

**République Algérienne Démocratique et populaire**



**Ministère de l'enseignement supérieur et de  
la recherche scientifique**

**Université 20 août 1955-Skikda**

**Faculté des sciences**

**Mémoire de Fin d'études pour L'obtention du Diplôme de Master en  
Informatique**

**Option : Systèmes Informatiques (SI)**

# **Traitement d'images numériques**

✓ **RÉALISÉ PAR :**

- **DJAGHROUD Assia**

✓ **ENCADRÉ PAR :**

- **M<sup>me</sup> MAGROUN Hanane**

**2024-2025**



# Remerciements

*Louanges tout d'abord à ALLAH qui est l'origine de toute réussite dans notre vie.*

*Je souhaite adresser mes sincères remerciements à ma dirigeante de mémoire, madame **MAGROUN Hanane** pour ses conseils et son encadrement. Sa présence, son soutien, sa disponibilité et ses conseils m'ont donné le grand courage afin d'accomplir ce projet.*

*Je voudrais aussi remercier les membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.*

*Enfin, mes remerciements s'adressent aussi à tous ceux qui ont participé, de près ou de loin, à l'élaboration de ce projet de fin d'études et en particulier ma famille.*





## *Dedicaces*

*Je tiens à remercier Dieu de m'avoir donné la force de réaliser ce  
modeste travail*

*Je dédie ce travail :*

- *A mes très chers parents pour leurs soutiens, leurs  
encouragements et leurs sacrifices.*
- *A mon mari*
  - *A mes chers frères.*
  - *A mes chères sœurs.*
  - *A tous mes amies.*
- *A toute ma famille.*
- *A tous ceux qui aime Assia*



## Abstract

Image processing is a rapidly growing field focused on the automated analysis and interpretation of visual data. It combines low-level techniques (filtering, restoration, segmentation) and high-level approaches (pattern recognition). This study presents an integrated application featuring preprocessing (contrast enhancement, noise reduction), geometric transformations, and compression methods. The implemented algorithms include Gaussian filtering, edge detectors (Sobel, Canny), and the K-means algorithm for image segmentation and classification. Structured in three parts, this work first covers the fundamentals of digital images, then explores processing techniques, and finally details the design and implementation of the application.

**Keywords:** Image processing, Segmentation, Compression, Filtering, K-means

## Résumé

Le traitement d'images est un domaine en pleine expansion qui vise à analyser et interpréter automatiquement les données visuelles. Il combine des techniques de bas niveau (filtrage, restauration, segmentation) et de haut niveau (reconnaissance de formes). Ce mémoire propose une application intégrée offrant des fonctionnalités de prétraitement (amélioration du contraste, réduction du bruit), des transformations géométriques et des méthodes de compression. Les algorithmes implémentés incluent le filtrage Gaussien, les détecteurs de contours (Sobel, Canny) ainsi que l'algorithme K-means pour la segmentation et la classification d'images. Structuré en trois parties, ce travail présente d'abord les fondamentaux des images numériques, puis les méthodes de traitement, et enfin la conception et la réalisation de l'application.

**Mots clés :** *Traitement* d'images, Segmentation, Compression, Filtrage, K-means.

## ملخص

معالجة الصور هي مجال يتوسع بسرعة يهدف إلى تحليل وتفسير البيانات المرئية تلقائيًا. فهو يجمع بين تقنيات المستوى المنخفض (التصفية، الاستعادة، التقسيم) والمستوى العالي (التعرف على الأنماط). تقدم هذه الدراسة تطبيقًا متكاملًا يوفر وظائف ما قبل المعالجة (تحسين التباين، تقليل الضوضاء)، والتحويلات الهندسية، وطرق الضغط. تشمل الخوارزميات المطبقة التصفية الغوسية، وكاشفات الحواف (سوبيل، كاني)، بالإضافة إلى خوارزمية **K-means** للتقسيم والتصنيف الصوري. ينقسم هذا العمل إلى ثلاثة أجزاء: يقدم أولاً الأساسيات حول الصور الرقمية، ثم يتناول تقنيات المعالجة، وأخيرًا يتطرق إلى تصميم وتنفيذ التطبيق.

الكلمات المفتاحية: معالجة الصور، التقسيم، الضغط، التصفية، خوارزمية K-means.

# Liste des Figures

## Chapitre I

<b>Figure I.1.</b> Image de projecteur de film 17.5mm sonore (AP-98-2060).....	.2
<b>Figure I.2.</b> Le processus général de l'acquisition d'une image numérique.....	.3
<b>Figure I.3.</b> Echantillonnage d'une image.....	3
<b>Figure I.4.</b> Quantification d'une image.....	4
<b>Figure I.5.</b> Déroulement de l'échantillonnage et de la quantification.....	4
<b>Figure I.6.</b> Exemple d'image en 255 niveaux de gris - Bureau. Bmp.....	5
<b>Figure I.7.</b> La représentation formelle d'une image numérique.....	6
<b>Figure I.8.</b> Positions des pixels dans le repère image. Chaque pixel est repéré par l'indice de sa ligne (entre 0 et $N - 1$ ) et l'indice de sa colonne (entre 0 et $M - 1$ ).....	7
<b>Figure I.9 :</b> Les pixels d'une image numérique .....	8
<b>Figure I.10.</b> Décomposition d'une image couleur en ses trois composantes Rouge, verte et bleue. ....	.10
<b>Figure I.11.</b> Image de couleur CMJN .....	11
<b>Figure I.12.</b> Image numérique en niveau de gris .....	11
<b>Figure I.13.</b> Image en noir et blanc.....	12
<b>Figure I.14.</b> Image en niveaux de gris (à gauche) et sa en représentation en pseudo couleur (à droite) .....	13
<b>Figure I.15.</b> Une image matricielle.....	14
<b>Figure I.16.</b> Image vectorielle (à gauche) et image matricielle (à droit).....	15

## Chapitre II

<b>Figure II.1.</b> Schéma d'un système de traitement d'images.....	20
<b>Figure II.2.</b> Exemple sur la restauration d'une image.....	22
<b>Figure II.3.</b> Histogramme d'une image numérique.....	23
<b>Figure II.4.</b> Courbe tonale de la normalisation (1).....	24
<b>Figure II.5.</b> Courbe tonale de la normalisation (2).....	25
<b>Figure II.6 :</b> Egalisation d'histogramme.....	25
<b>Figure II.7.</b> Filtre de Gauss.....	27
<b>Figure II.8.</b> Les étapes de filtrage par filtre filtre Median ;.....	28
<b>Figure II.9.</b> Image originale.....	28
<b>Figure II.10.</b> image filtrée par un filtre moyenneur de taille $3 \times 3$ .....	28

<b>Figure II.11.</b> image filtrée par un Filtre moyenneur de taille 7*7.....	28
<b>Figure II.12.</b> La segmentation de l'image en morceaux.....	30
<b>Figure II.13.</b> Image en nuances de gris d'un mur de briques et d'un <u>râtelier à vélo</u> (à gauche), Gradient avec le filtre de Prewitt.....	34
<b>Figure II.14.</b> Application du filtre de Sobbel.....	35
<b>Figure II.15.</b> Application du filtre de Roberts.....	36
<b>Figure II.16.</b> Application du filtre Sepia.....	36

### **Chapitre III**

<b>Figure III.1.</b> Diagramme de cas d'utilisation.....	42
<b>Figure III.2.</b> Diagramme d'activité.....	43
<b>Figure III.3.</b> Une fenêtre en java.....	46
<b>Figure III.4.</b> L'interface de l'Eclipse.....	47
<b>Figure III.5.</b> Schéma général de l'application.....	48
<b>Figure III.6.</b> La page d'accueil de l'application.....	49
<b>Figure III.7.</b> Fenêtre Saisie le mot de passe.....	49
<b>Figure III.8.</b> Ouvrir une image numérique.....	50
<b>Figure III.9.</b> Enregistrer une image numérique.....	50
<b>Figure III.10.</b> Exemple de la transformation en niveau de gris.....	51
<b>Figure III.11.</b> Exemple de la Conversion Image en Noir & Blanc.....	51
<b>Figure III.12.</b> Exemple d'un Négatif d'image.....	52
<b>Figure III.13.</b> Exemple de l'application d'un filtre Sepia.....	52
<b>Figure III.14.</b> Exemple de l'application d'un filtre Sobel.....	53
<b>Figure III.15.</b> L'affichage de l'histogramme d'une image.....	53
<b>Figure III.16.</b> Rotation à gauche et à droite.....	54
<b>Figure III.17.</b> Compression .....	54
<b>Figure III.18.</b> Barre d'accès rapide .....	55

# SOMMAIRE

<b>I. Remerciements</b>	
<b>II. Dédicaces</b>	
<b>III. Résumé</b>	
<b>IV. Liste des figures</b>	
<b>V. Introduction générale</b> .....	1

## Chapitre I: Généralités sur l'image numérique

1. Introduction .....	2
2. L'image ...../.....	2
3. Obtention des données images .....	3
4. L'image numérique .....	5
4.1. Formalisation .....	6
4.2. Discrétisation de l'espace image .....	6
5. Caractéristique d'une image numérique.....	7
5.1. Le pixel .....	7
5.2. La dimension (La définition).....	8
5.3. La résolution .....	9
5.4. Luminance .....	9
5.5. Contraste .....	9
6. L'espace couleur .....	9
6.1. Le modèle RVB / RGB .....	9
6.2. Le modèle CMJN .....	10
6.3. Le modèle niveau de gris .....	11
6.4. Le modèle noir et blanc.....	12
6.5. Les pseudos couleurs .....	12
7. Types d'image numérique .....	13

7.1. L'image matricielle (bitmap).....	13
7.2. L'image vectorielle.....	14
8. Formats de fichiers images .....	15
8.1. Formats de fichiers non compressés.....	15
8.1.1. Format PSD (Document Photoshop).....	16
8.1.2. Format bmp (bitmap ).....	16
8.1.3. Format TIFF (Tagged Image File Format).....	16
8.1.4. Format RAW .....	16
8.2. Format de fichiers compressés .....	17
8.2.1. Format JPG .....	17
8.2.2. Format GIF ( Graphics Interchange Format ).....	17
8.2.3. Format PNG (Portable Network Graphics) .....	17
9. Conclusion .....	18

## **Chapitre II: Traitement d'image numérique**

1. Introduction.....	19
2. Définition du traitement .....	19
2.1. Historique.....	19
2.2. Système de traitement d'images.....	20
2.2.1. Prétraitement d'images numériques .....	20
2.2.2. Analyse d'images numériques.....	21
2.2.3. Interprétation .....	21
3. Opérations de prétraitement.....	21
3.1. Restauration des images .....	21
3.2. Amélioration des images .....	22
3.2.1. Amélioration du rapport image/bruit .....	23
3.2.2. Amélioration par histogramme .....	23
3.3. Compression des images numériques .....	29

3.3.1. Les approches de compression .....	29
4. La segmentation d'images.....	30
4.1. Définition.....	30
4.2. Objectifs de la segmentation .....	31
4.3. Les différentes techniques de segmentation d'image.....	31
4.3.1. Technique de segmentation par seuillage de niveau de gris.....	31
4.3.2. Technique de segmentation par région .....	32
4.3.3. Technique de segmentation par contour.....	33
4.3.4. Technique de segmentation par classification .....	37
4.3.5. Technique de segmentation par modelé déformable .....	39
5. Conclusion .....	40

### **Chapitre III : Conception et réalisation de l'application**

1. Introduction .....	41
2. Conception .....	41
2.1. Diagramme de cas d'utilisation .....	41
2.2. Diagramme d'activité .....	43
3. Présentation des outils de développement.....	44
3.1. Le langage de programmation utilisé : Java .....	44
3.2. L'environnement de programmation : Eclipse .....	46
3.3. Schéma de l'application .....	48
4. Présentation de l'application .....	49
4.1. Interfaces et fonctionnalités .....	49
4.2. Quelques Scripts des commandes .....	55
5- Conclusion .....	58
VI.	
<b>Conclusion générale</b> .....	59

### **VII.Références bibliographiques**

# Introduction générale

# Introduction générale

Le traitement d'images est un domaine en constante évolution qui a connu un essor considérable au cours des dernières décennies. Son objectif principal est l'étude, la conception et la mise en œuvre de systèmes capables d'extraire l'information pertinente contenue dans les images, véritables vecteurs de données dans de nombreux domaines scientifiques et industriels.

Ce champ disciplinaire se structure généralement en deux niveaux complémentaires :

1. **Traitement de bas niveau** : incluant des opérations fondamentales telles que la restauration, l'amélioration et la segmentation d'images.
2. **Traitement de haut niveau** : concernant l'analyse sémantique via la reconnaissance de formes et l'interprétation automatique.

Un aspect particulièrement crucial concerne la compression d'images, technologie indispensable pour optimiser le stockage et la transmission des données visuelles.

## Cadre du Mémoire

Ce travail s'inscrit dans cette dynamique en proposant le développement d'une application intégrée de traitement d'images numériques. Notre solution regroupe les fonctionnalités essentielles suivantes :

- **Prétraitements de base** : amélioration (correction de contraste, réduction du bruit), segmentation (détection de régions d'intérêt)
- **Transformations géométriques** : rotation, redimensionnement (agrandissement/réduction)
- **Compression** : optimisation du poids des fichiers

L'implémentation a privilégié des algorithmes éprouvés :

- **Filtrage** : Gaussien (lissage), Robert et Sépia (effets visuels)
- **Détection de contours** : opérateurs de Sobel et Canny (précision des contours)

## **Structure du Document**

L'organisation de ce mémoire repose sur trois piliers fondamentaux :

### **Chapitre 1 : Généralités sur l'image numérique**

- Définition et modélisation mathématique
- Processus d'acquisition et caractéristiques physiques
- Typologie (images binaires, niveaux de gris, couleurs)
- Formats standards (JPEG, PNG, TIFF) et critères de choix

### **Chapitre 2 : Traitement d'image numérique**

- Techniques de restauration (filtrage adaptatif)
- Amélioration par manipulation d'histogramme
- Algorithmes de compression (avec perte/sans perte)
- Méthodes de segmentation (par région, par contour)

### **Chapitre 3 : Conception et réalisation de l'application**

- Choix technologiques (langage de programmation, IDE)
- Architecture logicielle (diagramme UML)
- Interface utilisateur (captures d'écran commentées)

# **Chapitre I :** **Généralités sur l'image** **numérique**

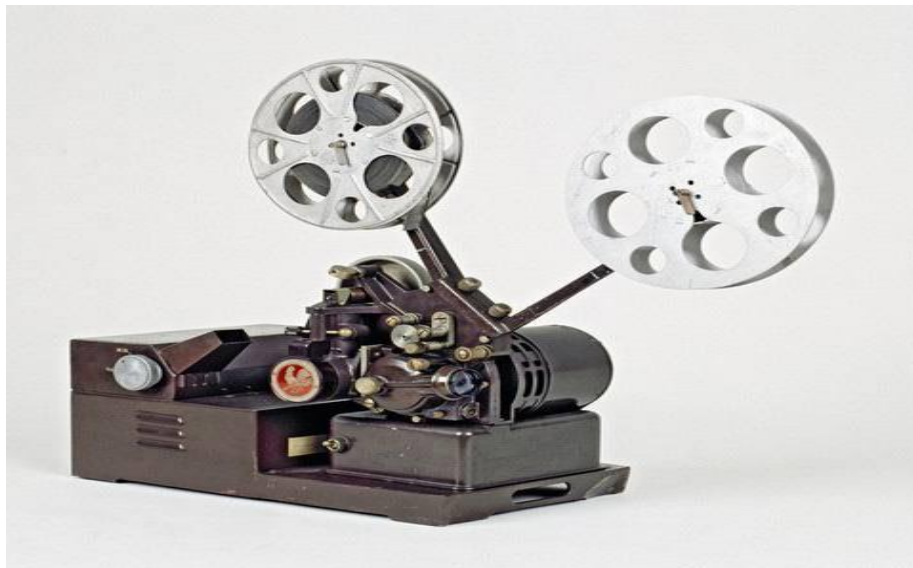
## 1. Introduction

L'image est partout, Elle est maintenant un des outils d'investigation les plus évalués de la recherche scientifique et technique. Le traitement et l'analyse d'images trouvent leurs applications dans des domaines extrêmement variés de l'industrie et de la recherche.

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information.

## 2. L'image

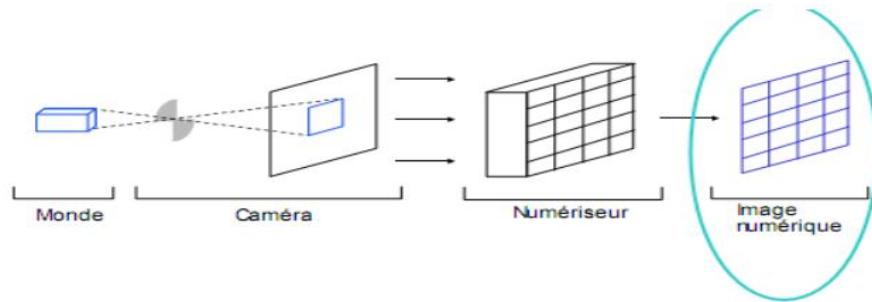
Une image est une reproduction exacte ou représentation analogique d'un être, ou d'une chose, elle est liée à un support matériel par exemple : pigments de peinture et toile, photo argentique, radiographie, bobines de film.



**Figure I.1.** Image de projecteur de film 17.5mm sonore (AP-98-2060)

### 3. Obtention des données images

Afin de rendre une image utilisable par l'ordinateur, il est nécessaire d'effectuer une transformation, grâce à une procédure de numérisation qui consiste à transformer l'image de base en une mosaïque de points et les convertir en nombre binaire traitable par le système informatique « Ordinateur » (figure I.2)



**Figure I.2.** Le processus général de l'acquisition d'une image numérique

Numériser une image consiste à transformer celle-ci de son état analogique (sous forme d'un signal continu) vers l'état numérique, cette opération se déroule en trois étapes :

#### ✚ Echantillonnage

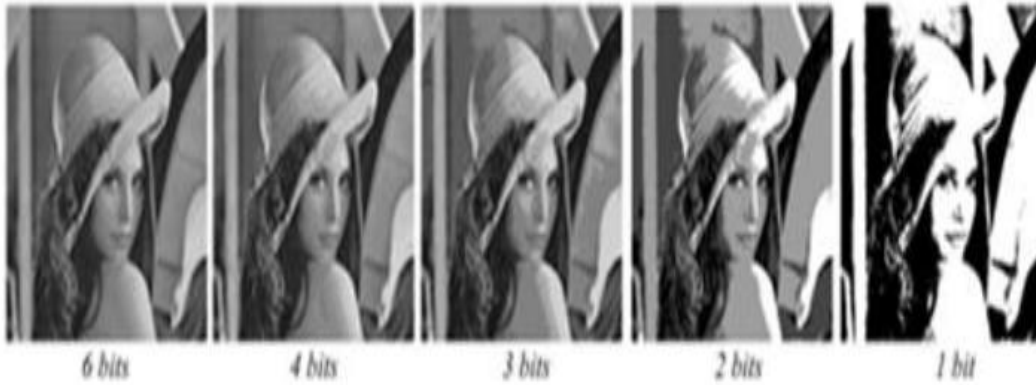
L'échantillonnage d'un signal (une dimension ou deux dimensions (cas d'image)) c'est le prélèvement des échantillons de ce même signal à des instants multiples de la période d'échantillonnage (Figure I.3).



**Figure I.3.** Echantillonnage d'une image.

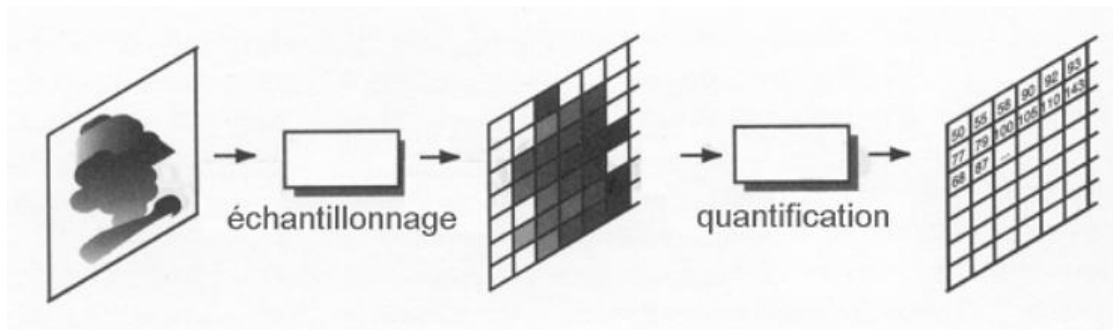
### Quantification

Elle consiste à affecter une valeur numérique à chaque échantillon. la quantification peut être monochrome ou couleur (figure I.4).



**Figure I.4.** Quantification d'une image.

L'échantillonnage est limité par la capacité du capteur, donc le nombre de pixels disponible (ou autre limite imposée), tandis que la quantification est limitée par la quantité de tons (de gris) définie dans l'intervalle.



**Figure I.5.** Déroulement de l'échantillonnage et de la quantification

### Codage

Le codage est une opération qui consiste à donner à chaque pixel de l'image un code qui représente une information complémentaire : sa couleur et sa précision. Ceci afin d'optimiser son stockage ou sa transmission.

## 4. L'image numérique

Une image numérique est un ensemble discret de points appelés PIXELS (Contraction de PICTURE ELEMENTS). Elle a pour vocation d'être affichée sur un écran. Chaque pixel possède une couleur.



**Figure I.6.** Exemple d'image en 255 niveaux de gris - Bureau. Bmp

L'appellation d' image numérique désigne :

toute image (dessin, icône, photographie...) acquise, créée, traitée et stockée sous forme binaire:

- ✚ Acquis par des convertisseurs analogique-numérique situés dans des dispositifs comme les scanners, les appareils photo ou les caméscopes numériques, les cartes d'acquisition vidéo (qui numérisent directement une source comme la télévision).
- ✚ Créée directement par des programmes informatiques, grâce à une souris, des tablettes graphiques ou par de la modélisation 3D (ce que l'on appelle, par abus de langage, les « images de synthèse »).
- ✚ Traitée grâce à des outils graphiques, de façon à la transformer, à en modifier la taille, les couleurs, d'y ajouter ou d'en supprimer des éléments, d'y appliquer des filtres variés, ...etc.
- ✚ Stockée sur un support informatique (clé USB, SSD, disque dur, CD-ROM...)

## 4.1. Formalisation

Une image est avant tout un signal 2D (x,y), souvent, cette image représente une réalité 3D (x,y,z). Alors du point de vue mathématique une image optique est généralement représentée par une fonction bidimensionnelle représentant des caractéristiques particulières du signal lumineux de l'image en chaque point de son espace (intensité, couleur,...).

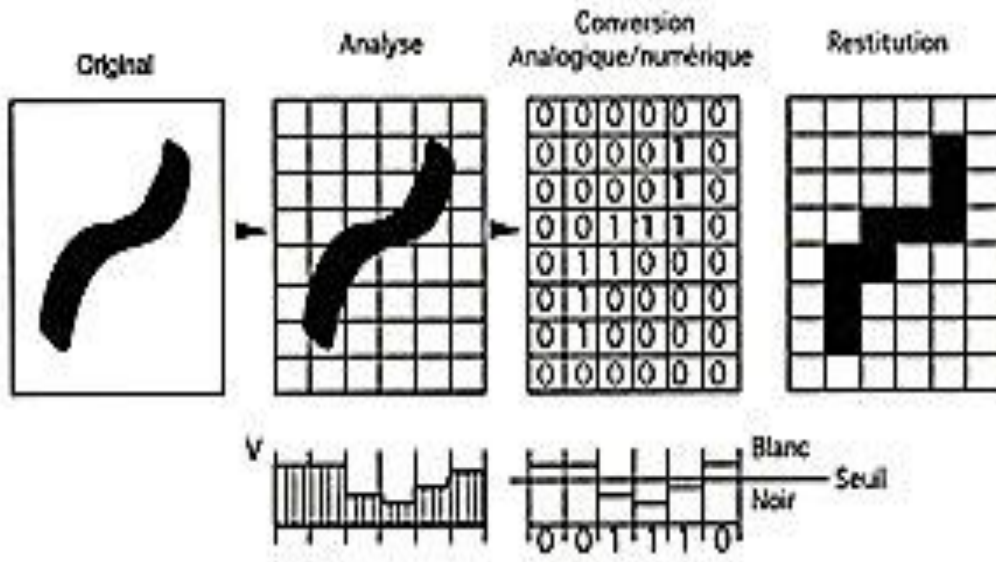


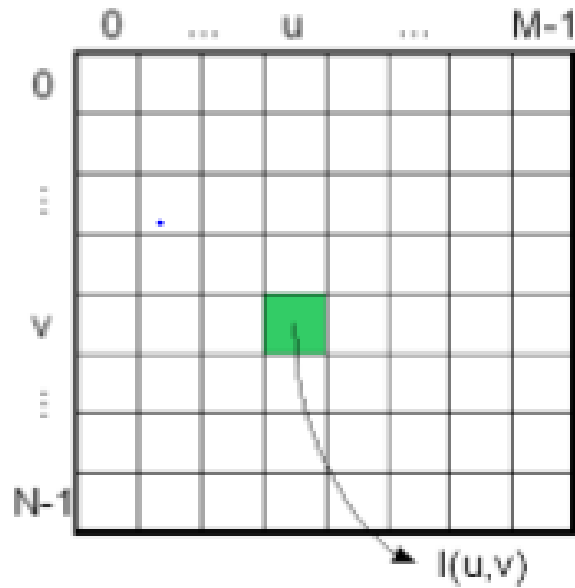
Figure I.7. La représentation formelle d'une image numérique

## 4.2. Discrétisation de l'espace image

Dans les images matricielles 2D, les valeurs sont définies uniquement pour chaque élément de la matrice image. De manière traditionnelle, l'origine de l'image se situe en haut à gauche, en comptant les colonnes vers la droite et les lignes vers le bas, tous les deux en commençant à 0. Les valeurs disponibles sont donc de la forme :

$$\mathbf{F}(\mathbf{x},\mathbf{y}) = \mathbf{I}(\mathbf{u},\mathbf{v}), \mathbf{u} \in [0;M - 1], \mathbf{v} \in [0;N - 1] \quad (\mathbf{I.1})$$

Où  $u$  et  $v$  sont des valeurs entières représentant les indices des colonnes et des lignes, et  $M$  et  $N$  représentent la largeur et la hauteur de l'image.



**Figure I.8.** Positions des pixels dans le repère image. Chaque pixel est repéré par l'indice de sa ligne (entre 0 et  $N - 1$ ) et l'indice de sa colonne (entre 0 et  $M - 1$ )

Où  $u$  et  $v$  sont des valeurs entières représentant les indices des colonnes et des lignes, et  $M$  et  $N$  représentent la largeur et la hauteur de l'image

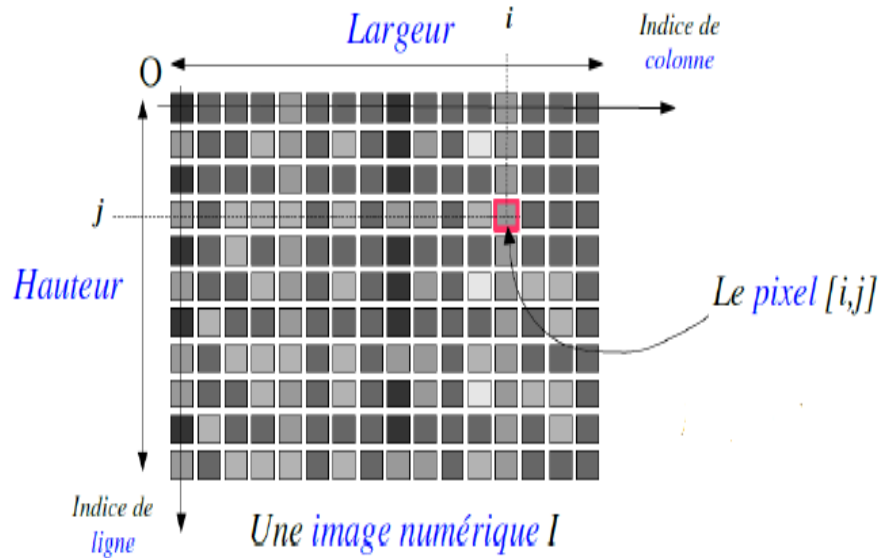
## 5. Caractéristique d'une image numérique

L'image est un ensemble structurée d'informations caractérisée par les parametres suivants :

### 5.1. Le pixel

Une image numérique contient un nombre fini de points. Ces points sont appelés Pixels (contraction des mots anglais "PICTure ELementS", c'est à dire élément d'image). Les pixels sont situés sur une grille régulière. A chaque pixel de la grille est associée une couleur ou une nuance de gris.

✚ Le passage d'une image continue à une grille de pixels s'appelle l'échantillonnage : on ne conserve que quelques points d'une image continue.



**Figure I.9 :** Les pixels d'une image numérique

- ✚ La taille du pixel définit la résolution par rapport à l'image analogique originale, c'est-à-dire la finesse de la grille. Plus la résolution baisse, plus le nombre de pixels dans l'image diminue, et plus la qualité de l'image numérique se dégrade.
- ✚ Les pixels sont codés sur un ou plusieurs bits selon le nombre de couleurs affichables :
  - ✓ En noir et blanc : un pixel correspond à un bit
  - ✓ En 16 couleurs (standard VGA) : un pixel correspond à 4 bits
  - ✓ En 256 couleurs : un pixel correspond à 8 bits (soit 1 octet)
  - ✓ En 65.536 couleurs : un pixel correspond à 16 bits
  - ✓ En mode 16 millions de couleurs : un pixel correspond à 24 bits

## 5.2. La dimension (La définition)

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.

$$\text{Dimension (nombre total des pixels)} = \text{nombre colonnes} \times \text{nombre lignes} \quad (\text{I.2})$$

### 5.3. La résolution

La résolution d'une image est définie par un nombre de pixels par unité de longueur de la structure à numériser. Ce paramètre est défini lors de la numérisation (passage de l'image sous forme binaire), et dépend principalement des caractéristiques du matériel utilisé lors de la numérisation. Plus le nombre de pixels par unité de longueur de la structure à numériser est élevé, plus la quantité d'information qui décrit cette structure est importante et plus la résolution est élevée.

La résolution est exprimée en nombre de pixels par unité de mesure ( pouce ou centimetre).

### 5.4. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie ainsi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface.

### 5.5. Contraste

Est une propriété intrinsèque d'une image qui désigne et quantifie la différence entre les parties claires et foncées d'une image (elle différencie les couleurs claires des couleurs foncées).

En photographie on le définit le contraste comme la différence entre la densité la plus forte et la plus faible d'une image. Le contrôle du contraste est un élément important de la pratique photographique. Le contraste final de l'image dépend à la fois du sujet, de la nature et du traitement du négatif et du positif.

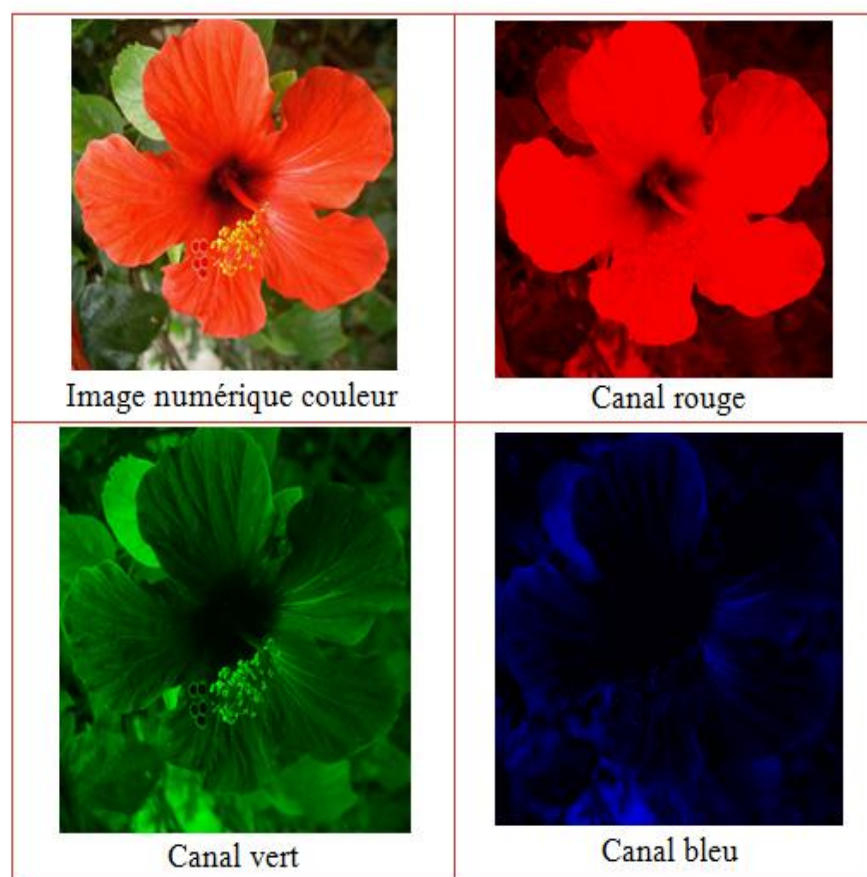
## 6. L'espace couleur

La notion de couleur est subjective: elle résulte de la perception de la lumière par le système visuel humain. Il existe de très nombreux systèmes de représentation des couleurs ou espaces de couleur, qui ont des applications et des propriétés différentes, les plus couramment utilisés sont :

### 6.1. Le modèle RVB / RGB

Une image couleur est en réalité composée de trois images, afin de représenter le rouge (R), le vert (V/G), et le bleu (B), chacune de ces trois images s'appelle

un canal. La figure suivante (Figure I.10) montre la décomposition d'une image couleur en ses trois canaux constitutifs.



**Figure I.10.** Décomposition d'une image couleur en ses trois composantes rouge, verte et bleue

Chaque pixel de l'image couleur contient ainsi trois nombres (R,V,B), chacun étant un nombre entier entre 0 et 255. Si le pixel est égal à  $(R,V,B)=(255,0,0)$ , il ne contient que de l'information rouge, et est affiché comme du rouge. De façon similaire, les pixels valant  $(0,255,0)$  et  $(0,0,255)$  sont respectivement affichés vert et bleu.

## 6.2. Le modèle CMJN

Le modèle de couleur CMJN (couleur de quadrichromie) est un modèle de couleur soustractif, utilisé pour l'impression couleur, et est également utilisé pour décrire le processus d'impression lui-même. CMJN fait référence aux quatre encres utilisées dans certaines impressions couleur: cyan, magenta, jaune et clé (noir).

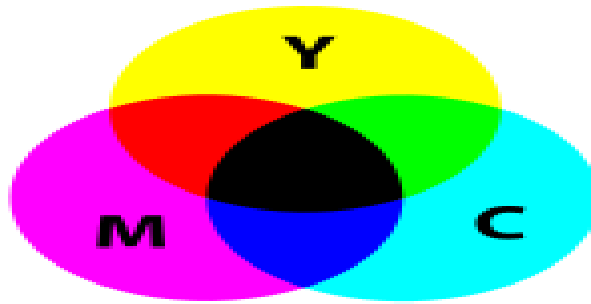


Figure I.11. Image de couleur CMJN

### 6.3. Le modèle niveau 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.

On ne code ici plus que le niveau de l'intensité lumineuse, généralement sur un octet (256 valeurs). Par convention, la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale).

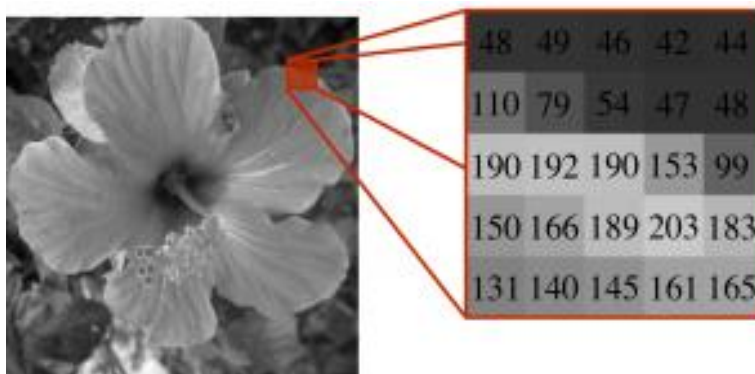


Image de taille 5x5

Figure I.12. Image numérique en niveau de gris

## 6.4. Le modèle noir et blanc

Avec ce mode, il est possible d'afficher uniquement des images en deux couleurs: noir et blanc, Chaque pixel ne peut avoir que les valeurs 0 et 1) .Le 0 correspond à un pixel noir et 1 à un pixel blanc. Le niveau de gris est donc codé sur un seul bit. Ces images sont appelées aussi images binaires.

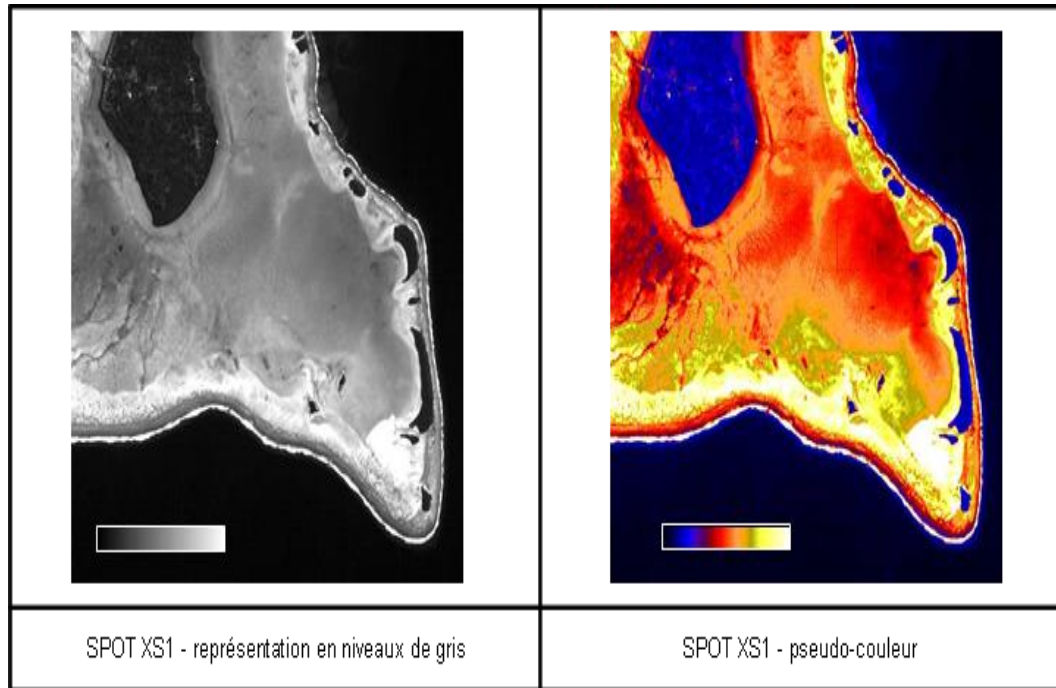
1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1

Figure I.13. Image en noir et blanc

## 6.5. Les pseudos couleurs

La représentation d'une image en pseudo-couleur est un processus un peu particulier qui ne concerne qu'une seule bande spectrale d'une image. Plutôt que de visualiser l'image en niveaux de gris, on attribue une couleur à chaque compte numérique. Cela permet de faciliter l'interprétation de l'image, l'œil humain ne permettant d'identifier de manière efficace qu'un nombre restreint de niveaux de gris.

Sur la figure ci-dessous (Figure I.13), l'image de gauche est représentée en niveaux de gris, les valeurs numériques allant de 0 (noir) à 255 (blanc). Sur l'image de droite, les valeurs numériques sont les mêmes, mais on a remplacé la table en niveaux de gris par une table couleurs s'étalant du bleu très sombre au jaune très pâle, en passant par le violet, le rouge, l'orangé, le vert et le jaune.



**Figure I.14.** Image en niveaux de gris (à gauche) et sa en représentation en pseudo couleur (à droite)

## 7. Types d'image numérique

Lorsque l'on affiche une image sur l'écran d'un ordinateur, ce que l'on voit n'est qu'une succession de points. Il existe pourtant deux manières différentes de stocker une image sur une machine, on parle d'image bitmap ou bien d'image vectorielle.

### 7.1. L'image matricielle (bitmap)

Consiste en un tableau à deux (ou plus) dimensions, dont chaque case contient la valeur d'un pixel (voir la figure 2.1). Les images fournies par les systèmes d'acquisition (microscope, appareil photo, scanner à plat...) sont de ce type. Elles sont stockées typiquement dans des fichiers aux formats .tif, .bmp, .png, .jpg...



**Figure I.15.** Une image matricielle

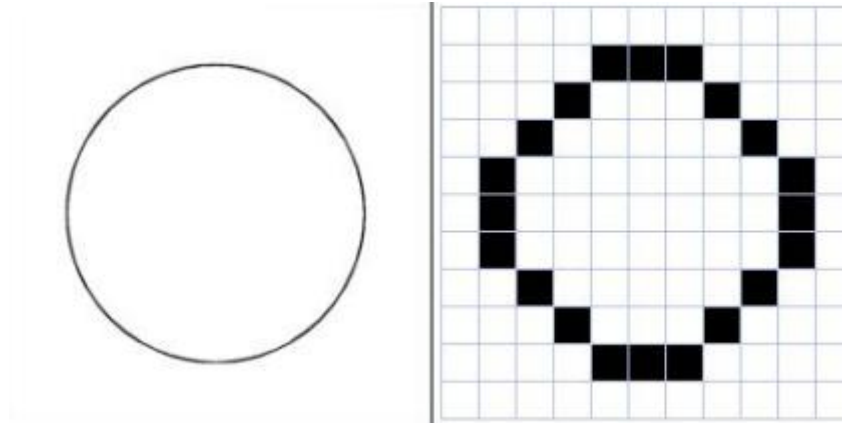
Dans une image matricielle (bitmap), les points (ou pixels) sont stockés dans un tableau. Dans chaque case du tableau se trouve la couleur que l'on doit afficher au point correspondant. Pourtant cette méthode a ses limites, si la résolution de l'image est plus grande ou plus petite que celle de l'écran, il faut agrandir ou rétrécir l'image. Si la réduction de l'image ne pose pas de problème, l'agrandissement de l'image conduit à une dégradation de l'aspect visuel de l'image.

## 7.2. L'image vectorielle

C'est une image présentée sous la forme d'une série de primitives géométriques : segment, point, cercle, polygone... Leur gros avantage est que contrairement aux images matricielles, on peut agrandir une image vectorielle autant que l'on veut sans perte de qualité. Elles sont souvent utilisées pour sauvegarder le résultat de graphiques ou de dessins techniques.

Dans une image vectorielle, les objets sont constitués de formes géométriques simples, telles que des vecteurs, rectangles, des cercles, des ellipses ... Le principal

avantage des images vectorielles par rapport aux images bitmap est qu'elles sont redimensionnables sans pertes de qualité.



**Figure I.16.** Image vectorielle (à gauche) et image matricielle (à droite)

## 8. Formats de fichiers images

La sauvegarde des images sous forme de fichier nécessite de stocker non seulement la liste des valeurs des pixels, mais aussi la taille de l'image et la manière dont sont codés les pixels.

Les images étant des données volumineuses, on peut être tenté de les compresser. Cependant, les algorithmes de compression efficaces pour les images courantes en milieu scientifique induisent une perte de données.

### 8.1. Formats de fichiers non compressés

Ce sont les formats de fichiers dit « non destructifs ». Ils enregistrent chaque pixel d'une image comme nous l'avons vu précédemment, et utilisent en général beaucoup de mémoire. De part leur poids élevé, ils ne sont pas adaptés pour le web mais doivent être utilisés lorsqu'on a besoin de préserver la totalité des informations d'une image pour retravailler dessus par exemple.

On trouve les formats suivant :

### **8.1.1. Format PSD (Document Photoshop)**

Le PSD, c'est un type de fichier conçu pour l'utilisation du logiciel Photoshop créé par Adobe. C'est un format de fichier dynamique qui prend en charge l'utilisation des calques et différents masque de traitement d'image. Il stocke les graphiques et images traités dans le logiciel Photoshop.

### **8.1.2. Format bmp (bitmap )**

L'extension BMP représente le fichier image Bitmap. Le fichier BMP contient des données graphiques matricielles qui sont indépendantes des périphériques d'affichage. Cela signifie qu'un fichier image BMP peut être visualisé sans adaptateur graphique.

Les images BMP sont généralement non compressées ou compressées avec une méthode de compression sans perte. Les fichiers peuvent stocker des images numériques bidimensionnelles en monochrome et en couleur.

### **8.1.3. Format TIFF (Tagged Image File Format)**

Le TIF est un des formats utilisés pour le stockage d'images numériques. Largement pris en charge par les différentes plateformes existantes, il est souvent approprié dans les environnements professionnels et précisément dans le domaine commercial. En effet, les fichiers au format TIF sont la référence en ce qui concerne les images de haute qualité. En dépit de leur grande consommation en termes d'espace de stockage, ils sont plus fiables pour ce qu'il en ait des impressions de haute résolution dans le domaine du textile.

### **8.1.4 Format RAW**

C'est un format brut qui « code » les images avec un maximum d'information suivant le capteur de l'appareil qui l'a créé. Il permet ensuite de développer numériquement ses photos en les enregistrant au format TIFF/EP et, pour les métadonnées, d'EXIF.

## 8.2. Format de fichiers compressés

Ce sont les formats de fichiers dit « destructifs ». Ils permettent, selon un algorithme particulier, de gagner plus ou moins de mémoire en supprimant certaines informations peu ou non perceptible par l'œil humain. Ils sont particulièrement adaptés à internet, mais ne doivent pas être utilisés lors d'un travail de création sous Photoshop car chaque nouvel enregistrement détériore un peu plus le fichier. On les utilisera donc pour exporter des images destinées à la visualisation sur internet ou l'archivage. On trouve les formats suivants :

### 8.2.1. Format JPG (JPG ou image JPEG : Joint Photographic Experts Group)

Le JPG est une extension du format JPEG qui est une norme de compression d'image numérique. C'est un type de fichier multimédia qui ne prend pas en charge la transparence des arrières plans, aussi son niveau de compression varie selon la qualité désirée.

Généralement, l'on opte pour une résolution basse pour les e-mails, une résolution moyenne pour les formats web et une grande résolution pour les documents commerciaux.

### 8.2.2. Format GIF ( Graphics Interchange Format )

Les GIF font part des fichiers multimédia les plus utilisés sur le web et pour les mails, ce sont généralement des fichiers de basse résolution, ils sont donc pris en charge par la plupart des navigateurs utilisant des algorithmes de compression pour garder leurs tailles initiales.

Contrairement au JPG, ils autorisent les fonds transparents et peuvent aussi stocker plusieurs images afin de créer des diapos ou des animations.

### 8.2.3. Format PNG (Portable Network Graphics)

Les PNG sont des formats de fichiers créés pour succéder au format GIF vu qu'elles offrent un meilleur résultat de compression sans perte de qualité (entre 5% et 25% de qualité conservée).

Tout comme les GIF, les formats PNG sont aussi répandus sur le net en raison de leur faible taille. Ce sont des images bitmap qui utilisent un algorithme de compression sans perte de données.

Toujours par analogie aux GIF, les PNG gèrent les fonds transparents. Ils sont utilisés pour l'affichage des images synthétiques destinées au web. En effet, les PNG sont recommandés par la W3C pour l'affichage des icônes, des logos, des images représentant des textes.

## **9. Conclusion**

Pour avoir l'image dans son format numérique celle-ci doit passer par tout un processus de numérisation. Dès qu'elle se présente sous sa forme numérique on peut y affecter par plusieurs traitements informatiques, dans le but d'améliorer sa qualité ou même d'analyser son contenu.

L'objectif du chapitre suivant est de présenter les principaux pré- traitements utiles dans l'analyse et l'exploitation des images numériques.

# Chapitre II : Traitement d'image numérique

## 1. Introduction

Le traitement des images est l'ensemble des opérations qui entraînent la modification des images numériques ces modification portent généralement sur la couleur des pixels. Plusieurs techniques sont utilisées pour manipuler ces données afin d'obtenir une plus grande lisibilité de l'image.

Ce chapitre présente une étude détaillée des différentes techniques de traitement d'image numérique, l'étude est centrée sur les opérations de prétraitement et la segmentation des images numérique.

## 2. Définition du traitement

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

### 2.1. Historique

Le traitement d'images commence à être étudié dans les années 1920 pour la transmission d'images par le câble sous-marin allant de New York à Londres. *Harry G. Bartholomew* et *Maynard D. McFarlane* effectuent la première numérisation d'image avec compression de données pour envoyer des fax de Londres à New York. Le temps de transfert passe ainsi de plus d'une semaine à moins de trois heures. Il n'y a pas vraiment d'évolution par la suite jusqu'à la période d'après-guerre<sup>1</sup>. Le traitement du signal prend de l'importance vers la fin de la Seconde Guerre mondiale avec l'arrivée du radar.

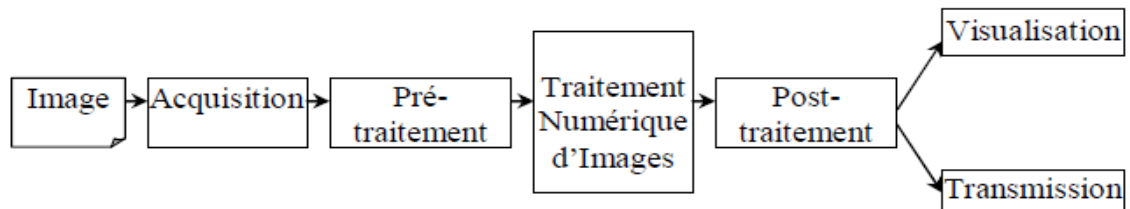
La prospection pétrolière participe aussi beaucoup au développement des techniques de traitement du signal. Le véritable essor du traitement d'images n'a lieu que dans les années 1960 quand les ordinateurs commencent à être suffisamment puissants pour travailler sur des images. Peu après, la redécouverte de la transformée de Fourier rapide (FFT) révolutionne le domaine, en rendant possible les manipulations du contenu fréquentiel des signaux sur ordinateur. Cependant, l'essentiel des recherches porte encore, à cette époque, sur l'amélioration des images et leur compression.

En 1980, *David Marr* formalise le premier la détection de contours de manière précise. Au cours des années 1980, un véritable engouement se fait jour 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 d'autant plus cuisant.

Les années 1990 voient l'amélioration constante des opérateurs. La recherche médicale devient un très gros demandeur en traitement d'images pour améliorer les diagnostics faits à partir des nombreuses techniques d'imagerie médicale, la technique reine étant l'IRM. Les publicitaires, puis le grand public se familiarisent avec la retouche d'image grâce au logiciel Photoshop, et le traitement d'images dans un objectif esthétique se répand avec l'apparition d'autres logiciels dédiés (The Gimp, Paint Shop Pro). Enfin, la décennie s'achève sur l'engouement pour les ondelettes et les images multimodales.

## 2.2. Système de traitement d'images

Un système de traitement numérique d'images est composé de :



**Figure II.1.** Schéma d'un système de traitement d'images

### 2.2.1. Prétraitement d'images numériques

La phase de prétraitement a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image, elle regroupe l'ensemble des techniques visant à améliorer l'aspect d'une image. Ces opérations sont diverses : en effet, il pourrait s'agir de renforcement de contraste, de réduction de bruit, ou d'extraction de primitives dans l'image (contours, points d'intérêt, etc..) ceci, dans le but de faciliter les traitements ultérieurs tel que segmentation ou la reconnaissance.

Les technique de prétraitement les plus courantes qu'on va présenter sont :

- ✚ La restauration : correction des défauts dus à une source de dégradation.
- ✚ L'amélioration : modification de l'image dans le but de la rendre plus agréable à l'œil.
- ✚ La compression : réduction du volume de l'image.

### 2.2.2. Analyse d'images numériques

Suite d'opérations pour l'extraction d'information contenue dans une image :

- ✚ Phase de segmentation avec partition de l'image.
- ✚ Techniques de description / modélisation pour obtenir la description structurelle de l'image.

### 2.2.3. Interprétation

Cette étape vise à déterminer d'autres primitives de haut niveau des objets à partir de celles données dans les étapes précédentes. Son objectif est de déterminer l'identité des objets perçus, leurs positions dans l'image ou par rapport au capteur et éventuellement d'autres informations nécessaires pour une application précise. Les méthodes de reconnaissance sont souvent couplées avec des méthodes de localisation. Passage de la description structurelle à la description sémantique en regard à certains objectifs : mesure de paramètres sur des formes, description du contenu de la scène en termes de concepts non mathématiques.

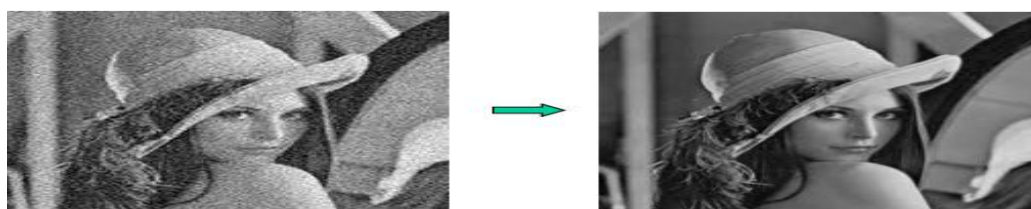
## 3. Opérations de prétraitement

Cette étape a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation, les opérations de prétraitement sont :

### 3.1. Restauration des images

La restauration d'images a pour objet la réduction, voire l'élimination des distorsions introduites (bruits) par le système ayant servi à acquérir l'image, son but est d'obtenir une image qui soit la plus proche possible de l'image idéale qui aurait été obtenue si le système d'acquisition était parfait.

La restauration cherche à rétablir la valeur exacte des pixels. On part à la recherche d'une estimation de l'image qui soit visuellement ou analytiquement intéressante, c'est-à-dire qu'elle minimise une fonction d'erreur.



(a) Image fortement bruitée

(b) Image résultante d'une restauration

**Figure II.2.** Exemple sur la restauration d'une image

### 🌈 Bruit

L'image acquise est composée d'une partie "utile" (signal) et d'une partie parasite indésirable appelée "bruit".

### 🌈 Types de bruit

Selon la cause, on peut distinguer les types de bruit suivants :

- ✓ Bruit lié au contexte de l'acquisition : bougé, dérive lumineuse, flou, poussière
- ✓ Bruit lié au capteur : distorsion de la gamme des niveaux de gris, distorsion géométrique, mauvaise mise au point, ...
- ✓ Bruit lié à la numérisation

## 3.2. Amélioration des images

L'amélioration a pour but de satisfaire l'œil de l'observateur humain. L'œil humain est essentiellement sensible aux forts contrastes. C'est pourquoi les techniques d'amélioration tentent d'augmenter ceux-ci dans le but d'accroître la séparabilité des régions composant une scène.

Il existe différentes méthodes afin d'améliorer la qualité d'une image. Il y a des méthodes qui passent par le filtrage, d'autre par le calcul d'histogramme ou encore par l'amélioration du rapport image/bruit. Ces méthodes doivent être utilisées en fonction de ce que l'on veut obtenir. En effet chacune de ces méthodes aurait un effet bien spécifique sur l'image que se soit pour augmenter les contrastes ou bien mettre en avant des petits détails de l'image.

### 3.2.1 Amélioration du rapport image/bruit

Cette méthode permet de diminuer le bruit dans une image. La technique consiste à prendre plusieurs fois l'image d'un même point de vue, on obtient donc  $N$  l'image avec  $N$  du bruit, ensuite il suffit d'appliquer un algorithme qui supprime les pixels différents entre chaque image et qui garde les pixels identiques. En effet, ces algorithmes se basent sur le fait que le bruit est différent à chaque prise tandis que l'image, elle, est identique.

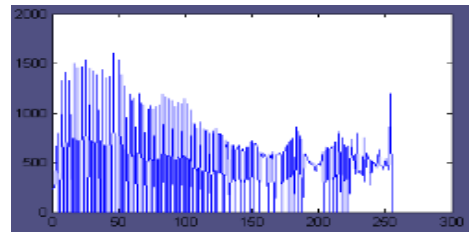
### 3.2.2 Amélioration par histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées.

Un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé au plus clair. (Figure II.3)



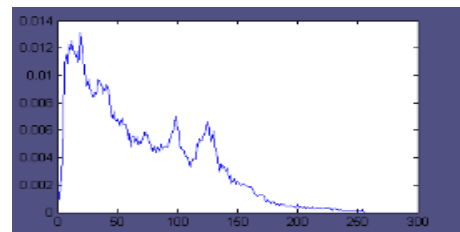
Image (a)



Histogramme correspondant à l'image (a)



Image (b)



Histogramme correspondant à l'image (b)

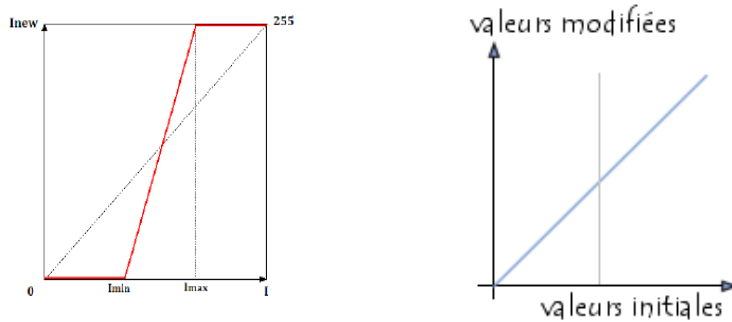
**Figure II.3.** Histogramme d'une image numérique

**Normalisation d'histogramme**

La normalisation d'histogramme est une opération qui consiste à modifier les valeurs de chaque pixel de manière à ce que l'image résultante exploite toute sa dynamique, améliorant ainsi le contraste de l'image. Ceci revient à un étirement de l'histogramme afin que la valeur d'intensité la plus faible  $I_{\min}$  soit à zéro et que la plus haute  $I_{\max}$  soit la valeur maximale (255) :

$$I_{\text{new}}(i, j) = (I(i, j) - I_{\min}) \times \frac{255}{I_{\max} - I_{\min}} \quad (\text{II.1})$$

La modification d'un histogramme est généralement représentée sur une courbe (appelée courbe tonale) indiquant la modification globale des composantes de l'image avec en abscisse les valeurs initiales et en ordonnées les valeurs après modification.



**Figure II.4.** Courbe tonale de la normalisation (1) d'histogramme

Donc si la courbe de modification de l'histogramme est située en dessous de la diagonale, les pixels auront des valeurs plus faibles et seront donc éclaircis. A l'inverse si la courbe est au dessus de la diagonale, les pixels seront assombris.

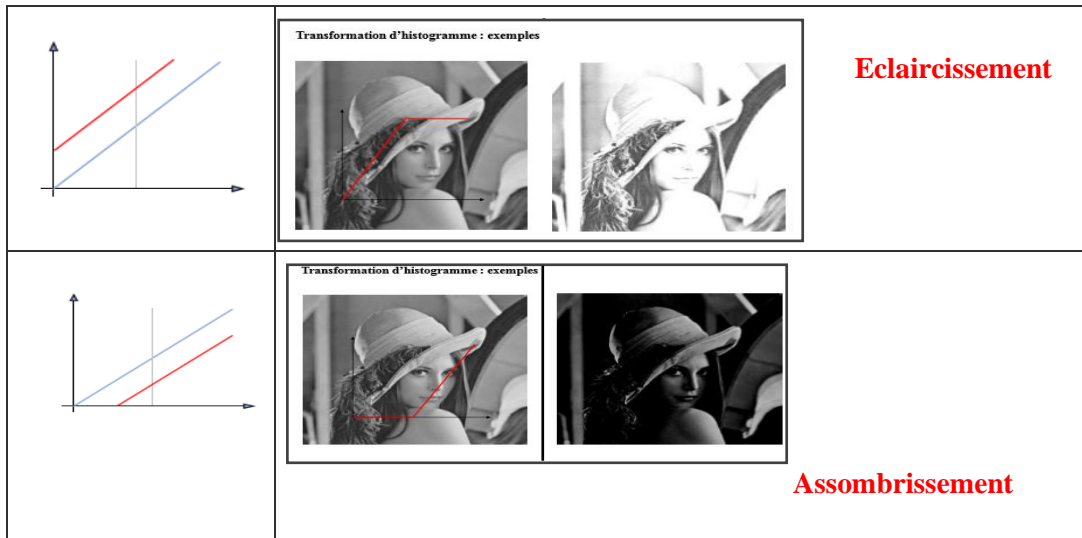


Figure II.5. Courbe tonale de la normalisation (2)

**Egalisation d'histogramme**

L'égalisation d'histogramme consiste à harmoniser la répartition des niveaux de luminosité de l'image, de manière à tendre vers un même nombre de pixel pour chacun des niveaux, ceci revient à l'aplatissement de l'histogramme. Cette technique améliore le contraste et permet d'augmenter artificiellement la clarté d'une image grâce à une meilleure répartition des intensités (Figure II.6).



Figure II.6 : Egalisation d'histogramme

La courbe tonale correspondant à l'égalisation de l'histogramme dépend totalement de l'image.

✓ La nouvelle image est calculée de la manière suivante :

$$I_{new}(i, j) = 255 \times \sum_{k=0}^{I(i,j)} h_n(k) \quad (\text{II.2})$$

### ✚ Amélioration par filtrage

Le filtrage peut agir de deux manières :

- ✚ Il peut s'agir d'une part, de filtrer les imperfections de l'image telle que le bruit
- ✚ D'autre part, de filtrer l'information portée par l'image de manière à en faire ressortir des primitives telles que les contours ou les points d'intérêts.

Il existe plusieurs types de filtres, nous allons citer :

#### ✓ Filtre Gaussien

Le filtre de Gauss est, en électronique et en traitement du signal, un filtre dont la réponse impulsionnelle est une fonction gaussienne.

Le filtre de Gauss est élaboré pour ne pas donner d'excédant à la fonction échelon tout en minimisant le temps de montée et descente.

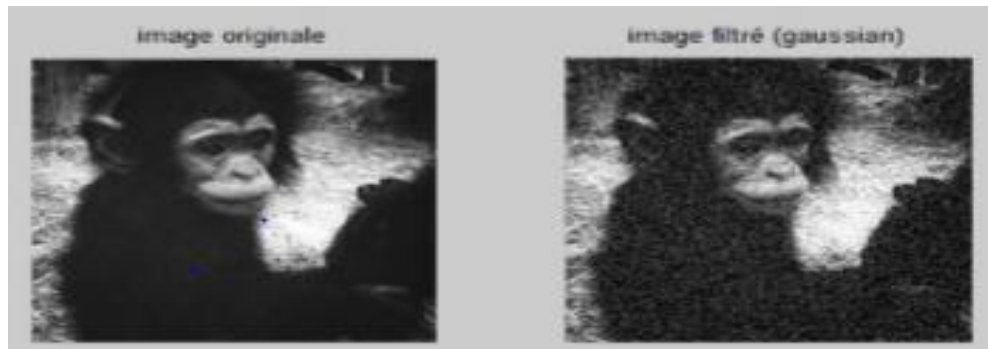
Cette propriété est étroitement liée au fait que le filtre de Gauss possède le moins de délai de groupe possible. En mathématiques, le filtre de Gauss modifie le signal entrant par une convolution avec une fonction gaussienne.

En une dimension le filtre de Gauss à une réponse impulsionnelle est de la forme suivante :

$$G(x) = \sqrt{a/\pi} * e^{-a*x^2} \quad (\text{II.3})$$

Ou bien avec la déviation standard en paramètre :

$$G(x) = \frac{1}{\sqrt{2*\pi*\sigma}} * e^{-\frac{x^2}{2\sigma^2}} \quad (\text{II.4})$$



**Figure II.7.** Filtrage par le filtre gaussien

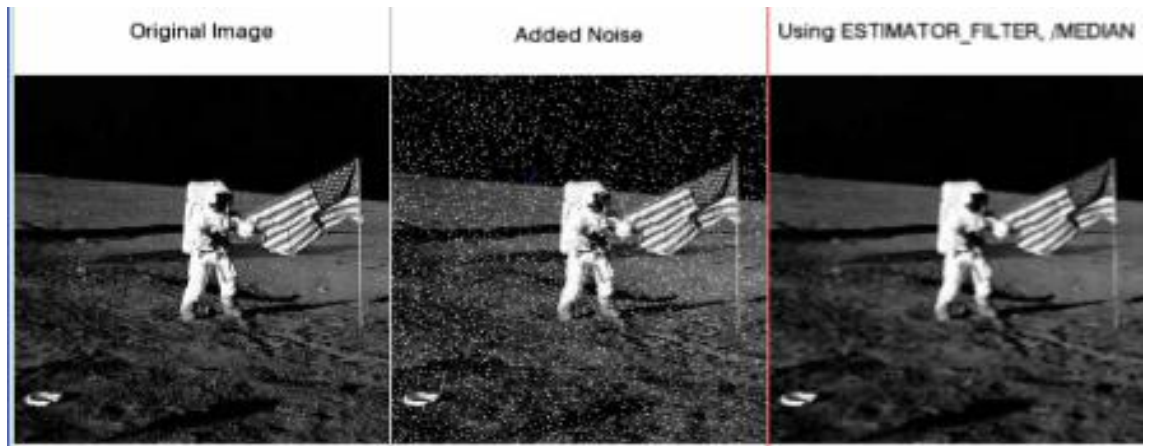
### ✓ *Filtre Médian*

La médiane est une mesure statistique représentant une alternative robuste à la moyenne. Considérant  $n$  valeurs numériques ou  $n$  est impair.

La valeur médiane correspond à la valeur du centre de la suite lorsque celle-ci est ordonnée du plus petit au plus grand. L'application du filtre médian dans une image consiste donc à remplacer chaque pixel par la valeur médiane de l'ensemble de son voisinage.

Le résultat que l'on obtient avec ce filtre est qu'il élimine les pixels isolés (par exemple un pixel noir parmi une surface blanche) tout en évitant de faire un flou trop important dans l'image (qui se verrait si c'était le cas).

Ce filtre est très utilisé pour éliminer le bruit sur une image qui peut être de différentes origines (poussières, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, ...). L'avantage de ce filtre réside dans le fait qu'il conserve les contours alors que les autres types de filtres ont tendance à les adoucir. (Figure II.8)



**Figure II.8.** Les étapes de filtrage par filtre médian

✓ **Filtre moyennneur**

Le filtre moyennneur est une opération de traitement d'images utilisée pour réduire le bruit dans une image et/ou flouter une image. Le filtre moyennneur permet de : lisser l'image (smoothing) , Remplace chaque pixel par la valeur moyenne de ses voisins , Réduit le bruit , Réduit les détails non-important , Brouille ou rend floue l'image (blur edges)



**Figure II.9.** image originale



**Figure II.10.**image filtrée par un filtre moyennneur de taille 3\*3



**Figure II.11.**image filtrée par un Filtre moyennneur de taille 7\*7

### 3.3. Compression des images numériques

La compression des images est une application de la compression des données sur des images numériques. Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement.

#### 3.3.1. Les approches de compression

Le but des algorithmes de compression est de réduire la taille des fichiers pour diminuer l'espace nécessaire à leur stockage sur le disque ou leur transfert par le réseau. Ils encodent d'une manière différente les données de l'image afin de les rendre plus compactes. Deux familles d'algorithmes de compression se distinguent :

- ✚ La compression sans perte d'information
- ✚ La compression avec perte d'information

#### ✓ La compression sans perte d'information

Dans ce type de compression, aucune information n'est perdue lors de la compression, et le critère objectif consiste à minimiser le code binaire en évitant la distorsion de l'image (erreur nulle entre l'image originale et l'image compressée). Il existe différents algorithmes/codages de la compression sans perte d'information, nous se limitons aux codages :

- ✚ RLE (Run Length Encoding) ou RLE
- ✚ Algorithme (Lempel-Ziv-Welch) ou LZW
- ✚ La méthode HUFFMAN
- ✚ La méthode entropique de Shannon...

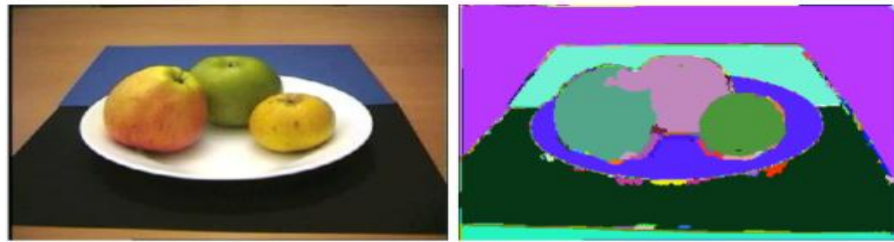
#### ✓ La compression avec perte d'information

Dans ce type de compression la taille de l'image est nettement réduite mais au détriment d'une perte d'information. Il existe différents codages de la compression avec perte d'information, nous se limitons aux codages :

- ✓ La transformée en cosinus discrète
- ✓ La compression par ondelettes
- ✓ La compression fractale

## 4. La segmentation d'images

Le traitement d'images joue aujourd'hui un rôle important dans de nombreux domaines. Dans un système de traitement d'images, l'opération la plus importante est la segmentation d'image. La segmentation automatique est la tâche la plus difficile en traitement d'images, plus la segmentation est meilleur plus l'étape de reconnaissance d'objets est réussite.



**Figure II.12.** La segmentation de l'image en morceaux

### 4.1. Définition

La segmentation d'image est une opération de traitement d'images qui a pour but de regrouper des pixels de l'image en régions. Ces régions vérifiant un critère d'homogénéité (par exemple sur les niveaux de gris ou sur la texture ... ). On cherche par ce traitement à obtenir une description compactée de l'image en régions qui constituent une partition de l'image.

Il peut s'agir par exemple de séparer les objets du fond. Si le nombre de classes est égal à deux, elle est appelée aussi binarisation.

Formellement : la segmentation d'une image numérique « I » utilisant un prédicat d'homogénéité P est communément définie comme une partition  $S = R_1, R_2, R_3, \dots, R_n$  (R : régions) telle que :

1.  $I = \cup_{i \in [1..n]} R_i$
2.  $R_i$  est connexe,  $\forall i \in [1..n]$
3.  $P(R_i) = \text{vrai}, \forall i \in [1..n]$
4.  $P(R_i \cup R_j) = \text{faux} \forall i \neq j$

## 4.2. Objectifs de la segmentation

La segmentation d'image a pour but, les points ci-dessous :

- ✚ Extraire, séparer les entités d'une image en régions homogènes (selon un critère donné), pour leur appliquer un traitement spécifique.
- ✚ Localiser de manière précise les contours des régions pour interpréter le contenu de l'image.
- ✚ Elle est réalisée avant les étapes d'analyse et de prise de décision dans plusieurs processus d'analyse d'image, tel que la détection des objets

## 4.3. Les différentes techniques de segmentation d'image

La segmentation est une étape primordiale en traitement d'images. Jusqu'à maintenant il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en cinq principales classes :

- ✚ La segmentation par seuillage de niveau de gris
- ✚ La segmentation par régions
- ✚ La segmentation par contours
- ✚ La segmentation par classification
- ✚ Ligne de partage des eaux
- ✚ La segmentation par modèle déformable

### 4.3.1. Technique de segmentation par seuillage de niveau de gris

Le seuillage est sûrement la méthode de segmentation la plus simple. Elle consiste à tester pour chaque pixel de l'image si sa valeur est supérieure ou inférieure à un certain seuil, et produit une image binaire regroupant les résultats. Pour son efficacité et sa simplicité théorique, le seuillage a inspiré beaucoup de recherches et plusieurs méthodes de seuillage ont été développées. Ces méthodes peuvent être divisées, entre autre, en deux catégories. Les méthodes de seuillage global et Les méthodes de seuillage local.

### ✚ **Seuillage global**

Dans ces méthodes, un seul seuil est calculé pour toute l'image à segmenter, donc, deux classes sont déterminées et les pixels sont classés dans l'une ou l'autre de ces deux classes, selon que la valeur de leurs niveaux de gris est inférieure ou supérieure à ce seuil.

### ✚ **Seuillage local**

Le principe du seuillage local est d'utiliser une étude localisée autour du pixel pour déterminer quel seuil utiliser. Pour réaliser cette étude locale, les techniques utilisent une fenêtre d'étude (le voisinage) centrée sur le pixel à étudier.

#### **4.3.2. Technique de segmentation par région**

La segmentation d'image par l'approche région consiste à découper l'image en régions. Dans cette approche on distingue les méthodes par croissance de région, et par division/fusion.

### ✚ **Segmentation par décomposition/fusion (Split and Merge)**

Cette technique enchaîne le découpage itératif et fin de l'image jusqu'à avoir des blocs contenant exclusivement des pixels similaires et puis la fusion de régions adjacentes selon un critère d'homogénéité.

- ✓ **Décomposition :** Le processus de division peut, dans le cas extrême, aboutir à un maillage de l'image sous forme de pixels indépendants. Son principe :
  - Faire une dichotomie par blocs de l'image
  - Pour cela, on commence par définir un bloc de la taille de l'image
  - Puis on examine le contenu de ce bloc :
  - Si le bloc est homogène alors on arrête la décomposition
  - Sinon, on découpe le bloc en 04 sous-blocs et on examine le contenu de chaque sous-bloc.

- Et ainsi de suite jusqu'à ce qu'il ait plus besoin de décomposer les blocs
  - Le résultat obtenu est donc un ensemble jointif de blocs de différentes tailles qui recouvrent entièrement l'image.
- ✓ **Fusion :**
- ✓ Il faut tout d'abord définir le critère de similarité entre blocs
  - ✓ Ainsi, on peut assimiler un bloc à un « gros » pixel en calculant sa valeur/couleur moyenne.

### **Segmentation par croissance de région**

Les méthodes par croissance de région, sont basées sur un processus itératif d'agrégation de pixels. Ces méthodes de par leur relative simplicité de mise en œuvre ont été l'une des premières familles d'outils utilisés pour la segmentation d'image.

#### **4.3.3. Technique de segmentation par détection de contour**

Un contour est un ensemble des points d'une image numérique qui correspond à un changement brutal de l'intensité lumineuse. Dans la technique de segmentation par détection de contour, on considère que les primitives à extraire sont les lignes de contrastes séparant des régions de niveaux de gris différents et relativement homogènes, ou bien des régions de texture différentes. Il existe plusieurs techniques de filtrage qui permettent d'effectuer des segmentations par détection de contours, on peut citer : le filtre de Prewitt, le filtre de Sobel, le filtre de Robert...etc.

#### **Le filtre de Prewitt**

Il tient son nom de Judith M. S. Prewitt. Ce filtre utilise deux matrices de dimensions 3, La première matrice donne la dérivée horizontale et la seconde donne la dérivée verticale. Si nous définissons A comme étant l'image source, et  $G_x$  et  $G_y$  les deux images dérivées horizontale et verticale (respectivement) de l'intensité lumineuse de l'image, on peut les calculer ainsi avec l'opération de convolution 2D suivante :

$$G_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{pmatrix} * A \quad (\text{II. 5})$$

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{pmatrix} * A \quad (\text{II. 6})$$

Cet opérateur ainsi défini pouvant se décomposer en un produit d'une moyenne et d'une différentielle, il calcule le gradient lissé.

Par exemple,  $G_x$  peut s'écrire :

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [-1 \quad 0 \quad 1] \quad (\text{II. 7})$$

Enfin, nous pouvons calculer l'amplitude du gradient via la norme du vecteur ainsi défini en tout point :

$$G = \sqrt{G_x^2 + G_y^2} \quad (\text{II.8})$$

De même, nous pouvons calculer sa direction :

$$\Theta = \arctan(G_y, G_x) \quad (\text{II.9})$$

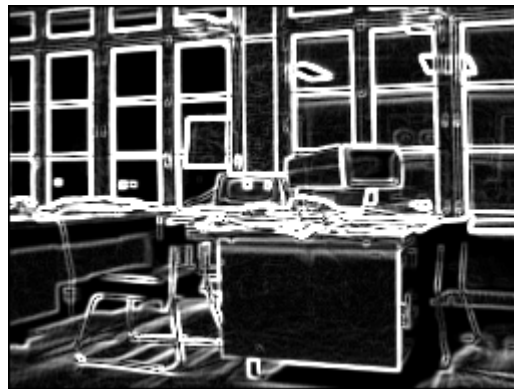


Par exemple,  $\Theta$  correspond à un front vertical qui est plus sombre du côté Droit.

**Le filtre de Sobel**

Il s'agit d'un des opérateurs les plus simples qui donne toutefois des résultats corrects. Ce filtre est une variante du filtre de Prewitt mais permet de mettre plus l'accent sur une direction particulière grâce aux 2 dans les trois matrices de ce filtre Horizontal, Verticale et Oblique

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \quad \begin{pmatrix} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{pmatrix} \quad (\text{II.1})$$



**Figure II.14.** Application du filtre de Sobel

**Le filtre de robert**

Le filtre de Roberts permet de calculer le gradient d'une image en suivant les diagonales grâce aux deux matrices suivantes : Matrice Sud-Ouest, Nord-Est et la Matrice Nord-Ouest, Sud-Est

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$



Figure II.15. Application du filtre de Roberts

✚ Le filtre Sepia :

En photographie, le sépia est une qualité de tirage qui ressemble au noir et blanc, mais avec des variations de brun, et non de gris. La couleur sépia dans le système RVB est S(94, 38, 18). Dans la transformation d'une image couleur en une image en nuances de sépia, on tient compte d'un seuil ( $0 < \text{seuil} < 255$ ), qui sépare le sépia assombri et le sépia éclairci. La transformation se fait alors pixel par pixel en deux temps. Pour chaque pixel, on calcule d'abord un niveau de gris qui est la moyenne  $m$  des intensités de rouge, vert et bleu. Puis, si le gris obtenu est foncé ( $m < \text{seuil}$ ), on le remplace par une couleur du segment NS, couleur d'autant plus proche du noir que  $m$  est petit. Si le gris est plus clair ( $m > \text{seuil}$ ), il est remplacé par une couleur du segment SB, couleur d'autant plus proche du blanc B que  $m$  est grand.

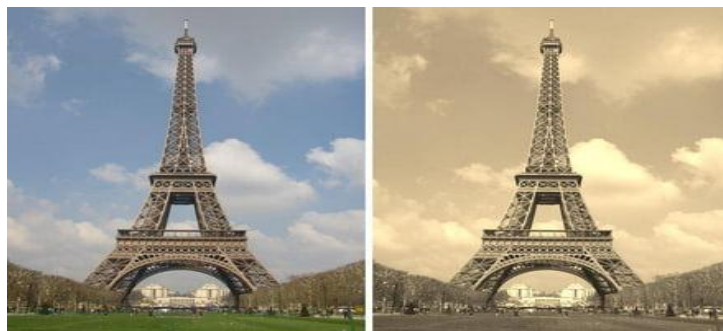


Figure II.16. Application du filtre Sépia

#### 4.3.4. Technique de segmentation par classification

Les méthodes de classification permettent de regrouper des objets en groupes ou classes d'objets plus homogènes. Les objets regroupés ont des caractéristiques communes, ils sont similaires mais se distinguent clairement des objets des autres classes. Autrement dit, ce type de segmentation a pour objectif d'affecter chaque pixel à une classe unique.

Il existe plusieurs algorithmes de classification, notons :

##### **Algorithme K-Means (K moyenne)**

C'est l'un des algorithmes de *clustering* les plus répandus. Il permet d'analyser un jeu de *données* caractérisées par un ensemble de *descripteurs*, afin de regrouper les données "similaires" en groupes (ou *clusters*).

**Entrée :** k le nombre de groupe cherchés

**DEBUT**

Choisir aléatoirement les centre des groupes

**REPTER**

i. Affecter chaque cas au groupe dont il est le plus proche à son centre

ii. Recalculer le centre de chaque groupe

**JUSQU'A** (stabilisation des centres)

**OU** (nombre d'itérations=t)

**OU** (stabilisation de l'inertie totale de la population)

**FIN**

**Algorithme II.1.** Déroulement de l'algorithme K-Means

**✚ Algorithme C-Moyenne Floues (FCM)**

L'algorithme des c-moyens flous (fuzzy c-means) est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité. L'algorithme nécessite de connaître le nombre de classes au préalable et génère les classes par un processus itératif en minimisant une fonction objectif. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une région donnée.

1. choisir le nombre de classes :  $c$  // information à priori, algorithme supervisé.
2. initialise la matrice de partition  $U$ , ainsi que les centres  $c_K$  (initialisation aléatoire) ;
3. faire évoluer la matrice de partition et les centres suivant les deux équations :
  - (1)  $u_{ik} = 1 / \left( \sum_{j=1, j \neq k}^c (d_{ik} / d_{ij})^{2/(m-1)} \right)$ , // mise à jour des degrés d'appartenances,
  - Ou :  $d_{ij} = \|x_i - c_j\|$ ,
  - (2)  $c_k = (\sum_i (u_{ik})^m \cdot x_i) / (\sum_i (u_{ik})^m)$ , // mise à jour des centres.
4. test d'arrêt :  $|J^{(t+1)} - J^{(t)}| < \text{seuil}$ .

**AlgorithmeII.2.** Déroulement de l'algorithme FCM

**✚ .Machine à vecteur de support**

Machine à vecteurs de support (ou Support Vector Machine en anglais) : Les SVMs sont une famille d'algorithmes d'apprentissage automatique qui permettent de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie. Ils sont connus pour leurs solides garanties théoriques, leur grande flexibilité ainsi que leur simplicité d'utilisation même sans grande connaissance de data mining.

### ✚ Champs Markov

Les champs de Markov permettant de représenter une relation de voisinage, ils sont particulièrement adaptés à la segmentation puisqu'il existe beaucoup de relations spatiales dans une image. La théorie des champs Markoviens a donc été largement développée ces dernières décennies et appliquée dans différents domaines du traitement d'image.

### ✚ Technique de segmentation par Ligne de Partage des Eaux (LPE)

L'algorithme de la ligne de partage des eaux (LPE), proposé par Digabel et Lantuéjoul, est un algorithme de segmentation d'images, ce qui revient à décomposer une image en régions homogènes. La ligne de partage des eaux utilise la description des images en termes géographiques. Une image peut en effet être perçue comme un relief si l'on associe le niveau de gris de chaque point à une altitude. Il est alors possible de définir la ligne de partage des eaux comme étant la crête formant la limite entre deux bassins versants.

#### 4.3.5. Technique de segmentation par modèle déformable

Les modèles déformables ont donné naissance à de nombreuses études depuis plus d'une décennie, ils permettent d'introduire une connaissance de la forme des objets recherchés mais aussi beaucoup d'autres informations comme la variabilité statistique, l'intensité des structures, etc. Les modèles déformables produisent directement une représentation géométrique des objets segmentés. En trois dimensions, leur représentation surfacique est particulièrement adaptée à la visualisation des objets qu'ils modélisent, ce qui correspond à l'une des premières attentes des praticiens du domaine de l'informatique médicale.

### ✚ Méthode des contours actifs (Snackes)

Un contour actif est un ensemble de points qu'on va tenter de déplacer pour lui faire épouser une forme. Il s'agit d'une technique d'extraction de données utilisée en traitement d'images. L'idée de cette méthode est de déplacer les

points pour les rapprocher des zones de fort gradient tout en conservant des caractéristiques comme la courbure du contour ou la répartition des points sur le contour ou d'autres contraintes liées à la disposition des points.

### ✚ Méthode d'ensemble de niveau zéro (Level Set)

La méthode des ensembles de niveau zéro est une méthode de simulation numérique utilisée pour l'évolution des courbes et des surfaces dans les domaines discrets.

L'idée de base de la méthode des Levels Sets est de considérer une courbe ( ou surface) en mouvement comme le niveau zéro d'une fonction de dimension plus élevée.

## 5. Conclusion

J'ai présenté, tout au long de ce chapitre, les différentes notions possibles de traitement d'images comme : la définition, l'objectif, aussi nous avons abordé le problème général de la segmentation d'images en mettant l'accent sur les méthodes de segmentation les plus utilisées.

Dans le dernier je présente l'organigramme conceptuel de mon application, puis je passe au langage et à l'environnement de développement utilisé, et je termine par la présentation du menu et des commandes de l'application ainsi que quelques scripts de code.

# **Chapitre III :** **Conception et réalisation** **de l'application**

## 1. Introduction

Le traitement d'images numériques est la mise en œuvre d'algorithmes de traitement destinés à extraire des informations significatives d'une image numérique prétraitée. L'objectif de ce chapitre est de développer une application qui présente un ensemble de commandes permettant le traitement d'images numériques.

Nous commençons par la description du langage de programmation utilisé, le choix de l'environnement de développement « Eclipse », ainsi que les étapes fondamentales de la conception de notre application. Nous terminons par une présentation des interfaces graphiques de l'application.

## 2. Conception

Cette section traite la représentation conceptuelle de l'application de traitement d'images qu'on a développé, nous employons le diagramme de cas d'utilisation UML pour donner une vision globale du comportement fonctionnel d'un système logiciel, ce dernier est suivi d'un schéma conceptuel récapitulant les différentes fonctionnalités de l'application.

### 2.1. Diagramme de cas d'utilisation de l'application

« L'utilisateur » étant l'acteur principal. Les cas d'utilisation de base qui vont être mis en évidence pour effectuer le traitement des images seront :

1. S'authentifier
2. Ouvrir une image
3. Enregistrer image
4. Convertir image
  -  Conversion en niveau de gris
  -  Conversion en noir et blanc
  -  Négatif
5. Amélioration image
  -  Amélioration par histogramme
  -  Sépia
  -  Filtre Moyenneur
  -  Filtre Médiane
  -  Filtre Gauss

6. Edition

- ✚ Rotation image
- ✚ Agrandir image
- ✚ Réduire image

7. Segmentation

- ✚ Détection de contour
  - Filtre Prewitt
  - Filtre Sobel
  - Filtre Canny
- ✚ Segmentation par methode K-Means

8. Compression & décompression

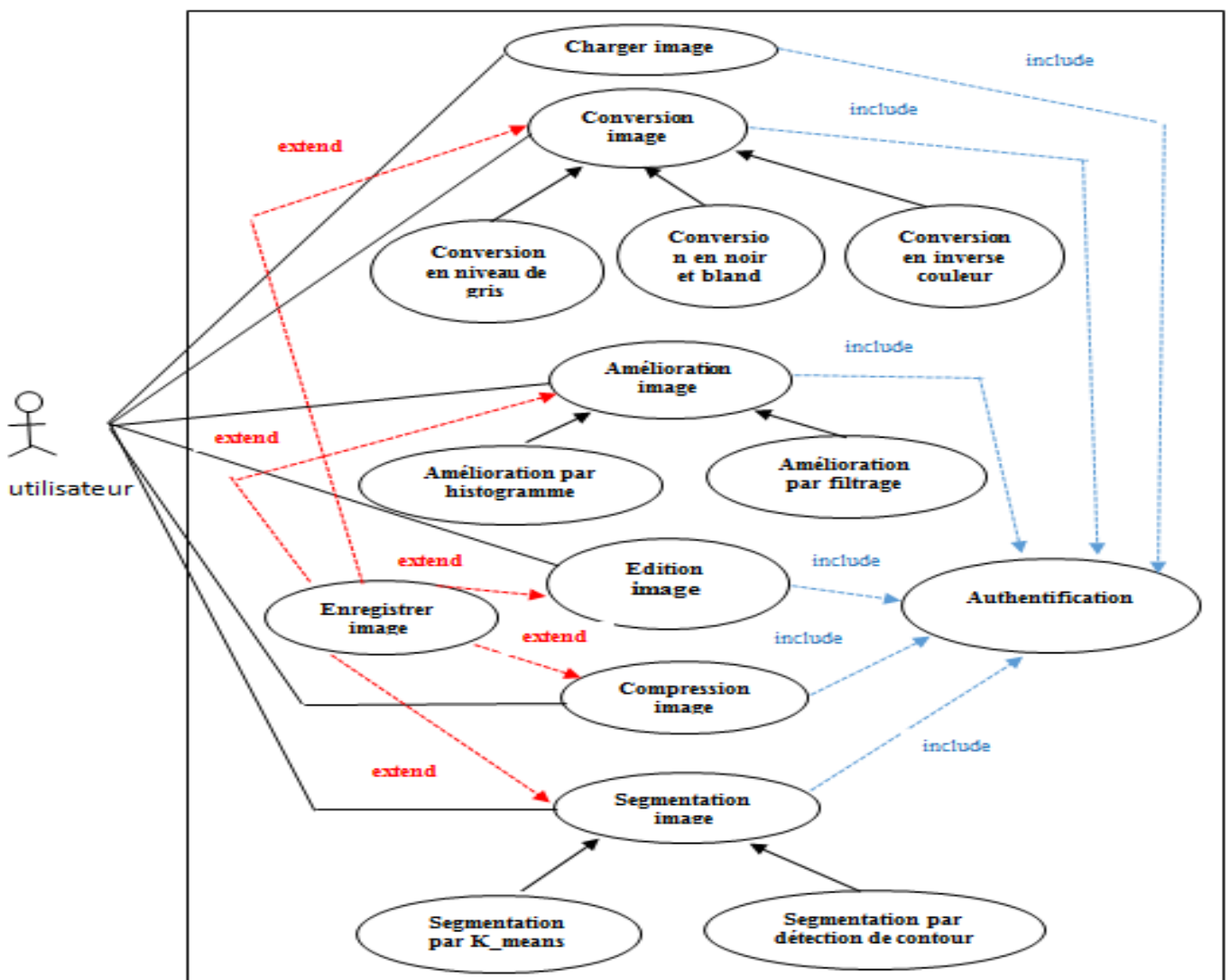


Figure III.1. Diagramme de Cas d'utilisation de l'application traitement d'image numérique

2.2. Diagramme d'activité de l'application

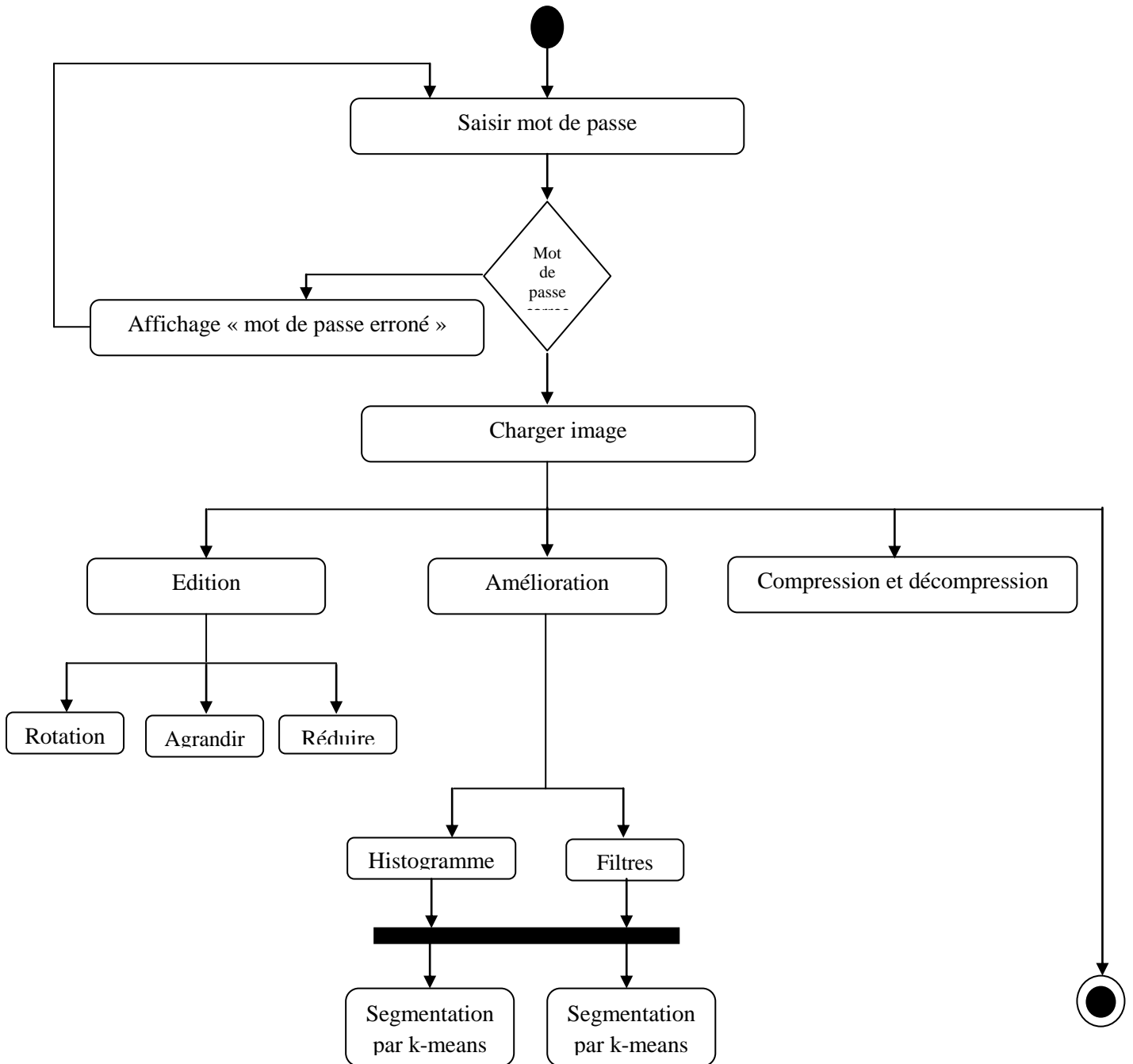


Figure III.2. Diagramme d'activité de l'application traitement d'image numérique

### 3. Présentation des outils de développement

La présente section est consacrée à la présentation des outils de développements sur lesquelles nous étions basées pour le développement de l'application de traitement d'images, il s'agit du langage de programmation Java et l'environnement de développement Eclipse.

#### 3.1. Le langage de programmation utilisé (java)



Le langage Java est un langage orienté objet créé en 1995 par Sun Microsystems. C'est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Aujourd'hui, le langage Java est très largement utilisé à la fois sur le Web, les stations de travail, les téléphones et autres tablettes. Il appartient à Oracle.

Java est un langage hybride, à la fois compilé et interprété. On dit qu'il est semi-compilé. Pour simplifier, disons qu'un programme Java est compilé dans un langage qui devra ensuite être interprété. Le résultat de la compilation n'est pas du langage machine directement exécutable (propre au processeur), mais un code intermédiaire appelé **byte-code**. Parmi les compilateurs java on site :

- ✚ JDK (Java Developer Kit) :
- Eclipse
- Netbeans
- JBulder
- Java Creator ...

Le byte-code est intermédiaire entre le code source et le langage machine. Pour exécuter le programme, le byte-code est interprété par un interpréteur appelé machine virtuelle Java (JVM).

Toutes les machines actuelles possèdent une JVM. Ainsi, le byte-code d'un programme peut être exécuté sur n'importe quel ordinateur (possédant une JVM). C'est pour cela que le langage Java est un langage portable.

Les versions de Java pour les systèmes Windows, Solaris et Linux sont disponibles sur Internet. Chaque version de Java est fournie sous deux formes :

- ✚ L'une pour les développeurs : le **JDK** (Java Development Kit) ou **SDK** (Software Development Kit) comprenant la machine virtuelle Java

pour un système d'exploitation, la bibliothèque des classes Java et les commandes pour développer en Java (commande de compilation ...)

- ✚ L'autre pour les utilisateurs : le **JRE** (Java Runtime Environment) comprenant la machine virtuelle Java et la bibliothèque des classes.

On Java on peut gérer les interfaces graphiques à travers deux types de bibliothèques de base:

- ✓ **awt (Abstract Window Toolkit, JDK 1.1)** : c'est la première API proposée pour développer des interfaces graphiques. Elle fait appel aux composants graphiques du système d'exploitation
- ✓ **swing** : proposée depuis le JDK 1.2. Swing est construit au-dessus de AWT :

- ✚ même gestion des événements

- ✚ les classes de *Swing* héritent des classes de AWT

Mais ses composants écrits complètement avec Java et sont indépendants de l'OS (système d'exploitation)

- ✚ Tous les composants de AWT ont leur équivalent dans Swing mais qui sont plus jolis avec plus de fonctionnalités
- ✚ Swing offre de nombreux composants qui n'existent pas dans AWT
- ✚ Actuellement, SWING est souvent choisi car il est fourni en standard dans le SDK

Il est préférable de n'en utiliser qu'un seul à la fois, tant les risques d'erreurs peuvent survenir entre les deux API graphiques (leurs syntaxes sont très proches).

Outre ces deux API d'autres API sont fournies tel que : **SWT** du projet Eclipse et plus récemment **JFace**, **GWT**.

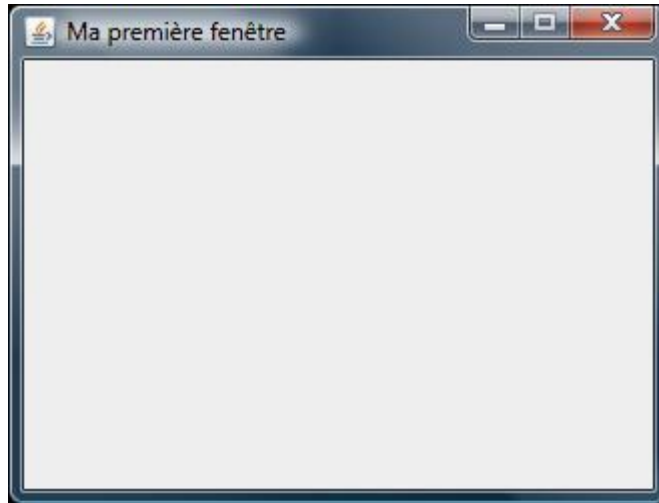


Figure III.3. Une fenêtre en java

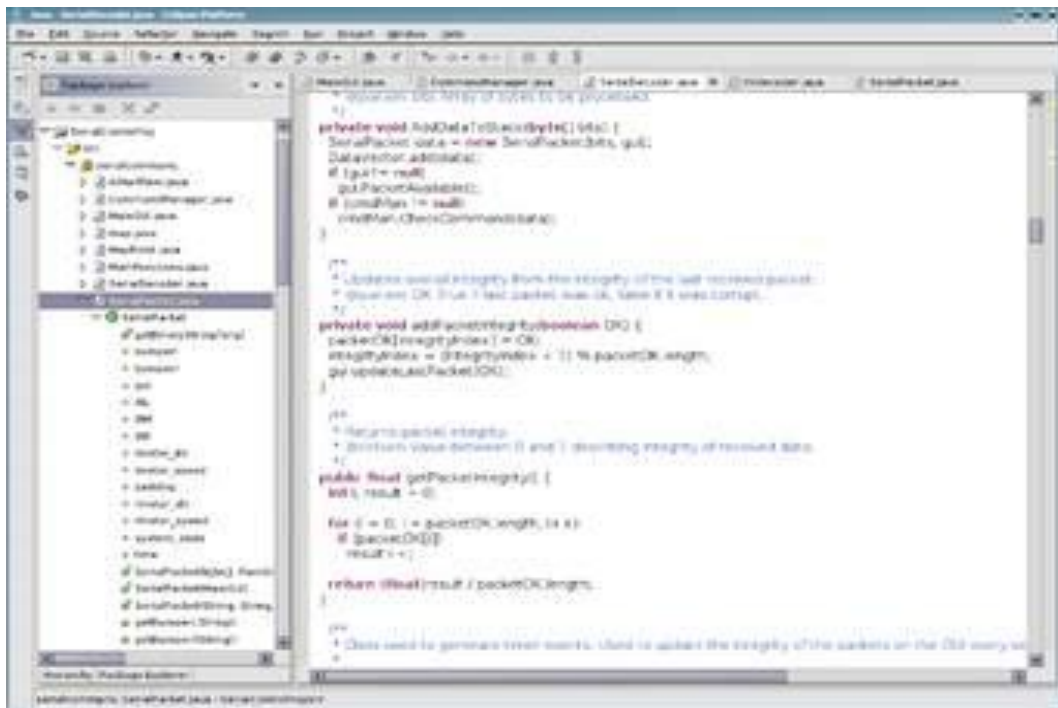
On peut distinguer :

- 1- Des composants qui sont destinés spécifiquement à recevoir d'autres éléments graphiques qui sont :
  - ✓ Les containers « *top-level* » dit aussi lourds : **JFrame**, **JApplet**, **JDialog**, **JWindow**
  - ✓ Les containers « intermédiaires » légers **JPanel**, **JScrollPane**, **JSplitPane**, **JTabbedPane**, **Box**
- 2- Des composants légers (*widgets*) : qui sont des composants simples dont un conteneur peut les contenir tel que : des boutons (**JButton**), des zone de texte (**JTextField**)...etc. La plupart des widgets de Swing sont des instances de sous-classes de la classe **JComponent**.

### 3.2. L'environnement de développement Eclipse

Eclipse est un environnement de développement intégré ( Integrated Development Environment ) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques. I.B.M. est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le cœur de son outil Websphere Studio Workbench ( WSW ), lui même à la base de la famille des derniers outils de développement en Java d'I.B.M. Tout le code d'Eclipse a été donné à la communauté par I.B.M afin de poursuivre son

développement. Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse. Les principaux modules fournis en standard avec Eclipse concernent Java.



**Figure III.4.** L'interface de l'Eclipse

Eclipse se compose de beaucoup bibliothèques parmi eux WindowBuilder est composé de SWT Designer et Swing Designer et facilite la création d'applications Java GUI sans passer beaucoup de temps à écrire du code. Utilisez le concepteur visuel WYSIWYG et les outils de mise en page pour créer des formulaires simples pour des fenêtres complexes; le code Java sera généré pour vous. Ajoutez facilement des contrôles par glisser-déposer, ajoutez des gestionnaires d'événements à vos contrôles, modifiez diverses propriétés des contrôles à l'aide d'un éditeur de propriétés, internationalisez votre application et bien plus encore.

WindowBuilder est conçu comme un plug-in pour Eclipse et les différents IDE basés sur Eclipse (RAD, RSA, MyEclipse, JBuilder, etc.). Le plug-in crée une arborescence de syntaxe abstraite (AST) pour naviguer dans le code source et utilise GEF pour afficher et gérer la présentation visuelle.

### 3.3. Schéma général de l'application

Le schéma général de l'application est illustré dans la figure suivante :

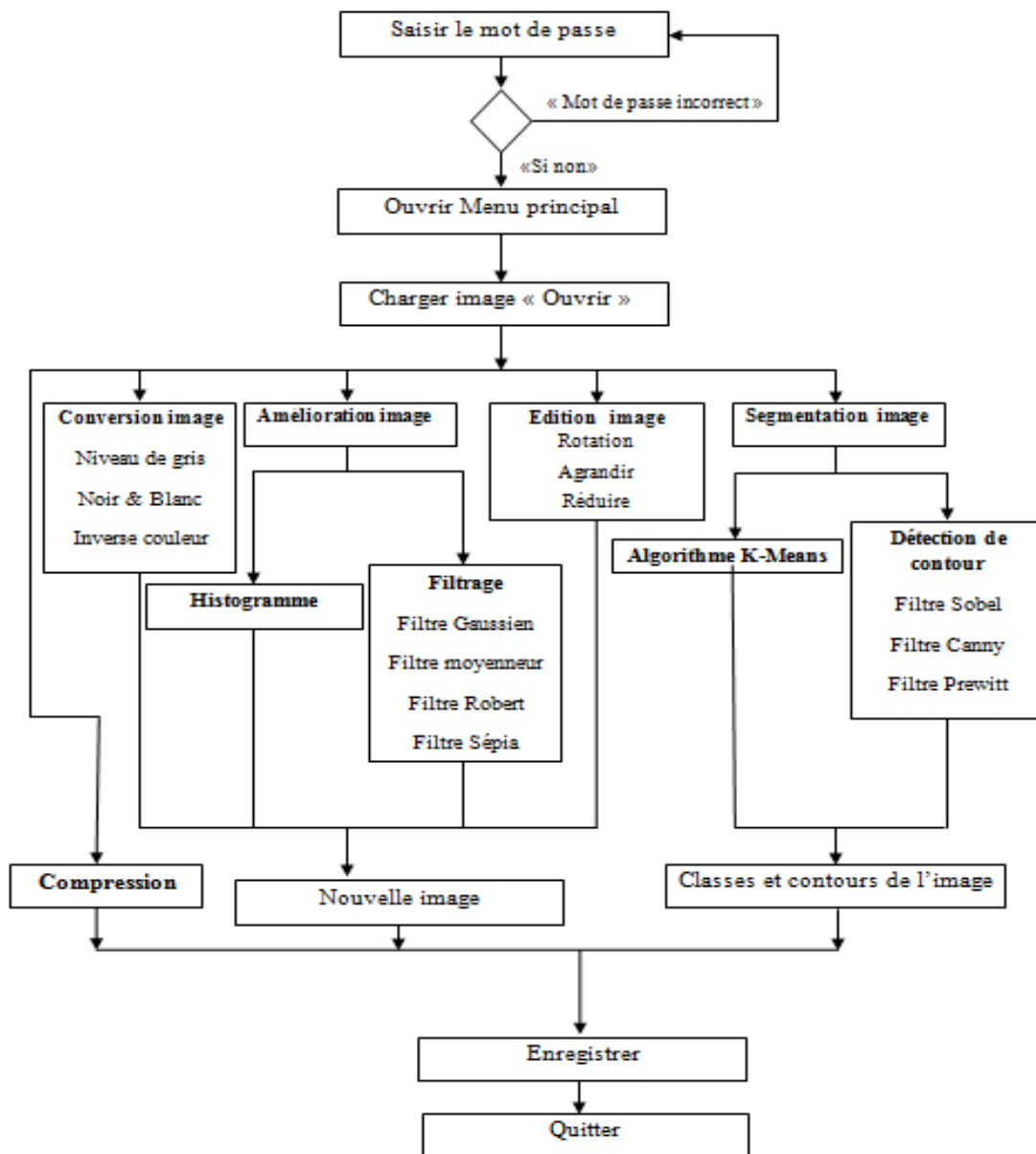


Figure III.5. Schéma général de l'application

## 4. Présentation de l'application

L'étude conceptuelle nous a permis de présenter les traitements d'images pour lesquelles nous avons opté et de fonder des idées très claires sur la manière de les structurer dans une interface graphique, toute en facilitant la tâche à l'utilisateur de l'application.

L'utilisateur de l'application doit s'authentifier pour accéder à son menu principal, il peut charger (ouvrir) une image, effectuer une rotation, la grandir ou la réduire, il peut aussi convertir l'image (en niveau de gris, en noir et blanc et en négatif). La commande amélioration image lui permet d'effectuer des améliorations en sélectionnant un filtre au choix. Le menu propose aussi un ensemble de filtres qui permettent de réaliser la segmentation de l'image par détection de contours, une compression ou décompression de n'importe quelle image.

### 4.1. Interfaces et fonctionnalités

Voici quelques fenêtres de l'application proposée pour le traitement d'images numériques.

#### a) La Fenêtre de mot passe

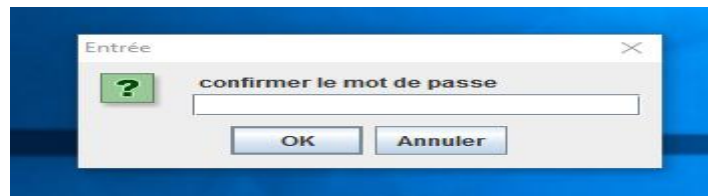
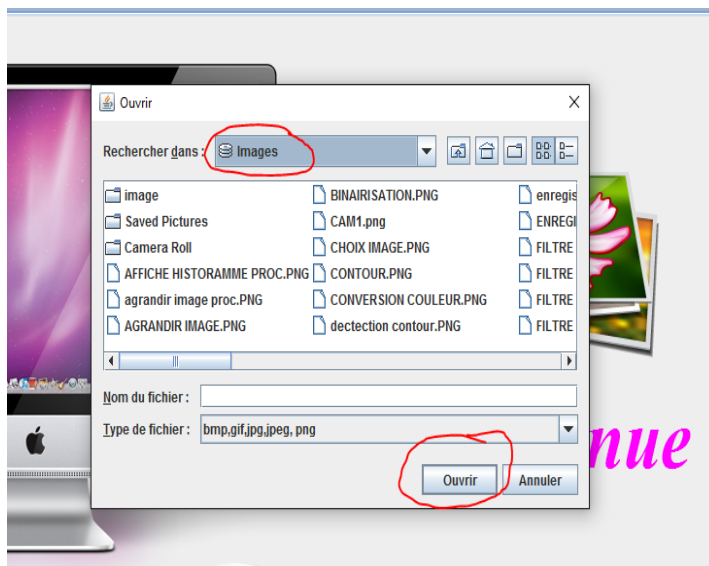
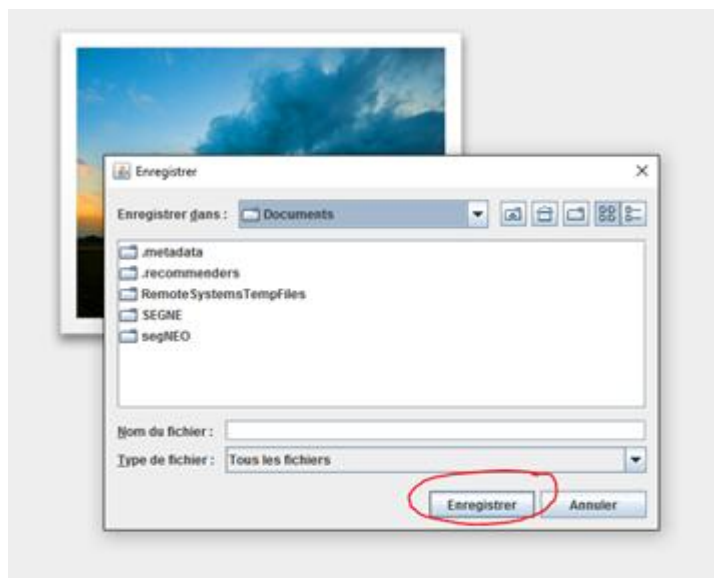


Figure III.6. Fenêtre Saisie le mot de passe

#### b) La page d'accueil de l'application



Figure III.7. La page d'accueil de l'application

**c) Ouvrir (Charger) une image****Figure III.8.** Ouvrir une image numérique**d) Enregistrer Image****Figure III.9.** Enregistrer une image numérique

e) Conversion Image

❖ Image au niveau de gris



Image converti en niveau de gris

Image originale

Figure III.10. Exemple de la Conversion Image en Niveau de gris

❖ Conversion Image en noir et blanc



Image converti en Noir & Blanc

Image originale

Figure III.11. Exemple de la Conversion Image en Noir & Blanc

❖ Inverse couleur



Inverse couleur

Image originale

Figure III.12. Exemple d'un inverse couleur d'image

f) Amélioration d'imgé par application Filtre

❖ Filtre Sepia



Image originale

Image avec filtre Sépia (b)

Figure III.13. Exemple de l'application d'un filtre Sépia

## g) Segmentation d'image

## ❖ Filtre Sobel



Figure III.14. Exemple de l'application d'un filtre Sobel

## h) Histogramme d'image

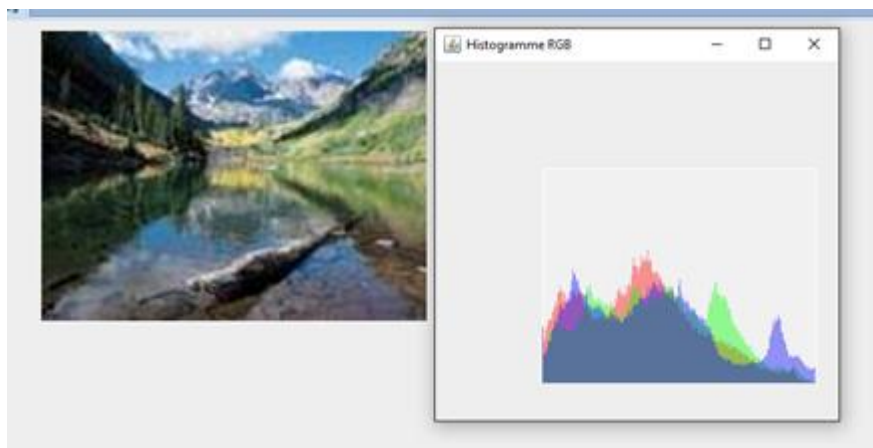


Figure III.15. L'affichage de l'histogramme d'une image

i) Rotation d'une image

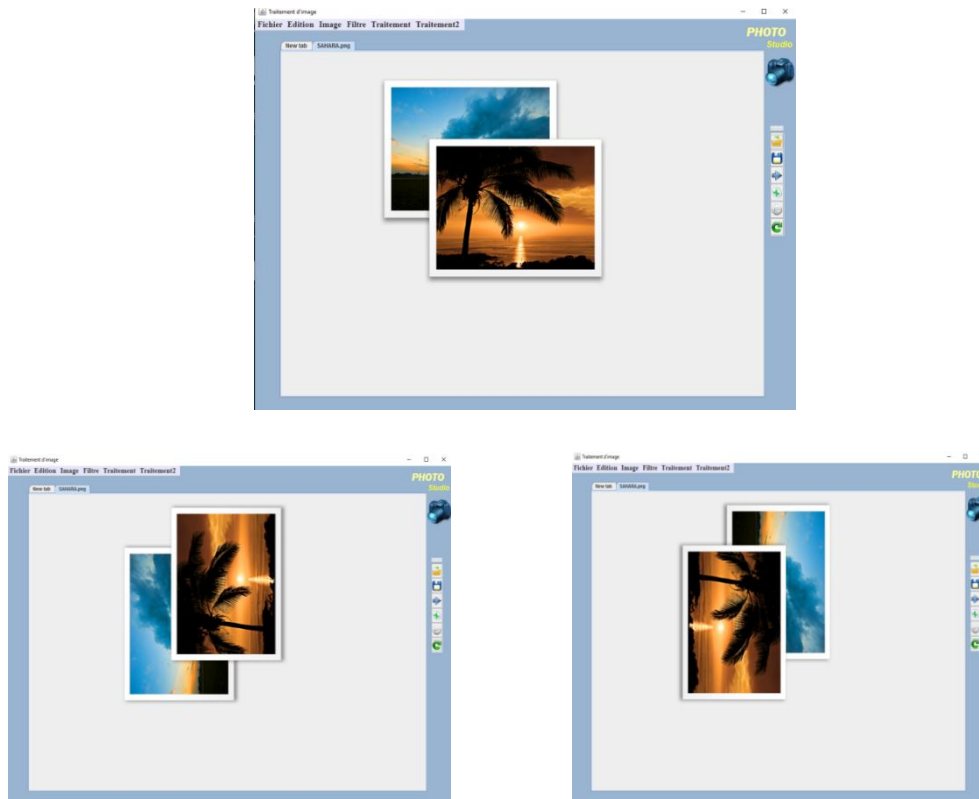


Figure III.16. Rotation à gauche et à droite

j) Compression

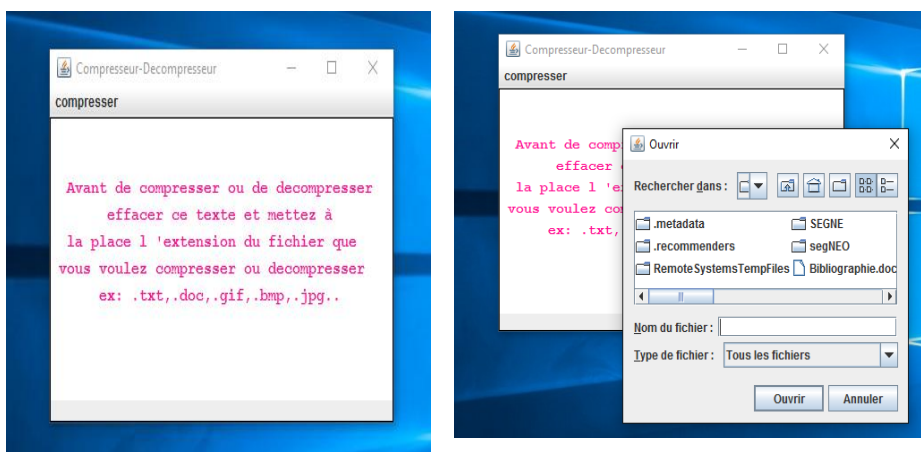


Figure III.17. Compression

**k) Barre accès rapide****Figure III.18.** Barre d'accès rapide**4.2. Scripts de quelques commandes****a) La commande « Ouvrir »**

Fait un appelle à la fenêtre de recherche qui ne permet de choisir un fichier de types image et afficher ce dernier sur la fenêtre principal.

```
public void loadImage(String name) {  
    BufferedImage imageZoomer = new BufferedImage((int)(monImage.getWidth()*1.5), (int)( monImage.getHeight()*1.5), monImage.getType(  
    Image loadedImage = Toolkit.getDefaultToolkit().getImage(name);  
  
    MediaTracker tracker = new MediaTracker(this);  
    tracker.addImage(loadedImage, 0);  
    try {  
        tracker.waitForID(0);  
    }  
    catch (InterruptedException e) {  
        JOptionPane.showMessageDialog(this, "Erreur:\n" + e.getMessage(), "monImage Processing :: Erreur", JOptionPane.ERROR_MESSAGE  
    }  
    monImage = new BufferedImage(loadedImage.getWidth(null),  
        loadedImage.getHeight(null),  
        BufferedImage.TYPE_INT_RGB);  
    Graphics2D g2 = monImage.createGraphics();  
    g2.drawImage(loadedImage, 0, 0, null);  
    repaint();  
}
```

**b) La commande « Sauvegarder »**

```
protected void enregistrerImage(File fichierImage)
{
    String format ="JPG";
    BufferedImage image = getImagePanneau();
    try {
        ImageIO.write(image, format, fichierImage);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

### c) Rotation

```
protected void reduireImage()
{
    BufferedImage imageReduite = new BufferedImage((int) (monImage.getWidth()*0.5), (int) ( monImage.getHeight()*0.5), monImage.getType())
    AffineTransform reduire = AffineTransform.getScaleInstance(0.5, 0.5);
    int interpolation = AffineTransformOp.TYPE_BICUBIC;
    AffineTransformOp reduireImage = new AffineTransformOp(reduire, interpolation);
    reduireImage.filter(monImage, imageReduite );
    monImage = imageReduite ;
    repaint();
}
```

### d) Agrandir l'image

```
protected void agrandirImage()
{
    BufferedImage imageZoomer = new BufferedImage((int) (monImage.getWidth()*1.5), (int) ( monImage.getHeight()*1.5), monImage.getType());
    AffineTransform agrandir = AffineTransform.getScaleInstance(1.5, 1.5);
    int interpolation = AffineTransformOp.TYPE_BICUBIC;
    AffineTransformOp agrandirImage = new AffineTransformOp(agrandir, interpolation);
    agrandirImage.filter(monImage, imageZoomer );
    monImage = imageZoomer ;
    repaint();
}
```

**e) Réduire l'image**

```
protected void reduireImage()
{
    BufferedImage imageReduite = new BufferedImage((int)(monImage.getWidth()*0.5), (int)( monImage.getHeight()*0.5), monImage.getType())
    AffineTransform reduire = AffineTransform.getScaleInstance(0.5, 0.5);
    int interpolation = AffineTransformOp.TYPE_BICUBIC;
    AffineTransformOp retaillerImage = new AffineTransformOp(reduire, interpolation);
    retaillerImage.filter(monImage, imageReduite );
    monImage = imageReduite ;
    repaint();
}
```

**f) Image en Negatif**

```
void negative() {
    byte negative[] = new byte[256];
    for (int i = 0; i < 256; i++)
        negative[i] = (byte) (255 - i);
    ByteLookupTable table = new ByteLookupTable(0, negative);
    LookupOp op = new LookupOp(table, null);
    filter(op);
}
```

**g) Noir et Blanc**

```
protected void imageBinaire()
{
    BufferedImage imgBinaire = new BufferedImage(monImage.getWidth(), monImage.getHeight(), BufferedImage.TYPE_BYTE_BINARY);
    Graphics2D surfaceImg = imgBinaire.createGraphics();
    surfaceImg.drawImage(monImage, null, null);
    monImage = imgBinaire;
    repaint();
}
```

## h) Histogramme

```

public Histogramme(BufferedImage monImage) {
    image = monImage;
    récupérerRVB();
    tracerHistogrammes();
}

protected void paintComponent(Graphics surface) {
    // surface.drawImage(image, 0, 0, null);
    surface.drawImage(histogramme, 100, 100, null);
}

private void récupérerRVB() {
    Raster trame = image.getRaster();
    int[] rgb = new int[3];
    int maximum = 0;
    for (int y=0; y<image.getHeight(); y++)
        for (int x=0; x<image.getWidth(); x++) {
            trame.getPixel(x, y, rgb);
            rouge[rgb[0]]++;
            vert[rgb[1]]++;
            bleu[rgb[2]]++;
        }
}

private void tracerHistogrammes() {
    histogramme = new BufferedImage(largeur, hauteur, BufferedImage.TYPE_INT_ARGB);
    dessin = histogramme.createGraphics();
    Rectangle2D rectangle = new Rectangle2D.Double(0, 0, largeur-1, hauteur-1);
    dessin.draw(rectangle);
    dessin.setPaint(new Color(1F, 1F, 1F, 0.2F));
    dessin.fill(rectangle);
    changerAxes();
    dessin.setPaint(new Color(1F, 0F, 0F, 0.4F));
    tracerHistogramme(rouge);
    dessin.setPaint(new Color(0F, 1F, 0F, 0.4F));
    tracerHistogramme(vert);
    dessin.setPaint(new Color(0F, 0F, 1F, 0.4F));
    tracerHistogramme(bleu);
}
}

```

## 5. Conclusion

A travers ce chapitre j'ai présenté l'application de traitement d'images que développée sous le langage de programmation Java et avec l'environnement de développement Eclipse.

J'ai présenté la conception générale de l'application de traitement d'images que j'ai proposé à travers un diagramme de cas d'utilisation UML et un schéma conceptuel, et une sélection de captures d'écran et quelques codes associés aux principales commandes.

# **Conclusion Générale**

# Conclusion générale

Le traitement et l'analyse d'images occupent une place centrale dans de nombreux domaines industriels et scientifiques, tels que les sciences des matériaux, la géographie, la robotique et la médecine. Ces techniques permettent d'extraire des informations essentielles à partir de données visuelles, ouvrant ainsi la voie à des applications innovantes.

Dans ce contexte, mon projet de fin d'études a porté sur le développement d'une application dédiée au traitement d'images numériques. Cette application intègre plusieurs techniques fondamentales, notamment :

- **L'amélioration d'images** (correction de contraste, réduction du bruit),
- **La conversion et la rotation d'images**,
- **La compression** (optimisation de la taille des fichiers),
- **La segmentation** (détection des régions d'intérêt).

Ce travail m'a permis d'explorer un domaine vaste et complexe, riche en concepts et en méthodes. Cependant, en raison des contraintes temporelles, certaines techniques avancées n'ont pu être approfondies.

## **Perspectives futures :**

Afin d'enrichir cette application, il serait pertinent d'implémenter des algorithmes de segmentation plus sophistiqués (par approche morphologique, par contours actifs, ou par apprentissage profond) et d'en évaluer les performances comparatives. Une intégration de l'intelligence artificielle pourrait également constituer une piste d'amélioration prometteuse.

## Bibliographie

1. Boulekrinet N., 2017. Méthodes floues en segmentation d'images médicales. Mémoire de Master 2 : Informatique, Systèmes informatiques. Université 20 Août 1955 Skikda 1,57.
2. Ziani L., 2017. Compression des images numériques par la technique des ondelettes (JPEG2000). Mémoire de Master 2 : Génie Electrique, Télécommunications. Université Akli Mohand Oulhadj–BOUIRA 1,33
3. Lamraoui D., Slimani H., 2017. Filtrage des images par différentes approches. Mémoire de Master 2 : Génie Electrique, Signal et Communications. Université M'hamed Bougara-BOUMERDES. 1,64
4. Refice I., 2012. Critères d'évaluation pour les méthodes de segmentation d'images. Mémoire de Master 2 : Génie Electrique, Système télécommunication numérique. Université de M'sila. 1,41

### Sites web

1. [https://ensiwiki.ensimag.fr/index.php?title=Fichier:Rapport\\_benhamadi.pdf](https://ensiwiki.ensimag.fr/index.php?title=Fichier:Rapport_benhamadi.pdf)
2. <http://dSPACE.univ-tlemcen.dz/bitstream/112/12907/1/Ms.Tel.MOUSSI%2BNEDJA.pdf>
3. <http://dSPACE.univ-tlemcen.dz/bitstream/112/5813/1/Developpement-dune-application-de-traitement-dimages.pdf>
4. <http://thesis.univ-biskra.dz/2271/6/Chapitre%2003.pdf>
5. <http://biblio.univ-annaba.dz/wp-content/uploads/2014/05/memoire-final.pdf>
5. <http://www.i3s.unice.fr/~mh/RR/2004/RR-04.05-D.LINGRAND.pdf>
7. [http://raphael.isdant.free.fr/traitement\\_numerique/2traitement\\_numerique\\_de\\_l'image.pdf](http://raphael.isdant.free.fr/traitement_numerique/2traitement_numerique_de_l'image.pdf)
8. <http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm>
9. <http://www-sop.inria.fr/members/Pierre.Kornprobst/journal/2006-aubert-vuibert-encyclopedia.pdf>
10. <http://bib.univ-oeb.dz:8080/jspui/bitstream/123456789/6896/1/m%C3%A9moire-segmentation-dimage.pdf>
11. <http://dSPACE.univ-tlemcen.dz/bitstream/112/6831/1/Segmentation-des-Images%20.pdf>
12. [http://glotin.univ-tln.fr/MCBIR/Segmentation\\_images\\_principes.pdf](http://glotin.univ-tln.fr/MCBIR/Segmentation_images_principes.pdf)