frac{\varphi_0}{2}\right)(x_L - x_D)} \quad (6)$$

A ce stade, il suffit d'avoir les coordonnées du pixel commun ainsi que la disparité pour obtenir en résultat la distance métrique séparant les caméras de l'objet.

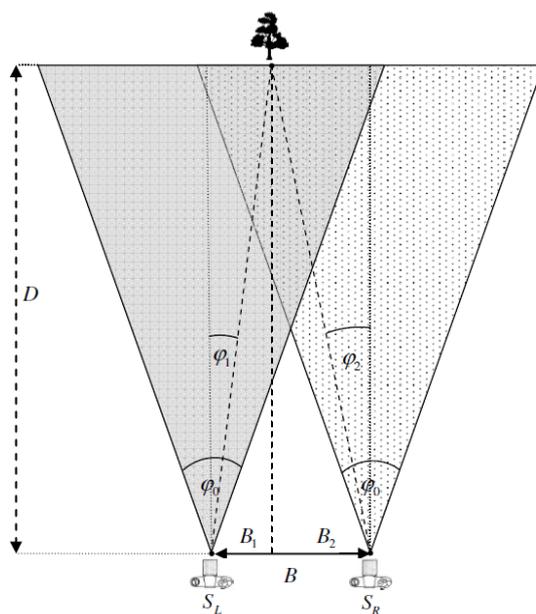


Figure 7 : Illustration d'un arbre pris par deux caméras

Avec :

$B$  : la distance entre les deux caméras

$x_0$  : le nombre de pixels horizontaux

$\varphi_0$  : l'angle de vue de la caméra

$(x_L - x_D)$  : la différence horizontale entre les mêmes pixels dans les deux images (disparité).

## 2.5 Architecture de la plateforme radar

### 2.5.1 Présentation de la plateforme

La plateforme vidéotheque permet l'enregistrement et le traitement de vidéos ou images haute résolution. Elle peut être utilisée pour plusieurs projets de traitement vidéo tel que le radar mono caméra et radar stéréoscopique qui sont les deux premiers projets testés sur cette plateforme par MAScIR.

### 2.5.2 Vue globale

La plateforme peut être modélisée par le système décrit par la figure ci-dessous :

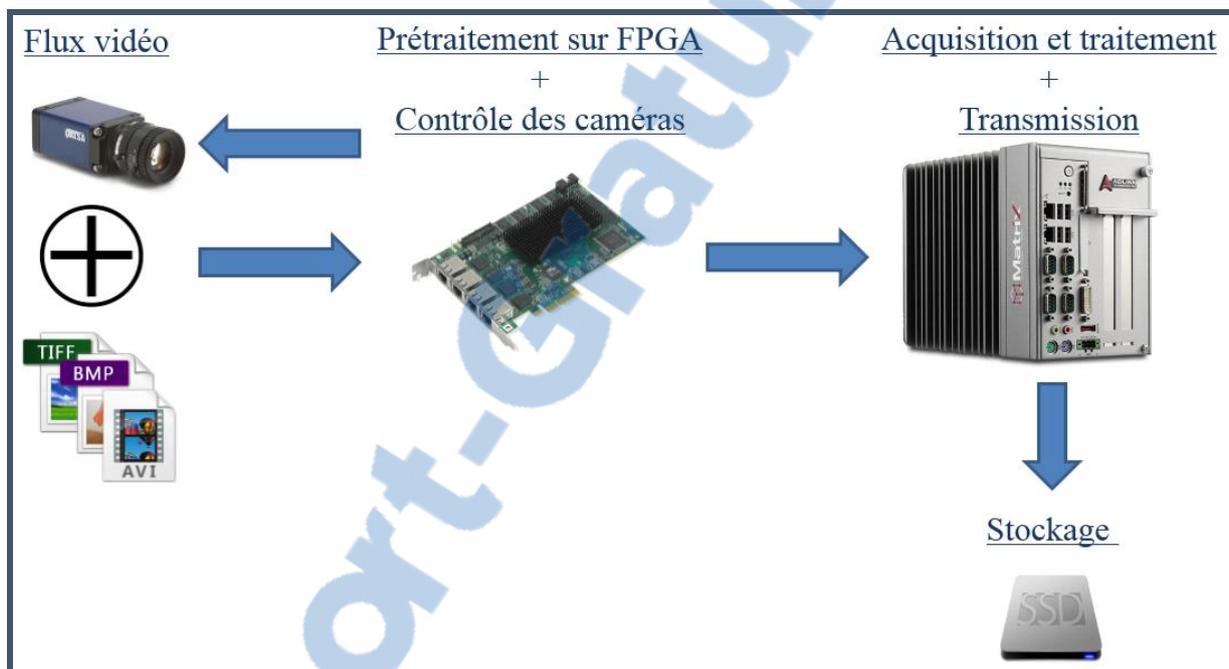


Figure 8 : Schéma de la plateforme

Le schéma de la plateforme résume les principales composantes de la plateforme, c'est un système dont l'entrée est un flux vidéo qui est soit reçu à partir du fichier image (BMP, TIFF) ou vidéo (AVI) ou d'une caméra IP qui donne en sortie des trames qui suivent le protocole GivE Vision, et la sortie du système est l'image après traitement stockée sur le disque dur.

### 2.5.3 Description du matériel

La plateforme est composée d'un certain nombre d'équipement qui doit être mis en place à l'endroit désigné pour les enregistrements.

- 2 Caméras Dalsa HC-1400
- PC MXC-6000

- FPGA MicroEnable IV
- SISO ME IV-TRIGGER I/O

La liste complète des composants est disponible en annexe A.

## 2.6 Conclusion

Dans ce chapitre nous avons présenté une définition générale de la stéréovision et les différentes étapes de la vision stéréoscopique. Nous avons aussi présenté les méthodes les plus répandues pour la mise en correspondance des pixels et finalement la formule permettant le calcul de distance entre le plan de la caméra et celui de l'objet. Ensuite nous avons donné une brève présentation de la plateforme du projet Radar.

Pour le calcul de la corrélation entre les images, nous avons opté pour la méthode locale se basant sur une cross-corrélation normalisée, car elle est simple et rapide ce qui nous permettra de gagner en terme de temps de calcul qui est un facteur très important pour un système fonctionnant en temps réel.

# Chapitre 3

---

## Architecture globale de l'algorithme

*Dans ce chapitre nous allons présenter l'architecture globale de l'algorithme et une description de chaque phase du processus de traitement.*

## 3 Architecture de l'algorithme

### 3.1 Architecture globale

L'algorithme d'estimation de vitesse par stéréoscopie se base sur le traitement des images stéréo prises par deux caméras, la figure ci-dessous présente les différents blocs de l'algorithme globale :

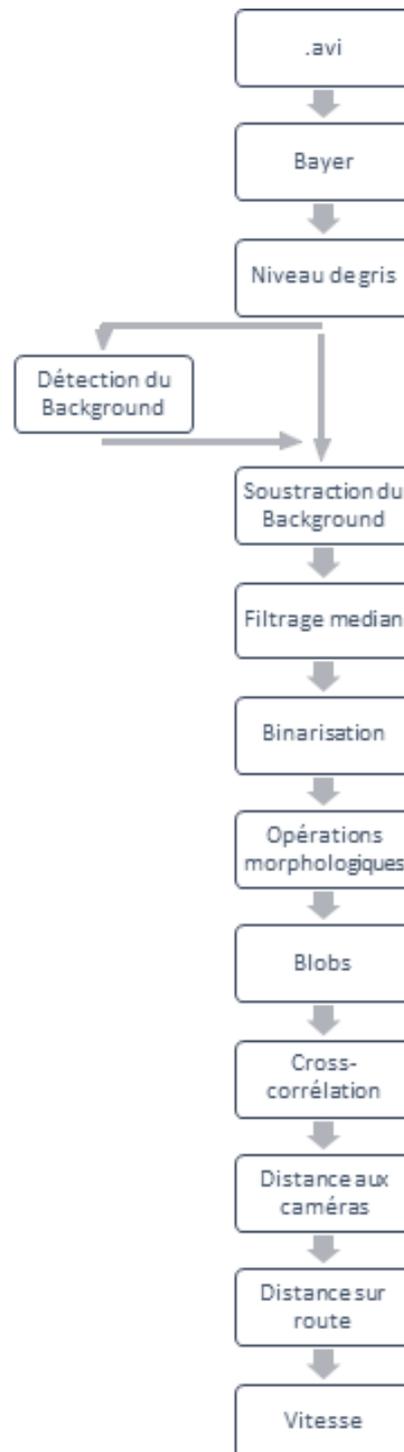


Figure 9 : Architecture de l'algorithme

A partir des vidéos prises par les caméras, nous procédons par une extraction des images en fonction du nombre de trames par seconde (FPS) avec lequel ces vidéos ont été enregistrées. Les images obtenues sont en format Bayer, on les convertis en niveaux de gris et on effectue en parallèle une détection du Background puis une soustraction de ce dernier. Ensuite on procède au filtrage afin d'éliminer le bruit résultant de l'opération précédente. Une fois l'image filtrée, on choisit un seuil de binarisation puis vient l'étape des opérations morphologiques sur l'image. Après cela on applique notre algorithme qui permet d'extraire les Blobs.

Pour calculer la disparité l'étape suivante est le calcul de la cross-corrélation entre la partie de l'image comprise dans le Blob et une sous partie de l'autre image (image droite). Une fois que nous avons obtenu la valeur de la disparité, on calcule la distance du plan de l'objet au plan parallèle au système et on en déduit la distance sur route en prenant en compte l'inclinaison des caméras. Finalement nous calculons la vitesse qui est la différence de deux distances calculées multipliée par le nombre de FPS et divisée par le nombre d'images sur lesquels nous avons travaillé.

## 3.2 Description

### 3.2.1 Images Bayer

Une matrice de Bayer, également appelé filtre de Bayer ou encore mosaïque de Bayer, est un type de Color filter array (en), c'est-à-dire une matrice de filtres de couleur placée devant un capteur photographique numérique afin de pouvoir enregistrer des photographies en couleur.

Une matrice de Bayer est constituée à 50 % de filtres verts, à 25 % de filtres rouges et à 25 % de filtres bleus.

Après la conversion des vidéos prises, nous obtenons des images en format Bayer qui seront converties par la suite en niveau de gris.

Les figures ci-dessous présentent un exemple d'images stéréo en format Bayer et en niveaux de gris

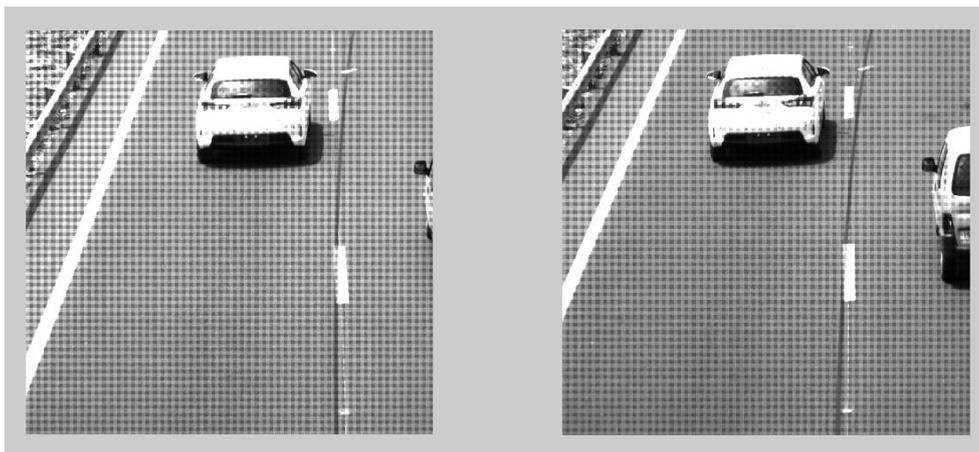


Figure 10 : Images Bayer

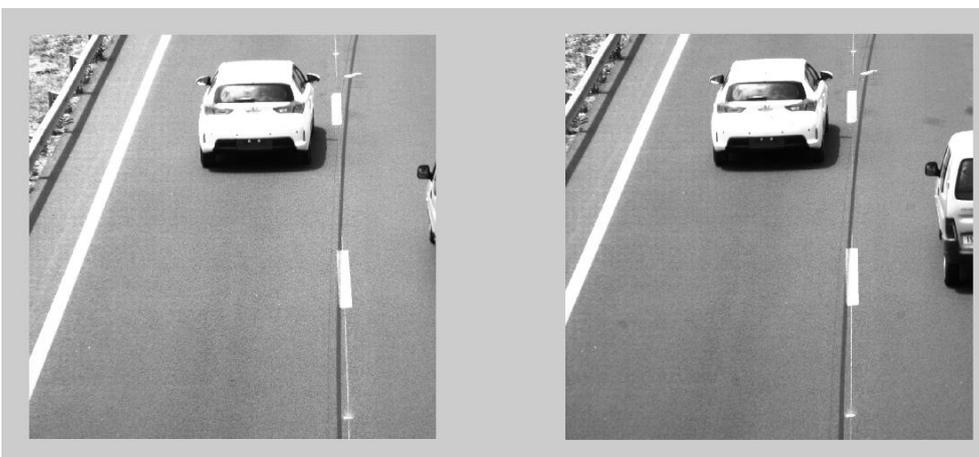
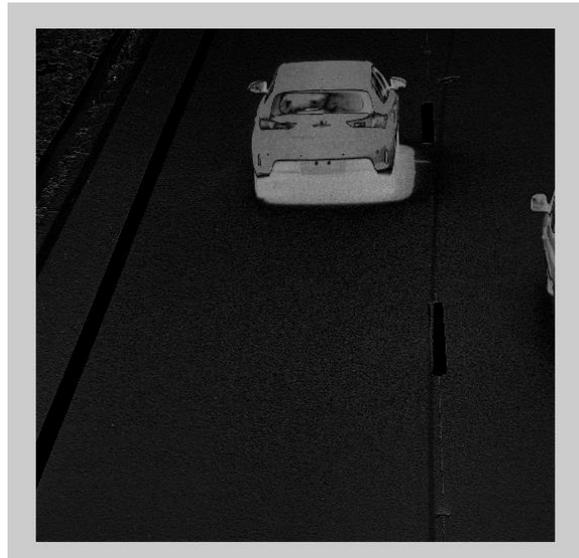


Figure 11 : Images en niveau de gris

### 3.2.2 Détection et soustraction du Background :

Pour éliminer les informations inutiles et qui peuvent alourdir le temps de calcul, on procède par une détection du Background afin de le soustraire par la suite et ne garder que les informations utiles, à savoir la position et les dimensions de l'image du véhicule en question.

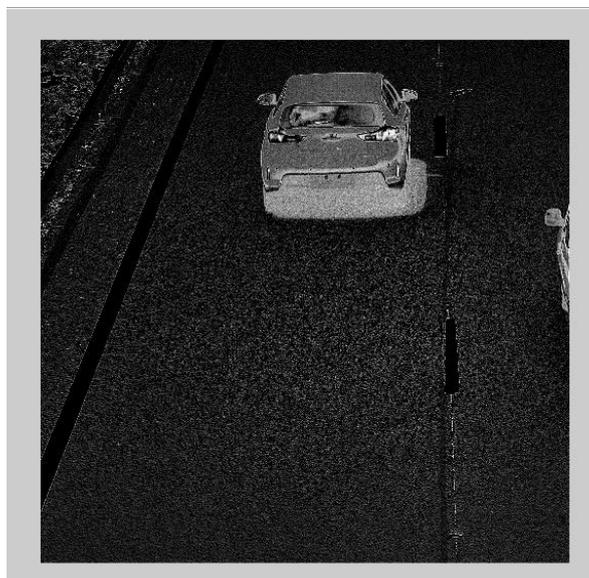
La figure ci-dessous présente l'image obtenue après soustraction du Background :



**Figure 12 : Image obtenue après soustraction du Background**

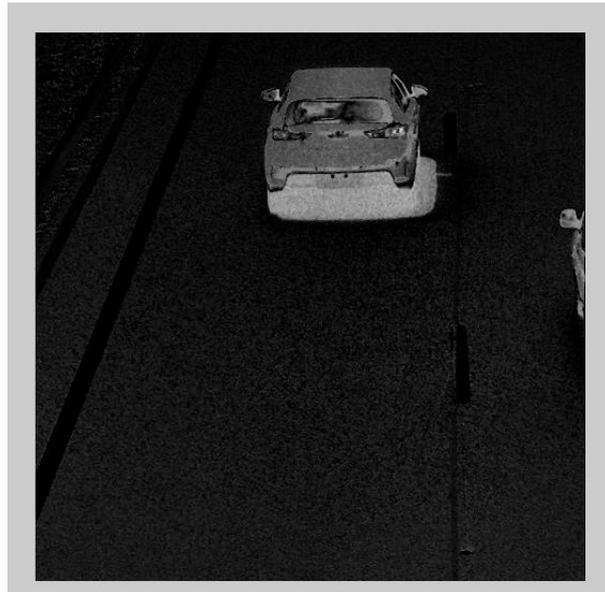
### 3.2.3 Filtrage

Après soustraction du Background, on applique un masque avec la fonction 'unsharp' de Matlab et un filtre médian afin d'éliminer le bruit restant sur l'image et résultant de l'opération de soustraction.



**Figure 13 : Image après application du filtre unsharp**

La figure ci-dessus présente l'image après application du masque.



**Figure 14 : Image obtenue après application du filtre median**

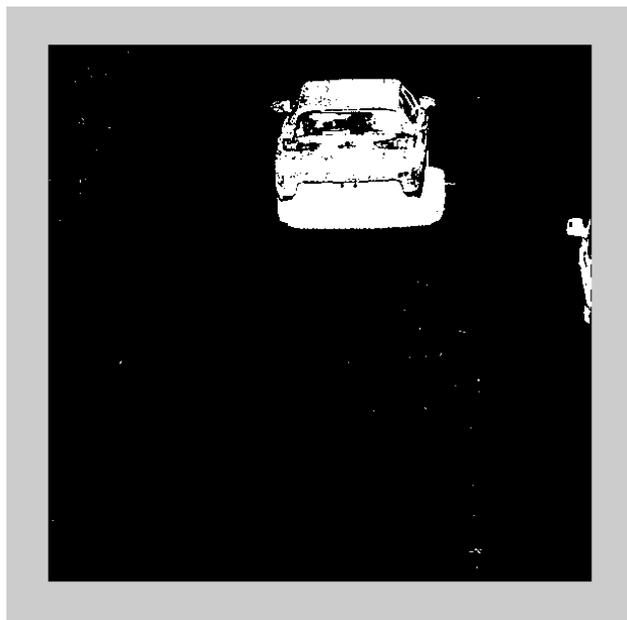
Après filtrage on obtient le résultat présenté dans la figure ci-dessus.

### 3.2.4 Binarisation

Après filtrage, vient l'étape de la binarisation. Dans cette étape, on commence par estimer l'indice de binarisation convenable afin d'éliminer le bruit restant de la partie filtrage et ne garder que les véhicules. Cette binarisation est donnée par l'algorithme suivant :

```
pour i variant de 1 à hauteur_image faire  
    pour j variant de 1 à largeur_image faire  
        si image(i,j) < seuil alors imageb(i,j) ← 0  
        sinon imageb(i,j) ← 1  
    fin pour  
fin pour
```

La figure ci-dessous montre le résultat de l'algorithme de binarisation :

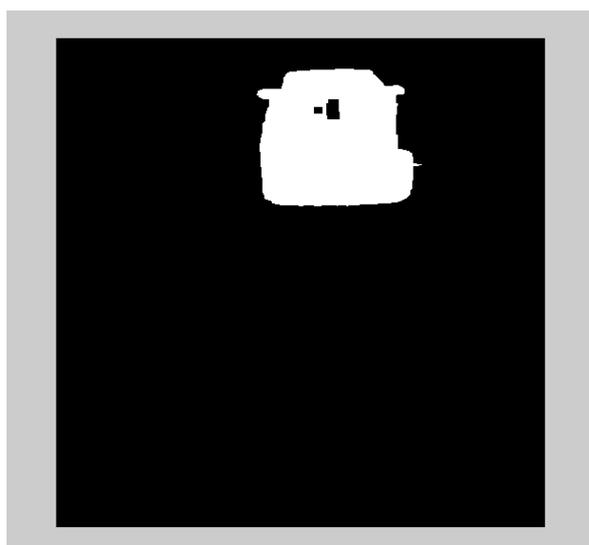


**Figure 15 : Image obtenue après Binarisation**

### 3.2.5 Opérations morphologiques

Une fois que l'image est binaire, nous appliquons des opérations morphologiques (dilatation, érosion, ouverture, fermeture) nécessaires pour ne garder que les véhicules et d'éliminer le bruit restant.

L'image obtenue après application des opérations morphologiques nécessaires est présentée ci-dessous :



**Figure 16 : Image après application des opérations morphologiques**

Nous appliquons ensuite la fonction 'imfill' de Matlab pour remplir les trous de tout objet connexe dans l'image, le résultat obtenu est donné sur la figure 17.

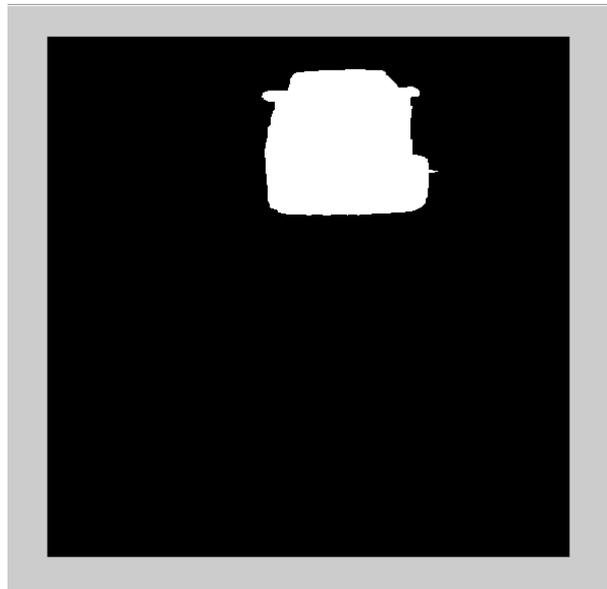


Figure 17 : Image obtenue après remplissage des trous

### 3.2.6 Détection de Blobs

C'est un algorithme qui permet d'avoir en sortie les coordonnées sur l'image du véhicule (position sur l'axe des x, position sur l'axe des y, largeur et hauteur de l'image) afin d'en extraire une partie qui sera va corrélée avec une partie plus grande de l'autre image stéréo.

### 3.2.7 Calcul de la cross-corrélation

Afin de définir la valeur de la disparité, nous procédons par le calcul de la cross-corrélation entre des sous-parties des images stéréo. Il en résulte un maximum de corrélation dont on prendra la position des coordonnées horizontales sur les deux images pour déduire la disparité. Cette partie et celle qui viendront par la suite seront détaillées dans le chapitre 4.

### 3.2.8 Calcul de la distance de l'objet au système de vision stéréoscopique

Après le calcul de la disparité, on applique la formule déjà mentionnée dans le chapitre 2 (équation (6)) pour calculer cette distance.

### 3.2.9 Calcul de la distance sur la route

Une fois la distance D entre le système et les caméras est calculée, on utilise les formules trigonométriques nécessaires pour calculer la distance sur route (Droute) selon la formule :

$$D_{\text{route}} = \frac{D}{\cos(\alpha)} \quad (7)$$

Avec :  $\alpha$  est l'inclinaison des caméras.

### 3.2.10 Calcul de la vitesse

Une fois la distance sur route est calculée, on calcul la pente des distances calculées en fonction des frames des images, on multiplie le résultat fois 3.6 pour obtenir la vitesse en km/h et on multiplie fois le nombre de FPS utilisé lors de l'enregistrement des vidéos.

## 3.3 Travail réalisé

La partie la plus importante et la plus critique de l'algorithme d'estimation de vitesse par stéréoscopie est la mise en correspondance de pixels, qui permet de calculer la disparité afin d'estimer la distance et par conséquent la vitesse du véhicule. Pour cela nous avons commencé par développer l'algorithme permettant de calculer la disparité entre les deux images stéréo, puis nous avons appliqué la formule d'estimation de distance entre le véhicule et le système et la distance sur route pour déduire finalement la vitesse.



Figure 18 : Algorithme de mesure de vitesse par stéréoscopie

## 3.4 Conclusion

Dans ce chapitre, nous avons présenté l'architecture globale de l'algorithme, en expliquant chaque phase et les résultats obtenue. Nous allons présenter dans un quatrième chapitre les résultats obtenus après application de l'algorithme, les problèmes rencontrés et les solutions proposées pour l'amélioration de cet algorithme.

# Chapitre 4

---

## Tests et résultats

*Dans ce chapitre nous allons présenter les résultats obtenus après le développement de l'algorithme ainsi que les différents problèmes rencontrés et les solutions proposées pour l'amélioration de ce dernier.*

## 4 Tests et résultats

### 4.1 Mode d'enregistrement

L'enregistrement des vidéos qui ont fait l'objet des tests de l'algorithme développé ont été enregistrés avec la plateforme qui comporte le matériel déjà cité en chapitre 2.

#### 4.1.1 Configuration

Avant de commencer l'enregistrement, nous configurons les paramètres des caméras, à savoir :

- La résolution de l'image,
- Le temps d'exposition,
- Le nombre de FPS,
- L'angle d'inclinaison des caméras.
- L'écart entre les caméras

#### 4.1.2 Calibration

L'étape la plus importante et la plus critique qui précède chaque enregistrement est celle de la calibration, car pour avoir de bon résultats d'estimation de la disparité et par la suite d'estimation de vitesse, il faut que les axes des caméras soient parfaitement parallèles. Aussi pour avoir les pixels qui permettent le calcul de la disparité sur la même droite épipolaire, il faut que les deux caméras soient dans le même plan horizontal.

#### 4.1.3 Les vitesses enregistrées

Après les étapes de configuration et calibration, nous filmons le passage d'un véhicule dont la vitesse varie pour chaque enregistrement dans le but de pouvoir comparer les vitesses du véhicule avec celles obtenues après application de l'algorithme.

La vitesse exacte avec laquelle roulait le véhicule est obtenue par GPS car ça nous permet d'avoir une erreur réduite de moins de 1%.

## 4.2 Réalisations

### 4.2.1 Algorithme pour calcul de la disparité

La première phase du travail est de développer un algorithme de calcul de disparité, pour ce faire, nous avons songé à utiliser un algorithme de mise en correspondance se basant sur une

méthode locale et une cross-corrélation normalisée. Nous avons utilisé pour ceci la fonction 'normxcorr2' de Matlab.

L'algorithme utilisé pour la fonction 'normxcorr2' est comme suit : [10]

- Calcul de la cross-corrélation dans le domaine temporel ou fréquentiel, ceci dépend de la taille de l'image.
- Calcul des sommes locales.
- Utiliser les sommes locales pour normaliser la cross-corrélation pour avoir les coefficients de corrélation.

La formule utilisée est :

$$\gamma(u,v) = \frac{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\left\{ \sum_{x,y} [f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2 \right\}^{0.5}} \quad (8)$$

Avec :

$f$  est l'image;

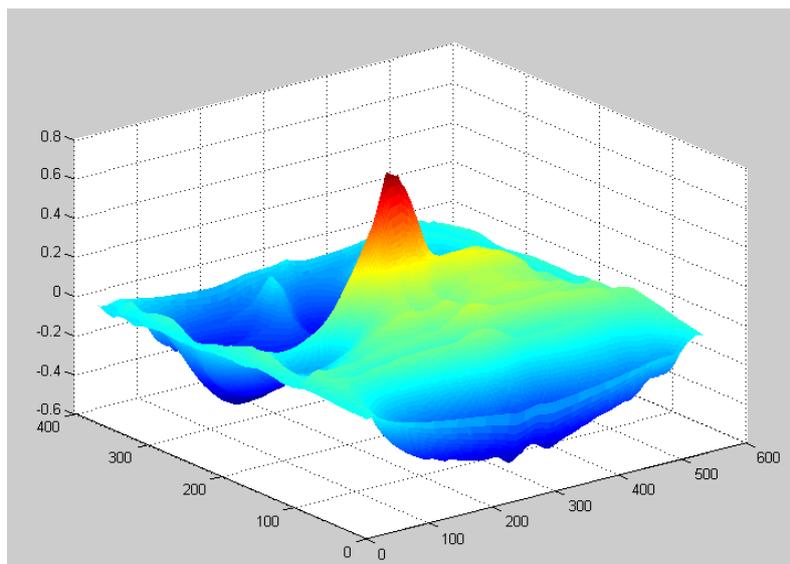
$\bar{t}$  est la moyenne du Template;

$\bar{f}_{u,v}$  est la moyenne de  $f(x,y)$  dans la région sous le Template.

(Template est l'image avec laquelle on va corrélérer est dont les dimensions sont plus grandes que l'image initiale).

Cette fonction nous délivre à sa sortie une courbe 3D qui nous permet d'en tirer les coordonnées du point dont correspond l'objet recherché dans la deuxième image. Nous prélevons de cette courbe le maximum de cross-corrélation.

La figure ci-dessous montre un exemple de courbe de corrélation entre des sous-parties de deux images stéréo :



**Figure 19 : Courbe 3D résultante d'une cross-corrélation avec la fonction 'normxcorr2'**

L'axe des x représente la somme des pixels horizontaux des deux images corrélées, l'axe des y représente la somme des pixels verticaux des images corrélées et l'axe des z représente la valeur de la cross-corrélation et qui est comprise entre -1 et 1.

Une fois qu'on extrait le maximum de cette fonction, on cherche dans l'axe des x la valeur du pixel où se trouve l'objet dans les deux images, on soustrait les deux valeurs pour avoir la valeur de la disparité recherchée.

### **4.2.2 Calcul de la distance sur route**

Pour calculer la distance sur route nous avons commencé par calculer la distance entre le véhicule et le système stéréoscopique. La formule utilise la distance focale de l'objectif des caméras, l'ouverture, la taille du pixel sur l'image, la largeur de l'image (en pixels), la distance qui leur sépare et la disparité déjà calculée.

Pour pouvoir calculer la distance sur route, il faut prendre en considération l'inclinaison du plan horizontal des deux caméras par rapport au plan horizontal de la route.

### **4.2.3 Calcul de la vitesse**

Pour calculer la vitesse obtenue on calcule la distance sur route obtenue dans chaque image et la pente de la courbe obtenue. On multiplie cette pente par le nombre de FPS fixé lors de la phase de configuration des caméras et multiplier par 3,6 pour avoir un résultat en km/h selon la formule :

$$V = P \times 3,6 \times nFPS \quad (9)$$

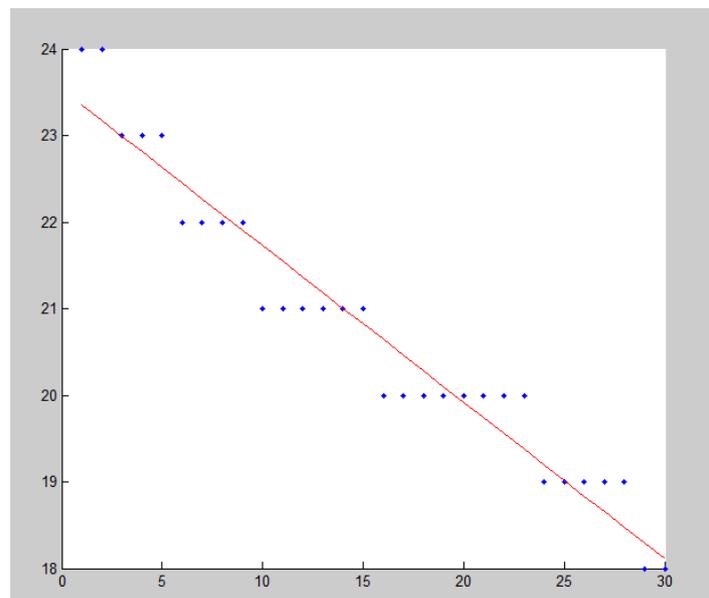
Avec :

P : La pente.

nFPS : Le nombre de FPS utilisé pour l'enregistrement.

### 4.3 Problèmes rencontrés

Après le développement de l'algorithme de calcul de disparité sous Matlab, les résultats obtenus sont reportés sur la figure ci-après :

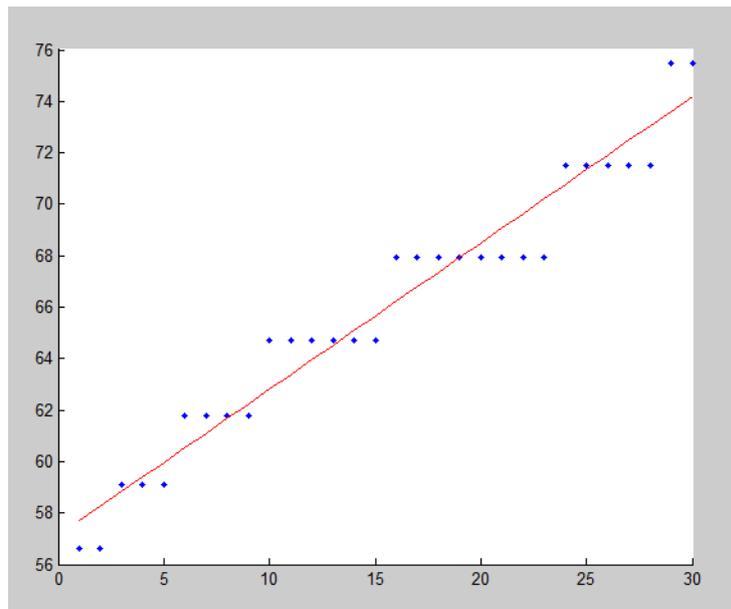


**Figure 20 : Courbe des disparités calculées pour 30 images**

La courbe ci-dessus présente la valeur de la disparité calculée en fonction du nombre d'images traitées.

On remarque que les valeurs obtenues sont discrètes car la valeur de la disparité est entière ce qui donne moins de précision lors des estimations de vitesse, d'où la nécessité de rendre ces valeurs sous forme d'un nombre réel afin d'atteindre une bonne précision.

Ceci entraîne le même effet sur les distances :



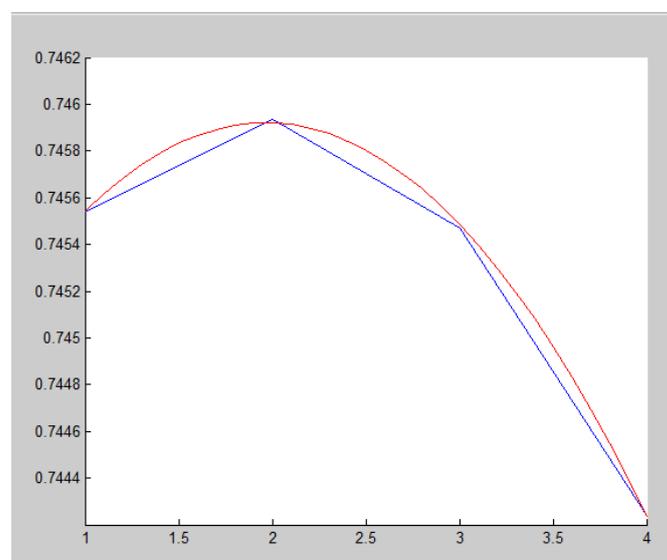
**Figure 21 : Courbe des distances calculées pour 30 images**

La courbe ci-dessus présente la valeur de la distance en fonction du nombre d'images traitées.

## 4.4 Solutions

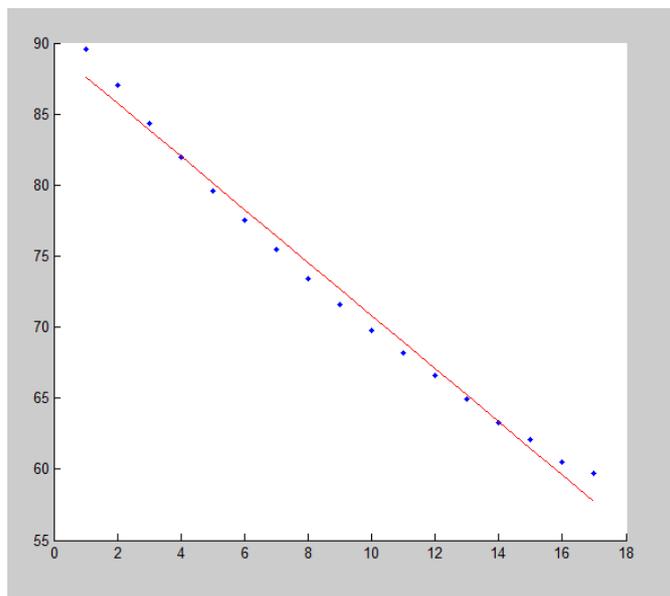
Pour remédier au problème déjà cité, nous proposons d'utiliser une approximation polynomiale avec un polynôme du deuxième ordre avec trois points autour du maximum puis interpoler ce polynôme afin d'avoir la position exacte du maximum.

La figure ci-après présente la valeur de disparité sur un intervalle de quatre points et le polynôme de deuxième degré.



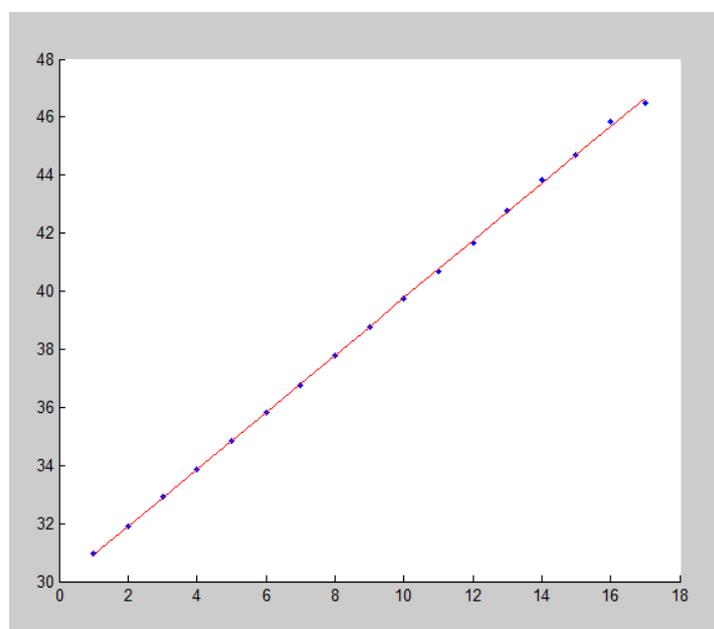
**Figure 22 : Une coupe de la courbe du maximum (en bleu) et l'approximation polynomiale de cette dernière (en rouge)**

La valeur de disparité en fonction du nombre d'images est présentée dans la courbe ci-dessous :



**Figure 23 : Les valeurs des disparités en utilisant une approximation polynômiale**

Ci-dessous la figure présentant les distances en fonction du nombre d'image traitées :



**Figure 24 : Courbe des distances en utilisant une approximation polynomiale**

En calculant la pente de la droite en rouge, nous pourrions calculer la vitesse avec laquelle roulait le véhicule.

## 4.5 Résultats obtenus

Après l'amélioration de l'algorithme, nous l'avons testé sur des vidéos prises de deux véhicules. Les résultats obtenus sont regroupés dans le tableau suivant :

|                          | Véhicule 1 (km/h) | Véhicule 2 (km/h) |
|--------------------------|-------------------|-------------------|
| <b>Vitesse_1 par GPS</b> | 105               | 96                |
| <b>Vitesse_1 obtenue</b> | 105,6026          | 96,4609           |
| <b>Vitesse_2 par GPS</b> | 118               | 113               |
| <b>Vitesse_2 obtenue</b> | 118,636           | 113,3523          |

**Tableau 1 : Table des résultats**

Le tableau ci-dessus présente les vitesses enregistrées par GPS, les estimations de vitesses obtenues en appliquant l'algorithme développé et le pourcentage d'erreur pour chaque estimation.

Les résultats obtenus sont satisfaisants, ceci est grâce à une bonne calibration du système stéréoscopique d'une part, car si les axes des caméras ne sont pas parallèles, on aura une mauvaise estimation de distance ce qui entraîne une estimation de vitesse erronée, et d'une autre part grâce à l'approximation polynômiale qui nous a permis d'augmenter la précision du calcul. Ainsi pour augmenter d'avantage cette précision, nous pourrions soit :

- Augmenter l'écart entre les caméras,
- Augmenter l'ouverture du capteur et par conséquent réduire la distance focale,
- Augmenter la largeur de l'image.

## 4.6 Conclusion

Dans ce chapitre nous avons présenté les étapes du mode d'enregistrement, de développement de l'algorithme et de calcul de vitesse. Nous avons aussi soulevé les problèmes rencontrés puis nous avons proposé des solutions proposées et quelques recommandations pour augmenter la précision de notre système.

## Conclusion générale

Dans ce rapport, nous avons présenté l'algorithme de calcul de vitesse par stéréoscopie, ce dernier se divise en trois étapes :

- La mise en correspondance et le calcul de disparité,
- Le calcul de la distance au système stéréoscopique,
- Le calcul de la distance sur route et le calcul de la vitesse.

L'étude bibliographique que nous avons menée nous a permis de choisir la méthode qui nous servira à réaliser l'étape de la mise en correspondance des pixels qui est l'étape la plus importante de l'algorithme nous permettant d'avoir la valeur de la disparité. Puisqu'on ne se focalise que sur le véhicule, nous avons choisi d'utiliser la méthode locale car elle est plus simple et nécessite un temps de calcul réduit. Nous avons ensuite calculé la distance séparant le véhicule et le système stéréoscopique pour pouvoir en déduire la distance sur route et par la suite la vitesse du véhicule.

La valeur entière de la disparité a impliqué une imprécision de l'estimation de vitesse, nous avons songé à l'approximer par un polynôme de deuxième degré pour avoir une valeur exacte de la disparité.

Des solutions possibles pourront être utilisées afin de rehausser la précision de l'algorithme et qui consistent à augmenter l'écart entre les caméras, l'ouverture du capteur pour réduire la distance focale et à augmenter la largeur de l'image.

Enfin, une perspective plus ambitieuse serait d'améliorer l'algorithme et de l'adapter aux différents milieux et différentes conditions dont lesquels le Radar pourrait fonctionner.

Pendant le déroulement de mon stage, j'ai eu l'opportunité de mettre en évidence mes compétences acquises en traitement d'images, de me familiariser avec la plateforme Radar et d'avoir une idée sur le déroulement des projets dans une entité qui travaille sur la recherche et développement. Sur le plan relationnel, cela m'a permis de travailler au sein d'une équipe et de voir de près le rôle d'un ingénieur R&D.

## Bibliographies

- [1] Blog at WordPress.com, *Correlation based similarity measures-Summary*, 2010, disponible sur : <http://siddhantahuja.wordpress.com/2010/04/11/correlation-based-similarity-measures-summary/>.
- [2] Sylvie Chambon, « Mise en correspondance stéréoscopique d'images couleur en présence d'occultations », *THESE présentée et soutenue publiquement le 14 décembre 2005 pour l'obtention du Doctorat de l'Université Paul Sabatier Toulouse III spécialité Informatique*, 2005.
- [3] Sylvie Chambon, Alain Crouzil, « Mise en correspondance par corrélation avec prise en compte des occultations », *Institut de Recherche en Informatique de Toulouse, IRIT, Université Paul Sabatier, UPS*, 2006.
- [4] Mihail Georgiev, Atanas Gotchev, Miska Hannuksela, « A FAST AND ACCURATE RE-CALIBRATION TECHNIQUE FOR MISALIGNED STEREO CAMERAS », *Tampere University of Technology, Tampere, Finland; Nokia Research Center, Tampere, Finland*, 2013.
- [5] Sébastien LEFEBVRE, « Approche monodimensionnelle de la mise en correspondance stéréoscopique par corrélation – Application à la détection d'obstacles routiers », *THESE présentée et soutenue publiquement le 3 juillet 2008 pour l'obtention du grade de Docteur de l'Université des Sciences et Technologies de Lille en Automatique, Génie informatique, Traitement du Signal et des Images*, 2008.
- [6] Julien Morat, « Vision stéréoscopique par ordinateur pour la détection et le suivi de cibles pour une application automobile », *THÈSE pour obtenir le grade de DOCTEUR DE L'INPG Spécialité : Imagerie, Vision et Robotique préparée au laboratoire LJK dans le cadre de l'École Doctorale : Mathématiques, Sciences et Technologies de l'Information, Informatique*, 2008.
- [7] Jernej Mrovlje and Damir Vrancic, « Distance measuring based on stereoscopic pictures », *9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, 2008.
- [8] Lionel SOMMELLIER, « Mise en correspondance d'images stéréoscopiques utilisant un modèle topologique », *THESE présentée devant L'UNIVERSITÉ CLAUDE BERNARD - LYON 1 pour l'obtention du DIPLOME DE DOCTORAT*, 1997.
- [9] Edwin Tjandranegara, « Distance Estimation Algorithm for Stereo Pair Images », *ECE Technical Reports Electrical and Computer Engineering*, 2005.
- [10] The MathWorks, Inc, *normxcrr2*, 2014, [En ligne], Disponible sur: <http://www.mathworks.com/help/images/ref/normxcrr2.html#bqkhhom-1,18/06/2014>.

## Annexe A

# Liste du matériel

## PC MXC-6000



**Figure 25: PC MXC-6000**

Fiche technique:

- Processeur Intel Core i7 cadencé à 2,0 GHz et un jeu de composants QM57
- Plus de puissance pour des applications de vision industrielle
- Possibilité d'utiliser des cartes PCI / PCIe standards
- Maximum de flexibilité grâce aux slots PCI configurables
- 16 entrées et sorties numériques opto-isolées
- Sortie DVI-I sur le panneau frontal permet la connexion directe d'un écran LCD
- Port eSATA
- Port interne USB 2.0 sur la carte d'adaptation de connexion garantit la protection fiable dans les applications de sécurité

Marchés & applications:

- Automatisation de machines
- Tâches de mesure dans l'imagerie et la vision industrielle

- Contrôle de la production
- Systèmes de test et de mesure
- Sécurité des réseaux

## Caméra Dalsa HC-1400



**Figure 26: Dalsa HC-1400**

Le Génie HC1400 offre une acquisition d'image à grande vitesse avec un ratio signal-bruit sans compromis, optimisant considérablement le rendu. Technologie de capteur exclusif de Teledyne DALSA permet des taux de 75 images par seconde en 1400 x 1024 de capture d'image tout en conservant une qualité d'image exceptionnelle. L'interface Ethernet gigabit GenieCam HC1400 tire pleinement parti de ce GenieCam à fréquence élevée des images par la transmission des données sur CAT-5e et CAT-norme 6 câbles à des distances allant jusqu'à 100 m. Comme tous les GenieCams, le HC1400 est basé sur l'AIA (Automated Imaging Association) GigE Vision standard pour relier directement l'appareil photo à un PC.

Fiche technique:

- Capteur CMOS d'un pouce avec global shutter
- 1400 x 1024 pixels
- Pixels carrée de 7.40  $\mu\text{m}$  x 7.40  $\mu\text{m}$
- Taille de capteur active 10.4 mm x 7.6 mm
- Débit d'images 60 images/s en pleine résolution
- Balayage partiel pour augmenter le taux d'images
- Temps d'exposition 10  $\mu\text{s}$  jusqu'à 4 s
- S/N ratio 48 dB
- Sortie numérique 8 bit via interface GigE Vision
- Alimentation de 12 VDC à 24 VDC
- Consommation d'électricité jusqu'à 4 W
- Connecteurs RJ-45 et Hirose, 12-pin

- Dimensions 29 x 44 x 67 (H x L x l mm)
- Poids ca. 125 g
- Optique Monture C
- Pilote CVB disponible
- incl. licence gratuite pour CVB 2011 CameraSuite MultiOS

## FPGA MicroEnable IV



**Figure 27: MicroEnable IV**

La famille IV microEnable de Silicon Software comprend des cartes d'acquisition d'images numériques dotées d'une interface PCI express et programmables via des mini-applications (Visual Applets). De la configuration Base CameraLink à la configuration FULL 10 Tap, il est possible d'y raccorder aussi bien des caméras couleur que monochromes à capteur matriciel ou linéaire.

Les cartes sont équipées d'interfaces PCIe x1 (single lane) à PCIe x4 (quad lanes). L'extension de la mémoire sur la carte d'acquisition s'étend de 128 MB à 512 MB DDR-RAM, qui garantissent une très faible latence et, en connexion avec l'interface PCI, un très haut débit de transmission de données.

La famille de produits microEnable IV peut être élargie de diverses manières. Ainsi, outre les cartes de déclenchement de base TTL ou bien à découplage optique avec huit entrées et huit sorties, la carte d'extension Pixelplant existe sous deux variantes dotées de processeurs FPGA supplémentaires et d'une mémoire de 256 ou 512 Mo (entrées/sorties numériques) est disponible en option.

Ces cartes d'acquisition puisent leur originalité dans la programmation libre des FPGA sur les cartes, via des mini-applications logicielles. De cette manière, il est possible de programmer simplement des mini-applications matérielles spéciales (programmes pour le FPGA) dans un diagramme de flux de données et de les charger lors de l'initialisation de la carte d'acquisition. Outre la capture d'image, de telles mini-applications peuvent contenir aussi des fonctions supplémentaires, telles que le prétraitement et l'amélioration de l'image. La grande originalité de cette famille réside dans le fait que ces fonctions sont directement traitées dans le matériel de la carte d'acquisition et sont exécutées en temps réel sans charger l'unité centrale.

## SISO ME IV-TRIGGER I/O



**Figure 28: SISO ME IV-Trigger I/O**

SISO ME IV-Trigger I/O fait partie des accessoires de la carte MicroEnable IV elle sert à donner des signaux trigger pour commander l'acquisition des images prise par la caméra et pour la synchronisation dans le cas de plusieurs caméras.

Fiche technique:

- 8 entrées - 8 sorties
- Connexion avec connecteur femelle 2\*D15
- Câble plat