



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Automatique

**Option : Automatique Avancée**

**Réf:.....**

**Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:**

**MASTER**

*Thème*

**Robot Mobile avec Suivi Visuel de ligne**

**Présenté par :**

-Belhouchet Khaled

-Khammar Kamelia

**Soutenu le : 05 Juin 2013**

**Devant le jury composé de :**

**Mr Messaoudi Abdelhamid**

**MAA**

**Président**

**Mr Boumehraz Mohamed**

**MAA**

**Encadreur**

**Mr Benelmir Okba**

**MCB**

**Examineur**

**Année universitaire: 2012 / 2013**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Automatique  
Option : Automatique Avancée

Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:

**MASTER**

*Thème*

**Robot Mobile avec Suivi Visuel de ligne**

**Présenté par :**

-Belhouchet Khaled  
-Khammar Kamelia

**Avis favorable de l'encadreur :**

- Boumehraz Mohamed *signature*

**Avis favorable du Président du Jury**

Messaoudi Abdelhamid

*Signature*

**Cachet et signature**



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Automatique  
Option : Automatique Avancée

## *Thème :*

### **Robot Mobile avec Suivi Visuel de ligne**

**Proposé par :** Belhouchet Khaled et Khammar Kamelia

**Dirigé par :** Boumehraz Mohamed

#### **RESUMES (Français et Arabe)**

Notre projet robot mobile avec suivi visuel de ligne, est de type uni-cycle composé par un châssis et 3 roues : deux roues motrices polarisées sur même axe et une roue libre pour assurer l'équilibre du robot. Pour commander les deux moteurs CC une carte ArduinoUno a été utilisée.

Le suivi de la ligne est assuré par un système à base d'un laptop et une webcam permettant l'acquisition en temps réel de l'image du terrain. Cet image utilisée pour la détection de la ligne et le calcul des commandes envoyées à la carte de commande des moteurs pour assurer le suivi de la ligne.

مشروعنا هو عبارة عن روبوت متحرك يتتبع بالبصرية المسار. يتكون الروبوت من اطار و 3عجلات عجلتين للقيادة متمركزتين على نفس المحور وعجلة حرة لتحقيق توازن الروبوت. لتحكم في المحركين من نوع تيار كهربائي مستمر استخدمنا بطاقة اردوينو اينو. ولضمان تتبع المسار استعملنا كمبيوتر محمول و كاميرا ويب تقوم بنقل مباشر لصورة الحقل . هذه الصورة

## Le mémoire contient par ordre d'apparition:

- Dédicaces
- Remerciements
- Liste des Tableaux
- Liste des Figures
- Liste des abréviations
- Résumés (bilingue)

### Sommaire :

- \* Introduction générale
- \* Chapitre 1, 2, 3...  
(Chaque chapitre commence par une Introduction et se termine par une conclusion)
- \* Conclusion générale et perspectives
- \* Bibliographie
- \* Annexe

### Type d'écriture:

- Times new Roman 12
- Les titres en Gras
- Interligne 1.5
- Formules et Figures sont numérotées par chapitre :  
exp. Fig. n°Chap. n°figure (Fig. I.1) ou (Equ. I.1)

---

### Remarques :

- Eviter le glaçage de la page de garde.
- Respecter le contenu par ordre d'apparition.
- Ecrire les résumés bilingues dans une seule page...

- عدم تجليد الصفحة الأولى
- 
- كتابة ملخص للمذكرة باللغتين  
(العربية والفرنسية)

...

# Dédicaces

Je dédie ce modeste travail :

A mes parents

A mes chères sœurs

A tout mes amis

**Khammar kamelia.**

## Remerciements

*Nous tenons* premier lieu remercier Dieu tout puissant de m'avoir accordé la force et le courage de mener ce travail à terme.

*Nous tenons* à adresser mes sincères remerciements à mon encadreur de thèse le Dr .Mohamed Boumehraz ,Professeur à l'Université Mohamed Kheidar de Biskra pour sa disponibilité, pour sa lecture , suggestion et remarques et surtout pour sa confiance sans limite mise en moi tout au long de ce projet de recherche. Nous vous prions de bien vouloir agréer le témoignage de ma plus vive reconnaissance et mon profond respect.

Nous remercions particulièrement ma famille pour leur soutien moral tout au long de ce travail, merci de m'avoir encouragé, et cru en moi.

Nous remercions également mes amis mes collègues de travail, sans oublier mes collègues du laboratoire qui m'ont encouragé tout au long de ce projet et m'ont beaucoup aidé.

Enfin, nous remercions les membres de jury qui m'ont fait l'honneur d'accepter de juger ce modeste travail.

## Liste des tableaux

<b>Tableau.III.2</b> : <i>Table de vérité de pont en H.....</i>	44
<b>Tableau.III.4</b> : <i>Les principales caractéristiques de l'Arduino Uno.....</i>	46

# Liste des figures

## **CHAPITRE I : Généralités sur les robots mobiles.**

<b>Figure. I.1 :</b> Représentation d'un robot mobile.....	02
<b>Figure. I.2 :</b> Exemple de plate-forme différentielle.....	04
<b>Figure. I.3 :</b> Urban Robot de la société iRobot .....	04
<b>Figure. I.4 :</b> Exemple de plate-forme omnidirectionnelle.....	04
<b>Figure. I.5 :</b> Exemple de plate-forme non holonome.....	05
<b>Figure. I.6:</b> Robot mobile: (a) MIABOT, (b) Pioneer P3-DX.....	06
<b>Figure. I.7 :</b> Robot mobile de type voiture.....	07
<b>Figure. I.8 :</b> Robucab.....	08
<b>Figure. I.9 :</b> Le robot martien Sojourner.....	10
<b>Figure. I.10 :</b> Le robot sous marin TAIPAN .....	10
<b>Figure. I.11 :</b> Drone de surveillance RESSAC de l'ONERA.....	10
<b>Figure. I.12 :</b> Localisation à l'estime.....	10
<b>Figure. I.13 :</b> Localisation absolue (méthode par triangulation).....	12
<b>Figure. I.14 :</b> Courbe B-spline.....	14
<b>Figure. I.15 :</b> Courbe de Bézier cubique.....	14
<b>Figure. I.16 :</b> Graphe de visibilité (à gauche) et diagramme de Voronoi (à droite).....	15
<b>Figure. I.17 :</b> Poursuite de trajectoires admissibles.....	19
<b>Figure. I.18 :</b> Genghis, développé par Rodney Brooks au MIT au début des années 1990.....	20
<b>Figure. I.19 :</b> Exemples de robots utilisés dans différentes applications.....	20

## **CHAPITRE II : Traitement d'image numérique-Suivi visuel.**

<b>Figure. II.1 :</b> Développement de la science traitement d'image depuis 1950 à 2005.....	26
<b>Figure. II.2 :</b> Le niveau de gris du pixel.....	27
<b>Figure. II.3 :</b> Image couleur.....	28
<b>Figure. II.4 :</b> Pixellisation.....	29
<b>Figure. II.5 :</b> Effets de la quantification à l'acquisition.....	29
<b>Figure. II.6 :</b> Les images sont prises dans différent de bruits.....	30

<b>Figure. II.7 :</b> <i>Comparaison entre un éclairage non uniforme et correction de l'éclairage.....</i>	30
<b>Figure. II.8 :</b> <i>Flou de bougé/filé dû à un temps de pose/intégration trop long.....</i>	30
<b>Figure. II.9 :</b> <i>Effet de lignage dû au balayage entrelacé des caméras vidéo.....</i>	31
<b>Figure. II.10 :</b> <i>Schémas présente familles d'approximation d'une droite.....</i>	31

### **CHAPITRE III : Réalisation pratique.**

<b>Figure. III.1 :</b> <i>Schéma de commande des moteurs avec le circuit L298.....</i>	42
<b>Figure. III.2 :</b> <i>Hacheur L298.....</i>	43
<b>Figure. III.3 :</b> <i>Schéma interne du L298.....</i>	43
<b>Figure. III.4 :</b> <i>Etat d'un pont en H : (a) Moteur déconnecté, (b) Rotation ou sens 1, (c) Rotation inverse ou sens inverse.....</i>	45
<b>Figure. III.5 :</b> <i>La première fenêtre concerne la webcam.....</i>	50
<b>Figure. III.6 :</b> <i>La deuxième fenêtre montre la détection de la ligne.....</i>	51
<b>Figure. III.7 :</b> <i>Schéma du principe du système de commande.....</i>	51

## Liste des abréviations

I.N.S.A : Institut National des Sciences Appliquées

L.A.A.S: Laboratoires d'analyses et d'Architecture des Systemes

GPS : Global Positionning System

EGNOS: European Geostationary Navigation Overlay System

GPSRTK : Real Time Kinematics

PMP : Planification de Mouvement Partiel

VFH : Vector Field Histogram

3 D: Dull, Dirty, Dangerous

RGB: Rouge-Vert Bleu

Bit: binary digit

HSV: Hue, Saturation, Value

YCbCr: Luminance, Chrominance bleue, Chrominance rouge

CAN : Convertisseur Analogique Numérique

CC : Courant Continue

GND : Modèle Géométrique Direct

VDD : alim ou niveau logique haut pour un circuit CMOS

VSS : alim ou niveau logique bas pour un circuit CMOS

PWM : Modulation de largeur d'impulsion

---

## Résumé

Notre projet robot mobile avec suivi visuel de ligne, est de type uni-cycle composé par un châssis et 3 roues : deux roues motrices polarisées sur même axe et une roue libre pour assurer l'équilibre du robot. Pour commander les deux moteurs CC une carte ArduinoUno a été utilisée.

Le suivi de la ligne est assuré par un système à base d'un laptop et une webcam permettant l'acquisition en temps réel de l'image du terrain. Cet image utilisée pour la détection de la ligne et le calcul des commandes envoyées à la carte de commande des moteurs pour assurer le suivi de la ligne.

**Mots clefs :** Robot Mobile-Suivi Visuel-Ligne.



مشروعنا هو عبارة عن روبوت متحرك يتتبع بالبصرية المسار. يتكون الروبوت من اطار و 3 عجلتين للقيادة متمركزتين على نفس المحور وعجلة حرة لتحقيق توازن الروبوت. لتحكم في المحركين من نوع تيار كهرباء مستمر استخدمنا بطاقة اردوينو اينو.

سار استعمالنا كمبيوتر محمول و كاميرا ويب تقوم بنقل مباشر لصورة الحقل . هذه

**الكلمات الاساسية :** - تتبع بالبصرية .

# SOMMAIRE

Introduction générale : .....	01
<b>Chapitre N°I: Généralités sur les robots mobiles.</b>	
I.1. INTRODUCTION.....	03
I.2. DEFINITION.....	03
I.2.1. Robot mobile.....	03
I.3. APERÇU HISTORIQUE.....	04
I.4. LES EFFECTEURS.....	05
I.4.1. Les plates-formes différentielles .....	05
I.4.2. Les plates-formes omnidirectionnelles .....	06
I.4.3. Les plates-formes non holonomes .....	07
I.4.4. Les plates-formes à pattes .....	07
I.5. LA MODELISATION D'UN ROBOT MOBILE.....	07
I.5.1. Le modèle cinématique du robot de type uni-cycle .....	07
I.5.2. Le modèle cinématique de type voiture .....	09
I.6. LES DIFFERENTS TYPES DE TERRAIN.....	10
I.7. LOCALISATION.....	12
I.8. PLANIFICATION DE TRAJECTOIRE.....	15
I.8.1. Approches polynomiales pour la génération de trajectoires.....	15
I.8.2. Carte routière (road map).....	16
I.8.3. Approche globale.....	17
I.8.3.1. Approche Par squelette.....	18
I.8.4. Approche locale.....	18
I.8.4.1. Méthodes des champs de potentiel.....	18
I.8.4.2. Méthode de HUANG.....	18
I.8.5. Approche mixte.....	19
I.8.6. Approches réactives.....	19
I.8.6.1. Planification de mouvement partiel.....	19
I.8.6.2. Histogramme de champs de vecteurs.....	19
I.9. EVITEMENT REACTIF D'OBSTACLES.....	20
I.9.1. Méthodes analytiques.....	20
I.9.2. Méthode de fenêtre dynamique.....	21
I.10. POURSUITE DE TRAJECTOIRE.....	21

I.11. EXEMPLES D'APPLICATIONS.....	22
I.12. LES TACTIQUE DE NAVIGATION DES ROBOTS MOBILES.....	23
I.12.1. Robots télémétriques sans mémoire .....	23
I.12.2. Optimisation de trajectoires dans un univers connu .....	24
I.12.3. Tactiques de navigation optimisées et interactives avec l'environnement.....	24
I.13. CONCLUSION.....	25
<b>Chapitre N°II : Traitement d'image numérique-Suivi visuel.</b>	
II.1. INTRODUCTION.....	25
II.2. HISTORIQUE DU TRAITEMENT D'IMAGES.....	25
II.3. VISUALISATION.....	25
II.3.1. Définition.....	26
II.3.2. Image binaire.....	26
II.3.3. Image en niveaux de gris .....	26
II.3.4. Image couleur .....	27
II.3.5. Image à valeurs réelles.....	27
II.4. ESPACE DE COULEURS.....	27
II.5. ACQUISITION: ECHANTILLONNAGE / QUANTIFICATION.....	28
II.5.1. Effets de l'échantillonnage : Pixellisation.....	28
II.5.2. Effets de la quantification à l'acquisition.....	28
II.5.3. Bruits lies à l'acquisition .....	29
II.6.RECONNAISSANCE DE FORMES.....	30
II.6.1. Définition.....	30
II.6.2. Méthode approximation d'une droite .....	30
II.6.2.1. Méthodes exactes .....	31
II.6.2.2. Approximation par estimation robuste.....	31
II.7. DOMAINES D'APPLICATION.....	32
II.8. TRAITEMENT D'IMAGE ET PERCEPTION VISUELLE.....	31
II.9. LES DIFFERENTS TYPES DE NAVIGATION.....	31
II.9.1 Guidage.....	31
II.9.2. Action associée à un lieu.....	32
II.9.3. Navigation topologique.....	32
II.9.4. Navigation métrique.....	32
II.10. CARACTÉRISTIQUE.....	32
II.11. LES REPRÉSENTATIONS DE L'ENVIRONNEMENT.....	33

II.11.1. Carte topologique.....	33
II.11.1.1. Description.....	33
II.11.1.2. Avantage.....	33
II.11.1.3. Inconvénient.....	34
II.11.1. Carte métrique.....	34
II.11.2.1. Description.....	34
II.11.2.2. Avantage.....	35
II.11.2.3. Inconvénient.....	35
II.12. DOMAINES D'APPLICATION.....	35
II.13. MODÉLISATION DU CHEMIN PAR LA FORME.....	35
II.13.1. Représentation de la forme des objets.....	36
II.13.1.1. APPROCHES GLOBALES.....	36
II.13.1.2. APPROCHES LOCALES.....	37
II.13.2. Catégorisation des chemins par shape context.....	37
II.14. MODÉLISATION ET CATÉGORISATION DE CHEMINS.....	38
II.14.1. Définition chemin.....	38
II.14.2. Indexation de chemins par la forme.....	38
II.15. CONCLUSION.....	39

## **Chapitre N°III : Réalisation pratique.**

III.1. INTRODUCTION.....	41
III.2. LA MÉCANIQUE.....	41
III.2.1. Le châssis.....	41
III.2.2. Principe de fonctionnement.....	41
III.2.3. Structure du système.....	41
III.3. L'ÉLECTRONIQUE.....	41
III.3.1. Alimentation.....	41
III.3.2. Commande des moteurs CC.....	42
III.3.2.1. Pont en H.....	43
III.3.2.2. Réalisation de module de commande des moteurs à CC.....	45
III.3.3. Détection de la ligne.....	45
III.3.4. La carte Arduino Uno.....	45
III.3.4.1. Définition.....	45
III.3.4.2 Entrées / Sorties.....	47
III.3.4.3 Communication .....	48

III.4. PROGRAMMATION.....	48
III.4.1 Python.....	48
III.4.1.1. Définition .....	48
III.4.1.2. Caractéristique du langage de programmation (python).....	49
III.4.1.3. Open CV.....	50
III.4.1.3.1. Fonctionnalités d’Open CV.....	50
III. 4.2. L’interface graphique: .....	50
III.5. CONCLUSION .....	51
Conclusion générale et perspectives.....	53
Bibliographie	
Annexe	

---

# INTRODUCTION GENERALE

**G**race à un succès qui ne se dément pas, La robotique est une technologie moderne et aujourd'hui la plupart sont automatisées à l'aide des robots. Merci à la haute technologie que la dépendance sur des êtres humains a été réduite dans une large mesure.

Un des grands avantages de l'automatisation des procédures est la précision des résultats. La robotique est utilisée dans plusieurs industries comme l'automobile, la médecine, l'électroménager et plusieurs autres. Le plus complexe des machines peuvent être assemblées à l'aide de la robotique.

Les robots peuvent être fixes (bras manipulateur) ou mobiles (robot marcheur, robot à roues) selon l'application, les robots fixes sont généralement utilisés dans les usines, les hôpitaux, domaine agricole...etc. Cependant les robots mobiles sont généralement utilisés dans les environnements dangereux : nucléaires, militaires, déminage,...etc. Ce dernier type de robot est notre domaine d'intérêt dans ce mémoire.

Le suivi d'objets en robotique mobile est devenu une des tâches les plus utilisées, Notre Projet consiste à réaliser un robot mobile suiveur de ligne de couleur sombre sur un terrain de couleurs claire.

Notre mémoire est organisé comme suit :

Le premier chapitre présente des généralités sur les robots mobiles ;

Le deuxième chapitre est divisé en deux parties :

-La première concerne le traitement d'image numérique : historique, définitions et applications.

- La deuxième traite la navigation : les différents types de la navigation.

Le troisième chapitre décrit la réalisation pratique : la partie pratique mécanique, la partie électronique et les programmes développés.

---

En fin, nous finissons notre mémoire par une conclusion qui présente le bilan de ce travail et les perspectives envisagées.

# Chapitre I

*Généralités sur les robots mobiles*

## I.1. INTRODUCTION :

Loin des images des films de science fiction, montrant hommes et robots cohabitant harmonieusement dans des environnements complexes, la robotique c'est pendant longtemps limitée à ses applications industrielles, se contentant de remplacer l'homme dans l'exécution de tâches simples et répétitives. Cette limitation venait notamment de l'impossibilité de créer des systèmes capables d'évoluer dans un environnement réel. En effet gérer l'extrême diversité des situations dans lesquels un robot peut se trouver est très difficile. Il est clairement impossible de prévoir tous les cas de figure pour préparer des comportements types en fonction des situations rencontrées. Il faut des lors que le robot soit à même de percevoir son environnement intelligemment pour apprendre et comprendre cet environnement en vue de planifier ses actions.

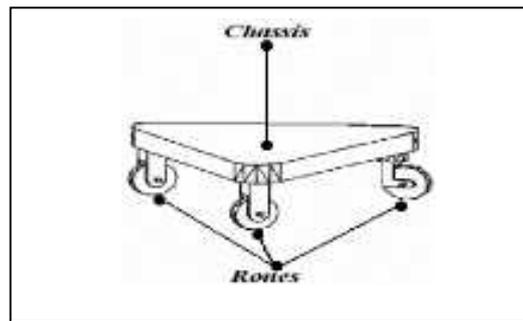
Loin d'avoir résolu totalement ce problème la robotique a cependant connu ces dernières années plusieurs victoires importantes qui ont conduit à de profonds changements. Qu'ils soient à pattes ou à roues, volant ou rampant, les robots sont désormais capables de se mouvoir dans des environnements aussi complexes que le sol martien ou un hall de musée avec très peu de supervision de la part de l'homme.

Dans ce chapitre nous avons présenté les différents types de robots mobiles, et des contraintes de terrain sur les qu'ils sont conçus pour évoluer, nous aborderons les contraintes principales liées à leur cinématique, et les solutions développées pour y remédier.

Enfin nous étudions les outils permettant aux robots de percevoir leur environnement et de s'y repérer. Etape primordiale nécessaire à l'autonomie totale des robots mobiles. Cet état de l'art offre une vision non exhaustive des thématiques de recherche associées au domaine de la robotique mobile, et présente l'ensemble des verrous scientifiques qu'il reste à lever pour aboutir au développement d'un robot autonome. Parmi ceux-ci, nous nous focalisons alors sur celui de la navigation d'un robot mobile. On en trouve ci-dessous un résumé d'analyses de quelques domaines particuliers ou Applications de la robotique.

## I.2. DEFINITION :

**I.2.1. Robot mobile :** De manière générale, on regroupe sous l'appellation *robots mobiles* l'ensemble des robots à base mobile (**Figure. I.1**).



**Figure. I.1 :** Représentation d'un robot mobile.

Ces machines sont constituées d'un châssis, et d'un ensemble de roues, ayant comme fonction la stabilité et la mobilité du système (nous n'allons nous intéresser qu'aux robots mobiles à roues, en faisant abstraction des autres types de plateformes, comme les robots marcheurs, ou rampants). La particularité de ces robots est leur capacité à se mouvoir dans des environnements relativement grands (sans influencer sur leur constitution), grâce à leur système de locomotion, c'est pour cela que nous allons dans ce qui suit évoquer les différents types de roues utilisées en robotique mobile [1].

### I.3. APERÇU HISTORIQUE :

La robotique est passée par plusieurs générations comme suit, [2] :

- ✓ 1947 : Premier manipulateur électrique télé-opéré.
- ✓ 1954 : Premier robot programmable.
- ✓ 1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une chaîne de montage de General Motors.
- ✓ 1961 : Premier robot avec contrôle en effort.
- ✓ 1963 : Utilisation de la vision pour commander un robot.
- ✓ 1978 : Le robot ARGOS .Développé à l'Université Paul Sabatier de Toulouse (France).Le robot ARGOS simule la navigation d'un robot mobile équipé d'un système de vision au fur et à mesure de ses déplacements.
- ✓ 1979 : le robot HILARE .les chercheurs de L.A.A.S. DE TOULOUSE (France) étudièrent la planification des trajectoires d'un robot mobile ponctuel, dans un environnement totalement connu.
- ✓ 1981 : Le robot VESA. Ce robot, construit à l'I.N.S.A(France).de Rennes, est équipée d'un arceau de sécurité pour réaliser la détection d'obstacles dans un environnement totalement inconnu.

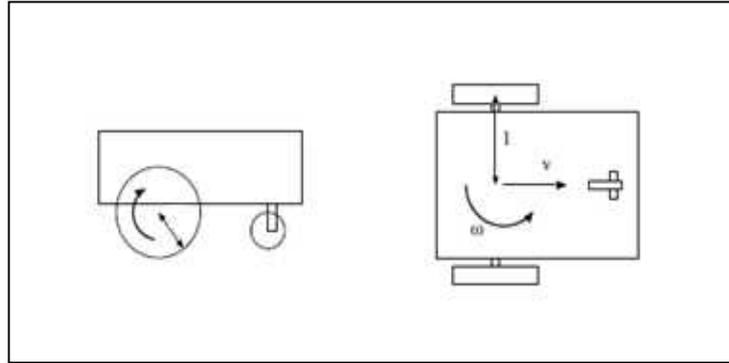
- ✓ 1984 : Le robot FLAKEY .Ce robot, conçu et construit au Stanford Research Institute et le reflet des améliorations apportées par 14 années de développement .Le robot FLAKEY est équipé de deux roues motrices avec encodeurs, mais sa vitesse maximale est de 66 cm /s au lieu de quelques centimètres par seconde .Ce robot est capable de naviguer dans des environnements réels.
- ✓ 1993 : Les robots ERRATIC et PIONNER. Le robot ERRATIC à été conçu par Kurt Konolige, au Stanford Research Institue, comme un robot mobile de faible cout pour ses cours de robotique.
- ✓ Les robots mobiles actuels : A présent la plupart des travaux de recherche portent sur les problèmes de perception. La planification de trajectoires, l'analyse et la modélisation de l'environnement de robot, appliqué sur des robots mobiles commerciaux .également la recherche actuelle sur la conception mécanique des robots mobiles pour des applications hautement spécialisées, comme l'exploration sous-marine, les robots volants et le micro robots [3].

#### I.4. LES EFFECTEURS:

Nous présentons ici les différents types de bases mobiles utilisées en robotique, en nous focalisant sur le milieu intérieur.

**I.4.1. Les plates-formes différentielles :** Une des configurations les plus utilisées pour les robots mobiles d'intérieur est la configuration différentielle, on va la utilisé dans notre projet qui comporte deux roues commandées indépendamment. Une ou plusieurs roues folles sont ajoutées à l'avant ou à l'arrière du robot pour assurer sa stabilité (**Figure. I.2**).

Cette plate-forme est très simple à commander, puisqu'il suffit de spécifier les vitesses des deux roues, et permet de plus au robot de tourner sur place. Cette possibilité permet de traiter le robot comme un robot holonome, c'est à dire un robot pouvant se déplacer dans toutes les directions depuis sa position courante (au contraire d'une voiture par exemple qui est non-holonome). Cette propriété simplifie énormément la planification de déplacement et la commande du robot.



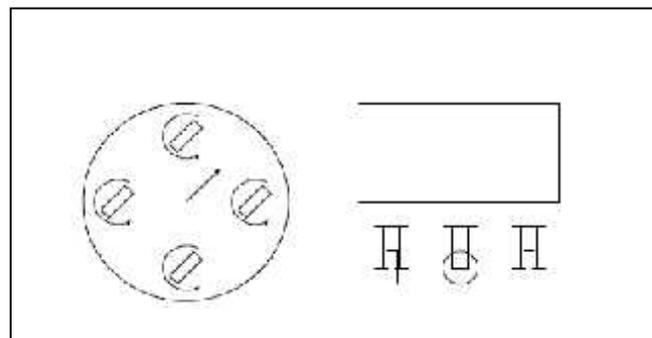
**Figure. I.2:** Exemple de plate-forme différentielle.

- Ce type de plate-forme peut également être utilisé avec des chenilles ce qui fournit une capacité de franchissement de petits obstacles intéressante (**Figure. I.3**). Ces plates-formes peuvent ainsi être utilisées en milieu urbain, ou dans des décombres. L'utilisation de chenilles conduit cependant à une odométrie très bruitée à cause du contact mal défini entre les chenilles et le sol [1].



**Figure. I.3:** Urban Robot de la société iRobot.

#### **I.4.2. Les plates-formes omnidirectionnelles :**

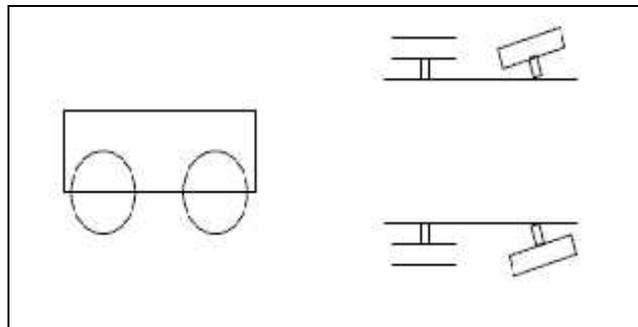


**Figure. I.4:** Exemple de plate-forme omnidirectionnelle.

Les plates-formes omnidirectionnelles permettent de découpler de manière encore plus nette le contrôle de la rotation et de la translation d'un robot et en font une plateforme réellement holonome. Elles utilisent pour cela 3 ou quatre roues qui tournent à la même vitesse pour fournir une translation et un mécanisme qui permet d'orienter simultanément ces roues dans la direction du déplacement souhaitée (**Figure. I.4**).

Le corps du robot lui-même n'effectue pas de rotation mais uniquement des translations. Ce système permet un contrôle très simple et relativement rapide car les changements de direction ne concernent que les roues et peuvent donc se faire très vite. Par contre ces plates-formes sont relativement limitées en capacité de franchissement et requièrent un sol très plan [1].

**I.4.3. Les plates-formes non holonomes :** Des plates-formes non holonomes, de type voiturent, sont également utilisées en robotique mobile (**Figure. I.5**). Ces plates-formes sont toutefois plus difficiles à commander car elles ne peuvent pas tourner sur place et doivent manœuvrer, ce qui peut être difficile dans des environnements encombrés. La commande de ces plates-formes pour réaliser un déplacement particulier est un problème à part entière [1].



**Figure. I.5 :** Exemple de plate-forme non holonome.

**I.4.4. Les plates-formes à pattes :** Des plates-formes à deux, quatre ou six pattes peuvent également être utilisées. Les plateformes à six pattes sont relativement pratiques car le robot est en équilibre permanent, ce qui facilite le contrôle. Les plates-formes à deux ou quatre pattes sont plus complexes à commander et le simple contrôle de la stabilité et d'une allure de marche correcte reste aujourd'hui difficile, ce qui les rend en général relativement lentes.

L'odométrie de ce type de plates-formes est de plus généralement de très faible qualité. Ces différents facteurs font que ces plates-formes sont rarement utilisées quand

l'application visée a un besoin précis de positionnement et de navigation. De telles plates-formes commencent cependant à apparaître à relativement grande échelle (par exemple le robot Aibo de Sony) et peuvent être utilisées en conjonction avec certaines méthodes de navigation précises [1].

## I.5. LA MODELISATION D'UN ROBOT MOBILE:

### I.5.1. Le modèle cinématique du robot de type uni-cycle :



**Figure .I.6:** Robot mobile: (a) MIABOT, (b) Pioneer P3-DX.

Le modèle d'un robot mobile uni-cycle correspond à celui d'une roue roulant dans un Plan. La configuration de l'uni-cycle est caractérisée par la donnée des nombres  $x, y, \theta$  et  $\varphi$  où  $x$  et  $y$  dénotent les coordonnées cartésiennes du point de contact de la roue avec le sol dans un repère cartésien donné,  $\theta$  l'orientation de la roue et  $\varphi$  l'angle de la roue mesurée à partir de la verticale. Sachant que le roulement est sans glissement le modèle cinématique de la roue s'exprime par les formules suivantes [4]:

$$\begin{cases} \dot{x} - \dot{\varphi} r \cos \theta = 0 \\ \dot{y} - \dot{\varphi} r \sin \theta = 0 \end{cases} \quad (\text{Equ. I. 1})$$

Où  $r$  représente le rayon de la roue.

Le modèle (1.1) peut être transformé pour faire apparaître les composantes des vitesses dans les plans de la roue et perpendiculaire à la roue, les contraintes cinématiques suivantes sont alors obtenues :

$$\begin{cases} \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \\ \dot{x} \cos \theta + \dot{y} \sin \theta = r \dot{\varphi} \end{cases} \quad (\text{Equ. I. 2})$$

Ces deux équations sont des contraintes de type non holonome, signifiant que l'on ne peut pas les intégrer de façon à ne faire apparaître que les coordonnées généralisées. Cela peut

se démontrer simplement à l'aide du théorème de Frobenius. En notant la vitesse longitudinale de l'uni-cycle avec  $v$

$v = r\dot{\theta}$  et  $\omega = \dot{\theta}$ , sa vitesse angulaire, on déduit de (1.1) le modèle cinématique du robot mobile non-holonome de type uni-cycle :

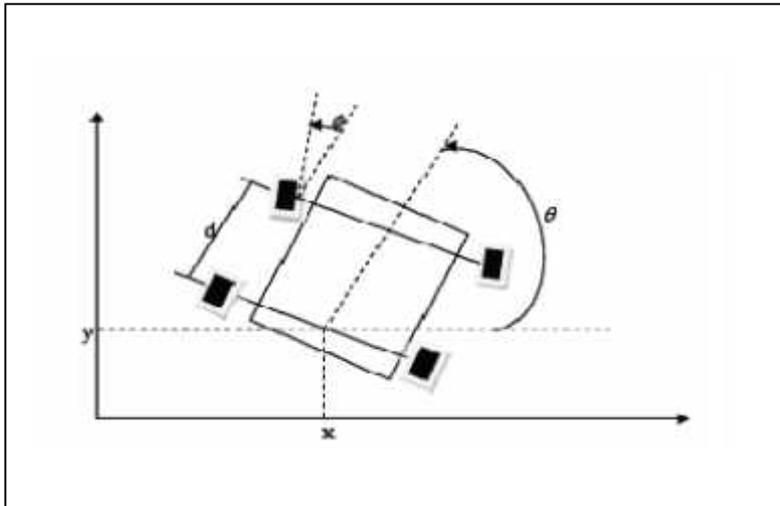
$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (\text{Equ. I. 3})$$

**I.5.2. Le modèle cinématique de type voiture :** Un robot mobile de type voiture est composé d'un train moteur à l'arrière du corps principal et de deux roues de direction et à braquage différentiel à l'avant (**Figure. I.7**).

La configuration du véhicule de type voiture peut être représentée par le quadruplet  $q = [x, y, \theta, \varphi]$ , où le point de coordonnées  $x, y$  est le centre de l'essieu arrière de la voiture,  $\theta$  est l'orientation du véhicule,  $\varphi$  l'orientation des roues avant et  $d$  la distance entre les essieux avant et arrière.

Les conditions de roulement sans glissement s'obtiennent en écrivant que les vitesses latérales des roues avant et arrière sont nulles [4]:

$$\begin{cases} \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \\ \dot{x} \sin(\theta + \alpha) - \dot{y} \cos(\theta + \alpha) - d\dot{\theta} \cos \alpha = 0 \end{cases} \quad (\text{Equ. I. 4})$$



**Figure. I.7:** Robot mobile de type voiture.

Ces contraintes sont non intégrables, le véhicule de type voiture est donc non holonome.

En utilisant des transformations sur l'équation(1.4) , le modèle cinématique d'un robot mobile non-holonome de type voiture est obtenu :

$$\begin{cases} \dot{x} = u_1 \cos \theta \\ \dot{y} = u_1 \sin \theta \\ \dot{\theta} = u_1 \frac{\tan \phi}{d} \end{cases} \quad (\text{Equ. I. 5})$$

$\phi = u_2$ , Où  $u_1$  correspond à la vitesse longitudinale du corps du robot, alors que  $u_2$  correspond à la vitesse angulaire des roues directrices par rapport au corps du véhicule.

### I.6. LES DIFFERENTS TYPES DE TERRAIN :

Nous rencontrons principalement 3 types d'espaces de navigation : les terrains plats, les Terrains accidentés et les espaces 3D. Les terrains plats sont généralement utilisés pour modéliser les milieux urbains et les intérieurs de bâtiments. Le robot évolue sur un plan 2D considéré sans pentes, et tout objet qui sort de cet espace 2D est considéré comme un obstacle (**Figure. I.8**).

Cette représentation est la plus simple à étudier et la plus répandue pour les robots mobiles aux roues. En première approche, elle permet de se concentrer sur les problèmes de contrôle et de navigation autonome du robot.



**Figure. I.8** : Robucab [10].

Les terrains accidentés (ou 2D 1/2) correspondent généralement aux milieux en extérieurs, comme des forêts, des champs en robotique agricole, ou encore des terrains rocheux (**Figure. I.9**). La différence avec les terrains plats est la présence des pentes, de bosses et de creux sur le terrain d'évolution du robot.

Cela interdit d'utiliser une métrique standard 2D et cela complique pour beaucoup la détection d'obstacles et la modélisation des déplacements du robot. De plus il devient également important de vérifier que le robot ne bascule pas quand il escalade une pente ou enjambe un obstacle. Le système de locomotion du robot doit dans ce cas être adapté à la topologie du terrain.



**Figure. I.9 :** *Le robot martien Sojourner [5].*

Enfin les espaces d'évolution 3D sont par exemple utilisés pour modéliser la navigation des robots sous-marins (**Figure. I.10**) et des drones volants (**Figure. I.11**). Les problèmes rencontrés sont spécifiques à l'application visée.

Chaque type de terrain correspond à des problématiques bien spécifiques. Le type de robot étudié dans ce mémoire est celui destiné à circuler en environnement urbain, la modélisation terrain plat sera utilisée. Cela signifie que l'on considère que tous les mouvements sont contenus dans un plan de navigation, parallèle au sol.



**Figure. I.10 :** *Le robot sous marin TAIPAN [12].*

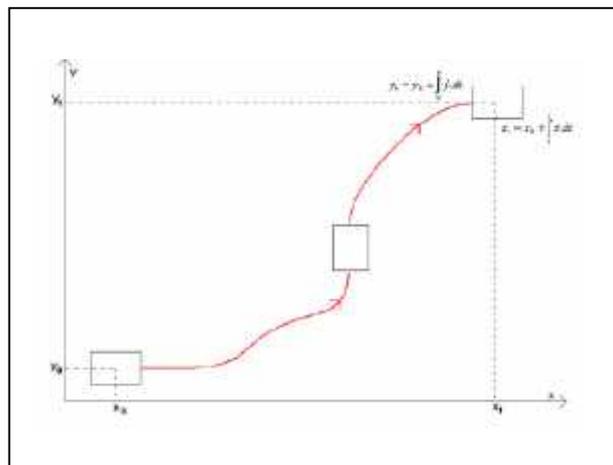


**Figure. I.11** : Drone de surveillance RESSAC de l'ONERA [8].

### I.7. LOCALISATION :

Les outils permettant la localisation d'un robot dans son environnement peuvent être classés en deux catégories : ceux par localisation à l'estime et ceux par localisation absolue [5]. Le principe de la première catégorie consiste à intégrer des informations sur les vitesses ou les accélérations fournies par des capteurs proprioceptifs (odomètres, centrales inertielles).

L'avantage de ces méthodes est qu'elles sont indépendantes de l'environnement, par contre leur souci est leur manque de précision dû à la dérive temporelle. En effet les Erreurs s'intégrant elles aussi au fur et à mesure du temps. Il est nécessaire d'apporter régulièrement des recalages (**Figure. I.12**).



**Figure. I.12** : Localisation à l'estime.

Parmi les méthodes de localisation à l'estime, le système le plus simple et le plus couramment utilisé pour la mesure de déplacement du robot est l'odométrie.

L'hypothèse de roulement sans glissement que nous avons vu précédemment, nous permet de relier directement les déplacements du robot à la vitesse de rotation des roues. Par intégration des déplacements à chaque instant, on en déduit la position relative du robot par rapport à son point de départ. L'odométrie est une méthode de localisation très courante, simple, mais également très rapidement imprécise.

En effet à cause du glissement des roues sur le sol, les erreurs s'accroissent au fur et à mesure que le robot avance, ce qui implique d'importantes erreurs sur les longs parcours s'il n'y a pas de recalage régulier.

Cette méthode est de ce fait fortement tributaire de la qualité du sol sur lequel le robot se déplace. Les incertitudes sur le diamètre exact des roues, sur les paramètres géométriques du robot, sur la résolution des codeurs, génèrent des erreurs de type systématique, qui vont s'accumuler très rapidement en odométrie.

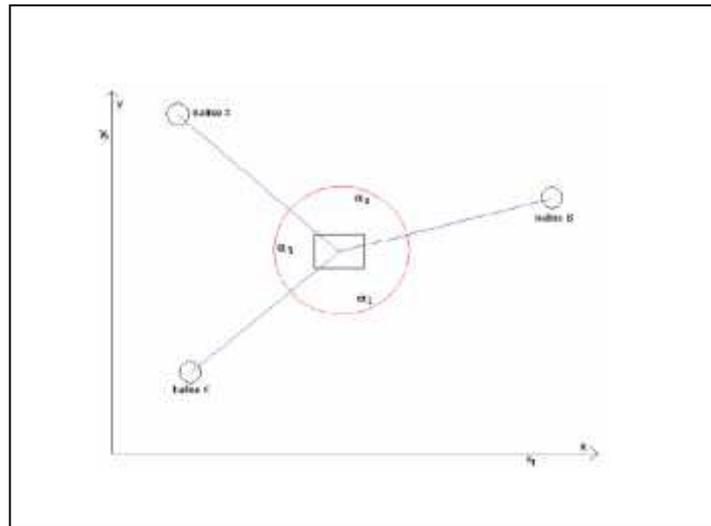
Cependant, ces erreurs peuvent être identifiées et évaluées pour faire un rééquilibrage du système et ainsi améliorer sa précision. Les erreurs non systématiques comme les glissements ou les irrégularités du sol, génèrent moins rapidement des erreurs, mais ne peuvent par contre pas être rééquilibrées puisqu'on ne peut pas les prévoir.

Pour l'exploration martienne, où le terrain est fortement accidenté, l'utilisation de Système d'odométrie classique est impossible. Pour cette application, Cheng a proposé une technique d'odométrie alternative, dite odométrie visuelle, basée sur la reconnaissance de points singuliers dans l'image vidéo du sol fournie par une caméra montée sur le robot [5] (les points de Harris).

Connaissant le positionnement de la caméra par rapport au robot, le déplacement de ces points dans l'image permet d'évaluer les vitesses de déplacement du robot, et par intégration, de retrouver sa localisation relative par rapport à sa position initiale.

Le second type de méthode pour la localisation est la localisation absolue. Ces méthodes utilisent des éléments repérables par le robot dans l'environnement de navigation, de position connue, pour permettre au robot de se repérer relativement à ceux-ci. Ces éléments sont appelés des balises ou amers et sont dits soit réels, s'ils ont été placés spécialement pour permettre la localisation, soit virtuels s'il s'agit d'éléments présents naturellement.

Les balises réelles sont dites passives si elles ont pour but de réfléchir un signal émis par un appareil de mesure du robot (laser ou infrarouge). Il existe deux méthodes pour utiliser ces balises pour la localisation du robot : la méthode télémétrique (calcul de la distance robot/balise), qui nécessite la présence de deux balises pour calculer la position du robot dans le plan ; et la méthode par triangulation, qui consiste à mesurer les angles entre chaque balise et le robot, et qui elle nécessite l'utilisation de 3 balises.



**Figure. I.13 :** Localisation absolue (méthode par triangulation).

Les balises réelles sont dites actives si elles émettent un signal captable par le robot. En Milieu extérieur, le système GPS (global positioning system) peut être utilisé pour obtenir des positions d'une précision de l'ordre du mètre.

A la base développée par l'armée américaine dans les années 80 (lancement du premier satellite GPS en 1978), il fut ouvert aux civils en 1995. Jusqu'en 2000 les mesures étaient volontairement entachées d'une erreur d'une centaine de mètres, l'armée américaine craignant que ce système soit un avantage pour leurs ennemis. Malgré le retrait de cette erreur volontaire, la précision du système restait de l'ordre du mètre, à cause des incertitudes sur l'orbite et l'horloge des satellites, ainsi que les retards engendrés par la traversée des couches atmosphériques.

Pour améliorer cette précision, nous pouvons utiliser les GPS différentiels : avec un second récepteur GPS sur une base fixe et de position connue. Il devient possible de mesurer l'erreur et d'en déduire la correction à apporter pour la zone environnante, pour que ce système fonctionne, il faut que la base mobile reste à une certaine portée de la base fixe. Cette distance varie suivant la gamme de fréquence utilisée pour l'envoi des

corrections, et peut atteindre quelques dizaines de kilomètres pour les besoins de la navigation maritime.

Des satellites géostationnaires permettent également de corriger certaines erreurs de position. Ils envoient des corrections sur les orbites et les horloges des satellites GPS.

Ainsi pour l'Europe c'est le système EGNOS (European Geostationary Navigation Overlay System) qui se charge d'envoyer ces corrections. La précision atteinte est de l'ordre de 3m. Le système de GPSRTK (real time kinematics) permet d'améliorer la précision à quelques centimètres, en utilisant la différence de phase de l'onde porteuse du signal, sa longueur d'onde étant d'une vingtaine de centimètres.

En contre partie la portée de la station de référence avec cette méthode n'est plus que de quelques kilomètres. Pour augmenter cette portée on a recours au RTK réseau, qui va utiliser la redondance d'informations pour communiquer les corrections à l'appareil, via un serveur de calcul à distance. Les systèmes de localisation GPS sont très intéressants en rase campagne ou en banlieue.

Cependant ils s'avèrent beaucoup moins efficaces en pleine ville ou en forêt. En effet pour fonctionner correctement un GPS a besoin de recevoir les informations de 4 satellites au moins.

La présence d'obstacles tels que les ponts et grands bâtiments en ville empêche parfois cette réception. Ainsi il n'est pas rare de perdre la localisation GPS pendant quelques minutes. C'est pour cela que beaucoup de recherches dans le domaine de la localisation sont aujourd'hui portées sur les méthodes alternatives au GPS en milieu urbain. La vision peut être un moyen pour compléter la localisation par GPS. Ainsi Cappella [6].

Utilise un modèle 3D de la place Stanislas à Nancy qui est confronté aux images obtenues par les caméras embarquées sur le robot pour déterminer précisément sa position. Yang propose d'utiliser les caméras du robot pour mettre au point un système d'odométrie visuelle [7].

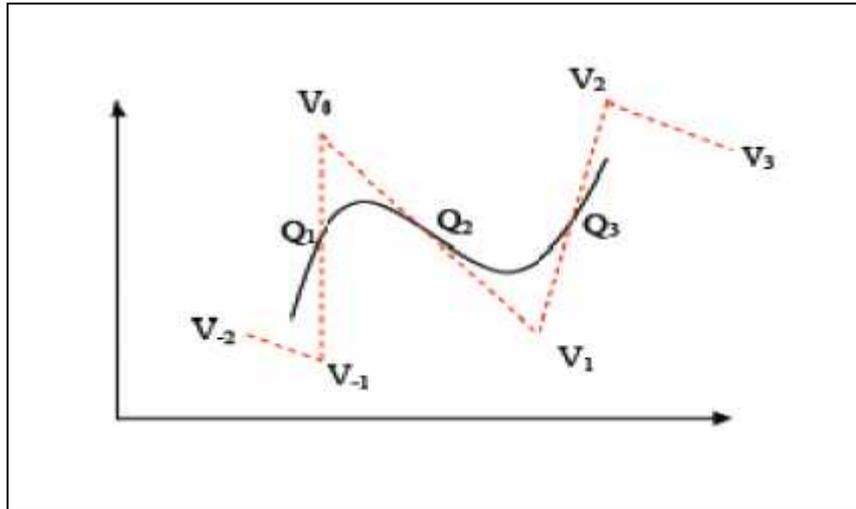
## **I.8. PLANIFICATION DE TRAJECTOIRE :**

- \* **Chemin:** Un chemin entre une situation initiale  $XI$  et une situation finale  $XF$  est une représentation géométrique du d'emplacement du corps qui ne considère pas le temps [8].

**I.8.1. Approches polynomiales pour la génération de trajectoires:** Différentes méthodologies de planification basées sur l'utilisation de trajectoires Polynomiales. Par exemple, l'utilisation de courbes de type spline ou B-spline. La trajectoire est conçue en

prenant en compte les contraintes cinématiques du robot mobile ainsi que la position des différents obstacles statiques à éviter.

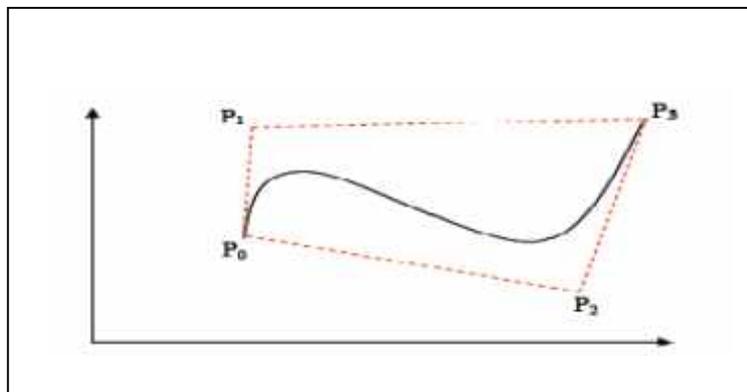
Une trajectoire générée par une courbe B-spline peut-être représentée à titre d'exemple comme sur la **Figure. I.14**.



**Figure. I.14** : Courbe B-spline.

Sur cette figure,  $Q_1$ ,  $Q_2$  et  $Q_3$  représentent les points de passage de la courbe et  $V_{-2}$ ,  $V_{-1}$ ,  $V_0$ ,  $V_1$ ,  $V_2$  et  $V_3$  sont les points qui forment le polygone de contrôle de la courbe. En pratique pour les robots mobiles de type uni-cycle ou véhicule, les B-splines cubiques s'avèrent être un choix suffisant pour définir une trajectoire vérifiant les contraintes cinématiques.

Une autre famille de trajectoires fréquemment utilisées pour la planification des robots mobiles non-holonomes est basée sur l'utilisation des courbes de Bézier. Un exemple de courbe de Bézier cubique est représenté sur la **Figure. I.15**.



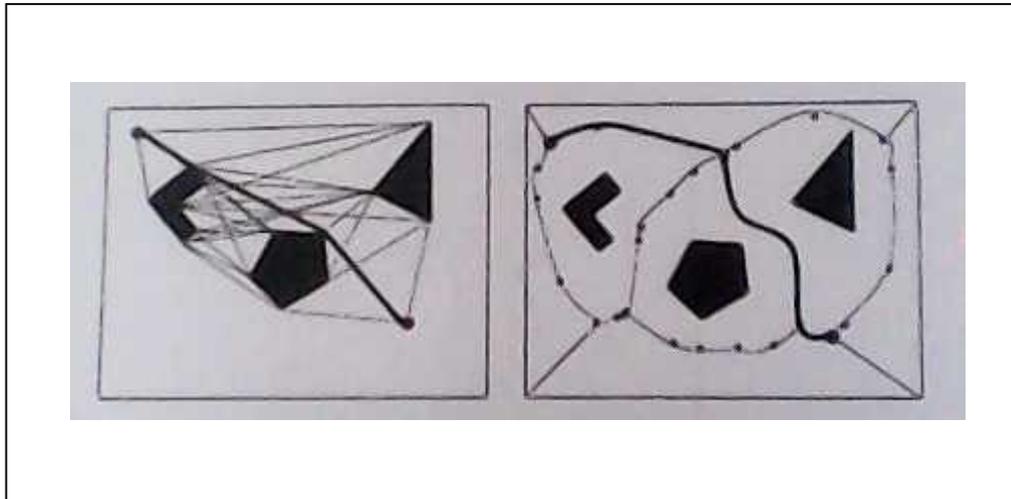
**Figure. I.15** : Courbe de Bézier cubique.

Les points  $P_0, P_1, P_2$  et  $P_3$  constituent le polygone de contrôle de la courbe [4].

### **I.8.2. Carte routière (road map) :**

Cette approche est basée sur l'identification d'un ensemble de routes dans l'espace libre. La base de cette approche est de capturer la connectivité de l'espace libre d'un robot dans un réseau de dimension 1 de courbes ou de lignes appelée carte routière. Une fois la carte routière construite, elle est utilisée comme étant un réseau de segments de routes (chemin) pour la planification des mouvements des robots.

L'objectif de cette approche est de construire un réseau routière qui rend le robot capable à visiter tous les points possible de l'environnement libre de navigation. On note qu'il y a deux approches différentes pour la construction de la carte routière. Dans le premier cas les routes sont le plus proche possible de l'obstacle, d'où l'appellation de « Graphe de visibilité » les chemins qui résultent sont de longueur minimum ; le deuxième cas est appelé « diagramme de Voronoi » les routes restent le plus loin possible de l'obstacle [9].



**Figure. I.16 :** *Graphe de visibilité (à gauche) et diagramme de Voronoi (à droite).*

**I.8.3. Approche globale:** Elle est utilisée lorsque le robot évolue dans un environnement entièrement connu. Le principe de cette approche est de capturer la connexité globale de l'espace libre dans un graphe qui est ensuite exploré. La connaissance de l'environnement nécessite sa modélisation :

### *I.8.3.1. Approche Par squelette :*

Un squelette est une concaténation de portions de courbes. Il est structuré en un graphe dont les arêtes sont ces portions de courbes et les nœuds leurs extrémités.

Il existe plusieurs types de squelettes entre autres : les graphes de visibilité et les 'freeways' :

🚦 **Graphe de visibilité** : Les nœuds du squelette sont les sommets des obstacles, tandis que les arêtes sont les segments joignant deux à deux les sommets visibles.

🚦 **Les 'freeways'** : L'espace libre est divisé en un ensemble de cônes dont les axes constituent les 'freeways' que le robot va pouvoir suivre. Les arêtes du graphe représentent les 'freeways', alors que les nœuds sont leurs extrémités.

Les méthodes se basant sur une approche globale ont l'avantage de générer un chemin optimal tandis qu'elles ont l'inconvénient de consommer beaucoup de temps de calcul et d'espace mémoire [10].

**I.8.4. Approche locale** : Contrairement à l'approche globale, cette approche ne nécessite pas une connaissance a priori de tout l'environnement, mais découvre celui ci au fur et à mesure à l'aide de systèmes de perception.

Cette approche est itérative (incrémentale). A chaque itération, on détermine l'incrément du déplacement. A titre d'exemple on peut citer deux méthodes [10] :

**I.8.4.1. Méthodes des champs de potentiel** : Le robot est assimilé à un point plongé dans un champ de potentiel artificiel. Ce champ de potentiel est défini comme la combinaison d'un potentiel qui attire le robot vers sa cible et de potentiels qui l'écartent des différents obstacles. Des améliorations de cette méthode ont vu le jour après le résultat satisfaisant qu'elle a donné.

**I.8.4.2. Méthode de HUANG** : Le principe de cette méthode est d'essayer de joindre la cible en ligne droite (reliant les positions initiale et finale). Si un obstacle intervient en cours du chemin, détecté par les ultrasons placés sur l'avant du robot, il est alors contourné. Des méthodes similaires ont été développées au paravent considérant le robot comme un point matériel et en utilisant d'autres systèmes de perception, mais ayant toujours un principe semblable.

Une adaptation de la méthode de **HUANG** permettant de prendre en compte les contraintes cinématiques du robot a été testée avec succès.

L'avantage des méthodes utilisant cette approche réside dans leurs efficacités en termes de temps de calcul qui permet leurs utilisations dans des applications temps réel. En

revanche elles présentent l'inconvénient de tomber dans des minima locaux et de générer une trajectoire souvent non optimale.

**I.8.5. Approche mixte:** L'intérêt de cette approche est de profiter des avantages des deux approches précédentes, à savoir :

- \* Temps de calcul faible pour l'approche locale.
- \* Chemin généré optimal pour l'approche globale.

L'idée de cette approche est de générer un chemin optimal, en utilisant une méthode globale, donnant l'allure générale de la trajectoire à suivre.

Ensuite, le robot exécutera cette trajectoire, à l'aide d'un générateur local de trajectoire et de son système de perception, tout en évitant de rentrer en collision avec des obstacles imprévus au moment de la modélisation (changements éventuels de l'environnement, obstacles mobiles) [10].

**I.8.6. Approches réactives:** Les approches réactives consistent à calculer à chaque pas de temps (après récupération des informations sur l'environnement fournies par les capteurs du système) le contrôle instantané à appliquer sur les actionneurs du système [11].

**I.8.6.1. Planification de mouvement partiel:** Une dernière approche réactive mérite d'être notée : il s'agit de la planification de mouvement partiel (en anglais : Partial Motion Planning – PMP utilisée. Celle-ci consiste à calculer relativement, en un temps de décision fixe, une trajectoire se rapprochant le plus possible du but. Cette méthode consiste en un algorithme à trois étapes répète à chaque pas de temps :

- \* Mise à jour du modèle de l'environnement à partir des entrées capteurs du robot.
- \* Recherche délibérative d'une trajectoire menant à l'état but. Si le but n'a pas été atteint après un temps de décision fixe, la trajectoire calculée s'en rapprochant le plus est choisie comme trajectoire à suivre.

Enfin, le mouvement planifié au pas de temps précédent est exécuté. Cette approche permet donc d'être réactive aux diverses évolutions de l'environnement tout en étant capable de sortir des impasses non détectées a priori. Elle reste sujette à des minima locaux, mais en est néanmoins bien plus robuste que les approches citées précédemment [11].

**I.8.6.2. Histogramme de champs de vecteurs:** Dans la lignée des approches par champs de potentiels, sont apparus les histogrammes par champs de vecteurs (en anglais : Vector Field Histogram - VFH). Ceux-ci, introduits par Koren et Borenstein sont nés de la combinaison des champs de potentiels et des grilles d'occupation [11] :

Un histogramme basé sur une grille cartésienne de l'environnement est construit et mis à jour au fur et à mesure de la navigation afin de reporter la présence d'obstacles à proximité du robot. Afin de choisir une direction à suivre, un histogramme polaire est construit à partir de la grille d'occupation : en discrétisant les différentes directions possibles autour du robot, l'histogramme polaire est construit en pondérant pour chaque secteur de la discrétisation polaire les cellules traversées de la grille d'occupation contenant des obstacles.

Une fois cet histogramme polaire construit, des "vallées candidates" sont d'exterminés comme les suites de secteurs contigus de l'histogramme polaire libres d'obstacles La direction à prendre par le système est alors déterminée par le milieu de la vallée menant le plus directement au but.

Initialement conçue pour la navigation de robots holonomes (pouvant naviguer dans toutes les directions), cette méthode a été étendue à plusieurs reprises afin de prendre en compte les dimensions du robot (par un espace de configuration implicite) et ses contraintes de vitesse. Plus tard, les VFH ont été combinés à une recherche A\* (VFH\*) afin de trouver un chemin menant vers le but et d'échapper ainsi aux minima locaux.

Les méthodes VFH disposent néanmoins encore de fortes limitations : Elles ne prennent en compte ni la dynamique du système robotique, ni l'éventuelle présence d'obstacles mobiles ; le mouvement instantané du robot est calculé uniquement à partir des informations sur la position courante des obstacles. Leur vitesse n'est en aucun cas considérée.

## **I.9. EVITEMENT REACTIF D'OBSTACLES:**

Dans cette partie, nous faisons un bref état de l'art des principales méthodes d'évitement réactif d'obstacles afin de mettre en lumière leurs avantages et leurs limites par rapport aux spécificités de notre problématique.

***I.9.1. Méthodes analytiques:*** Même en l'absence d'obstacles, commander un système non holonome pour l'amener d'une configuration de départ a une configuration d'arrivée n'est pas une chose aisée. En effet, il n'existe pas aujourd'hui d'algorithme général permettant de résoudre le problème pour n'importe quel système non holonome.

Des méthodes analytiques sont connues seulement pour certaines classes de systèmes. Pour les autres, on ne dispose que de méthodes numériques. En outre, la présence d'obstacles rend les méthodes analytiques inapplicables à des systèmes non holonomes [9].

**1.9.2. Méthode de fenêtre dynamique :** Cette technique proposée dans travaille dans l'espace des commandes du robot. La taille du domaine de recherche des vitesses accessibles (c'est-à-dire n'entraînant pas de collisions) est réduite par la prise en compte explicite du modèle cinématique du système. Les commandes envoyées au robot sont le résultat de la maximisation sur ce domaine de recherche d'une fonction cout liée à la configuration finale.

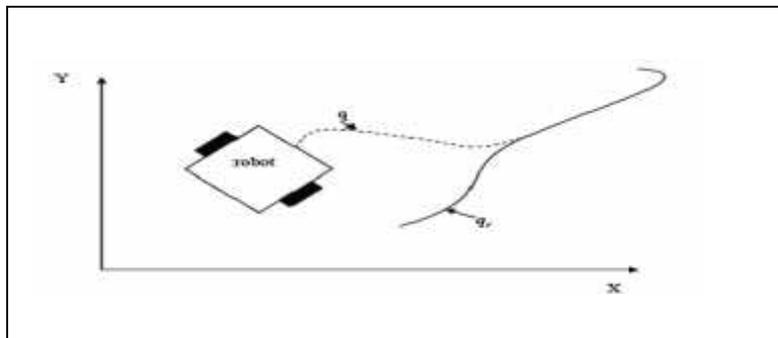
L'utilisation de cette méthode est très intéressante pour un robot se déplaçant rapidement ou pour un robot ayant des capacités d'accélération et de décélération limitées. Elle permet alors de produire un déplacement du robot sur et régulier. Son extension au cadre multi robots est cependant très délicate du fait de son manque de flexibilité [4].

### I.10. POURSUITE DE TRAJECTOIRE:

Une fois la trajectoire du robot déterminée dans le contexte de l'évitement d'obstacle ou non, il faut pouvoir la suivre. Le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonyme est apparu comme un problème de premier ordre pour la communauté roboticienne dans ces dernières années.

En effet, la forte utilisation des robots mobiles dans les domaines où l'être humain ne peut pas être présent, notamment dans les sites nucléaires à haut risque ou dans le cas de l'exploration spatiale, nécessite la mise en œuvre de lois de commande autonomes et performantes pour assurer les tâches assignées aux robots. Plusieurs travaux concernant la poursuite de trajectoire ont été développés dans ce contexte. Après une rapide formalisation de ce problème, une liste non exhaustive des différentes commandes disponibles dans la littérature est présentée [4].

 **Formulation du problème :** Une propriété des modèles de robot mobile non-holonyme (de type voiture ou uni-cycle) est qu'ils ne sont pas stables par un retour d'état continu et stationnaire. Ce corollaire est le résultat des travaux de [4].



**Figure. I.17 :** Poursuite de trajectoires admissibles.

### I.11. EXEMPLES D'APPLICATIONS :

Aujourd'hui, le marché commercial de la robotique mobile est toujours relativement restreint, mais il existe de nombreuses perspectives de développement qui en feront probablement un domaine important dans le futur.

Les applications des robots peuvent se trouver dans de nombreuses activités "ennuyeuses, salissantes ou dangereuses" (3 D's en anglais pour Dull, Dirty, Dangereuse), mais également pour des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées. Parmi les domaines concernés, citons [12] :

- ✚ La robotique de service (hôpital, bureaux).
- ✚ La robotique de loisir (aibo, robot 'compagnon').
- ✚ La robotique industrielle ou agricole (entrepôts, récolte de productions agricoles, mines).
- ✚ La robotique en environnement dangereux (spatial, industriel, militaire).



**Figure. I.18** : *Genghis*, développé par Rodney Brooks au MIT au début des années 1990.

A cela, s'ajoute à l'heure actuelle des nombreuses plates-formes conçues essentiellement

Pour les laboratoires de recherche.

La **Figure. I.19** montre quelques exemples de robot réels



**Figure. I.19:** Exemples de robots utilisés dans différentes applications.

## I.12. LES TACTIQUE DE NAVIGATION DES ROBOTS MOBILES :

On peut classer ces tactiques en trois groupes, suivant que le robot ne connaît rien de son univers de déplacement (robots télémétriques sans mémoires), qu'il le connaît totalement (univers connu) ou qu'il le découvre au fur et à mesure de son déplacement (navigation optimisée et interactive avec l'environnement) [8].

**I.12.1. Robots télémétriques sans mémoire :** Il s'agit de robots qui cherchent à se déplacer dans un univers inconnu pour atteindre un but, en évitant les obstacles à l'aide d'un télémètre à courte ou à longue portée (tactile, ultrasonore ou infrarouge). Pour ce faire, ils sont munis des primitives suivantes :

- ✚ Aller en ligne droite sur une longueur donnée.
- ✚ Tourner d'un angle donné.
- ✚ Viser au télémètre dans une direction donnée.

Certains de ces robots peuvent se repérer par rapport au but, à la fois en direction et en distance. Comme ils n'ont pas la mémoire de l'environnement, il n'est pas possible de planifier au départ leur chemin, ni leur donner des informations sur cet environnement.

Néanmoins ils possèdent quelques mots de mémoires qui leur permettent de discriminer l'enchaînement des tâches à accomplir.

**I.12.2. Optimisation de trajectoires dans un univers connu :** Dans ce cas, l'univers, connu parfaitement, est modélisé soit par une grille, soit par des polygones. La modélisation par grille a été utilisée dans les premiers travaux de recherche d'un chemin optimal car cette représentation se prête bien à une programmation directe. Le robot occupant une case ou étant modélisés par des cases hachurées.

L'univers est ainsi représenté par une matrice, appelée matrice d'adjacence, dont les éléments sont 0 ou 1 suivant que la case correspondante est libre ou occupée par un obstacle. Le déplacement du robot n'est en générale permis que suivant quatre directions perpendiculaires (horizontales et verticale ou huit directions (à 45°), suivant une direction quelconque.

Dans le modèle des polygones, les obstacles sont modélisés par des polygones dont il suffit de connaître la position des sommets pour que l'environnement soit parfaitement défini. Dans cette représentation, qui requiert moins de mémoire que la précédente, le chemin est constitué de segments joignant différents sommets de polygones. Ce chemin sera optimal lorsque la somme des coûts des arcs constituant le chemin dans le graphe (dont les nœuds sont les sommets des polygones) sera minimale.

**I.12.3. Tactiques de navigation optimisées et interactives avec l'environnement :**

Ces tactiques s'appliquent aux cas réels fréquents où l'environnement du robot n'est pas entièrement connu, soit à cause des limites de son organe de vision, soit parce que des obstacles ont pu être ajoutés à l'environnement initial mémorisé, soit encore par suite de l'imprécision dans la localisation du robot.

Il en résulte une combinaison des tactiques décrites précédemment. Au départ étant donné la connaissance de l'environnement dont dispose le robot, une trajectoire optimale est planifiée et le robot commence à le suivre. En cas de détection imprévue d'un obstacle,

le robot met en action son opérateur de vision pour réactualiser son modèle d'environnement, puis la procédure précédente est répétée (optimisation, suivi de la trajectoire, vision).

**I.13. CONCLUSION :**

La perception d'une part et la commande au sens large de l'autre part sont donc les deux thèmes majeurs de recherche pour obtenir un robot mobile parfaitement autonome. Parmi les problématiques liées à la commande, celle de la navigation qui tient un rôle important : elle consiste à déterminer les trajectoires que le robot sera capable de suivre pour lui permettre d'évoluer correctement au milieu d'obstacles.

# Chapitre II

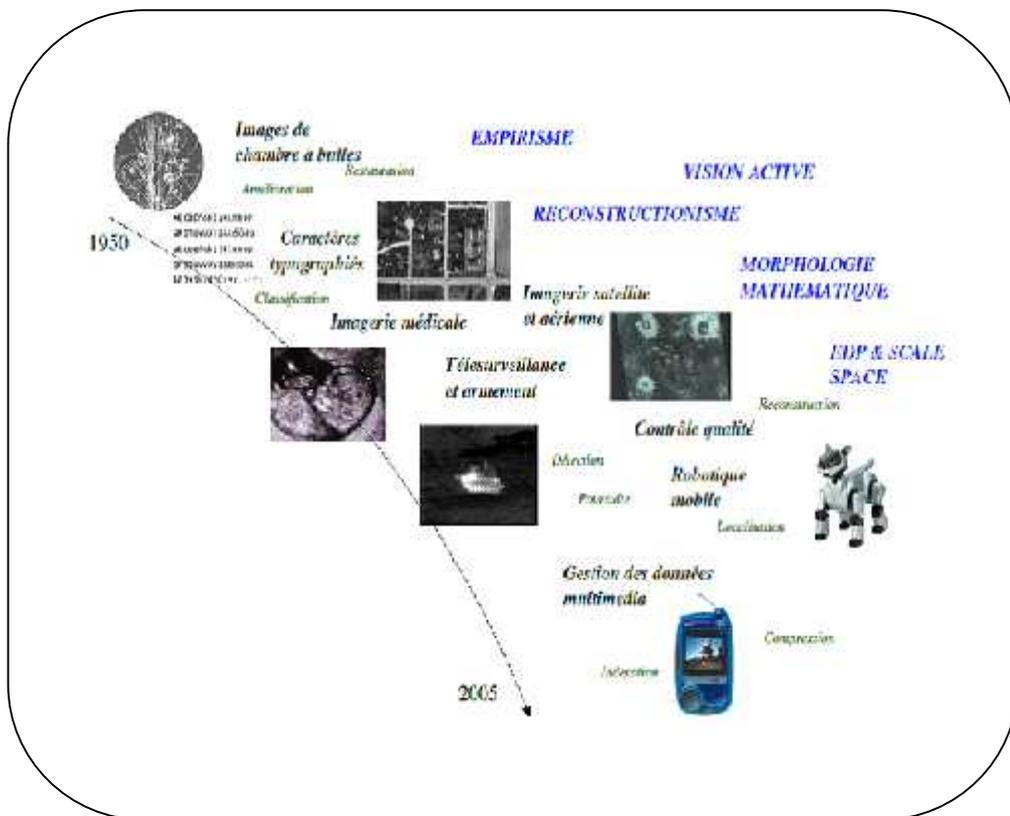
*Traitement d'images-Navigation*

## II.1. INTRODUCTION :

Dans ce chapitre on se concentrera sur les concepts fondamentaux du traitement d'images : historique, définition, Acquisition, domaine d'application, nous avons présenté le différent type de navigation, les représentations de l'environnement.

## II. 2. HISTORIQUE DU TRAITEMENT D'IMAGES :

La technologie numérique moderne est devenue omniprésente. Grâce à elle, il est devenu possible de traiter des signaux multidimensionnels avec des systèmes très divers, depuis les téléphones portables jusqu'aux ordinateurs massivement parallèles. Dans le domaine des images, Le traitement des images, opère une transformation image  $\rightarrow$  image.



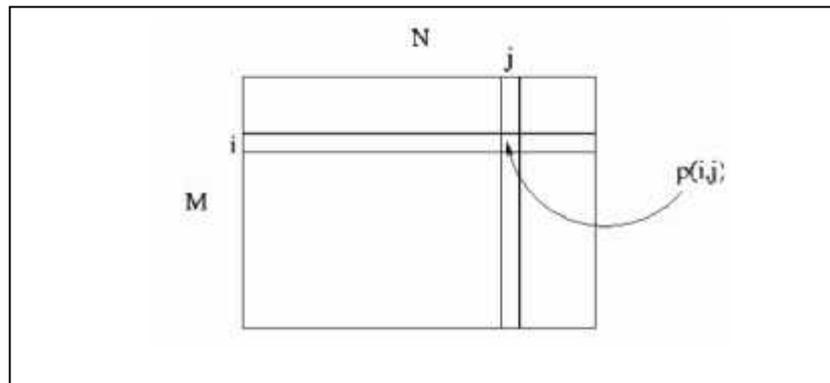
**Figure. II.1 :** Développement de la science traitement d'image depuis 1950 à 2005 [13].

## II.3. VISUALISATION:

Une image réelle est obtenue à partir d'un signal continu bidimensionnel comme par exemple un appareil photo ou une caméra... Sur un ordinateur, on ne peut pas représenter de signaux continus, on travaille donc sur des valeurs discrètes [14].

**II.3.1. Définition :** Une image numérique est définie comme un signal fini bidimensionnel échantillonné à valeurs quantifiées dans un certain espace de couleurs. Elle est constituée de points (pixels).

- \* **Signal fini :** une image possède des dimensions finies, exemple : 640x480, 800x600 points...
- \* **Signal bidimensionnel :** une image possède deux dimensions : largeur, hauteur.
- \* **Signal échantillonné :** les pixels d'une image sont régulièrement espacés sur une grille carrée.
- \* **Valeurs quantifiées :** les valeurs des pixels appartiennent à un intervalle borné Connu.
- \* **Espace de couleur :** il existe de nombreuses façons de percevoir les couleurs d'une image, l'espace de représentation le plus connu est l'espace **RGB** (rouge-vert bleu). Autrement dit, une image est une matrice  $M \times N$  de valeurs entières prises sur un intervalle borné  $[0, N_g]$  où  $N_g$  est la valeur maximale du niveau de gris [14].



**Figure II.2 :** Le niveau de gris du pixel.

$p(i, j)$  Est le niveau de gris du pixel de coordonnées ligne  $i$  et colonne  $j$  dans l'image.

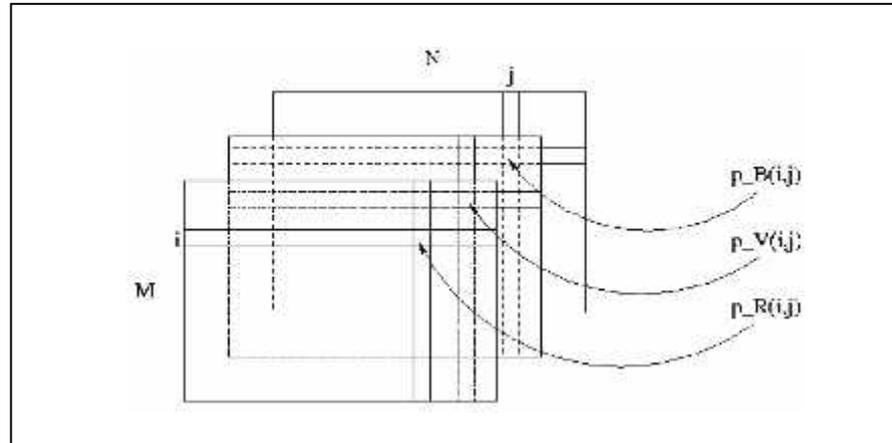
$p(i, j) \in [0, N_g]$ . Les valeurs des niveaux de gris sont des entiers.

**II.3.2. Image binaire:** Une image binaire est une image  $M \times N$  où chaque point peut prendre uniquement la valeur 0 ou 1. Les pixels sont noirs (0) ou blancs (1). Le niveau de gris est codé sur un bit. Avec  $N_g = 2$  et la relation sur les niveaux de gris devient:  $p(i, j) = 0$  ou  $p(i, j) = 1$ .

**II.3.3. Image en niveaux de gris :** Une image aux niveaux de gris autorise un dégradé de gris entre le noir et le blanc. En général, on code le niveau de gris sur un octet (8 bits) soit 256 nuances de dégradé. L'expression de la valeur du niveau de gris avec  $N_g = 256$  devient:

$p(i, j) \in [0, 255]$ .

**II.3.4. Image couleur :** Une image couleur est la composition de trois (ou plus) images en niveaux de gris sur trois (ou plus) composantes. On définit donc trois plans de niveaux de gris, un rouge, un vert et un bleu. La couleur finale est obtenue par synthèse additive des ces trois (ou plus) composantes.



**Figure.II.3 :** Image couleur.

On a les relations sur les niveaux de gris:  $p_R(i, j) \in [0, 255]$ ,  $p_V(i, j) \in [0, 255]$ ,  $p_B(i, j) \in [0, 255]$ . On voit bien sur la figure qu'une image couleur est en fait l'association de trois plans de niveau de gris, chacun d'eux étant une couleur de base [14].

**II.3.5. Image à valeurs réelles :** Pour les images, le résultat peut ne pas être entier, il est donc préférable de définir l'image de départ et l'image résultat comme des images à valeurs réelles [14]. En général, une image à valeurs réelle est telle que le niveau de gris est un réel compris entre 0.0 et 1.0. On a dans ce cas pour une image à niveaux de gris [14]:  $p(i, j) \in [0.0, 1.0]$ . Pour une image couleur, la relation devient  $p_R(i, j) \in [0.0, 1.0]$ ,  $p_V(i, j) \in [0.0, 1.0]$ ,  $p_B(i, j) \in [0.0, 1.0]$ .

#### II.4. ESPACE DE COULEURS:

La couleur est une donnée importante pour une image, elle modifie la perception que l'on a de l'image. L'espace de représentation standard décompose une image en trois plans de couleur: **RGB**. Les couleurs finales sont obtenues par synthèse additive de ces trois couleurs primaires.

Il existe cependant des problèmes qui peuvent nécessiter de changer d'espace de couleur pour percevoir différemment l'image. Il y a des Images où la couleur importe peu, par exemple des photographies de cellules vivantes (pseudo-transparentes), des images

radar, des images satellites... Dans ce cas, l'espace **RGB** n'est plus utilisé. On lui préfère d'autres espaces comme **HSV** ou **YCbCr** [14].

## II.5. ACQUISITION: ECHANTILLONNAGE / QUANTIFICATION :

### II.5.1. Effets de l'échantillonnage :

-*Pixellisation* [16] :

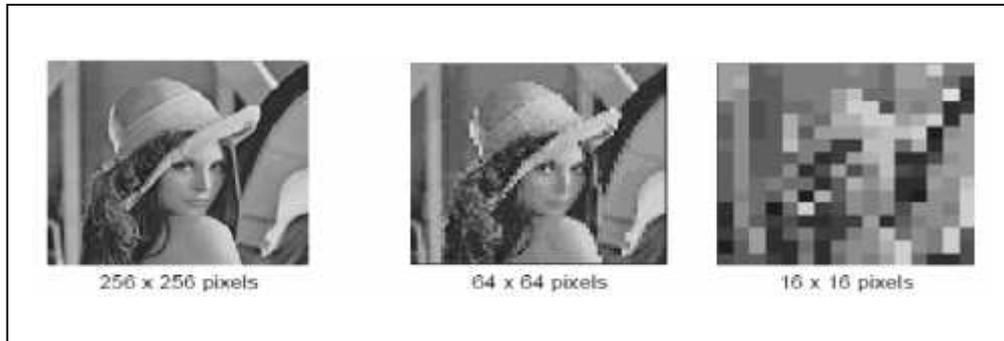


Figure. II.4 : *Pixellisation.*

- ✚ Contours en marche d'escalier.
- ✚ Perte de netteté.
- ✚ Détails moins visibles/ moins précis.
- ✚ Perte de résolution.

### II.5.2. Effets de la quantification à l'acquisition : [16]

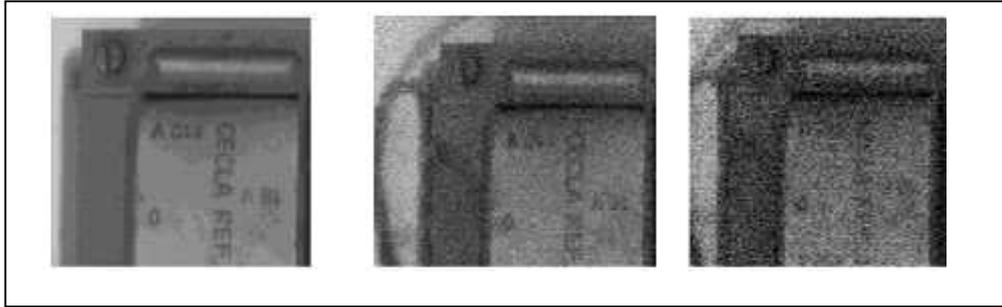
- ✚ CAN sur les systèmes d'acquisition d'images.
- ✚ Codage de la valeur de chaque pixel sur N bits (En général 8 bits).



Figure .II.5 : *Effets de la quantification à l'acquisition.*

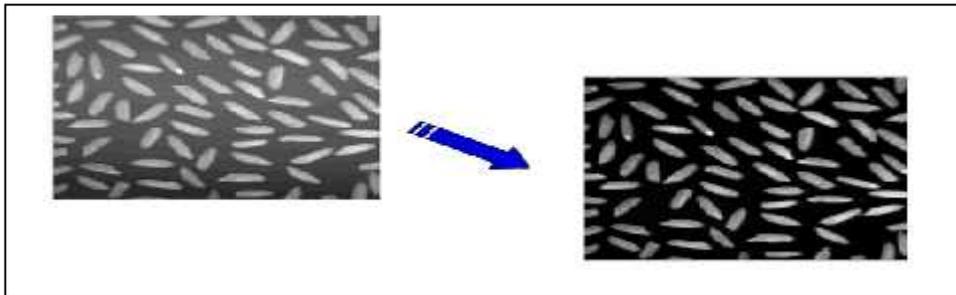
- ✚ Apparition de faux contours.
- ✚ Bruit de quantification.
- ✚ Effet visible à l'œil en dessous de 6/7 bits.
- ✚ Quantification sur 8 bits pour l'affichage.

**II.5.3. Bruits liés à l'acquisition :** Les images sont souvent entachées de bruit, parfois non visible à l'œil, et qui perturberont les traitements [16].

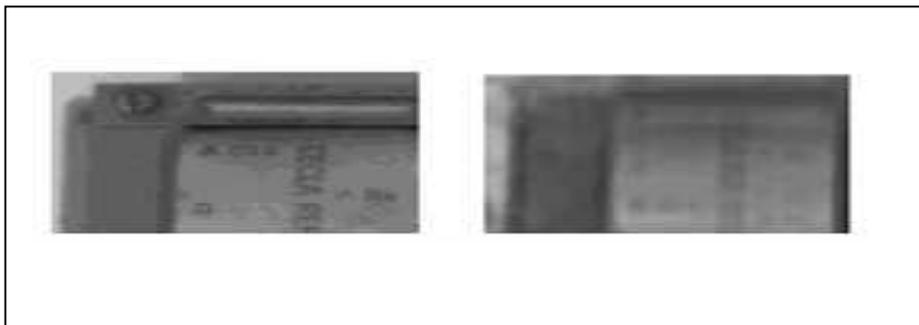


**Figure. II.6 :** Les images sont prises dans différent de bruits.

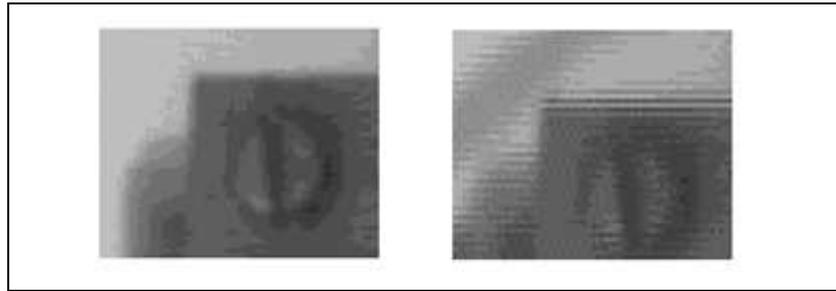
- ❖ Optimiser les conditions d'éclairage.
- ❖ Attention à l'éclairage ambiant.
- ❖ Diaphragme ouvert = faible profondeur de champ.
- ❖ Eclairage important = dégagement de chaleur.



**Figure. II.7 :** Comparaison entre un éclairage non uniforme et correction de l'éclairage.



**Figure .II.8 :** Flou de bougé/filé dû à un temps de pose/intégration trop long.



**Figure .II.9 :** Effet de lignage dû au balayage entrelacé des caméras vidéo.

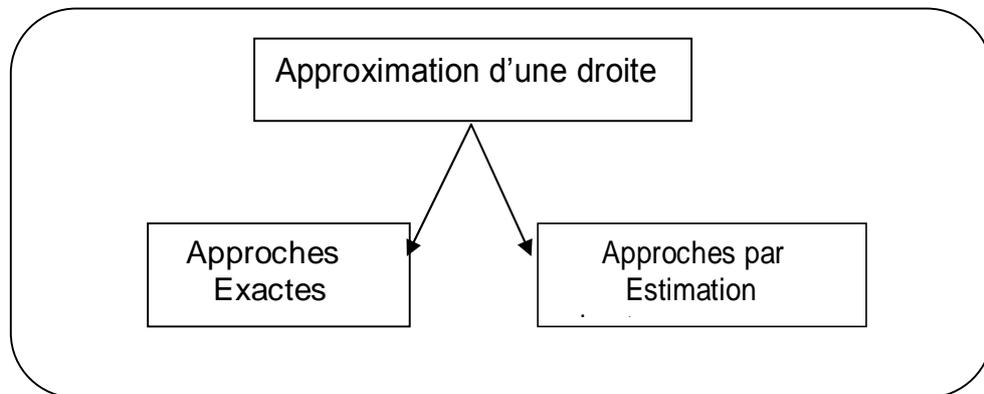
Une bonne acquisition  Des traitements facilités

## II.6.RECONNAISSANCE DE FORMES :

**II.6.1. Définition :** A partir d'un nuage de points (pixels) obtenu après une phase de traitement, on cherche à retrouver une forme connue a priori qui décrit «au mieux» les points extraits [17].

**II.6.2. Méthode approximation d'une droite :** Méthodes basées sur une approche par moindres carrés obtenue à partir de la minimisation d'une distance.

Il existe de nombreuses méthodes pour résoudre ce problème apparemment simple.



**Figure. II.10 :** Schémas présente d'approximation d'une droite.

**II.6.2.1. Méthodes exactes :** On considère que tous les points extraits participent avec le même poids à la description de la droite.

**II.6.2.2. Approximation par estimation robuste :** Recherche de la meilleure droite représentant au mieux l'ensemble des points sous l'hypothèse d'un bruit entachant la position des points [17].

**II.7. DOMAINES D'APPLICATION:**

Permet ces domaines [16] :

- \* Contrôle industriel.
- \* Robotique.
- \* Télédétection.
- \* Multi media.
- \* Médecine.

**II.8. TRAITEMENT D'IMAGE ET PERCEPTION VISUELLE :**

Une difficulté fondamentale de la vision artificielle vient du manque de connaissance profonde des mécanismes qui régissent la compréhension des images dans la nature. La vision humaine est en effet extrêmement performante (déplacement, lecture, reconnaissance), mais nous n'avons aucun retour conscient sur les mécanismes mis en jeu (à la différence du jeu d'échec par exemple).

En cela l'étude des mécanismes physiologiques et psychologiques de la vision est une source très importante d'information, et d'inspiration [13].

**II.9. LES DIFFERENTS TYPES DE NAVIGATION :**

**II.9.1 Guidage :** Cette capacité permet d'atteindre un but qui n'est pas un objet matériel directement visible, mais un point de l'espace caractérisé par la configuration spatiale d'un ensemble d'objets remarquables, ou amers, qui l'entourent ou qui en sont voisins [12] .

- La stratégie de navigation, souvent une descente de gradient également, consiste alors à se diriger dans la direction qui permet de reproduire cette configuration.
- Cette capacité semble utilisée par certains insectes, comme les abeilles, et a été utilisée sur divers robots.
- Cette stratégie utilise également des actions réflexes et réalise une navigation locale qui requiert que les amers caractérisant le but soient visibles.

**II.9.2. Action associée à un lieu :** Cette capacité est la première capacité réalisant une navigation globale, c'est-à-dire qui permette de rejoindre un but depuis des positions pour lesquelles ce but ou les amers qui caractérisent son emplacement sont invisibles [12].

- ✓ Elle requiert une représentation interne de l'environnement qui consiste à définir des lieux comme des zones de l'espace dans lesquelles les perceptions restent similaires, et à associer une action à effectuer à chacun de ces lieux.

- ✓ L'enchaînement des actions associées à chacun des lieux reconnus définit une route qui permet de rejoindre le but.
- ✓ Ces modèles permettent donc une autonomie plus importante mais sont limités à un but fixé.
- ✓ Une route qui permet de rejoindre un but ne pourra en effet pas être utilisée pour rejoindre un but différent.
- ✓ Changer de but entraînera l'apprentissage d'une nouvelle route, indépendante des routes permettant de rejoindre les autres buts.

**II.9.3. Navigation topologique :** Cette capacité est une extension de la précédente qui mémorise dans le modèle interne les relations spatiales entre les différents lieux [12].

- Ces relations indiquent la possibilité de se déplacer d'un lieu à un autre, mais ne sont plus associées à un but particulier. Ainsi le modèle interne est un graphe qui permet de calculer différents chemins entre deux lieux arbitraires.
- Ce modèle ne permet toutefois que la planification de déplacements parmi les lieux connus et suivant les chemins connus.

**II.9.4. Navigation métrique :** Cette capacité est une extension de la précédente car elle permet au robot de planifier des chemins au sein de zones inexplorées de son environnement. Elle mémorise pour cela les positions métriques relatives des différents lieux, en plus de la possibilité de passer de l'un à l'autre.

Ces positions relatives permettent, par simple composition de vecteurs, de calculer une trajectoire allant d'un lieu à un autre, même si la possibilité de ce déplacement n'a pas été mémorisée sous forme d'un lien [12].

## II.10. CARACTÉRISTIQUE :

En robotique mobile, comme pour l'homme ou certains animaux, l'utilisation de cartes est quasiment indispensable pour permettre d'effectuer des tâches de navigation dans des conditions environnementales complexes, qui ne sont pas spécialement adaptées pour le robot.

La construction et l'utilisation de telles cartes posent cependant de nombreux problèmes, notamment pour garantir l'adéquation entre la carte et le monde réel. Pour cette raison, la plupart des robots trouvent aujourd'hui un compromis entre une approche

réactive et une approche utilisant une carte afin de bénéficier de la rapidité et de la robustesse de la première et de la capacité de déplacement à long terme de la seconde [12].

## II.11. LES REPRÉSENTATIONS DE L'ENVIRONNEMENT :

### II.11.1. Carte topologique :

**II.11.1.1. Description:** Représenter l'environnement du robot sous forme de graphe: Les nœuds du graphe correspondent à des lieux, c'est-à-dire des positions que le robot peut atteindre et Les arêtes liant les nœuds marquent la possibilité pour le robot de passer directement d'un lieu à un autre et mémorisent en général la manière de réaliser ce passage.

La détection et la mémorisation des lieux reposent en général sur deux procédures qui utilisent les perceptions : La première permet simplement de comparer deux perceptions et donc de reconnaître un lieu de la carte ou de détecter un lieu nouveau et La seconde procédure permet de mémoriser un nouveau lieu ou d'adapter la définition d'un lieu lors des passages successifs du robot en ce lieu.

La première procédure peut donner des résultats erronés. Par exemple, un lieu déjà visité peut ne pas être reconnu, ou un lieu nouveau peut être confondu avec un lieu déjà mémorisé. Pour résoudre ces problèmes, la reconnaissance des lieux fera donc appel aux données proprioceptives en plus des perceptions [12].

**II.11.1.2. Avantage :** Un avantage important des cartes topologiques est qu'elles ne requièrent pas de modèle métrique des capteurs pour fusionner les données proprioceptives et les perceptions au sein d'une représentation unifiée de l'environnement.

Le bruit sur les mesures des capteurs influe principalement sur la reconnaissance des lieux, tandis que le bruit sur les données proprioceptives influe principalement sur la position associée à chaque lieu.

La mémorisation de l'environnement sous forme d'un ensemble de lieux distincts autorise en général une définition des lieux plus directement reliée aux capacités perceptives du robot.

Il n'y a pas de limitation au type de capteurs utilisables, Cette utilisation directe des perceptions permet meilleure mise en relation du robot avec son environnement.

La discrétisation de l'environnement correspondant au choix des lieux représentés dans la carte est un autre point fort des cartes topologiques. Cette discrétisation est en effet très utile pour la planification des mouvements du robot, qui se réduit alors à la recherche de chemin dans un graphe [12].

**II.11.1.3. Inconvénient :** Nécessitent en général une exploration très complète de l'environnement pour le représenter avec précision.

Représentant des lieux avec une assez grande densité spatiale, cela peut être un inconvénient, car l'exploration complète de l'environnement demandera un temps important.

La reconnaissance des lieux de l'environnement peut également être difficile dans le cas de capteurs très bruités, ou d'environnements très dynamiques. Il devient alors très difficile d'estimer correctement la position du robot au sein de cette carte et de lui ajouter de nouvelles informations sans erreurs.

La représentation de l'environnement peut être assez proche des données brutes des capteurs du robot, ce qui peut être un avantage du point de vue de l'autonomie du robot [12].

### **II.11.1. Carte métrique:**

**II.11.2.1. Description:** L'environnement est représenté par un ensemble d'objets auxquels sont associées des positions dans un espace métrique, généralement en deux dimensions.

Cette espace est, la plupart du temps, celui dans lequel s'exprime la position du robot estimée par les données proprioceptives qui permettent, en utilisant un modèle métrique des capteurs, de détecter ces objets et d'estimer leur position par rapport au robot.

La position de ces objets dans l'environnement est alors calculée en utilisant la position estimée du robot.

Carte de l'environnement correspond directement à l'espace libre, c'est-à-dire à l'espace dans lequel le robot peut se déplacer [12].

**II.11.2.2. Avantage :** Représente l'ensemble de l'environnement, Et non un petit sous-ensemble de lieux comme le font les cartes topologiques.

La position du robot est définie de manière non ambiguë par ses coordonnées au sein de l'espace dans lequel la carte est représentée.

Il s'ensuit une utilisation simple et directe de toutes les informations métriques fournies par les données proprioceptives ou les perceptions [12].

**II.11.2.3. Inconvénient :** Le calcul de chemin au sein des cartes métriques peut être plus complexe, car la planification se déroule dans un espace continu et non dans un espace préalablement discrétisé, comme c'est le cas pour les cartes topologiques.

De nombreux modèles recourent d'ailleurs à l'extraction d'une carte topologique depuis la carte métrique pour réaliser cette opération de planification [12].

## II.12. DOMAINES D'APPLICATION :

Objectif de créer des modèles riches de l'environnement qui peuvent avoir plusieurs utilisations. Ils peuvent notamment permettre à un opérateur distant d'avoir une bonne compréhension de l'environnement du robot, ou au robot d'avoir un comportement plus adapté à son environnement [18].

## II.13. MODÉLISATION DU CHEMIN PAR LA FORME:

Beaucoup d'applications de vision par ordinateur ont pour but la reconnaissance d'objets. Dans ces applications, la forme joue un rôle fondamental surtout quand d'autres attributs comme la couleur et la texture ne donnent pas d'information complémentaire suffisante sur la nature de l'objet. Par exemple, si nous parvenons à identifier (par couleur et texture) une région du type chemin sur la scène, nous ne savons rien sur sa catégorie (i.e., s'il s'agit d'un chemin « tout droit », d'un « virage à gauche », d'un « virage à droite », d'une « bifurcation », d'une « intersection », etc.) ce qui peut être déterminées de manière naturelle en utilisant la forme. L'apparence d'un objet est en effet, largement caractérisée par des mesures de dimensions physiques basées sur la forme.

Il n'y pas de définition universelle pour décrire ce qu'on appelle la « forme ».

Chez l'homme, l'impression de forme est généralement accompagnée par des informations de couleur et d'intensité (texture) qui permettent de faire émerger une représentation géométrique caractéristique de l'objet. Ainsi, l'identification des objets est réalisée de

manière courante en utilisant leurs propriétés visuelles ; nous définissons la forme comme un modèle géométrique constitué par un ensemble de points, courbes, surfaces solides, etc.

La mise en correspondance des formes va nécessiter leur transformation dans un espace de représentation dans lequel nous pourrions les comparer, trouver leurs ressemblances, en utilisant certaine mesure de similarité [19].

**II.13.1. Représentation de la forme des objets :** L'un des principaux problèmes pour la reconnaissance automatique des formes consiste à déterminer une représentation ou description convenable à cet effet. Elle doit être à la fois unique pour l'objet étudié et capable d'accepter certaines variations pour caractériser les éléments d'une même classe.

Les schémas de représentation de formes doivent satisfaire quelques propriétés pour donner des résultats satisfaisants : invariance à la translation, à la rotation et aux changements d'échelle. Par définition, ce type de transformations ne modifient pas la forme de l'objet et c'est plutôt en utilisant les transformations affines que l'on peut obtenir une certaine indépendance au point de vue.

D'autres facteurs peuvent gêner la mise en correspondance de formes dans des scènes naturelles, par exemple les occultations et la variation dans l'éclairage de l'objet. Dans notre application, les objets de l'arrière-plan comme les arbres, les bâtiments, les voitures ou les poteaux ont une forte influence sur l'aspect (la forme) des chemins. Par conséquent, il est important de modéliser non seulement les propriétés d'un objet mais également les relations entre ceux-ci et les objets du fond de la scène.

Les différentes représentations possibles pour définir la forme des objets 2D, peuvent être classifiées suivant les caractéristiques qu'elles utilisent :

#### **II.13.1.1. APPROCHES GLOBALES :**

- \* surface, périmètre, largeur, hauteur,
- \* moments invariants, moments de Zernike,
- \* morphologie mathématique,
- \* élongation, circularité,

### II.13.1.1.2. APPROCHES LOCALES :

- Coins ou sommets (position, angles, . . .).
- Segments (position, longueur, . . .).
- Facettes ou régions (position, surface, couleurs, texture, . . .), en utilisant les contours de l'objet, squelette.
- Représentation par des histogrammes (Shape Context, signature polaire).
- Descripteurs de Fourier, représentation par vecteurs propres, splines . . .

La reconnaissance des objets 2D est menée par des méthodes différentes :

- Appariement de gabarit « template matching » : les objets à détecter sont représentés par des échantillons (imagelettes) ou gabarits (templates). L'image étudiée est balayée avec le template en comparant les pixels du template à ceux de la portion de l'image recouverte.

Une mesure de comparaison du type coefficients de corrélation ou somme de différences (Maximales ou absolues) sert à quantifier la similarité entre les template et une portion de l'image.

- Méthodes basées sur l'apparence (en exploitant l'espace propre)
- Méthodes fondées sur des caractéristiques : Arbres d'interprétation, alignement etc.

La catégorisation d'un chemin perçu dans une scène doit utiliser une représentation qui exploite le contour de la région correspondante de l'image.

Le descripteur choisi pour représenter la forme d'un chemin est appelée Shape Context, considère comme une alternative fiable de classification et d'identification d'objets à partir de leurs points de contour. Du fait de la polyvalence et de la richesse de ce descripteur, nous l'avons adopté pour la classification et la catégorisation de chemins (de terre, goudronne...) à partir de leurs caractéristiques morphologiques (carrefours, lignes droites, intersections en T, impasse, virages, etc.). Notre principale motivation vise l'exploitation des intersections détectées dans un réseau de chemins, pour la construction d'une carte topologique exploitée ensuite pour la navigation autonome.

**II.13.2. Catégorisation des chemins par shape context :** La signature polaire d'une forme, décrite par un ensemble de points, est donnée par la distribution grossière de la forme par rapport à un point de référence ; la distribution est représentée par un histogramme polaire des points de la forme autour du point de référence.

Le descripteur Shape Context est l'ensemble des signatures polaires de la forme par rapport à chacun des points qui la décrit. La mise en correspondance entre deux formes décrites par Shape Context, se fait en deux étapes :

- (1) on recherche pour chaque point échantillon sur une forme, le point échantillon sur l'autre qui a la signature polaire la plus semblable.
- (2) La maximisation de similarités globales de tous les points est résolue par l'optimisation d'un graphe bipartite.

Cette méthode n'impose aucune restriction sur les propriétés inhérentes des points représentant la forme. C'est-à-dire qu'ils ne correspondent pas à des points de courbure, ni à des maxima, des minima, des points d'inflexion, etc. Pour la simplicité algorithmique, nous avons préféré travailler avec des points uniformément séparés.

D'après nos expériences sur l'extraction de contours issus des scènes naturelles, ou les bords des régions sont souvent assez irréguliers, cette approche nous semble assez souple pour la représentation des chemins.

## II.14. MODÉLISATION ET CATÉGORISATION DE CHEMINS :

**II.14.1. Définition chemin :** Un chemin entre une situation initiale  $X_I$  et une situation finale  $X_F$  est une représentation géométrique du d'emplacement du corps qui ne considère pas le temps [9].

**II.14.2. Indexation de chemins par la forme :** Pour l'indexation de chemins à partir de la forme des régions étiquètent Chemin dans les images, nous ne stockons que les points de contours lisses de ces régions. Les descripteurs « Shape Context » sont calculés pour toutes les images de la base de données, et stockées en mémoire pour leur réutilisation.

Le but de l'indexation est de ranger ces descripteurs dans une structure qui permette de retrouver facilement les descripteurs approximativement égaux à un descripteur donné. Nous avons un critère de comparaison du type plus proche voisin entre les descripteurs de

la base de données et celui de la région à identifier, i.e., la forme apprise ayant la distance la plus petite a la forme a classifier.

En conclusion, nous considérons possible d'exploiter cette technique d'indexation, comme élément additionnel dans la navigation autonome et la construction de cartes topologiques. Pour ce faire, les chemins caractéristiques rencontrés dans la nature (disons : carrefours, intersection, virages, etc.) sont pré-classifiés en tant qu'ayant des propriétés semblables, ce qui va nous permettre d'augmenter la base de données.

La catégorisation d'un chemin devient alors le processus d'indexation de cette forme dans notre base de données [19].

### **II.15.CONCLUSION :**

On a commencé ce chapitre par un bref historique sur la science de traitement d'image ensuite on a présenté une notion de base de traitement d'image numérique, l'espace de couleurs, l'acquisition d'image et comment reconnaître la forme aussi sont domaine d'application, traitement d'image et la perception visuelle.

Nous avons implanté les différents types de navigations comme le guidage, l'action associée à un lieu la navigation topologique et métrique.

# Chapitre III

*Réalisation pratique*

### III.1. INTRODUCTION :

L'objectif de ce travail est de rendre d'un robot mobile capable de suivre visuellement une ligne dans un terrain.

Ce chapitre est composé de trois parties :

- ✚ Dans la première partie on fait une description mécanique du robot mobile.
- ✚ La deuxième partie est consacrée à la partie électronique.
- ✚ Dans la troisième partie, on présente les différentes procédures et programmes utilisés pour la commande du robot.

### III.2. LA MÉCANIQUE :

La partie mécanique se compose :

- D'un châssis.
- D'un bloc équipé de deux moteurs.
- D'une roue libre.

La partie de commande comprend :

- ✓ Une camera (webcam).
- ✓ Une carte Arduino Uno.

**III.2.1. Le châssis :** On a utilisé cette configuration (robot mobile à roue différentielle) à cause de sa simplicité.

**III.2.2. Principe de fonctionnement :** La camera détecte la position du robot par rapport à la ligne jaune tracé sur la piste et transmettent cette commande à Notebook, ce dernier envoyé la commande à la carte Arduino Uno pour commander les deux moteurs CC.

**III.2.3. Structure du système :** le robot mobile est composé d'une plate forme mobile capable de porter les composants du système .La plate forme utilisée est très simple composée d'un châssis, deux moteurs et une roue de commande libre.

### III.3. L'ÉLECTRONIQUE :

**III.3.1. Alimentation :** L'unité d'alimentation du robot se compose d'une batterie 12V :

Un régulateur 7805a été utilisé pour obtenir une tension régulée très stable de +5V pour l'alimentation de la partie logique du circuit L298.

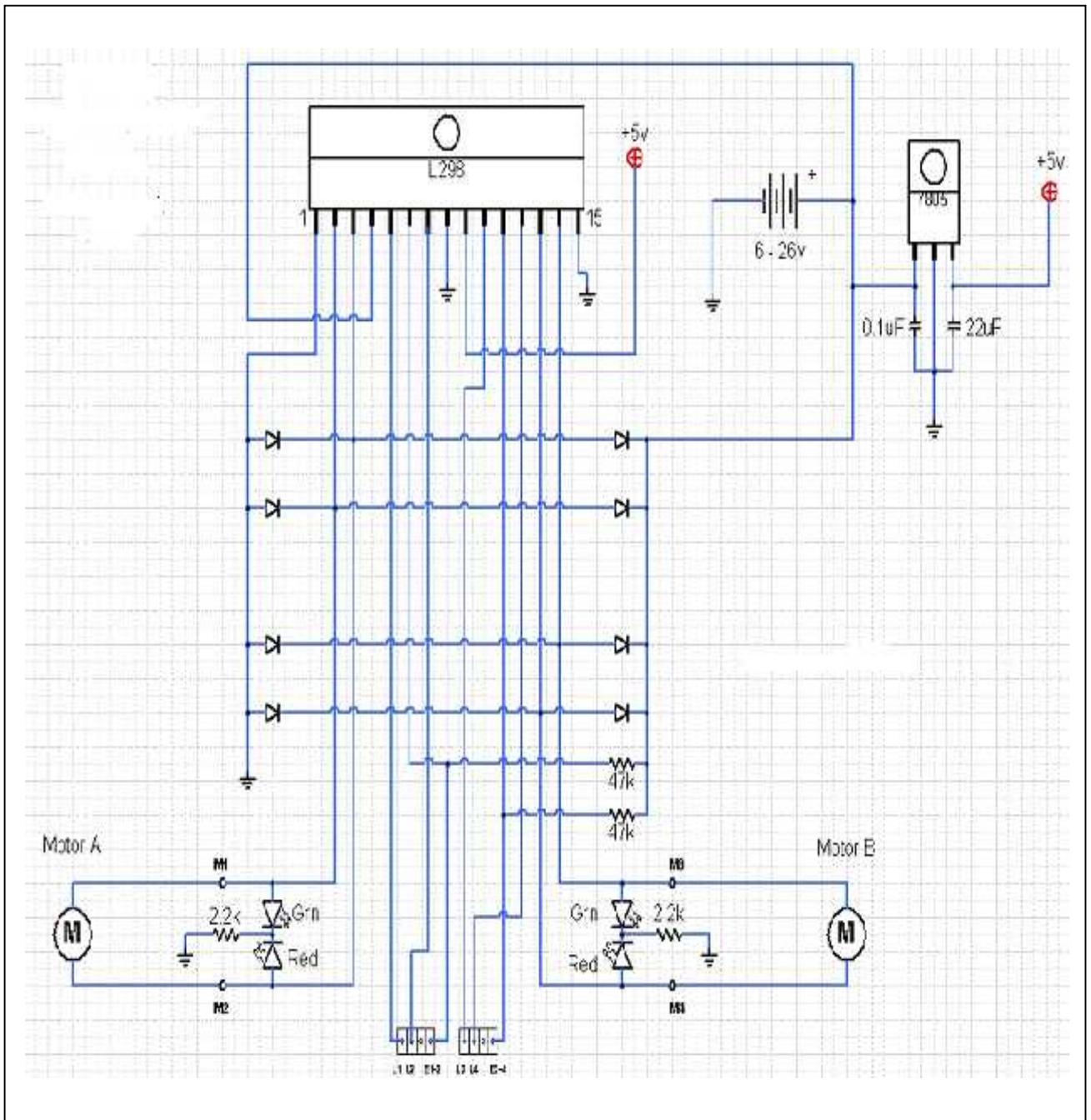


Figure. III. 1 : Schéma de commande des moteurs avec le circuit L298.

**III.3.2. Commande des moteurs CC :** Pour assurer le déplacement du robot, on a utilisé deux moteurs CC .Ces dernier fonctionnent sous une tension de 12V.

Le circuit intégré L298N à été choisi pour commander les moteurs.

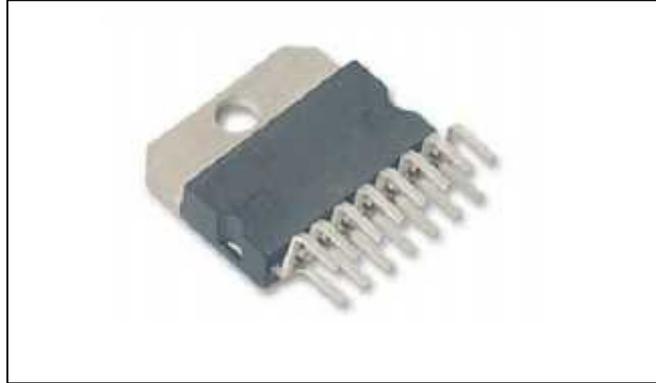


Figure .III.2 : Hacheur L298 [Annexe A].

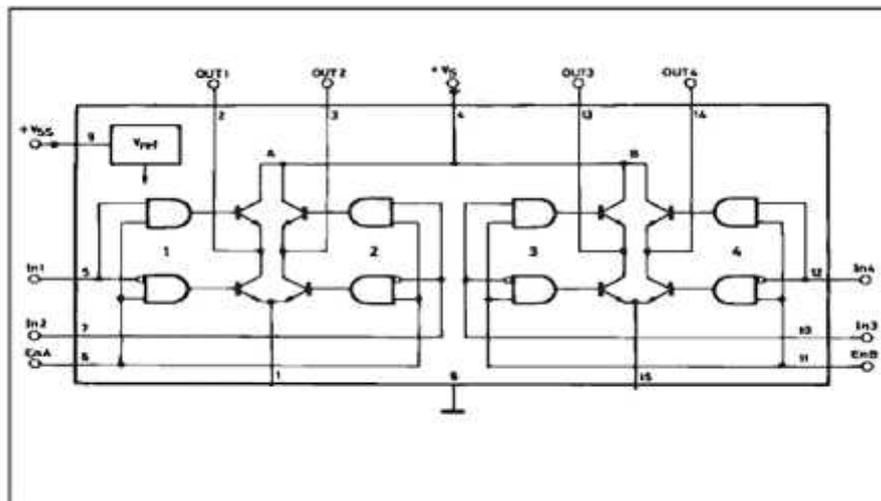


Figure .III.3 : Schéma interne du L298 [Annexe A].

- **Principales caractéristiques :**

- Tension d'alimentation du circuit logique : 5V
- Tension d'alimentation du circuit puissance : 0V → 48V
- Tension minimale de sensibilité MLI : 2.3V
- Courant maximal du circuit puissance : 3A

**III.3.2.1. Pont en H :** Une structure électronique servant à contrôler polarité aux bornes d'un dipôle. Il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom. Les commutateurs peuvent être des relais, des transistors, ou autres éléments de commutation en fonction de l'application visée.

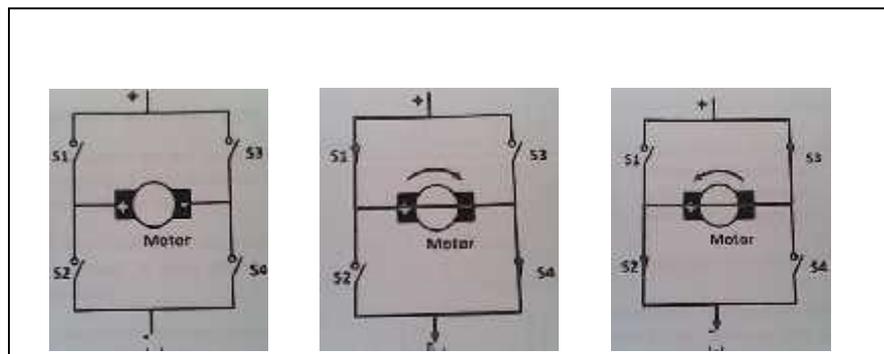
Le pont en H (Voir la **Figure. III.4**) permet de contrôler la polarité de la tension aux bornes du moteur, ou de déconnecter l'alimentation des moteurs (moteur arrêté). Les commutateurs sont actionnés deux par deux soit S1-S4 ou S2-S3 pour faire tourner le moteurs dans un sens ou dans l'autre sens. Les paires de relais peuvent avoir des puissances différentes selon que l'utilisation ne requiert pas le même couple dans un sens que dans l'autre sens.

De plus, le pont en H permet d'effectuer un freinage magnétique s'il est capable d'en dissiper la puissance générée. Cette opération s'effectue en actionnant soit les deux commutateurs supérieurs ou inférieurs en même temps, ce qui court-circuit les bornes du moteur et le fait par conséquent freiner.

Les états d'un pont en H sont représentés sur le **Tableau .III.2**.

Moteur	Commutateur				0 : commutateur ouverte 1 : commutateur fermée
	S1	S2	S3	S4	
Roue libre	0	0	0	0	
Sens 1	1	0	0	1	
Sens 2	0	1	1	0	
Arrêt	1	0	1	0	
	0	1	0	1	

**Tableau .III.2 :** Table de vérité de pont en H.



**Figure .III.4 :** Etat d'un pont en H : (a) Moteur déconnecté, (b) Rotation ou sens1, (c) Rotation inverse ou sens inverse.

**III.3.2.2. Réalisation de module de commande des moteurs à CC :** Le module de commande utilise un circuit L298N comme double pont en H pour la commande des deux moteurs dans les deux directions (dans le sens des aiguilles d'une montre, et le sens inverse des aiguilles d'une montre), huit diodes ont été utilisées pour former deux ponts en diode pour la protection de circuit L298N contre les surtensions provoquées par les charges inductives.

Les entrées IN1, IN2, IN3 et IN4 du circuit L298N sont utilisées pour commander le sens de la rotation du moteur 1 et du moteur 2 respectivement.

Les broches ENA et ENB sont directement reliées aux sorties (PWM) du MCU pour commander la vitesse du robot par un signal PWM.

Les broches SENSEA et SENSB doivent être liées à la masse directement ou à travers une petite résistance. En utilisant une petite résistance et en connectant le SENSX on peut vérifier si les moteurs sont surchargés ou si le L298N consomme un courant important.

Un connecteur est utilisé pour la connexion du module de commande avec la carte de commande : carte Arduino UNO.

### **III.3.4. La carte Arduino Uno :**

**III.3.4.1. Définition :** Arduino Uno est une carte microcontrôleur basée sur l'ATmega328. Elle est dotée de 14 broches en Entrées /Sortie numérique (dont 6 peuvent être utilisées comme sorties PWM-Modulation de largeur d'impulsion), 6 entrées analogiques, un oscillateur à quartz 16MHZ, une connexion USB, un jack d'alimentation, un support ICSP, un bouton 'reset'.

Elle contient tout le nécessaire pour fonctionner ; il suffit de la connecter à un ordinateur par le câble USB et de l'alimenter à l'aide d'un adaptateur USB ou des piles.

La carte Arduino Uno peut être alimentée par l'USB ou par une alimentation externe.

La source est sélectionnée automatiquement.

La tension d'alimentation extérieure (hors USB) peut être délivrée par un adaptateur AC-DC ou des piles. L'adaptateur peut être connecté grâce à un 'jack' de 2.1 mm positif au centre.

Le raccordement vers un bloc de piles peut utiliser les bornes Gnd et Vin du connecteur d'alimentation (POWER).

La carte peut fonctionner à l'aide d'une tension extérieure de 7 à 12 volts.

Les broches (pins) d'alimentation sont les suivantes :

- VIN : C'est la tension lorsque la carte est raccordée à une source extérieure (par opposition à la tension de 5 volts venant de l'USB ou d'un régulateur de tension).
- 5V : La tension d'alimentation régulée : elle peut venir de VIN via le régulateur sur la carte, ou être fournie par l'USB ou un régulateur extérieur. Le courant maximum est de 50 mA.
- GND : Masse.

La carte Uno diffère des précédentes cartes en ce qu'elle n'utilise pas le *ship* convertisseur USB-Série, FTDI, mais le microcontrôleur ATmega8U2-sur la carte - programmée comme convertisseur USB-Série.

“Uno “ signifie “ Un“ en italien et ce nom annonce la sortie de l'Arduino 1.0.

<b>Les principales caractéristiques de l'Arduino Uno</b>	
Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'Input (recommandée)	7-12V
Tension d'Input (limites)	6-20V
Pins I/O digitales	14(dont 6 sont des PWM- modulation de largeur d'impulsion)
Pins Input Analogiques	6
Courant DC par pin I/O	40mA
Courant DC pour la broche 3.3V	50mA
Mémoire flash	32KB (ATmega328) dont 0.5KB utilisé par le 'bootloader'
SRAM	2KB5ATmega328)
Fréquence d'horloge	16MHz

**Tableau.III.4** : Les principales caractéristiques de l'Arduino Uno.

### III.3.4.2 Entrées / Sorties :

Chacune des 14 broches digitales de la Uno peut être utilisée en entrée ou sortie, en utilisant les fonctions `pinMode()`, `digitalWrite()`, et `digitalRead()`.

Elles fonctionnent en logique 0V-5V ; chacune pouvant fournir (source) ou recevoir un courant maximal de 40 mA et dispose si besoin d'une résistance interne de 'pull-up'.

En outre, certaines broches ont des fonctions spécialisées:

- **Serial:** 0 (RX) and 1 (TX). Permet de recevoir (RX) et de transmettre (TX) des données séries TTL. Ces broches sont raccordées à leurs homologues sur le chip ATmega8U2 spécialisé dans la conversion USB-to-TTL série.
- **Interruptions externes** 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur LOW, sur un front montant ou descendant, ou encore sur le changement de valeur.
- **PWM:** 3, 5, 6, 9, 10, and 11. Output 10-bits de PWM avec la fonction `analogWrite()`.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches fournissent le support de communication SPI en utilisant la 'library' spécialisée.
- **LED:** 13. Il y a une LED connectée à la broche digitale 13.

La carte Uno a 6 broches d'entrées analogiques, A0 à A5, chacune avec 10 bits de résolution.

Par défaut les mesures sont effectuées de la masse à 5V (valeur de référence), mais il est possible de spécifier la valeur de référence en utilisant la broche VREF et la fonction `analogReference()`.

En outre, certaines broches ont des fonctions spécialisées :

- **I2C:** 4 (SDA) and 5 (SCL). Permettent le support du bus I2C (TWI) en utilisant la 'library' Wire.

Il y a encore quelques autres broches sur la carte:

- **AREF.** Tension de référence déjà mentionnée.
- **Reset.** Permet au niveau bas (LOW<sup>o</sup>) de faire un reset du contrôleur. Elle est utilisée typiquement pour monter un bouton 'reset' aux cartes additionnelles ('shields') bloquant celui de la carte principale.

**III.3.4.3 Communication :** La carte Arduino Uno a de nombreuses facilités de communications avec un ordinateur, une autre Arduino, ou un autre microcontrôleur. L'ATmega328 permet la communication série UART TTL, grâce aux broches digitales 0 (RX) et 1 (TX).

Un contrôleur ATmega8U2 sur la carte, canalise cette communication série vers l'USB et apparaît comme un port de communication série virtuel pour le logiciel sur l'ordinateur. Le firmware de l'8U2 utilise le protocole USB, et aucun driver externe n'est nécessaire.

Logiciel de programmation de l'arduino uno :

- Le logiciel de l'Arduino inclut un moniteur série permettant l'envoi et la réception de simples données textuelles. Les leds RX et TX sur la carte clignotent pour indiquer la transmission de données via le chip USB-to-serial vers l'ordinateur.
- Une bibliothèque 'SoftwareSerial' permet la transmission de données série à partir de chacune des broches digitales du Uno.
- L'ATmega328 supporte le bus I2C (TWI) et le protocole synchrone maître-esclave SPI. Le logiciel de l'Arduino inclut un ensemble de fonctions pour mettre en œuvre ces fonctions.

Pour la réalisation de notre programme on a utilisé le langage Python, et la bibliothèque open CV.

- Logiciel de programmation de l'arduino uno. Arduino IDE 1.0 pour Windows.

## III.4. PROGRAMMATION :

### III.4.1 Python :

**III.4.1.1. Définition :** Python est un langage de programmation très puissant qui est de plus en plus utilisé pour développer des applications rapidement et en même temps efficaces. Il est très utilisé dans le monde scientifique pour sa syntaxe aérée et une vitesse d'exécution rapide.

Ce langage est aussi excellent pour la création de prototypes car sa simplicité permet d'implémenter un projet aussi vite que l'on y réfléchit.

**III.4.1.2. Caractéristique du langage de programmation (python) :**

- IL est portable, non seulement sur les différentes variantes d'UNIX, mais aussi sur les OS propriétaires: MacOS, BeOS, NeXTStep, MS-DOS et les différentes variantes de Windows.
- Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux.
- Python convient aussi bien à des scripts d'une dizaine de lignes qu'à des projets complexes de plusieurs dizaines de milliers de lignes.
- La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles. A fonctionnalités égales, un programme Python est souvent de 3 à 5 fois plus court qu'un programme C ou C++ (ou même Java) équivalent, ce qui représente en général un temps de développement de 5 à 10 fois plus court et une facilité de maintenance largement accrue.
- Peut être étendu avec des modules en C.
- Le langage Python est un langage de programmation complet grâce aux nombreuses bibliothèques spécialisées, appelées modules.

**📦 NumPy :**

- NumPy est une extension du langage de programmation [Python](#), destinée à manipuler des [matrices](#) ou tableaux multidimensionnels. Il contient:
  - un objet puissant, tableau à N dimensions.
  - Des outils pour l'intégration de C / C++ et Fortran.
  - l'algèbre linéaire, transformée de Fourier.

**📦 SciPy :**

- SciPy est un projet visant à unifier et fédérer un ensemble de [bibliothèques Python](#) à usage scientifique. Scipy utilise les tableaux et matrices du module NumPy. Cette distribution de modules est destinée à être utilisée avec le langage interprété [Python](#) afin de créer un environnement de travail scientifique.

📦 **PySerial** : est un module d'accès au [port série](#) spéciale pour [Python](#).

📦 **Pywin32**: est un module de Python pour Windows Extensions.

### III.4.1.3. Open CV :

Open CV (Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société Willow Garage assure le support de cette bibliothèque depuis 2010.

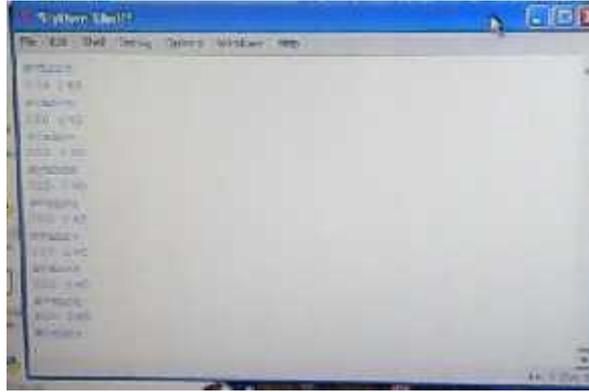
**III.4.1.3.1. Fonctionnalités d'Open CV:** La bibliothèque Open CV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

**III. 4.2. L'interface graphique:** Après l'exécution du programme on obtient deux fenêtres sur le bureau du PC :

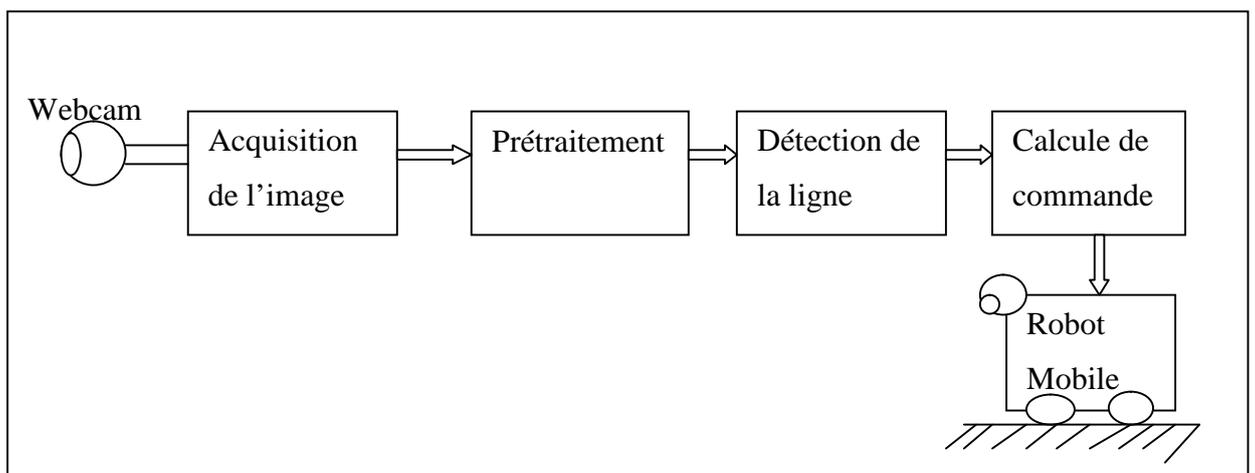
La première permet le choix de la couleur de la ligne à poursuivre par notre robot, et la deuxième permet d'afficher les informations (comme la détection de la ligne, avancé, droite et tourné à gauche,...).



Figure. III.5 : La première fenêtre concerne la webcam.



**Figure. III.6 :** La deuxième fenêtre montre la détection de la ligne.



**Figure. III.7 :** Schéma du principe du système de commande.

### III.5. CONCLUSION :

D'après les résultats obtenus et au cours de la réalisation de ce robot on peut conclure que ce travail peut être divisé en deux grandes parties, la partie software qui est basée essentiellement sur la programmation en langage Python, et la partie hardware qui est la réalisation de la partie électronique et mécanique du robot. Le travail est l'intégration entre ces deux parties pour la réalisation du robot capable de suivre en toute autonomie la ligne tracé, et les résultats obtenus montrent que le robot peut détecter et poursuivre la ligne détecté.

---

## Conclusion générale et perspectives

Ce projet couvre un grand nombre de domaines :

- La partie hardware : on a utilisé une carte de commande Arduino uno, qui se caractérise essentiellement par la programmation directe par un PC sans un programmeur, pour la commande de deux moteurs DC 12V, ces composants sont monté sur le châssis du robot.
- La partie software : on a utilisé le langage Python, et la librairie open CV pour la réalisation de notre programme après la configuration de ces deux logiciels et des autres modules annexes.

Consiste à l'étude et la mise en œuvre d'un robot suiveur de ligne.

Ce travail exige la maîtrise de plusieurs notions techniques aussi bien le domaine électronique et celui de l'informatique, ce que fait que nous avons eu l'occasion à découvrir, la programmation bas niveau par *La carte **Arduino Uno***, Langage **PYTHON**, et la librairie **Open CV**.

D'après le planning que nous nous étions fixés, on peut dire que nous sommes arrivés à terme du projet dans le temps.

Au terme de notre travail, nous prévoyons des perspectives :

Une première perspective on a besoin d'utilisés des cameras de haute qualité à cause de la perturbation de l'environnement (luminosité, les obstacles inévitables, ...)

Une deuxième perspective consiste à utilisés une méthode de détection des objets basé sur les couleurs.

Une deuxième perspective consiste à utiliser d'autres capteurs pour permettre l'évitement d'obstacles.

# BIBLIOGRAPHIE

- [1] Melle AKLI Isma. «Elaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant ». MEMOIRE de MAGISTER, Université Houari Boumediene, Oran ,Alger, le : 14 / 07 / 2007.
- [2] A. ALLOUI, A.HAJ Brahim. « Proposition d'une solution multi-agent pour la commande et la coopération multi-robot mobile ».Mémoire d'ingénieur d'état en automatique, Université Biskra, juin 2007.
- [3] M. GHAOUI. « Planification d'un mouvement pour un robot mobile »Thèse de magistère université de Batna.année 1997.
- [4] El-Hadi GUECHI. « Suivi de trajectoires d'un robot mobile non holonome : approche par modèle flou de Takagi-Sugeno et prise en compte des retards ». Thèse Doctorat présentée à UNIVERSITÉ DE VALENCIENNES ET DU HAINAUT CAMBRÉSIS Mention Automatique, le 28 juin 2010.
- [5] A, Pruski , “Robotique mobile, la planification de trajectoire”, ED. Hermes, Paris 1996.
- [6] C, Cappelle, El Badaoui El Najjar, M., Pomorski, D., Charpillet, F., "Détection, suivi Et géolocalisation d'obstacles à l'aide d'un modèle 3D géo-référencé, une caméra et Un GPS : validation avec un lidar", Conférence Internationale Francophone d'Automatique, CIFA'08, Bucarest, Roumanie, 3-5 Septembre, 2008.
- [7] Yang Cheng, Maimone, M.W., Matthies, L., “Visual odometry on the Mars Exploration rovers- a tool to ensure accurate driving and science imaging”, IEEE Robotics & Automation Magazine, Volume 13, Issue 2, June 2006. Page(s):54 – 62.
- [8] Hacéne ABERKANE & Khireddine Messaoudene, « Mini robot mobile», MFE ingénieur, Université de BATNA, 2001.
- [9] Mohamed Selim BEN HIMANE. « Vers une approche unifiée pour le suivi temps-réel et l'asservissement visuel ». Thèse Doctorat, L'ECOLE NATIONALE SUPERIEURE DES MINES DE PARIS, le 06 Décembre 2006.
- [10] Khiat Abdelaziz, Ouzaid Saïd. « Etude de la manœuvrabilité d'un robot mobile autonome non-holonome évoluant dans un espace restreint ». Mémoire d'ingénieur d'état

en informatique de l'Institut National de formation en Informatique (I.N.I) (ex: C.E.R.I.), Promotion 1997.

[11] Bouali Abdelmalek Ingénieur d'état en électronique. « Planification de trajectoire pour un robot mobile ». Mémoire Présenté pour obtenir le diplôme de magister Electronique, Université El-Hadj Lakhdar – BATNA – le : 30 /06/2012.

[12] David FILLIAT. « Robotique Mobile (cour) ».Ecole Nationale Supérieure de Techniques Avancées :

[13] Antoine MANZANERA. « Cours TERI – Master IAD ». UPMC Paris 6.

[14] Image Processing Toolbox. Version 2.2 (R11) 05-Aug-1998, Release information.

[16] Antoine MANZANERA. « Traitement d'images et vision artificielle ». Cours ENSTA .D9-2.

[17] Olivier bernard. « Traitement d'images numérique ». Université INSA. LYON.

[18] David Filliat. «Navigation, perception et apprentissage pour la robotique». Habilitation à Diriger des Recherches, l'Université Pierre et Marie Curie, Paris, le 12 juillet 2011, [https://www.google.dz/david filliat les robotique mobile](https://www.google.dz/david%20filliat%20les%20robotique%20mobile).

[19] JUAN GABRIEL AVIÑA CERVANTES. « Navigation visuelle d'un robot mobile dans un environnement d'extérieure semi-structure ». Thèse Doctorat, de l'Institut National Polytechnique de Toulouse, le 15 Février 2005.