

# TABLE DES MATIERES

TABLE DES MATIERES.....	i
INTRODUCTION.....	1
<b>CHAPITRE 1 : AQUISITION D'IMAGE VIDEO</b>	
1.1. Historique.....	2
1.2. Le capteur image.....	3
1.2.1. Le rapport signal sur bruit.....	4
1.2.2 La sensibilité.....	4
1.2.3. La sensibilité spectrale.....	4
1.2.4 La fonction de transfert.....	4
1.2.5. La correction Gamma .....	4
1.2.6. La résolution .....	7
1.2.7. Temps d'intégration (pour les capteurs CCD).....	7
1.2.8. La saturation.....	7
1.2.9. La rémanence.....	7
1.3. Le capteur a tube.....	8
1.3.1. Principe.....	8
1.3.2. Caractéristiques.....	8
1.3.3. Avantages.....	8
1.3.4. Inconvénients.....	8
1.4. Le capteur a CCD.....	9
1.4.1. Principe.....	9
1.4.2.1 Le transfert de trame.....	9
1.4.2.2 Le transfert interligne .....	10
1.4.2.3 Domaine d'application de chaque technologie.....	10
1.4.3. Caractéristiques.....	10
1.4.4. Avantages .....	11
1.4.5. Inconvénients .....	11
<b>CHAPITRE 2 : TRAITEMENT A PARTIR DE L'HISTOGRAMME</b>	
2.1. Les sources de dégradations .....	12

2.1.1. Bruit lié au contexte de l'acquisition.....	12
2.1.2. Bruit lié au capteur.....	13
2.1.3. Bruit lié à l'échantillonnage.....	13
2.1.4. Bruit lié à la nature de la scène.....	13
2.2. Présentation des images.....	13
2.2.1. Image à niveaux de gris.....	14
2.2.2. Histogramme d'une image.....	14
2.2.3. Histogramme cumulé d'une image.....	15
2.2.4. Image couleur.....	15
2.3. Les améliorations.....	15
2.3.1. Extension de la dynamique.....	16
2.3.2. Luminosité.....	18
2.3.3. Correction d'exposition.....	19
2.3.4. Contraste.....	20
2.3.4.1 L'étirement d'histogramme.....	20
2.3.5. Rehaussement des niveaux de gris.....	21
2.3.5.1 Inverse et dual d'un rehaussement.....	22
2.3.5.2 Exemples de fonctions de rehaussement.....	24
2.3.5.3 Rehaussement linéaire par morceaux.....	25
2.3.5.4 Egalisation d'histogramme d'une image discrète à niveau de gris.....	26
2.3.6. Amélioration du contraste par considération du voisinage.....	30
2.3.6.1 La méthode .....	31
2.3.6.2 Méthodes classiques .....	31
2.3.6.3 Méthode de Gordon .....	31
2.3.6.4 Méthode de Beghdadi .....	32

## CHAPITRE 3 : RESTAURATION PAR FILTRAGE

3.1. Le filtrage.....	36
3.1.1. Principes.....	36
3.2. Filtrage linéaire.....	36
3.2.1. Filtre par produit de convolution.....	36
3.2.1.1 Passage d'une image dans un système linéaire.....	36

3.2.1.2	Produit de convolution.....	37
3.2.1.3	Pavages et maillages.....	38
3.2.1.4	Adjacences.....	40
3.2.1.5	Voisinage.....	42
3.2.2.	<i>Les différents sortes de filtre linéaire.....</i>	44
3.2.2.1	Lissage.....	44
3.2.2.2	Filtrage gaussien.....	47
3.2.2.3	Filtre passe-haut.....	49
3.3.	Filtre non-linéaire.....	50
3.3.1.	<i>Filtre median.....</i>	50
3.3.1.1	Propriétés.....	51
3.3.2.	<i>Filtre à moyenne seuillée.....</i>	60
3.3.3.	<i>Filtre de Nagao ou filtre à sélection de voisinage.....</i>	61

## CHAPITRE 4 : APPLICATION

4.1.	Introduction.....	62
4.2.	Généralités sur MATLAB.....	62
4.2.1.	<i>Environnement.....</i>	62
4.2.2.	<i>Principe du help/demo.....</i>	62
4.3.	Simulation.....	64
4.3.1.	<i>Présentation du logiciel de simulation.....</i>	65
4.3.1.1	Lancement du logiciel.....	65
4.3.1.2	Filtrage.....	66
4.3.1.3	Contraste.....	67

CONCLUSION .....	71
------------------	----

## BIBLIOGRAPHIE



## INTRODUCTION

Depuis la naissance des signaux numériques, le traitement de l'information a connu une évolution exponentielle. Aussi, dans le domaine du traitement d'image, principalement basé sur les propriétés de l'œil humain, on est appelé à connaître une forte évolution en raison de la forte demande de qualité. Cette qualité est en effet subjective par rapport à son application. Par exemple : pour la télévision, on veut améliorer l'image en vue d'un meilleur rendu visuel, c'est-à-dire rendre l'image belle pour les téléspectateurs ; dans l'imagerie biomédicale la qualité dépend des informations que l'on veut atteindre,

Ces fortes demandes de qualité nécessitent en effet l'intégration de plusieurs disciplines, en outre, elles font appel au traitement du signal, l'analyse numérique, l'informatique, les études statistiques, l'optique, l'électronique...Le traitement d'image est donc l'ensemble des méthodes et techniques opérant sur celles-ci dans le but de rendre cette opération possible, plus simple et plus efficace, ainsi on améliore l'aspect visuel de l'image pour en extraire les informations pertinentes.

La vidéo quand à lui n'est qu'une succession d'image. Il suffit alors d'extraire chaque image et de les traiter selon notre vouloir. Contrairement à ce que l'on peut penser, les données numériques images ne sont pas forcément parfait, ils peuvent présenter plusieurs dégradations.

En effet, dans le domaine de la télévision, plusieurs traitements et prétraitements de la vidéo sont déjà réalisés par différent logiciel. Dans ce mémoire nous allons voir les différentes méthodes de prétraitement numérique et d'en apporter quelques améliorations en faisant abstraction à la numérisation et à la compression.

Le plan d'étude se divisera en quatre chapitres. Le premier chapitre nous introduira sur l'acquisition d'image vidéo, il nous décrit en quelque page les caractéristiques d'un capteur image. Le second chapitre abordera comment devra t-on traiter l'image à partir de l'histogramme afin de l'améliorer. Le troisième chapitre développera ensuite la restauration par les différents filtrages. Enfin, dans le dernier chapitre application, nous allons présenter le logiciel MATLAB suivie de la présentation de la simulation des différents traitements d'images étudiés sur les chapitres antérieurs.

# CHAPITRE 1 : ACQUISITION D'IMAGE VIDEO

## 1.1 Historique [12]

Un film est une succession d'images fixes projetées à une certaine cadence pour donner l'illusion du mouvement. La plupart des films sont projetés à la vitesse de 24 images par seconde.

Les techniques de transformation d'images animées en signaux électriques sont connues depuis la fin du XIXe siècle : en 1899, le physicien allemand Zenneck fabrique la première caméra à tube sous vide, et en 1911, apparaît le tube cathodique, issu des recherches du Russe Boris Rosing. La conjugaison de ces expériences diverses a abouti à la naissance de la télévision, souvent confondue avec celle de la vidéo. Cependant, de 1927, date de la première émission publique produite par la BBC, jusqu'à la fin des années quarante, la télévision reste cantonnée à la transmission d'images en direct, faute de procédés d'enregistrement. De ce fait, des traitements sur ses images ne sont pas encore envisageables.

Les premiers systèmes expérimentaux d'enregistrement et lecture électronique d'images sur bande magnétique utilisent des techniques empruntées à celles du son. Mais ces appareils consomment d'énormes quantités de bande, pour un résultat à peine satisfaisant. En effet, le signal vidéo étant composé d'une quantité d'informations plus de trois cents fois supérieure à celui du son (bande passante dépassant les 6 Mhz contre 20 Khz en audio), il nécessite des vitesses de défilement beaucoup plus élevées jusqu'à 914 cm par seconde.

En 1956, aux États-Unis, Ampex crée le magnétoscope à bande de deux pouces (1 pouce équivalant à 2,54 cm), fondé sur une technologie d'enregistrement et lecture radicalement différente. Plutôt que de suivre le sens de défilement de la bande, le signal se retrouve retranscrit perpendiculairement à celle-ci, grâce à un tambour tournant à grande vitesse (14 000 tours par minute) équipé de quatre têtes d'enregistrement et lecture montées sur sa circonférence. Ce système, appelé quadruplex, permet de couler de nombreuses pistes vidéo sur un espace restreint, ce qui réduit considérablement la vitesse de défilement de la bande (38 ou 29 cm par seconde).

Très vite, le magnétoscope s'impose comme un outil de montage et de production indispensable.

A son utilisation vient se greffer toute une série d'équipements offrant de nombreuses possibilités de truquages et de manipulations électroniques de l'image.

Depuis, les traitements sur l'amélioration, et sur les effets spéciaux se sont évolués, car contrairement au cinéma, les traitements requièrent un inévitable passage en laboratoire à cause de son procédé photochimique d'enregistrement.

Mais la grande révolution sur les traitements des images a vu le jour entre 1985 et 1990, les formats numériques professionnels se sont succédés. Le premier intérêt du numérique est tout d'abord d'améliorer la qualité de l'enregistrement, reproductible sans altération presque à l'infini. Mais c'est aussi surtout, l'étendue des outils de traitement désormais possibles qui lui assure une place privilégiée au sein des postes de prétraitement jusqu'aux régies de post-production. En effet, les traitements ne s'obtiennent plus par une manipulation électronique de l'image, mais par une multitude de calculs effectués sur sa composition numérique.

## **1.2 Le capteur image [1] [7] [12]**

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

D'une façon générale, un capteur image est caractérisé par les paramètres suivants :

- Son rapport signal sur bruit,
- Sa sensibilité
- Sa sensibilité spectrale.
- Sa fonction de transfert qui tient compte de la correction Gamma nécessaire.
- Sa résolution maximale.
- Son temps d'intégration.

- Son seuil de saturation.
- Sa rémanence.

### ***1.2.1. Le rapport signal sur bruit***

C'est le rapport entre l'amplitude du signal et l'amplitude du bruit. Il détermine la faculté que l'on aura à faire ressortir une information par rapport au bruit présent dans une image. Il permet alors au constructeur de caractériser le nombre de niveaux de gris effectivement discriminable par la caméra si il n'y a pas de correction du bruit.

### ***1.2.2. La sensibilité***

Elle définit une variation électrique pour un éclairage donné. Les conditions opératoires étant souvent mal définies dans les documentations données par les constructeurs, mais il est plus commode de comparer le niveau d'éclairage sur le capteur à la saturation.

Remarquons que le rapport signal sur bruit et la sensibilité sont deux paramètres en corrélation pour l'évaluation des qualités d'une caméra en présence de faible niveau de lumière.

### ***1.2.3. La sensibilité spectrale***

C'est la sensibilité du capteur en fonction de la longueur d'onde de la source lumineuse à flux énergétique constant.

### ***1.2.4. La fonction de transfert***

Relie la tension de sortie de la caméra à l'intensité lumineuse reçue par les capteurs.

### ***1.2.5. La correction Gamma***

Les photorécepteurs des caméras et des appareils photos numériques transforment l'intensité lumineuse en courant électrique; inversement, les tubes cathodiques ou les écrans plats restituent des photons à partir d'un courant électrique. Or, la réponse de ces derniers n'est pas une fonction linéaire du voltage de l'entrée, mais une fonction puissance avec un exposant, appelé « gamma », de l'ordre de 2,5 et varie selon les technologies utilisées. Si l'on veut que les intensités lumineuses produites par le tube soient proportionnelles à celles de la scène captée par le photorécepteur, il



faut corriger la non linéarité encore appelée effet gamma, cette correction s'appelle correction gamma, et on la fait toujours au niveau de l'électronique du photorécepteur.

Elle a pour équation :  $n' = n^{1/\gamma}$  ,

La correction nous permet ainsi d'avoir une relation linéaire

Avant la correction, on a

$$I_{\text{émise}} = k.n^\gamma \quad (1.01)$$

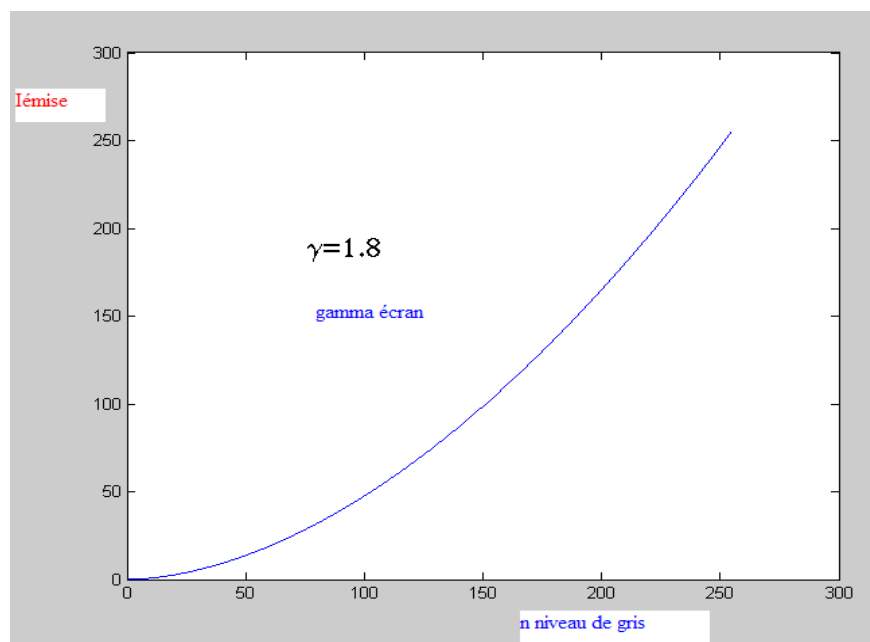


Figure 1.01: réponse du tube sans correction

Correction  $n' = n^{1/\gamma}$  (1.02)

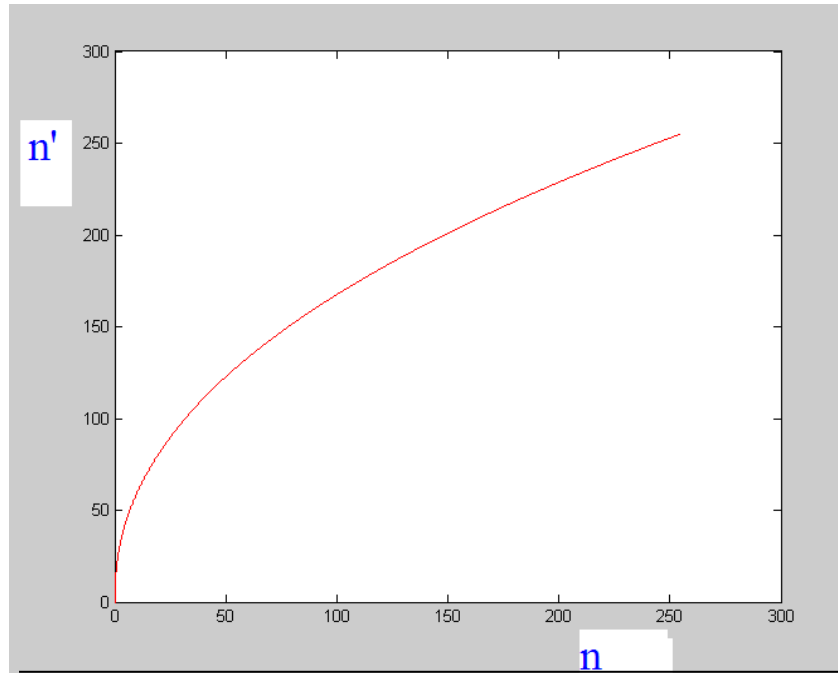
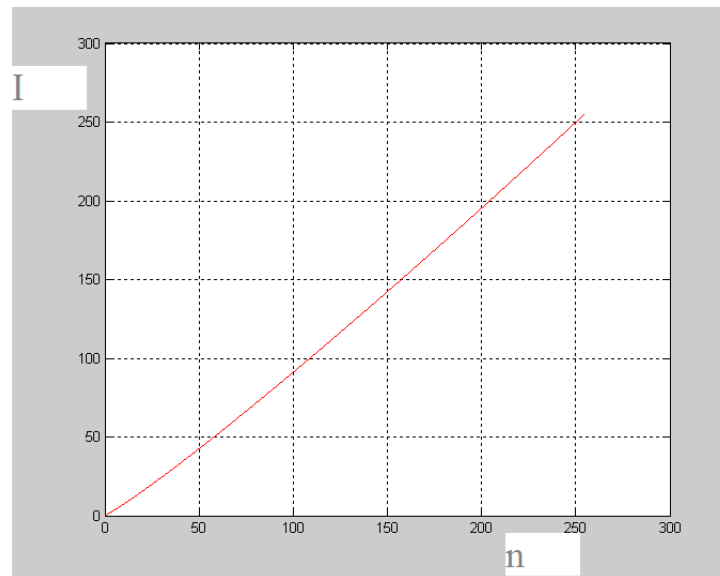


Figure 1.02 : correction gamma

Après correction on aura :

$$I_{\text{émise}} = k.n' = k.(n^{1/\gamma})^\gamma = k.n \quad (1.03)$$

Selon certaines normes, comme NTSC (standard TV des USA), on prend  $1/\text{gamma} = 0.45$ , ce qui correspond à  $\text{gamma} = 2.2$ . Cependant un gamma moyen vaut plutôt 2.45, or  $2.45 \times 0.45 = 1.1025$ . Donc l'effet global que donne la correction-gamma sera d'élever l'intensité lumineuse à une puissance d'environ 1.1, ce qui peut correspondre un peu à un étirement des contrastes .



*Figure 1.03 : réponse après correction*

### **1.2.6. La résolution**

Elle détermine la taille des plus petits détails discriminables. Habituellement cette résolution s'exprime en lignes pour les cameras à tubes et en pixels pour les cameras à CCD.

La résolution en pixels est le nombre de points élémentaires présents sur un CCD.

### **1.2.7. Temps d'intégration (pour les capteurs CCD)**

C'est le temps pendant lequel la surface sensible est exposée à la lumière avant le transfert des charges électriques. Il varie suivant les cameras de  $10^{-4}$ s à  $10^{-2}$ s. Ce temps ne doit pas être confondu avec la durée d'une trame qui est un temps de transmission d'information.

### **1.2.8. La saturation**

Par principe, les capteurs génèrent des charges électriques ou un phénomène similaire quand ils sont éclairés. Les capteurs n'acceptent qu'une certaine densité de charge par unité de surface, il existe alors un seuil à partir duquel les charges excédentaires vont se diffuser dans le matériel, ce qui modifie l'image des zones très lumineuses.

### ***1.2.9. La rémanence***

Elle exprime la persistance de charges résiduelles après transfert de l'information. Elle n'existe pas avec les CCD.

## **1.3 Capteur à tube**

### ***1.3.1. Principe***

La zone photosensible voit sa résistance électrique varier en fonction de l'éclairement. Nous nous trouvons donc avec une image virtuelle de la scène sur le capteur sous forme d'un champ de résistances. La cathode va émettre un faisceau d'électrons (donc un courant), et donc une tension aux bornes de la résistance. Il suffira donc de dévier le faisceau d'électrons de façon adéquate pour obtenir un balayage par lignes de toutes les résistances.

### ***1.3.2. Caractéristiques***

La surface photosensible peut être constituée de matériaux très divers, et dont dépendront directement les paramètres suivants :

- Sensibilité à saturation : Elle peut atteindre le 1/10 de lux
- Sensibilité spectrale : En général, le visible ou le proche de l'infrarouge mais certains matériaux dépassent ces limites
- Rémanence : Toujours présente, elle peut atteindre 400ms.

### ***1.3.3. Avantages***

- Sur le prix : à résolution égale, un tube est encore moins cher qu'un CCD.
- Sur la réponse spectrale et la sensibilité : le choix de ce matériel permet de trouver des tubes adaptées à la bande spectrale sur laquelle nous voulons travailler. En effet, la sensibilité est encore un élément qui fait parfois choisir le tube.
- Sur la résolution : certains tubes (coûteux) permettent d'obtenir des résolutions supérieures à 1000 lignes TV.

### ***1.3.4. Inconvénients***

- Sur la fragilité : Trop fragile, leur tenue aux chocs ne dépasse pas 5g.

- Sur le volume : Diamètres variant de 1 à 3cm, longueurs de 10 à 20 cm.
- Sur les tensions de commandes : Plusieurs centaines de volts.
- Sur la sensibilité aux champs magnétiques
- Sur la distorsion géométrique
- Mauvaise fidélité

## **1.4 Capteur à CCD**

### ***1.4.1. Principe***

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

### ***1.4.2. Transfert de trame et transfert interligne***

Une matrice CCD est formée de Y lignes de X colonnes, chaque pont élémentaire (ou pixel) étant physiquement un rectangle de silicium de quelques microns. Il va falloir déplacer les charges accumulées dans ce pixel avant de les convertir en tension électrique, ce qui n'est pas instantané. Il faut donc disposer d'une zone de stockage intermédiaire pour découpler le temps d'intégration du temps de transfert des données. C'est sur la définition de cette zone de stockage que deux technologies s'opposent :

#### **1.4.2.1. Le transfert de trame**

A la fin du temps d'intégration, les charges sont transférées ligne par ligne dans une zone équivalente mais protégée de la lumière par un film mélique. Les charges sont ensuite converties pixel par pixel pour donner le signal vidéo.

- Avantage :

Toute la surface du pixel est affectée à l'intégration de charges, ce qui donne la plus grande sensibilité.

➤ Inconvénient :

Pendant la durée du transfert qui est un peu longue à cause de la distance à parcourir, les pixels situés dans le haut de l'image vont être éclairés par la scène, et des charges parasites vont s'ajoutées (smearing).

#### 1.4.2.2. Le transfert interligne

Ici la zone de transfert se trouve entre les lignes, et les charges n'ont plus que quelques distances à parcourir pour être sauvegardées.

➤ Avantage :

Le phénomène de smearing est donc théoriquement supprimé.

➤ Inconvénient :

La proportion utile du pixel tombe à environ 50%, ce qui réduit d'autant la sensibilité.

#### 1.4.2.3. Domaine d'application de chaque technologie

Les CCD à transfert de trame ne sont plus conservées que pour les applications en très faible lumière (astronomie, biologie) où l'on peut accepter des temps de pose importants.

Les CCD à transfert interligne autorisent par contre des temps d'intégration très courts (inférieurs à la milliseconde), ce qui permet de figer des scènes en mouvement notamment dans le domaine de la télévision.

#### ***1.4.3. Caractéristiques***

➤ Sensibilité : 1/100 Lux

➤ Sensibilité spectrale : Le visible et l'infra-rouge

➤ Résolution : Actuellement 4096\*4096 pixels

➤ Temps d'intégration : 1/100 sec.

#### ***1.4.4. Avantages***

- Solide, léger et compact
- Aucune dérive géométrique de l'image
- Aucun brûlage
- Aucune rémanence
- Durée de vie et fidélité très fortement accrues
- Correction numérique des défauts éventuels envisageable

#### ***1.4.5. Inconvénients***

- Moins de choix en gamme spectrale.
- Sensibilité inférieure à certains tubes.
- Résolution inférieure aux tubes de haut de gamme.

## CHAPITRE 2 : TRAITEMENT À PARTIR DE L'HISTOGRAMME

Ici, le traitement, souvent appelé prétraitement, regroupe toutes les techniques visant à améliorer la qualité d'une image. De ce fait, la donnée de départ est l'image initiale et le résultat est également une image. La notion de qualité est une notion très subjective. En effet cette qualité dépend de l'objectif de l'opérateur humain. C'est la raison pour laquelle différentes techniques sont mise en œuvre.

La restauration a pour but d'inverser l'effet du phénomène dégradant. Il s'agit donc de produire une image la plus proche de la réalité physique de la scène observée. Donc la restauration d'images a pour objet la réduction, voire l'élimination des distorsions introduites par le système ayant servi à acquérir l'image. Son but est d'obtenir une image qui soit la plus proche possible de l'image idéale qui aurait été obtenue si le système d'acquisition était parfait. Le plus souvent, cette étape est la première dans la chaîne de traitements constituant un système de vision.

L'amélioration quand à elle a pour but de satisfaire l'oeil de l'observateur humain. Ici, on peut avoir une image très différente de la réalité, le seul but est de satisfaire l'œil de l'observateur. Comme par exemple, dans l'imagerie médicale, on peut être amené à renforcer une certaine zone de l'image dans le seul but d'en extraire des informations.

### 2.1 Les sources de dégradations [2] [4]

Il existe en général quatre sources de dégradation possibles :

- bruit lié au contexte de l'acquisition
- bruit lié au capteur
- bruit lié à l'échantillonnage
- bruit lié à la nature de la scène

#### 2.1.1. *Bruit lié au contexte de l'acquisition*

Dans cette première catégorie, on retrouve tous les événements inattendus, modifiant les conditions de l'acquisition du signal. L'exemple le plus proche est le bougé. On peut aussi imaginer une modification ponctuelle des conditions d'éclairage, conduisant à une sous ou sur illumination de l'objet observé. Le caractère inattendu de ces événements fait qu'il est pratiquement impossible d'en corriger les effets sans l'intervention d'un opérateur humain pour



activer le bon choix de processus de restauration par exemple :

- Correction de flou
- Correction de dérive lumineuse...

### **2.1.2. Bruit lié au capteur**

Le capteur, si il est de mauvaise qualité ou si il est mal utilisé, peut introduire toute sorte de bruits. On peut ainsi obtenir une distorsion de la gamme des niveaux de gris, provoquant une saturation ou bien une distorsion géométrique qui a le même effet visuel qu'un miroir grossissant par exemple. En effet, une mauvaise mise au point peut aussi provoquer un flou.

### **2.1.3. Bruit lié à l'échantillonnage**

Ici, l'image est une forme discrète. Il est donc nécessaire de passer du domaine continu au domaine discret, au niveau de la scène et au niveau de l'intensité lumineuse provenant de celle-ci.

De manière générale, le spectre des intensités lumineuses noir-blanc est quantifié sur 256 niveaux de gris différents nécessitant 8 bits pour le stockage en chaque point. Si le capteur est bien calibré, cette quantification ne provoque pas de bruit important. Il faut savoir que cette précision est de loin supérieure à ce que l'oeil humain utilise dans ses tâches quotidiennes.

Un objet très petit peut ne pas être représenté ou bien ne peut être représenté que par un pixel. Cet effet donne l'impression de bruit et est connu sous le nom d'effet "poivre et sel". Ce bruit est aussi généré par une texture dont les caractéristiques sont trop fines par rapport à la fréquence d'échantillonnage. Un exemple classique provient des images satellitaires lorsqu'un nuage est trop petit pour impressionner plus d'un pixel qui de ce fait a une intensité non cohérente avec son voisinage.

### **2.1.4. Bruit lié à la nature de la scène**

L'environnement dans lequel se situe l'acquisition n'est pas toujours maîtrisable. En télédétection, la couverture nuageuse dégrade la qualité des images satellites. De même, la poussière dans un atelier de production peut être difficile à éliminer

## **2.2. Présentation des images [1] [6] [11]**

La description d'une forme nécessite un certain nombre d'opération qui transforment

l'image originale en une représentation qui peut être compatible avec les appareils de visualisation et les traitements à effectuer. Dans notre cas, on a besoin d'une représentation sous forme numérique qui peut nous donner toutes les caractéristiques de l'image.

### 2.2.1. Image à niveaux de gris [6]

Ici la mise en correspondance de l'intensité lumineuse par des niveaux numériques permet des distinctions convenables entre les différents objets de l'image. De plus sur un écran de télévision ou d'ordinateur, une image est représentée par une matrice « nombre de ligne » x « nombre de colonne ». Chaque élément de la matrice, nommé pixel, représente les niveaux compris entre 0 et 255, soit 256 niveaux de gris. Le niveau de gris 0 correspond au noir tandis que 255 est représenté en blanc

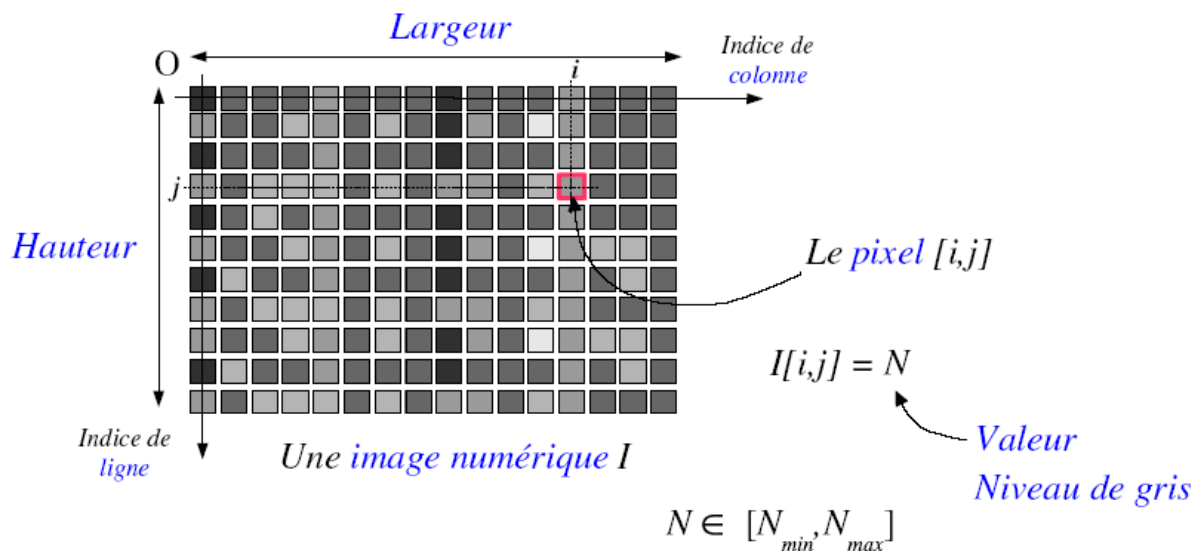


Figure 2.01 : représentation numérique d'une image

Ici, on nomme :

$$N_{max} - N_{min} = \text{nombre de niveau de gris}$$

$$\text{Log}(N_{max} - N_{min}) = \text{Dynamique}$$

### 2.2.2. Histogramme d'une image

D'après cette mode de représentation matricielle de l'image, on peut avoir un modèle statistique de l'image. Le plus utiliser est l'histogramme. En effet l'histogramme est un vecteur de dimension 256. Chaque élément  $h(i)$  représente le nombre de pixels de l'image possédant le

niveau de gris  $i$ . C'est un outil de base pour l'étude des capteurs ou de la dynamique d'une scène, il est utilisé par certains opérateurs d'analyse.

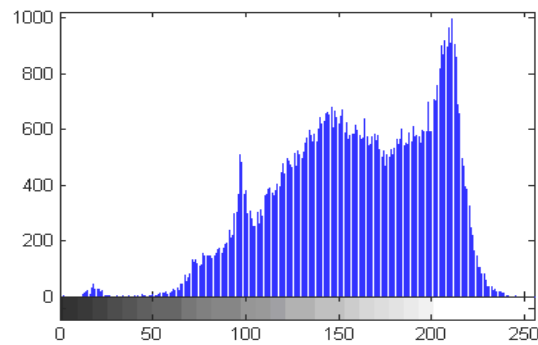


Figure 2.02 : a) une image à niveau de gris

b) son histogramme

### 2.2.3. Histogramme cumulé d'une image

C'est aussi un vecteur de dimension 256. Chaque élément  $h_c(i)$  représente le nombre de pixels de l'image possédant un niveau de gris inférieur ou égal à  $i$ . L'histogramme cumulé peut donc être assimilé, à un facteur de normalisation près, à la fonction de répartition des niveaux de gris. Celui-ci peut être estimé à partir de l'histogramme en faisant une somme discrète :

$$h_c(i) = \sum_{j=0}^i h(j) \quad (2.01)$$

$h_c(x)$  est le taux de pixels dont le niveau de gris est inférieur à  $x$ .

### 2.2.4. Image couleur

Une image couleur contient 3 plans couleurs. La plus part des cameras amènent aux trois plan Rouge, Vert, Bleu (R V B). Chaque plan est codé comme une image à niveau de gris, avec des valeurs allant de 0 à 255. Donc, on peut alors calculer l'histogramme couleur de chaque plan de la même manière que tout à l'heure.

Lorsque  $R=V=B$ , la couleur associé est un niveau de gris, D'autre part on peut passer d'une image couleur à un image à niveau de gris, on réalise alors :

$$I(y, x) = \frac{R(y, x) + V(y, x) + B(y, x)}{3} \quad (2.02)$$

## 2.3. Les améliorations

Le problème de l'amélioration est un problème subjectif. La question qui se pose est donc :  
Quand pourra-t-on dire qu'une image est améliorée ?

En effet l'oeil humain est essentiellement sensible aux forts contrastes. C'est pourquoi les techniques d'amélioration tentent d'augmenter ceux-ci dans le but d'accroître la séparabilité les régions qui composent une scène.

Différentes techniques sont possibles :

Il existe deux grandes familles de méthodes :

➤ Les méthodes globales :

Dites aussi ponctuelles, elles modifient chaque point de l'image indépendamment de ses voisins à partir d'une information globale sur l'image.

On peut distinguer :

- La modification de l'histogramme
- Le rehaussement de contraste...

➤ Les méthodes locales :

Elles travaillent sur des voisinages de pixels et donc localement dans l'image.

On distingue :

- Le filtrage fréquentiel. Cependant cette approche, au formalisme simple, met en oeuvre des processus de calculs trop lourds pour être réellement applicable.
- Le filtrage linéaire
- Le filtrage non-linéaire...

### ***2.3.1. Extension de la dynamique***

Cette méthode consiste à utiliser au mieux la dynamique de niveau de gris. Ainsi, si une image possède un niveau de gris entre  $G_{min}$  et  $G_{max}$ , on va étendre la plage des niveaux de gris pour la ramener à une dynamique comprise entre 0 et 255.

Cette étendue réduite est peut être due à un temps de pose incorrecte ou à un éclairage de la scène trop faible. La transformation mise en place est illustrée dans la *figure 2.03*.

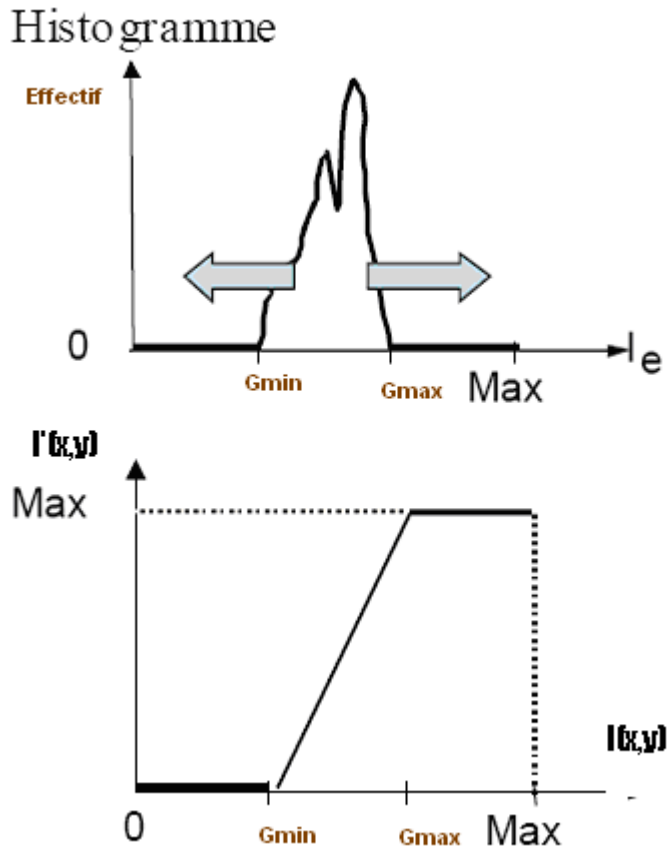


Figure 2.03 : principe de l'extension de la dynamique

On aura alors :

$$I'(x,y) = \frac{I(x,y) - G_{\min}}{G_{\max} - G_{\min}} \quad (2.03)$$



Figure 2.04 : a) image originale

b) image normalisée

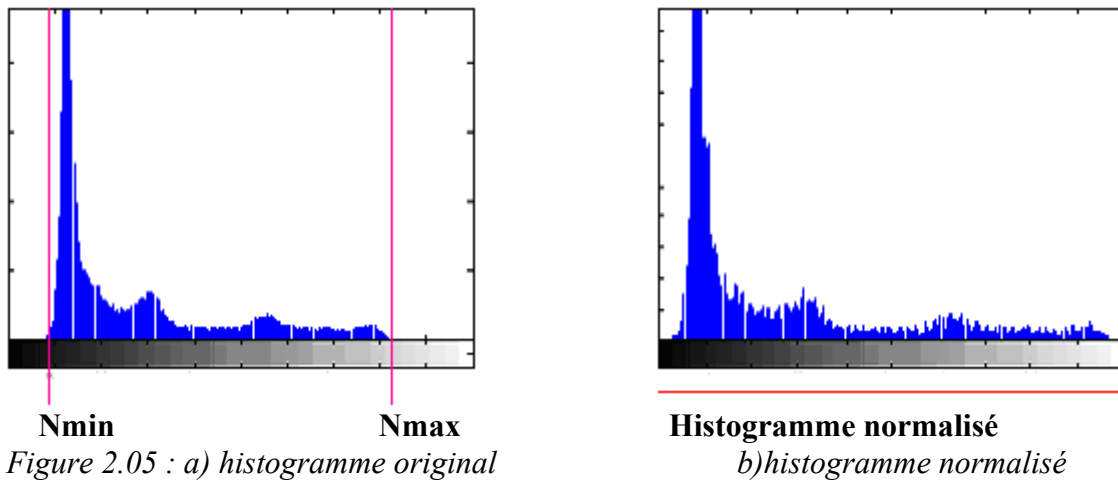


Figure 2.05 : a) histogramme original

b) histogramme normalisé

On peut remarquer sur la *Figure 2.05*, qu'après extension, tous les niveaux possibles sont utilisés

### 2.3.2. Luminosité

Afin de régler la luminosité, il suffit que l'on procède à une addition ou une soustraction d'une même constante  $C$  à tous les pixels. Lorsque des valeurs dépassent l'intervalle  $[0,255]$ , par exemple si l'on veut avoir une image un peu sombre, donc il faut soustraire, or, après soustraction, il peut y avoir des valeurs qui peuvent être négatives car la soustraction concerne tous les pixels même si ces derniers sont déjà suffisamment proche de 0. Pour cela, on met systématiquement les valeurs négatives à 0. De même pour les valeurs qui peuvent dépasser la limite de 255 après addition, il faut les forcer à 255.

➤ Augmentation de la luminosité

$$P_1 = P_{in} + C \quad (2.04)$$

$$P_{out} = P_1, \quad \text{pour } 0 \leq P_1 \leq 255 \quad (2.05)$$

$$P_{out} = 0, \quad \text{pour } P_1 \leq 0 \quad (2.06)$$

$$P_{out} = 255, \quad \text{pour } 255 \leq P_1 \quad (2.07)$$

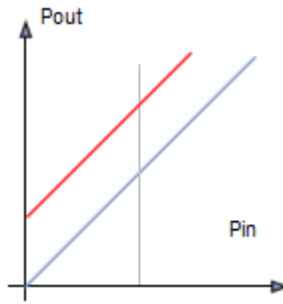


Figure 2.06 : augmentation de la luminosité

➤ Diminution de la luminosité

$$P_1 = P_{in} - C \quad (2.08)$$

$$P_{out} = P_1, \quad \text{pour } 0 \leq P_1 \leq 255 \quad (2.09)$$

$$P_{out} = 0, \quad \text{pour } P_1 \leq 0 \quad (2.10)$$

$$P_{out} = 255, \quad \text{pour } 255 \leq P_1 \quad (2.11)$$

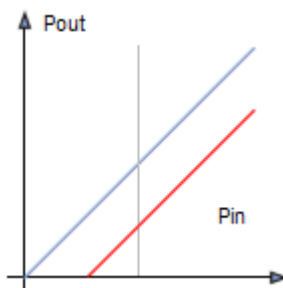


Figure 2.07 : diminution de la luminosité

### 2.1.3. Correction d'exposition

Puis que nous sommes dans la partie amélioration, on peut être alors amené renforcer certaine plage de niveau de gris, au détriment d'autres plages pour mettre certains objets en valeur. Dans ce cas, la transformation des luminances n'est plus linéaire. Ici, il faut choisir une fonction croissante qui permet de modifier la luminosité de certain niveau de gris de l'image sans affecter les autres pixels surtout les pixels noirs et blanc.

Par exemple :

$$P_{out} = 255^{n-1/n} P_{in}^{1/n} \quad (2.12)$$

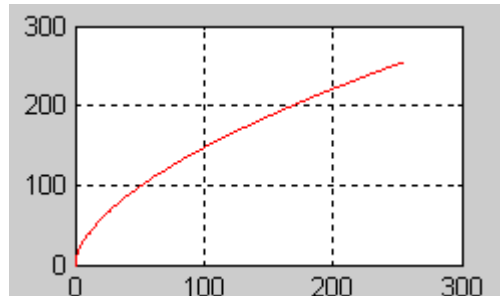


Figure 2.08 : fonction de correction d'exposition correspondant à une forte pente sur les bas niveaux

Dans cet exemple, on fait une forte correction d'exposition surtout pour les pixels de niveau proche de zéro.

### 2.3.4. Contraste

#### 2.3.4.1. L'étirement d'histogramme

Aussi appelé "*linéarisation d'histogramme*", il consiste à répartir les fréquences d'apparition des pixels sur la largeur de l'histogramme. Ainsi il s'agit d'une opération consistant à modifier l'histogramme de telle manière à répartir au mieux les intensités sur l'échelle des valeurs disponibles. De cette façon, si les valeurs de l'histogramme sont très proches les unes des autres, l'étirement va permettre de fournir une meilleure répartition afin de rendre les pixels clairs encore plus clairs et les pixels foncés proches du noir.

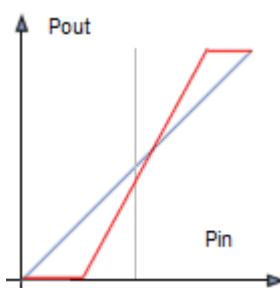


Figure 2.09 : étirement d'histogramme





$$P_{out} = \frac{255}{255 - 2d} * (P_{in} - d) \quad (2.13)$$

Ici, donnons un exemple sur les niveaux du plan rouge de l'image originale à gauche *Figure 2.10*



*Figure 2.10 : a) image originale*

*b) image étirée*



*Figure 2.11 : a) histogramme original*

*b) histogramme étiré*

Il est ainsi possible d'augmenter le contraste d'une image. Par exemple une image trop foncée pourra devenir plus "visible". Toutefois cela ne donne pas toujours de bons résultats.

### **2.3.5. Rehaussement des niveaux de gris [2]**

Soit  $M$  le niveau de gris maximum (par exemple,  $M = 255$ ). On suppose que le niveau de gris minimum est 0. Le rehaussement consiste à appliquer aux niveaux de gris de l'image une fonction croissante  $f$  telle que  $f(0) = 0$  et  $f(M) = M$ .

Le graphe de  $f$  indique le type de rehaussement appliqué à l'image: les plages de niveaux de gris où la pente de  $f$  est forte seront rehaussées au dépens de celles où cette pente est faible.

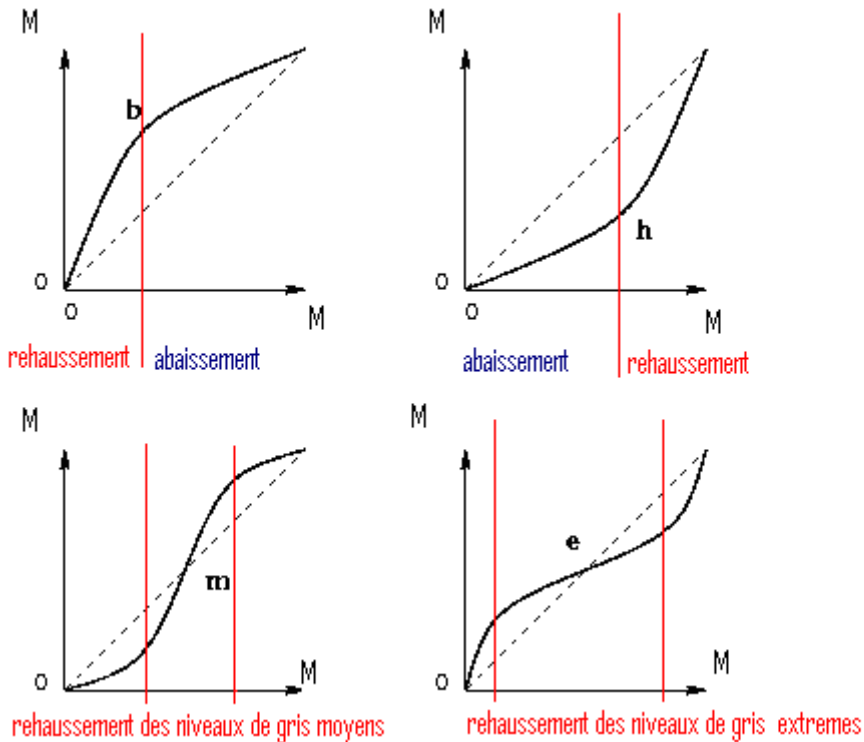


Figure 2.12 : quelques fonctions de rehaussement

Par exemple dans l'illustration ci-dessus, **b** rehausse les bas niveaux de gris (zones sombres), tandis que **h** rehausse les hauts niveaux de gris (zones claires), **m** rehausse les niveaux de gris moyens, et **e** rehausse les niveaux de gris extrêmes.

### 2.3.5.1. Inverse et dual d'un rehaussement

Soit  $f$  une fonction de rehaussement des niveaux de gris. L'inverse de  $f$  est la fonction  $f^{-1}$  définie par  $y = f^{-1}(x)$  si  $x = f(y)$ . Le graphe de  $f^{-1}$  est obtenu à partir de celui de  $f$  par symétrie par rapport à la diagonale  $y = x$ . Comme la pente de  $f^{-1}$  est l'inverse de celle de  $f$ , l'effet de rehaussement de  $f^{-1}$  est l'opposé de celui de  $f$ : là où  $f$  augmente les contrastes,  $f^{-1}$  les diminuera, et vice versa. Le dual de  $f$  est la fonction  $f^*$  obtenue en appliquant successivement un négatif,  $f$ , puis encore un négatif, en d'autres termes

$$f^*(x) = M - f(M-x)$$

Le graphe de  $f^*$  est obtenu à partir de celui de  $f$  par symétrie centrale (rotation de  $180^\circ$ ). La pente de  $f^*$  en  $x$  est celle de  $f$  en  $M-x$ , donc l'effet de rehaussement de  $f^*$  sur une plage de niveaux de gris sera celui de  $f$  sur la plage opposée. Notons enfin  $(f^*)^{-1}$ , le dual inverse de  $f$ , dont le graphe est obtenu à partir de celui de  $f$  par symétrie par rapport à la diagonale  $y = M - x$ .

Les relations d'inverse, de dual, et de dual inverse sont symétriques, c'est-à-dire

$$(f^{-1})^{-1} = (f^*)^* = (f^*)^{-*} = f.$$

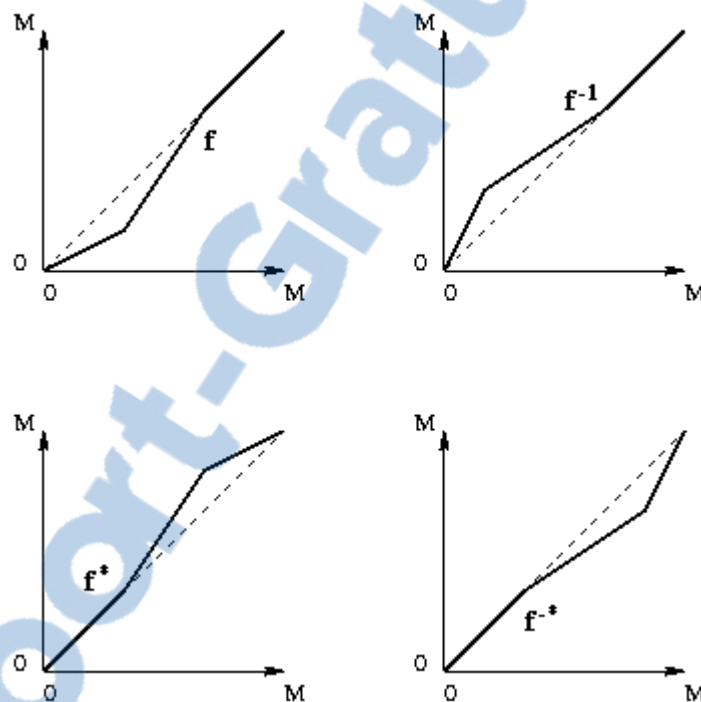


Figure 2.13 : inverses et dual d'une fonction de rehaussement

Nous illustrons dans la figure ci-dessus cette construction à partir d'une fonction  $f$  (en haut à gauche) de son inverse  $f^{-1}$  (en haut à droite), son dual  $f^*$  (en bas à gauche), et de son dual inverse  $f^{-*}$  (en bas à droite). On peut comparer les effets de chacune des quatre sur les contrastes :

- avec  $f$ , il est *diminué* dans les sombres, *augmenté* dans les moyens, et *maintenu* dans les clairs ;
- avec  $f^{-1}$ , il est *augmenté* dans les sombres, *diminué* dans les moyens, et *maintenu*

dans les clairs ;

- avec  $f^*$ , il est *maintenu* dans les sombres, *augmenté* dans les moyens, et *diminué* dans les clairs ;
- avec  $f^{*-1}$ , il est *maintenu* dans les sombres, *diminué* dans les moyens, et *augmenté* dans les clairs.

### 2.3.5.2. Exemples de fonctions de rehaussement

Nous allons définir de deux façons différentes de transformation de niveaux de gris  $\mathbf{b}$  et  $\mathbf{h}$ , où  $\mathbf{h} = \mathbf{b}^{-1}$ , la fonction inverse de  $\mathbf{b}$ , c'est-à-dire  $y = \mathbf{b}(x)$  si  $x = \mathbf{h}(y)$ . Ici  $\mathbf{b}$  rehaussera les *bas* niveaux de gris (zones *sombres*), et inversement  $\mathbf{h}$  rehaussera les *hauts* niveaux de gris (zones *claires*).

1) Soient A et B deux réels  $> 0$  tels que  $M = A \log(1 + MB)$ . Posons:

$$\mathbf{b}(x) = A \log(1 + Bx)$$

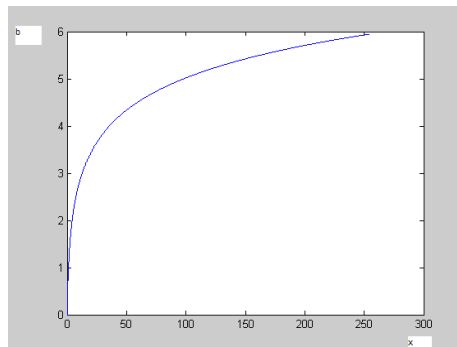


Figure 2.14 : rehaussement pour les niveaux de gris faible

$$\mathbf{h}(x) = (e^{x/A} - 1) / B$$

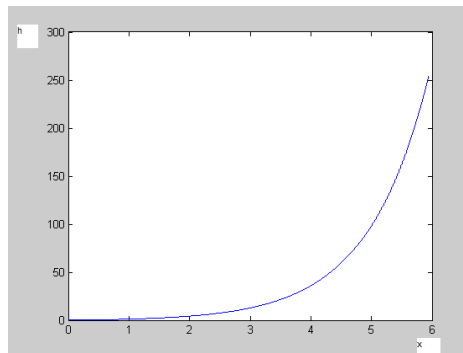


Figure 2.15 : rehaussement pour les niveau de gris élevés

Lorsque A diminue, B augmente, et l'effet de rehaussement se renforce.

### 2.3.5.3. Rehaussement linéaire par morceaux

Dans bien des cas, on peut mesurer les niveaux de gris sur différentes zones de l'image correspondant à certains objets spécifiques, qu'on comparera à des valeurs souhaitables. Par exemple sur une photo d'identité à 256 niveaux de gris, on peut avoir les niveaux de gris suivants :

Zone	niveau de gris mesuré	niveau de gris souhaitable
fond	105	70
cheveux	120	105
chemise	150	190
visage	180	225

Tableau 2.01 : table de valeur de couleur de quelques objets

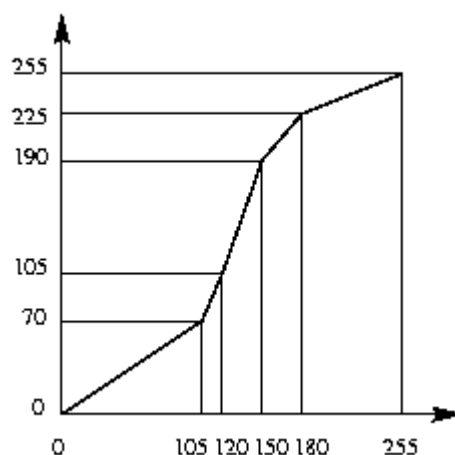


Figure 2.16 : fonction croissante par morceau

Pour rehausser l'image, on appliquera aux niveaux de gris une transformation  $f$  linéaire par morceaux définie par

$$f(0) = 0, f(105) = 70, f(120) = 105, f(150) = 190, f(180) = 225, f(255) = 255,$$

avec une interpolation linéaire entre les points de contrôle 0, 105, 120, 150, 180, et 255, comme illustré sur l'image ci-contre.

Plus généralement, pour une image à niveaux de gris compris entre 0 et M, on peut avoir  $n$

niveaux de gris  $a_1, \dots, a_n$  (tels que  $0 < a_1 < \dots < a_n < M$ ) à transformer en  $n$  niveaux de gris  $b_1, \dots, b_n$  (où l'on aura le plus souvent  $0 < b_1 < \dots < b_n < M$ ),

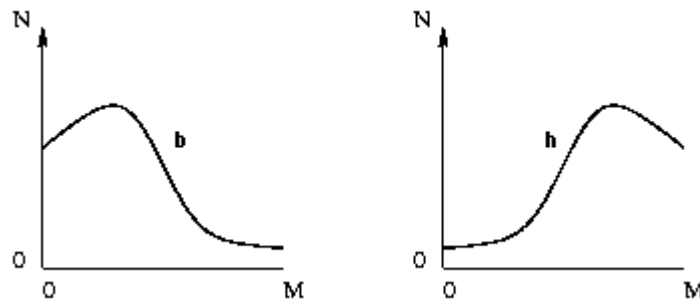
et on choisira une transformation  $f$  linéaire par morceaux donnée par

$$f(0) = 0, \quad f(a_1) = b_1, \quad \dots, \quad f(a_n) = b_n, \quad \text{et} \quad f(M) = M. \quad (2.14)$$

#### 2.1.2.6. Egalisation d'histogramme d'une image discrète à niveau de gris

L'égalisation d'histogramme est une transformation des niveaux de gris dont le principe est d'équilibrer le mieux possible la distribution des pixels dans l'histogramme. Cette opération a pour but de rendre l'histogramme le plus plat possible, c'est-à-dire qu'elle vise à augmenter les nuances dans l'image. On souhaite ainsi que chaque niveau de gris soit également représenté dans l'image.

La présentation visuelle de l'histogramme d'une image permet en effet de se rendre compte des défauts de contraste dans l'image.



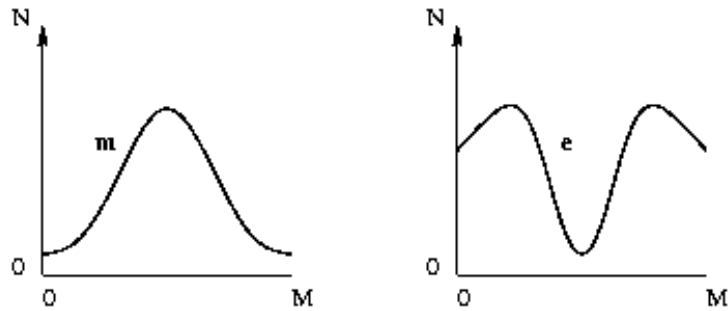


Figure 2.17 : quelques histogrammes d'images mal contrastés

Par exemple dans l'illustration de la *figure 2.17*,

- l'histogramme **b** est concentré sur les bas niveaux de gris, c'est -à- dire l'image est trop sombre;
- l'histogramme **h** est concentré sur les hauts niveaux de gris, c'est -à- dire l'image est trop claire;
- l'histogramme **m** est concentré sur les niveaux de gris moyens, c'est -à- dire l'image manque de contraste;
- l'histogramme **e** est concentré sur les niveaux de gris extrêmes, c'est -à- dire l'image manque de nuances intermédiaires entre le sombre et le clair.

Explication :

Pour ce faire, on considère des images discrètes définies sur un ensemble discret de points  $E$ , et dont les niveaux de gris sont des entiers compris entre le niveau de gris minimum 0 et le niveau de gris maximum  $M$  (par exemple,  $M = 255$ ). Donc une telle image peut être vue comme une fonction  $I$  de  $E$  vers  $\{0, \dots, M\}$ , associant à un point  $p$  de  $E$  le niveau de gris  $I(p)$ .

$$\begin{aligned}
 I : E &\rightarrow \{0, \dots, M\} \\
 p &\mapsto I(p)
 \end{aligned}
 \tag{2.15}$$

Pour un ensemble fini  $X$ , et écrivons  $\text{card}(X)$  le nombre d'éléments de  $X$  avec  $n = \text{card}(E)$ .

Soit  $H_I$  l'histogramme de l'image  $I$

On a alors  $H_I(g) = \text{card}\{p \mid I(p) = g\}$ . ( 2.16)

Par conséquent, étant donnés deux niveaux de gris a et b (avec  $a < b$ ) de l'intervalle  $\{0, \dots, M\}$ , le nombre de points de p de E dont le niveau de gris est compris entre a et b est donné par la somme :

$$H_I(a) + \dots + H_I(b) \quad ( 2.17)$$

En particulier, on a

$$H_I(0) + \dots + H_I(M) = \text{card}(E) = n \quad ( 2.18)$$

Nous définissons l'*histogramme cumulatif* de l'image I comme la fonction  $H_{ci}$  définie sur l'intervalle de niveaux de gris  $\{0, \dots, M\}$  et à valeurs entières non négatives, associant à tout niveau de gris g le nombre de points dans E ayant un niveau de gris inférieur ou égal à g dans l'image I. On a donc:

$$H_{ci}(g) = H_I(0) + \dots + H_I(g), \quad ( 2.19)$$

$$H_{ci}(0) = H_I(0),$$

Donc,

$$H_{ci}(g) = H_{ci}(g - 1) + H_I(g) \quad \text{pour } g = 1, \dots, M. \quad ( 2.21)$$

En particulier  $H_{ci}(M) = \text{card}(E) = n$ .

L'égalisation d'histogramme d'une image discrète utilise une discrétisation de la [formule obtenue dans le cas continu](#) pour obtenir un histogramme entièrement plat.

On applique alors à l'image I la fonction de rehaussement des niveaux de gris  $f$  définie par



$$f(g) = \frac{M \cdot H_{ci}(g)}{n} \quad (\text{on le fait arrondir à l'entier le plus proche}). \quad (2.22)$$

et

$$f(0) = \frac{MH_I(0)}{n} \quad (2.23)$$

$$f(g) - f(g-1) = \frac{MH_I(g)}{n} \quad \text{pour } g=1, \dots, M \quad (2.24)$$

En particulier  $f(M) = M$ .

L'effet de cette fonction de rehaussement  $f$  est de mettre à gauche de chaque niveau de gris un écart dont la taille est proportionnelle au nombre de points de ce niveau de gris dans l'image I originale. Donc tout niveau de gris fortement représenté dans l'image se trouvera écarté des niveaux de gris immédiatement plus sombres. En pratique l'image aura un aspect où les zones claires, moyennes et sombres s'équilibreront.

Bien sûr, la qualité visuelle d'une image ne dépend pas uniquement de son histogramme, mais surtout de la nature et du but de l'image. Par exemple, Dans une photo d'identité, il importe d'établir un bon contraste entre la tête et le fond.

Par exemple :



Figure2.18 : a) image originale

b) image égalisée

Image de départ

Image égalisée

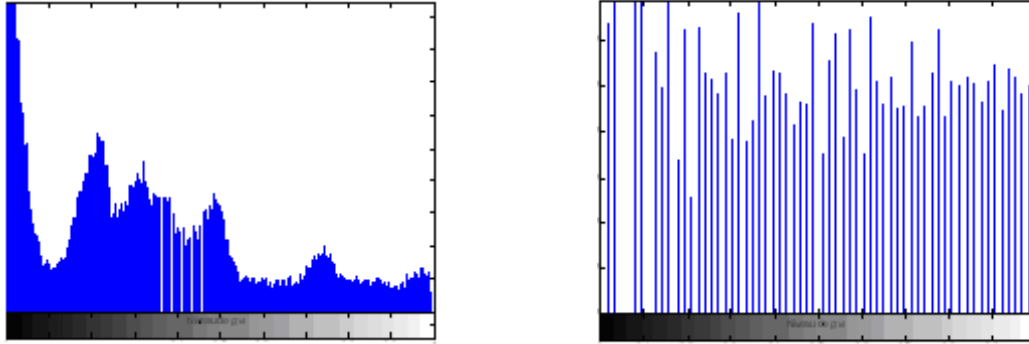


Figure 2.19 : a) histogramme de l'image originale

b) histogramme égalisé

L'histogramme obtenu après égalisation s'étale bien sur toute l'échelle des niveaux de gris avec un espacement accru.

Mais, dans une image bien contrastée, l'égalisation d'histogramme peut donner une image de qualité visuelle moindre.

### 2.3.6. Amélioration du contraste par considération du voisinage [6] [8]

Beaucoup de techniques de rehaussement de contraste ont été proposées afin de rendre optimale la qualité d'une image en jouant sur l'histogramme. Le problème de ces méthodes c'est qu'elles agissent sur le contraste sans le définir explicitement.

Ici, nous ne considérons le contraste qu'au niveau local, c'est-à-dire, associé à chaque pixel dans un voisinage donné. Le principe à adopter consiste à mesurer les différences entre le niveau de gris d'un pixel et celui de son voisinage pour les augmenter ensuite.

Gordon fut l'un des premiers à proposer une méthode basée sur ces considérations; nous étudierons dans un premier temps sa méthode puisque celle proposée par la suite reprend ses fondements.

#### 2.3.6.1. La méthode

Les problématiques dans les méthodes de rehaussement de contraste sont les suivantes :

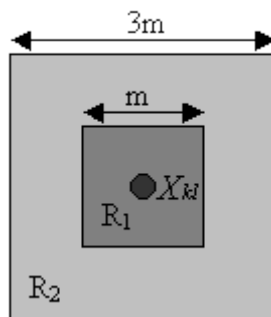
- Beaucoup d'entre elles ne définissent pas le contraste (méthodes classiques)
- Non respect des formes des objets

### 2.3.6.2. Méthodes classiques

Elles sont généralement basées sur des modifications, locales ou globales, de l'histogramme de l'image. On peut également effectuer un filtrage passe-haut de manière à accentuer les différences. Le principal problème de ces méthodes est qu'elles rehaussent le contraste sans se demander ce que c'est !

### 2.3.6.3. Méthode de Gordon

Gordon fut un des premiers à réellement proposer une définition du contraste dans sa méthode. Il le définit en considérant une fenêtre possédant deux régions qui respecte le schéma de la *figure 2.20* ci-après.



*Figure 2.20 : principe de la fenêtre de Gordon*

Il donne ainsi une définition du contraste :

$$C_{kl} = \frac{|\overline{X_1} - \overline{X_2}|}{|\overline{X_1} + \overline{X_2}|} \quad (2.25)$$

où  $X_1$  et  $X_2$  représente respectivement les niveaux de gris moyens dans les régions  $R_1$  et  $R_2$ .

Gordon mesure ainsi la diminution de la luminance dans l'image. On remarque que mise à part le cas où  $m$  est égal à 1,  $C_{kl}$  n'est pas directement relié au niveau de gris du pixel en cours. Ce moyennage va rendre la méthode inefficace pour les grandes valeurs de  $m$ .

#### 2.3.6.4. Méthode de Beghdadi

Cette méthode est très inspirée de celle de Gordon ; cependant, elle tient compte des contours des objets, ce qui la rend d'autant plus efficace.

*Définitions 2.01:*

Considérons la fenêtre carrée  $W_{kl}$  de taille  $m$  impaire centrée sur le pixel  $X_{kl}$  :

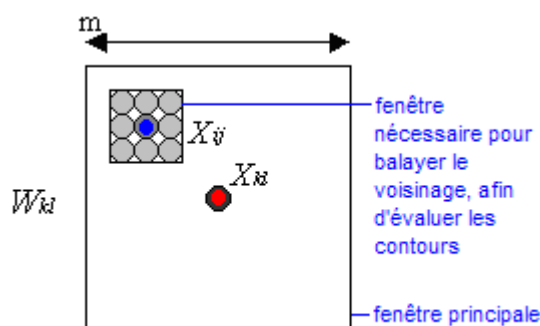


Figure 2.21 fenêtre proposée par Beghdadi

Nous utilisons ensuite l'opérateur défini de la manière suivante :

$$\Delta_{ij} = |X_{ij} - \bar{X}| \quad (2.26)$$

où  $\bar{X}$  représente le niveau de gris moyen des huit plus proches voisins de  $X_{ij}$ . L'avantage d'un tel opérateur est qu'il est maximum au niveau des fortes transitions ; de plus, le fait de considérer les huit plus proches voisins permet d'être efficace dans toutes les directions de l'espace. Ainsi, après balayage de toute la fenêtre d'analyse, nous obtenons une représentation des contours (zones de fortes transitions) à l'intérieur de cette fenêtre. En pondérant ensuite chaque pixel de la fenêtre par le  $\Delta_{ij}$  associé, nous obtenons le niveau de gris moyen situés au niveau des transitions, c'est à dire sur les contours de l'objet. Cette valeur, notée  $E_{kl}$  servira par la suite de seuil pour rehausser ou au contraire aplatir le niveau de gris de chaque pixel.

$$\overline{E}_{kl} = \frac{\left( \sum_{(i,j) \in W_{kl}} \Delta ij X_{ij} \right)}{\left( \sum_{(i,j) \in W_{kl}} \Delta ij \right)} \quad (2.27)$$

A partir de cette valeur seuil, nous pouvons proposer la définition suivante du contraste :

$$C_{kl} = \frac{|X_{kl} - \overline{E}_{kl}|}{|X_{kl} + \overline{E}_{kl}|} \quad (2.28)$$

On peut remarquer dans cette dernière expression que le contraste  $C_{kl}$  est directement relié au niveau de gris du pixel  $X_{kl}$ , mais pas avec sa moyenne. Ceci présente l'avantage de rendre la méthode de Beghdadi efficace quelque soit la taille de la fenêtre : en effet, faire varier cette dernière ne va influencer que sur la valeur de seuil  $E_{kl}$ .

Algorithme :

Tout d'abord, nous faisons passer le contraste précédemment défini d'une valeur  $C_{kl}$  à une valeur  $C'_{kl}$  en utilisant une fonction croissante sur  $[0, 1]$  respectant les conditions suivantes :

- Pour un rehaussement de contraste

$$C_{kl} \in [0,1], \text{ avec } F(C_{kl}) \geq C_{kl} \quad , \quad F(C_{kl}) \in [0,1] \quad (2.29)$$

- Pour diminuer le contraste

$$C_{kl} \in [0,1], \text{ avec } F(C_{kl}) \leq C_{kl} \quad , \quad F(C_{kl}) \in [0,1] \quad (2.30)$$

A partir de cette nouvelle valeur du contraste, on remplace  $C_{kl}$  par  $C'_{kl}$  il nous faut maintenant calculer le niveau de gris du pixel considéré ; ceci s'obtient en inversant l'expression de  $C_{kl}$  :

$$\text{Soit } C'_{kl} = F(C_{kl})$$

$$\text{On a alors } X'_{kl} = \frac{1 - C'_{kl}}{C'_{kl} + 1} \cdot \bar{E}_{kl} \quad \text{si } X_{kl} \leq \bar{E}_{kl} \quad (2.31)$$

$$X'_{kl} = \frac{1 + C'_{kl}}{1 - C'_{kl}} \cdot \bar{E}_{kl} \quad \text{si } X_{kl} \geq \bar{E}_{kl} \quad (2.32)$$

Résultats et interprétation sur quelques images tests :

Nous allons commencer par observer le comportement de l'algorithme sur des images de scènes dites "normales", en faisant varier d'une part la taille de la fenêtre, et d'autre part la fonction utilisée pour le rehaussement du contraste.



Figure 2.22 Image originale



Figure 2.23 : a)  $f(x) = 1 - \exp(-2x)$ , fenêtre 3\*3



b)  $f(x) = \ln(1+2x)$ , fenêtre 3\*3



Figure 2.24 : a)  $f(x) = 1 - \exp(-2x)$ , fenêtre 9\*9

b)  $f(x) = \ln(1+2x)$ , fenêtre 9\*9

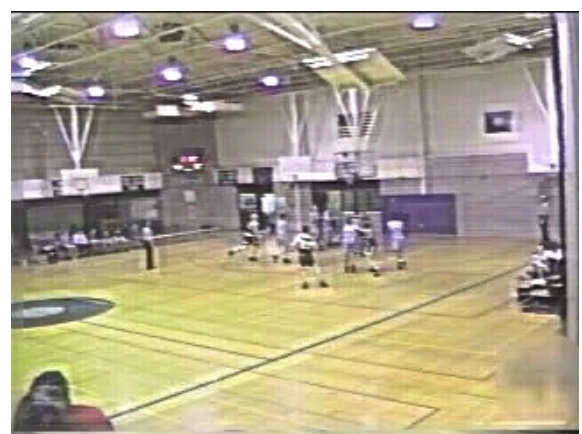


Figure 2.25: a)  $f(x) = 1 - \exp(-2x)$ , fenêtre 15\*15

b)  $f(x) = \ln(1+2x)$ , fenêtre 15\*15

On ne constate pas de grandes différences en ce qui concerne les fonctions ; plus la taille de la fenêtre augmente, plus le contraste se renforce, et on constate l'apparition d'un halo autour des objets pour une fenêtre de taille importante (15\*15).

## CHAPITRE 3 : RESTAURATION PAR FILTRAGE

### 3.1 Le filtrage

#### 3.1.1. Principes

Le filtrage d'une image consiste à appliquer des traitements locaux à cette image, c'est-à-dire que la valeur numérique d'un point va être réaffectée en fonction de ses voisins, et donc en fonction de la répartition spatiale des points et non pas en fonction des valeurs numériques des points.

La restauration de l'image est essentiellement obtenue par cette opération de filtrage. Il est utilisé pour modifier ou pour mettre en valeur une image, par exemple, on pourra filtrer une image pour accentuer certains attributs ou pour en supprimer d'autres. L'objectif d'un filtrage est d'éliminer les perturbations induites par les procédés d'acquisition d'une image tout en respectant l'intégrité de la scène originale comme les transitions entre les régions homogènes, les éléments significatifs de l'image doivent quand même être préservés au mieux.

Différentes méthodes de filtrage ont été développés suivant le type et l'intensité de bruit auxquels l'image est affectée.

Les premières et les plus simples sont basés sur le filtrage linéaire, mais les limites atteintes par ces techniques nous conduiront ensuite au développement des filtres non linéaires.

### 3.2 Filtrage linéaire [2] [8] [9]

Le filtrage linéaire d'une image peut s'envisager de deux manières.

- Le filtrage peut tout d'abord se réaliser dans le domaine spatial en faisant le produit de convolution
- ou peut se réaliser dans le domaine fréquentiel en affectant la TF de l'image par la fonction de transfert du filtre.

$$x(t) \quad , \quad h(t) \quad ; \quad y(t) = x(t) * h(t)$$

$$X(f) \quad , \quad H(f) \quad ; \quad Y(f) = X(f).H(f)$$

Mais, ici nous allons nous pencher sur les filtrages par convolution.

#### 3.2.1 Filtre par produit de convolution



### 3.2.1.1 Passage d'une image dans un système linéaire

Propriétés :

Soient  $g_1(x,y)$  et  $g_2(x,y)$  les images de sortie du système relatives aux images d'entrée  $f_1(x,y)$  et  $f_2(x,y)$ .

Si le filtre est linéaire, on a les propriétés suivantes :

➤ Linéarité

Toute combinaison linéaire  $a.f_1(x,y)+b.f_2(x,y)$  produit en sortie après filtrage.

$$a.g_1(x,y)+b.g_2(x,y)$$

➤ Invariance

$$f(x-t, y-t) \longrightarrow g(x-t, y-t)$$

### 3.2.1.2. Produit de convolution

La convolution est définie par la relation suivante :

$$\begin{aligned} g(t) &= f(t) * h(t) \\ &= \int_{-\infty}^{+\infty} f(\tau)h(t-\tau)d\tau \end{aligned}$$

Une extension dans les deux dimensions s'écrit :

$$g(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\alpha, \beta).h(x-\alpha, y-\beta).d\alpha .d\beta \quad (3.01)$$

En ce qui concerne la convolution dans le domaine du traitement d'image, chaque pixel est remplacé par une combinaison linéaire des pixels qui l'entourent. En effet un produit de convolution est une opération mathématique que l'on utilise pour multiplier des matrices entre elles.

Dans le cas qui nous intéresse, nous mettons en jeu deux matrices très différentes :

- Une matrice image très grande
- Une matrice plus petite qu'on appelle le noyau, parce que c'est le cœur de tous les changements qui vont affecter l'image.

Ce noyau représente le filtre elle même, il est connu sous le nom de convolution de Kernel. Le noyau va donc agir sur chacun des pixels c'est à dire sur chacun des éléments de la matrice image :

Pour le cas d'une image numérique, on prend le produit de convolution dans le cas discret

$$g(x,y) = \sum_u \sum_v h(u,v).f(x-u, y-v) \quad (3.02)$$

$$g(x, y) = \sum_u \sum_v f(u, v).h(x - u, y - v) \quad (3.03)$$

Un filtre est constitué d'une matrice carrée comportant un nombre impair de lignes et de colonnes et un pixel central auquel va être appliqué le filtrage. Le principe consiste à remplacer la valeur du pixel central par une combinaison prédéfinie des valeurs des pixels adjacents sur la matrice.

➤ Fonction linéaire sur les valeurs en Ng des pixels voisins :

$$g(x, y) = \frac{1}{D} \sum_u \sum_v f(u, v).h(x - u, y - v) \quad (3.04)$$

Avec D facteur de division qui représente la somme de tous les coefficients

➤ On peut alors représenter les coefficients et le facteur de division par la matrice ci-dessous:

$$\frac{1}{D} \begin{bmatrix} h(-1, -1) & h(0, -1) & h(1, -1) \\ h(-1, 0) & h(0, 0) & h(1, 0) \\ h(-1, 1) & h(0, 1) & h(1, 1) \end{bmatrix} \quad (3.05)$$

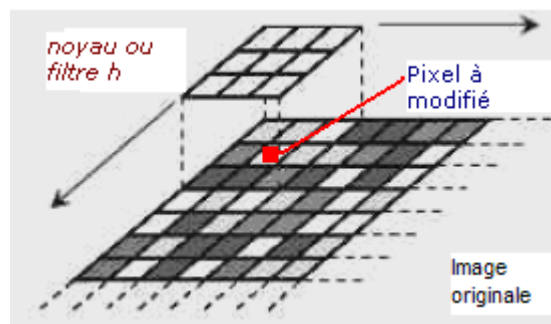


Figure3.01 : illustration du principe du filtrage

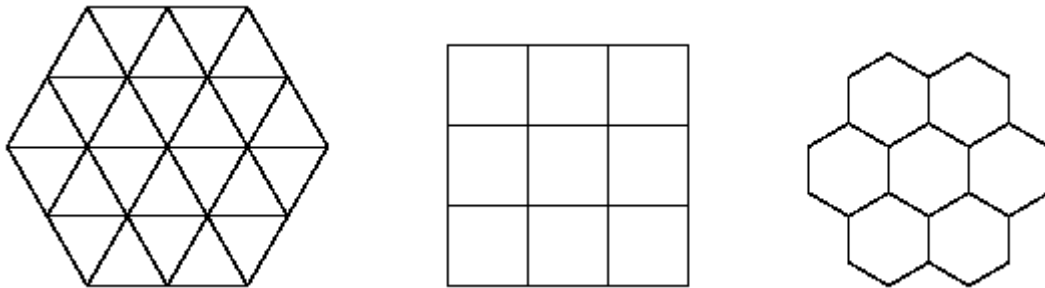
Selon les différentes formes de  $h$  on peut alors définir plusieurs variantes de filtres linéaires.

### 3.2.1.3. Pavages et maillages

Une abstraction mathématique des pixels comme taches lumineuses consiste à considérer qu'ils forment des ensembles connexes de points du plan euclidien, de telle sorte que deux pixels voisins ne peuvent s'intersectés que sur leur bord, et que l'ensemble des pixels recouvre le plan.

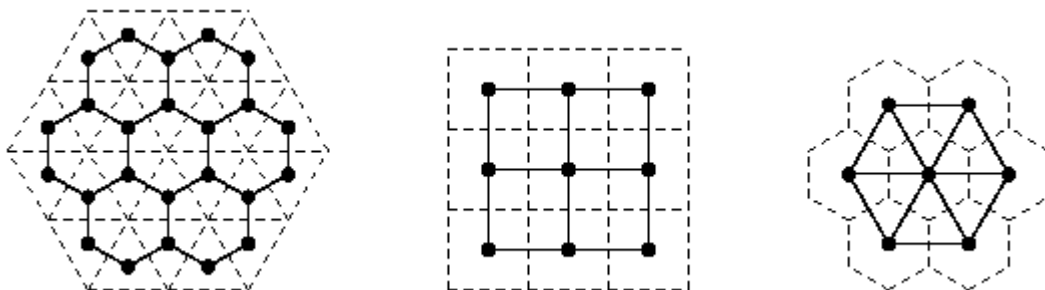
Une telle décomposition du plan s'appelle un *pavage*.

Il n'y a que 3 types de pavages dont les pavés sont des polygones réguliers : triangulaire, carré, et hexagonal. Nous les illustrons ci-dessous :



*Figure 3.02 : les différents sortes de pavages*

On place un point au centre de chaque pavé, et on joint par une ligne ceux parmi ces points dont les pavés correspondants se touchent par un côté. Cela donne le *maillage* correspondant au pavage, comme illustré ci-dessous :



*Figure 3.03 : maillages correspondants à chaque pavage*

Comme on peut le voir, le pavage carré donne un maillage carré, tandis que le pavage triangulaire donne un maillage hexagonal et vice versa. C'est ce qu'on appelle la *dualité* entre pavages et maillages.

Les centres des pavés seront les points discrets correspondant aux pixels. Pour les pavages carré et hexagonal, ces points forment un *réseau*, c'est-à-dire, ils coïncident avec les points à coordonnées entières selon deux axes ; ces axes sont orthogonaux pour le pavage carré, et forment un angle de  $60^\circ$  ou de  $120^\circ$  pour le pavage hexagonal, comme on le voit ci-dessous :

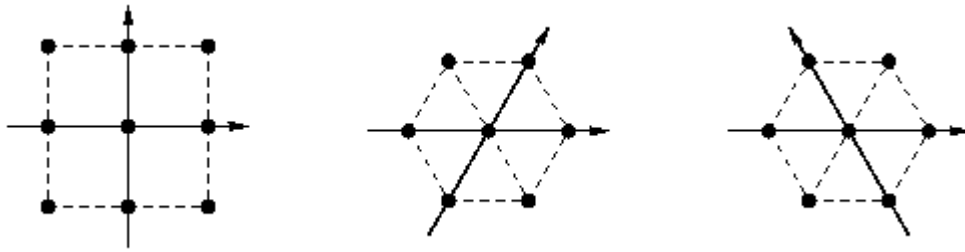


Figure 3.04 : repère dans un maillage

Pour le pavage triangulaire il est trop difficile de faire le repère des pixels voisins. Par conséquent, on n'utilise jamais le pavage triangulaire comme modèle de pixels.

Dans le pavage carré, chaque pavé a 8 pavés voisins se répartissant en deux types, à savoir 4 pavés le touchant par un côté et 4 le touchant par un sommet. En d'autres termes, dans le maillage carré, chaque point a 8 voisins, 4 selon les axes et 4 selon les diagonales. Par contre dans le pavage hexagonal, chaque pavé a 6 pavés voisins tous du même type, qui sont les 6 pavés le touchant par un côté. Donc dans le maillage triangulaire correspondant au pavage hexagonal, chaque point a 6 voisins, tous du même type.

Par conséquent le pavage hexagonal a la topologie la plus simple du point de vue mathématique et algorithmique (6 voisins au lieu de 8, et d'un seul type au lieu de deux). De plus, il se prête mieux à la modélisation de phénomènes naturels. Par exemple sur un réseau triangulaire, la modélisation discrète de la dynamique des molécules d'un fluide (liquide, gaz) permet de retrouver à l'échelle macroscopique les lois de la physique des fluides ; ce n'est pas le cas si on prend un maillage carré.

Cependant, on utilise presque toujours le pavage et le maillage carrés, car ils correspondent à nos habitudes cartésiennes. En particulier les points lumineux d'un écran sont toujours disposés suivant un maillage carré.

#### 3.2.1.4. Adjacences

Un point du maillage sera appelé *pixel*. Comme le maillage forme un réseau, les pixels sont les points à coordonnées entières selon deux axes. Donc tout pixel se code comme un couple  $(i, j)$  d'entiers, et l'ensemble des pixels correspond à  $\mathbf{Z}^2$ . En maillage carré il est d'usage de prendre le

premier axe  $i$  orienté vers le bas et le deuxième axe  $j$  orienté vers la droite ; ainsi les coordonnées correspondent à la notation matricielle, le pixel  $(i,j)$  se trouvant à l'intersection de la ligne  $i$  et de la colonne  $j$ .

Dans un maillage carré (aussi appelé *grille*), tout pixel a 2 types de voisins, à savoir ses 4 voisins selon les axes, et ses 4 voisins selon les diagonales. La relation de proximité entre deux voisins axiaux est plus forte que celle entre deux voisins diagonaux. Par conséquent nous définissons deux relations d'adjacence sur les pixels de ce maillage : deux pixels  $p$  et  $q$  sont dits

- *4-adjacents* s'ils sont voisins suivant un axe,
- *8-adjacents* s'ils sont voisins suivant un axe ou une diagonale.

Les nombres 4 et 8 correspondent au nombre de pixels adjacents à un pixel donné pour le type d'adjacence choisi. Pour le maillage triangulaire (correspondant au pavage hexagonal), il n'y a qu'une seule relation d'adjacence, appelée *6-adjacence*. Les 3 relations d'adjacence sont illustrées ci-dessous :

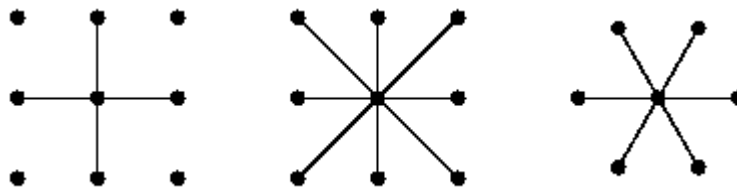


Figure3.05 : adjacence selon les maillages considérés

Etant donné un pixel  $(i,j)$ , en maillage carré les pixels adjacents à  $(i,j)$  sont :

- $(i+1,j)$ ,  $(i-1,j)$ ,  $(i,j+1)$ , et  $(i,j-1)$  pour la 4-adjacence ;
- les mêmes, plus  $(i+1,j+1)$ ,  $(i+1,j-1)$ ,  $(i-1,j+1)$ , et  $(i-1,j-1)$  pour la 8-adjacence.

On vérifie les relations suivantes :

- $(i,j)$  est 4-adjacent à  $(i',j')$  si et seulement si  $|i-i'| + |j-j'| = 1$ .
- $(i,j)$  est 8-adjacent à  $(i',j')$  si et seulement si  $\max(|i-i'|, |j-j'|) = 1$ .

En maillage triangulaire, les pixels 6-adjacents à  $(i,j)$  sont :

- $(i+1,j)$ ,  $(i-1,j)$ ,  $(i,j+1)$ ,  $(i,j-1)$ ,  $(i+1,j-1)$ , et  $(i-1,j+1)$  pour des axes formant un angle de  $60^\circ$  ;
- $(i+1,j)$ ,  $(i-1,j)$ ,  $(i,j+1)$ ,  $(i,j-1)$ ,  $(i+1,j+1)$ , et  $(i-1,j-1)$  pour des axes formant un angle de  $120^\circ$ .

### 3.2.1.5. Voisinage

La notion de voisinage est important dans le domaine du filtrage car il faut savoir à combien de voisin on va effectuer la combinaison. En effet, le voisinage d'un pixel est un ensemble de réflexif:

A dans voisinage de B  $\longrightarrow$  B dans voisinage de A

Dans le domaine du filtrage, on définit de façon classique un voisinage en 4 connexités, en 8 connexités, et ainsi de suite

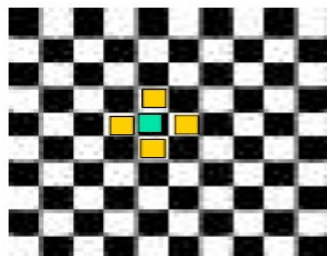


Figure3.06 : voisinage en 4- connexité (en croix, 4 voisins)

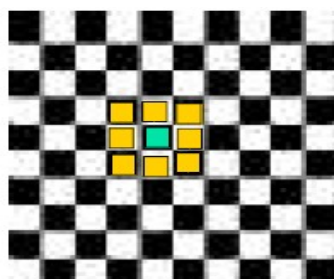


Figure3.07: voisinage en 8- connexité (+ diagonale, 8 voisins)

Le principe du filtrage linéaire est de remplacer le niveau d'un pixel par une combinaison linéaire

des niveaux des pixels environnants.

Exemple : Pour un filtre linéaire avec le voisinage à 4-connéxité nous proposons alors un exemple d'algorithme qui filtre chaque pixel en fonction des 4 pixels voisins.

Algorithme :

- pour j allant de 1 à hauteur- 1 FAIRE
  - pour i allant de 1 A largeur- 1 FAIRE
    - P0 = ima.Pixel(i,j)
    - P1 = ima.Pixel(i,j- 1)
    - P2 = ima.Pixel(i- 1,j)
    - P3 = ima.Pixel(i+ 1,j)
    - P4 = ima.Pixel(i,j+ 1)
    - $P = \text{filtrage}(P0, P1, P2, P3, P4)$
    - res.Pixel(i,j) = P
- FIN

Cette combinaison linéaire est usuellement représentée par le masque.

	P1	
P2	P0	P3
	P4	

Figure3.08 : masque à voisinage en 4-connéxité

Considérons par exemple un pixel (x, y) de l'image de niveau I(x,y). On peut remplacer la couleur de ce pixel par une combinaison linéaire des couleurs de tous les points immédiatement voisins ( y compris lui-même) :

$$g(x, y) = \frac{1}{9} \left[ \begin{array}{l} I(x- 1, y- 1) + I(x- 1, y) + I(x- 1, y+ 1) + I(x, y- 1) + I(x, y) + I(x, y+ 1) + \\ I(x+ 1, y- 1) + I(x+ 1, y) + I(x+ 1, y+ 1) \end{array} \right] \quad (3.06)$$

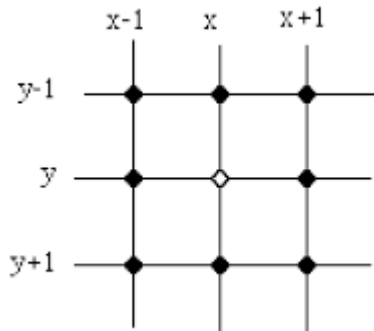


Figure 3.09 : repérage

ou encore :

$$g(x, y) = \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} h_1(i, j) \cdot I(x+i, y+j) \quad (3.07)$$

Ici, on a donc le noyau  $h_1$  qui est défini par la matrice :

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

On peut généraliser le procédé : pour un filtre  $h_p$  de taille  $(2n+1) \times (2n+1)$  :

Les matrices ci-dessous donnent des exemples de filtres linéaires où le pixel central est doté d'un poids plus important que les pixels périphériques :

$$h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_3 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

### 3.2.2 Les différents sortes de filtre linéaire

#### 3.2.2.1 Lissage



But :

On peut considérer une image comme étant constituée de plusieurs zones homogènes représentant les objets. Dans la réalité, des fluctuations des niveaux de gris sont présentes à cause du bruit. On cherchera donc à diminuer l'amplitude de ces perturbations, sans toucher aux zones de transitions.

3.1.4.2. Filtre moyennneur :

L'idée est de réaliser une moyenne des niveaux de gris autour du pixel central. Pour cela, on peut utiliser un noyau du type:

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

Exemple:

Soit le morceau d'image originale ci-dessous

30	50	60
40	100	30
30	60	10

Avec ce type de masque

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

On aura alors les différentes étapes suivantes:

- La valeur centrale 100 sera remplacée par  
 $(100*1)+(30*1)+(50*1)+(60*1)+(40*1)+(30*1)+(30*1)+(60*1)+(10*1)= 410$
- La somme est normalisée :  $X= 410/9 = 56.6$
- La valeur 100 de départ, deviendra donc 56.6, et la nouvelle fenêtre image sera :

30	50	60
40	56.6	30
30	60	10

- Une fois ce calcul effectué, le filtre passera à la valeur suivante en colonne (ici 30) et appliquera le calcul par rapport aux voisins immédiats de 30 en ayant bien évidemment prit la valeur d'origine du point précédent et non la valeur filtrée.
- Au bout de l'avant dernière colonne, il ne prend que les valeurs où le filtrage est possible, et donc, une image filtrée possède systématiquement des marges non-filtrées, il passera à la ligne suivante, en occurrence ici, à la valeur 60.

Pour définir un filtre :

- On détermine le type de filtre à appliquer
- L'image sur lequel l'appliquer
- La taille de la fenêtre filtrante. Pour un filtre 3 x 3 : 1 pixel de part et d'autre du pixel transformé, soit 9 pixels. Pour un filtre 5 x 5 : 2 pixels de part et d'autre du pixel transformé, soit 25 pixels. Pour un filtre 7 x 7 : 3 pixels de part et d'autre du pixel transformé, soit 49 pixels...

La taille du masque est alors un paramètre variable qui plus le masque sera de grande dimension, plus l'effet du filtrage sera fort. De manière générale, on essayera de réaliser un filtrage qui a le même effet dans toutes les directions. Pour cela, le voisinage considéré devra avoir une forme circulaire:

$$\frac{1}{29} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Afin de ne pas introduire de gain sur l'information portée par la fonction image. Il faut que la somme des éléments du masque soit égale à 1.



Figure 3.10 : a) image bruitée

b) image filtrée

Bien que cette méthode soit très simple à mettre en oeuvre, elle possède un inconvénient majeur, en effet ce filtrage introduit un effet de flou et les contours sont dégradés.

### 3.2.2.2. Filtrage gaussien

L'usage du filtrage gaussien pour lisser une image est devenu extrêmement populaire. En effet, par rapport au filtre moyenneur, le filtre gaussien accorde une grande importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci. On utilise alors la fonction de Gauss pour avoir les éléments du masque à utiliser

$$h(x, y) = g_{2D}(x, y) = \left( \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{x^2}{2\sigma^2}\right)} \right) \cdot \left( \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{y^2}{2\sigma^2}\right)} \right) \quad (3.08)$$

De plus, l'intérêt du filtre gaussien et que l'on règle très facilement le degré de filtrage à travers le paramètre  $\sigma$ .



Figure 3.11 : a) image bruitée

b) filtre avec  $\delta=0.2$



Figure 3.12 : a) image bruitée

b) filtre avec  $\delta=0.5$

Le seul inconvénient majeur est qu'il a le même défaut que le filtre moyenneur par le non respect des contours en introduisant un flou. En effet, le filtrage s'accompagne d'un étalement des transitions. La détermination des coefficients du filtre résulte ainsi d'un compromis entre filtrage et dégradation.

Remarquons que comme pour le filtre moyenneur, plus la taille de la fenêtre filtrante est grande plus l'effet du filtrage sera fort.

Notons que dans MATLAB, on a une fonction qui permet de régler la taille et le  $\sigma$ .

$H = \text{fspecial}('gaussian', \text{HSIZE}, \text{SIGMA})$  retourne un filtre passe-bas Gaussien symétriques par rapport au point central, avec une dimension qui est  $\text{HSIZE}$  avec un niveau de déviation  $\text{SIGMA}$  (positif).  $\text{HSIZE}$  peut être un vecteur qui spécifie le nombre de lignes et colonnes dans  $H$  ou un scalaire dans lequel  $H$  est une matrice carrée.

Par défaut le  $\text{HSIZE}$  est  $[3\ 3]$ , et le  $\text{SIGMA}$  est 0.5.

On a alors :

$$H = \begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$

### 3.2.2.3. Filtre passe-haut

Opération inverse des lissages. Sa fonction est d'accentuer les différences entre les pixels voisins, en particulier les contours des zones homogènes.

Rehaussement :

Le rehaussement de fréquence est un filtre de type "passe-haut" permettant l'amélioration les contours et détails d'une image. Il met en valeur les pixels centraux apparemment uniformes.

C'est un opérateur qui remplace le pixel central par la somme de la différence avec ses voisins.

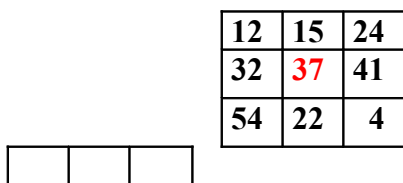


Figure 3.13 : Image originale

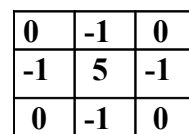


Figure 3.14 : Filtre



$$f(37) = (0 * 12) + (-1 * 15) + (0 * 24) + (-1 * 32) + (5 * 37) + (-1 * 41) + (0 * 54) + (-1 * 22) + (0 * 4)$$

$$f(37) = 75$$

la nouvelle fenêtre image sera :

12	15	24
32	75	41
54	22	4

Figure 3.15 : nouvelle image

### 3.3 Filtre non-linéaire

#### 3.3.1. Filtre médian [2] [4] [5]

Les filtres non-linéaires ont été développés pour palier aux insuffisances des filtres linéaires, principalement la mauvaise conservation des contours. Un des filtres non linéaire le plus connus est le filtre médian.

Le filtre médian a été proposé par Tukey en 1970. Ce filtre est basé sur la notion du médian des éléments où dans une telle approximation, on peut remplacer un élément par le médian des éléments considérés. Il peut être définie comme un élément séparant l'ensemble des éléments en deux parties égales.

Le filtre médian est capable de réduire certains types de bruit en dégradant très peu les contours, il a surtout pour but de supprimer les bruits impulsionsnels, qui ne sont que les points présents dans une image qui sont très différents de leurs voisins.

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

Par exemple, soit  $n = 9$ , et considérons les 9 valeurs

17, 1, 3, 5, 3, 1, 12, 2, 7.

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

En les ordonnant de la plus petite à la plus grande, on obtient la suite

1, 1, 2, 3, 3, 5, 7, 12, 17.

La valeur au milieu de cette suite ordonnée est la 5ème, à savoir 3, qui est donc la médiane des 9 valeurs.

On écrira  $\text{med}(x_1, \dots, x_n)$  la médiane des valeurs  $x_1, \dots, x_n$ . Ainsi dans l'exemple ci-dessus, on a

$$1, 1, 2, 3, \mathbf{3}, 5, 7, 12, 17 \longrightarrow \text{med}(17, 1, 3, 5, 3, 1, 12, 2, 7) = 3$$

Notons que, tout comme la moyenne, la médiane ne dépend pas de l'ordre dans lequel sont données les  $n$  valeurs.

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

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

En d'autre terme, le calcul de  $J(p)$  se fait en [parallèle](#), et le calcul en un pixel  $p$  n'influe pas sur le calcul en un pixel voisin  $q$ .

Vu que la médiane, en tant que mesure statistique, représente une alternative à la moyenne, le filtre médian constituera une forme de lissage qu'on peut utiliser à la place du lissage linéaire.

### 3.3.1.1. Propriétés

Le filtre médian a quelques propriétés qui conditionnent son utilisation.

- Non-création de niveaux de gris

L'opération de médiane choisit un de ses arguments, en fonction de la relation d'ordre entre eux.

Elle se distingue ainsi de celle de la moyenne, ou de toute combinaison linéaire, qui calcule une nouvelle valeur, ce qui peut nécessiter des arrondis. Aussi quand on applique le filtre médian à une image  $I$ , la valeur de niveau de gris  $J(p)$  obtenue en un pixel  $p$  sera une des valeurs de niveaux de gris  $I(q)$  d'un pixel  $q$  (dans l'image de départ) dans la fenêtre  $W(p)$  autour de  $p$ . Par conséquent, l'ensemble des niveaux de gris  $J(p)$  de la nouvelle image  $J$  sera inclus dans celui des niveaux de gris  $I(p)$  de l'image de départ  $I$ .

Il s'ensuit que dans une image ayant peu de niveaux de gris distincts, l'application du filtre médian n'augmentera pas le nombre de niveaux de gris, elle peut par contre le diminuer. Ainsi le filtre médian est absolument incapable de créer des dégradations. Un avantage de la non-crédation de niveaux de gris est qu'il ne faudra pas traiter les problèmes d'arrondi ou de précision des calculs. À partir de niveaux de gris entiers, on obtiendra des niveaux de gris entiers. Enfin, le filtre médian peut s'appliquer aux images binaires (à valeurs 0 et 1), et le résultat restera une image binaire.

Sur ce point, le filtre médian est en total contraste avec les filtres linéaires, qui créent de nouvelles valeurs de niveaux de gris, posent des problèmes d'arrondi et dans certains cas d'épanchement de valeurs, il ne s'appliquent pas aux images binaires, mais peuvent par contre être utilisés pour créer des dégradations et atténuer le phénomène de faux contours.

➤ Invariance par étirement de contraste

Soit  $f$  une transformation croissante des niveaux de gris appliquée à tous les pixels de l'image (par exemple,  $f$  peut être un [rehaussement des niveaux de gris de l'image](#)). Alors  $f$  peut commuter avec le filtre médian, ce qui signifie qu'on peut :

- soit appliquer d'abord le filtre médian, ensuite la transformation  $f$  des niveaux de gris,
- soit appliquer d'abord la transformation  $f$  des niveaux de gris, ensuite le filtre médian,

le résultat sera le même. C'est ce qu'indique le diagramme de *figure 3.16* ci-contre (ici  $f$  transforme le blanc en gris clair, et le gris sombre en noir).



Cela provient du fait que le choix de la médiane d'un ensemble de nombres est déterminé uniquement par la relation d'ordre entre ces nombres. Donc si  $J(p)$  est la médiane des  $I(q)$  pour  $q$  dans  $W(p)$ , alors  $f(J(p))$  sera la médiane des  $f(I(q))$  pour  $q$  dans  $W(p)$ , parce que la relation d'ordre entre les  $I(q)$  sera préservée pour les  $f(I(q))$ .

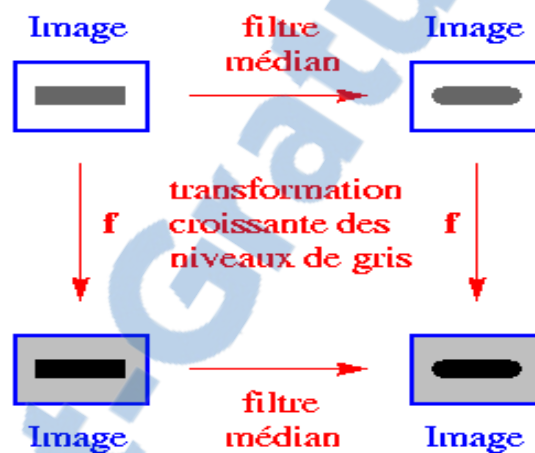


Figure 3.16 : Invariance par étirement de contraste

➤ Auto-dualité par inversion

Le filtre médian est [auto-dual](#), c'est-à-dire, son propre dual par inversion, en d'autres termes il commute avec l'opération de négatif dans une image. C'est ce qu'illustre le diagramme de la *figure 3.17*.

En effet, si on a  $n$  valeurs numériques  $x_1, \dots, x_n$  qui s'ordonnent de la plus petite à la plus grande  $x_{i1}, \dots, x_{in}$ , ce qui donne la médiane  $x_{i(n+1)/2}$ , alors comme le négatif inverse l'ordre, leurs négatifs  $N(x_1), \dots, N(x_n)$  s'ordonneront  $N(x_{in}), \dots, N(x_{i1})$ , ce qui donnera la médiane  $N(x_{i(n+1)/2})$  (si on inverse la suite ordonnée, le milieu ne change pas de position). Donc si  $J(p)$  est la médiane des  $I(q)$  pour  $q$  dans  $W(p)$ , alors  $N(J(p))$  sera la médiane des  $N(I(q))$  pour  $q$  dans  $W(p)$ .

L'auto-dualité du filtre médian signifie qu'il se comportera de la même façon sur la figure et le

fond (pour les images binaires), ou sur les zones claires et sombres (pour les images à niveaux de gris).

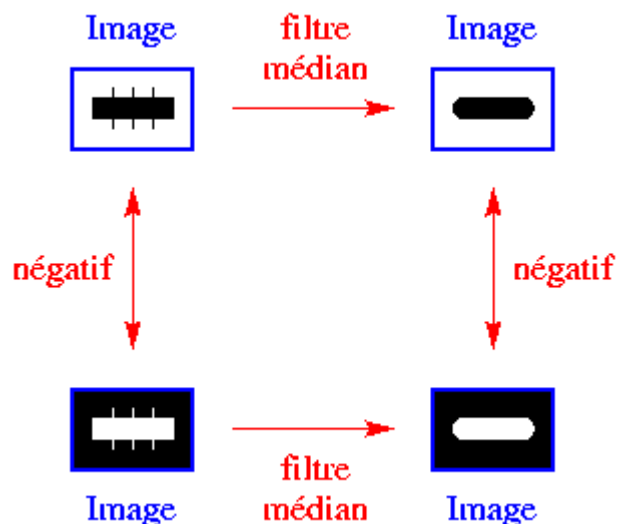


Figure 3.17 : Auto-dualité par inversion

➤ Préservation des marches et rampes rectilignes

Le filtre médian est réputé ne pas introduire de flou dans les images. Ce n'est pas totalement vrai, nous verrons qu'il peut étendre un dégradé présent le long d'une arête courbe, donnant donc un effet visuel de léger flou.

Cette qualité provient d'une propriété du filtre médian : la préservation des marches et rampes rectilignes. Une marche représente une augmentation abrupte du niveau de gris, tandis qu'une rampe signifie une augmentation progressive et monotone du niveau de gris. Une marche ou rampe est rectiligne si l'arête qu'elle constitue dans l'image est rectiligne, en d'autres termes si le niveau de gris est à peu près constant dans la direction perpendiculaire à celle de l'augmentation. Les marches et rampes sont des modèles typiques d'arêtes dans les images; un autre modèle est donné par la ligne, nous verrons plus loin comment le filtre médian se comporte sur celle-ci.

Nous illustrons ci-dessous le profil de niveaux de gris d'une marche et d'une rampe (dans le sens de la montée), et ensuite une marche et une rampe rectilignes où le niveau de gris augmente de bas en haut.

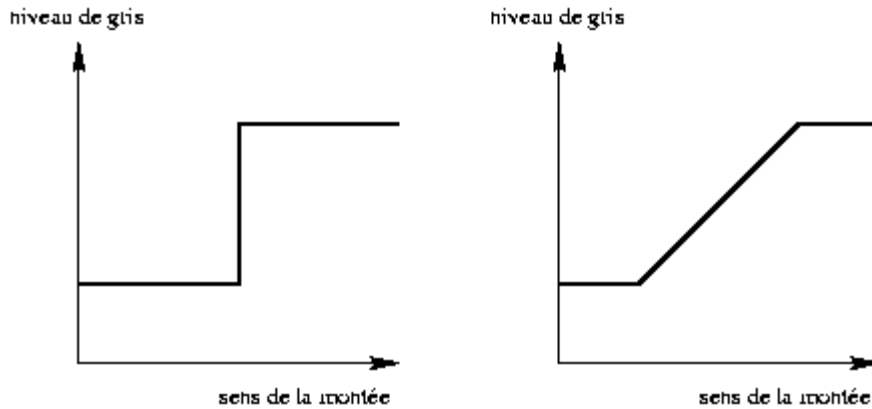


Figure 3.18 : Profils de niveaux de gris d'une marche (à gauche) et d'une rampe (à droite).



Figure 3.19 : Marche rectiligne



Figure 3.20 : Rampe rectiligne.

Cette propriété (de préservation des marches et rampes rectilignes) présuppose que la fenêtre  $W(p)$  correspondant au pixel  $p$  est toujours *symétrique* par rapport à  $p$ . Pour simplifier, considérons d'abord le cas unidimensionnel. Une marche ou une rampe donne un profil de niveaux de gris monotone croissant dans le sens de l'augmentation. Donc pour un pixel  $p$  quelconque, les niveaux de gris dans la fenêtre  $W(p)$  seront croissants, et la médiane de leurs valeurs sera celle du milieu, donc le niveau de gris du pixel au milieu de cette fenêtre  $W(p)$  ; mais nous avons supposé celle-ci est symétrique par rapport à  $p$ , donc le pixel au milieu de  $W(p)$  est  $p$ , et la médiane des niveaux de gris dans  $W(p)$  est celui de  $p$ . Par conséquent, le niveau de gris d'un pixel  $p$  quelconque ne sera pas modifié par le filtre médian. Il s'ensuit qu'un profil de niveaux de gris croissant est invariant par le filtre médian à fenêtres symétriques, et c'est le cas pour les marches et rampes.

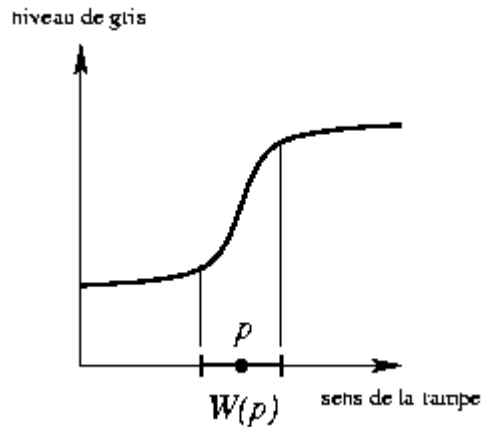


Figure 3.21 : comportement sur la rampe

Plaçons-nous maintenant dans le cas bidimensionnel. Nous supposons les niveaux de gris monotones croissants dans un sens, et constants dans la direction perpendiculaire, comme illustré ci-dessous. On peut donner un ordre de balayage de la fenêtre  $W(p)$  autour d'un pixel  $p$  : on décompose la fenêtre en segments dans la direction selon laquelle le niveau de gris est constant (donc perpendiculaire au sens de croissance), on balaye les lignes l'une après l'autre dans le sens de croissance des niveaux de gris, et chaque ligne dans un sens donné. Dans cet ordre de balayage, les niveaux de gris des pixels croissent dans la fenêtre  $W(p)$ , et la médiane de leurs valeurs sera celle du milieu, et comme la fenêtre est symétrique, cette valeur au milieu est le niveau de gris de  $p$ .

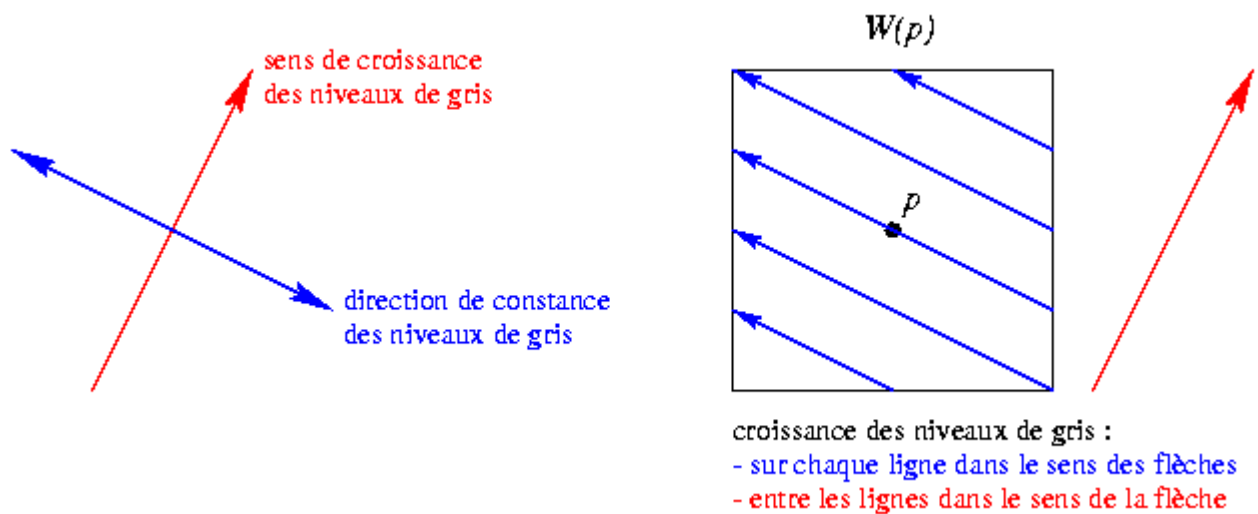


Figure 3.22 : considérations des deux sens, de niveau croissante et de niveau constante

Une *ligne* est un profil de niveau de gris qui correspond à deux marches ou rampes successives, donc dans une direction donnée, le niveau de gris présente soit une augmentation rapide suivie d'une diminution rapide (ligne claire), soit une diminution rapide suivie d'une augmentation rapide (ligne sombre). Le comportement du filtre médian sur une ligne est de niveler cette ligne de façon à former un plateau entre l'augmentation et la diminution de niveaux de gris, de largeur  $(n+1)/2$ , où  $n$  est la largeur de la fenêtre autour d'un pixel. Si la ligne est large de moins de  $(n+1)/2$  à sa base, elle sera entièrement effacée.

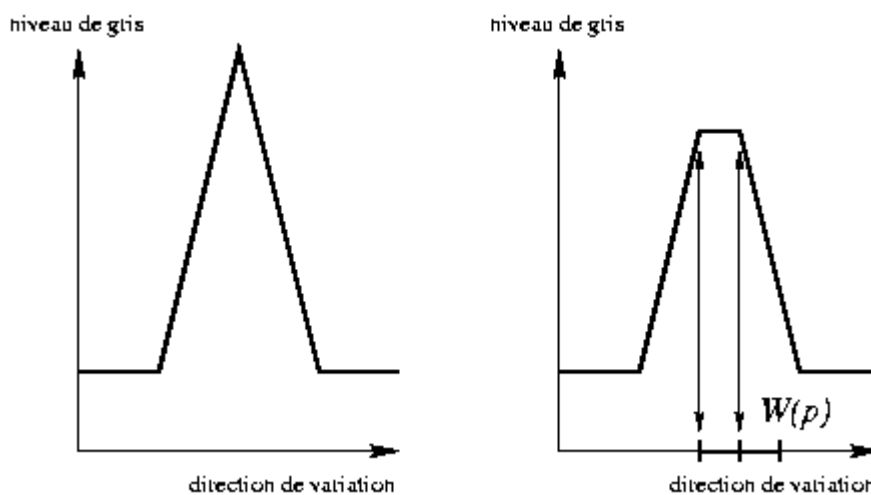


Figure 3.23 : À gauche : profil de niveau de gris d'une ligne claire. À droite : effet du filtre médian.

Du "point de vue" du filtre médian, une ligne filtrée dont le sommet forme un tel plateau, large de plus de la moitié de la taille de la fenêtre, représente une succession de deux marches ou rampes ; la fenêtre  $W(p)$  centrée autour d'un pixel  $p$  ne pourra jamais contenir à la fois une portion de la montée et une portion de la descente, elle ne peut donc "voir" que la montée ou la descente.

➤ Erosion des convexités

Les seules arêtes préservées par le filtre médian sont les marches ou rampes rectilignes. Quand une telle arête est courbée, le filtre médian érodera la zone convexe bordée par l'arête. C'est ce que nous illustrons ci-dessous. Le filtre médian érode uniformément un disque, ce qui revient à diminuer légèrement le diamètre de celui-ci. Par ailleurs, il arrondit les coins d'un rectangle (qui sont les points de forte courbure du bord de celui-ci), ce qui revient à réduire la courbure de l'arête aux coins. Quand le coin du rectangle présente un dégradé, celui-ci verra sa surface agrandie, ce qui peut donner une impression de léger flou. Enfin, sur un damier légèrement flou, le dégradé de gris du coin est fortement agrandi, ce qui donne un effet visuel d'accroissement du flou.

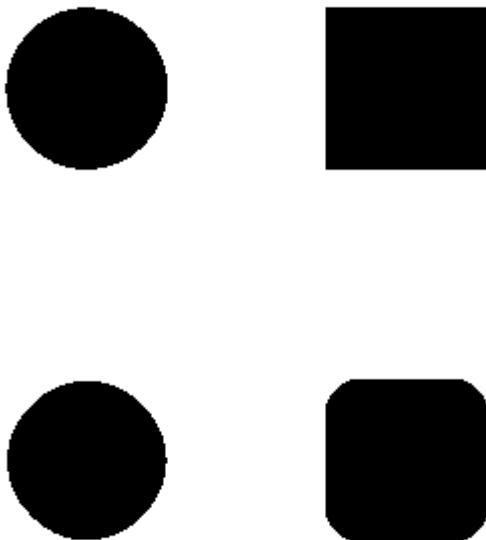


Figure 3.24 : comportement sur les zones convexes et les coins

➤ Elimination du bruit poivre et sel

Le *bruit impulsif*, aussi appelé *bruit poivre et sel*, est une dégradation de l'image où certains

pixels deviennent aléatoirement soit blancs, soit noirs. On suppose généralement que la probabilité qu'un pixel devienne blanc (respectivement noir) est constante sur l'image, et que le devenir de chaque pixel est indépendant de celui des autres.

Contrairement au lissage linéaire, le filtre médian est bien adapté au filtrage du bruit impulsif. Il s'appliquera aussi pour éliminer des griffes dans une image. C'est ce que nous illustrons dans les exemples ci-dessous.

On remarquera quand même que le filtre médian de fenêtre 5\*5 peut éliminer certains détails qui seront préservés par celui de fenêtre 3\*3, mais que ce dernier laisse subsister une partie du bruit, qui nécessitera pour être enlevé, une 2ème, voire une 3ème application du filtre.

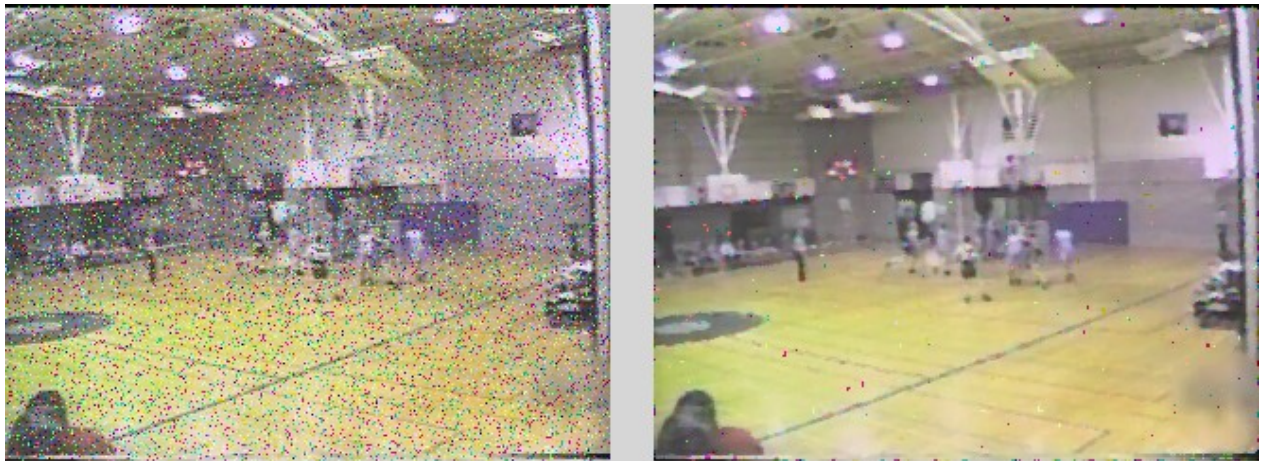


Figure 3.25 : a) image bruitée

b) filtrage par le filtre median 3\*3



Figure 3.26 : a) image bruitée

b) filtrage par 2\* le filtre median 3\*3





Figure 3.27 : a) image bruitée

b) filtrage par le filtre median 5\*5

La bonne performance du filtre médian dans ces situations découle de la propriété de la médiane d'être, contrairement à la moyenne, quasiment insensible aux valeurs extrêmes. Mais cela suppose que la probabilité qu'un pixel soit bruité "poivre" (très sombre), et celle qu'il soit bruité "sel" (très clair), sont toutes deux inférieures à 1/2. Ainsi, on pourra supposer que pour la plupart des pixels  $p$ , la fenêtre  $W(p)$  ne contient qu'une minorité de pixels "poivre", et de même pour "sel".

### 3.3.2. Filtre à moyenne seuillée [4] [5] [6]

L'idée générale consiste à réaliser un filtrage uniquement en dehors des zones de transitions. Celles-ci peuvent être repérées grâce à leur variance élevée.

#### 3.3.2.1. Principe

Autour de chaque pixel, on calcule la variance des niveaux de gris.

- Si elle est inférieure à un seuil, c'est-à-dire que l'on est dans une zone homogène, donc, on calcule la moyenne des niveaux de gris.
- Sinon, on garde le niveau de gris d'origine

#### **Inconvénient :**

Le seul inconvénient est que, plus on augmente le seuil, plus on peut introduire de flou, car en augmentant le seuil, on ignore les détails.



### 3.3.3. Filtre de Nagao ou filtre à sélection de voisinage [7] [8] [9]

Ce filtre est également appelé filtre à sélection de voisinage. Chaque fenêtre de 5\*5 sera divisée en les 9 domaines ( $D_0, D_1, \dots, D_9$ )

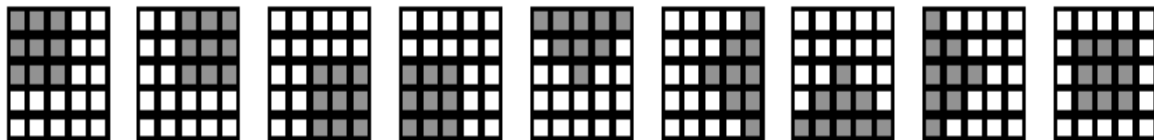


Figure 3.28 Les neufs domaines de considération selon toutes les directions pour un meilleur effet contre l'effet flou après filtrage

Pour chaque domaine, on calcule la moyenne et la variance : Puis on va ensuite chercher le domaine où la variance est la plus faible, on peut supposer alors qu'aucun contour ne passe par ce domaine, on affecte alors la moyenne ou la valeur médiane des niveaux de gris de ce domaine au pixel central.

Ce filtre permet aussi de limiter les pixels hors normes (ou bruit impulsif) mais les avantages majeurs par rapport aux autres c'est qu'il préserve et même améliore le contraste, les zones avant et après les transitions sont plus homogènes, on a une conservation de la dynamique des luminances le long des contours et en fin, on respecte mieux les angles des formes.

Dans notre application, pour avoir un peu de gain en temps de calcul, on remplace la variance par un autre paramètre de dispersion qui est l'étendue  $\max(\text{gris}) - \min(\text{gris})$ , et on remplace la moyenne par le médian.



Figure 3.29: a) image bruitée



b) image filtrée par le filtre Nagao

## CHAPITRE 4: APPLICATIONS

### 4.1 Introduction

Ce chapitre présente quelque simulation sous MATLAB version 6.5 des traitements à effectuer sur des images appartenant à un fichier vidéo numérique (.AVI). Pour cela, on va simuler quelques restaurations possibles d'une image appartenant au fichier .AVI préalablement bruitée par des bruits tel que les sels- poivres, puis, on va essayer des améliorations sur la même image mais non-bruité en jouant sur les différents techniques de rehaussement de contraste du chapitre 1 à savoir l'étirement d'histogramme, l'égalisation, et la méthode de Beghdadi.

Ce chapitre se divise en deux parties : la première partie concerne une présentation générale du logiciel MATLAB, la deuxième et dernière partie nous illustre une brève présentation de la simulation effectuée sur les images.

### 4.2 Généralités sur MATLAB

MATLAB ou *Matrix laboratory* est un langage de calcul scientifique très performant intégrant le calcul, la programmation et la visualisation dans un environnement simple à utiliser. C'est un système interactif qui permet de manipuler directement des données structurées (matrices et vecteurs), ce qui n'est pas le cas de C, Java, Fortran ou Pascal par exemple. C'est d'ailleurs la raison du choix de ce logiciel.

#### 4.2.1. Environnement

Matlab est à la fois un langage et un logiciel. Il possède une fenêtre principale ou de commande et un éditeur de programmes (M-files). Dans l'éditeur de programme, plusieurs fenêtres (donc programmes) peuvent être ouvertes à la fois. Un éditeur de figure est aussi disponible pour les graphiques.

#### 4.2.2. Principe du help/demo

Le help de Matlab est très bien adapté à l'auto-apprentissage autant théorique que pratique sur le langage, les fonctions disponibles et les techniques qu'il comporte le rend très souple (notamment grâce à ses toolboxes).

Taper la commande **help** permet d'afficher tout le contenu de matlab, c'est-à-dire les familles de fonctions qu'il comporte. Taper **help famille** affiche toutes les fonctions qui appartiennent à la

famille en question et **help fonction** donne la définition de la fonction, ses options et sa syntaxe. (figure 4.01).

Une documentation pdf plus élaborée est aussi disponible et accessible par la commande *Help Desk*. Cette documentation comporte des supports théoriques pour la plupart des techniques offertes par Matlab notamment via les toolboxes. On pourrait y trouver par exemple des cours théoriques et pratiques sur le traitement d'image, la logique floue, les réseaux de neurones, les ondelettes, ect...

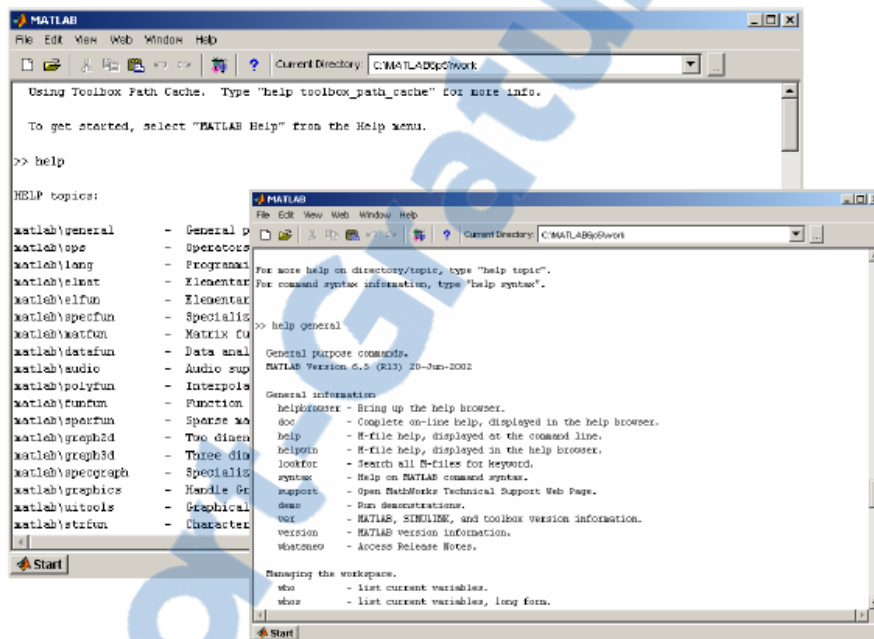


Figure 4.01 : les aides disponibles sur MATLAB

Le **demo**, apprendre tout: taper **demo** (démonstration) affiche un environnement interactif qui permet de naviguer dans les différentes ressources de matlab et de lancer des simulations de programmes où les codes (instructions) et les actions résultantes sont affichés en même temps.

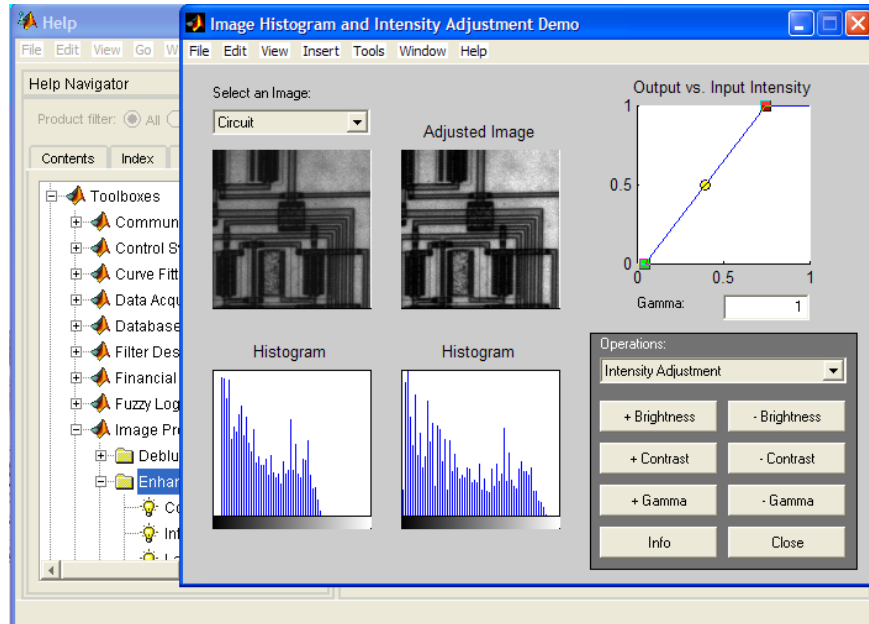


Figure 4.02 : démonstration de MATLAB sur le traitement numérique d'image

#### 4.2.4. Programmation

Toutes les instructions permises par Matlab peuvent être lancées à partir de la fenêtre de commande, il suffit d'écrire ces instructions en respectant leur syntaxe et de taper la touche *enter* pour valider. Si nous avons un nombre très réduit d'instructions à exécuter, il est en effet possible

de les écrire directement dans cette fenêtre de commande. Toutefois, le plus souvent nous avons besoin d'écrire des programmes assez longs et surtout de sauvegarder ces programmes. L'éditeur de programmes ou de fichiers d'extension **.m** est fait pour cela. Pour l'activer, aller dans le menu **file** et cliquer sur **open** pour ouvrir un fichier existant ou bien sur **new** puis **M-file** si on veut créer un nouveau fichier **.m**.

Pour lancer (exécuter) un programme, on peut soit à partir de la fenêtre de programme faire **debug** suivi de **run** ou bien écrire carrément le nom du programme dans la fenêtre de commande suivi de *enter*.

### 4.3 Simulation

Cette simulation qui, en effet se divise en deux parties traite en premier temps quelques images appartenant à la vidéo, en minimisant les bruits préalablement introduits. Nous utiliserons alors les principales techniques présentées dans du chapitre 3, à savoir les filtres linéaires, et les filtres non-linéaires tels que le filtre médian, le filtre à moyenne seuillée et le filtre de Nagao. En

suite, en deuxième partie, nous allons réaliser des améliorations de contraste à partir de l'étirement d'histogramme, puis l'égalisation d'histogramme, et enfin le rehaussement par la méthode de Beghdadi.

#### 4.3.1. Présentation du logiciel de simulation

Le logiciel de simulation comportera alors deux principaux champs de commande : la première est le filtrage, la deuxième est le contraste

##### 4.3.1.1. Lancement du logiciel

Après avoir lancé MATLAB, taper *accueil* dans la fenêtre de commande puis valider pour lancer le logiciel. La fenêtre d'accueil est représentée par la *Figure 4.03*. Cette fenêtre ne comporte qu'un seul bouton de lancement LANCER ; ce bouton nous permet alors d'accéder à l'interface utilisateur de notre simulation *Figure 4.04*.



Figure 4.03 : fenêtre d'accueil pour la simulation

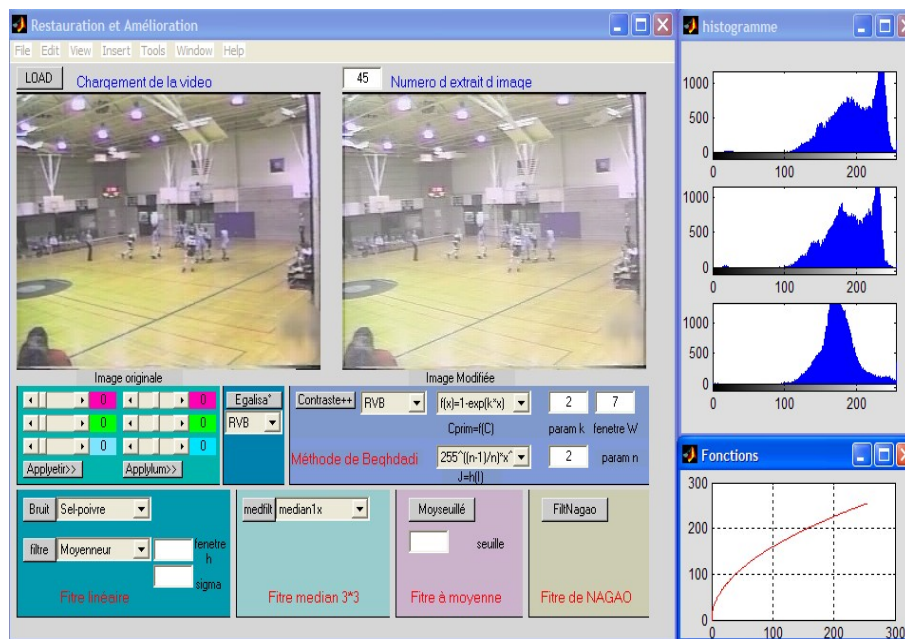


Figure 4.04 : interface utilisateur

#### 4.3.1.2. Filtrage

Description :

Dans cette partie de commande, on a quatre possibilités de filtrage :

- Le filtrage linéaire, comportant les filtres moyenneur et gaussien. Avant de filtrer, il faut choisir le filtre à utilisé, puis paramétrer le filtre en fonction de notre besoin, par exemple, pour le filtre gaussien, il faut définir la taille du filtre et la valeur de  $\sigma$

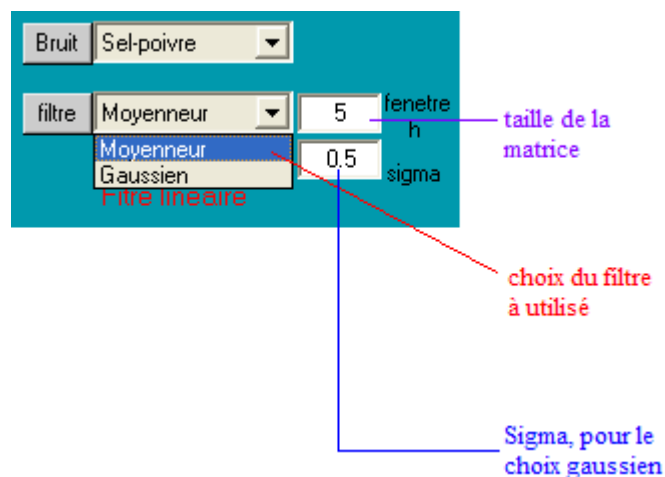


Figure 4.05 : commande sur les filtre linéaire

- Filtrage médian, ne comportant qu'un seul bouton, avec trois choix qui définie le

nombre de filtre à appliqué.

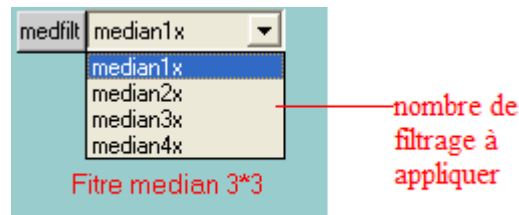


Figure 4.06 : commande sur le filtre médian

- Filtrage à moyenne seuillée, ne comportant qu'un seul bouton et un seul paramètre qui es le seuil de variance. Ici, la fenêtre à utiliser est de 3\*3 pour être plus efficace.



Figure 4.07 : commande sur le filtre à moyenne seuillée

- Filtre de Nagao, présenter par un seul bouton et sans paramètre.



Figure 4.08 : commande sur le filtre de Nagao

#### 4.3.1.3. Contraste

##### Description

Dans cette deuxième partie de commande, on a trois possibilités de rehaussement du contraste :

- L'étirement d'histogramme, comportant trois curseurs correspondant respectivement à l'étirement des trois couleurs RVB, les positions des curseurs permettent de régler les niveaux d'étirement. Trois autres curseurs permettent de régler la luminosité.

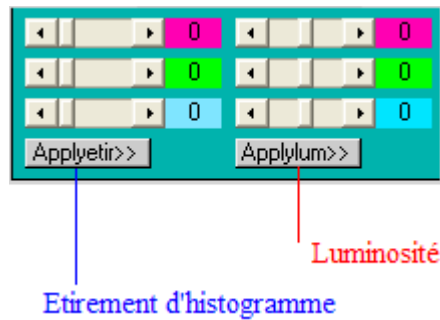


Figure 4.09 : commande sur les rehaussements classiques

- L'égalisation d'histogramme, ne comportant qu'un seul bouton, mais, on peut choisir tout les combinaisons d'égalisation des trois plans de couleurs possibles.

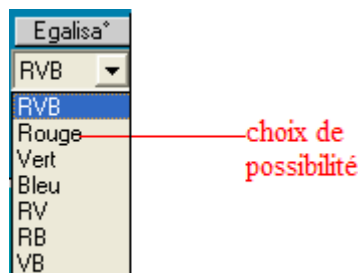


Figure 4.10 : Commande sur l'égalisation d'histogramme

- Méthode de Beghdadi, composé d'un premier bouton réalisant le rehaussement suivant la combinaison choisie, et de deux paramètres qui définissent la fenêtre de considération des voisins et le paramètre sur la fonction qui réalise le rehaussement de contraste . Le deuxième bouton réalise une correction d'exposition suivant le choix de fonction à utilisé en prenant un paramètre pour régler cette même fonction.



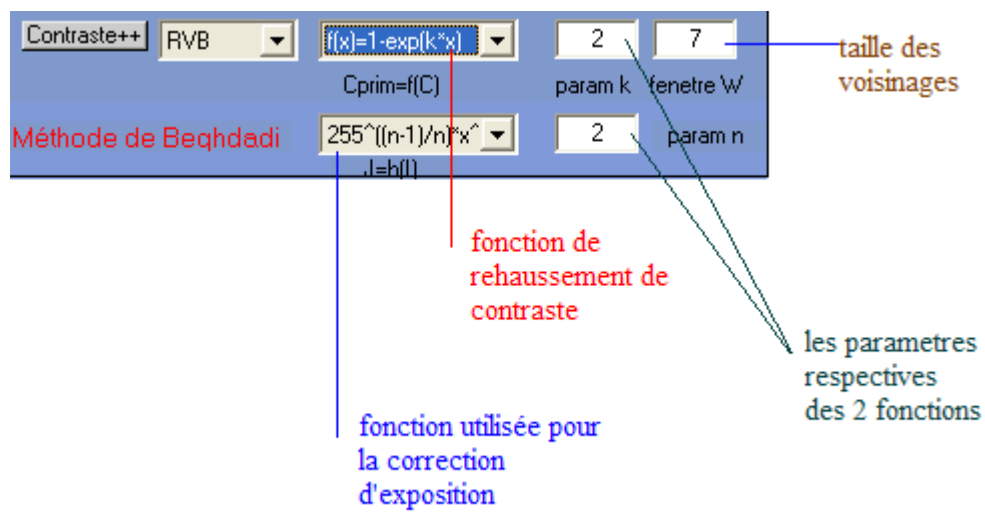


Figure 4.11 : commande sur le rehaussement utilisant la méthode de Beqhdadi

Enfin, on peut voir l'histogramme et les différentes fonctions utilisées par les traitements à droite de l'écran Figure 4.12.

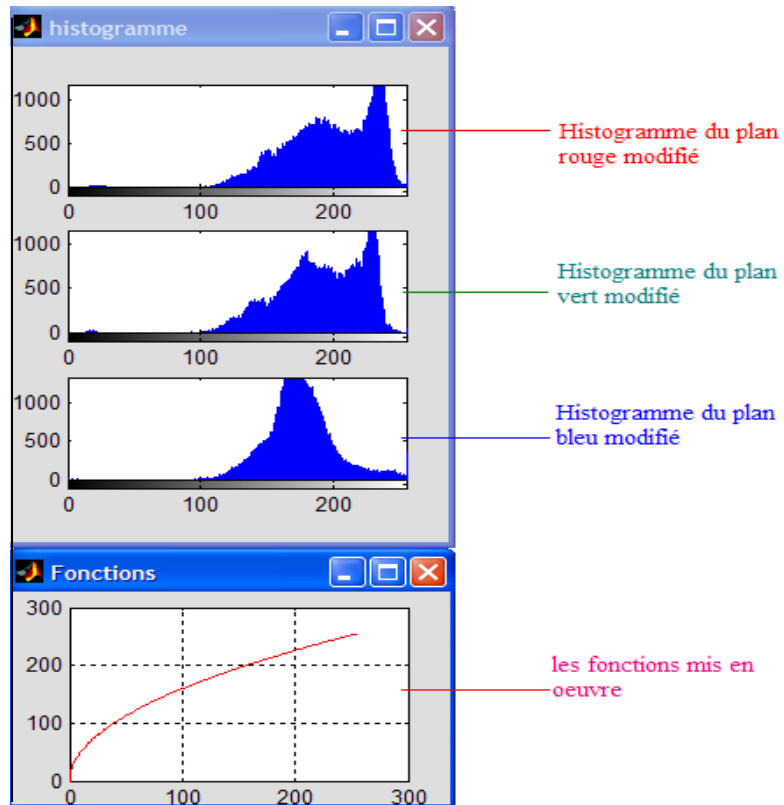


Figure 4.12 : visualisation de l'histogramme et des fonction nécessaire au traitement

Les graphes situés à droite nous permettent de suivre et de comprendre l'évolution de la fréquence d'apparition des niveaux de gris de chaque plan RVB, ainsi cela nous permet de nous aider dans le choix du traitement concernant surtout sur les rehaussements de contraste.

## CONCLUSION

La demande fulgurante sur les exploitations de l'information nécessite toujours des recherches sans limite, en effet, le traitement d'image est utilisé pour des différentes applications et peut dans plusieurs des cas être un élément clé. Par exemple, dans le domaine militaire, un meilleur traitement de l'information, plus particulièrement l'image est primordial pour faciliter les différentes approches stratégiques, en effet il a déjà démontré son utilité dans plusieurs conflits, notamment pendant la guerre froide. Dans un domaine similaire, par exemple, pour les enquêtes policières, la restauration et amélioration peuvent être l'élément de basculement. Enfin, dans d'autre domaine très différent tel que la médecine, le traitement d'image a apporté beaucoup d'évolution, notamment dans le domaine chirurgicale, il est désormais possible d'introduire des appareils munis de petit camera afin de permettre la navigation dans le corps du patient. Le chirurgien n'a plus qu'à guider l'appareil en fonction de la vidéo capté par la caméra.

En effet, plusieurs logiciels de traitement numérique d'image sont déjà sur le marché, avec des mis à jour qui dans certain cas ne dépasse pas un mois. Le problème qui se pose est que ces logiciels coûtent trop chers, c'est-à-dire que les sociétés tel que les stations de télévision ne peuvent pas suivre tout le temps les évolutions trop rapide de ces évolutions. À cet effet, nous avons montré que l'on peut améliorer les traitements déjà existants en connaissant les théories de base de la restauration et amélioration. Ces traitements sont en effet très nécessaires dans le domaine de la télévision avant les divers montages vidéo. La limite de notre simulation est en effet le problème de temps réel, en effet, on ne peut pas traiter instantanément la vidéo.

Enfin, un sujet de recherche intéressant serait dans l'avenir, de minimiser le temps de calcul sur ces techniques. On pourra alors dans ce cas intégrer ces traitements dans les appareils de capture d'image. Les images vidéos sortants n'auront plus à être restauré avant utilisation.

## BIBLIOGRAPHIE

- [1] M BOTO ANDRIANANDRASANA Jean Espérant, *Cours de Télévision 4<sup>ème</sup> année*, ESPA, département Télécommunication.
  
- [2] M RAKOTOMALALA Mamy Alain, *Cours de traitement numérique d'image 4<sup>ème</sup> année*, ESPA, département Télécommunication.
  
- [3] V. Lozano, *Contribution de l'analyse d'image couleur au traitement des images textile* thèse de doctorat, Université Jean-Monnet, Saint-Étienne, janvier 1998.
  
- [4] Coster et Chermant , *Précis d'analyse d'image*, Edition du CNRS 1985
  
- [5] La Recherche 289, *l'Imagerie Médicale*, Juillet-Août 1996
  
- [6] Catherine Achard, *Cours de traitement d'images*
  
- [7] Groupe Adobe Dynamic Media, *Initiation à la vidéo numérique*
  
- [8] T. Boudier , N. Kamech , *Imagerie Numérique*
  
- [9] T. Boudier, *Imagerie Numérique en Biologie*
  
- [10] Louis VOLONT, *cours de « Image TV & Vidéo »*
  
- [11] Gérard-Michel Cochard , *Image numérique*
  
- [12] *Encarta 2004*

[13] <http://www.commentcamarche.net/encyclopedie>

[14] <http://www.linternaute.com/>

[15] [http://www.arthuraulin.com/tuto\\_photoshop\\_aulin\\_maj\\_2005-06\\_05.pdf](http://www.arthuraulin.com/tuto_photoshop_aulin_maj_2005-06_05.pdf)

Rapport-Gratuit.com

## ANNEXE

### Les principaux codes sources des simulations

- **Chargement du fichier vidéo .AVI**

```
elseif strcmp(action,'loadimage')
H=get(gcf,'Userdata');
h = waitbar(0,'LOADING.....');for i=1:1000,waitbar(i/100,h),end,close(h)
M = aviread('video', 1:141);
numframes = length(M)
for i = 1:numframes
M(i) = im2frame(M(i).cdata);
%image(f(i).cdata)
end
axes(H(7))

movie(M,1,24)
```

- **Etirement d'histogramme**

```
f=get(H(12),'string');
n=str2num(f)
axes(H(7))
movie(M(n))
x1=frame2im(M(n));

y1=double(x1);
g1=get(H(4),'value');
y2(:,1)=(255/(255-(2*g1)))*(y1(:,1)-g1);
g2=get(H(8),'value');
y2(:,2)=(255/(255-(2*g2)))*(y1(:,2)-g2);
g3=get(H(10),'value');
y2(:,3)=(255/(255-(2*g3)))*(y1(:,3)-g3);

h = waitbar(0,'LOADING.....');for i=1:1000,waitbar(i/100,h),end
y1=uint8(y1);
y2=uint8(y2);
figure(2)
subplot(3,1,1)
```

```

imhist(y2(:,:,1))
subplot(3,1,2)
imhist(y2(:,:,2))

subplot(3,1,3)
imhist(y2(:,:,3))
f1(1)=im2frame(y2);

axes(H(5))
movie(f1)
t=[0:0.0001:1];
y=1-exp(-v*t);
figure(3)
subplot(1,1,1)
plot(t,y,'r')
close(h)

```

- **Luminosité**

```

f1=get(H(14),'string');
f2=get(H(16),'string');
f3=get(H(18),'string');
n1=str2num(f1)
n2=str2num(f2)
n3=str2num(f3)
y2(:,:,1)=y2(:,:,1)+n1;
y2(:,:,2)=y2(:,:,2)+n2;
y2(:,:,3)=y2(:,:,3)+n3;
y1=uint8(y1);
y2=uint8(y2);
figure(2)
subplot(3,1,1)
imhist(y2(:,:,1))

```

```

subplot(3,1,2)
imhist(y2(:,:,2))

```

```

subplot(3,1,3)
imhist(y2(:,:,3))
f4(1)=im2frame(y2);

```

```

axes(H(5))
movie(f4)

```

- **Simulation de bruit**

```

switch f1
case 'Sel-poivre'
    N=imnoise(N,'salt & pepper',0.3);

case 'Bruit Gaussien'
    N=imnoise(N,'gaussian');
case 'Poisson'
    N=imnoise(N,'poisson') ;
case 'J=I+n*I'
    N=imnoise(N,'speckle');%J = I + n*I

otherwise
    error('Erreur!0');
end

M(n) = im2frame(N);
axes(H(7))
movie(M(n))

```

```
M(n) = im2frame(N);
```

```
f2 = get(H(29), {'value','String'});
f2 = deblank(f2 {2} (f2 {1},:));
```

```

switch f2
case 'Moyenneur'
    f3=get(H(27),'string');
    n3=str2num(f3)

    if n3==3
        H1=(1/9)*[1 1 1;1 1 1;1 1 1];
    elseif n3==5
        H1=(1/21)*[0 1 1 1 0;1 1 1 1 1;1 1 1 1 1;1 1 1 1 1;0 1 1 1 0];
    elseif n3==7
        H1=(1/29)*[0 0 0 1 0 0 0;0 1 1 1 1 1 0;0 1 1 1 1 1 0; ...
        1 1 1 1 1 1 1;0 1 1 1 1 1 0;0 1 1 1 1 1 0; ...
        0 0 0 1 0 0 0];
    end

case 'Gaussien'

    f3=get(H(27),'string');
    n3=str2num(f3);
    HSIZE=[n3 n3]

```



```

f4=get(H(28),'string');
n4=str2num(f4);
SIGMA=n4;
H1 = FSPECIAL('gaussian',HSIZE,SIGMA)
figure(3)
subplot(1,1,1)
surf(H1)

otherwise
    error('Erreur!01');
end

y2=imfilter(N,H1);

y2=uint8(y2);

v(1)=im2frame(y2,[]);

axes(H(5))
movie(v)

figure(2)
subplot(3,1,1)
imhist(y2(:,1))

subplot(3,1,2)
imhist(y2(:,2))

subplot(3,1,3)
imhist(y2(:,3))

```

- **Contraste par Beghdadi**

```

M=240;
N=320;
f=get(H(19),'string');
m=str2num(f)
ff=get(H(20),'string');
v=str2num(ff)

o=1;
p=1;
%I(:,1);
%I=double(I);
g2 = get(H(22),{'value','String'});

```

```

g2 = deblank(g2{2}(g2{1},:));
g1 = get(H(21),{'value','String'});
g1 = deblank(g1{2}(g1{1},:));
%g1=get(H(21),'string') diso be
d1 = get(H(24),{'value','String'});
d1 = deblank(d1{2}(d1{1},:));
switch d1
case 'Sans correction'
switch g1
case 'RVB'
for i=(m-1)/2+1 : N-(m-1)/2
for j=(m-1)/2+1 : M-(m-1)/2
o=1;
p=1;

%MANOMBOKA
if m==5

for l=i-1 : i+1
%l=i-(m-1)/2 +2 : i+(m-1)/2-2
for k=j-1 : j+1%k=j-(m-1)/2 +2 : j+(m-1)/2-2
Xmoy=1/9*(y2(k-1,l-1,1)+y2(k-1,l,1)+y2(k-1,l+1,1)+...
y2(k,l-1,1)+y2(k,l,1)+y2(k,l+1,1)+...
y2(k+1,l-1,1)+y2(k+1,l,1)+y2(k+1,l+1,1));

delta(k,l,1)=abs(y2(k,l,1)-Xmoy);
DELTA(o)=delta(k,l,1);
o=o+1;
Pe(p)=y2(k,l,1);
p=p+1;
end

end
elseif m==3
l=i;
k=j;
Xmoy=1/9*(y2(k-1,l-1,1)+y2(k-1,l,1)+y2(k-1,l+1,1)+...
y2(k,l-1,1)+y2(k,l,1)+y2(k,l+1,1)+...
y2(k+1,l-1,1)+y2(k+1,l,1)+y2(k+1,l+1,1));

delta(k,l,1)=abs(y2(k,l,1)-Xmoy);
DELTA(o)=delta(k,l,1);
o=o+1;
Pe(p)=y2(k,l,1);
p=p+1;
elseif m>5
for l=i-(m-1)/2 +2 : i+(m-1)/2-2
for k=j-(m-1)/2 +2 : j+(m-1)/2-2
Xmoy=1/9*(y2(k-1,l-1,1)+y2(k-1,l,1)+y2(k-1,l+1,1)+...
y2(k,l-1,1)+y2(k,l,1)+y2(k,l+1,1)+...

```

```

        y2(k+1,l-1,1)+y2(k+1,l,1)+y2(k+1,l+1,1));

        delta(k,l,1)=abs(y2(k,l,1)-Xmoy);
        DELTA(o)=delta(k,l,1);
        o=o+1;
        Pe(p)=y2(k,l,1);
        p=p+1;
        end

    end
end
    if m==3
        Sommedelta=delta(k,l,1)+0.00001;
    elseif m>3
        Sommedelta=sum(DELTA);
    end
    Num=DELTA.*Pe;
    Num=sum(Num);

    E(j,i,1)=Num/Sommedelta;

    C(j,i,1)=abs((y2(j,i,1)-E(j,i,1))/(y2(j,i,1)+E(j,i,1)));

    g2 = get(H(22),{'value','String'});
    g2 = deblank(g2{2}(g2{1},:));
    %g1=get(H(21),'string') diso be
    switch g2
    case 'f(x)=1-exp(k*x)'
        X(j,i,1)=-v*C(j,i,1);%resaka fotsiny
        Cc(j,i,1)=1-exp(X(j,i,1));%fonction mapiakatra
        %ny contraste
    case 'f(x)=ln(1+2x)'
        X(j,i,1)=v*C(j,i,1);
        Cc(j,i,1)=log(1+X(j,i,1));
    case 'f(x)=x^(1/k)'
        X(j,i,1)=C(j,i,1).^(1/v);
        Cc(j,i,1)=C(j,i,1).^(1/v);
    otherwise
        error('Erreur! 1');
    end

    if y2(j,i,1)<=E(j,i,1)
        y2(j,i,1)=((1-Cc(j,i,1))/(Cc(j,i,1)+1))*E(j,i,1);
    elseif y2(j,i,1)>E(j,i,1)
        y2(j,i,1)=((1+Cc(j,i,1))/(1-Cc(j,i,1)))*E(j,i,1);
    end
end
end
end

```

- Filtrage de Nagao

```

switch fl
case 'Sel-poivre'
    N=imnoise(N,'salt & pepper',0.3);

case 'Bruit Gaussien'
    N=imnoise(N,'gaussian');
case 'Poisson'
    N=imnoise(N,'poisson') ;
case 'J=I+n*I'
    N=imnoise(N,'speckle');%J = I + n*I

otherwise
    error('Erreur!0');
end
N=double(N);
for k=3 : 318
    for l=3 : 238

        X1=[((N(l-2,k-2,1)+N(l-2,k-2,2)+N(l-2,k-2,3))/3) ...
            ((N(l-1,k-2,1)+N(l-1,k-2,2)+N(l-1,k-2,3))/3) ...
            ((N(l,k-2,1)+N(l,k-2,2)+N(l,k-2,3))/3) ...
            ((N(l-2,k-1,1)+N(l-2,k-1,2)+N(l-2,k-1,3))/3) ...
            ((N(l-1,k-1,1)+N(l-1,k-1,2)+N(l-1,k-1,3))/3) ...
            ((N(l,k-1,1)+N(l,k-1,2)+N(l,k-1,3))/3) ...
            ((N(l-2,k,1)+N(l-2,k,2)+N(l-2,k,3))/3) ...
            ((N(l-1,k,1)+N(l-1,k,2)+N(l-1,k,3))/3) ...
            ((N(l,k,1)+N(l,k,2)+N(l,k,3))/3)];

        X2=[((N(l-2,k,1)+N(l-2,k,2)+N(l-2,k,3))/3) ...
            ((N(l-1,k,1)+N(l-1,k,2)+N(l-1,k,3))/3) ...
            ((N(l,k,1)+N(l,k,2)+N(l,k,3))/3) ...
            ((N(l-2,k+1,1)+N(l-2,k+1,2)+N(l-2,k+1,3))/3) ...
            ((N(l-1,k+1,1)+N(l-1,k+1,2)+N(l-1,k+1,3))/3) ...
            ((N(l,k+1,1)+N(l,k+1,2)+N(l,k+1,3))/3) ...
            ((N(l-2,k+2,1)+N(l-2,k+2,2)+N(l-2,k+2,3))/3) ...
            ((N(l-1,k+2,1)+N(l-1,k+2,2)+N(l-1,k+2,3))/3) ...
            ((N(l,k+2,1)+N(l,k+2,2)+N(l,k+2,3))/3)];

        X3=[((N(l,k,1)+N(l,k,2)+N(l,k,3))/3) ...
            ((N(l+1,k,1)+N(l+1,k,2)+N(l+1,k,3))/3) ...
            ((N(l+2,k,1)+N(l+2,k,2)+N(l+2,k,3))/3) ...
            ((N(l,k+1,1)+N(l,k+1,2)+N(l,k+1,3))/3) ...
            ((N(l+1,k+1,1)+N(l+1,k+1,2)+N(l+1,k+1,3))/3) ...
            ((N(l+2,k+1,1)+N(l+2,k+1,2)+N(l+2,k+1,3))/3) ...
            ((N(l,k+2,1)+N(l,k+2,2)+N(l,k+2,3))/3) ...
            ((N(l+1,k+2,1)+N(l+1,k+2,2)+N(l+1,k+2,3))/3) ...
            ((N(l+2,k+2,1)+N(l+2,k+2,2)+N(l+2,k+2,3))/3)];
    end
end

```

$$\begin{aligned}
X4 = & [((N(1,k-2,1)+N(1,k-2,2)+N(1,k-2,3))/3) \dots \\
& ((N(1+1,k-2,1)+N(1+1,k-2,2)+N(1+1,k-2,3))/3) \dots \\
& ((N(1+2,k-2,1)+N(1+2,k-2,2)+N(1+2,k-2,3))/3) \dots \\
& ((N(1,k-1,1)+N(1,k-1,2)+N(1,k-1,3))/3) \dots \\
& ((N(1+1,k-1,1)+N(1+1,k-1,2)+N(1+1,k-1,3))/3) \dots \\
& ((N(1+2,k-1,1)+N(1+2,k-1,2)+N(1+2,k-1,3))/3) \dots \\
& ((N(1,k,1)+N(1,k,2)+N(1,k,3))/3) \dots \\
& ((N(1+1,k,1)+N(1+1,k,2)+N(1+1,k,3))/3) \dots \\
& ((N(1+2,k,1)+N(1+2,k,2)+N(1+2,k,3))/3)];
\end{aligned}$$

$$\begin{aligned}
X5 = & [((N(1-2,k-2,1)+N(1-2,k-2,2)+N(1-2,k-2,3))/3) \dots \\
& ((N(1-2,k-1,1)+N(1-2,k-1,2)+N(1-2,k-1,3))/3) \dots \\
& ((N(1-2,k,1)+N(1-2,k,2)+N(1-2,k,3))/3) \dots \\
& ((N(1-2,k+1,1)+N(1-2,k+1,2)+N(1-2,k+1,3))/3) \dots \\
& ((N(1-2,k+2,1)+N(1-2,k+2,2)+N(1-2,k+2,3))/3) \dots \\
& ((N(1-1,k-1,1)+N(1-1,k-1,2)+N(1-1,k-1,3))/3) \dots \\
& ((N(1-1,k,1)+N(1-1,k,2)+N(1-1,k,3))/3) \dots \\
& ((N(1-1,k+1,1)+N(1-1,k+1,2)+N(1-1,k+1,3))/3) \dots \\
& ((N(1,k,1)+N(1,k,2)+N(1,k,3))/3)];
\end{aligned}$$

$$\begin{aligned}
X6 = & [((N(1-2,k+2,1)+N(1-2,k+2,2)+N(1-2,k+2,3))/3) \dots \\
& ((N(1-1,k+2,1)+N(1-1,k+2,2)+N(1-1,k+2,3))/3) \dots \\
& ((N(1,k+2,1)+N(1,k+2,2)+N(1,k+2,3))/3) \dots \\
& ((N(1+1,k+2,1)+N(1+1,k+2,2)+N(1+1,k+2,3))/3) \dots \\
& ((N(1+2,k+2,1)+N(1+2,k+2,2)+N(1+2,k+2,3))/3) \dots \\
& ((N(1-1,k+1,1)+N(1-1,k+1,2)+N(1-1,k+1,3))/3) \dots \\
& ((N(1,k+1,1)+N(1,k+1,2)+N(1,k+1,3))/3) \dots \\
& ((N(1+1,k+1,1)+N(1+1,k+1,2)+N(1+1,k+1,3))/3) \dots \\
& ((N(1,k,1)+N(1,k,2)+N(1,k,3))/3)];
\end{aligned}$$

$$\begin{aligned}
X7 = & [((N(1+2,k-2,1)+N(1+2,k-2,2)+N(1+2,k-2,3))/3) \dots \\
& ((N(1+2,k-1,1)+N(1+2,k-1,2)+N(1+2,k-1,3))/3) \dots \\
& ((N(1+2,k,1)+N(1+2,k,2)+N(1+2,k,3))/3) \dots \\
& ((N(1+2,k+1,1)+N(1+2,k+1,2)+N(1+2,k+1,3))/3) \dots \\
& ((N(1+2,k+2,1)+N(1+2,k+2,2)+N(1+2,k+2,3))/3) \dots \\
& ((N(1+1,k-1,1)+N(1+1,k-1,2)+N(1+1,k-1,3))/3) \dots \\
& ((N(1+1,k,1)+N(1+1,k,2)+N(1+1,k,3))/3) \dots \\
& ((N(1+1,k+1,1)+N(1+1,k+1,2)+N(1+1,k+1,3))/3) \dots \\
& ((N(1,k,1)+N(1,k,2)+N(1,k,3))/3)];
\end{aligned}$$

$$\begin{aligned}
X8 = & [((N(1-2,k-2,1)+N(1-2,k-2,2)+N(1-2,k-2,3))/3) \dots \\
& ((N(1-1,k-2,1)+N(1-1,k-2,2)+N(1-1,k-2,3))/3) \dots \\
& ((N(1,k-2,1)+N(1,k-2,2)+N(1,k-2,3))/3) \dots \\
& ((N(1+1,k-2,1)+N(1+1,k-2,2)+N(1+1,k-2,3))/3) \dots \\
& ((N(1+2,k-2,1)+N(1+2,k-2,2)+N(1+2,k-2,3))/3) \dots \\
& ((N(1-1,k-1,1)+N(1-1,k-1,2)+N(1-1,k-1,3))/3) \dots \\
& ((N(1,k-1,1)+N(1,k-1,2)+N(1,k-1,3))/3) \dots \\
& ((N(1+1,k-1,1)+N(1+1,k-1,2)+N(1+1,k-1,3))/3) \dots \\
& ((N(1,k,1)+N(1,k,2)+N(1,k,3))/3)];
\end{aligned}$$

$X9 = [((N(l-1,k-1,1)+N(l-1,k-1,2)+N(l-1,k-1,3))/3) \dots$   
 $((N(l,k-1,1)+N(l,k-1,2)+N(l,k-1,3))/3) \dots$   
 $((N(l+1,k-1,1)+N(l+1,k-1,2)+N(l+1,k-1,3))/3) \dots$   
 $((N(l-1,k,1)+N(l-1,k,2)+N(l-1,k,3))/3) \dots$   
 $((N(l,k,1)+N(l,k,2)+N(l,k,3))/3) \dots$   
 $((N(l+1,k,1)+N(l+1,k,2)+N(l+1,k,3))/3) \dots$   
 $((N(l-1,k+1,1)+N(l-1,k+1,2)+N(l-1,k+1,3))/3) \dots$   
 $((N(l,k+1,1)+N(l,k+1,2)+N(l,k+1,3))/3) \dots$   
 $((N(l+1,k+1,1)+N(l+1,k+1,2)+N(l+1,k+1,3))/3)];$

$XX(1) = \text{var}(X1);$   
 $XX(2) = \text{var}(X2);$   
 $XX(3) = \text{var}(X3);$   
 $XX(4) = \text{var}(X4);$   
 $XX(5) = \text{var}(X5);$   
 $XX(6) = \text{var}(X6);$   
 $XX(7) = \text{var}(X7);$   
 $XX(8) = \text{var}(X8);$   
 $XX(9) = \text{var}(X9);$

$YY = \min(XX);$   
 if  $YY == XX(1)$

$a = [N(l-2,k-2,1) N(l-1,k-2,1) N(l,k-2,1) \dots$   
 $N(l-2,k-1,1) N(l-1,k-1,1) N(l,k-1,1) N(l-2,k,1) \dots$   
 $N(l-1,k,1) N(l,k,1)];$

for  $i=1:9$   
 if  $a(1) == \min(a)$   
 $b(i) = a(1);$   
 $a(1) = a(1) + 1000;$   
 elseif  $a(2) == \min(a)$   
 $b(i) = a(2);$   
 $a(2) = a(2) + 1000;$   
 elseif  $a(3) == \min(a)$   
 $b(i) = a(3);$   
 $a(3) = a(3) + 1000;$   
 elseif  $a(4) == \min(a)$   
 $b(i) = a(4);$   
 $a(4) = a(4) + 1000;$   
 elseif  $a(5) == \min(a)$   
 $b(i) = a(5);$   
 $a(5) = a(5) + 1000;$   
 elseif  $a(6) == \min(a)$   
 $b(i) = a(6);$   
 $a(6) = a(6) + 1000;$   
 elseif  $a(7) == \min(a)$   
 $b(i) = a(7);$   
 $a(7) = a(7) + 1000;$

```
elseif a(8)==min(a)
    b(i)=a(8);
    a(8)=a(8)+1000;
elseif a(9)==min(a)
    b(i)=a(9);
    a(9)=a(9)+1000;
end
end
Y2(l,k,1)= b(5) ;
```

```
end
```

```
end
```

Rapport-Gratuit.com

## **PAGE DE RENSEIGNEMENTS**

Nom : RAJAONARIVONY  
Prénoms : Malala Tsimaniry  
Adresses : Logt N°1295 Cité 67Ha Antananarivo(101)  
Tél : 032 02 57102  
E-mail : [rajaonarivonytsimaniry@yahoo.fr](mailto:rajaonarivonytsimaniry@yahoo.fr)

Titre de mémoire : « RESTAURATION ET AMELIORATION DE LA VIDEO PAR  
TRAITEMENT NUMERIQUE D'IMAGE »

Nombre de pages : 71  
Nombre de tableaux : 01  
Nombre de figures : 69

Mots clés : Restauration  
Amelioration  
Histogramme  
Filtrage linéaire  
Filtrage non-linéaire

Directeur de mémoire : M. BOTO ANDRIANANDRASANA Jean Espérant



### ***Résumé***

Le traitement numérique d'image est l'une des parties la plus importante pour plusieurs applications, à savoir les traitements de la vidéo. Il est indispensable que ce domaine soit un domaine de recherche pointu et très évolutif à cause de la forte demande de qualité. Pour cela, il faut alors connaître les différentes techniques et approche à utiliser. Ce mémoire est consacré à l'étude des différentes techniques d'amélioration de contraste et de restauration d'image par filtrage. La connaissance sur ces techniques nous a permis alors d'effectuer quelques simulations sous MATLAB. En effet, ce langage permet de faire les manipulations des matrices qui paraissent trop difficile pour certain langage.

### ***Abstract***

The numeric treatment of image is one of the most important parts for many applications, as the movie treatments. It is indispensable that this domain evolve in research domain because the are many demand concerning the movie quality. For that, it is necessary to know the different techniques and approach. This memory treats the different techniques of contrast improvement and an image's restoration by filtering. The knowledge on these techniques allowed us then to do some simulations with MATLAB. Indeed, this language permits to make the manipulations of the matrixes which appear so difficult for other language