

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
وزارة التعليم العالي والبحث العلمي  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
جامعة سعد دحلب البليدة  
UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك  
FACULTE DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT D'ELECTRONIQUE

**MEMOIRE DE  
PROJET DE FIN D'ETUDES  
POUR L'OBTENTION DU DIPLÔME  
D'INGENIEUR D'ETAT EN ELECTRONIQUE**

Option : contrôle

**Suivi visuel d'un objet dans une scène**

**Proposé par:**

**Pro Mr:** AMMOUR.N

**Co-pro Mr:** BENSLAMA.Z

**Présenté par :**

SLIMANI Mohammed

ABDELAZIZ Mohammed

**N° : CT....**

**Session: septembre 2009**



*A mes parents*

*A mes frères*

*A mes sœurs*

*A tous mes amis*

*SLIMANI Mohammed*

*A mes parents*

*A mes frères*

*A mes sœurs*

*A tous mes amis*

*ABDELAZIZ Mohammed*

# Remerciements

En premier lieu, nous remercions chaleureusement Monsieur AMMOUR Nassim pour avoir accepté d'encadrer ce travail, et pour nous avoir proposé un sujet passionnant. Nous remercions également Monsieur BENSLAMA.Z pour son aide. Nous remercions aussi tous ceux qui ont participé de près ou de loin à l'achèvement de ce travail.



# Résumé

La gestion d'une chaîne de production basée sur la vision industrielle est de plus en plus répandue. A l'heure actuelle, on retrouve des caméras dans tous les secteurs d'activité : la surveillance, la médecine, la production. La plupart des industries utilisent ces systèmes dans le but d'automatiser ces actions. Cette automatisation nécessite l'utilisation de systèmes capables de capturer une image, de l'analyser et d'en tirer une information permettant une action.

# Abstract

The management of a production chain based on industrial vision is more and more widespread. On present hour, we find cameras in all sectors of activity: the surveillance, medicine, the production. Most industries use these systems in the goal to automate these actions. This automation requires the use of systems capable to capture a picture, to analyze it and to extract information permitting an action of it.

## ملخص

إن استعمال الرؤية الاصطناعية في آلات الإنتاج من الأمور المنتشرة أكثر فأكثر. في الوقت الحالي، نجد آلات التصوير في شتى المجالات: المراقبة، الطب، التصنيع. أغلب المصانع تستعمل هذه التقنيات من أجل التحكم الذاتي في جميع مراحل التصنيع. هذا التحكم الذاتي يتطلب استعمال أنظمة قادرة على التقاط الصورة، تحليلها و استخراج معلومات منها، تستعمل في ضبط هذه المراحل.



# Table des matières

<b>Remerciements</b>	<b>iv</b>
<b>Résumé</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>ملخص</b>	<b>viii</b>
<b>Table des matières</b>	<b>x</b>
<b>Liste des figures</b>	<b>xii</b>
<b>Liste des tableaux</b>	<b>xiii</b>
<b>Introduction générale</b>	<b>1</b>
<b>I Robotique industrielle</b>	<b>4</b>
I.1 Introduction .....	4
I.2 Présentation .....	4
I.2.1 Définition.....	5
I.2.2 Domaines d'utilisation .....	6
I.3 Types de robots industriels .....	6
I.3.1 Les robots polaires.....	6
I.3.2 Les robots cartésiens.....	8
I.4 Fonctionnement .....	8
I.5 Conclusion.....	9
<b>II Traitement d'image</b>	<b>11</b>
II.1 Introduction .....	11
II.2 Définition du traitement d'image .....	11
II.3 Types de données manipulées .....	13
II.4 Acquisition d'une image .....	14
II.5 Opérateurs de traitement d'images.....	14
II.5.1 Opérateurs point à point .....	16
II.5.2 Opérateurs locaux .....	17
II.5.3 Filtres linéaires .....	19
II.5.4 Opérateurs morphologiques.....	22
II.6 Construction d'une application de traitement d'images .....	23
II.7 Reconnaissance d'objets .....	23
II.8 Quelques exemples concrets de traitement d'images.....	24

II.9	Segmentation d'image.....	26
II.9.1	Introduction .....	26
II.9.2	Segmentation par approche « région » .....	27
II.9.3	Segmentation par approche « frontière ».....	28
II.9.4	Segmentation par classification ou seuillage.....	29
II.9.5	Segmentation par la méthode des k-means .....	29
II.10	Conclusion.....	34
<b>III</b>	<b>Réalisation pratique</b>	<b>36</b>
III.1	Introduction .....	36
III.2	Présentation du système .....	37
III.3	Le schéma synoptique du système.....	38
III.4	La partie logicielle .....	39
III.4.1	Introduction .....	39
III.4.2	L'algorithme de segmentation.....	39
III.4.3	Définition du centre d'objet à suivre .....	42
III.4.4	L'algorithme du suivi .....	43
III.4.5	Génération des signaux de commande .....	46
III.4.6	Conclusion.....	47
III.5	La partie électronique .....	47
III.5.1	Introduction .....	47
III.5.2	La carte de commande.....	48
III.5.3	La carte de l'isolation optique .....	51
III.5.4	La carte de puissance.....	52
III.5.5	Conclusion.....	53
III.6	Testes et essais.....	53
III.7	Conclusion.....	54
	<b>Conclusion générale</b>	<b>56</b>
	<b>Bibliographie</b>	<b>59</b>

# Liste des figures

Figure I.1 Bras manipulateur .....	7
Figure I.2 Robot cartésien .....	8
Figure III.1 Le système du suivi visuel .....	37
Figure III.2 Le schéma synoptique du système .....	38
Figure III.3 L'organigramme de k-means .....	40
Figure III.4 Image captée avec la webcam .....	41
Figure III.5 Image segmentée avec k-means avec $k=2$ .....	41
Figure III.6 Centre de gravité de l'objet à suivre .....	42
Figure III.7 L'organigramme de la définition du centre d'objet .....	43
Figure III.8 La distance entre les axes à superposer .....	44
Figure III.9 La bande morte B .....	45
Figure III.10 L'organigramme du suivi .....	46
Figure III.11 Le schéma développé de la carte de commande .....	48
Figure III.12 L'organigramme de l'exécutable qui réside dans la mémoire du pic .....	49
Figure III.13 Le schéma développé de la carte d'isolation optique .....	51
Figure III.14 Le schéma développé de la carte de puissance .....	52
Figure III.15 L'objet à l'état initial .....	53
Figure III.16 L'objet après mouvement .....	53
Figure III.17 Superposition des deux axes .....	54

# Liste des tableaux

Tableau II.1 La matrice contour .....	21
Tableau III.1 Table de vérité des signaux de commandes.....	47
Tableau III.2 Table de vérité des signaux envoyés à la carte de puissance.....	50
Tableau III.3 Table de vérité de l'état du moteur en fonction des entrées.....	52



# Introduction générale

Pour manipuler des objets dans le domaine industriel, on utilise souvent la technique de détection de présence (capteurs infrarouges, capteurs ultrasons, lasers...). Ceci est valable dans des cas simples, où l'objet peut prendre des positions connus à l'avance. Dans les cas compliqués, l'objet peut se trouver n'importe où dans une scène ; donc l'utilisation de cette technique devienne difficile, et même impossible dans les cas les plus compliqués. D'autres techniques basées sur la vision artificielle sont développées, pour garantir une bonne manipulation des objets, et avec une très grande précision.

L'objectif de ce travail, est de présenter une technique basée sur la vision artificielle, qui permet le suivi visuel d'un objet dans une scène à deux dimensions. On doit donc réaliser un système capable de capter l'image de la scène avec une webcam, de reconnaître l'objet et son environnement en utilisant un ordinateur, et de superposer le centre de cet objet et l'axe passant par le centre du champ de vision du webcam, en contrôlant le sens de rotation du moteur qui porte cette webcam, à l'aide d'une interface port parallèle. Le centre du champ de vision de la webcam peut se déplacer seulement suivant l'axe horizontal. On ne peut pas balayer toute la surface de la scène, donc on s'intéresse seulement à l'axe vertical qui peut changer de position.

Tout ça doit être fait en temps réel. En plus de ça, il faut aussi afficher la vidéo de la scène avec des axes imaginaires (l'axe vertical qui passe par le centre du champs de vision de la webcam, et les axes qui passent par le centre de l'objet), pour bien visualiser la superposition.

La thèse est articulée autour des chapitres suivants :

Dans le premier chapitre nous rappelons les notions essentielles sur la robotique industrielle. Dans le chapitre suivant, nous abordons des généralités sur le traitement et la segmentation d'image. Le troisième chapitre concerne plus spécifiquement notre travail, c'est la réalisation pratique (les algorithmes que nous avons développé et les circuits électroniques fabriqués). Finalement on va terminer avec une conclusion générale.

Dans cette introduction générale, nous avons vu l'objectif de notre projet, les tâches à réaliser, ainsi que la structure de la thèse. Maintenant on va attaquer le premier chapitre. Comme on a dit précédemment, dans ce chapitre, on va présenter le domaine de la robotique industrielle.



# Chapitre 1

## Robotique industrielle

### **I.1 Introduction**

Dans ce chapitre, on va présenter le domaine de la robotique industrielle, les différents types des robots industriels et leur fonctionnement. On va citer aussi quelques exemples d'utilisation.

### **I.2 Présentation**

La robotique industrielle est officiellement définie par l'ISO comme un contrôle automatique, reprogrammable, polyvalent manipulateur programmable dans trois ou plusieurs axes. Les applications typiques incluent des robots de soudage, de peinture et d'assemblage. La robotique industrielle inspecte les produits, rapidement et précisément.

Les robots industriels sont beaucoup utilisés en automobile. Leur conception nécessite une bonne connaissance et un très haut niveau dans le domaine de l'ingénierie. [4]

### **I.2.1 Définition**

Un robot industriel est un système polyarticulé à l'image d'un bras humain souvent composé de six degrés de liberté, trois axes destinés au positionnement et trois axes à l'orientation permettant de déplacer et d'orienter un outil (organe effecteur) dans un espace de travail donné. [4]

Un robot se compose d'une partie mécanique, le bras lui même, d'une armoire de commande composée d'une unité centrale qui pilote les électroniques de commande d'un ou plusieurs axes qui en assure l'asservissement, de variateurs de vitesse et d'un langage de programmation spécialisé qui permet de commander le robot qui intègre un transformateur de coordonnées pour transformer une valeur cartésienne en données codeur du moteur. [4]

Certains robots disposent d'un mode d'apprentissage qui permet de répéter les mouvements réalisés librement à la main, l'élément essentiel étant la fidélité<sup>1</sup> (la capacité du robot à atteindre successivement la même position dans une tolérance définie), une procédure de calibration permet de reprendre le zéro de chacun des axes. Ils peuvent être associés à un système de vision artificielle qui leur permet de corriger les déplacements. [4]

Pour des raisons de sécurité, ces robots sont protégés par des cages ou des carters pour interdire à l'homme de les approcher de trop près.

---

<sup>1</sup> Certains auteurs parlent aussi de répétabilité

## **I.2.2 Domaines d'utilisation**

Les robots industriels ont d'abord été développés pour intervenir dans les milieux à risques (nucléaire, forte corrosion...), Ils servent aussi beaucoup dans le maniement d'objets lourds, et Le petit assemblage de précision sur des petites séries. Certains robots sont programmés pour exécuter fidèlement des actions spécifiques répétitives. Ils sont programmés avec un haut degré de précision. D'autres robots sont beaucoup plus flexibles. Ils sont par exemple utilisés en peinture. L'intelligence artificielle est en passe de devenir un facteur important dans la robotique industrielle. [4]

## **I.3 Types de robots industriels**

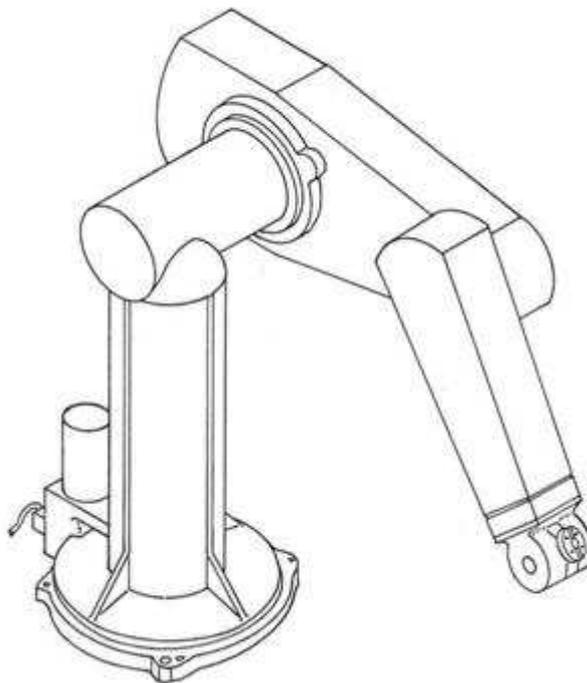
### **I.3.1 Les robots polaires**

On appelle robot polaire les robots ayant uniquement des articulations de type rotoïde. Pour pouvoir déplacer et orienter l'organe effecteur dans toutes les directions en 3D, un tel robot a besoin de six axes : trois pour le déplacement, trois pour l'orientation. Dans un environnement à deux dimensions, il suffit de trois axes : deux pour le déplacement, un pour l'orientation. [4]

#### **I.3.1.1 Les bras manipulateurs**

Pour faire face à la compétition grandissante, les industries ont de plus en plus recours à l'automatisation pour faire diminuer les prix des produits tout en augmentant Leur qualité. Les bras manipulateurs occupent une place importante dans le processus de production automatisée. Tout en étant très flexible, ils peuvent exécuter une variété de tâches avec régularité et rapidité. [4]

Un bras manipulateur est le bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce manipulateur sont reliés par des axes permettant, soit du mouvement de rotation (comme dans un robot articulé) ou de translation (linéaire) de déplacement. Il peut être autonome ou contrôlé manuellement et peut être utilisé pour effectuer une variété de tâches avec une grande précision. Les bras manipulateurs peuvent être fixes ou mobiles (c'est-à-dire à roues) et peuvent être conçus pour des applications industrielles. On les utilise par exemple dans les robots d'assemblage, de soudure, robots manipulateurs... [4]



**Figure I.1 Bras manipulateur**

### I.3.2 Les robots cartésiens

On appelle robot cartésien les robots ayant des articulations de type prismatique pour le déplacement de l'outil, mais forcément trois rotoïdes pour l'orientation de celui-ci. Pour pouvoir déplacer et orienter l'organe effecteur dans toutes les directions en 3D, un tel robot a besoin de six axes : trois prismatiques pour le déplacement, trois rotoïdes pour l'orientation. Dans un environnement à deux dimensions, il suffit de trois axes : deux pour le déplacement, un pour l'orientation. [4]

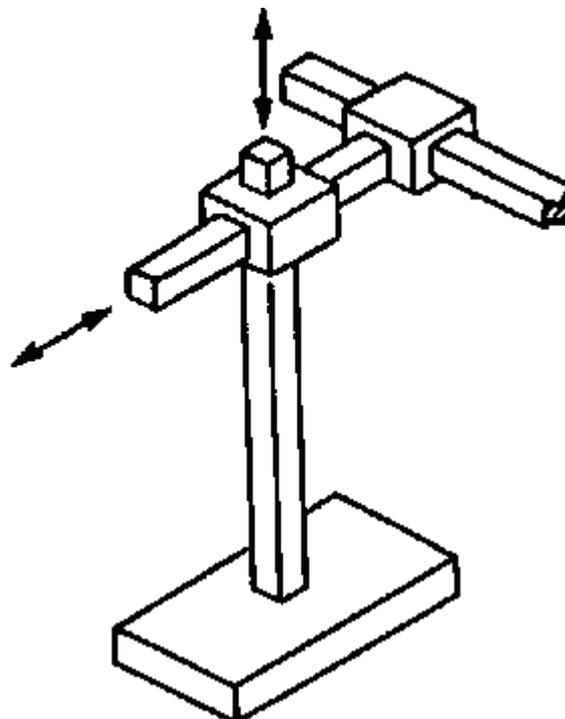


Figure I.2 Robot cartésien

### I.4 Fonctionnement

Certains robots utilisent des moteurs électriques, d'autres utilisent des vérins hydrauliques. Les premiers sont plus rapides, ces derniers sont plus forts et avantageux dans des

applications telles que la pulvérisation de peinture. On les utilise par exemple dans les tables traçantes, les robots de chargement... [4]

## **I.5 Conclusion**

Dans ce chapitre, nous avons présenté le domaine de la robotique industrielle, les différents types des robots industriels ainsi que quelques exemples d'utilisation. Dans le chapitre suivant on va entrer dans un autre domaine, c'est le domaine du traitement d'image.



## Chapitre 2

# Traitement d'image

### II.1 Introduction

Dans ce chapitre on va traiter des notions de base sur le traitement d'image, les types de données manipulés, l'acquisition d'une image, les opérateurs du traitement d'image, la reconnaissance d'objet et quelques exemples concrets de traitement d'image. Ensuite on va étudier la segmentation d'image. Finalement on va présenter la segmentation à l'aide de l'algorithme k-means que nous avons choisi pour notre projet.

### II.2 Définition du traitement d'image

Le traitement d'images désigne une discipline des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en

extraire de l'information. Il s'agit donc d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelles). [2]

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du domaine). [3]

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant : [2]

- La résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions,
- Les réglages optiques utilisés, (dont la mise au point) qui déterminent par exemple la netteté de l'image,

- Les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées,
- Le bruit de la chaîne de transmission d'image.

Quelques exemples types d'informations qu'il est possible d'obtenir d'une image numérique : [2]

- La luminance moyenne
- Le contraste moyen
- La couleur prédominante
- Le taux d'acuité moyen (précis ou flou)
- Le taux d'uniformité des couleurs
- La présence ou l'absence de certains objets

### II.3 Types de données manipulées

Le traiteur d'image dispose principalement d'images numériques, donc échantillonnées. Il dispose également de données intermédiaires de diverses natures : cartes de régions, listes de points connexes, tableaux de valeurs mesurées, etc. En ce qui concerne les images proprement dites, elles sont vues comme des fonctions de  $\mathbb{Z} \times \mathbb{Z}$  dans  $\mathbb{Z}^n$  ( $\mathbb{Z}$  représente

les entiers relatifs et  $n$  le nombre de composantes de l'image, 1 pour du niveau de gris, 3 pour RGB, plus pour les images spectrales). La représentation la plus utilisée est un tableau à plusieurs dimensions (représentant les dimensions spatiales de l'image), dans lequel les valeurs ont une sémantique dépendant du type de signal qu'elles codent (l'intensité lumineuse du point, la distance à un point de référence, ou le numéro de la région d'appartenance par exemple). [2]

## II.4 Acquisition d'une image

L'étude de cette étape passe inmanquablement par le système d'acquisition qui fait référence : l'œil. On peut utiliser des webcams, des appareils photo numériques, des caméras industrielles, des caméras infrarouges... En médecine, on utilise des imageurs IRM, TEP, scanner X, échographie Doppler, échographie, scintigraphie, Tomographie, etc. Tous ces systèmes peuvent être comparés à des capteurs. Il ne faut pas oublier qu'il y a une étape de conversion analogique/numérique. C'est souvent cette étape qui limite la résolution de l'image. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'intercorrélacion de deux éléments voisins : plus cette intercorrélacion est faible, meilleure est l'image. [2]

## II.5 Opérateurs de traitement d'images

Par analogie avec les opérateurs mathématiques, on appelle opérateurs de traitement d'images des traitements plus ou moins complexes prenant en entrée une image ou un ensemble d'informations relatif à une image, et produisant une image ou un ensemble d'informations relatif aux données initiales. [2]

On classe généralement les opérateurs en différentes familles, en fonction des informations qu'ils acceptent en entrée et qu'ils fournissent en sortie, et en fonction des transformations qu'ils font subir aux données. Ainsi, par exemple, on distingue (cette liste est loin d'être exhaustive) : [2]

Opérateurs image  $\rightarrow$  image :

- opérateurs de modifications pixel à pixel (aussi appelés opérateurs point à point) : changement de la dynamique de l'image, opérateurs binaires pixel à pixel (et, ou, xor, etc.) ;
- opérateurs locaux (traitent les pixels en fonction de leur voisinage) : opérateurs de flou, opérateurs morphologiques (érosion, dilatation, squelette), opérateurs de détection de contours ;
- opérateurs dans l'espace fréquentiel : opérateurs de réduction du bruit, filtres passe-bande (souvent utilisés en première approche pour améliorer l'image, on les appelle alors des opérateurs de prétraitement) ;
- opérateurs globaux : calcul des distances.

Opérateurs image  $\rightarrow$  ensemble d'informations :

- opérateurs de segmentation en frontières, en régions ;

- opérateurs de classification de pixels ;
- opérateurs de calcul de paramètres.

Opérateurs ensemble d'informations → image

- constructeurs d'image à partir d'une carte de régions ou d'une liste de frontières.

Les parties suivantes s'attachent à détailler les différents opérateurs et leurs applications habituelles, puis à présenter la manière dont ils sont combinés pour construire une application de traitement d'images.

### **II.5.1 Opérateurs point à point**

Cette amélioration peut servir dans un premier temps à faciliter la visualisation de l'image sur un écran d'ordinateur. Les capacités de vision de l'être humain étant limitées, il est indispensable d'adapter la dynamique de l'image à notre vision. On parle souvent de Lookup Table ou LUT, que l'on retrouve également dans les FPGA. Il s'agit de l'opérateur le plus simple qu'on puisse trouver puisqu'en chaque pixel de l'image on modifie le niveau de gris à l'aide d'une fonction. Ainsi, pour éclaircir une image, on applique la fonction  $\log$  () à chaque niveau de gris. Au contraire pour rendre plus sombre une image un peu trop saturée, on applique une fonction exponentielle. On peut remarquer que le seuillage (binaire) n'est rien d'autre qu'une table d'affichage particulière, celle qui associe le noir à tous les niveaux inférieurs à un certain seuil et le blanc à tous les autres. Il s'agit d'un opérateur très simple et particulièrement utilisé mais qui cache une grande difficulté, trouver le seuil adéquat et de manière automatique ! Ces opérations point à point,

qualifiées ainsi car elles ne travaillent que sur un pixel (et non sur un voisinage), ont un effet bien limité. En présence de bruit elles ne sont d'aucune utilité. [2]

## II.5.2 Opérateurs locaux

Il faut alors utiliser des opérateurs de traitement plus complexes scindés bien souvent en deux sous-catégories : [2]

- les opérateurs linéaires.
- les filtres non linéaires.

La première sous-catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent). [2]

La deuxième sous-catégorie comprend le domaine de la morphologie mathématique, ainsi que d'autres traitements comme les détecteurs de points caractéristiques, l'opérateur de Di-Zenzo (détecteur de contour généralisé au cas couleur), le filtre Retinex, ainsi que les opérateurs homomorphiques (ceux qui travaillent sur le log de l'image), mais aussi tous les opérateurs permettant d'extraire par exemple des informations sur la texture de l'image (matrice de cooccurrence, indice fractal, longueur de plage...). [2]

On a souvent l'habitude de voir un détecteur de contours s'appliquer après un filtre linéaire passe-bas... qui rend l'image floue ! La plupart du temps il faut combiner astucieusement

filtre non linéaire et filtre linéaire afin de détecter ce que l'on souhaite tout en faisant abstraction du bruit. Une fois le bruit éliminé et l'image restaurée afin de compenser les déformations introduites par le milieu de transmission et l'optique d'acquisition, on peut passer à l'étape de segmentation qui doit permettre de réaliser une partition de l'image en ensembles connexes homogènes. [2]

Il existe deux grandes catégories de segmentations : [2]

- la segmentation de région
  
- la segmentation de contour ; on se trouve alors confronté à un problème de représentation du résultat par des primitives simples.

La segmentation orientée contour connaît de nombreux progrès autour de l'utilisation de contours actifs ou des ensembles de niveaux. L'introduction d'aspects probabilistes (chaîne de Markov et champs de Markov) a permis de travailler en réduisant la connaissance a priori nécessaire pour obtenir un traitement satisfaisant. Dans cette étape on retrouve souvent une partie de classification des pixels en classes. On essaye de regrouper au sein d'un même ensemble, aussi appelé classe, les pixels présentant une même caractéristique : niveau de gris compris dans un certain intervalle ou dérivée seconde supérieure à un certain seuil. [2]

## II.5.3 Filtres linéaires

### II.5.3.1 Généralités

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution. Le temps de calcul est souvent réduit lorsqu'on veut séparer un filtre en deux filtres dont la convolution mutuelle permet de le reconstituer. Cette remarque est utilisée en particulier pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical. [1]

### II.5.3.2 Lissage

Ceux-ci sont des filtres passe-bas qui coupent plus ou moins les plus hautes fréquences. Ils sont utilisés pour atténuer les bruits d'origines les plus diverses qui polluent l'information, en particulier dans la détection de contours considérée ci-après. Techniquement, il s'agit de traductions discrètes de filtres continus qui, comme ceux-ci, ne modifient pas le niveau global du signal. Les termes de la matrice de convolution sont donc généralement des entiers à diviser par leur somme. [1]

- Filtre uniforme : Il est obtenu par convolution de deux filtres unidimensionnels rectangulaires. Toutes les composantes de la matrice ont la même valeur. L'imperfection de ce filtre réside dans le fait qu'il introduit des déphasages.

- Filtre pyramidal : La convolution d'un filtre rectangulaire avec lui-même conduit à un filtre triangulaire grâce auquel les phases ne sont plus modifiées. Le filtre pyramidal est obtenu à partir de filtres triangulaires dans les deux directions.
  
- Filtre gaussien : Ce filtre très populaire utilise la loi de probabilité de Gauss (voir Loi normale multidimensionnelle). Des approximations de plus en plus précises peuvent être obtenues, selon le Théorème de la limite centrale par itération de l'un des filtres précédents.

Le filtre gaussien est utilisé comme constituant du masque flou qui améliore la netteté apparente des photographies numériques. Bien qu'il soit popularisé par la photographie plus ou moins artistique, il est également utilisé dans certaines techniques, comme l'astronomie. [1]

### **II.5.3.3 Détection de contours**

Dans une image en niveaux de gris, un contour est caractérisé par un changement brutal de la valeur. Le but de l'opération est de transformer cette image en une autre dans laquelle les contours apparaissent par convention en blanc sur fond noir. Dans une section horizontale (ou verticale) de l'image rectangulaire, les variations de la valeur sont décrites par une courbe. Sur celle-ci un point d'un contour est associé à un maximum de la pente, c'est-à-dire à un extremum (maximum ou minimum) de la dérivée première. Cet extremum peut aussi s'interpréter comme un zéro de la dérivée seconde. Dans une image numérisée, à chaque pixel est associée une valeur qui est en général différente de la valeur des pixels voisins. La notion de dérivée correspondant à une variation infiniment petite doit donc être

remplacée par l'approximation différence finie utilisée en calcul numérique. Le problème est simplifié car on ne s'intéresse ici qu'aux comparaisons entre dérivées indépendamment de leurs valeurs. Ainsi la dérivée première au niveau d'un pixel se réduit à la différence entre les valeurs des deux pixels voisins, la dérivée seconde étant la différence entre les dérivées moyennées aux frontières des pixels. [1]

	N	
W	C	E
	S	

**Tableau II.1 La matrice contour**

Si on considère maintenant l'image rectangulaire, les dérivées premières dans les deux directions perpendiculaires s'écrivent pour le pixel centre en fonction des valeurs des pixels aux quatre points cardinaux :

$$D_x = E - W \quad (\text{II.1})$$

$$D_y = N - S \quad (\text{II.2})$$

Les dérivées secondes sont données par :

$$D_{2x} = E - 2C + W \quad (\text{II.3})$$

$$D_{2y} = N - 2C + S \quad (\text{II.4})$$

Enfin, il faut remarquer qu'en considérant classiquement les fonctions comme des sommes de sinusoides, la dérivation multiplie leur amplitude par leur pulsation : c'est donc un filtre passe-haut qui introduit un bruit haute fréquence à l'origine de faux contours. Les dérivées ne sont donc en principe pas utilisées telles quelles mais associées à des filtres passe-bas). L'utilisateur peut effectuer lui-même un prétraitement avec un filtre gaussien par exemple. Des méthodes plus élaborées incluent ce prétraitement dans le filtre. [1]

A priori, l'utilisation des dérivées secondes est plus commode que celle des dérivées premières : au lieu de chercher les maximums de l'intensité du gradient, on cherche l'annulation du laplacien, somme des dérivées secondes dans deux directions, soit  $N + S + E + W - 4C$ . La contrepartie de la simplicité est l'amplification du bruit par la seconde dérivation. Ce bruit est atténué par l'adjonction d'un filtre gaussien mais les méthodes du second ordre ne semblent pas compétitives avec celles du premier ordre évoquées précédemment. [1]

#### **II.5.4 Opérateurs morphologiques**

La morphologie mathématique offre des opérateurs non linéaires particulièrement utiles pour filtrer, segmenter et quantifier des images. Initialement destinée au traitement des images binaires, elle a très vite été généralisée aux images à niveaux de gris, puis aux images en couleurs et multi-spectrales. La nature des opérateurs morphologiques fait qu'ils se prêtent bien au développement de circuits électroniques spécialisés (ou bien à l'utilisation de FPGA) dans les opérateurs morphologiques. [2]

## II.6 Construction d'une application de traitement d'images

Les objectifs des applications peuvent être de différentes natures : [2]

- détecter la présence d'un objet ou son absence.
- calculer les caractéristiques d'un ou de plusieurs éléments de l'image.

Dans tous les cas, l'idée est, en partant d'une image initiale, d'en extraire des informations. Pour cela, on va utiliser les opérateurs à la manière de briques logicielles, en les combinant et en les enchaînant. Ces techniques sont la base des systèmes de vision industrielle. [2]

## II.7 Reconnaissance d'objets

La reconnaissance d'objets est une branche de la vision artificielle et un des piliers de la vision industrielle. Elle consiste à identifier des formes pré-décrites dans une image numérique, et par extension dans un flux vidéo numérique. Attention à ne pas confondre reconnaissance d'objets (object recognition ou shape recognition en anglais) et reconnaissance de formes (pattern recognition en anglais). La première s'attache à reconnaître des formes géométriques dans une image, alors que la seconde cherche à identifier des motifs dans des données statistiques. La confusion vient du fait qu'on utilise souvent la reconnaissance de formes comme technique appliquée à la reconnaissance d'objets. [2]

## II.8 Quelques exemples concrets de traitement d'images

- Contrôle de présence/absence. Sur des chaînes de production, on vérifie en bout de chaîne avec une caméra vidéo la présence d'une pièce dans un ensemble plus complexe. Pour cela bien souvent il suffit de faire un simple seuillage dans une région spécifique.
- Contrôle du niveau de maturation des fruits sur une chaîne de conditionnement. Il s'agit de reconnaître à la couleur et à la texture du fruit son degré de maturité et donc la catégorie sous laquelle il sera emballé puis vendu.
- Construction et correction de cartes géographiques d'après des images satellites ou des images aériennes. On recale d'après des informations topographiques les images reçues, puis on les met sur la carte en correspondance avec les informations trouvées dans l'image : voies de communication, voies et plans d'eau, parcelles agricoles...
- Surveillance et évaluation de la production agricole. Il est possible de déterminer le degré de maturation des cultures, la quantité d'eau nécessaire pour l'irrigation, le rendement moyen... On peut ainsi établir des prévisions à large échelle de la récolte à venir.
- Reconnaissance de l'écriture. La reconnaissance de l'écriture manuscrite progresse de jour en jour. Elle est suffisamment opérationnelle pour que la majorité des

adresses, même manuscrites, soient reconnues automatiquement sur le courrier postal.

- Recherche d'image par le contenu. L'objectif de cette technique est de rechercher, parmi une base de données d'images, les images similaires à une image exemple, ou ayant certaines caractéristiques, par exemple rechercher toutes les images comportant un vélo.
- Analyse de la vidéo. L'objectif de cette technique devenue une discipline depuis les années 2000 (lorsque la puissance des processeurs peu onéreux et en particulier des PC a permis des traitements puissants en temps réel) est d'interpréter les faits observés à l'image afin de signaler ou d'enregistrer des faits marquants. Le plus souvent, la caméra est fixe et observe les mouvements d'une scène. Les applications sont nombreuses : Protection des biens (détection d'intrusion, détection d'objet abandonné ou déposé...), Identification (biométrie faciale), Sécurité des personnes (détection de chutes de personnes....), Animations (planchers animés selon les mouvements des danseurs en boîte de nuit), Détection de feux (industriel, forêts, tunnels, ...), Surveillance de tunnels (comptage, mesure de vitesse, détection de fuites/anomalies dans les plafonds), surveillance de tuyaux et autres applications industriels, ...

## II.9 Segmentation d'image

### II.9.1 Introduction

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond. Si le nombre de classes est égal à deux, elle est appelée aussi binarisation. [2]

Si l'homme sait naturellement séparer des objets dans une image c'est grâce à des connaissances de haut niveau (compréhension des objets et de la scène). Mettre au point des algorithmes de segmentation de haut niveau (chaque région est un objet sémantique) est encore un des thèmes de recherche les plus courants en traitement d'images. [2]

La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes : [2]

- Segmentation fondée sur les régions (en anglais : region-based segmentation). On y trouve par exemple : la croissance de région (en anglais : region-growing), décomposition/fusion (en anglais : split and merge)
  
- Segmentation fondée sur les contours (en anglais : edge-based segmentation)

- Segmentation fondée sur classification ou le seuillage des pixels en fonction de leur intensité (en anglais : classification ou thresholding)
- Segmentation fondée sur la coopération entre les trois premières segmentations

## II.9.2 Segmentation par approche « région »

Les méthodes appartenant à cette famille manipulent directement des régions. Soit elles partent d'une première partition de l'image, qui est ensuite modifiée en divisant ou regroupant des régions, et on parle alors de méthodes de type décomposition/fusion (ou split and merge en Anglais) ; soit elles partent de quelques régions, qui sont amenées à croître par incorporation de pixels jusqu'à ce que toute l'image soit couverte, et on parle alors de méthodes par croissance de régions. Des méthodes fondées sur la modélisation statistique conjointe de la régularité des régions et des niveaux de gris de chaque région existent également. [2]

Les algorithmes de type décomposition/fusion exploitent les caractéristiques propres de chaque région (surface, intensité lumineuse, colorimétrie, texture, etc.). On cherche des couples de régions candidates à une fusion et on les note en fonction de l'impact que cette fusion aurait sur l'apparence générale de l'image. On fusionne alors les couples de régions les mieux notés, et on réitère jusqu'à ce que les caractéristiques de l'image remplissent une condition prédéfinie : nombre de régions, luminosité, contraste ou texture générale donnée, ou alors jusqu'à ce que les meilleures notes attribuées aux couples de régions n'atteignent plus un certain seuil (dans ce dernier cas, on parle d'un algorithme avec minimisation de fonctionnelle). [2]

Les algorithmes par croissance de régions partent d'un premier ensemble de régions, qui peuvent être calculées automatiquement (par exemple, les minima de l'image), ou fournies par un utilisateur de manière interactive. Les régions grandissent ensuite par incorporation des pixels les plus similaires suivant un critère donné, tel que la différence entre le niveau de gris du pixel considéré et le niveau de gris moyen de la région. Les algorithmes de segmentation par ligne de partage des eaux, développés dans le cadre de la morphologie mathématique, appartiennent à cette catégorie. [2]

Les algorithmes fondés sur une modélisation statistiques conjointe des régions et des niveaux de gris, notamment ceux s'appuyant sur les Champs de Markov Cachés, reposent sur la minimisation d'une fonction de vraisemblance (ou énergie). Cette fonction prend simultanément en compte la vraisemblance de l'appartenance du pixel à une région considérant son niveau de gris, et les régions auxquelles appartiennent les pixels voisins. Cette fonction effectue un compromis entre la fidélité à l'image initiale et la régularité des régions segmentées. [2]

### **II.9.3 Segmentation par approche « frontière »**

Cette approche cherche à exploiter le fait qu'il existe une transition détectable entre deux régions connexes. Les méthodes les plus anciennes utilisent des opérateurs de traitement d'images, tels que le filtre de Canny, pour mettre en évidence les pixels qui semblent appartenir à un contour. La construction d'une partition est alors souvent difficile. On peut aussi faire intervenir des modèles déformables à l'aide de courbes paramétriques (courbe de Bézier, spline...) ou de polygones (par exemple algorithme à bulle). Pour initier le processus, on recherche des points remarquables de l'image, tels que des points à

l'intersection de trois segments au moins. De tels points sont appelés des graines (seeds en anglais). L'intérêt principal des méthodes de segmentation selon l'approche frontières est de minimiser le nombre d'opérations nécessaires en cas d'itération du processus sur des séries d'images peu différentes les unes des autres (cas des images vidéo notamment). En effet, une fois que les contours des régions ont été trouvés dans la première image, l'application du modèle déformable à l'image suivante est plus efficace que de tout recalculer, si la différence entre les images est peu importante. [2]

## **II.9.4 Segmentation par classification ou seuillage**

On part ici d'un rapport qu'entretient chaque pixel individuellement avec des informations calculées sur toute l'image, comme par exemple la moyenne des niveaux de gris de l'ensemble des pixels, ou la médiane, permettant de construire n classes d'intensité. Lorsque les classes sont déterminées par le choix d'un seuil, on parle de seuillage. Les pixels appartenant à une même classe et étant connexes forment des régions. [2]

## **II.9.5 Segmentation par la méthode des k-means**

### **II.9.5.1 Introduction**

La méthode des k-means est un outil de classification classique qui permet de répartir un ensemble de données en k classes homogènes. La plupart des images (photos, dessins vectoriels 2D, synthèses 3D, ...) vérifient localement des propriétés d'homogénéité, notamment en termes d'intensité lumineuse. L'algorithme des k-means permet donc d'apporter une solution à la segmentation d'images. [2]

Ici on va étudier la méthode k-means. On va voir les résultats dans la partie de la réalisation pratique.

### II.9.5.2 Présentation

La méthode des k-means a été introduite par J. McQueen en 1971 et mise en œuvre sous sa forme actuelle par E. Forgy. De nombreuses variantes se sont succédées depuis afin d'étendre ses capacités de classification (séparations non linéaires) : kernel k-means (k-means basée sur des méthodes à noyaux), améliorer ses performances : global-k-means, k-Harmonic means, automatiser le choix du nombre de clusters : Gaussian-means, x-means. Dans le cadre de la classification non supervisée, on cherche généralement à partitionner l'espace en classes concentrées et isolées les unes des autres. Dans cette optique, l'algorithme des k-means vise à minimiser la variance intra-classe, qui se traduit par la minimisation de l'énergie suivante : [2]

$$E = \frac{1}{2} \sum_{c \in C} \sum_{x \in c} \|x - m_c\|^2 \quad (\text{II.5})$$

$$E = \frac{1}{2} \sum_{c \in C} \# c \times V(c) \quad (\text{II.6})$$

$$E = \frac{1}{2} \sum_{x \in D} \min_{c \in C} \|x - m_c\|^2 \quad (\text{II.7})$$

Avec  $C$  l'ensemble des clusters et pour chaque cluster  $c$ ,  $m_c$  son centre (appelé noyau),

$V(c)$  sa variance,  $\#c$  le nombre de ses éléments et  $D = \bigcup_{c \in C} c$  l'ensemble des données que

l'on cherche à classer.

### II.9.5.3 L'algorithme k-means

La minimisation de l'énergie (II.5) peut se réaliser par une descente de gradient sur les noyaux. Elle peut se traduire par les étapes suivantes : [2]

- 1) Initialisation des noyaux.
- 2) Mise à jour des clusters.
- 3) Réévaluation des noyaux.
- 4) Itérer les étapes 2) et 3) jusqu'à stabilisation des noyaux.

Cet algorithme présente ces étapes de façon plus formelle : [2]

On note  $\mathbb{E}(c)$  la fonction qui associe à un cluster  $c$  son centre de gravité défini par  $\frac{1}{\#c} \sum_{x \in c} x$ .

**Début**

// Initialisation

$\forall c \in \mathcal{C}$

$m_c \leftarrow x / x \in \mathcal{D}$  // Tirage uniforme dans la base de données

$c \leftarrow \{x \in \mathcal{D} / \min_{q \in \mathcal{C}} \|x - m_q\|^2 = \|x - m_c\|^2\}$

**Fin**  $\forall$

**Tant que**  $\exists c \in \mathcal{C} / m_c \neq \mathbb{E}(c)$

$\forall c \in \mathcal{C}$

$m_c \leftarrow \mathbb{E}(c)$

$c \leftarrow \emptyset$

**Fin**  $\forall$

$\forall x \in \mathcal{D}$

$\hat{c} \leftarrow \arg \min_{c \in \mathcal{C}} \|x - m_c\|^2$

$\hat{c} \leftarrow \hat{c} \cup x$

**Fin**  $\forall$

**Fin tant que**

**Fin**

#### II.9.5.4 Choix du nombre k de clusters

L'algorithme classique des k-means laisse un paramètre libre : le nombre de clusters, ce qui dans le cas de la segmentation d'images correspond au nombre d'intensités utilisées pour représenter l'image. Généralement le choix de k est fait empiriquement en sélectionnant la valeur de k qui minimise l'énergie (II.5). [2]

Différents critères permettent d'estimer le nombre de clusters en minimisant la distance intra-classes et à maximiser la distance inter-classes (par exemple en segmentation d'images couleurs). [2]

De nombreuses stratégies permettent de déterminer le nombre de clusters pendant le déroulement de l'algorithme. Ces méthodes sont basées sur un processus itératif au cours duquel on choisit de subdiviser les clusters précédemment établis en se basant sur un critère statistique (par exemple grâce aux BIC : Bayesian Information Criterion) : Gaussian-means et x-means. [2]

Une méthode spécialement adaptée à la segmentation d'images couleurs a été développée par L. Lucchese. Cette technique réalise une première classification par la méthode des k-means dans l'espace des chrominances UV, suivie de plusieurs classifications successives dans l'espace des luminances Y. [2]

#### **II.9.5.5 K-means et segmentation d'images**

Les méthodes de classification (clustering) issues de l'analyse de données permettent de regrouper des objets possédant des propriétés similaires. Elles constituent donc une approche naturelle pour réaliser une segmentation d'images. [2]

La méthode des k-means a été très utilisée dans ce contexte, d'une part pour sa simplicité de mise en œuvre et d'autre part car elle peut fournir une bonne approximation de la segmentation recherchée. Néanmoins cette méthode souffre d'un défaut qui a son importance en segmentation d'images : elle introduit des discontinuités spatiales assez fortes aux frontières des classes. Des méthodes de régularisation sont donc généralement employées pour renforcer la connexité et ainsi réduire le nombre de composantes connexes de chaque classe. [2]

## **II.10 Conclusion**

Dans ce chapitre on a présenté des notions de base sur le traitement d'image, ainsi que la segmentation à l'aide de l'algorithme k-means. Le chapitre suivant concerne la réalisation pratique, les différentes parties du projet (électronique, informatique,..) ainsi que les résultats obtenus.



# Chapitre 3

## Réalisation pratique

### III.1 Introduction

Dans ce chapitre on va présenter toutes les démarches à faire pour construire notre système. On va commencer avec une explication sur ses différentes parties, ensuite on va passer à la partie logicielle. Dans cette partie on va discuter les algorithmes développés pour l'acquisition, le traitement, l'analyse et l'extraction des informations des images, ainsi que la génération des signaux de commande, on va passer ensuite à la partie électronique. Dans cette partie on va détailler le fonctionnement de l'interface qui assure le contrôle du moteur à partir de l'ordinateur. Finalement on va donner les résultats de notre travail.

### III.2 Présentation du système

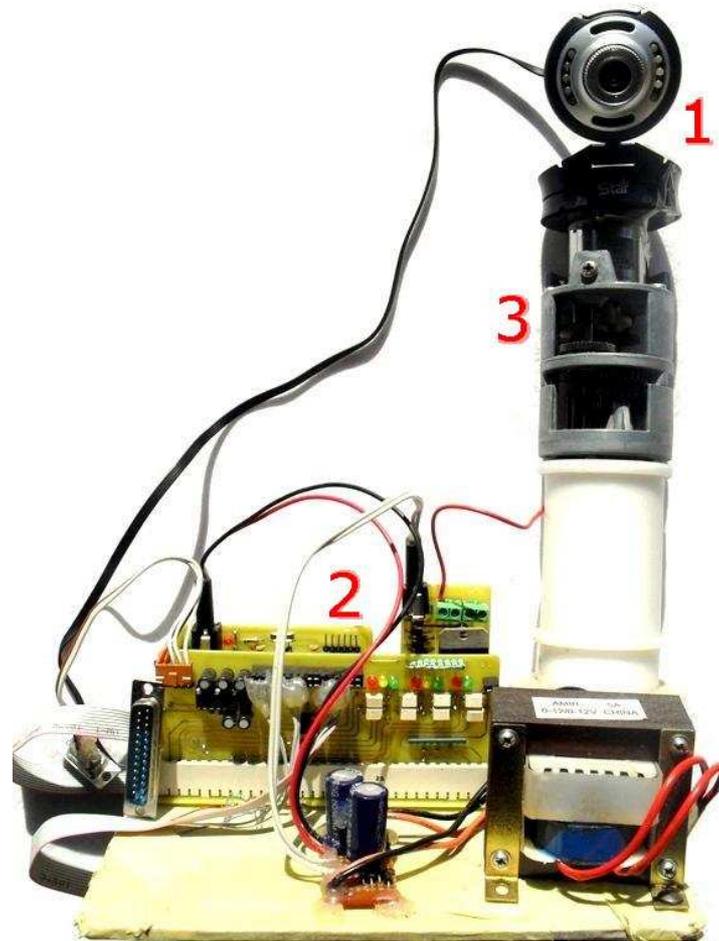


Figure III.1 Le système du suivi visuel

Voici les parties qui composent le système :

- 1) Webcam : utilisée pour l'acquisition d'image.
- 2) Interface de contrôle (partie électronique) : permet le contrôle du moteur à partir de l'ordinateur.
- 3) Moteur de rotation : permet le déplacement rotatif de la webcam.

### III.3 Le schéma synoptique du système

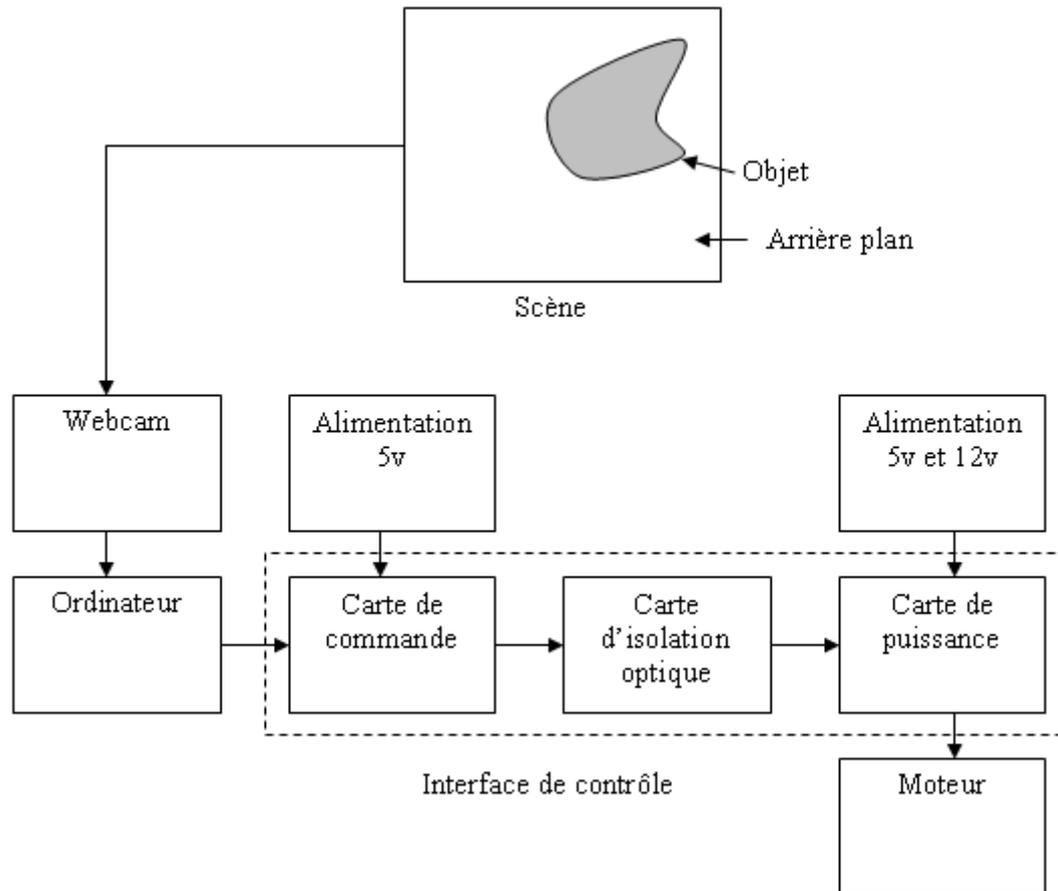


Figure III.2 Le schéma synoptique du système

L'acquisition d'image est faite avec la webcam. Cette image est transmise à l'ordinateur via le port USB. L'ordinateur fait le traitement et la segmentation de cette image, ensuite il détermine le centre de l'objet, et donne l'ordre via le port parallèle à la carte de commande. Cette carte va traduire les signaux envoyés par l'ordinateur pour donner les signaux de commande à la carte de puissance qui contrôle le moteur. Une carte intermédiaire appelé carte d'isolation optique permet la séparation des alimentations de la carte de commande et de la carte de puissance. On utilise deux alimentations différentes, une pour la carte de commande et l'autre pour la carte de puissance, pour éviter les bruits. Les trois cartes de

commande, d'isolation optique et de puissance forme ce qu'on appelle l'interface de contrôle.

Maintenant on va détailler les algorithmes développés pour le traitement et la segmentation de l'image de la scène, ainsi que la détection et la superposition des différents centres.

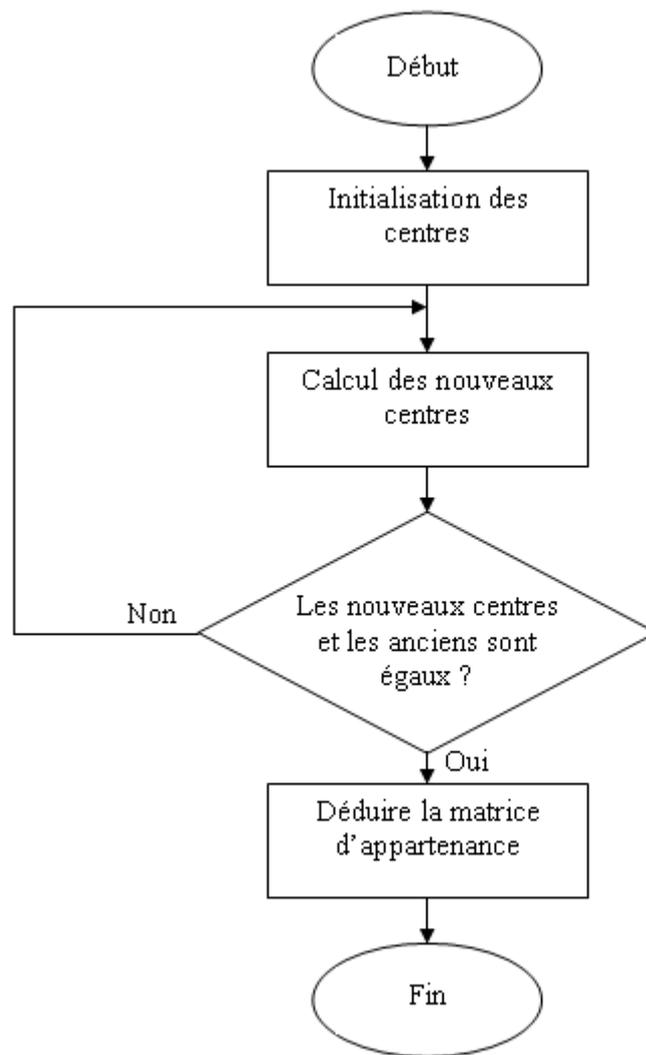
## **III.4 La partie logicielle**

### **III.4.1 Introduction**

Cette partie est divisée en trois sous parties. Dans la première sous partie on va expliquer l'algorithme utilisé dans la segmentation. Dans la deuxième sous partie on va expliquer l'algorithme utilisé dans la détection du centre de l'objet à suivre. Dans la dernière sous partie on va voir l'algorithme de suivi de l'objet, ainsi que comment générer les signaux de commande.

### **III.4.2 L'algorithme de segmentation**

On utilise l'algorithme k-means pour définir les deux classes de la scène (l'objet et l'arrière plan). Voici l'organigramme de k-means :



**Figure III.3 L'organigramme de k-means**

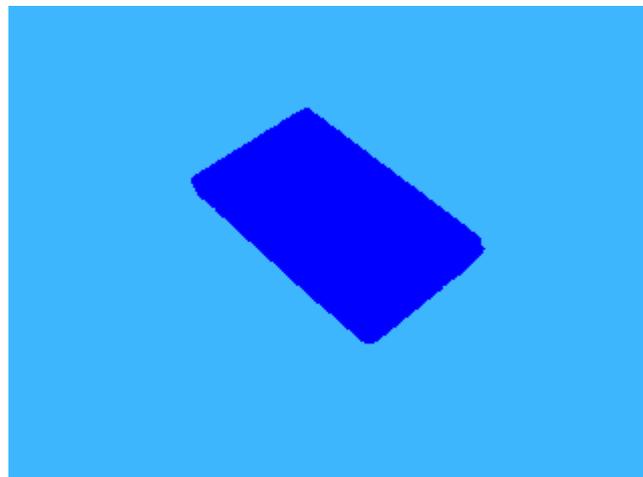
Au début il faut donner des valeurs initiales aux différents centres, ces valeurs doivent être différentes, ensuite il faut calculer les nouveaux centres, le calcul se fait comme suit :

On effectue un balayage de toutes les pixels, on calcul ensuite les distances euclidiennes entre le niveau de gris de chaque pixel et les centres, puis on affecte chaque pixel au centre le plus proche (distance euclidienne minimale). On calcule les nouveaux centres en

utilisant les formules présentés dans le chapitre précédent. On répète cette étape jusqu'à ce que les centres ne changent pas de valeur. Enfin on construit la matrice d'appartenance (matrice des étiquettes), qui contient les étiquettes des différentes classes.



**Figure III.4 Image captée avec la webcam**



**Figure III.5 Image segmentée avec k-means avec k=2**

### III.4.3 Définition du centre d'objet à suivre

On a utilisé une scène à deux objets seulement, l'objet à suivre et l'arrière plan. L'arrière plan est en blanc, mais l'objet peut prendre des couleurs différentes, à part la couleur blanche bien sûr. Donc les niveaux de gris des pixels appartenant à l'objet sont supérieurs aux niveaux de gris des pixels appartenant à l'arrière plan. Après qu'on a déduit la matrice d'appartenance dans l'étape de la segmentation, on l'utilise pour définir le centre de gravité de l'objet à suivre.

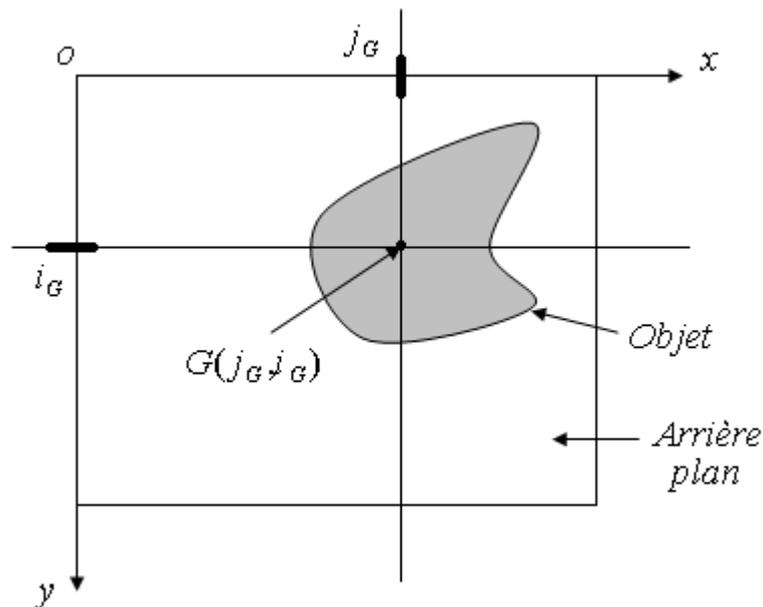


Figure III.6 Centre de gravité de l'objet à suivre

Voici l'organigramme de la méthode de définition du centre de gravité :

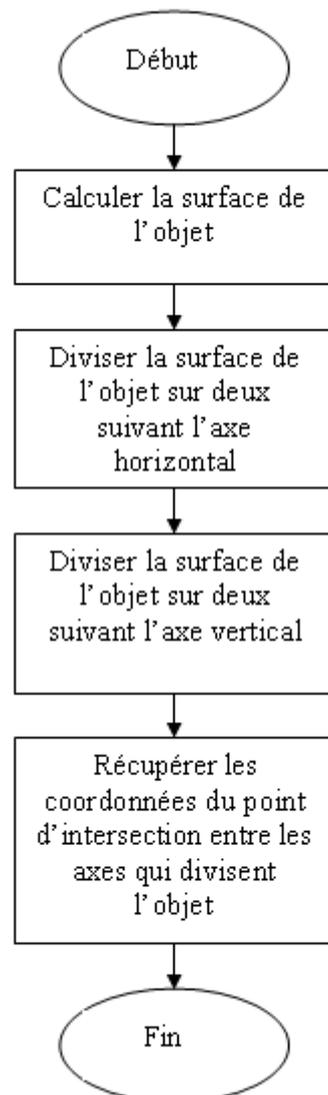


Figure III.7 L'organigramme de la définition du centre d'objet

#### III.4.4 L'algorithme du suivi

Après qu'on a récupéré le point  $G(j_G, i_G)$ , on va comparer la position de l'axe  $x = j_G$  et la

position de l'axe  $x = \frac{w}{2}$  où  $w$  est la largeur de l'image (scène entière).

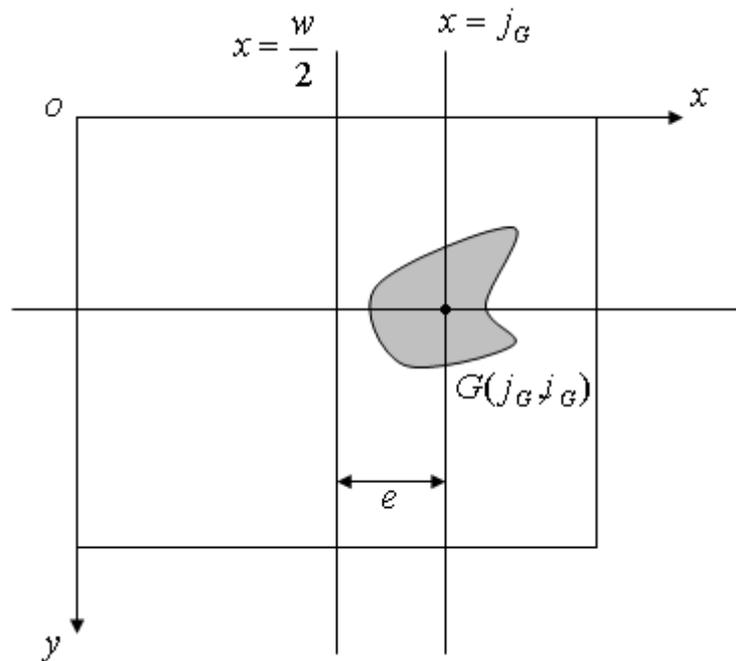


Figure III.8 La distance entre les axes à superposer

$$e = j_G - \frac{w}{2} \quad (\text{III.1})$$

Où  $e$  est l'erreur entre la position  $x = \frac{w}{2}$  et la position  $x = j_G$ . On cherche à minimiser cette erreur, elle doit être proche du zéro.

- Si  $e > 0$ , il faut tourner le moteur à gauche.
- Si  $e < 0$ , il faut tourner le moteur à droite
- Si  $e = 0$ , il faut arrêter le moteur (théoriquement)

On sait qu'on ne peut jamais arriver à  $e=0$ , donc le moteur va osciller autour de l'axe  $x = \frac{w}{2}$ . Pour éliminer ces oscillations on va utiliser une bande morte  $B$ .

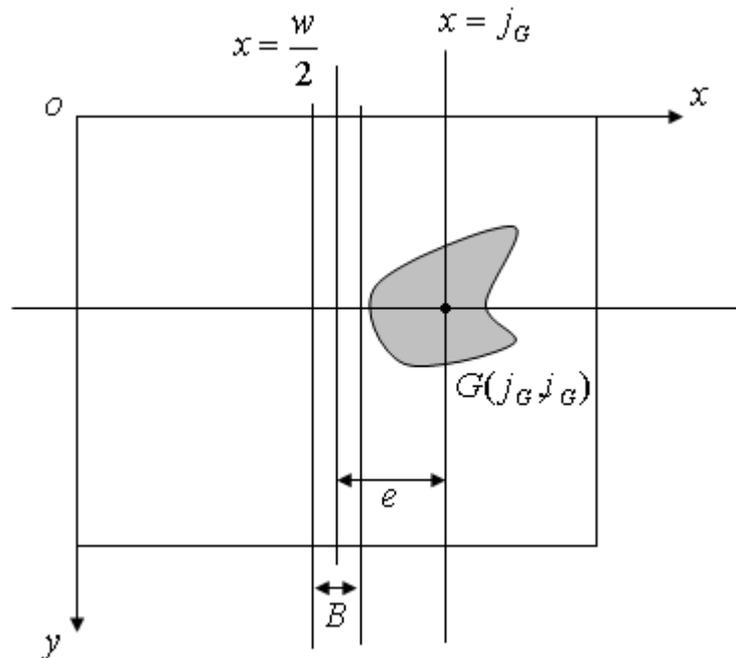


Figure III.9 La bande morte B

Donc :

- Si  $j_G > \frac{w}{2} + \frac{B}{2}$ , il faut tourner le moteur à gauche.
- Si  $j_G < \frac{w}{2} - \frac{B}{2}$ , il faut tourner le moteur à droite.
- Si  $\frac{w}{2} - \frac{B}{2} < j_G < \frac{w}{2} + \frac{B}{2}$ , il faut arrêter le moteur.

Voici donc l'organigramme du suivi :

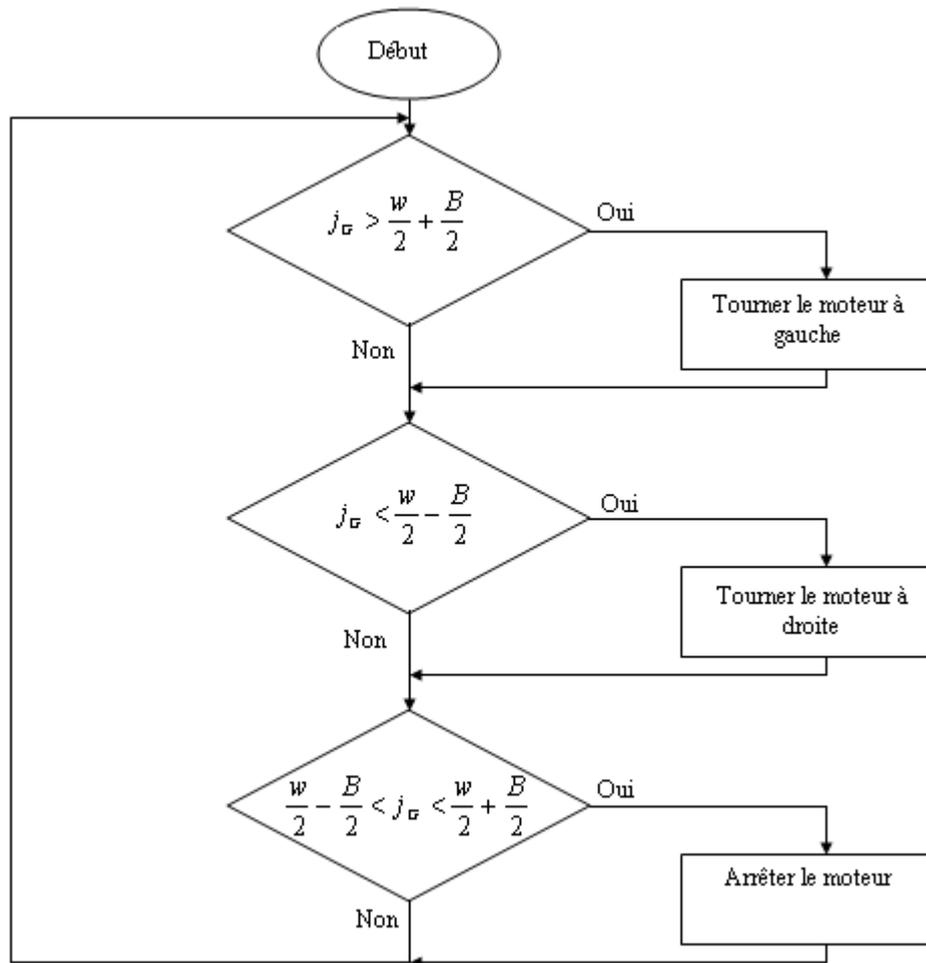


Figure III.10 L'organigramme du suivi

### III.4.5 Génération des signaux de commande

Il nous reste maintenant de définir comment notre logiciel va communiquer avec le port parallèle. Le port parallèle possède un registre de données à huit bits, où on peut écrire un nombre allant de 0 à 255. Comme on a seulement trois probabilités (moteur tourne à droite, à gauche ou moteur arrêté). On va utiliser les deux premiers bits du poids faible D1 et D0.

Voici la table de vérité des signaux :

---

D1	D0	Action
0	0	Moteur arrêté
0	1	Moteur tourne à droite
1	0	Moteur tourne à gauche
1	1	x

Tableau III.1 Table de vérité des signaux de commandes.

### III.4.6 Conclusion

Dans cette partie on a vu comment notre logiciel fait la segmentation, le calcul de l'erreur entre les positions des deux axes à superposer, et enfin comment générer les signaux de commande, ainsi que les différents circuits utilisés pour contrôler le moteur.

## III.5 La partie électronique

### III.5.1 Introduction

Après la segmentation d'image de la scène, la définition du centre d'objet à suivre et l'envoi des signaux de commande via le port parallèle, on va voir comment ces signaux provoquent la rotation et l'arrêt du moteur, à l'aide du circuit électronique qu'on a conçu. Ce circuit est composé de trois modules : la carte de commande, la carte de puissance, et la carte d'isolation optique. Dans cette partie on va détailler le fonctionnement de chaque module, en suivant les signaux de l'entrée jusqu'à l'arrivée au moteur.

### III.5.2 La carte de commande

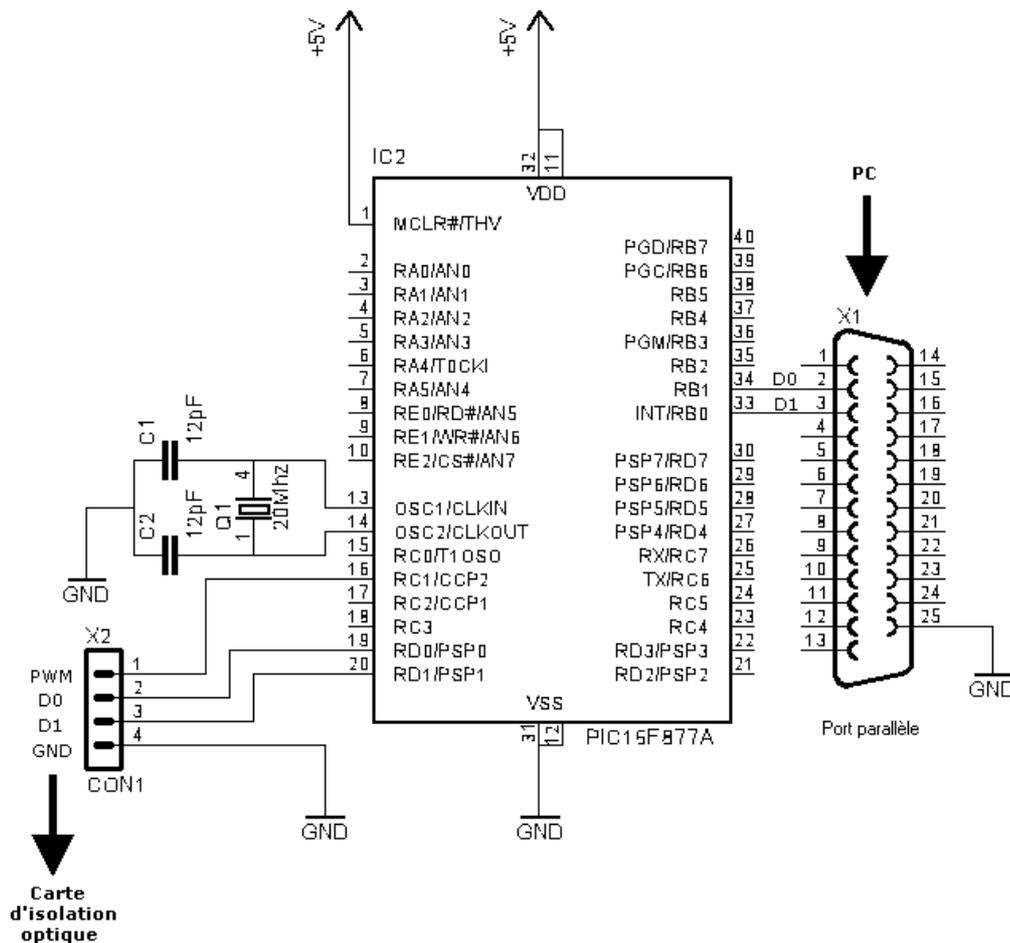


Figure III.11 Le schéma développé de la carte de commande

Cette carte est basée sur le microcontrôleur 16F877A de MICROCHIP. Le programme qui réside dans la mémoire de ce microcontrôleur va lire les signaux du port parallèle à travers le port B, et traduit ces signaux à des ordres à envoyer vers le prochain module. L'autre fonction dédiée à ce programme est de régler la vitesse du moteur à l'aide d'un signal PWM généré à travers la pince C1 du port C. Voici l'organigramme de ce programme :

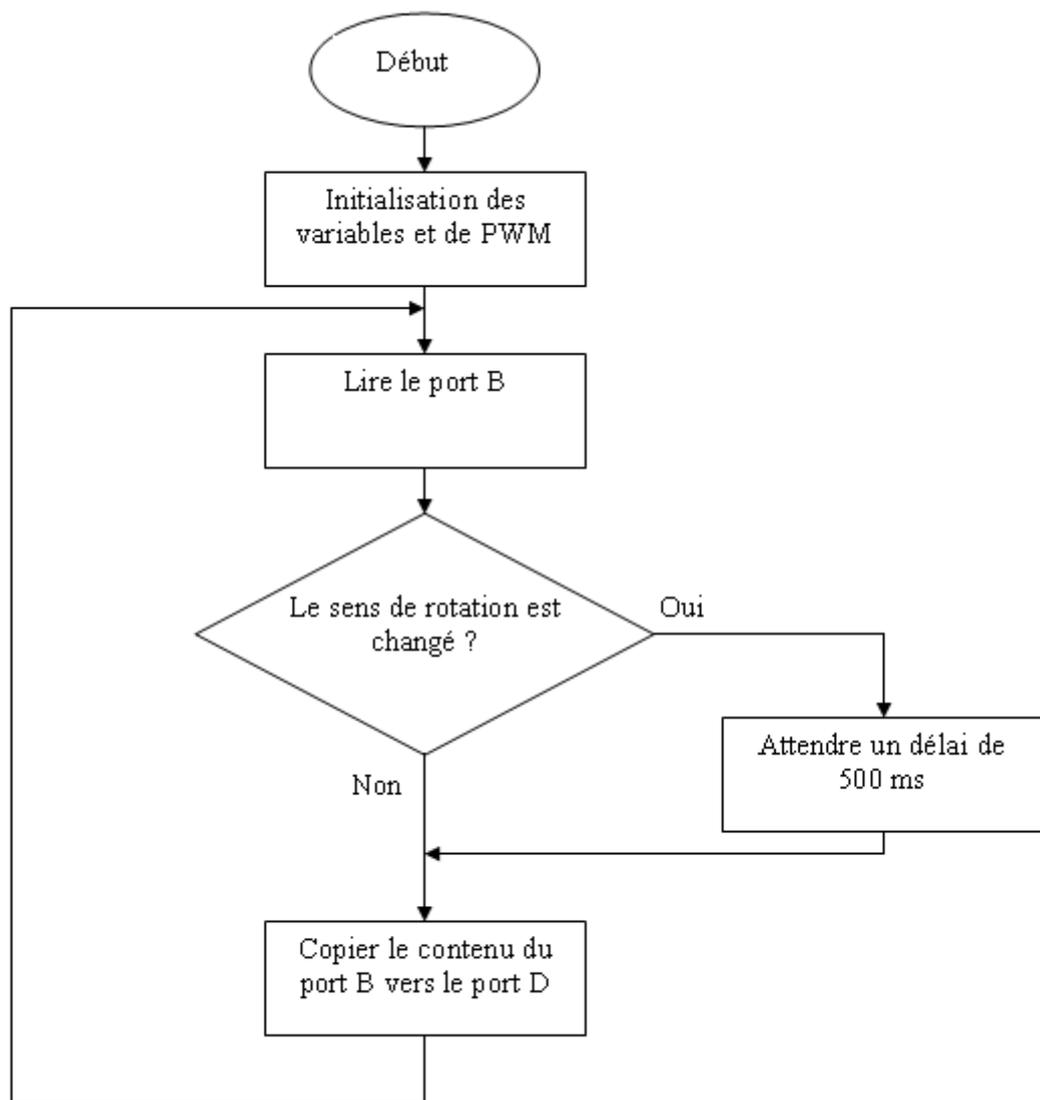


Figure III.12 L'organigramme de l'exécutable qui réside dans la mémoire du pic

Les ordres envoyés sont des signaux numériques. Le port utilisé pour cette fonction est le port D. Comme il y a seulement trois probabilités, on utilise les deux pins du poids faible D1, D0. Voici la table de vérité des signaux envoyés.

B1	B0	D1	D0
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

**Tableau III.2 Table de vérité des signaux envoyés à la carte de puissance.**

Le port B est copié directement sur la sortie (port D), donc pourquoi ne pas relier directement le port parallèle de l'ordinateur à la prochaine carte ? La première raison est qu'on ne peut pas contrôler la vitesse du moteur avec précision directement avec l'ordinateur. Le microcontrôleur a un module intégré chargé de cette fonction. La deuxième raison est que si le moteur change brusquement le sens de rotation, le courant augmente énormément, donc soit le moteur va brûler, soit le pont H va surchauffer et brûler lui aussi. Donc le microcontrôleur va dans ce cas attendre un délai de 500 ms avant de copier le port B vers le port D.

On a vu comment le microcontrôleur traite les signaux envoyés de l'ordinateur pour les envoyer vers la carte de puissance, mais cette carte alimente directement le moteur, et on sait que les moteurs génèrent des bruits et des surtensions, il faut donc isoler cette carte de la carte de commande. On utilise deux alimentations différentes, une pour la partie commande, et l'autre pour la partie puissance. La communication entre ces deux cartes est faite par le biais d'une carte intermédiaire appelée la carte d'isolation optique.

### III.5.3 La carte de l'isolation optique

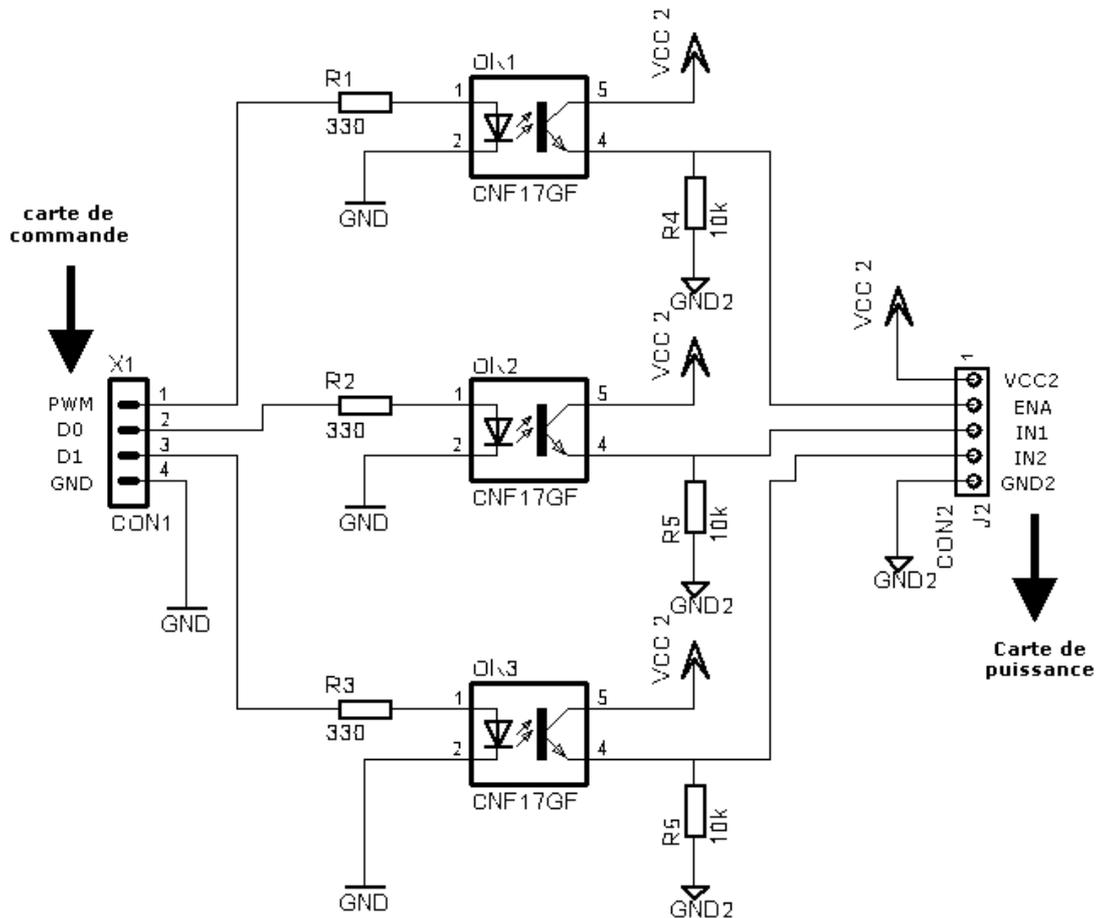


Figure III.13 Le schéma développé de la carte d'isolation optique

Cette carte est composée de trois optocoupleurs, deux pour isoler les deux signaux du port D, et un pour isoler le signal PWM. Maintenant on est sûr que les surtensions et les bruits n'ont pas d'influence sur la partie commande.

Après qu'on a parlé sur la carte d'isolation, on va passer à la carte qui a une interaction directe avec le moteur, c'est la carte de puissance.

### III.5.4 La carte de puissance

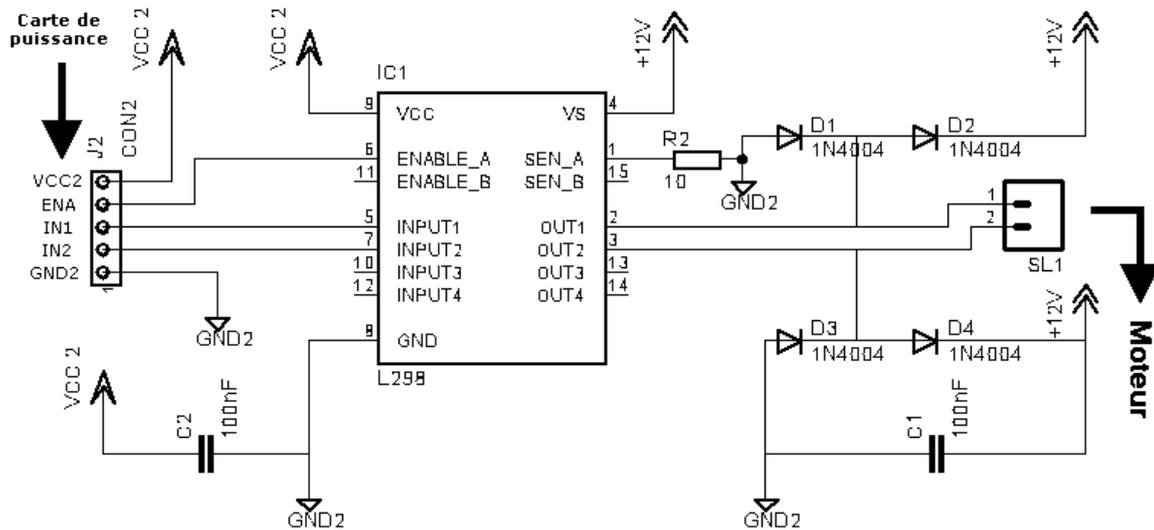


Figure III.14 Le schéma développé de la carte de puissance

Le cœur de cette carte est le pont H « L298 », ce circuit intégré peut contrôler deux moteurs. Voici la table de vérité de l'état du moteur en fonction des entrées :

EN	IN1	IN2	Moteur
0	x	x	Libre
1	0	0	Libre
1	0	1	Rotation à droite
1	1	0	Rotation à gauche
1	1	1	freiné

Tableau III.3 Table de vérité de l'état du moteur en fonction des entrées.

L'entrée EN est reliée directement à la sortie PWM. Les deux entrées IN1 et IN2 sont reliées aux sorties du port D (à travers la carte d'isolation).

### III.5.5 Conclusion

Après réaliser la partie électronique, le système maintenant est complet. On doit faire quelques essais, pour assurer que le système fonctionne comme on veut.

### III.6 Testes et essais

Au début, on a fait des essais sans moteurs, donc visualiser une scène qui contient deux objets (un objet plus l'arrière plan). Les résultats sont étonnants, l'objet est détecté et le centre de gravité est calculé avec une grande précision.

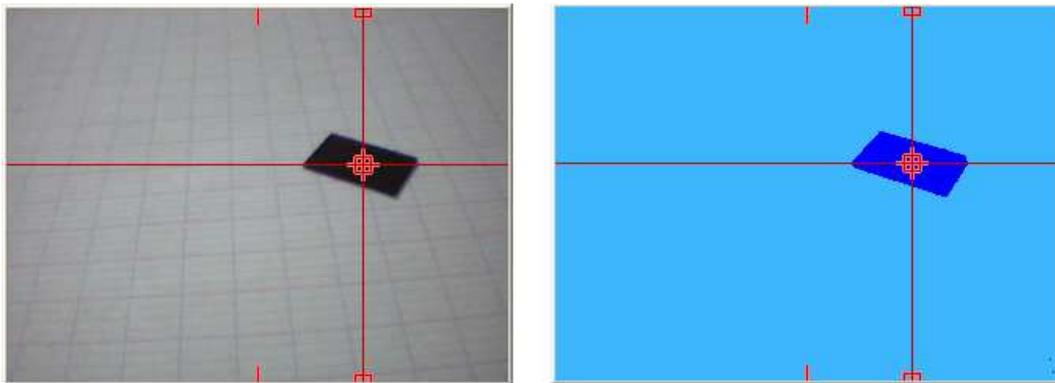


Figure III.15 L'objet à l'état initial

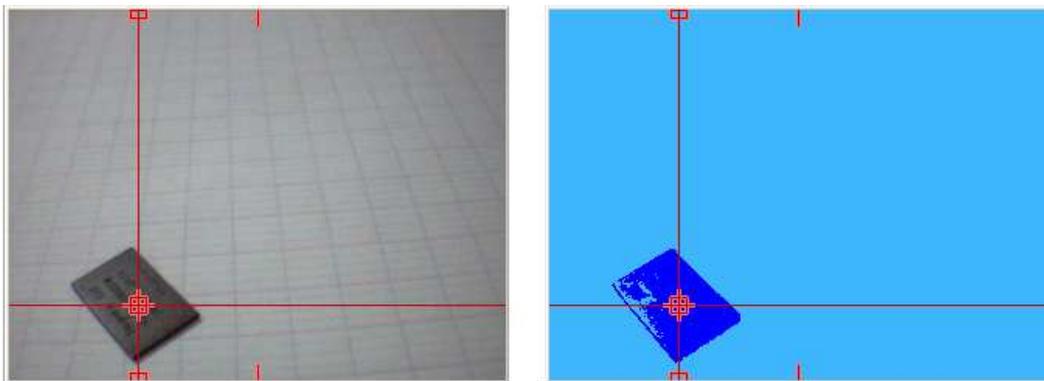


Figure III.16 L'objet après mouvement

Après ça, on a branché le câble du port parallèle, et on a branché les alimentations. Au début le mouvement du moteur était très rapide, donc il dépasse parfois la bande morte. On a abaissé la vitesse jusqu'à ce qu'on est tombé sur la bonne configuration.

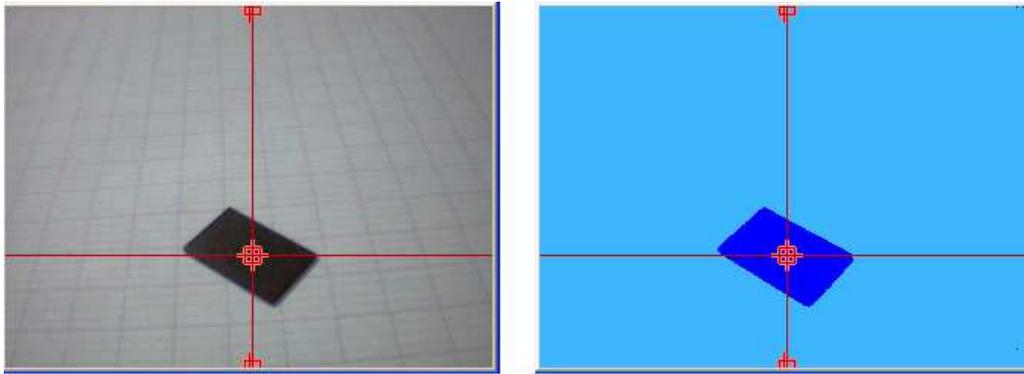


Figure III.17 Superposition des deux axes

### III.7 Conclusion

Dans ce dernier chapitre, on a traité toutes les parties de la réalisation pratique. On a commencé par la partie logicielle, ensuite on a vu le fonctionnement des différentes cartes dans la partie électronique. On a testé, et régler ensuite notre système.



# Conclusion générale

Dans ce travail, on a réalisé un système capable de suivre un objet dans une scène à deux dimensions, après les réglages et les testes, on a arrivé à notre but, le système est fonctionnel à 100%, malgré les problèmes rencontrés durant la réalisation.

Au début on a utilisé un moteur avec un réducteur d'un faible rapport de réduction, la vitesse de rotation était élevée, donc on n'a pas pu suivre l'objet avec la précision voulu, on a décidé donc d'abaisser la vitesse mais ça n'a pas marché. Avec une petite vitesse le moteur ne peut pas démarrer, il nécessite un grand couple de démarrage. La solution apportée à ce problème était de changer complètement le moteur. Cette fois on a utilisé un moteur avec un réducteur d'un très grand rapport de réduction, on a pu atteindre des très petites vitesses, et un très grand couple. Cette fois on a pu atteindre la précision voulu.

Un autre problème était rencontré, le problème cette fois est de définir le centre d'objet après la segmentation. Au début on a utilisé une méthode de balayage. Le premier point rencontré qui appartient à la classe de l'objet à suivre n'appartient pas toujours à l'objet, il appartient parfois à des points de bruit. Donc les résultats sont complètement faux. La solution à ce problème est la méthode du centre de gravité, qu'on a discuté dans le chapitre de la réalisation pratique. Même s'il existe un peu de bruit, sa surface totale (la somme des

surfaces de tous les points de bruits) est négligeable devant la surface de l'objet à suivre. Donc le centre de l'objet calculé reste très proche du centre original. Cette méthode était très efficace.

La réalisation de ce système nous a permis de se familiariser avec le monde du traitement d'image, le monde de la robotique et la programmation. Ce projet nous a permis aussi de travailler avec un système en temps réel. On a travaillé aussi avec des nouveaux outils, tel que la bibliothèque du traitement d'image OPEN CV (Open Computer Vision), qui contient toutes les fonctions nécessaires, pour l'acquisition et le traitement d'image en temps réel. On a utilisé l'environnement Visual c++ pour la programmation, et la bibliothèque INPOUT32 pour la communication avec le port parallèle. Ce projet nous a permis aussi de se familiariser avec les microcontrôleurs, et avec le logiciel PICC de la programmation, basé sur le langage C.

On peut améliorer ce système dans le futur, pour balayer la scène entière, et même dans l'espace 3D. On peut aussi ajouter l'asservissement dans le contrôle du moteur, par exemple un contrôle PID. Ça permet plus de précision.

Notre travail est original, et c'est le premier de son genre dans notre université, espérons donc que notre projet va ouvrir des portes pour les autres étudiants pour travailler sur des sujets dans ce domaine (traitement d'image + robotique).



# Bibliographie

- [1] H. Maître, *La détection de contour dans les images*, Editions Dunod, 1997.
- [2] P. Bernas, *Traitements d'images pour la sécurité globale*, Optics Valley, 2006.
- [3] Radu HORAUD, Olivier MONGA, *Vision par ordinateur*, Hermès, 2003.
- [4] S. Renard, *La robotique industrielle*, Ecole spécialisée de suisse, 2005.

