

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de projet de fin d'études

présenté par

Mr. BOUCETTA Youcef

&

Mr. HADDOUCHE Mohammed

Pour l'obtention du diplôme de master en électronique option système de vision et robotique

Détection Poursuite et Comptage d'Objets par la Vision Artificielle

Proposé par : Dr. NAMANE Abderrahmane

Dr. AREZKI Madjid

Année Universitaire 2016-2017

Dédicace

Je dédie ce modeste travail

*A mes chères parents pour leurs soutient, leur amour,
leur patience.*

*A ma femme pour les sacrifices qu'elle a consentis pour
avenir et pour le soutient moral.*

*A ma petite princesse de zomois pour la joie
et le bonheur qu'elle n'a cessai d'offrir.*

*A mes frères et ma sœur amine et sa petite famille,
Seddik et Amina et sa petite famille.*

*A ma belle mère mes belle sœurs Ilhem, Afaf,
Manel et la petite hadjer .*

A tous mes oncles et tentes.

A tous mes amis.

A mon cousin Billel.

A mon binôme youcef.

*A tous ces personnes que j'ai sentis redoutable
je leur dédie ce modeste travail.*

MOHAMMED



Dédicace

Je dédie ce modeste travail à :

Ma grande mère qui m'a soutenu avec ses prières tout au long de ma vie.

Ma mère la voix de la sagesse et de l'amour qui a prié et rêvé de voir ses enfants toujours parmi les meilleurs.

Mon père qui m'a donné tous pour être où je suis.

Ma femme et mes enfants Mohamed zino et hamza.

Mes frères et sœurs

Mon promoteur M. Namane qui m'a poussé beaucoup et m'a donné le nécessaire pour achever Mon travail.

Mes amis : Zoubir, Mohamed, Djamel, Toufik,

L'arbi, Ahmed, Yacine, Sid Ali,

Hamza, Ismail

A mon binôme Mohamed.

A ceux qui m'ont soutenu pendant toute la durée de mes études, je dédie ce mémoire

Yousef



Remerciements

Tout d'abord, nous remercions dieu de tout puissant de nous avoir aidés à surmonter tous les dures épreuves et moments pour atteindre la fin de ce projet.

Naturellement, nous exprimons notre gratitude au Professeur NAMANE Abderrahmane, le promoteur pour avoir dirigé notre projet de fin d'études et pour ses conseils, et pour son encadrement et son aide inestimable. Nous remercions Dr AREZKI Madjid pour avoir accepté d'être notre co-promoteur. Nous sommes très reconnaissants envers tout les enseignants d'électronique pour leurs aides, Mr. MAAMOUN, Mr. AIT SAADI, Mr. DAHMANI, Mr. HENHEN, Mr. KAZED, Mme. BEN BLIDIA et Mme. REGGUIG, et n'oublions pas Mr. BENCHERCHALI.

Enfin, nos pensées vont à toutes les personnes qui nous ont apportés leurs soutiens.

Merci..

ملخص

يتناول هذا المشروع الكشف عن الأشياء المتنقلة وتتبعها في بيئة ديناميكية، باستخدام كاميرا وناقلة تنقل الأجسام. ويهدف إلى الكشف عن الأجسام المتحركة في مشهد من خلال تحليل انتقالها الواضح للصور، وعد هذه الأجسام بعد المعالجة المسبقة، عن طريق البرمجة باستعمال الماتلاب. ويمثل هذا الموضوع تحديا مهما يفرضه وسط الالتقاط الذي يتطلب الدقة لجعل التطبيق يعمل في الوقت الحقيقي، وفي البيئة ذات الإضاءة الغير مستقرة دائما.

الكلمات الافتتاحية: ناقل، كاميرا، كشف الأجسام، عد الأجسام، البرمجة على الماتلاب.

Résumé

Ce projet traite la détection et le suivi d'objets mobiles dans un environnement dynamique, en utilisant une caméra et un convoyeur qui déplace des objets. Il a pour objectif la détection des objets mobiles dans une scène par une analyse de leurs déplacements apparents dans les images, et compter ces objets après un prétraitement, moyennant une programmation par MATLAB. Ce sujet représente un défi important imposé par le moyen d'acquisition qui nécessite une faible résolution pour faire fonctionner l'application en temps réel, et l'environnement dont l'éclairage qui n'est pas toujours stable.

Mots-clés: Convoyeur, Caméra CCD, Détection des Objets, Comptage d'objets, Programmation par MATLAB.

Abstract

This project treats the detection and tracking of mobile objects in a dynamic environment, using a camera and a conveyor that moves objects. It aims at detecting moving objects in a scene by analyzing their apparent displacements in the images, and counting these objects after pre-processing, by means of programming by MATLAB. This subject represents a major challenge imposed by the acquisition means which requires a low resolution to make the application work in real time, and the environment whose lighting is not always stable.

Keywords: Conveyor, CCD Camera, Object Detection, Object Counting, Programming on MATLAB.

Table des matières

Liste des figures

Liste des tableaux

Résumés

Introduction général	1
Chapitre 1: Généralité sur la Vision Artificiel et le Traitement d'Image	
1.1 Introduction	4
1.2 Vision Artificiel	5
1.2.1 Acquisition d'une image numérique par cameras CCD (Charge Coupled Device)	5
1.2.2 Restituer une image en couleur	6
1.2.3 Comparaison entre les capteurs CCD et CMOS et leurs applications	8
1.3 Traitement d'image	8
1.3.1 Introduction	8
1.3.2 Traitement d'image	9
1.3.3 Définition de l'image	10
1.3.4 Acquisition d'une image	11
1.3.5 Image numérique	11
1.3.6 Caractéristiques d'une image numérique	12
a) Pixel	12
b) Dimension	12
c) Résolution	13
d) Bruit	13
e) Histogramme	14
f) Contours	15
g) Luminance	15
h) Contraste	15
1.3.7 Qualité d'image numérique	16
1.3.8 Les types d'images	16
a) Les images de type bitmap	17
b) Les images de type vectoriel	18
1.3.9 Prétraitement	19
1.3.10 Le filtrage	20
a) Le filtrage linéaire	20

b) Le filtrage non linéaire	23
1.3.11 La binarisation	24
a) La binarisation global	25
b) La binarisation locale	26
1.3.12 La détection de contour	27
1.4 Conclusion	29

Chapitre 2: Les analyses sur le traitement d'image

2.1 Introduction	31
2.2 Les images numériques	32
2.2.1 Définition d'une image numérique	32
2.2.2 Image vectorielle	32
2.2.3 Image matricielle	32
2.2.4 Résolution d'une image	32
2.2.5 Le codage de l'information d'une image numérique	34
2.3 Les couleurs	34
2.3.1 La synthèse additive de la lumière, ou le mode RVB	34
2.3.2 La synthèse soustractive de la lumière, ou le mode CMJN	35
2.3.3 Images aux Niveaux de Gris	36
2.4 Prétraitement	36
2.4.1 Histogramme de l'image	36
a) Luminance ou brillance d'une image	37
b) Contraste	37
2.4.2 Egalisation d'histogramme	38
2.5 La binarisation	39
2.5.1 Le Seuillage (threshold)	39
2.5.2 Définition la méthode d'OTSU	40
2.5.3 Implémentation de la méthode d'OTSU sur MATLAB	40
2.5.4 Réduction avec seuillage	41
2.6 Le Bruit	41
2.6.1 Bruit lié aux dispositifs d'acquisition	42
2.6.2 Bruit lié à la scène	42

2.6.3	Bruit lié à la transmodulation	42
2.6.4	Bruit lié à l'échantillonnage	42
2.7	Le filtrage	42
2.7.1	le Filtrage linéaire	42
2.7.2	Filtrage médian	43
2.7.3	Filtrage spatial	43
2.7.4	Filtres médians	45
2.8	Etiquetage	47
2.9	Conclusion	52

Chapitre 3: Implémentation Pratique et Résultats de Simulations

3.1	Introduction	54
3.2	Acquisition des images par une caméra CMOS	54
3.3	Traitement Expérimentale	55
3.3.1	Lecture et visualisation de l'image et affichage de la matrice correspondante	55
3.3.2	Convertir une image couleur en niveau de gris	57
3.3.3	Seuillage	57
3.3.4	Binarisation	58
3.3.5	Etiquetage	60
3.3.6	Détection des contours	60
3.4	Conclusion	63

Conclusion générale	65
----------------------------------	----

Bibliographie.

Annexes.

Liste des figures

Figure 1.1 Systèmes à base de TI	4
Figure 1.2 Principe de l'acquisition d'une image CCD	5
Figure 1.3 Définition des couleurs : (a) Fonctions colorimétriques de l'observateur standard, (b) Diagramme chromatique (CIE 1931).....	6
Figure 1.4 (a) Réseau de Bayer, (b) Reconstruction de la couleur	7
Figure 1.5 Lettre A affichée comme un groupe de pixels.	12
Figure 1.6 Un histogramme	14
Figure 1.7 Niveaux de gris	16
Figure 1.8 Filtrage moyen	21
Figure 1.9 Filtrage gaussien	22
Figure 1.10 Filtrage médian	24
Figure 1.11 Binarisation globale	25
Figure 1.12 Binarisation locale	26
Figure 1.13 Détection de contour	28
Figure 2.1 Schéma synoptique de la chaîne de traitement d'image	31
Figure 2.2 pixels par pouce soit environ 7 pixels par cm	33
Figure 2.3 pixels par pouce soit environ 30 pixels par cm	33
Figure 2.4 Les couleurs RVB	35
Figure 2.5 Les couleurs CMJN	36
Figure 2.6 Exemple d'une luminance d'image	37
Figure 2.7 Exemple d'une image Contrasté	37
Figure 2.8 Egalisation d'histogramme	38
Figure 2.9 Exemple d'un egalisation d'un histogramme	38
Figure 2.10 Fonction « seuillage »	39
Figure 2.11 Exemple d'un Seuillage à 70	39
Figure 2.12 Fonction « fenêtre d'intensité »	39

Figure 2.13 l'Image original et leur Histogramme	40
Figure 2.14 Seuillage avec fenêtre d'intensité entre 30 et 100	40
Figure 2.15: illustration de réduction avec seuil	41
Figure 2.16 Le noyau de convolution du filtre κ est a support compact inclus dans $[x_1, x_2] \times [y_1, y_2]$	44
Figure 2.17 Filtres séparables	44
Figure 2.18 Exemple d'un filtre médiane	45
Figure 2.19 Image bruitée « Poivre et Sel »	45
Figure 2.20 a)Filtre de moyenne : rayon 3, b)Filtre de moyenne : rayon 5	45
Figure 2.21 Filtre de moyenne : rayon 7	46
Figure 2.22 a)Filtre median : rayon 3, b)Filtre médian : rayon 5	46
Figure 2.23 Filtre médian : rayon 7	46
Figure 2.24 types de connexité	47
Figure 2.25 Image binaire et sa matrice initiale correspondante	48
Figure 2.26 premier passage de l'étiquetage	50
Figure 2.27 Deuxième passage de l'étiquetage	51
Figure 2.28 Carte finale des étiquètes	51
Figure 2.29 Exemples d'étiquetage	52
Figure3.1 Système de Détection poursuite et comptage d'objets par un Caméra placer au dessus de convoyeur	54
Figure 3.2 : Illustration de la représentation des images numériques	55
Figure 3.3 Taille d'image traitée	56
Figure 3.4 Image initiale	56
Figure 3.5 Afficher le niveau de gris d'image	57
Figure 3.6 Afficher la valeur « level » correspond le programme	58
Figure 3.7 Transformation à l'image binaire	59
Figure3.8 Etiquetage des objets d'image par des couleurs déférentes	60
Figure 3.9 Détermination les contours en rectangle et les centres des objets	61
Figure 3.10 Détermination les contours des objets	62

Liste des tableaux

Tableau 1.1 Avantages principaux des capteurs CCD et APS.....	8
Tableau 1.2 Voisinage de $f(x,y)$	26
Tableau 2.1 Définition des Grandeurs des quelques couleur	34

Introduction générale

Dans les productions à haute cadence, le souci constant d'amélioration de la qualité et la recherche de gain économique poussent de plus en plus les industriels à automatiser les moyens de production. La vision industrielle est une réponse à ces préoccupations pour les opérations de contrôles de la production. En effet les machines de vision industrielle permettent un contrôle de la production à haute cadence et assurent une bonne répétabilité du contrôle.

Les systèmes de vision sont également énormément utilisés pour gérer des flux d'objets, par exemple pour la lecture optique des codes barres ou le tri d'objets (pièces mécaniques, produits alimentaires) par des couleurs différentes avant l'emballage. Enfin la vision industrielle peut être un moyen de guidage pour un système mobile autonome (comme un robot) lorsque ses mouvements ne peuvent pas être déterminés par avance, comme par exemple la préhension d'objets sur un tapis roulant. Une caméra est alors embarquée sur la tête du robot et permet le positionnement de celui-ci au point désiré. En étudiant l'évolution des technologies, on peut se rendre compte que les systèmes de vision industrielle sont de plus en plus présents dans notre environnement.

Dans l'univers du multimédia, la vision industrielle est présente sous forme de système embarqués temps réel s'appuyant sur des traitements d'images et de reconnaissance de formes. Dans le monde de l'entreprise, de plus en plus de caméras dites « intelligentes », c'est à dire permettant à la fois d'acquérir une image, d'extraire l'information utile et de la transmettre, s'imposent comme des outils efficaces et rentables. La présence de plus en plus forte de la vision industrielle dans les entreprises s'explique finalement facilement par la nécessité d'être compétitif et donc de rendre un certain nombre de tâches automatiques, contrôlées et régulées.

L'objectif de ce mémoire est de concevoir et réaliser un système basé sur la vision artificielle pour faire la détection et le comptage d'objets placés sur un convoyeur en utilisant une programmation MATLAB.

Ce mémoire est établi sur trois chapitres. Dans le premier chapitre, nous présentons les généralités sur la Vision Artificiel et le Traitement d'Image ainsi les différents domaines d'utilisation de traitement d'images.

Le deuxième chapitre est consacré à Les analyses sur le traitement d'image et différents étapes de prétraitement d'images.

Dans le troisième chapitre, nous présentons l'acquisition d'images par une caméra et la détection et comptage des objets placés sur un convoyeur, on utilise le logiciel d'exécution MATLAB, on explique les différent+s instructions utilisés.

Enfin, nous terminerons ce mémoire par une conclusion générale qui résume notre travail dans sa partie théorique et simulation des résultats.

1.1 Introduction :

Ce Chapitre a pour objectif la présentation des techniques du traitement d'images pour mieux comprendre le travail réalisé dans ce mémoire. Par traitement d'images, on désigne l'ensemble des opérations sur les images numériques, qui transforment une image en une autre image, ou en une autre primitive formelle, la figure 1.1 illustre un Systèmes à base de traitement d'image.

La vision par ordinateur désigne la compréhension d'une scène ou d'un phénomène à partir d'informations « image », liant intimement perception, comportement et contrôle.

Les domaines traités dans la vision artificielle vont du traitement du signal à l'intelligence artificielle, on ne saurait donc prétendre à l'exhaustivité, mais on vise plutôt l'exploration d'un certain nombre de techniques importantes et actuelles.

Traitement d'images et vision sont des disciplines relativement récents (~années 60) et qui évoluent rapidement. Elles sont en plein expansion et donnent lieu chaque année à une profusion de travaux, académiques, technologiques, industriels. [7]

Cette profusion s'explique par le caractère ardu du sujet : complexité algorithmique dû aux énormes volumes de données, caractère mal posé des problèmes et difficultés à formaliser une faculté biologique « évidente ».

D'autre part l'engouement pour ces disciplines s'explique par la multiplication permanente d'applications et d'enjeux industriels dans des domaines aussi variés que : médecine, télécommunications, automobile, météorologie, défense, jeux vidéo, art, écologie...

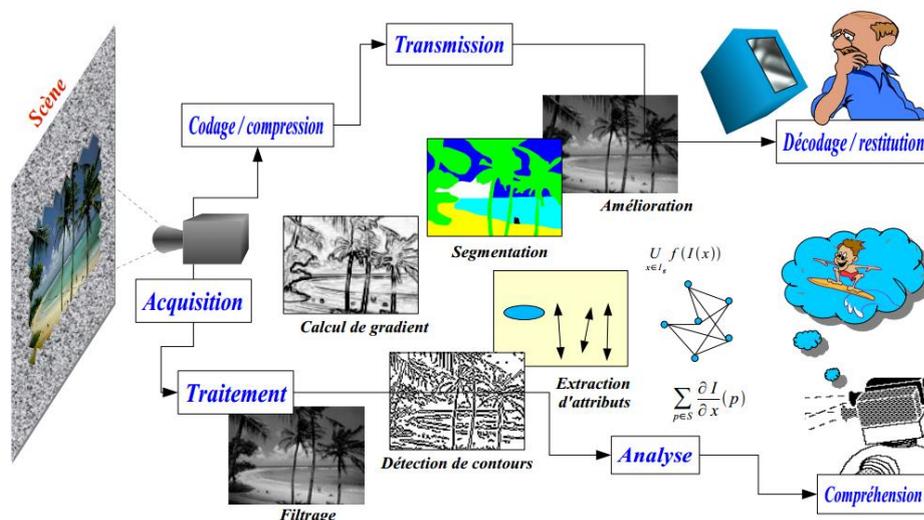


Figure 1.1 Systèmes à base de TI

1.2 Vision Artificiel :

1.2.1 Acquisition d'une image numérique par cameras CCD (Charge Coupled Device) :

L'acquisition d'images à partir des caméras CCD (charge coupled device) repose sur deux principes. Le premier concerne les propriétés électro-optiques du matériau utilisé (silicium). En effet, la réponse spectrale du silicium s'étalant entre 400 nm et 1100 nm permet à ce matériau d'absorber des photons de lumière à partir du proche ultraviolet jusqu'à l'infrarouge.

Le deuxième principe concerne la technologie CCD, introduite en 1970 par Boyle et Smith des laboratoires Bell. Les dispositifs CCD se présentent sous la forme d'une série de capacités MOS (metal-oxide semiconductor) couplées, constituant ainsi des registres analogiques où les charges électriques représentant l'information à transmettre sont stockées puis décalées en série vers la sortie du dispositif. Ainsi, l'intégration dans une même « puce » électronique de séries de photo-éléments associés à des registres analogiques CCD permet d'obtenir des capteurs d'images à l'état solide. À l'aide de ces circuits, l'image est acquise en deux temps : - un temps d'intégration ; - un temps de transfert. Durant le temps d'intégration, la lumière incidente, d'énergie $h\nu$, est absorbée par les photo-éléments puis transformée en paquets de charges électriques proportionnelles à l'intensité lumineuse incidente (figure 1.2 a).

Les paquets de charge sont alors transférés vers les registres analogiques via une porte de transfert et décalés en série vers la sortie du CCD (figure 1.2 b). L'organisation topologique des photo-éléments permet de définir deux types de capteurs d'images CCD : - les capteurs d'images CCD linéaires ; - les capteurs d'images CCD matriciels.

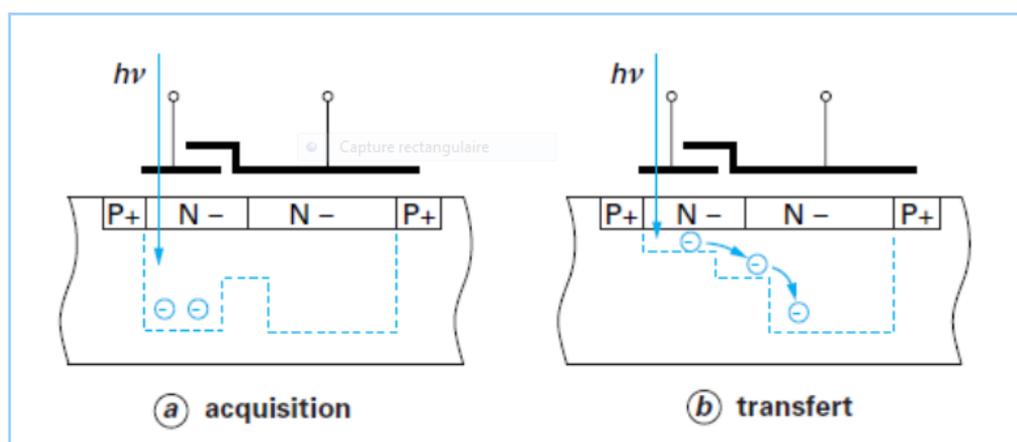


Figure 1.2 Principe de l'acquisition d'une image CCD

1.2.2 Restituer une image en couleur :

Confrontant la technologie au processus de vision biologique, la restitution numérique de la couleur est un défi qui concerne la majorité des capteurs destinés à une application grand public. Les objets, éclairés sous un spectre lumineux visible, renvoient un spectre lumineux dépendant des propriétés optiques de leurs surfaces et du spectre reçu. L'œil et le cerveau attribuent à tout spectre une couleur, en se basant sur la réponse relative de trois types de cônes, des cellules sensibles à la couleur dans la partie centrale de la rétine. Il a été montré qu'il était possible de décrire toutes les couleurs perçues par la superposition de trois sources mono chromatiques, et le système de référence utilise trois sources virtuelles dénommées illuminants X, Y et Z. Les amplitudes relatives qu'il faut attribuer aux trois illuminants X, Y et Z pour percevoir toutes les couleurs mono chromatiques sont données en figure 1.3 a pour un observateur standard.

Une transformation en coordonnées réduites, plus commode à l'usage, permet de décrire l'ensemble des couleurs dans un plan (x, y) pour une luminance donnée. Dans ce plan, les couleurs mono chromatiques de l'arc-en-ciel déterminent la frontière courbe du diagramme. La ligne des pourpres rejoint les deux extrémités de cette courbe mono chromatique et clôt l'espace des couleurs visibles. Le plan colorimétrique en coordonnées réduites x y est représenté en figure 1.3b.

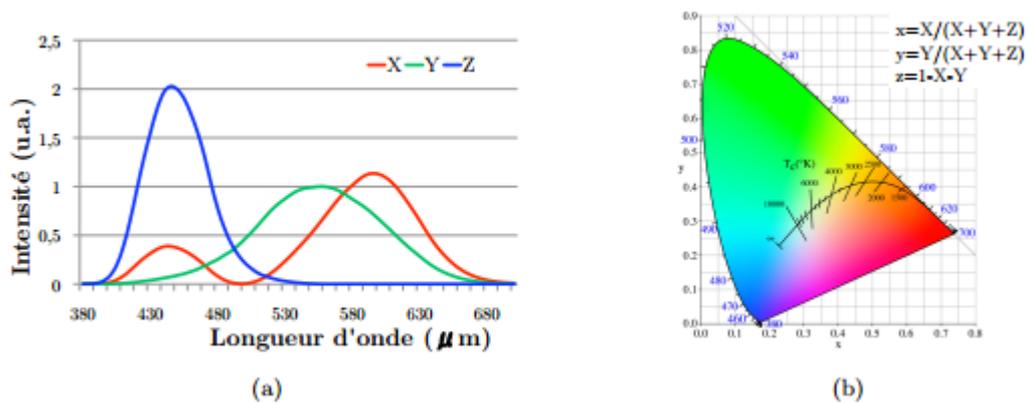


Figure 1.3 Définition des couleurs : (a) Fonctions colorimétriques de l'observateur standard, (b) Diagramme chromatique (CIE 1931)

Plusieurs méthodes de restitution ont été élaborées, d'abord destinées aux CCD puis appliquées aux APS. Elles reposent sur un filtrage de la longueur d'onde du rayonnement incident par des filtres rouge, vert et bleu dont les courbes de réponse sont ajustées de manière à être, à une combinaison linéaire près, identiques à celles de l'observateur standard décrites en figure 1.3. Il est ainsi possible d'obtenir les trois composantes nécessaires à la reconstruction des couleurs.

Solutions pour les capteurs mono chromes Le premier système exploitant ce principe est la roue à filtres devant un capteur monochrome. Celui-ci prend trois images successives, apportant les trois composantes. Ce système offre une grande résolution, il est utilisé dans le domaine de la photographie professionnelle.

Pour des images à partir de scènes rapides, trois capteurs monochromes peuvent être embarqués: les composantes rouge, verte et bleue sont séparées à travers un prisme et dirigées sur chacun des capteurs ; l'image est reconstituée par addition des trois composantes. Si ces systèmes sont courants dans les applications audiovisuelles, le coût et l'encombrement de cette solution restent un frein à une plus large diffusion. Réseaux de filtres colorés La solution la plus répandue consiste à partitionner la matrice de pixels en trois composantes : rouge, verte et bleue, à l'aide de filtres colorés organisés en damier à la surface du capteur. Le plus courant est le damier de Bayer, inventé par Eastman Kodak dans les années 70 [7], dans lequel sont disposés des pixels verts et bleus sur une ligne; verts et rouges sur la ligne suivante (figure 1.4 a). La couleur verte est privilégiée car elle correspond au pic de sensibilité de l'œil humain. Les réseaux de filtres sont réalisés par photolithographie de résines colorées, généralement dans l'ordre : vert, bleu, rouge. Les épaisseurs typiques évoluent entre 600 nm et 1 µm. [7]

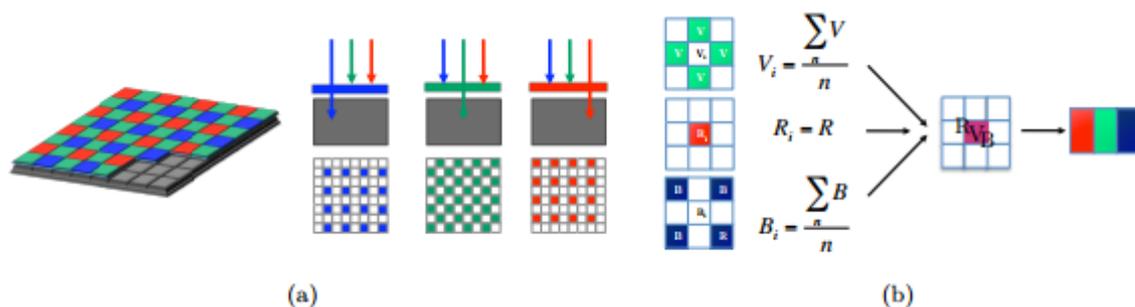


Figure 1.4 (a) Réseau de Bayer, (b) Reconstruction de la couleur

Chaque pixel donne un signal correspondant à l'une des trois couleurs, les composantes manquantes sont obtenues par interpolation des pixels voisins 1.4 b. La figure illustre ce principe dans le cas d'une matrice de neuf pixels. Dans la même idée, Toshiba Corporation a développé un filtre utilisant les trois couleurs complémentaires : jaune, cyan, magenta. [7]

1.2.3 Comparaison entre les capteurs CCD et CMOS et leurs applications:

La comparaison entre les CCD et les APS est largement abordée dans la littérature; le tableau 1.1 dresse les principaux avantages de chaque technologie. [7]

Les capteurs CCD ont été intensément étudiés au cours des dernières décennies et sont parvenus à un niveau de performances très robuste, rendant possible la détection de photons unitaires. Néanmoins ils comportent, par nature, certains inconvénients. Leur fonctionnement capacitif nécessite des tensions de polarisation de l'ordre de 10V et des courants d'horloges pouvant atteindre plusieurs ampères. La consommation d'énergie est 50 à 100 fois plus importante que pour les APS. Le processus de lecture en mode balayage ne permet pas la lecture aléatoire d'un pixel et limite la vitesse de lecture. Finalement, ce type de capteur requiert une technologie de fabrication devenue incompatible avec celles des circuits intégrés, la plupart des procédés ayant été adaptés pour réduire le niveau de bruit et améliorer le transfert de charge.

CCD	CMOS
Faible bruit	Faible consommation
Faible courant d'obscurité	Capable d'une forte intégration
Facteur de remplissage de 100%	Faible coût
Grande sensibilité	Accès aléatoire au pixel

Tableau 1.1 Avantages principaux des capteurs CCD et APS

1.3 Traitement d'image :

1.3.1 Introduction :

Le traitement d'image est la sous-branche du [traitement du signal](#) qui se consacre aux images. C'est aussi toute la partie de calcul d'une chaîne de vision artificielle.

C'est une science au carrefour des [mathématiques](#), de la [physique](#), et de l'[informatique](#).

Ponctuellement on arrive à y glisser un peu de [biologie](#).

En effet, on parle alors de photographie, de [télévision](#) ou encore d'[holographie](#). Par contre dans la partie numérique on emploie le terme de traitement d'image, sous entendu numérique, et on fait souvent clairement référence au traitement du signal.

Nous ne considérerons donc ici que le cas des images numériques.

La compréhension du traitement d'image commence par la compréhension de ce qu'est une image. Ensuite on trouve une étape d'acquisition de l'image. Dans beaucoup de cas c'est la qualité de cette étape qui va déterminer la faisabilité de la chaîne de traitement d'image qui va suivre. Ensuite on trouve en général ce que l'on nomme des prétraitements, cela consiste souvent à corriger des petits défauts de l'image pour passer au traitement à proprement parler.

1.3.2 Traitement d'image :

Avec la parole, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

Le traitement d'image a commencé à être étudié dans les années 1920 pour la transmission d'image par le câble sous-marin allant de New-York à Londres [1]. Harry G. Bartholomew et Maynard D. McFarlane effectuèrent la première digitalisation d'image avec compression pour envoyer des fax de Londres à New-York. Le temps de transfert passa ainsi de plus d'une semaine à moins de trois heures. Il n'y a pas vraiment eu d'évolution pour l'image par la suite jusqu'à la période d'après-guerre.

On peut tout de même noter que le traitement du signal pris son essor à la fin de la guerre avec l'arrivée du radar. La prospection pétrolière participa aussi beaucoup au développement des techniques de traitement du signal.

L'essor du traitement d'image n'a eu lieu que dans les années 1960 quand les [ordinateurs](#) commencèrent à pouvoir travailler sur des images [1].

Le monde du traitement du signal en général a connu une révolution le jour où la transformation de Fourier rapide ([FFT](#)) a été redécouverte. L'ensemble des manipulations fréquentielles devenaient envisageables sur ordinateur.

L'essentiel des recherches portent sur l'amélioration des images et leurs compressions. David Marr serait le premier à bien formaliser la détection de contour.

Au cours des années 1980 [2], il y a eu un véritable engouement pour le traitement de l'image et surtout pour la compréhension de l'image par des systèmes experts. Les ambitions étaient beaucoup trop grandes. L'échec fut à la hauteur.

Les années 1990 ont vu l'amélioration constante des opérateurs. La recherche médicale est devenue un très gros demandeur en traitement d'image pour améliorer les diagnostics fait à partir des nombreuses techniques d'imageries médicales, la technique reine étant l'[IRM](#). Le grand public se familiarise avec le traitement d'image grâce au logiciel Photoshop. Enfin la décennie s'achève sur l'engouement pour les ondelettes et les images multimodales.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

1.3.3 Définition de l'image :

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain.

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation.

1.3.4 Acquisition d'une image :

L'étude de cette étape passe immanquablement par le système d'acquisition qui fait référence : l'œil. On utilise plus couramment des caméras vidéo, ou récemment pourquoi pas des appareils photos numériques. En médecine, on utilise des imageurs IRM, TEP, scanner X, écho doppler, échographie, scintigraphie etc.

Tous ces systèmes peuvent être comparés à des capteurs. Il ne faut pas oublier qu'il y a une étape de conversion analogique/numérique. C'est souvent cette étape qui limite la résolution de l'image.

L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément ([pixel](#)).

1.3.5 Image numérique :

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits).

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy .) en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x, y)$ où :

x, y : coordonnées cartésiennes d'un point de l'image.

$f(x, y)$: niveau de gris en ce point

Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont généralement rangées sous formes de tableau I de n lignes et p colonnes. Chaque élément $I(x, y)$ représente un pixel de l'image et à sa valeur est associé un niveau de gris codé sur m bits (2^m niveaux de gris ; 0 = noir ; $2^m - 1$ = blanc). La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur.

1.3.6 Caractéristiques d'une image numérique :

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

a. Pixel :

Contraction de l'expression anglaise « Picture éléments » : éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous (**Figure 1.5**) :

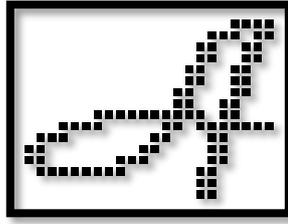


Figure 1.5 Lettre A affichée comme un groupe de pixels.

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

b. Dimension :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

c. Résolution :

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre), elle peut s'exprimer en DPI (« dots per inch » : points par pouce) ou en PPP (points par pouce). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution.

Le choix de la résolution dépend de la destination finale de l'image : la résolution d'un écran informatique (moniteur) est de 72 DPI ; en conséquence, toute image destinée à l'affichage sur l'écran ne nécessite qu'une résolution de 72 voire 75 DPI. Dans le cas d'une numérisation, on peut calculer avec précision la résolution idéale. Pour affichage à écran, la formule de calcul est : Taille de destination (en points) / Taille d'origine (en pouces).

Exemple : pour un document de 21 cm de large (8,27 pouces) qui doit occuper 800 pixels d'écran (horizontalement) : $800 / 8,27 \implies 96$: Il faut donc choisir la résolution la plus proche de 96 DPI.

d. Bruit :

Le bruit est un terme issu du domaine de l'acoustique et désigne un signal parasite. Que ce soit pour le son ou pour l'image, le principe est identique : sur tout signal de base vient s'adjoindre un ensemble d'informations parasites aléatoires.

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins. Sur une image, le bruit est particulièrement visible dans les zones sombres. Pourquoi ? Simplement car le capteur perçoit moins de lumière et dispose donc de moins d'informations électriques. Si le signal diminue, la proportion de bruit augmentera. La solution : éclairer.

De la même manière, la sensibilité influe sur la qualité de l'image car le capteur n'est pas nativement apte à supporter de hautes sensibilités. Plus on augmente la sensibilité et plus la mesure de la lumière sera approximative, donc plus le signal sera amplifié, tout comme le bruit par la même occasion.

e. Histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

L'histogramme est une courbe en deux dimensions pour laquelle l'axe vertical représente le nombre de pixel de l'image ou encore la proportion de ces pixel dans l'image, tandis que l'axe horizontal représente l'ensemble des valeurs de luminosité potentielles de l'image comme se présente la figure ci-dessous (**Figure 1.6**) :

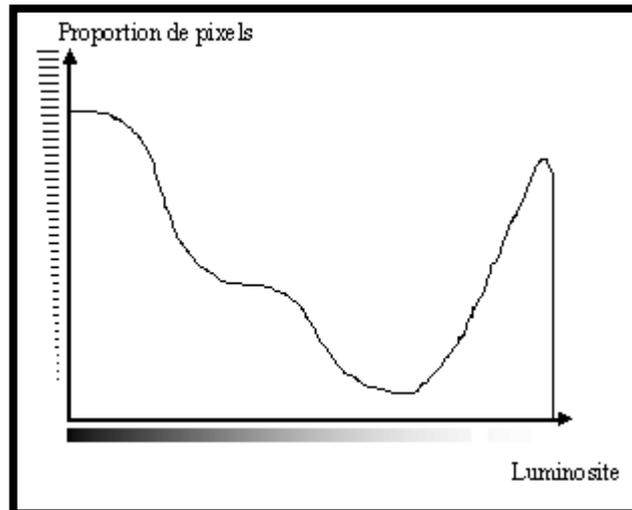


Figure 1.6 Un histogramme.

Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée [1].

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

f. Contours :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

g. Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes) ;

- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir ; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.

- L'absence de parasites.

h. Contraste :

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport : $C = \frac{L_1 - L_2}{L_1 + L_2}$

- **Images à niveaux de gris :**

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

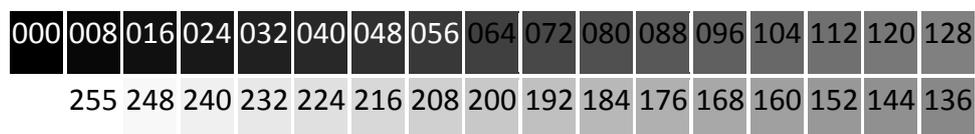


Figure 1.7 Niveaux de gris

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

1.3.7 Qualité d'image numérique :

Elle dépend, d'une part, de la qualité des images d'origine et, d'autre part, des moyens mis en œuvre pour convertir un signal analogique en signal numérique. Elle dépend aussi de :

- La qualité des périphériques de numérisation de l'image, du nombre de niveaux de gris ou de couleurs enregistrées, etc.
- La qualité de l'affichage à l'écran : définition de l'écran, nombre de teintes disponibles simultanément, calibrage de l'écran, etc.

Les critères d'appréciation de la qualité d'une image, tels que cités succinctement ci-dessus, dépendent largement de la structure même de l'image réaliste ou conceptuelle et de son mode de représentation (bitmap ou vectorielle).

1.3.8 Les types d'images :

Les images appartiennent à deux grandes familles : bitmap (image-bit) et vectorielle.

Alors qu'une image vectorielle est décrite à l'aide de courbes et d'équations mathématiques, une image bitmap est constituée de pixels et se réduit donc à une matrice de points.

Si les images vectorielles peuvent être manipulées avec beaucoup de facilité, les modifications de taille, par exemple, apportées à une image bitmap ne sont pas sans incidence.

a. Les images de type bitmap :

Une image bitmap est composée en mode point. Le système de codage le plus universel consiste en effet à décomposer la représentation graphique, l'image, en un certain nombre de points élémentaires caractérisés par leurs coordonnées spatiales et leur couleur. Il s'agit donc d'une représentation graphique définie par l'ensemble des points qui la composent.

➤ Avantage :

- Le mode de codage des images bitmap (24 bits, codage RBG) les rend adaptées au fonctionnement des principaux périphériques, notamment les contrôleurs d'écran "true colors" (point allumé ou non, codé sur x bits).
- Elles conviennent fort bien aux images complexes, principalement d'origine analogique, qui ne peuvent être codées qu'en mode point.

- Elles se laissent manipuler et traiter par des opérations techniques "naturelles" pour un graphiste qui retrouve des outils et les manipulations très proches de ceux qui caractérisent son métier et sa pratique professionnelle de type analogique.

- Elles permettent des opérations comme l'anti-aliasing (suppression du crénelage produit à la rencontre de deux couleurs éloignées), le rehaussement des contours, les modifications locales de l'image (contraste, colorimétrie, effets, filtres, etc.).

- Le mode de codage point par point étant relativement universel, une fois cette opération effectuée, le transcodage demande des calculs répétitifs mais relativement simples: la compatibilité est aisée entre les différents formats.

La structure du fichier est telle que des dégradations minimales des données -mais non pas de l'en tête- ne le rendent pas nécessairement inutilisable.

➤ **Inconvénient :**

- Les images bitmap ont une résolution fixe : aussi la qualité maximale sur un périphérique d'affichage ou d'impression donné rend-elle nécessaire de travailler, dans la majorité des cas, dans la résolution native de ce périphérique.

Concrètement cela veut dire qu'une résolution de type écran donnera d'assez mauvais résultats sur un imageur photographique. Les images bitmap sont donc dépendantes du périphérique.

- Elles supportent mal les opérations de redimensionnement, réduction ou agrandissement. Les deux opérations se traduisent par une perte d'information. Après une réduction de taille, l'image réduite présentera souvent des effets d'escaliers plus marqués que ceux de l'image source.

Un agrandissement se traduira par la multiplication à la taille voulue de chacun des pixels pris séparément : chaque point se voit grossi, mais la résolution demeurant identique, la définition de l'image sera de qualité inférieure.

- Les images bitmap sont "lourdes" : les fichiers, lorsque l'on traite des images en haute définition, ont des tailles qui varient entre 10 et 30 Mo par image.... Elles sont donc encombrantes, difficiles à faire passer sur le réseau, etc.

b. Les images de type vectoriel :

La description vectorielle d'une image consiste en une description géométrique : le document numérisé prend donc la forme d'une suite de formules mathématiques décrivant les formes élémentaires constituant l'image (carrés, rectangles, ellipses, cercles, courbes, etc.). Chaque forme élémentaire constitue un objet et se voit assigné un certain nombre d'attributs tels que la couleur [1], la transparence, l'épaisseur du trait, le type de trait (pointillé, etc.)

➤ **Avantage :**

- L'image numérique doit être calculée avant de pouvoir être affichée par le périphérique (opération qui porte le nom de rasterisation). Cette opération peut être faite pour n'importe quelle résolution du périphérique : l'image vectorielle est réellement indépendante du périphérique.
- Toutes les modifications spatiales de l'image (réduction, agrandissement, translation, rotation, etc.) sont aisées et n'occasionnent aucune perte d'information. Il suffit en effet de modifier les coordonnées des points de contrôle qui définissent l'objet.
- L'image vectorielle est particulièrement adaptée aux représentations schématiques et stylisées constituées de formes géométriques, uniformément remplies par des à-plats de couleur ou des motifs.
- Un fichier vectoriel est bien plus compact qu'un fichier en mode point. Sa taille varie en fonction de la complexité de l'image, mais pas en fonction de la résolution demandée. Un fichier de 2 Mo peut représenter une image déjà forte complexe.

➤ **Inconvénient :**

- Une image vectorielle ne peut coder une image analogique telle qu'une image photographique.
- Le travail sur des objets graphiques isolés qu'il faut ensuite associer ou grouper est peu familier au graphiste ou à l'illustrateur. Il faut donc une certaine habitude et un apprentissage de nouvelles procédures de création.
- Certaines manipulations telles que les modifications de couleurs sont difficiles sur une zone d'un objet, sur un objet simple ou sur un groupe d'objets.

- Un fichier vectoriel est plus fragile qu'un fichier bitmap dont l'en-tête surtout doit être intacte. La moindre dégradation de l'information est souvent irréparable.

- Chaque format de fichier vectoriel possédant ses propres attributs, la compatibilité entre les formats est difficile.

1.3.9 Prétraitement :

L'image brute est une structure simple dans son organisation mais complexe dans son contenu. Ceci est dû à plusieurs facteurs, d'une part la grande quantité d'information qui y réside et d'autre part le nombre important de processus indépendants de l'utilisateur qui tendent à y introduire des distorsions indésirables, donc au dégrader.

Parmi les éléments de perturbation on trouve :

- L'éclairage qui peut engendrer des ombres, des reflets, être très puissant et saturer la caméra ou bien trop faible et rendre l'image trop sombre.

- La qualité optique de l'objectif : aberration, astigmatisme, etc...., et celle de prise de vue : mise au point.

- Le bruit électronique inhérent au capteur qui engendre une imprécision dans la restitution des niveaux de gris.

Le rôle du prétraitement dans sa définition la plus général est de remédier aux dégradations ayant affecté l'image et de rendre cette image mieux adaptée à une application particulière. Donc le prétraitement consiste à améliorer la qualité de l'image en éliminant les défauts dus à l'éclairage et au processus d'acquisition.

Les prétraitements utilisent des opérations sur les images telles que :

- Filtrages (gaussien, moyen, médian)

- La binarisation (ou le seuillage).

- La détection de contour.

1.3.10 Le filtrage :

Le filtrage d'image a pour but d'améliorer la qualité d'une image numérique.

Le filtrage manipule uniquement les données de l'image numérisée. On ne fait aucune supposition sur ce que représente l'image. Le résultat obtenu suite à un filtrage dépend donc énormément de la qualité du signal de l'image d'origine. Si le signal est très détérioré, le filtrage seul sera très peu efficace. Dans ce genre de situation le filtrage est juste une première étape dans un traitement plus complexe qui fera intervenir des informations sémantiques sur l'image.

Chaque filtre cherche à atténuer un type de défaut bien précis. Il n'y a pas de filtre universel capable de corriger tous les défauts. Il faut choisir les bons filtres suivant les défauts que nous désirons corriger.

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

a. Le filtrage linéaire :

Le filtre est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage.

➤ Le filtre moyen :

Le filtre moyen consiste à conformer un pixel avec son voisinage en le remplaçant par la moyenne des niveaux de gris de son voisinage.

Ce type de filtre utilise la moyenne non pondérée des voisins et peut être mis sous la forme d'un

$$\text{masque H : } H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Il faut diviser tous les coefficients du masque par leur somme, pour que l'image de résultat soit bien normalisée entre 0 et 255 et dans ce cas la somme est de 9 d'où le $\frac{1}{9}$ [1].

La matrice, celle qu'on a appelée le masque, ne changera pas de taille : ce sera toujours une matrice 3x3 (ou 5x5, 7x7...), mais qui va se déplacer sur toute l'image.

Pour chaque pixel P de l'image, on effectuera alors la convolution entre le masque et la sous-image de même taille que le masque et centrée sur P, et on remplacera la valeur de P par le résultat obtenu.



Image originale



Image filtrée par un filtre moyen
de taille 3*3



Par un filtre moyen de taille 5*5



par un filtre moyen de taille 7*7

Figure 1.8 Filtrage moyen.

Le filtre moyen à pour :

- Permet de lisser l'image (smoothing).
- Remplace chaque pixel par la valeur moyenne de ses voisins.
- Réduit le bruit.
- Réduit les détails non-importants.
- Brouille ou rend floue l'image (blur edge).

➤ **Le filtre gaussien :**

Il s'agit de filtre que l'on utilise pour lisser une image, mais en tenant compte du fait que plus un pixel du voisinage est proche du pixel actuellement traité, plus il a de chances de faire partie du même objet, et donc d'avoir la même couleur.

Il faut diviser tous les coefficients du masque H par leur somme, pour que l'image de résultat soit

bien normalisée entre 0 et 255. $H = 1/16 \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Un inconvénient évident du filtrage gaussien est qu'il ne lisse pas uniquement le bruit, mais il gomme aussi les contours, les rendant difficilement identifiables. Le filtre gaussien donnera un meilleur lissage et une meilleure réduction du bruit que le filtre moyen



Image originale



Image filtrée par un filtre Gaussien de taille 3*3



Par un filtre Gaussien de taille 5*5



Par un filtre Gaussien de taille 7*7

Figure 1.9 Filtrage gaussien

b. Le filtrage non linéaire :

Si le filtre ne peut pas être exprimé par une combinaison linéaire, il est appelé " non-linéaire ". Les filtres non-linéaires sont plus complexes à mettre en œuvre que les filtres linéaires. Cependant les résultats obtenus avec les filtres non-linéaires sont très souvent de meilleure qualité que ceux obtenus par les filtres linéaires.

➤ **Le filtre médian :**

La médiane est une mesure statistique qui représente une alternative robuste à la moyenne. Considérons n valeurs numériques x_1, \dots, x_n (pas nécessairement distinctes), où n est impair. On les ordonne de la plus petite à la plus grande, ce qui donne la suite permutée x_{i_1}, \dots, x_{i_n} , où $\{i_1, \dots, i_n\}$ est une permutation de $\{1, \dots, n\}$. La médiane est alors la valeur placée au milieu de cette suite ordonnée, à savoir x_{i_m} pour $m = (n+1)/2$.

Par exemple, soit $n = 9$, et considérons les 9 valeurs : 17, 1, 3, 5, 3, 1, 12, 2, 7.

Notons que certaines valeurs peuvent être répétées, et qu'il ne faut pas supprimer les répétitions !

En les ordonnant de la plus petite à la plus grande, on obtient la suite 1, 1, 2, 3, 3, 5, 7, 12, 17.

La valeur au milieu de cette suite ordonnée est la 5ème, à savoir 3, qui est donc la médiane des 9 valeurs. On écrira $\text{med}(x_1, \dots, x_n)$ pour la médiane des variables x_1, \dots, x_n . Ainsi dans l'exemple ci-dessus, on a $\text{med}(17, 1, 3, 5, 3, 1, 12, 2, 7) = 3$. Notons que - tout comme la moyenne - la médiane ne dépend pas de l'ordre dans lequel sont données les n valeurs.

Le filtre médian est un filtre spatial qui calcule en chaque pixel la médiane des niveaux de gris des pixels de sa fenêtre (ou bien masque), ce qui donnera le niveau de gris du pixel dans l'image filtrée. Plus précisément, pour tout pixel p , soit $W(p)$ la fenêtre associée à p ; on suppose que sa taille $|W(p)|$ est impaire (c'est ainsi le cas pour une fenêtre centrée en p , par exemple une fenêtre 3×3 ou 5×5). Etant donnée une image I à niveaux de gris, le filtre médian transformera I en une image J à niveaux de gris, telle que pour tout pixel p , le niveau de gris $J(p)$ est la médiane des niveaux de gris $I(q)$ des pixels q dans la fenêtre $W(p)$:

$$J(p) = \text{med} (I(q) \mid q \text{ dans } W(p)) .$$

Notons que ce calcul de $J(p)$ se fait en parallèle, en d'autres termes le calcul en un pixel p n'influe pas sur le calcul en un pixel voisin q . Vu que la médiane, en tant que mesure statistique, représente une alternative à la moyenne, le filtre médian constituera une forme de lissage qu'on peut utiliser à la place du lissage linéaire. Les deux types de filtres ont des propriétés différentes, qui selon les cas peuvent donner l'avantage à l'un ou l'autre.



Figure 1.10 Filtrage médian

1.3.11 La binarisation :

La binarisation (le seuillage) est la technique de classification la plus simple où les pixels de l'image sont partagés par un seul seuil S en deux classes : ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et S est retenue et l'autre non, ou vice-versa.

Soit l'image I ($M \times N$), supposons que $f(x, y)$ représente le niveau de gris du pixel aux coordonnées (x, y) , $0 \leq x \leq M$, $0 \leq y \leq N$ et S est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à S et les autres ayant le niveau de gris supérieur à S sont des pixels du fond. Alors, l'image binarisée g est déterminée par les pixels (x, y) dont la valeur est :

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) > S \\ 0 & \text{si } f(x,y) \leq S \end{cases}$$

Travailler avec des images binaires est plus simple, dans une certaine mesure, qu'avec des images en niveaux de gris, pour les raisons suivantes :

- on peut très facilement identifier des objets à condition qu'ils soient bien séparés (on appelle ça "l'analyse par composantes connexes") ;
- on a beaucoup moins d'informations à traiter, donc les calculs sont globalement plus rapides;
- si notre image est bien binarisée, les algorithmes sont dans l'ensemble très robustes.

Évidemment, il y a un certain nombre d'inconvénients à binariser une image comme :

- il faut choisir un seuil correct, ce qui n'est pas toujours facile, comme vous allez le constater;
- on passe d'un espace de couleurs de dimension 255 à un nouvel espace de dimension 2 : on perd énormément d'informations qui pourraient s'avérer utiles !
- la plupart des algorithmes de binarisation sont très sensibles au bruit, on est souvent obligé de lisser l'image en prétraitement.

Donc binariser ou non, c'est un choix qui dépend beaucoup du type d'images que l'on traite, ainsi que du problème que l'on cherche à résoudre.

Il existe deux grandes techniques de sélection du seuil S : global, local (adaptative).

a. La binarisation globale :

Dans la méthode de binarisation globale un seuil unique s est calculé à partir d'une mesure globale sur toute l'image. Il nous permet de décider l'appartenance d'un pixel à l'objet ou au fond.

$$S = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) .$$

Si $f(x,y) > S$ alors $g(x,y)=255$.

Si $f(x,y) \leq S$ alors $g(x,y)= 0$.



Figure 1.11 Binarisation globale

b. La binarisation locale :

Dans ce type de seuillage, le seuil n'est pas une valeur unique pour toute l'image, mais il dépend essentiellement à la région de taille fixée par l'utilisateur (la marge).

Donc chaque pixel doit être comparé au seuil calculé par sa région.

La table suivante représente le voisinage de pixel de coordonnées (x,y) :

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

Tableau 1.2 Voisinage de f(x,y)

$$S = \frac{1}{(2n+1)^2} \sum_{k=-n}^n \sum_{l=-n}^n f(x+k, y+l).$$

Avec : n = la marge

Si $f(x,y) > S$ alors $g(x,y)=255$.

Si $f(x,y) \leq S$ alors $g(x,y)= 0$.



Figure 1.12 Binarisation locale

1.3.12 La détection de contour :

La détection de contours est une technique de réduction d'information dans les images, qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image.

Si les structures extraites sont simples à manipuler (courbes fines, régulières, stables...), elles peuvent être utiles pour la mise en correspondance d'images (robotique, indexation,...)

La méthode que nous avons adaptée celle du suivi de contour permet de représenter le contour détecté par un vecteur ordonné dans le sens des pixels détectés 1^{er}, ..., Nième. Cette méthode est très rapide car elle ne nécessite que peu de calculs à chaque étape. Elle présente l'avantage de détecter les pixels frontières suivant toutes les directions.

Tous les pixels seront traités une seule fois. La méthode permet de détecter les contours d'objets d'images binaires pouvant être représentés de la façon suivante :

Les objets sont constitués de pixels « noir » de valeur '1' tandis que le fond est formé de pixels « blanc » de valeur '0'.

Ces contours sont classés selon deux catégories :

- Ceux qui englobent un objet, ils sont dit extérieurs.
- Ceux qui sont noyés dans un objet, ce sont des contours intérieurs.

Après la détection du premier pixel contour, la nature de transition donne le type du contour :

- Si cette transition est de la nature fond forme (0-1), c'est un contour de type externe.
- Sinon, le contour est de type interne(1,0).

Cette méthode se résume de balayer l'image jusqu'à trouver une transition, c'est le premier pixel contour. Étiqueter les huit voisinages de ce pixel, tester si ces pixels appartiennent au contour, à chaque fois qu'un pixel objet est rencontré, on lui étiquette ses huit voisinages et continuer jusqu'à la rencontre du premier pixel trouvé [1]. On peut réaliser cette méthode en suivant les étapes suivantes :

1^{ère} Étape : Balayer l'image jusqu'à la rencontre d'un pixel objet.

2^{ème} Étape : Balayer ses voisins dans le sens horaire.

3^{ème} Étape : Evaluer les coordonnées de ses voisins.

4^{ème} Étape : Si le K ème voisin est un pixel noir, il sera le point suivant du contour et on définit ses coordonnées en ce point.

5^{ème} Étape : Si le K ème voisin est un pixel blanc, prendre le voisin suivant et aller à l'étape 3.

6^{ème} Étape : Continuer le processus jusqu'à la rencontre du premier point contour détecté.

7^{ème} Étape : Affecter à tous les points contours découverts une valeur différente de 0 et de 1 comme intensité du contour et numéro du contour. Cela permet d'éviter la détection du même contour une autre fois.

8^{ème} Étape : Poursuivre le balayage de l'image pour détecter de nouveaux contours jusqu'à la fin de l'image



Figure 1.13 Détection de contour

1.4. Conclusion :

En effet, l'image est un objet à deux dimensions, censé représenter un espace à trois dimensions, ce qui a deux conséquences majeures :

- Le volume des données à traiter est beaucoup plus important ;
- La structure de ces données est nettement plus complexe.

Il en résulte que la manipulation, le stockage et la représentation de ces données se heurtent à certaines limitations. Grâce au traitement d'image, ces contraintes sont levées ou contournées. En effet, 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.

Le traitement d'images trouve son application dans des domaines extrêmement variés de l'Industrie et de la Recherche. Ces méthodes sont utilisées dans de nombreuses disciplines scientifiques, citons en particulier les sciences des matériaux, les sciences de la terre, la géographie (dont la cartographie et la géomorphologie), la robotique (pour le tri et la vérification de pièces électroniques) ou bien encore dans des domaines aussi variés que ceux qui ont trait à l'astronomie, l'identification, la pharmacologie.

En biologie, ces techniques sont particulièrement bien adaptées à l'étude des images macroscopiques (radiologie, IRM, imagerie fonctionnelle...) mais aussi à celle des images microscopiques et constituent un outil performant pour visualiser et décrire des structures. La représentation numérique permet notamment de combiner des images obtenues sous différents modes.

2.1 Introduction :

La vision nous permet de percevoir et d'interpréter le monde qui nous entoure. La vision artificielle a pour but de reproduire certaines fonctionnalités de la vision humaine au travers de l'analyse d'images. C'est un problème difficile en raison du fait que l'information disponible n'est pas parfaite (numérisation des capteurs, déformation des objectifs, bruitages).

L'analyse d'images regroupe plusieurs disciplines que l'on classe en deux catégories :

- Les processus de bas-niveaux : qui nécessitent très peu d'informations sur le contenu des images. Il s'agit ici des processus de filtrage, d'amélioration et de restauration d'images, processus que nous regroupons sous le terme traitement d'images, ainsi que d'extraction d'indices.
- Les processus de haut-niveaux : qui fonctionnent en aval de ceux de bas niveaux, et qui peuvent nécessiter des informations sur le contenu des images.

Il s'agit de la reconstruction tri -dimensionnelle, la reconnaissance de formes, les processus cognitifs de façon générale.

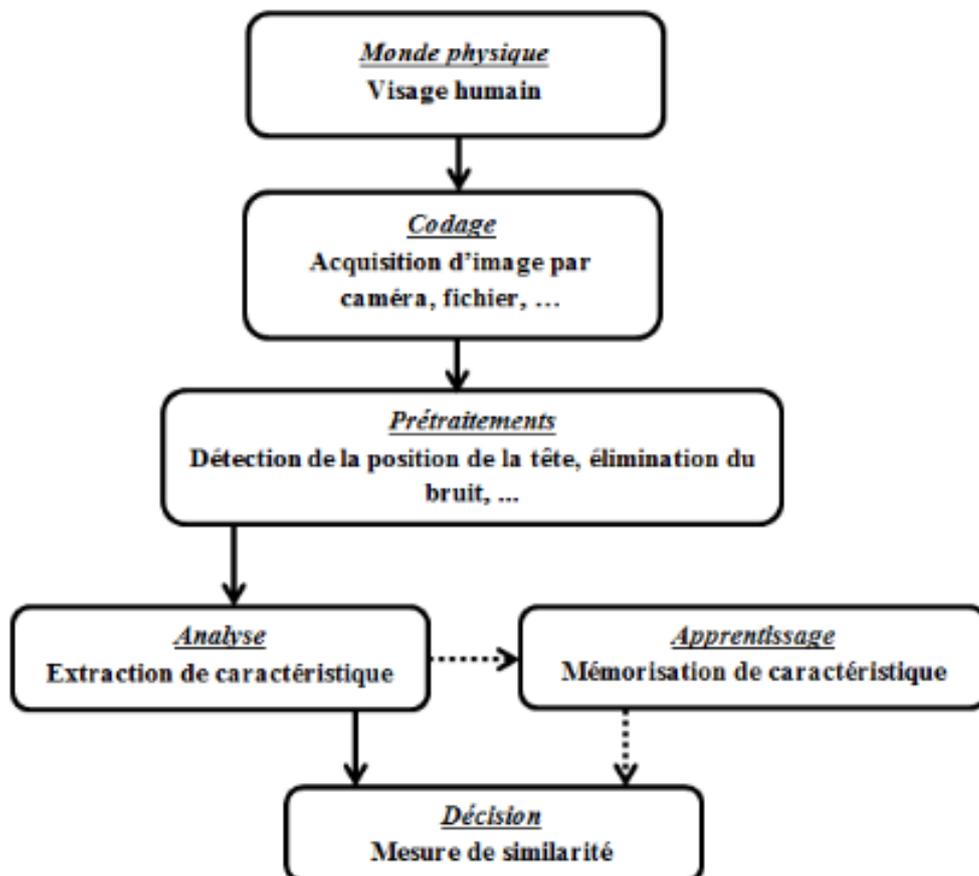


Figure 2.1 Schéma synoptique de la chaîne de traitement d'image.

2.2 Les images numériques :

2.2.1 Définition d'une image numérique :

Une image numérique est un ensemble des points. Ces points sont appelées « pixels » qui une abréviation en anglais « Picture élément », en fait ces points des petites carrées. La représentation graphique d'une image peut se faire d'une palette de gris allant de noir jusqu'au blanc. Il existe deux sortes d'images numériques : les images matricielles et les images vectorielles.

2.2.2 Image vectorielle :

Dans une image vectorielle les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique. Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans. Ces images présentent deux avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

2.2.3 Image matricielle :

Une image matricielle est formée d'un tableau de points ou pixels. Plus la densité des points est élevée, plus le nombre d'informations est grand et plus la résolution de l'image est élevée. Corrélativement la place occupée en mémoire et la durée de traitement seront d'autant plus grandes.

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.

2.2. 4 Résolution d'une image :

Une image est divisée en points ou pixels. Considérons une image de 10 cm sur 10 cm. Avec une résolution très faible de 10 pixels par cm, elle est codée sur $100 \times 100 = 10000$ pixels. Et avec une résolution convenable de 100 pixels par cm (un pixel mesure 0,1 mm), elle serait codée sur $1000 \times 1000 = 1000000$ pixels = 1 M pixels. Le symbole M signifiant million.

En général on utilise l'unité de longueur anglo-saxonne le pouce ou inch. La résolution d'une image s'exprime alors en pixels par pouce (ppp) ou dots per inch (dpi) en anglais.

Les figures suivantes illustrent la même image avec deux résolutions différentes :



Figure 2.2 pixels par pouce soit environ 7 pixels par cm.

Dans ce cas on observe l'effet de pixellisation.



Figure 2.3 pixels par pouce soit environ 30 pixels par cm.

Cette dernière résolution correspond à celle d'un écran, elle est idéale pour visualiser une image sur l'écran. Plus les nombres des pixels est élevée par cm plus la résolution est élevée, plus on a une bonne qualité d'image.

2.2.5 Le codage de l'information d'une image numérique :

L'information est codée en binaire. Le support évolue mais le principe est toujours le même, un même élément peut se trouver dans 2 états différents stables. Il constitue une mémoire élémentaire ou bit. Conventionnellement on attribue le symbole 0 à l'un de ces 2 états et le symbole 1 à l'autre.

-A l'aide de 1 bit on a donc 2 possibilités 0/1 ou ouvert/fermé ou noir/blanc.

-A l'aide de 2 bits on a $2 \times 2 = 4$ possibilités : 00/01/10/11

-A l'aide de 3 bits on a $2 \times 2 \times 2 = 8$ possibilités : 000/001/010/011/100/101/110/111.

L'image de 10000 pixels ainsi codée occupe $10000 \times 3 = 30000$ octets. C'est ce codage de la couleur qui est utilisé par la plupart des écrans d'ordinateurs actuellement. On constate qu'il est très gourmand en mémoire. Pour faciliter le stockage des images en mémoire on utilise d'autres formes de codage.

Couleur	noir	bleu pâle	vert pâle	rose	bleu	vert	rouge	blanc
R	0	120	120	150	120	120	255	255
V	0	120	150	120	120	255	120	255
B	0	150	120	120	255	120	120	255

Tableau 2.1 Définition des Grandeurs des quelques couleur.

2.3 Les couleurs :

2.3.1 La synthèse additive de la lumière, ou le mode RVB :

L'image est obtenue par superposition de trois rayonnements lumineux : le rouge (R), le vert (V) et le bleu (B). Dans le cas d'un écran cathodique, ces 3 rayonnements sont obtenus en bombardant les luminophores photosensibles de l'écran.

Une image RVB est composée de la somme de trois rayonnements lumineux rouge, vert, et bleu dont les faisceaux sont superposés. A l'intensité maximale ils produisent un ray de lumière blanche, et à l'extinction une zone aussi noire que l'éclairage ambiant le permet (c'est la raison pour laquelle vous avez installé le moniteur de votre ordinateur dans une pièce où règne, autant que faire se peut une certaine pénombre, et qu'un rayon de soleil qui frappe en plein un écran le rend presque illisible).

Si vous mettez le nez sur l'écran de votre moniteur (ou mieux encore sur celui de votre téléviseur, dont le pas est plus grossier) vous distinguerez les trois sources RVB en forme de nid d'abeille, ou de grille, qui donnent l'illusion d'un fond blanc. En fait, croyant voir un point blanc, vous percevez simultanément trois lumières rouge, vert et bleu. La gamme des couleurs reproductibles par ce mode, quoique conditionnée par la qualité du matériel employé, est très étendue, et reproduit bien les couleurs saturées. En contrepartie, elle convient mal à la restitution des nuances délicates des lumières intenses et des tons pastels.

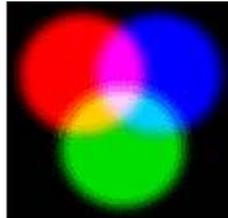


Figure 2.4 Les couleurs RVB.

2.3.2 La synthèse soustractive de la lumière, ou le mode CMJN :

Les couleurs sont obtenues par mélange des pigments colorés. Lorsqu'ils sont éclairés par de la lumière blanche, les pigments absorbent une partie de la lumière qu'ils reçoivent ce qui les fait apparaître colorés. Les trois couleurs "primaires" sont le cyan (C), le magenta (M) et le jaune (J). Les autres sont obtenues par mélange.

Les encres déposées sur le papier agissent comme des filtres qui absorbent la lumière. Leur superposition devrait produire théoriquement un noir total : plus de lumière, ce qui n'est, en pratique, pas le cas. Les trois encres additionnées ne donnent pas plus qu'un brun sombre, que l'on se doit de renforcer par un quatrième passage d'encre noire (que l'on note donc en toute logique : "N", comme Noir). Il faut donc garder à l'esprit qu'une impression, même d'excellente qualité, ne saurait restituer de blanc plus lumineux que celui du papier utilisé, ni de noir plus puissant que celui produit par l'addition des densités des encres mises en œuvre. La gamme des couleurs reproductibles par le mode CMJN est plus restrictive que celle de la gamme RVB. Elle est, de surcroît, particulièrement sensible aux variations inévitables dues aux conditions mécaniques et physiques de l'impression en machine.

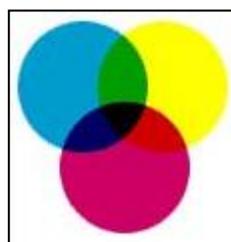


Figure 2.5 Les couleurs CMJN.

2.3.3. Images aux Niveaux de Gris :

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise entre 0 et 255. Par convention, la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale). Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

2.4. Prétraitement :

Prétraitement consiste à améliorer la qualité de l'image en éliminant les défauts dus à l'éclairage et au processus d'acquisition. Les prétraitements d'une image consistent à effectuer des opérations visant à :

- * améliorer sa qualité visuelle
- * restaurer l'image en éliminant les défauts et les bruits et en renforçant l'information utile qu'elle contient.
- * Compresser l'image afin de réduire l'espace nécessaire à son stockage et diminuer le temps de sa transmission.

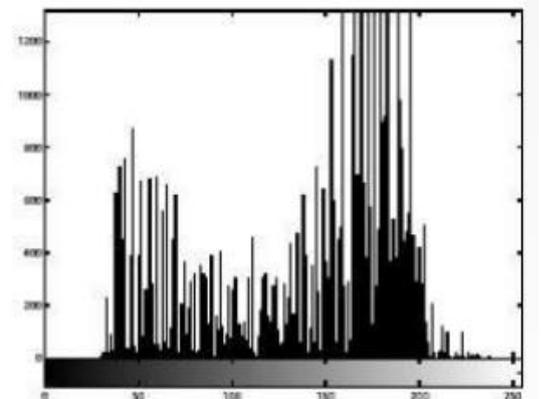
2.4.1. Histogramme de l'image :

On veut souvent avoir une information sur la distribution des valeurs des pixels (niveaux) dans une image, pour cela on utilise souvent une table qui donne le nombre de pixels de chaque niveau dans l'image. Cette table, souvent représentée graphiquement, est appelée Histogramme de l'image, est notée $h(v)$.

A chaque valeur on associe le nombre de pixels dans l'image ayant cette valeur.



(a)



(b)

Figure 2.5 a) Image originale, b) Histogramme de l'image.

a) Luminance ou brillance d'une image :

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 : $I_0(i, j) = I(i, j) + b$

Dans les deux images suivantes, seule la luminance est différente

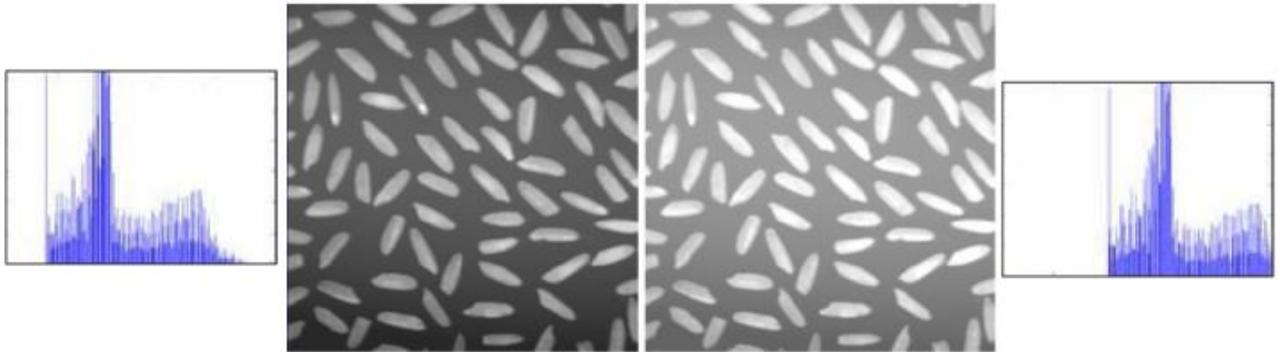


Figure 2.6 Exemple d'une luminance d'image.

b) Contraste :

Le contraste peut être défini de plusieurs façons :

- Variance des niveaux de gris (N nombre de pixels dans l'image).

$$\frac{1}{N} \sum_{n=1}^N (I(i, j) - Moy)^2 \quad (2.1)$$

- Variation entre niveaux de gris max et min.

$$\frac{\max[I(i, j)] - \min[I(i, j)]}{\max[I(i, j)] + \min[I(i, j)]} \quad (2.2)$$

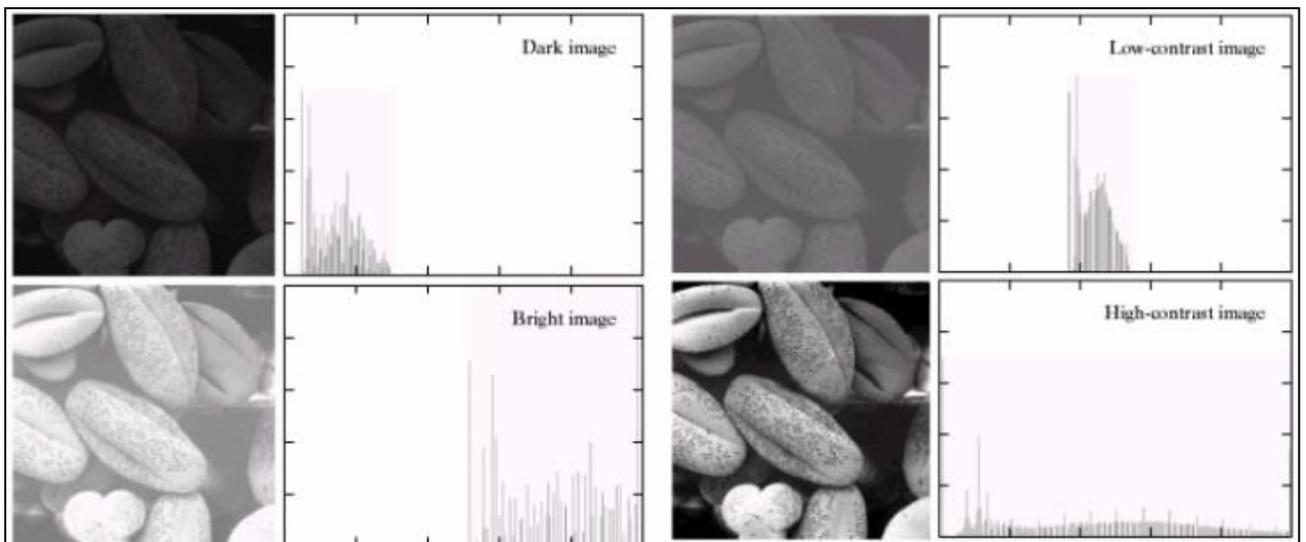


Figure 2.7 Exemple d'une image Contrasté.

c) Egalisation d'histogramme :

On cherche à aplanir l'histogramme :

1. Calcul de l'histogramme $h(k)$ avec $k \in [0, 255]$.
2. Histogramme cumulé.

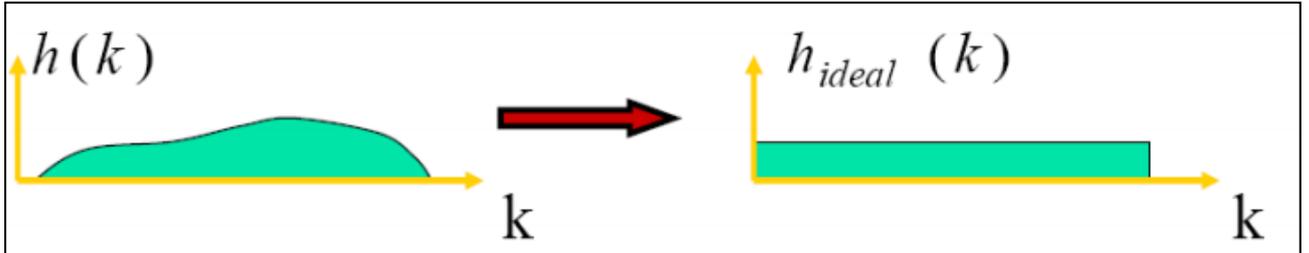


Figure 2.8 Egalisation d'histogramme

$$C(k) = \sum_{i=1}^k (h(i)) \quad (2.3)$$

3. Transformation des niveaux de gris de l'image.

$$I'(x, y) = \frac{C(I(x, y)) * 255}{N} \quad (2.4)$$

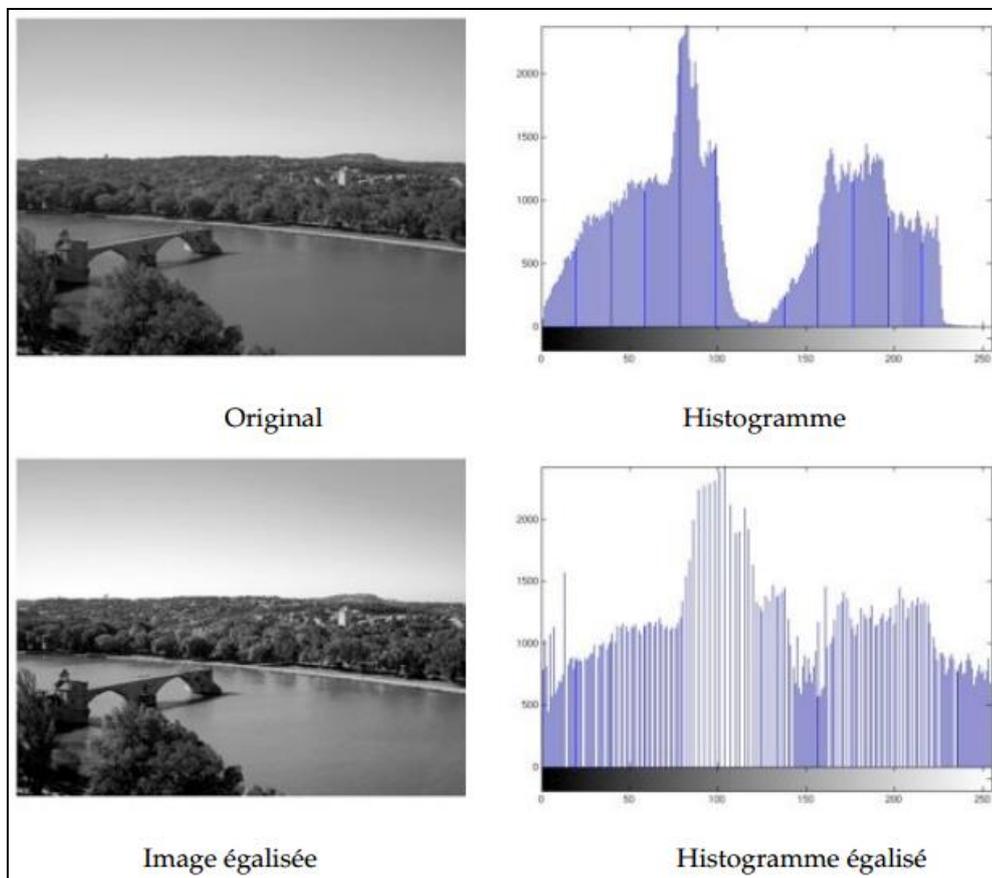


Figure 2.9 Exemple d'un egalisation d'un histogramme.

2.5 La binarisation :

2.5.1 Le Seuillage (threshold) :

Le but de la binarisation d'une image est d'affecter un niveau uniforme aux pixels pertinents et d'éliminer les autres. Le seuillage consiste à affecter le niveau 255 aux pixels dont la valeur est supérieure à un seuil S et 0 le niveau aux autres. Le graphe de la transformation correspondante est le suivant.

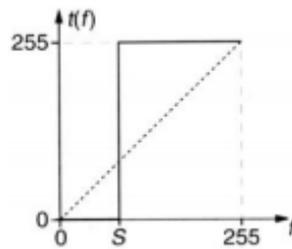


Figure 2.10 Fonction « seuillage ».

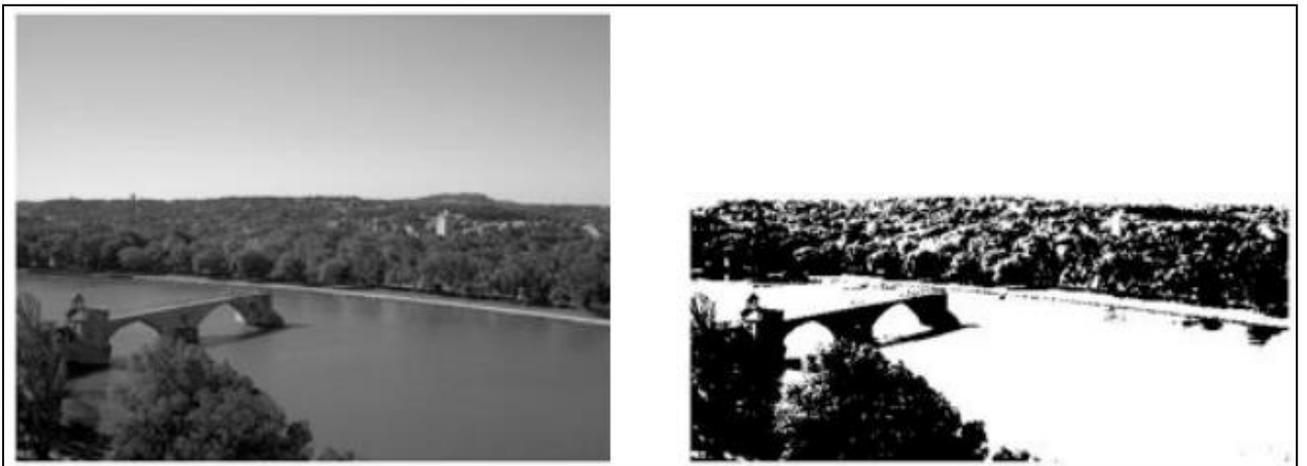


Figure 2.11 Exemple d'un Seuillage à 70.

Extraction d'une fenêtre d'intensité avec la transformation décrite ci-dessous, la nouvelle image ne visualise que les pixels dont le niveau d'intensité appartient à l'intervalle $[a, b]$. Sous réserve d'une connaissance à priori de la distribution des niveaux de gris des objets de l'image originale, cette technique permet une segmentation d'objets particuliers de l'image.

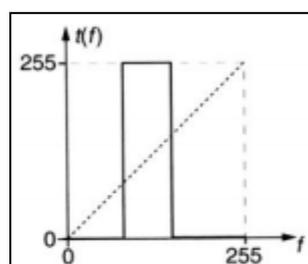


Figure 2.12 Fonction « fenêtre d'intensité »

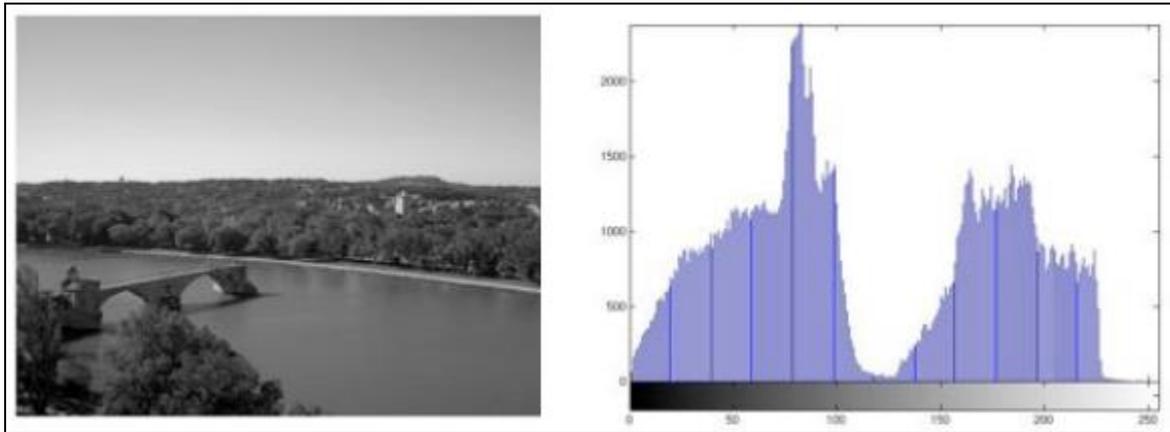


Figure 2.13 l'Image original et leur Histogramme.

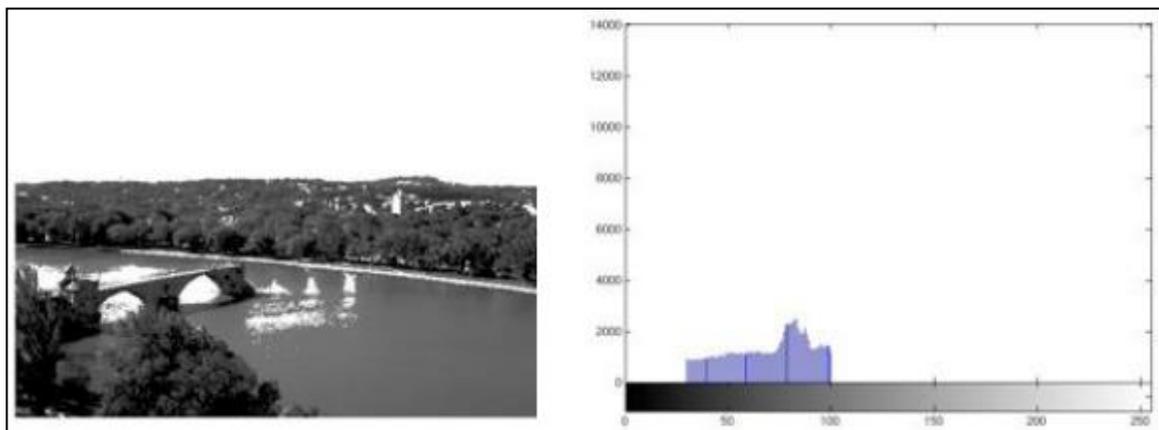


Figure 2.14 Seuillage avec fenêtre d'intensité entre 30 et 100

2.5.2 Définition la méthode d'OTSU

La méthode d'OTSU est utilisée pour effectuer un seuillage automatique à partir de la forme de l'histogramme de l'image. Cette méthode nécessite donc le calcul préalable de l'histogramme de l'image. L'algorithme suppose alors que l'image à binariser ne contient que deux classes, (Les objets et l'arrière-plan). L'algorithme itératif calcule alors le seuil optimal T qui sépare ces deux classes afin que la variance intra-classe soit minimale et que la variance inter-classe soit maximale.

2.5.3 Implémentation de la méthode d'OTSU sur MATLAB.

Il existe plusieurs façons d'implémenter la méthode d'OTSU sur MATLAB. Dans le cas présent, vous allez voir deux méthodes distinctes offrant le même résultat mais avec des caractéristiques différentes.

La première méthode consiste à utiliser la fonction « graythresh ». Cette méthode est la plus simple et la plus rapide car elle nécessite peu de lignes de commande et pas de calcul.

La seconde méthode consiste à implémenter manuellement la méthode d'OTSU. C'est une méthode plus complexe, elle nécessite le calcul de la variance ainsi que de la moyenne de chaque classe pour chaque seuil T possible. Cette méthode est plus calculatoire que la première. On est donc plus amené à faire des erreurs lors de l'implémentation.

Le seuillage est une technique de segmentation simple, non contextuelle et efficace :

- Seuillage d'intensité
- Classification des pixels en deux catégories
- Création d'une image binaire (binarisation)

2.5.3 Réduction avec seuillage :

On considère une image binaire dont un pixel de fond porte la valeur '0' et un pixel de l'objet porte la valeur '1'. On divise cette image en blocs de taille 2×2 pixels. Chaque bloc de 4 pixels sera remplacé par un seul pixel, la valeur de ce pixel est soit '0' ou bien '1' dépendamment à la valeur choisie pour le seuil S avec : $1 \leq S \leq 4$.

Si la valeur de la somme des 4 pixels est supérieure ou égale à S : Le bloc est remplacé par '1'

Si non : Le bloc est remplacé par '0'.

Après une seule opération de réduction (appelé aussi réduction 4×1) avec un seuil donné le nombre de pixels de l'image est réduit de 2^n à 2^{n-2} [11]. Le concept de cette opération est illustré sur la (figure 2.14).

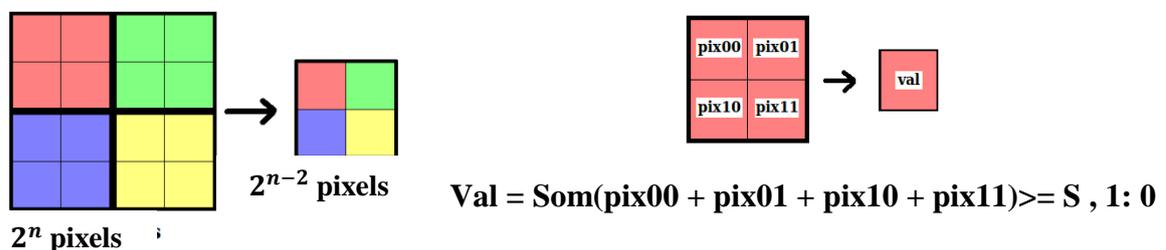


Figure 2.15: illustration de réduction avec seuil.

2.6 Le bruit :

En traitement d'image, toute brusque fluctuation d'un pixel par rapport à ses voisins étant considéré comme un bruit provoquant ainsi la dégradation de la qualité d'image que se soit de luminance ou de contraste. Ses sources les plus fréquentes sont :

2.6.1 Bruit lié aux dispositifs d'acquisition

La caméra, les câbles et tous autres blocs de connexion engendrent un bruit qui a un effet multiplicatif, pour cela, il serait plus judicieux de mettre les sources qui produisent plus de bruit dans les étages inférieurs pour diminuer leur contribution générale au bruit général.

2.6.2 Bruit lié à la scène :

Il se trouve parfois que la scène soit polluée de poussière, ou couverte de nuages, brouillard ou buée qui dégrade la qualité des images. Cependant, là encore, la connaissance a priori du phénomène perturbateur permet d'envisager une modélisation et donc une correction systématique. Ce type de bruit a un effet de bruit additif.

2.6.3 Bruit lié à la transmodulation :

Les images qui sont capturées à partir d'une séquence vidéo qui regroupe le son et l'image dans un même signal sont atteintes par un bruit dû à la transmodulation qui se produit du fait qu'on utilise une double modulation : modulation audio et vidéo. Il arrive, ainsi, des cas où des composantes du signal audio brulent notre signal vidéo.

2.6.4 Bruit lié à l'échantillonnage :

Une image est un signal discret. Il est donc nécessaire de passer du domaine continu au domaine discret au niveau de la scène et de l'intensité lumineuse émanant de celle-ci. De manière générale, le spectre des intensités lumineuses noir et blanc est quantifié sur 256 niveaux de gris différents si on admet que chaque pixel est codé sur un octet.

Pour éliminer le bruit et rendre l'image plus nette, on utilise les techniques de filtrage, qui seront abordées dans la section suivante.

2.7 Le filtrage :

2.7.1 le Filtrage linéaire :

Un filtre linéaire est un système linéaire continu et invariant. En d'autres termes, on se donne deux espaces normés X et Y ainsi qu'un opérateur linéaire continu, invariant par translation $A : X \rightarrow Y$. Un filtre idéal est un filtre qui élimine totalement les bandes de fréquence indésirables sans transition et sans déphasage dans les bandes conservées.

2.7.2 Filtrage médian :

Le célèbre format bitmap, qui tire son nom de l'anglais « bitmap » pour « carte de bits » montre qu'une image est avant tout un domaine spatial sur lequel on peut se promener avec la souris de l'ordinateur : les distances en pixels dans l'image I sont des lors liées aux distances réelles en mètres dans la scène réelle S.

La fréquence spatiale est un concept délicat qui découle du fait que les images appartiennent au domaine spatial. Pour commencer on peut rappeler que la fréquence est une grandeur qui caractérise le nombre de phénomènes qui se déroulent au cours d'un temps donné: en voiture le long d'une route vous voyez 2 bandes blanches PAR seconde : c'est une fréquence temporelle.

Il est ensuite facile de comprendre que ce concept de fréquence « temporelle » peut aussi se traduire en disant qu'il y a 200 bandes blanches PAR kilomètre : c'est une fréquence spatiale.

2.7.3 Filtrage spatial

Le filtrage spatial est essentiellement une opération de convolution. Si f est l'image à filtrer (ou à rehausser) et g le filtre spatial (ou PSF ou masque) on a :

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

G est la fonction de transfert du filtre.

Une image numérique étant essentiellement discrète (pixels et niveaux de gris) nous allons présenter les filtres dans le cas discret. Dans tout ce qui suit x et y sont des entiers (coordonnées des pixels) et f est a valeurs entières (dans {0, ..., 255}).

On ne fait pas en général une convolution globale mais une transformation basée sur le voisinage d'un point (x, y) :

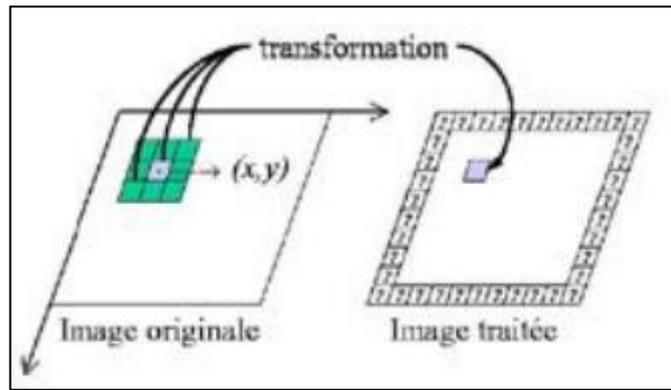


Figure 2.16 Le noyau de convolution du filtre κ est a support compact inclus dans $[x_1, x_2] \times [y_1, y_2]$:

$$g(x, y) = (f * \kappa)(x, y) = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} f(x - i, y - j) \kappa(i, j) .$$

Généralement le filtre est de dimension d impaire et est symétrique. Dans ce cas

$$[x_1, x_2] = [y_1, y_2] = [-d/2, d/2]$$

$$(f * \kappa)(x, y) = \sum_{i=-(d-1)/2}^{(d-1)/2} \sum_{j=-(d-1)/2}^{(d-1)/2} f(x + i, y + j)$$

❖ Exemple d'un Filtrés séparables.

$$\begin{bmatrix} a & b & c \end{bmatrix} \otimes \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} a\alpha & b\alpha & c\alpha \\ a\beta & b\beta & c\beta \\ a\gamma & b\gamma & c\gamma \end{bmatrix}$$

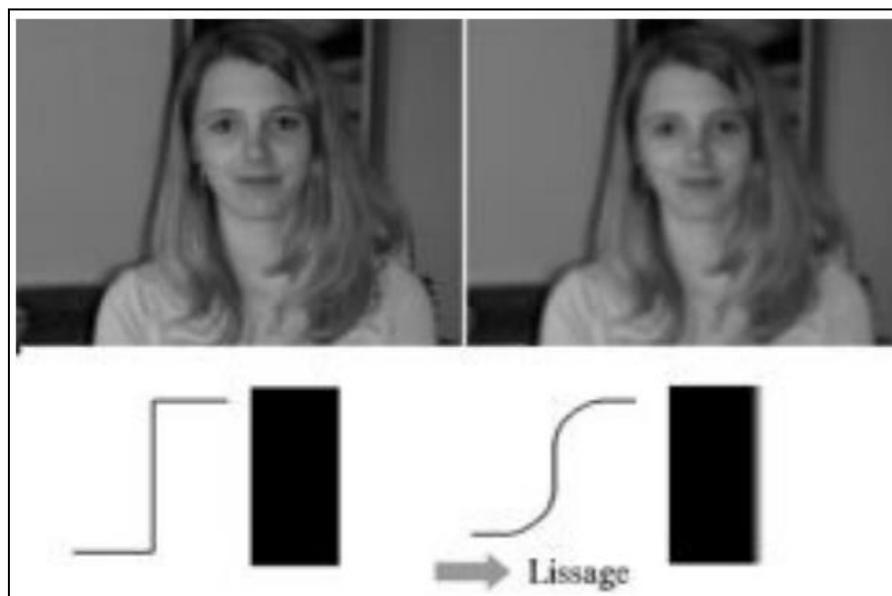


Figure 2.17 Filtrés séparables.

2.7.4 Filtres médians :

Ce ne sont pas des filtres de convolution, ni des filtres linéaires.

$$g(x, y) = \text{médian} \{ f(n, m) \mid (n, m) \in S(x, y) \},$$

où $S(x, y)$ est un voisinage de (x, y) .

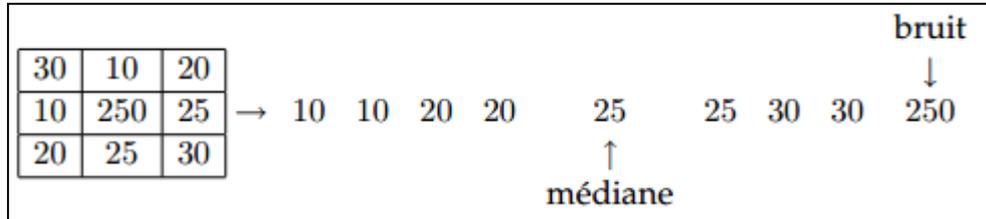


Figure 2.18 Exemple d'un filtre médiane.

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.

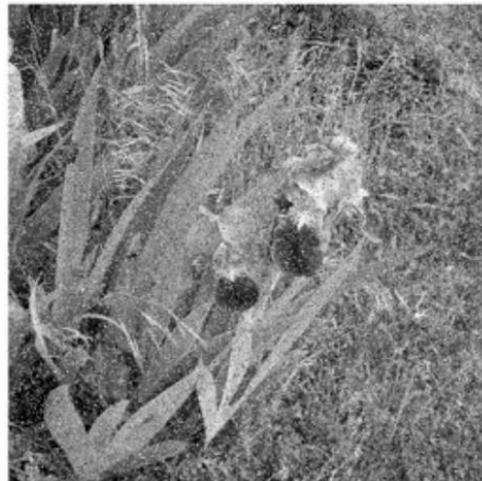


Figure 2.19 Image bruitée « Poivre et Sel »



(a)



(b)

Figure 2.20 a) Filtre de moyenne : rayon 3, b) Filtre de moyenne : rayon 5



Figure 2.21 Filtre de moyenne : rayon 7



(a)



(b)

Figure 2.22 a)Filtre median : rayon 3, b)Filtre médian : rayon 5.



Figure 2.23 Filtre médian : rayon 7

Si le bruit $Pet S$ est supérieur à la moitié de la dimension du filtre, le filtrage est inefficace.

2.8 Etiquetage :

L'étiquetage des pixels (L'extraction des composantes connexes, procédure également appelée capture des connexités), est largement utilisé en Reconnaissance des Formes (RdF) pour segmenter les images binaires. La technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé (figure 2.31). La 4-connexité est distinguée de la 8-connexité suivant que le critère de voisinage comprend les 4 ou les 8 voisins d'un pixel.

Il existe deux principales méthodes pour accomplir cette tâche :

- la première est basée sur une procédure de suivi de contour : en parcourant le contour d'un objet et en revenant au point de départ, une composante connexe est délimitée, à l'exclusion cependant des contours intérieurs correspondant aux éventuels trous.

- la seconde méthode procède par une propagation d'un étiquetage des pixels lorsque l'on effectue un balayage des lignes et des colonnes de l'image [1].

Nous avons élaboré un algorithme de ce type fonctionnant en plusieurs passes (2 ou 4 passes), suivant le critère de 4-connexité.

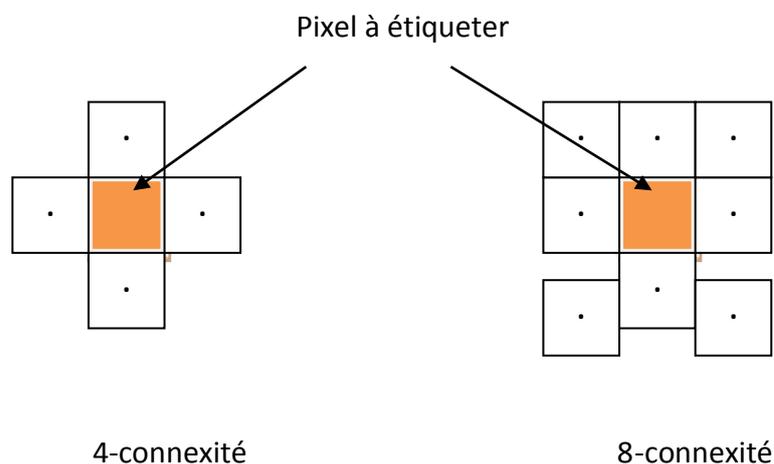


Figure 2.24: types de connexité.

Algorithme : Approche par double parcours (two-pass)

Pour l'explication de cette méthode on note B l'image binaire à étiqueter et par L le résultat obtenu, c'est la matrice (ou carte) des étiquettes [1].

Premier parcours de l'image, dans le sens classique (raster-scan order) :

<ul style="list-style-type: none"> Étape 4 affecter au pixel l'étiquette de son voisin gauche 	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	1	1	1																																																																																			
0	1	1	0	0	0	0	1																																																																																			
0	1	0	0	1	1	1	1																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	2	2	2																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
<ul style="list-style-type: none"> Étape 5 nouvelle étiquette (nbLabels=3) 	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	2	2	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	1	1	1																																																																																			
0	1	1	0	0	0	0	1																																																																																			
0	1	0	0	1	1	1	1																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	2	2	2																																																																																			
0	3	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
<ul style="list-style-type: none"> Étape 6 les voisins haut et gauche du pixel suivant à 1 dans B sont étiquetés différemment ⇒ affecter au pixel l'étiquette minimale 	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	2	2	2	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	1	1	1																																																																																			
0	1	1	0	0	0	0	1																																																																																			
0	1	0	0	1	1	1	1																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	2	2	2																																																																																			
0	3	1	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
<ul style="list-style-type: none"> Étape 7 affecter au pixel l'étiquette de son voisin haut 	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	2	2	2	0	3	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	1	1	1																																																																																			
0	1	1	0	0	0	0	1																																																																																			
0	1	0	0	1	1	1	1																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	2	2	2																																																																																			
0	3	1	0	0	0	0	2																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
<ul style="list-style-type: none"> Étape 8 affecter au pixel l'étiquette de son voisin haut 	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	2	2	2	0	3	1	0	0	0	0	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	1	1	1																																																																																			
0	1	1	0	0	0	0	1																																																																																			
0	1	0	0	1	1	1	1																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	2	2	2																																																																																			
0	3	1	0	0	0	0	2																																																																																			
0	3	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
<ul style="list-style-type: none"> Étape 9 nouvelle étiquette (nbLabels=4) 	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td></tr> <tr><td>0</td><td>3</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	0	1	1	0	2	2	2	0	3	1	0	0	0	0	2	0	3	0	0	4	0	0	0	0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	1	1	1																																																																																			
0	1	1	0	0	0	0	1																																																																																			
0	1	0	0	1	1	1	1																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			
0	0	1	1	0	2	2	2																																																																																			
0	3	1	0	0	0	0	2																																																																																			
0	3	0	0	4	0	0	0																																																																																			
0	0	0	0	0	0	0	0																																																																																			

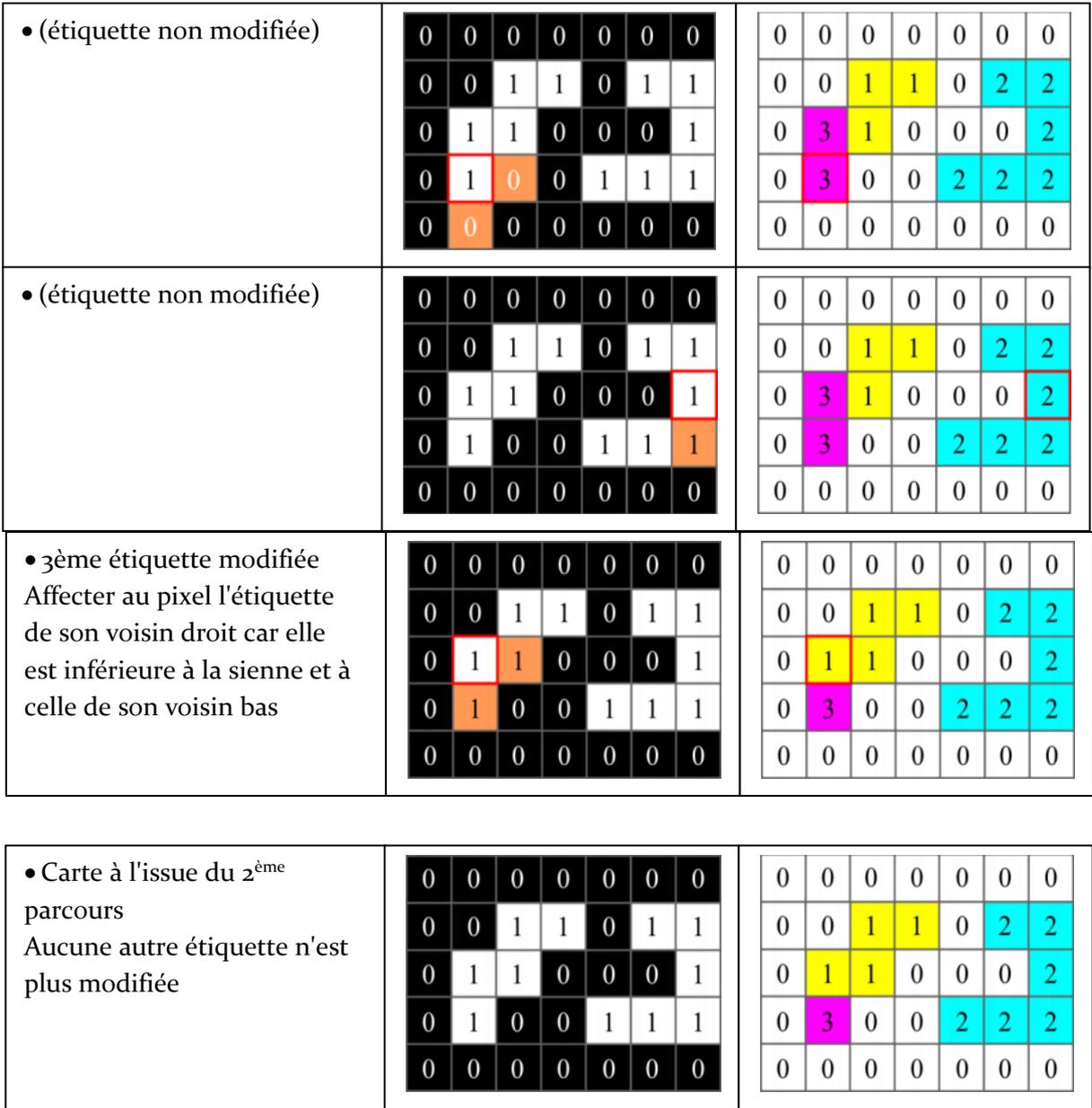


Figure 2.27 Deuxième passage de l'étiquetage

L'opération d'étiquetage (en deux passages) se répète jusqu'à ce qu'il n'y ait plus de changement sur l'image.

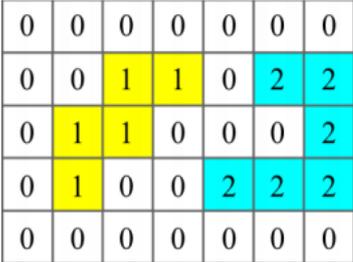


Figure 2.28 Carte finale des étiquètes.



Figure 2.29 : Exemples d'étiquetage.

2.9 Conclusion :

Les méthodes de traitement d'images désignent les outils utilisés pour transformer une image numérique en une nouvelle image, ou même combiner plusieurs images entre elles. Elles permettent de compresser l'information en vue d'un transfert informatique rapide par les réseaux informatiques, de restaurer une image, d'améliorer la visualisation de certains détails, de combiner des images de sources.

Restauration, amélioration et segmentation Les sources de dégradation d'une image (flous, distorsion géométrique, hétérogénéité de l'éclairage) sont multiples et dépendent du processus de formation de l'image.

L'analyse d'images est la dernière étape, elle consiste à extraire de façon quantitative l'information contenue dans une image, et à donner une description objective, finalisée et précise de l'image, ou de certains éléments. Elle part d'une image et conduit à l'évaluation de mesures. Elle conduit à une évaluation quantitative ou à une reconnaissance de formes. Elle se pose chaque fois que l'on veut préciser une observation visuelle ou la remplacer par une analyse automatique.

3.1 Introduction :

Les procédures de traitement des séquences d'images exposées ont été développées dans l'optique d'une mise en œuvre orientée temps réel afin d'être utilisables dans le contexte général de la surveillance du trafic.

Dans le cadre d'une automatisation, le temps réel est en fait le temps maximum que l'on peut octroyer au système de commande pour qu'il génère l'action nécessaire en fonction de l'environnement. [8]

Un langage de programmation sur MATLAB analyse les images. La méthode de détection et de comptage des objets qui passent sur le tapis roulant est visualisée sur la figure 3.1. Le système comporte de quatre étapes essentielles :

- acquisition de l'image par un Caméra ;
- Prétraitement de l'image capturé ;
- Spécifier la ligne virtuel ;
- Comptage des Objets.

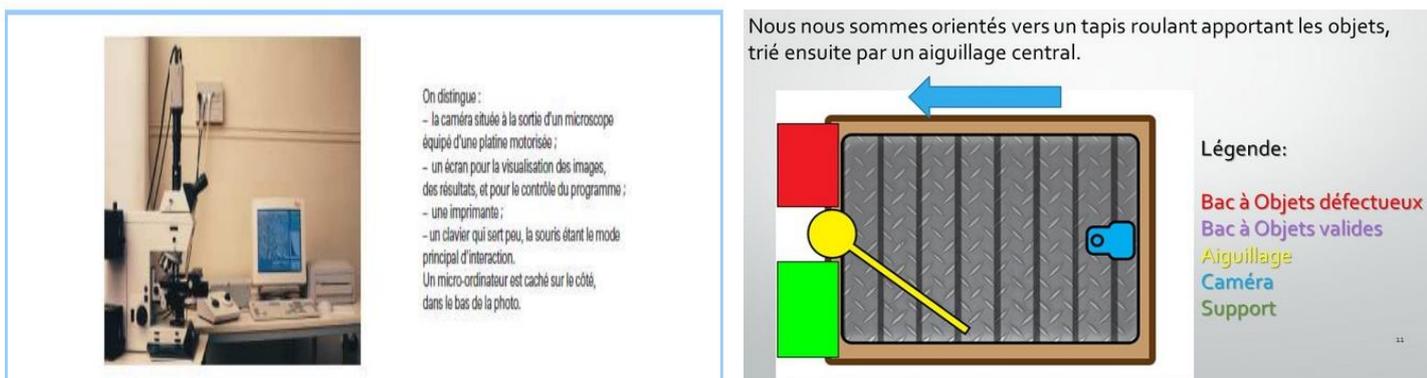


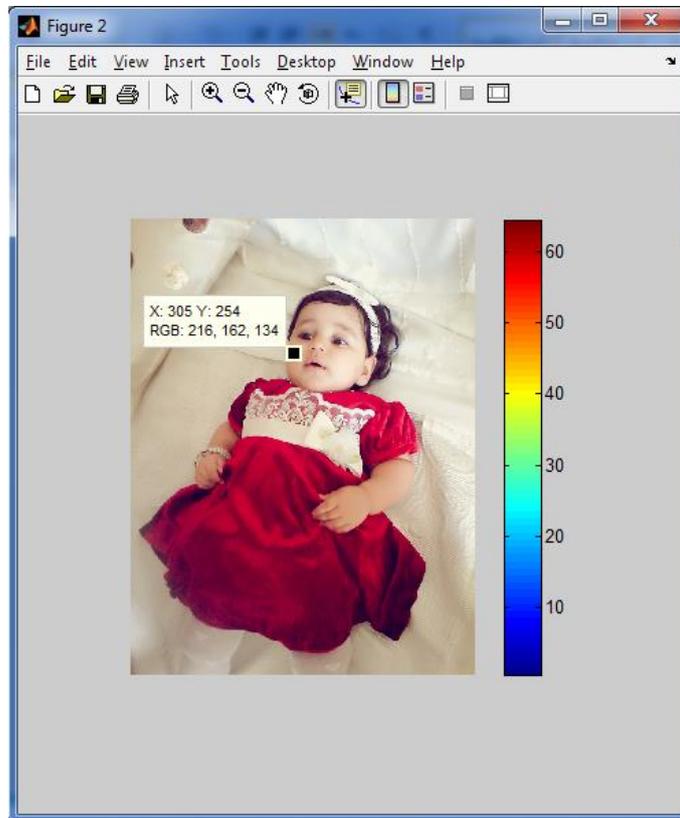
Figure 3.1 Système de Détection poursuite et comptage d'objets par un Caméra placé au dessus de convoyeur.

3.2 Acquisition des images par une caméra CMOS :

Les images sont acquises par une caméra matricielle de type CMOS qui délivre un signal vidéo standard à la cadence de 20 ms par trame, soit 40 ms par image, une image vidéo étant composée d'une trame paire et d'une trame impaire. Les images sont ensuite numérisées à cette même cadence vidéo sur 8 bits et selon une résolution de 640 x 480 pixels.

La caméra est placée sur un axe fixé au dessus de convoyeur bordant la scène observée. Cette installation ne permettait pas d'obtenir une prise de vue à la verticale de la scène, mais, comme nous le verrons l'inclinaison de la caméra, lorsqu'elle demeure faible, ne porte pas préjudice à la méthode.

Nous nous intéressons plus particulièrement dans cette partie à l'acquisition et l'analyse d'images. Une fois une telle image acquise, on peut la représenter comme une matrice de dimension où chaque composante correspond à un pixel de coordonnées (x, y) et où chaque valeur correspond à l'intensité d'un pixel associée. Figure 3.2 donne une illustration d'une telle



représentation.

Figure 3.2 : Illustration de la représentation des images numériques.

3.3 Traitement Expérimentale :

3.3.1 Lecture et visualisation de l'image et affichage de la matrice correspondante:

Lire et écrire des images sous MATLAB est capable de lire et de décoder les fichiers images sous MATLAB peut être représentée sous plusieurs formes, mais toujours sous forme d'une matrice. Avant de traiter une image dans MATLAB, il faut la lire et décoder son format afin de la transformer en une matrice de valeurs. Les exemples qui contiennent permettent de lire une image et de la décoder dans la variable image et de l'afficher à l'écran dans une figure. La commande `axis ('image')` rend l'image affichée carrée pour garder les proportions. L'appel à `axis` on permet l'affichage des graduations des axes. Enfin, on calcule les Objets qui passent par la ligne virtuelle spécifique.

PROCEDURE EN MATLAB

Pour lire et afficher l'image sur MATLAB on commence par la vérification que le fichier image 'PFE 2017.png' existe dans le répertoire C:\MATLAB\work.

En suite dans la fenêtre commande window :

On exécute les instructions suivantes :

MATLAB nous affiche l'image 'PFE 2017' et sa matrice correspondante.

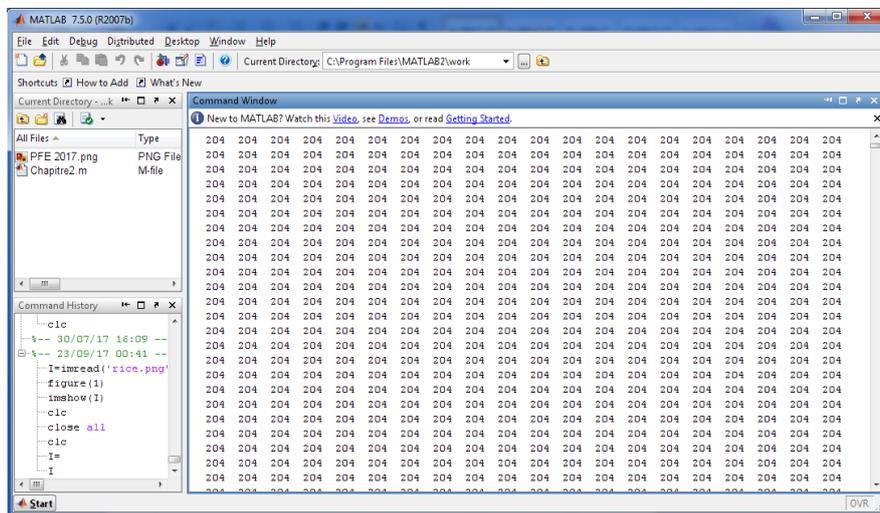


Figure 3.3 Taille d'image traitée.

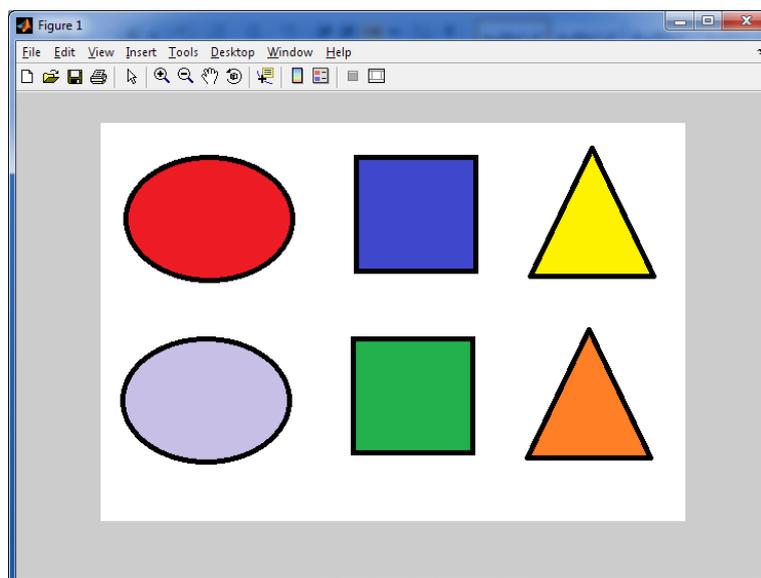


Figure 3.4 Image initiale

3.3.2 Convertir une image couleur en niveau de gris

La phase d'acquisition a permis d'associer à une image un ensemble de données. Des logiciels vont permettre d'améliorer l'image, puis de la décomposer, de la segmenter en plusieurs phases (composants de même nature), afin de sélectionner la ou les régions d'intérêt, en vue d'une quantification ou d'une reconnaissance de formes. L'image initiale est en général traitée, transformée en une autre image numérique, puis seuillée pour donner une image binaire (figure 3.5), souvent encore transformée en d'autres images binaires (figure 3.7). À chaque étape, des comparaisons et des opérations entre images numériques ou binaires peuvent se faire.

Pour convertir une image couleur en niveau de gris il faut remplacer, pour chaque pixel les trois valeurs représentant les niveaux de rouge, de vert et de bleu, par une seule valeur représentant la luminosité, à enregistrer dans le fichier ou à envoyer à un terminal en noir et blanc (écran ou imprimante). Si l'écran ou le fichier prend un format d'image en couleurs, les trois valeurs (rouge, vert, bleu) sont égales.

Une fonction utile à connaître est la fonction de passage d'une image couleur à une image en niveaux de gris. Cette fonction **rgb2gray** permet d'obtenir une image en niveaux de gris en partant d'une image couleur. Dans certains cas, cette transformation est très utile.

```
B= rgb2gray(A)
figure(2), imshow(B)
```

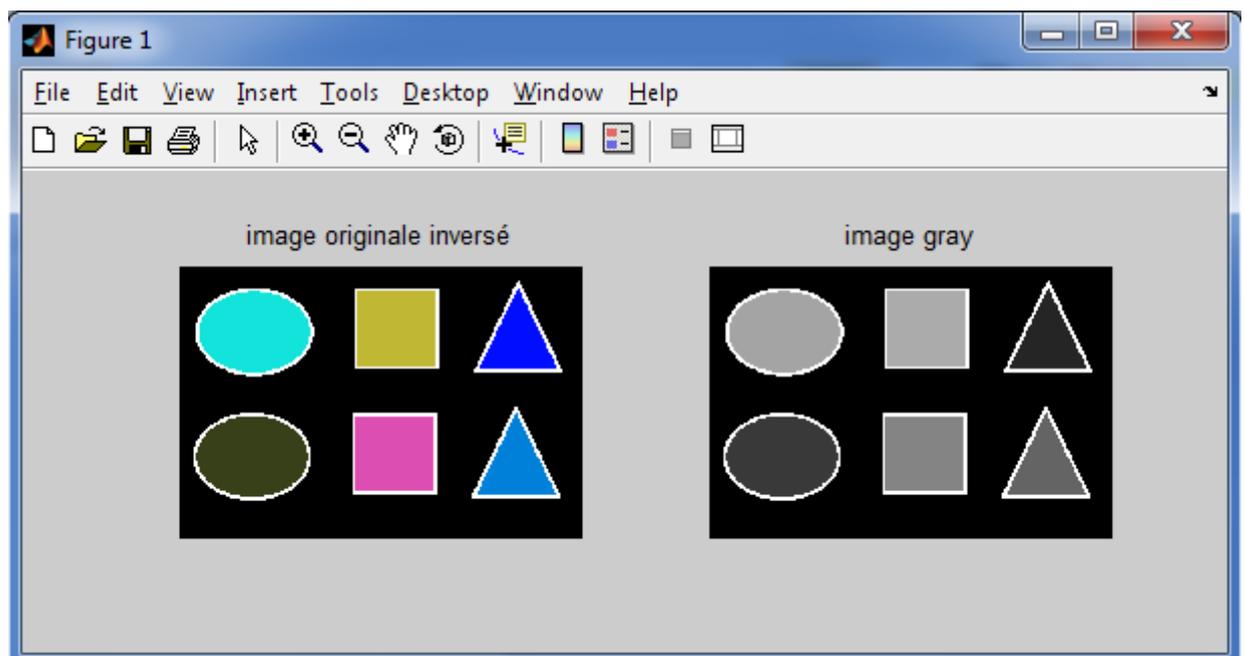


Figure 3.5 Afficher le niveau de gris d'image.

3.3.3. Seuillage

Le seuillage est aussi une transformation point par point. Il permet de passer d'une image numérique (niveaux de gris) à une image binaire (noir et blanc). Il permet de ne retenir dans l'image que les points, qui ont un niveau de gris supérieur ou inférieur à un seuil donné (points les plus sombres, ou points les plus clairs), ou ceux qui sont compris entre deux valeurs de gris $G_1 - G_2$ (phase grise). Les échantillons doivent toujours être préparés dans des conditions qui optimisent les contrastes. Une mise au point des techniques préparatives pour l'analyse d'images est souvent obligatoire.

Il consiste à déterminer la valeur du seuil à partir duquel tous les pixels ayant un niveau de gris inférieur à cette valeur sont représentés par un zéro « 0 » le noir, et tout les pixels de niveau de gris supérieur auront la valeur un « 1 » (le blanc). La valeur du seuil est déterminée à partir de l'histogramme du niveau de gris se trouvant dans la vallée entre les deux pics de l'histogramme figure.

```
level=graythresh(B);
```

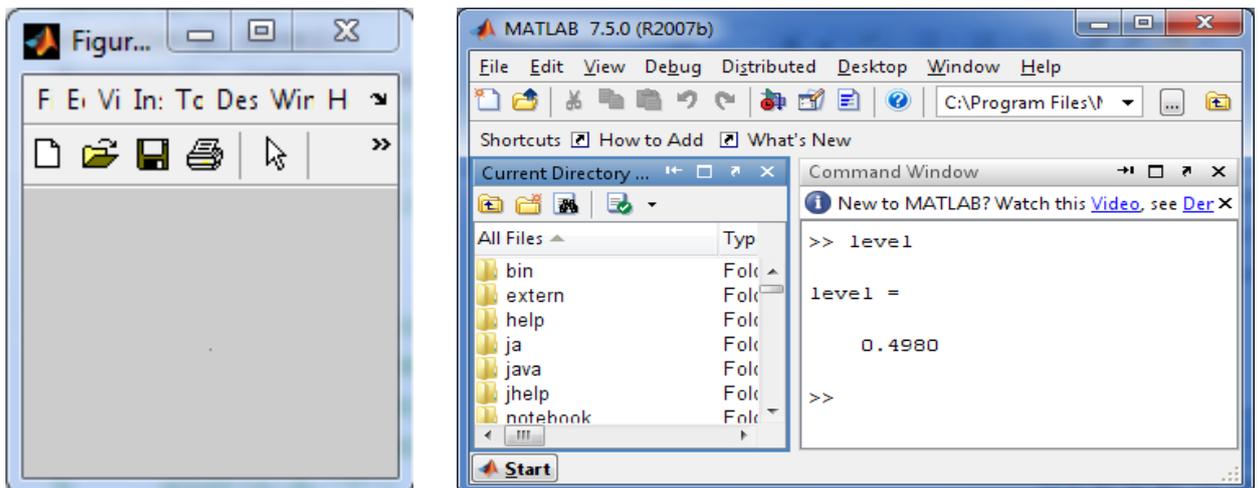


Figure 3.6 Afficher la valeur « level » correspond le programme.

3.3.4 Binarisation :

La binarisation est la technique de classification la plus simple où les pixels de l'image sont partagés par un seuil S en deux classes : ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et S est retenue et l'autre non, ou vice-versa. Le fond de l'image contient généralement du texte mais également du graphique qui sont quelques fois assez proches, le texte risque donc de fusionner des différentes parties du graphique à cause du changement faible d'intensité des pixels du fond et ceux de l'objet. Une binarisation efficace permet de séparer clairement le fond et l'objet. Cette étape prépare l'image entrée pour l'étape de reconnaissance.

Travailler avec des images binaires est plus simple, dans une certaine mesure, qu'avec des images en niveaux de gris, pour les raisons suivantes :

- On peut très facilement identifier des objets à condition qu'ils soient bien séparés (on appelle ça "l'analyse par composantes connexes") ;
- on a beaucoup moins d'informations à traiter, donc les calculs sont globalement plus rapides ;
- si notre image est bien binarisée, les algorithmes sont dans l'ensemble très robustes.

Évidemment, il y a un certain nombre d'inconvénients à binariser une image comme :

- il faut choisir un seuil correct, ce qui n'est pas toujours facile.
- on passe d'un espace de couleurs de dimension 255 à un nouvel espace de dimension 2: on perd énormément d'informations qui pourraient s'avérer utiles.

La plupart des algorithmes de binarisation sont très sensibles au bruit, on est souvent obligé de lisser l'image en prétraitement.

```
C=im2bw(B,level);  
figure(3),imshow(C),title('image binaire')
```

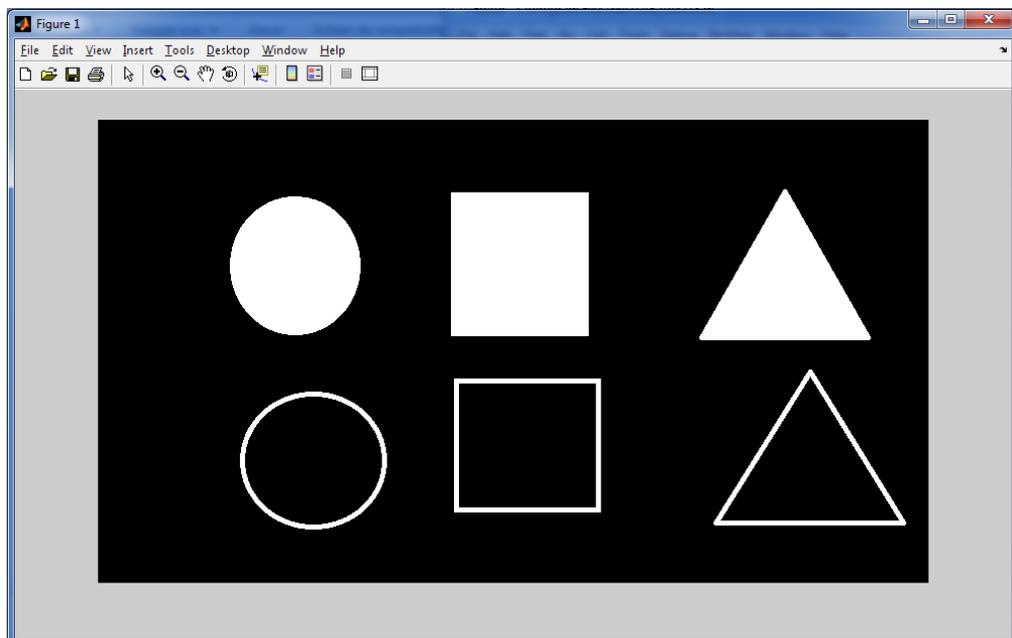


Figure 3.7 Transformation à l'image binaire.

3.3.5 Etiquetage:

L'étiquetage des pixels (L'extraction des composantes connexes, procédure également appelée capture des connexités), est largement utilisé en Reconnaissance des Formes (RdF) pour segmenter les images binaires. La technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé.

```
[label Ne]=bwlabel(c);  
f=label2rgb(label);  
figure(3),imshow(f),title('Etiquetage')
```

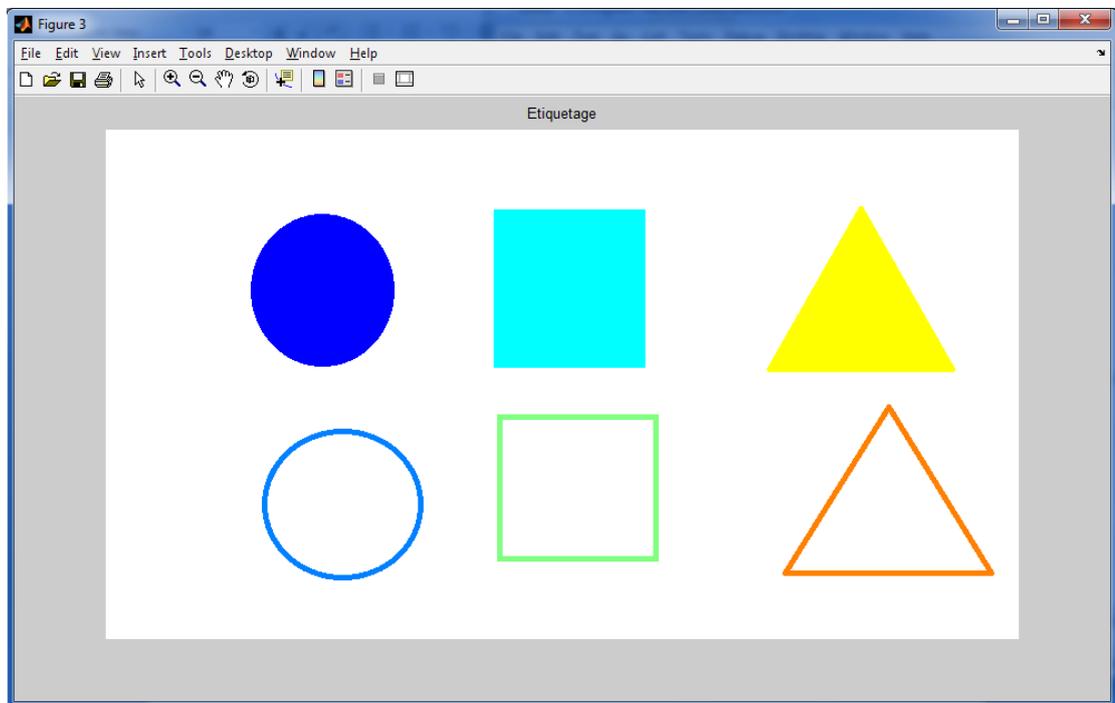


Figure3.8 Etiquetage des objets d'image par des couleurs différentes.

3.3.6 Détection des contours

La détection des contours constitue l'une des procédures de base de la vision artificielle. Une littérature aussi abondante que récente l'atteste. [8]

Un contour peut être défini comme une zone de transition entre deux régions relativement homogènes d'une image. Dans le cas où l'on cherche à différencier des objets du fond sur lequel ils apparaissent, un contour peut être considéré comme une zone de l'image à fortes variations locales de niveaux du gris.

La reconstitution des contours des objets stationnaires ou en mouvement dans leurs positions précédentes et suivantes ne présentant aucun intérêt dans le cadre de cette étude. Par contre, la nécessaire reconstitution des contours des objets en mouvements dans leurs positions courante sera examinée après application de l'opérateur de détection des contours des objets en mouvement.

La fonction de traitement de l'image « **regionprops** » donne des mesures basées sur la forme des régions d'image.

```
prop=regionprops (label)
imshow (f);
hold on
for n=1:length (prop)
    rectangle ('Position',prop (n) .BoundingBox, 'EdgeColor', 'g', 'LineWidth', 2)
    x=prop (n) .Centroid (1);
    y=prop (n) .Centroid (2);
    plot (x,y, '*')
end
hold off
```

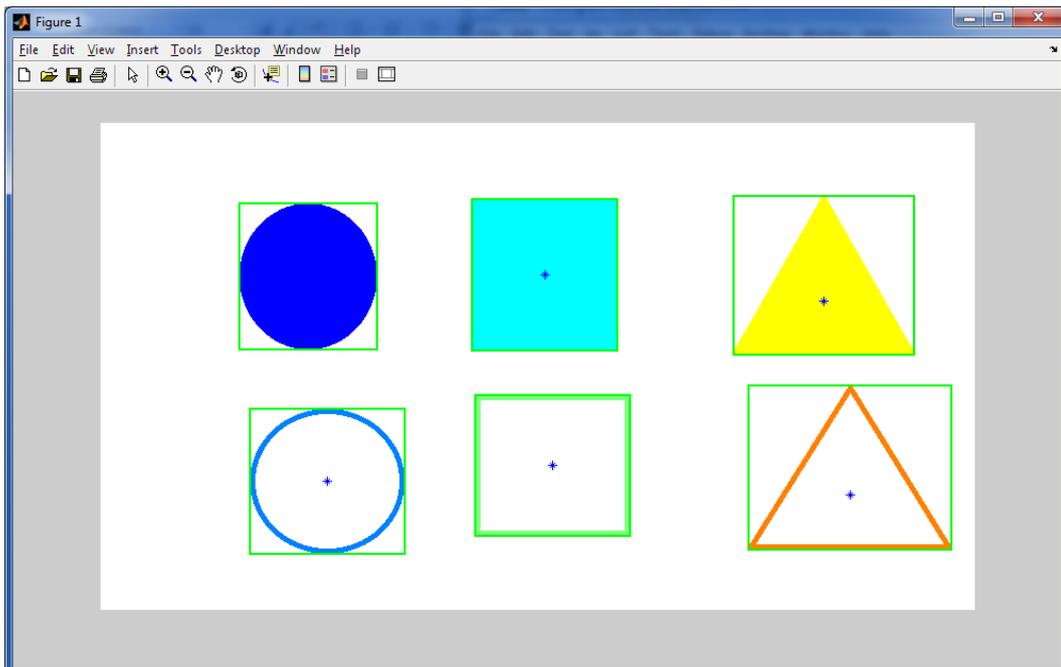


Figure 3.9 Détermination les contours en rectangle et les centres des objets

Le rectangle est, après le triangle, la forme géométrique la plus élémentaire permettant de modéliser d'un objet considéré comme bidimensionnel.

Dans une optique robotique, Stuck [8] a développé une procédure permettant d'encadrer tout amas de points connexes par un rectangle de cotés parallèles et perpendiculaires au balayage vidéo. Cette modélisation, très peu couteuse en calculs a été utilisée par Stelmaszyk [8]. Elle est malheureusement très sensible aux variations d'orientation de l'objet. En effet, les dimensions du modèle s'éloignent d'autant plus de celles de l'objet que celui-ci se déplace en diagonale par rapport au balayage vidéo.

Le rectangle exinscrit dont les dimensions correspondent le mieux à celles de l'objet à modéliser est le rectangle exinscrit de périmètre minimal qui tient compte de l'orientation de l'objet. Quoique idéal pour notre application, la détermination de ce rectangle nécessite de nombreux calculs: calcul du tenseur d'inertie, de ses axes principaux d'inertie, etc... . Il fut utilisé par Ferrie pour la modélisation de cellules en complément à d'autres paramètres. [8]

Les quatre cotés définis par les quatre valeurs suivantes:

- Xmin, correspondant au point de contour le plus à gauche, définit le coté Ouest;
- Xmax, correspondant au point de contour le plus à droite, définit le coté Est;
- Ymin, correspondant au point de contour le plus haut, définit le coté Nord ;
- Ymax, correspondant au point de contour le plus à bas, définit le coté Sud;

```
prop=regionprops(label)
imshow(f);
hold on
for n=1:length(prop)

rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2)
x=prop(n).Centroid(1);
y=prop(n).Centroid(2);
plot(x,y,'*')
end
hold off
```

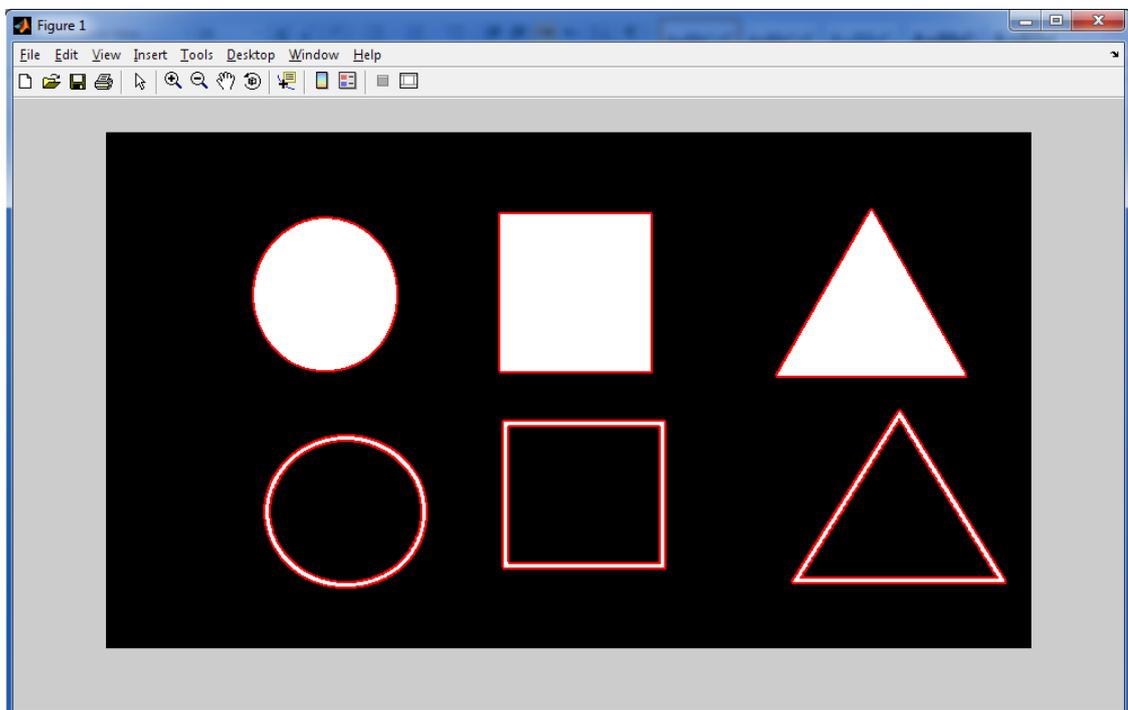


Figure 3.10 Détermination les contours des objets.

3.4 Conclusion

Le traitement d'images trouve son application dans des domaines extrêmement variés de l'industrie et de la recherche. Ces méthodes sont utilisées dans de nombreuses disciplines scientifiques, comme les sciences des matériaux, les sciences de la terre, la géographie (dont la cartographie et la géomorphologie), la robotique (pour le tri et la vérification de pièces électroniques) ou bien encore dans des domaines aussi variés que ceux qui ont trait à l'astronomie, l'identification, la pharmacologie. En biologie, ces techniques sont particulièrement bien adaptées à l'étude des images macroscopiques (radiologie, IRM, imagerie fonctionnelle...) mais aussi à celle des images microscopiques et constituent un outil performant pour visualiser et décrire des structures.

Citons en particulier le domaine de détection et comptage des objets par la vision artificiel comme le cas de notre travail la détection et comptage des objets est basé essentiellement sur des techniques et algorithmes de traitement d'image.

Conclusion générale

Le travail présenté dans ce mémoire a été effectué au sein du département d'électronique université de Blida.

Pour réaliser un tel travail nous avons été confrontés à des problèmes liés à l'aspect expérimental. L'environnement étant très complexe, les premiers problèmes concernaient le choix de la résolution, le changement de luminosité, la grande variation de vitesse du convoyeur et surtout la grande différence de niveau de gris entre le niveau des objets et celui du convoyeur (couleur, position par rapport à la caméra, etc.).

Nous avons présenté en premier lieu la vision assistée par ordinateur, sa définition, domaine d'utilisation puis on a détaillé quelques outils de traitement d'image qu'on a utilisé dans nos programmes.

En deuxième lieu, nous avons réalisé une étude sur les méthodes utilisées dans ce travail, on se focalisant surtout sur la détection d'objets.

En troisième étape, on a conçu un algorithme et l'implémenté, avec La conception d'un convoyeur qui était une tâche assez importante, ensuite, nous avons tenté d'améliorer notre application en testant ses performances pour différentes valeurs de paramètres

Une étape importante est l'évaluation des performances, ou on a testé cette application sur des vidéos réelles depuis notre caméra. Les résultats sont satisfaisants et le travail fonctionne en temps réel.

Au cours de ce projet et à travers le travail fait, nous avons pu acquérir beaucoup de connaissances. Premièrement, nous avons utilisé les outils appris au cours de notre formation, et enrichi nos connaissances dans le domaine du traitement d'images en apprenant de nouveaux outils. Nous avons aussi appris à utiliser le langage de programmation Matlab.

Nous estimons que notre module pourra être exploité dans l'industrie. Nous proposons comme perspective la réalisation d'un système embarqué basé sur la vision artificielle capable de compter le nombre de voiture circulant dans une autoroute.

Annexe 1 :

1.1 Choix des caméras :

La caméra dont les modes d'acquisition sont présentés figure n° 1 a les caractéristiques suivantes :

- fait l'acquisition d'une image en résolution VGA en codage RVB à la cadence maximale de 15 images par seconde,
- les autres modes sont codés en YUV, avec compression des composantes de chrominance :1/2: pour le format 4:2:2 et 1/4 pour le format 4:1:1.
- le format YUV non comprimé 4:4:4 est possible pour la résolution 160x120, avec une cadence de 30 images par seconde.

Il n'est pas nécessaire d'avoir une caméra permettant l'acquisition VGA à 30 Hz si l'ordinateur n'a pas la puissance de calcul suffisante pour traiter l'image à la cadence souhaitée !

Picture modes progressive VGA uncompressed, modes selection by FireWire link

Resolutions 640x480, 320x240, 160x120

Codings YUV, RGB, Monochrome

Video Modes / Frame rates	30 fps	15 fps	7.5 fps	3.75 fps
640x480 RGB-24 (24 bits)	-	X	X	X
640x480 YUV 4:2:2 (16 bits)	-	X	X	X
640x480 YUV 4:1:1 (12 bits)	X	X	X	X
640x480 Y-Mono8 (8 bits)	X	X	X	X
320x240 YUV 4:2:2 (16bits)	X	X	X	X
160x120 YUV 4:4:4 (24 bits)	X	X	X	-

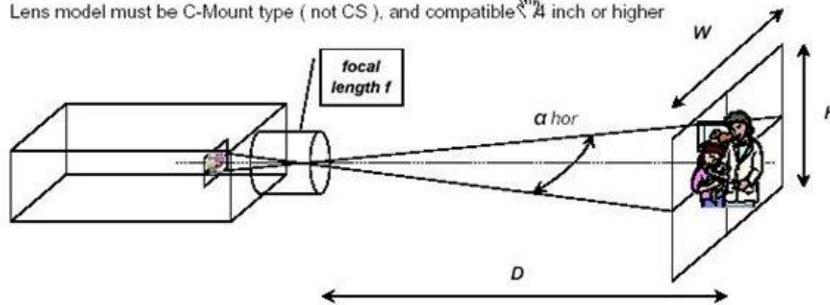
Figure n°1 : Différents Modes d'Acquisition

La figure n° 2 aide au choix de l'objectif en monture C en fonction de l'application. Pour éviter les distorsions, il faut prendre la focale la plus longue possible. Le tableau, obtenu sur le site web du fabricant Unibrain : www.unibrain.com est très pratique. Il indique en fonction de la distance focale choisie la dimension de la scène pour deux distances : 50 cm et 5 m qui correspondent par exemple à deux types d'applications courantes : respectivement :

- le contrôle qualité : caméra en « douche » au-dessus du tapis roulant,
- la robotique mobile et autonome : le robot mobile se déplaçant dans son environnement.

Selecting a lens for unibrain Fire-i400 camera

Lens model must be C-Mount type (not CS), and compatible $\frac{1}{4}$ inch or higher



Lens focal length (see note 1)	Horizontal viewing angle	Vertical viewing angle	Diagonal viewing angle	Size of the reality for D = 0,5 m (see note 2)		Size of the reality for D = 5 m	
f	α_{hor}	α_{vert}	α_{diag}	W	H	W	H
(mm)	(°)	(°)	(°)	(m)	(m)	(m)	(m)
2,00	83,72	67,80	96,48	0,896	0,672	8,96	6,72
3,00	61,70	48,26	73,49	0,597	0,448	5,97	4,48
4,00	48,26	37,14	58,50	0,448	0,336	4,48	3,36
6,00	33,26	25,25	40,94	0,299	0,224	2,99	2,24
8,00	25,25	19,07	31,28	0,224	0,168	2,24	1,68
12,00	16,99	12,78	21,15	0,149	0,112	1,49	1,12
16,00	12,78	9,60	15,94	0,112	0,084	1,12	0,84
25,00	8,20	6,15	10,24	0,072	0,054	0,72	0,54
36,00	5,86	4,40	7,32	0,051	0,038	0,51	0,38
50,00	4,11	3,08	5,13	0,036	0,027	0,36	0,27
75,00	2,74	2,05	3,42	0,024	0,018	0,24	0,18
100,00	2,05	1,54	2,57	0,018	0,013	0,18	0,13

Note 1 : exact focal length depends on available models from lens manufacturers
 Note 2 : each lens has a minimum object distance, that may be longer than 0,5 m

Figure n°2 : Choix de l'Objectif

1.2 Types d'images, de capteurs :

<i>Phénomène physique</i>	<i>Grandeur mesurée</i>	<i>Capteur</i>
Émission et réflexion de la lumière visible	Réflectance, luminance,...	CCD, CMOS, Barettes CCD,...
Rayonnement infra-rouge	Luminance IR (chaleur), ...	Bolomètres, ...
Écho ultrasonore	Distance, densité de tissus,...	Échographie, sonar,...
Résonance magnétique	Présence d'un corps chimique,...	IRM, RMN,...
Écho électromagnétique	Distance, spécularité de surfaces,...	Radar, SAR,...
Absorption des rayons X	Densité de tissus,...	Radiographie, tomographie,...

Figure 1.2 Différents Type de Capteur

Annexe 2 :

2.1 Applications des traitements d'images :

- Robotique - Industrie
 - Assemblage, reconnaissance de pièces`
 - Contrôle de qualité`
 - Véhicule autonome`
 - etc ...

- Télédétection
 - Météo`
 - Cartographie
 - Analyse des ressources terrestres
 - Astronomie
 - Restauration
 - etc...

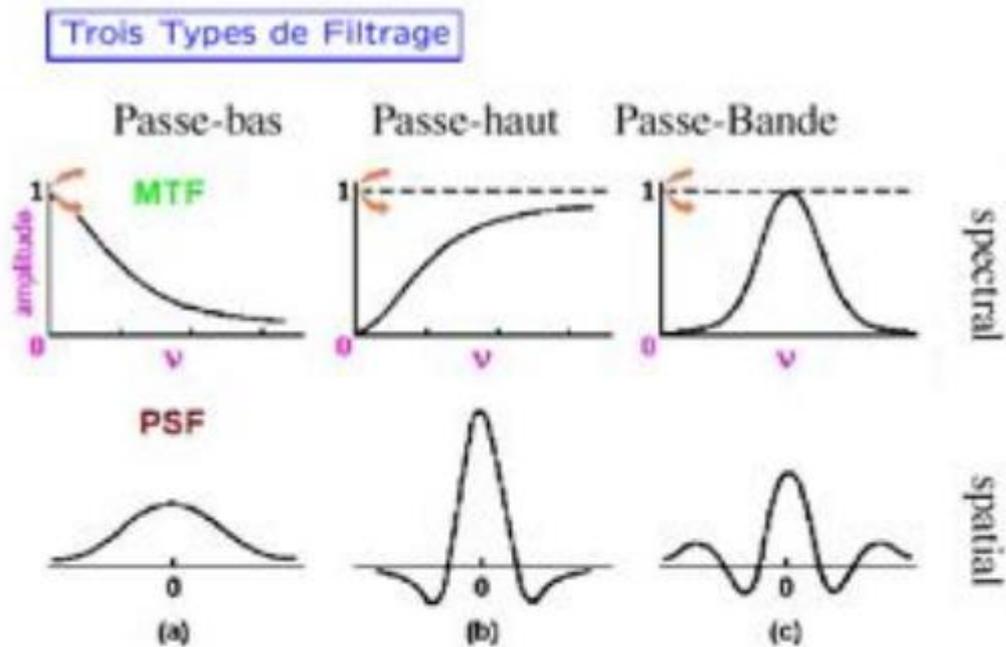
- Applications militaires
 - Guidage de missile
 - Reconnaissance (aérienne, sous-marine, etc ...)`
 - etc ...

- Imagerie médicale
 - Tomographie
 - Aide au diagnostic
 - Comptage (nombre de cellules)
 - Suivi de formes anatomiques
 - Restauration
 - etc...

- Sécurité
 - Reconnaissance (d'empreintes, visages, signatures)
 - Détection de mouvement`
 - etc...

Annexe 3 :

3.1 Types de filtrage :



- **PSF** : Point Spread Function
(ou Fonction d'Étalement Spectrale)
- **MTF** : Modulation Transfer Function
(ou Fonction de Transfert)
- ▶ **Filtre Passe-bas** : diminue le bruit mais atténue les détails de l'image
- ▶ **Filtre Passe-haut** : accentue les contours et les détails de l'image mais amplifie le bruit
- ▶ **Filtre Passe-bande** : élimine certaines fréquences indésirables présentes dans l'image

Référence Bibliographique :

- [1] LOURCI Mohamed Amine, "Segmentation de Blocs d'Images de documents Imprimés", Mémoire de Magister, Université de SAAD DAHLEB de BLIDA, Juin 2015.
- [2] Maitine Bergounioux, "Quelques méthodes mathématiques pour le traitement d'image", DEA. Cours M2 - Université d'Orléans , 2008-2009.
- [3] Mr Cyril Meurie, "Segmentation d'images couleur par classification pixellaire et hiérarchie de partitions", Thèse de Doctorat en Informatique, Université de CAEN/BASSE-NORMANDIE, P59, octobre 2005.
- [4] Dora Luz Almanza-Ojeda, Détection et suivi d'objets mobiles perçus depuis un capteur visuel embarqué, Thèse de Doctorat, Université Paul Sabatier-TOULOUSE III, Janvier 2011.
- [5] Perceval COUDRAIN, "Contribution au développement d'une technologie d'intégration tridimensionnelle pour les capteurs d'images CMOS à pixels actifs", Thèse de Doctorat, Université de TOULOUSE, octobre 2009.
- [6] **Patick.Bonnin** , "Les Bases du Traitement d'Image", <https://www.developpez.com/>, **Publié le 12 mai 2016.**
- [7] Antoine Manzanera , "Traitement d'images et vision artificielle", Cours ENSTA D9-2, - Unité d'Électronique et d'Informatique.
- [8] VIEREN Christphe, "Segmentation de Scènes Dynamiques en Temps REEL. Application au Traitement de Séquences d'images pour la surveillance de carrefours routiers", Thèse de Doctorat, Université des Science et Techniques de LILLE, FLANDRES ARTOIS, avril 1988.
- [9] <https://www.mathworks.com>
- [10] <http://www.techniques-ingenieur.fr>