

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERRI, Tizi-Ouzou**



Faculté de Génie Electrique et d'Informatique  
Département d'Automatique

## MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme

*MASTER ACADEMIQUE EN AUTOMATIQUE*  
*OPTION : COMMANDE DES SYSTEMES*

# Thème

**Filtrage local et non local des images**

Proposé par : M<sup>me</sup> Sadia ALKAMA

Présenté par :

Kahina SAIDI

Samira RAHOUI

Dirigé par : M<sup>me</sup> Sadia ALKAMA

Soutenu le : 03 /07 /2013

*Promotion 2013*

# Remerciement

*Avant tout développement de ce travail, il apparait opportun de commencer par des remerciements.*

*Nous tenons à remercier Allah le Tout-Puissant de nous avoir donné le courage, la volonté et la patience de mener à terme ce présent travail.*

*Nous tenons tout particulièrement à remercier Mme ALKAMA Fatiha, notre promotrice qui nous a formés et accompagner tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie, pour sa disponibilité permanente, ses discussions fructueuses, mais aussi pour l'aide qu'elle nous a offert pendant la réalisation de ce travail.*

*Tous nos remerciements vont aussi aux membres de jury ayant accepté d'analyser et de corriger notre modeste travail. Nos reconnaissances vont également à l'ensemble des enseignants du département d'automatique qui nous ont énormément transmis tout au long de notre cursus.*



*Dédicace*

*Je dédie ce travail :*

*A mes très chers parents et ma grande mère qui sont toujours mon soutien et ma force. Le joyau de ma vie, que dieu me les garde.*

*A mon cher frère et à ma chère sœur pour leurs aides et leurs précieux conseils (Hakim et Tassadit).*

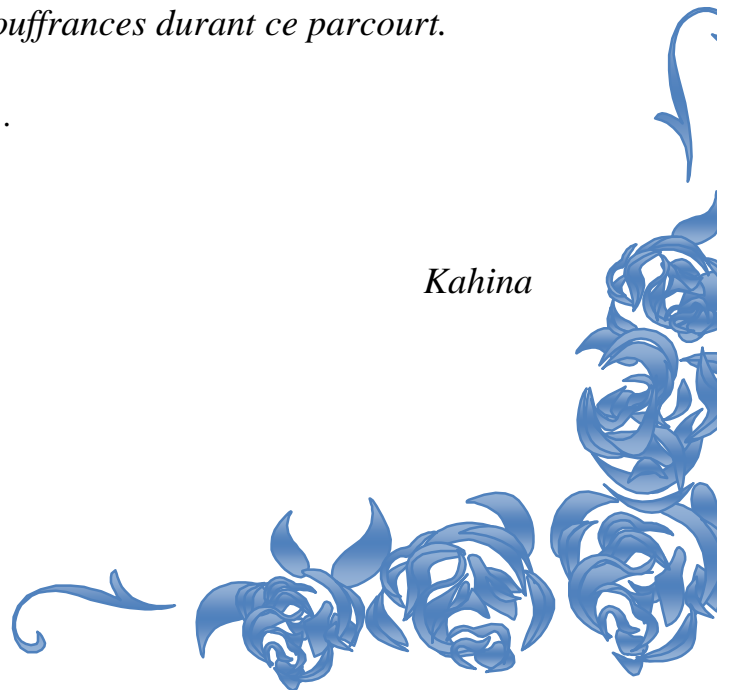
*A mes amies, pour leur amitiés sincères (Djedjiga, Rima, Samira, Sara, Nadia.....).*

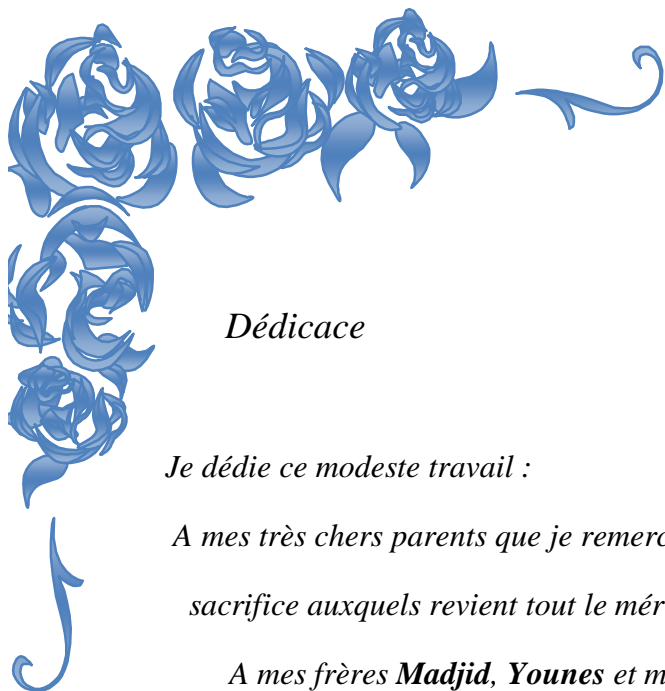
*A tout mes proches paternels et maternels.*

*A tous les collègues de notre promotion, à tous les gents qui ont partagés mes moments de bonheur et de souffrances durant ce parcours.*

*A toute personne qui m'apprecie.....*

*Kahina*





*Dédicace*

*Je dédie ce modeste travail :*

*A mes très chers parents que je remercie beaucoup pour leur patience et sacrifice auxquels revient tout le mérite,*

*A mes frères **Madjid, Younes** et ma grand-mère,*

*A mes oncles et tentes cousins et cousine,*

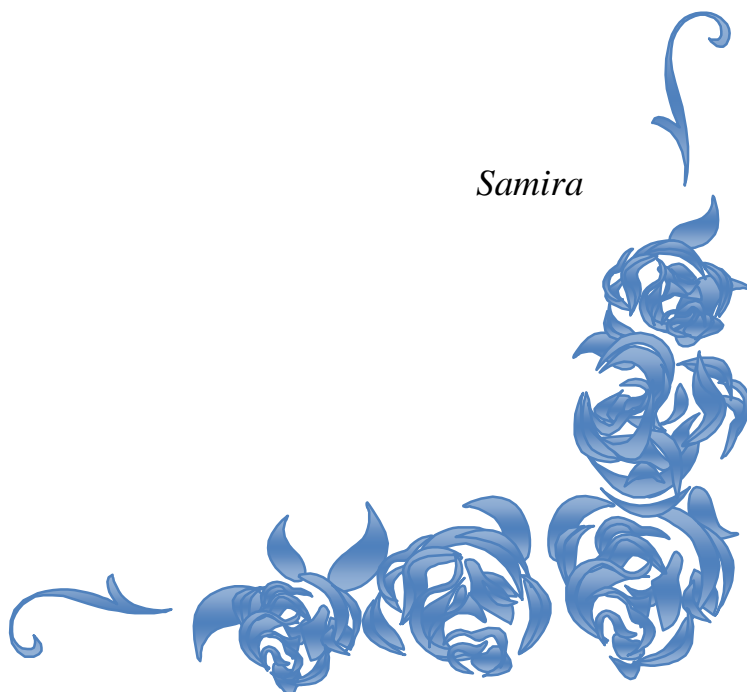
*A **Aghiles Boudedja** et toute sa famille.*

*A ma meilleure amie **Nadjet**,*

*A ma binôme **kahina** et sa famille.*

*A tous mes amis, A toute la promotion automatique 2012/2013.*

*Samira*



# ***Sommaire***

## *Sommaire*

Introduction générale.....	1
<b><i>Chapitre I : Théorie des graphes</i></b>	
I.1 introduction.....	2
I.2 Définition d'un graphe .....	3
I.3 Graphe orienté .....	3
I.4 Graphe non orienté.....	4
I.5 Relation entre graphe orienté et graphe non orienté .....	4
I.6 Sous-graphes.....	5
I.7 Degré d'un sommet .....	5
I.8 Notions de chemin, chaîne, cycle et circuit .....	5
I.9. Distance entre deux sommets .....	6
I.10. Notions de connexité.....	7
I.11. Représentation des graphes.....	8
I.11.1 Matrice d'adjacence.....	9
I.11.2 Matrice associée .....	9
I.11.3 Matrice incidence aux arcs .....	9
I.12 Graphes particuliers.....	10
I.12.1 Graphes complets.....	10
I.12.2 Graphes planaires.....	10
I.12.3 Graphes bipartites .....	11
I.12.4 Graphe eulérien.....	11
I.12.5 Graphe hamiltonien.....	12
I.12.6 Arbres .....	12

I.12.7 Graphe de $\varepsilon$ -voisinage .....	13
I.12.8 Graphe des k-plus proches voisins.....	13
I.13 Conclusion .....	14

## ***Chapitre II : imagerie et filtrage***

II.1 Introduction .....	15
II.2 Définition de l'image et de l'image numérique.....	15
II.3 La Texture .....	15
II.3.1 Types de texture .....	15
II.3.1.1 Les textures structurales .....	15
II.3.1.2 Les textures aléatoires .....	16
II.3.1.3 Les textures directionnelles .....	16
II.3.2 Analyse de la texture .....	17
II.4 Filtrage .....	17
II.4.1 Filtrage global .....	17
II.4.2 Filtrage local.....	18
II.4.2.1 Filtrage local linéaire.....	18
II.4.2.1.1 Filtre passe-bas (lissage).....	19
II.4.2.1.2 Filtre passe-haut .....	20
II.4.2.2 Filtrage non linéaire.....	22
II.4.2.2.1 Filtre médian.....	22
II.4.2.2.2 Filtre de Nagao.....	23
II.4.2.2.3 Filtres morphologiques.....	24
a) Erosion .....	24
b) Dilatation.....	25
c) Ouverture.....	25
d) Fermeture .....	26
II.4.3 Filtrage non local.....	26
II.5 Conclusion.....	27

## ***Chapitre III : Tests et résultats***

III.1 Introduction.....	28
III.2 Modèle de filtres réalisés .....	28
III.2.1 Filtre local.....	28
III.2.2 Filtre non local .....	29
III.2.3 filtre non local avec patch.....	30
III.3 Méthodes de test et d'évaluations adoptées .....	31
III.4 Filtre moyennneur .....	32
III.4.1 Filtre local.....	32
III.4.2 Filtre non local .....	32
III.4.3 Filtre non local avec patch.....	33
III.4.4 Tests avec bruits.....	35
III.5 Filtre médian .....	36
III.5.1 Filtre local.....	36
III.5.2 Filtre non local .....	37
III.5.3 Filtre non local avec patch.....	38
III.5.4 Tests avec bruits.....	39
III.6 Erosion .....	40
III.6.1 Erosion local .....	40
III.6.2 Erosion non local.....	41
III.6.3 Erosion non local avec patch .....	41
III.6.4 Testes avec bruits .....	42
III.7 Dilatation.....	44
III.7.1 Dilatation local.....	44
III.7.2 Dilatation non local .....	44
III.7.3 Dilatation non local avec patch.....	45
III.7.4 Testes avec bruits .....	45
III.8 Conclusion .....	47
Conclusion générale .....	48

# ***Introduction générale***

Depuis les premiers travaux dérivés du traitement du signal, le traitement et l'analyse des images numériques est devenu un domaine de recherche en pleine expansion et dont l'intérêt ne s'est jamais démenti. Le traitement d'image est généralement synonyme d'amélioration des images avec pour but d'extraire des informations souvent noyées dans du bruit ou abîmées par les défauts optiques des instruments. En effet, l'interprétation automatique des scènes bruitées est un défi très important pour les chercheurs en traitement d'images. Pour réduire les bruits et même dans le but de les éliminer, plusieurs types et modèles de filtres ont été développés. Les images texturées sont souvent difficile à filtrer ou à débruiter car dans les filtres classiques la texture n'est pas préservée. Pour cela et pour d'autres raisons de nouveaux modèles de filtres dit non locaux ont été créés.

C'est dans ce cadre que s'inscrit notre travail de fin d'étude. Nous nous sommes en effet intéressés aux différentes approches possibles permettant le filtrage des images.

Nous étudions dans ce travail trois modèles de filtrages à savoir local, non local et non local avec patch. Pour rendre compte de l'efficacité de chaque modèle de filtre, Nous l'appliquons à une image texturée et entachée de différents types de bruits.

Pour ce faire, nous avons structuré notre mémoire en trois chapitres.

Le premier chapitre est consacré à la théorie des graphes qui permettent de modéliser une grande variété de problèmes et que nous utiliserons pour modéliser le filtrage non local.

Dans le deuxième chapitre, nous donnerons quelques notions sur l'imagerie numérique et développerons plus particulièrement le filtrage d'images.

Dans le troisième chapitre, nous expliquons dans le détail la mise en œuvre des modèles de filtrages local, non local et non local avec patch, donnerons les différents résultats et leurs interprétations.

Nous terminons ce mémoire par une conclusion générale.

# ***Chapitre I***

# ***Théorie des graphes***

## I.1 introduction

La théorie des graphes débute avec les travaux d'Euler [01]. L'histoire veut que Léonard Euler en visite dans la ville de Königsberg en Prusse orientale (aujourd'hui elle se nomme Kaliningrad et appartient à la fédération de Russie) ait tenté de répondre à un problème : « est-il possible de trouver un circuit qui emprunte une seule fois chacun des sept ponts disposés comme indiqué sur la figure I.1 ? ».

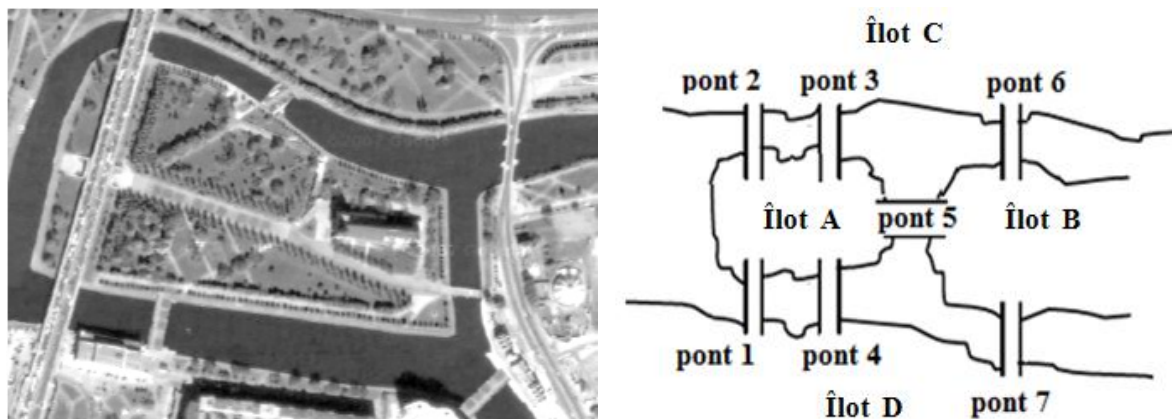


Figure I.1: Les sept ponts de Königsberg

Euler choisit, pour étudier ce problème, de le représenter sous forme de graphe, c'est-à-dire de diagramme où il relie aux quatre lieux possibles les sept ponts de la façon schématisée par la figure I.2.

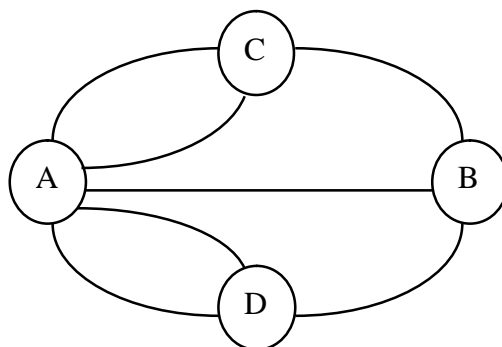


Figure I.2 : Représentation en graphe des sept ponts de Königsberg

Cette modélisation permet de traduire le problème initial en problème autour des propriétés du graphe : « peut-on circuler sur le graphe à partir d'un des points en empruntant une fois et une seule chaque lien? ». Euler démontra que ce problème n'a pas de solution. De la même façon, de nombreuses méthodes, propriétés, procédures ont été imaginées ou trouvées à partir d'un dessin ou d'un schéma. C'est un des principes fondateurs de la théorie des graphes. En

effet, un graphe permet de représenter la structure et les connexions d'un ensemble complexe en exprimant les relations entre ses éléments. Les graphes permettent de modéliser une grande variété de problèmes en se ramenant à l'étude des sommets et de liens.

### I.2 Définition d'un graphe

Le mot « graphe » a été introduit par l'anglais J.J.Sylvester en 1822. Un graphe est un dessin géométrique défini par la donnée d'un ensemble de points (appelées sommets ou nœuds), reliés entre eux par un ensemble de lignes ou de flèches (appelées arêtes ou arcs).

Mathématiquement, un graphe est un couple  $G = (S, A)$  où :

- ◆  $S$  est un ensemble fini d'éléments appelés sommets ou nœuds avec  $S = \{s_1, s_2, \dots, s_n\}$ .
- ◆  $A$  est un ensemble fini de couples de sommets appelés arcs ou arête avec  $A = \{a_1, a_2, \dots, a_m\}$  tel que  $a_i \subset S \times S$ .

En présence d'une arête  $a_k = (s_i, s_j)$  qui peut être notée simplement  $(s_i, s_j)$ , on dit que  $s_i$  et  $s_j$  sont les extrémités de  $a_k$ . Si les deux extrémités d'un arc sont égales, l'arc est une boucle.

Un graphe est sans boucle si  $A$  ne contient pas d'arête de la forme  $(s_i, s_i)$  c'est-à-dire joignant un sommet à lui-même. Le nombre de sommets est appelée ordre du graphe.

### I.3 Graphe orienté

Un graphe orienté, noté  $G = (S, A)$  est un graphe dont les arêtes sont orientées. Ainsi pour une arête  $a_k = (s_i, s_j)$ ,  $s_i$  constitue le début (ou l'origine) de  $a_k$  et  $s_j$  constitue la fin (ou l'extrémité finale) de  $a_k$ . On dit que  $s_j$  est le successeur de  $s_i$  et que  $s_i$  est le prédécesseur de  $s_j$  [02]. Un exemple de graphe orienté est donné sur la Figure I.3.

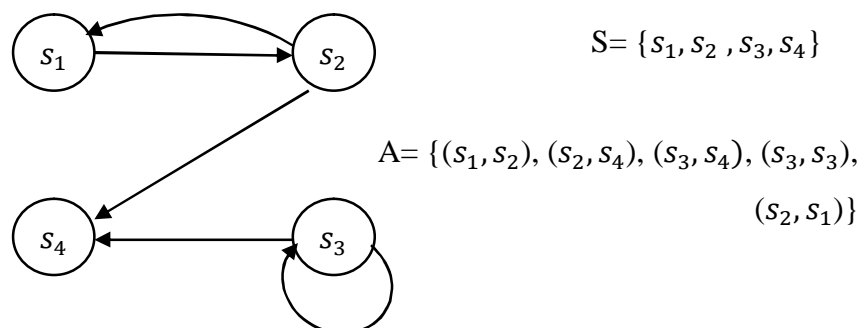


Figure I.3: Exemple d'un graphe orienté

**I.4 Graphe non orienté**

Un graphe non orienté, noté  $G = (S, A)$  est un graphe dont la notion d'ordre des sommets est inexistante. On représente les sommets par des cercles et les arêtes par des courbes simples reliant les deux sommets qui sont ses extrémités. Un exemple de graphe non orienté est donné sur la Figure I.4.

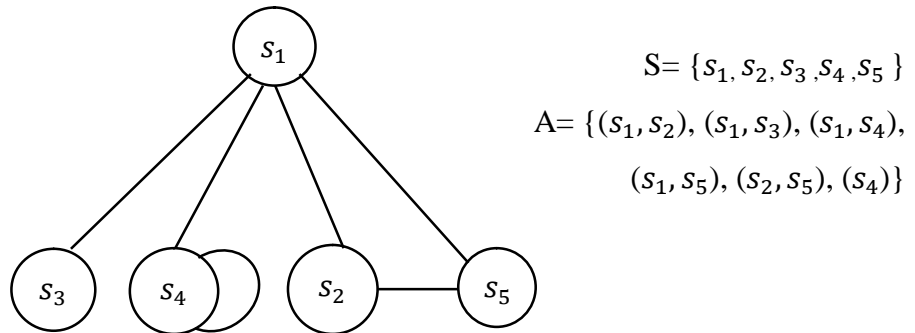


Figure I.4 : Exemple de graphe non orienté

**I.5 Relation entre graphe orienté et graphe non orienté**

A chaque graphe orienté  $G = (S, A)$  on peut associer un graphe non orienté  $G' = (S', A')$ , appelé graphe non orienté associé ou sous-jacent, tel que le graphe sous-jacent  $G'$  possède le même ensemble de sommets ( $S' = S$ ) et l'ensemble des arêtes  $A$  vérifie la relation :

$$(s_i, s_j) \in A' \Leftrightarrow (s_i, s_j) \in A \text{ ou } (s_j, s_i) \in A.$$

Un exemple de graphe orienté et graphe non orienté associé est donné sur la figure I.5.

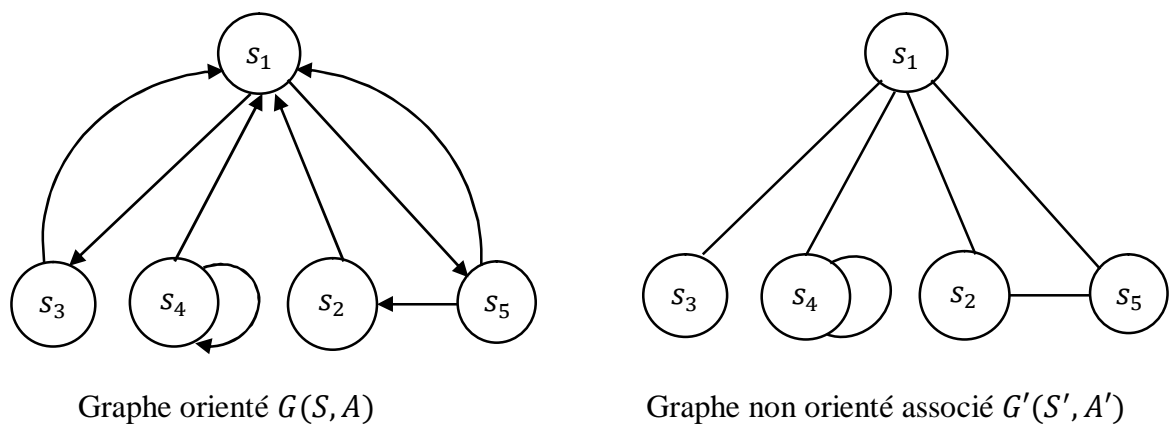


Figure I.5: Exemple de graphes associés

### I.6 Sous-graphes

Soit  $G = (S, A)$  un graphe (orienté ou non). Un sous-graphe de  $G$  est un graphe  $G' = (S', A')$  tel que  $S' \subset S$  et  $A' \subset A$  [03].

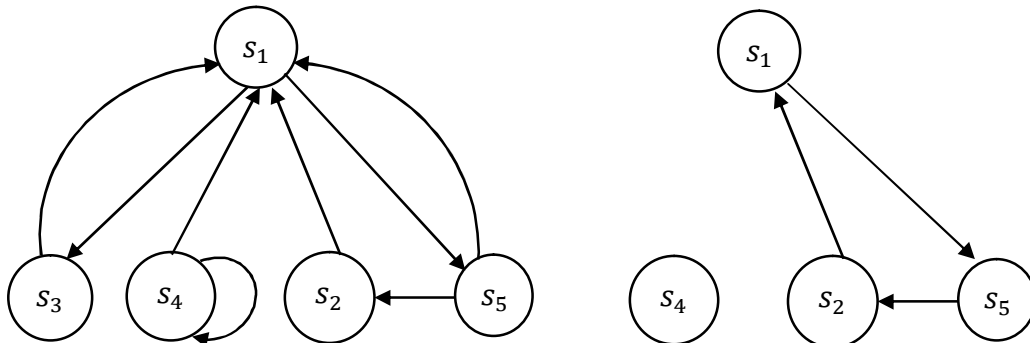


Figure I.6 : Exemple de graphe orienté et de sous graphe

### I.7 Degré d'un sommet

Le degré du sommet  $s_i$ , noté par  $d(s_i)$ , est égal au nombre d'arcs incidents (connecté) au sommet. Un graphe ayant pour chaque sommet le même degré est dit graphe régulier et un sommet ayant le degré égal à 0 est dit sommet isolé [04].

◆ On appelle degré entrant d'un sommet  $s_i$  et on le note  $d_-(s_i)$  le nombre d'arcs dont l'extrémité est  $s_i$ , c'est-à-dire le nombre de prédécesseurs de  $s_i$ .

◆ On appelle degré sortant d'un sommet  $s_i$  et on le note  $d_+(s_i)$ , le nombre d'arcs dont l'origine est  $s_i$ , c'est-à-dire le nombre de successeurs de  $s_i$ .

◆ On appelle degré total d'un sommet  $s_i$  et on le note  $d(s_i)$  le nombre d'arcs dont  $s_i$  est l'origine ou l'extrémité ( $d(s_i) = d_-(s_i) + d_+(s_i)$ ).

Si  $G$  est un graphe non orienté, on définit de la même façon le degré  $d(s_i)$  d'un sommet  $s_i$  comme le nombre d'arêtes ayant  $s_i$  pour extrémité (en cas de boucles on compte deux fois).

### I.8 Notions de chemin, chaîne, cycle et circuit

Dans un graphe orienté, un chemin d'un sommet  $s_i$  vers un sommet  $s_j$  est une séquence  $\langle u_0, u_1, u_2, \dots, u_k \rangle$  de sommets tels que  $s_i = u_0$ ,  $s_j = u_k$ , et  $(u_{l-1}, u_l) \in A$  pour  $l \in \{1, \dots, k\}$ . La longueur du chemin est le nombre d'arcs dans le chemin, c'est-à-dire  $k$ . On dira que le chemin contient les sommets  $u_0, u_1, u_2, \dots, u_k$ , et les arcs  $(u_0, u_1), (u_1, u_2), \dots, (u_{k-1}, u_k)$ .

S'il existe un chemin de  $s_i$  à  $s_j$ , on dira que  $s_j$  est accessible à partir de  $s_i$ . Un chemin est élémentaire si les sommets qu'il contient sont tous distincts [05].

Dans un graphe orienté, un chemin  $\langle u_0, u_1, u_2, \dots, u_k \rangle$  forme un circuit si  $u_0 = u_k$  et si le chemin comporte au moins un arc ( $k \geq 1$ ). Ce circuit est élémentaire si en plus les sommets  $u_0, u_1, u_2, \dots, u_k$  sont tous distincts. Une boucle est un circuit de longueur 1.

Considérons par exemple le graphe orienté de la Figure I.7.

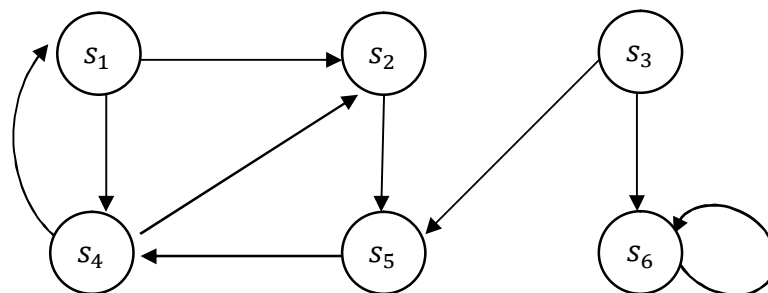


Figure I.7: Exemple de graphe orienté

Un chemin élémentaire dans ce graphe est  $\langle s_1, s_4, s_2, s_5 \rangle$ .

Un chemin non élémentaire dans ce graphe est  $\langle s_3, s_6, s_6, s_6 \rangle$ .

Un circuit élémentaire dans ce graphe est  $\langle s_1, s_2, s_5, s_4, s_1 \rangle$ .

Un circuit non élémentaire dans ce graphe est  $\langle s_1, s_2, s_5, s_4, s_2, s_5, s_4, s_1 \rangle$ .

On retrouve ces différentes notions de cheminement dans les graphes non orientés. Dans ce cas, on parlera de chaîne au lieu de chemin, et de cycle au lieu de circuit. Un graphe sans cycle est dit acyclique.

### I.9 Distance entre deux sommets

Soit  $G = (S, A)$  un graphe orienté. Pour tout  $(s_i, s_j) \in S^2$ , la distance entre le sommet  $s_i$  et le sommet  $s_j$  est le nombre  $d(s_i, s_j)$  défini par :

$$d(s_i, s_j) = \begin{cases} \infty & \text{si il n'existe pas un chemin de } s_i \text{ à } s_j \\ \min I(c) & \end{cases}$$

$c$  : est un chemin de  $s_i$  à  $s_j$ .

$I(c)$  : désigne la longueur du chemin.

La matrice des distances du graphe est alors définie par  $D = (d_{i,j}) = (d(s_i, s_j))$  [06].

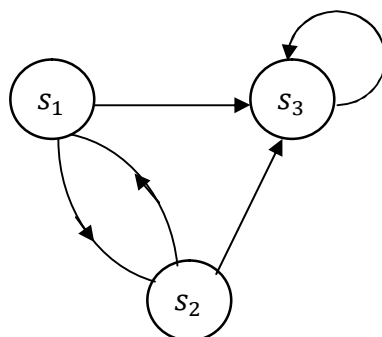


Figure I.8: Exemple de distance entre deux sommets

La matrice des distances du graphe de la figure I.8 est donnée par la matrice  $D$  suivante:

$$D = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ \infty & \infty & 0 \end{pmatrix} \quad \text{I.1}$$

### I.10 Notions de connexité

Un graphe non orienté est connexe si chaque sommet est accessible à partir de n'importe quel autre sommet. Autrement dit, si pour tout couple de sommets distincts  $(s_i, s_j) \in S^2$ , il existe une chaîne entre  $s_i$  et  $s_j$ . Quand un graphe n'est pas connexe, il est possible que certains sous-ensembles de ce graphe le soient. Nous nommons cette partie une composante connexe [05].

Une composante connexe d'un graphe non orienté  $G$  est un sous-graphe  $G'$  de  $G$  qui est connexe et maximal (c.-à-d. qu'aucun autre sous-graphe connexe de  $G$  ne contient  $G'$ ). Par exemple, le graphe non orienté de la figure I.9 n'est pas connexe car il n'existe pas de chaîne entre les sommets  $s_1$  et  $s_5$ . En revanche les deux composantes définies par les sommets  $\{s_1, s_2, s_3, s_4\}$  et les sommets  $\{s_5, s_6, s_7\}$  sont connexes.

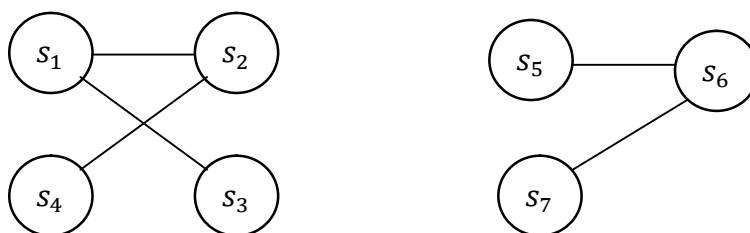


Figure I.9 : Exemple de graphe non connexe avec composantes connexes

On retrouve ces différentes notions de connexités dans les graphes orientés, en remplaçant naturellement la notion de chaîne par celle de chemin : on parle de graphe fortement connexe au lieu de connexe, de composante fortement connexe au lieu de composante connexe.

Plus précisément, un graphe orienté est fortement connexe si chaque sommet est accessible à partir de n'importe quel autre. Autrement dit, si pour tout couple de sommets distincts  $(s_i, s_j) \in S^2$ , il existe un chemin de  $s_i$  vers  $s_j$ , et un chemin de  $s_j$  vers  $s_i$ .

Une composante fortement connexe d'un graphe orienté  $G$  est un sous-graphe  $G'$  de  $G$  qui est fortement connexe et maximal (c'est à dire qu'aucun autre sous-graphe fortement connexe de  $G$  ne contient  $G'$ ). Le graphe orienté de la Figure I.10 n'est pas fortement connexe, car il n'existe aucun chemin entre les sommets  $s_5$  et  $s_3$  alors que les deux composantes définies par les sommets  $\{s_1, s_2, s_3, s_4\}$  et les sommets  $\{s_5, s_6, s_7\}$  sont fortement connexes.

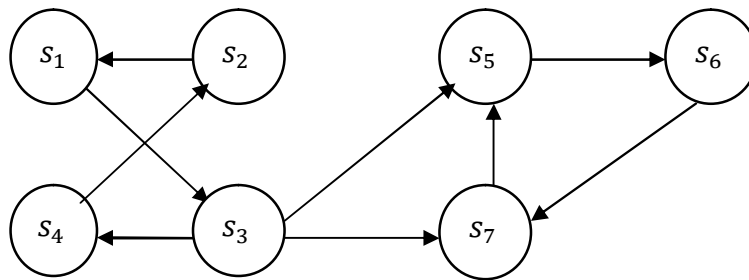


Figure I.10: Exemple de graphe orienté non fortement connexe

### I.11 Représentation des graphes

Certains graphes peuvent être complexes, notamment s'ils contiennent plusieurs arcs ou plusieurs sommets. Un nouveau concept basé sur les matrices est introduit pour palier à certaines difficultés rencontrées dans la représentation des graphes.

A un graphe  $G = (S, A)$  contenant  $n$  sommets et  $m$  arcs, c'est-à-dire:  $\text{card}(S) = |S| = n$  et  $\text{card}(A) = |A| = m$ , on associera trois types de matrices [07]. Pour une meilleure illustration, on définira ces matrices dans le cas du graphe  $G$  de la figure I.11.

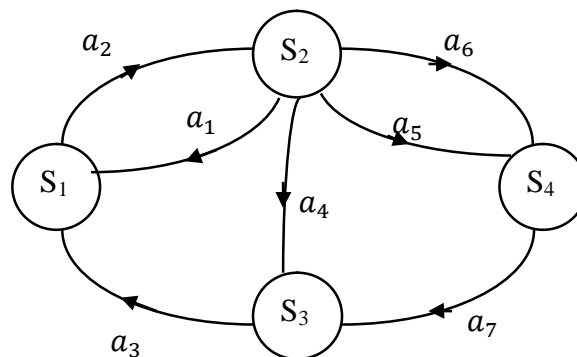


Figure I.11: Exemple de graphe orienté

**I.11.1 Matrice d'adjacence**

La matrice d'adjacence d'un graphe  $G = (S, A)$  est une matrice  $n \times n$  ; ses éléments prennent deux valeurs 1 ou 0. Chaque ligne et chaque colonne représente un sommet du graphe. Ainsi chaque élément de la matrice indique la relation qui existe entre deux sommets : le 1 signifie que les deux sommets sont reliés par un arc, alors que 0 signifie que les deux sommets ne sont pas reliés par un arc. Pour le graphe  $G$  de la figure I.11, la matrice d'adjacence est :

$$M1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{I.2}$$

**I.11.2 Matrice associée**

La matrice associée d'un graphe  $G = (S, A)$  est une matrice  $n \times n$ , où chaque ligne et chaque colonne correspondent à un sommet du graphe. Chaque élément indique le nombre de relations entre deux sommets, c'est-à-dire le nombre d'arcs (arêtes) reliant deux sommets. Pour le graphe  $G$  de la figure I.11, la matrice associée est :

$$M2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{I.3}$$

De la matrice associée, on déduit le demi-degré extérieur de  $s_i$  noté  $d_{G^+}(s_i)$  comme étant la somme des valeurs de la ligne correspondant au sommet  $s_i$  et le demi-degré intérieur de  $s_i$  noté  $d_{G^-}(s_i)$  comme étant la somme des valeurs de la colonne correspondant au sommet  $s_i$ .

**I.11.3 Matrice d'incidence aux arcs**

La matrice d'incidence aux arcs d'un graphe  $G = (S, A)$  est une matrice à  $n$  lignes et  $m$  colonnes. Ses éléments prennent les valeurs 1, 0 ou -1. Chaque ligne de la matrice est associée à un sommet et chaque colonne à un arc. Chaque élément de la matrice indique la relation entre un sommet et un arc comme suit : +1 signifie que le sommet est une extrémité initiale de l'arc, -1 signifie que le sommet est une extrémité terminale de l'arc, et 0 signifie qu'il n'existe pas de relations entre le sommet et l'arc. La matrice d'incidence ne convient pas pour les graphes avec boucles. Pour le graphe  $G$  de la figure I.11, composé de 7 arcs, la matrice d'incidence aux arcs est :



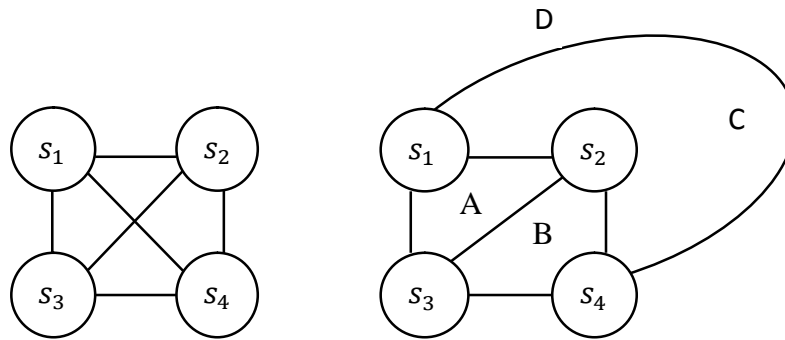


Figure I.13: Exemple du graphe planaire

Un graphe planaire divise le plan en régions internes et en région externe. Dans le graphe de la figure I.13, les régions internes sont A, B et C et la région externe est D.

**I.12.3 Graphes bipartites**

Un graphe bipartite (ou biparti) est un graphe non orienté tel que l'ensemble des sommets peut être partitionné en deux sous-ensembles disjoints  $X$  et  $Y$  tels que chaque arête du graphe à la forme  $\{s_i, s_j\}$ , avec  $s_i \in X$  et  $s_j \in Y$ .

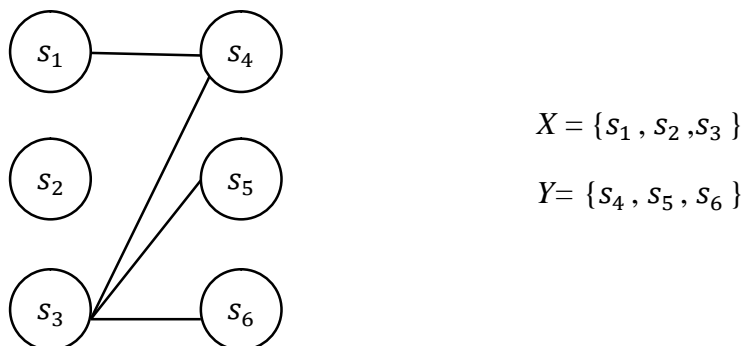


Figure I.14 : Exemple du graphe biparti

**I.12.4 Graphe eulérien**

Un graphe est dit eulérien s'il est possible de trouver un cycle passant une et une seule fois par toutes ses arêtes. Plus simplement, on peut dire qu'un graphe est eulérien s'il est possible de dessiner le graphe sans lever le crayon [04]. Le graphe de la figure I.15 est un graphe eulérien qui donne par exemple le cycle :  $s_1, s_3, s_4, s_2, s_3, s_5, s_2, s_1, s_4$ .

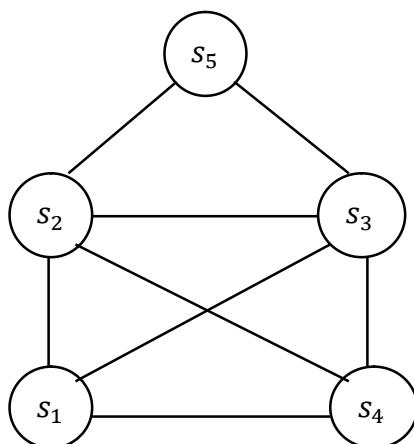


Figure I.15 : Exemple de graphe eulérien

### I.12.5 Graphe hamiltonien

Un graphe est dit hamiltonien s'il est possible de trouver un cycle passant une et une seule fois par tous ses sommets. Le graphe de la figure ci-dessous est un graphe hamiltonien [04].

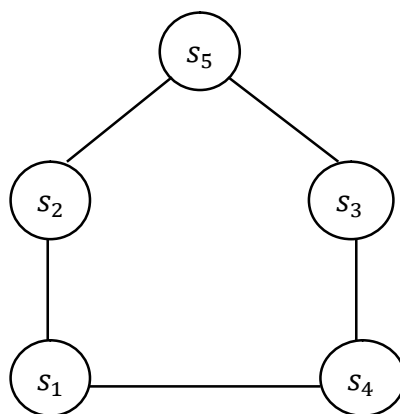


Figure I.16 : Exemple de graphe hamiltonien

### I.12.6 Arbres

Un arbre est un graphe non orienté connexe sans boucle et sans cycle. Entre deux sommets d'un arbre, il y a une arête simple unique [09]. Un arbre avec au moins deux sommets a au moins deux sommets de degré 1.

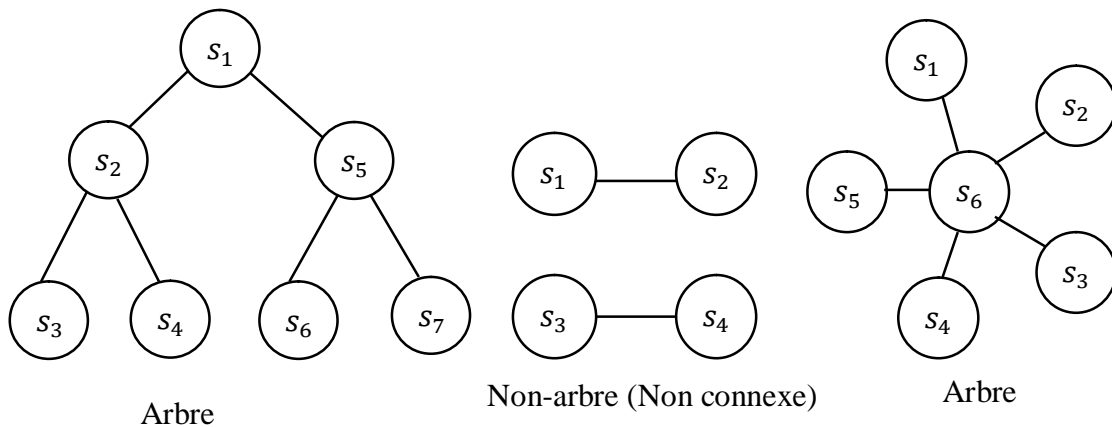


Figure I.17: Exemples d'arbres

**I.12.7 Graphe de  $\epsilon$ -voisinage** (The  $\epsilon$ -neighborhood graph)

Le graphe  $\epsilon$ -voisinage est un graphe dont tous les couples de sommets dont la distance est inférieure à  $\epsilon$  sont reliés. C'est un graphe non orienté [10]. La Figure I.18 montre un exemple d'un graphe de voisinage  $\epsilon$  pour  $\epsilon = 1$  et  $\epsilon = 2$ .

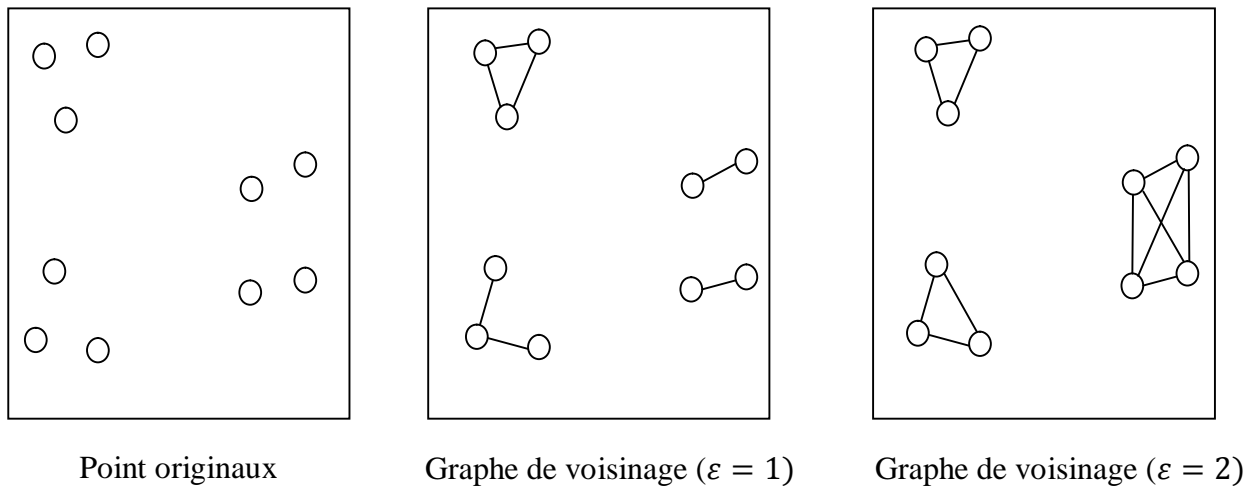


Figure I.18 : Graphe de voisinage  $\epsilon$

**I.12.8 Graphe des k-plus proches voisins** ( $K$ -nearest neighbor graph)

Dans ce type de graphe, un sommet  $s_i$  est lié au sommet  $s_j$  si ce dernier est parmi les  $K$  plus proches voisins (KPPV) du sommet  $s_i$ . Cependant, cette définition conduit à un graphe orienté, et la relation de voisinage n'est pas symétrique. Il existe deux façons de rendre ce graphe non orienté. La première façon consiste tout simplement à ignorer le sens des arêtes, en reliant le sommet  $s_i$  et le sommet  $s_j$  avec une arête non orienté si  $s_i$  est parmi les  $K$  plus

proches voisins de  $s_j$  ou si  $s_j$  est parmi les  $K$  plus proches voisins de  $s_i$ . Le graphe qui résulte est généralement appelé « graphe des  $K$ -plus proches voisins standard ». Le second choix consiste à relier le sommet  $s_i$  avec le sommet  $s_j$  si  $s_i$  est parmi les  $K$  plus proches voisins de  $s_j$  et si  $s_j$  est parmi les  $K$  plus proches voisins de  $s_i$ . Le graphe qui résulte est appelé « graphe des  $K$ -plus proche voisins mutuel ».

La Figure I.19 montre les deux types de graphe des  $K$  plus proches voisins (standard et mutuel) pour  $K=3$  (3 plus proches voisins).

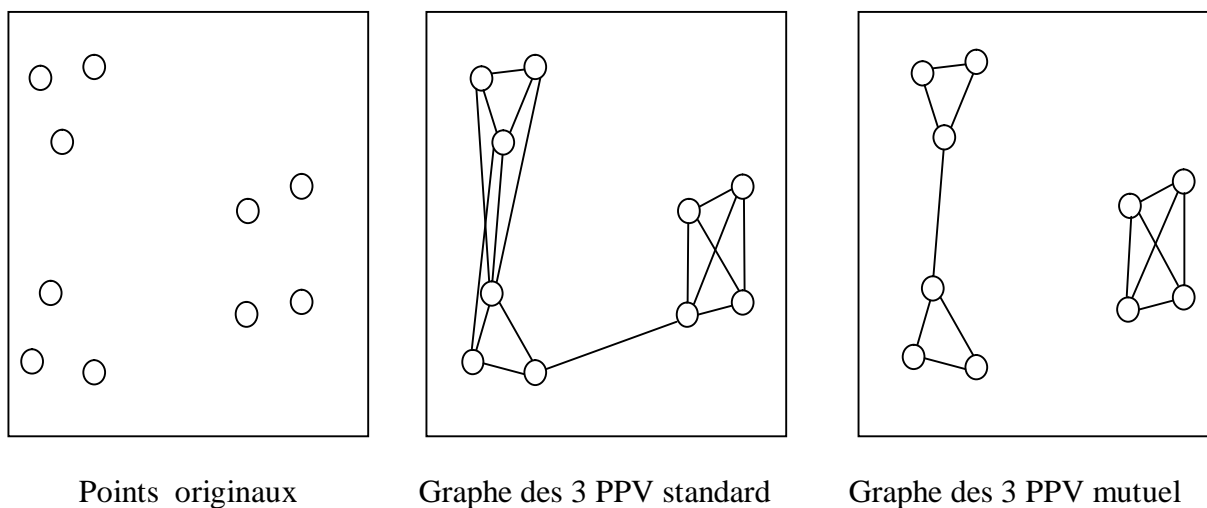


Figure I.19 : Graphes des  $K$  plus proches voisins standard et mutuel

### I.13 Conclusion

Les graphes permettent de modéliser toute situation dans laquelle il y a des interactions entre les objets. On peut donner comme exemples les réseaux routiers, les réseaux de chemin de fer, les réseaux sociaux ou encore le réseau internet. Les techniques utilisées en théorie de graphes permettent de répondre à beaucoup de problèmes algorithmiques posés sur ces réseaux. En effet, étudier les propriétés de ces derniers revient à étudier les propriétés structurelles de leurs topologies représentées par des graphes.

Dans ce chapitre nous avons présenté brièvement quelques notions sur la théorie des graphes, dans les prochains chapitres on verra que la théorie des graphes est une technique couramment utilisée pour résoudre à des problèmes algorithmiques sur le filtrage en traitement d'image.

# ***Chapitre II***

## ***Imagerie et filtrage***

## II.1 Introduction

On désigne par le traitement d'images, l'ensemble des techniques permettant de modifier une image numérique dans le but de l'améliorer ou d'en extraire les informations les plus pertinentes ou tout simplement pour fournir une image plus perceptible à l'œil humain. C'est un domaine très vaste, qui trouve de plus en plus d'applications. Nous donnerons dans ce chapitre quelques notions de traitement d'images, nous décrirons plus particulièrement la notion de texture et développerons la notion de filtrage d'images.

## II.2 Définition de l'image et de l'image numérique

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, ...etc. L'image numérique est définie comme étant une représentation bidimensionnelle d'une scène tridimensionnelle ou à chaque élément de base de cette image ou pixel est associée une grandeur numérique  $I(x,y)$ .  $x$  et  $y$  représentent les coordonnées discrètes du pixel dans le plan et  $I(x,y)$  la grandeur attachée à ce pixel. Bien souvent cette grandeur correspond à l'intensité lumineuse qui est souvent codée sur 8 bits ( $0 < I(x,y) < 255$ ), on parle alors d'images en niveaux de gris. L'image numérique peut aussi être binaire ou chaque pixel est codé sur 1 bit ou couleur dans ce cas le pixel est codé par trois valeurs généralement de 8 bits chacune.

## II.3 La Texture

L'image numérique est décrite par son intensité ou sa couleur, mais elle peut aussi être décrite par d'autres paramètres comme sa texture. Il n'existe pas de définition générique d'une texture, mais on retrouve parfois certaines définitions dont notamment celle qui stipule que la texture est la répétition spatiale d'un même motif dans différentes directions de l'espace. Elle a été décrite comme un phénomène à deux dimensions : la première concerne la description d'éléments de base ou primitives (le motif) à partir desquelles est formée la texture, la deuxième est relative à la description de l'organisation spatiale de ces primitives.

En pratique, nous distinguons trois classes de texture: les textures structurelles, appelées aussi périodiques ou macro-textures, les textures aléatoires et les textures directionnelles.

### II.3.1 Types de texture

#### II.3.1.1 Les textures structurelles

On peut considérer les textures structurelles comme étant la répétition spatiale de motifs

élémentaires de base suivant des règles de direction et de placement de manière ordonnée on parle alors de texture ordonnée. Les méthodes décrivant ce type de texture essaient de découvrir et de caractériser le motif de base (motif générateur). La figure II.1 donne trois exemples de textures structurées.

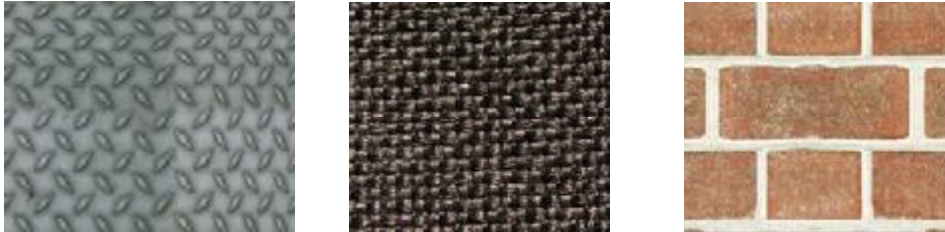


Figure II.1 : Trois exemples de textures structurées

### II.3.1.2 Les textures aléatoires

Contrairement aux textures structurées, les textures aléatoires ne contiennent pas de motif de base et chaque pixel semble avoir été tiré aléatoirement (voir figure II.2). Toutefois ces textures donnent une impression d'homogénéité. Une solution pour analyser ces textures est de procéder à une analyse statistique.

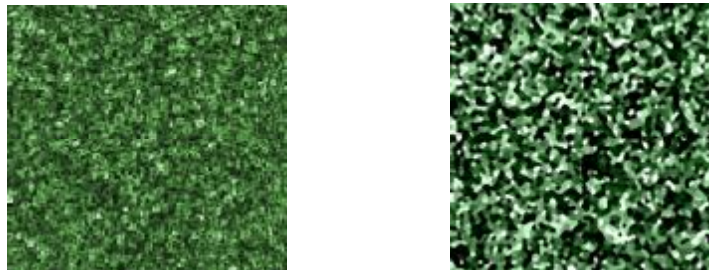


Figure II.2 : exemples de textures aléatoires

### II.3.1.3 Les textures directionnelles

Les textures directionnelles n'utilisent pas de motif de base et ne sont pas pour autant aléatoires. Les intensités des pixels qui les composent forment des motifs organisés selon des directions bien précises (voir figure II.3).

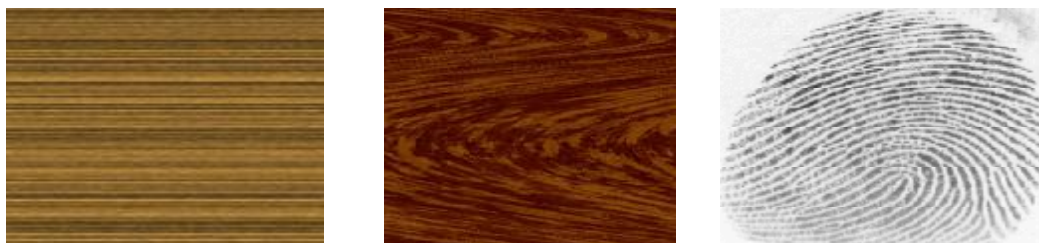


Figure II.3: Exemples de textures directionnelles.

### II.3.2 Analyse de la texture

L'analyse de texture regroupe un ensemble de techniques mathématiques permettant de quantifier les différents niveaux de gris présents dans une image en termes d'intensité et de distribution. Le but de l'analyse de texture est de formaliser les descriptifs de la texture par des paramètres mathématiques qui serviraient à l'identifier, Parmi les méthodes d'analyse de la texture nous pouvons citer les méthodes statiques et les méthodes structurelles.

**Les méthodes statistiques** étudient les relations entre un pixel et ses voisins. Elles sont utilisées pour caractériser des structures fines, sans régularité apparente. L'étude se fait sur des structures tout à fait aléatoires et le plus souvent non homogènes, c'est pourquoi ce type de méthode sera préférentiellement utilisé pour les micro-textures. Dans ce cas, la texture est décrite par les statistiques de la distribution de ces niveaux de gris.

**Les méthodes structurelles** tiennent compte de l'information structurelle et contextuelle d'une forme et sont particulièrement bien adaptées aux textures macroscopiques. Elles permettent de décrire la texture en définissant les primitives et les "règles" d'arrangement qui les relient. En effet, les textures ordonnées possèdent des primitives qui se répètent dans les images en des positions selon une certaine loi.

## II.4 Filtrage

Le filtrage d'image a pour but d'améliorer la qualité d'une image numérique. Il consiste essentiellement soit à atténuer le bruit, soit à accentuer les contours pour mieux extraire les informations voulues. Il existe un grand nombre de filtres où chaque filtre est souvent destiné 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 à corriger et suivant le type et la qualité de l'image d'origine. Les filtres utilisés en traitement d'images peuvent être classés en trois catégories, filtrage global, filtrage local et filtrage non local.

### II.4.1 Filtrage global

Dans le filtrage global, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ. Dans cette catégorie on trouve, par exemple, les opérations sur les histogrammes [12].

Par exemple le filtrage par égalisation d'histogramme consiste à rendre le plus plat possible l'histogramme des niveaux de gris de l'image. Cette méthode permet souvent de renforcer le

contraste sur des détails de l'image qui sont masqués par des variations d'intensité de plus grande amplitude à plus grande échelle (voir figure II.4).



Image original

Image après égalisation d'histogramme

Figure II.4: Exemple d'égalisation d'histogramme

### II.4.2 Filtrage local

Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrice) de taille impaire. Le pixel traité est au centre de ce tableau. Un exemple de voisinage 3x3 est donné ci-dessous :

$$\begin{bmatrix} I_{x-1,y-1} & I_{x,y-1} & I_{x+1,y-1} \\ I_{x-1,y} & I_{x,y} & I_{x+1,y} \\ I_{x-1,y+1} & I_{x,y+1} & I_{x+1,y+1} \end{bmatrix}$$

Dans le cas où le nouveau pixel calculé est une combinaison linéaire des valeurs des pixels du voisinage ce filtre local est dit linéaire sinon il est dit non linéaire.

#### II.4.2.1 Filtrage local linéaire

Le filtre linéaire consiste à calculer les nouveaux pixels de l'image  $I'(x, y)$  en réalisant un produit de convolution de l'image d'origine  $(I(x, y))$  par une matrice « noyau »  $(h(x, y))$ . L'équation décrivant cette convolution est donnée par la relation ci-dessous.

$$I'(x, y) = I(x, y) \otimes h(x, y) = \sum_{m=-d}^d \sum_{n=-d}^d h(m, n) \cdot I(x + m, y + n)$$

La taille du masque de convolution est  $(2d+1) \times (2d+1)$ , avec  $d=1,2,\dots$

Les filtres linéaires les plus utilisés sont les filtres passe bas et les filtres passe haut.

**II.4.2.1.1 Filtre passe-bas (lissage)**

Ce type de filtre atténue les composantes de hautes fréquences telles que les bruits et les irrégularités de l’image. Cette opération de filtrage peut être répétée plusieurs fois, ce qui crée un effet de flou. En pratique il faut choisir un compromis entre l’atténuation du bruit et la conservation des détails et des contours significatifs. La fonction de voisinage dans ce type de filtre est définie généralement par :

$$H = \frac{1}{N} \sum_{i,j} w_{ij}$$

où  $\frac{1}{N}$  est le facteur de normalisation

◆ Dans le cas où  $N=9$ , il est dit Filtre moyeneur. Il consiste à remplacer chaque pixel par la moyenne des valeurs des pixels adjacents et du pixel central. Son noyau est donc donné par la matrice suivante :

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Un exemple de résultats de filtrage par le filtre moyeneur est donné sur la figure II.5



Figure II.5: Exemple de filtrage moyeneur

◆ Dans le cas où  $N=16$ , il est dit filtre de Smooth. Son noyau est donc donné par la matrice suivante :

$$H = \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

C’est le filtre passe-bas le plus utilisé, il affecte au pixel central un poids plus grand par rapport à son voisinage. Un exemple de résultat de ce type de filtre est donné sur la figure II.6



Image originale

Smooth 3x3

Smooth 5x5

Figure II.6: Exemple de filtrage par le filtre de Smooth

#### II.4.1.2 Filtre passe-haut

Comme dans le cas des filtres passe-bas on peut attribuer à chaque poids du voisinage d'une fenêtre passe-haut, des poids statistiques différents, mais choisis de telle sorte que leur somme soit toujours égale à zéro. Ce type de filtre n'élimine pas le bruit, bien au contraire, plus qu'une augmentation du contraste de l'image, il permet la mise en évidence de contours entre des plages de niveaux de gris différentes. Un exemple de masque de convolution d'un filtre passe-haut est :

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Le résultat de ce filtrage par le masque précédent est donné sur la figure ci-dessous.



Image Originale

Image filtré

Figure II.7 : Filtrage passe-haut

Il existe plusieurs autres matrices de convolution réalisant le filtrage passe-haut tel que les opérateurs de Robert, de Sobel et de Prewitt.

L'opérateur de Robert est très sensible au micro-contour, mais aussi aux bruits. Il donne des contours dont la qualité est assez bonne. Appliquer l'opérateur de Roberts à une image revient à la convoluer avec les deux masques suivants :

Un exemple d'image filtrée par l'opérateur de Robert est donné sur la figure ci-dessous.

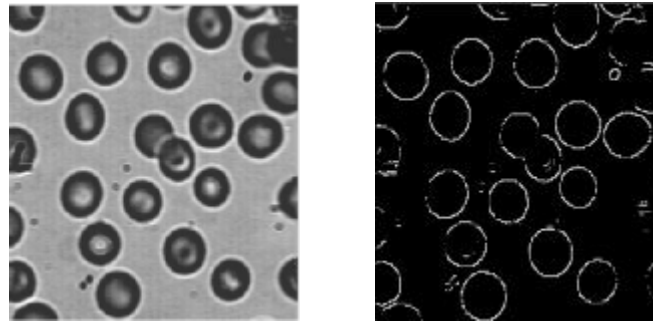


Image original

image filtrée

Figure II.8 : filtrage par l'opérateur Robert

L'opérateur de Robert est très sensible aux bruits, il est utilisé pour la détection des contours correspondant à des changements brusques des propriétés physiques ou géométriques de la scène. Les masques de détection de contour de Sobel détectent séparément les contours verticaux et horizontaux. On obtient après convolution une image des gradients. Cette image pourra par la suite être modifiée dans une phase de seuillage pour ne contenir que des pixels à 1 ou à 0 correspondant respectivement à un point de contour ou à un point de non contour. Les masques de Sobel sont donnés par les matrices ci-dessous.

Un exemple de filtrage par l'opérateur de Sobel est donné sur la figure II..



Image originale

image filtrée et seuillée

Figure II.9 : filtrage par l'opérateur de Sobel

L'opérateur de Prewitt, présente l'avantage d'être moins sensible aux bruits, il favorise les contours à structure linéaire mais la qualité des contours est meilleure.

Les masques de Prewitt sont donnés par les matrices ci-dessous.

Un exemple de filtrage par l'opérateur de Prewitt est donné sur la figure II.10.



Image originale



Image filtrée et seuillée

Figure II.10: Filtrage par l'opérateur de Prewitt

#### II.4.2.2 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 ces filtres sont très souvent de meilleure qualité que ceux obtenus par les filtres linéaires. Parmi les filtres non linéaires, on peut citer le filtre médian, filtre de Nagao, les filtres morphologiques,...

##### II.4.2.2.1 Filtre médian

Pour réaliser le filtre médian il faut déplacer une fenêtre de taille impaire sur le support image et remplacer le pixel central (sur lequel est positionnée la fenêtre) par la valeur médiane des pixels inclus dans la fenêtre (Voir la Figure II.11).

Par exemple, le 0 centrée dans une fenêtre de taille 3x3 dans l'image originale est remplacé par la valeur médiane de la fenêtre qui est de 36.

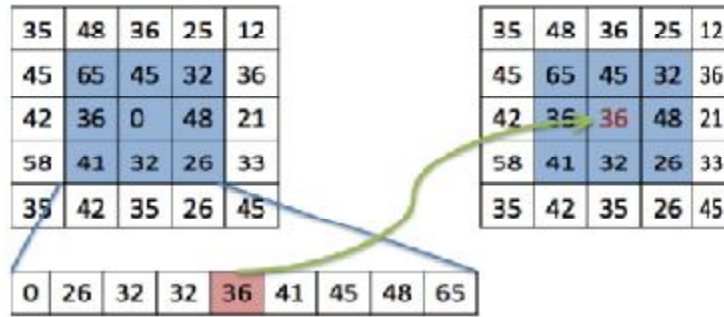


Figure II.11: schéma explicatif du Filtre Médian

Ce filtre est très adapté au bruit type « poivre et sel », il préserve les contours et réduit le bruit additif uniforme ou gaussien (lissage de l'image). Si le bruit est supérieur à la moitié de la taille du filtre, alors le filtre est inefficace. Une illustration des effets du filtre médian est donnée sur la Figure II.12

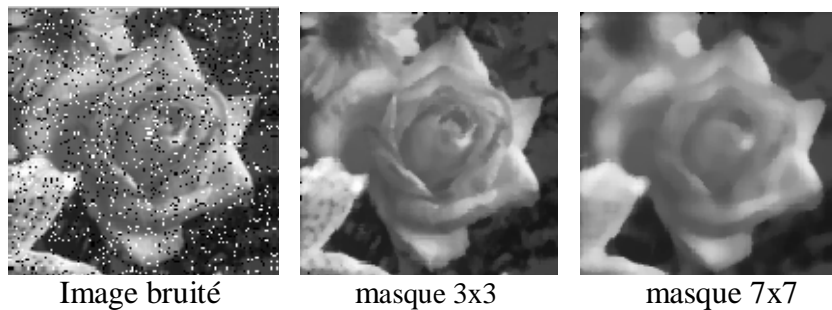


Figure II.12: Résultat du filtre médian

#### II.4.2.2.2 Filtre de Nagao

Pour chaque pixel, on considère 9 sous-fenêtres  $F_k$ ,  $k=1,\dots,9$  comme indiqué sur la figure II.13.



Figure II.13 : les sous fenêtres de Nagao dans un voisinage 5x5

La nouvelle valeur sera la moyenne des niveaux de gris du voisinage le plus homogène c'est-à-dire ayant une variance minimale.

L'image présentée ci-dessous est le résultat d'un filtrage de Nagao avec un voisinage 9x9.

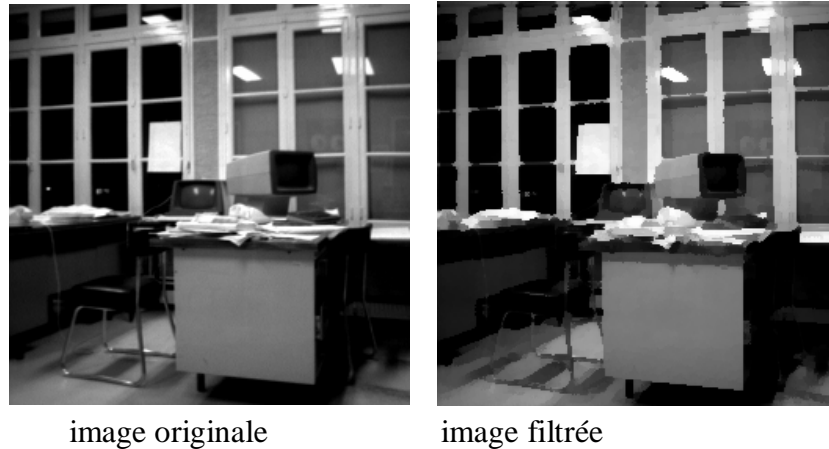


Figure II.14 : Exemple de filtrage par le filtre de Nagao 9x9

#### II.4.2.2.3 Filtres morphologiques

Ce type de filtre repose sur la morphologie mathématique basée sur une description ensembliste des images. Les opérateurs morphologiques privilégient la notion de forme plutôt que l'information sur l'amplitude des signaux. Ce filtrage non linéaire fait appel à deux opérateurs de base, l'érosion et la dilatation et à deux opérateurs complémentaires combinant les deux premières, l'ouverture et la fermeture.

##### a) Erosion

L'érosion d'une image  $I$  par l'élément structurant  $B$  consiste à déplacer cet élément structurant de manière à occuper successivement toutes les positions de l'image et de remplacer le pixel central de cet élément structurant par le pixel présentant la plus faible valeur dans  $B$ . L'élément structurant  $B$  est souvent une fenêtre carrée de taille impaire  $(2d+1) \times (2d+1)$ .

$$I' = I \ominus B = \{I'(x, y) = \inf(B_I(x, y))\}$$

Avec  $I'$  est l'image érodée

$I$  : L'image originale

$B$  : L'élément structurant

$B_I(x, y)$  : L'ensemble des pixels de  $I$  présent dans l'élément structurant  $B$  centré en  $(x, y)$

Un exemple d'érosion par un élément structurant carré de taille  $(5 \times 5)$  est donné sur la figure II.15.

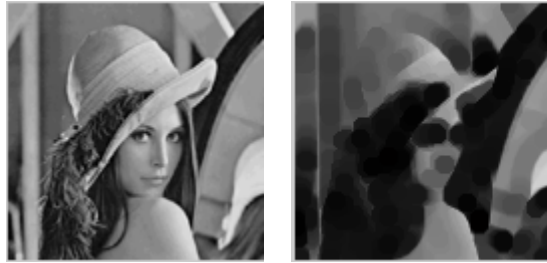


Figure II.15 : Erosion par élément structurant carré de taille 5x5

### b) Dilatation

La dilatation d'une image  $I$  par l'élément structurant  $B$  consiste à déplacer cet élément structurant de manière à occuper successivement tout les positions de l'image et de remplacer le pixel central de cet élément structurant par le pixel présentant la plus élevée valeur dans  $B$ .

L'élément structurant  $B$  est souvent une fenêtre carrée de taille impaire  $(2d+1) \times (2d+1)$

$$I' = I \oplus B = \{I'(x, y) = \sup(B_I(x, y))\}$$

Avec  $I'$  est l'image dilatée

$I$  : L'image originale

$B$  : L'élément structurant

$B_I(x, y)$  : L'ensemble des pixels de  $I$  présent dans l'élément structurant  $B$  centré en  $(x, y)$ .

Un exemple de dilatation par un élément structurant carré de taille  $(5 \times 5)$  est donné sur la Figure II.16.

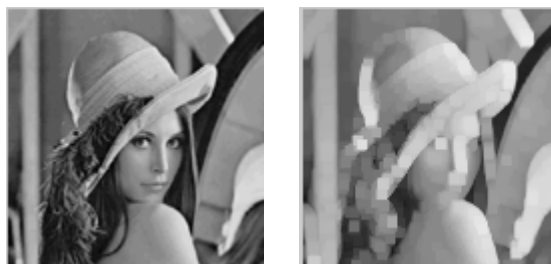


Figure II.16: Dilatation par élément structurant carré de taille 5x5

### c) Ouverture

L'ouverture est l'application de l'opérateur érosion puis de l'opérateur dilatation avec le même élément structurant.  $I'$  qui est l'image obtenue par l'ouverture de  $I$  par  $B$ , est donnée par la relation :

$$I' = I \circ B = (I \ominus B) \oplus B$$

Figure II.17 donne un exemple de l'opération ouverture appliquée sur l'image Lena.



Figure II.17 : Ouverture par élément structurant carré de taille 3x3

#### d) Fermeture

La fermeture est l'opération inverse de l'ouverture. Elle consiste en une dilatation suivie d'une érosion avec le même élément structurant.

$$I' = I \bullet B = (I \oplus B) \ominus B$$

La Figure II.18 donne un exemple de l'opération fermeture appliquée sur l'image Lena.

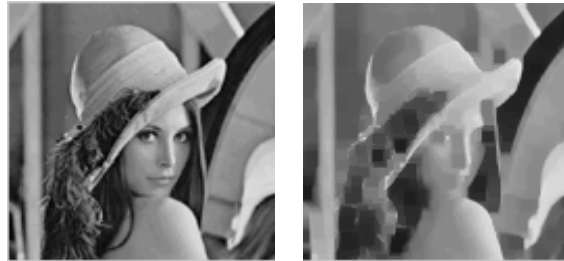


Figure II.18 : Fermeture par élément structurant carré de taille 3x3

### II.4.3 Filtrage non local

Tous les filtres cités précédemment (à part les filtres globaux) combinent les pixels voisins dans l'image c'est pour cela qu'ils sont appelés filtres locaux.

Or il est assez rare, qu'un détail dans une image ne soit pas répété plusieurs fois. Cette remarque très simple n'a commencé à être prise en compte qu'assez récemment pour le filtrage et la restauration des images, elle est à la base de nouveaux filtres dit non locaux.

La méthode consiste à rechercher dans toute l'image ou dans une vaste portion de l'image tous les pixels qui « ressemblent » au pixel que l'on veut traiter et réaliser l'opération souhaité en utilisant ces pixels.

On ne cherche plus les pixels proches au sens spatial, mais des pixels « ressemblant ».

Plusieurs auteurs [14] et [15] ont utilisé cette méthodologie pour le débruitage d'images qui peut être adaptée à différents modèles de bruits. Cette méthode se base principalement sur les filtres à moyennes non locales. Il a été constaté que pour ces filtres, le bruit présent dans

l'image originale est bien atténué dans les régions homogènes et les contours des objets sont bien conservés.

La ressemblance entre pixels qui est initialement déterminée en fonction de la distance entre les niveaux des pixels a été ensuite étendue à une distance entre patches.

Un patch est un carré de taille fixe et centré sur un pixel [16].

La ressemblance entre deux pixels sera alors en fonction de la mesure de la distance euclidienne entre patches.

La figure fournie un exemple de patches de l'image les plus similaires.



Figure III.19 : Exemple de patches de l'image les plus similaires

Le filtrage par les moyennes non locales en utilisant la ressemblance entre patches est donc un algorithme de filtrage utilisant la redondance présente naturellement dans les images [17].

En plus de son utilisation pour le débruitage, le principe de la non localité et de la non localité avec patch est utilisé dans la synthèse de textures [18]et dans l'inpainting [19].

## II.5 Conclusion

Dans ce chapitre nous avons présenté quelques types de filtres classiques dit locaux tels que les filtres passe bas, les filtres passe haut et les filtres morphologies. Nous avons par la suite présenté la méthodologie utilisée pour construire des filtres non locaux et des filtres non locaux avec patches. Dans le chapitre suivant nous développerons ces différents modèles de filtres avant de les appliquer aux images.

# ***Chapitre III***

## ***Tests et résultats***

### III.1 Introduction

Nous allons dans ce chapitre tester plusieurs modèles de filtrage avec différents types de filtres. Nous regrouperons dans le même modèle les différents types de filtres (par exemple les filtres passe-bas, passe-haut, morphologiques, ...) qui traitent les pixels de l'image en considérant le même type de graphe. Nous commencerons alors ce chapitre par définir les trois modèles de filtres que nous testerons. Avant de donner les différents résultats, nous expliquerons la technique que nous utilisons pour leur évaluation.

Nous testerons les différents modèles de filtres sur une image à laquelle nous avons éventuellement ajouté du bruit. Tous les résultats seront bien-sûr interprétés. Les différents programmes sont réalisés sous Matlab7.9 .

### III.2 Modèles de filtres réalisés

Nous présenterons dans ce chapitre les résultats obtenus en utilisant quelques filtres bien connus qui sont : le filtre moyenneur, médian, érosion et dilatation. Ces filtres seront utilisés de trois manières différentes, à savoir local, non local et non local avec patch.

En premier lieux nous donnerons la définition des modèles de filtrage local (L), non local (NL) et non local avec patch (NLP) en utilisant un filtre de type moyenneur. Une extension pour les trois autres types de filtres sera par la suite évidente.

#### III.2.1 Filtre local

Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine.

L'idée est de représenter une image par un graphe  $G = (S, A)$  avec  $S = \{s_1, s_2, \dots, s_n\}$  qui est un ensemble fini de sommets représentant les pixels de l'image et  $A$  est un ensemble fini d'arrêtes. Plusieurs types de graphe peuvent être construits tel que le  $\mathcal{E}$ -voisinage graphe, le k-plus proche voisin graphe ou le graphe complètement connecté. Dans le cas du filtrage local, nous utiliserons le graphe des k-plus proches voisins en reliant chaque pixel de l'image avec les k pixels qui lui sont les plus proches dans l'espace au sens d'une distance euclidienne spatiale.

Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré en considérant donc pour chaque pixel ses  $((2d+1) \times (2d+1) - 1)$  voisins les plus proches avec  $d=1,2,3,\dots$ . Ces voisinages sont donc assimilables à des matrices appelées éléments structurant

ou bien masque de taille impaire. Le problème des voisinages d'un pixel de bord peut être résolu avec l'extension de la matrice par l'effet miroir par exemple.

La figure III.1 donne un exemple de mise en œuvre d'un filtre moyennneur en utilisant un voisinage 3x3.

103	159	124	101	98
105	155	135	101	70
116	143	142	162	79
191	169	150	128	69
183	175	147	105	79

Avant filtrage

122	130	129	106	93
123	131	136	112	95
143	145	143	115	92
163	157	147	118	94
181	169	145	112	88

Après filtrage

Figure III.1: Filtrage local moyennneur avec masque 3x3

Le point central qui avait une valeur de 142 sera remplacé par la moyenne des neuf éléments du masque à savoir par la valeur 143, tous les autres pixels seront traités de la même manière.

### III.2.2 Filtre non local

Dans le filtrage non local, on ne parle plus des voisins traditionnels d'un pixel (ceux qui l'entour), mais plutôt des voisins dont la valeur radiométrique est la plus proche à celle du pixel à traiter.

L'idée dans ce formalisme est de représenter une image par un graphe orienté  $G = (S, A)$  avec  $S = \{s_1, s_2, \dots, s_n\}$  qui est un ensemble fini de sommets représentant les pixels de l'image et  $A$  est un ensemble fini d'arrêtes.

Plusieurs types de graphe peuvent être construits tel que le  $\varepsilon$ -voisinage graphe, le k-plus proche voisin graphe ou le graphe complètement connecté. Dans la suite de ce travail, nous utiliserons le graphe des k-plus proches voisins en reliant chaque pixel de l'image avec les k pixels qui lui sont radiométriquement les plus similaires au sens d'une distance euclidienne radiométrique.

La figure III.2 donne l'ensemble des arrêtes dont le sommet en rouge est l'origine ainsi que ses 10 plus proches voisins au sens du graphe  $G$  qui correspond au 10-plus proches voisins. Pour réaliser le filtre moyennneur il suffira de remplacer la valeur de ce sommet par la moyenne de ses 10 plus proches voisins. Dans cet exemple, la valeur 98 sera remplacée par 100. Dans ce type de filtre, il n'y a aucune difficulté à traiter les pixels des bords.

103	159	124	101	98	94	85	87
105	155	135	101	70	109	90	89
116	143	142	162	79	87	117	84
191	169	150	128	69	71	89	135
183	175	147	105	79	63	102	96
161	175	147	98	56	81	123	117
178	170	138	102	72	151	106	136
162	138	143	111	75	104	145	136

Figure III.2 : Exemple de construction d'un graphe des 10-plus proches voisins

### III.2.3 Filtre non local avec patch

Dans le filtrage non local avec patch, le traitement entre deux pixels se fait patch par patch. Rappelons qu'un patch est une fenêtre rectangulaire ou carrée centrée autour du pixel.

L'idée dans ce formalisme est toujours de représenter une image par un graphe  $G = (S, A)$  avec  $S$  l'ensemble des sommets qui représente les pixels de l'image et  $A$  l'ensemble des arrêtes. Dans ce cas de filtre, on construit aussi un graphe des  $k$ -plus proches voisins en reliant chaque pixel de l'image avec les  $k$  pixels qui lui sont les plus similaires sauf que la similarité dans ce cas est définie par une distance de norme 1 entre deux patches. Cette distance est donnée par :

$$d(I(x, y), I(x', y')) = \sum_{i=-d}^d \sum_{j=-d}^d |I(x + i, y + j) - I(x' + i, y' + j)|$$

Avec  $I(x, y)$  est la valeur du pixel (sommets) se trouvant à la position  $(x, y)$ , et le patch est défini par une fenêtre carrée de taille  $(2d + 1) \times (2d + 1)$ .

Les pixels de bord seront traités en réalisant une extension de la matrice par l'effet miroir.

Dans la figure III.3, le pixel de valeur 142 (en rouge) représente le sommet du graphe  $G$ , les 5 arrêtes indique les 5 centres des patches les plus similaires au patch centré en 142. Pour réaliser le filtre moyenneur il suffira de remplacer la valeur de ce sommet par la moyenne des centres de ces 5 patches. La valeur 142 sera donc remplacée par 146.

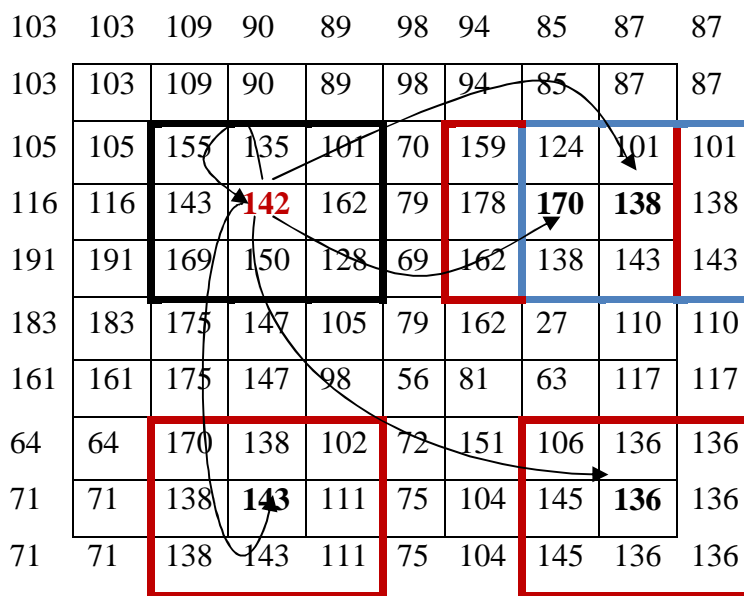


Figure III.3 : Exemple de construction d'un graphe des 5-plus proches voisins défini sur des patchs 3x3

### III.3 Méthodes de test et d'évaluations adoptées

Nous appliquons les trois modèles de filtre à savoir L, NL et NLP sur l'image réelle texturée woman. Pour se rendre compte de l'efficacité de chacun de ces modèles, nous les appliquerons successivement sur cette image réelle, puis sur cette même image entachée de bruit. Nous testerons 3 types de bruit à savoir les bruits « poivre et sel », « gaussien » et « poisson ». Pour réaliser une évaluation objective, nous calculerons pour chaque résultat un critère donnant la qualité de l'image filtrée.

Nous avons choisie d'utiliser le critère PSNR (Peak Signal to Noise Ratio) qui est un rapport signal sur bruit. Pour calculer ce rapport il faut d'abord mesurer l'erreur quadratique moyenne (en anglais Mean Square Error) entre l'image originale  $I$  et l'image reconstruite  $I'$  donnée par la formule suivante [20] :

$$MSE(I, I') = \frac{\sum_{x=1}^l \sum_{y=1}^c (I(x, y) - I'(x, y))^2}{l \times c}$$

Avec  $l$  qui est le nombre de lignes dans l'image et  $c$  le nombre de colonne de cette image. Le PSNR est alors défini par la relation :

$$PSNR(I, I') = 10 \log \frac{255^2}{MSE(I, I')}$$

Dans les paragraphes qui suivent, nous donnerons les différents résultats obtenus sur l'image woman et sur cette image bruitée en lui appliquant les filtres moyenneur, médian, érosion et dilatation de manière local (L), non local (NL) et non local avec patch (NLP).

### III.4 Filtre moyenneur

#### III.4.1 Filtre local

Rappelons qu'un filtre moyenneur local fait la moyenne des pixels d'un voisinage local. Le seul paramètre ajustable dans ce cas est la taille de la fenêtre de voisinage, elle est souvent carrée et impaire. Plus la taille de la fenêtre devient importante plus l'image résultat est floue ce qui est confirmé par la figure III.4. Nous remarquons aussi que ce filtrage ne préserve pas la texture.



Image original

Masque 3x3

Masque 5x5

Masque 9x9

Figure III.4 : Visualisation du résultat d'un filtre local moyenneur avec différents masques

#### III.4.2 Filtre non local

Dans le cas d'un filtre moyenneur non local le paramètre ajustable est le nombre  $k$  lors de la construction du graphe des  $k$ -plus proches voisins. Dans notre cas nous avons choisie de le prendre en fonction du nombre total des pixels de l'image. Nous l'indiquerons par un pourcentage de la taille total de l'image.

La Figure III.5 donne les résultats obtenus sur l'image Woman en appliquant un filtre moyenneur non local en utilisant un graphe des  $k$ -plus proches voisins dans le cas où  $k$  correspond à 0.1%, 1%, 10%, 50% et 80% du nombre totale des pixels de l'image c'est-à-dire  $k$  est respectivement égale à 65, 655, 6553, 32768 et 52428 pixels, car le nombre total des pixels de l'image est de  $256 \times 256 = 65536$ .



Figure III.5 : Visualisation du résultat d'un filtre non local moyennneur avec différentes valeurs de  $k$

Nous constatons que ce modèle de filtre présente un atout principal qui est de conserver la texture. Effectivement si nous observons le foulard et la chaise en arrière plan, leurs textures sont préservées jusqu'à des valeurs importantes de  $k$ . Quand cette valeur devient très importante, les détails de l'image commencent à disparaître. Effectivement ce résultat est prévisible car si  $k$  est important, le graphe alors construit tend vers un graphe complet. Dans le cas d'un graphe complet la moyenne calculée pour chaque pixel sera alors identique pour tous les pixels.

#### III.4.3 Filtre non local avec patch

Pour ce qui est du filtre NLP deux paramètres sont ajustable à savoir la taille du patch et le nombre des  $k$ -plus proches voisins. La Figure III.6 donne les résultats obtenus sur l'image Woman en appliquant un filtre non local moyennneur avec patch et en variant en même temps le paramètre  $k$  (0.1%, 1%, 10% et 50% du nombre totale des pixels de l'image) et la taille du patch (3x3, 5x5, 7x7 et 9x9) .

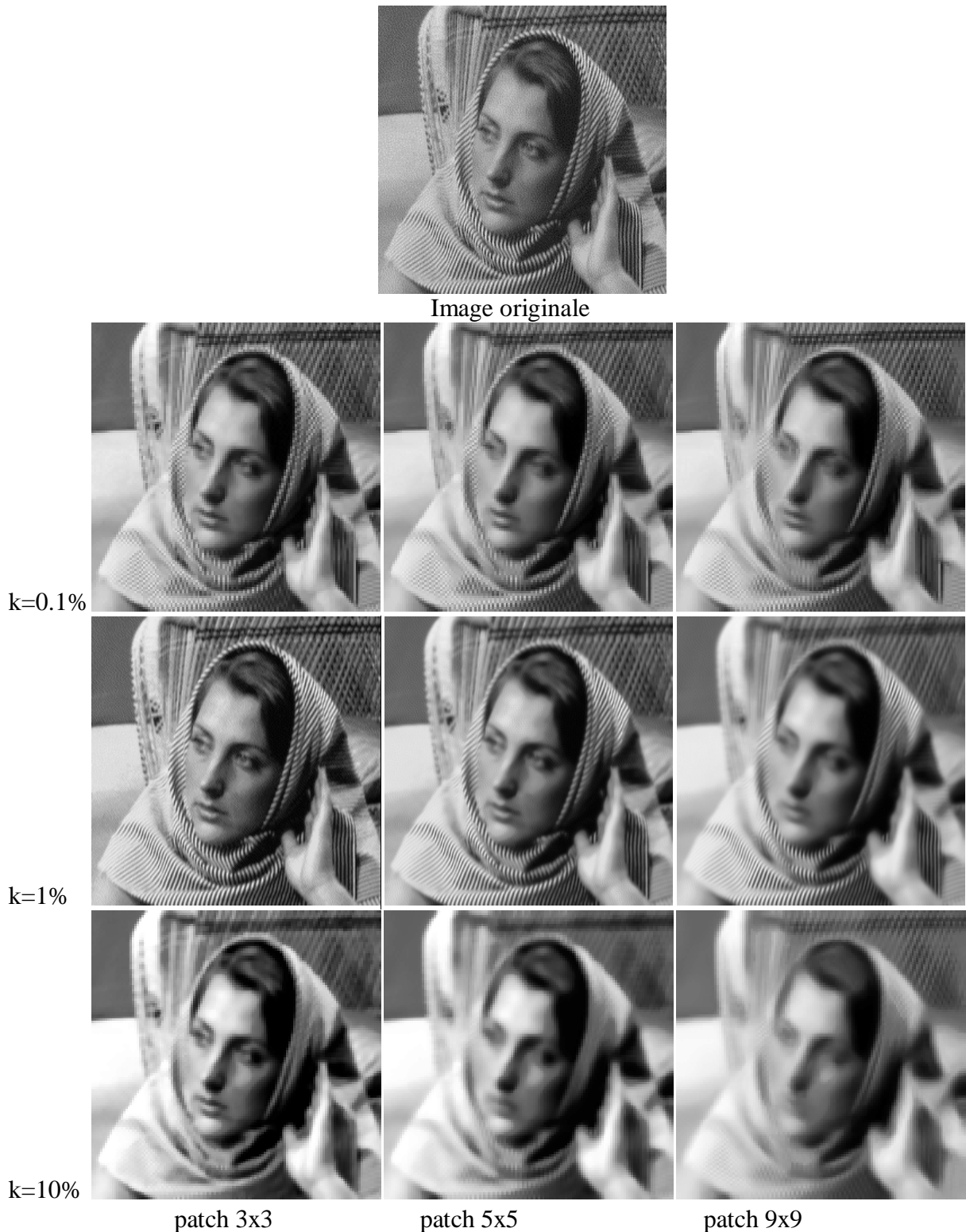


Figure III.6: Visualisation du résultat d'un filtre moyenneur non local avec Patch avec différentes valeurs de  $k$  et différentes tailles des patches

Il est assez difficile de choisir en même temps la taille du patch et le nombre  $k$  des plus proches voisins. On remarque que lorsque la taille du patch est importante et le nombre  $k$  l'est aussi l'image devient flou. Il nous apparait que pour l'image de la figure III.6, le meilleur

résultat est obtenu avec un patch de 3x3 et  $k=1\%$ .

**III.4.4 Tests avec bruits**

Pour se rendre compte de l'efficacité de chacun des trois modèles de filtres (L, NL, NLP), nous testerons l'image woman entachée de bruits "poivre et sel", "gaussien" et "poisson". Les différents résultats sont donnés sur la figure III.7.



Figure III.7: Visualisation des résultats des différents filtres moyennés sur l'image woman entachée de bruit a) poivre et sel b) poisson c) gaussien

Cette figure est organisée en colonne. Sur la première colonne, nous affichons l'image entachée de bruit, sur la seconde colonne les résultat après filtrage local en utilisant un

masque 3x3, sur la troisième colonne les résultats du filtrage non local avec  $k=1\%$  et sur la dernière colonne les résultats après filtrage non local avec patch en utilisant des patch de taille 3x3 et  $k=1\%$  pour le graph k-plus proches voisins. Afin d'évaluer quantitativement les résultats observés, nous calculons à chaque fois la valeur du rapport signal sur bruit (PSNR). Ces rapports sont récapitulés sur le tableau III.1.

Nous constatons sur la figure III.7 que dans le cas du bruit poivre et sel, le filtrage NLP réduit le bruit car nous ne retrouvons plus certains pixels bruités dans le résultat. Le filtre NL semble n'apporter aucune modification à l'image bruitée. Le bruit n'est alors ni éliminé ni même atténué dans ce cas de figure. Le filtre L, quand à lui étale le bruit présent sur un pixel à son voisinage. Donc dans le cas du bruit poivre et sel, le meilleur résultat est obtenu dans le cas du filtre NLP, ce qui est confirmé par la valeur du PSNR qui est la plus élevée dans ce cas là (voir tableau III.1).

Dans le cas du bruit gaussien, le filtre NLP fournit une image avec un meilleur contraste mais le bruit nous ne semble pas réduit. Il n'est pas réduit aussi dans le cas du filtrage NL. Il est peu réduit dans le cas du filtrage L. Effectivement, dans le cas de ce bruit le meilleur score du PSNR est obtenu pour ce dernier filtrage. L'inconvénient de ce filtre L est que la texture se trouve dégradée, même si cette dégradation n'est pas très importante.

Pour ce qui est du bruit de type poisson, les différents filtres semblent avoir le même effet sur ce bruit, il n'est pas nettement atténué. Le filtre NLP donne aussi dans ce cas une image avec un meilleur contraste. Le meilleur PSNR dans ce cas là est obtenu pour le filtrage NL.

Tableau III.1 : Récapitulatif des PSNR d'un filtrage moyenné avec différents bruits en utilisant les différents modèles de filtre.

Type de bruit	Local	Non local $k=1\%$	NLP 3x3 et $k=1\%$
Poivre et sel	48.9154	47.1699	49.7410
Poisson	50.2183	56.5821	52.9594
Gaussien	48.9142	45.7470	46.7750

### III.5 Filtre médian

#### III.5.1 Filtre local

Dans le filtrage Médian local, la valeur du pixel résultant représente la valeur médiane des voisinages du pixel dans l'image d'origine. Le paramètre à ajuster dans ce cas est la taille du masque. La figure III.8 donne les résultats de ce filtre en utilisant différentes tailles du masque. Nous remarquons que ce filtre détériore la texture surtout pour des masques dont la

dimension est supérieure à  $3 \times 3$ . Ce type de filtre est souvent utilisé pour réduire le bruit de type poivre et sel. Nous le testerons pour ce type de bruit et pour d'autre en utilisant un masque  $3 \times 3$  dans les paragraphes qui suivent.

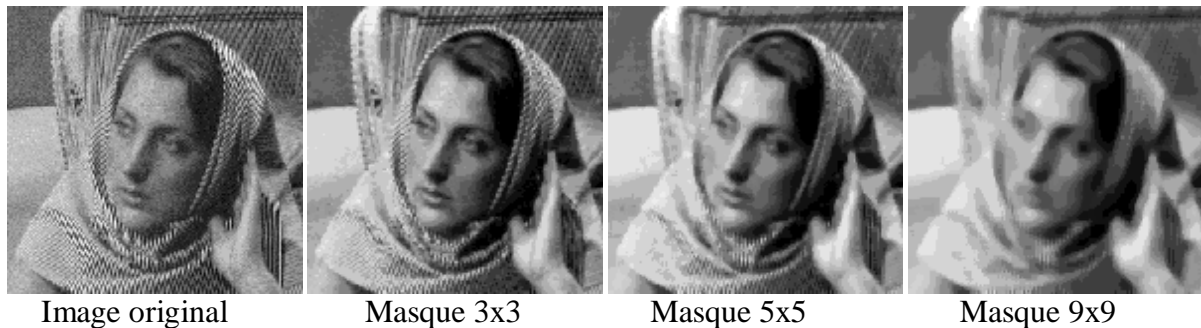


Figure III.8 : Visualisation du résultat d'un filtre médian local avec différents masques

### III.5.2 Filtre non local

Dans le cas du filtrage médian NL la valeur du pixel dans la nouvelle image représente la valeur médiane des  $k$ -plus proches voisins radiométrique du pixel d'origine.

La figure III.9 illustre le résultat du filtrage médian NL en variant le paramètre  $k$  de 1% à 80%. Nous constatons que ce filtre est efficace pour la préservation de la texture avec le bon choix de la variable  $k$ . dans notre cas les deux valeurs 1% et 10% sont bonne ce qui n'est pas le cas pour 50%.

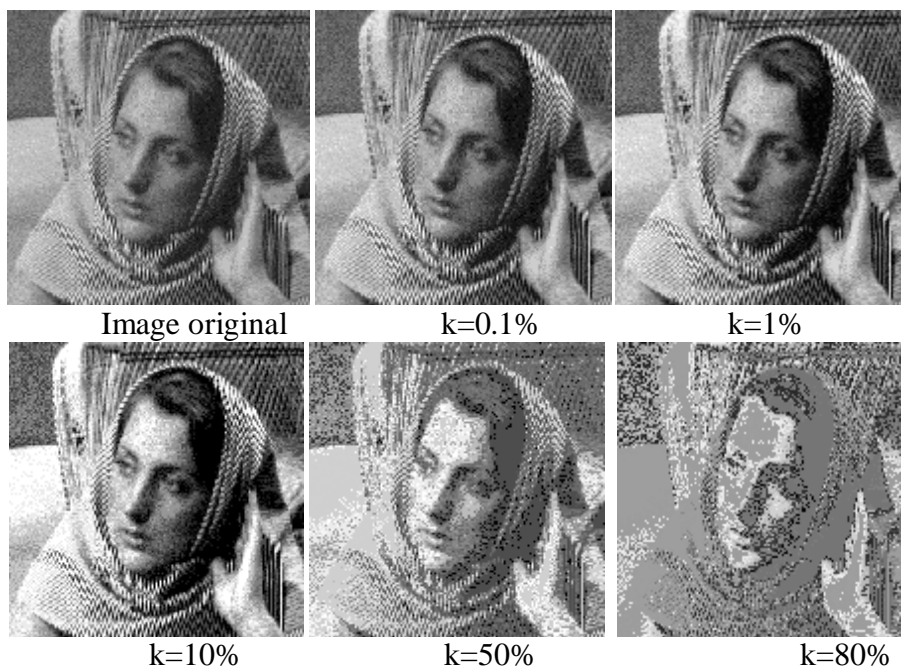


Figure III.9 : Visualisation des résultats du filtrage médian NL avec différentes valeurs de  $k$

Nous constatons que ce filtre est efficace pour la préservation de la texture, sauf que, lorsque la valeur de  $k$  devient très importante l'image est détériorée. Comme dans le cas du filtre

moyenleur, lorsque  $k$  est très grand, le graphe construit tend vers un graphe complet et le résultat dans ce cas est moins précis.

### III.5.3 Filtre non local avec patch

Le filtre NLP est obtenu en construisant un graphe  $G=(S, A)$  où chaque pixel est reliée à ses  $k$  plus proche voisin. Le voisinage dans ce cas est déterminé par une distance radiométrique sur les patches. La valeur du pixel est remplacée par la valeur médiane de ces  $k$  plus proches voisins qui sont au centre de chacun des patches.

La Figure III.10 donne les résultats obtenus sur l'image Woman en appliquant un filtre non local médian avec patch en variant le paramètre  $k$  (0.1%, 1%, 10% et 50% du nombre totale des pixels de l'image) et en fixant la taille du patch à 3x3 et 5x5.

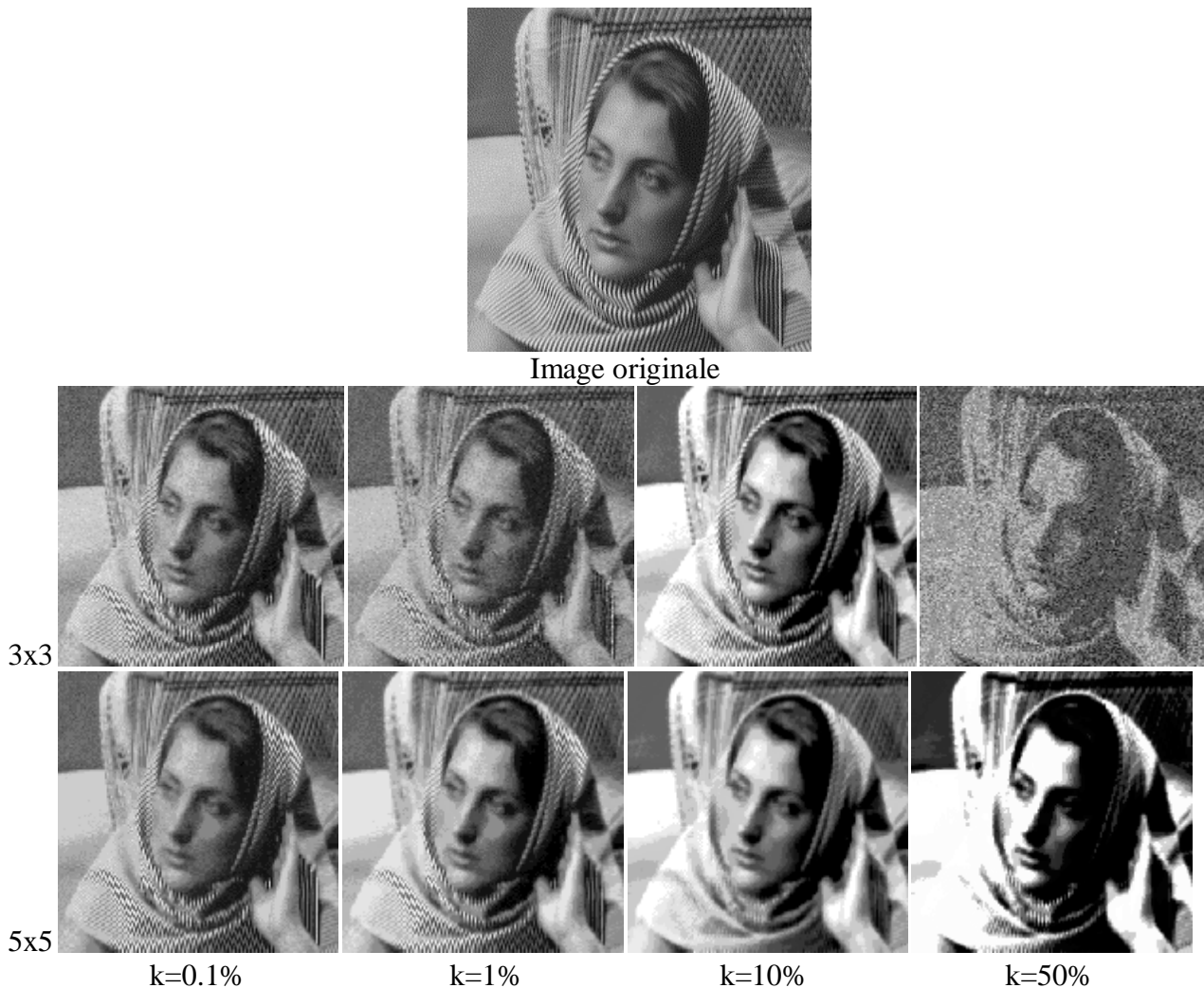


Figure III.10: Visualisation des résultats du filtre médian NLP avec différentes valeurs de  $k$

Nous avons testé différentes tailles pour les patches à savoir 3x3 et 5x5 et différentes valeurs de  $k$ . nous avons remarqué que lorsque  $k$  est élevée l'image résultat est dégradé. Les meilleurs résultats sont obtenus pour  $k=0.1\%$  et  $k=1\%$ .

### III.5.4 Tests avec bruits

Pour voir clairement l'efficacité des modèles de filtrage L, NL et NLP médian, nous visualisons dans la figure III.11 les résultats de ces filtrages sur l'image woman entachée de bruit. Sur la première colonne, nous affichons les images entachées des bruits « poivre et sel », « gaussien » et « poisson », sur la colonne suivante les résultats du filtrage médian local avec un masque 3x3, sur la troisième colonne celle du filtrage médian non local avec  $k=1\%$  et sur la dernière colonne le résultat du filtrage non local avec patch avec une taille de patch 3x3 et  $k=1\%$ .



Figure III.11 : Visualisation des résultats des différents filtrages médian sur l'image woman entachée de bruit a) poivre et sel b) poisson c) gaussien

Ce qui est très remarquable dans la figure III.11 est l'efficacité du filtrage NLP sur le bruit poivre et sel qui est presque totalement éliminé. Dans ce cas aussi la texture est parfaitement préservée.

Pour le bruit poisson le model de filtre qui nous semble donner de meilleurs résultats est le non local. Par contre dans le cas du bruit gaussien aucun des trois modèles n'ont apporté une amélioration remarquable pour l'image bruité. Le tableau III.2 donne les résultats du rapport signal sur bruit pour chaque types de bruit en fonction de chaque model de filtre. Nous remarquons que les valeurs du PSNR donnée dans ce tableau confirme les constatations visuelles précédentes. Effectivement le filtre NLP, dans le cas du bruit poivre et sel, donne un PSNR largement supérieur aux deux autres filtres. Dans le cas du bruit Poisson, le filtre NL, donne aussi un PSNR largement supérieur aux deux autres filtres. Par contre, dans le cas du bruit gaussien, les trois filtres donnent des PSNR assez semblables.

Tableau III.2 : Récapitulatif des PSNR d'un filtrage médian avec différents bruits en utilisant les différents modèles de filtre.

Type de bruit	Local	Non local k=1%	NLP 3x3 et k= 1%
Poivre et sel	45.8541	47.1595	60.8618
Poisson	46.8221	60.8416	51.8906
Gaussien	45.6352	46.3196	46.3443

## III.6 Erosion

### III.6.1 Erosion locale

Rappelons qu'un filtrage par érosion locale consiste à remplacer la valeur du pixel centrale par la plus faible valeur de son voisinage. Comme les filtres locaux précédents, le seul paramètre ajustable est la taille du masque de voisinage. La figure III.12 fournit les résultats de ce filtrage sur l'image women avec une taille du masque variable.



Image originale

masque 3x3

masque 5x5

masque 9x9

Figure III.12 : Visualisation du résultat d'un filtre érosion local avec différents masques

Plus la taille du masque est importante plus l'image est déformée. Ce filtre ne préserve pas la texture.

### III.6.2 Erosion non locale

Le filtrage par érosion non locale consiste à remplacer la valeur d'un pixel par la plus faible valeur des  $k$  pixels qui lui sont radiométriquement les plus proches. La figure III.13 donne les résultats de ce filtrage avec différentes valeurs de  $k$ .

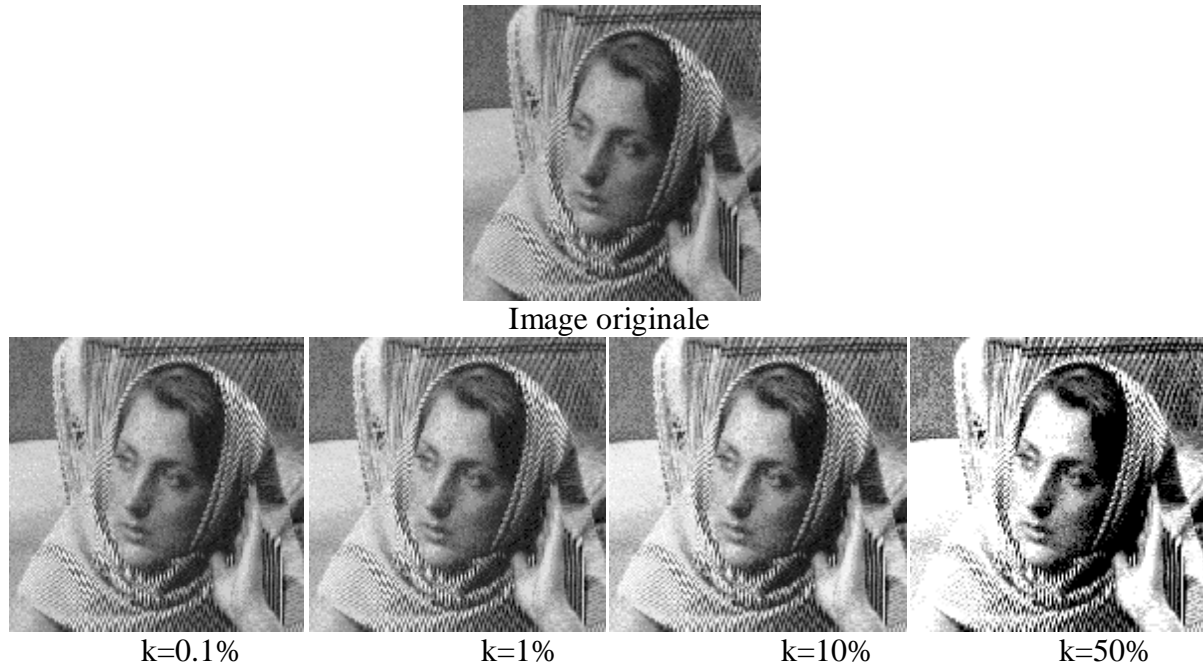


Figure III.13 : Visualisation des résultats d'un filtre érosion NL avec différentes valeurs de  $k$

Nous constatons que les résultats obtenus avec ce type de filtre sont nettement meilleurs que les résultats du filtrage par érosion locale. La texture dans ce cas est préservée.

### III.6.3 Erosion non locale avec patch

Le filtrage par érosion NLP consiste à remplacer la valeur d'un pixel par la plus faible valeur parmi ses  $k$  plus proches voisins. Rappelons que pour le NLP les voisins sont les pixels dont les distances entre patches est la plus faible. La figure III.14 fournit les résultats de ce filtre avec différentes valeurs de  $k$  et la même taille du masque qui est de  $3 \times 3$ .

Ce filtre préserve aussi la texture même pour des grandes valeurs de  $k$  (voir figure III.14  $k=50\%$ ) sauf que lorsque  $k$  est assez important l'image devient sombre.

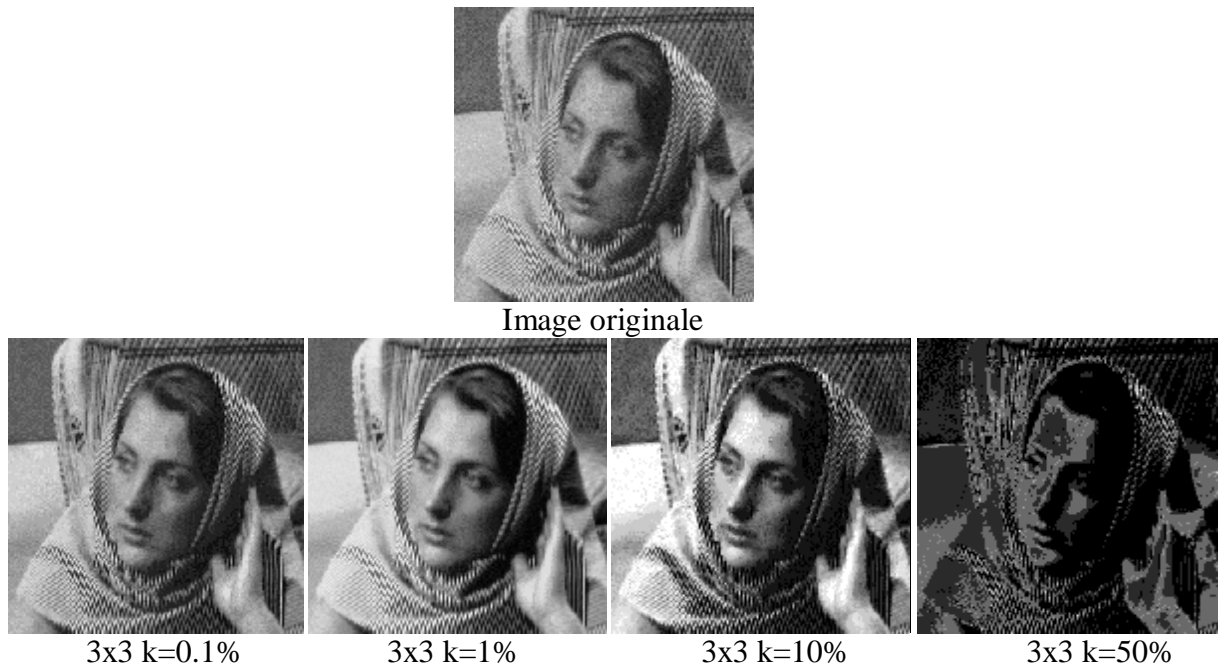


Figure III.14: Visualisation des résultats d'une érosion NLP avec taille du masque 3x3 et différentes valeurs de k

#### III.6.4 Testes avec bruits

Nous testons le filtrage par érosion L, NL et NLP sur image woman bruitée avec les bruits « poivre et sel », « poisson » et « gaussien » (Voir figure III.15).

Nous constatons que le bruit poivre et sel est assez difficile à éliminer avec l'érosion. Dans le cas de l'érosion L, le bruit en blanc est éliminé par contre le bruit en noir est étalé sur les pixels voisins.

Pour les filtres NL et NLP, le bruit poivre et sel est réduit surtout dans le cas du NL. Ces résultats sont confirmés par la valeur du PSNR donnée sur le tableau III.3.

Dans le cas du bruit poisson, le meilleur résultat est obtenu par le filtre NL (Voir tableau III.3).

Le filtre NL donne aussi un meilleur résultat pour le bruit gaussien.



Figure III.15: Visualisation des résultats des différents filtrages érosion sur l'image woman entachée de bruit a) poivre et sel b) poisson c) gaussien

Tableau III.3 : Récapitulatif des PSNR d'un filtrage érosion avec différents bruits en utilisant les différents modèles de filtre.

Type de bruit	Local	Non local k=1%	NLP 3x3 et k= 1%
Poivre et sel	PSNR = 29.8421	PSNR =99.8946	PSNR =33.4299
Poisson	PSNR = 38.9206	PSNR =56.2265	PSNR = 45.0544
Gaussien	PSNR = 31.7078	PSNR = 68.9410	PSNR = 33.0404

### III.7 Dilatation

#### III.7.1 Dilatation locale

La dilatation locale consiste à remplacer la valeur du pixel par la plus grande valeur de son voisinage. La figure III.16 fournit les résultats de la dilatation locale de l'image woman avec différentes tailles du masque de voisinage.

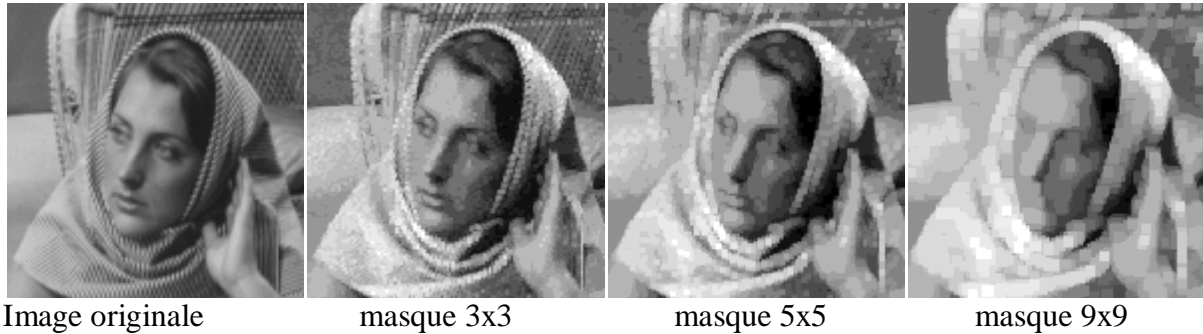


Figure III.16 : Visualisation des résultats d'un filtre dilatation local avec différents masques

On remarque que ce filtre ne préserve pas la texture. Lorsque la taille du masque est importante l'image résultat est dégradée.

#### III.7.2 Dilatation non locale

Le filtrage par dilatation non locale consiste à remplacer la valeur d'un pixel par la plus grande valeur des  $k$  pixels qui lui sont radiométriquement les plus proches. La figure III.17 donne les résultats de ce filtrage avec différentes valeurs de  $k$ .

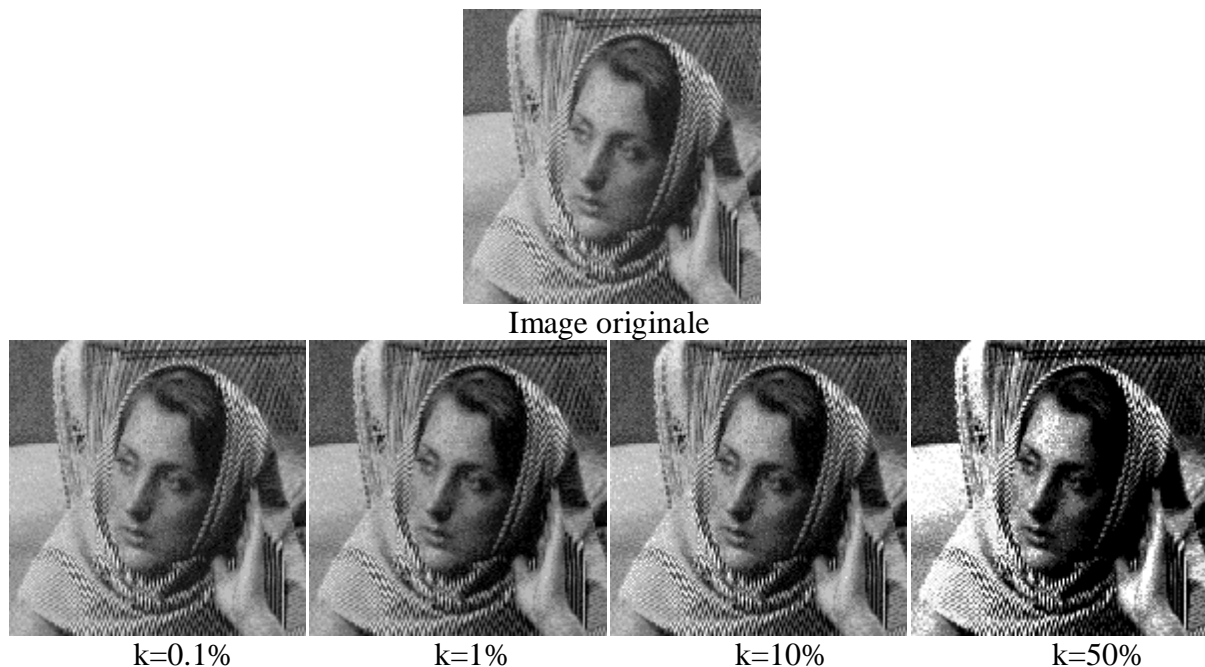


Figure III.17: Visualisation des résultats d'un filtre dilatation NL avec différentes valeurs de  $k$

Contrairement au filtre local, ce filtre NL préserve la texture. Même pour des valeurs importantes de  $k$ , la texture est globalement préservée. Néanmoins les meilleurs résultats sont observés pour des valeurs de  $k$  égales à 0.1%, 1% et 10%.

### III.7.3 Dilatation non local avec patch

Le filtrage par dilatation NLP consiste à remplacer la valeur d'un pixel par la plus grande valeur parmi ses  $k$  plus proches voisins. Le voisinage dans ce cas est déterminé par une distance de norme 1 entre patches. La figure III.18 donne les résultats de ce filtrage.

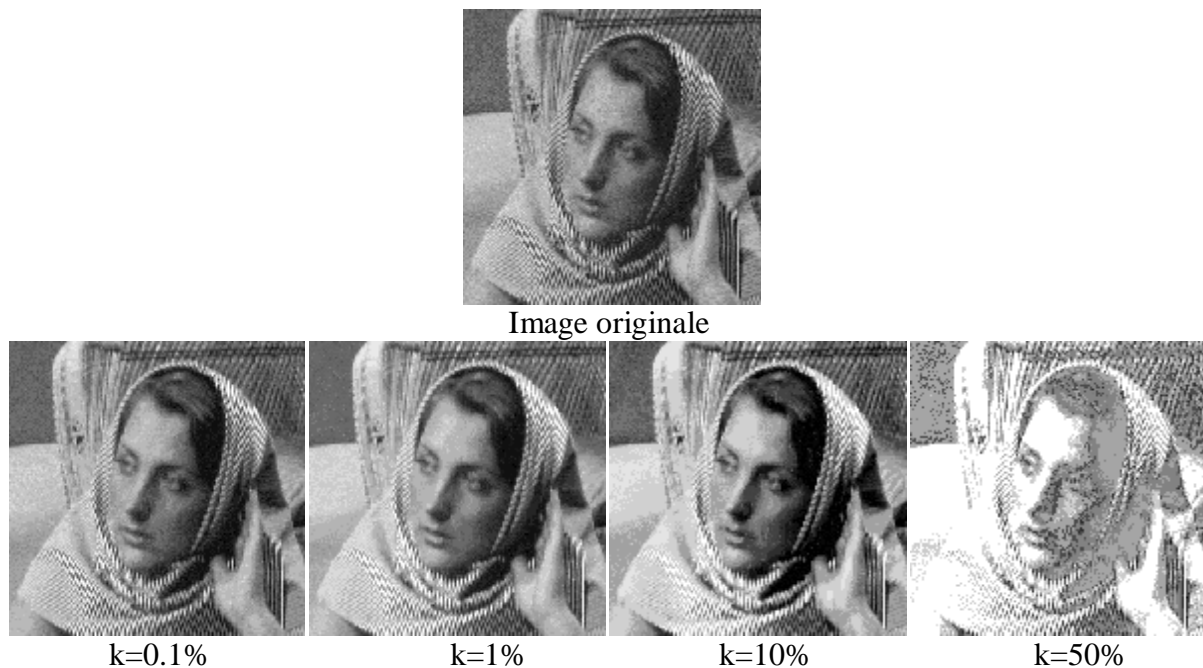


Figure III.18: Visualisation du résultat d'une dilatation non locale avec Patch avec taille du masque  $3 \times 3$  et différentes valeurs de  $k$

Ce filtre préserve la texture même pour une valeur élevée de  $k$ , sauf que lorsque  $k$  est assez important l'image résultat devient claire (blanche).

### III.7.4 Tests avec bruits

Nous testons le filtrage par dilatation L, NL et NLP sur l'image woman bruitée avec les bruits « poivre et sel », « poisson » et « gaussien » (Voir figure III.19).



Figure III.19: Visualisation des résultats des différents filtres érosion sur l'image woman entachée de bruit a) poivre et sel b) poisson c) gaussien

La dilatation  $L$  élimine le bruit en noir et étale le bruit en blanc sur les pixels voisins. Pour les filtres NL et NLP, le bruit poivre et sel est réduit surtout dans le cas du NL. Ces résultats sont confirmés par la valeur du PSNR donné sur le tableau III.4.

Dans le cas du bruit poisson, le meilleur résultat est obtenu par le filtre NL. En ce qui concerne le bruit gaussien, les trois filtres donnent des résultats assez semblables.

Tableau III.4 : Récapitulatif des PSNR d'un filtrage érosion avec différents bruits en utilisant les différents modèles de filtre.

Type de bruit	Local	Non local k=1%	NLP 3x3 et k= 1%
Poivre et sel	PSNR = 28.4472	PSNR =107.1935	PSNR = 34.0176
Poisson	PSNR =37.2639	PSNR =205.6344	PSNR =40.3564
Gaussien	PSNR = 31.8755	PSNR =31.7078	PSNR =33.0404

### III.9 Conclusion

D'après les résultats obtenus dans ce chapitre nous pouvons conclure que les modèles de filtres NL et NLP préservent la texture ce qui n'est pas le cas pour les filtres locaux.

D'après tous les tests réalisés nous avons constaté que le modèle et le type de filtre doit être choisi en fonction de l'image et du bruit traité. Les paramètres variables de ces différents filtres sont à choisir de manière empirique.

# ***Conclusion générale***

Le but du filtrage est d'éliminer au maximum les perturbations tout en respectant l'intégrité de la scène originale. Bien entendu, il n'y a pas de filtre universel capable de corriger tous les défauts.

Le travail qui nous a été proposé est le filtrage des images en utilisant trois méthodologies différentes. Il s'agit de déterminer la nouvelle valeur du pixel en utilisant un ensemble de pixels déterminé en utilisant trois types de graphes. Pour le filtrage local, il s'agit de construire un graphe des k plus proches voisins spatial, pour le filtrage non local, on construit le graphe des k plus proche voisin radiométriques et pour le filtrage non local avec patch, on construit le graphe des k plus proches patches.

Nous avons implémenté ces trois modèles de filtres et nous les avons appliqués sur quatre types de filtres à savoir, le moyenneur, le médian, l'érosion et la dilatation.

Nous avons constaté que les filtres NL et NLP, contrairement aux filtres locaux, préservent la texture, ce qui est un atout majeur lors du traitement des images texturées.

L'inconvénient de ce modèle de filtrage, est le temps d'exécution des programmes qui est important surtout dans le cas du filtrage non local avec patch.

Le travail réalisé dans ce mémoire présente plusieurs perspectives. Effectivement, il serait intéressant de tester ces modèles de filtre sur d'autres images texturées et non texturées mais aussi sur des images multi-composantes tel que les images couleur et satellitaires. Le temps d'exécution peut être aussi réduit en optimisant nos programmes mais aussi en utilisant des logiciels de programmation qui sont connue pour être plus rapide que le Matlab tel que le C++. Il est aussi intéressant de réfléchir à la manière d'appliquer la non localité à d'autres types de filtres tel que par exemple, les filtres de Sobel, Prewitt, Robert ainsi que d'autres filtres passe-haut, passe-bas et morphologiques.

# Bibliographie

- [01] Y. Champclaux, Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information, Thèse de doctorat de l'université de Toulouse, Paul Sabatier, 2009.
- [02] D. Maquin, Eléments de Théorie des Graphes, support de cours, Institut National Polytechnique de Lorraine, Ecole Nationale Supérieure d'Electricité et de Mécanique, 2003.
- [03] J. Cohen, Théorie des graphes et algorithmes, support de cours, université Paris-Est, 2006, [http://www.lacl.fr/cohen/poly\\_gr.pdf](http://www.lacl.fr/cohen/poly_gr.pdf).
- [04] M. Younsi, A. Salmi, Classification automatique de donnée basée sur la théorie des graphes, Thèse d'ingénieur d'état en Automatique, UMMTO, 2011.
- [05] C. Solnon, Théorie des graphes et optimisation dans les graphes, support de cours, INSA de Lyon, 2012, <http://liris.cnrs.fr/csolonon/polyGraphe.pdf>.
- [06] E. Sigward, Introduction a la théorie des graphes, support de cours, Académie de Nancy-Metz, 2002, <http://www.ac-nancy-metz.fr/enseign/maths/m2002/.../graphes/graphes.pdf>
- [07] N. Belharrat, théorie des graphes en recherche opérationnelle, éditions page blues, 2009.
- [08] C. Allène, Paradigmes de segmentation de graphe comparaisons et applications en traitement d'images, Thèse de doctorat en Informatique, université Paris-Est, 2009.
- [09] M. Rigo, Théorie des graphes, Support de cours de l'université de Liège, Département de mathématique, 2009–2010.
- [10] U. Von Luxburg, A Tutorial on Spectral Clustering, Statistics and Computing, 17 (4), 2007.
- [11] F. Mallek, segmentation d'images texturées couleurs par approche isosegment, Mémoire d'ingénieur d'Etat en électronique, UMMTO, 2011.
- [12] X. Philippeau, Les filtres usuels en traitement d'images, support de cours, 2007, <http://xphilipp.devlopez.com/articles/filtres/filtres.pdf>.
- [13] S. Benamrouz, S. Kettan, Segmentation d'images par méthode adaptative basée sur les matrices de cooccurrences, Mémoire d'Ingénieurs d'Etat en électronique, UMMTO, 2009.
- [14] A. Buades, B. Coll, J.-M. Morel, A non-local algorithm for image denoising, IEEE Int. Conf. On Computer Vision and Pattern Recognition, pp. 60-65, 2005.
- [15] C.-A Deledalle, L. Denis, F. Tupin, Débruitage Non-Local Itératif fondé sur un Critère de Similarité Probabiliste, GRETSI, Dijon, France, 2009.

- [16] J. Salmon, Méthodes à Patches pour le débruitage d'images numériques, Neuvième Colloque "Jeunes Probabilistes et Statisticiens", Université Paris Diderot - Paris 7, Mai 2010
- [17] A. Buades, B. Coll, J. Morel, A review of image denoising algorithms, *Multiscale Modeling and Simulation* 4 (2), pp. 490–530, 2005.
- [18] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling. *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2 , pp. 1033 – 1038, 1999.
- [19] A. Criminisi, P. Perez et K A. Criminisi, P. Perez, K. Toyama, Region filling and object removal by exemplarbased image inpainting, *IEEE Transactions on Image Processing*, 13, pp. 1200–1212, 2004.
- [20] S. Postec, Quelques remarques en débruitage des images liées à des propriétés de similarité, de régularité et de parcimonie, Thèse de Doctorat de l'Université de Bretagne-Sud Spécialité Mathématiques, 2012.