

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

## UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUEES  
DÉPARTEMENT DE GENIE ELECTRIQUE



# Polycopié de TP-Vision Artificielle

« Rappels et TPs résolus »

Par

**Dr. Abdelkader Moustafa Radwane GHELLAB**

**Matière** : TP-Vision Artificielle

**Promotion** : 2<sup>ème</sup> année Master Electronique des Systèmes Embarqués

**Semestre** : 03

**Expertisé par** :

- Mr Nasri Djilali                      Pr              Univ-Tiaret
- Mr Belhadji Youssef              MCA              Univ-Tiaret

# SOMMAIRE

Introduction générale .....	1
<b>I- TP 1 : Manipulation et amélioration d'images par traitement d'histogramme</b> .....	<b>2</b>
I-1 Rappel... ..	2
I-2 But de TP1.....	6
I-3 Travail demandé du TP1.....	6
I-4 Solution de TP1.....	7
<b>II- TP 2 : Filtrage par opération de convolution 2D.....</b>	<b>11</b>
II-1 Rappel... ..	11
II-2 But de TP2.....	15
II-3 Travail demandé du TP2.....	15
II-4 Solution de TP2.....	16
<b>III- TP 3 : Filtrage dans le domaine fréquentiel.....</b>	<b>21</b>
III-1 Rappel... ..	21
III-2 But de TP3.....	23
III-3 Travail demandé du TP3.....	23
III-4 Solution de TP3.....	24
<b>IV- TP 4 : Filtrage morphologique.....</b>	<b>28</b>
IV-1 Rappel... ..	28
IV-2 But de TP4.....	29
IV-3 Travail demandé du TP4.....	29
IV-4 Solution de TP4.....	29
<b>V- TP 5 : Détection de contours et rehaussement de contraste .....</b>	<b>33</b>
V-1 Rappel... ..	33
V-2 But de TP5.....	37
V-3 Travail demandé du TP5.....	37
V-4 Solution de TP5.....	38
<b>VI- TP 6 : Outils de segmentation .....</b>	<b>41</b>
VI-1 Rappel... ..	41
VI-2 But de TP6.....	52
VI-3 Travail demandé du TP6.....	52
VI-4 Solution de TP6.....	53
<b>VII- TP 7 : Détection d'objets similaires dans une image par opération de corrélation (détection directe).....</b>	<b>56</b>
VII-1 Rappel... ..	56

VII-2	But de TP7.....	56
VII-3	Travail demandé du TP7.....	56
VII-4	Solution de TP7.....	57
<b>VIII-</b>	<b>TP 8 : Détection de points d'intérêt pour la mise en correspondance des objets (méthodes de détection indirecte) .....</b>	<b>60</b>
VIII-1	Rappel... ..	60
VIII-2	But de TP8.....	64
VIII-3	Travail demandé du TP8.....	64
VIII-4	Solution de TP8.....	64
VIII-5	Annexe du TP8.....	67
VIII-5-1	Exploitation de la méthode SIFT pour l'estimation de mouvement d'objet mobile.....	67
<b>IX-</b>	<b>Conclusion générale .....</b>	<b>69</b>
<b>X-</b>	<b>Bibliographie.....</b>	<b>69</b>

## LISTE DES FIGURES

<b>Figure-I-1</b> Vocabulaire de l'image numérique. ....	2
<b>Figure-I-2</b> Résolution spatiale et résolution tonale. ....	3
<b>Figure-I-3</b> Schéma illustratif du concept d'histogramme pour une image. ....	4
<b>Figure-I-4</b> Schéma illustratif du diagnostic et de traitement par normalisation. ....	5
<b>Figure-I-5</b> Schéma illustratif du diagnostic et de traitement par égalisation.....	5
<b>Figure-I-6</b> Chargement et affichages des deux images à diagnostiquer et à traiter. ....	7
<b>Figure-I-7</b> Les deux images à diagnostiquer et leurs histogrammes respectifs ....	8
<b>Figure-I-8</b> Traitement de normalisation pour image satellite et d'égalisation pour image route et leurs histogrammes respectifs avant et après traitement.....	9
<b>Figure-I-9</b> Histogrammes cumulés des deux images avant et après traitement. ....	10
<b>Figure-II-1</b> Noyaux et concept de convolution 2D. ....	12
<b>Figure-II-2</b> Filtrage par convolution 2D. ....	13
<b>Figure-II-3</b> Filtrage médian sur image affectée par un bruit impulsionnel. ....	14
<b>Figure-II-4</b> image "eight.tif". ....	16
<b>Figure-II-5</b> Image et image affectée par un bruit de type « salt & pepper » de densité de 0.02. ....	17
<b>Figure-II-6</b> Résultats de filtrage linéaire (par convolution). ....	17
<b>Figure-II-7</b> Résultats de filtrage non linéaire (médian) comparé aux noyaux de filtrage linéaire. ....	18
<b>Figure-II-8</b> Résultats de filtrage linéaire et non linéaire sur une image affectée par un bruit blanc gaussien..	19
<b>Figure-II-9</b> Résultats de filtrage linéaire et non linéaire sur une image affectée par un bruit blanc de Poisson...	19
<b>Figure-III-1</b> Caractéristiques de la TFD bidimensionnelle. ....	21
<b>Figure-III-2</b> Filtrage par masquage utilisant la TFD bidimensionnelle. ....	22
<b>Figure-III-3</b> Chargement et affichage de l'image « phone.png ». ....	24
<b>Figure-III-4</b> Spectre et spectre réordonné de l'image « phone.png ». ....	25
<b>Figure-III-5</b> Etapes de filtrage passe-haut sur l'image « phone.png ». ....	25
<b>Figure-III-6</b> Etapes de filtrage passe-bas sur l'image « phone.png ». ....	26
<b>Figure-III-7</b> Etapes de filtrage passe-bas (rayon D=30) sur l'image « phone.png » affectée par un bruit blanc Gaussien. ....	27
<b>Figure-IV-1</b> érosion morphologique. ....	30
<b>Figure-IV-2</b> Ouverture morphologique. ....	31

<b>Figure-IV-3</b> Dilatation et fermeture morphologique. ....	31
<b>Figure-V-1</b> Détection de contours par filtre dérivateur ainsi que son utilisation pour rehausser le contraste de l'image. ....	34
<b>Figure-V-2</b> Illustration de résultat de filtrage dérivateur par différent noyaux dérivateurs. ....	35
<b>Figure-V-3</b> Détection de contours par le filtrage de Laplacien. ....	35
<b>figure-V-4</b> Rehaussement de contraste. ....	36
<b>figure-V-5</b> Détection des contours par filtre de Laplacien. ....	38
<b>Figure-V-6</b> Création de l'image dégradée à traiter. ....	39
<b>Figure-V-7</b> Rehaussement de contraste de l'image dégradée. ....	39
<b>Figure-VI-1</b> Segmentation d'image se basant sur le critère de la couleur. ....	41
<b>Figure-VI-2</b> Concept de pic, dans l'histogramme, correspondant à une couleur d'un objet homogène. ....	42
<b>Figure-VI-3</b> Processus itératif d'éliminations récursives des pics lors de définition de leur ensemble de pixels approprié (segmentation d'objets homogènes). ....	43
<b>Figure-VI-4</b> Processus itératif d'accroissement de région (Approche locale). ....	44
<b>Figure-VI-5</b> Processus récursive de division dans la méthode SPLIT & MERGE. ....	45
<b>Figure-VI-6</b> Avantage et inconvénients SPLIT & MERGE. ....	46
<b>Figure-VI-7</b> Création d'îlots hexagonaux chevauchés lors de la phase de division pour l'algorithme CSC. ....	47
<b>Figure-VI-8</b> Division récursive d'îlots. ....	47
<b>Figure-VI-9</b> Phase d'initialisation de l'algorithme CSC. ....	48
<b>Figure-VI-10</b> Phase de regroupement de l'algorithme CSC. ....	49
<b>Figure-VI-11</b> Phase de découpage récursif de l'algorithme CSC. ....	50
<b>Figure-VI-12</b> Formes obtenues des objets segmentés par l'algorithme CSC. ....	50
<b>Figure-VI-13</b> Illustration de la segmentation par la méthode « Snake ». ....	51
<b>Figure-VI-14</b> Segmentation de l'image chargée par la méthode « Snake ». ....	53
<b>Figure-VI-15</b> Image à segmenter et sa représentation en 3D. ....	54
<b>Figure-VI-16</b> Image segmentée par la méthode « Water thresholding ». ....	54
<b>Figure-VII-1</b> Image et <i>template</i> . ....	57
<b>Figure-VII-2</b> Détection d'objets de la même classe que le <i>template</i> à différents seuils de coefficients de corrélation. ....	58

<b>Figure-VII-3</b> Détection d'objets de la même classe que le <i>template</i> à différents seuils de coefficients de corrélation dans une image bruitée. ....	58
<b>Figure-VIII-1</b> Calcul direct vs calcul indirect du champ de mouvement d'un objet mobile. ....	60
<b>Figure-VIII-2</b> Illustration de la méthode HARRIS pour extraire les corners de l'objet à détecter. ....	61
<b>Figure-VIII-3</b> Schéma synoptique pour la méthode the SIFT appliqué au <i>template</i> . ....	62
<b>Figure-VIII-4</b> <i>Template</i> ou objet à extraire ses points caractéristiques pour une détection. ....	64
<b>Figure-VIII-5</b> Extraction des points caractéristiques (corners) par HARRIS. ....	65
<b>Figure-VIII-6</b> Extraction des points caractéristiques (corners) par HARRIS de l'objet subissant une rotation. ....	65
<b>Figure-VIII-7</b> Image « eight.tif ». ....	66
<b>Figure-VIII-8</b> HARRIS appliquée sur l'Image « eight.tif » pour extraire ses points caractéristiques. ....	66
<b>Figure-VIII-9</b> Trame extraite d'une vidéo de surveillance routière. ....	67
<b>Figure-VIII-10</b> Soustraction entre trames de la vidéo éliminant l'arrière plan (objets fixes) de la vidéo. ....	68
<b>Figure-VIII-11</b> Elimination de bruits par opération morphologique (érosion). ....	68
<b>Figure-VIII-12</b> Trame représentative de traitement de la vidéo : Véhicule détectés utilisant l'algorithme de détection indirecte SIFT. ....	69

## NOMENCLATURES

CC: Coefficient de Corrélation

CSC: Code de Structure de Couleur

FFT: *Fast Fourier Transform*

*Gamma* : Gain de rehaussement

Id : image dégradée

IFFT: *Inverse Fast Fourier Transform*

Ir : image rehaussée

SIFT: *Scale-Invariant Features Transform*

TF : Transformée de Fourier

TFD: Transformée Discrète de Fourier

Th : Seuil de corrélation pour les méthodes directes de détection (*Threshold*)

$\delta$  : Paramètre de largeur de bande du filtre exponentiel

$\nabla$  : Opérateur de Gradient

$\Delta$  : Opérateur de Laplacien

$\sigma$  : Paramètre de largeur de bande du filtre Gaussien



## Introduction générale :

Ce polycopié de "TP-vision Artificielle" est dédié pour les étudiants en Master-2, spécialité électronique des systèmes embarqués. Le but de ce polycopié de TPs est, dans un premier temps, le traitement numérique de l'image à base d'outils mathématiques afin d'améliorer son apparence (filtrage, rehaussement du contraste ..etc).

Ensuite, il s'agit d'extraire les caractéristiques de l'image (segmentation d'images, détection de caractéristiques et reconnaissance d'objet..etc) afin d'en exploiter celle les plus importantes évitant ainsi toute redondance pour rendre plus possible la prise de décision en temps réel tel que pour l'estimation de mouvement d'objet mobile dans une séquence vidéo.

N.B : Touts les TPs, dans ce polycopié, sont présentés de la manière suivante : Rappel, But de TP, Travail demandé, solution du TP.

## I- TP 1 : Manipulation et amélioration d'images par traitement d'histogramme

### I-1 Rappel :

#### I-1-1 Vocabulaire :

La figure-1-1 et la figure-I-2 résument le vocabulaire d'une image numérique [2]. Figure-I-1 illustre l'espace de l'image sous forme d'un repère orthogonal. Chaque pixel est caractérisé par ses coordonnées spatiales et par sa valeur de niveau de gris allant du noir (valeur 0) jusqu'au blanc (valeur maximale selon la profondeur =  $2^{\text{nombre de bit selon lequel est codé un pixel}-1}$ ).

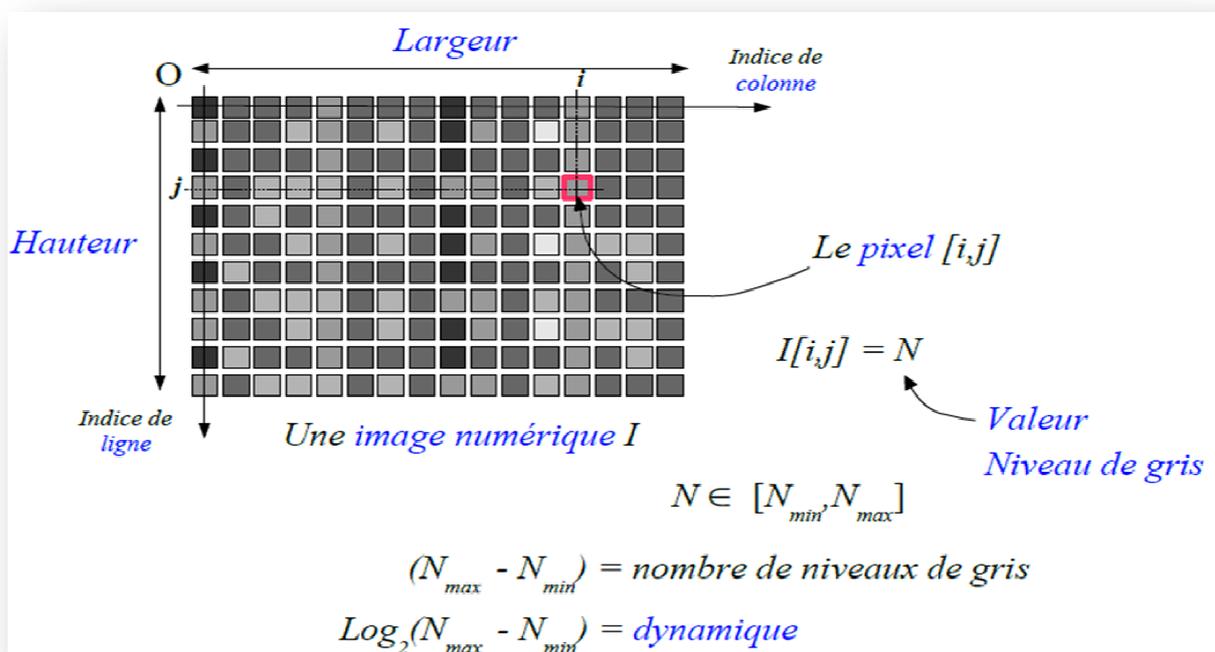


Figure-I-1 Vocabulaire de l'image numérique.

L'échantillonnage et la quantification étant les deux opérations essentielles de numérisation, la figure-I-2 illustre l'effet de l'échantillonnage sur la résolution spatiale de l'image et l'effet de la quantification (codage du pixel sur un nombre de bits) sur la résolution tonale.

*Résolution...*

*...spatiale :*

*Échantillonnage*



*...tonale :*

*Quantification*



**Figure-I-2** Résolution spatiale et résolution tonale.

**I-1-2 Amélioration de l'apparence d'image par traitement d'histogramme :**

Le type d'images concernées par le traitement à effectuer dans ce TP est les images présentant une information difficilement observable due aux conditions environnementales lors de la prise de vue telle que l'obscurité ou la faible luminosité. A notifier que les images bruitées et qui peuvent être aussi difficilement observables, ne sont pas considérées parmi les images cachant de l'information et ne font pas objet dans ce premier TP.

Généralement, la prise de décision, pour effectuer un traitement, nécessite au premier lieu un diagnostic correct et qui permettrait de décider de procéder un traitement adéquat remédiant au problème constaté lors du diagnostic visuel et analytique de l'image. Une image diagnostiquée visuellement contenant des informations cachées (difficilement observables) nécessite un deuxième diagnostic utilisant l'analyse de son histogramme afin de décider de quel traitement s'agit-il ?

La figure-I-3 illustre le concept d'histogramme permettant d'avoir une distribution des pixels selon leur niveau de gris (ou leurs couleurs pour une image couleur) [2].

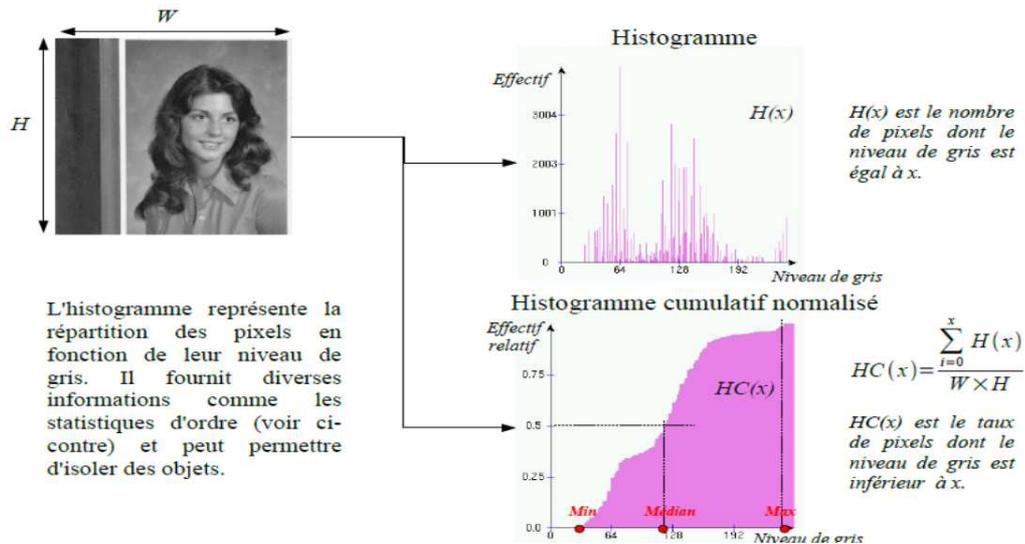


Figure-I-3 Schéma illustratif du concept d'histogramme pour une image.

Les images cachant de l'information dont il est question de diagnostiquer et de décider et effectuer le traitement adéquat présentent des histogrammes ayant généralement un des deux aspects suivants [2]:

- Ou bien des distributions localisées vers les faibles niveaux de gris concrétisées par une image sombre cachant la visibilité des structure de l'image nécessitant ainsi un traitement de normalisation au niveau de l'axe des abscisses de l'histogramme (celui des niveaux de gris) menant à une exploitation de toute la dynamique de l'axe des abscisses. De cette manière, la normalisation permettrait de faire réapparaître les structures visuellement cachées dans l'image.
- Ou des distributions relativement localisées majoritairement vers une bande étroite des niveaux de gris ce qui donne un aspect de pic au niveau des effectifs (axe des éloignements de l'histogramme). Une redistribution adéquate des effectifs est nécessaire afin de remédier à l'aspect majoritairement identique dans l'image. Ce traitement de redistribution des effectifs s'appelle l'égalisation.

Afin de décider quel est le traitement nécessaire pour une image présentant des informations visuellement cachées, l'analyse de son histogramme est primordiale pour une décision de traitement par normalisation ou égalisation.

Les figures-I-4 et I-5 illustrent un diagnostic et traitement successivement par normalisation et par égalisation. A noter qu'une égalisation nécessite l'utilisation de l'histogramme cumulé (figure-I-3).

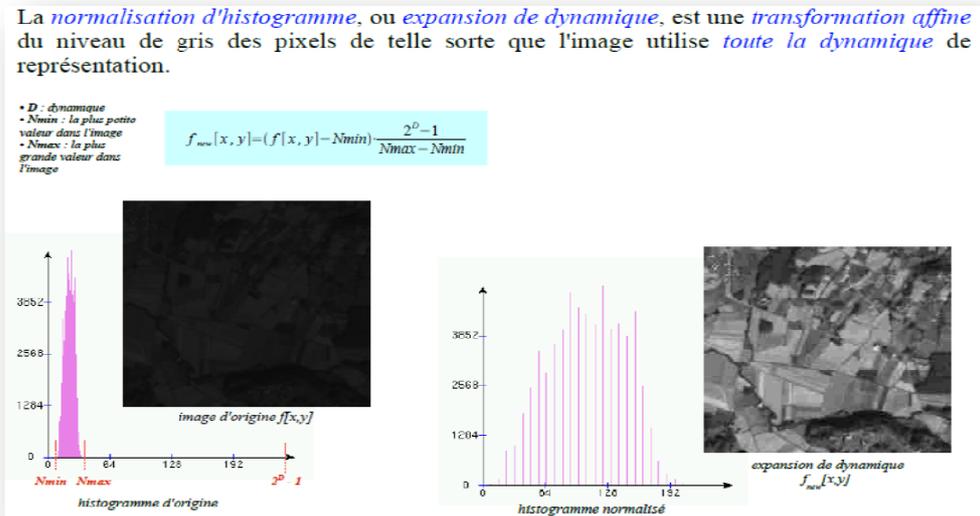


Figure-I-4 Schéma illustratif du diagnostic et de traitement par normalisation.

Une image normalisée (figure-I-4) permet d'exploiter toute la dynamique (tous les niveaux de gris) faisant ainsi améliorer l'apparence de l'image.

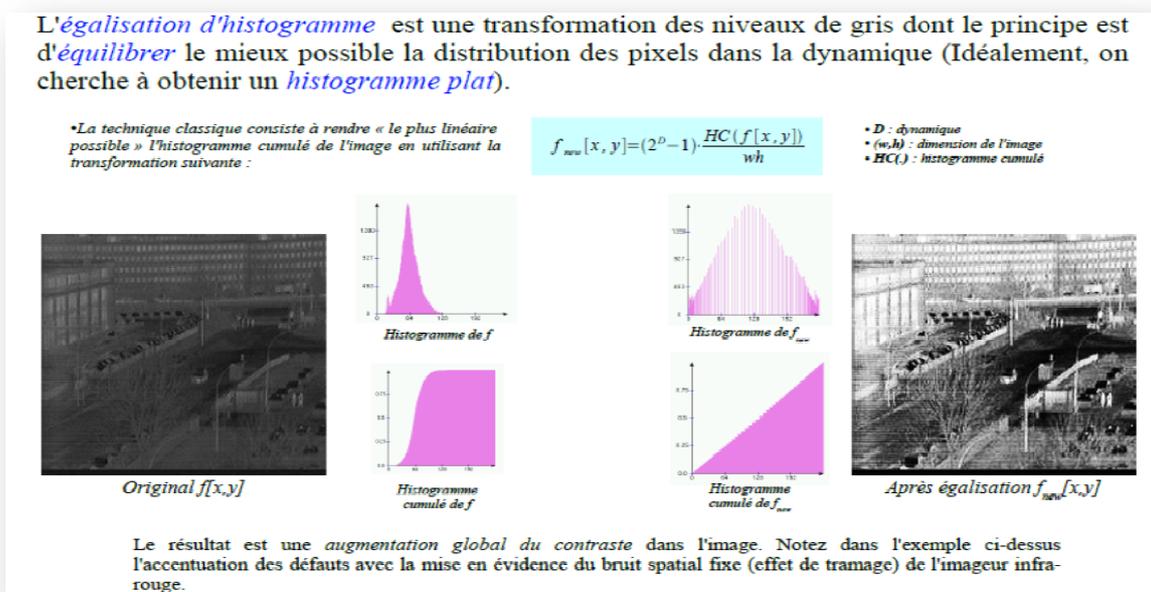


Figure-I-5 Schéma illustratif du diagnostic et de traitement par égalisation.

Une image égalisée (figure-I-5) permet de redistribuer les effectifs en remédiant au pic des effectifs dû à l'aspect identique dans l'image.

### **I-2 But de TP-1:**

- Se familiariser avec les commandes Matlab® de lecture et d'affichage des images.
- Apprendre à manipuler les histogrammes des images dans le but d'apporter les corrections adéquates aux images cachant de l'information.

### **I-3 Travail demandé de TP-1 :**

1. En utilisant les commandes « imread », « imshow » et « subplot », charger les images « route.png » et « sat.png » et afficher les dans une même figure (Aidez vous du help du Matlab® pour apprendre à utiliser ces commandes).
2. Comment vous pouvez juger ces images en termes d'apparence et d'observabilité ?
3. En utilisant la commande « imhist », afficher les images précédentes avec leurs histogrammes respectifs dans une même figure (Pour l'affichage dans une même figure, utiliser la commande « subplot » pour subdiviser le plan de la figure).
4. Faites vos remarques concernant ces histogrammes et juger sur la nature du traitement nécessaire à l'amélioration de chaque image.
5. Appliquer le traitement adéquat pour chaque image en utilisant la commande « histeq » pour l'égalisation d'histogramme et la commande « imadjust » pour la normalisation.
6. Afficher les images et leurs histogrammes respectifs avant et après traitement (dans une même figure). Commenter et conclure ?
7. Tracer dans une même figure, les histogrammes cumulés des deux images avant et après traitement. Faites vos remarques.

#### I-4 Solution du TP-1 :

1- Chargement et affichages des deux images à diagnostiquer et à traiter :

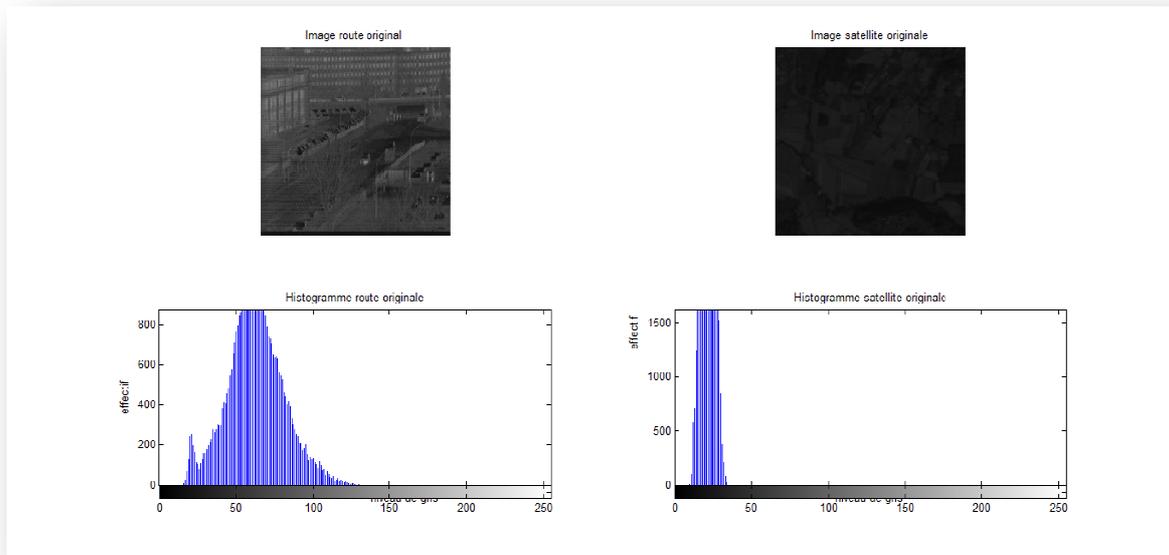


**Figure-I-6** Chargement et affichages des deux images à diagnostiquer et à traiter.

2- Diagnostic visuelle des deux images :

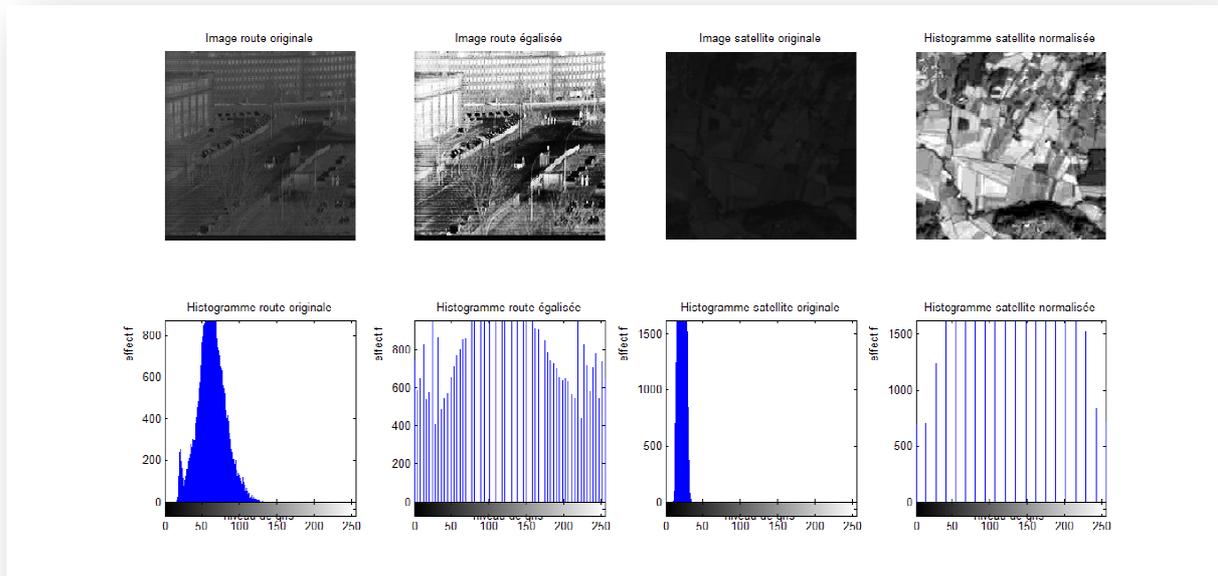
L'image satellite (figure-I-6) présente un aspect sombre ce qui rend difficile l'observation des structures. Tant disque l'image route (figure-I-6) présente un aspect identique (un certain niveau de gris majoritairement réparti sur toute l'image). Pour une décision de traitement par normalisation ou égalisation de ces deux images, l'analyse de leurs histogrammes devient nécessaire.

3- Les deux images à diagnostiquer et leurs histogrammes respectifs :



**Figure-I-7** Les deux images à diagnostiquer et leurs histogrammes respectifs.

- 4- A partir de la figure-I-7, étant que les deux images sont codées sur 8 bits (le maximum des niveaux de gris = 255), la répartition des niveaux de gris de l'image satellite est étroite et localisée vers les niveaux de gris sombre. Afin de mieux répartir les niveaux de gris de cette image, une normalisation sur celle-ci est nécessaire (paragraphe-I-1-2). Tant disque l'image route donne un histogramme présentant un pic indiquant qu'un effectif majoritaire prend une bande étroite de niveaux de gris causant ainsi un aspect identique sur toute l'image. Pour mieux répartir les effectifs sur les différents niveaux de gris de cette image, le traitement par égalisation est nécessaire (paragraphe-I-1-2).
- 5- Traitement de normalisation pour image satellite et d'égalisation pour image route et leurs histogrammes respectifs avant et après traitement :



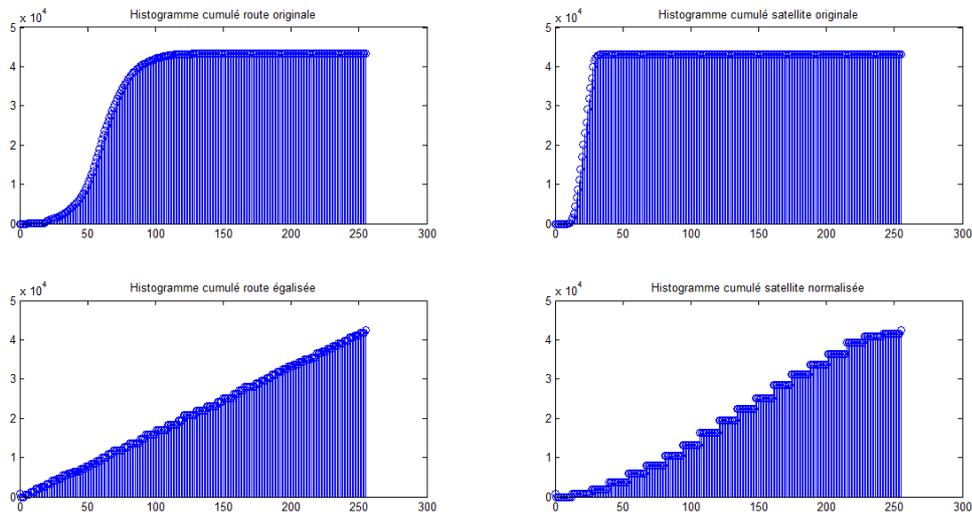
**Figure-I-8** Traitement de normalisation pour image satellite et d'égalisation pour image route et leurs histogrammes respectifs avant et après traitement.

6- Commentaires et conclusions :

L'image satellite (figure-I-8) présentant un aspect sombre donne un histogramme réparti sur une bande étroite des niveaux de gris et nécessite donc un traitement de normalisation. La normalisation de l'image satellite a permis d'exploiter toute la dynamique des niveaux de gris ce qui a permis d'améliorer l'apparence de l'image (figure-I-8). L'histogramme de l'image traitée (normalisée) explique l'effet, sur l'apparence améliorée de l'image, de l'exploitation de toute la dynamique des niveaux de gris (figure-I-8).

L'image route (figure-I-8) présentant un aspect identique donne un histogramme avec un effectif majoritaire d'une bande étroite des niveaux de gris (pic) et nécessite donc un traitement d'égalisation. L'égalisation de l'image route a permis de redistribuer (égaliser) les effectifs sur toute la dynamique des niveaux de gris ce qui a permis d'améliorer l'apparence de l'image (figure-I-8) tout en présentant différents niveaux de gris pour les différentes structures dans l'image (route, immeuble, arbre. etc). L'histogramme de l'image traitée (égalisée) explique l'effet, sur l'apparence améliorée de l'image, de la redistribution égalisée des effectifs sur toute la dynamique des niveaux de gris (figure-I-8).

7- Les histogrammes cumulés des deux images avant et après traitement.



**Figure-I-9** Histogrammes cumulés des deux images avant et après traitement.

Remarques et constatations sur les histogrammes cumulés des deux images avant et après traitement :

Avant traitement,

L'histogramme cumulé de l'image satellite se sature rapidement dès le niveau de gris 35 correspondant à la limite de la bande étroite (figure-I-5 et figure-I-9). Après cette limite, l'histogramme cumulé n'a pas d'effectif à cumuler et donc se sature. Tant que l'histogramme cumulé (figure-I-9) de l'image route présente une pente importante dans la bande [35 70] des niveaux de gris et qui correspond à la bande du pic d'effectif majoritaire de l'histogramme de cette image (figure-I-7).

Après traitement,

Etant normalisée exploitant ainsi toute la dynamique des niveaux de gris, l'image satellite normalisée donne un histogramme cumulé (figure-I-9) qui sature graduellement du noir jusqu'au niveau de gris maximal (le blanc). Tandis que l'image route donne un histogramme cumulé avec une pente quasi linéaire indiquant la redistribution égalisée des effectifs (figure-I-9).

## II- TP 2 : Filtrage par opération de convolution 2D

### II-1 Rappel :

#### II-1-1 filtrage par opération de convolution (filtrage linéaire) :

La convolution entre deux images est définie par analogie de la convolution de deux signaux monodimensionnels. C'est le produit dans l'espace entre le noyau de la convolution et la position actuelle correspondante dans l'image. Le noyau est déplacé pour balayer toute l'image. A chaque position, le pixel concerné par le filtrage (résultat de convolution) est le pixel central [1].

Soit  $I$  une image numérique.

Soit  $h$  une fonction de  $[x_1, x_2] \times [y_1, y_2]$  à valeurs réelles.

La *convolution* de  $I$  par  $h$  est définie par :

$$(I * h)[x, y] = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h[i, j] \cdot I[x-i, y-j]$$

#### Propriétés de la convolution :

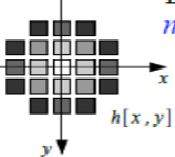
**COMMUTATIVITÉ**  $h * g = g * h$

**ASSOCIATIVITÉ**  $(h * g) * k = h * (g * k) = h * g * k$

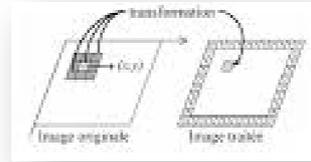
**DISTRIBUTIVITÉ**  $h * (g + k) = (h * g) + (h * k)$

La figure-II-1 illustre le concept de convolution 2D [1].

La fonction  $h$  est dite *noyau de convolution*



Les nouvelles valeurs du pixel sont calculées par *produit scalaire* entre le noyau de convolution et le voisinage correspondant du pixel.



La multiplication dans le domaine fréquentiel correspond à la convolution dans le domaine spatial. Un grand nombre de filtres de lissage peut être obtenu à partir de noyaux de convolution symétriques et normalisés (de somme égale à 1). Voici 3 famille de filtres parmi les plus utilisés :

*Moyenne*

\* Réponse impulsionnelle :

$$h(x, y) = \frac{1}{\lambda^2} \text{si } (x, y) \in [-\lambda/2, +\lambda/2]^2$$

$$h(x, y) = 0 \text{ sinon}$$

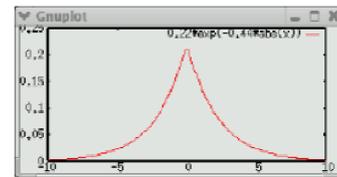
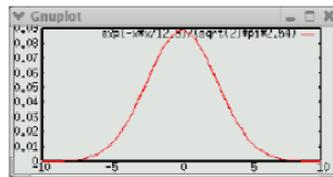
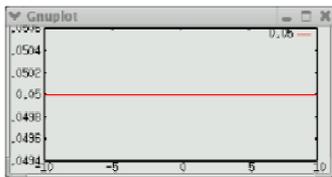
*Gauss*

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

*Exponentiel*

$$h(x, y) = \frac{\gamma^2}{4} \exp(-\gamma(|x| + |y|))$$

\* Représentation graphique de la réponse impulsionnelle (en 1d) :



\* Exemple de noyaux de convolution discrets :

$$\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Filtre moyenneur (5x5)

$$\frac{1}{864} \begin{pmatrix} 11 & 23 & 29 & 23 & 11 \\ 23 & 48 & 62 & 48 & 23 \\ 29 & 62 & 80 & 62 & 29 \\ 23 & 48 & 62 & 48 & 23 \\ 11 & 23 & 29 & 23 & 11 \end{pmatrix}$$

Filtre gaussien ( $\sigma = 1,41$ )

$$\frac{1}{80} \begin{pmatrix} 1 & 1 & 3 & 1 & 1 \\ 1 & 3 & 7 & 3 & 1 \\ 3 & 7 & 16 & 7 & 3 \\ 1 & 3 & 7 & 3 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{pmatrix}$$

Filtre exponentiel ( $\gamma = 0,8$ )

Figure-II-1 Noyaux et concept de convolution 2D.

La figure-II-2 illustre le résultat de filtrage par opération de convolution d'une image bruitée par trois différents noyaux : moyenneur, gaussien et exponentiel [1].

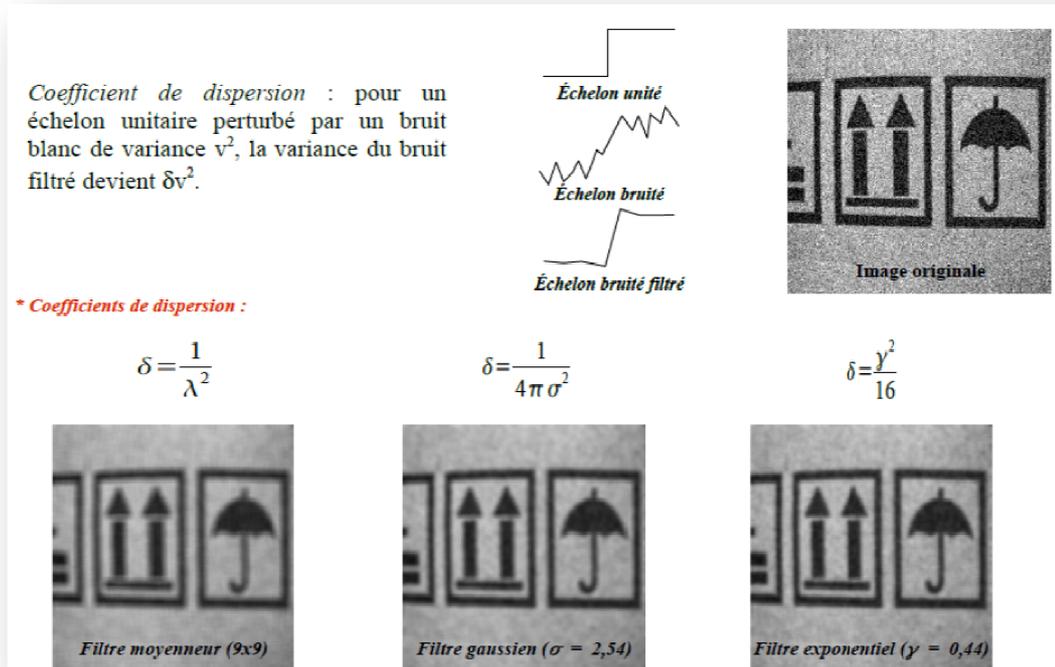


Figure-II-2 Filtrage par convolution 2D.

Nous pouvons constater que le filtre moyenneur en éliminant le bruit a causé un aspect flou (la moyenne rapproche les valeurs du voisinage du pixel central concerné par le filtrage). De ce fait, nous pouvons en déduire que le filtre moyenneur est adapté pour le bruit uniformément réparti sur l'image. Ce qui, souvent, n'est pas le cas.

Afin de remédier au problème du flou causé par l'opération de la moyenne, une pondération dans le noyau du pixel central est nécessaire. C'est le cas des deux noyaux de filtrage : Gaussien et exponentiel (figure-II-2) qui sont plus adaptés que le filtre moyenneur dans ce cas mais n'éliminent pas entièrement le bruit.

C'est parce que la convolution est une opération de filtrage linéaire : Les filtres linéaires éliminent mal les valeurs aberrantes.

### II-1-2 Filtrage non linéaire :

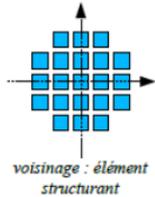
- Deux aspects du lissage sont concernés par le filtrage non linéaire :
- *Le bruit impulsif* : les filtres linéaires éliminent mal les valeurs aberrantes.
- *L'intégrité des frontières* : on souhaiterait éliminer le bruit sans rendre flous les frontières des objets.

#### II-1-2-1 Filtre médian :

- (a) Filtres d'ordre, médian
- (b) Filtres non linéaires divers – ex : Nagao
- (c) Filtres morphologiques

Dans le contexte ce TP, nous allons rappeler les filtres : d'ordre, médian et le filtre Nagao. Tandis que le filtrage morphologique (aussi non linéaire) sera entamé dans le quatrième TP.

Les filtres d'ordres procèdent en remplaçant les valeurs de chaque pixel par la valeur qui occupe *un certain rang* lorsqu'on trie les valeurs observées dans *un certain voisinage* du pixel.



les valeurs dans le voisinage de  $(x,y)$ :  $V(x,y) = \{a_1, a_2, \dots, a_N\}$

soit  $\{b_1, b_2, \dots, b_N\}$  permutation de  $\{a_1, a_2, \dots, a_N\}$  telle que  $b_1 \leq b_2 \leq \dots \leq b_N$

alors le filtre d'ordre de rang  $k$  est défini par :  $\rho_k[x,y] = b_k$

pour  $k=N/2$ , on parle de **filtre médian**, pour  $k=1$ , d'**érosion morphologique**, pour  $k=N$ , de **dilatation morphologique**.

*Implantations du médian :*

- calcul d'histogrammes locaux
- tri des valeurs dans le voisinage (Quick Sort)
- tri incrémental
- /

La figure-II-3 présente une image affectée par un bruit de type « salt & pepper » («sel et poivre » ; des tâches blanches et noires).

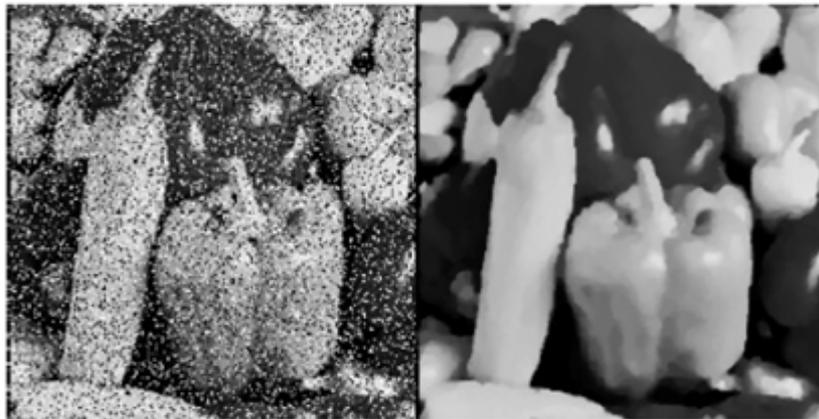


Figure-II-3 Filtrage médian sur image affectée par un bruit impulsif.

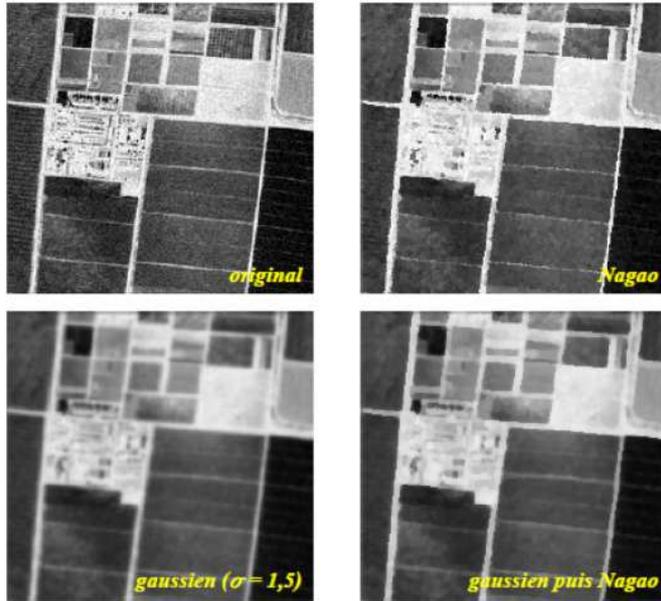
Le filtre médian est utilisé dans ce cas parce que c'est le filtre le plus adéquat pour ce type de bruit par le fait qu'il considère le rang du médian et met successivement les tâches noires et blanches dans les rangs minimaux et maximaux du filtre d'ordre.

### II-1-2-2 Filtre de Nagao :

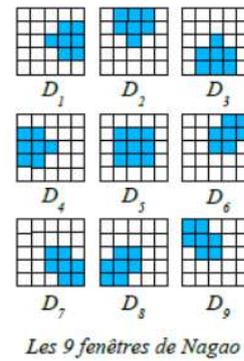
Nous rappelons que le filtre linéaire moyenneur cause un flou lors de l'élimination du bruit. Une solution linéaire était (paragraphe II-1-1) de procéder à pondérer l'élément central du noyau du filtrage tel que le cas des filtres Gaussien et Exponentiel (figure-II-1 et figure-II-2).

Une solution non linéaire de ce problème est le filtre de Nagao.

On trouve dans la littérature de nombreux filtres combinant *filtres d'ordre*, *moyennes robustes* (opérations linéaires éliminant les valeurs marginales), et *anisotropie* (le support des opérations s'adapte en fonction des frontières locales). Nous décrivons ici comme exemple le *filtre de Nagao*.



Le filtre de Nagao examine la fenêtre 5x5 centrée sur chaque pixel. 9 domaines sont définis dans cette fenêtre (voir figure). On calcule pour chaque domaine  $D_i$  la moyenne  $\mu_i$  et la variance  $v_i$ . Le résultat de l'opérateur est la moyenne du domaine qui présente la plus faible variance.



**II-2 But de TP-2:**

- Se familiariser avec la notion de convolution.
- Apprendre à manipuler les différents noyaux des filtres linéaires et les appliquer pour des opérations de filtrage.

**II-3 Travail demandé de TP-2:**

- 1) En utilisant les commandes « imread » et « imshow », charger et afficher l'image « eight.tif ».
- 2) En vous aidant du help et en utilisant la commande « imnoise », ajouter du bruit du type « salt & pepper », de densité de 0.02, à l'image précédente et afficher les images originale et bruitée dans une même figure.

Nous souhaitons maintenant filtrer **par opération de convolution** l'image bruitée. Pour cela on utilise trois filtres différents (filtre moyennneur, filtre gaussien et filtre exponentiel) :

$\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ <p><i>Filtre moyennneur (5x5)</i></p>	$\frac{1}{864} \begin{pmatrix} 11 & 23 & 29 & 23 & 11 \\ 23 & 48 & 62 & 48 & 23 \\ 29 & 62 & 80 & 62 & 29 \\ 23 & 48 & 62 & 48 & 23 \\ 11 & 23 & 29 & 23 & 11 \end{pmatrix}$ <p><i>Filtre gaussien (<math>\sigma = 1,41</math>)</i></p>	$\frac{1}{80} \begin{pmatrix} 1 & 1 & 3 & 1 & 1 \\ 1 & 3 & 7 & 3 & 1 \\ 3 & 7 & 16 & 7 & 3 \\ 1 & 3 & 7 & 3 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{pmatrix}$ <p><i>Filtre exponentiel (<math>\gamma = 0,8</math>)</i></p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 3) Créer dans Matlab® les noyaux 5X5 des trois filtres.
- 4) En utilisant la commande « conv2 », calculer le produit de convolution de l'image bruitée avec les trois noyaux de filtres. Afficher les résultats de filtrage avec l'image bruitée et l'image originale dans la même figure. Comparer les résultats de filtrage.
- 5) Utiliser la commande « medfilt2 » pour filtrer l'image bruitée avec le filtre médian. Afficher le résultat du filtrage dans la même figure précédente et comparer ce résultat avec les autres.
- 6) Appliquer d'autres types de bruit (bruit gaussien et bruit de poisson) à l'image et refaire le filtrage par les quatre noyaux définis précédemment. Faites vos remarques concernant la fiabilité de chaque filtre pour les différents types de bruit.

#### **II-4 Solution du TP-2 :**

- 1- Chargement et affichage de l'image :



**Figure-II-4** image "eight.tif"

- 2- Affichage de l'image originale avec l'image bruitée :

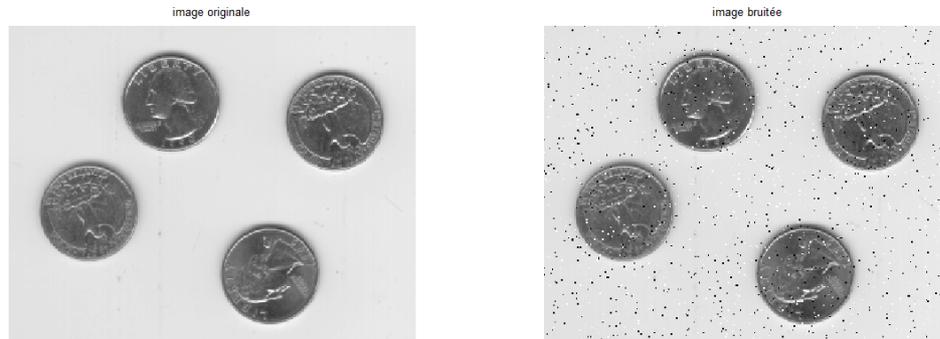


Figure-II-5 Image et image affectée par un bruit de type « salt & pepper » de densité de 0.02.

3- Création des trois noyaux de filtrage :

```

%%moyenneur
fm=ones(5,5)/25;%filtre moyenneur
%%filtre Gaussien
gs=[11 23 29 23 11;23 48 62 48 23;29 62 80 62 29;23 48 62 48 23;11 23 29
23 11]/864;
%%filtre exponentiel
ex=[1 1 3 1 1;1 3 7 3 1;3 7 16 7 3;1 3 7 3 1;1 1 3 1 1]/80;

```

4- Calcul de convolution de l'image bruitée avec les trois noyaux de filtrage, affichage de résultats et comparaisons :

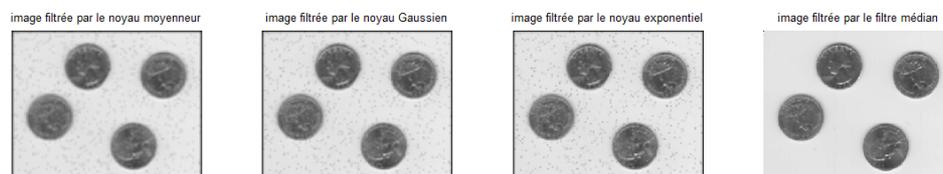


Figure-II-6 Résultats de filtrage linéaire (par convolution).

A partir de la figure-II-6, à part le problème de bords non traité de l'image et qui est dû à ce que les bords de l'image ne sont pas considérés par le filtrage par le fait que ces pixels ne peuvent pas être au centre du noyau lors de son balayage sur l'image pour effectuer la convolution, un autre problème bien attendu a surgit est que les trois noyaux de filtrage non pas pu éliminer totalement le bruit et cela revient à la nature du bruit non linéaire qui nécessite un filtrage non linéaire plus adéquat afin d'aboutir un résultat de filtrage plus satisfaisant.

A noter que le filtre moyenneur a causé un flou sur l'image car l'opération de la moyenne rapproche la valeur du pixel central concerné par le filtrage à son voisinage ce qui donne un aspect de flou sur l'image filtrée. Tandis que le noyau Gaussien et le noyau exponentiel ont pu éviter le flou en procédant à la pondération (figure-II-1) du pixel central par rapport à son voisinage.

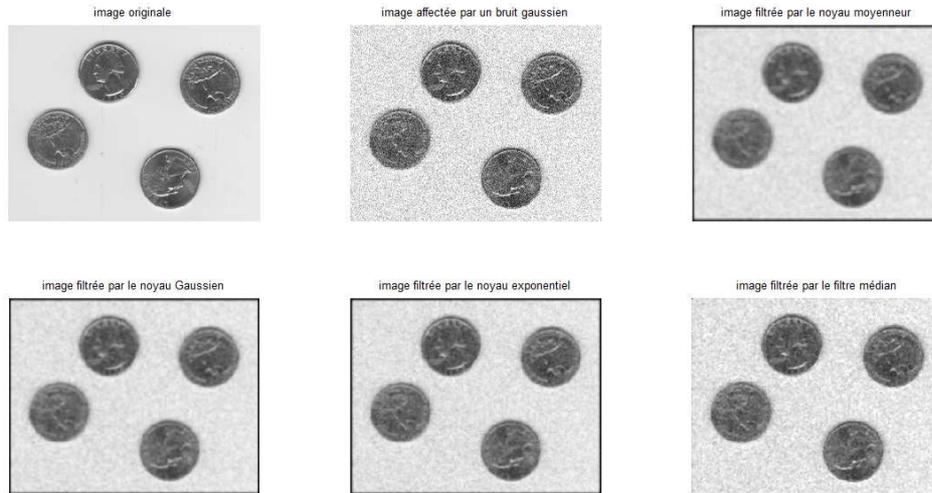
5- Filtrage médian (non linéaire) et sa comparaison avec le filtrage linéaire (précédant) :



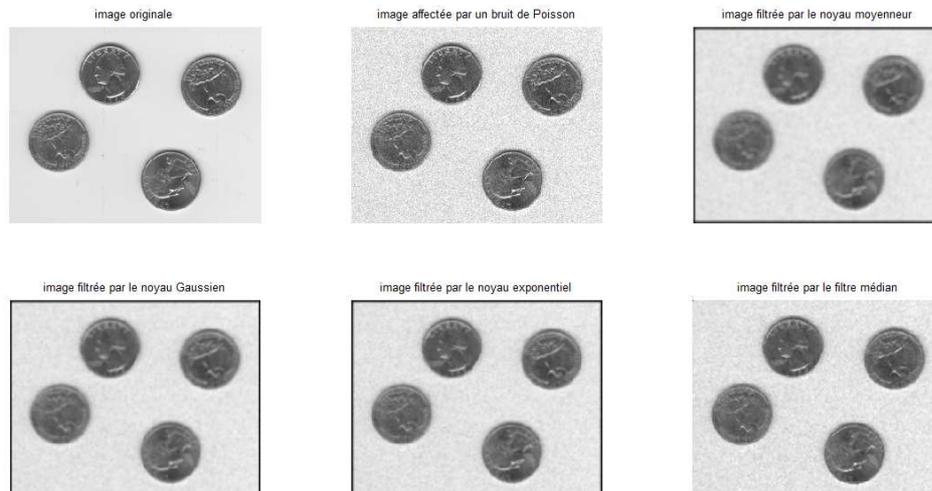
**Figure-II-7** Résultats de filtrage non linéaire (médian) comparé aux noyaux de filtrage linéaire.

La figure-7 illustre le résultat attendu du filtrage non linéaire (médian) envers le bruit non linéaire « salt & pepper ». Le filtre médian, au contraire du filtrage linéaire par convolution (moyenneur, Gaussien, exponentiel) a pu éliminer le bruit. Cela revient principalement à ce que le filtre médian considère la médiane du filtre d'ordre pour substituer la valeur du pixel bruité et tant que le bruit est sous forme de tâches noires et blanches, celles-ci seront localisées successivement dans les rangs minimales et maximales du filtre d'ordre et ne seront jamais localisées dans la médiane du filtre d'ordre.

6- Appliquons d'autres types de bruit (bruit gaussien et bruit de poisson) à l'image et refaisons le filtrage par les quatre noyaux définis précédemment. Faisons nos remarques concernant la fiabilité de chaque filtre pour les différents types de bruit :



**Figure-II-8** Résultats de filtrage linéaire et non linéaire sur une image affectée par un bruit blanc gaussien.



**Figure-II-9** Résultats de filtrage linéaire et non linéaire sur une image affectée par un bruit blanc de Poisson.

Un bruit blanc est une réalisation d'un processus aléatoire dans lequel la densité spectrale de puissance est la même pour toutes les fréquences. Autrement dit, le bruit blanc affecte toutes les fréquences et son élimination par filtrage (par élimination d'une bande de fréquences) n'est pas



possible. Ce qui explique qu'aucun des deux filtrages (linéaire et non linéaire) n'a pu éliminer totalement le bruit (figure-II-8 et figure-II-9).

Pour une telle problématique (les fréquences du signal utile et les fréquences indésirables sont dans la même gamme), un autre axe récent de recherche procède à localiser les fréquences dans l'espace 2D et ensuite séparer entre les fréquences utiles et les fréquences du bruit, bien qu'elles soient de même nature, en utilisant des connaissances préalables et des outils mathématiques de séparation telle que la méthode Delaunay de triangulation [7], [8].

### III- TP 3 : Filtrage dans le domaine fréquentiel

#### III-1 Rappel :

#### III-1-1 La Transformée de Fourier Discrète bidimensionnelle :

La transformée de Fourier permet la décomposition d'un signal  $f$  en *combinaison linéaire de sinusoïdes complexes*, dont les coefficients  $F[u,v]$  dit *coefficients de Fourier*, fournissent des informations sur les *fréquences*  $(u,v)$  et permettent des manipulations dans le *domaine fréquentiel*.

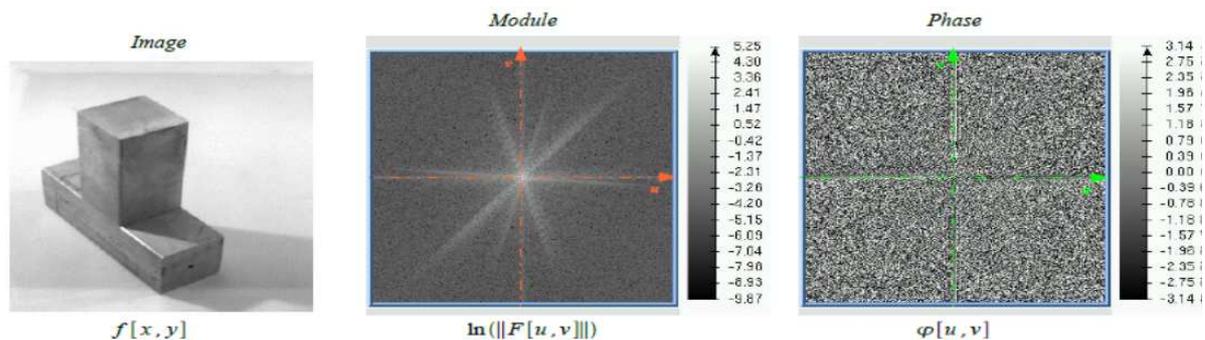
*Transformée de Fourier discrète bidimensionnelle :*

<p><math>(x,y)</math> sont les coordonnées du <i>domaine spatial</i></p>	<p><i>Directe :</i></p> $F[u, v] = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} f[x, y] e^{-2i\pi(ux+vy)/wh}$
<p><math>(u,v)</math> sont les coordonnées du <i>domaine fréquentiel</i></p>	<p><i>Inverse :</i></p> $f[x, y] = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} F[u, v] e^{2i\pi(ux+vy)/wh}$

Propriétés de la transformée de Fourier (1) :

<b>ÉCRITURE SOUS FORME MODULE / PHASE</b>	$F[u, v] = \ F[u, v]\  e^{i\varphi[u, v]}$
<b>PÉRIODICITÉ</b>	$F[u, v] = F[u+w, v+h]$
<b>SYMÉTRIE</b>	Si $F$ est la transformée de Fourier d'une fonction réelle $f$ : $F[u, v] = \overline{F[-u, -v]}$ et donc : $\ F[u, v]\  = \ F[-u, -v]\ $ et $\varphi[u, v] = -\varphi[-u, -v]$

La Transformée de Fourier Discrète bidimensionnelle, étant définie par analogie avec la Transformée de Fourier Discrète des signaux monodimensionnels, préserve les mêmes caractéristiques et qui sont résumées sur la figure-III-1 [1].



Propriétés de la transformée de Fourier (2) :

$\text{si } \begin{cases} f[x, y] \xrightarrow{\text{TF}} F[u, v] \\ f_1[x, y] \xrightarrow{\text{TF}} F_1[u, v] \\ f_2[x, y] \xrightarrow{\text{TF}} F_2[u, v] \end{cases}$	<b>LINEARITÉ</b> $a \cdot f_1[x, y] + b \cdot f_2[x, y] \rightarrow a \cdot F_1[u, v] + b \cdot F_2[u, v]$
<b>CORRESPONDANCE CONVOLUTION / PRODUIT</b> $f_1[x, y] * f_2[x, y] \rightarrow F_1[u, v] \cdot F_2[u, v]$ $f_1[x, y] \cdot f_2[x, y] \rightarrow F_1[u, v] * F_2[u, v]$	<b>TRANSLATIONS SPATIALES / FRÉQUENTIELLES</b> $f[x-x', y-y'] \rightarrow F[u, v] \cdot e^{-2i\pi(u'x'+v'y')/wh}$ $f[x, y] \cdot e^{2i\pi(u'x'+v'y')/wh} \rightarrow F[u-u', v-v']$
<b>DÉRIVATION</b> $\frac{\partial f[x, y]}{\partial x} \rightarrow iuF[u, v]$ et $\frac{\partial f[x, y]}{\partial y} \rightarrow ivF[u, v]$	<b>THÉORÈME DE PARSEVAL</b> $\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} \ f[x, y]\ ^2 = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} \ F[u, v]\ ^2$

**Figure-III-1** Caractéristiques de la TFD bidimensionnelle.

### III-1-2 Filtrage dans le domaine de Fourier :

- Dans le TP précédent (TP-2), le filtrage dans le domaine spatial de l'image s'effectue par l'opération de convolution. Dans ce TP, l'opération équivalente dans le domaine de Fourier et le produit du spectre de l'image (voir les caractéristiques de la TFD bidimensionnelle dans la figure-III-1) par un masque de filtrage.

La figure-III-2 illustre une opération de masquage (filtrage par élimination d'une bande de fréquences spatiales). Cette opération est effectuée en multipliant le spectre de l'image par un masque centré (étant donné que le spectre périodique de l'image est réordonné de telle façon à avoir les origines des périodes du spectre au centre de l'espace) [1].

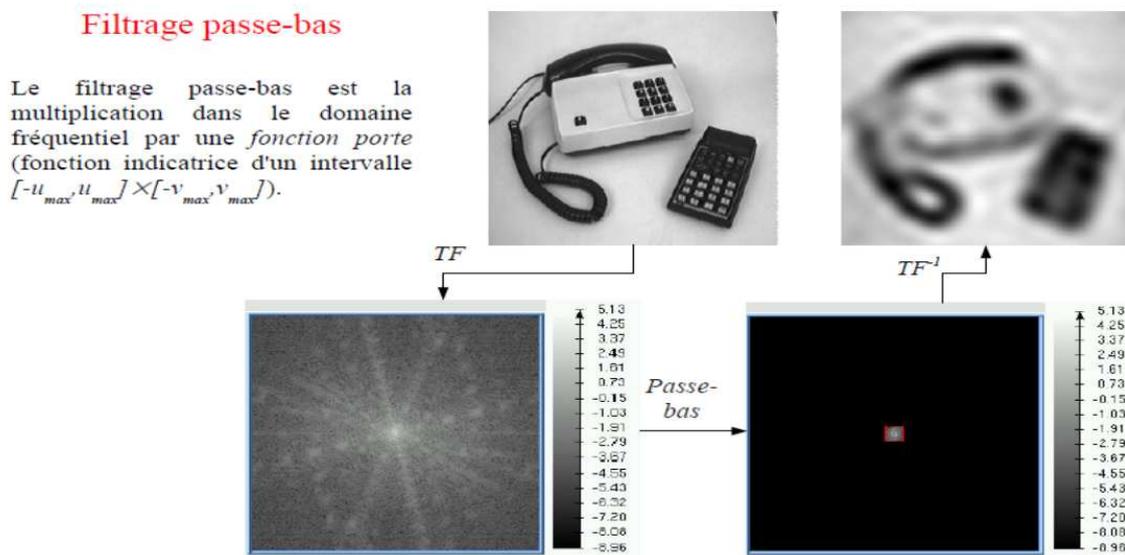


Figure-III-2 Filtrage par masquage utilisant la TFD bidimensionnelle.

A partir de la figure-III-2, nous constatons que la transformée inverse du spectre masqué par un masque passe-bas donne une image spatiale présentant les structure générales et approximations de l'image et perd les détails spatiaux fins tel que les contours des objets et les boutons de la calculatrice et du téléphone dans cet exemple.

A partir de ce résultat, nous pouvons retenir que les **basses fréquences** spatiales correspondent aux structures **générales** et aux **approximations** dans l'image. De même, nous pouvons retenir que les **hautes fréquences** spatiales éliminées lors du filtrage passe-bas correspondent bien aux structures éliminées dans l'image et qui sont les **détails spatiaux** et les **contours** dans l'image.

### III-2 But de TP-3 :

Etant donné que le TP précédent portait sur le filtrage de l'image utilisant la convolution avec un noyau dans le domaine spatial, ce TP porte sur le filtrage équivalent dans le domaine de Fourier par le produit scalaire avec un masque.

- De ce fait, le but de ce TP est, dans un premier temps, d'apprendre à réaliser les différents types de filtrage par masque dans le domaine fréquentiel et montrer la nature des basses et hautes fréquences spatiales d'une image en analysant l'effet de leur élimination.
- Ensuite, dans une deuxième partie, le filtrage dans le domaine de Fourier est utilisé afin de traiter des images bruitées.

### III-3 Travail demandé de TP-3:

#### Partie-A :

On veut maintenant analyser l'effet d'élimination de certaines fréquences spatiales sur les images afin de montrer leur nature. Pour cela on vous demande de :

1. Charger et afficher l'image « phone.png ». Essayer, par observation, de reconnaître les régions d'approximation et les détails dans cette image.
2. En utilisant la commande « fft2 », calculer et afficher, dans la même figure, le spectre en fréquence de cette image.
3. En utilisant la commande « fftshift » Réordonner les fréquences spatiales du spectre de l'image de telle sorte à avoir la fréquence 0 au centre du spectre. Réafficher le spectre réordonné dans la même figure.
4. On voudrait créer un masque de filtre passe-haut, de diamètre  $D=10$ , pour éliminer les basses fréquences de cette image. Voici la séquence sur Matlab permettant de créer ce masque :

```
[x y] = size(ft);% ici ft est le spectre de l'image
D = 10;
mask = fspecial('disk', D) == 0;
mask = imresize(padarray(mask, [floor((x/2)-D) floor((y/2)-D)], 1, 'both'), [x y]);
```

Afficher le masque dans la même figure.

5. Appliquer le filtre précédent sur le spectre réordonné de l'image et visualiser, dans la même figure, le spectre masqué (filtré) ainsi que l'image filtrée en utilisant la commande de transformée de Fourier inverse « ifft2 ». Analyser l'effet d'élimination des basses fréquences spatiales sur l'image.
6. Changer la nature du masque en passe-bas pour analyser l'effet d'élimination des hautes fréquences spatiales sur l'image.

#### Partie-B :

On veut maintenant appliquer le filtrage dans le domaine de Fourier pour but de traiter des images bruitées. Pour cela :

1. Ajouter un bruit de type « gaussian », écart type=0.01, sur l'image précédente.
2. Concevoir un masque convenable (en diamètre et en type) pour filtrer cette image.
3. Appliquer ce masque sur le spectre de l'image bruitée et afficher l'image bruitée, son spectre, le masque, le spectre masqué et l'image filtrée dans la même figure.
4. Donner vos remarques et commentaires.

### III-4 Solution du TP-3:

#### Partie-A :

- 1) Chargeons et affichons l'image « phone.png ». Ensuite, par observation, reconnaissons les régions d'approximation et les détails dans cette image :

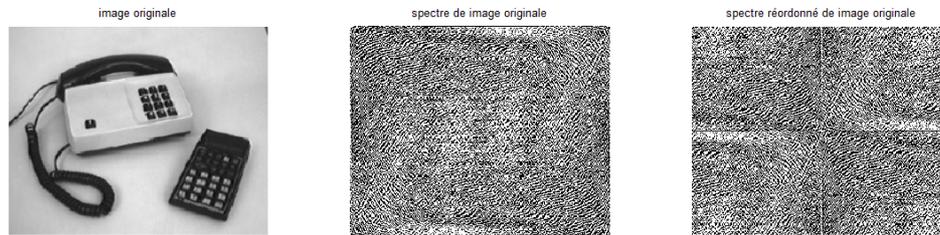
image originale



**Figure-III-3** Chargement et affichage de l'image « phone.png ».

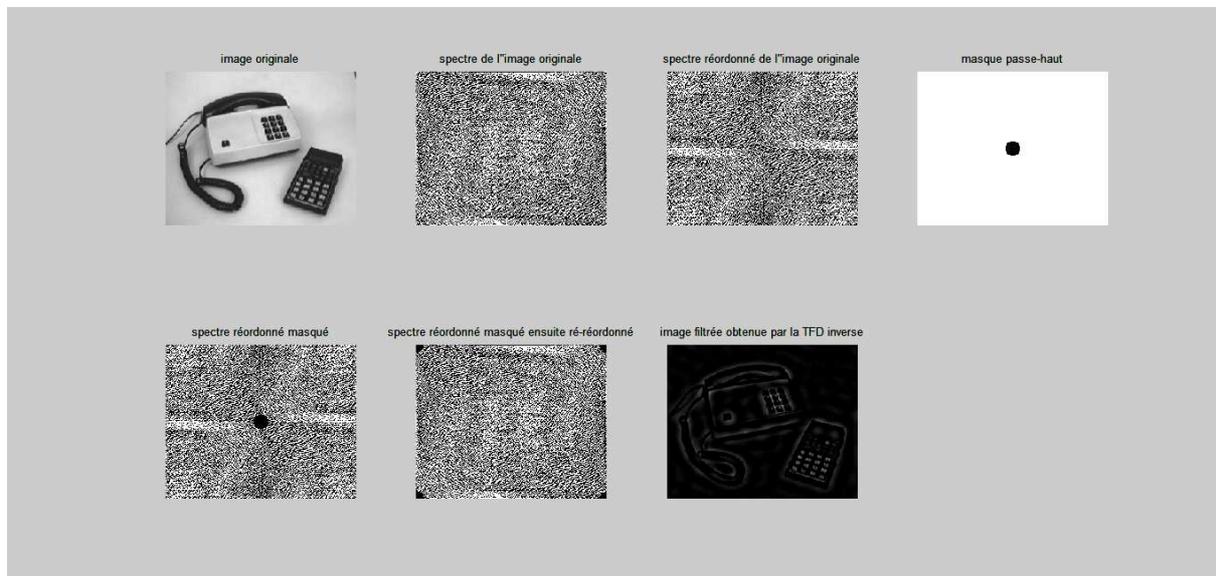
A partir de l'analyse visuelle de l'image de la figure-III-3, nous pouvons reconnaître les régions d'approximation et qui sont dans cet exemple : La surface sur laquelle sont déposés les objets, les deux boîtiers de la calculatrice et du téléphone (sans boutons et sans contours), le câble du téléphone (sans spirales). Tandis que les détails spatiaux dans cet exemple sont : les contours des objets, les spirales du câble de téléphone, les boutons du téléphone et de la calculatrice.

- 2) et 3) Calcul et affichage du spectre de l'image ainsi que le spectre réordonné de telle sorte à avoir la fréquence 0 au centre du spectre :



**Figure-III-4** Spectre et spectre réordonné de l'image « phone.png ».

4)- et 5)- Création et affichage du masque passe-haut ainsi que son application sur le spectre réordonné et affichage du spectre réordonné masqué. Ensuite ré-réordonné le spectre réordonné masqué pour obtenir le spectre filtré par un masque passe-haut. Enfin, Analysons l'effet d'élimination des basses fréquences spatiales sur l'image :



**Figure-III-5** Etapes de filtrage passe-haut sur l'image « phone.png ».

La figure-III-5 illustre les étapes de filtrage ainsi que l'effet de l'élimination des basses fréquences et de la préservation des hautes fréquences (filtrage passe-haut). L'image filtrée obtenue ne présente que les détails spatiaux fins et les contours ce qui indique que celles-ci correspondent aux hautes fréquences spatiales passées lors du filtrage passe-haut. D'autre part, ce filtrage passe-haut a éliminé les régions d'approximation reconnues dans la question-1 (le noir dans l'image filtrée) ce qui signifie que ces régions d'approximations correspondent bien aux basses fréquences spatiales éliminées lors du filtrage passe-haut.

6)- Changeons la nature du masque en passe-bas pour analyser l'effet d'élimination des hautes fréquences spatiales sur l'image.

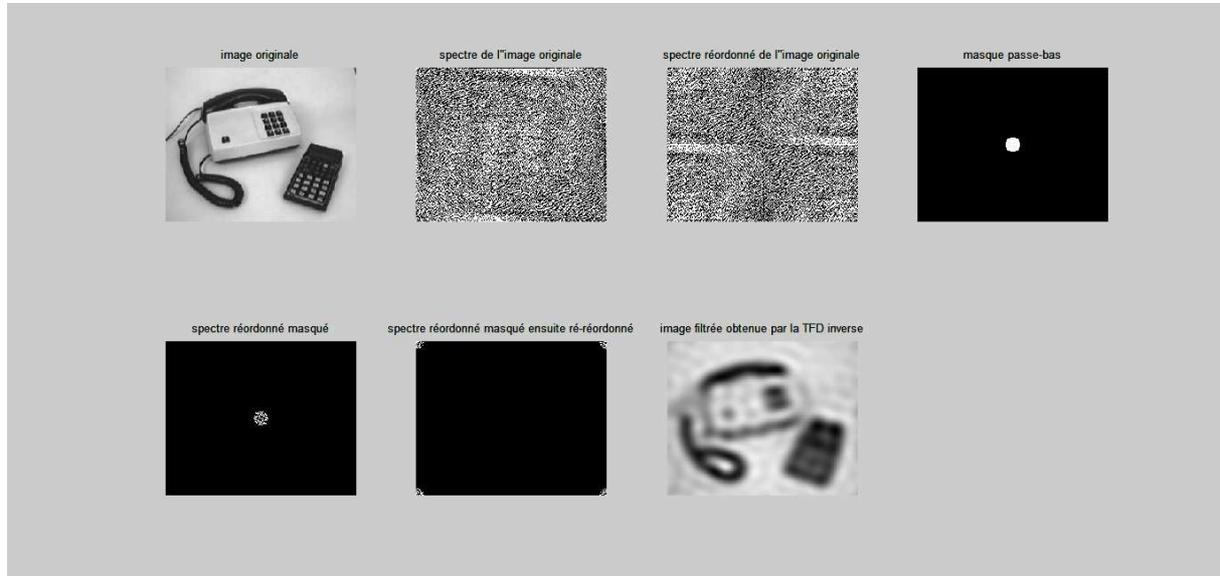


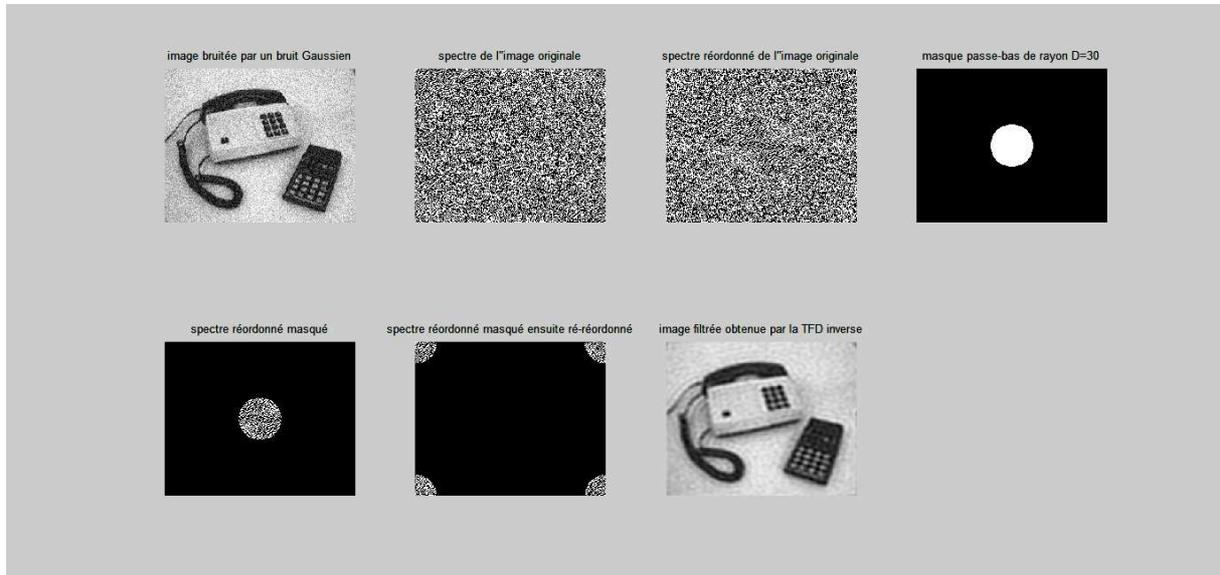
Figure-III-6 Etapes de filtrage passe-bas sur l'image « phone.png ».

Comme il a été attendu, la figure-III-6 confirme la nature, en déduites précédemment lors du filtrage passe haut, des basses fréquences et des hautes fréquences spatiales. Le filtrage passe-bas, en laissant passer les basses fréquences, préserve les approximations reconnues lors de la question-1 et éliminent les détails spatiaux et les contours correspondant aux hautes fréquences spatiales éliminées par le filtrage passe-bas.

**Partie-B :**

1)- et 2)- et 3)- et 4)-

On veut maintenant appliquer le filtrage dans le domaine de Fourier pour but de traiter des images bruitées. Pour cela. La figure-III-7 illustre un filtrage passe-bas de rayon  $D=30$  (plusieurs fréquences de coupure ont été essayées et  $D=30$  a été adopté) appliqué sur l'image affectée par un bruit blanc Gaussien.



**Figure-III-7** Etapes de filtrage passe-bas (rayon  $D=30$ ) sur l'image « phone.png » affectée par un bruit blanc Gaussien.

Comme il a été mentionné lors du TP précédent (filtrage par convolution), le bruit blanc affecte toutes les fréquences et son élimination entière par filtrage (par élimination d'une bande de fréquences spatiales) n'est pas possible. On ne peut, par filtrage du bruit blanc, qu'éliminer une partie des fréquences indésirables (réduction de l'effet de bruit) avec élimination des fréquences de l'information utile de la même bande. Pour une telle problématique, un autre axe récent de recherche procède à localiser les fréquences dans l'espace 2D et ensuite effectuer une séparation entre les fréquences utiles et les fréquences du bruit de la même bande en se basant sur des connaissances préalables et utilisant des méthodes mathématiques de séparation [7], [8].

---

## IV- TP 4 : Filtrage morphologique

---

### **IV-1 Rappel :**

Le filtrage morphologique est un filtrage non linéaire qui utilise le filtre d'ordre dans ses différentes opérations. Selon le diagnostic de l'image, Si elle nécessite de séparer les formes dans l'image, elle doit subir l'un des traitements morphologiques suivants: érosion morphologique, dilatation morphologique, ouverture morphologique ou fermeture morphologique [1].

Comme il a été évoqué lors du TP-2, le filtre d'ordre est un noyau qui prend ses valeurs depuis sa position actuelle dans l'image lors de son balayage. Ensuite ces valeurs seront ordonnées. Chaque opération morphologique utilisant le filtre d'ordre doit remplacer le pixel de l'image superposant au centre du noyau par un certain rang du filtre d'ordre pour sa position actuelle lors du balayage.

**IV-1-1 L'érosion morphologique** : Cette opération considère le rang minimale du filtre d'ordre et est utilisée pour injecter des niveaux de gris de faible rang (tel que le noir) afin de séparer des structures de niveaux de gris important tel que le blanc.

**IV-1-2 L'ouverture morphologique** : Lors de l'érosion morphologique, l'injection du rang faible entre les structures de niveaux de gris importants pour les séparer peut engendrer des déformations de leurs formes. Pour remédier à cette éventuelle déformation, l'érosion doit être suivie par un filtrage d'ordre supérieur (proche du rang maximal). Ce processus (érosion suivie par filtrage d'ordre supérieur) s'appelle l'ouverture morphologique.

**IV-1-3 La dilatation morphologique** : Par analogie inverse avec l'érosion, l'opération de dilatation morphologique considère le rang maximale du filtre d'ordre et est utilisée pour injecter des niveaux de gris de rang important (tel que le blanc) afin de séparer des structure de niveaux de gris faible tel que le noir.

**IV-1-4 La fermeture morphologique** : Lors de la dilatation morphologique, l'injection du rang important entre les structures de faibles niveaux de gris pour les séparer peut engendrer des déformations de leurs formes. Pour remédier à cette éventuelle déformation, la dilatation doit être suivie par un filtrage d'ordre inférieur (proche du rang minimal). Ce processus (dilatation suivie par filtrage d'ordre inférieur) s'appelle la fermeture morphologique.

#### **IV-2- But de TP-4 :**

- S'initier au filtrage morphologique via l'étude du filtre d'ordre et d'un ensemble d'opérations morphologiques (l'érosion, la dilatation, l'ouverture et la fermeture).

#### **IV-3 Travail demandé de TP-4:**

Charger et afficher l'image « circles.png ». On désire effectuer des opérations morphologiques sur cette image afin de séparer ses structures (les disques dans cette image) tout en préservant leur forme de disque. Pour cela, on vous demande de :

1)- Avec la commande « strel », créer un noyau de type « disk », de rayon R, pour l'utiliser dans les opérations morphologiques.

2)- Avec la commande « imerode », effectuer une érosion avec le noyau précédent sur l'image « circles.png ». Afficher l'image originale et l'image érodée dans la même figure. Varier la valeur du rayon R du noyau jusqu'à ce que les structures de l'image soient séparées. Quelles sont vos constatations après avoir séparé les structures (les disques blancs) ? Quelle est l'opération morphologique nécessaire pour remédier à cela ?

Pour poursuivre le traitement morphologique de cette image, il est nécessaire d'effectuer une ouverture (ouverture = érosion suivie par un filtre d'ordre supérieur). Pour cela :

3)- En utilisant la commande « ordfilt2 », effectuer un filtrage d'ordre supérieur (avec un noyau d'un domaine 7x7 sur l'image érodée. Quelles sont vos constatations ?

4)- Inverser l'image « circles.png » et afficher la. Effectuer les opérations morphologiques suivantes afin de séparer et réparer les disques Noirs :

a). La dilatation (la commande "imdilate"), b). La fermeture (fermeture= dilatation suivie par un filtrage d'ordre inférieur). Donner vos constatations pour les étapes a) et b).

#### **IV-4- Solution du TP-4 :**

1)- et 2) Création un noyau de type « disk », de rayon R=11 (le rayon est défini après quelques essais) et son utilisation pour séparer les structures de l'image « circles.png » nécessitant une érosion morphologique :

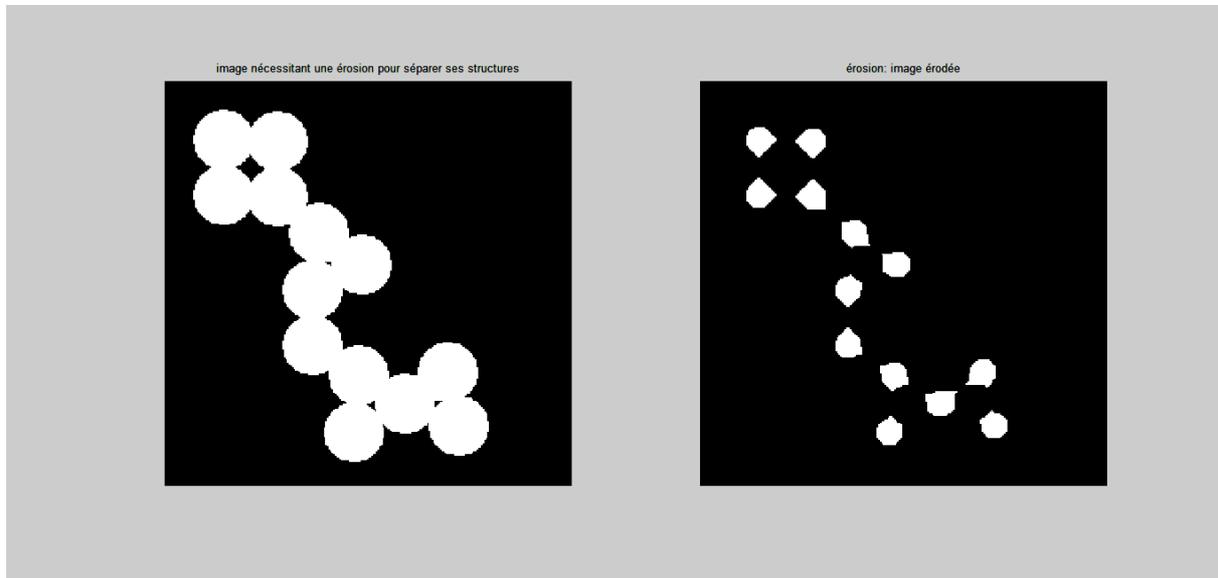


Figure-IV-1 érosion morphologique.

A partir de la figure-IV-1, nous pouvons constater que l'image « circles.png » nécessite l'injection du noir (niveau de gris du fond de l'image) afin de séparer les disques blancs. L'opération adéquate est l'érosion morphologique. Parfois, l'érosion morphologique peut être suffisante mais dans les cas où la séparation engendre des déformations de formes de la structure telle que c'est le cas avec l'érosion de cette image, il faut faire suivre l'érosion par un filtrage d'ordre supérieur afin de réparer ces déformations. Le processus complet s'appelle l'ouverture morphologique.

3)- En utilisant la commande « ordfilt2 », effectuons un filtrage d'ordre supérieur (avec un noyau d'un domaine 7x7 sur l'image érodée. Le noyau étant de 7x7 pixels, le rang maximal du filtre d'ordre est de 49. Prenant un rang supérieur, soit de 40 (proche de celui du maximum) :

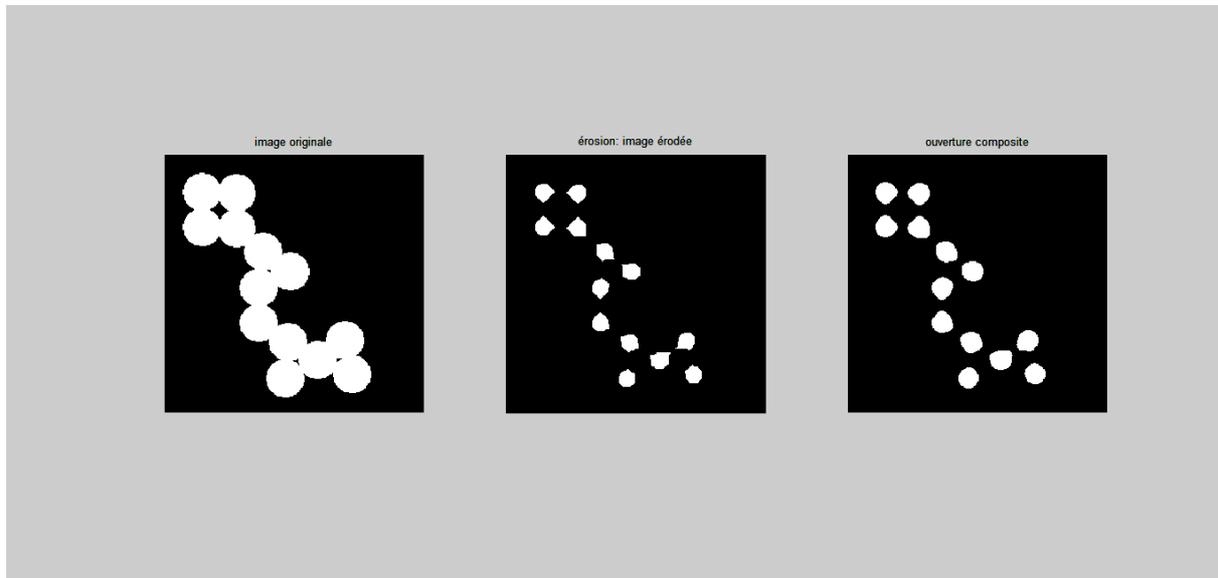


Figure-IV-2 Ouverture morphologique.

L'ouverture morphologique (figure-IV-2) a permis de remédier à la déformation des structures tout en les gardant séparées.

4)-a et 4)-b Traitement de l'image inversée (disques noirs sur fond d'image blanc) :

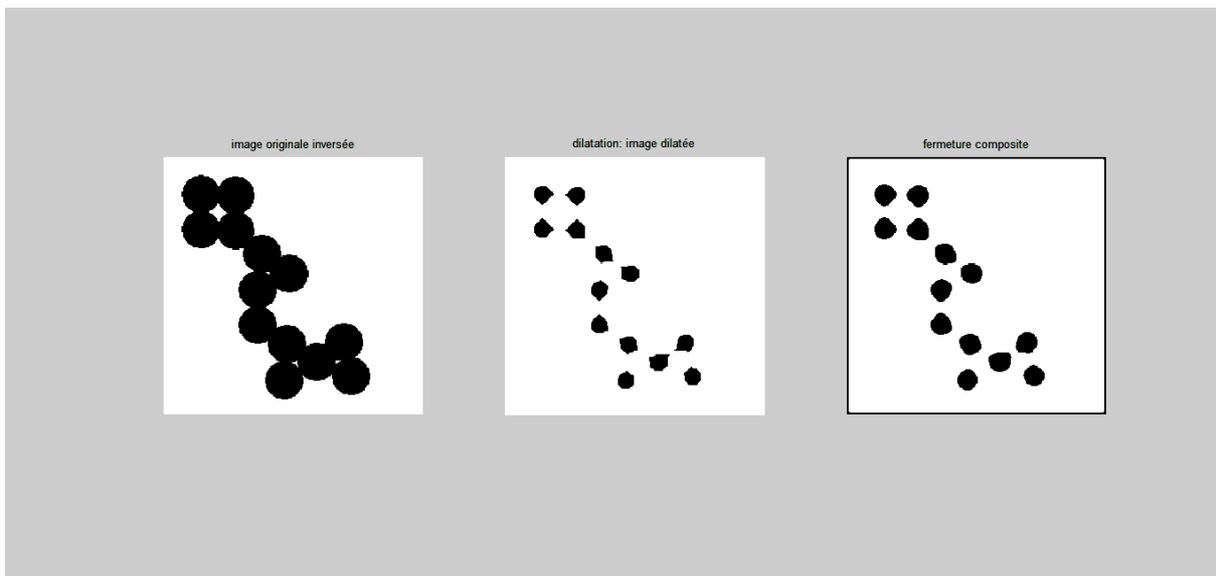


Figure-IV-3 Dilatation et fermeture morphologique.

L'image inversée nécessite une dilatation morphologique pour séparer les disques noirs (figure-IV-3). La séparation aboutie par la dilatation a engendré une déformation des formes

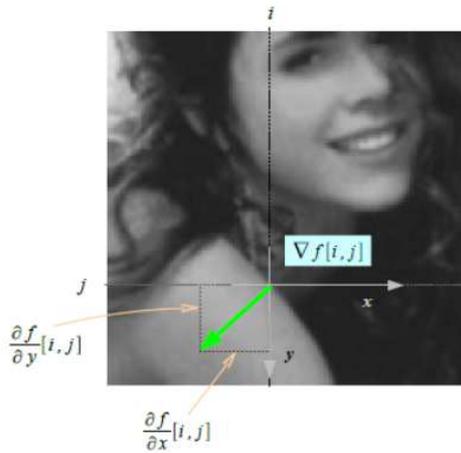


de disques noirs. Ce qui nécessite la poursuite de la dilatation morphologique par un filtrage d'ordre inférieur soit à un rang de 10. L'opération composite (fermeture) a permis de remédier à la déformation des disques tout en les gardant séparés (figure-IV-3).

## V- TP 5 : Détection de contours et rehaussement de contraste

### V-1 Rappel :

#### V-1-1 Filtrage dérivateur :



Les *variations locales d'intensité* constituent une source primordiale d'information en traitement d'images. Elles sont mesurées par le *gradient*, fonction *vectorielle* des pixels  $[i, j]$  :

$$\nabla f[i, j] = \left( \frac{\partial f}{\partial x}[i, j], \frac{\partial f}{\partial y}[i, j] \right)$$

D'autres grandeurs différentielles sont utilisées en traitement d'images, comme le *laplacien*, fonction *scalaire* de  $[i, j]$  :

$$\Delta f[i, j] = \frac{\partial^2 f}{\partial x^2}[i, j] + \frac{\partial^2 f}{\partial y^2}[i, j]$$

ou encore le *hessien*, fonction *matricielle* de  $[i, j]$  :

$$H_{,i,j} = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2}[i, j] & \frac{\partial^2 f}{\partial x \partial y}[i, j] \\ \frac{\partial^2 f}{\partial x \partial y}[i, j] & \frac{\partial^2 f}{\partial y^2}[i, j] \end{pmatrix}$$

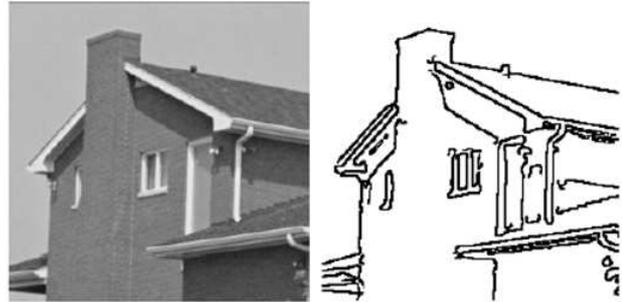
Le problème du calcul des filtres dérivateurs dans les images numériques est l'*approximation* de ces grandeurs différentielles dans notre espace discret ; on s'intéresse aussi à leur *utilisation* : rehaussement, détection de contours,...

La figure-V-1 illustre la détection de contours par filtre dérivateur ainsi que son utilisation pour rehausser le contraste de l'image [1].

Les filtres dérivateurs sont des opérateurs qui mettent en évidences certaines variations spatiales dans les images. Ils sont utilisés comme traitements de base dans de nombreuses opérations, comme le réhaussement de contraste ou la détection de contours :



réhaussement de contraste obtenu par combinaison linéaire avec le laplacien.



les contours (image de droite) constituent une simplification de l'image utile dans de nombreuses applications. Dans les approches linéaires, ils sont en général obtenus à partir des maxima locaux de la dérivée première, ou des passages par zéro de la dérivée seconde :

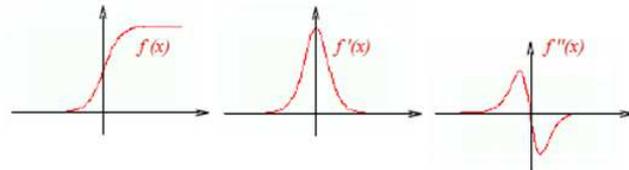


Figure-V-1 Détection de contours par filtre dérivateur ainsi que son utilisation pour rehausser le contraste de l'image.

Les approximations les plus simples des dérivées directionnelles se font par différences finies calculées par convolution avec des noyaux très simples :

Par ex :  $[-1 \ 1]$  , pour l'approximation de  $\frac{\partial f}{\partial x}$  , et :  $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$  , pour l'approximation de  $\frac{\partial f}{\partial y}$

On utilise plus souvent  $[-1 \ 0 \ 1]$  , respectivement  $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$  qui produisent des frontières plus épaisses, mais bien centrées (phase nulle).

Ces opérations étant très sensibles au bruit, on les combine en général avec un filtre lisseur dans la direction orthogonale à celle de dérivation, par ex par le noyau suivant (ou sa transposée) :  $[1 \ 2 \ 1]$

Le calcul des dérivées directionnelles en x et en y revient finalement à la convolution avec les noyaux suivants, respectivement :

$$f_x[i, j] = (f * h_x)[i, j] \quad , \quad \text{avec :} \quad h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (\text{Masques de Sobel})$$

$$f_y[i, j] = (f * h_y)[i, j]$$

On peut ensuite calculer la norme du gradient :

$$\|\nabla f[i, j]\|_2 = \sqrt{f_x[i, j]^2 + f_y[i, j]^2}$$

$$\|\nabla f[i, j]\|_1 = |f_x[i, j]| + |f_y[i, j]|$$

$$\|\nabla f[i, j]\|_\infty = \max(|f_x[i, j]|, |f_y[i, j]|)$$

Et son orientation :

$$\arg(\nabla f[i, j]) = \arctan\left(\frac{f_y[i, j]}{f_x[i, j]}\right)$$

La figure-V-2 illustre le résultat de filtrage dérivateur par différent noyaux dérivateur [1].

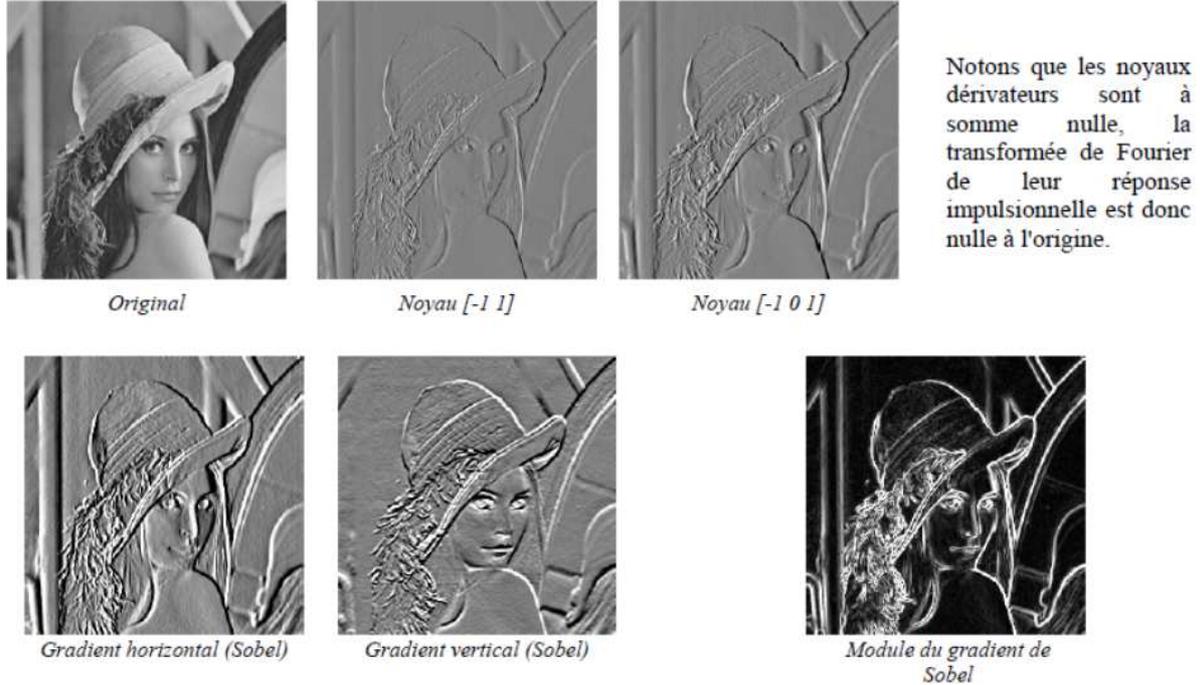


Figure-V-2 Illustration de résultat de filtrage dérivateur par différents noyaux dérivateurs.

Le filtre de Laplacien est souvent utilisé pour la détection des contours et pour le rehaussement de contraste. La figure-V-3 illustre différents noyaux pour le filtre Laplacien.

L'approximation par différences finies la plus simple de la dérivée seconde est la convolution par le noyau :

$$[1 - 2 1] \text{ , pour l'approximation de } \frac{\partial^2 f}{\partial x^2} \text{ , et : } \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \text{ , pour l'approximation de } \frac{\partial^2 f}{\partial y^2}$$

Le laplacien  $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$  peut donc être approximé par l'opérateur linéaire suivant :

$$\begin{bmatrix} 1 & & \\ 1 & -4 & 1 \\ & & 1 \end{bmatrix} \text{ Laplacien en 4-connexité , ou encore } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ Laplacien en 8-connexité}$$



Figure-V-3 Détection de contours par le filtrage de Laplacien.

**V-1-2 Rehaussement de contraste :**

Le rehaussement de contraste des images utilise essentiellement les filtres dérivateurs.

L'idée est que l'image sous contrastée est jugée contenant les deux types d'informations correspondantes aux basses et hautes fréquences spatiales. Néanmoins, les hautes fréquences spatiales, correspondant aux détails spatiaux et donc aux variations dans l'image, sont les première structure affectées par une dégradation tel que le flou.

Le flou est mathématiquement causé par le filtrage moyenneur qui rapproche la valeur du pixel filtré à son voisinage donnant un aspect identique dans l'image (le flou).

Bien que le flou affecte la perception visuelle de l'image en rapprochant les amplitudes (niveaux de gris) des hautes fréquences spatiales à leurs voisinages, les hautes fréquences spatiales peuvent être détectées par filtrage dérivateur puisqu'elles correspondent à des variations spatiales.

De ce fait, l'application du filtre dérivateur sur une image dégradée détecte (localise) les détails spatiaux et élimine les approximations car la dérivée élimine les fonctions de faible variation (continues).

Après avoir détecté les détails spatiaux, ils sont soustraits (à un gain près) de l'image dégradée afin de réduire leurs amplitudes qui ont été rapprochées (augmentées) à celles des approximations lors de dégradation.

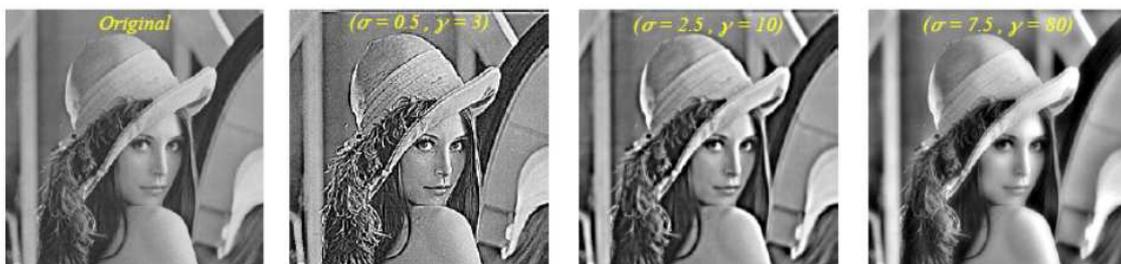
La figure-V-4 illustre le rehaussement de contraste [1].

Les opérateurs de traitement d'images possèdent une certaine portée correspondant à l'amplitude du voisinage qui interagit.

Un exemple, l'opération de réhaussement de contraste (unsharp masking) :

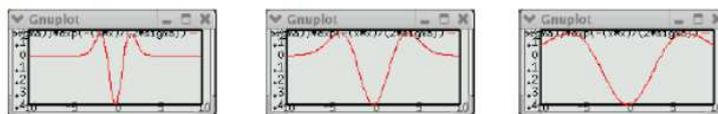
$$R_f[x, y] = f[x, y] - \gamma \cdot \Delta f[x, y] \quad (\gamma : \text{gain})$$

En soustrayant le laplacien à l'image original, on augmente le contraste.  
 Mais le contraste est une notion multi-échelle :



Il faut donc des mécanismes pour adapter la portée des opérateurs (voir cours espaces d'échelle).

*Laplaciens calculés par dérivées secondes de noyaux gaussiens, pour  $\sigma = 1$ , puis 5, puis 15 :*



**figure-V-4** Rehaussement de contraste.



### V-2 But de TP-5 :

- Détection de contours d'images par les filtres dérivateurs,
- Rehaussement de contraste pour les images floues ou sous contrastées.

### V-3 Travail demandé de TP-5 :

#### Partie-A :

Charger et afficher l'image « cameraman.tif ». On veut appliquer un filtrage dérivateur de type Laplacien afin de détecter ses contours. Pour cela :

1. En utilisant la commande « fspecial », créer un noyau dérivateur, h, de type Laplacien.
2. En utilisant la commande « imfilter », appliquer le noyau, h, de Laplacien sur l'image chargée afin de détecter ses contours. Afficher l'image et l'image filtrée (les contours) sur la même figure.
3. D'après l'analyse de l'image filtrée, que peut-on dire sur le gabarit du filtre dérivateur ?

#### Partie-B :

Afin d'appliquer le filtrage dérivateur pour le rehaussement (amélioration) de contraste des images floues (sous contrastées), on pondère les détails (contours ou les hautes fréquences spatiales). Pour cela on utilise l'équation suivante :  $I_r = I_d - \text{Gamma} * I_p$  ..... (eq-V-1) Tel que :

$I_r$  : L'image rehaussée (améliorée),

$I_d$  : l'image dégradée (floue ou sous contrastée),

$I_p$  : Les contours obtenus par l'application du filtre dérivateur, le Laplacien, sur l'image dégradée,

Gamma : Gain ou facteur de pondération.

- 1) Afin d'appliquer le rehaussement de contraste, il faut d'abord créer une image floue «  $I_d$  » (dégradée ou sous contrastée). Pour cela, en utilisant la commande « fspecial », créer un noyau de filtrage moyenneur et appliquer le sur l'image « cameraman.tif » en utilisant la commande « imfilter ».
- 2) Afficher l'image floue  $I_d$  sur la même figure. Que peut-on dire sur le gabarit du filtre moyenneur. Conclure.
- 3) L'image  $I_d$  nécessite un rehaussement de contraste (eq-V-1). Effectuer ce traitement. Faites varier le gain Gamma et donner vos constatations sur le résultat.

#### V-4 Solution du TP-5:

##### Partie-A :

1) et 2) Création du noyau du filtre dérivateur et son application sur l'image afin de détecter ses contours :



**figure-V-5** Détection des contours par filtre de Laplacien.

3)- D'après l'analyse de l'image filtrée (figure-V-5), l'image résultante ne contient que des détails spatiaux fins et des contours correspondant aux hautes fréquences spatiales. Ainsi les approximations ont été éliminées (régions noires dans l'image filtrée). De ce fait, nous pouvons dire que le filtre dérivateur se comporte comme un filtre passe-haut.

##### Partie-B :

1) Création de l'image dégradée (floue) en appliquant le filtre moyenneur qui permet de rapprocher les valeurs des pixels filtrés à celles de son voisinage :



**Figure-V-6** Création de l'image dégradée à traiter.

- 2) En analysant l'image dégradée (floue) de la figure-V-6, on constate que son apparence visuelle manque des détails fins comparativement à l'image originale. Ainsi le gabarit du filtrage moyennneur se comporte visuellement comme un filtre passe-bas. Néanmoins, les détails spatiaux dans l'image dégradée n'étant pas perceptibles visuellement mais ces détails existent dans l'image dégradée car l'opération de la moyenne n'est pas sensée d'éliminer les fréquences mais seulement de changer leurs amplitudes (niveaux de gris). De ce fait, la détection des détails et contours de l'image dégradée par un filtre dérivateur serait d'une grande utilité dans le rehaussement de son contraste.
- 3) Rehaussement de l'image dégradée (eq-V-1) à un gain  $\Gamma=5$  :



**Figure-V-7** Rehaussement de contraste de l'image dégradée.



L'application du filtre dérivateur sur l'image dégradée a permis de détecter les détails spatiaux et contours visuellement sous-contrastés. Après leur détection, leur soustraction de l'image dégradée (eq-V-1) a permis de leur retrouver des amplitudes (niveaux de gris) plus contrastés. Ainsi l'image rehaussée de l'image dégradée présente visuellement des détails plus rehaussés (détails fins sur cameraman, immeuble..etc).

## VI- TP 6 : Les outils de segmentation

### VI-1 Rappel :

#### Segmenter une image?

Il s'agit de Partitionner l'image en zone homogènes selon un critère déterminé: couleur, texture, niveau de gris, indice,...

Le but de partitionnement homogène (segmentation) est la définition des régions/frontières correspondant aux objets de l'image. De cette façon, la machine aura une vision plus intelligente envers une image numérique définie par sa matrice de pixels.

La figure-VI-1 illustre une segmentation d'image en se basant sur le critère de la couleur [1].

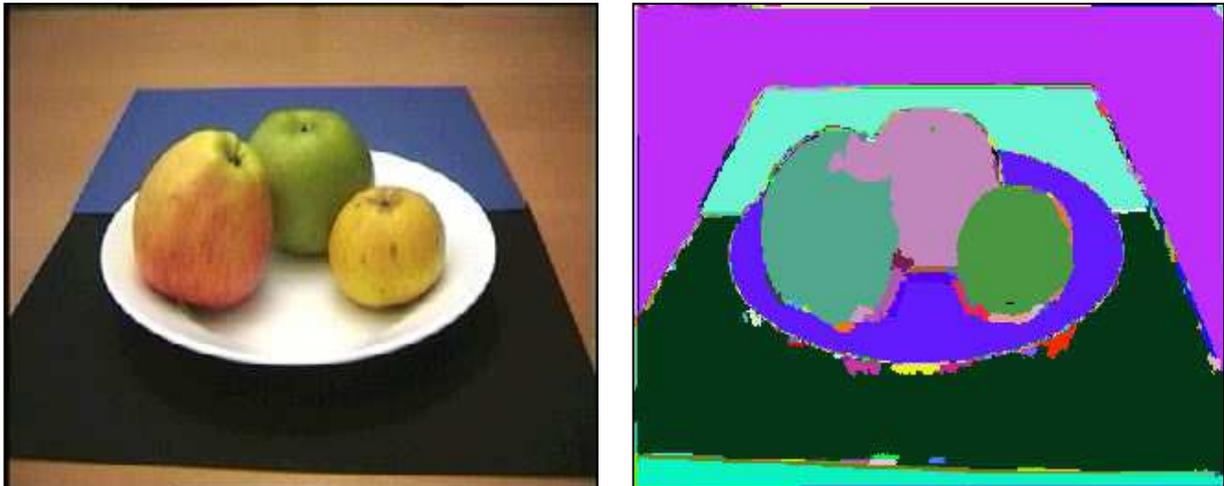


Figure-VI-1 Segmentation d'image se basant sur le critère de la couleur.

En se basant sur les régions de l'image, Il existe plusieurs approches de segmentation:

- Approches **GLOBALES**: utilisant l'histogramme de l'image
- Approches **LOCALES**: *region growing* (accroissement de région), contours
- Approches **HYBRIDES**: split & merge, CSC (Code de Structures de Couleur)

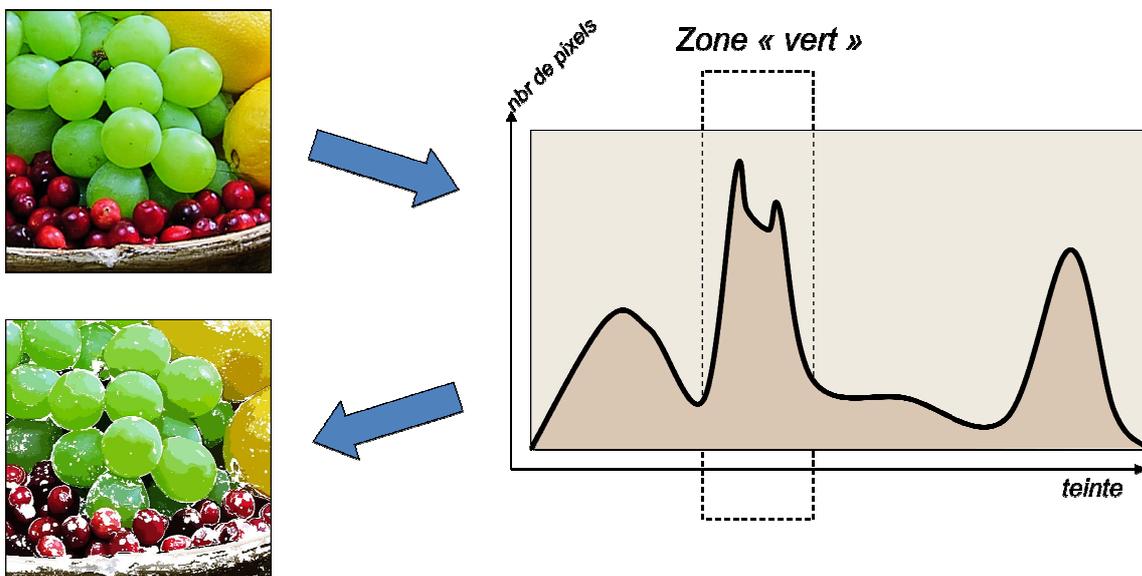
Selon le contexte de l'image, le type approprié de segmentation est décidé. En l'absence de contextes particuliers, les meilleurs résultats sont obtenus avec les méthodes hybrides.

**VI-1-1 Approches GLOBALES:** Utilisant l'histogramme de l'image.

**Idée:** Si les objets présents dans l'image ont des couleurs bien distinctes et uniformes, ils vont apparaître comme des pics dans l'histogramme.

La figure-VI-2 illustre le concept de pic, dans l'histogramme, correspondant à une couleur d'un objet homogène [1].

**Idée:** Si les objets présents dans l'image ont des couleurs bien distinctes et uniformes, ils vont apparaître comme des pics dans l'histogramme.



**Figure-VI-2** Concept de pic, dans l'histogramme, correspondant à une couleur d'un objet homogène.

La figure-VI-3 illustre le processus itératif d'éliminations récursives des pics lors de définition de leur ensemble de pixels approprié (segmentation d'objets homogènes) [1].

Chaque pixel est décrit selon certains channels: R,G,B,H,S,V,...

=> L'algorithme travaille sur plusieurs histogrammes, un par channel

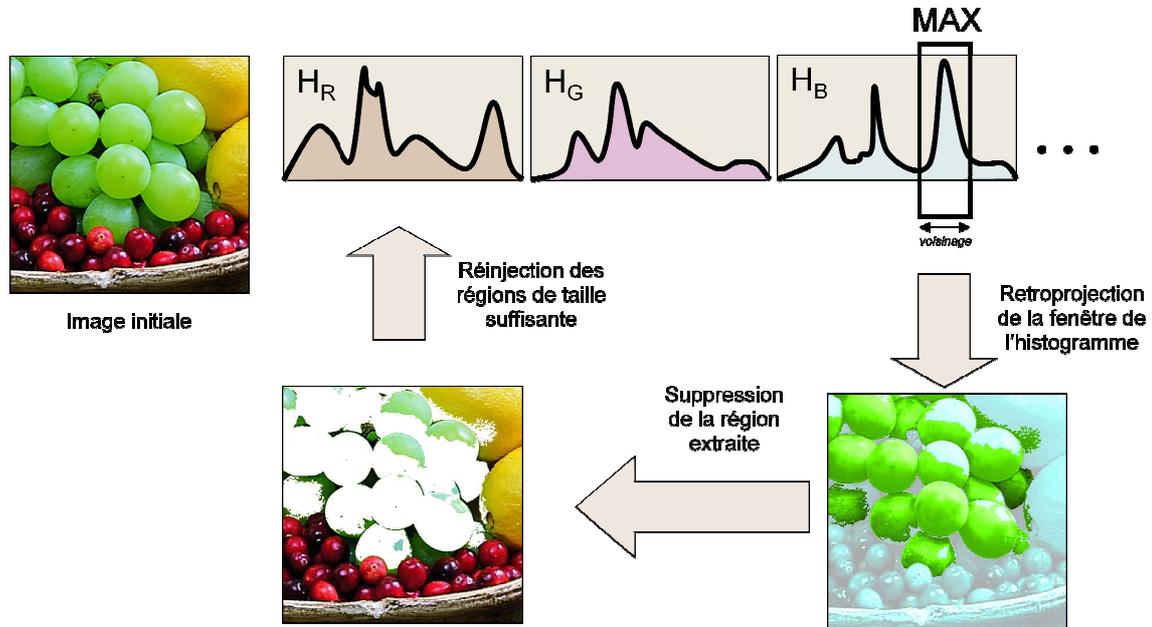


Figure-VI-3 Processus itératif d'éliminations récursives des pics lors de définition de leur ensemble de pixels approprié (segmentation d'objets homogènes).

**AVANTAGES :** Ce type de méthodes (globales) a les avantages suivants :

- Méthodes très rapides puisqu'elle traite un ensemble de pixels (des régions) correspondant aux pics dans l'histogramme,
- Peu sensibles au bruit car elles s'intéressent à une région et non plus au pixel.

**INCONVENIENTS :**

- \_Méthode globale: ignore les informations de proximité qui permettent d'utiliser des seuils variables locaux.
- Que se passe-t-il si deux objets ont la même couleur? => Nécessite en général un *Region Growing* pour détacher les composantes connexes.

Partant de l'inconvénient majeur des approches globales qui lorsqu'il s'agit deux objets (ou plus) différents ayant la même couleur (ou niveau de gris), ceux-ci seront segmentés comme étant le même objet. De plus, un objet ayant des couleurs différents (non homogènes) sera segmenté en plusieurs objets. De ce fait, les approches globales restent limitées sur les images présentant des **objets homogènes et distincts**. Pour des images ne vérifiant pas

l'hypothèse d'homogénéité et distinction d'objets, il faut penser à d'autres types d'approches de segmentation selon le concept approprié.

### **VI-1-2 Approches locales : Croissance de région (*Region Growing*)**

**Idée:** On part d'un point amorce (*seed*) et l'on l'étend en ajoutant les points de la frontière qui satisfont le critère d'homogénéité exigé. La figure-VI-4 illustre la segmentation par « *region growing* » [1].

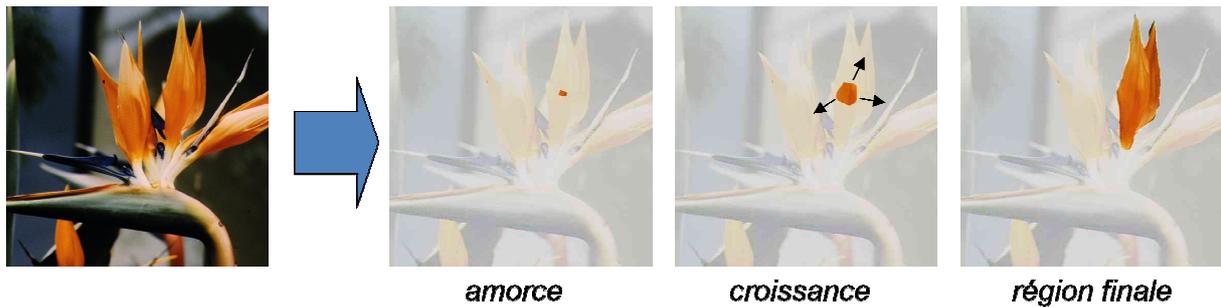


Figure-VI-4 Processus itératif d'accroissement de région (Approche locale).

### **AVANTAGES**

- Méthode rapide
- Conceptuellement très simple

### **INCONVENIENTS**

- Méthode locale: aucune vision globale du problème. En pratique, il y a presque toujours un chemin continu de points connexes de couleur proche qui relie deux points d'une image... Problème du gradient:



- Tenir compte de l'homogénéité globale donne un algorithme sensible à l'ordre de parcours des points (méthode par amorce)
- Algorithme très sensible au bruit puisqu'il s'intéresse au pixel (local), peu stable.

En cas d'image bruitée et en cas où le problème de gradient est présent dans l'image, la méthode *region growing* n'est pas appropriée.

### **VI-1-3 Approches hybrides :**

**VI-1-3-1 Division & fusion (Split & Merge)**

Dans l'absence de contexte particulier local/global, les méthodes hybrides sont les mieux appropriés.

**Idée:** Plutôt que de regrouper des pixels dans le *region growing*, pourquoi ne pas regrouper des zones homogènes pré-calculées sur l'image?

**Phase 1 :** Créer les zones homogènes = SPLIT

**Phase 2 :** Les regrouper = MERGE

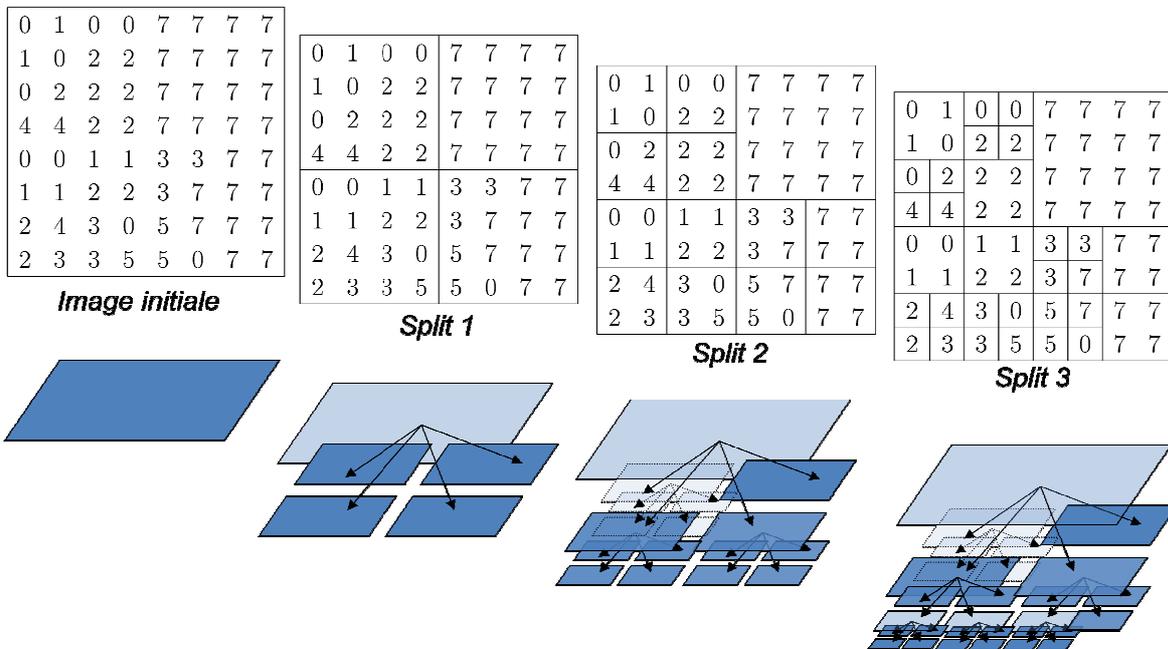
**SPLIT:** L'image est stockée dans un arbre.

Initialement, *arbre racine = image complète*

Récursivement, chaque feuille F est subdivisée en quatre si elle n'est pas assez homogène, et les quatre sous images sont ajoutées en tant que feuilles de F.

L'algorithme poursuit tant qu'il reste des feuilles non homogènes à diviser [1].

**Example**



**Figure-VI-5** Processus récursive de division dans la méthode SPLIT & MERGE.

**MERGE:** Dans cette phase les feuilles homogène voisines obtenues lors de la phase de division et qui vérifient un critère de ressemblance exigé sont regroupées pour former un seul objet segmenté [1].

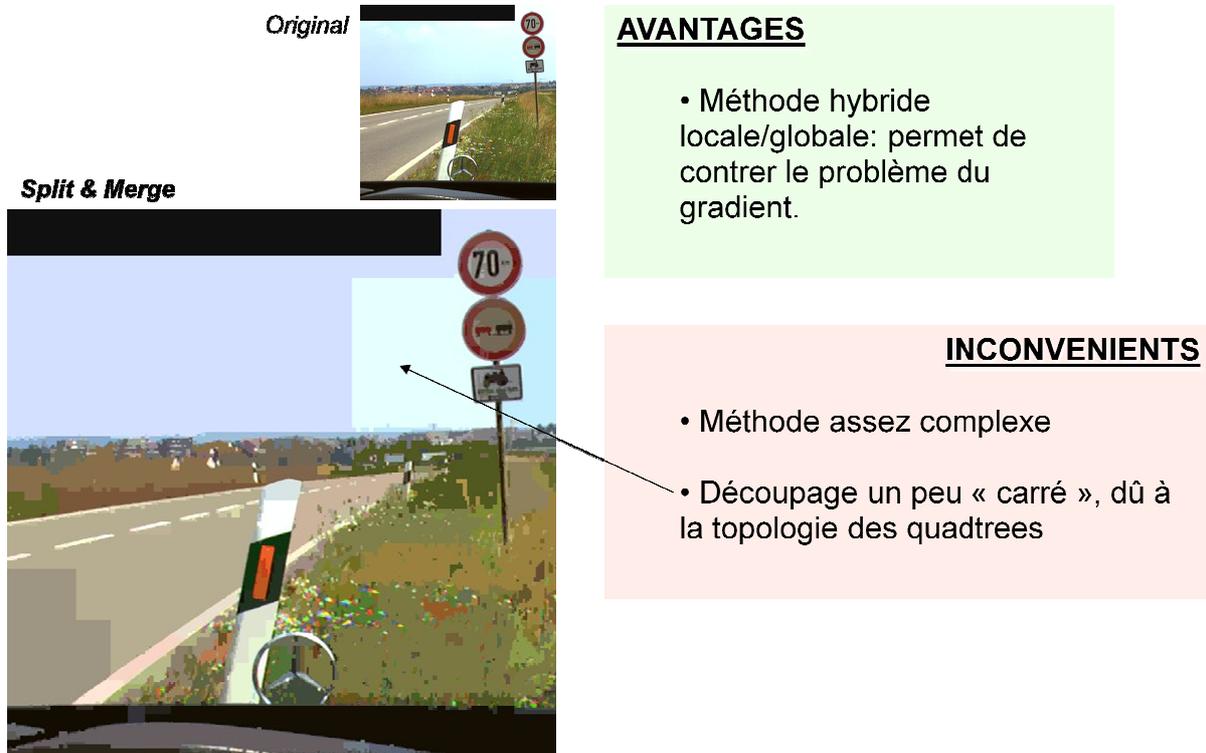


Figure-VI-6 Avantage et inconvénients SPLIT & MERGE.

L'inconvénient majeur de la méthode hybride SPLIT & MERGE est la forme carrée exigée aux objets segmentés. Cela est dû à la topologie de division carrée (figure-VI-5).

Pour remédier au problème de topologie de division carrée, une autre méthode hybride de division & fusion **Algorithme CSC (Color Structure Code)** utilise des structure hexagonales chevauchées pour permettre plus de flexibilité dans la forme des objets segmentés.

### VI-1-3-2 Division & Fusion : Algorithme CSC (Color Structure Code)

**Idée:** Regrouper des structures de pixels qui se chevauchent et découper ensuite les zones communes par une descente récursive [3].

L'algorithme CSC est un algorithme *Merge & Split* basé sur une *topologie hiérarchique hexagonale avec recouvrements*.

La figure-VI-7 illustre la création d'îlots hexagonaux chevauchés lors de la phase de division pour l'algorithme CSC permettant d'avoir différentes formes d'objets segmentés lors de la phase de division [3].

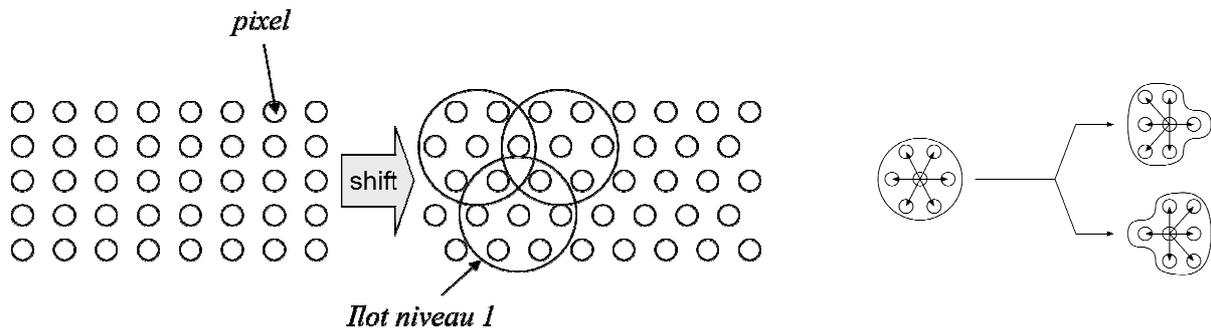


Figure-VI-7 Création d'îlots hexagonaux chevauchés lors de la phase de division pour l'algorithme CSC.

La figure-VI-8 montre la division récursive d'îlots dans des niveaux hiérarchique de l'espace de l'image [3].

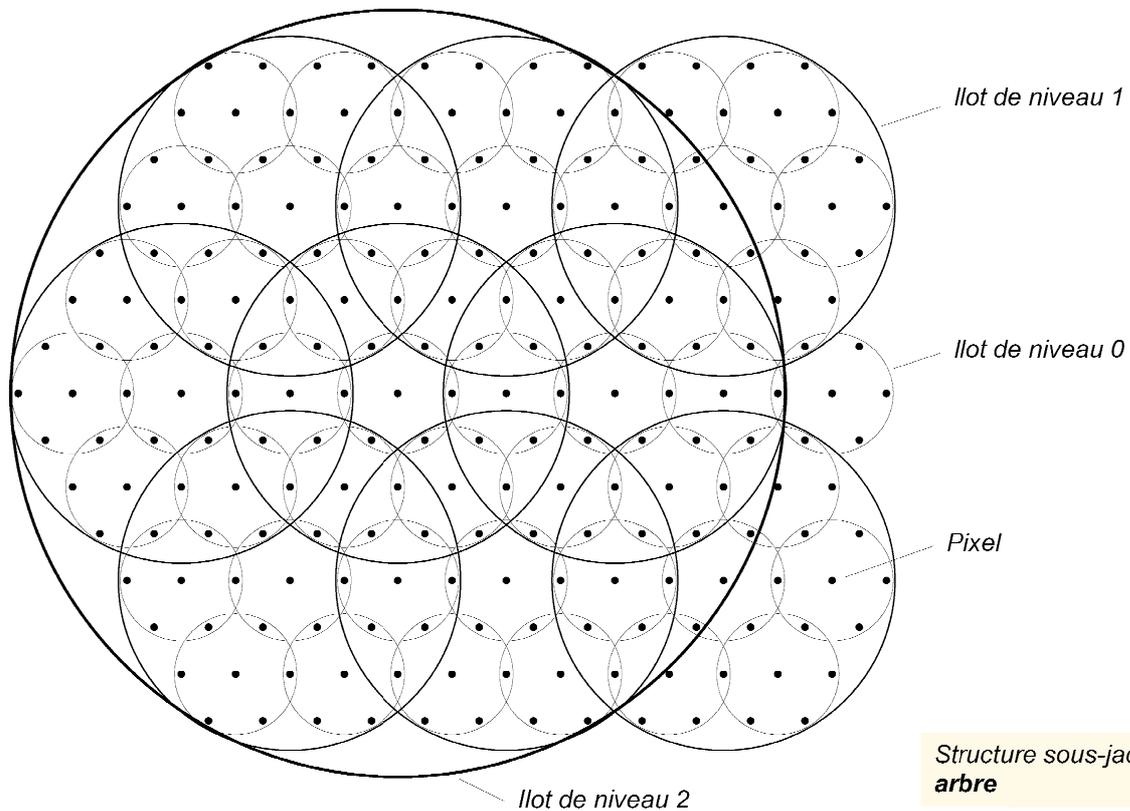


Figure-VI-8 Division récursive d'îlots.

Les phases de l'algorithme CSC sont résumées dans les figures suivantes [3]:

## PHASE 1 : INITIALISATION

Localement pour chaque îlot de niveau 0, on construit une partition en **éléments** (de niveau 0).

**élément** = région de couleur homogène

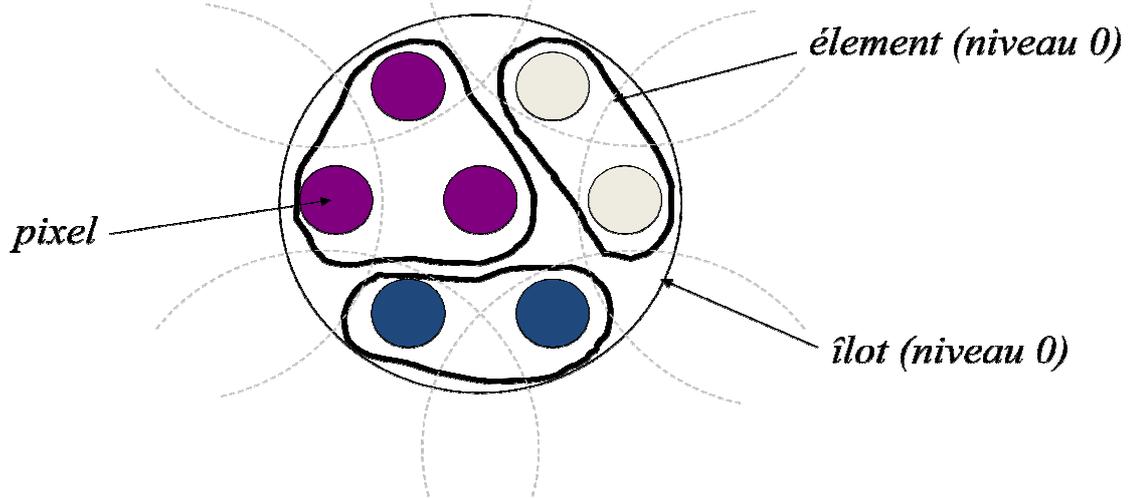


Figure-VI-9 Phase d'initialisation de l'algorithme CSC.

## PHASE 2 : REGROUPEMENT

On considère les éléments de niveau  $n$  contenus dans les îlots de niveau  $n$ . On se place dans un îlot de niveau  $n+1$  (ici  $n=0$ , pour l'exemple). Les éléments de niveau  $n$  sont regroupés en éléments de niveau  $n+1$  si:

- Ils sont de couleur proche
- Ils se touchent

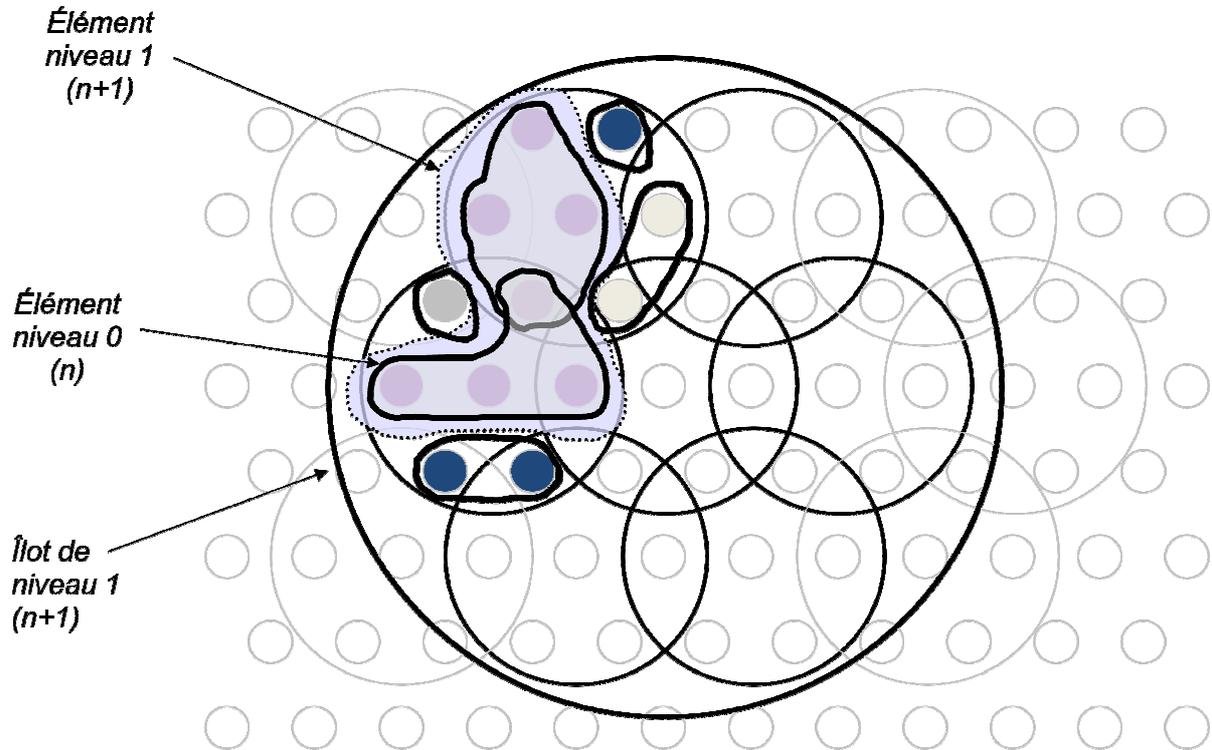


Figure-VI-10 Phase de regroupement de l'algorithme CSC.

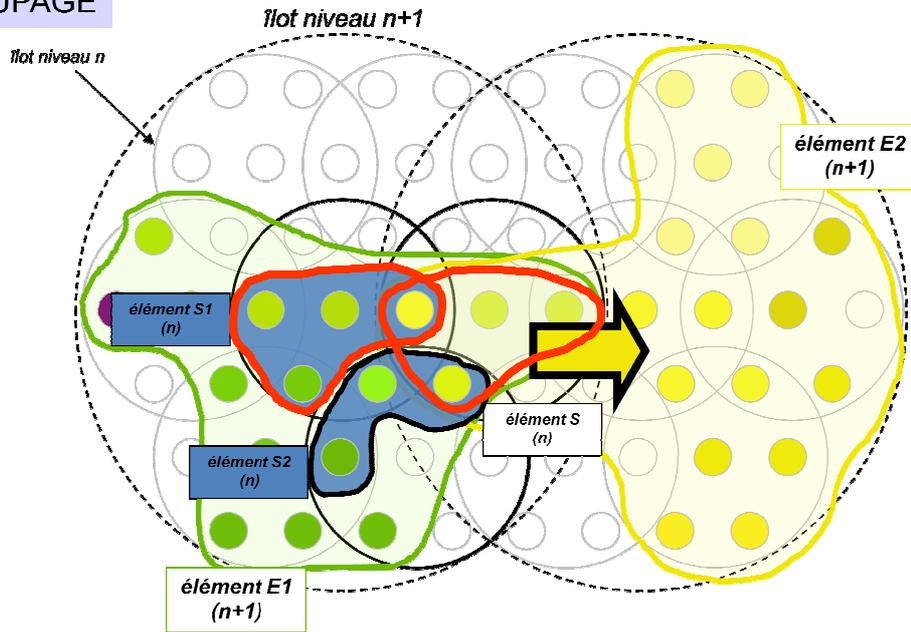
### PHASE 3 : DECOUPAGE

→Le problème du chaînage d'éléments localement identiques n'est pas résolu à ce stade  
**(Problème du gradient)**

L'algorithme CSC permet de gérer ce problème dans la phase de découpage, qui a lieu en parallèle de la phase de regroupement.

**Découpage:** Que se passe-t-il dans la phase de regroupement lorsque deux éléments de niveau  $n$  sont connectés mais de couleurs incompatibles? => Découpage récursif (figure-VI-11) [3].

**DECOUPAGE**

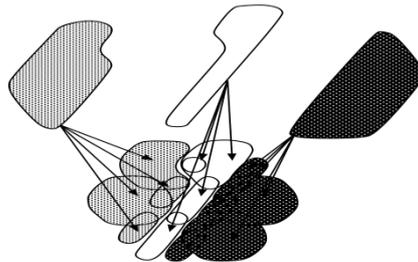


**=> découpage récursif**

**S attribué à E1 ou E2? Attribution à celui dont la couleur est la plus proche (ici E2)**

**Figure-VI-11** Phase de découpage récursif de l'algorithme CSC.

L'algorithme CSC est une méthode hybride permettant de remédier au problème de la topologie carrée dont les objets segmentés peuvent prendre différents formes (figure-VI-12) [3].



**Figure-VI-12** Formes obtenues des objets segmentés par l'algorithme CSC.

Rappelons la définition initiale de la segmentation évoquée au début de ce rappel pour ce TP; le but de partitionnement homogène (segmentation) est la définition des **régions/frontières** définissant des objets dans l'image. Toutes les méthodes évoquées jusqu'ici sont des méthodes basées sur la région (que ce soit locales, globales ou hybrides).

Il existe aussi des méthodes de segmentation basées sur les contours (et se sont considérés des approches locales).

Les méthodes génétiques utilisant des algorithmes modélisés à partir de comportement des être vivants (animaux..etc) ou de la nature (inondations..etc) peuvent offrir des algorithmes efficaces dans différents domaines tel que pour la segmentation d'image.

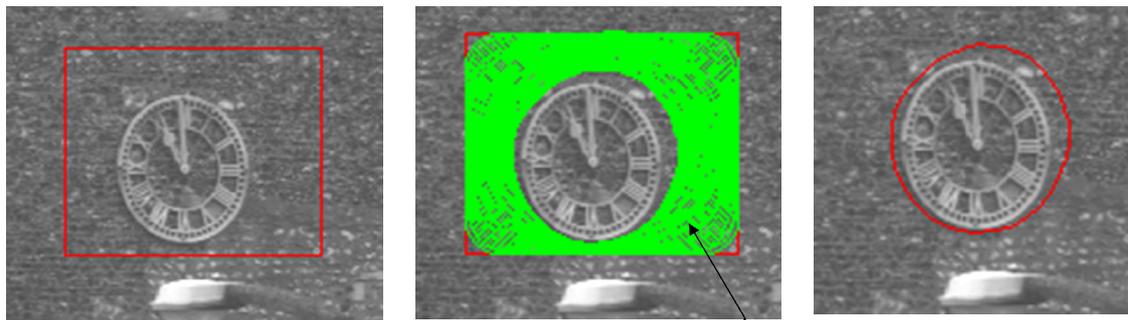
#### VI-1-4 Approches génétiques :

##### VI-1-4-1 méthode basée sur contours (*snake*) :

Etant donné que les algorithmes génétiques extraient leurs fonctionnement par modélisation du comportement des êtres vivants et de la nature, une méthode de segmentation basée sur les contours de l'image s'appuyant sur le comportement du *SNAKE* (serpent) qui est considéré le meilleur segmenteur dans la nature en détenant l'objet par ses frontières (contours).

**Idée:** Utiliser des courbes déformables qui sont « attirées » par les formes recherchées dans l'image [3].

Analogie avec la physique: *fonction d'énergie à minimiser.*



Le « snake » se contracte et s'adapte à la forme de l'horloge

Figure-VI-13 Illustration de la segmentation par la méthode « Snake ».

En minimisant l'énergie du contour actif (contour déformable vers les contours de l'image), on se situe sur les contours de l'image car ceux-ci sont caractérisés par des énergies faibles comparativement aux régions de l'image.

Un autre algorithme génétique, basé sur région, de segmentation sera évoqué lors de l'expérimentation du TP. C'est l'algorithme « *Water Thresholding* » inspiré d'un phénomène de la nature qui est les inondations.

### VI-2 But de TP-6 :

- Segmentation d'images utilisant une méthode basée sur les contours et une méthode basée sur les régions.

### VI-3 Travail demandé de TP-6 :

#### **Partie-A :**

Charger et afficher l'image 'coins.png'. On veut segmenter cette image utilisant une méthode basée sur les contours, soit la méthode du contour actif (*Snake*). Pour cela :

1. Préparer le contour initial en utilisant la séquence suivante sur Matlab:

```
mask = zeros(size(I));  
mask(25:end-25,25:end-25) = 1;
```

Afficher ce contour initial dans la même figure avec l'image.

2. Utiliser la fonction **activecontour** pour segmenter cette image dans la même figure tout en variant le nombre de boucle n en prenant successivement n=100 puis n=200 ensuite n=300 et enfin n=500. Donner vos constatations et conclusions.

#### **Partie-B :**

Utilisant la même image que la partie-A, on veut dans cette partie la segmenter utilisant une méthode basée sur la région, soit la méthode dite **water-thresholding**. Pour comprendre le fonctionnement de cette méthode, poursuivez les étapes suivantes :

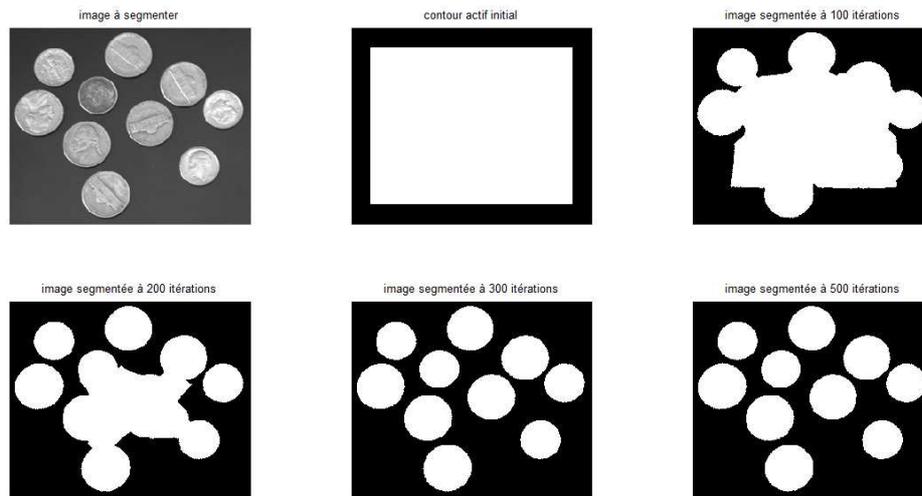
1. Convertir l'image 'coins.png' du format **uint8** en format **double** et afficher la en 3D utilisant la commande **mesh**.
2. A l'aide de l'icône **Rotate 3D** de la **barre d'outils** de cette figure en 3D, faites des rotations sur la figure tout en imaginant que vous la remplissez partiellement avec de l'eau et faites le lien avec la segmentation.
3. Trouver le seuil (niveau) de remplissage en eau utilisant la fonction **graythresh** pour l'image en format **uint8**.
4. Utiliser le seuil trouvé dans la question précédente pour segmenter l'image à l'aide de la fonction de binarisation d'image **im2bw**. Afficher l'image segmentée.

Comparer les deux résultats de segmentation dans les deux parties A et B et donner vos constatations et conclusions.

## VI-4 Solution de TP-6 :

### Partie-A :

1)- et 2)- Préparation du contour actif initial et segmentation par la méthode « Snake » de l'image pour 100, 200, 300 et 500 itérations :



**Figure-VI-14** Segmentation de l'image chargée par la méthode « Snake ».

Nous constatons, à partir de la figure-VI-14, que la méthode « snake » a pu segmenter tous les objets dans l'image. Néanmoins, la définition du nombre d'itérations n'est pas automatique. La définition d'un nombre important d'itérations  $n$  (comme  $n=500$  dans cet exemple) assure une segmentation satisfaisante mais prend un temps excessif supplémentaire de traitement. Pour cet exemple,  $n=300$  est suffisant mais inconnu préalablement. En conclusion, la définition non automatique du nombre d'itérations (du seuil en général) peut ou bien causer une segmentation incomplète (comme pour  $n=100$  et  $n=200$  dans cet exemple) ou une segmentation lourde dans le temps de calcul en effectuant des itérations en plus.

### Partie-B :

1) et 2) Affichage en 3D de l'image :

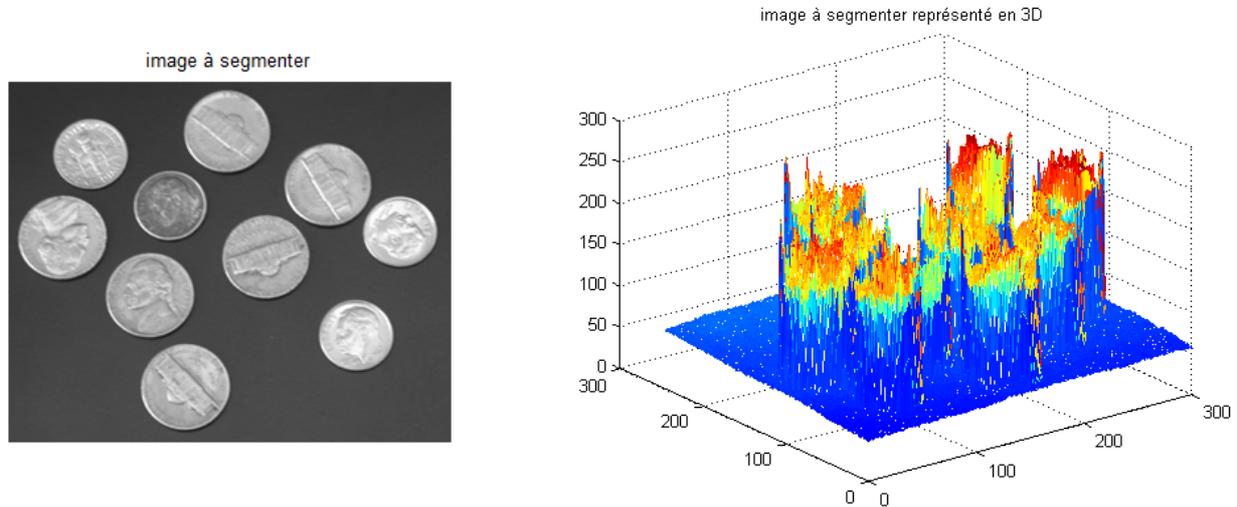


Figure-VI-15 Image à segmenter et sa représentation en 3D.

Pour comprendre la méthode « *Water thresholding* », imaginons la représentation 3D de l'image (figure-VI-15) comme des montagnes et commençons à remplir de l'eau (inondations). A un certain seuil (*threshold*) de remplissage d'eau, seuls les sommets de montagnes seront visibles de vision en haut ce qui est une sorte de segmentation d'objets pour l'image. Le seuil de remplissage est calculé comme étant le centre des distances proportionnelles entre le plus petit sommet en hauteur et le sol. Ensuite, ce seuil est utilisé pour binariser l'image inondée : La partie apparente au-dessus de l'eau remplie est mise à 1 et la partie émergée prend 0.

3)- Le seuil calculé dans cet exemple est de 0.4941 ce qui veut dire que 49.41% de l'hauteur maximale sera remplie en eau pour segmenter l'image.

4)- Utilisation du seuil calculé pour inonder l'image et la segmenter par binarisation :

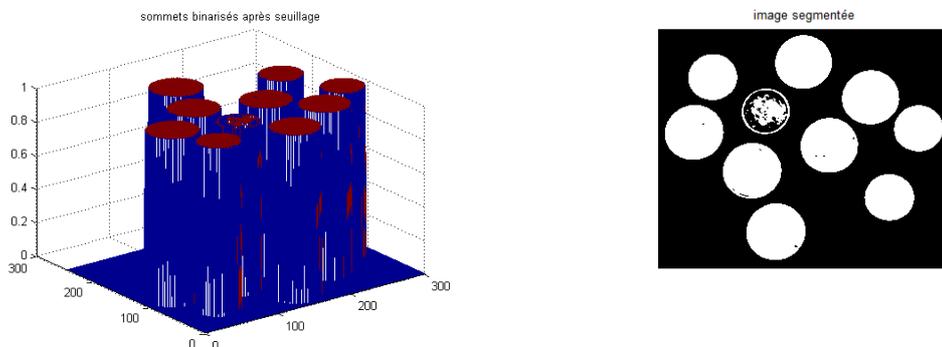


Figure-VI-16 Image segmentée par la méthode « *Water thresholding* ».



Comparativement à la méthode « *Snake* », la méthode « *Water Thresholding* » a permis la segmentation (figure-VI-16) en effectuant un calcul **automatique** du seuil de remplissage en eau. Néanmoins, la pièce de monnaie non homogène (figure-VI-16) n'a pas été segmentée en une seule pièce car l'eau a pu y émerger dans une partie de cette pièce non homogène. C'est l'inconvénient déjà évoqué pour les méthodes globales de segmentation dont la méthode « *Water Thresholding* » en fait partie.

## VII- TP-7 : Détection d'objets similaires dans une image par opération de corrélation (détection directe)

### VII-1 Rappel:

On donne le coefficient de corrélation calculé entre deux matrices  $G$  et  $H$  de même dimension

par :

$$\gamma = \frac{\left| \sum_s \sum_t (G(s,t) - \bar{G}) \cdot (H(s,t) - \bar{H}) \right|}{\sqrt{\sum_s \sum_t (G(s,t) - \bar{G})^2 \cdot \sum_s \sum_t (H(s,t) - \bar{H})^2}} \quad (\text{eq-VII-1})$$

Où  $\bar{G}$  et  $\bar{H}$  sont successivement les valeurs moyennes des matrices  $G$  et  $H$ .

Le coefficient de corrélation est compris entre  $[0, 1]$ .

Un **Template** (objet candidat pour détecter les objets de sa classe) contenant l'objet à détecter sa classe dans l'image est balayé sur celle-ci. Dans chaque position de balayage le coefficient de corrélation (eq-VII-1) est calculé entre le *template* et la matrice correspondante à la position actuelle sur l'image. Les coefficients de corrélations vérifiant un certain seuil prédéfini indique une détection d'objet de même classe que le template. L'action envisagée dans ce TP est d'encadrer l'objet détecté.

### VII-2 But du TP-7:

- Se familiariser avec la notion de détection d'objets similaires dans une image et mise en œuvre de l'opération de corrélation.

### VII-3 Travail demandé du TP-7:

- 1) Créer dans un fichier script la fonction suivante et lire le help des commandes utilisées dans cette fonction:

```
function findtemplate(im,temp,th)

out=normxcorr2(temp,im);
[m,n]=size(temp);

bw=out>th;
r=regionprops(bwlabel(bw));

imshow(im,[])
hold on
for i=1:length(r)
    rectangle('Position',[r(i).Centroid(1)-n,r(i).Centroid(2)-m,n,m]);
end
end
```

- 2) En exploitant cette fonction, faites la détection d'objet *template* de l'image « templ.tif » dans l'image « im.tif » pour différents seuils de coefficients de corrélation.
- 3) Faite vos remarques concernant les résultats de détection en changeant le seuil « th ».
- 4) Rajouter du bruit de type 'Salt & pepper' de densité de 0.3 à l'image « im.tif » et refaites le même travail. Faites vos remarques et tirer des conclusions concernant l'utilité et l'efficacité de la corrélation quand à la détection d'objets.

#### VII-4 Solution du TP-7:

- 1) et 2) Création de la fonction de détection et son application sur l'image avec différents seuils pour les coefficients de corrélation :

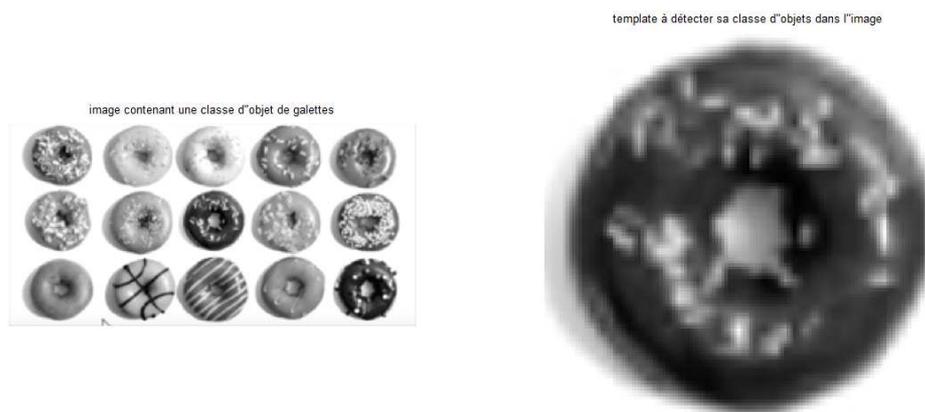
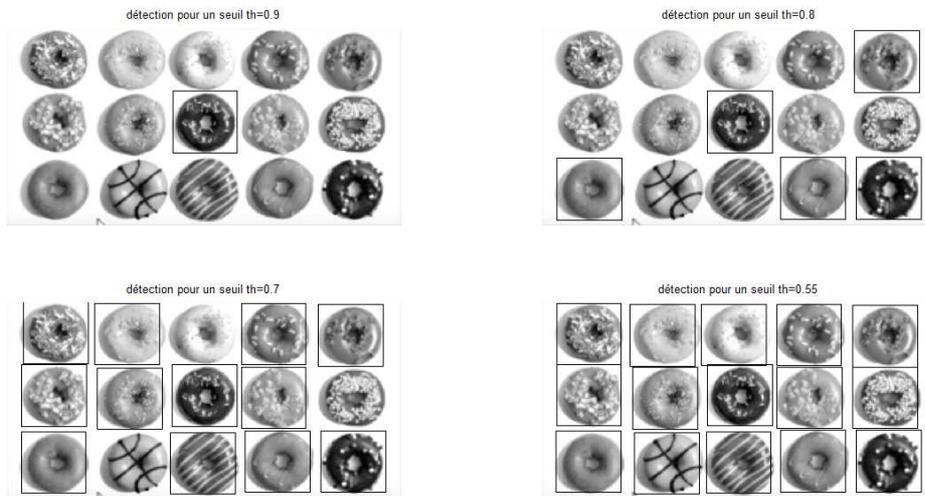


Figure-VII-1 Image et *template*.

3)- Pour différents seuils pour les coefficients de corrélation, nous avons obtenu (figure-VII-2) :

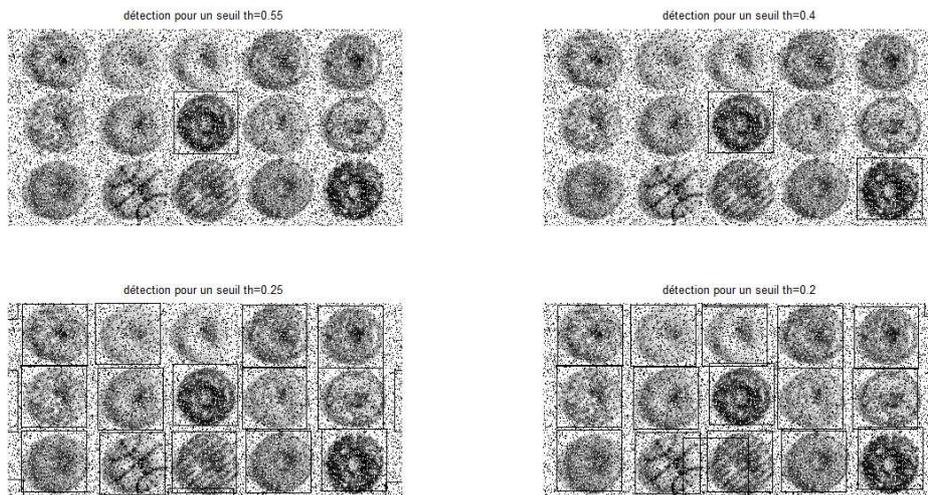
- pour  $th=0.9$ , une seule galette qui a été détecté. C'est la plus semblable au template,
- Pour  $th=0.8$ , quelques galettes ont été détectées,
- Pour  $th=0.7$ , la majorité des galettes ont été détectées,
- Pour  $th=0.55$ , toute les galettes ont été détectées.

Nous constatons que la définition du seuil n'est pas automatique. Cela est un inconvénient pour les méthodes directes (corrélation).



**Figure-VII-2** Détection d'objets de la même classe que le *template* à différents seuils de coefficients de corrélation.

4)- Rajoutons du bruit de type 'Salt & pepper' de densité de 0.3 à l'image « im.tif » et refaisons le même travail.



**Figure-VII-3** Détection d'objets de la même classe que le *template* à différents seuils de coefficients de corrélation dans une image bruitée.

Nous constatons que la détection directe (par coefficients de corrélation) est affectée par le bruit. Le seuil  $th=0.55$ , qui a été suffisant pour détecter toutes les galettes dans l'image saine non bruitée (figure-VII-2), n'est plus suffisant pour l'image bruitée (figure-VII-3) en ne détectant que deux objets.



La redéfinition des seuils plus faibles à cause du bruit, rend la détection incorrecte. Pour pouvoir détecter tous les objets, on était obligé de baisser le seuil jusqu'à  $th=0.2$ . Mais dans de tel faible seuil de coefficients de corrélations, d'autres objets à part les objets de la même classe que le template peuvent être détectés (fausses détection). C'est le cas pour cet exemple à  $th=2$  (figure-VII-3).

En conclusion, en plus des deux inconvénients en déduits expérimentalement pour les méthodes de détection directes (le seuil non automatiquement défini et la sensibilité aux bruits), la recherche exhaustive sur toute l'image tout en considérons tous les pixels du template rend le temps de calcul plus lourd et limite les chances d'utiliser les méthodes directes dans des applications multi-tâches exigeant le fonctionnement en temps réel.

C'est pour cela que les méthodes directes de détection restent limitées pour les milieux fermés où l'image est à l'abri du bruit et ne nécessite pas un fonctionnement en temps réel tel que pour le domaine d'aide au diagnostic dans l'imagerie médicale. Pour remédier au problème de définition non automatique du seuil, celui-ci est préalablement prédéfini pour chaque type d'image.

## VIII- TP 8 : Détection de points d'intérêt pour la mise en correspondance des objets (méthodes de détection indirecte)

### VIII-1 Rappel:

Partant de la limite des méthodes directes de détection envers le temps important parce qu'elles prennent en considération tout les pixels de l'objet à détecter (voir TP-7). A titre d'exemple lorsqu'il s'agit de détecter un objet mobile rapide afin de le poursuivre, les méthodes de type directes sont limitées. Les méthodes indirectes en définissant des points caractéristiques de l'objet à détecter réduisent considérablement le temps de calcul et se sont donc plus adaptées pour des applications en temps réel.

La figure-VIII-1 illustre la complexité du calcul direct par rapport à celui indirect exploitant des points caractéristique de l'objet.

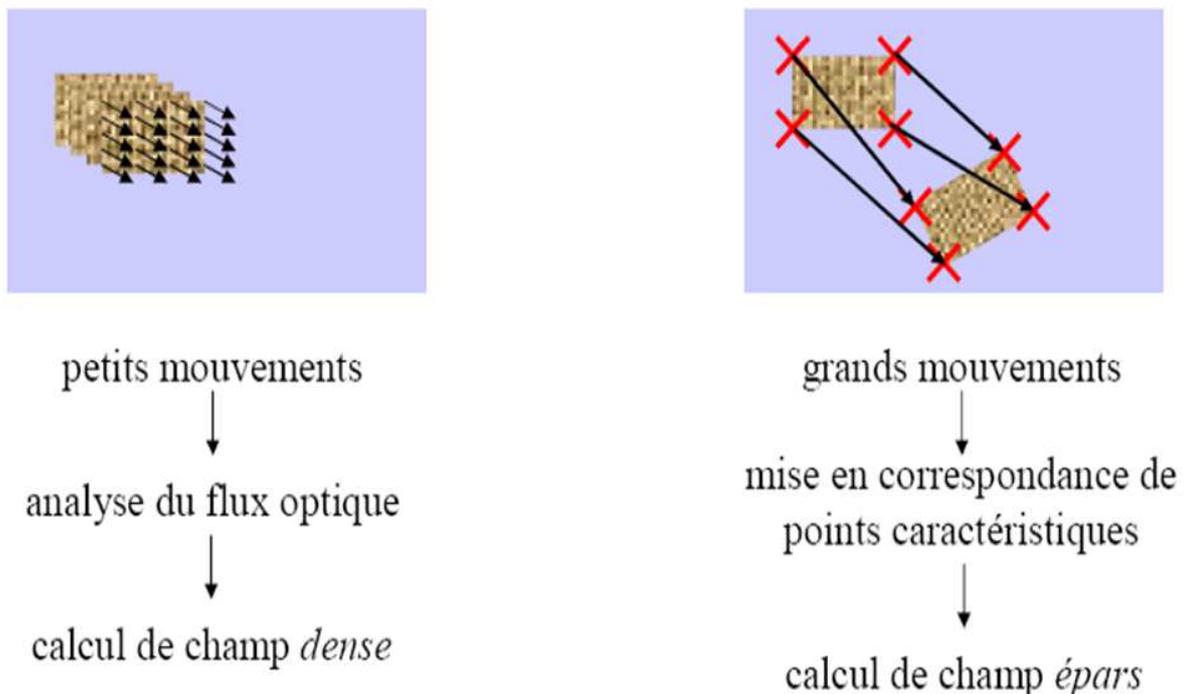


Figure-VIII-1 Calcul direct vs calcul indirect du champ de mouvement d'un objet mobile.

### VIII-1-1 La méthode de détection indirecte « HARRIS »

L'idée de cette méthode est d'extraire les corners du *template*, ou de l'objet à détecter en général, afin de ne pas considérer tout ses pixels lors de sa détection. La dérivée horizontale permet de détecter les lignes verticales des contours. Tandis que la dérivée verticale permet de détecter les lignes horizontales des contours. L'intersection des lignes verticales et

horizontales des contours définissent les corners de l'objet considérés par la méthode HARRIS (figure-VIII-2) [6].

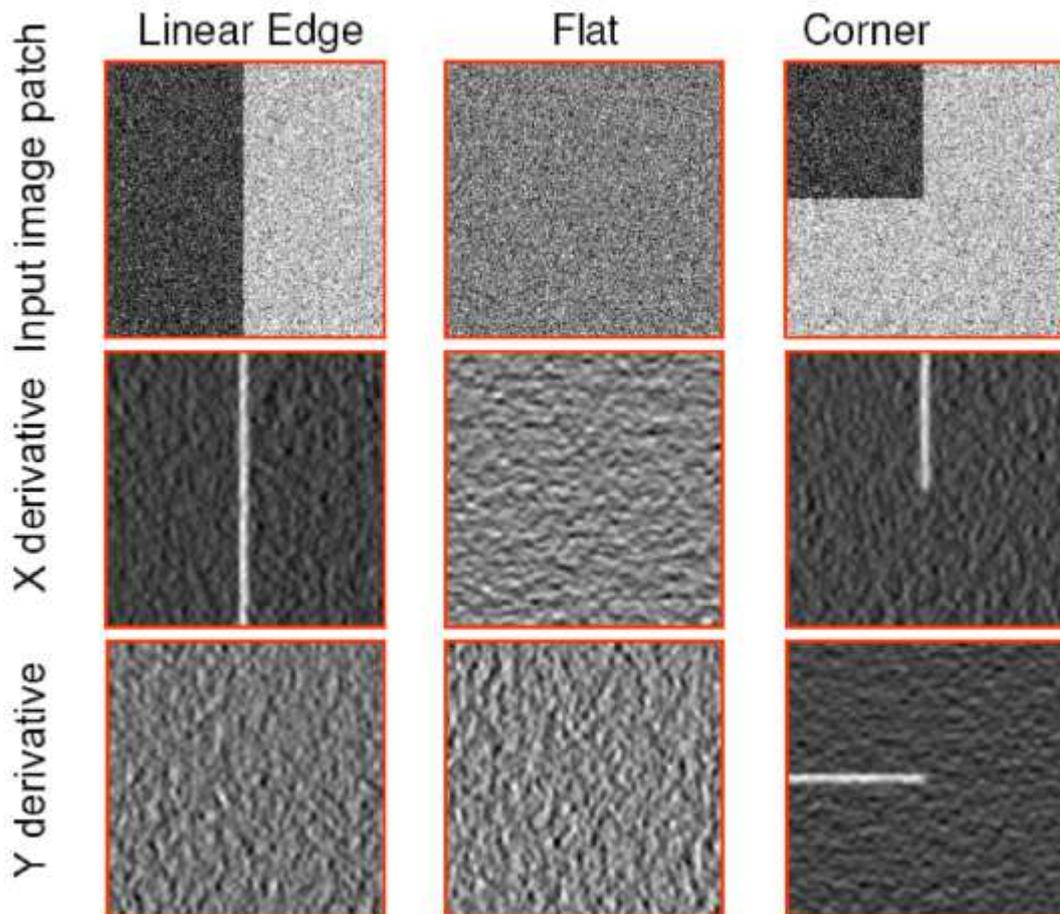


Figure-VIII-2 Illustration de la méthode HARRIS pour extraire les corners de l'objet à détecter.

A partir de la figure-VIII-2, nous constatons que la méthode HARRIS peut extraire les corners formés par des contours horizontaux ou verticaux. Dans le cas où les contours ne le sont pas, cette méthode de HARRIS n'arrive pas à extraire correctement les corners. Ainsi, de même, si un objet carré lors de son mouvement effectue des rotations, les corners du rectangle incliné peuvent ne pas être correctement extraits [6].

#### Avantages/Problèmes et limites pour HARRIS :

- Limites: Méthode peu efficace pour le matching (mise en correspondance) dans les échelles différentes (rotations..etc)

- Qualité de matching limitée pour la reconstruction 3D, estimation de mouvement..etc

**AVANTAGES:**

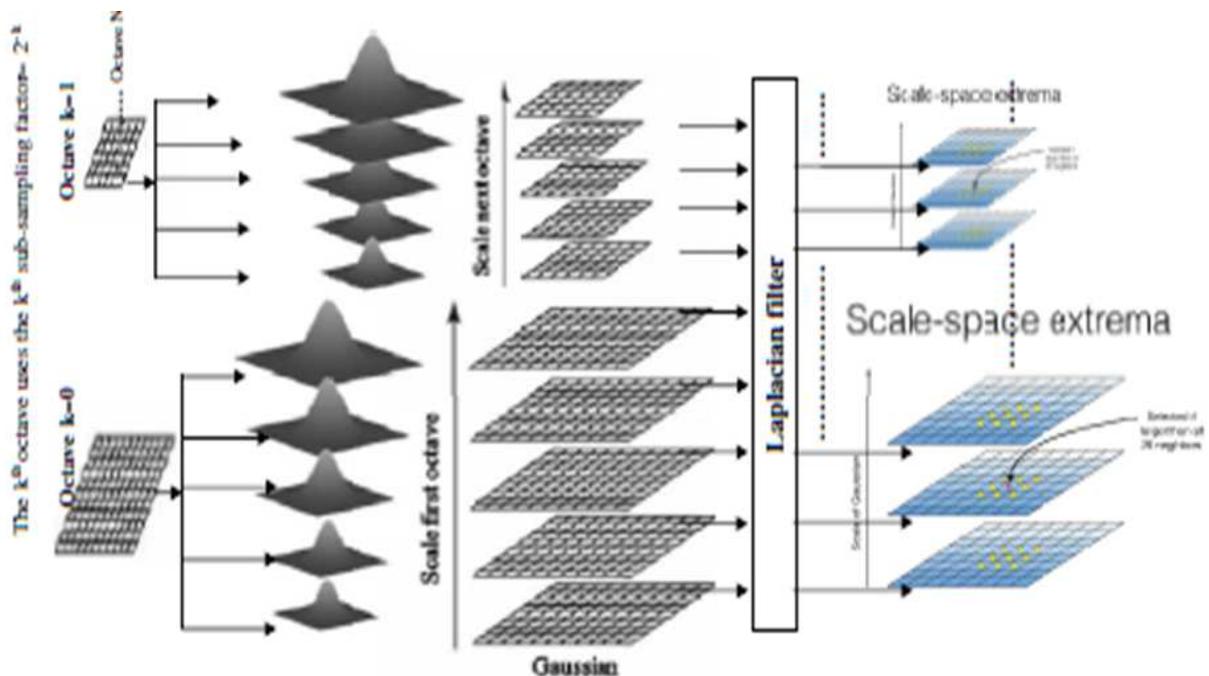
- Méthode rapide (temps réel),
- Peu sensible aux bruits.

**VIII-1-2 Détecteur SIFT-Détection des points d'intérêt :** (Scale-Invariant Features Transform)

**Idée:** Effectuer des dérivées dans plusieurs échelles. Ensuite rechercher les détails (points d'intérêt invariant en passant d'une échelle à l'autre) [4], [5].

**Questions:**

- Comment créer ces différentes échelles?
- Comment définir les détails invariants entre-échelles?



**Figure-VIII-3** Schéma synoptique pour la méthode the SIFT appliqué au *template*.

Les différentes échelles sont créées par plusieurs octaves. Chaque  $k^{\text{ième}}$  octave est un sous-échantillonnage d'image par un facteur de  $2^k$ .

Pour chaque octave, on applique plusieurs *Laplacian Of Gauss* (LOG). La variété entre les LOGs dans la même échelle réside dans la variation du paramètre  $\sigma$  définissant la bande passante du filtre GAUSS (figure-VIII-3 et figure-II-1).

Le LoG de chaque échelle donne plusieurs images contenant des détails spatiaux (obtenues par filtrage dérivateur de Laplacien). Les points qui persistent entant qu'extrema entre les images de la même échelle sont considérés des points caractéristiques de l'objet *template* et peuvent le remplacer lors de la détection [4], [5].

### **Avantages/Problèmes et limites pour SIFT :**

#### **AVANTAGES:**

- Méthodes efficace pour le matching dans les échelles différentes (rotations..etc)
- Qualité de matching meilleure pour la reconstruction 3D, estimation de mouvement..etc
- Peu sensible aux bruits.

#### **LIMITE:**

- Méthode rapide (temps réel), mais plus lourde comparativement à HARRIS

### VIII-2 But de TP-8:

- Se familiariser avec la notion de détection de points d'intérêts afin de mettre en correspondance (matching) les objets similaires dans une image,
- Exploitation de la méthode de HARRIS.

### VIII-3 Travail demandé du TP-:

1. Créer une image en utilisant la commande « checkerboard ».
2. Détecter les corners en utilisant la commande « detectHarrisFeatures » et afficher les sur la même figure que l'image.
3. En utilisant la commande « imrotate », faire une rotation de 45° de l'image de départ et refaire la question 2 sur l'image obtenue.
4. En analysant les deux résultats des questions 2/ et 3/, donner vos commentaires et conclusion vis-à-vis la détection de points d'intérêts utilisant la méthode de HARRIS.
5. Charger l'image « eight.tif ». Sans appliquer HARRIS, observer et analyser l'image ensuite prévoir la qualité de détection qui serait obtenue par l'application de la méthode de HARRIS sur cette image.
6. Appliquer la méthode de HARRIS sur l'image « eight.tif » et comparer le résultat obtenu avec le résultat prévu dans la question précédente.

### VIII-4 Solution du TP-8 :

1. Création d'une image en utilisant la commande « checkerboard » :

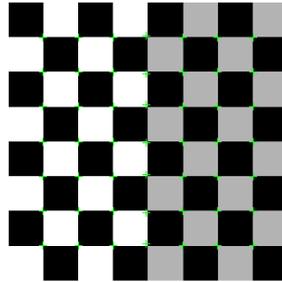
image checkerboard



Figure-VIII-4 *Template* ou objet à extraire ses points caractéristiques pour une détection.

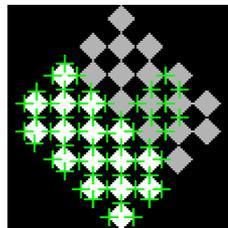
2. Détection des points caractéristiques de l'objet de l'image « checkerboard » :

Tous les points caractéristiques (corners) de l'objet « checkerboard » ont été détecté par HARRIS (figure-VIII-5).



**Figure-VIII-5** Extraction des points caractéristiques (corners) par HARRIS.

3. Faisons une rotation de 45° de l'image de départ et refaisons la question 2 sur l'image obtenue.



**Figure-VIII-6** Extraction des points caractéristiques (corners) par HARRIS de l'objet subissant une rotation.

4. Nous constatons que la méthode HARRIS n'est pas adaptée pour l'objet qui subit une rotation (figure-VIII-6) car elle ne peut pas extraire tous les points caractéristiques dans ce cas. De ce fait, la méthode indirecte de détection, HARRIS, est limitée lorsqu'il s'agit de la reconstruction 3D, estimation de mouvement..etc
5. Chargeons l'image « eight.tif ». Sans appliquer HARRIS, observons et analysons l'image ensuite prévoyons la qualité de détection qui serait obtenue par l'application de la méthode de HARRIS sur cette image :



Figure-VIII-7 Image « eight.tif ».

D'après l'analyse visuelle de l'image « eight.tif » dans la figure-VIII-7, La méthode HARRIS n'est pas adaptée pour détecter ses points caractéristiques parce que cette image n'est pas constituer de structures polygonales.

6. Appliquons HARRIS sur l'image « eight.tif » :

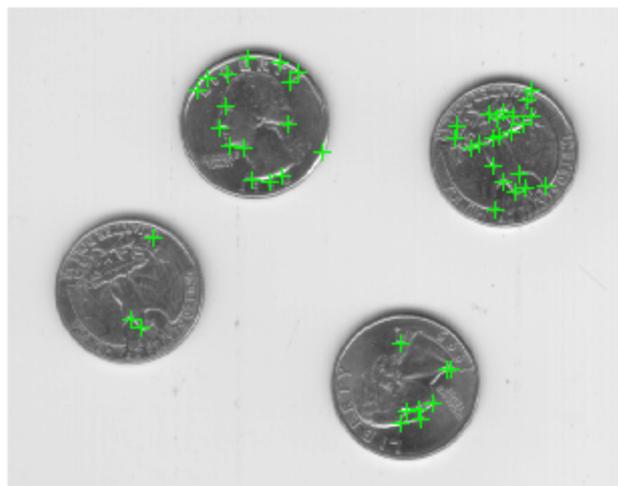


Figure-VIII-8 HARRIS appliquée sur l'Image « eight.tif » pour extraire ses points caractéristiques.

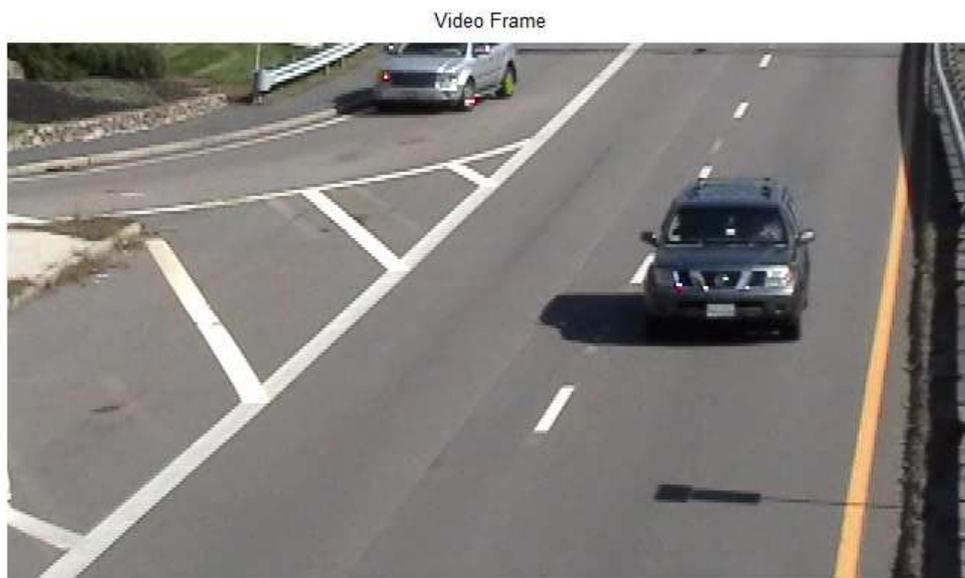
Comme il a été prévu dans la question précédente, HARRIS n'est pas adaptée pour l'extraction des caractéristiques de telle image ne contenant pas des structures polygonales. Les points détectés ne peuvent pas représenter les structures de disques de cette image.

Une autre méthode indirecte de détection, la méthode SIFT, peut donner des résultats plus satisfaisants. Celle-ci sera appliquée, dans le paragraphe suivant, sur une vidéo de surveillance routière afin de détecter et poursuivre l'objet mobile.

### **VIII-5 Annexe du TP-8 :**

#### **VIII-5-1 Exploitation de la méthode SIFT pour l'estimation de mouvement d'objet mobile:**

La méthode de détection indirecte, SIFT, est appliquée, dans l'exemple suivant, pour la détection et poursuite d'objet mobile dans une séquence vidéo. Une trame représentative de la vidéo traitée est illustrée sur la figure-VIII-9.



**Figure-VIII-9** Trame extraite d'une vidéo de surveillance routière.

Dans un premier temps (figure-VIII-10), une différence entre trames successives de la vidéo est effectuée afin de mettre les zones d'intérêt en reliefs (objets mobiles : *template*). Cette différence met l'accent sur les objets mobiles (véhicules dans cet exemple). Néanmoins, certains petit mouvements (bruits) sous forme de points blancs apparaissent dans la figure-VIII-10 et devraient d'abord être éliminés avant d'aller plus loin. L'élimination des petits mouvements (points blancs : bruits) s'effectue par opération morphologique : l'érosion. Le résultat de cette opération morphologique est concrétisé sur la figure-VIII-11.



Figure-VIII-10 Soustraction entre trames de la vidéo éliminant l'arrière plan (objets fixes) de la vidéo.

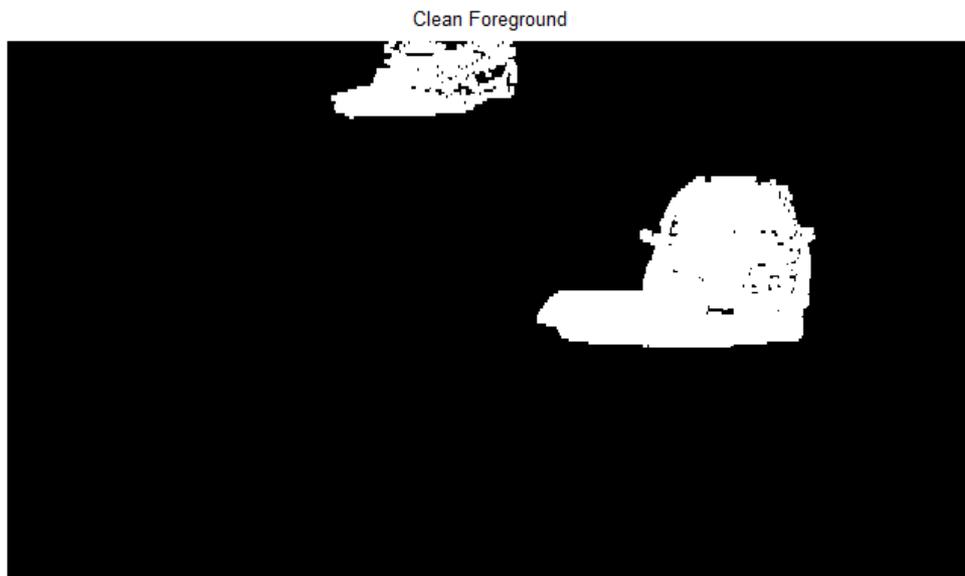
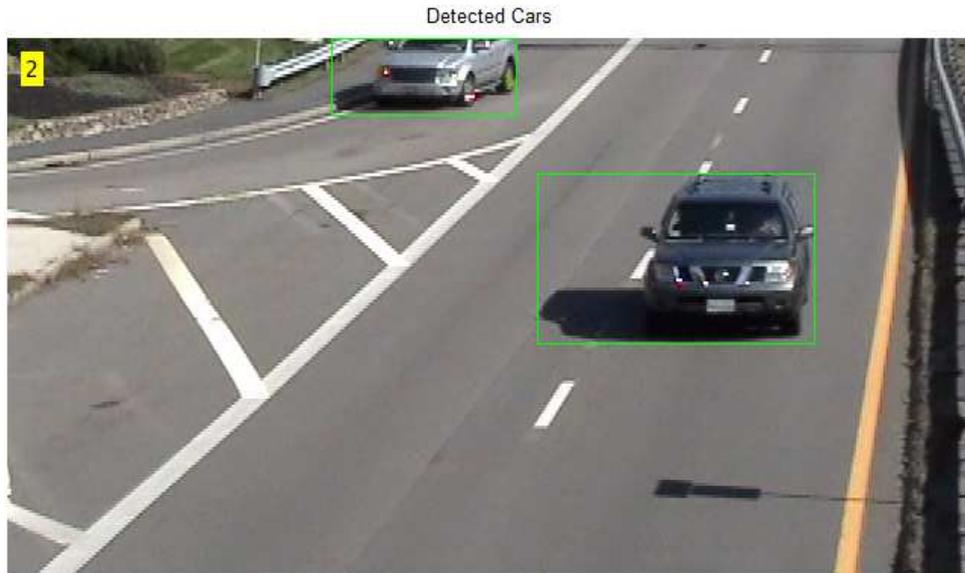


Figure-VIII-11 Elimination de bruits par opération morphologique (érosion).

Une fois, les objets d'intérêts (*template*) sont définis (figure-VIII-11), l'algorithme SIFT peut être appliqué afin d'en extraire les points caractéristiques d'objet mobile à détecter dans les trames qui suivent (poursuite d'objet mobile).

La figure-VIII-12 montre une trame représentative de la poursuite d'objet mobile où les véhicules en mouvement sont détectés et poursuivis (encadrés) le long de toute la vidéo.



**Figure-VIII-12** Trame représentative de traitement de la vidéo : Véhicule détectés utilisant l'algorithme de détection indirecte SIFT.

### IX- Conclusion générale :

Dans ce polycopié de TP-Vision Artificielle, nous avons, dans un premier temps, mis en œuvre les différents outils fondamentaux de traitement d'image (amélioration d'apparence, filtrage linéaire et non linéaire, filtrage morphologique, rehaussement de contraste..). Ces outils et avant d'être appliqués sur une image, celle-ci devrait d'abord être observée, analysée et diagnostiquée.

Ensuite, des algorithmes d'extraction de caractéristiques aidant la machine à différencier entre les composantes de l'image (outils de segmentation, méthodes de détection directes et indirectes) ont été étudiés et mis en œuvre. Grâce à l'extraction de ces caractéristiques, une machine deviendrait plus indépendante au diagnostic du programmeur (devient plus intelligente).

Enfin, les caractéristiques extraits des objets (*template*, objet mobile..) permettent de mieux détecter et localiser les objets et d'en prendre la décision adéquate en temps réel tel que pour l'estimation de mouvement.

### X- BIBLIGRAPHIE :

[1] Cocquerez and al, "Analyse d'images : Filtrage et Segmentation" , Ed. Dunod, 1995.

[2] H. Maître, "Le traitement des images", Hermes Science Publications , 2003.



[3] David A. Forsyth, Jean Ponce, “Computer Vision : a modern approach” , International Edition, Prentice Hall, 2003.

[4] David G. Lowe, « Object recognition from local scale-invariant features », dans *Proceedings of the International Conference on Computer Vision*, vol. 2, 1999.

[5] David G. Lowe, « Distinctive image features from scale-invariant keypoints », *International Journal of Computer Vision*, vol. 60, n° 2, 2004, p. 91–110.

[6] Chris Harris et Mike Stephens, « A combined corner and edge detector », dans *Proceedings of the fourth Alvey Vision Conference*, Manchester, 1988.

[7] Cédric Févotte, « Approche temps-fréquence pour la séparation aveugle de sources non-stationnaires », Thèse de Doctorat de l’Ecole Centrale de Nantes et de l’Université de Nantes, 2003.

[8] Yann Le Gall & Julien Bonnel, « Débruitage et séparation dans le plan temps-fréquence des contributions sonores de navires en mouvement », *Traitement du Signal*, octobre 2013, 30(3-4-5) :149-168. DOI : 10.3166/ts.30.149-168