الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Mention Électronique
Spécialité Systèmes Embarqués

présenté par

RAHMI RABIA

&

BOUCETTA ABDELDJALIL ETTAHAR

pour l'obtention du diplôme de Master en Électronique option électronique des système embarqués

Comptage et tris automatique d'objets placés sur un convoyeur en temps réel en utilisant le Rasberry Pi

Proposé par : Dr. NAMANE Abderrahmane

Année Universitaire 2019-2020

Louange à Allah, seigneur de l'univers, pour la volonté, la santé, le courage et la patiente pour accomplir ce modeste travail.

Nous tenons ainsi, à exprimer ici notre respect et toute notre reconnaissance à notre encadreur Monsieur : Namane Abderrahmane, pour sa bienveillance, sa gentillesse, son encouragement et ses conseils.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nous remercions aussi nos professeurs qui nous ont soutenus durant notre formation à l'université de BLIDA.

Nous remercions nos parents qui ont été là pour nous motiver et nous diriger pendant tout notre parcours scolaire.

Dédicace

À ma chère mère, et à mon cher père qui me manque tellement,

Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes frères, Pour leur soutien moral et leurs conseils précieux tout ou long de mes études.

A mes chères amis houda, tarek, abdou, hamid Pour leurs aides et supports dans les moments

Difficiles.

A tous mes autres amis(e)s.

A tous ceux que j'aime et ceux qui m'aiment.

RAHMI Rabia

Dédicace

A mes parents qui m'ont doté d'une éducation digne, et d'un amour qui a fait de moi ce que je suis aujourd'hui, Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez.

A ma chère sœur, en souvenir d'une enfance dont nous avons partagé les meilleurs et les plus agréables moments. Pour toute la complicité et l'entente qui nous unissent, ce travail est un témoignage de mon attachement et de mon amour.

A mes très chers amis IZZA ABDELAZIZ et MOHAMED SALAH.

A tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

A tous ceux qui me sont proches et que j'ai omis involontairement de citer.

BOUCETTA

ملخص:

يتناول هذا المشروع الكشف عن الأشياء فيصوره بعد حفظها وتتبعها معا الانتشار الواسع للصور الرقمية، باستخدام كاميرا وناقلة تنقل الاجسام.

المشروع يسمح بدراسة الوسائل المختلفة لتتبع جسم متحرك واحدا وأكثر في فيديو.

ويمثل هذا الموضوع تحديا مهما يفرضه وسط الالتقاط الذي يتطلب الدقة لجعلا لتطبيق يعمل في الوقت الحقيقي.

كلمات المفاتيح :الكشف،raspberry، open cv ،التمويه الغوسي ، التشكل ،التدرج الرمادي و الناقل.

Résumé :

Ce projet a pour objectif le tri automatique d'objets placés sur un convoyeur. Pour ce faire, un système utilisant une Rasberry PI, une carte Arduino et deux servo moteurs a été développé et réalisé. Le fonctionnement de ce système est géré par une programmation en langage Python et utilisant la bibliothèque Opencv pour la partie traitement d'images.

Cette réalisation représente un défi pour des applications en temps réel, notamment, la résolution du moyen de capture et la vitesse du convoyeur.

Mots clés : Détection d'objet, tri d'objet, Raspberry PI, Opencv, Convoyeur.

Abstract:

This project aims to automatically sort objects placed on a conveyor. To do this, a system using a Rasberry PI, an Arduino board and two servo motors was developed and realized. The operation of this system is managed by programming in Python language and using the Opencv library for the image processing part.

This realization represents a challenge for real time applications, in particular, the resolution of the capture device and the speed of the conveyor.

Keywords : Object detection, object sorting, Raspberry PI, Opencv, Conveyor.

Listes des acronymes et abréviations

CSI : Camera Serial Interface

CPU: Central Processing Unit

FPS: Frame par Second

GPIO: General Purpose Input/Output

GPIO: General Purpose Input/Output

GPU: Graphics Processing Unit

HDMI: High Definition MultiMedia Interface

HD: Haute Definition

KO: Kilo Octets

KB: Kilo Bytes

LCD :

MA : Milliampére

MP : Mega pixels

MHz: Mega hertz

Open CV: Open source Computer Vision

PWM: Pulse Width Modulation

RAM: Random Acces Memory

RCA: « Radio Corporation of America

RVB: Rouge, Vert, Bleu

SoC: System on Chip

SD: secure Digital

USB: Universal Serial Bus

V: Volts

Table des matières

Chapitre 3 Analyse d'images

Liste des figures

Liste des tableaux

Introduction générale

La vision est le sens qui nous fournit le plus d'informations sur le monde extérieur, elle nous permet de voir, de décrire, et de détecter les objets qui sont présent dans une scène. L'être humain est capable de détecté les objets rapidement, avec une très grande précision, Cela est réalisé grâce au système visuel humain qui est capable d'interpréter les informations sensorielles, c'est-à-dire les informations fournies par l'environnement au système visuel à une instante donnée.

L'interprétation est réalisée grâce au cerveau, qui peut détecter et situer les objets, détecter le mouvement, etc. La détection a besoin d'un modèle de l'objet. Pour l'être humain, ce modèle correspond à une représentation mentale de l'objet qui peut être apprise en retenant les caractéristiques les plus discriminantes de l'objet. Les caractéristiques peuvent être toutes sortes d'attributs de l'objet : forme, couleur, taille, volume, etc.

En d'autres termes, le problème que cherche à résoudre la détection des formes est d'associer une étiquette à une donnée qui peut se présenter sous la forme d'une image brute ou d'un signal.

Avec la généralisation de l'utilisation des images numériques, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi diverses que la vidéo surveillance, la compression vidéo, l'imagerie médicale, la robotique, l'interaction homme machine, l'analyse de séquences sportives...etc. En effet, les zones de mouvement d'une séquence d'images correspondent souvent à des événements sur lesquels un système de vision doit se focaliser. L'analyse du mouvement d'un objet est un vaste sujet qui englobe un certain nombre de problématiques :

- ✓ La détection des objets, c'est-à-dire la détection d'un ensemble de régions d'intérêt dans la scène observée,
- ✓ Le suivi de primitives ou de régions, dont le but est de déterminer la forme de chaque primitive ou région dans l'image à chaque instant.

Dans le premier chapitre on va parler sur la détection et le suivi des objets en générale avec les types de convoyeur ensuite dans le deuxième chapitre on va présenter des informations générales sur le Raspberry Pi, l'Arduino et le servo moteur avec la méthodologie du traitement après dans le troisième chapitre a pour but de préparer l'image pour un autre niveau de traitement afin de bien extraire les informations pertinentes , notamment, la conversion de la couleur en niveau de gris, le filtrage, la binarisation, la morphologie mathématique et le suivi de contour et à la fin on termine avec le chapitre quatre qui représente notre réalisation et test .

,

Chapitre 1 **Détection, suivi des objets et types de convoyeur**

1.1 Introduction

Avec la généralisation de l'utilisation d'images numériques, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi diverses que la vidéo surveillance, la compression vidéo, l'imagerie médicale, la robotique, l'interaction homme-machine, l'analyse de séquences sportives... En effet, les zones de mouvement d'une séquence d'images correspondent souvent à des événements sur lesquels un système de vision doit se focaliser.

L'analyse du mouvement est un vaste sujet qui englobe un certain nombre de problématiques. On peut notamment citer :

– la détection du mouvement, qui consiste à étiqueter chaque pixel d'une image suivant s'il correspond ou non à une région en mouvement dans la scène,

– la détection des objets en mouvement, c'est-à-dire la détection d'un ensemble de régions d'intérêt en mouvement dans la scène tridimensionnelle observée,

– la segmentation basée mouvement de la scène, pour laquelle chaque région de l'image ayant un mouvement distinct des autres est détectée et segmentée,

– l'estimation du mouvement, qui consiste à estimer, à partir d'une séquence d'images, le mouvement apparent des objets composants une scène tridimensionnelle,

– le suivi de primitives ou de régions, dont le but est de déterminer la position de chaque primitive ou région dans l'image à chaque instant,

– la reconnaissance et la modélisation d'activités ou de gestes.

Les trois premières problématiques (détection du mouvement, détection des objets en mouvement et segmentation basée mouvement), qui sont au cœur des travaux

présentés, sont en général une première étape pour des outils automatiques de vision par ordinateur. Ces outils peuvent avoir pour vocation, soit uniquement de détecter, soit de détecter et reconnaître, soit de détecter et suivre des objets pour, par exemple, analyser le comportement ou la trajectoire de ces objets. Ainsi, des méthodes de détection fiables et automatiques sont indispensables pour de nombreuses applications de vision par ordinateur.

Les autres problématiques sont toutes aussi importantes et nécessitent la mise en place de méthodes simples et robustes. Tous ces sujets font l'objet d'un grand nombre de travaux, mais il n'existe pas, à l'heure actuelle, d'algorithmes aboutis s'adaptant à n'importe quelle situation.

1.2 Modélisation d'un objet [1]

1.2.1 Caractéristiques d'un objet

Dans le domaine du mouvement d'un objet dans une séquence, un objet peut avoir différentes caractéristiques, il peut être :

- **Constant ou variable :** La forme, le mouvement, les couleurs ou les textures d'un objet varient ou non au cours du temps. Ces considérations jouent en faveur de l'adoption ou non d'un modèle de l'objet pour en améliorer le suivi.

- **Rigide ou non rigide :** pour la forme nous parlerons d'objet rigide (voiture) ou non-rigide (corps humain) c.à.d.la distance entre les points de l'objet est constante ou variable au cours du temps.

- **Unique ou multiple :** Dans le cadre d'un suivi labellisé où chaque objet porte une étiquette et n'en change pas durant l'opération de son suivi, une distinction entre chaque objet est nécessaire c.à.d. que chaque objet doit être unique. L'unicité repose, là aussi, sur un modèle. Cependant celui-ci n'est pas nécessairement connu au préalable. Cependant une connaissance des critères potentiellement discriminants permet une amélioration de la charge de calcul. La description de chaque objet selon des critères pertinents assure une labellisation fiable.

1.2.2 Représentation d'un objet

Un objet, dans un scénario de suivi, est une entité indépendante pourvue de son identité spatiale (sa forme, son contour, etc.) dans un environnement particulier. Cependant, on ne dispose pas toujours de toutes les informations qui caractérisent l'objet à suivre, mais on utilise un modèle de représentation de son état à surveiller. Une modélisation d'objets par catégories peut être présentée.

A. Modélisation par points

L'objet à suivre est représenté soit par un point qui représente le centre de gravité de l'objet concerné, soit par un ensemble de points [3], (voir figure 1.1). En général cette dernière est efficace pour le suivi d'objets de petites tailles ou régions.

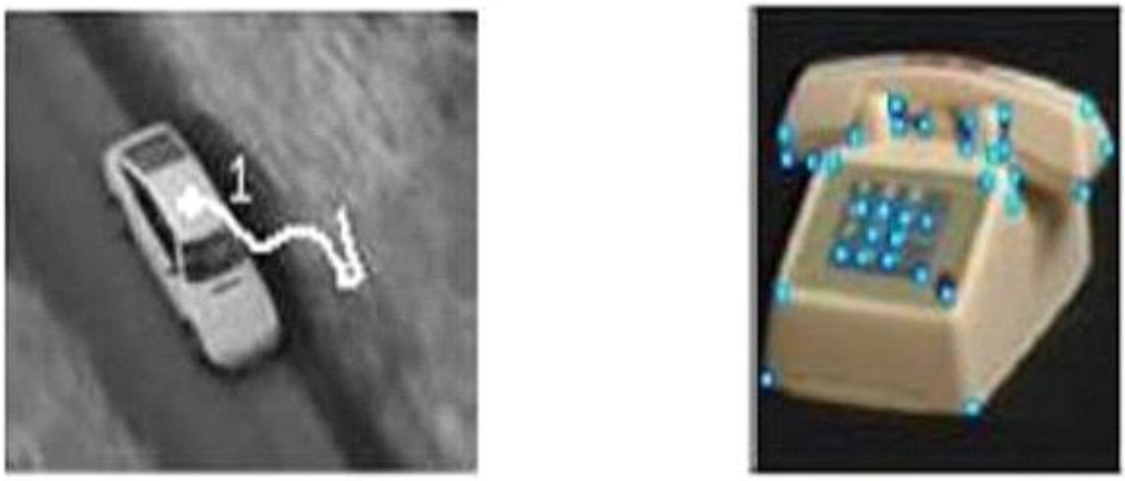


Figure 1.1 : Points de contrôle

B. Modélisation par formes géométriques

La forme d'un objet est représentée par un rectangle ou (boîtes englobantes) (Figure 1.2) ou une ellipse, le mouvement d'objet défini par cette représentation est souvent modélisé par une translation, projection et autre transformation, cependant ces formes géométriques sont plus appropriées à représenter les objets rigides simples, elles sont également employées pour suivre les objets souples.

Figure 1.2 : Boîtes englobâtes

C. Modélisation par silhouette et contour

Par définition, un contour est la frontière qui sépare deux objets dans une image. La région à l'intérieur du contour s'appelle la silhouette de l'objet (Figure1.3). La représentation par le contour et la silhouette est appropriée pour représenter des objets non rigides et les modèles déformables complexes.

Figure 1.3 : Modelé silhouette

D. Modélisation par des modèles de formes articulées

Les objets articulés sont composés de parties du corps qui sont liés avec des joints. Par exemple le corps humain est un objet articulé avec torse, jambes, mains, tête et pieds reliés par des joints (Figure1.4). On peut modéliser ces parties en utilisant des cylindres

ou des ellipses, les rapports entre eux sont régis par des modèles de cinématique de mouvement, par exemple l'angle commun etc.

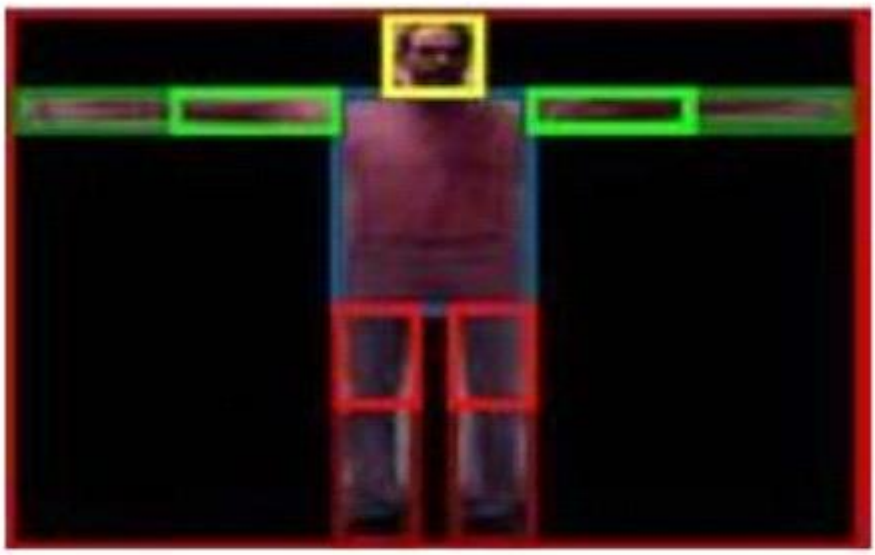


Figure 1.4 : Modelé d'apparence articulé

E. Modèles squelettiques

Le squelette d'objet peut être extrait en appliquant la transformation de l'axe médiane à la silhouette d'objet (Figure1.5). Ce modèle est généralement utilisé pour la reconnaissance de l'objet et du suivi des modèles cinématiques complexes et la reconnaissance de mouvement.

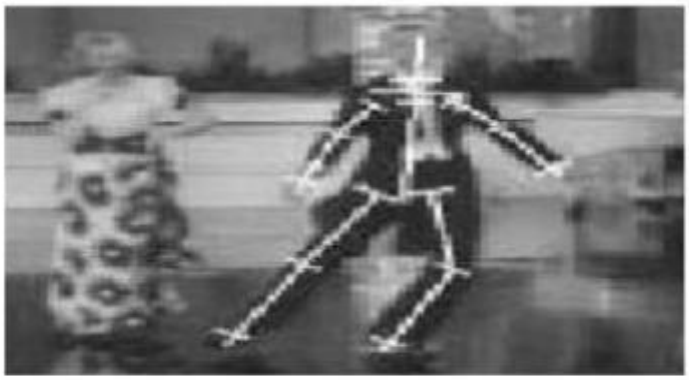


Figure 1.5 : Modelé squelette

1.3 Détection d'objets [1]

Dans le domaine de la vision par ordinateur, la détection consiste à percevoir une scène statique ou dynamique, et de savoir le changement de mouvement approprié en

déduisant ainsi la cinématique de tous les objets mobiles par rapport à la caméra. La détection est une tâche difficile, elle dépend généralement de la nature de la scène et de l'application correspondante.

- **Catégorisation des méthodes de détection**

Parmi Les méthodes de détection on peut Citer :

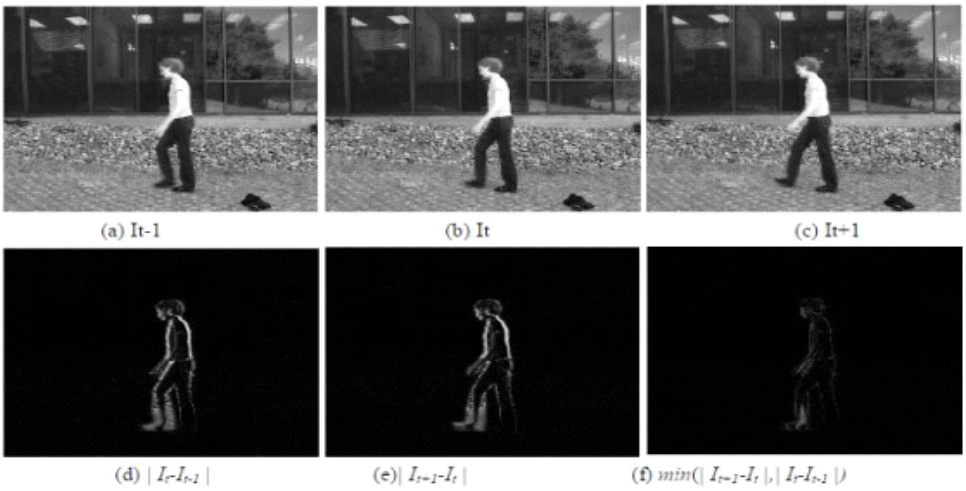
1. Les détections basées sur la différence inter-images.

2. Les détections basées sur la modélisation du fond.

3. Les détections utilisant la notion de cohérence.

1.3.1 Détections basées sur la différence inter-images

La méthode de détection de mouvement la plus simple consiste à faire la soustraction entre deux images successives. Elle porte aussi le nom de méthode de différence temporelle dans d'autres littératures. Les pixels dont l'intensité résultante est proche de zéro sont assimilés comme étant les pixels du fond. Les méthodes basées sur la différence inter-images s'adaptent très bien aux environnements dynamiques mais



laisse des "trous" dans les objets en mouvements. Par exemple, la différence temporelle souillée permet de détecter des objets en mouvement dans une séquence vidéo. Une amélioration de cette méthode consiste à extraire trois images de différence au lieu de deux permettant de décider si un pixel a bougé.

Figure 1.6 : Différence inter-images

1.3.2 Détections basées sur la modélisation du fond

Une autre catégorie de méthodes, très populaire en détection de mouvement, repose sur la modélisation du fond. Plusieurs conditions sont nécessaires à l'utilisation de ces

techniques. Tout d'abord, la caméra doit être maintenue fixe et les occultations de parties du fond par des objets en mouvement doivent rester temporellement minoritaires. De plus, il est préférable d'avoir, au début de la séquence, un certain nombre d'images sans objets en mouvement afin de pouvoir apprendre correctement le fond.

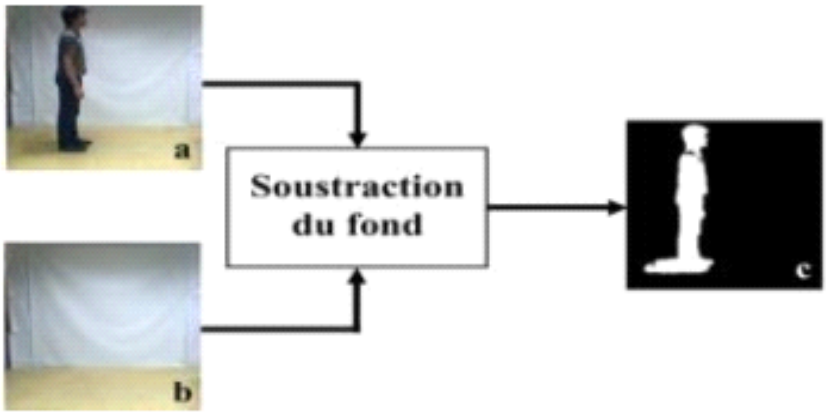


Figure 1.7 : détections basées sur la modélisation du fond

1.3.3 Détections utilisant la notion de cohérence

Une dernière approche consiste à définir un objet mobile comme une région ayant un mouvement cohérent. Dans ce contexte, un mouvement cohérent peut être défini comme un mouvement ayant de grande chance de provenir d'une "cible" classique (personne, véhicule). Cette définition a été utilisée pour la segmentation de mouvement en ajoutant l'hypothèse suivante : un objet avec un mouvement cohérent se déplace dans une direction approximativement constante pendant une courte période (en pratique quelques images).

Les modèles de mouvement dérivés du flot optique sont utilisés comme primitives pour la détection ou le suivi. Le flot optique a pour rôle de décrire le mouvement cohérent des points entre des images successives, des mesures de flot optique cohérentes en direction sont accumulées pendant quelques pas de temps. Une estimation du déplacement de chaque pixel à travers une séquence d'images est ainsi obtenue. Il est alors possible de distinguer les objets mobiles, qui se déplacent avec une direction constante, des mouvements parasites. Une région cohérente peut être directement vue comme une couche. La différence et l'avantage ici est que seuls les

objets cohérents sont détectés. Cela permet l'application directe d'autres traitements tels que l'identification ou le suivi.

1.4 Suivi des objets [1]

1.4.1 But du suivi

Le suivi de plusieurs objets dans une séquence d'images peut s'avérer difficile selon les scènes complexes qu'il traite, cette complexité réside là où les objets peuvent avoir des tailles différentes. Ces objets peuvent être rigides ou non-rigides et s'occulter les uns les autres de ce fait, des méthodes de suivi sont apparues, le but de ces méthodes est d'estimer au fil du temps les paramètres d'une cible (ou plusieurs) présente dans le champ de vision de la caméra et initialement détectée par un moyen quelconque. Les paramètres peuvent être divers position dans l'image, à laquelle peuvent s'ajouter la taille et l'orientation apparente, l'attitude, l'apparence, etc. La plupart des systèmes de suivi actuels ne marchent pas bien avec ces conditions difficiles.

1.4.2 Méthodes de suivi

Les méthodes existantes pour le suivi simultané de plusieurs objets dans une séquence d'images impliquent généralement que des caractéristiques telles que des histogrammes couleurs, des mesures de corrélation, des estimations de vitesse, ou des mesures de distance entre régions puissent être utilisées pour suivre chaque objet. Plusieurs classifications des méthodes de suivi d'objets ont été proposées dans la littérature qui peuvent être divisées en trois principales catégories :

A. Suivi par appariement de détection ("detect-before-track"),
B. Suivi par segmentation dynamique.
C. Suivi déterministe par détection séquentielle.

A. Suivi par appariement de détections [2]

Le suivi se fait alors en assignant les observations aux pistes en cours d'estimation. Ces méthodes, dites "detect-before-track" sont très populaires en pistage sonar et radar.

Elles peuvent être déterministes ou probabilistes. On est confronté dans les deux cas à un problème combinatoire d'association.

- **Méthodes déterministes**

Le principe des méthodes déterministes est d'associer les observations aux pistes en cours de suivi en minimisant une distance calculée sur certaines caractéristiques de l'objet. Les caractéristiques des objets couramment utilisées sont l'apparence peuvent être des densités (histogrammes de couleur ou de contour), une carte de contours (contour ouvert ou fermé de l'objet) ou une combinaison de ces modèles.

- **Méthodes probabilistes**

Les observations obtenues par un algorithme de détection sont très souvent corrompues par du bruit. De plus, le mouvement ou l'apparence d'un objet peut légèrement varier entre deux images consécutives. Les méthodes probabilistes permettent de gérer ces variations en ajoutant une incertitude au modèle de l'objet et aux modèles des observations. Le suivi d'une seule cible est alors obtenu par des méthodes de filtrage (filtres de Kalman, filtrage particulaire). Le suivi de plusieurs objets peut lui aussi se faire avec ces méthodes de filtrage mais une étape préalable d'association de l'objet avec l'observation la plus probable doit être ajoutée.

B. Segmentation dynamique [1]

Figure 1.8 : Contour actif [1]

Les méthodes de suivi par segmentation dynamique sont utilisées lorsque l'on souhaite extraire la silhouette de la cible à chaque instant, et ce sans connaissance a priori sur sa Forme. Ces approches reposent sur une succession de segmentations. Elles font généralement Evoluer le contour de l'objet à l'instant précédent jusqu'à sa nouvelle position à l'instant courant.

C. Suivi déterministe de fenêtre englobant par détection séquentielle [2]

La dernière catégorie de méthodes évoquée dans cette partie regroupe les méthodes de suivi d'image. Une image est une boîte (en général un rectangle mais parfois aussi une ellipse) entourant l'objet à suivre. Il s'agit en fait d'une petite portion de l'image. Les techniques de suivi considérées dans cette section sont basées sur la conservation de l'apparence (généralement couleur et/ou luminance) de l'objet pendant au moins deux instants consécutifs. On distingue les algorithmes faisant l'hypothèse de conservation de l'apparence localement (en chaque point de l'objet) et ceux utilisant une caractérisation globale de la cible (hypothèse globale de conservation de l'apparence).

- **Suivi différentiel d'image**

Cette approche consiste à chercher la portion d'image la plus proche dans l'image courante de celle à l'instant précédent. La recherche se fait souvent autour de la position de l'objet à l'instant précédent. Ces approches sont appelées "block matching" ou "template matching". Elles font appel à une mesure de similarité telle que la corrélation ou la SSD ("sum of square différences"). Les intensités et les couleurs étant sensibles aux changements d'illumination, il est aussi possible d'exploiter les gradients de l'image. Le coût de calcul de ce type de méthodes peut être élevé. Il dépend de la taille du voisinage dans lequel l'image est recherchée.

- **Suivi de distributions**

L'utilisation de distributions dans la recherche de la cible à l'instant courant se base alors sur des distributions (histogrammes ou mélanges de gaussiennes) des couleurs dans une région de géométrie simple. La méthode la plus utilisée dans cette catégorie est le suivi par « mean shift ». La distribution est un histogramme de couleur.

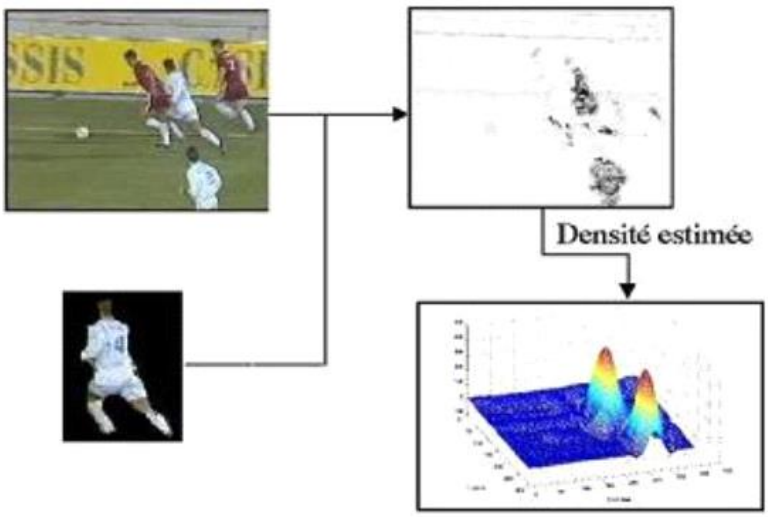


Figure 1.9 : Suivi de distributeur [2]

L'algorithme consiste alors à déplacer une fenêtre d'analyse (noyau) de manière à trouver l'histogramme contenu dans la fenêtre qui coïncide le mieux avec l'histogramme de référence.

1.4.3 Caractéristiques d'un bon algorithme de suivi [1]

Il ressort des techniques de suivi présentées précédemment des points importants permettant de définir la qualité d'un algorithme de suivi.

Les caractéristiques d'un bon algorithme de suivi sont les suivantes :

- La méthode doit être capable d'initialiser automatiquement les cibles et doit gérer les arrêts et les sorties du champ de la caméra.
- Elle doit de plus être robuste aux changements d'illumination et aux éventuels changements de topologie. La topologie n'est importante que dans le cas où une segmentation de la cible est exigée.
- Enfin, la méthode doit permettre de continuer à suivre la cible même en cas d'occultations partielles ou totales par un autre objet ou par le fond.

La première caractéristique ne peut être obtenue qu'avec l'ajout d'une méthode de détection des objets, des observations étant apportées à chaque instant ou à des instants espacés par un faible pas de temps.

Les changements d'illumination ou de photométrie ne sont bien pris en compte que par les méthodes utilisant des histogrammes ou des mélanges de gaussiennes sur l'intensité ou la couleur. Ces distributions doivent régulièrement être mises à jour.

Enfin, la gestion des occultations n'est bien considérée qu'avec l'utilisation d'observations ou en ajoutant une loi dynamique au processus de suivi, la difficulté étant alors de déterminer cette loi. La plupart du temps les objets peuvent avoir un mouvement quelconque qu'aucune loi ne peut caractériser.

1.5 Convoyeur

un convoyeur est un mécanisme composé de plusieurs éléments dont le

but de transporter une charge isolée (cartons, bacs, sacs ...) ou de produit en vrac

(terre, poudre, aliments...) d'un point A à un point B.

1.5.1 Les types de convoyeur

1.5.1.1 Convoyeur a bande [4]

Les convoyeurs à bande (figure 1.10) sont caractérisés par le type de bande transporteuse utilisée (matériaux, texture, épaisseur) et par la position du groupe de motorisation (central ou en extrémité).

Dans tous les cas, un convoyeur à bande se compose

- D'un tambour de commande et de sa moto réductrice
- D'un rouleau d'extrémité
- D'un châssis porteur avec une sole de glissement qui assure le soutien de la bande
- D'une bande transporteuse.

Les convoyeurs à bande modulaire permettent, grâce à leur bande rigide en acétal, d'accumuler des charges (avec frottement entre la bande et les objets transportés). La bande est en fait une chaîne en plastique qui vient s'engrener dans des pignons également en plastique. En matière de maintenance, l'avantage est de ne pas avoir de centrage et de tension de bande à effectuer, contrairement à un convoyeur à bande classique [4].

Figure 1.10 : Vue générale du convoyeur a bande

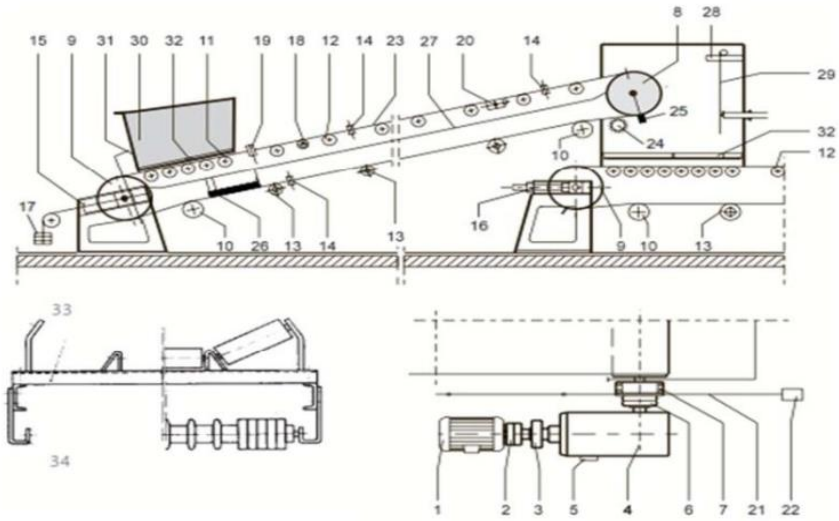


Figure 1.11 : Convoyeur a bande [4]

1. moteur
2. Moteur accouplement
3. Frein
4. Pilote de transmission
5. Anti retour
6. Rouler d'accouplement
7. Roulement de poulie
8. Rouler
9. Filer de poulie
10. Déviation ou repousser poulie
12. Support de poulie avant
13. Retour de poulie avant
14. Rouleau de guidage
15. Compteur de poids
16. Vis de graisse
17. Contre poids
18. Compteur vitesse de tapis
19. Commende réduction de tapis de roulement
20. Ceinture direction de poulie avant
21. Tirer de fil
23. Bande convoyeuse
24. Rouleau a brosse
25. Grattoir recaler
26. Recaler
27. Plaque de couverture
28. Capot
29. Bar cloison
30. Livraison goulotte
31. Garniture de goulotte
32. Hotte planche

a) Convoyeur à bandes métalliques

Ces convoyeurs sont principaux ment utilisés dans le domaine de la métallurgie, ils permettent de transporter des pièces coupantes, abrasives, lourdes et à des températures élevées [4]. Ces convoyeurs sont particulièrement adaptés à l'évacuation des chutes de découpe et de copeaux métallique et non ferreux les rendant incompatibles avec un convoyeur magnétique.

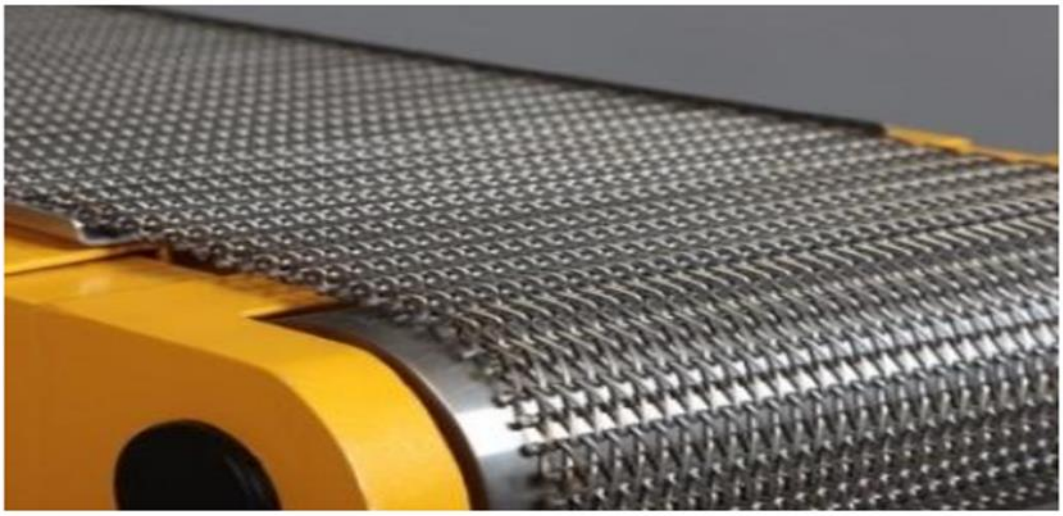


Figure 1.12 : Convoyeur a bandes métalliques

b) Convoyeur à bandes textiles

Les bandes transporteuses à carcasse textile ont, suivant leur domaine d'utilisation, des revêtements avec différentes propriétés ainsi que des carcasses textiles à un ou plusieurs plis. Ce sont des produits endurants pour une multitude d'opérations de transport dans la construction mécanique en général, ainsi, que dans de nombreux autres secteurs industriels [4].

Figure 1.13 : Convoyeur a bandes textile

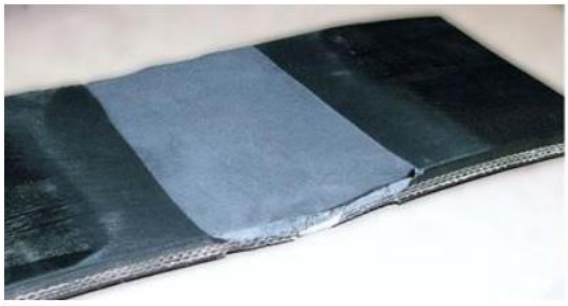


Figure 1.14 : bandes en textile

1.5.1.2 Convoyeur à raclette

Le convoyeur à raclette est un engin de transport continu dont l'organe de traction est une chaîne ou deux sans fin portant des raclettes. Lors du déplacement de la chaîne, les raclettes accrochent la matière chargée et ka déplacent dans le couloir en tôle dans le sens du mouvement de la chaîne.

Les convoyeurs à raclettes (figure 1.15) se composent des éléments suivants [4].

- Tête motrice
- Chaîne de traction
- Raclettes
- Étoile de retour
- Dispositif de tension
- Couloir du convoyeur

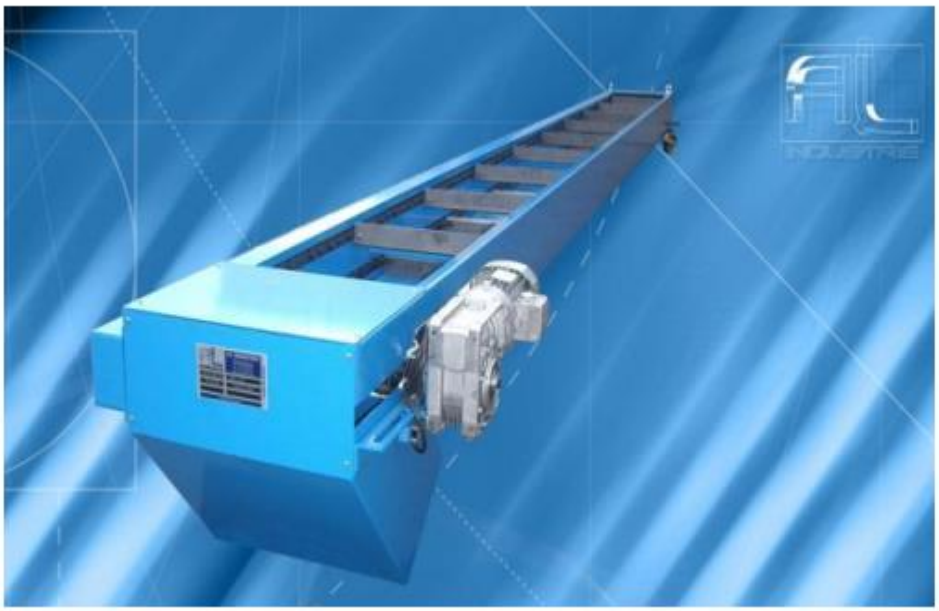


Figure 1.15 : Convoyeur à raclette

1.5.1.3 Convoyeur à magnétique

Est un appareil muni d'une bande avec une partie magnétique qui est placé en dessous de la bande permet d'attirer les produits métalliques vers le bas leur donnant ainsi plus de stabilité.

Les convoyeurs à tambour magnétique permettent la séparation des particules ou déchets métalliques. Souvent employé en fonderie pour extraire les déchets métalliques d'un transporteur de sable après l'opération de décochage [4].

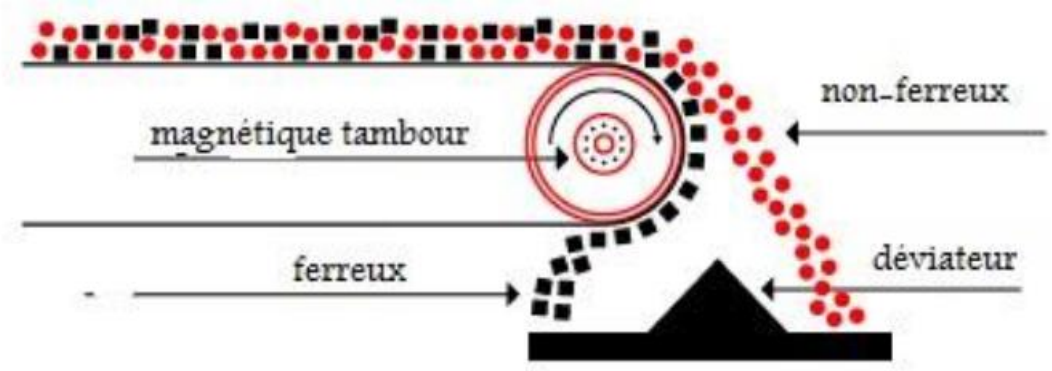


Figure 1.16 : Bande magnétique

1.5.1.4 Convoyeur à chaîne

Les convoyeurs à chaînes permettent le déplacement de charges qui ne pourraient pas l'être sur des convoyeurs à rouleaux (cas des palettes ou containers dont les "skis" sont perpendiculaires au sens de déplacement).

Selon la rigidité de la charge à transporter, le nombre de chaînes est augmenté de sorte à réduire l'entre-axe des chaînes. Il existe des convoyeurs à deux, trois, quatre, voire cinq chaînes et plus.

Ces convoyeurs se caractérisent par le nombre de chaînes, les matériaux des chaînes (acier, inox, plastique) ainsi que la robustesse de leur châssis porteur qui dépend de la charge à supporter.

L'accumulation est en général non préconisée. Pour le passage d'un convoyeur à l'autre, il est quelque fois conseillé d'imbriquer les convoyeurs entre eux en variant les entre axes des chaînes. L'entrainement des charges est alors assuré en permanence, y compris durant le transfert [4].

Figure 1.17 : Convoyeur à chaine

1.5.1.5 Convoyeur à rouleaux

Ils sont utilisés pour le transport ou l'accumulation de produits suffisamment longs pour ne pas tomber entre deux rouleaux. Les colis à transporter doivent être également à fond plats et rigides (voir méthode de détermination dans la rubrique Lien externe) [4].

A. Convoyeur en courbe

Il existe des convoyeurs à rouleaux coniques pour décrire des courbes à 45, 90 et 180°. La conicité des rouleaux est en effet nécessaire pour appliquer au colis une vitesse linéaire différente en fonction de sa position par rapport au rayon de la courbe.

Une autre méthode plus économique, et appliquée généralement aux convoyeurs à rouleaux libres consiste à réaliser plusieurs voies de rouleaux cylindriques, parallèles entre elles, et permettant la différenciation des vitesses [4].

B. Pente des convoyeurs gravitaires

En fonction de la nature de la charge à transporter (c'est-à-dire en fonction de la rigidité de sa face de contact) et de sa masse, la pente nécessaire sera comprise entre 1,5 et 5 % (soit une élévation 1,5 à 5 cm/m de convoyeur) [4].

Figure 1.18 : Convoyeur à rouleaux libres

Figure 1.19 : Convoyeur à rouleaux conique

1.6 Conclusion

De nombreuses applications en vision par ordinateur nécessitent la détection et le suivi des objets en mouvement dans une séquence d'images. La plupart des méthodes existantes ne donnent pas de bons résultats que pour de séquence avec de fond qui présente un peu de changement, ou si le fond et les objets sont rigides.

La mise en œuvre de l'évaluation d'un système de suivi d'objets est un problème difficile du fait de la complexité du système lui-même, ainsi, Le degré de précision d'une méthode de suivi est plus ou moins grand, selon le type d'application utilisée. Un modèle de représentation des objets bien choisi permet souvent d'améliorer la qualité d'un suivi afin d'obtenir la précision souhaitée en rapport avec l'application.

Déterminer le type du convoyeur à bande avec un, bon dimensionnement de calcul dans le but de nous assurer la longue durée de vie de ses composants et d'éviter les risques et les incidences sur la sécurité de ces derniers.

Chapitre 2 **Matériels utilisé et méthodologie du traitement**

2.1 Introduction

Ce chapitre présente des informations générales sur le Raspberry Pi, l'Arduino et le servo moteur. Nous commençons par des informations générales telles que les modèles publiés et leurs composants de base. Il présente également des informations sur l'historique. La bibliothèque OPENCV sera utilisée dans notre projet pour suivre des objets en utilisant une camera Raspberry Pi Rev 1.3.

2.2 Raspberry Pi [5]

Raspberry Pi est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier…). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (C, C++, python, scratch...) et d'être utilisé dans les systèmes embarqués. Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo.

Figure 2.1 : Carte Raspberry Pi

2.2.1 Historique de Raspberry Pi [6]

RASPBERRY Pi, tel qu'illustré à la figure 2.2, est un ordinateur de bord unique ayant un format aussi petite qu'une carte de crédit de la fondation RASPBERRY Pi. Une idée de production RASPBERRY Pi a commencé en 2006 avec la prise de conscience de l'absence de la jeune génération des connaissances sur le fonctionnement de l'ordinateur. Un groupe d'universitaires et d'ingénieurs de l'Université d'Ottawa.

L'Université de Cambridge a donc décidé de mettre au point un très petit ordinateur qui permet d'effectuer les tâches suivantes que tout le monde peut se permettre d'acheter pour créer un environnement d'apprentissage dans la programmation. Le RASPBERRY Pi projet est devenu prometteur avec l'apparition de bon marché et puissant des processeurs mobiles avec de nombreuses fonctionnalités avancées permettant un développement possible de RASPBERRY Pi qui s'est poursuivi sous la fondation RASPBERRY Pi spécialement créée à cet effet avec le premier produit lancé en2012

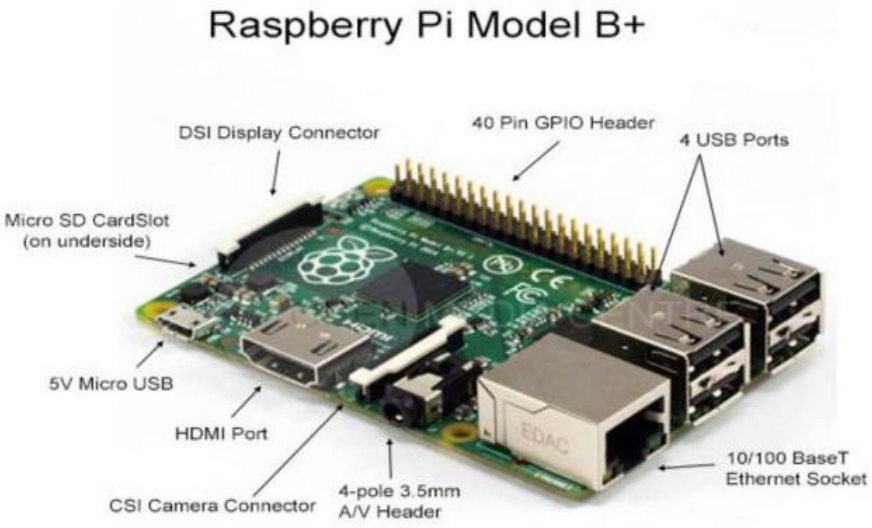


Figure 2.2 : Les portes de Raspberry Pi

Le développement de mini-ordinateur RASPBERRY Pi est un processus continu. Jusqu'au printemps 2015, la fondation RASPBERRY Pi a publié cinq modelé en deux générations d'ordinateur.

La première génération se compose de quatre modelé. D'abord est le modèle B suivi respectivement du modèle A, du modèle B+ et du modèle A+. Ces deux derniers modelés sont des versions mises à jour de leurs versions précédentes pour rendre

l'ordinateur plus efficace et plus pratique pour les utilisateurs, en particulier en ayant une consommation d'énergie plus faible et plus de ports USB (Universal Serial Bus). La deuxième génération se compose d'un seul modèle appelé raspberry Pi 2 modèle B dont la spécification est basée sur Raspberry Pi modèle B+ mais avec un CPU plus rapide et plus de mémoire.

RASPBERRY Pi modèle B, comme le montre la figure 2.3 (b), a été publié au début de 2012 avec la spécification de 256 Mo de RAM, deux porte USB et un port Ethernet. Plus tard dans la même année, dans une nouvelle version, la quantité de RAM a été augmentée a 512 Mo, suivie de la sortie de la carte a spécifications inferieures, modèle A, comme le montre la figure 2.3(a). Le modèle A est sortie avec la même quantité de RAM que l'ancien modèle B à 256 Mo, mais avec un port USB et pas de port Ethernet. Ces ordinateurs de première génération utilisent Broadcom SoC, BCM2835, qui intègre un processeur ARM1176JZF-S à cœur unique de 700 MHz, un GPU VideoCore et une variété de périphériques. Alors que le modèle B peut être utilise dans n'importe quelle application, le modèle A, moins cher, est utile dans des applications spécifiques qui nécessitent un poids léger et une faible consommation d'énergie, comme la robotique ou tout service de médias portables.

En 2014, une version améliorée des deux modèles précédents a été lancée, sous les noms B+ et A+, comme le montrent les figures 2.3(c) et2.3(d). Bien que les deux nouveaux modèles aient le même processeur, GPU et RAM que leurs versions précédentes, plusieurs mises à jour et de nouvelles fonctionnalités ont été ajoutées, telles qu'un support de mémoire micro SD remplaçant un emplacement pour carte SD, une version améliorée de GPIO de 26 à 40 broches, un son à faible bruit et une consommation d'énergie réduite de 0,5 à 1 W en ayant un régulateur a découpage au lieu d'un régulateur linéaire. De plus, le modèle 1A + est environ 2 cm plus court que son prédécesseur, tandis que le modèle B+ fournit quatre ports USB au lieu de deux dans le modèle B.

Au début de 2015, la nouvelle génération d'ordinateur Raspberry Pi, appelée Raspberry Pi modèle B, est sortie. Le modèle représente une amélioration significative des capacités de Raspberry Pi et a été accueilli avec enthousiasme par la communauté. Raspberry Pi 2 modèle B utilise un BCM2836 beaucoup plus puissant qui offre 900 MHz quadri-cœur ARM Cortex-A7 CPU, 1 Go de RAM et les mêmes spécifications de

processeur graphique qu'auparavant, ce qui lui permet de fonctionner environ six fois plus rapide que ses prédécesseurs sans changement prix. De plus, les supports de la version de Windows 10 Internet des objets (IdO) sont disponibles gratuitement pour le Raspberry Pi 2 modèle B, offrant ainsi un grand potentiel pour une utilisation future.

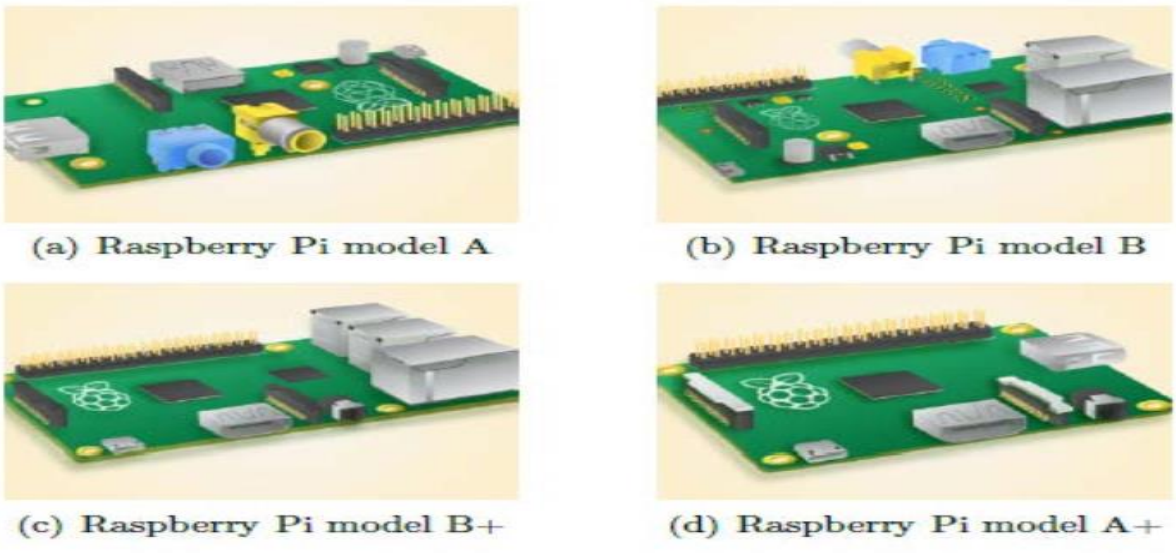


Figure 2.3 : Les modelés de Raspberry Pi.

Chacun de ces modèles ayant des caractéristiques bien spécifiques pour cela nous avons décidé de présenter les caractéristiques de chaque modèle dans un schéma qui fait une comparaison entre ces différents modèles (tableau 2.1) [6] :
comparaison entre ces différents modèles (tableau 2.1) :

Raspberry Pi	Génération 1			Génération 2	
Spécification	Modelé A	Modelé B	Modelé A+	Modelé B+	Modelé B
Power	300mA	700mA	200mA	600mA	900mA
Ethernet Port	No	Yes	No	Yes	Yes
USB Port	1	2	1	4	4
GPIO	26	26	40	40	40
SD Carte Slot	SD	SD	MicroSD	MicroSD	MicroSD
SoC	BCM2835	BCM2835	BCM2835	BCM2835	BCM2856

CPU	700MHz ARM11 Single-core	700MHz ARM11 Single-core	700MHz ARM11 Single-core	700MHz ARM11 Single-core	900MHz ARM Cortex-A7 Quad-core
RAM	256MB	512MB	256MB	512MB	1GB

Tableau 2.1 : les principales caractéristiques des modèle RASPBERRY Pi

2.2.2 Composants standards de Raspberry Pi [5]

La figure 2.4 suivante représente un Raspberry standard :

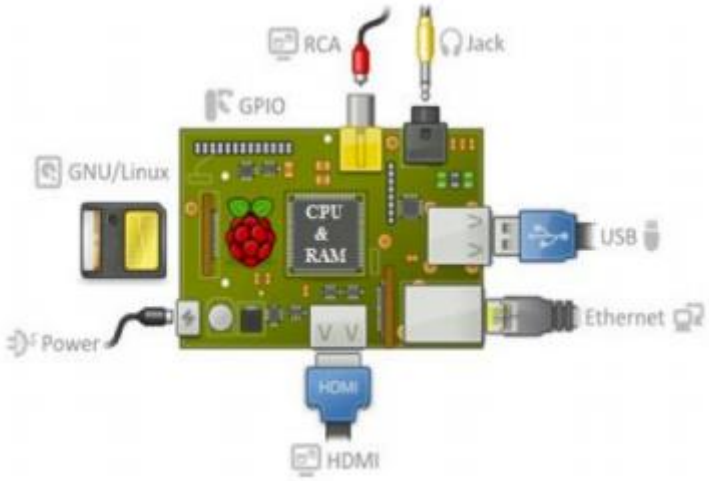


Figure 2.4 : Les composants standards d'un Raspberry Pi

- **Processeur ARM :** Les architectures ARM sont des architectures de processeurs, à faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd ».
- **Mémoire vive RAM :** C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement.

Une connectique variée :

- **HDMI :** « High Définition MultiMedia Interface » permet de relier le Raspberry PI à un dispositif compatible : écran LCD ou un vidéoprojecteur…
- **Port USB 2.0 :** Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques.

- **Port Ethernet :** C'est un port qui correspond au protocole international ETHERNET de réseau local à commutation de paquets.
- **Prise RCA :** « Radio Corporation of America » est un connecteur électrique utilisé dans le domaine audio/vidéo.
- **Un slot les cartes SD :** Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe.
- **Une prise jack :** C'est une connectique audio-vidéo.
- **GPIO :** « General Purpose Input/Output » sont des ports d'Entrée/Sortie.

a. Module camera raspberry Pi

Dans notre projet on va utiliser une caméra pour prendre des photos en cas d'une intrusion à la salle du serveur, le meilleurs choix est d'utiliser le module de caméra dédie au Raspberry Pi ; La caméra achetée est « Raspberry Pi Camera Rev 1.3 » dispose d'un capteur de 5 méga pixels Sony IMX219(figure2.5).

Le module de caméra peut être utilisée pour prendre la vidéo haute définition (HD), ainsi que de prendre des photos. Il prend en charge 1080p30, 720p60 et modes vidéo VGA90. L'appareil fonctionne avec tous les modèles de Raspberry Pi 1, 2 et 3, et ce qui est très important c'est qu'il y a de nombreuses bibliothèques déjà construites pour la commande de cette caméra. Parmi elles, on trouve le **Picamera** bibliothèque Python qu'on va utiliser par la suite [7].

Caractéristiques de la camera :

- La caméra se branche sur le connecteur CSI existant sur la carte Raspberry Pi par câble plat à l'interface 15-pin.
- Capteur 5 Mégapixels.
- Résolutions photo 2592 x 1944
- Résolutions vidéo 1080p à 30fps (30 images/s), 720p à 60fps et640x480p à 60fps ou 90fps
- Dimension du capteur 3280 x 2464 pixels

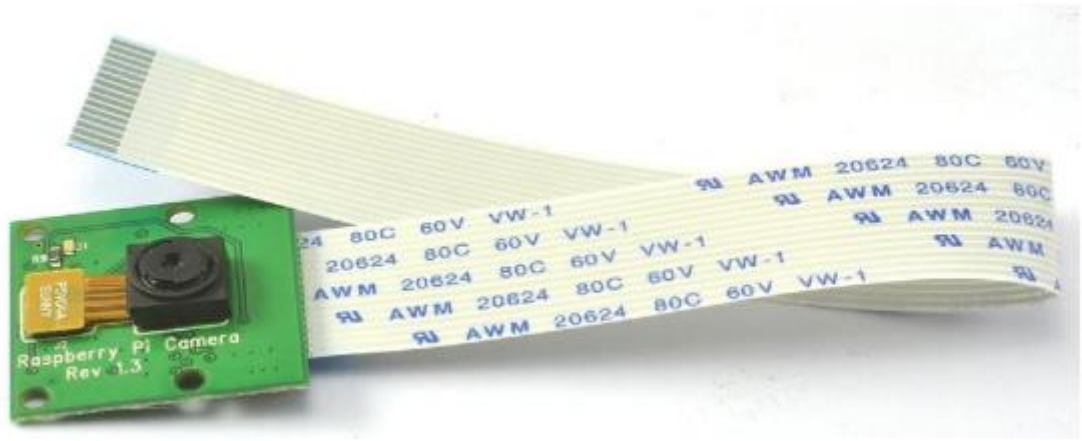


Figure 2.5 : Le module camera raspberry Pi Rev 1.3

2.3 Carte Arduino [7]

Arduino est le nom d'une gamme de cartes à microcontrôleur programmable et a plusieurs entrées et sorties numériques et analogiques. Elle permet de piloter un système à partir du programme mis dans sa mémoire. Elles utilisent toutes un même logiciel de programmation multiplateforme appelé logiciel Arduino également.

Il existe toute une gamme de carte Arduino chacune ses caractéristiques, utilisées selon le besoin de l'utilisateur. Dans ce projet on s'intéresse à la carte Arduino Méga et Uno.

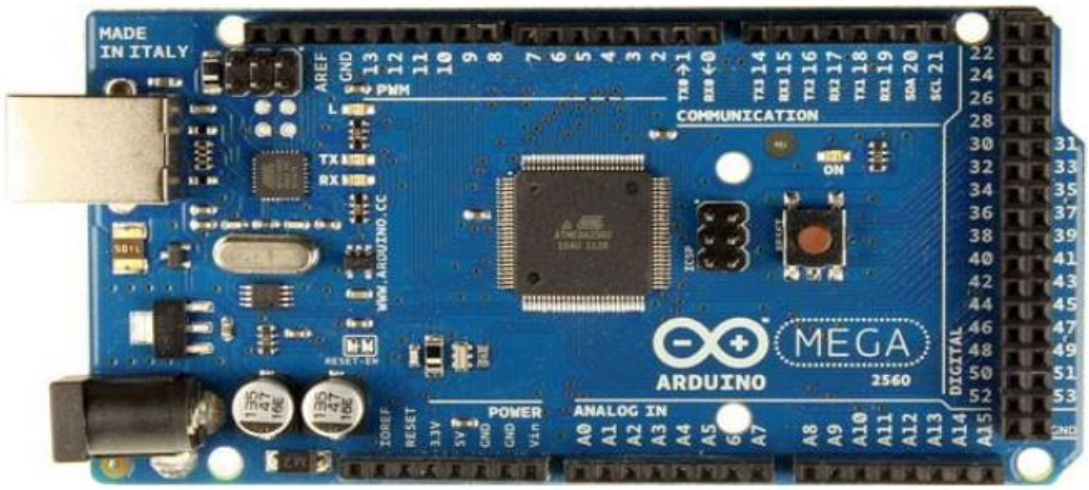


Figure 2.6 : La carte Arduino Méga

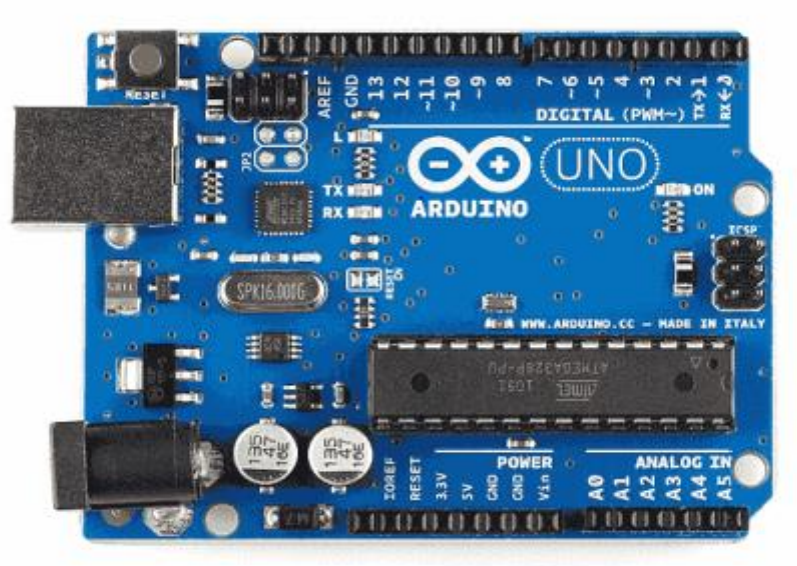


Figure 2.7 : La carte Arduino Uno

Les caractéristiques techniques des deux cartes Uno et Méga [7] :

	Carte Arduino Uno	Carte Arduino Méga
Microcontrôleur	ATmega328	ATmega2560
Tension de fonctionnement	5 V	5V
Tension d'entrée (recommandée)	7 à 12 V	7 à 12 V
Tension d'entrée (limites)	6 à 20 V	6 à 20 V
Bronches E/S numériques	14 (dont 6 fournissent la sortie PWM)	54 (dont 14 fournissent la sortie PWM)
Bronches d'entrée analogiques	6	16
Courant alternatif par broche d'E/S	40 mA	40 mA
Courant continu pour la bronche de 3.3V	50 mA	50 mA
Mémoire flash	32 KB (dont 0.5 KB utilises par le chargeur initial de programme)	256 Ko (dont 8 Ko utilises par le chargeur initial de programme)
SRAM	2 KB	8 Ko
EEPROM	1 KB	4 Ko
Vitesse de l'horloge	16 MHz	16 MHz

Tableau 2.3 : Les caractéristiques techniques des carte Arduino Uno et Méga

2.4 Répartition des taches [7]

Dans ce projet, l'Arduino et Raspberry Pi forment une association importante. Chaque carte est censée transmettre à l'autre, les bonnes informations afin d'assurer le bon fonctionnement du système. Pour profiter des avantages de chaque carte, il est important de distribuer les taches selon la puissance, et la robustesse de chacune. L'Arduino est adaptée pour un type de travail répété, mais Raspberry Pi peut faire des choses plus complexes. Etant donné que le Raspberry Pi est 40 fois plus rapide que l'Arduino, et grâce à son module camera, il va s'occuper de la partie traitement d'images, quant à l'Arduino, elle assure le bon fonctionnement des actionneurs et des capteurs du projet selon les informations reçues.

2.5 Servo moteur

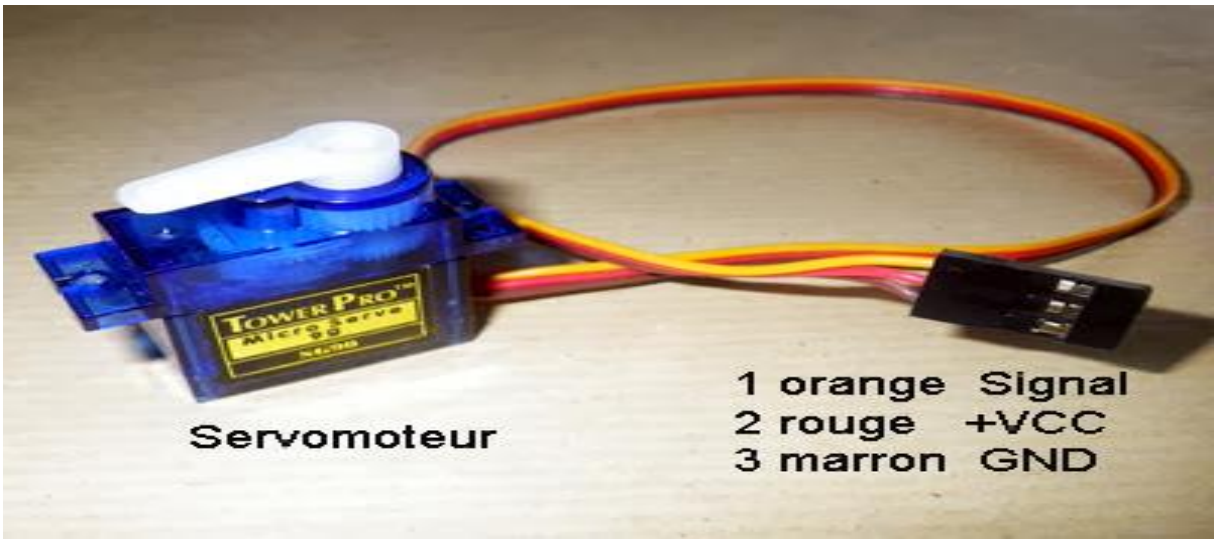


Figure 2.8 : servomoteur

Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe, C'est un actionneur (système produisant une action) qui mélange l'électronique, la mécanique et l'automatique.

Un servomoteur est composé :

- d'un moteur à courant continu
- d'un axe de rotation

- un capteur de position de l'angle d'orientation de l'axe (très souvent un potentiomètre)

- une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu

Un servomoteur est capable d'attendre des positions prédéterminées dans les instructions qui lui on était donné, puis de les maintenir.

Le servomoteur à l'avantage d'être asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur respectera la consigne d'instruction que vous lui avez envoyée en son entrée. Même si un obstacle si tiens sur la route, qui viendrait à lui faire changer l'orientation de sa trajectoire, le servomoteur essaiera de conserver la position. Pour un ajustement précis de la position, le moteur et son réglage sont équipés d'un système de mesures qui détermine la position courante.

2.5.1 Gestion du servo moteur

-Déplacer le servo moteur à la position désirée et s'arrêter, plage limitée de 180 ° dans 2 directions

3 fils : * rouge = puissance

* marron = sol

* jaune = contrôle

Circuit de contrôle + potentiomètre + Moteur DC + arbre de position avec engrenages

Le signal de commande utilise PWM, largeur d'impulsion = position

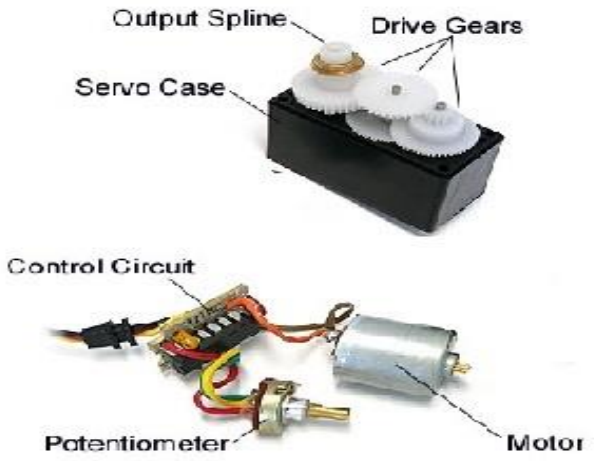


Figure 2.9 : Composants du servomoteur

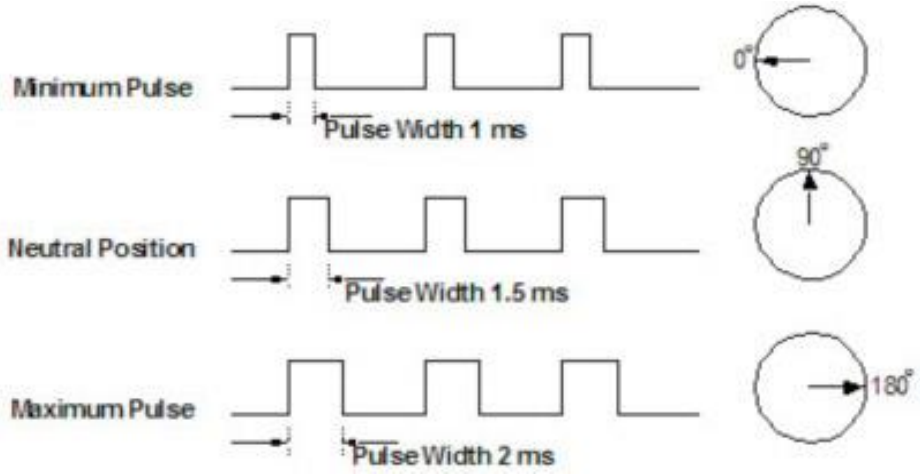


Figure 2.10 : Position en fonction de la pulsation

Calibrez le moteur :

- Connecter le servomoteur à la broche PWM, à la puissance et à la masse.
- Ouvrir skeleton_motor_calibrate.
- Trouver les positions "ouvertes" et "fermées" en balayant l'arbre.

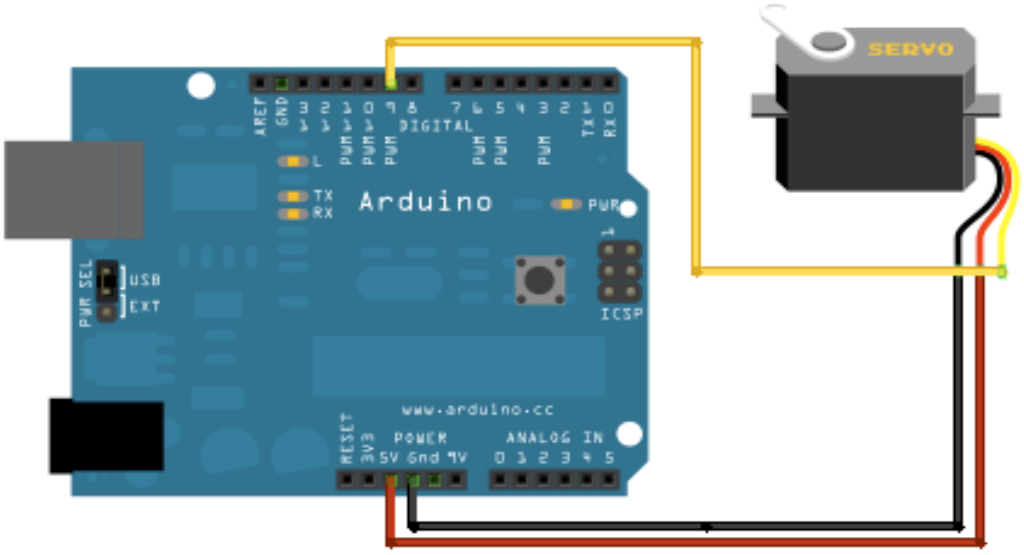


Figure 2.11 : Montage simple d'un servomoteur avec l'Arduino Position en fonction de la pulsation

Bibliothèque servo :

#include <Servo.h>

Créer un objet

Object.attach (broche)

Object.write (position) * en degrés

2.6 Présentation de la librairie open cv

2.6.1 Qu'est-ce qu'open cv

Une multitude d'outils logiciels et de bibliothèques pour manipuler les images et les vidéos sont disponibles, mais pour toute personne souhaitant développer une application de vision intelligente, la bibliothèque OpenCV est l'outil à utiliser. Depuis son introduction en 1999, il a été largement adopté comme principal outil de développement par la communauté de chercheurs et développeurs en « computer vision ». Vous trouverez ci-dessous la liste des principales raisons pour lesquelles on a choisi OpenCv pour développer notre projet [8].

2.6.2 Bibliothèque multiplateforme

OpenCV est une bibliothèque multiplateforme, permettant aux utilisateurs de différents systèmes d'exploitation pour l'utiliser sans complications. Il est même accessible sur les systèmes mobiles comme iOS et Android, ce qui en fait une bibliothèque vraiment portable.

2.6.3 Vastes algorithmes

OpenCV (Open source Computer Vision) est une bibliothèque qui contient plus de 500 algorithmes optimisés pour l'analyse d'images et de vidéos. Cette vaste bibliothèque permet aux programmeurs d'effectuer une multitude de tâches dans leur logiciel telles que l'extraction des modèles 3D pour les objets, détection et suivi d'objets, etc.

2.6.4 Utilisation intensive

OpenCV est utilisé par des entreprises géantes comme Google,IBM, Toyota et des startups comme Applied Minds et Zeitera ainsi que des organisations dans les pays du monde pour mener des tâches multiples. Cela donne aux utilisateurs l'assurance qu'ils sont à bord d'une bibliothèque qui est largement utilisé par les entreprises et le gouvernement institutions. De plus, OpenCV dispose d'une vaste communauté où les utilisateurs peuvent demander de l'aide et offrir de l'aide à leurs collègues

développeurs au cas où ils auraient des questions concernant les codes ou la plateforme. Cela permet aux développeurs accéder aux informations de personnes réelles sur la bibliothèque et ses codes.

2.6.5 Efficacité

OpenCV a été conçu pour une efficacité de calcul et avec forte concentration sur les applications en temps réel. Écrit en C / C ++, la bibliothèque peut tirer parti de traitement multicœur. Activé avec OpenCL, elle peut aussi profiter de l'accélération matérielle du plate-forme de calcul hétérogène.

2.6.6 Fonctionnalité [8]

La bibliothèque OpenCV 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. Elle propose la plupart des opérations classiques en traitement bas niveau des images et des vidéos.

- **Traitement d'images**

Cette partie propose la plupart des opérations classiques en traitement bas niveau des images :

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- Lissage, filtrage.
- Seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- Segmentation (composantes connexes, GrabCut).
- Morphologie mathématique.

- **Traitement vidéo**

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra).

- Détection de droites, de segment et de cercles par Transformée de Hough.
- Détection de visages par la méthode de Viola et Jones.
- Cas cade de classifier boostés.
- Détection de mouvement, historique du mouvement.
- Pour suite d'objets par mean-shiftou Camshift.
- Détection de points d'intérêts.
- Estimation de flux optique (Méthode de Lucas–Kanade).
- Triangulation de Delaunay.
- Diagramme de Voronoi.
- Enveloppe convexe.
- Ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés.

2.7 Méthodologie utilisée

2.7.1 Identification des contours et techniques utilisées

Ce projet a été développé selon trois directions principales. La première direction se concentre sur les techniques de traitement d'image afin de détecter et de classer des formes régulières présentes dans une image en utilisant leurs contours ou leurs délimitations par rapport au fond (convoyeur). La deuxième direction se concentre sur la façon dont la sortie produite dans la première étape sera transmise en Raspberry et comment la Raspberry interagira avec cette sortie. La troisième direction est la construction d'une matérielle plateforme dans laquelle tous les résultats obtenus seront combinés et solidifiera le but de ce travail.

« Un contour est une courbe fermée de points ou de segments de ligne, représentant les limites d'un objet dans une image. » **[9]**

En d'autres termes, les contours représentent les formes d'objets trouvés dans une image. Si le détail interne est visible dans une image, l'objet peut produire plusieurs contours, qui sont renvoyés dans des structures de données hiérarchiques. Une fois les contours des objets sont détectés, nous pouvons déterminer le nombre d'objets dans

une image, classer les formes des objets, mesurer la taille des objets, ……etc. L'entrée du processus de recherche de contour est une image binaire, elle est produite en appliquant d'abord un seuil et / ou un bord technique de détection. Dans notre travail cette image binaire représente des objets de couleur blanche sur un fond de couleur noire. Un point qui mérite d'être mentionné c'est qu'il ne suffit pas d'identifier simplement les frontières pixels d'un motif afin d'extraire son contour, mais il faut extraire une séquence ordonnée des pixels de la frontière à partir de laquelle nous pouvons extraire la forme générale du modèle. n. **[10]**

Le traçage des contours est l'un des nombreuses techniques de prétraitements effectué sur des images numériques afin d'extraire des informations sur leur forme générale. Une fois le contour d'un motif donné est extrait, ces différentes caractéristiques seront examinées et utilisées comme caractéristiques pour plus tard dans la classification des modèles.

Par conséquent, une extraction correcte du contour va produire des fonctionnalités plus précises qui augmenteront la chance de classer correctement un modèle donné.

« Le seuillage est une opération non linéaire qui convertit une image en niveaux de gris en une image binaire où les pixels sont divisés en deux groupes en fonction d'une valeur définie comme valeur seuil. Les pixels qui ont une valeur supérieure à la valeur seuil reçoivent la valeur de 1 tandis que d'autres prennent la valeur 0. » **[11]**

Le seuillage examine simplement les intensités et voit si chaque valeur est plus petite ou plus grande et on obtient respectivement des points "edge".

2.8 Conclusion

Dans ce chapitre nous avons étudié la partie matérielle (Raspberry, arduino, servo moteur). La Bibliothèque OpenCV permet aux utilisateurs de créer des applications de vidéo et d'images complexes. La détection de formes ou détection de contour dans les images est un des problèmes les plus difficiles en vision par ordinateur puisque celle-ci est indispensable pour toutes sortes d'applications

3.1 Introduction

Ce chapitre a pour but de préparer l'image pour un autre niveau de traitement afin de bien extraire les informations pertinentes. Cette préparation nécessite l'utilisation de différentes étapes qui sont expliqués dans ce chapitre, notamment, la conversion de la couleur en niveau de gris, le filtrage, la binarisation, la morphologie mathématique et le suivi de contour.

3.2 Prétraitement de l'image :

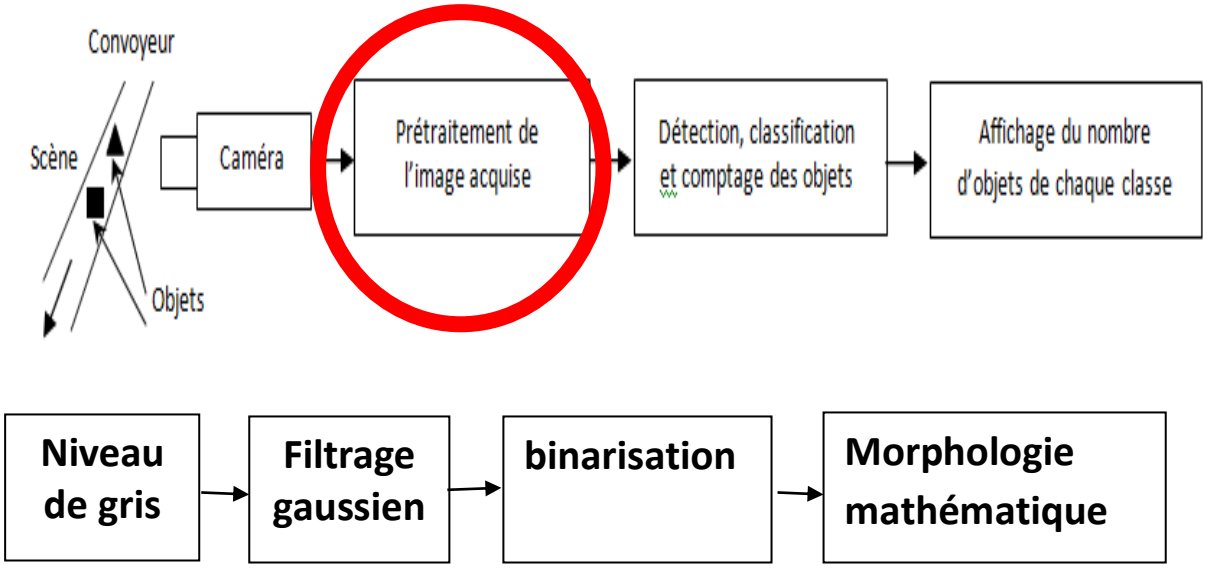


Figure 2.11.1 : Partie prétraitement de l'image

Cette préparation nécessite l'utilisation de différentes étapes qui sont expliqués dans ce schéma, notamment, la conversion de la couleur en niveau de gris, le filtrage, la binarisation, la morphologie mathématique.

3.3 Passage d'une image couleur vers une image à niveaux de gris

Pour passer d'une image couleur (RVB) vers une image à niveaux de gris ou noir et blanc, on doit exploiter la relation suivante :

G = 0,299*R + 0,587*V + 0,114*B…… (1)

Avec G est le niveau de gris du pixel, et R, V et B sont les composantes rouge, vert et bleue du pixel. Pour afficher une image à niveaux de gris sur un écran couleur, les trois composantes RVB doivent être identiques comme suit :

R = V = B = G……. (2)

L'application de cette conversion est illustrée par la **figure 3.1** :

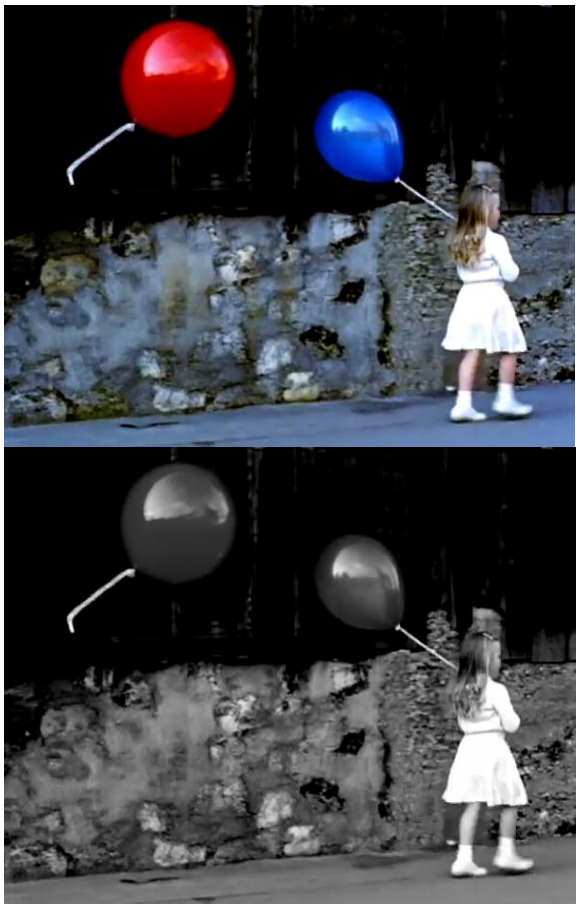


Figure 3.1: Conversion d'image couleurs en niveaux de gris

3.4 Filtrage

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

Le filtrage a pour but de réduire les effets du bruit sans affecter trop le signal ou l'image. On peut considérer deux types de filtres ; les filtres linéaires et les filtres non-linéaires.

3.4.1 Filtrage linéaire

L'hypothèse fondamentale derrière le filtrage linéaire est que la moyenne de plusieurs échantillons devrait réduire le bruit (i.e. l'écart-type du signal résultat du moyennage de N échantillons devrait être plus faible que celui de la distribution de laquelle proviennent ceux-ci).

Le filtrage linéaire se fera donc par la convolution d'un filtre exprimé dans le domaine spatial avec l'image. On appelle ce filtre un opérateur de convolution. Cet opérateur de convolution prend la forme d'un masque ou noyau ("kernel") de convolution. L'opération de filtrage consiste à convoluer ce masque avec l'image.

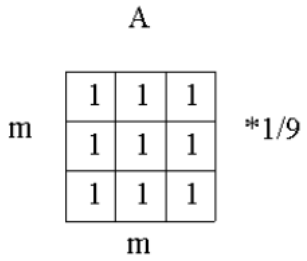


Figure 3.2 : Exemple de masque de convolution « le filtre moyenneur »

L'équation de convolution pour le filtrage linéaire d'une image E(x,y) avec un filtre de noyau A(h,k) est la suivante pour chaque pixel d'illuminance E(i,j):

$$E_{filtre}(i,j) = A(h,k) \bullet E(i,j) = \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h,k)E(i-h,j-k) \quad (3)$$

Pour chaque pixel (i,j), les valeurs d'illuminance E(i,j) des pixels couverts par le masque A(h,k) sont multipliées par les valeurs du masque et additionnées pour produire la sortie du filtre.

La réponse de tels _ltres est tout simplement une moyenne pondérée des valeurs des pixels du voisinage. La pondération dépend du masque utilisé. Ces masques sont ainsi parfois appelés _ltres moyenneurs, ou encore _ltres passe-bas, ce dernier nom étant en relation avec leur équivalent dans le domaine fréquentiel. Deux exemples de tels filtres se trouvent sur la **figure 3.3**. **[12]**

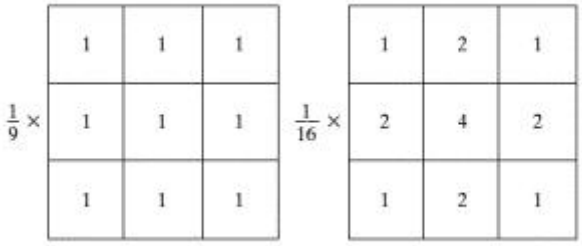


Figure 3.3 : Deux masques moyenneurs classiques

3.4.2 Filtre moyenneur

Le filtre moyenneur le plus simple est associé au premier masque de la figure.
La réponse à un tel filtrage se traduit par :

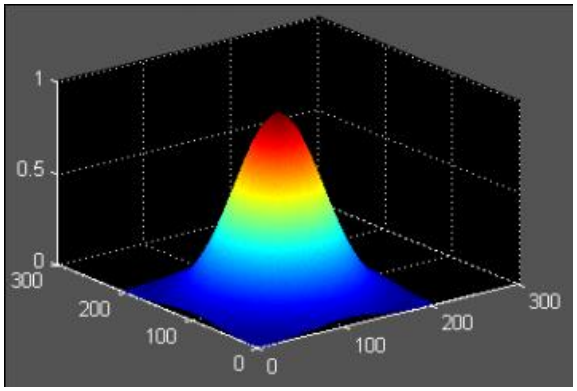
$$R = \frac{1}{9}\sum_{i=1}^{i=9} z_i \qquad (4)$$

Il s'agit tout simplement de la moyenne des valeurs de 9 pixels appartenant au voisinage 3×3 du pixel considéré. On remarque que les coefficients du masque sont 1 et non 1/9. Il est en effet moins coûteux de réaliser en chaque pixel une opération d'addition et de diviser toutes les valeurs des pixels de l'image par 9 une fois que l'image a été entièrement filtrée. La **figure 3.3** montre l'application du filtre moyenneur pour différentes tailles du masque.

3.4.3 Filtre gaussien

Comme son nom le dit, le filtre gaussien possède un noyau de convolution de forme gaussienne La forme du filtre est obtenu de l'équation d'une gaussienne :

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \qquad (5)$$



A

m

1	2	1
2	4	2
1	2	1

*1/16

m

Figure 3.4 : Exemple de masque de convolution « le filtre gaussien »

Le filtre gaussien préserve les contours mieux que le filtre moyenneur qui tends à lisser d'avantage les contours pour de grande taille du masque. La **figure 3.5** montre la différence entre l'application du filtre moyenneur et gaussien

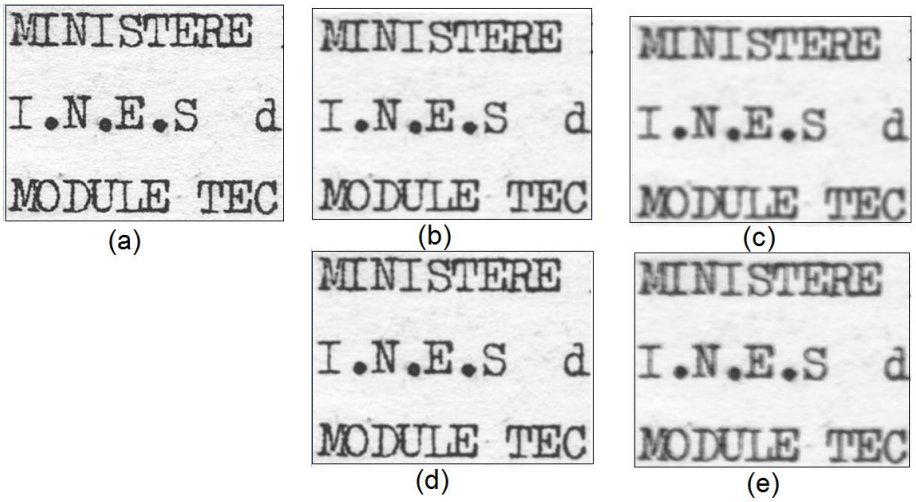


Figure 3.5 : Résultats de l'application des deux filtres ; moyenneur et gaussien. L'image (a) représente l'image d'entrée, les image (b) et (c) sont des images filtrées un filtrage moyen avec des masques de taille 3x3 et 5x5. Les image (d) et (e) sont des images filtrées un filtrage gaussien avec des masques de taille 3x3 et 5x5.

3.5 Seuillage

Le seuil du seuillage ou binarisation consiste à transformer l'image codée sur 6, 8 ou 16 bits, en une image binaire créé sur 1 bit.

Le seuillage permet de sélectionner les parties de l'image qui intéressent l'opérateur, par exemple 2 types de grains (blancs et gris) dans un mélange. On peut donc, par exemple, attribuer à tous les pixels de l'image numérique qui ont un niveau de gris compris entre deux valeurs i1 et i2, choisies par l'opérateur, la valeur 1 à tous les autres pixels est attribuée la valeur 0.

$$I_b(i,j) = \begin{cases} 1 & si\ \ I_b(i,j) \geq s \\ 0 & si\ non \end{cases}$$

Après seuillage, les parties des images sélectionnées seront traduites en noir et blanc.

3.5.1 Seuillage global

La première solution pour fixer le seuil est de procéder par tâtonnement à partir de l'image en noir et blanc résultante censée mettre en évidence les contours.

Le seuil correspond à une valeur réelle dans l'intervalle [0,1]. Au préalable, il est nécessaire de normaliser l'image du module du gradient afin que tous les pixels se trouvent également dans l'intervalle [0, 1]. Cette méthode est très simple mais peu efficace.

3.5.2Seuillage local

Dans cette technique de seuillage contrairement aux précédentes, le traitement n'est pas identique en tout point de l'image. On s'intéresse ici aux pixels avoisinant les contours les plus significatifs de l'image. L'idée est de garder les contours les plus forts de l'image mais en essayant d'assurer leur continuité. Deux seuils sont nécessaires pour implanter la technique : un seuil haut Sh et un seuil bas Sb. Le seuil haut va servir à sélectionner les contours les plus significatifs dans l'image du module du gradient. Ces contours sont contenus dans l'image résultante en noir et blanc. Le seuil bas permet de mettre en évidence des contours moins forts de l'image. Ces contours sont conservés dans l'image résultante seulement s'ils sont situés dans le voisinage des contours les plus significatifs mis en évidence par le seuillage avec Sh. Généralement le voisinage est défini par les huit voisins.

3.5.3 Binarisation par seuillage

Cette technique consiste à réduire la dynamique d'une image (plusieurs niveaux de gris) à deux niveaux (noir et blanc). Ceci revient à séparer les pixels de l'image en deux classes, la première ayant un niveau maximal (typiquement 255) et la seconde un niveau minimal (0).

La méthode consiste à calculer un seuil d'après l'histogramme de l'image, ce seuil doit être le Plus adéquat possible pour ne pas altérer les informations pertinentes de l'image (**figure3.6**) **[13]**.

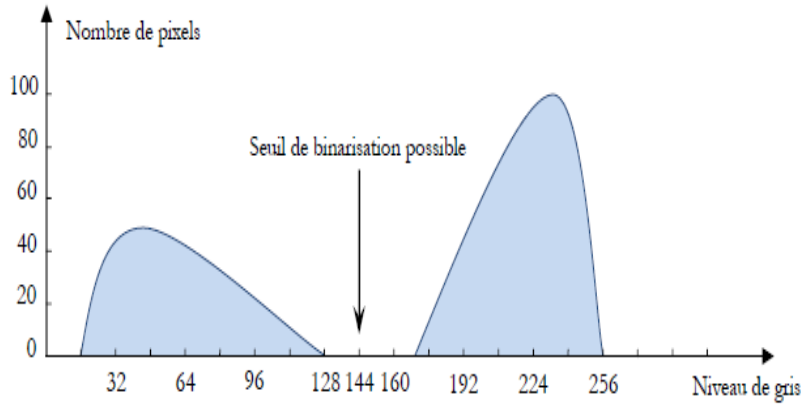


Figure 3.6 : Choix du seuil de binarisation sur l'histogramme.

3.6Méthode d'Otsu

3.6.1 Définition

En vision par ordinateur et traitement d'image, la méthode d'Otsu est utilisée pour effectuer un seuillage automatique à partir de la forme de l'histogramme de l'image, ou la réduction d'une image à niveaux de gris en une image binaire. L'algorithme suppose que l'image à binariser ne contient que deux classes de pixels, (c'est-à-dire le premier plan et l'arrière-plan) puis calcule le seuil optimal qui sépare ces deux classes afin que leur variance intra-classe soit minimale.

Image originale Image binarisée par l'algorithme d'Otsu

Figure 3.7 *:* Image binarisée par la méthode d'Otsu**.**

Dans la méthode d'Otsu, le seuil qui minimise la variance inter-classe est recherché à partir de tous les seuillages possibles :

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \text{ .....(6)}$$

Les poids ω_i représentent la probabilité d'être dans la n-ième classe, chacune étant séparée par un seuil t. Finalement, les σ_i^2 sont les variances de ces classes.

Otsu montre que minimiser la variance inter-classe revient à maximiser la variance intra-classe :

$$\boldsymbol{\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2} \text{ .....(7)}$$

Qui est exprimée en termes des probabilités de classe ω_i et des moyennes de classes μ_i qui à leur tour peuvent être mises à jour itérativement. Cette idée conduit à un algorithme efficace.

3.6.2 Les étapes de l'algorithme

Cette méthode de binarisation nécessite au préalable le calcul de l'histogramme. Puis, la séparation en deux classes est effectuée.

a) Calcul de l'histogramme

Le calcul de l'histogramme est très simple.

On initialise un tableau hist avec des 0. Généralement, ce tableau est constitué de 255 cases correspondant aux 255 niveaux de gris d'une image.

Ensuite, si p (i, j) représente la valeur du pixel au point (i, j), on balaye toute l'image et on compte le nombre de fois ou un niveau de gris apparait.

b) Séparation en 2 classes

La séparation se fait à partir des moments des deux premiers ordres : la moyenne et l'écart-type. Pour que le procédé soit indépendant du nombre de points dans l'image N, on normaliser l'histogramme : $p_i = n_i/N$ ou n_i représente le nombre de pixels de niveau i.

On peut calculer alors les deux moments utilisés :

$$\boldsymbol{\mu(k) = \sum_{i=1}^{i=k} i * P_i} \qquad \boldsymbol{et} \qquad \boldsymbol{\omega(k) = \sum_{i=1}^{i=k} P_i} \text{ ..... (8)}$$

On note $\mu_T = \mu(256)$, où 256 est le nombre totale de niveaux de gris.

Si on appelle $omega_0$ la probabilité de la classe C_0 et $omega_1$ la probabilité de la classe C_1, alors:

$\omega_0 = \omega(k^*)$ où k* représente le niveau de seuil.

Et $\omega_0 = 1 - \omega(k^*)$.

Si on note de même μ_1 et μ_0 avec :

$\mu_0 = \mu(k^*)/ \omega(k^*)$ $\mu_1 = (\mu_T - \mu(k^*))/(1- \omega(k))$

Or l'image totale conserve certaines propriétés, d'où on peut tirer les relations :
$\omega_0^* \mu_0 + \omega_1^* \mu_1 = \mu_T$ et $\omega_0 + \omega_1 = 1$

En introduisant un paramètre pour évaluer la qualité du niveau de seuillage, on obtient :

$S^2 = \omega_0 * \omega_1 (\mu_1 - \mu_0)^2$. La valeur précédente est fonction de k.

On calcule donc cette valeur pour les 256 niveaux de gris de l'image. En fait on peut déjà enlever les valeurs 0 et 255 qui correspondent à affecter tous les pixels à la même classe.

A partir de ω(k) et μ(k), on calcule donc :

$$S^2(k) = \omega(k)*(1- \omega(k))*(\mu_T * \omega(k) - \mu(k))^2$$

La valeur du seuil k* est obtenue pour le maximum de S^2.

Il ne reste plus qu'à comparer la valeur de tous les pixels de l'image au seuil ainsi trouvé. Les résultats de l'application de cette méthode sont représentés par la **figure 2.8**

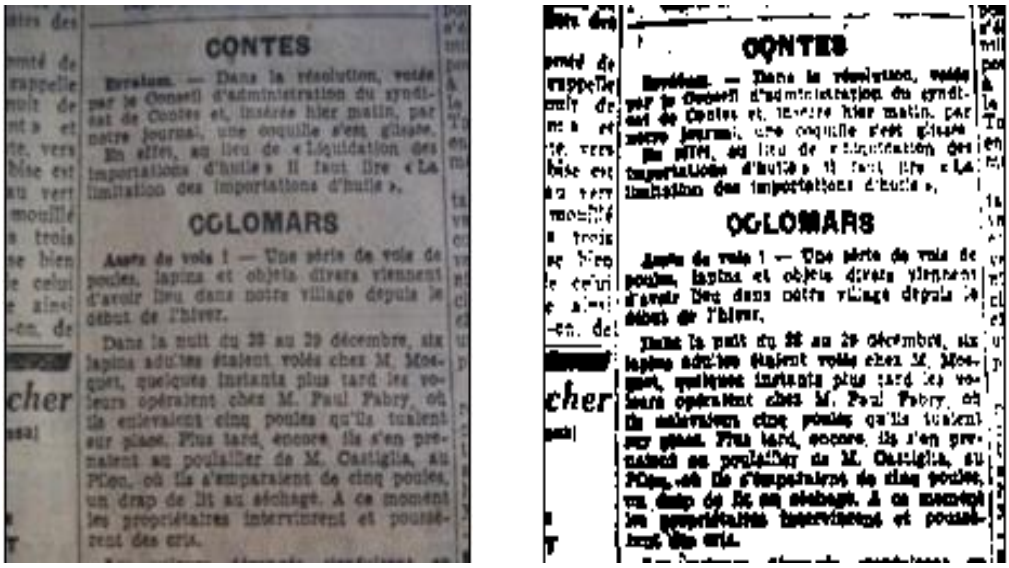


Image initiale. Image binaire.

Figure 3.8 : Image binarisée par Otsu**.**

On remarque la présence d'un bruit important dans le fond de l'image. Cependant on arrive à distinguer convenablement les caractères.

3.7 Filtres morphologiques

La morphologie mathématique est dédiée à l'analyse des structures spatiales. Elle est dite morphologique parce qu'elle vise à analyser la forme (morphologie) des objets. Une image est vue comme un ensemble de formes (voir **figure 3.9**).

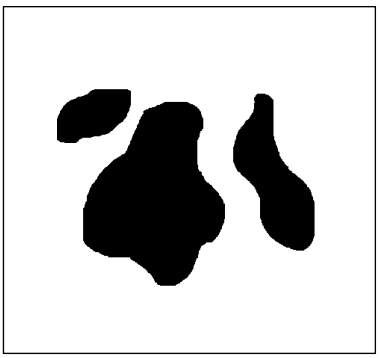


Figure 3.9 : Image formée d'objets.

Les filtres morphologiques sont souvent utilisés pour éliminer des pixels isolés considérés comme un bruit dans une image. Pour analyser les images, ces méthodes utilisent des formes connues appelées éléments structurants (en anglais ; structuring elements) comme l'indique la **figure 3.10**.

Un élément structurant est un masque binaire muni d'un point d'ancrage (point de fixation) comme l'indique la **figure 3.10**.

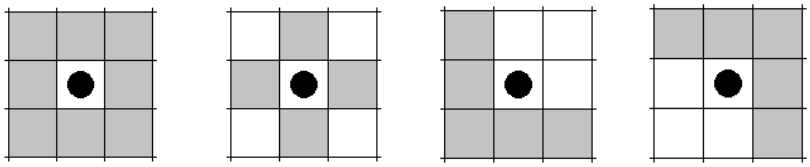


Figure 3.10 : Eléments structurants.

L'analyse d'images par la morphologie mathématique nécessite l'utilisation d'opérations morphologiques, Parmi ces opérations il y a la dilatation, l'érosion, l'ouverture et la fermeture mathématiques.

3.7.1 Dilatation

La dilatation a pour objectif de dilater les objets ou formes dans une image. Pour une image binaire X et un élément structurant B, nous avons :

$$D^{B}(x) = X \oplus B = \left\{ x \,\middle|\, Bx \cap X \neq \Phi \right\} \quad (9)$$

où $$Bx = \left\{ b + x \,\middle|\, b \in B \right\} \quad (10)$$

B_x l'ensemble de B translaté de x.

Dans une image binaire cette opération permet d'éliminer les pixels noirs isolés mais ajoute des pixels blancs au contour des objets présents dans l'image. Le résultat de cette opération est l'augmentation de la taille de ces objets.

3.7.2 Erosion

L'érosion a pour objectif de d'éroder les objets ou formes dans une image. Pour une image binaire X et un élément structurant S, nous avons :

$$\mathrm{E}^{\mathrm{B}}(x) = \mathrm{X} \oplus \mathrm{B} = \left\{ x \,\middle|\, \mathrm{B}x \subseteq \mathrm{X} \right\} \quad (11)$$

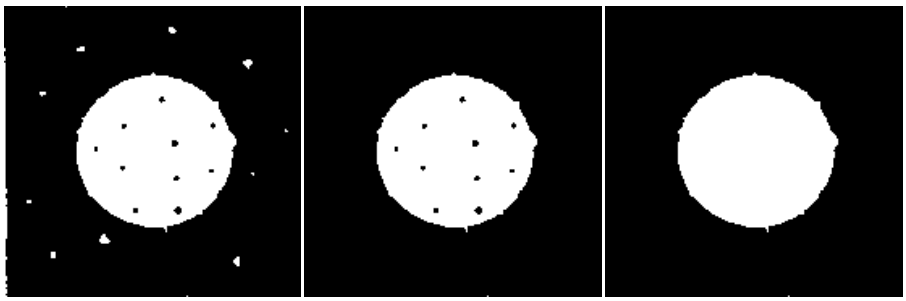
Elle permet d'éliminer les pixels blancs isolés au milieu des parties noires de l'image. Le résultat de cette opération est la diminution de la taille des objets présents dans l'image.

3.7.3 Ouverture

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet d'éliminer les taches blanches dans le fond de l'image.

3.7.4 Fermeture

La fermeture est l'opération inverse de l'ouverture, qui consiste à faire subir à l'image une dilatation suivie d'une érosion. Elle permet d'éliminer les noirs qui se trouvent dans l'objet **[14].**

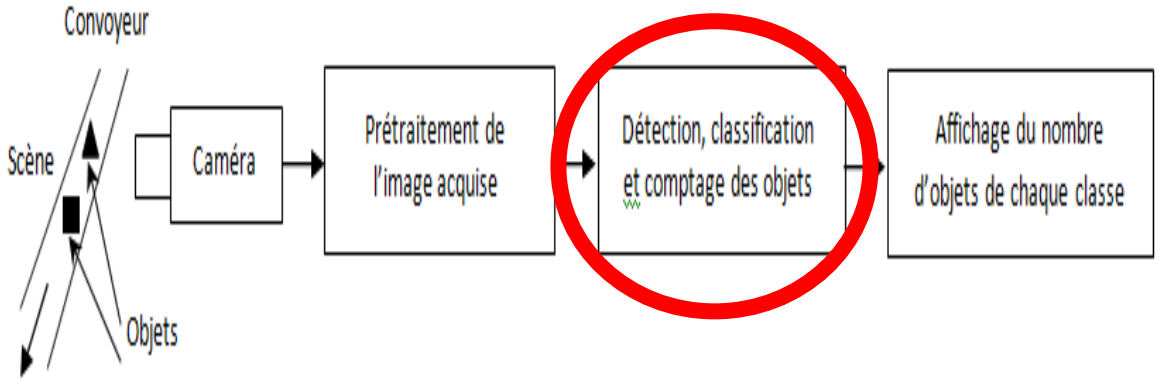


(a) (b) c)

Figure 3.11 : Analyse d'images par morphologie (a) Image d'entrée binaire. (b) Résultat d'ouverture avec un disque de 9x9. (c) (b) Résultat de fermerture avec un disque de 9x9.

Figure 3.12 : Prétraitement de l'image acquise par caméra ; (a) Image binarisé. (d) Résultat d'ouverture suivi de fermeture avec un élément structurant disque de taille 7x7.

3.8 Détection, classification et comptage des objets



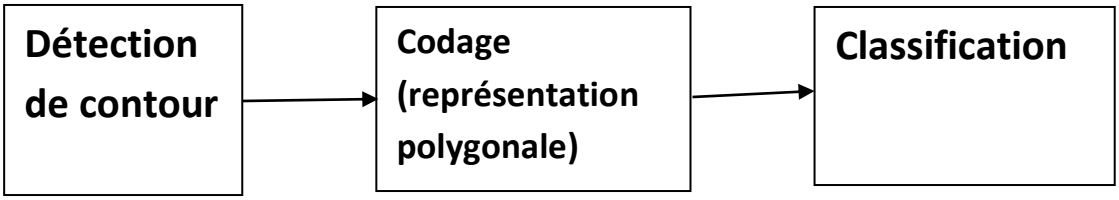


Figure 3.12.1 : Partie détection ,classification et comptage des objets .

Dans cette partie nous allons présenter les étapes qui sont expliqués dans ce schéma. Notamment, la détection de contour, codage (représentation polygonale), et classification.

3.9 Présentation de la méthode du suivi de contour

Cette méthode est très rapide car elle ne nécessite que peu de calculs à chaque étape.

Elle présente l'avantage de détecter les pixels frontières suivant toutes les directions. Tous les pixels seront traités une seule fois.

Le programme de suivi ainsi réalisé, permet de détecter les contours d'objet d'images binaires pouvant être représentées de la façon suivant :

Les objets sont constitués de pixels « noirs », la valeur « 1 » leur sera attribués, tandis que l'extérieur (fond) est formé de pixels « blancs » de valeur « 0 ».

Ces contours sont classés selon deux catégories :

- Ceux qui englobent un objet. Ils sont dits extérieurs, c'est la silhouette de l'objet.
- Ceux qui sont noyés dans un objet, ce sont les contours intérieurs.

Après la détection du premier pixel contour, la nature de la transition détermine sont type :

3 Si cette transition est de nature fond forme (0 – 1), c'est un contour de type externe.

4 Sinon (1 – 0), le contour est de type interne (voir **figure 3.13**)

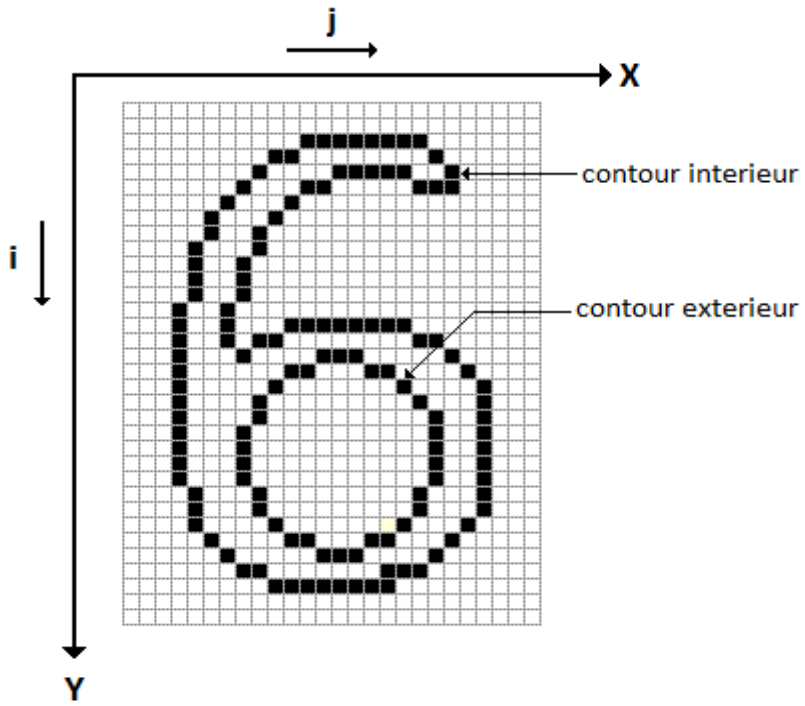


Figure3.13 : Image de référence.

Cette méthode se résumé à balayer l'image jusqu'à trouver une transition. C'est le premier pixel contour.

Etiqueter les huit voisinages de ce pixel. Tester si ces pixels appartiennent au contour. A chaque fois qu'un pixel objet est rencontré, on lui étiquette ses huit voisinages et continuer jusqu'à la rencontre du premier pixel trouvé **[15].**

3.9.1 Implémentation de la méthode

Les principales étapes de l'algorithme sont les suivantes :

1er étape :

Balayage de l'image jusqu'à la rencontre d'un pixel objet

Premier pixel contour à (i_1,j_1). Le pixel qui le précède est noté par (i_D,j_D).

2eme étape :

A partir du pixel (i_D,j_D) en tournant autour du pixel (i_1,j_1) de façon à balayer ses voisins dans le sens horaire, on numérote les sept autres voisins du pixel (i_1,j_1) comme 2 ,……,8 .

3eme étape :

Evaluer les coordonnées Lx(k), Ly(k) du k^{ieme} voisin de (i_1,j_1) en utilisant la table de la figure 23

	Lx(k)	Ly(k)
1	i_d	j_d
2	Lx(1)+k1	Ly(1)-k2
3	Lx(2)-k2	Ly(2)-k1
4	Lx(3)-k2	Ly(3)-k1
5	Lx(4)-k1	Ly(4)+k2
6	Lx(5)-k1	Ly(5)+k2
7	Lx(6)+k2	Ly(6)+k1
8	Lx(7)+k2	Ly(7)+k1

Coordonnées du pixel contour (i_1, j_1)

Coordonnées du 1er pixel voisin (i_d,j_d)

K1 = $j_d - j_1$

K2 = $i_d - i_1$

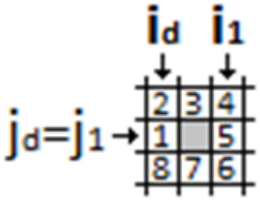


Figure 3.14: Définition du point contour

4eme étape : Si le k^{ieme} voisin est un point noir, il sera le point suivant du contour et on définit (i_1, j_1) en ce point. (i_d, j_d) sera le (k – 1) ^{ieme} point de la table aller à l'étape 2.

5eme étape : Si le k^{ieme} voisin est un pixel blanc, prendre k=k+1 et aller à l'étape 3.

6eme étape : Continuer le processus jusqu'à la rencontre du premier point contour détecté.

7eme étape : Affecter à tous les points contours découverts une valeur différente de 0 et de 1 comme intensité du contour et numéro du contour. Cela permet d'éviter la détection du même contour une notre fois.

8eme étape : Poursuivre le balayage de l'image pour détecter de nouveaux contours jusqu'à la fin de l'image.

L'organigramme de la figure 3.15 illustre le fonctionnement du programme de suivi de contour.

3.9.2 Tableau comparatif

Méthode	Vitesse d'exécution	Sensibilité au bruit	Difficulté d'implémentation
Gradient	Rapide	Très sensible	Moin facile
Laplacien	Moin rapide	Sensible	Difficile
Suivi du contour	Très rapide	Moin sensible	Facile

Tableau 3.9.2: Tableau comparatif

3.10 Organigramme de la détection de contour

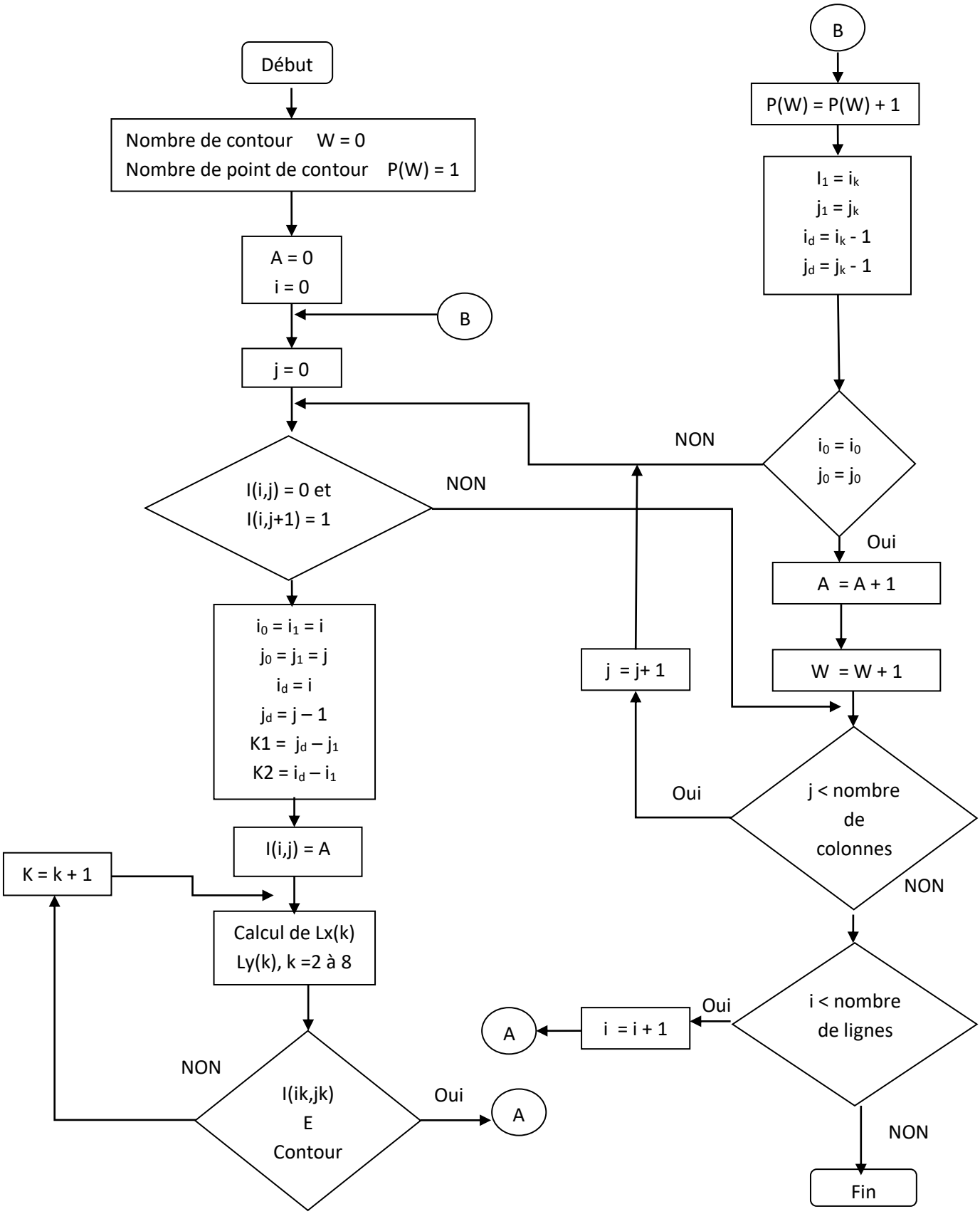


Figure 3.15: Organigramme de suivi de contour de Chottera.

3.11 Représentation polygonale

Une représentation polygonale d'une courbe numérique peut réduire considérablement la quantité de données à traiter tout en préservant des informations importantes sur la courbe. Il peut être appliqué dans l'analyse de forme, la classification de motifs, la compréhension d'image, la reconstruction 3D et les applications de conception assistée par ordinateur (CAO).

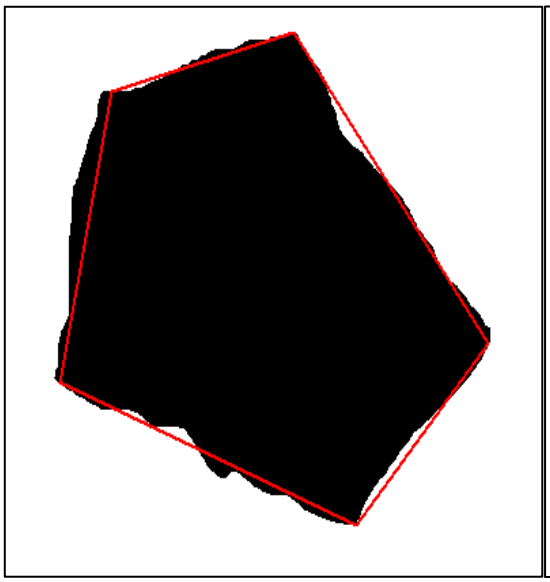


Figure 3.15.1:

Forme approximée par 5 segments droites

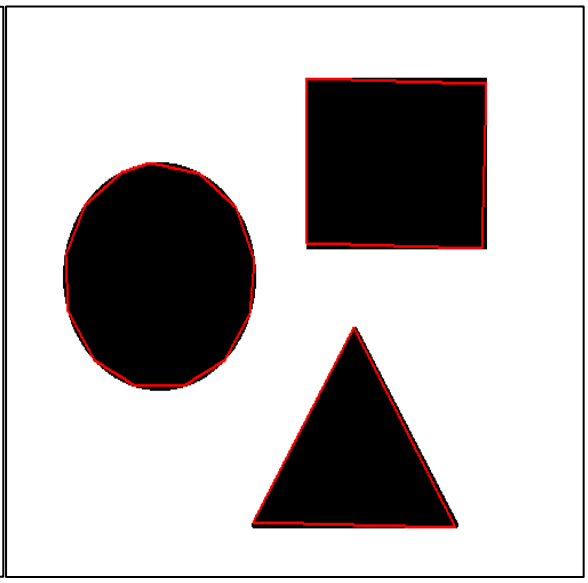


Figure 3.15.2:

Cercle, carré et triangle approximés de par 13, 4 et 3 droites respectivement.

3.12 Mesures d'erreurs

Une courbe d'approximation doit satisfaire certains critères d'erreurs, qui sont spécifiés de manière appropriée pour chaque application. En pratique, la plupart des mesures d'erreurs pratiques utilisées sont basées sur la distance entre les sommets de la courbe d'entrée et les segments linéaires d'approximation ou segment de droites.

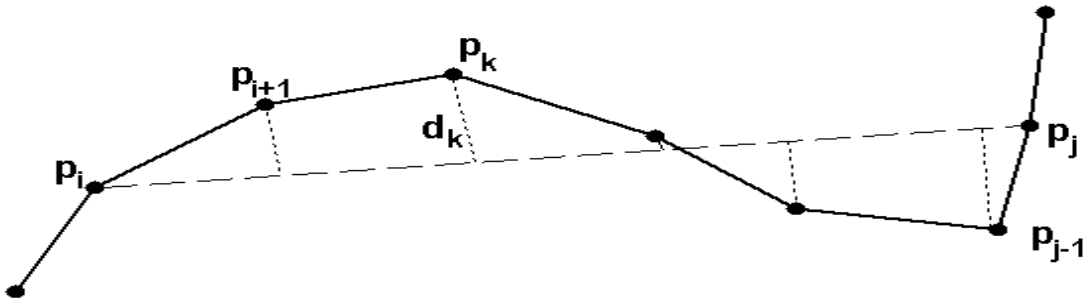


Figure 3.15.3: Distance d_k du point p_k au segment de droite (p_i, p_j).

La distance $d_k(i, j)$ du point $p_k=(x_k, y_k)$ au segment de droite (p_i, p_j) est défini par:

$$d(k;i,j) = \frac{|y_k - a_{i,j}x_k - b_{i,j}|}{\sqrt{1+a_{i,j}^2}} \quad (12)$$

Où les coefficients a_i,j and b_i,j sont les paramètres du segment de droite (p_i, p_j):

$$a_{i,j} = (y_j - y_i)/(x_j - x_i)$$
$$b_{i,j} = y_i - a_{i,j}x_i. \quad (13)$$

La somme cumulative des erreurs L_p pour le segment $\{p_i, p_j\}$ est définie par la somme des distances $d(k; i, j)$ pour tous les sommets dans le segment $\{p_i, p_j\}$ comme suit:

$$e_p(i,j) = \sum_{k=i+1}^{k=j-1} d^p(k;i,j) \quad (14)$$

3.13 Conclusion

Nous avons vu dans ce chapitre quelques notions sur le traitement d'images. Les méthodes de traitement d'images désignent les outils utilisés pour transformer une image numérique en une nouvelle image présentant de meilleurs caractéristiques pour rendre fiable son interprétation.

Chapitre 4 **Réalisation et test**

4.1 Introduction

Dans ce dernier chapitre, nous présentons notre projet qui a été réalisé en utilisant la carte Raspberry PI, la carte Arduino et les servo moteurs. C'est un système de tri automatique d'objets placés sur un convoyeur. Nous évaluons dans ce chapitre les résultats obtenus sur des objets ayant trois formes géométriques ; triangle carré et cercles.

4.2 Principe de l'application :

4.2.1 Environnement du Travail :

Nous avons utilisé :

- **Un Raspberry Pi 4 :**

- Processeur 1.5GHz quad-core 64-bit ARM Cortex-A72

- 4GB de mémoire SDRAM LPDDR4

- Ethernet Gigabit

- Wifi Dual-band 802.11ac

- Bluetooth 5.0

- Deux ports USB 3.0 et deux ports USB 2.0

- Support double écran

- GPU VideoCore VI

- Pi camera 5mp

4.2.2 Présentation de l'application :

Dans cette partie nous allons détailler les étapes principales et présenter la mise en œuvre de notre application en se basant sur le schéma synoptique de la **figure 4.1**

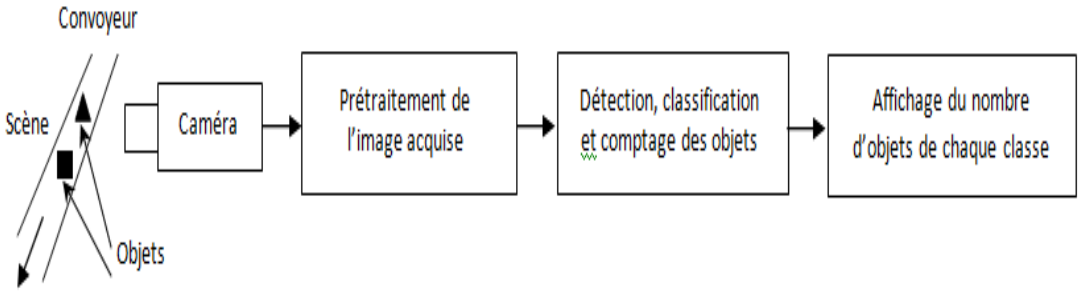


Figure 4.1: Schéma synoptique de l'application proposée.

4.3 Algorithme de détection de contour dans OpenCv :

Les algorithmes de détection de contour peuvent généralement être

Classés en trois types comme suit

• Suivi des pixels,

• Suivi de sommet

• Suivi basé sur les données d'exécution

OpenCv a une méthode intégrée qui implémente un algorithme de détection de contour : cv2.findContours (). Cette méthode prend en entrée trois paramètres : le premier est image source, le deuxième est le mode de récupération de contour et le troisième est la méthode d'approximation de contour. Les sorties sont : image, contours et hiérarchie. Contours est une liste de tous les contours de l'image. Chaque contour individuel est un tableau NumPy de (x, y) coordonnées des points limites de l'objet.

Après avoir détecté tous les contours d'une image, une approximation de chaque contour sera effectué. Comme son nom l'indique, l'approximation des contours est un algorithme de réduction du nombre de points dans une courbe avec un ensemble réduit de points - d'où le terme approximation. Cet algorithme est communément appelé l'algorithme « Ramer-Douglas-Peucker », ou simplement l'algorithme de fractionnement et de fusion. L'approximation du contour est fondée sur l'hypothèse qu'une courbe peut être approximée par une série de segments de ligne courts. Cela conduit à une courbe approximative résultante qui se compose d'un sous-ensemble de points définis par courbe originale. Nous avons utilisé la fonction d'approximation des contours de la bibliothèque OpenCV via le « cv2.approxPolyDP () ». **[16]**

Afin d'effectuer une approximation du contour, il faut tout d'abord calculer le périmètre du contour puis faire la construction du contour réel comme comme suit :

```
peri = cv2.arcLength(c, True)

approx=cv2.approxPolyDP(contours[i], peri*0.02, True)
```

La valeur du deuxième paramètre à cv2.approxPolyDP est normalement comprise entre 1 et 5% du périmètre du contour d'origine. Compte tenu de notre contour approché, on peut passer à la réalisation de la détection de forme.

Il est important de comprendre qu'un contour se compose d'une liste des sommets. Nous pouvons vérifier le nombre d'entrées dans cette liste pour déterminer la forme d'un objet. Par exemple, si le contour approximatif à trois sommets, alors ce doit être un triangle comme indiqué ci-dessous

```
if len(approx)==3:
        shape = "triangle"
    elif len(approx)==4:
        (x, y, w, h) = cv2.boundingRect(approx)
        ratio = w / float(h)
        if ratio >= 0.95 and ratio <= 1.05:
            shape = "carre"
        else:
            shape = "rectangle"
    elif len(approx)==5:
        shape = "pentagone"
    elif len(approx)==6:
        shape = "hexagone"
    else:
        shape= "circle"
    cv2.putText(image, shape, (cX, cY), cv2.FONT_HERSHEY_SIMPLEX,0.5, (255, 255, 255), 2)
```

Figure 4.2 : Algorithme de détection de contour avec OpenCv

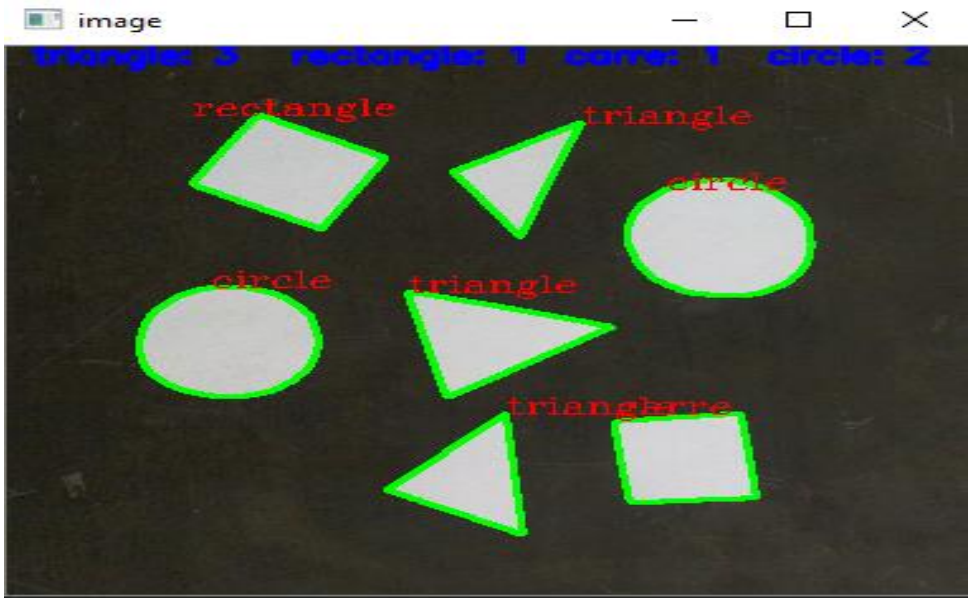


Figure 4.3 : Classification d'objet par classe

4.4 Détection et comptage d'objets :

Dès que l'objet rentre dans le champs de vision de la camera, grâce à la fonction « (x,y,w,h)=cv2.boundingRect(approx) » qui va faire dessiner un rectangle virtuel sur tous les contours détectés ,un rectangle avec les cordonnées (x,y,w,h).

Trouver le centre d'un contour :

Avant de faire le comptage il faut d'abord détecter l'objet et le suivre et pour le faire il faut trouver le centre du contour ou bien le centre de l'objet et pour cela nous avons utilisé les cordonnées x,y,w,h :

Le centre est un point de cordonnèes (cx,cy)

Nous allons nommé : x=limite_gauche ,y=limite_basse,h=limite_droite,w=limite_haute (voire **figure 4.4**)

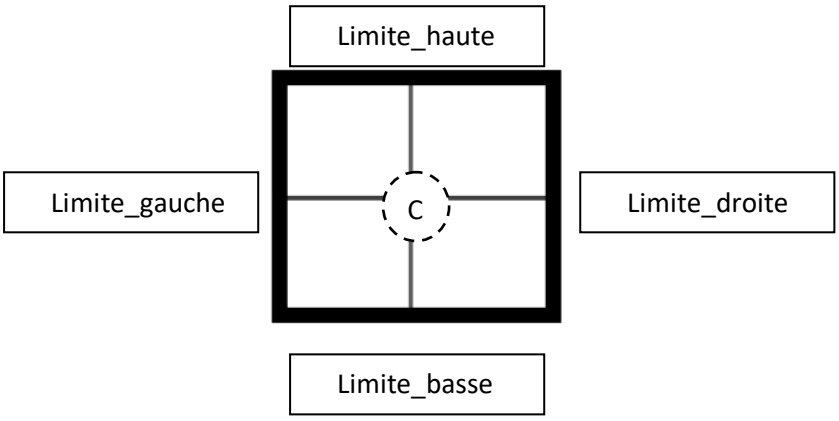


Figure4.4 : les différentes limites qui se trouve dans le rectangle virtuel

D'après la figure on constate que :

Cx=limite_gauche+1/2 limite_haute

Cy=limite_basse+1/2 limite_droite

Avec l'rentrée de l'objet dans champ de vision notre système enregistre directement ces cordonnées ; (x,y,w,h), et avec le mouvement de l'objet ces cordonnées vont changer et pour bien le suivre l'objet nous avons mis des conditions pour cx et cy .

-si cx est supérieure à la limite_gauche et cx est inférieure à la somme de la limite_gauche avec la limite_haute, et si cy est supérieure à la limite_basse et cy est inférieure à la somme de la limite_basse avec la limite_droite, donc l'objet est bien suivi, sinon il s'agit de la détection d'un autre objet.

Maintenant, après que l'objet est bien suivi, nous avons mis une ligne virtuelle à la fin du trajet et dès que l'objet traverse cette ligne on incrémente le nombre d'objet passés et par rapport à sa forme (cercle, triangle, rectangle) on fait l'incrémentation de la forme identifiée (voir figure 4.6)

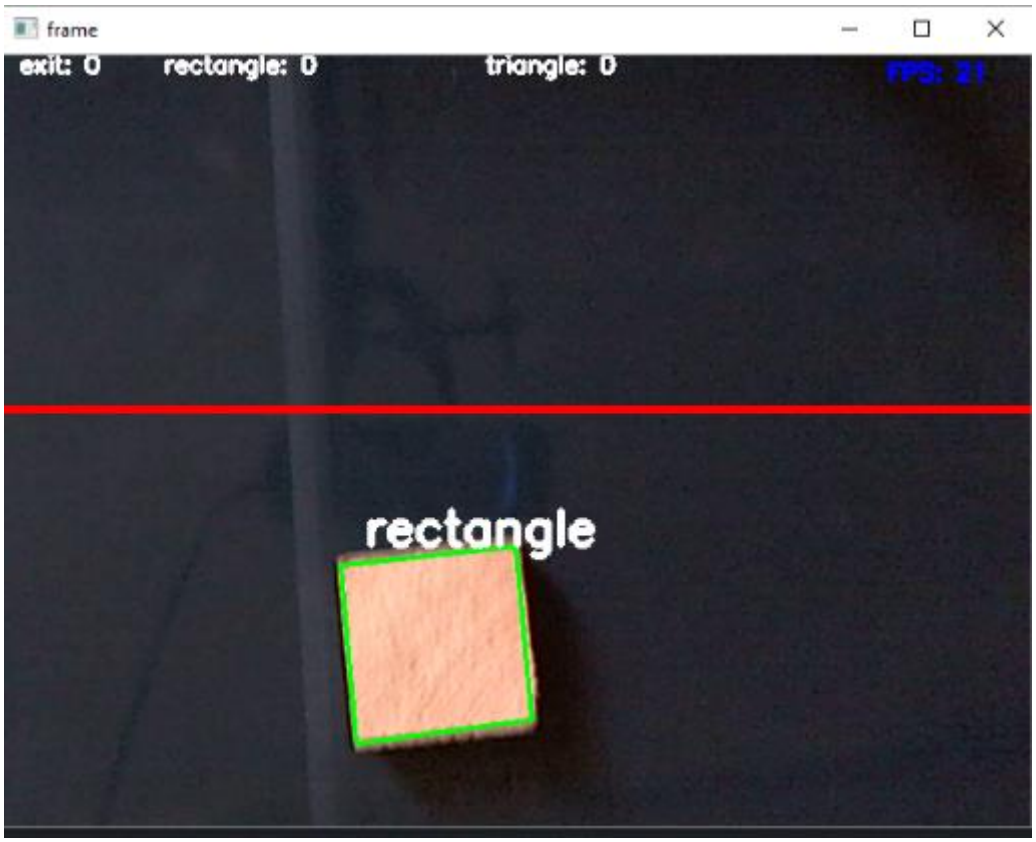


Figure 4.5 : Détection d'objet avant la ligne rouge

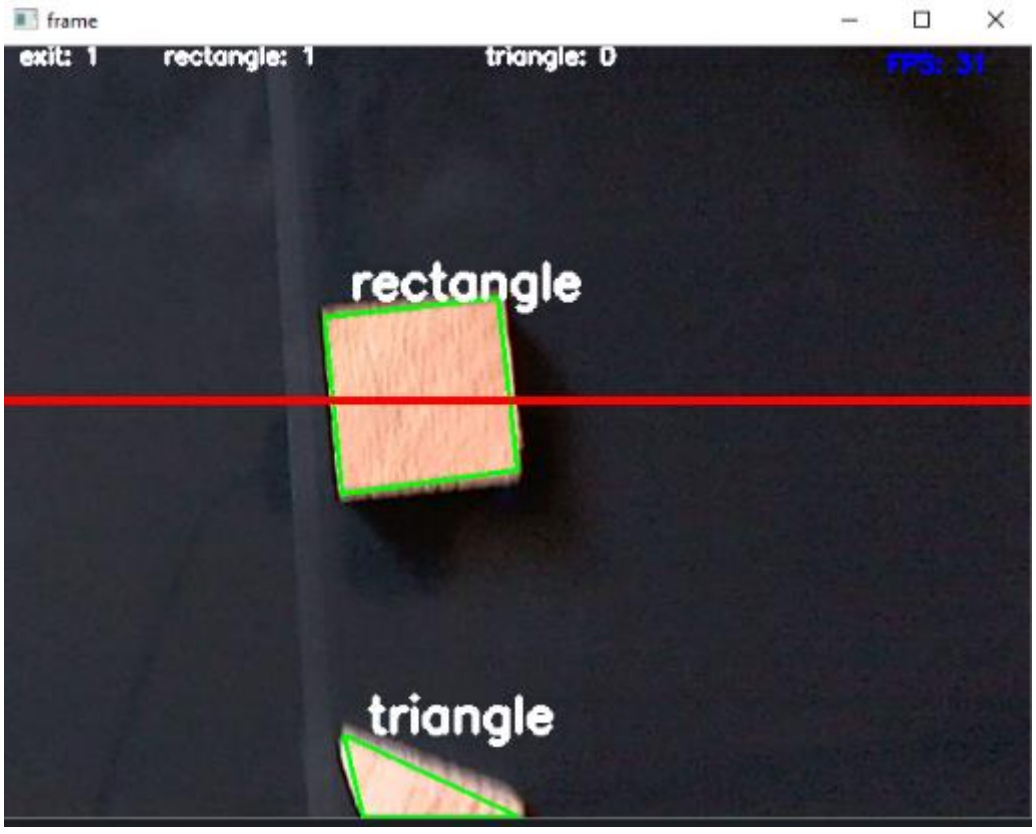


Figure 4.6 : détection d'objet sur la ligne rouge

4.5 Activation du servo moteur :

Après la détection et le comptage, dès que les objets passent la ligne virtuelle, chaque objet doit être acheminé ou mis dans sa propre place par rapport à sa forme.

Pour notre projet nous avons utilisé 3 formes ; cercle, triangle et rectangle. Si l'objet et un triangle le programme active le servo moteur gauche pour faire tourner l'objet à gauche et va le mettre dans la place des triangles, si l'objet est un rectangle le programme active le servo moteur droit pour faire tourner l'objet à droite pour le mettre dans la place des rectangles et enfin si l'objet est un cercle le programme va laisser l'objet aller jusqu'au bout dans la place des cercles. L'arduino contrôle les 2 servos moteurs et la communication entre le raspbery pi et l'arduino est par une communication I2C.

Figure 4.6.1 : notre convoyeur avec l'emplacement des servo moteurs

4.6 Schéma de circuit :

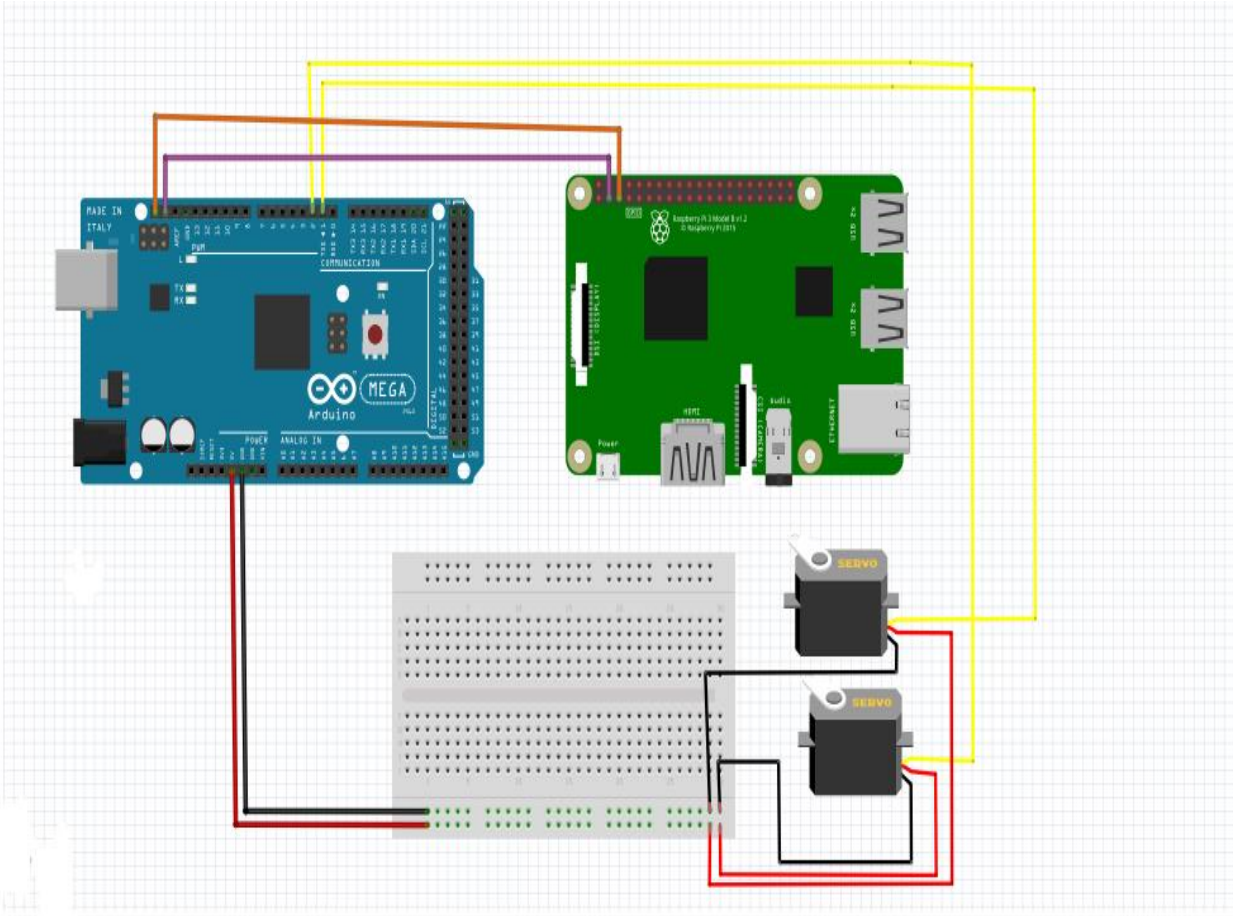


Figure 4.6.2 : schéma de circuit

Organigramme de notre programme :

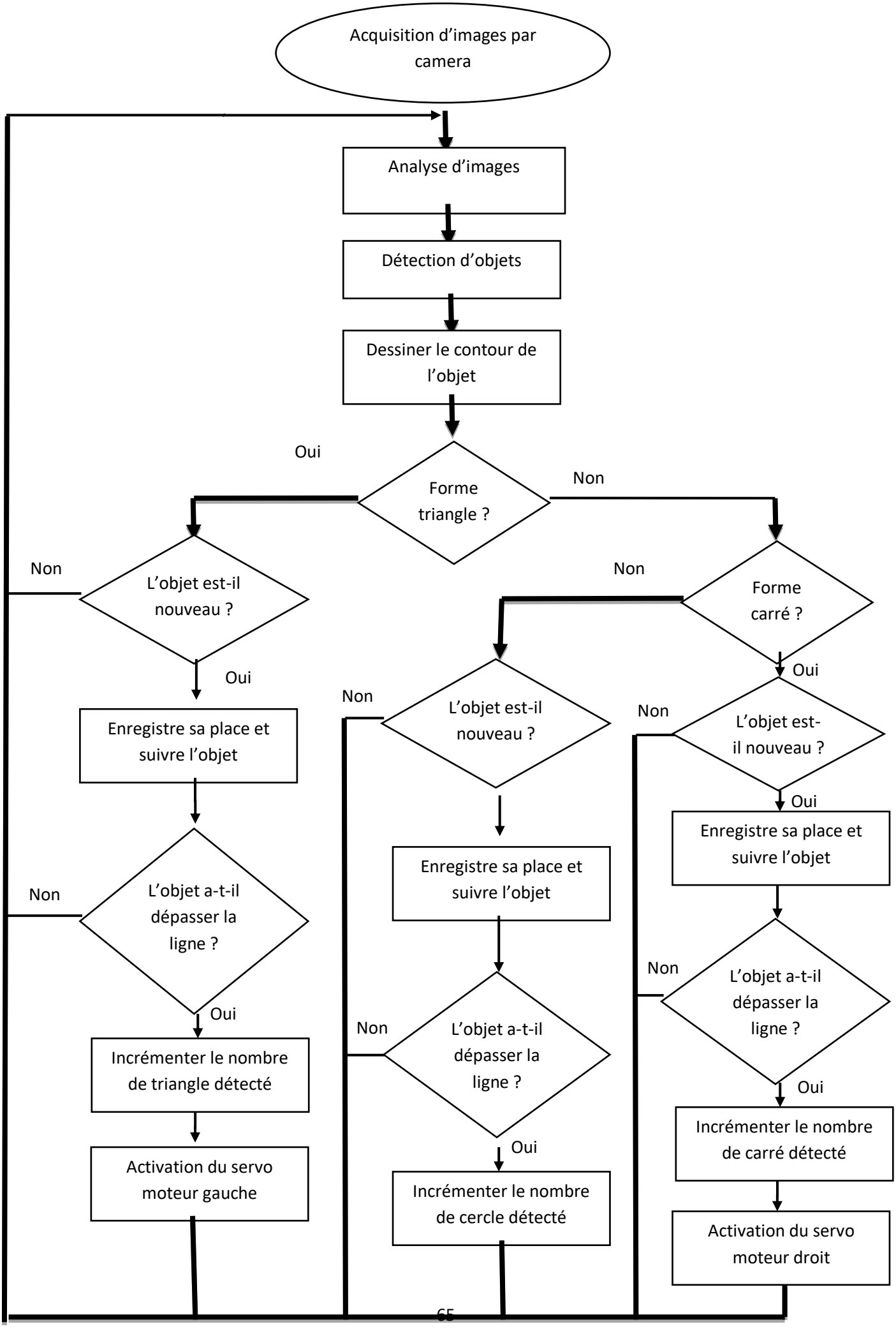

Acquisition d'images par camera
Analyse d'images
Détection d'objets
Dessiner le contour de l'objet
Forme triangle ?
Oui
Non
L'objet est-il nouveau ?
Non
Oui
Enregistre sa place et suivre l'objet
L'objet a-t-il dépasser la ligne ?
Non
Oui
Incrémenter le nombre de triangle détecté
Activation du servo moteur gauche
Forme carré ?
Non
Oui
L'objet est-il nouveau ?
Non
Oui
Enregistre sa place et suivre l'objet
L'objet a-t-il dépasser la ligne ?
Non
Oui
Incrémenter le nombre de cercle détecté
L'objet est-il nouveau ?
Non
Oui
Enregistre sa place et suivre l'objet
L'objet a-t-il dépasser la ligne ?
Non
Oui
Incrémenter le nombre de carré détecté
Activation du servo moteur droit

Figure 4.7 : Organigramme de la détection et comptage d'objet

4.6 Résultat du comptage en temps réel :

Les résultats du comptage d'objets transportés par notre convoyeur sont présentés au tableau 4.1. :

Test	Triangle	Carrée	Cercle	Triangle Détecté	Carrée Détecté	Cercle Détecté	Pourcentage %
1	5	6	5	5	6	5	100 %
2	2	10	20	2	10	20	100 %
3	0	6	0	0	6	0	100 %
4	5	6	0	5	6	0	100 %
5	30	25	14	30	25	14	100 %
6	43	33	0	43	33	0	100 %
7	40	45	43	40	45	43	100 %

Tableau 4.6 : Résultat du comptage.

D'après les résultats du tableau 4.1 nous pouvons conclure que le système que nous avons réalisé est robuste, fiable et précis. Un taux de 100% a été obtenu pour un nombre réduit d'objets, reste à l'expérimenter à grand échelle.

4.7 Conclusion

Dans ce chapitre, nous avons vu la mise en œuvre en temps réel d'un système de localisation, comptage et tri automatique d'objets en utilisant La Raspberry PI, Arduino et servo moteur. Tout le système est géré par une programmation en langage Python avec la bibliothèque OpenCV. Cette programmation utilise les techniques du traitement d'images. Nous avons utilisé un ordinateur portable et une caméra reliée directement à la Raspberry PI. Les résultats que nous avons obtenus étaient assez bons, l'efficacité et la précision du système étaient excellentes avec un taux de classification ou de tri de 100%.

Conclusion générale

Le travail présenté dans ce mémoire a été effectué au sein du Département d'Electronique

Université de Blida1.

Nous avons présenté en premier lieu les méthodes de détection d'objets en mouvement, puis l'analyse d'images numériques, sa définition, domaine d'utilisation puis nous avons détaillé quelques outils du traitement d'images ainsi que leurs implémentations. En deuxième lieu, nous avons réalisé une étude sur les méthodes utilisées dans ce travail, on se focalisant surtout sur la détection d'objets. En troisième lieu, nous avons conçu une procédure de détection d'objets ainsi que son implémentation, suivi de là de la conception d'un convoyeur qui était une tache assez importante. L'utilisation de servo-moteur contrôlés par la carte Raspberry PI a été introduite pour assurer un tri automatique d'objets en temps réel. En quatrième lieu nous avons présenté la partie expérimentale.

Au cours de ce projet et à travers le travail fait, nous avons pu acquérir beaucoup de connaissances. Nous avons utilisé les outils appris au cours de notre formation, et enrichi nos connaissances dans le domaine du traitement d'images en apprenant de nouveau outils. Nous avons aussi appris à utiliser le langage de programmation Python et l'utilisation de la carte Raspberry PI.

Nous estimons que notre module pourrait être exploité dans l'industrie. Nous proposons comme perspective la réalisation d'un système embarqué basé sur la vision artificielle capable de compter le nombre d'objets réels (paquets de produit alimentaire, pièces mécaniques, sacs de ciments…etc.) placés sur un convoyeur et dans un milieu industriel.

Bibliographie

[1] H. YEDJOUR. Détection de contours et suivi d'objet dans une séquence d'images par les réseaux de neurones impulsionnels Université Mohamed Boudiaf (USTO) ORAN Thèse Magister 2010.

[2] A. BUGEAU .Détection et suivi d'objets en mouvement dans des scènes complexes, application à la surveillance des conducteurs, Thèse de Docteur Université Rennes 1, 2007.

[3] J. PIATER, P. GABRIEL, J. VERLY and A. GENON. The state of the art in multiple object tracking under occlusion in video sequences. Advanced Concepts for Intelligent Vision Systems, 2003.

[4] Latreche kaddour, boumagouda loubna, Conception d'un convoyeur à bande, thème de master, 2011.

[5] Yasser Ben Aissia et Ghassene Chaieb. Gateway d'un système de monitoring d'un malade à un serveur d'urgence. Master's thesis, Ecole Nationale d'Ingénieurs de Tunis, 2015.

[6] Gâad Mohamed et Rahmi Bachir, Conception et Réalisation d'un Emetteur OFDM à Base d'une Carte Raspberry pi, Thème de mester,2018.

[7] Si-Ahmed Yasmine et Ouchefoun Nassima, Conception d'un Robot Mobile avec Vison Intelligente Embarquée, thème de master,2016.

[8] http://fr.wikipedia.org/wiki/OpenCV.

[9] Richard Szeliski Computer Vision : Algorithms and Applications 2003.

[10] Alexander Mordvintsev & Abid K OpenCVPython Tutorials Documentation Release 1 2017

[11] L. Shapiro, G. Stockman. Computer Vision (1st Edition), February 2001.

[12] Gonzalez, R. and Woods, R. (1992) Digital Image Processing. Addison Wesley, 5, 414-428.

[13] Nobuyuki Otsu, « A threshold selection method from gray-level histograms », IEEE Trans. Sys., Man., Cyber., vol. 9,? 1979, p. 62–66.

[14] Xuelong Hu, Min Zhang, Weiping Yang, Nan Jiang and Xiang Yina, "Image De-noising and Granularity Detection Based on Morphology", Physics Procedia, Volume 24, Part C, 2012, Pages 1822-1829.

[15] A. Chottera; M. Shridhar, "Feature extraction of manufactured parts in the presence of spurious surface reflections", Canadian Electrical Engineering Journal, Volume: 7, Issue: 4, 1982,Page(s):29 - 33.

[16] Jan Erik Solem Programming Computer Vision with Python 2012.