



Université Hadj Lakhdar Batna  
Faculté de Technologie  
Département Génie Industriel



## Mémoire de Magister

présenté au

Laboratoire d'Automatique et Productique

En vue de l'obtention de diplôme de

Magister

Option : Génie Industriel et Productique

Par

**BELLAL SALAH EDDINE**

*Ingénieur d'état en informatique*

Thème

**RECONNAISSANCE D'OBJETS DANS UNE IMAGE**  
**APPLICATION AU TRI / CONTROLE DE QUALITE**

DIRECTRICE DE THESE : Dr. L.H. Mouss

### MEMBRES du JURY

M. Djamel Mouss	MC-A	Université de Batna	Président
L. Hayet Mouss	MC-A	Université de Batna	Rapporteur
Mhania GUERTI	Pr.	Ecole Nationale Polytechnique	Examineur
Samir Abdelhamid	MC-A	Université de Batna	Examineur
R. Bensaadi	MA-B	Université de Batna	Co-encadreur
N. Menina	Ingenieur MBA	Gérant Minitec/Schneider	Invité

Année Universitaire 2011-2012



## REMERCIEMENTS

*Tous mes remerciements vont à Mme Mouss Hayet dont l'auguste autorité scientifique et les précieux conseils ont permis ce modeste travail.*

*Qu'elle trouve ici le sincère témoignage de reconnaissance et de gratitude qu'elle mérite et qui n'égale jamais la patience, et le temps consacré à la direction de ce travail.*

*Je remercie encore mon co-encadreur Mr R. Bensaadi pour son soutien et son assistance.*

*Que les membres de jury trouvent ici mes sincères remerciements pour le temps imparti à la lecture de ce mémoire et à son examinateur.*

*Mes remerciements vont également au Mr. Said Saidi de l'Université de Constantine dont les judicieuses suggestions en méthodologie furent très constructives.*

*Ma gratitude la plus entière à tous les enseignants, mes collègues du département et à tout le personnel administratif.*

*Pour les efforts aimablement consentis à l'élaboration de cette modeste recherche, que Mr. N. Menina et Mr. A. younsi partenaire de Schneider, soient amplement remerciés et trouvent ici la sincère expression de reconnaissance.*

*Enfin, un grand merci à ceux que je n'ai pas cités ici mais qui ont croisé mon sillage, à tous mes amis.*

# TABLE DES MATIERES

Table des matières.....	i
Tables des figures.....	iv
Résumé.....	vii
Abstract.....	viii
 Introduction Générale.....	 1
Objectif et Problématique .....	3
Présentation du mémoire.....	6
Cahier des charges.....	8
 <b>Chapitre -1 Concepts de la Vision Humaine et de la Vision Artificielle .....</b>	 <b>10</b>
1.1 Introduction .....	10
1.2 Vision Humaine.....	11
1.3 Le système visuel humain .....	11
1.3.1 Sensibilité à la lumière.....	12
1.3.2 Sensibilité à la couleur.....	12
1.4 Vision Artificielle.....	13
1.4.1 Avantages des applications de Vision Artificielle.....	15
1.5 Processus de reconnaissance d’objets.....	15
1.5.1 Introduction.....	15
1.5.2 Etapes du processus de reconnaissance d’objets Introduction.....	17
1.5.2.1 Introduction.....	17
1.5.3 Composants d’un système de vision .....	18
1.5.4 Description du processus de reconnaissance.....	19
1.5.5 Acquisition.....	21
1.5.5.1 Introduction.....	21
1.5.5.2 Caméras, capteurs d’images.....	21
1.5.5.3 Eclairage.....	23
1.6 Conclusion.....	24
 <b>Chapitre -2 Traitement Numérique d’Images.....</b>	 <b>25</b>
2.1 Introduction.....	25
2.2 Prétraitement ou Traitement bas niveau.....	26

2.2.1	Définition d'image.....	26
2.2.1.1	Définition Pixels.....	27
2.2.1.2	Types et caractéristiques d'images.....	28
2.2.2	Voisinage dans les images.....	34
2.2.2.1	Voisinage d'un pixel.....	34
2.2.2.2	Distances entre pixels.....	34
2.3	Principe General sur le Seuillage et la Binarisation.....	35
2.3.1	Définition du seuillage.....	35
2.3.1.1	Le seuillage fondé sur l'analyse de l'histogramme.....	35
2.3.1.2	Seuillage adaptatif.....	37
2.4	Binarisation –seuillage.....	37
2.4.1	Le Filtrage.....	39
2.4.1.1	Filtrage linéaire.....	40
2.4.1.2	Filtrage non linéaire.....	52
2.5	Morphologie mathématique.....	56
2.5.1	Opérateur morphologique de dilatation.....	57
2.5.1.1	Définition.....	57
2.5.2	Opérateur morphologique d'érosion.....	58
2.5.2.1	Définition.....	58
2.5.3	Ouverture (érosion + dilatation).....	58
2.5.3.1	Définition.....	58
2.5.4	Fermeture (dilatation + érosion).....	59
2.6	Conclusion.....	60
		70
<b>Chapitre -3</b>	<b>Segmentation des images.....</b>	<b>61</b>
3.1	Introduction.....	61
3-2	Les principes de la segmentation.....	62
3.2.1	La segmentation par régions.....	62
3.2.2	La segmentation par seuillage.....	63
3.2.2.1	Détection automatique du seuil.....	64
3.2.2.2	Seuillage à hystérésis .....	65
3.2.3	La segmentation par contours.....	65
3.2.3.1	Détection de contours .....	87
3.2.4	La transformée de Hough.....	74
3.2.5	Segmentation par Etiquetage de composantes connexes.....	77
3.2.5.1	Analyse en composantes connexes.....	78
3.2.5.2	Organigramme de reconnaissance d'objets par composantes Connexes.....	79
3.2.6	La Segmentation par LPE .....	80
3.2.6.1	Définition.....	80
3.3	Conclusion.....	84

<b>Chapitre -4 Description et Interprétation.....</b>	<b>85</b>
4.1 Introduction.....	85
4-2 Invariants de formes Contours.....	86
4.2.1 Code de Freeman.....	86
4.2.1.1 Principe du codage.....	86
4.2.2 Les descripteurs de Fourier.....	87
4.3 Invariants de formes Regions.....	90
4.3.1 Moments de Zernike.....	90
4.3.1.1 Effet de la rotation.....	91
4.3.1.2 Effet d'échelle.....	91
4.3.2 Moments invariants de Hu.....	92
4.4 La corrélation.....	94
4.4.1 Définition.....	94
4.5 Caractéristiques géométriques d'un objet.....	96
4.6 Conclusion.....	98
 <b>Chapitre -5 Mise en œuvre du processus.....</b>	 <b>99</b>
5.1 Introduction.....	99
5.2 Environnement du travail.....	99
5.2.1 Environnement matériel.....	99
5.2.2 Environnement logiciel.....	100
5.2.2.1 Langage de développement : MATLAB 7.9.0 (R2009b).....	100
5.2.2.2 Logiciel de conception.....	100
5.3 Interface de l'application MATLAB.....	103
5.3.1 Reconnaissance d'objets.....	103
5.3.2 Reconnaissance d'objets par Automate.....	107
5.3.2.1 Communication entre le PC et l'automate.....	108
5.3.2.2 Organigramme du programme de communication entre PC et l'automate M340.....	109
5.4 Interface homme-machine IHM.....	111
5.5 Conclusion.....	112
 Conclusion Générale et Perspectives.....	 113
BIBLIOGRAPHIE.....	116
ANNEXE A.....	119
ANNEXE B.....	121

## Table des Figures

Figure 1.1 : Domaine de la vision.....	10
Figure 1.2 : Domaine de la vision.....	12
Figure 1.3 : Comparaison entre le capteur naturel, l'œil et un capteur numérique.....	13
Figure 1.4 : Contrôle de qualité par vision artificielle .....	14
Figure 1.5 : Le scannage des plaques est réalisé par un système de vision artificielle .....	14
Figure 1.6 : Des yeux numériques trient les agrumes.....	14
Figure 1.7 : Identification et manipulation des pièces à la géométrie complexe par et système de vision artificielle.....	14
Figure 1.8 : Détection de salissures/ de corps étrangers à l'intérieur d'une coupelle.....	14
Figure 1.9 : Principales étapes de la reconnaissance d'objets.....	18
Figure 1.10 :L'implantation d'une simulation qui représente toutes les étapes de reconnaissance d'objets pour l'opération de tri et de contrôle de qualité.....	19
Figure 1.11 : Convoyeur.....	20
Figure 1.12 : Recadrage d'image.....	20
Figure 1.13 : Webcam.....	22
Figure 1.14 : Dispositif d'éclairage.....	23
Figure 2.2 : Transformation image RGB en image en niveau de gris.....	26
Figure 2.3 : Exemple de réseau de pixels.....	27
Figure 2.4 : Matrice S sous forme d'un tableau.....	28
Figure 2.5 : Espace RGB.....	30
Figure 2.6 : Représentation d'image en RGB et la valeur d'intensité en long d'une ligne...	30
Figure 2.7 : Espace HSV.....	31
Figure 2.8 : Transformation Image RGB en HSV.....	31
Figure 2.9 : Représentation d'une image en niveaux de gris.....	32
Figure 2.10 : Différence d'Histogramme Image RGB et image contrastée.....	33
Figure 2.11 : Connexité et voisinage.....	34
Figure 2.12 : Cas du seuillage d'un histogramme bimodal.....	36
Figure 3.13 : Cas du seuillage d'un histogramme multimodal.....	36
Figure 2.14 : Seuil de binarisation.....	38
Figure 2.15 : Transformation d'image en niveau de gris en image binaire.....	39
Figure 2.16 : Filtrage linéaire.....	40
Figure 2.17 : Filtrage fréquentiel.....	43
Figure 2.18 : La Transformation de Fourier de l'image originale et de l'image filtrée passe bas.....	44
Figure 2.19 : Différent types de filtrage.....	45
Figure 2.20 : Transformation d'un point $i,j$ .....	46
Figure 2.21 : Exemple de convolution Locale de l'image.....	46

## Table des Figures

Figure 2.22 : Représentation du filtrage spatial.....	46
Figure 2.23 : Filtre spatial pour un masque 3x3.....	48
Figure 2.24 : Filtre moyennneur.....	49
Figure 2.25 : Filtre de bruit Gaussien.....	51
Figure 2.26 : image bruitée par un moyennneur 3 x 3.....	53
Figure 2.27 : La réduction de bruit avec le filtre médian.....	54
Figure 2.28 : Image dilatée.....	57
Figure 2.29 : Erosion d'image .....	58
Figure 2.30 : Ouverture d'image.....	59
Figure 2.31 : Fermeture d'image.....	59
Figure 3.1 : La méthode OTSU seuil=136 .....	64
Figure 3.2 : La méthode OTSU seuil=102.....	64
Figure 3.3 : Seuillage par hystérésis.....	65
Figure 3.4 : Différentes formes de contours.....	66
Figure 3.5 : Détection de contours à l'aide du filtre de Prewitt (seuil : 0.1).....	68
Figure 3.6 : Détection de contours à l'aide du filtre de Prewitt (seuil : 0.3).....	69
Figure 3.7 : Détection de contours à l'aide du filtre de Prewitt (seuil : 0.05).....	69
Figure 3.8 : Détection de contours à l'aide du filtre de Sobel (seuil : 0.1).....	70
Figure 3.9 : Détection de contours à l'aide du filtre de Sobel (seuil : 0.3).....	70
Figure 3.10 : Détection de contours à l'aide du filtre de Sobel (seuil : 0.05).....	70
Figure 3.11 : Détection de contours à l'aide du filtre de Robert (seuil : 0.1).....	71
Figure 3.12 : Détection de contours à l'aide du filtre de Robert (seuil : 0.3).....	71
Figure 3.13 : Détection de contours à l'aide du filtre de Robert (seuil : 0.05).....	71
Figure 3.14 : Laplacien de l'image.....	73
Figure 3.15 : Filtre de Canny .....	74
Figure 3.16 : Composante connexe.....	78
Figure 3.17 : Etiquetage en Composantes Connexes.....	79
Figure 3.18 : Organigramme de reconnaissance d'objets par composantes connexes.....	79
Figure 3.19 : Ligne de Partage des Eaux.....	80
Figure 3.20 : Module du gradient.....	80
Figure 3.21 : Transformation de LPE à Amplitude de Gradient.....	81
Figure 3.22 : Ouverture et fermeture par reconstruction.....	81
Figure 3.23 : Maxima régionaux ouverture et fermeture par reconstruction	
Maxima régionaux superposé sur image originale.....	81
Figure 3.24 : Ouverture et fermeture de LPE par reconstruction.....	82
Figure 3.25 : Lignes de partage des eaux.....	82
Figure 3.26 : Superposition des marqueurs et lignes de partages sur l'image.....	83
Figure 3.27 : Matrice colorée des régions appliquée sur l'image d'origine.....	83



## Table des Figures

Figure 4.1 : Représentation à 8 directions pour le code chaîne.....	87
Figure 4.2 : Le carré à côté sa séquence uniforme de K points complexes obtenue avec les Fourier descripteurs.....	88
Figure 4.3 : Exemple de reconstruction de contour à partir des Fourier descripteurs P le nombre de coefficients utilisés dans le processus de reconstruction.....	89
Figure 4.4 : Exemple d'images appliquées.....	91
Figure 4.5 : Extraction de l'objet dans l'image.....	91
Figure 4.6 : Ensembles d'objets.....	95
Figure 5.1: Menu principal.....	103
Figure 5.2: Interface Acquisition vidéo.....	104
Figure 5.3: Interface Acquisition image.....	104
Figure 5.4: Interface traitement image.....	105
Figure 5.5: Interface localisation objets.....	105
Figure 5.6: Interface Identification d'objets.....	106
Figure 5.7: Port parallèle (LPT1) par Matlab.....	107
Figure 5.8: Numérotation de pines (LPT1).....	107
Figure 5.9: Organigramme du programme de communication entre PC et l'Automate.....	110
Figure 5.10: Interface Vijeo Citect.....	111

## RESUME

Dans ce mémoire nous présentons les méthodes et techniques de la vision par ordinateur existantes et celles qui font l'objet de recherches actuelles. La reconnaissance d'objets est l'un des domaines d'étude les plus actifs de la vision par ordinateur, une des ces tâches principales est de pouvoir identifier et localiser ces objets à partir d'informations reçues de l'environnement.

La reconnaissance d'objets a pour but d'extraire automatiquement et efficacement le contenu intéressant, pertinent et utile de la scène c'est-à-dire de déterminer l'identité des objets formant la scène et leur disposition spatiale dans le contexte d'une tâche ou d'un processus à réaliser. Il s'agit d'un point très important en vision par ordinateur puisqu'il est indispensable pour toutes sortes d'applications.

Dans ce travail nous nous sommes intéressés à bâtir un système de reconnaissance d'objets dans une image pour une application de tri.

L'axe de travail consiste à reconnaître et à trier les objets (formes géométriques) dans une image en vision artificielle. Les images proviennent d'une caméra fixe ayant un champ de vision régulier et permanent avec un angle conséquent sur un convoyeur, porteur des objets à observer.

La reconnaissance d'objets implique de connaître les caractéristiques de notre caméra. Ainsi pour pouvoir calibrer cette caméra, nous avons proposé une simple camera Webcam orientée de manière à obtenir une photo par le centre du convoyeur. Il est dès lors primordial de démarrer la conception d'une chaîne de traitement d'images par la question de l'éclairage. Un bon éclairage permet parfois de simplifier considérablement l'étape de traitement et donne un très bon résultat au système de reconnaissance d'objets.

L'objectif est la recherche d'objets dans des images, il s'agira d'images unitaires et non pas de séquences d'images. Dans ce cas la mise en place de méthodes automatiques est nécessaire afin d'assurer la rapidité du tri mais impose des traitements compliqués afin de le rendre efficace.

Nous proposons le test de plusieurs techniques de filtrage sur des images fixes et aussi les méthodes de segmentation. Nous avons également fait un tour d'horizon des différentes approches existantes de la segmentation. L'approche détection de contour et étiquetage semblent être habituellement retenue, ainsi que l'approche LPE, basée sur la détection des régions dans l'image sous forme de matrice qui présente des inconvénients tels que la taille mémoire et le temps de calcul des paramètres élevé.

Pour remédier nous avons introduit l'approche corrélation qui est une méthode efficace et rapide pour la localisation de certaines formes géométriques qui consiste à comparer l'objet à reconnaître (la cible), en général figurant dans une scène et des objets prédéterminants (base de connaissances) par le calcul de centre de gravité de l'objet. Enfin nous proposons d'intégrer un automate programmable qui servira l'échange des données via l'application Matlab par le port LPT1 sous un outil interface de vision Homme machine (HMI) qui facilitera une supervision en temps réel.

**Mots clés :** Vision artificielle, Reconnaissance d'objets, Traitement d'images, Analyse d'images

## ABSTRACT

In this paper we present methods and techniques of computer vision and existing ones that are the subject of current research. The object recognition is one area of study the most active computer vision, one of the main tasks is to identify and locate these objects based on information received from the environment.

Object recognition aims to automatically and efficiently extract the content interesting, relevant and useful to the scene that is to say, to determine the identity of objects forming the scene and their spatial arrangement in the context of a task or process to achieve. This is a very important issue in computer vision because it is essential for all applications.

In this work we were interested in building a system for recognizing objects in an image to a triage application. The line of work is to recognize and sort objects (geometric shapes) in an image in computer vision. The images come from a fixed camera with a field of view with a regular and permanent result on a conveyor angle, enabling objects to be observed.

Object recognition requires knowledge of the characteristics of our camera. Thus in order to calibrate the camera, we proposed a simple webcam camera oriented to get a picture through the center of the conveyor. It is therefore important to start the design of a chain of image processing by the issue of lighting. Good lighting can sometimes greatly simplify the processing step and gives very good results in object recognition system. The goal is finding objects in images, it will be unit image and not image sequences. In this case the introduction of automatic methods is necessary to ensure the rapid screening but requires complicated treatments to make it effective.

We offer the test several filtering techniques on still images and also segmentation methods, We also made an overview of different existing approaches to segmentation, edge detection approach and labeling appear to be generally accepted, and EPA's approach, based on the detection of regions in the image matrix form has drawbacks such as memory size and computation time of high parameters.

To remedy we have introduced the correlation approach is an effective and rapid localization of certain geometric shapes that compares to recognize the object (the target), usually appearing in a scene and objects predetermining (based knowledge) by calculating the center of gravity of the object. Finally we propose to incorporate a PLC that will exchange data via the application by Matlab LPT1 as a tool of vision Human-Machine Interface (HMI) which will facilitate real-time monitoring.

**Keywords :** Computer vision, object recognition, image processing, image analysis.



# Introduction Générale

Etre non voyant représente un lourd handicap et une assistance permanente dans la vie de tous les jours et à tout moment car parmi les cinq sens attribués aux êtres vivants et plus particulièrement à l'être humain, la vue est sans doute celui qui est considéré comme le plus important. Agir aveuglément, avancer à l'aveuglette, avoir une vision à long terme, avoir la vision courte, voir venir quelqu'un, pour voir venir, etc. Il existe ainsi des dizaines d'expressions qualifiées d'imagées que nous employons continuellement et qui démontrent l'importance prépondérante de la vue dans notre perception du monde. Mais de nos jours la vue naturelle, humaine, peut ne pas suffire, ou être incapable de soutenir longtemps certaines cadences. A vrai dire une forme de vision artificielle ou d'assistance à la vision a vu le jour avec Galilée lorsqu'il a fabriqué l'ancêtre du télescope d'aujourd'hui. Ce qui a permis de découvrir le système solaire. L'homme, actuellement perdu entre l'infiniment petit et l'infiniment grand, ne peut se passer d'outils de vision.

Il est donc tout à fait naturel que la vision artificielle, c'est-à-dire la conception, la fabrication, l'utilisation de machines imitant cette faculté, soit l'objet de tant d'efforts scientifiques et technologiques et soit au centre d'enjeux économiques aussi considérables. Au delà des progrès vertigineux de l'électronique en général et de l'informatique en particulier, c'est la mise au point de la technologie dite CCD (Charge-Coupled Device) qui a ouvert les portes du développement de la vision artificielle[1].

La vision repose sur deux organes liés intimement l'un à l'autre, l'oeil et le cerveau. L'image est captée par l'œil, l'information contenue dans cette image est prétraitée et comprimée dans l'oeil lui-même avant d'être transmise aux cellules cérébrales proprement dites pour y être traitée et interprétée. C'est cette association d'un capteur d'images et d'un système de traitement fournissant un ensemble d'informations de haut niveau que nous appellerons un système de vision.

Le principe de fonctionnement du système de vision artificielle est simple : une machine associant un capteur d'images et un système de traitement qui transforme et interprète cette image pour fournir une information pertinente sur la scène observée. Du moins, c'est le système optimal dont dispose la technologie actuellement.

Les principaux problèmes industriels rencontrés au niveau de nos entreprises sont liés aux contrôles qualité, classification et vérification. Certains de ces problèmes nécessitent l'intervention humaine répétitive et contraignante ou demandent une installation mécanique

classique, complexe et peu évolutive. Or, la qualité et le temps sont souvent les seuls critères retenus par les applications, notamment en temps réel. Pour cela nous envisageons, dans le cadre de ce projet une application d'un système de vision qui doit être doté d'une représentation interne permettant de surmonter ces difficultés et de trouver une solution au problème complexe de la vision dans le domaine du tri et de la détection des défauts.

Ce système de vision, devient indispensable dans la fabrication d'objets en très grande série, où la phase finale et en même temps décisive est le contrôle de qualité.

Devant le nombre, très important, et par conséquent le défilement très rapide de ces objets, la vision artificielle, en l'occurrence, une Webcam, reliée à un ordinateur, se charge de visualiser ces mêmes objets se déplaçant sur un convoyeur. Elle acquiert ainsi, une image, qui sera traitée et ensuite analysée par des algorithmes de plus en plus complexes. L'analyse déterminera simultanément le recensement et le comptage arithmétique puis la détection de défauts basés principalement sur la comparaison de la forme d'un objet avec une forme préalablement définie et par cette opération fera le tri. Un système annexe et complémentaire se chargera grâce à des capteurs et un automate programmable pour envoyer un signal aux bras poussoirs qui ont pour fonction d'éjecter chaque objet dans le bac correspondant.

Il s'agit d'une certaine manière de remplacer la vision humaine par une autre, artificielle, qui doit avoir les mêmes performances en plus de la rapidité, de l'endurance et de l'absence totale de distraction ou de perturbation.

De nombreuses applications de la vie courante, réalisées de manière informatique, utilisent la reconnaissance d'objets, notamment le tri des fruits et légumes sur un tapis roulant ou le tri du courrier par reconnaissance des codes postaux.

Nous nous proposons, dans les prochains chapitres d'étudier successivement ces grandes parties dans leurs principes et de présenter les principales méthodes utilisées dans le système de vision par ordinateur.

## Objectif et Problématique

L'objectif de ce travail est de présenter les méthodes et techniques de la Vision par Ordinateur existantes et celles qui font l'objet de recherches actuelles. Cette discipline aux nombreux problèmes non encore résolus se trouve au carrefour des mathématiques appliquées et de l'informatique. La Vision par Ordinateur est à la base de tout système de vision artificielle qui prend en entrée une ou plusieurs images numériques, et qui effectue des traitements sur ces images afin d'en extraire des informations relatives à la scène observée.

Dans ce travail, nous allons tenter de bâtir un système de reconnaissance d'objets dans une image pour une application de tri et de contrôle de qualité. L'objectif est la recherche d'objets dans des images. Il s'agira d'images unitaires et non pas de séquences d'images.

Reconnaître un objet dans une image, c'est reconnaître sa forme, sa couleur, sa taille... Bref tout ce qui caractérise cet objet et le différencie d'un autre. Ceci est important pour traiter les images de façon automatique (informatique) en vue d'applications industrielles ou informatiques. Le problème est de reconnaître plusieurs objets en 2 dimensions (2D) dans une image. Dans ce cas la mise en place de méthodes automatiques est nécessaire afin d'assurer la rapidité du tri mais impose des traitements compliqués afin de le rendre efficace. Il existe plusieurs méthodes pour y arriver. Elles diffèrent par leur rapidité, leur robustesse au bruit (perturbation), leur insensibilité aux transformations géométriques et photométriques (luminosité), leur réaction face à des événements imprévus.

Cependant pour toutes les méthodes le schéma général reste quasiment identique. A savoir, un premier traitement de l'image pour la rendre plus facilement exploitable, (élimination du bruit, filtrage, rehaussement de contraste, détection des contours,...), l'extraction de diverses caractéristiques (longueur, aire, histogramme, etc.) qui serviront à reconnaître l'objet pour pouvoir le comparer de manière efficace et finalement permettre sa classification en fonction des descripteurs.

Il y a plusieurs façons de recueillir de l'information sur l'environnement, mais la façon que nous avons préconisée est l'utilisation d'une caméra ordinaire, d'un traitement et d'une analyse de l'image assez profonds. C'est la méthode qui permet d'avoir à la fois accès à une image que l'on pourrait par la suite projeter et mettre le plus à profit des connaissances informatiques.

Nous évoquerons ensuite les conditions de luminosité et les dispositifs d'éclairage afin de corriger les défauts optiques induits par le matériel et l'environnement ayant servi à acquérir

l'image. Il est également possible de nettoyer l'image lors de la numérisation par des concepts de filtrage. Ces filtres possèdent des caractéristiques spectrales; on parle ainsi de filtres linéaires et de filtres non linéaires.

La vision n'est pas un domaine facile. Repérer un objet simple dans une image demande beaucoup d'opérations. Comme nous utilisons une caméra simple pour recueillir de l'information, il est nécessaire de faire un traitement et une analyse sur les images en provenance de la caméra.

Ainsi, tout d'abord nous avons procédé à une recherche afin de trouver des solutions permettant de facilement recevoir les images en provenance de la caméra et de leur faire subir un traitement approprié le plus simplement possible.

En effet, certains phénomènes peuvent apparaître. Notamment de bruit (par exemple lors d'une mauvaise réception de données), de flou (dus à une mauvaise mise au point), ou encore des pertes de qualité (dus à une mauvaise luminosité).

Les dégradations que subissent les images sont en général dans les phases que l'on vient d'énoncer. Notre problème consistera donc à récupérer une image proche de l'image originale à partir d'une image de mauvaise qualité. Dans cette phase, nous ne nous concentrerons que sur le problème du bruit.

Une étape clé du traitement de l'image est le seuillage c'est-à-dire le passage de l'image numérique à l'image binaire. Différents traitements peuvent être réalisés, que ce soit sur l'image numérique ou l'image binaire. Effectués après le seuillage, les traitements binaires sont destinés à définir le plus précisément possible les objets à analyser. La binarisation devrait prendre compte autant de la différence des couleurs que de la différence de luminance.

Ensuite, on utilise les morphologies mathématiques : dilatation, érosion, ouverture et fermeture. Ces opérations nous permettent de changer la forme d'une image (toujours par rapport à sa forme originale) pour mettre en évidence certaines caractéristiques.

Exprimée simplement l'opération de dilatation est équivalente à « gonfler » une image, tandis qu'une opération d'érosion équivaut à la « dégonfler ».

Dans une opération de dilatation, la valeur de chaque pixel de sortie est la valeur maximale de tous les pixels de son voisinage. Il y a plusieurs façons de définir un tel voisinage. Pour l'instant, on se contente d'établir un voisinage carré de  $3 \times 3$ .

L'étape suivante de notre processus de reconnaissance d'objet est la segmentation qui est une étape essentielle et reste un problème complexe.



La segmentation est un processus de la vision par ordinateur, généralement c'est la première étape de l'analyse d'image qui vient après le prétraitement. C'est un passage obligé pour la vision par ordinateur. Donc la segmentation est l'extraction de caractéristiques de l'objet, ce qui permet une distinction entre l'objet et le fond. Une erreur dans la segmentation de l'objet à reconnaître augmente forcément le risque d'une mauvaise reconnaissance. Il existe une multitude de méthodes de segmentation dont l'efficacité reste difficile à évaluer.

Une fois les objets isolés, la prochaine étape pour la reconnaissance d'objets est la représentation et l'interprétation des données (descripteurs) représentées par les caractéristiques. La comparaison de l'objet à reconnaître (la cible) qui figure dans la scène et des objets de la base de connaissances en appliquant la méthode de corrélation par le calcul de centre de gravité de l'objet afin d'améliorer le temps du calcul et le taux de reconnaissance.

Viendra, en dernier, la phase expérimentale ou l'application par le langage MATLAB pour valider la conception théorique du processus de reconnaissance d'objets. Après cette réalisation, une deuxième réalisation consiste à envoyer des signaux par le biais du port parallèle (LPT1) à un automate qui actionne les bras poussoirs selon les caractéristiques de l'objet. Lorsque l'objet n'est pas identifié, il progresse et il est éjecté dans le palettiseur des objets inconnus.

## Présentation du mémoire

**Le premier chapitre** décrit une description sur la vision humaine dans la vie active et avec plus de détails de position sur la vision artificielle aussi il est réservé aux étapes de processus de la reconnaissance d'objets avec ces différentes technologies des moyens d'acquisition de l'information ; et les moyens utilisés pour notre projet. Nous justifions également le choix d'utilisation d'une simple caméra pour le traitement et l'analyse de l'image, ainsi que les dispositifs d'éclairage.

**Le deuxième chapitre** est essentiellement consacré aux techniques de traitements de l'image. Pour ainsi dire, l'emploi d'outils ou d'algorithmes de seuillage, filtrage et les opérations mathématiques de morphologie afin de récupérer une image proche de l'image originale pour la rendre plus facilement exploitable, constitue la seconde partie du processus.

**Le troisième chapitre** est consacré à un problème intéressant en analyse numérique d'images. C'est un sujet d'actualité et un problème qui reste ouvert où l'on retrouve de très nombreuses approches de segmentation qu'on va traiter. L'utilisation d'algorithmes de segmentation de l'image afin d'extraire des caractéristiques essentielles à la localisation et à la reconnaissance des objets dans l'image, représente une étape primordiale dans notre processus de reconnaissance d'objets.

**Le quatrième chapitre** est l'étape de représentation et interprétation de l'image, cette étape intervient après la segmentation dont les pixels voisins ont besoin d'être représentés et/ou décrits pour bien interpréter la scène. La représentation dépend de la tâche à réaliser. On doit ensuite être capable de décrire une région, à partir d'une représentation, pour cela on a utilisé les descripteurs de contour avec et les descripteurs de région . L'interprétation conduit à attribuer une signification à l'ensemble d'objets reconnus. Elle consiste à comparer l'objet à reconnaître (la cible), en général figurant dans une scène et des objets prédéterminants (base de connaissances) par le calcul de centre de gravité de l'objet, afin d'améliorer le temps de calcul et le taux de reconnaissance.

**Le cinquième chapitre** est la dernière étape des résultats de l'application par un langage de programmation MATLAB. Les outils de programmation sont généralement des algorithmes de traitement d'images, reconnaissance de forme, identification, classification et détection de défaut. Afin de progresser dans notre application MATLAB pour une fin industrielle, nous avons élaboré un outil d'interface de vision homme machine ( HMI) qui

facilitera une supervision en temps réel. Nous avons implanté un automate programmable qui servira l'échange des données via l'application Matlab par le port LPT1.

# Cahier des Charges

L'objectif est de concevoir un cahier des charges pour une application vision et intégrer cette application en choisissant les composants et les outils de traitement adéquats et ainsi découvrir les fondements du traitement d'images pour la reconnaissance d'objets.

L'axe de notre travail consiste à reconnaître et à trier les objets dans une image en vision artificielle. Les images proviennent d'une caméra fixe ayant un champ de vision régulier et permanent avec un angle conséquent sur un convoyeur, porteur des objets à observer.

La reconnaissance d'objets implique de connaître les caractéristiques de notre caméra. Ainsi pour pouvoir calibrer une caméra, nous avons proposé une simple camera Webcam orientée de manière à obtenir une photo par le centre du convoyeur.

Pour estimer la position et l'orientation entre les objets et une caméra, à partir d'une seule image, nous prévoyons d'élaborer un convoyeur qui va en se rétrécissant de manière à éviter l'encombrement en faisant passer un objet à la fois. Incontestablement cela rendrait optimale la qualité de vision. Lorsque la position de l'objet correspond à la position d'acquisition, le convoyeur s'arrête. Grâce à un système d'acquisition de données, les données constituant l'image numérique seront transmises à un ordinateur. Les fonctions correspondantes sont réalisées par un programme informatique que nous avons implanté (langage : MATLAB).

Le tri se fera pratiquement selon les objets qui défileront régulièrement et selon un rythme et une fréquence déterminée par le goulot de rétrécissement du convoyeur.

Le système est équipé du matériel suivant :

- Un convoyeur.
- Une camera fixée au dessus du convoyeur (position en vue verticale).
- Un micro-ordinateur (PC).

L'étude expérimentale est réalisée en laboratoire grâce à des objets qu'on a fabriqué à l'extérieur, déposés sur un support dit convoyeur. L'expérience réalisée par une caméra Webcam. La mise en place des objets est réalisée manuellement par nous même, afin de suivre le déplacement des objets sur le convoyeur.

Une fois l'acquisition vidéo réalisée, cette dernière est ensuite numérisée. Notre travail est bâti sur une image fixe et non une séquence d'images. Cette image est analysée, permettant la localisation et l'identification des objets, à l'aide d'outils informatiques.

Notre projet consiste à développer un logiciel qui automatise la phase d'analyse de l'image. Donc il devra déterminer l'identification des objets en fonction de leur position et leur forme dans le convoyeur.

Pour la simulation et la concrétisation en vue d'une application à une échelle industrielle, les éléments suivant sont indispensables :

- Un micro-ordinateur
- Une camera
- Un convoyeur
- Un automate (API M 340)
- Deux capteurs
- Une application de supervision VIJEO CITECT pour l'interface homme machine (IHM)

Cette partie est décrite en détail dans le 5<sup>ème</sup> chapitre.

## Concepts de la Vision humaine et de la Vision artificielle

*La vision permet la reconnaissance des objets et du monde en relief autour de l'être humain. La reproduction de ce monde a constitué une préoccupation permanente de l'homme depuis les anciennes peintures rupestres, les sculptures antiques, jusqu'aux films actuels, où la caméra remplace l'œil humaine ''voit '' le monde d'une certaine manière de telle sorte que l'on a abouti à une double vision: celle de la caméra d'abord et celle des scènes ensuite. Dans ce chapitre nous allons décrire une description sur la vision humaine et la vision artificielle et nous justifions le choix d'une caméra simple pour le traitement et l'analyse de l'image ainsi les dispositifs d'éclairage.*

### 1.1 Introduction

L'univers est, par nature, fait de volumes, c'est-à-dire de corps, remplissant un vide plus ou moins important. Et de ce fait, l'espace qui nous entoure a une structure tri-dimensionnelle (3D). Lorsque l'on demande à une personne de décrire ce qu'elle voit elle n'éprouve aucune difficulté à nommer les objets qui l'entourent téléphone, table, livre,... [2]. Pour définir la vision par ordinateur en une phrase, nous dirons que cette discipline s'attache à extraire de manière automatique des informations à partir d'images, en vue d'effectuer des tâches spécifiques.

La vision est un terme générique englobant plusieurs aspects de l'imagerie par ordinateur (computer imaging) [3]. On peut définir le domaine de la vision comme le regroupement de trois disciplines comme l'indique la figure 1.1: la vision artificielle (computer vision), le traitement d'image (image processing) et l'infographie (computer graphics).

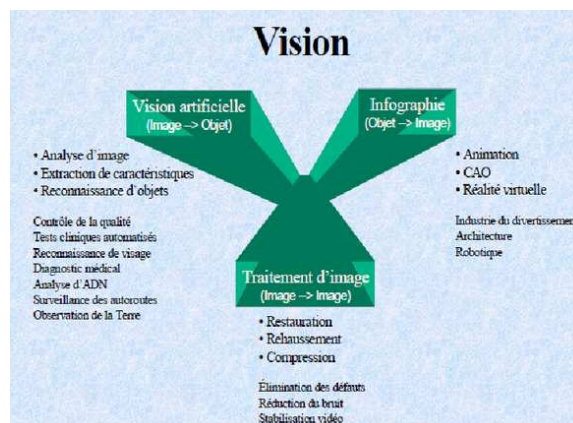


Figure 1.1 : Domaine de la vision

La vision est incontestablement le sens le plus développé chez l'homme : Les humains ont une capacité de remarque, d'analyse et d'interprétation en temps réel des scènes complexes. Chaque pixel contient des informations sur la lumière (quantité et contenu spectral/couleur) reçue en ce point de la rétine.

Les objets (téléphone, voiture...) n'existent pas sur la rétine, et pourtant on les voit : leur interprétation est le résultat du processus visuel.

Par ailleurs, la vision par ordinateur est devenue ces dernières années un domaine en plein essor de par ses multiples applications en médecine, robotique, surveillance etc.

Ainsi, l'étude du système visuel humain ou animal dans le but d'en comprendre et copier les mécanismes pour la vision par ordinateur est particulièrement intéressante [4].

La contribution de la vision permet d'améliorer la productivité en gardant une réactivité acceptable. Par ailleurs, la vision permet la localisation ainsi que la minimisation du nombre de pièces en rebut. Le remplacement des capteurs classiques par un système de vision permettra un gain matériel et financier, et ce, au niveau du coût de l'installation et de sa maintenance.

## **1.2 Vision Humaine**

Souvent imité, mais jamais égalé, l'œil, roi des organes sensoriels, permet aux êtres vivants de se localiser en dynamique dans un environnement. Il est également complémentaire des autres organes extéroceptifs qu'utilisent les humains. L'œil occupe une place privilégiée chez les êtres vivants. Il n'est pas étonnant que les scientifiques essaient de le reproduire artificiellement à l'aide de caméra, d'ordinateur et d'algorithme [8].

L'information qui est réellement disponible sur la rétine de nos yeux n'est ni plus ni moins une collection de points très nombreux. En chaque point ou pixel (Picture element), il y a tout simplement une information qui donne une indication quant à la quantité de lumière et la couleur qui proviennent de l'espace environnant et qui ont été projetées à cet endroit de la rétine. Le téléphone, la table ou le livre n'existent pas sur la rétine [9].

C'est leur reproduction en négatif sur cette même rétine, grâce notamment à la réflexion lumineuse que la vision peut avoir lieu. C'est pour quoi l'obscurité totale empêche la vision.

De même que l'excès de lumière perturbe énormément celle-ci et provoque l'éblouissement.

## **1.3 Le système visuel humain**

L'information visuelle est reçue par l'œil puis transmise au cerveau après différents traitements préliminaires réalisés par la rétine. Les traitements de plus haut niveau sont réalisés dans le

cerveau au niveau des aires visuelles que l'on trouve dans la partie arrière des deux hémisphères cérébraux (figure 1.2) [10].

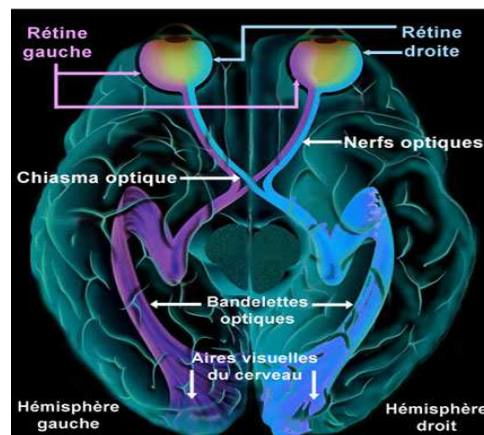


Figure 1.2 : Domaine de la vision

Les traitements rétiniens, après transmission de l'information via le nerf optique, puis les traitements effectués par la première aire du cortex visuel primaire [11].

### 1.3.1 Sensibilité à la lumière

La lumière traverse les couches neuronales de la rétine et arrive au niveau des photorécepteurs (cônes et bâtonnets). Ces cellules ont pour but d'effectuer la transduction du stimulus lumineux en un potentiel de membrane. Mais, les photorécepteurs ne peuvent donner une réponse fiable que sur une dynamique d'une à deux décades seulement alors que la dynamique à coder par ces capteurs est de l'ordre de 10 unités logarithmiques [12].

### 1.3.2 Sensibilité à la couleur

Les photorécepteurs présentent des sensibilités spectrales (couleur) et des sensibilités en amplitude (quantité de lumière) différentes. On trouve deux types de photorécepteurs:

Les bâtonnets :

- responsables de la vision scotopique : faible intensité lumineuse (vision de nuit);
- très sensibles à la lumière;
- réponse lente aux variations d'illumination.

Les cônes :

- responsables de la vision photopique : vision dans les hautes intensités lumineuses (vision de jour);
- responsables de la vision de précision (haute résolution);
- déterminent la luminosité des scènes visualisées;



- permettent l'extraction des couleurs grâce à 3 types de cônes.

Le système de vision doit fournir les connaissances nécessaires afin de permettre une interprétation non ambiguë [9].

Le modèle de vision vers lequel la vision par ordinateur essaie de tendre, car :

- l'oeil est une caméra très performante ;
- le cerveau est un ordinateur très performant.

La figure 1.3 montre une comparaison entre le capteur naturel, l'œil et un capteur numérique

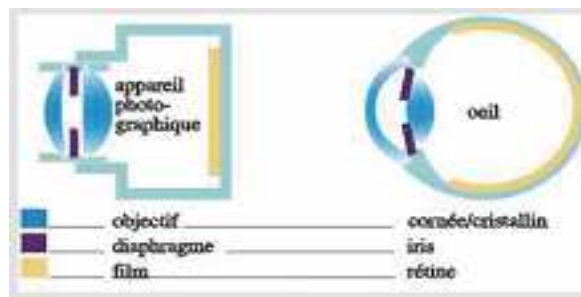


Figure 1.3 : Comparaison entre le capteur naturel, l'œil et un capteur numérique

## 1.4 Vision Artificielle

La vision artificielle ou vision par ordinateur est le cadre général de ce travail. Pour la définir, il est nécessaire de rappeler que cette discipline s'est attachée depuis ses débuts à développer des méthodes d'interprétation automatique des images, en particulier, des méthodes permettant de reconnaître ou d'identifier des objets présents dans ces images.

Cette discipline consiste à convertir une image en données objets ou, plus explicitement, à identifier les objets contenus dans l'image par l'extraction et l'analyse de caractéristiques abstraites (features) à partir des pixels, suivant un processus de reconnaissance de forme similaire à celui opéré par l'humain [5]. Quelques images réelles qui utilisent la technologie de la vision artificielle.



Figure 1.4 : Contrôle de qualité par vision artificielle



Figure 1.5 : Le scannage des plaques est réalisé par un système de vision artificielle



Figure 1.6 : Des yeux numériques trient les agrumes



Figure 1.7 : Identification et manipulation des pièces à la géométrie complexe par et système de vision artificielle

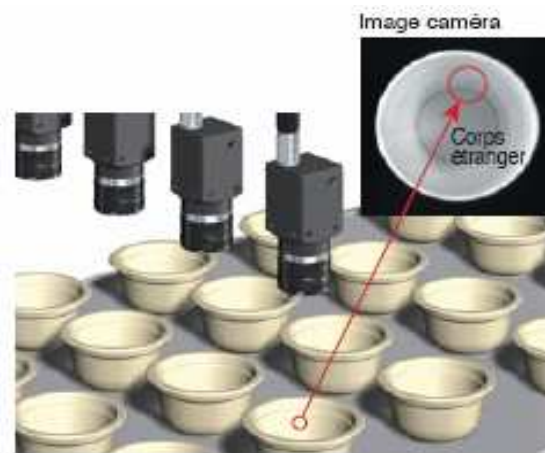


Figure 1.8 : Détection de salissures/ de corps étrangers à l'intérieur d'une coupelle

Le développement d'une machine de vision impose de maîtriser les paramètres d'acquisitions des images, de traiter ces images, d'extraire de ces images les caractéristiques significatives et finalement de poser un diagnostic en fonction du contexte [6] [7].

### **1.4.1 Avantages des applications de Vision Artificielle**

Parmi les applications de la vision par ordinateur, on compte notamment le tri et le contrôle de la qualité sur une chaîne de production, l'identification d'un individu par biométrie (par ex., l'analyse automatisée des empreintes digitales), le diagnostic médical, la classification de terrains (forêts de conifères ou de feuillus, plaines, etc.) à partir d'images satellites.

Grâce au progrès de l'électronique et de l'informatique, la puissance actuelle des ordinateurs nous permet de contrôler plusieurs produits par seconde. La gamme de capteurs vidéo (caméras couleur, noir et blanc, linéaires,...) offre la possibilité d'analyser des produits fabriqués en continu (tissus, pâtes, produits d'extrudeuses...) ou unitairement (bouteilles, pièces de voitures...)

Les domaines observables s'étendent du très petit (cellules, cachets) aux objets de taille imposante (sièges de voitures, pneus...). Les produits peuvent être arrêtés au moment de l'analyse ou en déplacement. C'est dire que le domaine de la vision artificielle est vaste. La vision artificielle s'est considérablement améliorée au point de devenir un capteur intelligent.

Dans ce vaste domaine qu'est la vision par ordinateur, nous nous intéressons ici à la sous discipline qu'est la reconnaissance d'objets, et plus particulièrement à la localisation de classes d'objets dans des images (par exemple localiser les objets comme le carré, le cercle ou le triangle dans une image). Ce type de technique combine des approches de traitement d'images et l'identification automatique dans le but de mener à bien cette tâche de reconnaissance des objets présents dans les images.

## **1.5 Processus de reconnaissance d'objets**

### **1.5.1 Introduction**

Le mot "reconnaître" se décompose naturellement en re- et -connaître. Dans un premier temps, il y a une phase d'acquisition de connaissances (-connaître), il s'agit d'apprendre à caractériser l'objet qui nous intéresse. Qu'est-ce qui fait que nous pouvons reconnaître une voiture? Qu'est ce qui la différencie des autres objets? Il y a ensuite une phase d'utilisation de cette connaissance (re-), qui analyse de nouveaux objets pour déterminer s'ils appartiennent à la catégorie apprise[9].

L'idée de construire des machines capables d'imiter les êtres humains afin de les aider dans certaines tâches, voire de les remplacer, était antérieure aux ordinateurs [14].

On sait que, le plus souvent, des équations différentielles ou aux dérivées partielles sont au cœur de ces réussites. Et l'on sait également qu'il reste toujours à en calculer les solutions !

La discipline scientifique de la reconnaissance des formes (RdF - Pattern Recognition, en anglais) s'est développée depuis la fin des années 60, en même temps que l'intelligence artificielle (IA), et ceci au fur et à mesure de l'augmentation de la puissance et des capacités de mémoire des ordinateurs. Très tôt, dès le début des années 70, les deux disciplines se sont séparées.

Malgré de nombreux efforts des uns et des autres, les résultats des premiers programmes d'IA ou de RdF se sont révélés très éloignés des performances humaines ou même animales. Soulignons qu'une machine informatique programmable - un ordinateur - est capable d'opérer n'importe quelle transformation d'une donnée de départ, la représentation, en une image d'arrivée, une interprétation [16].

Cette capacité a ouvert des domaines d'application bénéfiques au monde moderne et a ses nombreuses technologies. Parmi ces domaines :

- L'analyse automatique des images médicales obtenues de microscopes, de digitaliseurs, d'appareils à résonance magnétique, d'appareils à rayons X, d'appareils de médecine nucléaire, de caméras conventionnelles, etc.
- Inspection automatique de pièces sur une ligne d'assemblage.
- Assemblage automatique de pièces.
- Reconnaissance de la parole par ordinateur.
- Reconnaissance de texte par ordinateur.
- Classement automatique de bois, acier, fruits, etc.
- Classification de signaux sismiques.
- Reconnaissance et classement d'objets différents dans l'ordre souhaité.
- Contrôle de qualité.
- Identification des personnes à partir des empreintes digitales, de la forme des mains, des empreintes rétiniennes, de la voix, etc.
- Analyse automatique d'images satellitaires pour le monitoring des cultures, des forêts, des changements climatiques, des réserves d'eau, des icebergs, des catastrophes naturelles (fuite de pétrole, volcan, ouragan), etc.

- Classification des électrocardiogrammes selon les catégories de maladies cardiaques.
- Télésurveillance.

Ce sont les domaines extrêmement variés, vitaux pour les civilisations d'aujourd'hui, toutes exigées sur l'observation d'abord et la compétence de plus en plus pointue ensuite.

Il s'agit d'une tâche fondamentale qui fait l'objet de recherches très actives en tant qu'étape indispensable pour de multiples applications de vision par ordinateur.

Et qui sera l'objet de ces prétraitements, traitements et l'analyse décrits dans les chapitres suivants.

## **1.5.2 Etapes du processus de reconnaissance d'objets**

### **1.5.2.1 Introduction**

Le monde réel est composé essentiellement d'objets tridimensionnels. Une des tâches principales d'un système de vision artificielle est de pouvoir identifier et localiser ces objets à partir d'informations reçues de l'environnement.

La reconnaissance d'objets dans les images est un des problèmes les plus difficiles en vision par ordinateur puisqu'il est indispensable pour toutes sortes d'applications. Elle est en même temps une étape primordiale pour la mise en oeuvre de plusieurs applications actuelles qui nécessitent une interprétation de haut niveau d'images.

Dans ce travail, nous allons tenter de bâtir un système de reconnaissance d'objets dans une image pour une application de tri et de contrôle de qualité. L'objectif est la recherche d'objets dans des images : il s'agira d'image unitaire et non pas de séquences d'images.

De façon symbolique, un système de vision artificielle pourra toujours être représenté par le schéma de la figure 2.1, où on trouve bien les étapes du processus de reconnaissance d'objets

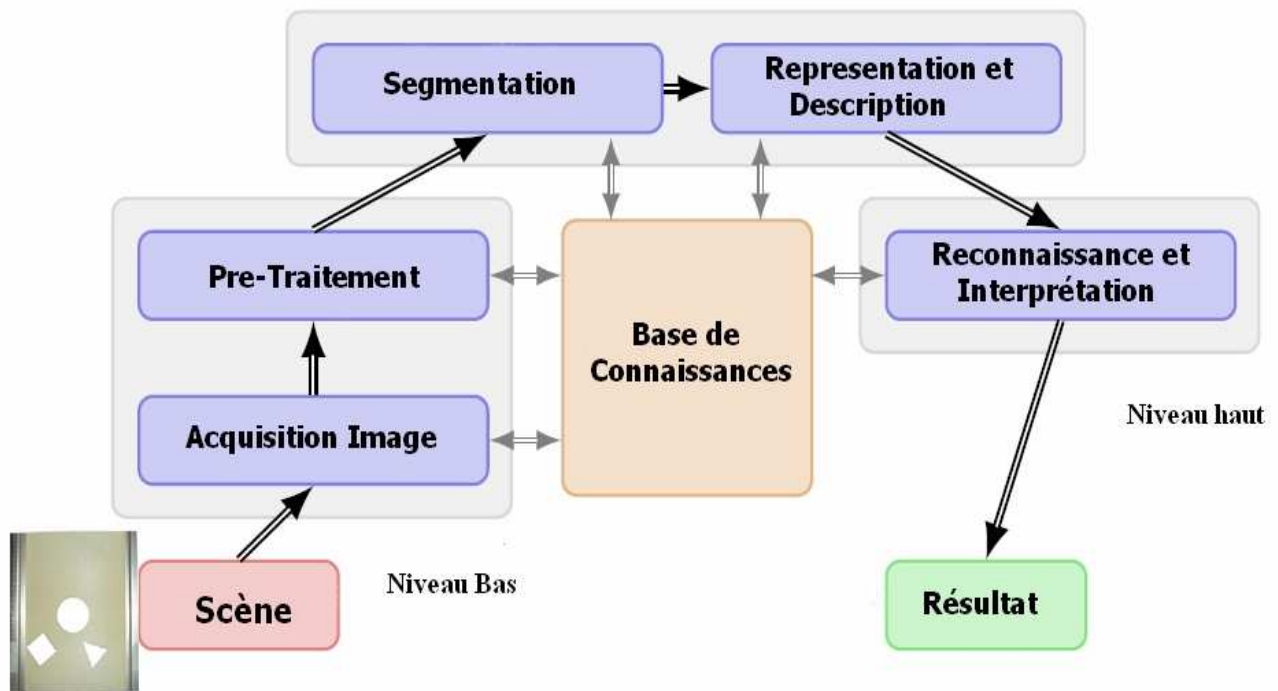


Figure 1.9 : Principales étapes de la reconnaissance d'objets

Chacune des étapes ci-dessus exige une connaissance préalable telle que: la taille de l'objet, la forme, la distance entre les objets, les conditions environnementales, le nombre d'objets à reconnaître dans l'image. Toutes ces informations sont nécessaires pour l'utilisation des techniques de traitement d'images et entre autres les développements d'algorithmes pour la reconnaissance d'objets. Chacune de ces étapes est décrite plus en détail dans les chapitres suivants.

### 1.5.3 COMPOSANTS D'UN SYSTEME DE VISION

Les composants matériels requis pour un système de vision artificielle sont décrits dans la figure 1.10. On peut distinguer l'objet à traiter, éclairage, caméra, le convoyeur, les bras poussoirs, l'automate, les capteurs et le processus de traitements et d'analyse d'images par le micro-ordinateur qui procède à l'opération de tri et de rejet qu'on va détailler dans les prochains chapitres.

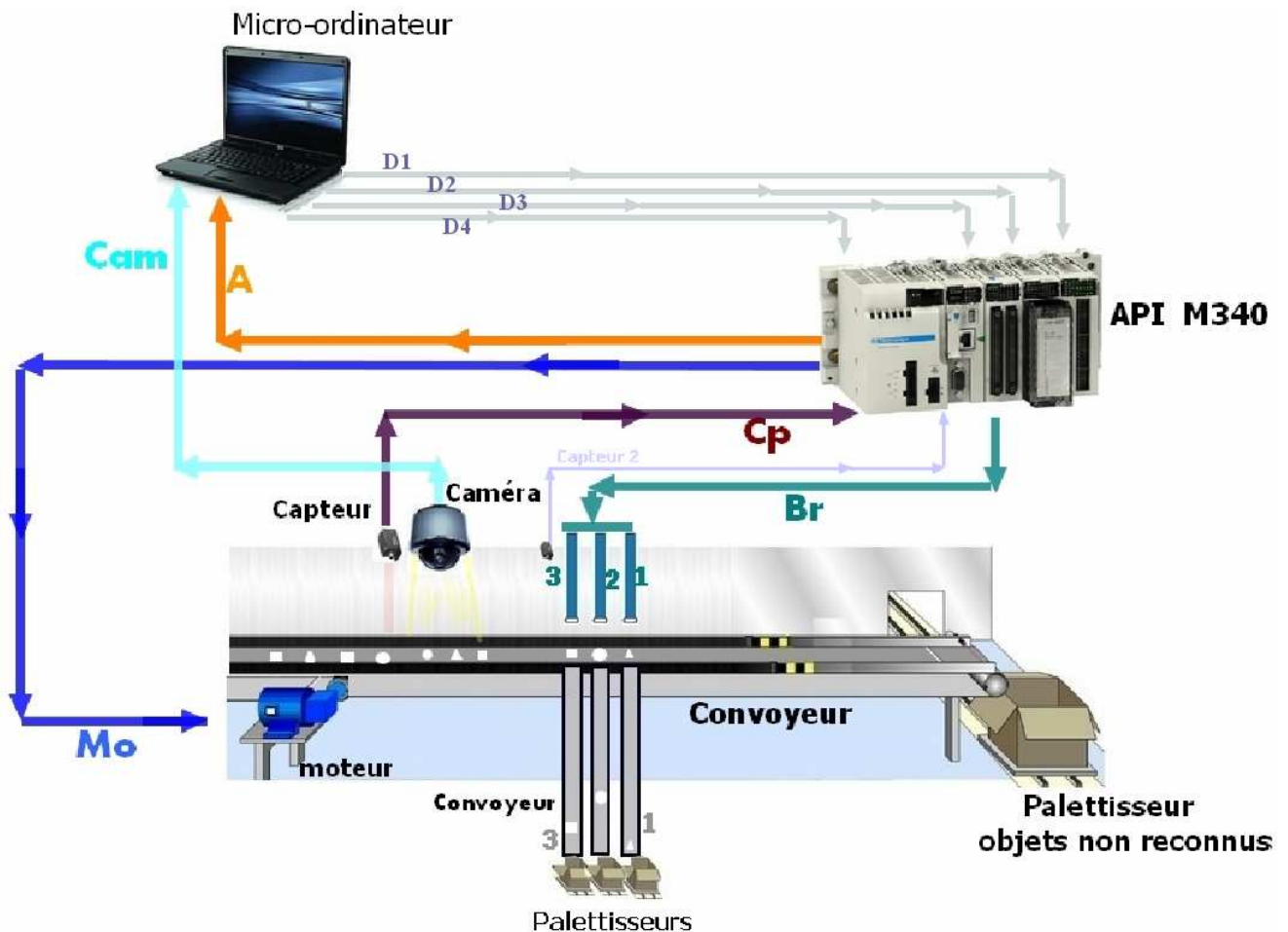


Figure 1.10 :L'implantation d'une simulation qui représente toutes les étapes de reconnaissance d'objets pour l'opération de tri et de contrôle de qualité.

### 1.5.4 Description du processus de reconnaissance

Par « images » il faut plutôt entendre « l'image » d'une scène, ou d'un objet, formée dans le plan focal d'une caméra (c'est cette « image de la scène », ou « image de l'objet » qui devient le fichier informatique source destiné à être traité par ordinateur).

**La scène :** Le processus étudié concerne un système de tri des objets de différentes formes (cercle, carré et triangle). En effet, les objets arrivent d'une façon aléatoire sur un convoyeur. L'équipement matériel utilisé dans ce processus est constitué d'un convoyeur, d'une caméra et d'un micro-ordinateur.

Le convoyeur permet de faire défiler les objets sous une caméra numérique - La caméra numérique prend les images des objets et les envoie vers l'ordinateur. L'ordinateur reçoit les images, les analyse et prend la décision pour trier et contrôler les objets pour une décision finale. Le convoyeur s'arrête au moment de la prise d'image par la caméra (figure 1.11). La suite de cette opération est d'éliminer la bordure du convoyeur afin de cadrer l'image. La scène est



représentée par un convoyeur, des objets (cercles, carrés, triangles) posés sur le convoyeur, une caméra et une lampe d'éclairage.

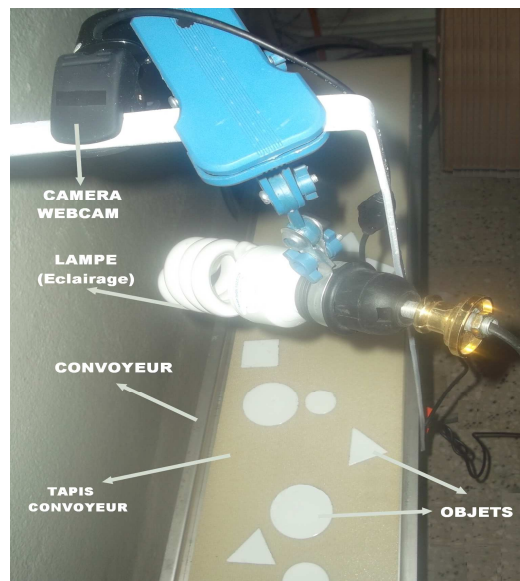


Figure 1.11 : Convoyeur

**Recadrage d'images :** Ces opérations géométriques sont utilisées pour modifier la taille d'une image (ou d'une fenêtre d'image)



Figure 1.12 : Recadrage d'image

La réduction d'image va consister à remplacer un groupe de pixels par un seul pixel. Le niveau de gris correspondant pouvant être un des niveaux de gris d'un pixel du groupe ou par une valeur intermédiaire calculée à partir de tous les niveaux de gris des pixels du groupe.

L'image est spécifiée en termes de coordonnées spatiales, les éléments de la largeur et la hauteur de l'image ne correspondent pas toujours exactement avec la taille de l'image de sortie, comme indique la figure 2.3. Par exemple, supposons que le cadre de l'image est un rectangle de  $[20 \ 20 \ 40 \ 30]$ , en utilisant le système de coordonnées spatiales par défaut. Le coin supérieur gauche du rectangle spécifié est le centre du pixel  $(20,20)$  et le coin inférieur droit est le centre du pixel  $(50,60)$ . L'image de sortie résultant est de 31 par 41, pas 30 par 40, car l'image de sortie inclut tous les pixels de l'image d'entrée qui sont totalement ou partiellement fermés par le rectangle.



## **1.5.5 Acquisition**

### **1.5.5.1 Introduction**

La caméra est l'un des éléments principaux de la chaîne d'acquisition d'une image. De nombreux types de caméras existent dans le domaine de l'industrie : Caméras analogiques ou numériques, caméras linéaires ou matricielles, caméras multi spectrales.

Le capteur d'images transforme l'énergie lumineuse renvoyée dans sa direction en un signal électrique. Un numériseur convertit la tension électrique (U: Volt) de chaque pixel en compte numérique (CN) codé sur un nombre prédéfini de bits. Le signal est ensuite envoyé à un ordinateur à travers une carte d'acquisition. C'est la phase de numérisation de l'image. Cette carte effectue une conversion analogique/numérique du signal [17]. L'image est ensuite mémorisée dans l'ordinateur pour être ultérieurement traitée et analysée, en vue d'une éventuelle prise de décision.

Le signal est ensuite envoyé à un ordinateur à travers une carte d'acquisition. C'est la phase de numérisation de l'image.

### **1.5.5.2 Caméras, capteurs d'images**

Une caméra, ou plus généralement, un capteur d'images comporte deux parties , une optique et un élément sensible à la lumière. Dans le cas le plus simple l'optique est simplement constituée d'un objectif. Son rôle est d'assurer la projection d'une image nette sur la surface sensible [1].

Dans le cadre de notre travail, concernant la mise en œuvre du processus, le choix de la caméra est porté sur un matériel simple mais suffisant, donc un équipement, pas cher mais satisfaisant pour valider la faisabilité du processus. Si le convoyeur est à l'arrêt, toutes les caméras convergeront. Si par contre le convoyeur ne s'arrête pas devant la caméra, il est recommandé de choisir une caméra progressive, ou à balayage progressif, capable de synchroniser l'acquisition avec le passage du produit tout en gardant sa pleine résolution. Pour cela, le choix s'est porté sur l'utilisation de la caméra « Webcam USB » pour les essais au laboratoire (figure 1.13).

La décision d'utiliser une Webcam, consiste à limiter les coûts et à valider la faisabilité avec du matériel basique et non dédié à la vision.



Figure 1.13 : Webcam

Une caméra USB CCD (Charge-Coupled Device), numérique qui s'installe sur des ports de PC comme le port USB de résolution  $320 \times 240$  et a une vitesse d'acquisition de 15 images par seconde. L'image est de format RGB (3 plans de base Red, Green, Blue). La technologie CCD (Charge-Coupled Device) est inventée en 1975, a connu au cours des trente dernières années d'importantes évolutions. Elle constitue aujourd'hui la technologie de référence.

### a) Capteur à CCD

Le principe du capteur à CCD (Charge-Coupled Device), ou capteur à transfert de charge est un composant basé sur une technologie de composants à semi-conducteurs. L'arrivée de photons sur le silicium va créer des charges électriques, charges qui vont s'accumuler dans des zones appelées "puits". Ces zones sont la résultante de différentes tensions électriques appliquées localement. En jouant sur ces tensions, nous allons modifier la position des puits, et donc forcer les charges à se déplacer. [18]

Une matrice CCD est formée de Y lignes de X colonnes, chaque point élémentaire (ou pixel) étant physiquement un rectangle de silicium de quelques microns.

### b) Capteur à CMOS

Parallèlement à cela, est apparue dans les années 90, une seconde technologie, baptisée CMOS (Complementary Metal Oxide Semiconductor), qui devait révolutionner le monde de la vision. Jugée moins performante que son aînée, celle-ci est restée pendant longtemps cantonnée aux applications bas de gamme, ne nécessitant pas une qualité d'image très élevée.

### c) Technologie Caméras linéaires

Depuis plusieurs années les caméras linéaires jouent un rôle prépondérant dans le monde de l'imagerie. Le principe de base de l'acquisition par une caméra linéaire consiste en la conversion des rayons lumineux (aussi appelés « stimuli ») réfléchis par la scène à observer en un signal vidéo à l'aide d'un capteur de forme linéaire, parfois désigné sous le terme de barrette

dans la littérature. Ce capteur linéaire est constitué d'éléments photosensibles (parfois appelés pixels par abus de langage mais aussi photo-sites ou photo-récepteurs).

Leur disposition linéaire ne permet que l'observation d'une ligne de la scène restituée sous la forme d'une image-ligne, contrairement aux caméras matricielles dont les photo-récepteurs disposés sous la forme d'une matrice permettent l'acquisition d'images bidimensionnelles.[19]

### 1.5.5.3 Eclairage

La détermination des conditions d'éclairage est un des éléments essentiels à la bonne maîtrise de la vision artificielle, un bon éclairage permet parfois de simplifier considérablement l'étape de traitement et donne un très bon résultat au système de reconnaissance d'objets. Il est dès lors primordial de démarrer la conception d'une chaîne de traitement d'images par la question de l'éclairage. La lumière projetée sur l'objet à observer présente une structure parfaitement définie. Cette structure représente un motif connu qui se déforme si la surface de l'objet observé présente un relief. Il faut prévoir les variations de l'éclairage naturel en fonction du temps. Paramètre qui peut fausser certaines données de vision artificielle.

Il existe quelques techniques d'éclairage dans le domaine de vision qu'on trouvera dans la partie Annexe A.

Le but est d'éviter les effets d'ombre pour obtenir une illumination aussi homogène que possible de toute leur surface. Dans notre travail nous avons utilisé un éclairage par une lampe lumière simple pour éclairer le champ de vision et pour pouvoir donner à l'image acquise un traitement approfondi (figure 1.14).

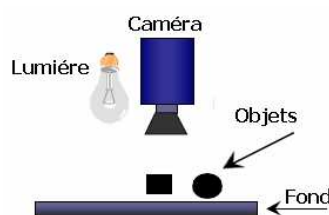


Figure 1.14 : Dispositif d'éclairage

## 1.6 Conclusion

La vision humaine est extrêmement complexe. La vision par ordinateur ne cherche pas à comprendre ou à reproduire la vision humaine, mais à construire un modèle algorithmique qui, vu de l'extérieur, possède des propriétés semblables. D'où la vision artificielle est d'extraire des caractéristiques à partir des images. C'est une mise en correspondance image/reconnaissance.

La reconnaissance d'objets dans les images est un des problèmes les plus difficiles en vision par ordinateur puisqu'il est indispensable pour toutes sortes d'applications industrielles

Connaître la forme et le nombre d'objets sur un convoyeur nécessite une bonne acquisition. Le choix du type de caméra est très important, l'éclairage est l'aspect le plus critique de toute application de vision artificielle, si l'objet est bien éclairé on peut arriver à de très bons résultats de notre processus de reconnaissance d'objets. De nombreuses applications ont ainsi échoué par manque d'éclairage. Mais l'éclairage n'est pas le seul paramètre. Il y en a d'autres pour le traitement numérique de l'image, où un appareillage plus complexe, est indispensable.

## Traitement Numérique d'Images

*Dans toutes les pratiques industrielles, la fée électricité est omniprésente. L'image fournie par le capteur est transformée en un signal électrique. C'est le cas des prétraitements au cours desquels le signal peut toujours être identifié à une image. Dans un deuxième temps, les traitements visent à extraire des caractéristiques qui constitueront les données sur lesquelles l'identification finale sera menée. L'emploi d'outils ou d'algorithmes de traitement comme le seuillage, le filtrage et les opérations morphologiques afin de récupérer une image proche de l'image originale pour la rendre plus facilement exploitable.*

### 2.1 Introduction

L'image fournie par le capteur est transformée en un signal électrique. De ce signal il faut extraire les informations recherchées sur le contenu de la scène dont l'image a été captée.

Les premières bases du traitement d'images sont directement issues du traitement du signal, phénomène normal puisque toute image, qu'elle soit continue ou numérique, peut être considérée comme un signal à 2 dimensions.

La vision n'est pas un domaine facile, car repérer un objet simple dans une image demande beaucoup d'opérations.

L'objectif de ce travail est de montrer comment il est possible de développer des algorithmes de traitement d'images pour réaliser par exemple une segmentation et détection d'objets dans une image.

Le traitement, souvent appelé prétraitement, regroupe toutes les techniques visant à améliorer la qualité d'une image. De ce fait, la donnée de départ est l'image initiale et le résultat est également une image.

La représentation la plus élémentaire correspond à l'image binaire pour laquelle chaque pixel ne peut prendre qu'une valeur parmi deux autres. Pour les images monochromes, chaque pixel peut prendre une valeur parmi  $N$ .  $N$  correspond généralement à une puissance de 2, ce qui facilite la représentation de l'image en machine. Par exemple, pour une image en niveau de gris chacun des pixels peut prendre une valeur parmi 256. Sa valeur est alors codée par un octet de donnée.

Voir quelques données quantitatives liées à la représentation des images. Une image est constituée par une matrice X lignes et Y colonnes de pixels chacun codé par x bits. Les données quantitatives liées à la représentation des images sont représentées dans le tableau suivant (Table 2.1) :

1 bit	2 couleurs (noir & blanc)
4 bits	16 couleurs
8 bits	256 couleurs ou niveaux de gris
16 bits	65536 couleurs
24 bits	16777216 couleurs (vraies couleurs)
32 bits	4 294 967 296 couleurs

Table 2.1 : Données quantitatives des images

Dans notre cas l'image est considérée comme une image couleur 320\*240 où chaque couleur est codée sur 8 bits. On obtient ainsi environ 786 Ko de données nécessaires pour stocker cette image. La figure 2.2 représente une image correspondant à la représentation RGB (Red , Green Bleu) , on associe un octet à chaque pixel pour définir le niveau d'intensité du pixel la transformation d'une image en niveau de gris, on associe un octet à chaque pixel pour définir le niveau de gris du pixel (255 pour le blanc et 0 pour le noir).



Figure 2.2 : Transformation image RGB en image en niveau de gris.

## 2.2 Prétraitement ou Traitement bas niveau

Avant de présenter les différentes étapes de prétraitement, il est nécessaire de définir ce qu'est une image en l'occurrence une image bruitée et le type d'image que nous allons travailler.

### 2.2.1 Définition d'image

Une image est constituée d'un ensemble de points pixels (Picture Element). Il représente le plus petit élément constituant d'une image numérique (on parle d'image numérique lorsque les quantités physiques qui caractérisent l'image sont converties par des valeurs numériques).

L'ensemble des ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

Les axes de l'image sont orientés de la façon suivante :

- L'axe X est orienté à droite (largeur).

- L'axe Y est orienté de haut en bas (hauteur), contrairement aux notations conventionnelles en mathématiques, ou l'axe Y est orienté vers le haut.

Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case est codée par un certain nombre de débits déterminant la couleur ou l'intensité du pixel, la figure 2.3 présente un réseau de pixels (L'élément grisé encadré par la couleur rouge correspond aux coordonnées  $i, j$ ).

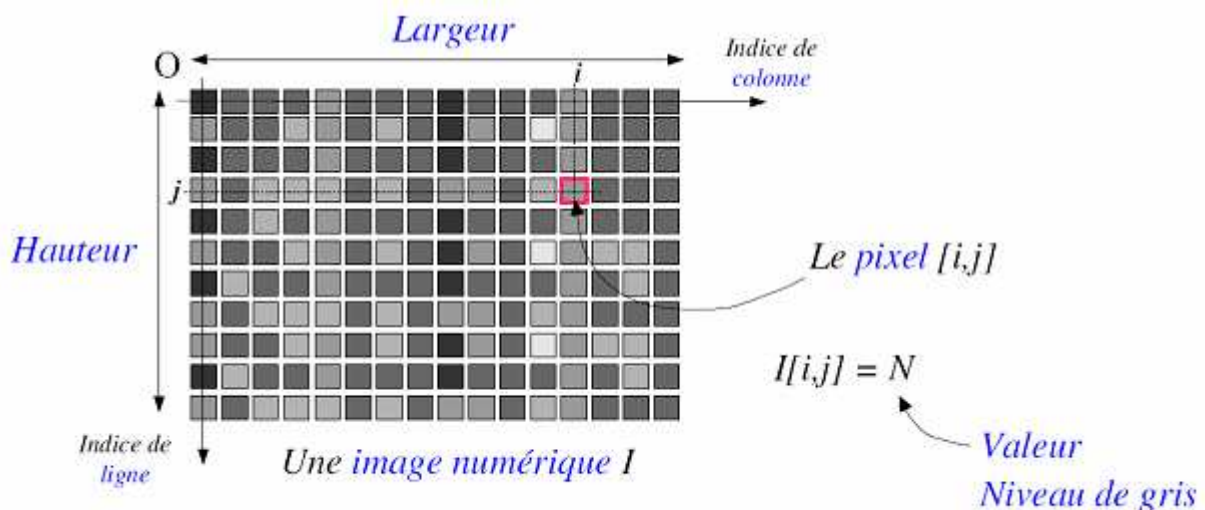


Figure 2.3 : Exemple de réseau de pixels

### 2.2.1.1 Définition Pixel

Le terme pixel est la contraction de l'anglais « Picture element ». C'est le plus petit élément de l'image. Sortie de l'appareil photo une image est composée d'un nombre  $x$  de pixels. Un pixel a une couleur exprimée (codée) en langage binaire mathématique (uniquement des 0 et des 1). On comprend donc tout de suite qu'il faut cette quantité minimum de nuances pour avoir une qualité correcte, appelée parfois qualité image.

On dit que l'image est codée en 8 bits par couche ( $2^8 = 256$  niveaux) soit 8 pour le R, 8 pour le V, 8 pour le B au total 24 bits en RVB

### 2.2.1.2 Types et caractéristiques d'images

Il existe 2 sortes d'images numériques : les images matricielles et les images vectorielles.

#### a) Image matricielle

Comme nous l'avons décrite, elle est formée d'une grille de points ou pixels, chacun pouvant avoir une couleur différente.

Une image matricielle est caractérisée notamment par sa dimension en pixels, sa résolution et son mode colorimétrique.

Les images vues sur un écran de télévision ou une photographie sont des images matricielles. On obtient également des images matricielles à l'aide d'un appareil photo numérique, d'une caméra vidéo numérique ou d'un scanner.

On représente souvent cette matrice numérique sous forme d'un tableau (une matrice notée S dans la figure 3.4). Les pixels sont ainsi numérotés en fonction de leur position (numéro de ligne, numéro de colonne)

$$S = \begin{bmatrix} s(1,1) & s(1,2) & s(1,3) & \dots & \dots \\ s(2,1) & s(2,2) & s(2,3) & \dots & \dots \\ s(3,1) & s(3,2) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & s(i,j) & \dots & \dots \end{bmatrix}$$

Figure 2.4 : Matrice S sous forme d'un tableau

Cette disposition a l'avantage de respecter les voisinages spatiaux des pixels

#### b) Image vectorielle

Elle n'est pas composée de pixels mais définie par des fonctions mathématiques qui décrivent des lignes, des courbes etc. Dans ce cas on manipule des objets et non des pixels.

Par exemple, un cercle est décrit par une fonction du type (cercle, position du centre, rayon). Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans.

Ces images présentent 2 avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

#### c) Résolution d'une image

La résolution d'une image est définie par le nombre de pixels par unité de longueur de la structure à numériser (classiquement en dpi (dots per inches) ou ppp (points par pouce)). Plus le nombre de pixels est élevé par unité de longueur de la structure à numériser, plus la quantité



d'information qui décrit cette structure est importante et plus la résolution est élevée. La résolution d'une image numérique définit le degré de détail qui va être représenté sur celle-ci. Les phénomènes de numérisation dépendent des 2 équations suivantes :

$$(X \times \text{résolution}) = x \text{ pixels}$$

$$(Y \times \text{résolution}) = y \text{ pixels}$$

Où X et Y représentent la taille géométrique (en pouces ou centimètres) de la structure à numériser, résolution représente la résolution de numérisation, puis x et y représentent la taille (en pixels) de l'image.

#### **d) Taille d'une image**

Une image de 1 pouce x 1 pouce scannée à 100 dpi aura une taille (x, y) de 100 pixels sur 100 pixels c'est-à-dire  $(1 \times 100) \times (1 \times 100) = 100 \text{ pixels sur } 100 \text{ pixels} = 10.000 \text{ pixels}$ .

**Remarque :**

$$1 \text{ pouce} = 2,54 \text{ cm}$$

$$1 \text{ pouce} = 25,4 \text{ mm} = 100 \text{ pixels}$$

$$1 \text{ inch} = 2,54 \text{ cm} = 1 \text{ pouce}$$

#### **e) Images couleur**

L'oeil humain possède des récepteurs pour les trois couleurs primaires rouge, vert et bleu. Ainsi, tous les espaces couleur ont trois dimensions. Il y a plusieurs espaces de couleurs disponibles, les plus connus étant RVB (Rouge-Vert-Bleu), HSV (Hue-Saturation-Valeur) et NTSC.

Ces images sont en général codées en utilisant le codage des trois couleurs fondamentales (rouge, vert, bleu), on parle alors d'images RVB.

Chaque couleur est codée sous forme d'un octet, d'où la composante Rouge, intensité de 0 à 255. image RVB + la composante Verte, intensité de 0 à 255 + la composante Bleue, intensité de 0 à 255, on code ainsi  $2^{24} = 16\,777\,216$  couleurs différentes.

La couleur peut aussi être codée sur un octet (256 couleurs) et l'affichage étant réalisé après passage dans une table de couleurs (transcodage).

Le codage peut ainsi être réalisé en affectant 3 bits au rouge et au vert et 2 bits au bleu (pour tenir compte par exemple de la plus faible sensibilité de la vision humaine au bleu) [24].

### e.1) L'espace RGB

L'espace RGB est vectoriel engendré par les 3 composantes primaires (Rouge, Vert, Bleu). L'ensemble des couleurs se présente comme l'intérieur d'un cube voir figure 3.5

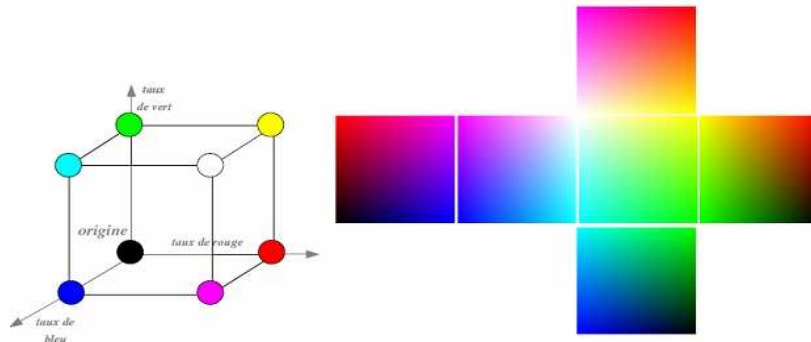


Figure2.5 : Espace RGB

Dans notre cas l'image est en RGB de 320\*240 où chaque couleur est codée sur 8 bits. Les objets sont en blanc et la couleur de l'arrière plan est spécifique au convoyeur (figure 3.6).

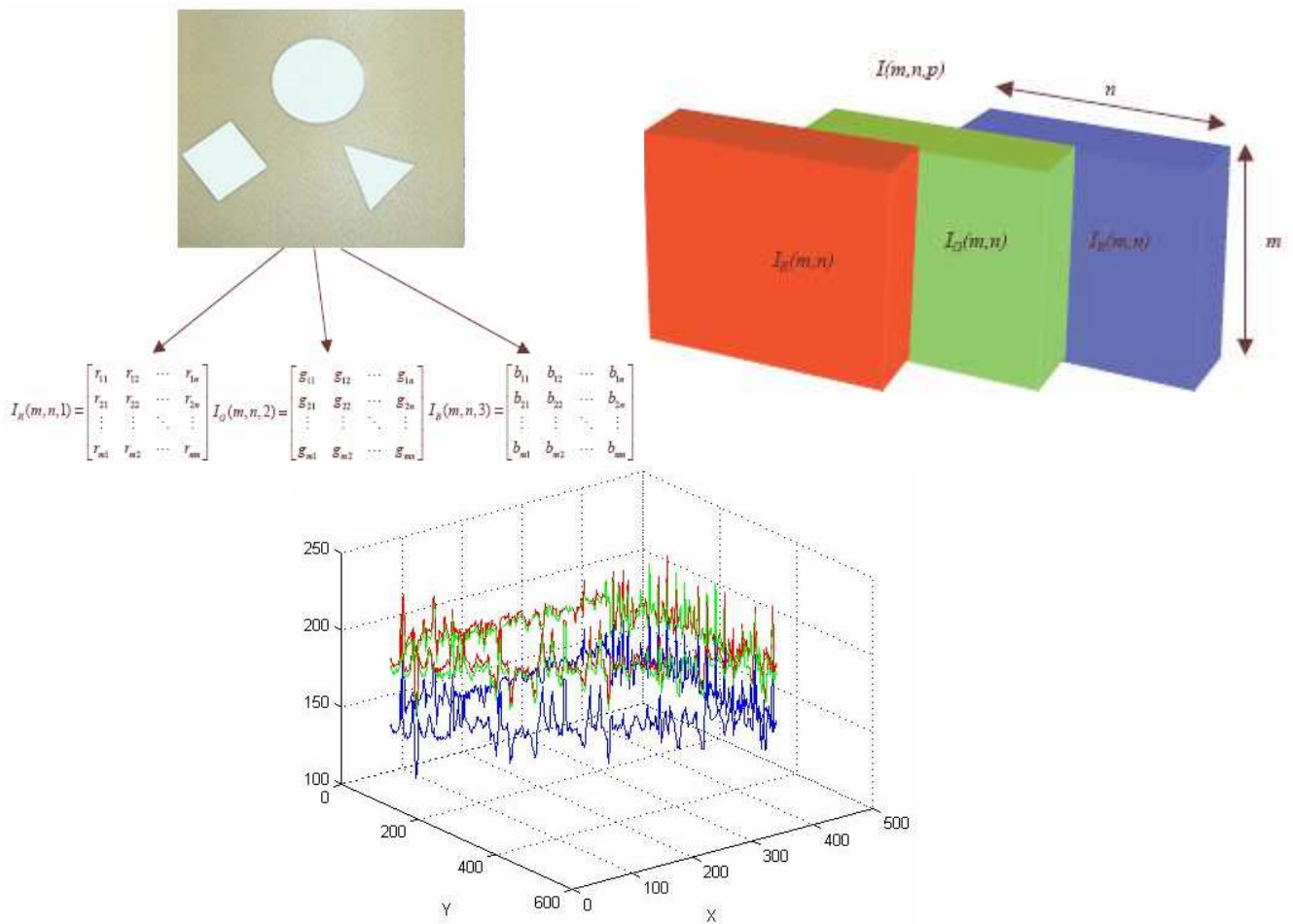


Figure 2.6 : Représentation d'image en RGB et la valeur d'intensité en long d'une ligne.

### e.2) L'espace HSV

Le principe de l'espace HSV est de caractériser les couleurs de façon plus intuitive, conformément à la perception des couleurs, en termes de : Teinte, Saturation, Valeur. (Figure 3.7)

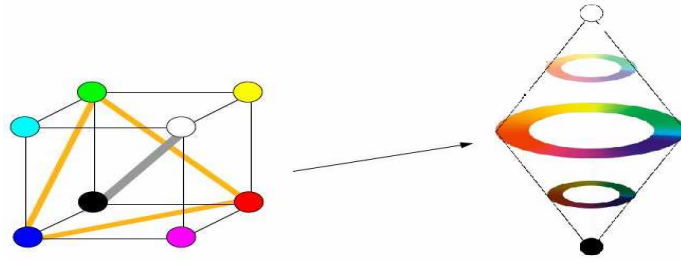


Figure 2.7 : Espace HSV

Le passage de RGB à HSV se fait par une transformation non linéaire. Plusieurs opérateurs ont été proposés pour la conversion. Nous proposons une transformation de l'image originale (RGB) en image HSV voir figure 3.8

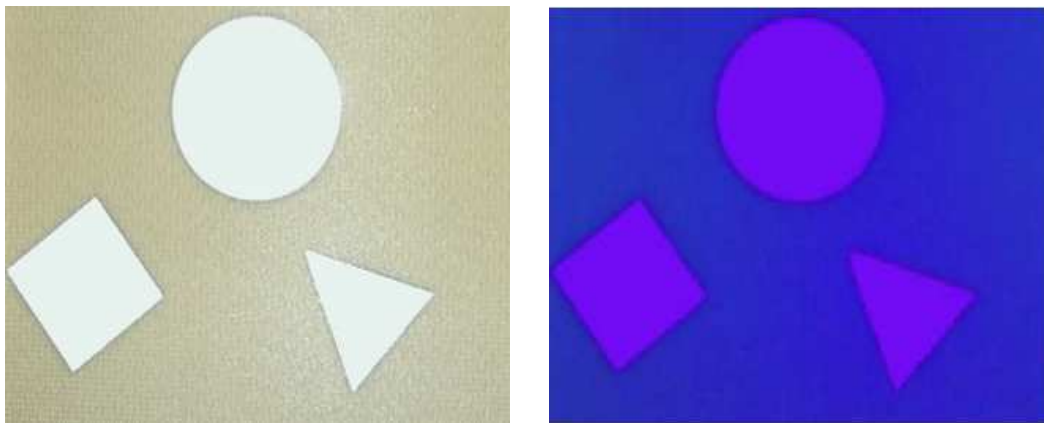


Figure 2.8 : Transformation Image RGB en HSV

### e.3) Image en niveau de gris

On passe en niveaux de gris en considérant comme seule composante la luminosité. Il existe plusieurs manières de convertir une image RGB en niveaux de gris.

La plus simple est de faire :  $gris = \frac{rouge + vert + bleu}{3}$

3

Cela équivaut aussi à affecter la couleur en niveau de gris à chacune des trois composantes RGB.

L'idéal est de faire ressortir la luminosité d'un pixel. Celle-ci vient principalement de la présence de la couleur verte. On emploie généralement les coefficients suivants :


$gris = 0,299 \cdot rouge + 0,587 \cdot vert + 0,114 \cdot bleu$

Dans ce cas on dispose d'une échelle de teintes de gris, et la plupart du temps on dispose de 256 niveaux de gris avec:

0 ----> noir, .....127 ---> gris moyen, ....., 255 ---> blanc

Certaines images peuvent être codées sur deux octets ou plus (certaines images médicales, des images astronomiques,...) ce qui peut poser des problèmes dans la mesure où les systèmes de traitement d'images courants supposent l'utilisation des pixels d'un octet.

Notons au passage que l'humain standard ne reconnaît au plus que 64 niveaux de gris

Valeurs des niveaux de gris et teintes de gris correspondantes														
0	20	40	50	80	100	120	140	160	180	200	220	240	255	
														

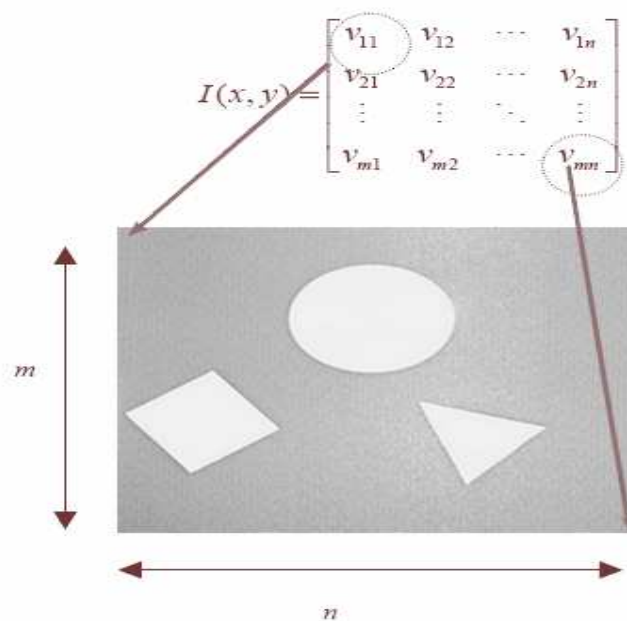


Figure 2.9: Représentation d'une image en niveaux de gris

Dans cette image, on peut régler le contraste et la distribution d'intensité en utilisant la fonction égalisation d'histogramme.

- La luminance (ou brillance) est définie comme la moyenne de tous les pixels de l'image.
- Pour augmenter la luminance, il suffit de décaler l'histogramme :

$$I'(i; j) = I(i; j) + b$$

- Dans les deux images suivantes, seule la luminance est différente voir figure 3.10

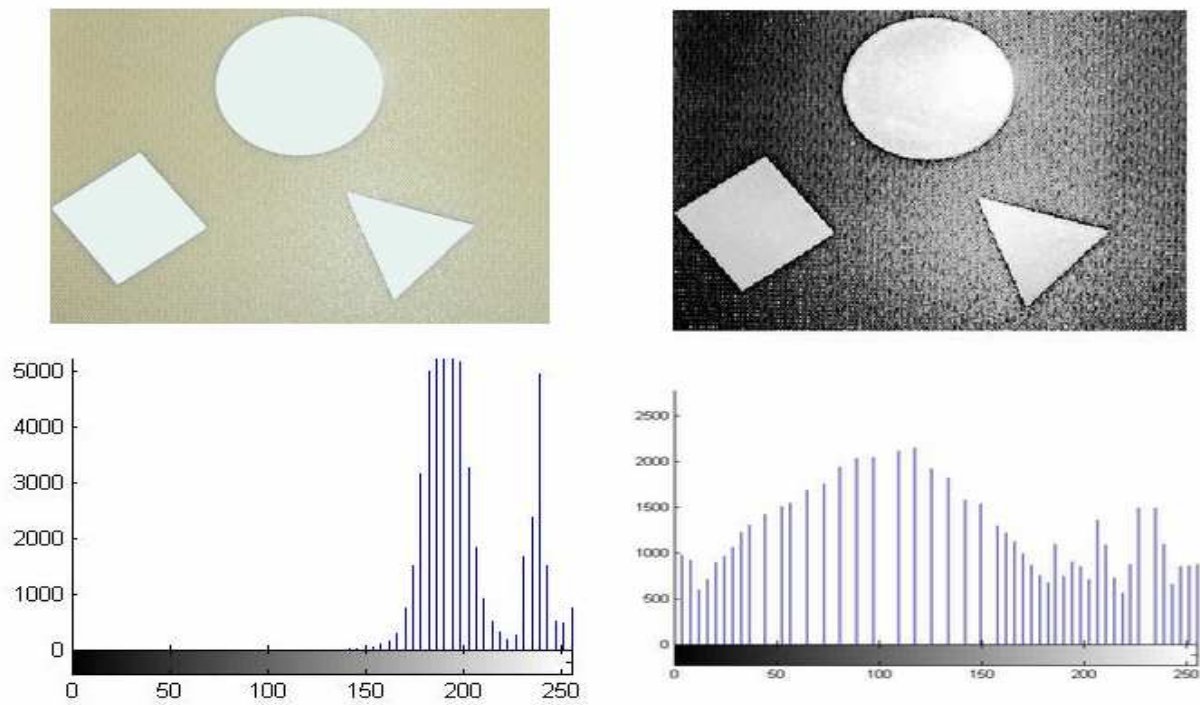


Figure 2.10 : Différence d'Histogramme Image RGB et image contrastée

- La luminance (ou brillance) est définie comme la moyenne de tous les pixels de l'image.
- Pour augmenter la luminance, il suffit de décaler l'histogramme :

$$I'(i, j) = I(i, j) + b$$

Le contraste peut être défini de plusieurs façons :

- Ecart-type des niveaux de gris (N nombre de pixels dans l'image)

$$\frac{1}{N} \sum_{n=1}^N (I(i, j) - Moy)^2$$

- Variation entre niveaux de gris max et min

$$\frac{\max[I(i, j)] - \min[I(i, j)]}{\max[I(i, j)] + \min[I(i, j)]}$$

max=intensité maximale (zone blanche)

min= intensité maximale (zone noire)

Une fois que l'histogramme d'une image est connu, les niveaux de gris de l'image peuvent être manipulés pour changer l'histogramme de la manière souhaitée. On peut par exemple désirer améliorer le contraste, changer le niveau de brillance ou faire correspondre l'histogramme à celui d'une autre image.

## 2.2.2 Voisinage dans les images

La partie que nous voulons toucher ici concerne les relations entre pixels dans les images numériques et ceci pour préparer l'étude des divers algorithmes de traitement d'images impliquant des opérations entre pixels voisins [20].

Notamment la notion de connexité entre pixels est une notion particulièrement importante pour la détection des frontières d'objets dans une image et des pixels composant un objet :

Deux pixels seront considérés comme connexes (appartenant au même objet donc) s'ils satisfont deux critères:

- Critère de similarité (par exemple même niveau de gris)
- s'ils sont adjacents (voisins)

### 2.2.2.1 Voisinage d'un pixel

La notion de connexité représente ici la notion de voisinage par contact direct. En général, on utilise deux règles de connexité:

- la connexité 4 où sont considérés connexes les quatre voisins latéraux (droite, gauche, dessus, dessous).
- la connexité 8 où sont considérés connexes les quatre voisins latéraux (droite, gauche, dessus, dessous) et les quatre voisins diagonaux (Figure 2.11).

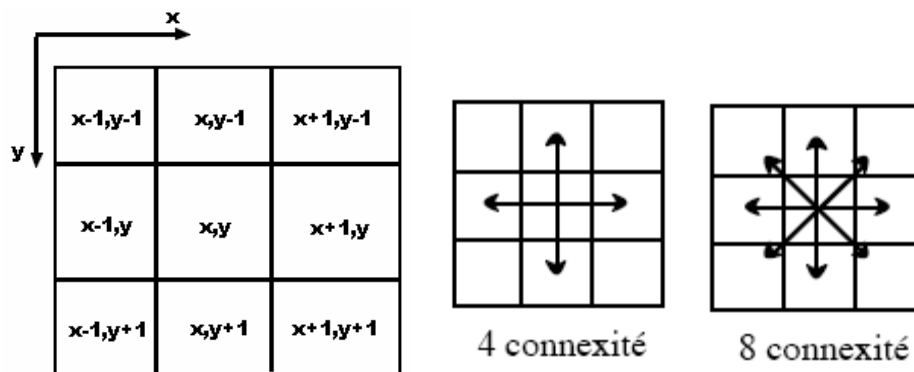


Figure 2.11 : Connexité et voisinage

### 2.2.2.2 Distances entre pixels

Tout pixel d'une image est caractérisé par un couple de coordonnées  $(x, y)$ . On peut donc calculer des distances entre pixels. Les distances les plus courantes sont (pour deux pixels  $(x_1, y_1)$  et  $(x_2, y_2)$ ):

$$\text{Distance de Manathan : } |x_1 - x_2| + |y_1 - y_2| \quad 2.1$$

$$\text{Distance Euclidienne : } \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad 2.2$$

$$\text{Distance de l'Echiquier : } \max(|x_1 - x_2|, |y_1 - y_2|) \quad 2.3$$

La distance utilisée est généralement la distance euclidienne, mais Manhattan et l'Echiquier sont plus rapides dans le calcul. Le type de distance à utiliser dépend de l'application, c'est-à-dire sur des facteurs tels que le temps d'exécution, le coût, la performance, la précision, etc.

## 2.3 Principe General sur le Seuillage et la Binarisation

### 2.3.1 Définition du seuillage

L'extraction de l'information dans l'image passe par la réduction du nombre de gris utilisé pour son codage, la limite inférieure à ce nombre est deux, ce qui conduit à produire une image binaire dans laquelle les pixels sont noirs ou blancs.

- Choix d'une valeur de seuil ( $T$ )
  - Si  $f(x,y) \geq T$  alors  $f(x,y) \in \textit{objet}$
  - Sinon  $f(x,y) \in \textit{fond}$  (*background*)

- Formation d'une image binaire  $g$

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) \geq T \\ 0 & \text{si } f(x,y) < T \end{cases}$$

Il existe deux grandes familles de seuillage:

- Le seuillage fondé sur l'analyse de l'histogramme,
- Le seuillage adaptatif, par information de contour.

#### 2.3.1.1 Le seuillage fondé sur l'analyse de l'histogramme

L'algorithme de seuillage va déterminer un seuil ou plusieurs seuils, cela va dépendre en fait de la nature de l'image et de son histogramme associé. En fait si celui-ci correspond à une image bimodale l'algorithme trouvera un seul seuil. Si celui-ci correspond à une image multimodale l'algorithme trouvera plusieurs seuils.

Prenons le cas d'un histogramme bimodal:

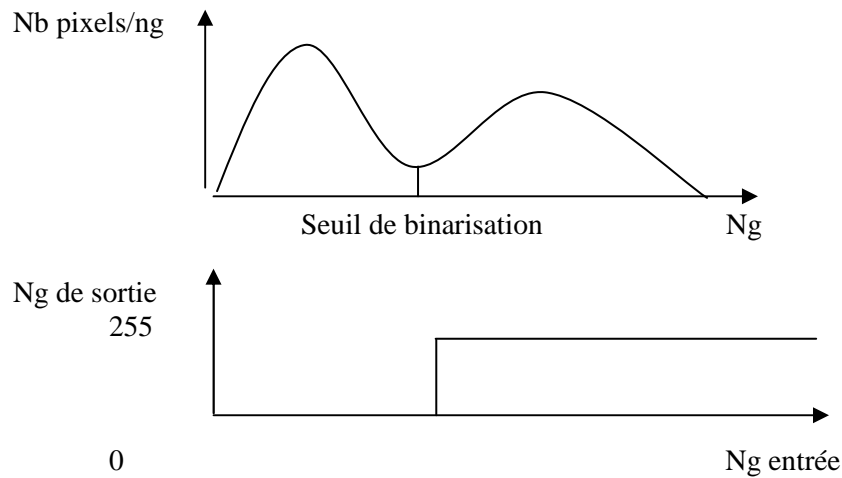


Figure 2.12 : Cas du seuillage d'un histogramme bimodal

Le cas d'un histogramme multimodal:

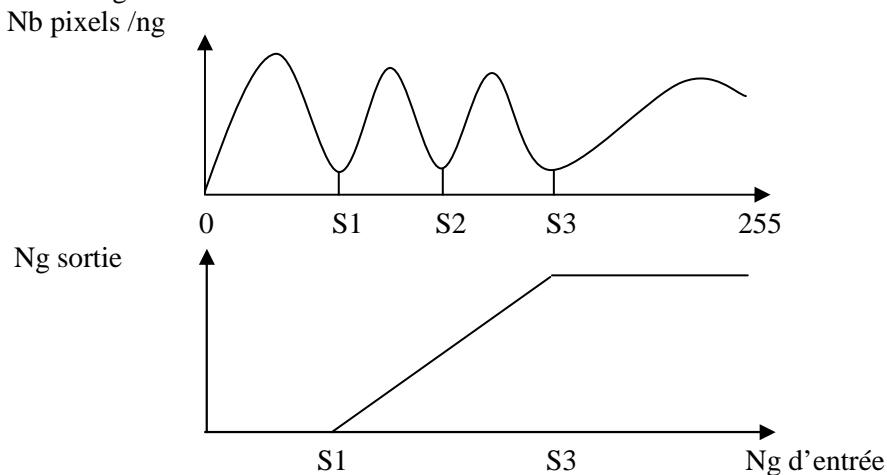


Figure 2.13 : Cas du seuillage d'un histogramme multimodal

L'histogramme d'une image est une fonction donnant, par exemple, le nombre de pixels à un niveau de gris particulier en fonction du niveau de gris.

L'histogramme ne contient aucune information relative à l'emplacement des pixels ni sur la proximité relative de deux pixels. Par contre, l'information qu'il contient peut concerner notamment la brillance apparente et le contraste d'une image, et il est utilisé en traitement d'images pour manipuler ces caractéristiques d'une image [28].

L'histogramme est calculé pour les valeurs d'intensité de l'image et analysé pour déterminer l'établissement d'un seuil permettant de séparer un objet du fond. Pour un



histogramme d'un objet assez uniforme peut être approché par une gaussienne de variance faible et l'histogramme d'une image comportant un objet sur un fond est bimodal (deux pics).

### **2.3.1.2 Seuillage adaptatif**

Le seuillage adaptatif est un type de seuillage paramétrique, c'est-à-dire qu'il tient compte des variations d'éclairements présents dans l'image. En fait ce type d'opération fait intervenir les plus proches voisins d'un pixel.

On doit écrire en fait une équation de type  $ng = f(x, y)$  pour obtenir en fait le niveau de gris d'un pixel par rapport à ses coordonnées dans l'image.

En traitement d'image on va donc utiliser un élément structurant sous forme de matrice. Le seuillage, comme nous venons de le dire, permet de produire une image binarisée, c'est-à-dire en noir et blanc. L'objet est généralement représenté par la couleur blanche et le fond par la couleur noire. L'opération inverse peut évidemment être réalisée.

La recherche automatique d'un seuil pour la binarisation est effectuée à partir de la statistique de répartition des niveaux de gris de l'image donnée par l'histogramme.

Dans certaines recherches un minimum dans l'histogramme et on y place la valeur du seuil. Dans d'autres on procède à un partitionnement progressif de la population de niveaux de gris. D'une façon générale, toutes les techniques de classification peuvent être utilisées pour résoudre ce problème.

Le seuillage est un passage obligatoire pour toute analyse morphologique. Il permet de faciliter la localisation d'objet dans l'image.

## **2.4 Binarisation -seuillage**

Le but de la binarisation d'une image est d'affecter un niveau uniforme aux pixels pertinents et d'éliminer les autres.

Le seuillage va permettre de produire une matrice image binarisée c'est-à-dire une matrice ne contenant que des valeurs égales à 0 ou 255 si on raisonne dans l'ensemble  $[0..255]$ , ou de 0 ou de 1 si on travaille dans l'ensemble  $[0..1]$ .

L'algorithme qui en résulte est très simple, il suffit d'attribuer à tous les pixels ayant un niveau de gris supérieur à la valeur seuil la couleur blanche (1 ou 255). Dans le cas contraire on leur attribue la couleur noire (0).

L'image est mise dans la matrice carrée  $X$  des valeurs de niveaux de gris pour chaque pixel

```
for n = 1:hauteur de l'image
  for m = 1:largeur de l'image
    if  $X(n,m) > ng$ 
       $X(n,m) = 256$  (attribution de la couleur blanche aux pixels)
    end
    if  $X(n,m) < ng$ 
       $X(n,m) = 1$  (attribution de la couleur noire aux pixels)
    end
  end
end
```

L'algorithme permet de binariser une image initiale en niveaux de gris, cette technique permet une segmentation d'objets particuliers de l'image.

Le schéma général représentant le seuil de binarisation (Figure 2.14 )

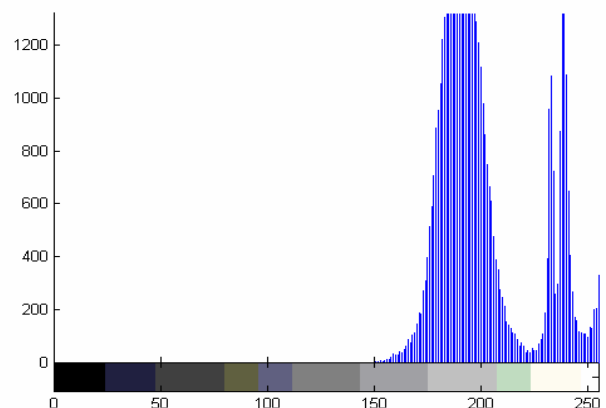
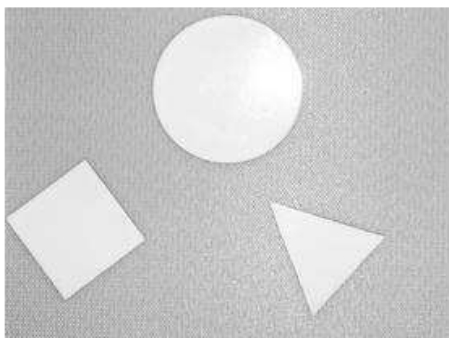
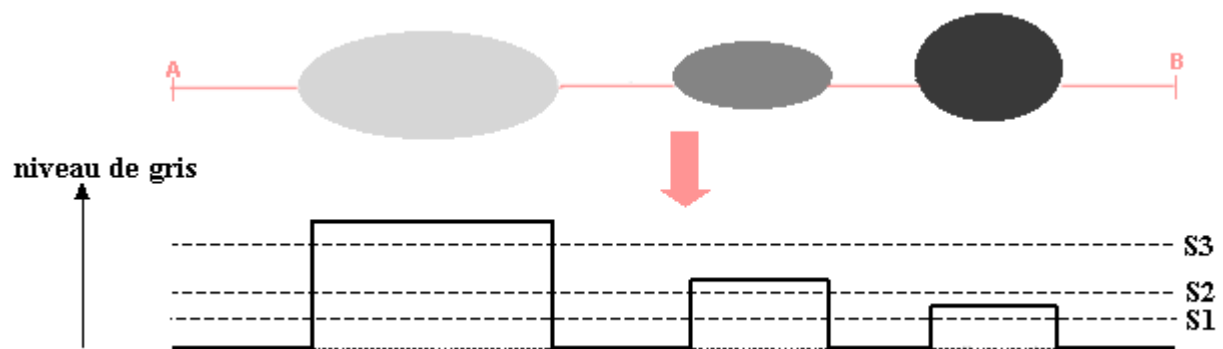


Figure 2.14 : Seuil de binarisation

Le seuil de binarisation peut alors être choisi de différentes façons. Dans le cas de notre image, on remarque que l'histogramme présente deux pics de population de pixels autour des niveaux de gris 210 et 190. On peut par exemple choisir la valeur de niveau de gris médiane entre ces deux pics. Néanmoins, il existe de nombreuses autres manières d'effectuer le seuillage : valeur moyenne de l'intensité sur l'ensemble de l'histogramme, valeur médiane de l'intervalle des niveaux de gris  $[0, 255]$ , qu'on va voir par la suite

Prenons par exemple, l'image en niveaux de gris suivante, et considérons un profil AB la traversant:



Elle contient 3 objets : un noir, un gris foncé et un gris clair.

Le seuil va correspondre à une valeur de niveau de gris susceptible de partitionner l'image en 2 composantes : les objets d'une part, le fond d'autre part. Le résultat est une simplification de l'image en une image binaire. Après seuillage, les parties de l'image qui auront été sélectionnées seront traduites en blanc et le reste de l'image en noir (codé à 0). On a donc le schéma suivant :

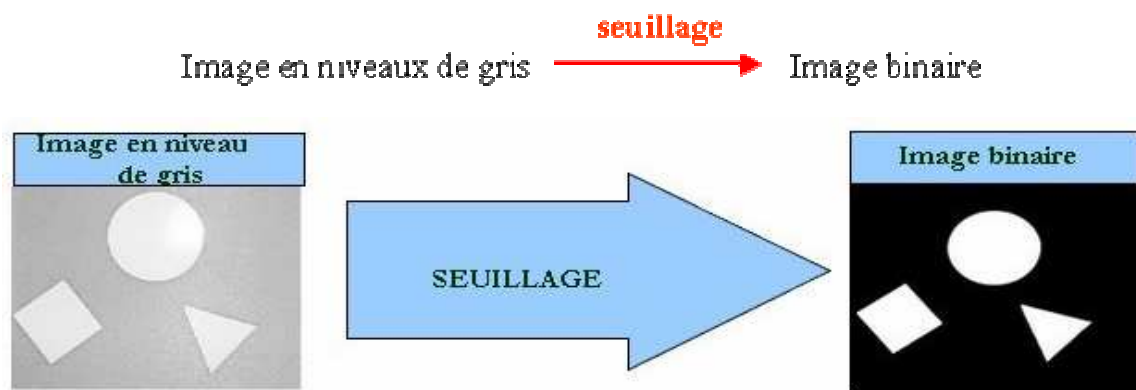


Figure 2.15 : Transformation d'image en niveau de gris en image binaire

Il est également possible de nettoyer l'image lors de la numérisation par des concepts de filtrage dans le but de faciliter une analyse ultérieure, ces filtres possèdent des caractéristiques spectrales; nous évoquerons dans cette partie en détail le filtrage.

### 2.4.1 Le Filtrage

Le Filtrage des images a la même finalité que celui des signaux 1D. Il s'agit essentiellement de nettoyer l'image lors de la numérisation par des concepts de filtrage.

Notamment de bruit (par exemple lors d'une mauvaise réception de données), de flou (dûs à une mauvaise mise au point), ou encore des pertes de qualité (dues à une mauvaise luminosité).

Dans une image, les détails se répètent fréquemment sur un petit nombre de pixels, on dit qu'ils ont une fréquence élevée : c'est le cas pour les textures fines et certains contours de l'image. Au

contraire, les fréquences basses correspondent à des faibles variations étendues sur de grandes parties de l'image, par exemple des variations de fond de ciel.

Différentes méthodes de filtrage ont été développées suivant le type de l'intensité du bruit.

Ces filtres possèdent des caractéristiques spectrales; on parle ainsi des filtres linéaires et des filtres non linéaires.

#### 2.4.1.1 Filtrage linéaire

L'information contenue dans l'image peut être représentée de différentes façons afin de mieux mettre en évidence certaines propriétés des images.

Le filtrage linéaire correspond à la première opération utilisant le proche voisinage qui vient à l'esprit. Il consiste, en effet à remplacer la valeur d'un pixel par une combinaison linéaire des valeurs des pixels voisins.

Pour une atténuation d'un bruit qui se traduit par des variations locales aléatoires de luminance, un filtre dit moyenneur ou passe –bas sera souvent utilisé. Cette atténuation peut être effectuée à l'aide des deux méthodes de filtrage soit par filtrage dans le domaine fréquentiel ou filtrage dans le domaine spatial (convolution) [21]. Certaines théories neurophysiologiques ont également fait évoluer le traitement des images vers des méthodes combinant les deux domaines. En représentation fréquentielle, on utilise notamment la transformée de Fourier.

##### -Réduction de bruit avec un filtre linéaire

On peut jouer sur l'intensité du filtre et on peut remarquer que l'image est plus nette lorsque la valeur de l'intensité est plus importante. Le résultat est visualisé sur la figure 2.16

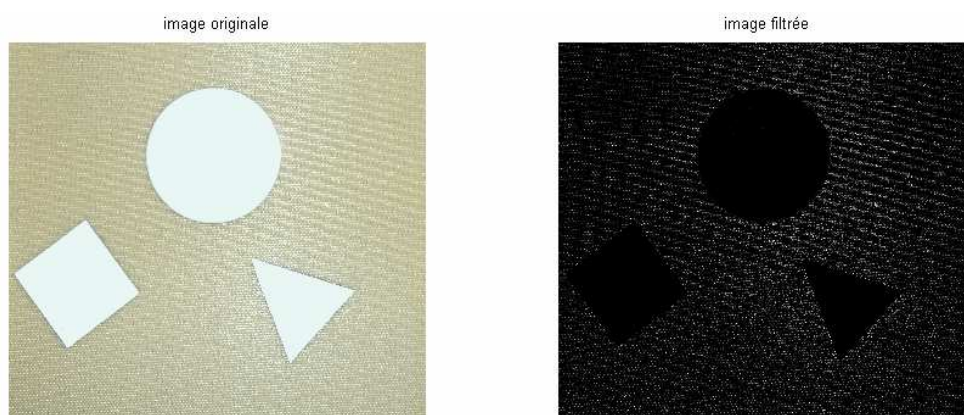


Figure 2.16 : Filtrage linéaire

## a) Filtrage dans le domaine fréquentiel

Il y a plusieurs manières de transformer une image (signal 2D) en d'autres représentations, plus adéquates pour certains types d'analyse d'images. Lorsque l'on considère un signal, il est indispensable d'avoir présent à l'esprit deux représentations possibles de ce signal [22]. C'est-à-dire le temps et fréquence : dans la représentation continue en temps du signal  $y(t)$  ou discret  $y(n)$  la variable  $t$  ou  $n$  est la durée qui s'écoule continûment ou par sauts discrets. La représentation fréquence du signal fait jouer un rôle central aux fréquences pures qui constitueront la base de description des signaux. La transformée de Fourier est l'outil le plus communément utilisé pour réaliser ce type de transformations et d'analyse des données.

La transformée de Fourier (TF) permet la décomposition d'un signal  $f$  (ici notre image) en combinaison linéaire de sinusoides complexes, dont les coefficients  $F[u,v]$  dit coefficients de Fourier, fournissent des informations sur les fréquences  $(u,v)$  et permettent des manipulations dans le domaine fréquentiel. L'image obtenue n'est plus représentée par la valeur de ses pixels en fonction de leur position dans l'espace, mais par l'amplitude et la phase de sinusoides en fonction de leur fréquence [21].

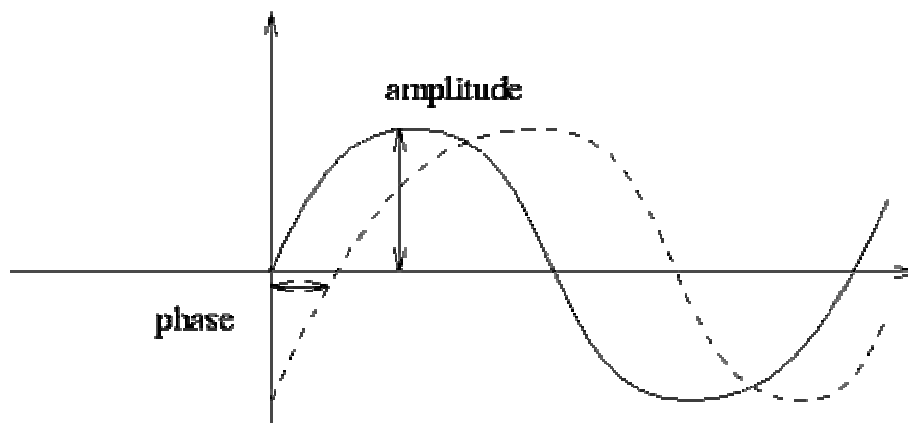
Soit un signal continu  $f(x)$ , vérifiant la condition d'existence d'une énergie finie suivante :

$$\int_{-\infty}^{+\infty} |f(x)|^2 dx$$

La paire de la transformée de Fourier est définie par :

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi ux} dx \quad \text{Son inverse est : } f(x) = \int_{-\infty}^{+\infty} F(u) e^{i2\pi ux} du$$

Amplitude et angle de phase d'une sinusoïde à une fréquence particulière :



La norme de l'amplitude est appelée spectre de Fourier de la fonction (ou module), et l'exposant est appelé angle de phase. Le carré de l'amplitude est simplement appelé puissance spectrale de la fonction.

Les intégrales des formules précédentes seront remplacées par des sommes partielles. On obtient alors la paire de transformées discrètes suivante :

$$F(u, v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) e^{\frac{-i2\pi}{N}(uj+vk)} \quad 2.4$$

$$|F(u, v)| = \sqrt{F_{re}(u, v)^2 + F_{im}(u, v)^2} \quad 2.5$$

La fonction  $|F(u, v)|^2$  est connue sous le nom de spectre de puissance.

Le spectre de phase associé aux fréquences est défini de la manière suivante :

$$\phi(u, v) = \tan^{-1} \left( \frac{F_{im}(u, v)}{F_{re}(u, v)} \right) \quad 2.6$$

En général, la transformation d'un signal réel dans le domaine fréquentiel est évaluée comme une fonction complexe, c'est-à-dire qui a un module et une phase.

Un filtre passe-bas est un filtre qui laisse passer les basses fréquences et qui atténue les hautes fréquences, c'est-à-dire les fréquences supérieures à la fréquence de coupure. [23].

D'autant plus que la transformée de Fourier a l'avantage d'être très facilement inversible, de l'image on passe facilement aux fréquences, et des fréquences on passe aisément à l'image. Ainsi, on peut (comme avec un gaussien) supprimer les hautes fréquences de l'image en créant ce qu'on appelle un filtre passe-bas :

1. On passe dans l'espace de Fourier (espace fréquentiel)
2. On supprime les données relatives aux hautes fréquences
3. On inverse la transformée de Fourier pour revenir à l'image (mais cette fois les hautes fréquences ont disparu).

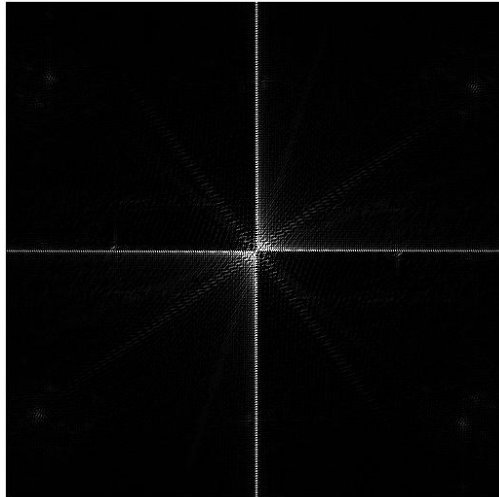


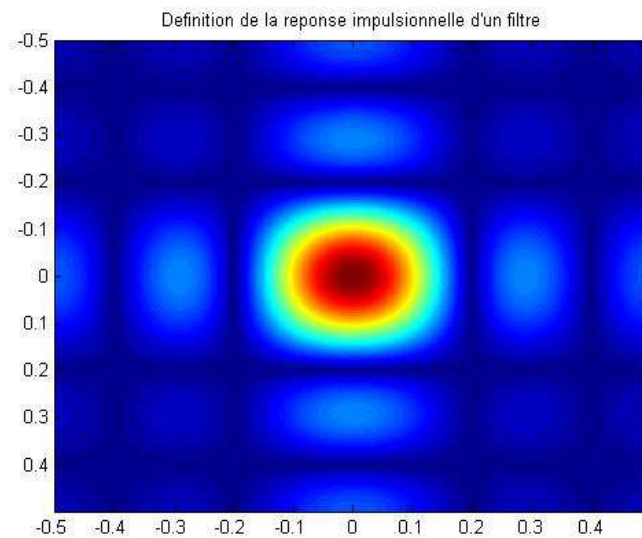
Figure 2.17 : Filtrage fréquentiel

Le filtre passe-bas est défini comme suit :

$$\begin{cases} F_{pb}(u, v) = F(u, v) & \text{si } u \leq u_m \text{ et } v \leq v_m \\ F_{pb}(u, v) = 0 & \text{sinon} \end{cases} \quad 2.7$$

Le calcul de la fonction de transfert se fait de la manière suivante :

On calcule la transformée de Fourier puis on définit les vecteurs de fréquences normalisées et on affiche le module de la FFT.



Le module du spectre avant et après filtrage est représenté dans la figure 2.18.

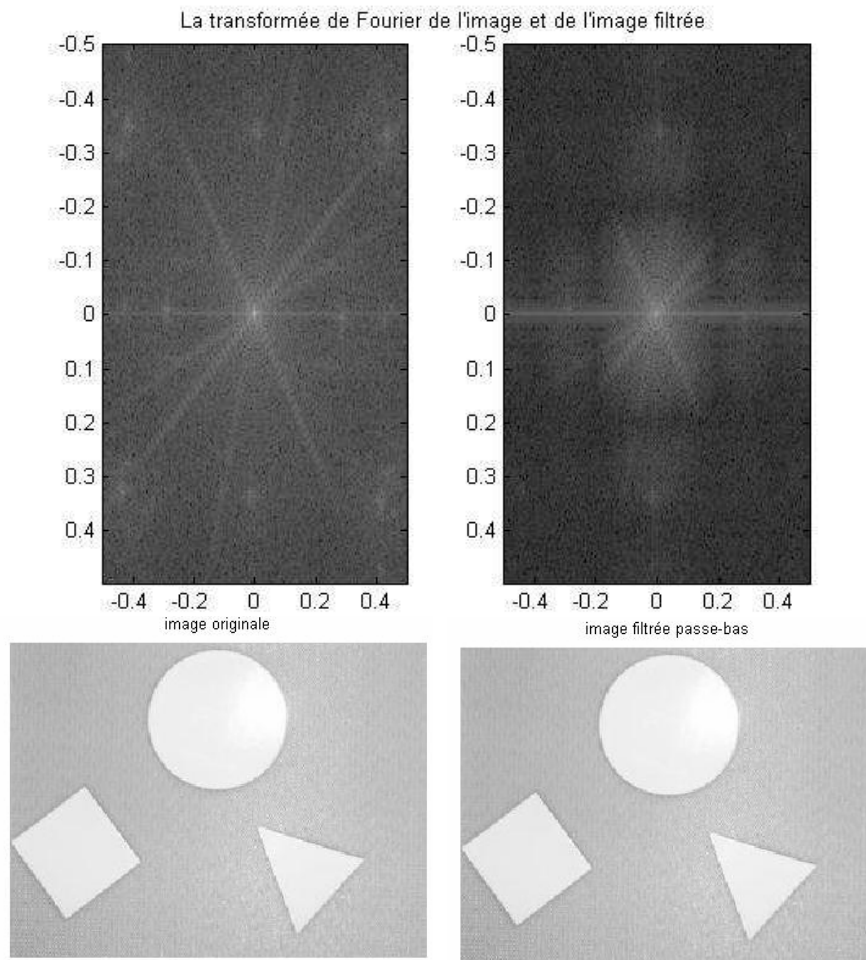


Figure 2.18 : La Transformation de Fourier de l'image originale et de l'image filtrée passe bas

L'image à gauche représente le module du spectre avant filtrage. On aperçoit une zone claire au centre du spectre, qui correspond à la composante continue de l'image. Le nuage de points autour de cette composante continue correspond à quelques composantes moyennes et hautes fréquences de l'image (contours, détails, zones de transition, ...).

L'image de droite représente le spectre après filtrage passe-bas. La composante continue qui est basse fréquence est bien conservée. Le nuage de points représentant les moyennes et hautes fréquences a été globalement supprimé. Cependant, selon les axes des fréquences spatiales horizontales et verticales pures, des gains importants apparaissent pour les moyennes et hautes fréquences.

Dans la mesure où un bruit est généralement un signal haute fréquence, filtrer un signal bruité en ne gardant que les hautes fréquences revient à garder seulement le bruit



## b) Filtrage dans le domaine spatial (par convolution)

Une image est avant tout un signal, elle possède des hautes fréquences et des basses fréquences. Dans une image une basse fréquence est caractérisée par une région homogène, un flou, tandis qu'une haute fréquence est caractérisée par un contour, un changement brusque d'intensité, un bruit. La plus grande partie de l'énergie d'une image se situe dans les basses fréquences.

Le filtrage linéaire spatial est essentiellement une opération de convolution (2D). Si  $f$  est l'image à filtrer et  $g$  le filtre spatial on a :

$$f(x, y) * k(x, y) = F^{-1} \left\{ F(f(x, y)) \cdot \underbrace{F(g(x, y))}_{k(u, v)} \right\}$$

ou  $F$  est la transformation de Fourier et  $K$  la fonction de transfert du filtre. Dans le contexte du traitement d'image le noyau  $K$  est appelé masque.

Trois types de filtrage

1. Le filtre Passe-bas diminue le bruit mais atténue les détails de l'image (flou plus prononcé)
2. Le filtre Passe-haut accentue les contours et les détails de l'image mais amplifie le bruit
3. Le filtre Passe-bande élimine certaines fréquences indésirables présentes dans l'image.

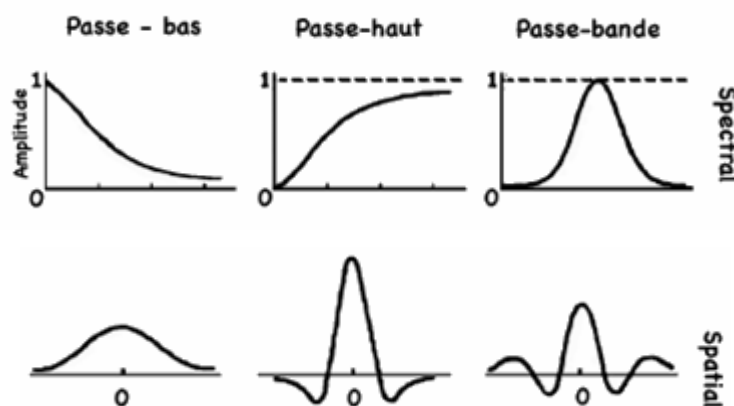


Figure 2.19 : Différent types de filtrage

Une image numérique étant essentiellement discrète (pixels et niveaux de gris) nous allons présenter les filtres dans le cas discret. Dont  $i$  et  $j$  sont des entiers (coordonnées des pixels) et  $f$  est à valeurs entières (dans  $\{0, \dots, 255\}$ ). On ne fait pas en général une convolution globale mais une transformation locale, basée sur le voisinage d'un point  $i, j$  : voir figure 2.20

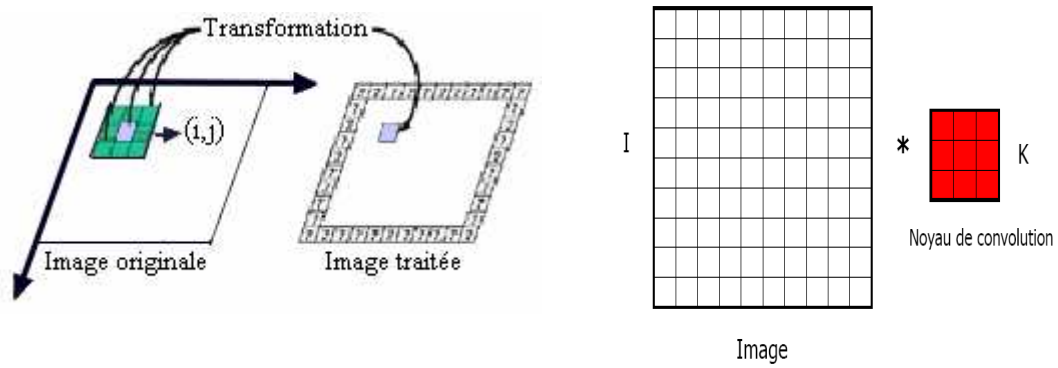


Figure 2.20 : Transformation d'un point  $i, j$

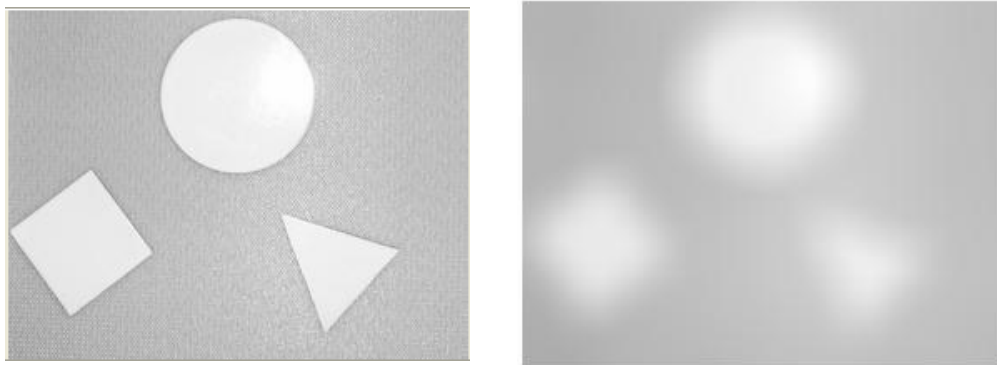


Figure 2.21 : exemple de convolution Locale de l'image

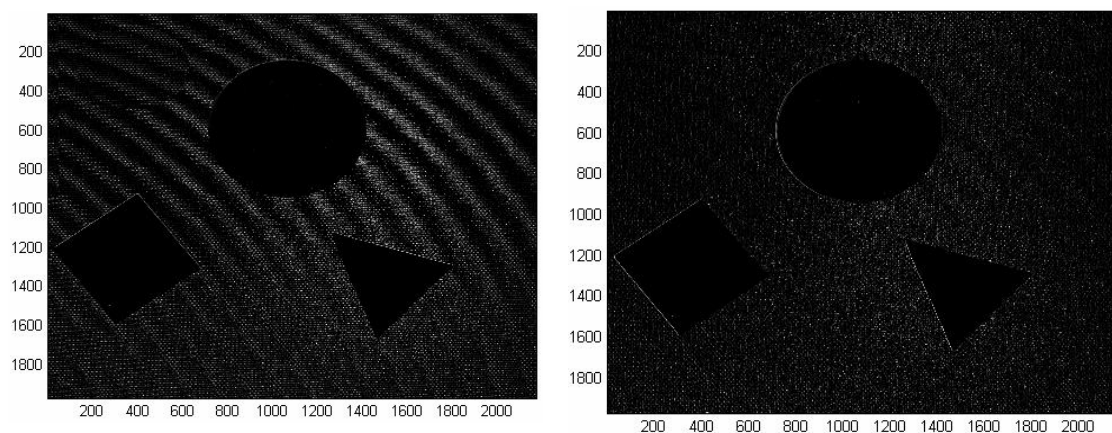


Figure 2.22 Représentation du filtrage spatial

## -Convolution

Transformation locale : utilisation du voisinage de chaque pixel.

Un filtre de convolution (ou masque ou noyau) est généralement une matrice  $n \times n$ ,  $n$  impair.

Pour éviter de modifier la luminance de l'image, la somme des coefficients du filtre doit être égale à 1.

Le filtrage peut être effectué dans l'espace image (domaine spatial) en utilisant l'opérateur de convolution, bien connu dans le traitement signal.

Le produit de convolution d'un signal 2D  $f(i, j)$  (une image) avec un filtre  $h(i, j)$  est donné par :

$$f'(i, j) = (f * h)(i, j) = \sum_{n=i-l}^{i+1} \sum_{m=j-l}^{j+1} f(i, j) h(n-i, m-j) \quad 2.8$$

Exemple : si  $h(n, m)$  est un masque carré de taille  $d=3$  impair :

$$\begin{aligned} f'(i, j) = & h(1,1)f(i-1, j-1) + h(1,0)f(i-1, j) + h(1,-1)f(i-1, j+1) + h(0,1)f(i, j-1) \\ & + h(0,0)f(i, j) + h(0,-1)f(i, j+1) + h(-1,1)f(i+1, j-1) + h(-1,0)f(i+1, j) \\ & + h(-1,-1)f(i+1, j+1) \end{aligned} \quad 2.9$$

## -Principe de calcul de la convolution pour un filtre.

Soit un filtre RIF de réponse impulsionnelle  $h(n, m)$  au pixel  $p = f(i, j)$  :

1- Faire une rotation de  $\pi$  du noyau par rapport à son centre :

$$h(n, m) \Rightarrow h(-n, -m) = g(n, m)$$

2- Centrer le filtre sur  $p$  en le superposant à l'image

3- Effectuer la somme pondérée entre les pixels de l'image et les coefficients du filtre  $g(n, m)$

4- Le pixel  $p$  dans l'image but (filtrée) aura comme valeur cette somme pondérée

Cas avec un filtre de taille  $d = 3$

$$h = \begin{pmatrix} w_9 & w_8 & w_7 \\ w_6 & w_5 & w_4 \\ w_3 & w_2 & w_1 \end{pmatrix} \quad g = \begin{pmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{pmatrix}$$

La convolution au pixel (i ; j) de f par le noyaux h est donnée par :

$$\begin{aligned}
 f'(i, j) = & w_1f(i-1, j+1) + w_2f(i-1, j) + w_3f(i-1, j-1) \\
 & + w_4f(i, j-1) + w_5f(i, j) + w_6f(i, j+1) + w_7f(i+1, j-1) \\
 & + w_8f(i+1, j) + w_9f(i+1, j+1)
 \end{aligned}
 \tag{2.10}$$

Pour conserver la moyenne originale de  $f$  dans  $f'$ , on normalise les coefficients du filtre, on a donc :

$$\sum_{i=1}^t w_i = 1 \text{ où } t = d^2 \text{ est le nombre de coefficients du filtre .}$$

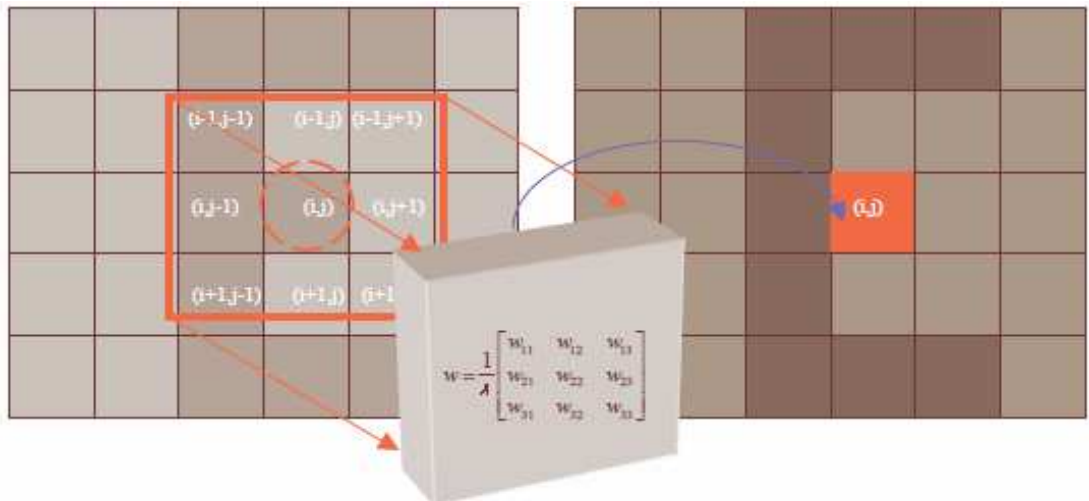


Figure 2.23 : Filtre spatial pour un masque 3x3

La multiplication dans le domaine fréquentiel correspond à une convolution dans le domaine spatial. Un grand nombre de filtres peut être obtenu à partir de noyaux de convolution symétriques et normalisés (de somme égale à 1). Quelques méthodes de filtres les plus utilisés.

### -Filtre moyennneur

Les filtres moyennneurs sont un des types passe-bas dans le principe est de calculer la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue. Exemple d'algorithme du filtre moyennneur.

Algorithme :

```

Pour i allant de 1 à largeur_image faire
  Pour j allant de 1 à hauteur_image faire
    Som = 0
    Pour k allant de -1 à 1 faire
      Pour l allant de -1 à 1 faire
        Som = Som + Image1(i+k, j+l)
      Fin
    Image2(i, j) = Som/9
  Fin
Fin
  
```

Le filtre moyennneur remplace la valeur d'un pixel par la moyenne de sa valeur et de la valeur des 8 pixels qui l'entourent. La matrice est donc une 3x3 dont chacun des membres vaut 1/9.

**Réponse impulsionnelle :**

$$h(i, j) = \begin{cases} \frac{1}{\lambda^2} & \text{si } |i, j| \leq \frac{\lambda}{2} \\ 0 & \text{Autrement} \end{cases}$$

Le filtre de taille d = 3 :

$$h = 1/9 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

-D'une manière générale, si on a un filtre de taille d, tous les coefficients du filtre ont comme valeur  $w_i = 1/d^2$

- Plus d est grand, plus le lissage sera important, et plus l'image filtrée perd les détails de l'image originale.

Le filtre moyennneur réalise en effet la moyenne pour chaque pixel sur un voisinage (3 x3).voir figure 2.23

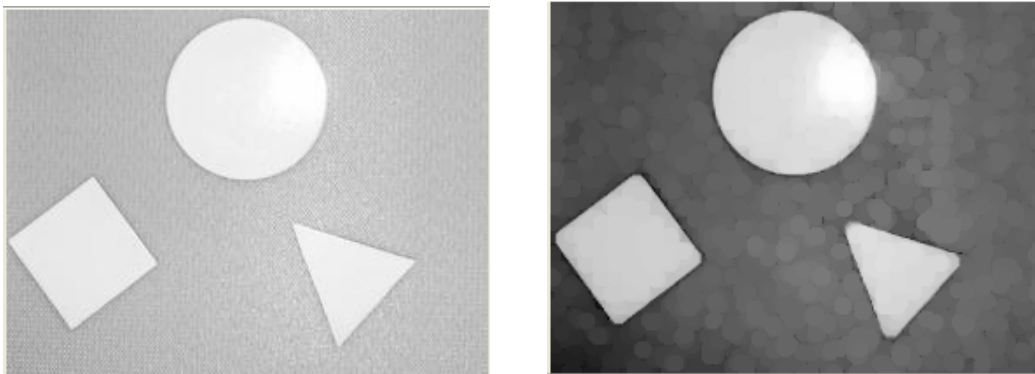


Figure 2.24 : Filtre moyennneur

Il existe un autre type de filtrage qui permet d'atténuer le bruit sur une image, c'est le filtre gaussien.

### **-Filtre Gaussien**

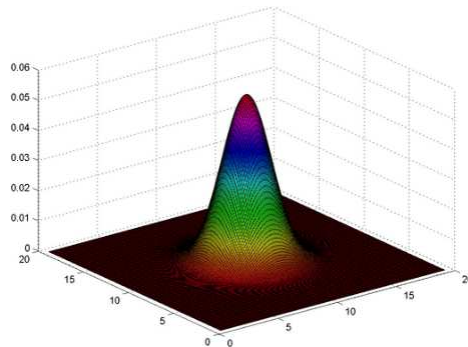
L'intérêt du filtre gaussien est que l'on règle très facilement le degré de filtrage à travers le paramètre  $\sigma$ . On induit une forme de flou qui a pour avantage de retirer les petits défauts. D'un

point de vue "signal", ces petits défauts correspondent à des variations importantes d'intensités sur une petite longueur. Si on parlait d'un signal, on dirait que ce sont des "hautes fréquences".

Le paramètre sigma s'appelle la déviation standard, et détermine la largeur de la cloche Gaussienne.

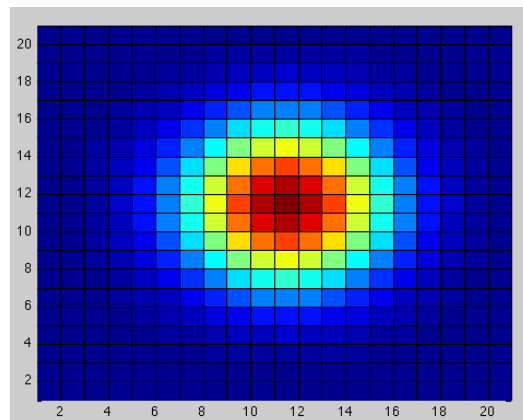
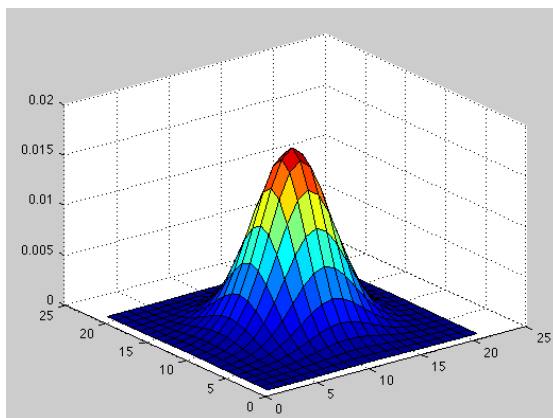
Par rapport au filtre moyenneur, le filtre gaussien accorde une grande importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci. Même effet de flou que le filtre moyenneur [29].

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



### -Du continu au discret

Le noyau gaussien est défini par un ensemble de coefficients qui sont des échantillons de la Gaussienne 2D



Etant donné que les images numériques sont composées de valeurs discrètes, le noyau gaussien sera composé aussi de valeurs discrètes:

Il faut noter que plus le sigma est grand, plus la cloche Gaussienne est large et plus le flou appliqué à l'image sera marqué.

Pour calculer un flou gaussien, il existe deux principes :

- produit de convolution avec un noyau gaussien
- passage dans l'espace de Fourier

Dans le premier cas : Si l'on veut un flou qui prenne en compte un grand nombre de pixel autour du pixel de destination, il est nécessaire d'augmenter le rayon du noyau (dans le cas d'un produit de convolution standard avec un noyau gaussien). On applique le filtre de bruit gaussien dans la figure 2.24

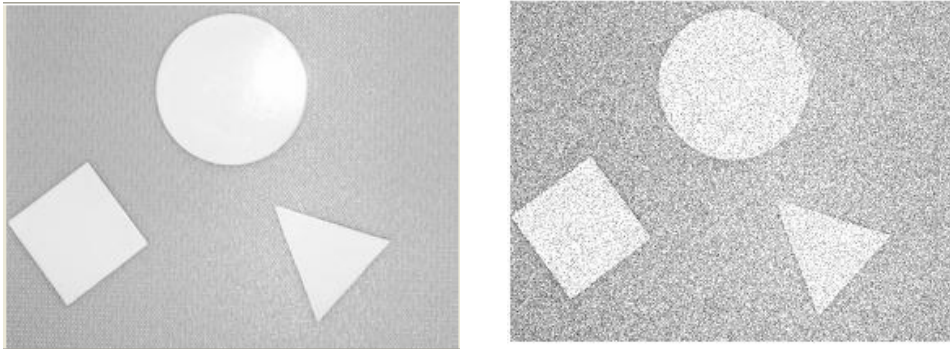


Figure 2.25 : Filtre de bruit gaussien

L'avantage de passer dans l'espace de Fourier, c'est que l'on peut appliquer un flou gaussien de n'importe quel rayon (dans ce cas, ce n'est plus vraiment un rayon) pour un même temps d'exécution.

Pour l'appliquer, nous avons un noyau de convolution  $G$  et une image  $I$ , et  $G * I$  le produit de convolution entre  $G$  et  $I$ .

Alors, si on note  $F$  la transformée de Fourier, on a la propriété suivante :

$F(G * I) = F(G).F(I)$  Le produit de convolution (opération lente) se transforme en produit simple (opération rapide). Pour un noyau gaussien, l'avantage étant que  $F(G)$  est une formule connue.

Le filtre gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyenneur, il préserve mieux l'information contenue dans l'image que les moyenneurs. On considère souvent que le filtre gaussien est le meilleur compromis entre lissage et préservation de l'information, ce qui implique pour un masque de convolution très étendu et donc une charge de calcul enlevée. C'est pourquoi dans la pratique cette solution est rarement retenue et se contente d'un masque peu étendu.

On constate simplement, sur l'exemple de filtrage de notre image que les contours des objets sont réduits, le contraste après filtrage est moins bon.

Ce défaut est inhérent au principe de filtrage linéaire et on ne peut y remédier qu'en changeant de méthode de traitement. Une solution consiste à autoriser des opérations non-linéaires entre les voisins, c'est ce qu'on appelle le filtrage non-linéaire.

### **2.4.1.2 Filtrage non linéaire**

#### **- Introduction**

Contrairement au filtrage par convolution (filtrage linéaire), le filtrage non-linéaire fait intervenir les pixels voisins suivant une loi non-linéaire. Il existe ainsi une infinité de filtres non-linéaires, mais deux grandes catégories sont parmi les plus utilisées en vision artificielle ; celle des filtres d'ordre et celle qui regroupe les opérations morphologie mathématique.

Nous décrirons en détail cette partie qui est les filtres d'ordre et ensuite la catégorie morphologie mathématique.

Les filtres non-linéaires opèrent sur un voisinage donné. Lorsqu'un masque de convolution est lui-même fonction des pixels de l'image, le filtrage devient non-linéaire.

Seuls des filtres non-linéaires peuvent permettre de préserver des contours en s'adaptant au contenu des images. Le plus connu dans le filtrage non-linéaire est le filtre médian (cas particulier du filtrage d'ordre).

#### **- Filtre médian**

Ce sont des filtres non linéaires et donc ce ne sont pas des filtres de convolution [25].

Ces filtres sont utiles pour contrer l'effet " Poivre et Sel " (P& S) c'est à dire des faux « 0 » et « 255 » dans l'image.

Le bruit « poivre-et-sel » utilisé ici consiste à mettre, aléatoirement, plusieurs pixels aux valeurs 255 ou 0 (valeurs extrêmes de l'intervalle des niveaux de gris). Ce type de bruit impulsionnel peut apparaître par numérisation d'une image ou au cours d'une transmission.



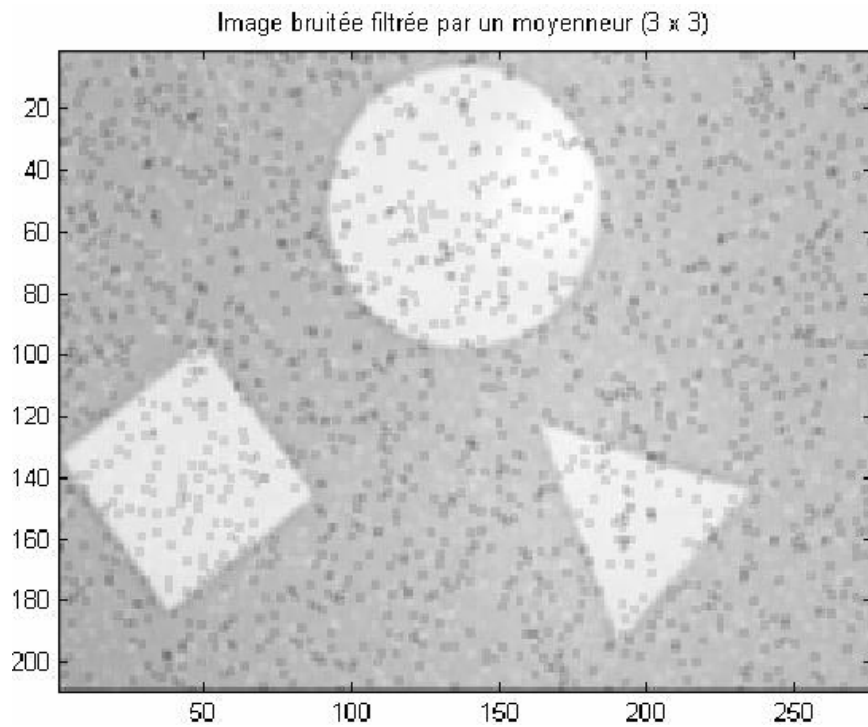


Figure 2.26 : image bruitée par un moyeneur 3 x 3

Le bruit « salt-and-pepper » est faiblement réduit. On distingue encore nettement les grains dans l'image. Le filtre moyeneur réalise en effet la moyenne pour chaque pixel sur un voisinage (3 x3).

En effet un filtre de convolution affecte au pixel traité un barycentre des valeurs des niveaux de gris des pixels situés dans un voisinage. Un pixel dont le niveau de gris est très différent des autres va donc affecter le résultat de la convolution. On préfère, dans ce cas remplacer la valeur du pixel par la valeur médiane et non la valeur moyenne . L'algorithme suivant écrit les étapes de calcul du filtre médian.

Algorithme :

```

Pour i allant de 1 à largeur_image faire
  Pour j allant de 1 à hauteur_image faire
    Index = 0
    Pour k allant de -1 à 1 faire
      Pour l allant de -1 à 1 faire
        Tab[index] = Image(i+k, j+l)
        index=index+1
    Fin
    Ranger les valeurs
    P = tab[4]
    image2(i, j) = P

```

Fin

On remplace la valeur du pixel par la valeur médiane ou la valeur moyenne. Ce filtre est utile pour contrer l'effet « Poivre et Sel » (P& S) c'est à dire des faux « 0 » et « 255 » dans l'image.

7	5	3
4	2	7
6	5	2

Le filtre médian range par ordre croissant les intensités du voisinage :

2 2 3 4 5 5 6 7 7

Il affecte au pixel central l'intensité qui se trouve au milieu du rangement ci-dessus.

La méthode basée sur le filtre médian est particulièrement adaptée lorsque le bruit est constitué de points isolés ou de lignes fines. Cependant, elle n'est applicable qu'aux images en niveaux de gris, contrairement au lissage. La réduction de bruit avec le filtre médian est représentée par la figure suivante 2.26 en plusieurs étapes.

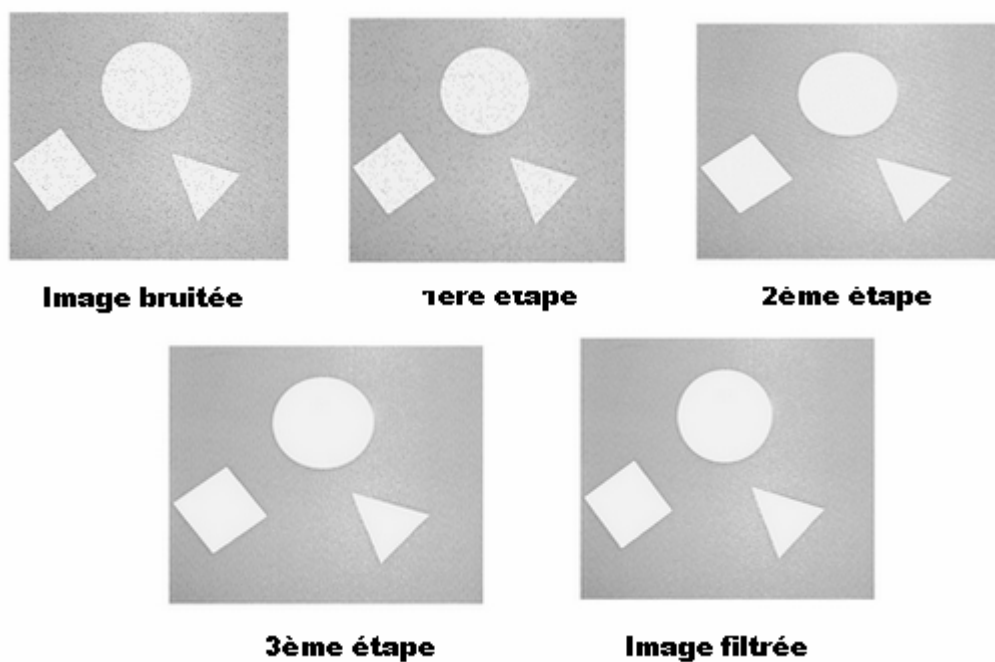


Figure 2.27 : La réduction de bruit avec le filtre médian

### Intérêt du filtre médian:

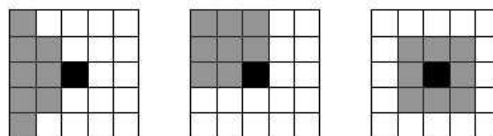
- Un pixel non représentatif dans le voisinage affectera peu la valeur médiane.
- La valeur médiane choisie étant le niveau de gris d'un des pixels considérés, on ne crée pas alors de nouveaux niveaux de gris dans l'image. Ainsi lorsque le filtre passe sur un contour très marqué il le préservera mieux, mais il est coûteux en temps de calcul.

### -Le filtrage de Nagao

L'idée de Nagao [26] est de faire un partitionnement de la fenêtre d'analyse, de taille 5x5, en plusieurs fenêtres de filtrage. Après calcul de la variance empirique sur chaque domaine la sortie du filtre est la moyenne sur le domaine ayant la variance la plus faible.

Sous certaines conditions, Nagao montre que les domaines ayant la plus grande variance sont révélateurs de présence d'au moins deux régions. Les performances du filtre ne changent pas par suppression du domaine 9 [27].

Le principe de ce filtrage est de remplacer chaque pixel de l'image par la valeur moyenne des pixels contenus dans une fenêtre particulière. Il s'agit de choisir la fenêtre la mieux adaptée parmi un certain nombre de fenêtres prédéfinies.



9 fenêtres sont ici définies, chacune contenant 9 pixels, dont le pixel à remplacer. Elles s'inscrivent dans une fenêtre de taille 5x5 centrée sur le pixel à modifier [30].

### Algorithme de Nagao

L'algorithme de Nagao est un peu plus complexe. Il permet d'adoucir l'image sans pour autant rendre les bords flous. Dans un carré de 5x5 entourant le pixel à traiter, la variance de 9 différentes régions est calculée. La valeur du pixel central est remplacée par la valeur moyenne de la région qui a la plus faible variance.

- Pour chaque pixels de l'image :
- Filtrer l'image par chacune des 9 matrices de Nagao
- Calculer la variance des 9 résultats ci-dessus

- Rechercher l'ensemble qui a la plus faible variance
- Remplacer la valeur du pixel par la moyenne de l'ensemble ci-dessus

Comme dans le cas du filtre médian, le filtre de Nagao est appliqué sur les trois composantes RGB indépendamment.

Ceci paraît facile, mais les boucles imbriquées sont très lentes. Par conséquent, cet algorithme fonctionne relativement lentement sous Matlab. Ceci peut devenir un inconvénient majeur si le nombre d'images à traiter est important.

## 2.5 Morphologie mathématique

La morphologie mathématique est une catégorie particulière de filtre d'ordre donc théorie de traitement non linéaire qui est aujourd'hui très largement utilisée en analyse d'images binaires, mais qui a depuis été étendue aux images en niveaux de gris. La morphologie mathématique ne s'appuie pas sur le traitement de signal, mais repose sur la théorie des ensembles, ce qui en fait une discipline relativement « auto contenue » et formant un tout cohérent[31] .

La plus grande partie de cette théorie a été développée au Centre de Morphologie Mathématique (CMM) de l'Ecole des Mines de Paris.

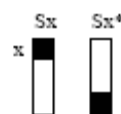
Les opérateurs morphologiques travaillent aussi sur le voisinage local de chaque pixel. La forme de ce voisinage est appelée **élément structurant**. Comme pour les filtres linéaires ou médians, on peut utiliser des éléments structurants de taille et de forme variées. Les éléments structurants couramment utilisés sont le cercle, isotrope, et le carré, qui permet d'accélérer les calculs.

### Notations et définitions

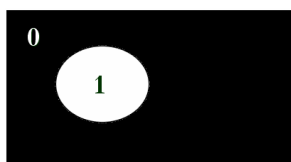
- Objets :  $X = \{ x / x = 1 \}$

- Fonds :  $\sim X = \{ x / x = 0 \}$

- Élément structurant :



$S_x$  caractérisé par ses dimensions et sa forme ,son centre  $x$  est le point d'application de l'opération locale



Tout d'abord, nous présenterons les transformations morphologiques basiques que nous projetons d'utiliser. Parmi les opérateurs morphologiques, on distingue :

- La dilatation et l'érosion morphologiques.
- L'ouverture et la fermeture morphologiques.

Ce sont l'érosion, la dilatation, l'ouverture et la fermeture qui reposent sur des opérations de supérieur et d'inférieur entre l'image d'origine et une forme géométrique que l'on appelle élément structurant.

Le principe de base de l'analyse morphologique est d'extraire les formes de l'image à partir des réponses fournies à différents tests (transformations). Il a été prouvé que ces méthodes sont très efficaces pour la reconnaissance des objets comme pour leur analyse.

## 2.5.1 Opérateur morphologique de dilatation

### 2.5.1.1 Définition

Ensemble des pixels recouverts par l'élément structurant, lorsque celui-ci parcourt l'objet (son centre restant dans l'objet).

Permet d'ajouter à l'objet initial les pixels qui lui sont proches, dans des directions dépendant de l'élément structurant choisi.

$$A \oplus B = \left\{ x \mid (\hat{B})_x \cap A \neq \emptyset \right\} \quad \text{où } B_x \text{ est l'élément structurant centré en } x$$

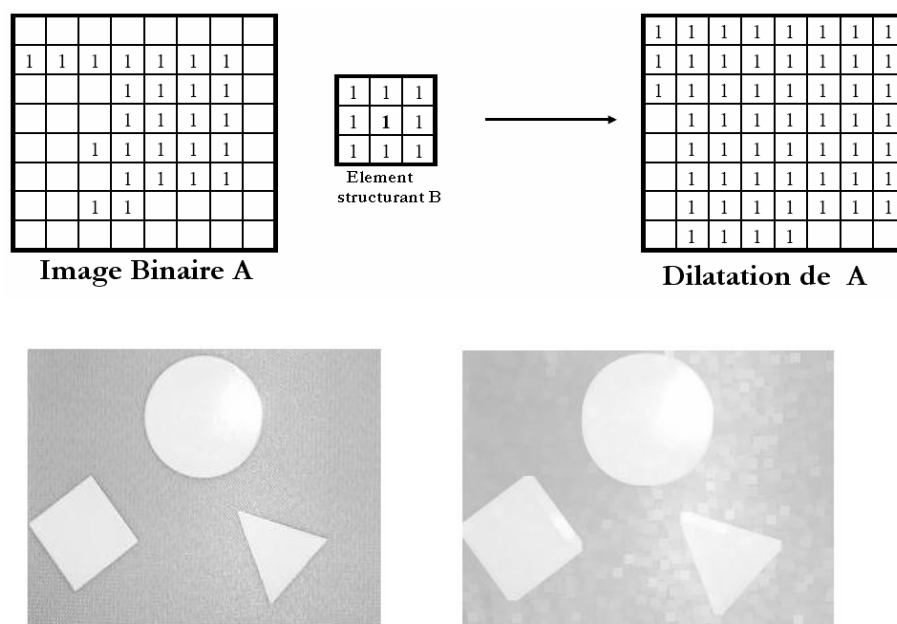


Figure 2.28 : Image dilatée

## 2.5.2 Opérateur morphologique d'érosion :

### 2.5.2.2 Définition

-Opération complémentaire à la dilatation.

-Ensemble des pixels non recouverts par l'élément structurant, lorsque celui-ci parcourt l'objet (son centre restant hors de l'objet).

Permet d'enlever à l'objet initial les pixels qui sont proches de ses bords, dans des directions dépendant de l'élément structurant choisi.

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad \text{où } B_x \text{ est l'élément structurant centré en } x$$

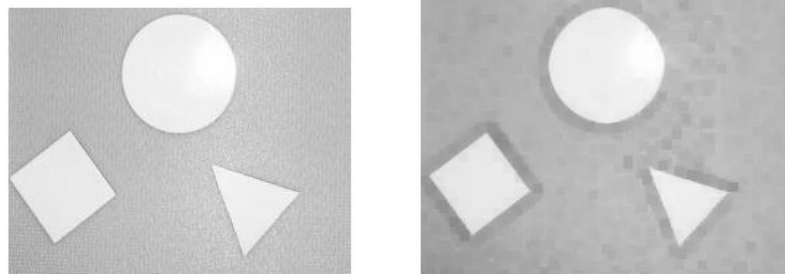
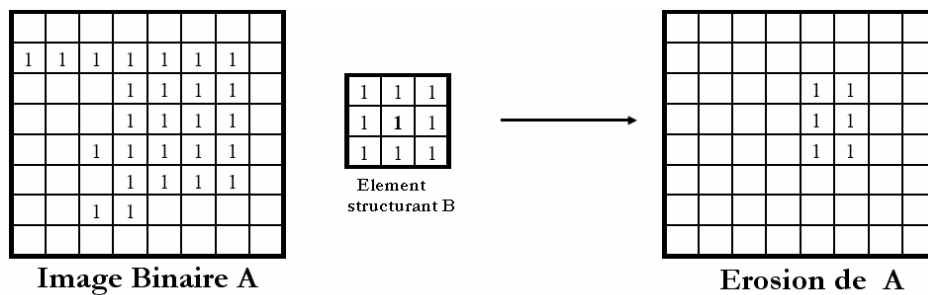


Figure 2.29 : Erosion d'image

## 2.5.3 Ouverture (érosion + dilatation) :

### 2.5.3.1 Définition

Lisse (nettoie) les contours, supprime les petits îlots ou parties saillantes tout en gardant la taille originale de l'objet

$$A \circ B = (A \ominus B) \oplus B$$

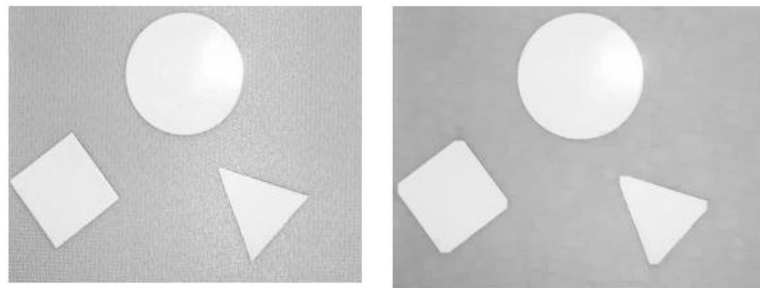
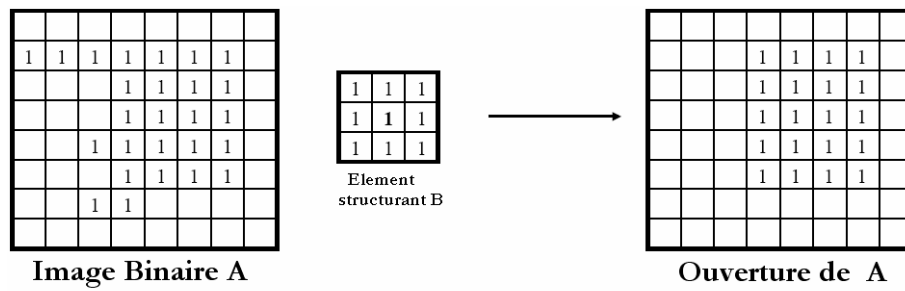


Figure 2.30 : Ouverture d'image

## 2.5.4 Fermeture (dilatation + érosion) :

Supprime les trous, complète les défauts de forme, connecte les objets proches

$$A \bullet B = (A \oplus B) \ominus B$$

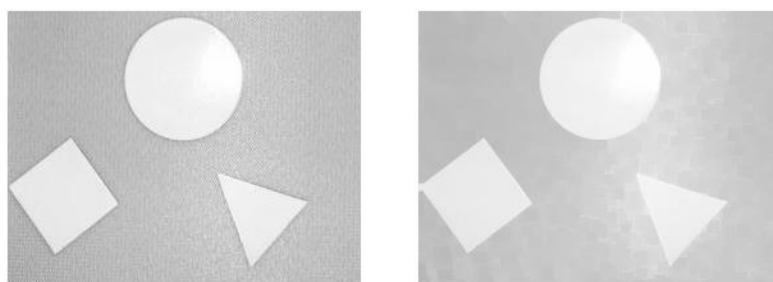
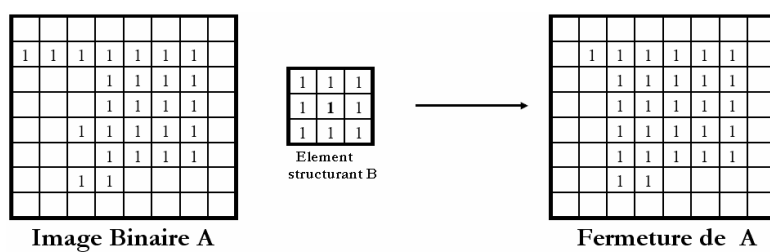


Figure 2.31 : Fermeture d'image

## 2.6 Conclusion

Le filtrage médian est plus adapté que le filtrage linéaire pour réduire le bruit impulsif.

Il élimine les événements de taille inférieure à la moitié de la taille de la fenêtre d'observation.

Le filtre médian est le plus simple et son action est tout à fait remarquable sur les images affectées de perturbations puisqu'il permet de les nettoyer pour la netteté des contours.

Cet effet, les filtres non linéaires, constitue un avantage certain par rapport aux filtres linéaires qui ont un effet « passe bas » global agissant aussi bien sur le bruit que sur l'information utile présente dans l'image.

En morphologie mathématique, filtrer, c'est conserver l'image en supprimant certaines structures géométriques (en général implicitement définies par un ou plusieurs éléments structurants).

Le filtre morphologique simplifie l'image en lui préservant la structure, mais il perd en général de l'information. Le filtre morphologique est stable et possède une classe d'invariance connue.

Cette opération, nous a apporté de nouveaux concepts de transformation géométrique d'une image de notre système de vision. Notamment l'opération de dilatation et l'opération d'érosion ainsi que leurs combinaisons appelées ouverture (érosion puis dilatation) ou fermeture (dilatation puis érosion) de même l'ouverture suivie de fermeture. Pour réaliser une reconnaissance d'objets il est nécessaire après un bon prétraitement de faciliter la phase de segmentation qui va être décrite dans le chapitre suivant.



## Segmentation des images

*L'image, comme tout élément, apparaît dans sa totalité pour être intelligible. Mais dans la réalité, elle est la résultante d'un ensemble d'éléments plus ou moins nombreux, plus ou moins petits, selon la distance de vision, son angle, et surtout l'appareillage par le biais duquel l'image est observée. Si l'homme sait naturellement séparer des objets dans une image c'est grâce à des connaissances de haut niveau (compréhension des objets et de la scène). Mettre au point des algorithmes de segmentation de haut niveau (chaque région est un objet sémantique) est encore un des thèmes de recherche les plus courants en traitement d'images. Dans ce chapitre nous allons aborder les différentes méthodes de segmentation, mais malgré que ce soit impossible d'évaluer automatiquement la segmentation, nous pouvons définir des critères sur le résultat de la segmentation pour avoir une idée d'évaluation.*

### 3.1 Introduction

La segmentation est une étape essentielle en traitement d'images et reste un problème complexe. La segmentation est un processus de la vision par ordinateur, généralement c'est la première étape de l'analyse d'image qui vient après le prétraitement. La segmentation est l'extraction de caractéristiques de l'objet, ce qui permet une distinction entre l'objet et le fond.

Elle aide à localiser et à délimiter les entités présentes dans l'image. Il existe une multitude de méthodes de segmentation dont l'efficacité reste difficile à évaluer.

La segmentation des images est le procédé qui conduit à un découpage de l'image en un nombre fini de régions (ou segments) bien définies qui correspondent à des objets, des parties d'objets ou des groupes d'objets qui apparaissent dans une image. C'est une transformation très utile en vision artificielle.

Une erreur dans la segmentation de la forme à reconnaître augmente forcément le risque d'une mauvaise reconnaissance.

Essentiellement, l'analyse de l'image fait appel à la segmentation où l'on va tenter d'associer à chaque pixel de l'image un label en s'appuyant sur l'information portée (niveaux de gris ou couleur), sa distribution spatiale sur le support image, des modèles simples (le plus souvent des modèles géométriques).

## **3-2 Les principes de la segmentation**

De nombreux travaux ont été réalisés sur ce sujet, dans des domaines aussi variés que le domaine médical, militaire, industriel, géophysique, etc... C'est toujours un sujet d'actualité et un problème qui reste ouvert où l'on retrouve de très nombreuses approches :

- La segmentation par régions
- La segmentation par seuillage
- La segmentation par contours
- La Transformée de Hough
- La segmentation par étiquetage en composantes connexes
- La segmentation par LPE

Toutes ces approches visent à l'extraction des caractéristiques. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. Il n'existe pas d'algorithme universel de segmentation, à chaque type d'images correspond une approche spécifique. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation.

### **3.2.1 La segmentation par régions**

La segmentation va consister à regrouper les pixels de l'image en régions (composantes connexes). Ces régions vérifiant un critère d'homogénéité (par exemple sur les niveaux de gris ou sur la texture...). On cherche par ce traitement à obtenir une description de l'image en régions, elle peut être située dans le cadre plus général de la segmentation de données.

Le problème de base de la segmentation de données est le suivant :

On connaît :

- Un ensemble d'entités.
- Un ensemble d'attributs caractérisant ces entités.
- Des relations topologiques entre ces entités.
- Des attributs relationnels entre ces entités.

On cherche :

- Une (ou des) partition(s) de ces données ayant des propriétés intéressantes par rapport aux attributs et aux relations topologiques.

En général, la segmentation peut être définie comme un partitionnement en régions  $R_i$  connexes suivant un prédicat  $p$  telles que :

$$I = \bigcup_{i=1}^N R_i$$

$R_i$  est une région connexe

$$R_i \cap R_j = \emptyset$$

$$P(R_i) = TRUE$$

$$P(R_i \cup R_j) = FALSE \quad (i \neq j)$$

A chacune de ces régions, doit correspondre un objet dans l'image car, dans ce procédé d'analyse d'images, l'objectif ultime est d'être en mesure de décomposer une image en un groupe d'objets distincts. En général, ces objets ont des propriétés qui leur sont propres par rapport à l'image elle-même. Ainsi, il est possible de distinguer de tels objets par différentes mesures telles:

- Leur aspect connexe
- Leur couleur cohérente
- Leurs contours
- Leur texture
- Des informations a priori

L'intérêt de détecter des régions est de pouvoir les manipuler ensuite pour extraire des caractéristiques de forme, de position, de taille ...

### 3.2.2 La segmentation par seuillage

Dans le chapitre précédent, nous avons vu que la distribution des niveaux de gris dans l'image constitue déjà une information précieuse que l'on peut mettre à profit pour identifier des régions particulières de l'image.

Donc le seuillage a pour objectif de segmenter une image en plusieurs classes en n'utilisant que l'histogramme. On suppose donc que l'information associée à l'image permet à elle seule la segmentation, c'est à dire qu'une classe est caractérisée par sa distribution de niveaux de gris.

A chaque pic de l'histogramme est associée une classe.

Nous avons vu qu'il existe de très nombreuses méthodes de seuillage d'un histogramme. La plupart de ces méthodes s'appliquent correctement si l'histogramme contient réellement des pics séparés. De plus, ces méthodes ont très souvent été développées pour traiter le cas particulier de la segmentation en deux classes (bimodale).

L'approche la plus simple est celle où on ne fait intervenir qu'un seuil (on dira que l'image est alors formée d'un fonds et d'un objet). Elle consiste à tester pour chaque pixel de l'image si sa

valeur est supérieure ou inférieure à un certain seuil, et produit une image binaire regroupant les résultats.

### 3.2.2.1 Détection automatique du seuil

Il existe plusieurs méthodes pour calculer de manière automatique la valeur du seuil à appliquer. L'une des plus répandue est la méthode d'Otsu. Elle consiste à faire l'hypothèse que l'image contient deux classes, décrites chacune par une partie de l'histogramme de l'image. La qualité du seuillage est quantifiée en mesurant la variance des niveaux de gris de chaque classe. On cherche la valeur de seuil qui minimise les variances des deux classes. La figure 3.1 et la figure 3.2 ci-dessous montrent la binarisation par la méthode OTSU d'un seuil = 136 et d'un seuil = 102

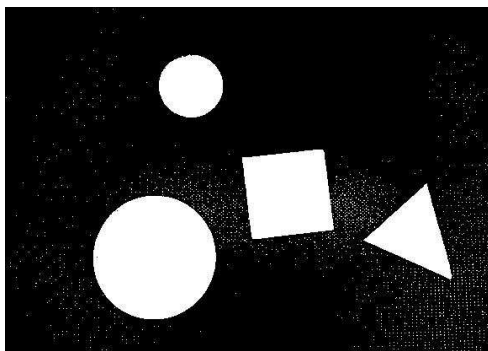


Figure 3.1 : La méthode OTSU seuil=136

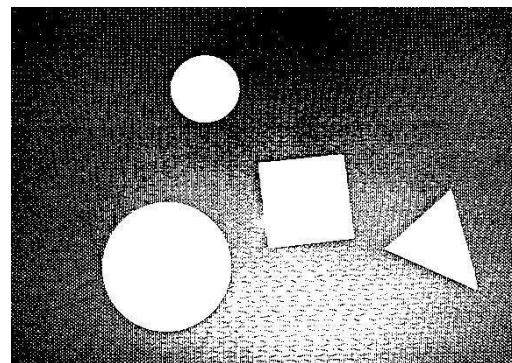


Figure 3.2 : La méthode OTSU seuil=102

### 3.2.2.2 Seuillage à hystérésis

Le seuillage à hystérésis est une méthode un peu plus perfectionnée, qui permet d'améliorer la détection des contours des zones contrastées, tout en évitant de détecter des zones du fond. Le principe est d'utiliser deux seuils : un seuil « haut » et un seuil « bas ». On sélectionne d'abord l'ensemble des pixels au-dessus du seuil haut, puis l'ensemble des pixels au-dessus du seuil bas. On ne garde ensuite que les composantes connexes du seuil bas qui contiennent au moins un pixel au-dessus du seuil haut. En pratique, cette détection peut se faire soit par reconstruction géodésique, soit par étiquetage des composantes connexes de l'image des seuils bas et détection des étiquettes présentes dans l'image des seuils hauts. L'exemple ci-dessous nous montre le seuillage par hystérésis figure 3.3

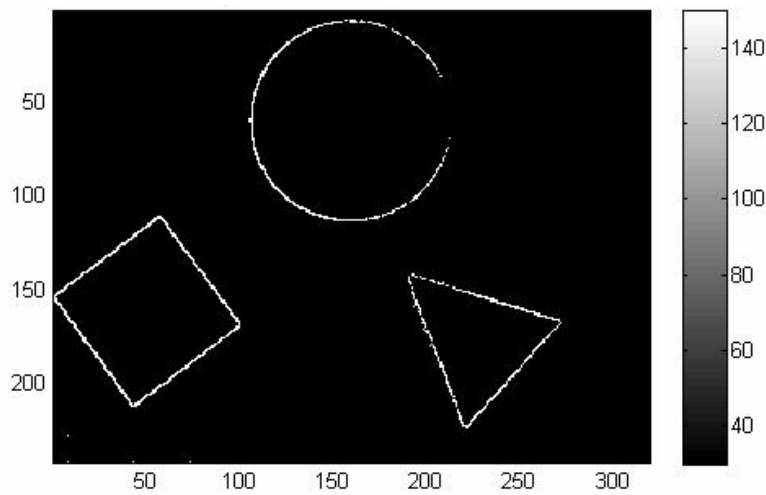


Figure 3.3 : Seuillage par hystérésis

### 3.2.3 La segmentation par contours

De nombreux problèmes pratiques d'analyse d'image nécessitent une analyse préalable où il faut découper l'image en régions homogènes séparées par des contours. En général les régions homogènes sont des régions où l'intensité lumineuse varie lentement en fonction des coordonnées spatiales. Les contours séparant ces régions sont des portions de l'image de faible largeur (largeur de l'ordre d'un pixel) où les variations d'intensité sont importantes. Il existe une multitude de méthodes pour détecter les contours, d'où une autre approche de segmentation des images est la recherche des contours des régions. Les différentes méthodes de détection des contours utilisés sont Sobel, Prewitt, Robert, Canny. Ces méthodes ont été proposées pour détecter des transitions dans les images.

#### 3.2.3.1 Détection de contours

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. La détection des contours d'une image réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurales importantes de l'image [32].

L'extraction de contours, d'un point de vue algorithmique, est une fonction économique qui permet d'alléger considérablement les processus de reconnaissance et localisation d'objet.

Un contour peut être défini comme une "marche d'escalier" si le contour est net, comme une "rampe" si le contour est plus flou ou comme un "toit" s'il s'agit d'une ligne sur un fond uniforme figure 3.4

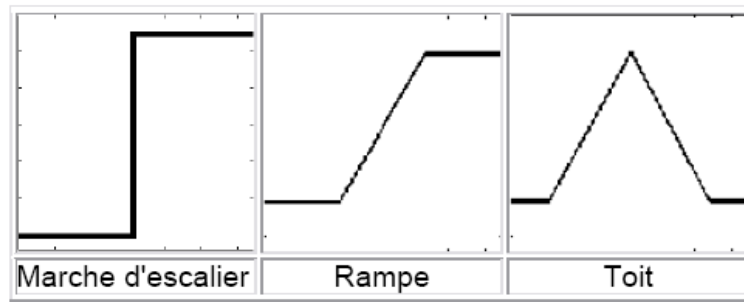
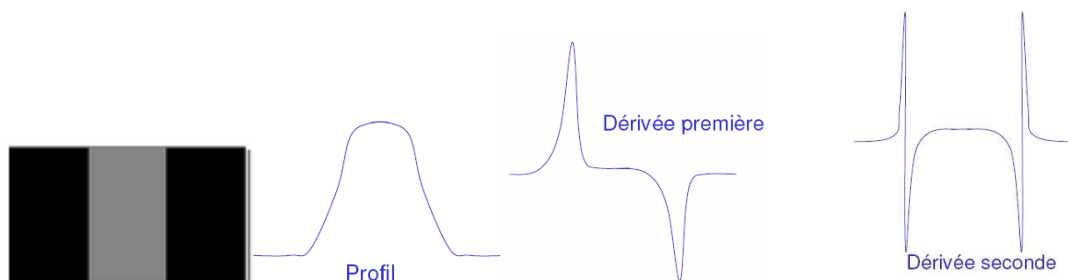


Figure 3.4 : Différentes formes de contours

La détection de contours dans notre cas permet de déterminer les limites d'objets pour la segmentation, dont le but est la séparation des objets dans l'image. Nous allons d'abord présenter les méthodes d'extraction de points de contour, par des techniques de filtrage gradient et Laplacien.

La notion de contour étant liée à celle de variation, il est évident qu'une telle définition nous amène tout naturellement vers une évaluation de la variation en chaque pixel. Une variation existera si la dérivée première (gradient) est localement maximum ou si la dérivée seconde (Laplacien) présente un passage par zéro.



## - Opérateur du premier ordre

Soit  $f(x,y)$  la fonction de répartition de la luminance dans l'image. Le gradient en coordonnées cartésiennes est exprimé par :

$$G_x(x, y) = \frac{\partial f(x, y)}{\partial x} \text{ et } G_y(x, y) = \frac{\partial f(x, y)}{\partial y} \quad 3.1$$

Si les dérivées partielles sont supposées continues en un point quelconque de l'image, alors il est admis que la différentielle de la fonction Image en ce point est une fonction linéaire qui est notée :

$$G(x, y) = f'_x(x, y).dx + f'_y(x, y).dy \quad 3.2$$

Écrit plus généralement sous cette forme :

$$G(x, y) = \frac{\partial f(x, y)}{\partial x} . dx + \frac{\partial f(x, y)}{\partial y} . dy \quad 3.3$$

$G(x, y)$  ou  $\nabla f(x, y)$  est appelé Gradient et a pour amplitude :

$$\|\nabla f(x, y)\| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad 3.4$$

Son orientation est donnée par :

$$\phi = \text{Arc tan} \frac{\frac{\partial f(x, y)}{\partial y}}{\frac{\partial f(x, y)}{\partial x}} \quad 3.5$$

L'analyse du vecteur gradient permet de mettre en évidence un contour. En effet l'amplitude du gradient indique une plus ou moins forte discontinuité et sa direction est par définition normale au contour. Le gradient est un opérateur directionnel.

## - Génération du gradient

La méthode la plus simple de la génération de gradient est de former la différence des pixels le long des lignes et des colonnes de l'image, les dérivées partielles font l'objet d'une approximation par une méthode de différentiation.

$$\frac{\partial f(x, y)}{\partial x} \text{ est évalué par } \Delta_x f(x, y) = f(x, y) - f(x-1, y) \quad 3.6$$

$$\frac{\partial f(x, y)}{\partial y} \text{ est évalué par } \Delta_y f(x, y) = f(x, y) - f(x, y-1) \quad 3.7$$

Qui correspondent à des convolutions avec les noyaux  $\begin{pmatrix} -1 & 1 \end{pmatrix}$  qui est l'approximation de

$$\frac{\partial f(x, y)}{\partial x} \text{ et } \begin{pmatrix} -1 \\ 1 \end{pmatrix} \text{ qui est l'approximation de } \frac{\partial f(x, y)}{\partial x}$$

Ces dérivées sont calculées par convolution de l'image avec un masque de différences

$\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$  pour la dérivée horizontale et  $\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$  pour la dérivée verticale qui constitueront des

frontières bien centrées. Nous allons maintenant présenter un type de filtre à une sensibilité au bruit un peu moins important : le filtre de Prewitt.

### -Filtre de Prewitt

L'application d'un filtre de Prewitt est constitué de deux étapes. Dans un premier temps, on applique les deux filtres de convolution à l'image. Le premier de ces filtres permet de détecter les contours verticaux, s'écrit  $h_x = [-1 \ 0 \ 1]$  tandis que le second filtre permet de détecter les contours horizontaux de l'image (transposé). Nous obtenons ainsi deux images.

Il suffit ensuite de fusionner les deux images en utilisant un calcul de norme.

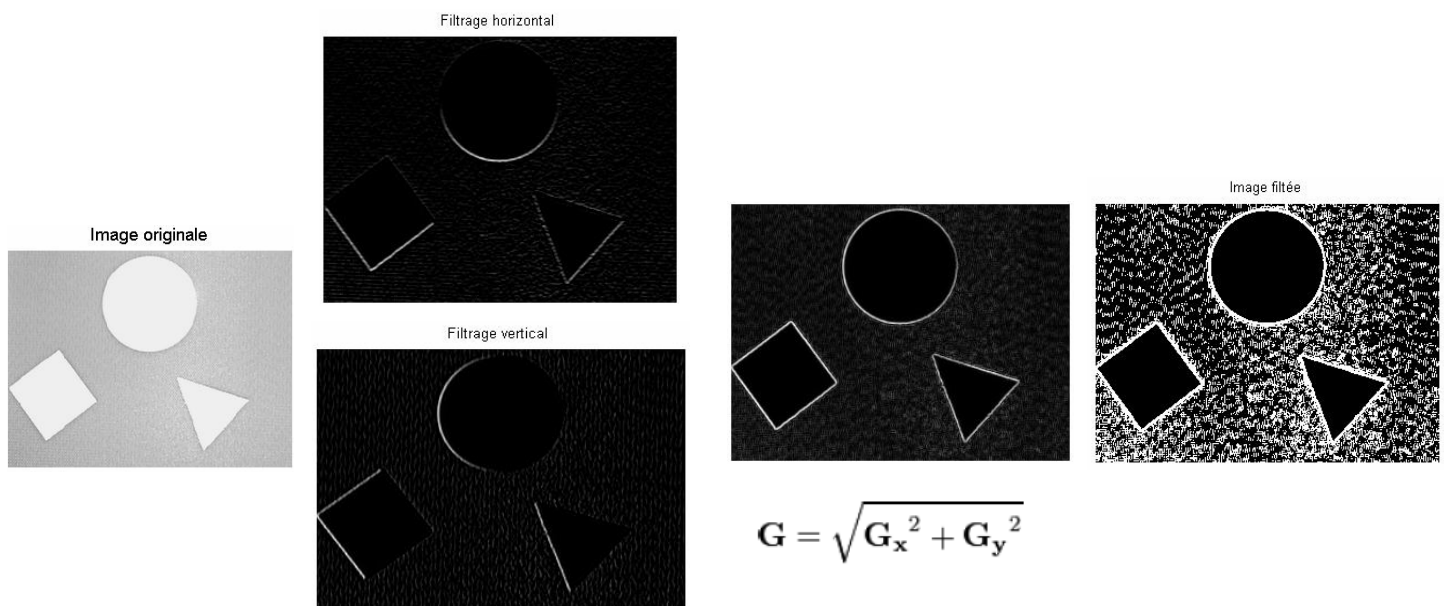


Figure 3.5 : Détection de contours à l'aide du filtre de Prewitt (seuil : 0.1)



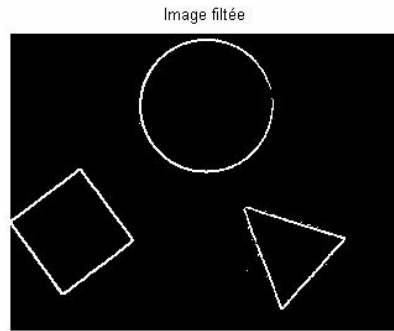


Figure 3.6 : Détection de contours à l'aide du filtre de Prewitt (seuil : 0.3)

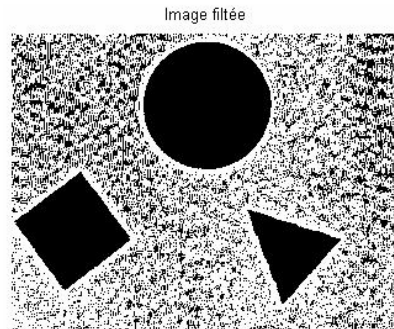


Figure 3.7 : Détection de contours à l'aide du filtre de Prewitt (seuil : 0.05)

### -Filtre de Sobel

Le principe de ce filtre est que l'opérateur calcule le gradient de l'intensité de chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords.

L'opérateur utilise des matrices de convolution. La matrice (généralement de taille 3×3) subit une convolution avec l'image pour calculer des approximations des dérivées horizontale et verticale. Soit  $A$  l'image source,  $G_x$  et  $G_y$  deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point.

L'opérateur utilise des matrices de convolution. La matrice (généralement de taille 3×3) subit une convolution avec l'image pour calculer des approximations des dérivées horizontale et verticale. Soit l'image source, et deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point.

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad \text{et} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

En chaque point, les approximations des gradients horizontaux et verticaux peuvent être combinées comme suit pour obtenir une approximation de la norme du gradient:

$$G = \sqrt{G_x^2 + G_y^2}$$

Ces images sont calculées comme suit:

Les figures montrent les images calculées par des seuils différents :

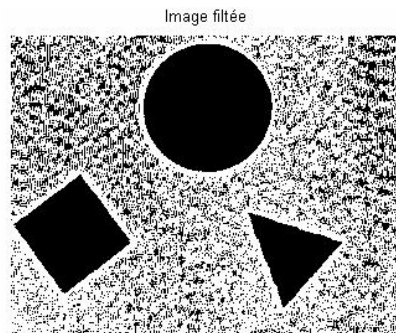


Figure 3.8 : Détection de contours à l'aide du filtre de Sobel (seuil : 0.1)

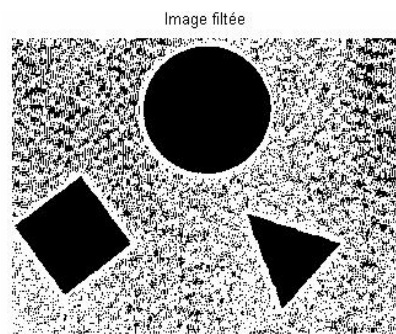


Figure 3.9 : Détection de contours à l'aide du filtre de Sobel (seuil : 0.3)

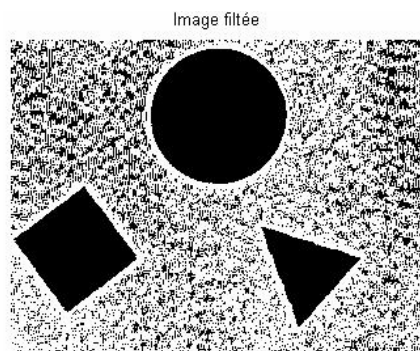


Figure 3.10 : Détection de contours à l'aide du filtre de Sobel (seuil : 0.05)

Nous pouvons voir que les résultats obtenus sont similaires à ceux obtenus à l'aide du gradient de Roberts. Toutefois, le filtre de Sobel permet de filtrer une partie des parasites lorsque le seuil est trop faible comme en témoigne la capture d'écran réalisée avec un seuil de 0.05.

### -Filtre de Robert

Le gradient de Robert est constitué de deux étapes. Dans un premier temps, on applique les deux filtres de convolution à l'image. Le premier de ces filtres permet de détecter les contours ayant une direction allant du coin supérieur gauche de l'image au coin inférieur droit (ou l'inverse) tandis que le second filtre permet de détecter les contours ayant une direction allant du coin inférieur gauche de l'image au coin supérieur droit (ou l'inverse). Nous obtenons ainsi deux images. Il suffit ensuite de fusionner les deux images en utilisant un calcul de norme.

Les filtres de Robert sont une approche discrète de la dérivée d'une fonction: le gradient de cette fonction

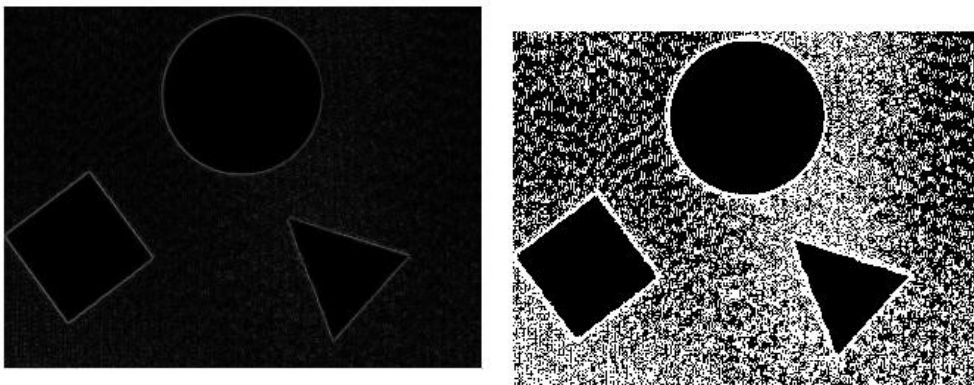


Figure 3.11 : Détection de contours à l'aide du filtre de Robert (seuil : 0.1)

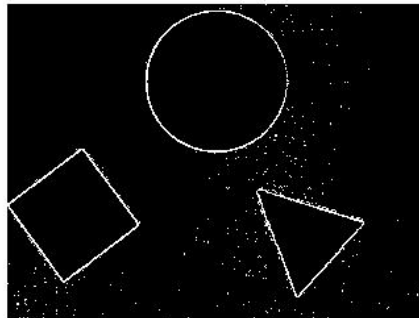


Figure 3.12 : Détection de contours à l'aide du filtre de Robert (seuil : 0.3)

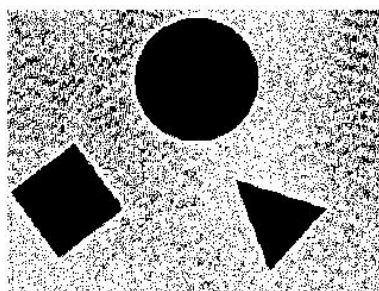


Figure 3.13 : Détection de contours à l'aide du filtre de Robert (seuil : 0.05)

Nous pouvons voir que les résultats obtenus sont similaires à ceux obtenus à l'aide du gradient de Robert .Toutefois, le filtre de Sobel permet de filtrer une partie des parasites lorsque le seuil est trop faible comme en témoigne la capture d'écran réalisée avec un seuil de 0.05.

## - Opérateur de second ordre

Ces méthodes ont été proposées en 1976 [Marr et Hildreth, 1980]. Elles ont eu une grande importance historique, étant considérées comme le prototype du détecteur de contour inspire des systèmes biologiques Elles utilisent le fait que le passage par zéro du Laplacien permet de bien mettre en évidence les extremums de la dérivée [34]. Le Laplacien d'une image représente sa dérivée seconde. En effet, si par hypothèse de départ, la fonction image est supposée continue. Alors les propriétés de la dérivée seconde d'une fonction sont utilisées pour caractériser un contour par la présence d'un extremum local par le passage à zéro de la dérivée seconde.

Le Laplacien de la fonction Image est défini par :

$$\nabla^2 f = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad 3.8$$

Ainsi en utilisant également les propriétés de différentiation l'opérateur Laplacien peut s'écrire sous la forme suivante :

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad 3.9$$

Généralement écrit par le masque de convolution suivant :

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ ou } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Le filtre Laplacien 4-connexe a le même principe que le filtre Laplacien 8-connexe.Ce qui change est le masque utilisé.

Voici le masque utilisé pour ce filtre figure 3.14 Nous utilisons donc cette formule pour appliquer ce filtre :

$$\begin{aligned} image(i-1, j-1) = & 8*image2(i, j) - image2(i-1, j-1) - image2(i-1, j) - image2(i-1, j+1) \\ & - image2(i+1, j-1) - image2(i+1, j) - image2(i+1, j+1) - image2(i, j-1) - image2(i, j+1) \end{aligned}$$

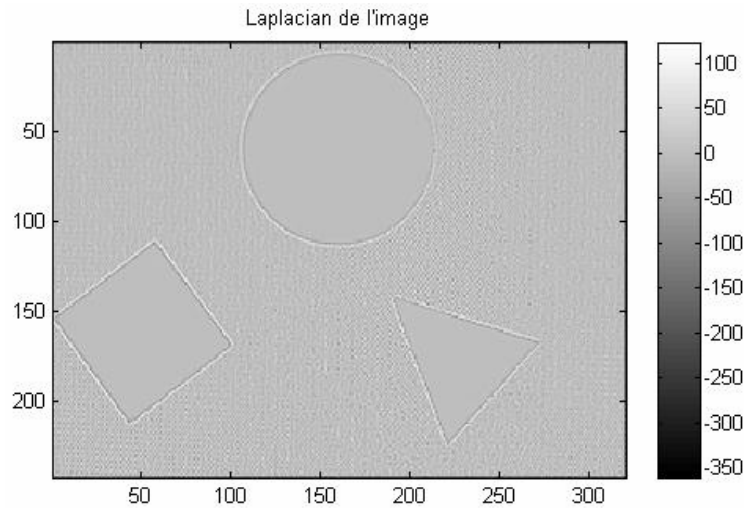


Figure 3.14 : Laplacien de l'image

Nous remarquons que l'opérateur Laplacien est symétrique quelle que soit l'orientation choisie, il n'est donc pas directionnel. L'opérateur Laplacien est fortement sensible aux bruits. Ce qui nous conduit à essayer d'atténuer les bruits par l'emploi des filtres qu'on a déjà implantés dans les chapitres précédents.

### -Détection de contours par la méthode de Canny

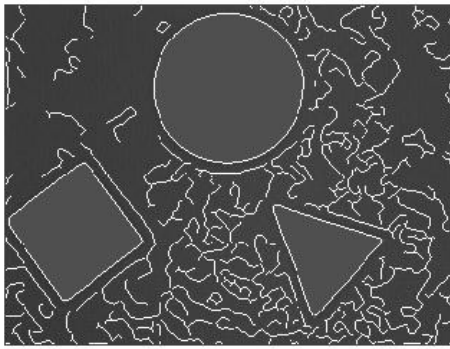
Cet algorithme a été conçu pour être optimal suivant trois critères :

- Bonne détection : faible taux d'erreur dans la signalisation des contours
- Bonne localisation : minimisation des distances entre les contours détectés et les contours réels
- Clarté de la réponse : une seule réponse par contour et pas de faux positifs.

La détection des contours par Canny se déroule en plusieurs étapes. Tout d'abord il s'agit de réduire le bruit de l'image afin d'optimiser la détection des contours. Ceci permet de réduire de fortes réponses lors du calcul du gradient, conduisant ainsi à de faux positifs. Un filtre Gaussien est utilisé afin de "lisser" l'image originale et donc de réduire le bruit.

Le filtre de Canny est basé sur l'algorithme de Sobel, il dispose d'étapes supplémentaires qui permettent de travailler sur des images plus complexes. Cette méthode permet, en changeant l'intensité de détection de contour, d'éliminer les défauts de linéarisation et du bruit et en même temps d'avoir des contours larges. La largeur est importante pour l'algorithme de localisation par corrélation qui suit cet algorithme de détection de contour. Nous présentons dans ce qui suit les images obtenues par les différentes valeurs de  $\sigma$  avec un seuil =  $[3e-03 \ 6e-02]$  ainsi la combinaison des deux valeurs pour différencier la détection des contours

Canny,  $\sigma = 2$ , seuil par défaut = [0.0188, 0.0469]



Canny,  $\sigma = 4$ , seuil par défaut = [0.0313, 0.0781]

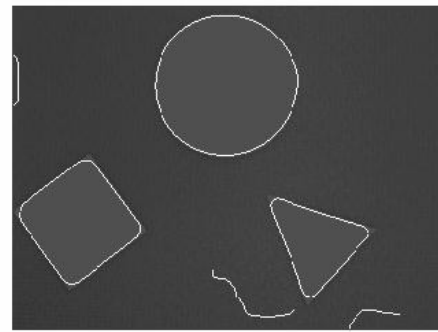
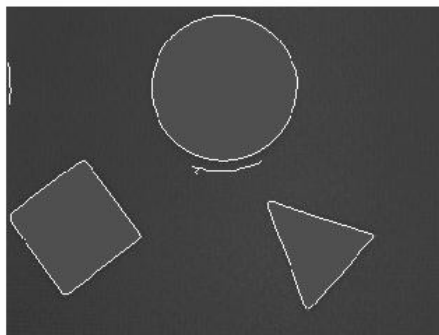
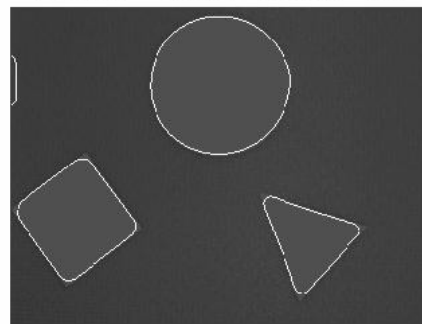


Figure 3.15 : Filtre de Canny

Canny,  $\sigma = 2$ , seuil = [0.005, 0.1]



Canny,  $\sigma = 4$ , seuil = [0.005, 0.1]



Donc on remarque avec  $\sigma = 4$  et un seuil = [0.005, 0.1] on obtient un contour fin

### 3.2.4 La transformée de Hough

La transformée de Hough est une technique de reconnaissance de formes inventée en 1962 par Paul Hough. Cette technique nous permet de reconnaître les lignes (droite), les cercles ou n'importe quelle forme présente dans une image [35].

L'application la plus simple permet d'identifier la présence de lignes dans une image. Elle peut être très facilement étendue à la reconnaissance de cercle aux d'autres modèles mathématiques simples. Il existe également une transformée généralisée de Hough.

Le principe est le suivant. Par un point donné peut passer une infinité de droite

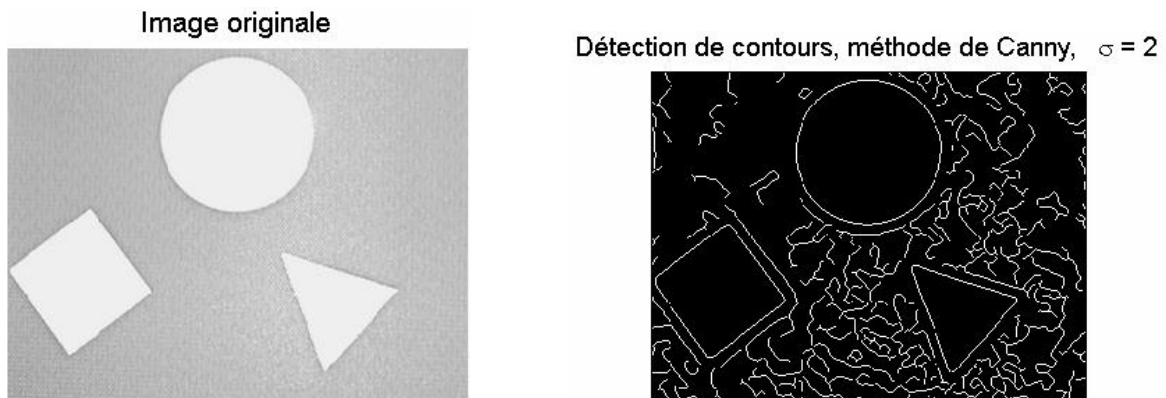
$y = a \cdot x + b$  (ou  $r = x \cos \theta + y \sin \theta$  en coordonnées polaires).

Si l'on trace  $r$  en fonction de  $\theta$ , on obtient une sinusoïde représentant l'ensemble des droites de paramètre  $(r, \theta)$  passant par ce point. C'est ce que l'on appelle « espace de Hough ». Si l'on trace

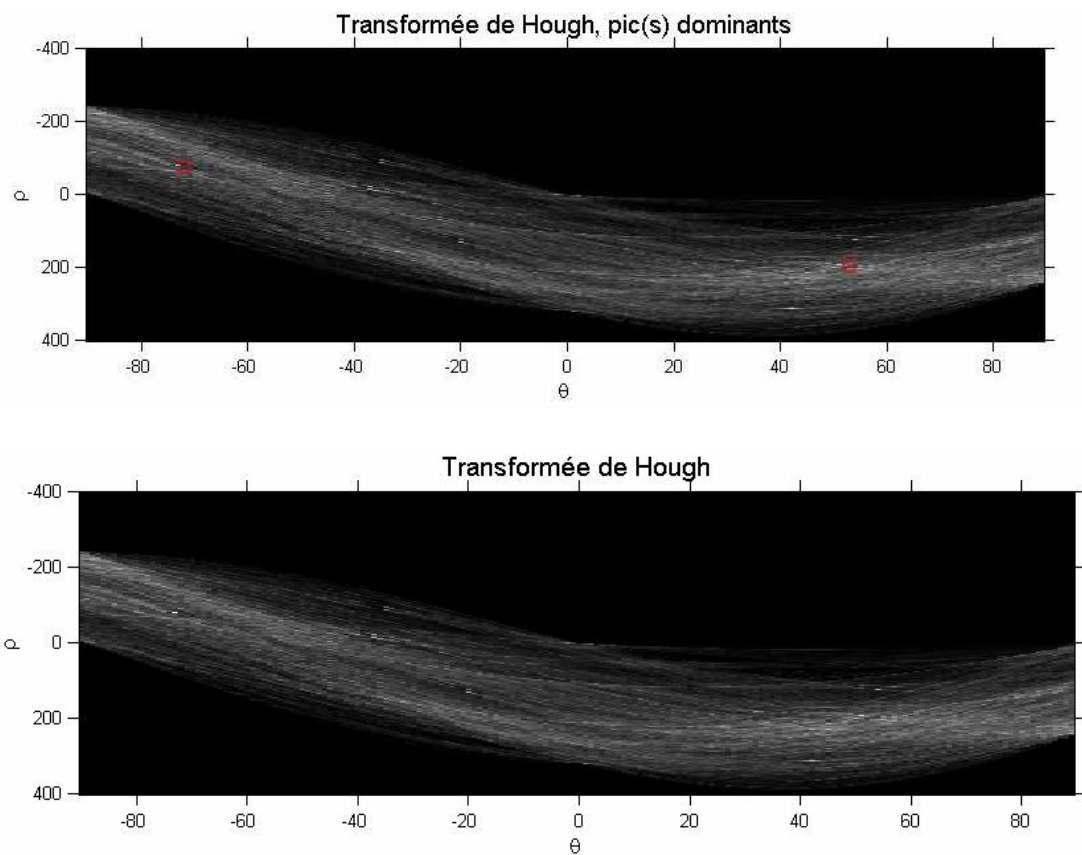


alors une autre sinusoïde par un autre point, l'intersection des deux sinusoïdes donne la valeur de  $(r, \theta)$  de la droite passant par ces deux points. On va donc chercher à tracer pour l'ensemble des points chacune des sinusoïdes correspondante dans l'espace de Hough. On cherchera alors à savoir quels sont les lieux ou les courbes qui se croisent pour en déduire la présence de droites.

Voici quelques figures de la transformée de Hough par la méthode de canny avec la valeur  $\sigma = 2$

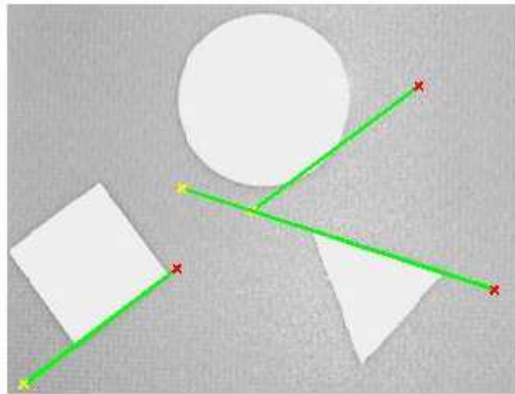


L'image de la transformée de Hough où les 2 pics en carré rouge



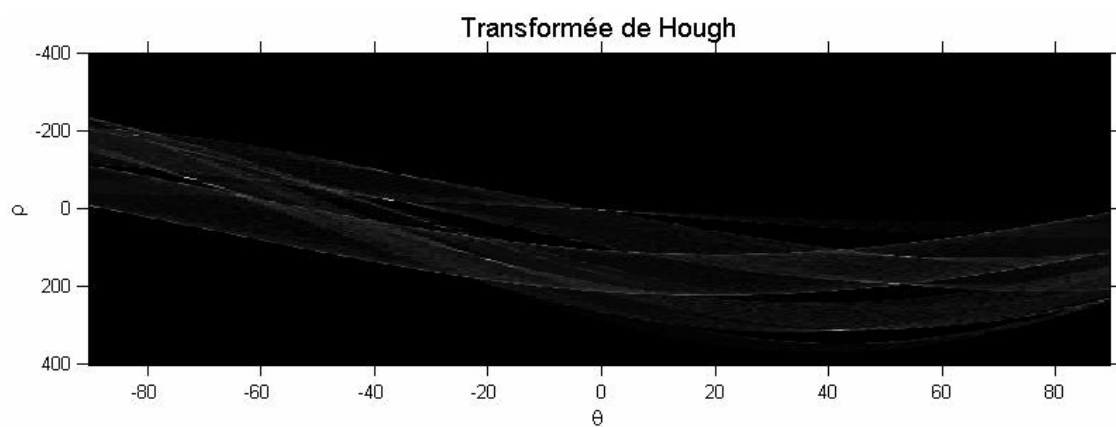
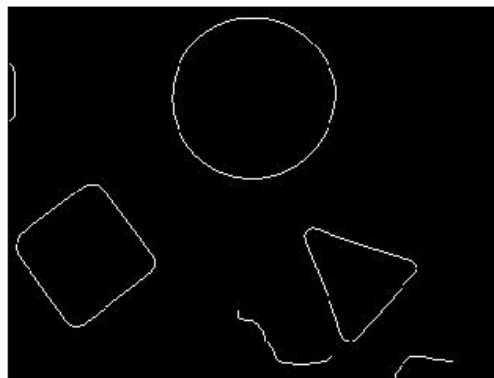
Le résultat du parcours et le tracé des lignes par la transformée de Hough.

Résultat du traitement, distance de connexion 45

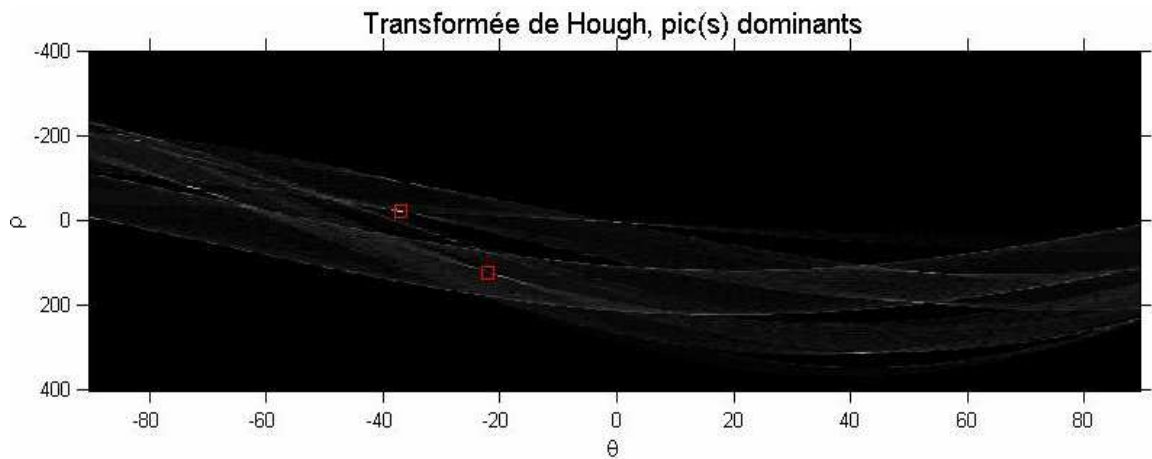


Canny avec  $\sigma = 4$

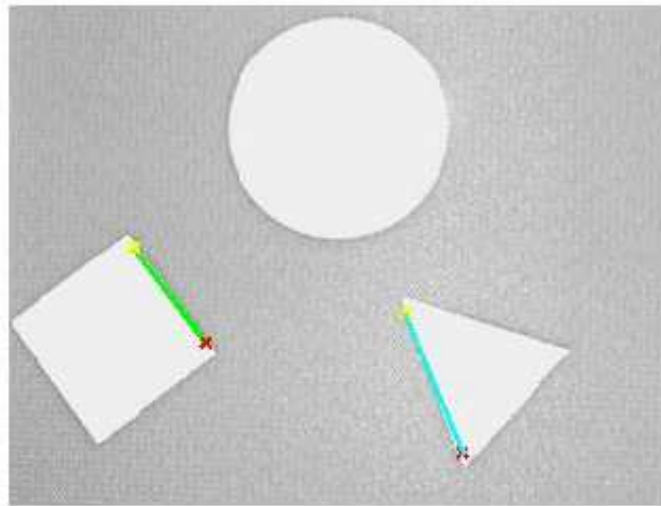
Détection de contours, méthode de Canny,  $\sigma = 4$







Résultat du traitement, distance de connexion 42



Dans cette partie nous distinguons que la détection de contours est très satisfaisante, puisqu' on a pu avoir les bords extérieurs du carré et le triangle avec la valeur de sigma égale à 4.

Nous avons finalement opté, pour la détection la plus simple, c'est-à-dire simplement en appliquant un filtre de Canny. Car les résultats observés en faisant des prétraitements n'étaient pas convaincants. La détection de contours avec Canny n'est certes pas parfaite mais honorable, et tous les prétraitements appliqués n'apportent aucun plus dans cette détection de contours.

### 3.2.5 Segmentation par Etiquetage en composantes connexes

C'est l'opération qui consiste dans une image binaire (forme/fonds) à trouver toutes les classes d'équivalence de la forme, c'est à dire les différentes composantes connexes et à les étiqueter. L'étiquetage consiste à attribuer à chacune des régions séparées (composantes connexes) une étiquette c'est-à-dire un numéro unique qui sera attribué à chacun des pixels appartenant à une même composante connexe (objet isolé). Exemple de la figure 3.16. Sur la notion de chemin

connexe : **c** et **d** peuvent être reliés sans avoir à passer par le fond (chemin connexe), alors que **a** et **b** ne peuvent être reliés sans avoir à passer sur le fond (chemin non-connexe).

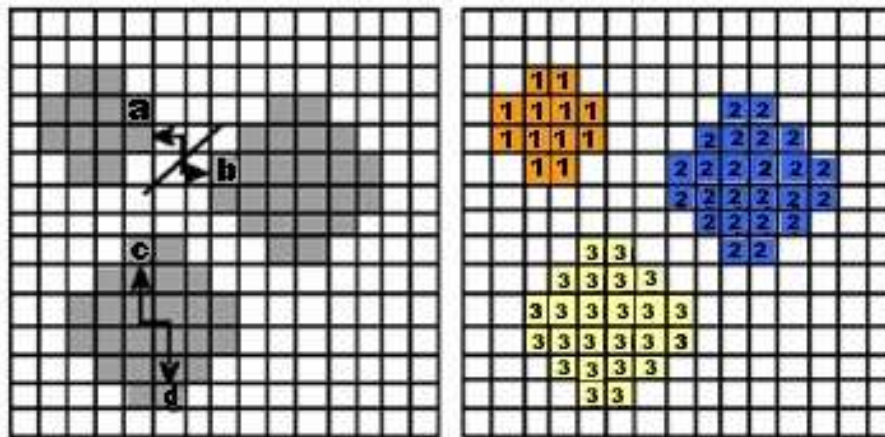


Figure 3.16 : Composante connexe

Exemple d'algorithme :

- Balayer les pixels de l'image ligne par ligne et ne considérer que ceux de la forme (critère de similarité, pixel à 1,...).

Si le pixel n'a pas de voisin connexe déjà étiqueté, créer une nouvelle étiquette et l'affecter à ce pixel.

Si le pixel a exactement un seul pixel connexe étiqueté, lui affecter cette étiquette.

Si le pixel a plus d'un pixel connexe (avec des étiquettes différentes) lui affecter une de ces étiquettes et mémoriser que toutes ces étiquettes sont équivalentes.

- Refaire une passe sur l'image en groupant les étiquettes équivalentes sur une seule étiquette

### 3.2.5.1 Analyse en composantes connexes

Le principe est le suivant :

Partant d'une image binaire, trouver les groupes de pixels connectés, appelés composantes connexes (connected components) ou blobs.

On obtient une image dans laquelle chaque « objet » est identifié.

Cette opération s'appelle analyse (ou étiquetage) en composantes connexes (connected-component analysis / labeling, ou encore blob extraction)

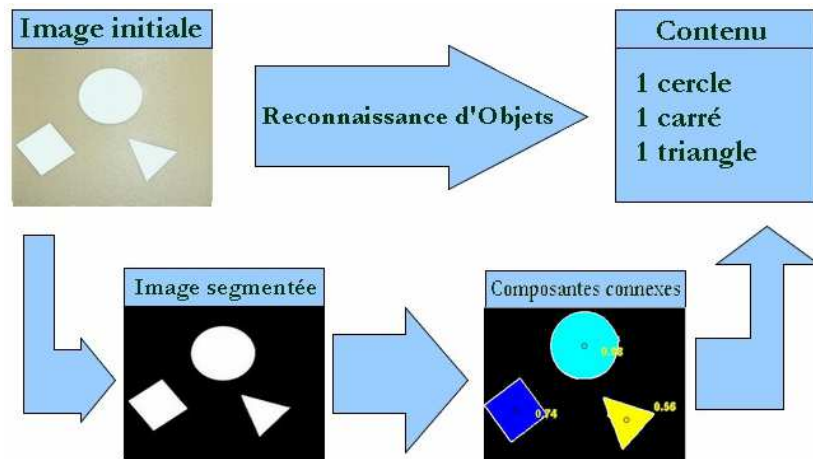


Figure 3.17 : Etiquetage en Composantes Connexes

### 3.2.5.2 Organigramme de reconnaissance d'objets par composantes connexes

On cherche à détecter les différents objets dans l'image, leurs positions et leurs formes en temps réel. Le principe utilisé dans ce programme consiste à calculer, dans chaque image, la surface de l'objet, en nombre de pixels, puis de comparer cette surface à une valeur seuil, correspondant à d'objet dans l'image (figure 3.17). Nous décrivons le synoptique de l'algorithme à mettre en œuvre [36].

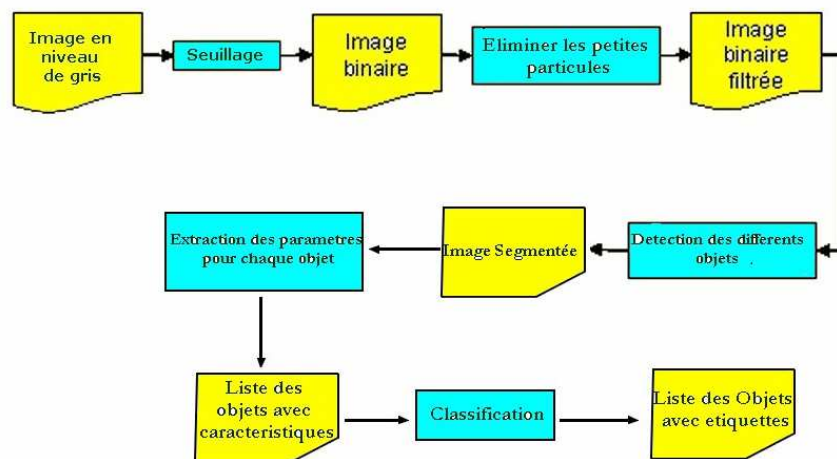


Figure 3.18 : Organigramme de reconnaissance d'objets par composantes connexes

### 3.2.6 La segmentation par LPE (Ligne de Partage des Eaux)

En morphologie mathématique, l'algorithme de la ligne de partage des eaux (LPE), proposé par *Digabel* et *Lantuéjoul*, est un algorithme de segmentation d'images, ce qui revient à décomposer une image en régions homogènes.

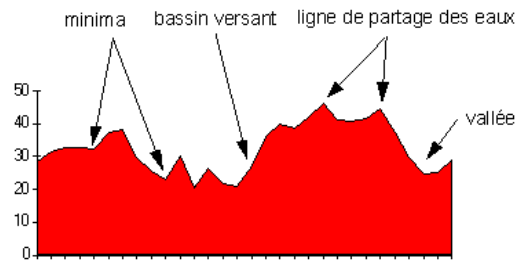


Figure 3.19 : Ligne de Partage des Eaux

#### 3.2.6.1 Définition

- minimum local : c'est le point du plateau d'où on ne peut pas atteindre un point plus bas sans être obligé de remonter.
- bassin versant : c'est la zone d'influence d'un minimum local. Une goutte d'eau s'écoulant dans le bassin versant arrive au minimum local
- ligne de partage des eaux : C'est la ligne séparant 2 bassins versants (de cette ligne, une goutte d'eau peut s'écouler vers au moins 2 minima locaux distincts).

On commence par générer l'image du module du gradient c'est-à-dire les crêtes correspondantes aux contours, on obtient alors la figure suivante :

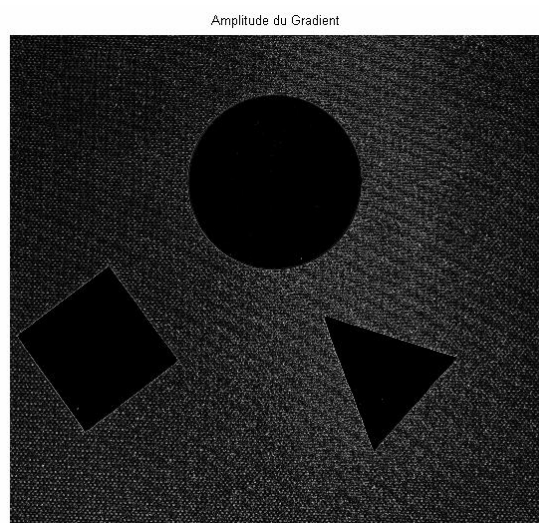


Figure 3.20 : Module du gradient

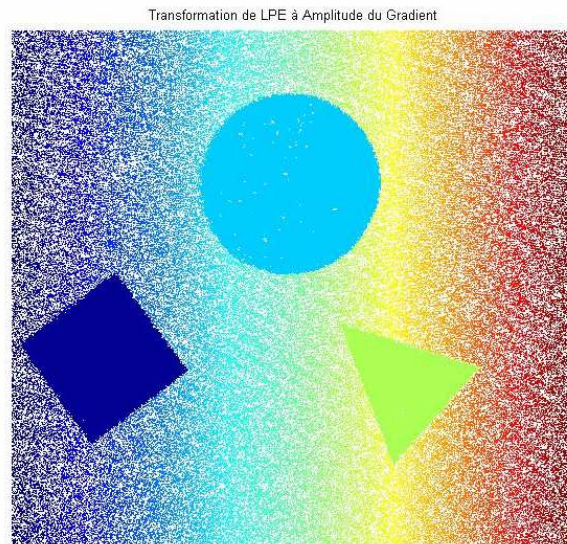


Figure 3.21 : Transformation de LPE à Amplitude de Gradient

Puis on procède à une évaluation de la détection de contours par la LPE et cela en détectant les contours fermés et faisant de la sur-segmentation

L'application d'une dilatation et d'une érosion nous sert à détecter les maxima régionaux dans l'image.

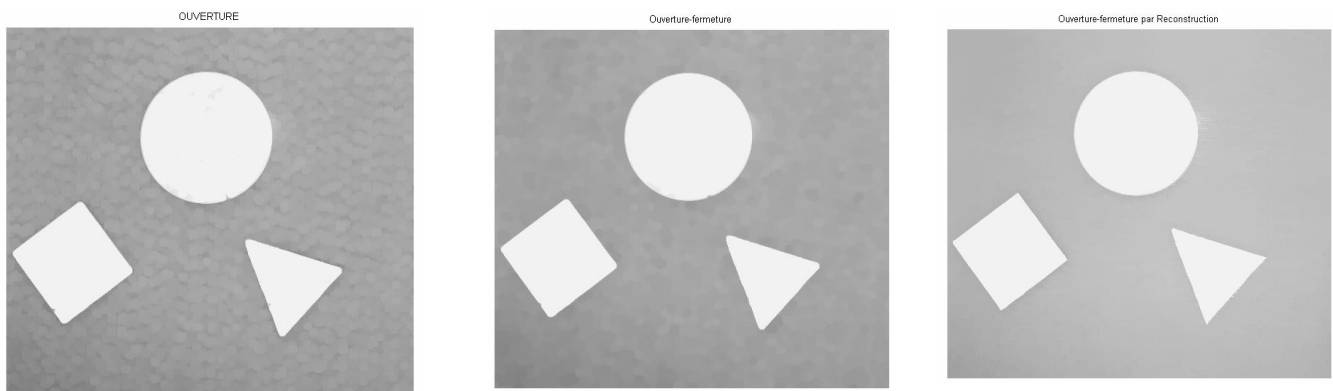


Figure 3.22 : Ouverture et fermeture par reconstruction

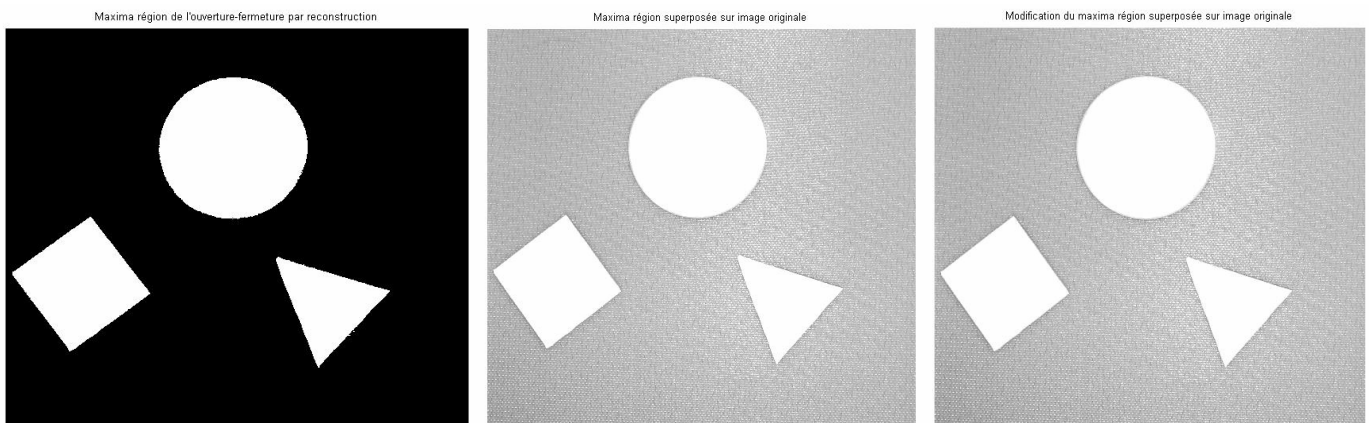


Figure 3.23 : Maxima régionaux ouverture et fermeture par reconstruction  
Maxima régionaux superposé sur image originale



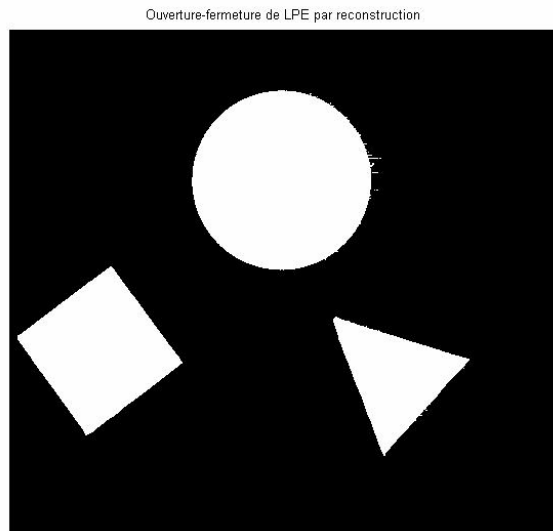


Figure 3.24 : Ouverture et fermeture de LPE par reconstruction

On procède ensuite à la superposition des maxima régionaux sur l'image d'origine puis on refait les mêmes opérations de dilatation et d'érosion. Ces différentes opérations nous permettent la détection des régions dans l'image et on les sépare en utilisant des lignes appelées des lignes de partage des eaux.

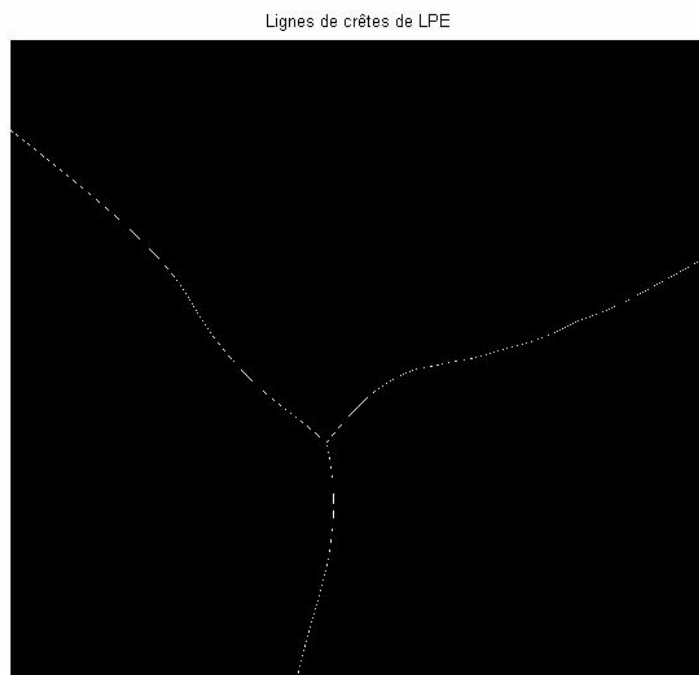


Figure 3.25 : Lignes de partage des eaux

On s'intéresse à présent à ce qu'on appelle les marqueurs qui existent dans les objets recherchés dans l'image qui sont dans notre cas les bords de l'image

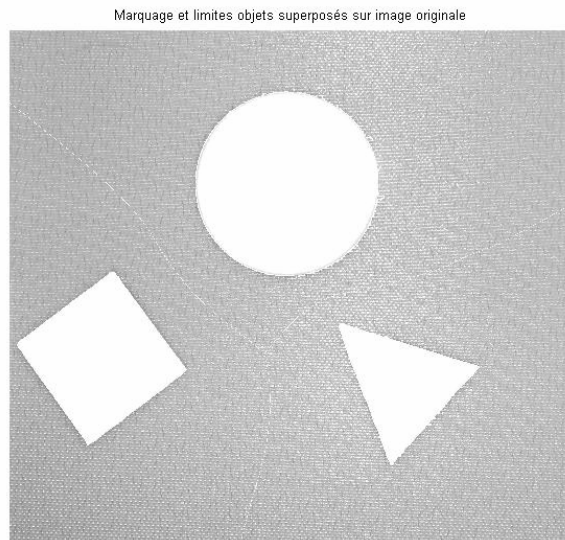


Figure 3.26 : Superposition des marqueurs et lignes de partages sur l'image

On obtient enfin la matrice correspondante à l'application de ces contours et marqueurs et qui définit les différentes régions de l'image.

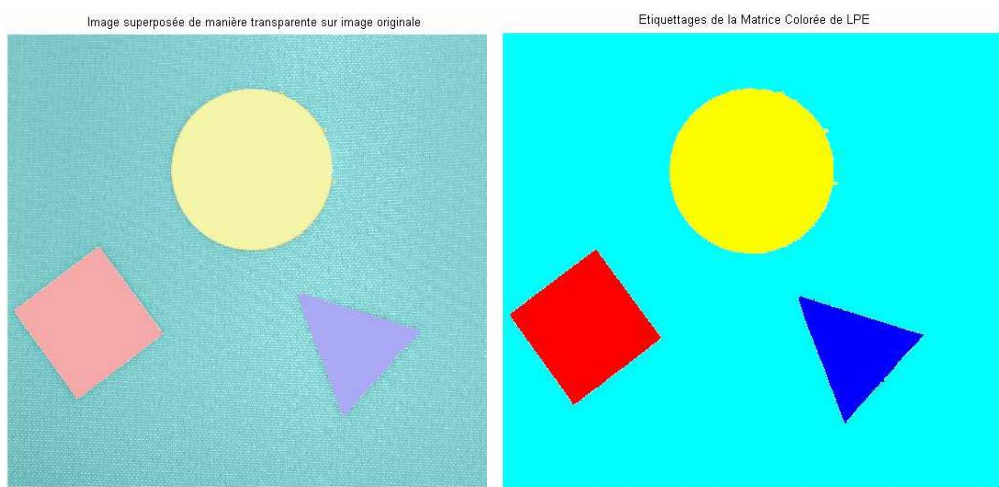


Figure 3.27 : Matrice colorée des régions appliquée sur l'image d'origine

### 3.3 CONCLUSION

Ce domaine cherche à détecter la présence de certaines formes, certains contours ou certaines textures d'un modèle connu, c'est le problème de la détection.

Chacune des méthodes de segmentation a ses avantages, mais a également ses inconvénients. Comme dans la plupart des problèmes, c'est une histoire de compromis.

Les méthodes demandant des paramètres sont très délicates à manipuler. Elles fonctionnent correctement dans le cas où les paramètres sont choisis judicieusement. Il est évident pour une personne de savoir si la segmentation est correcte, mais pour une machine cela pose réellement de grandes difficultés.

Un aspect important lors du choix de l'une de ces méthodes est le temps d'exécution.

Un bon prétraitement serait envisageable. Il permettrait d'améliorer considérablement le résultat final. Chaque méthode impose de concevoir un prétraitement adapté à celle-ci. Le résultat de cette opération est une image de même dimension que l'image d'origine mais décrite par des pixels dont la valeur varie de 1 à  $r$ ;  $r$  est le nombre de régions ou d'étiquettes. On considère généralement qu'il ne s'agit donc plus d'un traitement bas niveau. Enfin c'est le lien avec la description et la reconnaissance qu'on va décrire dans le chapitre suivant.



## Description et Interprétation

*Pour reconnaître une forme, il faut, au préalable la décrire, c'est-à-dire lui attribuer des caractéristiques distinctives. La description et l'interprétation est une étape essentielle dans tous les systèmes de vision artificielle. Une fois les objets isolés du fond, étape suivante de la reconnaissance d'objets est la description des caractéristiques qui décrivent les propriétés de l'objet bien que une interprétation des caractéristiques (descripteurs) pour classer les objets traités, nous avons introduit l'approche corrélation qui est une méthode efficace et rapide pour la localisation de certaines formes géométriques. Les méthodes de reconnaissance de forme proposées par Hough ou Freeman sont des méthodes d'étude morphologique, mais elles nécessitent un temps de calcul important dans le cas des formes géométriques complexes. Étant donné que la corrélation est un outil de reconnaissance à caractère global, la tâche essentielle dans la mise en oeuvre du corrélateur est la conception du filtre approprié.*

### 4.1 Introduction

La forme est reconnue comme étant l'une des principales caractéristiques qui décrivent le contenu d'une image. La perception des objets chez l'homme procède de l'analyse de la forme des objets ainsi que des transformations géométriques qui opèrent sur les objets [37]. Cela revient à dire qu'un objet est reconnu comme appartenant à une certaine catégorie d'objets que nous connaissons (par exemple cette boîte carrée est une télé), et qui a subi des transformations géométriques (réduction d'échelle ou rotation). Le problème fondamental en reconnaissance d'objets est de déterminer dans quelle mesure deux objets sont similaires indépendamment de leur pose et de leur orientation dans l'image. C'est pourquoi la plupart des techniques de reconnaissance/discrimination d'objets font appel à une transcription de la représentation de l'image originale sous la forme de descripteurs invariants par rapport aux transformations géométriques. Ainsi, à partir d'un seul exemple d'objet, on peut décrire tous les objets dérivés selon une rotation, une translation ou un changement d'échelle.

Une fois les objets d'intérêt isolés, l'étape suivante de la reconnaissance d'objets est la représentation et description. Dans notre travail, l'image que nous avons choisie est un ensemble de caractéristiques qui décrivent les propriétés physiques des objets telles que: la forme, la

surface, le périmètre, la largeur, la hauteur, le centre de gravité, l'orientation. La couleur est un élément important pour la classification et la description de l'objet en vision par ordinateur. Pour cela on a préféré utiliser simplement des objets en blancs pour faciliter la détection.

Etant donnée les différentes complexités de formes que l'on peut rencontrer, la reconnaissance d'objets dans les images basée sur ces objets ou formes présente trois problèmes principaux :

La représentation de la forme elle-même, la comparaison des mesures issues de cette forme avec d'autres valeurs de référence, et l'identification des formes. Parmi ces trois points, la représentation de la forme est l'aspect le plus délicat à traiter. Plusieurs algorithmes le résolvant existent déjà. Ils peuvent être séparés en deux catégories : les invariants de formes contours et les invariants de forme régions.

Dans le cas des invariants de formes contours, seuls les contours de l'objet sont exploités. Les deux méthodes les plus utilisées sont la chaîne du code de Freeman et les descripteurs de Fourier.

Dans le cas des invariants régionaux, tous les pixels à l'intérieur de la forme sont pris en compte. Les invariants de cette catégorie les plus utilisés sont les moments invariants de formes tels que ceux de Hu, Zernike ou les moments complexes.

## **4-2 Invariants de formes Contours**

### **4.2.1 Code de Freeman**

#### **4.2.1.1 Principe du codage**

Le code de Freeman est une méthode très utile pour représenter le contour d'un objet sur les images. C'est une technique de représentation des directions. Elle consiste à décrire le contour d'une forme en indiquant les pixels voisins immédiats par l'une des 8 directions possibles. Les 8 directions indiquent les 8 voisins immédiats possibles du pixel analysé.

Le codage d'un contour se fait donc de la façon suivante :

1. transmission des coordonnées absolues du point de départ.
2. transmission de la liste des codes de déplacement d'un point du contour au suivant sur le maillage.

Le code de Freeman se fait une fois l'objet est localisée. Le centre de l'objet et les coordonnées de chaque pixel formant le contour, il sera possible de calculer la distance entre le centre de l'objet et le contour.

Plusieurs critères peuvent être identifiés pour retrouver certaines informations morphologiques ou critères géométriques de l'objet. Cette solution sera détaillée dans la partie caractéristique géométrique d'un objet.

On peut utiliser deux types de représentations : celle à 4 directions et l'autre à 8 directions. Chaque direction est codée par un chiffre comme la montre la figure 4.1.

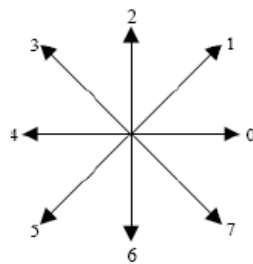
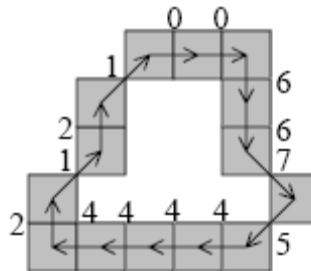


Figure 4.1 : Représentation à 8 directions pour le code chaîne.

La figure 3.2 présente un exemple de ce que peut donner le code chaîne.



La séquence associée à cette forme est donc : 21210066754444

Alors qu'un carré de 5 pixel sera codé par 0 0 0 0 2 2 2 2 4 4 4 4 6 6 6 6

Ainsi le pixel de départ peut être redéfini de sorte que la séquence de nombre forme le plus petit entier possible. D'après l'exemple précédent on aurait alors : 00667544442121.

Néanmoins cette méthode rend le code invariant à la rotation si la forme elle-même est invariante à la rotation.

## 4.2.2 Les descripteurs de Fourier

Outil permettant de décrire l'enveloppe externe d'une forme, c'est-à-dire son contour. Si on considère un plan cartésien 2-D, le contour issu des Fourier descripteurs peut être tracé et

rééchantillonné dans le sens des aiguilles d'une montre de sorte à trouver une distribution uniforme de K points comme la montre la figure 3.3.

Chaque point peut être représenté par un couple de coordonnées  $(x_0, y_0), \dots, (x_{K-1}, y_{K-1})$ . Ces coordonnées peuvent être exprimées autrement sous la forme  $x(k) = xk$  et  $y(k) = yk$ . On peut donc représenter le contour du carré par une suite de points complexes :

$$s(k) = x(k) + jy(k), k = 0, 1, \dots, K-1$$

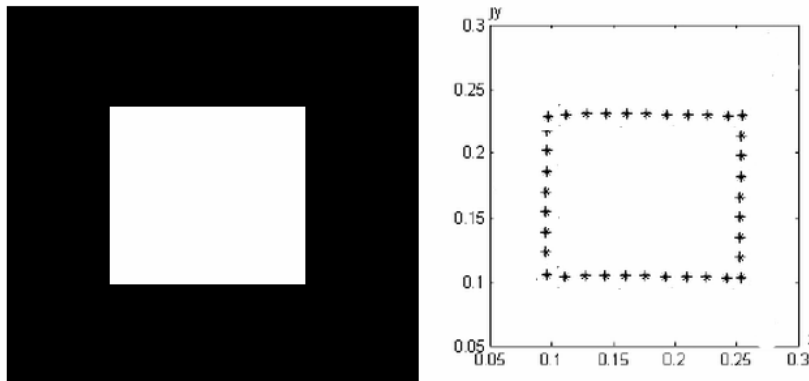


Figure 4.2 : Le carré à côté sa séquence uniforme de K points complexes obtenue avec les Fourier descripteurs.

Les coefficients de la transformée de Fourier discrète de cette séquence de nombres complexes sont donnés par :

$$z(u) = \frac{1}{K} \sum_{k=0}^{K-1} s(k) \exp(-j \frac{2\pi uk}{K}) \quad u = 0, 1, \dots, K-1 \quad 4.1$$

Les coefficients complexes de  $z(u)$  sont les Fourier descripteurs.

La figure 4.3 présente un exemple de reconstruction d'un carré avec un nombre de coefficients allant de 2 à 62. On voit bien que plus le nombre de coefficients est élevé plus la reconstruction est proche de l'original.

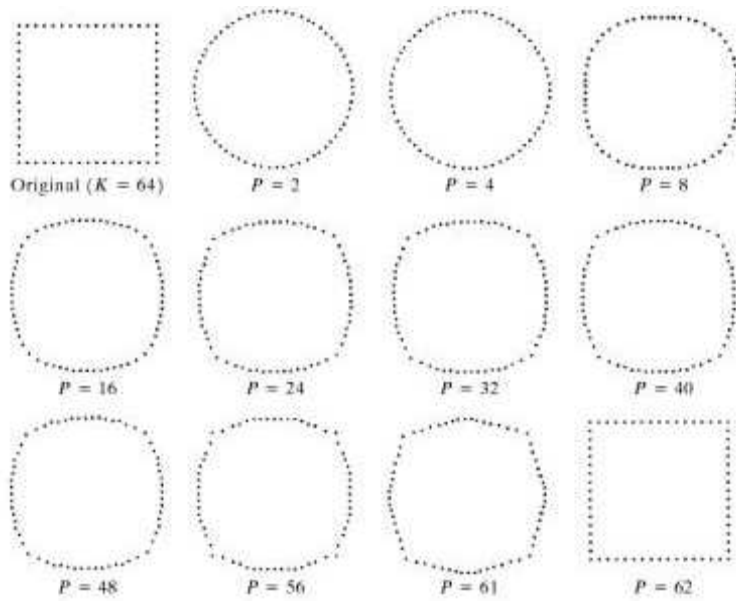


Figure 4.3 : Exemple de reconstruction de contour à partir des Fourier descripteurs P le nombre de coefficients utilisés dans le processus de reconstruction.

On peut reconstruire le contour à partir de la transformée de Fourier inverse :

$$s'(k) = \frac{1}{K} \sum_{u=0}^{P-1} a(u) \exp(j \frac{2\pi u k}{K}) \quad k = 0, 1, \dots, K-1 \quad 4.2$$

Afin d'implémenter les descripteurs de Fourier, il faut que l'image doit être binarisée (Image binarization) par un seuillage (méthode décrite dans 2.4 Binarisation -seuillage) .On se retrouve donc avec une image avec seulement deux valeurs de pixels : soit 1 pour le blanc soit 0 pour le noir.

Le tracé de contour est une fonction qui renvoie les coordonnées X et Y de chaque pixel du contour du caractère à reconnaître. Le tracé du contour commence à partir du pixel se situant à l'extrémité gauche du contour lorsqu'il est analysé dans le sens des aiguilles d'une montre. Toutes les coordonnées sont sauvegardées dans une matrice 2xN. N est le nombre total de pixels formant le contour. La première ligne de la matrice représente les valeurs X, la deuxième les valeurs de Y.

Pour calculer la transformée de Fourier de cette série de nombres complexes, il est nécessaire de rééchantillonner uniformément les pixels du contour de l'image de sorte à ce qu'ils soient au nombre d'une puissance de 2. Cette étape compte le nombre de pixels formant le contour de l'objet et le rééchantillonner à un nombre fixe , dans notre cas le nombre vaut 42 et les coordonnées des pixels rééchantillonnées sont sauvegardées dans des matrices 2x42.

Les Fourier descripteurs permettent de passer d'un problème à 2-D à un problème à 1-D, tout comme le code chaîne. Les Fourier descripteurs les plus importants peuvent être utilisés comme base pour la classification de différents types de contour.

### 4.3 Invariants de forme régions

#### 4.3.1 Moments de Zernike

Les moments Zernike sont définis comme des polynômes complexes qui forment un ensemble orthogonal complet  $\{V_{nm}(x, y)\}$ , en coordonnées polaires. Présenté par Zernike (1934), cet ensemble polynomial est défini par l'équation (4.3)

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) R_{nm}(\rho) \exp(im\theta) \quad 4.3$$

où  $n$  est un entier positif ou nul ;  $m$  est un entier soumis à des contraintes :  $n - m$  est constant et  $n - m \geq 0$  ;  $r$  est la longueur du vecteur entre l'origine et le pixel de coordonnées  $(x, y)$  ;  $\theta$  est l'angle entre le vecteur  $\rho$  et l'axe des  $x$  mesuré dans le sens des aiguilles d'une montre ;  $(\rho)_{nm} R$  est un polynôme radial définit de la façon suivante :

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \quad 4.4$$

Le moment de Zernike d'ordre  $n$  de répétition  $m$  pour la fonction  $f(x, y)$  est définit comme :

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x, y) V_{nm}^*(x, y) dx dy \quad 4.5$$

ou  $V_{nm}(x, y) = V_{n, -m}(x, y)$  .

Supposons que tous les moments de Zernike  $A_{nm}$  de  $f(x, y)$  jusqu'à l'ordre  $N$  sont connus. D'après les propriétés d'orthogonalités des moments de Zernike, on peut reconstruire l'image de base à partir de ces  $N$  moments à l'aide de la formule suivante.

$$f'(x, y) = \sum_{n=0}^N \sum_m A_{nm} V_{nm}(x, y) \quad 4.6$$

Les moments Zernike sont bien connus pour être invariants à la rotation.

Une normalisation d'image est nécessaire pour rendre ces moments invariants à la translation et au facteur d'échelle.

#### 4.3.1.1 Effet de la rotation

Dans notre base d'image (Figure 4.4), pour tester l'invariance des moments  $A_{nm}$ . Nous calculons comme exemple les 5 premiers invariants de Zernike.

Nous avons utilisé pour la reconnaissance seulement les invariants qui possèdent  $m=0$ , car ils sont moins sensibles au bruit.

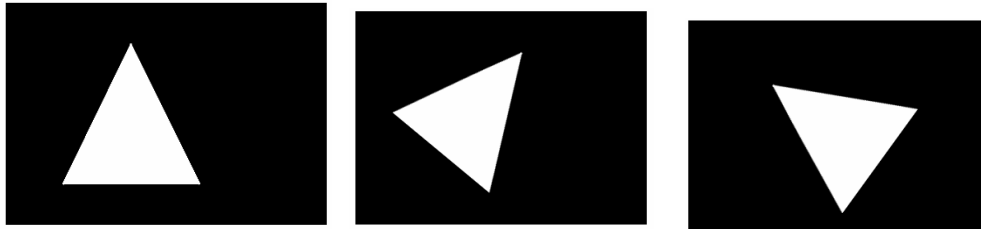


Figure 4.4 : Exemple d'images appliquées.

#### 4.3.1.2 Effet d'échelle

La taille de l'échelle spatiale d'une image peut être obtenue en modifiant les coordonnées cartésiennes de l'image source:

$$x_i = a_x x'_i \quad 4.7$$

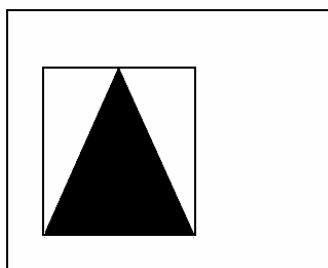
$$y_j = a_y y'_j \quad 4.8$$

Où  $a_x$  et  $a_y$  sont des valeurs positives constantes, appelées facteur d'échelle.

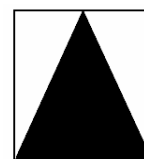
Si  $a_x$  et  $a_y$  sont chacun plus grand que l'unité, le calcul des adresses à partir des équations (4.7-4.8) conduira à un grossissement. Réciproquement, si  $a_x$  et  $a_y$  sont chacun moins que l'unité, le résultat est une réduction. Pour améliorer le temps de calcul de ces primitives invariantes, les objets ont été amenés à une taille égale à 42x42 pixels.

$x'_i$ ,  $y'_j$  sont les pixels source, représentant les positions des objets dans le champ de vision.

Extraction de l'objet de l'image originale en utilisant la méthode de rectangle.



a.Sélection de l'objet par la méthode de rectangle.



b.Objet sélectionné à une taille de 42x42

Figure 4.5 : Extraction de l'objet dans l'image

Calculer les deux facteurs d'agrandissement et (ou) de réduction  $a_x$  et  $a_y$  pour changer la taille de l'objet à une taille égale 42x42 pixels

$$a_x = \frac{\text{largeur de rectangle}}{42} \quad 4.9$$

$$a_y = \frac{\text{longueur de rectangle}}{42} \quad 4.10$$

### 4.3.2 Moments invariants de Hu

Depuis la théorie des moments algébriques, H. Ming-Kuel a extrait une famille de sept invariants par similitude plane (c'est-à-dire rotation, translation et changement d'échelle) appelés invariants de Hu [39]. Ces invariants découlent de combinaisons non linéaires des moments géométriques centrés, réduits, d'ordre inférieur à trois, extraits de l'image.

Les moments géométriques diffèrent des moments statistiques car ils font intervenir la position du pixel dans l'image. En associant le niveau de gris d'un point de l'image à la masse d'un corps en chaque point, on peut reprendre le même formalisme pour décrire la répartition des niveaux de gris dans un objet. Dans le cas d'un objet binaire où  $g(k,l)$  prend la valeur « 1 » à l'intérieur (si le point appartient à l'objet) et « 0 » à l'extérieur, les différents moments fournissent des informations importantes concernant l'arrangement spatial des points de l'objet [40].

Pour une image numérisée et discrète de taille  $K \times L$ , alors  $g(k,l)$  est la fonction décrivant le contenu de l'image en chacun de ses points de coordonnée  $k,l$  ;  $i$  et  $j$  étant deux entiers naturels quelconques :

$M_{ij}$  est la définition du moment géométrique d'ordre  $(i+j)$  avec la répétition  $|i - j|$

$$M_{ij} = \sum_{k=1}^K \sum_{l=1}^L (K)^i (L)^j g(k,l) \quad 4.11$$

où  $g(k,l)$  est le niveau de gris de l'image définie.  $(k)_i (l)_j$  est la base

Les moments géométriques sont particulièrement adaptés pour traiter les changements d'échelle. Prenons 2 images,  $f$  et  $g$  représentant des objets identiques à un facteur d'échelle près, c-à-dire  $g(x,y) = f(\frac{x}{\alpha}, \frac{y}{\alpha})$ , alors la relation entre les moments centrés s'écrit :

$$\forall (p,q) \in \mathbb{N} * \mathbb{N}, \mu_g(p,q) = \alpha^\gamma \mu_f(p,q), \text{ avec } \gamma = p + q + 2. \quad 4.12$$



Pour obtenir l'invariance par rotation Hu [39] a construit un ensemble de 7 descripteurs appelés invariants de Hu. Les invariants de Hu sont devenus des outils classiques en reconnaissance d'objets depuis les 30 dernières années. Ces descripteurs sont construits à partir de combinaisons non-linéaires des moments géométriques :

Les moments invariants de Hu prennent en considération tous les pixels de l'image ce qui induit un temps de calcul des algorithmes autrement plus élevé que pour les invariants contours. Cet aspect est d'autant plus marqué que le nombre de pixels de l'image, donc la résolution, est élevée.

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{21}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \\
\phi_6 &= (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) \\
&\quad - (\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)
\end{aligned}$$

Les six premiers moments sont invariants aux translations et aux changements, problématique quand il s'agit de reconnaître des images miroirs.

Ces descripteurs sont invariants par rotation, translation et rapport d'échelle mais ne constituent pas une description invariante par rotation complète du plan. Beaucoup de travaux effectués depuis ont cherché à améliorer ces invariants et tenter de les généraliser, mais aussi à les utiliser dans des domaines variés [41].

## 4.4 La corrélation

### 4.4.1 Définition

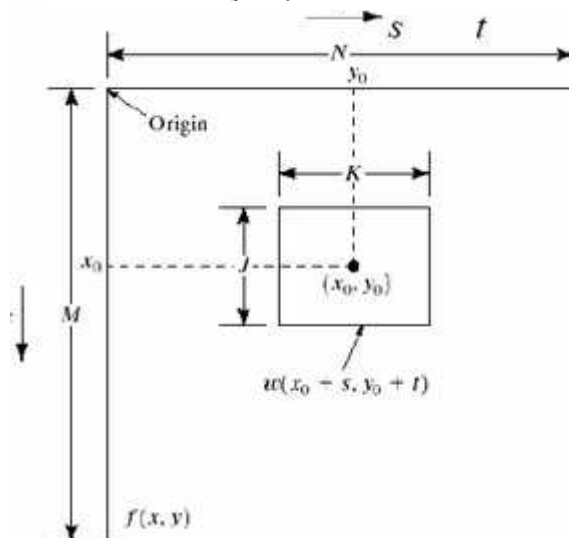
La corrélation est une méthode efficace et rapide pour la localisation de certaines formes géométriques. Les méthodes de reconnaissance de forme proposées par Hough ou Freeman sont des méthodes d'étude morphologique, mais elles nécessitent un temps de calcul important dans le cas des formes géométriques complexes (Hough) ou des contours discontinus (Freeman).

Son principe est très simple et consiste à comparer l'objet à reconnaître (la cible), en général figurant dans une scène et des objets prédéterminants (base de connaissances).

L'algorithme de corrélation est une méthode efficace qui permet de localiser une forme bien précise en corrélant un contour type avec les contours détectés dans l'image (image de référence). L'algorithme de corrélation est sensible au changement de l'échelle, mais reste assez compétitif du fait que la méthode de détection de contour de Canny nous permet de ressortir des contours assez larges qui conviennent parfaitement à la localisation par corrélation.

La corrélation entre  $f(x,y)$  et  $w(x,y)$  est :

$$c(x, y) = \sum_s \sum_t f(s, t) w(x+s, y+t) \quad 4.13$$



Problème de la rotation

- Soit on doit "tourner" l'objet, si certaines informations permettent de connaître l'orientation.

- Soit on doit "tourner"  $w(x,y)$  par petits incréments et tester toutes l'image plusieurs fois.

Problème du facteur d'échelle

- Soit on a un indice quant à la taille de l'objet.
- Soit on doit essayer de corrélér l'objet à différentes échelles.

La reconnaissance des objets implique l'indentification correcte d'un objet indépendamment de la rotation, du facteur d'échelle et de la translation. Un des moyens les plus fondamentaux de détection d'objet dans un champ d'image est par la correspondance de modèle (template ), qui est stocké sous la forme d'une matrice de pixels noirs et blancs, comparée à tous les objets inconnues dans le champ de l'image . Si la correspondance entre un objet inconnu et le modèle est suffisamment étroite, l'objet inconnu est désigné comme le modèle idéal. (Figure 4.6).

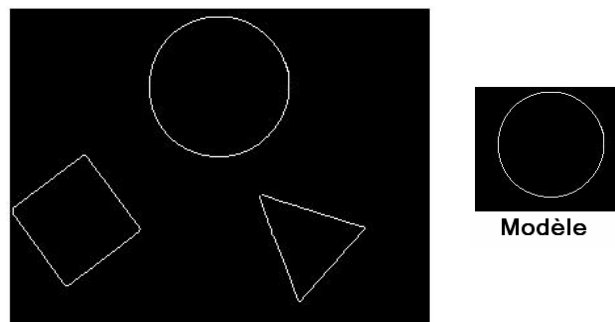
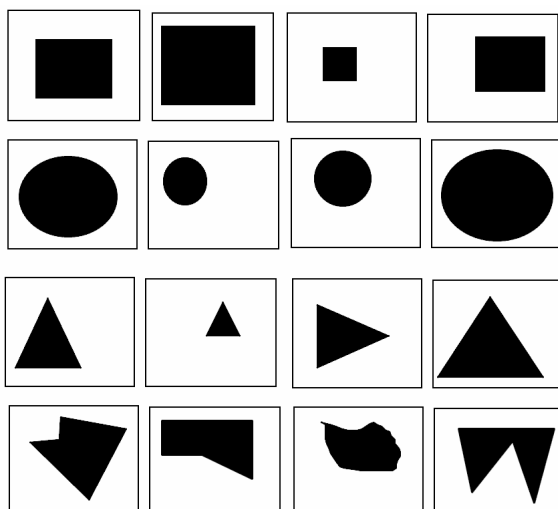


Figure 4.6 Ensembles d'objets

Base de connaissances



Le modèle parcourt séquentiellement l'image et la région commune entre le modèle et le champ d'image est comparé pour la similitude.

Les résultats obtenus à travers l'opération de prétraitement sont des objets séparés avec des orientations et des positions inconnues, et pour appliquer la méthode de la correspondance de modèle sans parcourir le modèle sur le champ d'image, la forme de l'objet est représentée par le centre de gravité, où le modèle est comparé avec la forme à reconnaître.

## 4.5 Caractéristiques géométriques d'un objet

Ce sont des mesures qui peuvent être utilisées pour décrire un objet, nous les obtenons à partir des pixels qui composent la région de l'objet ou les pixels déduits à partir du contour de l'objet.

**Surface :** La zone d'un objet dans une image binaire est égale à la somme de tous des points correspondants à l'objet. Il balaye à travers toutes les colonnes C et toutes les lignes R de l'image la somme des pixels ajoutés à l'objet:

$$A = \sum_{x=0}^C \sum_{y=0}^R f(x, y) \quad 4.14$$

**Périmètre -.** La formule est semblable à la zone, seuls les pixels sont ajoutés au contour de l'objet. Un pixel est un contour (voisinage N4).

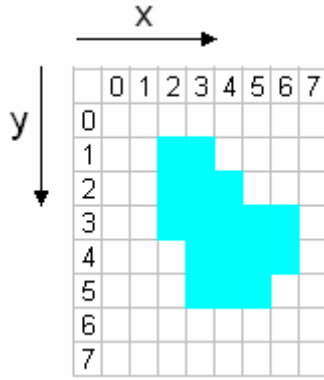
$$P = \sum_{x=0}^C \sum_{y=0}^R f(x, y) \quad \forall f(x, y) \quad \text{avec le pixel } N_4(f(x, y)) = 0 \quad 4.15$$

**Centre de gravité :** Est un point géométrique par rapport au centre de l'objet, les coordonnées de ce point sont calculées en utilisant les formules suivantes:

$$\bar{x} = \frac{\sum_{x=0}^C \sum_{y=0}^R x * f(x, y)}{A} \quad 4.16$$

$$\bar{y} = \frac{\sum_{x=0}^C \sum_{y=0}^R y * f(x, y)}{A} \quad 4.17$$

Par exemple: Nous voulons calculer la surface, le périmètre et le centre de gravité de l'objet suivant



En appliquant la formule 1 nous obtenons le calcul de la surface:

$$\text{Surface} = \sum_{x=0}^C \sum_{y=0}^R f(x, y) = 3 + 5 + 4 + 3 + 2 = 17$$

Le calcul du périmètre de l'image qui détermine la frontière du pixel du Voisinage N4:

$$\text{Périmètre} = 12 \quad \forall f(x, y) \quad \text{avec le pixel } N_4(f(x, y)) = 0$$

Si l'on applique les formules 4.16 et 4.17 le calcul du centre de gravité est le suivant:

$$\bar{x} = \frac{\sum_{x=0}^C \sum_{y=0}^R x * f(x, y)}{A} = \frac{6 + 15 + 16 + 15 + 12}{17} = \frac{64}{17} = 3.76$$

$$\bar{y} = \frac{\sum_{x=0}^C \sum_{y=0}^R y * f(x, y)}{A} = \frac{6 + 15 + 14 + 12 + 7}{17} = \frac{54}{17} = 3.18$$

On arrondi les valeurs obtenues pour déterminer les coordonnées du centre de gravité donc nous aurons : C (4,3).

Facteur de compacité (FC) : Cette fonction est très importante et largement utilisée pour la classification. Elle est définie comme suit :

$P^2$  est le périmètre carré divisé par  $4\pi$  fois la surface de l'objet et compare ainsi la forme de l'objet avec un cercle.

$$FC = \frac{P^2}{4\pi(A)}$$

Dans le tableau (4.1), nous observons que la valeur du facteur de compacité  $FC = 1$  pour les objets circulaires, un peu plus de "1" pour les objets carrés et grandes valeurs obtenues pour les objets autres que du triangle.

L'importance de cette fonction est qu'elle est invariante aux translations, rotations et changements d'échelle.



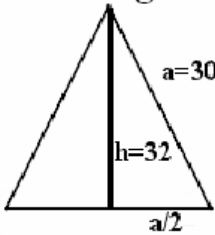
Forme	Mésures	Facteur de compacité
<b>Cercle</b>  $r = 1$	Circonférence = $2\pi r = 6.28$ Surface = $\pi r^2$	$FC = \frac{(6.28)^2}{4(3.1416)(3.1416)} = \frac{39.44}{39.45} \approx 1$
<b>Carré</b>  5	Périmètre = 20 Surface = 25	$FC = \frac{(20)^2}{4(3.1416)(25)} = \frac{400}{314.16} = 1.27$
<b>Triangle</b>  $a=30$ $h=32$ $a/2$	$P = 3a$ $h = \frac{\sqrt{3}}{2} a$ Périmètre = 90 Surface = $a^2 \frac{\sqrt{3}}{4} = 389.7$	$FC = \frac{(90)^2}{4(3.1416)(389.7)} = \frac{8100}{4897.12} = 1.65$

Tableau 4.1 : Calcul de facteur de compacité des différentes formes

## 4.6 Conclusion

La reconnaissance est le processus qui attribue une étiquette à un objet sur la base des informations fournies par des descripteurs. L'interprétation conduit à attribuer une signification à l'ensemble d'objets reconnus. Dans ce chapitre nous avons présenté quelques caractéristiques géométriques d'un objet dans une image, Son principe est très simple et consiste à comparer l'objet à reconnaître (la cible), en général figurant dans une scène et des objets prédéterminants (base de connaissances) par le calcul de centre de gravité de l'objet, on améliorant le temps du calcul et le taux de reconnaissance.

## Mise en œuvre du processus

*Après avoir décrit, les étapes préliminaires du processus de reconnaissance d'objets dans l'image, succèdera la mise en œuvre du processus dans le domaine industriel. Les applications les plus simples peuvent parfaitement être mise en œuvre à l'aide de matériel standard : Webcam , PC. Cet équipement de base permet une mise en œuvre logicielle de l'application envisagée, il a l'avantage considérable d'être flexible et de pouvoir s'adapter aux évolutions des techniques de traitement. Dans ce chapitre nous avons pu intégrer une application sous une seule interface graphique lié par une camera des outils de traitement d'images en temps réel ainsi une liaison directe par le port LPT1 vers un API (Automate Programmable Industriel) pour une opération de tri et de contrôle de qualité.*

### 5.1 Introduction

La dernière étape du processus de développement concerne l'implémentation de l'application en fonction des technologies choisies. Dans ce cadre, nous avons appliqué ce projet en deux phases, la première phase qui porte sur le développement d'une application permettant d'intégrer sous une seule interface graphique la manipulation de la caméra et d'utiliser des outils de traitement d'images en temps réel. Cette application sera développée sous MATLAB à l'aide du toolbox GUIDE, la deuxième phase permet à l'utilisateur de visualiser l'application par des outils de vision industrielle.

Nous commencerons, tout d'abord, par une présentation de l'environnement matériel et logiciel utilisés pour le développement de notre projet.

### 5.2 Environnement du travail

Dans cette section, nous présenterons l'environnement matériel et logiciel de notre travail.

#### 5.2.1 Environnement matériel

Afin de mener à bien ce projet, il a été mis à notre disposition un ensemble de matériels :

- Un Micro-ordinateur
- Un Convoyeur
- Une camera
- Un Automate
- Capteurs

## 5.2.2 Environnement logiciel

### 5.2.2.1 Langage de développement : MATLAB 7.9.0 (R2009b)

#### a) Qu'est ce que MATLAB ?

MatLab est un système interactif de programmation scientifique, pour le calcul numérique et la visualisation graphique. Développé à l'origine pour le calcul matriciel (le nom MatLab est dérivée de cette représentation MatLab = Matrix Laboratory). Il contient des bibliothèques spécialisées (toolbox) qui répondent à des besoins spécifiques : analyse numérique, traitement du signal, traitement de l'image, etc.

MatLab est un logiciel qui permet de faire des calculs mathématiques et numériques, et non un logiciel de calcul formel et symbolique comme Maple.. Matlab connaît un grand nombre d'opérations ou de fonctions mathématiques : fonctions usuelles, calcul matriciel, fonctions plus spécifiques du signal (FFT, etc).

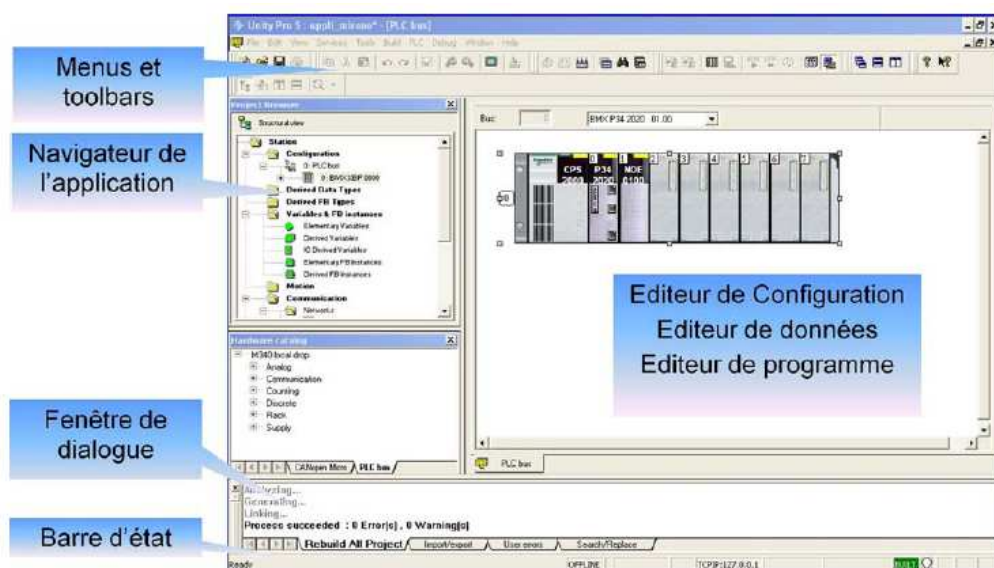
### 5.2.2.2 Logiciel de conception

#### 1- Logiciel unity Pro

##### a) Présentation du logiciel unity Pro

Unity pro est le logiciel de programmation des gammes d'automates :Modicon M340 ;Premium ; Atrium et quantum : Il supporte les cinq langages de norme IEC 6113-3 qui sont : langage à blocs fonction (FBD), Langage à contact (LD), Diagramme fonctionnel en séquence(SFC) , liste d'instructions (IL) et littéral structuré (ST).

##### b) L'interface Utilisateur d'Unity PRO





La création d'un projet sous Unity Pro passe par plusieurs étapes afin de faciliter la tâche d'automatisation. Ces étapes sont :

- 1) Configuration matérielle.
- 2) Configuration du réseau.
- 3) Définition et édition des variables (variables élémentaires, structures, tableaux...)
- 4) Structurer l'application sous forme de tâches, de sections et de modules fonctionnels.
- 5) Edition du programme dans les cinq langages de programmation mentionnés précédemment.
- 6) Compilation et simulation des programmes avant leur implémentation dans l'automate.

La première chose à faire est simuler le programme utilisateur et son transfert vers l'automate pour de générer (compiler) le programme et ceci afin de vérifier qu'il n'y a pas d'erreurs aussi bien dans la configuration matérielle que dans la syntaxe du programme. Puis, le transfert du projet vers l'automate.

Durant cette étape, nous avons vu que les automates programmables Schneider sont dotés d'un logiciel de programmation très performant qui est Unity Pro intégrant les cinq langages de programmation normalisés et possédant une interface graphique interactive et simple à utiliser. Ce logiciel sera exploité par la suite pour la création des programmes de notre application.

## **2- Logiciel SCADA Vijeo Citect**

### **a) Introduction**

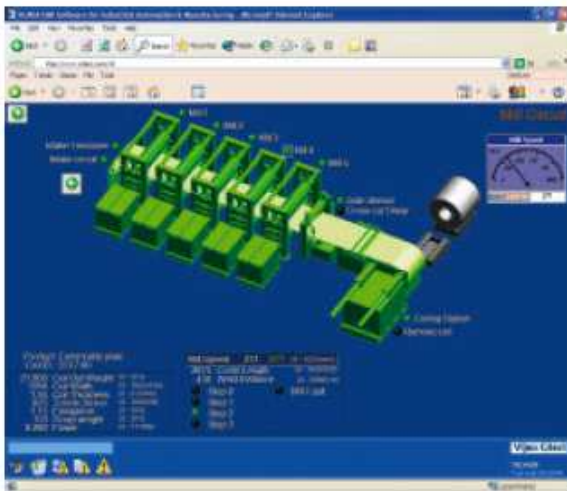
Vijeo Citect est un logiciel pour un système d'acquisition et de contrôle des données (SCADA ou Supervisory Control And Data Acquisition), ce logiciel permet de gérer et de surveiller des processus de fabrication, de distribution et de gestion d'installations.

Il est à noter que le logiciel Vijeo Citect est un logiciel polyvalent pouvant être utilisé aussi bien dans l'industrie que pour la Gestion Technique du Bâtiment, son utilisation en GTB permet de gérer l'ensemble de l'immeuble, surveiller la consommation d'énergie et les différentes alarmes, commander l'ouverture ou la fermeture des portes, commander l'éclairage, modifier la consigne de température pour les systèmes CVC..., enfin bref, tout ce qu'on peut imaginer.

## b) Caractéristiques et avantages du logiciel Vijeo Citect

Le logiciel Vijeo Citect dispose de plusieurs caractéristiques faisant de lui une solution de choix pour la supervision des systèmes de GTB, parmi ces caractéristiques on peut citer les suivantes :

- 1- Analyse et présentation de tendances historiques et en temps réels sous forme de courbes et d'histogrammes (tel que la consommation d'énergie électrique des différents secteurs d'un bâtiment)
- 2- Gestion des alarmes (alarmes relatives aux systèmes de sécurité (anti-intrusion), alarmes déclenchées par des pannes d'équipements tel que les équipements de CVC), de plus le logiciel permet aussi la génération de rapports donnant l'état des équipements ainsi que les différents événements enregistrés (alarmes, pannes,...etc.).
- 3- Deux langages de programmation Cicode (langage évolué) et CitectVBA.
- 4- Eléments graphiques contenant Génie, Super Génie et objets ActiveX (ces éléments vont être explicités plus loin dans ce chapitre).
- 5- Possibilité d'utiliser des *grappes (ou Cluster)* permettant de regrouper plusieurs systèmes de contrôle en un seul système, ce qui permet de diminuer les couts, ainsi, si on utilise plusieurs systèmes de supervision pour des sites différents, on peut utiliser un Cluster pour les regrouper un une seule station permettant de superviser les différents sites et donnant ainsi la possibilité de comparer les données et tendances des différents sites.



### 5.3 Interface de l'application MATLAB

Cette interface présente le menu principal qui permet une sorte de passerelle pour accéder à l'application de reconnaissance d'objets dans une image



Figure 5.1: Menu principal

#### 5.3.1 Reconnaissance d'objets

Cette interface montre la capture de la vidéo et images. La caméra utilisée est la Webcam qui transmet via une connexion USB les images au PC.

Tout d'abord, nous saisissons l'image à traiter soit à partir de la Webcam, si elle est connectée si non on aperçoit un message d'erreur que la Webcam n'est pas connectée donc l'image capturée est automatiquement se trouve dans le répertoire de travail.

Le programme permet le passage d'une image vidéo vers une image fixe qu'on va par la suite traiter et analyser. Pour capturer une image il faut cliquer sur le bouton capture.

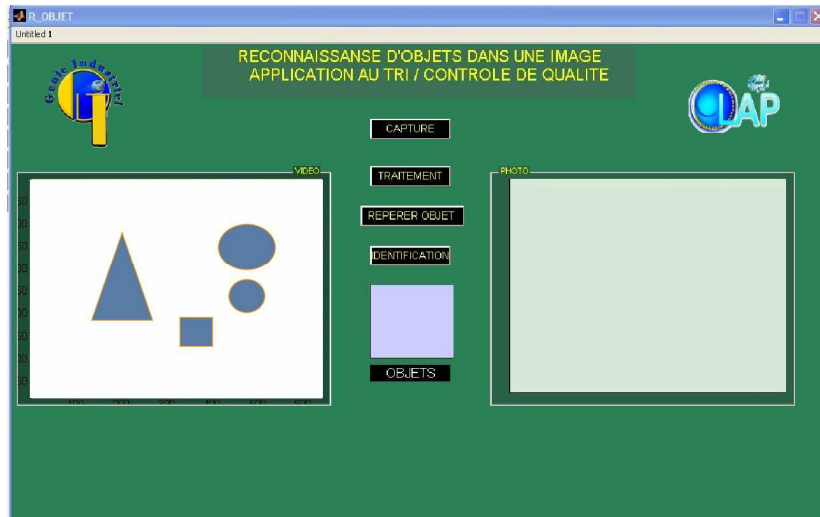


Figure 5.2: Interface Acquisition vidéo

Le bouton **capture** transmet la vidéo vers une image fixe, cette image couleur contient des objets géométriques.

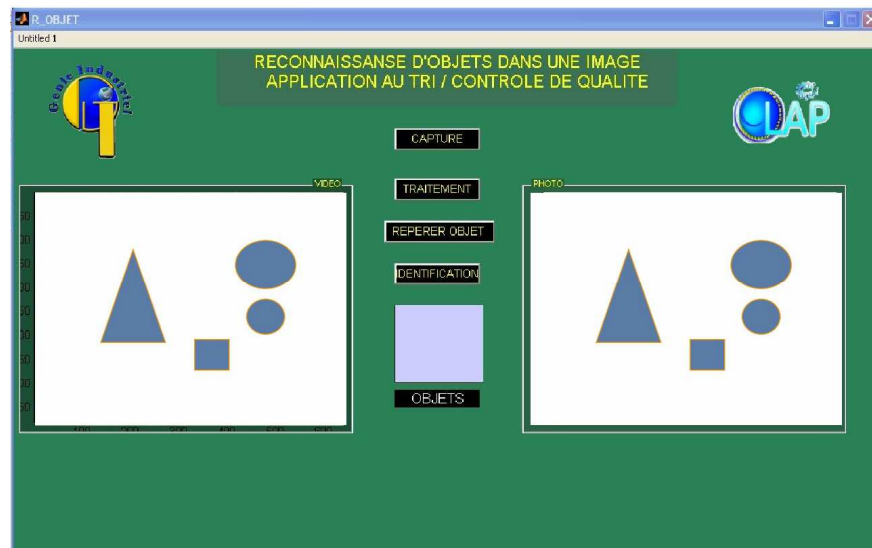


Figure 5.3: Interface Acquisition image

Ensuite le bouton **traitement** est un programme qui exécute tout les outils de traitements d'images qu'on a abordés tout en long de notre travail par une conversion d'image couleur en niveau de gris ensuite un seuillage automatique puis nettoiera l'image de tous les objets inutiles. Des opérations morphologiques mathématiques (érosion, dilatation, ouverture, fermeture). Enfin on isolera les différents objets et on les nommera (suivant le nombre de côte qui constituent l'objet) et on calcule la surface de chaque objet. Grâce à ce programme, on peut réaliser toute une série de traitements sur l'image.

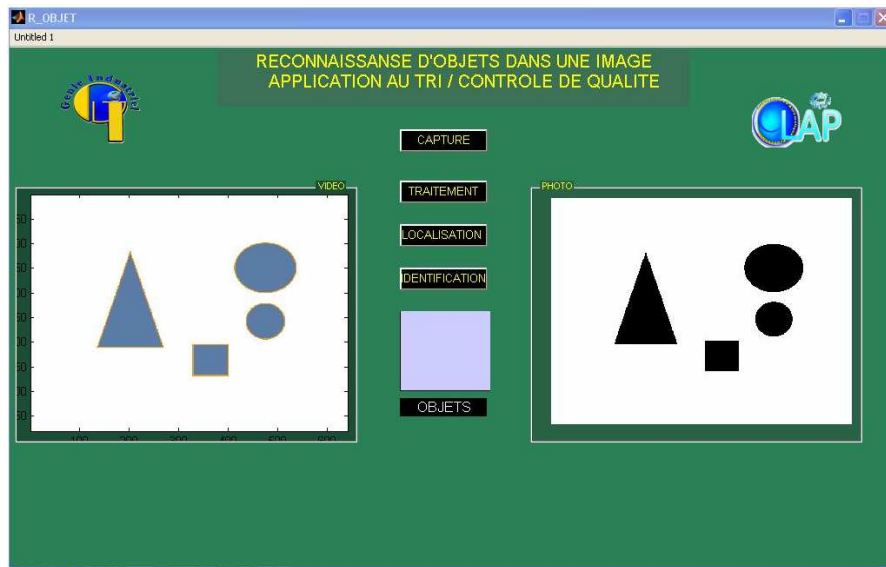


Figure 5.4: Interface traitement image

Le bouton **Localisation objet** est un programme qui compte le nombre d'objets dans l'image puis sauvegardé dans une matrice qui va calculer les limites de chaque objet dans toutes les directions , puis le calcul du centre de gravité de chaque objet trouvé par un marquage \* (astérisque) ensuite on sauvegarde l'objet dans une nouvelle matrice.

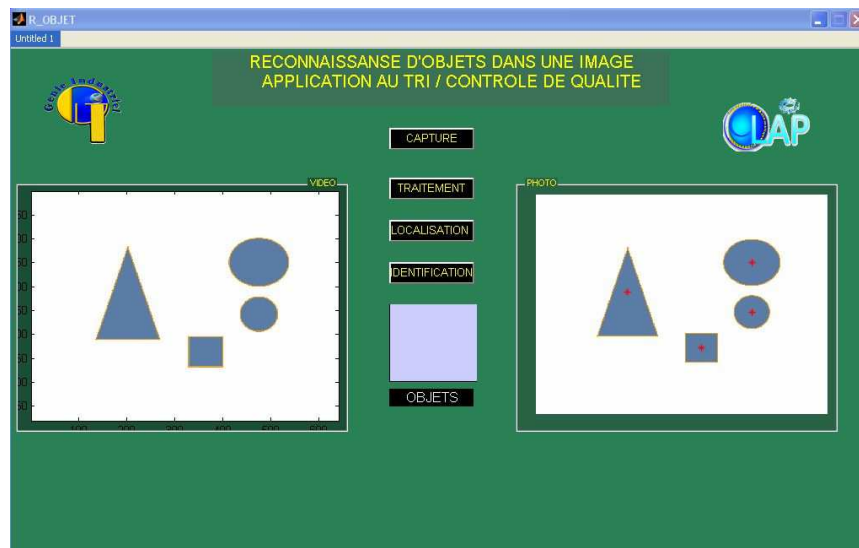


Figure 5.5: Interface localisation objets

Le bouton **Identification** est un programme de corrélation et comptage d'objets, cette séquence de programme affecte chaque objet trouvé, un préfixe qui se trouve dans un modèle (templates)

et après vérification dans la base de connaissances on identifiant l'objet puis on désigne l'objet par son nom.

TRI= Triangle

CER= Cercle

CAR= Carré

INC= Inconnu

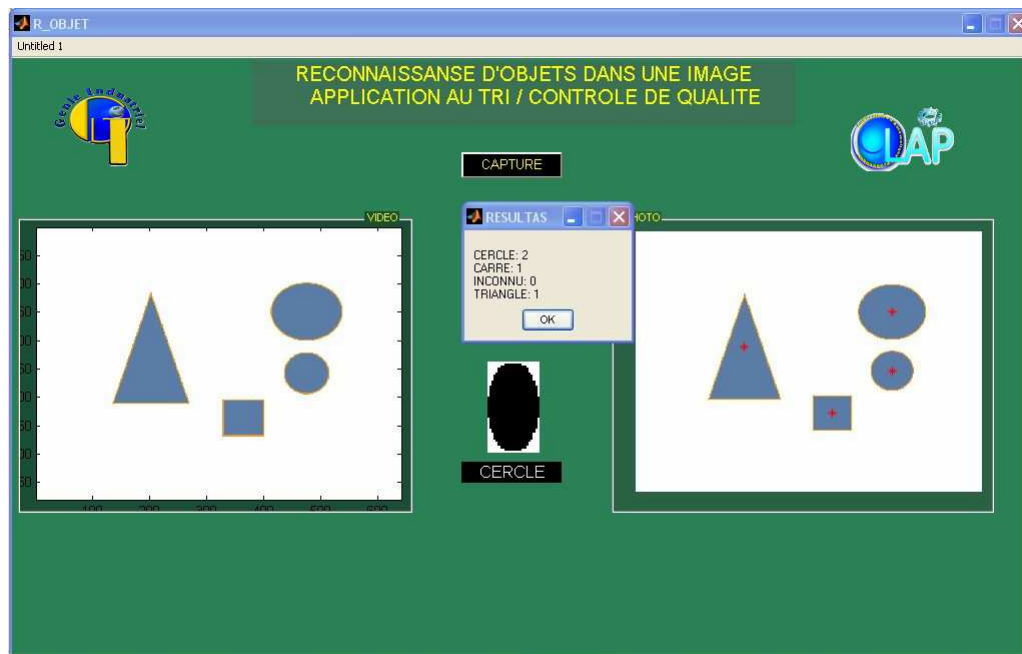


Figure 5.6: Interface Identification d'objets

Enfin nous avons pu réaliser un programme qui peut reconnaître des formes géométriques dans une image pour une opération de tri et en même temps un contrôle de qualité suivant un temps d'exécution rapide pour une opération industrielle.

Afin de perfectionner notre Application MATLAB pour une fin industrielle, nous avons pensé à l'intégrée dans une interface de vision homme machine qui facilitera une supervision en temps réel.

### 5.3.2 Reconnaissance d'objets par Automate

Une deuxième réalisation consiste à envoyer des signaux par le biais du port parallèle (LPT1) à un automate qui actionne les bras poussoirs selon les caractéristiques de l'objet.

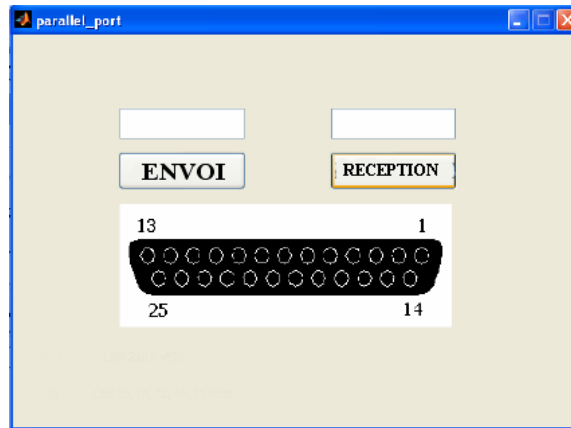


Figure 5.7: Port parallèle (LPT1) par Matlab

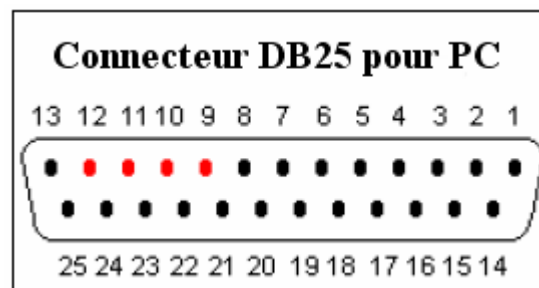


Figure 5.8: Numérotation de pins (LPT1)

A la fin de notre application de reconnaissance d'objets dans une image on définit trois variables ob1, ob2 et ob3 comme suit :

Si le premier objet est un carré alors ob1= carré

Si le premier objet est un cercle alors ob1= cercle

Si le premier objet est un triangle alors ob1= triangle

Si le premier objet est ni carré, ni cercle et ni triangle alors ob1= n\_reconnu

Si le deuxième objet est un carré alors ob2= carré

Si le deuxième objet est un cercle alors ob2= cercle

Si le deuxième objet est un triangle alors ob2= triangle

Si le deuxième objet est ni carré, ni cercle et ni triangle alors ob2= n\_reconnu

S'il y a pas de deuxième objet alors ob2=rien

Si le troisième objet est un carré alors ob3= carré

Si le troisième objet est un cercle alors ob3= cercle

Si le troisième objet est un triangle alors ob3= triangle

Si le troisième objet est ni rectangle, ni cercle et ni triangle alors ob3= n\_reconnu

S'il y a pas de troisième objet alors ob3=rien

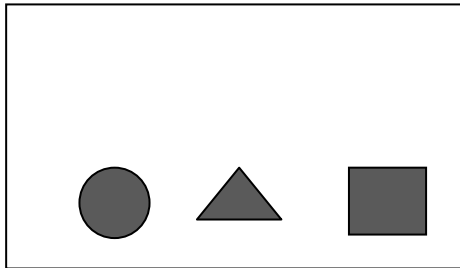
### 5.3.2.1 Communication entre le PC et l'automate

L'objet de cette étape est de communiquer à l'automate le type d'ordre des objets arrivant sur le convoyeur (reconnus par la partie vision).

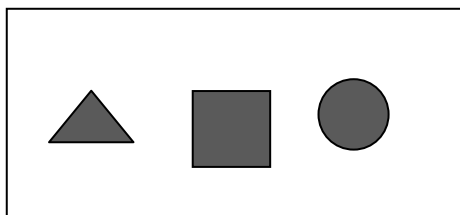
Les quatre lignes D1, D2, D3 et D4 du port parallèle (LPT1) du Micro-ordinateur sont connectées à quatre entrées de l'automate.

- Pour indiquer qu'un objet est carré il faut mettre la ligne D1 à 1 pendant 0.5s.
- Pour indiquer qu'un objet est circulaire il faut mettre la ligne D2 à 1 pendant 0.5s.
- Pour indiquer qu'un objet est triangulaire il faut mettre la ligne D3 à 1 pendant 0.5s.
- Pour indiquer qu'un objet est ni circulaire ni carré et ni triangulaire il faut mettre la ligne D4 à 1 pendant 0.5s.

L'ordre de mise à 1 des quatre lignes correspond à l'ordre de positionnement des objets sur le convoyeur. Les figures suivantes représentent les séquences D1 D2 D3 D4 à envoyer à l'automate pour quatre exemples d'images.

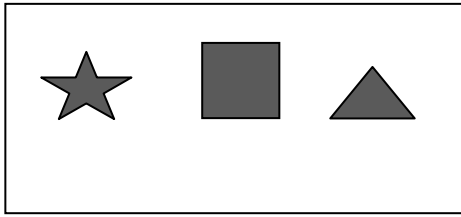


Envoyer pendant 0.5s la séquence 1000  
Puis envoyer pendant 0.5s la séquence 0010  
Puis envoyer pendant 0.5s la séquence 0100  
Puis envoyer la séquence 0000

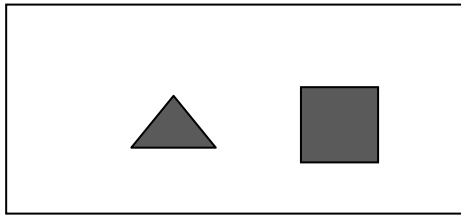


Envoyer pendant 0.5s la séquence 0100  
Puis envoyer pendant 0.5s la séquence 1000  
Puis envoyer pendant 0.5s la séquence 0010  
Puis envoyer la séquence 0000





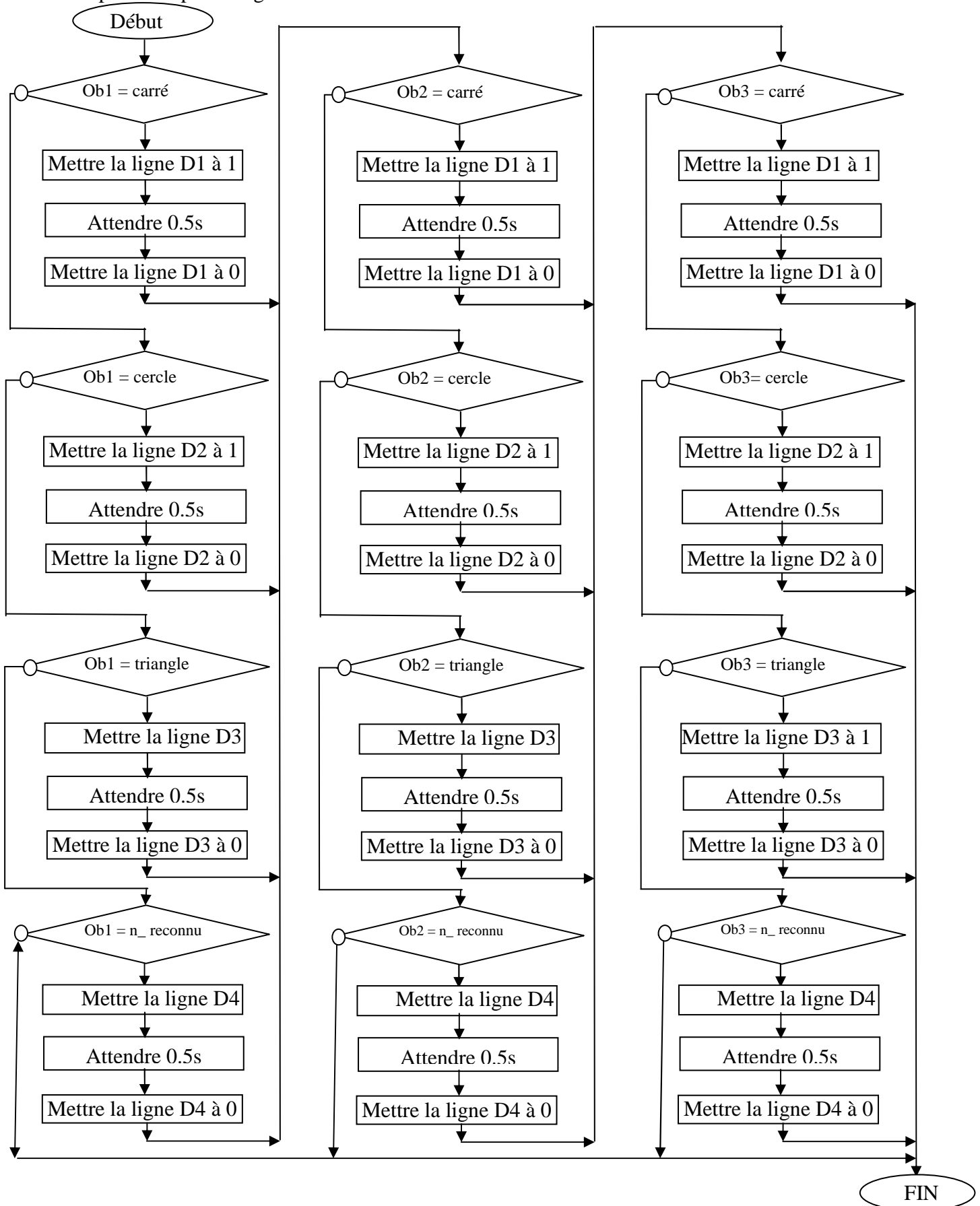
Envoyer pendant 0.5s la séquence 0010  
Puis envoyer pendant 0.5s la séquence 1000  
Puis envoyer pendant 0.5s la séquence 0001  
Puis envoyer la séquence 0000



Envoyer pendant 0.5s la séquence 1000  
Puis envoyer pendant 0.5s la séquence 0010  
Puis envoyer la séquence 0000

### 5.3.2.2 Organigramme du programme de communication entre PC et l'automate

L'organigramme du programme permettant de réaliser la communication entre PC et l'automate est présenté par la figure 5.9 suivante :



## 5.4 Interface Homme-Machine ( IHM)

L'interface homme machine est définie comme étant le moyen matériel ou logiciel permettant à une personne de communiquer avec un programme (configuration ou supervision). Des exemples d'IHM sont les fenêtres, les boutons et menus déroulants.

L'interface est un programme créé pour simplifier l'accès de l'homme à la configuration ou la mise en service d'un matériel configurable par informatique. En effet, le cas le plus fréquent dans le domaine de la vision consiste à avoir une interface visuelle pour la programmation et la configuration.

Dans le cadre de notre travail, une interface est mise en place pour la configuration du système de commande. Pour concrétiser notre application via interface **Vijeo Citect** (figure 5.10), nous allons décrire un enchaînement d'opérations détaillées successives et complémentaires d'un processus de reconnaissance d'objets pour une application de tri et contrôle de qualité

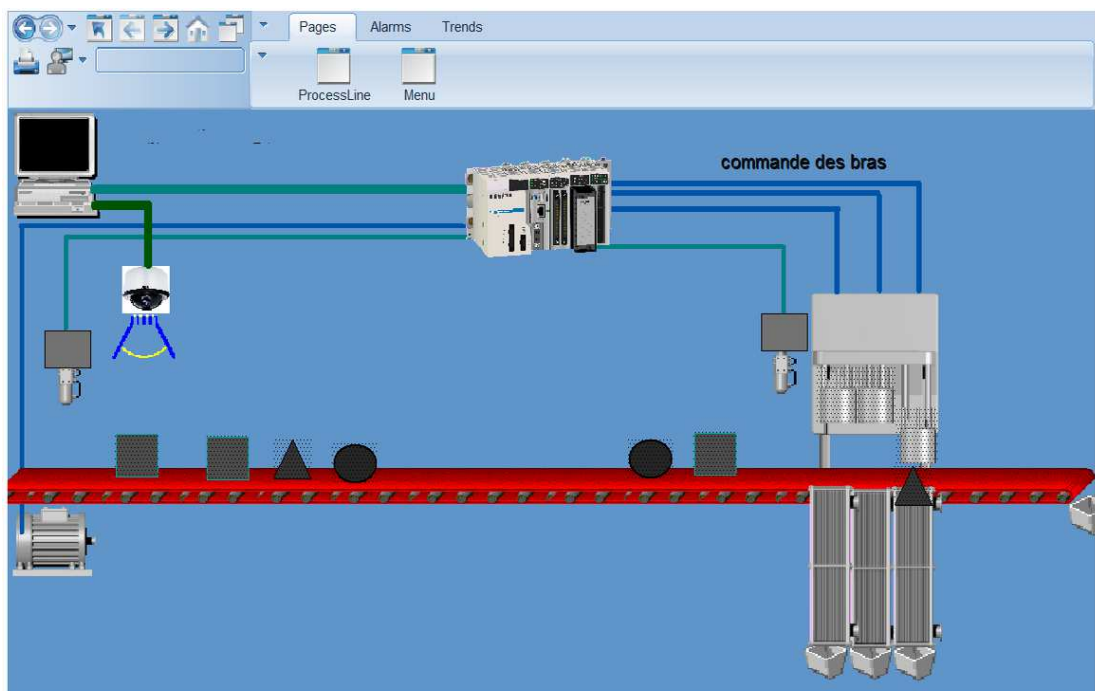


Figure 5.10 : Interface Vijeo Citect

La simulation consiste en :

La détection par le premier capteur concomitant à la caméra, de l'objet entraîne une réaction de l'automate consistant à envoyer un signal au moteur actionnant le convoyeur principal pour un arrêt (BFI 26) permettant à la caméra de saisir l'image. Laquelle image est à son tour envoyée à l'ordinateur chargé d'analyser et reconnaître l'objet de l'image (FBI 31).

Il s'agit en fait de la phase suivante du processus, car c'est à partir de là, que l'automate reçoit l'ordre d'actionner les bras poussoirs.

Cette deuxième opération, initiée par l'écriture d'un Mot correspondant à l'entrée détectée dans un registre type FIFO, se poursuit par l'action du 2<sup>ème</sup> capteur placé avant les bras poussoirs que l'automate actionne selon la valeur du Mot sélectionné dans le registre FIFO.

La commande de l'un de ses bras, pousse l'objet sur son convoyeur spécifique qui le placera dans le bac correspondant (FBI 23, FBI 24, FBI 25).

Dans le cas de non détection de l'un des trois objets sélectionnés, cela entraîne l'absence de signal vers l'automate, épargne l'objet, et le conduit vers le palettiseur des objets non reconnus.

Il est à signaler que tout le processus se complète dans son aspect utilitaire et applicationnel par l'opération de comptage, lui-même commande le changement des bacs, une fois la capacité maximale atteinte (exemple 90 objets). Les détails du programme Automate cité en annexe B

## **5.5 Conclusion**

A terme, cette étude peut conduire au développement d'un outil d'analyse pouvant être par exemple appliqué à l'étude de tri et de contrôle de qualité dans une industrie liées à la fabrication des produits.

## CONCLUSION GENERALE ET PERSPECTIVES

La Vision par Ordinateur est à la base de tout système de vision artificielle qui prend en entrée une ou plusieurs images numériques, et qui effectue des traitements sur ces images afin d'en extraire des informations relatives à la scène observée.

La reconnaissance d'objet dans les images est une discipline qui, bien qu'elle soit encore dans une phase de recherche et de perfectionnement, est utilisée pour de nombreuses applications. Elle permet à un ordinateur de « comprendre » l'image d'où un lien étroit avec l'intelligence artificielle.

Dans ce projet, nous avons tenté de bâtir un système de reconnaissance d'objets dans une image pour une application de tri et de contrôle de qualité. L'objectif est la recherche d'objets dans des images : il s'agira d'image unitaire et non pas de séquences d'images.

Ce projet a été vraiment très intéressant car cela nous a permis d'approfondir et mettre en pratique nos connaissances dans le domaine du traitement d'images mais aussi d'être confronté aux difficultés des images réelles : qualité, bruit, résolution, etc...

Il nous a fallu pas mal de temps pour rechercher les documents nous permettant de mener à terme ce projet, surtout en ce qui concerne l'identification et la localisation des objets dans l'image, le sujet est certes beaucoup trop vaste mais incontournable. Nous sommes confrontés à des difficultés d'acquisition ; puisque nous avons utilisé une simple caméra Webcam qui n'est pas destinée à l'usage industriel, ainsi que l'éclairage qui est l'aspect le plus critique de toute application de vision artificielle, car si l'image est bien éclairée le processus de reconnaissance d'objets peut atteindre de très bon résultats.

L'acquisition d'image, nous permet de récupérer les données d'une image en temps réel tout en visualisant l'image, cette fonction nous donne en retour l'image que l'on exploite sous forme matricielle. Celle-ci montre trois matrices représentant l'image dans les couleurs de base. Notre travail est justifié par une transformation d'image en niveau de gris puisque les algorithmes de traitement d'images sont très coûteux en temps d'application.

Nous avons analysé quelques points très importants dans la reconnaissance d'objets, de la façon d'appliquer toutes sortes de méthodes de filtrage pour éliminer le bruit d'image, nous avons mis une méthode pour la segmentation d'images sur un fond blanc, les objets en noir, c'est une étape de binarisation, nous avons introduit l'approche corrélation qui est une méthode efficace et rapide pour la localisation de certaines formes géométriques. Les méthodes de reconnaissance de forme proposées par Hough ou Freeman sont des méthodes d'étude morphologique, mais elles nécessitent un temps de calcul important dans le cas des formes géométriques complexes ou des

contours discontinus (Freeman). L'algorithme de corrélation est une méthode efficace et simple qui permet de localiser une forme bien précise en corrélant un contour type avec les contours détectés dans l'image (image de référence). L'algorithme de corrélation est sensible au changement de l'échelle, mais reste assez compétitif du fait que la méthode de détection de contour de Canny nous permet de ressortir des contours assez larges qui conviennent parfaitement à la localisation par corrélation.

Dans ce projet, nous nous sommes intéressés aux défauts de forme. Concernant l'identification de défauts par les deux méthodes connues, Hough ou Freeman, l'algorithme de traitement s'applique une fois que l'objet est localisé par corrélation.

En dernier, la phase expérimentale ou l'application par le langage MATLAB pour valider la conception théorique du processus de reconnaissance d'objets. Après cette réalisation, une deuxième réalisation consiste à concrétiser à une fin industrielle un outil d'interface de vision Homme machine (HMI) qui facilitera une supervision en temps réel.

Nous avons montré que, Vijeo Citect constitue une solution complète et homogène dans le domaine industriel.

Nous avons implanté un automate programmable qui servira l'échange des données via l'application Matlab par le port LPT1.

Après plusieurs tests des images acquises par la Webcam et des images obtenues de façons aléatoire, le taux de reconnaissance obtenu (90 %) est très satisfaisant, mais comme dans toutes les applications, des améliorations restent toujours à apporter.

De grandes perspectives sont prévisibles après ce mémoire, comme le fond de la scène (le tapis du convoyeur) qui joue un rôle important en simplifiant les étapes de traitement d'images (telles que la segmentation). Il doit être aussi homogène que possible pour permettre d'identifier facilement les objets. Toute tache dans l'arrière-plan, peut causer des erreurs dans la phase de reconnaissance. Aussi l'un des problèmes les plus importants dans le monde de la vision est l'éclairage, le champ de vision risque d'être perturbé par le changement de la luminosité ambiante, où la moindre fluctuation de la luminosité, engendre des réponses non conformes.

En ce qui concerne la caméra, elle doit d'abord être choisie en fonction de la dynamique de l'objet sur le convoyeur au moment de l'acquisition. Si le convoyeur est à l'arrêt, toutes les caméras conviendront. Si par contre l'objet ne s'arrête pas devant la caméra, il est recommandé de choisir une caméra progressive, capable de synchroniser l'acquisition avec le passage de l'objet tout en gardant sa pleine résolution.

On utilisera une caméra numérique haute résolution (plus chère et plus lente qu'une caméra à résolution standard) lorsqu'elle apporte effectivement un gain en précision nécessaire.

En terme de notre perspective, l'un des axes principaux d'amélioration du résultat de ce projet est d'utiliser des objets couleurs tridimensionnels et d'intégrer deux caméras du type stéréoscopique.

## BIBLIOGRAPHIE

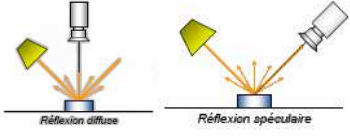
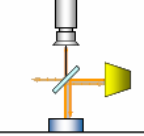
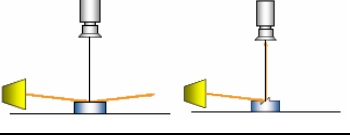
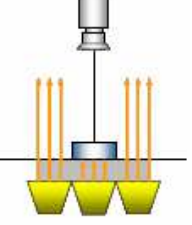
- [1] **Frederic Truchetet**, Introduction a la vision artificielle industrielle, Année 2005.
- [2] **Horaud Radu et Monga Olivier**, Vision par ordinateur : outils fondamentaux INRIA Deuxième édition mai 2011.
- [3] **Umbaugh S. E.**, Computer Vision and Image Processing : A Practical Approach Using Cviptools, Année 1998.
- [4] **Bugea Aurélie**, Attention visuelle multi-echelle, Année 2004.
- [5] Le grand dictionnaire terminologique. Version 3.70. Année 1997.
- [6] **N. Zuech, Marcel Dekker**, Understanding and Applying Machine Vision Deuxième Edition, Année 2000.
- [7] **H. R. Myler**, "Fundamentals of Machine Visions", Vol. TT33, SPIE Press, 1999.
- [8] **Bénallal Mohamed**, Système de calibration de Caméra Localisation de forme polyédrique Par vision monoculaire, Année 2002.
- [9] **Horaud Radu et Monga Olivier**, Vision par ordinateur : outils fondamentaux INRIA Deuxième édition mai 2011.
- [10] **Benoit Alexandre**, Le système visuel humain au secours de la vision par ordinateur Année 2007.
- [11] **Bullier J** Architecture fonctionnelle du système visuel, Année 1998.
- [12] **Beaudot W.** Le traitement neuronal de l'information dans la rétine des vertébrés : Un creuset d'idées pour la vision artificielle, Année 1994.
- [13] **Nowak, Eric**, Reconnaissance de catégories d'objets et d'instances D'objets à l'aide de représentations locales, Mars 2008.
- [14] Brève introduction à la Reconnaissance des Formes Document CNRS "Interaction Homme-Machine", juillet 1997.
- [15] Revue la recherche site [www.larecherche.fr](http://www.larecherche.fr)
- [16] **Fernandez-Maloigne**, Introduction aux techniques de traitement et d'analyse d'images.
- [17] **El Sahmarani Khaled**, Commande multi capteur d'un processus de Fabrication, octobre 2008.
- [18] **L. Busin, N. Vandenbroucke, L. Macairer**, Contrôle d'aspect par vision linéaire couleur, décembre 2008.

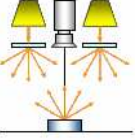
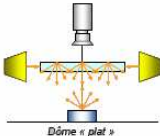






- [19] **Jacky Desachy**, cours informatique analyse d'images version 1.3 janvier 2001.
- [20] **Gonzole Raphael, woods Richard, Eddin steven**, Digital Image Processing using Matlab édition 2004.
- [21] **Gerard Blanchet, Maurice Charbit**, Digital Image Processing using Matlab édition 2006.
- [22] **J. Max, J.L Lacoume**, Méthodes et techniques de traitement du signal, Année 2004.
- [23] Wikipédia
- [24] **Yaovi Gagouy**, Cours de Traitement d'Image, Année 2007-2008.
- [25] **J. W. Tukey**, "Non linear method for smoothing data", in Conf. Rec. EASCON1974, P. 673.
- [26] **M. Nagao, T. Matsuyama**, edge preserving smoothing , CVGIP, 1979, Vol.9, pp. 394-407.
- [27] **J. P. Cocquerez, J. Devars**, "Détection de contours dans les images aériennes: nouveaux opérateurs", Traitement du Signal, Vol. 2, no. 1, pp. 45-65, 1985.
- [28] **Pecaud Christopher**, seuillage et segmentation d'images texturées, Année 2001,
- [29] **J.P. Cocquerez et S. Philipp**, Analyse d'images : filtrage et segmentation Ed. Masson, Année 1995.
- [30] **Etienne Corteel et Jérôme Lavadou**, Comparaison entre les filtres de Tomita/Tsuji et de Nagao/Matsuyama Projet MBAI - Mars 2000.
- [31] **Antoine Manzanera**, cours de Morphologie mathématique. Année2005.
- [32] **Bernd Jahne**, Digital Image Processing , édition 2002.
- [33] **Haytham Elghazel** , Analyse et identifications de bactéries dans les images microscopiques couleur de biofilms, Année 2003.
- [34] **Henri Maitre** , La détection des contours dans les images.
- [35] **Phan Viet Anhe**, Développement d'un module de segmentation pour un système de reconnaissance biométrique basé sur l'iris, novembre 2008.
- [36] **Bellal Se, Mouss L.H et Bensaadi Rafik** , Un processus de reconnaissance d'objets dans une image pour une application de tri ,ICIEM 2012.
- [37] **Stéphane Derrode** , Représentation de formes planes à niveaux de gris par différentes approximations de Fourier-Mellin analytique en vue d'indexation de bases d'images, Année1999.

- [38] **Pratt, William K**, Digital Image Processing, quatrieme edition USA 2007.
- [39] **M.K. Hu**, Visual Pattern Recognition by moment invariants. IRE Trans. Inform. Theory, 8 :179–187, 1962.
- [40] **M. Kunt**, Reconnaissance des formes et analyse de scènes », collection électricité, traitement de l'information : volume 3, année 2000.
- [41] **Olivier Caignart**, Détection d'objets dans les images spot, Année 2003.

Les techniques d'éclairage dans le domaine de vision par le tableau ci dessous :

Techniques d'éclairage	Avantages	Inconvénients
<b>Éclairage frontal</b> -Éclairage directionnel (bright field) 	Avec un bon contraste et une image nette, il permet de réaliser du contrôle de présence et de la mesure • En réflexion spéculaire, il permet de détecter des défauts de surface • En réflexion diffuse, il permet de contrôler les surfaces brillantes de métaux polis ou en plastique	• Génère des contours, des ombres et des reflets • Ne convient pas pour les pièces transparentes ou translucides • Effet de perspective en réflexion spéculaire
<b>Éclairage coaxial (On Axis Lighting (AOL))</b> 	• Même avantage que l'éclairage directionnel • Ne génère ni ombres ni reflets	• Avec un AOL, l'épaisseur du miroir peut produire une image double mais il permet d'éclairer de petite cavité • Nécessite d'avantage de puissance lumineuse
<b>Éclairage rasant (dark field)</b> 	Permet une très bonne détection de défauts sur surfaces planes opaques ou translucides	• Nécessite une source lumineuse très puissante
<b>Rétro-éclairage</b> <b>Éclairage diascopique (backlight)</b> 	• Mise en évidence de la silhouette (contraste maximum entre le fond, blanc et la forme, noire) • Permet le contrôle dimensionnel et l'absence/présence de trous • Permet de contrôler les matériaux transparent et translucide (verre, tissu, circuit imprimé)	• Les détails de la surface sont perdus
<b>Lumière diffuse</b> -Diffuseur et sources diffuses	• Éclairage uniforme • Peu ou pas d'ombres et de reflets • Existe aussi en DAOL (Diffuse	• Contours flous • Faible contraste

 <p>-Dome Illuminator</p>  <p>Dôme « plat »</p>	<p>On Axis Lighting)</p>	
<p><b>Lumière collimatée(ou telecentrique)</b></p>  <p>Lumière collimatée : Les rayons lumineux sont parallèles</p>  <p>Lumière non collimatée : Les rayons lumineux se dispersent</p>	<p>La surface éclairée et l'intensité lumineuse restent les mêmes quelque soit la distance</p> <ul style="list-style-type: none"> <li>• Permet de mettre en évidence les reliefs (creux, bosses, ...)</li> <li>sur des surfaces planes et réfléchissantes</li> <li>• En rétro-éclairage, permet de mettre très nettement en évidence les bords des objets quelque soit leur opacité (pas de phénomène de réfraction)</li> </ul>	<ul style="list-style-type: none"> <li>• Nécessite un dispositif optique</li> <li>• Prix</li> </ul>
<p><b>Lumière focalisée</b></p>  <p>Lumière non focalisée : Les rayons lumineux divergent</p>  <p>Lumière focalisée : Les rayons lumineux convergent</p>	<ul style="list-style-type: none"> <li>• Augmente l'intensité lumineuse émise sur la surface éclairée</li> <li>• Eclaire une partie spécifique d'un objet</li> <li>• Permet l'éclairage structuré</li> <li>• Particulièrement bien adapté à la vision linéaire nécessitant d'éclairer la ligne de visée de la caméra avec une forte intensité lumineuse</li> </ul>	<ul style="list-style-type: none"> <li>• N'éclaire pas la totalité de l'objet</li> </ul>

## PROJET

Projet	Reconnaissance d'objets dans une image application au tri et controle de qualité
Concepteur	Bellal S.E , Mouss L.H , Bensaadi R.
Application	PLC.stu
Version logicielle	Unity Pro XL V5.0
Date de création	2011-2012
Date de dernière modification	
Automate cible	BMX P34 2020 02.10CPU 340-20 Modbus Ethernet

## 0.1 : BMX DDM 16025

### Identification du module :

Réf. commerciale : BMX DDM 16025  
Adresse : 0.1

Désignation : Dig 8I 24 Vdc 8Q Relays  
Symbole :

### Paramètres communs [0-7]

Surveillance alimentation : Actif  
Tâche : MAST

### Paramètres de voie d'entrée [0-7]

Voie	Adresse	Symbole
0	%IO.1.0.0	
1	%IO.1.1.0	
2	%IO.1.2.0	
3	%IO.1.3.0	
4	%IO.1.4.0	
5	%IO.1.5.0	
6	%IO.1.6.0	
7	%IO.1.7.0	

### Paramètres communs [16-23]

Tâche : MAST  
Mode de repli : Repli

### Paramètres de voie de sortie [16-23]

Voie	Adresse	Symbole	Valeur de repli
0	%Q0.1.16.0		0
1	%Q0.1.17.0		0
2	%Q0.1.18.0		0
3	%Q0.1.19.0		0
4	%Q0.1.20.0		0
5	%Q0.1.21.0		0
6	%Q0.1.22.0		0
7	%Q0.1.23.0		0

## 0.0 : BMX P34 2020

### Identification du module :

Réf. commerciale : BMX P34 2020  
Adresse : 0.0

Désignation : CPU 340-20 Modbus Ethernet  
Symbole :

### Mode de marche

Entrée Run/Stop : Non  
Protection mémoire : Non  
Démarrage Auto/Run : Non  
RAZ MWi : Oui  
Démarrage à froid uniquement : Non

### Données

Nombre de bits : 512  
Nombre de mots : 1024  
Nombre de constantes : 256  
Nombre de bits système : 128  
Nombre de mots système : 168

### Voie 0 :

Fonction métier : Liaison Modbus  
Type de voie : Voie intégrée  
Tâche : MAST  
Type : Esclave  
Vitesse de transmission : 19 200 bits/s  
Stop : 1 bit  
Délai inter-trames : 2 ms  
Numéro d'esclave : 1  
Ligne physique : RS485

Données : 8 bits  
Parité : Paire

### Voie 3 :

Fonction métier : ETH TCP IP  
Type de voie : Voie intégrée  
Lien réseau : Ethernet\_1  
Tâche : MAST

## Variables et instances FB

### CTD

Nom	Commentaire	Valeur	Utilisé	DG
<b>FBI_32</b>			1	
<entrées>				
CD	Trigger input			
LD	Load data			
PV	Preset value			
<sorties>				
Q	Indicator output			
CV	Count value			
<b>FBI_33</b>			1	
<entrées>				
CD	Trigger input			
LD	Load data			
PV	Preset value			
<sorties>				
Q	Indicator output			
CV	Count value			
<b>FBI_34</b>			1	
<entrées>				
CD	Trigger input			
LD	Load data			
PV	Preset value			
<sorties>				
Q	Indicator output			
CV	Count value			

### EBOOL

Nom	Const	Adresse	Commentaire	Valeur	Utilisé	DG
<b>BR1_Triangle</b>	NON	%M117			4	NON
<b>BR2_Carre</b>	NON	%M119			4	NON
<b>BR3_Cercle</b>	NON	%M121			4	NON
<b>Capteur</b>	NON	%M0			4	NON
<b>Capteur2</b>	NON	%M205			2	NON
<b>Carre</b>	NON	%M2			3	NON
<b>Cercle</b>	NON	%M3			2	NON
<b>Conv_Carre</b>	NON	%M120			1	NON
<b>Conv_Cercle</b>	NON	%M122			1	NON
<b>Conv_Princ</b>	NON	%M116			1	NON
<b>Conv_Triangle</b>	NON	%M118			1	NON
<b>RESET</b>	NON	%M206			1	NON
<b>Triangle</b>	NON	%M1			2	NON

### FIFO

Nom	Commentaire	Valeur	Utilisé	DG
<b>FBI_0</b>			0	
<entrées>				
R	1: stack will be cleared			
SET	1: write value to stack			
GET	1: read value from stack			
X	Stack register input			
N_MAX	Max no. of elements in stack			
<sorties>				
FULL	1: stack register is full			



# AffectationES : [MAST]

	1	10	20	30	40	50	60	70	80	90	100	110
120		134										
1	%M0	:=	%IO.1.0 ;	(*	Capteur d'objet Cp	*)						
2	%M1	:=	%IO.1.1 ;	(*	Triangle	*)						
3	%M2	:=	%IO.1.2 ;	(*	Carre	*)						
4	%M3	:=	%IO.1.3;	(*	Cercle	*)						
5	%M4	:=	%IO.1.4;	(*	*)							
6	%M5	:=	%IO.1.5;	(*	*)							
7	%M6	:=	%IO.1.6;									
8	%M7	:=	%IO.1.7;									
9	%Q0.1.16 := %M116			;	(*	Moteur convoyeur Principale Mo	*)					
10	%Q0.1.17 := %M117			;	(*	Bras pousseur BR1	*)					
11	%Q0.1.18 := %M118			;	(*	Convoyeur Triangle	*)					
12	%Q0.1.19 := %M119			;	(*	Bras pousseur BR2	*)					
13	%Q0.1.20 := %M120			;	(*	Convoyeur Carre	*)					
14	%Q0.1.21 := %M121			;	(*	Bras pousseur BR3	*)					
15	%Q0.1.22 := %M122			;	(*	Convoyeur cercle	*)					
16	%Q0.1.23 := %M123			;	(*	*)						

# command : [MAST]

