

ECOLE SUPERIEURE D'INFOTRONIQUE D'HAÏTI (ESIH)

Support de cours

TRAITEMENT NUMERIQUE D'IMAGES

Réalisé par

Dr Narcisse Talla
narcisse.talla@gmail.com

2011-2012

Chapitre 1. Généralités sur l'image

1.1 Introduction

Ce premier chapitre a pour but de présenter ce que c'est qu'une image en général, et une image de télédétection en particulier. De façon générale, une image est un support d'informations. Elle présente les éléments d'une scène qui a été captée soit par un appareil photographique, soit par un satellite. Les images de télédétection sont généralement issues de capteurs embarqués à bord de satellites, et aussi parfois de photographies aériennes. Une image peut avoir plusieurs définitions, selon les contextes. En traitement du signal, on définit une image comme étant un *signal bidimensionnel*. Mathématiquement parlant, une image est une application IMG d'un sous-ensemble $M \times N$ de $\mathbf{IR} \times \mathbf{IR}$ vers l'ensemble des réels \mathbf{IR} , qui, à chaque couple de réels (x, y) associe le réel $IMG(x, y)$:

$$\begin{aligned} IMG & : (M \times N) \longrightarrow \mathbf{IR} \\ (x, y) & \mapsto IMG(x, y) \end{aligned} \quad (1.1)$$

Le sous-ensemble $(M \times N)$ est constitué de couples d'entiers (x, y) tels que $x \in \{0, 1, 2, \dots, NC\}$, et $y \in \{0, 1, 2, \dots, NL\}$. Comme nous le verrons plus loin une image numérique est à la base un tableau à deux dimensions. NC et NL représentent les **dimensions** de l'image numérique. NC est le nombre de colonnes de l'image et NL est le nombre de lignes de l'image.

Une image est constituée d'un ensemble de points élémentaires appelés **pixels** (vient du mot anglais **picture element**). Le couple de réels (x, y) représente la position spatiale d'un pixel, et la valeur $IMG(x, y)$ représente le **niveau de gris** du pixel. Le niveau de gris d'un pixel est une grandeur proportionnelle à l'intensité du signal réfléchi par ce pixel lorsqu'il est radié notamment par une onde électromagnétique. Les niveaux de gris de pixels ont une gamme variée de domaines de valeurs, selon le type d'image. Par exemple, certaines images en couleur ont des niveaux de gris variant entre 0 et 65500. La plupart des images utilisées en télédétection ont des niveaux de gris variant entre 0 et 255. Ce dernier type d'images est justement appelé **image en niveaux de gris codée sur 8 bits**. On peut encore trouver plusieurs détails du traitement numérique des images dans (Rosenfeld and Kak, 1982), (Schalkoff, 1989), (Jain, 1989), (Toumazet, 1990), et (Horaud et Monga, 1995). Sur le terrain, la taille d'un pixel correspond à la distance à partir de laquelle un capteur peut distinguer deux objets à la surface du sol. Cette distance est appelée **résolution** de l'image. Plus la résolution d'une image est fine, mieux on peut distinguer des objets rapprochés sur cette image. La

tendance actuelle est de mettre au point des capteurs ayant de plus en plus des résolutions fines. A titre d'exemples, le satellite Européen ERS1 offre une résolution de 12.5 mètres, et le satellite Français SPOT offre une résolution de 20 mètres. Les tableaux 1.1 et 1.2 présentent quelques exemples de satellites et engins spatiaux de télédétection, avec leur altitude, leur résolution spatiale, leurs longueurs d'onde et les types de capteur embarqués sur leur plateforme. Nous présentons respectivement dans les paragraphes qui suivent : la représentation d'une image numérique, les différents formats d'images, les images monobande et les images multibande.

Tableau 1.1 Exemples de satellites et engins spatiaux de télédétection. On y trouve : leur année de mise en orbite, leur altitude, leur résolution spatiale et la dimension d'une scène captée par les instruments à bord de ces engins spatiaux.

Satellite Ou Engin Spatial	Année de mise en orbite ou de vol	Altitude	Résolution spatiale	Dimension d'une scène
LANDSAT 5 (USA)	Mars 1985	705 km	30 x 30 m	185 x 172 km
SPOT (France)	Février 1986	822 km	20 x 20 m (10 x 10 m en mode Panchromatique)	60 x 60 km
RADARSAT (Canada)	Novembre 1995	793 km	10, 25, 50 et 100 m	50 à 500 km
ERS (Europe)	Juillet 1991	785 km	25 x 25 m (12.5 x 12.5 en mode PRI)	100 x 100 km
JERS (Japon)	Février 1992	568 km	18 x 24 m	75 x 75 km
Radar Aéroporté ESAR (USA)	1995	8 km	6 x 6 m	

Tableau 1.2 Exemples de satellites et engins spatiaux de télédétection. On y trouve : leurs longueurs d'onde, le type de capteur qu'ils transportent, leur fréquence de passage, et la disponibilité de leurs images au Cameroun.

Satellite Ou Engin Spatial	Fréquence ou Longueur d'onde	Type de capteur	Fréquence de passage	Disponibilité au Cameroun
LANDSAT 5 (USA)	0.76 – 0.90 µm (IR) 1.55 – 1.75 µm (IRM) 10.4 – 12.5 µm (IRT) 2.08 – 2.35 µm (IRM)	Thematic Mapper (TM) (radiomètre à balayage)	16 jours	Partielle (au Nord)
SPOT (France)	0.50 – 0.59 µm (B) 0.61 – 0.68 µm (V) 0.79 – 0.89 µm (IR) 0.51 – 0.73 µm (Panchromatique)	HRV (Haute Résolution Visible) (Radiomètre à barettes)	3 à 26 jours	Partielle (au Nord)
RADARSAT (Canada)	5.3 Ghz (Bande C) 5.66 cm Polarisation HH	RSO (Capteur actif)	16 jours (3 jours au Canada)	Totale

ERS (Europe)	5.3 Ghz (Bande C) 5.66 cm Polarisation VV	RSO (Capteur actif)	35 jours	Totale
JERS (Japon)	0.52 – 0.60 μm (B) 0.63 – 0.69 μm (V) 0.76 – 0.86 μm (R) 0.76 – 0.86 μm (IR) 1.60 – 1.71 μm (IRM) 2.01 – 2.12 μm (IRM) 2.13 – 2.15 μm (IRM) 2.27 – 2.40 μm (IRM) 5.3 Ghz (Bande C) 5.66 cm Polarisation VV	SOP (Senseur Optique) RSO (capteur actif)	44 jours	Partielle Localisée (Région de (Kribi)
Radar Aéroporté ESAR (USA)	5.3 Ghz (Bande C) 5.66 cm Polarisation VV	RSO (Capteur actif)		Localisée (Région de Douala)

1.2 Représentation d'une image numérique

Une image numérique est à la base un tableau bidimensionnel dont les valeurs représentent les niveaux de gris des pixels se trouvant aux coordonnées correspondantes. Un exemple de portion d'une image en niveaux de gris est présenté sur la figure 1.2a. La représentation d'une image telle qu'elle apparaît sur un écran d'ordinateur ou sur un écran de télévision se fait par une correspondance entre les niveaux de gris et l'aspect plus ou moins lumineux de cet écran. En télédétection, on associe alors à chaque niveau de gris (ou à des niveaux de gris appartenant à un certain intervalle) une couleur particulière. La représentation de l'image numérique de la figure 1.2a est représentée sur la figure 1.2b. L'algorithme de correspondance utilisé est présenté sur la figure 1.1. La figure 1.3 présente une portion 375 x 200 pixels d'une image réelle en niveaux de gris. Il s'agit d'une image radar ERS1 de la ville de Douala.

Pour chaque pixel Pix en (x,y)

Si $0 \leq f(x, y) \leq 50$ alors attribuer à Pix la couleur NOIR SINON

Si $51 \leq f(x, y) \leq 100$ alors attribuer à Pix la couleur ROUGE SINON

Si $101 \leq f(x, y) \leq 150$ alors attribuer à Pix la couleur BLEU SINON

Attribuer à Pix la couleur BLANC.

Figure 1.1 Exemple d'algorithme de correspondance entre les niveaux de gris d'une image et différentes couleurs.

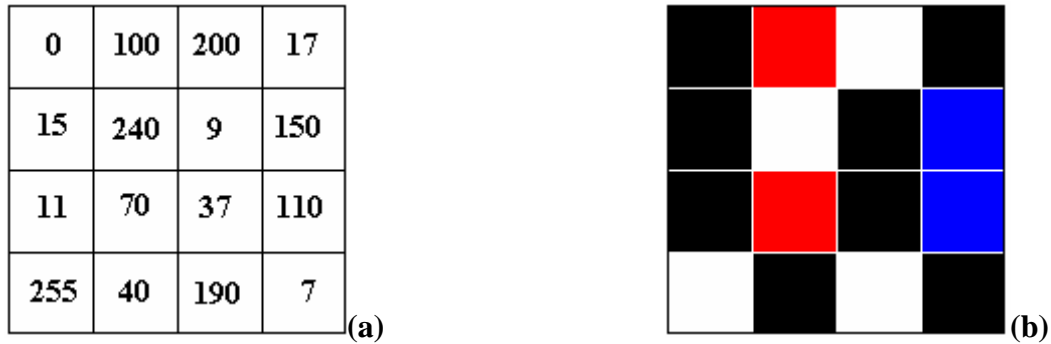


Figure 1.2 (a) Portion 4x4 d'une image en niveaux de gris (b) représentation colorée de la même portion d'image selon l'algorithme de la figure 1.1.



Figure 1.3 Portion 375 x 200 pixels d'une image en niveaux de gris. Il s'agit d'une image ERS1 (12.5 m de résolution) de la côte atlantique Camerounaise.

1.3 Les différents formats d'images

Les images numériques sont enregistrées dans des supports de données informatiques selon différents formats. Chaque valeur de niveau de gris est représentée sur un certain nombre d'octets, selon le nombre de niveaux de gris maximal. Ainsi, il y a par exemple des images codées sur 8 bits et des images codées sur 16 bits. Pour les images 8 bits, chaque niveau de gris est représenté sur un octet, et le nombre total de niveaux de gris possible est donc 2^8 , soit 256. Le procédé de conversion d'une image 16 bits en une image 8 bits est présenté au chapitre 2.

Dans la plupart des formats d'images, les valeurs de niveaux de gris sont stockées de façon séquentielle dans un fichier. Généralement, un certain nombre d'octets se trouvant en début de fichier contiennent des informations sur les dimensions de l'image (nombre de lignes et nombre de colonnes) et sur d'autres types d'informations (format, date d'acquisition, capteur, polarisation...) Dans certains cas, le fichier image est accompagné d'un fichier séparé (souvent appelé **fichier entête**) contenant ces informations supplémentaires. C'est le cas des images ERS produites par l'Agence Spatiale Européenne (ESA ou European Space Agency).

1.4 Les images monobandes et les images multibandes

Les capteurs d'images embarqués dans des satellites de télédétection comportent souvent plusieurs bandes spectrales. Dans de tels cas, une même zone est captée sous des longueurs d'onde différentes, produisant plusieurs images de la même région. L'ensemble de ces images constitue une seule image appelée image **multibande**. Chaque image de l'ensemble est appelée **canal** ou encore **bande spectrale**. C'est par exemple le cas du satellite français SPOT qui a trois bandes spectrales dénommées XS1, XS2 et XS3. La prise de vue d'une même région sur plusieurs bandes spectrales offre l'avantage d'obtenir plus de renseignements sur la zone filmée. En effet, certains objets n'étant pas sensibles à une certaine longueur d'onde peuvent l'être à une autre, et vice-versa. Il est donc avantageux d'avoir des images sur plusieurs bandes spectrales pour être mieux renseigné sur la zone étudiée. La figure 1.4 présente trois portions 180x180 pixels d'une image SPOT de la ville de Yaoundé. Ces images ont été prises avec des longueurs d'onde différentes. Une image prise à une seule bande spectrale est appelée image **monobande**.

Dans une image multibande, chaque pixel a plusieurs niveaux de gris, chaque niveau de gris correspondant à une bande spectrale. On peut donc définir mathématiquement une image multibande (n bandes avec $n > 1$) comme étant une application I d'un sous-ensemble $M \times N$ de $\mathbb{R} \times \mathbb{R}$ vers \mathbb{R}^n , qui, à chaque couple (x,y) appartenant à $M \times N$ associe un n-uplet $(I_1(x,y), I_2(x,y), \dots, I_n(x,y))$, chaque $I_k(x,y)$ étant le niveau de gris du pixel positionné en (x,y) dans la bande spectrale k :

$$\begin{aligned}
 I & : (M \times N) \longrightarrow \mathbb{R}^n \\
 (x, y) & \mapsto (I_1(x, y), I_2(x, y), \dots, I_n(x, y)) \qquad (1.2)
 \end{aligned}$$

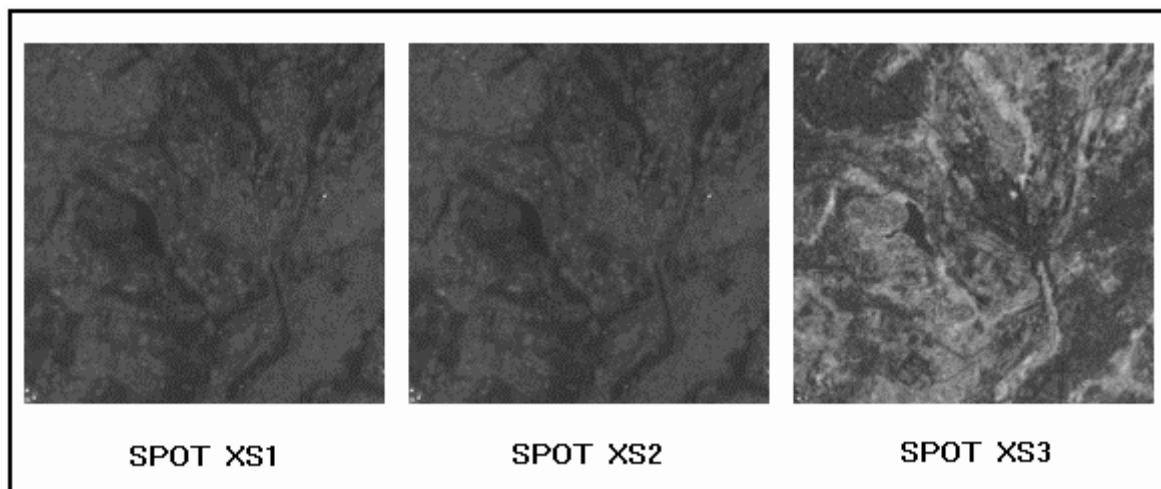


Figure 1.4 Trois portions 180x180 pixels d'une image multibande SPOT (20 mètres de résolution) de la ville de Yaoundé.

1.5 Exercices

- ◆ Qu'est-ce qu'une image numérique ?
- ◆ Comment obtient-on une image ?
- ◆ Qu'appelle-t-on résolution d'une image ?
- ◆ Comment représente-t-on une image sur un écran d'ordinateur ?
- ◆ Comment sont stockées les images dans les supports de données informatiques ?
- ◆ Qu'appelle-t-on fichier-entête ?
- ◆ Qu'est-ce qu'une image monobande et une image multibande ?
- ◆ Quel est l'intérêt d'avoir une image multibande ?
- ◆ Etant donnée l'image numérique présentée sur la figure 1.5, représentez-la en couleur en utilisant l'algorithme de la figure 1.1. Indiquez les couleurs par des lettres : N pour noir, R pour rouge, B1 pour bleu et B2 pour blanc.

10	99	100	255	200	180	55
78	87	23	45	33	77	90
0	55	23	67	88	114	45
77	99	92	90	87	70	88
159	95	200	77	0	0	63
87	74	0	11	74	14	43
44	77	98	49	84	45	23

Figure 1.5 Une image numérique de dimensions 7x7 pixels.

Chapitre 2. La lecture et l'affichage d'une image

2.1 Introduction

Le but de ce chapitre est de présenter la technique de lecture et d'affichage d'une image stockée dans un support de données informatique (disque dur d'un ordinateur ou disquette, par exemple). Pour programmer la lecture et l'affichage d'une image sur un écran d'ordinateur, il est nécessaire de connaître le format dans lequel l'image est stockée dans le support de données. Dans le cadre de ce chapitre, nous supposons que le nombre de lignes et le nombre de colonnes de l'image à afficher sont connus. Cela implique que nous traiterons directement le fichier de données, et non le fichier entête. Dans les lignes qui suivent, la lecture et l'affichage des images au format 8 bits sont d'abord abordées. Par la suite, la méthode de lecture et d'affichage d'une image au format 16 bits est présentée.

2.2 Lecture et affichage des images au format 8 bits

Nous avons vu au chapitre 1 que chaque valeur de niveau de gris d'une image au format 8 bits est représentée sur un octet, et que les valeurs de niveaux de gris sont stockées de façon séquentielle dans le support de données. La figure 2.1 schématise ce mode de stockage de données en prenant pour exemple l'image numérique de la figure 1.5.

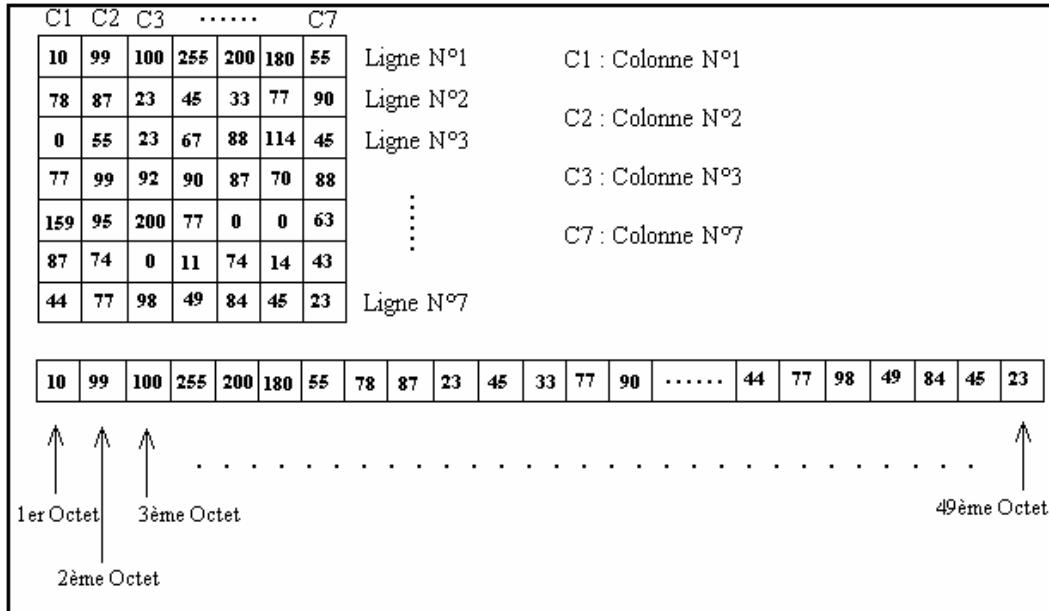


Figure 2.1 Représentation séquentielle des octets contenant des valeurs de niveaux de gris.

On peut voir sur cette figure que les valeurs de niveaux de gris sont stockées les unes après les autres, en commençant par la première ligne de l'image, et ensuite par la seconde, jusqu'à parcourir toutes les lignes de l'image. Une fois que ce principe de stockage est compris, il est maintenant facile de programmer la lecture et l'affichage d'une image sur un écran

d'ordinateur. Nous allons présenter trois méthodes de lecture : la méthode de lecture par pixel, la méthode de lecture par ligne, et la méthode de lecture globale.

2.2.1 La méthode de lecture d'une image pixel par pixel

Cette méthode consiste simplement à lire le fichier image octet par octet, et d'afficher les couleurs correspondantes aux valeurs de niveaux de gris au fur et à mesure qu'elles sont lues. Cette méthode est très lente et n'est plus utilisée dans les logiciels de traitement d'images. L'algorithme de cette méthode est présentée sur la figure 2.2.

```
Programme Lecture1_8bits  
Variables d'entrées :  
    NL : Nombre de lignes de l'image ;  
    NC : Nombre de colonnes de l'image ;  
    NomFic : Nom du fichier image (fichier d'octets).  
Variables locales : x ,y : de type entier ;  
    Pix : de type octet.  
DEBUT  
♦ Ouvrir le fichier image associé à NomFic;  
♦ POUR y variant de 0 à NL-1 FAIRE  
  POUR x variant de 0 à NC-1 FAIRE  
    DEBUT  
      -Lire un octet dans le fichier et le mettre dans la variable Pix ;  
      -Afficher la couleur correspondante à la valeur Pix à la position  
        (x,y) de l'écran ;  
    FIN  
♦ Fermer le fichier image ;  
FIN du programme.
```

Figure 2.2 Algorithme de lecture d'une image 8 bits par la méthode **pixel par pixel**.

2.2.2 La méthode de lecture d'une image ligne par ligne

Cette méthode consiste à définir une variable de type ligne d'image et de parcourir le fichier image en lisant une ligne entière à chaque fois. La différence avec la méthode de lecture par pixel est que cette fois-ci, on lit une ligne entière chaque fois qu'on accède au fichier image, au lieu de lire un seul pixel. Par expérience, nous avons constaté que cette méthode est au moins deux fois plus rapide que la méthode de lecture pixel par pixel. L'algorithme de cette méthode est présenté sur la figure 2.3.

```
Programme Lecture2_8bits
Variables d'entrées :
    NL : Nombre de lignes de l'image ;
    NC : Nombre de colonnes de l'image ;
    NomFic : Nom du fichier image ;
              (fichier d'octets)
Variables locales : x , y : de type entier ;
    LL : de type vecteur ligne d'octets.
DEBUT
♦ Ouvrir le fichier image associé à NomFic;
♦ POUR y variant de 0 à NL-1 FAIRE
    DEBUT
    Lire NC octets dans le fichier et mettre les données
    dans la variable LL ;
    POUR x variant de 0 à NC-1 FAIRE
    DEBUT
        -Afficher la couleur correspondante à la valeur LL[x] à la
        position (x,y) de l'écran ;
    FIN
    FIN
♦ Fermer le fichier image ;
FIN du programme.
```

Figure 2.3. Algorithme de lecture d'une image 8 bits par la méthode *ligne par ligne*.

2.2.3 La méthode de lecture globale d'une image

Cette méthode consiste à définir une variable de type image et de lire en une seule fois l'image entière. Cette méthode, beaucoup plus rapide que les méthodes précédentes, est celle qui est utilisée dans la plupart des logiciels modernes de traitement d'images. L'algorithme de cette méthode est présenté sur la figure 2.4.

```

Programme Lecture3_8bits
Définition de Type :
    Image : Tableau bidimensionnel contenant
              des entiers de type octet.
Variable d'entrée :
    NomFic : Nom du fichier image ;
              (fichier de type image)
Variable locale :
    BitMap : Pointeur vers une variable de type image.

DEBUT
    ♦ Ouvrir le fichier image associé à NomFic ;
    ♦ Allouer de l'espace mémoire au pointeur BitMap ;
    ♦ Lire en une seule fois le contenu du fichier et le mettre dans
      variable pointée par BitMap ;
    ♦ Envoyer le contenu de variable pointée par BitMap vers la
      mémoire d'écran de l'ordinateur (affichage) ;
    ♦ Fermer le fichier image ;
    ♦ Libérer l'espace mémoire alloué au pointeur BitMap ;
FIN du programme.
    
```

Figure 2.4. Algorithme de lecture d'une image 8 bits par la méthode de lecture globale.

2.2.4 Lecture et affichage d'une zone d'intérêt

Dans certains cas, on peut être intéressé à ne lire l'image qu'à partir d'un point particulier, de manière à extraire une zone d'intérêt (figure 2.5).

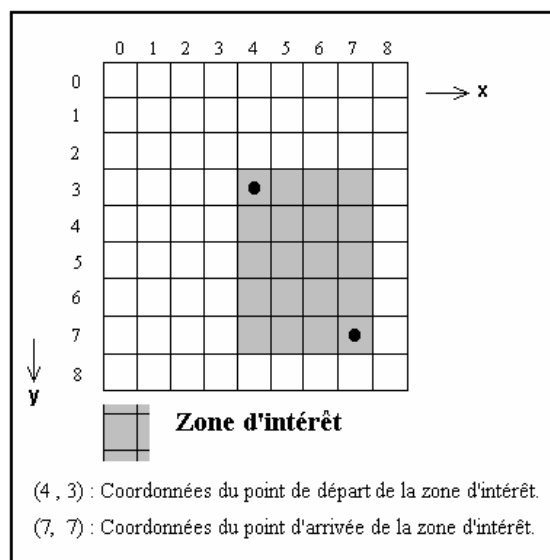


Figure 2.5. Illustration d'une zone d'intérêt sur une image numérique.

Dans ce cas, les coordonnées du point de départ et les coordonnées du point d'arrivée de la zone d'intérêt doivent être prises en compte lors de la lecture du fichier image. En particulier, il faut remarquer que le lecteur de fichier doit être positionné sur l'octet correspondant au point de départ, dans le cas de la lecture octet par octet. L'octet de départ est l'octet numéro

$((Y_o * NC) + X_o)$, (X_o, Y_o) étant les coordonnées du point de départ, et NC étant le nombre de colonnes de l'image. Il faut aussi noter que l'image représentant la zone d'intérêt a pour dimensions $(X_a - X_o + 1)$ colonnes et $(Y_a - Y_o + 1)$ lignes, X_a et Y_a étant les coordonnées du point d'arrivée de la zone d'intérêt. Dans le cas d'une lecture octet par octet, le lecteur de fichier doit être positionné après chaque lecture de $(X_a - X_o + 1)$ octets, sinon on lira des octets qui ne font partie de la zone d'intérêt. L'algorithme de lecture et d'affichage d'une zone d'intérêt est présenté sur la figure 2.6, pour le cas d'une lecture octet par octet.

La lecture et l'affichage d'une zone d'intérêt est plus simple avec la méthode de lecture ligne par ligne. Il suffit de positionner le lecteur de fichier à la ligne numéro Y_o , Y_o étant l'ordonnée du point de départ de la zone d'intérêt. Il faut ensuite prendre soin de n'afficher les pixels qu'à partir de la position X_o , jusqu'à la position X_a ; X_o et X_a étant respectivement l'abscisse du point de départ et l'abscisse du point d'arrivée de la zone d'intérêt. L'algorithme de lecture et d'affichage d'une zone d'intérêt est présenté sur la figure 2.7, pour le cas d'une lecture ligne par ligne.

```

Programme Lecture4_8bits
Variables d'entrées :
    Xo, Yo : Coordonnées du point de départ de la zone d'intérêt ;
    Xa, Ya : Coordonnées du point d'arrivée de la zone d'intérêt ;
    NL : Nombre de lignes de l'image ;
    NC : Nombre de colonnes de l'image ;
    NomFic : Nom du fichier image (fichier d'octets).
Variables locales : x ,y : de type entier ;
                    Pix : de type octet.
DEBUT
    ♦ Ouvrir le fichier image associé à NomFic;
    ♦ POUR y variant de Yo à Ya FAIRE
        DEBUT
            Positionner le lecteur de fichier sur l'octet n°  $(y * NC) + Xo$  ;
            POUR x variant de Xo à Xa FAIRE
                DEBUT
                    -Lire un octet dans le fichier et le mettre dans la variable Pix ;
                    -Afficher la couleur correspondante à la valeur Pix à la position
                       $(x - Xo, y - Yo)$  de l'écran ;
                FIN
            FIN
    ♦ Fermer le fichier image ;
FIN du programme.
    
```

Figure 2.6. Algorithme de lecture d'une zone d'intérêt sur une image 8 bits par la méthode pixel par pixel.

```

Programme Lecture5_8bits
Variables d'entrées :
    Xo, Yo : Coordonnées du point de départ de la zone d'intérêt ;
    Xa, Ya : Coordonnées du point d'arrivée de la zone d'intérêt ;
    NL : Nombre de lignes de l'image ;
    NC : Nombre de colonnes de l'image ;
    NomFic : Nom du fichier image ;
                (fichier de type octets)
Variables locales : x ,y : de type entier ;
                LL : de type vecteur ligne d'octets.
DEBUT
    ♦ Ouvrir le fichier image associé à NomFic;
    ♦ Positionner le lecteur de fichier sur l'octet numéro Yo*NC
    ♦ POUR y variant de Yo à Ya FAIRE
        DEBUT
            Lire NC octets dans le fichier et mettre les données
            dans la variable LL ;
            POUR x variant de Xo à Xa FAIRE
                DEBUT
                    -Afficher la couleur correspondante à la valeur LL[x] à la
                    position (x-Xo, y-Yo) de l'écran ;
                FIN
            FIN
        ♦ Fermer le fichier image ;
    FIN du programme.
    
```

Figure 2.7. Algorithme de lecture d'une zone d'intérêt sur une image 8 bits par la méthode **ligne par ligne**.

Dans le cas d'une lecture par octet, le fichier doit être défini comme étant un fichier d'**octets** (**file of byte**, dans le langage Turbo Pascal, par exemple). Et dans ce cas, le déplacement du lecteur de fichier se fait octet par octet. Dans le cas de la méthode de lecture par lignes, le fichier est aussi défini comme un fichier d'**octets**, mais la lecture se fait par blocs d'octets (on lit plusieurs octets par une seule instruction). Les valeurs correspondant aux octets lus sont stockées dans une variable de type **ligne d'octets**. Après chaque lecture, le lecteur de fichier se déplace d'un nombre d'octets égal au nombre d'octets lus.

2.3 Lecture et affichage des images au format 16 bits

Un moyen possible d'afficher une image au format 16 bits consiste à d'abord la convertir au format 8 bits, et de l'afficher ensuite par l'une des méthodes présentées dans les paragraphes précédents. Nous allons donc présenter la méthode de conversion d'une image 16 bits en une image 8 bits.

2.3.1 Conversion d'une image au format 16 bits en une image au format 8 bits

Les pixels des images au format 16 bits ont des niveaux de gris représentés par des valeurs complexes. Ce sont donc des valeurs à partie réelle et à partie imaginaire. On trouve

couramment deux types d'images 16 bits : les images en amplitudes et les images en intensités. Pour les images en intensités, le niveau de gris de chaque pixel a pour valeur l'intensité du nombre complexe représentant ce niveau de gris, c'est-à-dire la somme du carré de la partie réelle et du carré de la partie imaginaire. Ce nombre est très grand, et c'est pour cela qu'il est représenté sur 16 bits au lieu de 8 bits. Pour les images en amplitudes, chaque niveau de gris est représenté par la racine carrée de l'intensité du nombre complexe correspondant à ce niveau de gris. Il importe de souligner qu'il existe aussi des images d'amplitudes obtenues à partir d'un moyennage d'amplitudes. Les niveaux de gris des pixels de ce dernier type d'image sont déjà codés sur 8 bits. Nous allons présenter la méthode de conversion de chacun des types d'images 16 bits en une image au format 8 bits.

2.3.1.1 Cas des images 16 bits en amplitudes

Pour convertir une image au format 16 bits en amplitudes en une image au format 8 bits, il faut d'abord calculer les statistiques de l'image 16 bits. Etant donné un tableau image (**ima**) de **NL** lignes par **NC** colonnes, la moyenne (**moy**), la moyenne des carrés (**moycarr**) et l'écart-type (**sigma**) des données contenues dans ce tableau se calculent par les expressions suivantes :

$$\left\{ \begin{array}{l} \text{moy} = \frac{1}{\text{NL} \times \text{NC}} \sum_{y=0}^{\text{NL}-\text{INC}-1} \sum_{x=0}^{\text{NC}-1} \text{ima}[y, x] \\ \text{moycarr} = \frac{1}{\text{NL} \times \text{NC}} \sum_{y=0}^{\text{NL}-\text{INC}-1} \sum_{x=0}^{\text{NC}-1} (\text{ima}[y, x])^2 \\ \text{sigma} = \sqrt{\text{moycarr} - (\text{moy})^2} \end{array} \right. \quad (2.1)$$

L'algorithme de conversion d'une image 16 bits en une image 8 bits est présenté sur la figure 2.8. L'image 8 bits résultante est contenue dans un tableau d'entiers de type **octets**. La lecture d'une zone d'intérêt dans une image au format 16 bits se fait de la même manière que celle d'une image au format 8 bits. L'algorithme de lecture d'une zone d'intérêt sur une image au format 16 bits (en amplitudes) est présenté sur la figure 2.9.

```

Programme Conversion16_8bits_amp
Variables d'entrées :
    NL, NC : Nombre de lignes et de colonnes de l'image ;
    NomFic : Nom du fichier image 16 bits(fichier d'octets) ;
Variables locales : x ,y : de type entier ;
    LL : de type vecteur ligne d'octets (ayant au moins NC
    éléments).
    IMA : tableau NL x NC de réels
    IM8 : tableau NL x NC d'octets (image 8 bits) ;
DEBUT
    ♦ Ouvrir le fichier image associé à NomFic;
    ♦ POUR y variant de 0 à NL-1 FAIRE
        DEBUT
            Lire NC octets dans le fichier et mettre les données
            dans la variable LL ;
            POUR x variant de 0 à NC-1 FAIRE IMA[y,x] ← LL[x] ;
        FIN
    ♦ Calculer la moyenne (moy), la moyenne des carrés (moycarr) et
    l'écart-type (sigma) des données du tableau réel IMA ;
    ♦ POUR tous les points (x,y) du tableau IM8 FAIRE :
        SI (IMA[y,x] > 255) ALORS IM8[y,x] ← 255
        SINON IM8[y,x] ← ENT((255*IMA[y,x])/(moy + 3*sigma)) ;
        (ENT représente la partie entière)
        (l'image 8 bits est contenue dans le tableau IM8)
    ♦ Fermer le fichier image ;
FIN du programme.
    
```

Figure 2.8. Algorithme de conversion d'une image 16 bits en amplitudes en une image 8 bits.


```

Programme LectZone_16bits
Variables d'entrées :
    NL, NC : Nombre de lignes et de colonnes de l'image ;
    NomFic : Nom du fichier image 16 bits(fichier d'octets) ;
    (Xo,Yo,Xa,Ya) : Coordonnées de la zone d'intérêt ;
Variables locales : x ,y : de type entier ;
    LL : de type vecteur ligne d'octets (ayant au moins NC
    éléments).
    IMA : tableau NL x NC de réels
    IM8 : tableau (Ya-Yo) x (Xa-Xo) d'octets (image 8 bits) ;
DEBUT
    ♦ Ouvrir le fichier image associé à NomFic;
    ♦ Positionner le lecteur de fichier sur l'octet numéro Yo*NC ;
    ♦ POUR y variant de Yo à Ya FAIRE
        DEBUT
            Lire NC octets dans le fichier et mettre les données
            dans la variable LL ;
            POUR x variant de Xo à Xa FAIRE IMA[y-Yo,x-Xo] ← LL[x] ;
        FIN
    ♦ Calculer la moyenne (moy), la moyenne des carrés (moycarr) et
    l'écart-type (sigma) des données du tableau réel IMA ;
    ♦ POUR tous les points (x,y) du tableau IM8 FAIRE :
        SI (IMA[y,x] > 255) ALORS IM8[y,x] ← 255
        SINON IM8[y,x] ← ENT((255*IMA[y,x])/(moy + 3*sigma)) ;
        (ENT représente la partie entière)
        (l'image 8 bits est contenue dans le tableau IM8)
    ♦ Fermer le fichier image ;
FIN du programme.
    
```

Figure 2.9. Algorithme de lecture d'une zone d'intérêt sur image au format 16 bits.

2.3.1.2 Cas des images 16 bits en intensités (images PRI)

Pour convertir une image au format 16 bits en intensités en une image au format 8 bits, il faut tenir compte du fait que la valeur sur 8 bits de chaque niveau de gris est fonction de deux octets consécutifs, dans l'enregistrement de l'image. Avant de calculer les statistiques de l'image, il faut d'abord faire la conversion 16 bits / 8 bits, selon l'algorithme suivant :

$$\left\{ \begin{array}{l} P_{16} = (256 \times P_L) + P_F \quad (P_{16}: \text{Valeur sur 16 bits}) \\ P_8 = P_{16}/3 \quad (P_8: \text{Valeur sur 8 bits}) \\ \text{Si } P_8 \geq 255 \quad \text{Alors } P_8 = 255 \end{array} \right. \quad (2.2)$$

Dans les expressions (2.2), P_L représente la valeur de l'octet de poids lourd, et P_F représente la valeur de l'octet de poids faible. Quand on écrit un programme de lecture d'une image 16 bits en intensités, il faut bien noter que le nombre de pixels par ligne de l'image est égal à la moitié du nombre d'octets représentant une ligne dans le fichier image, puisque le niveau de

gris de chaque pixel est représenté sur deux octets. La figure 2.11 illustre notamment le principe de conversion d'une image 16 bits en intensités en une image 8 bits.

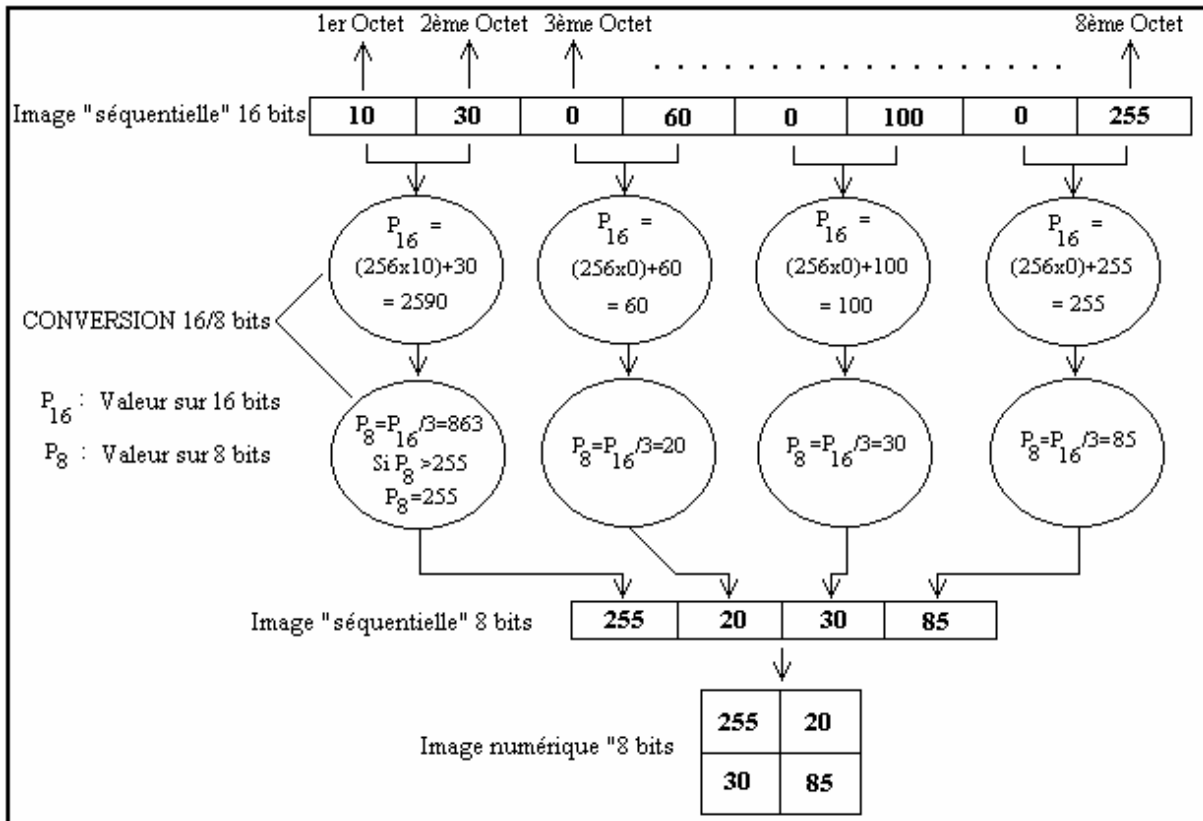


Figure 2.11. Illustration du principe de conversion d'une image 16 bits en intensités en une image au format 8 bits.

L'algorithme détaillé de conversion d'une image 16 bits en intensités en une image au format 8 bits est présenté sur la figure 2.12.

```

Programme Conversion16_8bits_int
Variables d'entrées :
    NL, NC : Nombre de lignes et de colonnes de l'image ;
    NomFic : Nom du fichier image 16 bits(fichier d'octets) ;
Variables locales : x1, x2, y : de type entier ; p8, p16, pL, pF : réels ;
    LL : de type vecteur ligne d'octets ayant au moins 2*NC
    éléments.
    IMA : tableau NL x NC de réels
    IM8 : tableau NL x NC d'octets (image 8 bits) ;
DEBUT
    ♦ Ouvrir le fichier image associé à NomFic;
    ♦ POUR y variant de 0 à NL-1 FAIRE
        DEBUT
            Lire 2*NC octets dans le fichier et mettre les données
            Dans la variable LL ;
            x1 ← 0; x2 ← 0;
            TANT QUE (x1 ≤ (2*NC)-1) FAIRE
                DEBUT
                    PL ← LL[x1]; pF ← LL[x1+1]; p16 ← (256*pL) + pF ;
                    p8 ← p16 / 3; SI (p8 > 255) ALORS p8 ← 255;
                    IMA[y,x2] ← p8; x1 ← x1 + 2; x2 ← x2 + 1;
                FIN
            FIN
        ♦ Calculer la moyenne (moy), la moyenne des carrés (moycarr) et l'écart-
        type (sigma) des données du tableau réel IMA ;
        ♦ POUR tous les points (x,y) du tableau IM8 FAIRE :
            SI (IMA[y,x] > 255) ALORS IM8[y,x] ← 255
            SINON IM8[y,x] ← ENT((255*IMA[y,x])/(moy + 3*sigma)) ;
            (ENT représente la partie entière)
            (l'image 8 bits est contenue dans le tableau IM8)
        ♦ Fermer le fichier image ;
    FIN du programme.
    
```

Figure 2.12. Algorithme de conversion d'une image 16 bits en intensités en une image 8 bits.

2.4 Exercices

- ♦ Comment sont disposées des données images dans un support informatique ?
- ♦ Expliquer les différentes méthodes de lecture et d'affichage d'une image numérique ;
- ♦ Une image au format 16 bits en intensités est représentée par la série d'octets présentée sur la figure 2.14.
 - 1) Quels sont les nombres de lignes et de colonnes de cette image ?
 - 2) Convertir cette image en une image au format 8 bits.
- ♦ Ecrire un programme de lecture et d'affichage d'une image au format 16 bits (en intensités et en amplitudes) dans un langage informatique de votre choix ;

0	177	23	55	0	10	70	1	90	255	0	200	100	80	40	20
---	-----	----	----	---	----	----	---	----	-----	---	-----	-----	----	----	----

Figure 2.14. Valeurs des octets représentant une image 16 bits en intensités.

Chapitre 3. La convolution d'une image avec un masque linéaire

3.1 Introduction

Dans le domaine des signaux et des systèmes, la convolution est une opération qui permet d'obtenir la sortie y d'un système, lorsqu'on connaît le signal d'entrée x et la réponse impulsionnelle h du système, comme illustré sur la figure 3.1 ci-dessous.

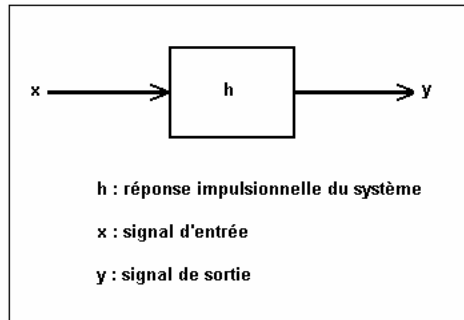


Figure 3.1. Illustration d'un système avec ses éléments d'entrée et de sortie.

Etant donné un tel système, le signal de sortie y est obtenu par la convolution du signal d'entrée x avec la réponse impulsionnelle h du système, selon l'expression suivante

$$y = x * h \Leftrightarrow y(t) = \int_0^t x(t - \lambda)h(\lambda)d\lambda \tag{3.1}$$

Dans le cas des signaux numériques tels que les images numériques, cette opération de convolution se fait de façon discrète. Dans ce cas, l'opérateur d'intégration devient une simple sommation, selon l'expression suivante.

$$y(k,l) = \sum_{m=0}^M \sum_{n=0}^N x(k - m,l - n) \cdot h(k,l) \tag{3.2}$$

Dans cette expression, x représente une fonction image et h représente un masque discret de convolution. M et N représentent les dimensions de l'image et y est le résultat de la convolution de l'image x avec le masque h . Nous présentons dans ce qui suit, le principe détaillé de la convolution d'une image avec un masque linéaire.

3.2 Principe

Un masque linéaire est une matrice à coefficients entiers, par exemple :

$$M = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

La convolution d'une image avec un masque linéaire comporte les étapes suivantes :

- 1) Positionner le masque sur chaque point de l'image ;
- 2) Calculer la combinaison linéaire des coefficients du masque avec les niveaux de gris des points voisins du point central (point sur lequel est centré le masque) ;
- 3) Remplacer le niveau de gris du point central par le résultat de la combinaison linéaire ;
- 4) Mettre le résultat obtenu à l'échelle des niveaux de gris de l'image originale.

Voici un exemple de calcul de combinaison linéaire. Considérons la portion d'image suivante :

$$I = \begin{bmatrix} 4 & 0 & 100 & \dots & 240 \\ 2 & 7 & 4 & \dots & 100 \\ 6 & 3 & 1 & \dots & 50 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 190 & 0 & 255 & 75 & 40 \end{bmatrix}$$

Positionnons le masque M sur le point I(1,1), de niveau de gris 7. Ce point correspond donc à la valeur 8 du masque. Les points autour du 7 correspondent aux valeurs -1 autour du 8 dans le masque. L'application du masque au point 7 donne le résultat suivant :

$$C = (-1 \times 4) + (-1 \times 0) + (-1 \times 100) + (-1 \times 2) + (8 \times 7) + (-1 \times 4) + (-1 \times 6) + (-1 \times 3) + (-1 \times 1)$$

On obtient $C = -64$. On normalise le résultat d'après la relation suivante :

$$NG_{\text{new}} = \text{ENT} \left(\frac{NG_{\text{old}} \times C}{NG_{\text{max}}} \right). \tag{3.3}$$

NG_{new} est le niveau de gris du pixel courant dans l'image résultat. NG_{old} est le niveau de gris de ce même pixel dans l'image originale. C est le résultat de la combinaison linéaire effectuée sur le pixel courant. NG_{max} est le niveau de gris maximal de l'image originale, et ENT représente l'opérateur "partie entière". En supposant que le niveau de gris maximal de l'image originale c'est 200, dans l'image résultat, le pixel traité aura pour niveau de gris :

$$\text{ENT} \left(\frac{7 \times -64}{200} \right) = \text{ENT}(-2.24) = -2 \tag{3.4}$$

La normalisation est nécessaire pour pouvoir visualiser nettement l'image résultat. Elle permet de ramener les niveaux de gris de l'image résultat dans le domaine de variation des niveaux de gris de l'image originale. Pour obtenir la convolution de l'image entière avec le masque proposé, on applique le principe précédent à tous les pixels de l'image. L'algorithme général de convolution est présenté ci-dessous.

Algorithme de convolution

Cet algorithme comporte deux parties :

a) Sous-programme CONV1(Xo, Yo, Ima, M, L, H, C)

CONV1(Xo, Yo, Ima, M, L, H, C) calcule le produit de convolution d'une image avec un masque M de dimensions L x H uniquement au point de coordonnées (Xo, Yo) de l'image. Le résultat est placé dans la variable C.

◆ Description des variables :

(Xo, Yo) : Coordonnées du point où est positionné (centré) le masque.

M : Matrice de dimensions L x H représentant le masque de convolution.

Ima : Tableau bidimensionnel contenant l'image.

C : résultat de l'application du masque M au point de coordonnées (Xo, Yo).

◆ Corps de l'algorithme :

C ← 0;

Pour y variant de (Yo - H/2) à (Yo + H/2) **Faire**

Pour x variant de (Xo - L/2) à (Xo + L/2) **Faire**

Début

I ← x - Xo + (L/2);

J ← y - Yo + (H/2);

C ← C + (Ima[x,y] * M[J, I]);

Fin

//Normalisation

C ← ENT(C*Ima[x,y]/Max(Ima(i,j))); //ENT : partie entière

Fin du Sous-programme.

b) Sous-programme CONV(ImaIn, M, L, H, ImaOut)

Ce sous-programme calcule le produit de convolution d'une image de NC colonnes et NL lignes, contenue dans un tableau **ImaIn** avec un masque **M** de dimensions **L x H**. Le résultat de la convolution est placé dans un tableau **ImaOut**.

Début

Pour y variant de 0 à NL-1 **Faire**

Pour x variant de 0 à NC-1 **Faire**

Début

CONV1(x, y, Ima, M, L, H, C) ;

ImaOut[x,y] ← C ;

Fin

Fin du Sous-programme

3.3 Le traitement des points de bordure

Il y a plusieurs façons de gérer les points de bordure. Nous ne présenterons que les 3 méthodes suivantes : la méthode des zéros, la méthode symétrique et la méthode de la symétrie circulaire.

3.3.1 La méthode des zéros

Dans cette approche, on considère que tous les points qui ne font pas partie de l'image ont le niveau de gris 0. On remplit donc de zéros les points extérieurs voisins des points de bordure, avant de faire la convolution. Le nombre de points extérieurs à considérer est fonction de la taille du masque de convolution. La figure suivante montre un exemple pour un masque de taille 3 x 3.

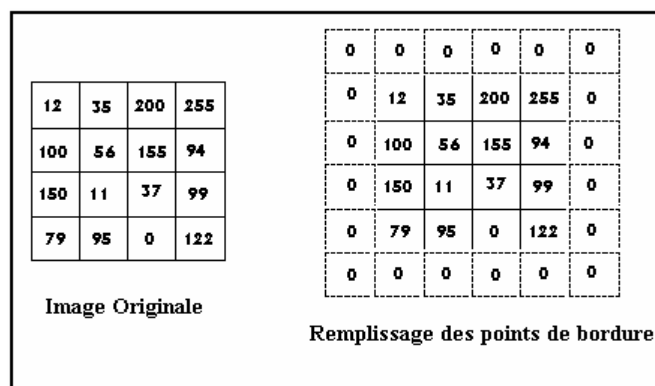


Figure 3.2. Remplissage des points de bordure selon la méthode des zéros.

3.3.2 La méthode symétrique

Dans cette approche, on attribue aux points extérieurs les niveaux de gris de leurs symétriques par rapport à la bordure la plus proche se trouvant dans l'image. Cette approche est illustrée sur la figure suivante.

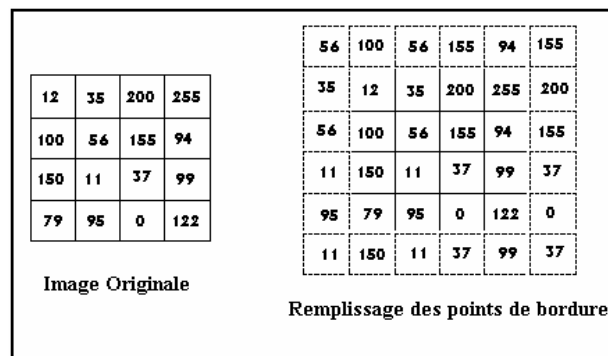


Figure 3.3. Remplissage des points de bordure selon la méthode symétrique.

3.3.3 La méthode de la symétrie circulaire

Dans cette approche, on agit comme si l'image se refermait sur elle-même, et les points extérieurs sont vus comme s'ils se retrouvaient de l'autre côté de l'image. L'image est vue

comme un tore. Il convient de noter que cette approche n'est pas réaliste sur le plan pratique, puisque la nature des pixels se trouvant sur un bord de la scène peut être complètement différente de celle des pixels se trouvant à l'autre extrémité. Cette approche est illustrée sur la figure suivante.

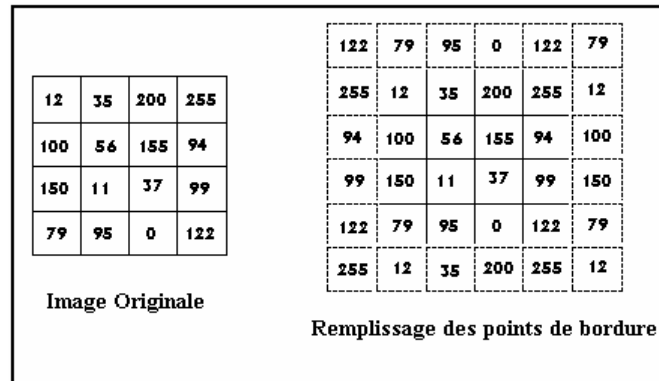


Figure 3.4. Remplissage des points de bordure selon la méthode de la symétrie circulaire.

3.4 Exercices

- ◆ Ecrire un programme de convolution d'une image avec un masque de taille quelconque, dans un langage informatique de votre choix. Utiliser le principe de symétrie pour les points de bordure.
- ◆ Faire la convolution de l'image présentée sur la figure 3.6 avec le masque présenté sur la même figure. Utiliser les trois principes de gestion des points de bordure illustrés dans le cours.

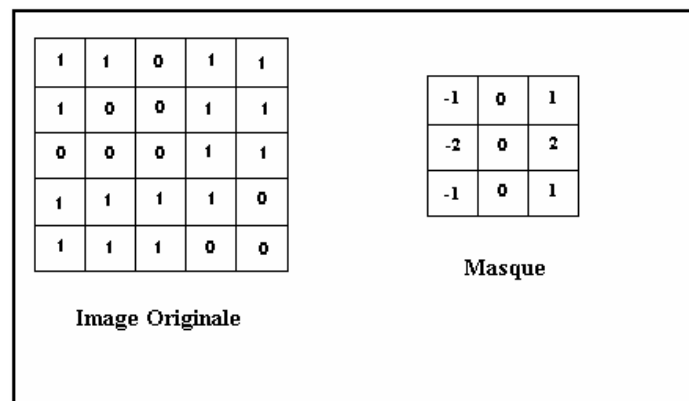


Figure 3.5. Image numérique et masque de convolution.

Chapitre 4. Le filtrage des images

4.1 Introduction

Les images captées par des satellites comportent souvent du bruit. Ce bruit est dû soit à la nature environnante, soit à la nature même des capteurs embarqués sur ces satellites. Cela fait que les images ont souvent un aspect brouillé à la réception. Il est alors nécessaire de filtrer ces images avant de les traiter. Il existe des filtres linéaires et des filtres non linéaires. Les images radar comportent souvent un type de bruit particulier appelé chatoiement (Nicolas, 1998). Le paragraphe 4.3 présente ce type de bruit et quelques algorithmes de filtrage de ce type de bruit. Le paragraphe qui suit présente quelques techniques de filtrage linéaire des images.

4.2 Le filtrage linéaire

Le filtrage linéaire consiste à appliquer à l'image des masques de convolution linéaires. Un masque de filtrage linéaire classique est le masque de la moyenne :

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Pour filtrer une image avec le filtre moyenne, il suffit de faire la convolution de l'image avec le masque de la moyenne, selon le principe présenté au chapitre 3. Les masques suivants sont généralement utilisés pour la détection des contours.

1) Masques de **Roberts** :

$$W_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad W_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad W_x: \text{direction horizontale, } W_y: \text{direction verticale.}$$

2) Masques de **Sobel** :

$$W_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad W_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad W_x: \text{direction horizontale, } W_y: \text{direction verticale.}$$

3) Masques de **Prewitt** :

$$W_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad W_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad W_x: \text{direction horizontale, } W_y: \text{direction verticale.}$$

4) Masques de **Kirsh** :

$$W_x = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad W_y = \begin{bmatrix} -3 & -3 & 3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad W_x: \text{direction horizontale, } W_y: \text{direction verticale.}$$

 5) Masques de **Roberts** pour 4 directions :

$$W_{0^\circ} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad W_{90^\circ} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad W_{45^\circ} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad W_{135^\circ} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

 6) Masques de **Sobel** pour 4 directions :

$$W_{0^\circ} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad W_{90^\circ} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad W_{45^\circ} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad W_{135^\circ} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

 7) Masques de **Prewitt** pour 4 directions :

$$W_{0^\circ} = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad W_{90^\circ} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad W_{45^\circ} = \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad W_{135^\circ} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

 8) Masques de **Kirsh** pour 4 directions :

$$W_{0^\circ} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad W_{90^\circ} = \begin{bmatrix} -3 & -3 & 3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad W_{45^\circ} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \quad W_{135^\circ} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

La figure 4.2 présente un exemple d'utilisation du masque de Sobel pour la détection des contours sur une image radar ERS1 de la côte atlantique camerounaise. Le masque de Sobel est utilisé ici pour les 4 directions classiques (0° , 45° , 90° et 135°). On effectue une convolution avec chacun des masques directionnels en chaque pixel de l'image, et on retient le maximum des amplitudes trouvées. Cette valeur maximale est considérée comme l'amplitude du gradient au pixel considéré. L'image des contours obtenue est présentée sur la figure 4.3.

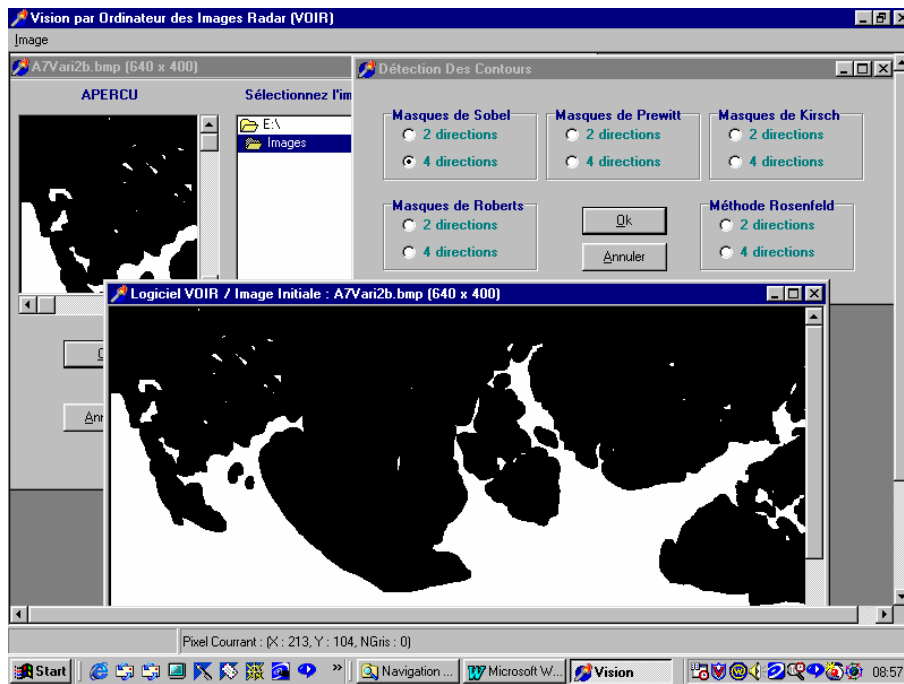


Figure 4.2 Choix d'une image pour la détection des contours par masques de Sobel. Il s'agit d'une image ERS1 de la côte atlantique camerounaise (source : logiciel VOIR).

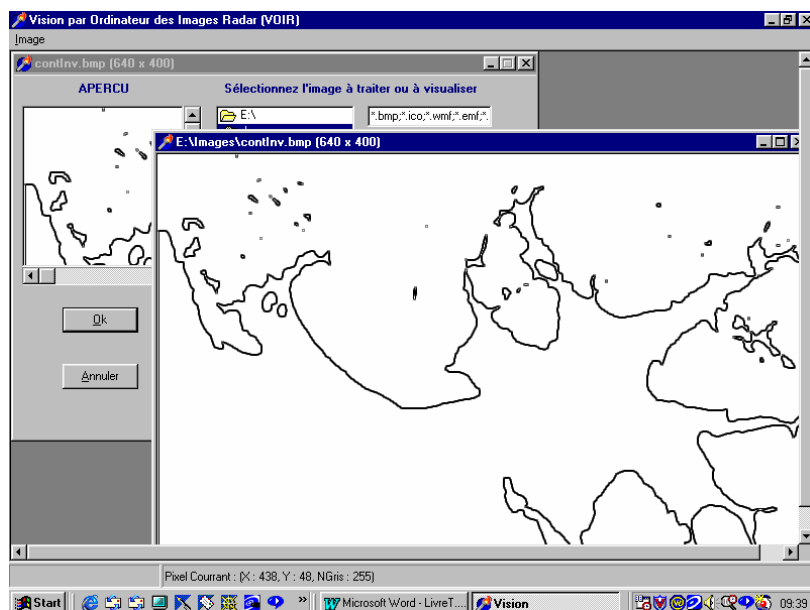


Figure 4.3 Résultat de la détection des contours de l'image de la figure 4.2 par les masques directionnels de Sobel (source : logiciel VOIR).

4.3 Le filtrage du chatolement des images RADAR

Les images radar comportent un bruit appelé chatolement, qui est dû à la propagation d'un faisceau cohérent dans une atmosphère caractérisée par des variations aléatoires. Le chatolement complique l'interprétation et le traitement des images radar. Plusieurs méthodes ont été proposées pour la réduction du chatolement. En général, il y a deux approches de filtrage des images RSO (Radar à Synthèse d'Ouverture) à une seule polarisation. La première

approche consiste globalement en un moyennage des valeurs radiométriques contenues dans les bandes spectrales disponibles. Ce moyennage entraîne une diminution de la variance du bruit. Dans la deuxième approche, le chatoiement est réduit dans le domaine spatial après que l'image soit formée. Dans les années passées, des filtres tels que le filtre médian et le filtre moyenne ont été appliqués aux images radar, mais sans grand succès. Cet échec est dû au fait que le chatoiement est un bruit multiplicatif et que les filtres mentionnés ne sont pas adaptatifs. Récemment, des techniques plus sophistiquées basées sur le modèle multiplicatif du bruit ont été développées. Les techniques de filtrage peuvent ou non supposer un modèle de chatoiement. Quand un modèle est supposé, il est généralement multiplicatif. On a dans ce cas-là :

$$z(k,l) = y(k,l)u(k,l)$$

où $z(k, l)$ est la valeur mesurée (intensité ou amplitude) du pixel (k, l) d'une image RSO, $y(k,l)$ sa valeur radiométrique réelle et $u(k, l)$ le bruit, caractérisé par une distribution de moyenne unité et d'écart-type σ_u . Ainsi, il y a une relation linéaire entre la moyenne et l'écart-type dans les régions homogènes, ce qui permet de tester l'hypothèse d'un bruit multiplicatif et de confirmer l'homogénéité d'une région. Le modèle multiplicatif est approprié pour le filtrage du chatoiement.

4.3.1 Le filtre des statistiques locales de LEE

En l'absence d'un modèle précis pour le signal original y , on utilise l'image elle-même pour estimer la moyenne et la variance à priori du signal, à partir de la moyenne locale \bar{z} et de la variance locale s_z^2 dans une fenêtre $n \times n$. A partir du modèle multiplicatif du bruit, on obtient un estimé de la moyenne et de la variance à priori :

$$\bar{y} = \frac{\bar{z}}{u} = \bar{z} \quad \text{et} \quad s_y^2 = \frac{s_z^2 - \bar{z}^2 s_u^2}{s_u^2 + 1}$$

On suppose qu'on a un filtre linéaire de la forme $\tilde{y} = a\bar{y} + bz$ où \tilde{y} est l'estimation minimum de la racine carrée de y , et a et b sont choisis de manière à minimiser l'erreur au sens des moindres carrés. Il a été prouvé que le meilleur estimé de y est :

$$\tilde{y} = \bar{z} + b(z - \bar{z}) \quad \text{avec} \quad b = \frac{s_y^2}{s_z^2}, \quad \bar{y} = \bar{z}$$

Le seul paramètre d'entrée du filtre est s_u qui dépend du traitement "multi-vues" RSO. Il est à veiller que s_y^2 ne soit pas négatif. Au cas où s_y^2 serait négatif, il faut lui attribuer la valeur zéro. Ce problème pourrait être évité si une limite inférieure d'homogénéité des régions est

convenablement fixée. Dans la méthode originale de Lee, un modèle linéaire est introduit, dans lequel la valeur optimale de b a l'expression suivante :

$$b' = \frac{s_y^2}{\bar{z}^2 s_u^2 + s_y^2}$$

Par substitution, on obtient : $b = \frac{s_y^2}{\bar{z}^2 s_u^2 + (1 + s_u^2) s_y^2}$, ce qui permet de comparer b et b' . Vu

que $(1 + s_u^2) \approx 1$, cette linéarisation n'affecte pas le filtrage du chatoiement dans le cas des images 4-vues en amplitude. Dans le cas des images RSO 1-vue, la formule exacte doit être utilisée. L'estimation de l'écart-type du bruit est fonction du nombre de vues et d'un facteur de qualité de l'image RSO. Avant de présenter les différents algorithmes de filtrage, nous présentons ci-dessous l'algorithme d'estimation de l'écart-type du chatoiement.

Algorithme d'estimation de l'écart-type du chatoiement

1. Fonction Gamma(n)

-Paramètre d'entrée : n , entier représentant le **nombre de vues** de l'image ;

-Variables locales : a, t : réels, k : entier.

Début

$a \leftarrow 1.7724538509$;

$t \leftarrow 1$;

Pour k variant de 1 à n **Faire** $t \leftarrow t * (2k - 1)$

$\text{Gamma} \leftarrow (a * t) / 2^n$

Fin

2. Fonction Ecart-Type_Bruit(n, Q)

-Paramètres d'entrée : n , entier représentant le **nombre de vues** de l'image ;

Q : Facteur représentant le type d'image à filtrer ;

$Q = 0$ pour une image en intensités (16 bits);

$Q = 1$ pour une image en amplitudes de type racine carrée des intensités (8 bits);

$Q = 2$ pour une image en amplitudes de type moyenne des amplitudes (8 bits);

Début

Si ($Q = 0$) **Alors** $\text{Ecart-Type_Bruit} \leftarrow 1/\sqrt{n}$ **Sinon**

Si ($Q = 1$) **Alors** $\text{Ecart-Type_Bruit} \leftarrow \sqrt{0.273/n}$ **Sinon**

Si ($Q = 2$) **Alors** $\text{Ecart-Type_Bruit} \leftarrow \sqrt{n \times \left(\frac{(n-1)!}{\text{Gamma}(n)} \right)^2 - 1}$

Fin

L'algorithme de filtrage d'une image radar par le filtre de Lee est présenté ci-dessous :

Algorithme du filtrage de LEE

1. Créer un tableau image intermédiaire $ima_f[]$ de type réel et y affecter l'image à filtrer. Effectuer les étapes suivantes pour chaque pixel (i, j) de l'image intermédiaire.

2. Calculer la moyenne locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$\bar{z} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ima_f[i, j] / n^2$$

3. Calculer la variance locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$S_z^2 = (1/(n^2-1)) \times \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (ima_f[i, j] - \bar{z})^2$$

4. Estimer l'écart-type du bruit dans la fenêtre :

$$\left\{ \begin{array}{l} S_u \leftarrow \text{Ecart_type_Bruit}(n, Q) \\ n : \text{Nombre de vues de l'image} \\ Q : \text{Facteur représentant le type d'image} \end{array} \right.$$

5. Estimer la moyenne et la variance a priori dans la même fenêtre :

$$\left\{ \begin{array}{l} \bar{y} = \bar{z} \\ S_y^2 = (S_z^2 - S_u^2) / (S_u^2 + 1) \end{array} \right.$$

6. Vérifier le signe de la variance a priori : Si $S_y^2 \leq 0$ Alors $S_y^2 = 0$

7. Calculer le coefficient b suivant : $b = \frac{S_y^2}{\bar{z}^2 \cdot S_u^2 + S_y^2}$

8. Changer le niveau de gris $ima_f[i, j]$ du pixel (i, j) selon l'affectation suivante :

$$ima_f[i, j] \leftarrow \bar{z} + b(ima_f[i, j] - \bar{z})$$

9. Transformer les niveaux de gris de l'image intermédiaire en des entiers et mettre le résultat dans le tableau image d'entiers de type **octets** :

Pour tous les pixels (i, j) de l'image intermédiaire, **Faire** :

$$ima[i, j] \leftarrow \text{ENT}(ima_f[i, j])$$

ENT est la fonction partie entière. Le tableau image $ima[]$ contient maintenant l'image filtrée.

4.3.2 Le filtre de LEE alterné (Lee, 1980 et 1981)

Le filtre de LEE alterné est une variante du filtre de LEE précédemment présenté. Son algorithme est présenté sur les lignes qui suivent.

Algorithme du filtre de LEE alterné

1. Créer un tableau image intermédiaire $ima_f[]$ de type réel et y affecter l'image à filtrer.
Effectuer les étapes suivantes pour chaque pixel (i, j) de l'image intermédiaire.

2. Calculer la moyenne locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$\bar{z} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ima_f[i, j] / n^2$$

3. Calculer la variance locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$S_z^2 = (1/(n^2 - 1)) \times \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (ima_f[i, j] - \bar{z})^2$$

4. Estimer l'écart-type du bruit dans la fenêtre :

$$\left\{ \begin{array}{l} S_u \leftarrow \text{Ecart_type_Bruit}(n, Q) \\ n : \text{Nombre de vues de l'image} \\ Q : \text{Facteur représentant le type d'image} \end{array} \right.$$

5. Estimer l'écart-type a priori dans la même fenêtre :

$$\left\{ \begin{array}{l} \text{Si } (\bar{z} \neq 0) \text{ Alors } S_y \leftarrow \sqrt{S_z^2 / \bar{z}} \\ \text{Sinon } S_y \leftarrow 1 \end{array} \right.$$

6. Comparer l'écart-type à priori et l'écart-type du bruit, et en déduire la valeur du niveau de gris du pixel filtré :

Si $(S_y < S_u)$ **Alors** $ima[i, j] \leftarrow \text{ENT}(\bar{z})$

Sinon effectuer les opérations suivantes

Début

$$a \leftarrow S_y^2 + S_u^4$$

$$b \leftarrow ima_f[i, j] \times (S_y^2 - S_u^2) + \bar{z} \times (S_u^2 + S_y^4)$$

$$b \leftarrow b/a$$

$$ima[i, j] \leftarrow \text{ENT}(b)$$

Fin

ENT est la fonction partie entière. Le tableau image $ima[]$ contient maintenant l'image filtrée.

4.3.3 Le Filtre à Maximum de Probabilité à Postérieur (MAP Filter, Kuan et al., 1987)

Ce filtre adaptatif est basé sur la maximisation de la probabilité à postérieur (MAP), $p(y/z)$ du signal $y(k,l)$ étant donné $z(k,l)$: $p(y/z) = \frac{p(z/y)p(y)}{p(z)}$. Pour une image RAS N-look en intensités, $p(z/y)$ a une distribution χ carrée :

$$p(z/y) = \frac{N^N z^{N-1}}{(N-1)! y^N} \exp\left(\frac{-N \cdot z}{y}\right)$$

Contrairement au filtre de Lee qui ne requiert aucun modèle pour le signal, le filtre MAP suppose que le signal y a une distribution gaussienne :

$$p(y) = \frac{1}{s_y \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{y - \bar{y}}{s_y}\right)^2\right)$$

ayant pour moyenne \bar{y} et pour variance s_y^2 , lesquels sont estimés par calcul des statistiques locales à l'intérieur d'une fenêtre, selon le même procédé utilisé dans le calcul du filtre de Lee. La valeur de \tilde{y} est obtenue par maximisation de la grandeur $\log[p(z/y)] + \log[p(y)]$ par rapport à y . La solution est une racine de l'équation cubique :

$$\tilde{y}^3 - \bar{z} \cdot \tilde{y}^2 + N \cdot s_y^2 \cdot \tilde{y} - N \cdot s_y^2 \cdot z = 0$$

La solution est la racine qui est positive, réelle et qui maximise la probabilité $p(y/z)$. La solution optimale est comprise entre z et \bar{z} , comme dans le cas du filtre de Lee. Le nombre N de "vues" utilisé pour générer l'image est généralement connu, à défaut il pourrait être estimé par comparaison de la distribution réelle et de la distribution théorique de l'image. Dans le dernier cas, la distribution réelle du chatoiement peut être évaluée en prenant s_z / \bar{z} valeurs dans des régions homogènes. Certains auteurs ont remplacé le modèle Gaussien par un modèle plus réaliste Gamma. Dans ce cas, on obtient une simple équation du second degré. L'algorithme de ce filtre est présenté ci-dessous.

Algorithme du filtre MAP (Kuan)

1. Créer un tableau image intermédiaire ima_f de type réel et y affecter l'image à filtrer. Effectuer les étapes suivantes pour chaque pixel (i, j) de l'image intermédiaire.
2. Calculer la moyenne locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$\bar{z} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ima_f[i, j] / n^2$$

3. Calculer la variance locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$S_z^2 = (1/(n^2-1)) \times \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (\text{ima}_f[i, j] - \bar{z})^2$$

4. Estimer l'écart-type du bruit dans la fenêtre :

$$\left\{ \begin{array}{l} S_u \leftarrow \text{Ecart_type_Bruit}(n, Q) \\ n : \text{Nombre de vues de l'image} \\ Q : \text{Facteur représentant le type d'image} \end{array} \right.$$

5. Estimer l'écart-type à priori dans la même fenêtre :

$$\left\{ \begin{array}{l} \text{Si } (\bar{z} \neq 0) \text{ Alors } S_y \leftarrow \sqrt{S_z^2 / \bar{z}} \\ \text{Sinon } S_y \leftarrow 0 \end{array} \right.$$

6. Comparer l'écart-type à priori et l'écart-type du bruit, et en déduire la valeur du niveau de gris du pixel filtré :

Si ($S_y < S_u$) **Alors** $\text{ima}[i, j] \leftarrow \text{ENT}(\bar{z})$

Sinon effectuer les opérations suivantes

Début

$$a \leftarrow S_y^2 \times (1 + S_u^2)$$

$$b \leftarrow \text{ima}_f[i, j] \times (S_y^2 - S_u^2) + (\bar{z} \times S_u^2 \times (1 + S_y^2))$$

$$b \leftarrow b/a$$

$$\text{ima}[i, j] \leftarrow \text{ENT}(b)$$

Fin

ENT est la fonction partie entière. Le tableau image $\text{ima}[]$ contient maintenant l'image filtrée.

4.3.4 Le filtre Gamma

Ce filtre est semblable au précédent, sauf qu'il tient compte du nombre de vues de l'image à filtrer. Voici son algorithme.

Algorithme du filtre Gamma

1. Créer un tableau image intermédiaire $\text{ima}_f[]$ de type réel et y affecter l'image à filtrer. Effectuer les étapes suivantes pour chaque pixel (i, j) de l'image intermédiaire.
2. Calculer la moyenne locale dans une fenêtre n x n centrée sur le pixel courant :

$$\bar{z} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \text{ima}_f[i, j] / n^2$$

3. Calculer la variance locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$S_z^2 = (1/(n^2-1)) \times \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (\text{ima}_f[i, j] - \bar{z})^2$$

4. Estimer l'écart-type du bruit dans la fenêtre :

$$\left\{ \begin{array}{l} S_u \leftarrow \text{Ecart - type_Bruit}(n, Q) \\ n : \text{Nombre de vues de l'image} \\ Q : \text{Facteur représentant le type d' image} \end{array} \right.$$

5. Estimer l'écart-type à priori dans la même fenêtre :

$$\left\{ \begin{array}{l} \text{Si } (\bar{z} \neq 0) \text{ Alors } S_y \leftarrow \sqrt{S_z^2 / \bar{z}} \\ \text{Sinon } S_y \leftarrow 0 \end{array} \right.$$

6. Comparer la variance à priori et la variance du bruit, et en déduire la valeur du niveau de gris du pixel filtré :

Si $(S_y^2 - S_u^2) \neq 0$ **Alors** on effectue les opérations suivantes

Début

$$\begin{aligned} \alpha &\leftarrow (1 + S_u^2) / (S_y^2 - S_u^2) \\ t_1 &\leftarrow \bar{z} \cdot (\alpha - N_{\text{vues}} - 1) \\ t_2 &\leftarrow t_1^2 + (4\alpha \cdot N_{\text{vues}} \cdot \text{ima}_f[i, j] \cdot \bar{z}) \\ t &\leftarrow t_1 + \sqrt{t_2} \\ t &\leftarrow t / (2\alpha) \\ \text{ima}[i, j] &\leftarrow \text{ENT}(t) \end{aligned}$$

Fin

ENT est la fonction partie entière, et N_{vues} est le nombre de vues de l'image à filtrer. Le tableau image $\text{ima}[\]$ contient maintenant l'image filtrée.

4.3.5 Le filtre de Frost (Frost et al., 1981)

Ce filtre adaptatif fait l'hypothèse que l'information utile a une fonction d'auto-corrélation exponentielle, c'est-à-dire que l'image est stationnaire dans un voisinage significatif du pixel filtré. Dans ce cas, on peut vérifier que le meilleur estimé du signal est :

$$\tilde{y} = \sum_{n,n} k_1 \cdot a \cdot \exp[-a(|k - k_0| + |l - l_0|)] \cdot z(k, l)$$

où (k_0, l_0) représente les coordonnées du pixel central, $z(k,l)$ la valeur du pixel aux coordonnées (k,l) , k_1 une constante de normalisation choisie tel que les poids du filtre aient toujours une somme égale à l'unité. La constante a est telle que $a = k_2 \cdot s_z^2 / \bar{z}^2$, où k_2 est une constante de normalisation telle que $k_2 = 4 / (n \cdot s_u^2)$. La valeur du pixel central est alors remplacée par une somme pondérée dont les poids diminuent avec la distance. De plus, les pixels centraux augmentent en poids quand la variance de la fenêtre croît, c'est-à-dire que le pixel central est plus ou moins filtré selon l'homogénéité de la fenêtre. L'algorithme du filtrage de Frost est présenté sur les lignes qui suivent.

Algorithme du filtrage de FROST

1. Créer un tableau image intermédiaire $ima_f[]$ de type réel et y affecter l'image à filtrer. Effectuer les étapes suivantes pour chaque pixel (i, j) de l'image intermédiaire.
2. Calculer la moyenne locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$\bar{z} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ima_f[i, j] / n^2$$

3. Calculer la variance locale dans une fenêtre $n \times n$ centrée sur le pixel courant :

$$S_z^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (ima_f[i, j] - \bar{z})^2$$

4. Estimer l'écart-type du bruit dans la fenêtre :

$$\left\{ \begin{array}{l} S_u \leftarrow \text{Ecart_type_Bruit}(n, Q) \\ n : \text{Nombre de vues de l'image} \\ Q : \text{Facteur représentant le type d'image} \end{array} \right.$$

5. Calculer le coefficient k suivant : $k = 4 / (n \cdot S_u^2)$
6. Calculer la constante a suivante : $a = k \cdot S_z^2 / \bar{z}^2$
7. Remplacer le niveau de gris $ima_f[i, j]$ du pixel (i, j) par la valeur suivante :

$$\left\{ \begin{array}{l} ima_f[i, j] = \sum_{m=i-(n/2)}^{i+(n/2)} \sum_{l=j-(n/2)}^{j+(n/2)} a \cdot ima_f[m, l] \cdot \exp(X) \\ \text{avec } X = -a(|m - i| + |l - j|) \end{array} \right.$$

8. Remettre les niveaux de gris de l'image intermédiaire à l'échelle des octets et mettre le résultat dans un tableau image d'entiers de type octets :

Pour tous les pixels (i, j) de l'image intermédiaire, **Faire** :

$$\text{ima}[i, j] \leftarrow \text{ENT}\left(\text{ima}_f[i, j] \times 255 / \text{ima}_{f_{\text{MAX}}}\right)$$

$\text{ima}_{f_{\text{MAX}}}$ est le niveau de gris maximal de l'image intermédiaire et ENT est la fonction partie entière. Le tableau image $\text{ima}[]$ contient maintenant l'image filtrée.

4.4 Exemples et illustration

Nous allons présenter un exemple de filtrage de LEE sur une portion d'image numérique de taille 3 x 3. L'image à filtrer est présentée sur la figure 4.4. On y voit également le remplissage des points de bordure selon le principe de symétrie. Nous allons d'abord filtrer cette image par le principe de LEE en utilisant une fenêtre de filtrage de taille 3 x 3.

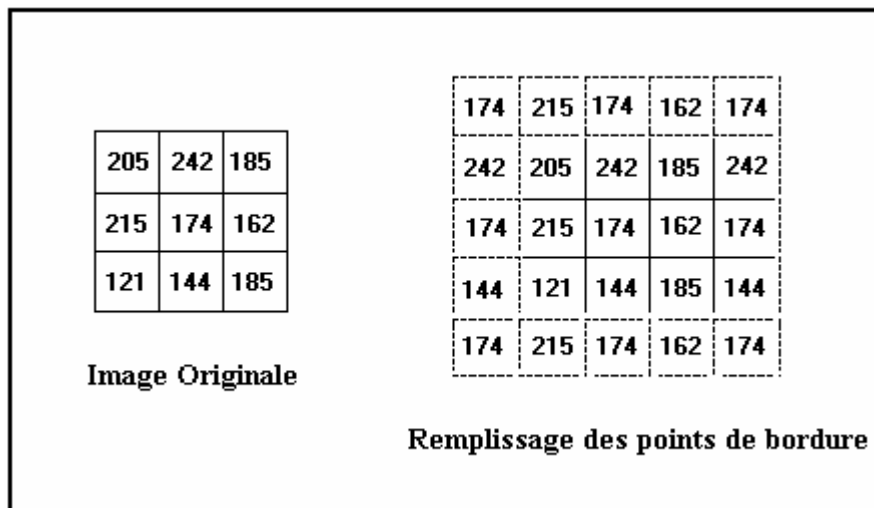


Figure 4.4 Portion d'image numérique à filtrer.

Nous allons procéder étape par étape en suivant l'algorithme de filtrage de LEE. Considérons une fenêtre de taille 3 x 3 centrée autour du pixel de niveau de gris 205. Cette fenêtre contient les valeurs suivantes :

$$W_{205} = \begin{bmatrix} 174 & 215 & 174 \\ 242 & 205 & 242 \\ 174 & 215 & 174 \end{bmatrix}$$

1- Calcul de la moyenne locale de la fenêtre :

$$\bar{z} = (174 + 215 + 174 + 242 + 205 + 242 + 174 + 215 + 174) / (3 \times 3) = 201.66$$

2- Calcul de la variance locale de la fenêtre :

$$\begin{aligned}
 S_z^2 &= ((174 - 201.66)^2 + (215 - 201.66)^2 + (174 - 201.66)^2 \\
 &\quad + (242 - 201.66)^2 + (205 - 201.66)^2 + (242 - 201.66)^2 \\
 &\quad + (174 - 201.66)^2 + (215 - 201.66)^2 + (174 - 201.66)^2) / (9 - 1) \\
 &= 6682 / 8 = 835.25
 \end{aligned}$$

3- Estimation de l'écart-type du bruit (on suppose que c'est une image en intensités (Q=1) et que le nombre de vues est n = 1) : $S_u \leftarrow \sqrt{0.273/1} = 0.5225$

4- Estimation de la moyenne et de la variance à priori :

$$\begin{cases} \bar{y} = \bar{z} = 201.66 \\ S_y^2 = (S_z^2 - S_u^2) / (S_u^2 + 1) = (835.25 - 0.5225^2) / (0.5225^2 + 1) = 655.9 \end{cases}$$

5- Vérification du signe de la variance à priori :

S_y^2 est positif, donc aucun changement.

6- Calcul du coefficient b :

$$b = \frac{S_y^2}{\bar{z}^2 \cdot S_u^2 + S_y^2} = \frac{655.9}{201.66^2 \times 0.5225^2 + 655.9} = 0.05578$$

7- Remplacement du niveau de gris 205 par la valeur suivante (dans le tableau image réel) :

$$\bar{z} + b(\text{ima}_r[i, j] - \bar{z}) = 201.66 + 0.05578 \times (205 - 201.66) = 201.84$$

8- Remplacement du niveau de gris 205 par la partie entière du réel 201.84 dans le tableau image résultat. Le pixel correspondant aura donc la valeur 201 dans l'image filtrée. Les fenêtres centrées autour des autres pixels de l'image sont représentées sur la figure 4.5.

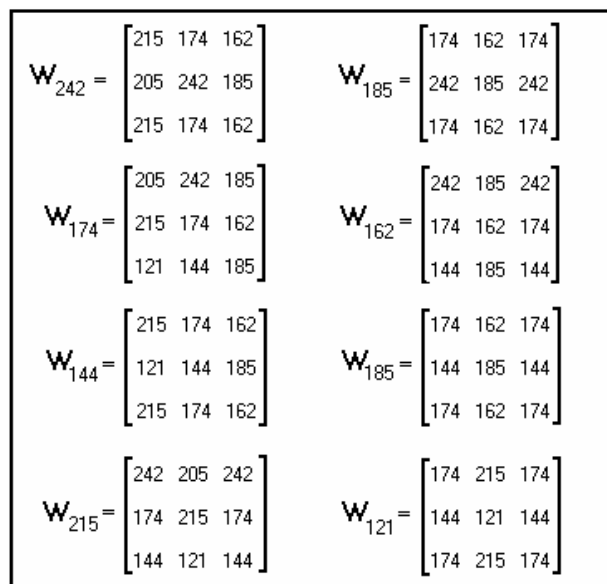


Figure 4.5 Différents masques centrés autour des pixels de l'image de la figure 4.4.

Le calcul effectué pour obtenir le filtrage du pixel de niveau de gris 205 est également effectué pour tous les autres pixels de l'image originale. Les résultats des calculs intermédiaires sont présentés sur le Tableau 4.1, et l'image filtrée est présentée sur la figure 4.6.

Masque	W_{242}	W_{185}	W_{174}	W_{162}	W_{144}	W_{185}	W_{215}	W_{121}
\bar{z}	192.66	187.66	181.44	183.55	172.44	165.88	184.55	170.55
S_z^2	777.5	997	1357.77	1326.53	932.3	202.10	1937	983
S_u	0.5225	0.5225	0.5225	0.5225	0.5225	0.5225	0.5225	0.5225
S_y^2	610.54	782.97	1066.4	1041.83	732	158.54	1521.4	771.97
b	0.05682	0.0753	0.1060	0.1017	0.0827	0.02067	0.1406	0.0886
$ima_f[i, j]$	195.46	187.46	180.65	181.35	170.08	166.27	188.83	166.15
$ima[i, j]$	195	187	180	181	170	166	188	166

Tableau 4.1. Résultats des calculs intermédiaires pour le filtrage de l'image numérique de la figure 4.4 par filtre de LEE.

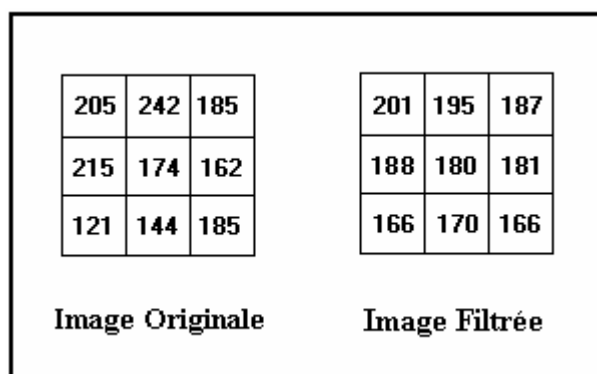


Figure 4.6 Résultat du filtrage de l'image numérique de la figure 4.4 par filtre de LEE.

De façon générale, aucune méthode de filtrage du chatoiement n'est meilleure par rapport aux autres. Toutefois, une méthode peut s'avérer meilleure par rapport aux autres dans une application particulière.

Chapitre 5. La morphologie mathématique appliquée aux images

5.1 Introduction

La morphologie mathématique est une technique d'analyse de l'image basée sur la notion des ensembles. Il va s'agir de faire des mesures sur une image, afin d'en faire une quantification. Pour cela, il faudrait construire l'ensemble dans lequel portera la mesure. Cette construction s'effectue par des transformations successives qui, partant de l'image brute, vont peu à peu mettre en évidence l'ensemble à mesurer. Les opérations de morphologie mathématique s'appliquent aux images binaires, mais elles peuvent aussi être appliquées aux images en niveaux de gris. Les opérations morphologiques se font à l'aide d'un élément structurant B qui a des formes géométriques simples : rond, carré, hexagone, octogone (figure 5.1).

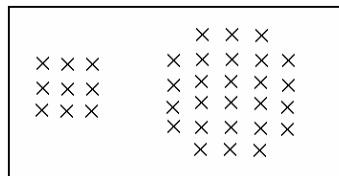


Figure 5.1. Eléments structurants carré et hexagonal.

Nous allons maintenant présenter les opérateurs morphologiques les plus courants, mais auparavant voyons ce que c'est que la binarisation.

5.2 La binarisation d'une image

La binarisation d'une image consiste à la transformer en une image ne possédant que deux niveaux de gris. Le résultat de la binarisation produit une image binaire. Pour binariser une image, on se fixe deux seuils de binarisation s_1 et s_2 . L'algorithme de binarisation est présenté ci-dessous.

Algorithme de binarisation d'une image

Pour tous les pixels (i, j) de l'image originale **Faire**

Début

Si $(\text{ima}[i, j] \geq S_1)$ ET $(\text{ima}[i, j] \leq S_2)$

Alors $\text{ima}_b[i, j] \leftarrow 1$

Sinon $\text{ima}_b[i, j] \leftarrow 0$

Fin

Le tableau ima_b contient l'image binaire.

La figure 5.2 montre un exemple de binarisation réalisé sur une portion d'image ERS1 autour de la ville de Douala.

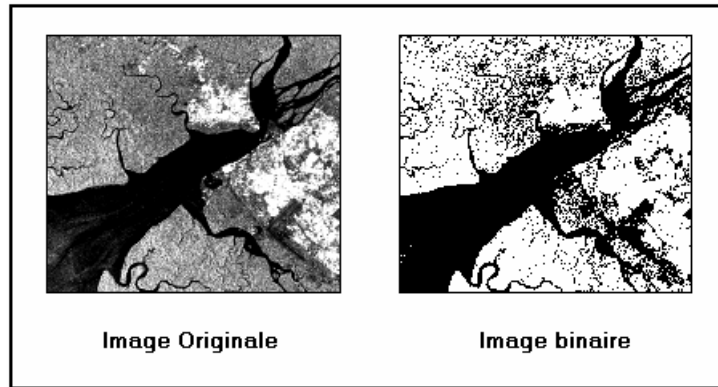


Figure 5.2. Binarisation d'une portion d'image ERS1 (183 x 159 pixels) de la région de Douala avec $S_1=0$ et $S_2=100$ pour seuils. (source : logiciel **VOIR**)

5.3 L'érosion d'une image binaire

L'érosion consiste à réduire les entités. Pour cela, on fait promener un élément structurant B sur le pourtour des entités. Le centre de l'élément structurant donne le contour de l'entité réduite. Une érosion peut être répétée plusieurs fois sur une même image, on parle alors d'**érosion de taille n, ou d'ordre n** : $E^{Bn}(X)$. L'algorithme de l'érosion est présenté ci-dessous.

Algorithme d'érosion

Créer deux tableaux ima_{b1} et ima_{b2} à **valeurs booléennes**, et de mêmes dimensions que l'image binaire initiale ima .

Pour tous les pixels (i,j) de niveau de gris $ima[i,j]$ de l'image initiale **Faire**

Début

$ima_{b1}[i, j] \leftarrow$ FAUX

$ima_{b2}[i, j] \leftarrow$ FAUX

Si $(ima[i, j] > 0)$ Alors $ima_{b1}[i, j] \leftarrow$ VRAI

Fin

Pour tous les pixels (i,j) **Faire**

Début

Si $(ima_{b1}[i - 1, j - 1])$

ET $(ima_{b1}[i - 1, j])$

ET $(ima_{b1}[i - 1, j + 1])$

ET $(ima_{b1}[i + 1, j - 1])$

ET $(ima_{b1}[i + 1, j])$

ET $(ima_{b1}[i + 1, j + 1])$

Alors $ima_{b2}[i, j] \leftarrow ima_{b1}[i, j]$

Sinon $ima_{b2}[i, j] \leftarrow$ FAUX

Fin

Noter bien que le ET est l'opérateur ET logique s'appliquant aux valeurs booléennes entre parenthèses. L'expression "Si $(\text{ima}_{b_2}[i, j]) \dots$ " veut dire "Si la valeur booléenne de $(\text{ima}_{b_2}[i, j])$ est VRAI...".

Pour tous les pixels (i, j) **Faire**

Début

Si $(\text{ima}_{b_2}[i, j])$ **Alors** $\text{ima}[i, j] \leftarrow 255$ (attribution de la couleur blanche au pixel positionné en (i, j))

Sinon $\text{ima}[i, j] \leftarrow 0$ (attribution de la couleur noire au pixel positionné en (i, j))

Fin

Afficher le tableau image ima contenant le résultat de l'érosion.

La figure 5.3 montre un exemple d'érosion appliquée sur l'image binaire de la figure 5.2.

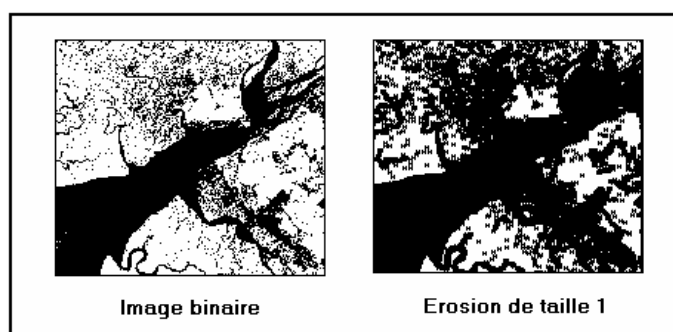


Figure 5.3. Erosion d'une portion d'image ERS1 (183 x 159 pixels)
De la région de Douala (source : logiciel VOIR)

5.4 La dilatation d'une image binaire

La dilatation a pour effet d'augmenter la taille des entités. Pour cela, l'élément structurant parcourt l'extérieur de l'entité. De même que pour l'érosion, le centre de l'élément structurant donne le contour de l'entité agrandie. Une dilatation peut aussi être répétée plusieurs fois sur une même image, on parle alors **de dilatation de taille n, ou d'ordre n : $D^{Bn}(X)$** . L'algorithme de la dilatation est présenté ci-dessous.

Algorithme de dilatation

1. Créer deux tableaux ima_{b_1} et ima_{b_2} à **valeurs booléennes**, et de mêmes dimensions que l'image binaire initiale ima.
2. **Pour** tous les pixels (i, j) de niveau de gris $\text{ima}[i, j]$ de l'image initiale **Faire**

Début

$\text{ima}_{b_1}[i, j] \leftarrow \text{FAUX}$

$\text{ima}_{b_2}[i, j] \leftarrow \text{FAUX}$

Si $(\text{ima}[i, j] > 0)$ **Alors** $\text{ima}_{b_1}[i, j] \leftarrow \text{VRAI}$

Fin

3. **Pour** tous les pixels (i,j) **Faire**

Début

Si $(\text{ima}_{b1}[i - 1, j - 1])$
 OU $(\text{ima}_{b1}[i - 1, j])$
 OU $(\text{ima}_{b1}[i - 1, j + 1])$
 OU $(\text{ima}_{b1}[i + 1, j - 1])$
 OU $(\text{ima}_{b1}[i + 1, j])$
 OU $(\text{ima}_{b1}[i + 1, j + 1])$
 Alors $\text{ima}_{b2}[i, j] \leftarrow \text{VRAI}$
 Sinon $\text{ima}_{b2}[i, j] \leftarrow \text{ima}_{b1}[i, j]$

Fin

Noter bien que le OU est l'opérateur OU logique s'appliquant aux valeurs booléennes entre parenthèses. L'expression "Si $(\text{ima}_{b2}[i, j]) \dots$ " veut dire "Si la valeur booléenne de $(\text{ima}_{b2}[i, j])$ est VRAI...".

4. **Pour** tous les pixels (i,j) **Faire**

Début

Si $(\text{ima}_{b2}[i, j])$ **Alors** $\text{ima}[i, j] \leftarrow 255$ (attribution de la couleur blanche au pixel positionné en (i,j))

Sinon $\text{ima}[i, j] \leftarrow 0$ (attribution de la couleur noire au pixel positionné en (i,j))

Fin

5. Afficher le tableau image ima contenant le résultat de la dilatation.

La figure 5.4 montre un exemple de dilatation appliquée sur l'image binaire de la figure 5.2.

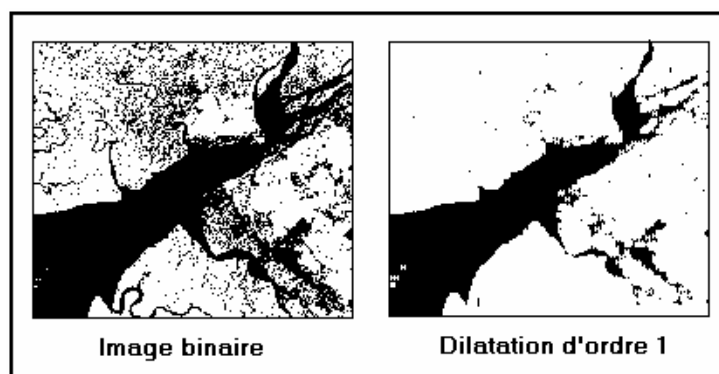


Figure 5.4. Dilatation d'une portion d'image ERS1 (183 x 159 pixels) de la région de Douala (source : logiciel **VOIR**)

5.5 L'ouverture d'une image binaire

Une ouverture consiste à éliminer des presque îles étroites des entités. Pour cela, on applique d'abord une érosion, et ensuite une dilatation : X_B (Serra, 1982). On peut aussi faire une **ouverture de taille n** en effectuant n érosions puis n dilatations : X_{bn} . L'algorithme de l'ouverture consiste simplement à faire d'abord une érosion telle que présentée au paragraphe 5.3, et ensuite une dilatation telle que présentée au paragraphe 5.4. La figure 5.5 montre un exemple d'ouverture appliquée sur l'image binaire de la figure 5.2.

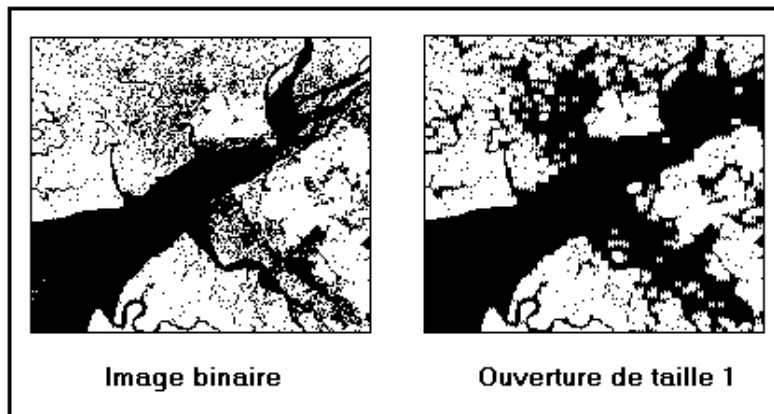


Figure 5.5. Ouverture d'une portion d'image ERS1 (183 x 159 pixels) De la région de Douala (source : logiciel VOIR)

5.6 La fermeture d'une image binaire

A l'inverse de l'ouverture, la fermeture est une opération qui ferme les golfes et les petits trous. Pour faire une fermeture, on applique d'abord une dilatation (voir algorithme du paragraphe 5.4), et ensuite une érosion (voir algorithme du paragraphe 5.3) : X^B (Serra, 1982). Une **fermeture de taille n (ou d'ordre n)** se définit par n dilatations suivies de n érosions : X^{bn} . La figure 5.6 montre un exemple de fermeture appliquée sur l'image binaire de la figure 5.2.

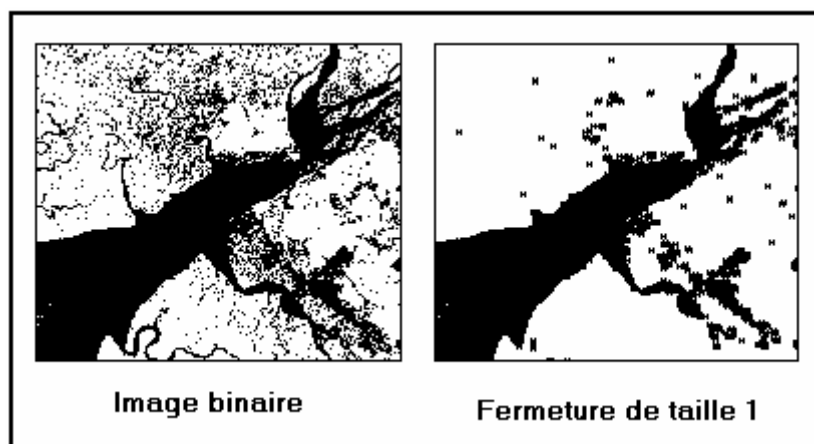


Figure 5.6. Fermeture d'une portion d'image ERS1 (183 x 159 pixels) de la région de Douala (source : logiciel VOIR)

5.7 Le Chapeau Haut de Forme

Les propriétés de la fermeture $f_b(x)$ et de l'ouverture $f^{Bn}(x)$ en teintes de gris permettent de définir des filtres appelés Chapeaux Haut de Forme qui sont utilisés pour extraire les pics de la fonction $f(x)$, correspondant aux petites zones claires de l'image ou aux vallées correspondant aux petites zones sombres. Ces zones sont représentées sur le graphe de la fonction $f(x)$ par des pics (resp. des vallées étroites) et l'extraction des pics d'épaisseur inférieure à n exige l'utilisation d'un élément structurant B_n de taille n (Debaine et al., 1988). On définit ces filtres ainsi :

- 1) Le Chapeau Haut de Forme Blanc ou White Top Hat : $WTH(X) = f(X) - f_{B_n}(X)$. Il est obtenu en faisant la différence entre l'image binaire originale et la fermeture de la même image. La figure 5.7 en présente une illustration.
- 2) Le Chapeau Haut de Forme Noir ou Black Top Hat : $BTH(X) = f^{B_n}(X) - f(X)$. Il est obtenu en faisant la différence entre l'image binaire originale et l'ouverture de la même image. La figure 5.8 en présente une illustration.

Les Chapeaux Haut de Forme Blanc et Noir sont des filtres dits morphologiques et sont adaptés pour l'extraction d'objets linéaires fins (sombres ou clairs), d'une épaisseur donnée. Ils ont été exploités par Legeley et Mering (1997) pour extraire les failles à partir d'images SPOT panchromatiques.

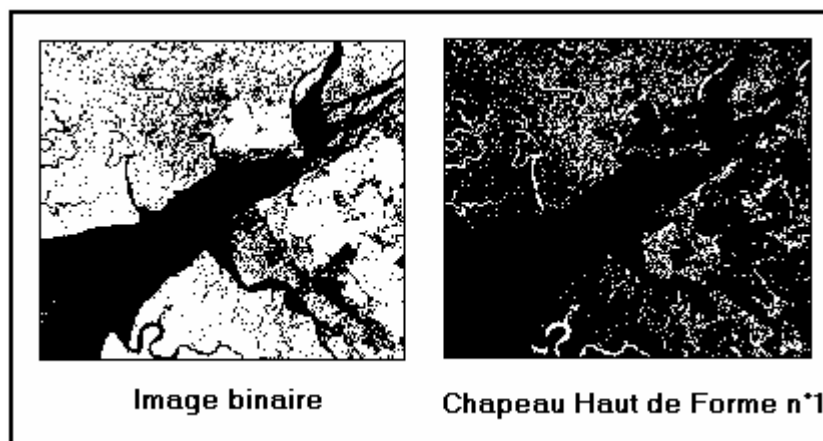


Figure 5.7 Chapeau Haut de Forme Blanc d'une portion d'image ERS1 (183 x 159 pixels) de la région de Douala (source : logiciel **VOIR**)

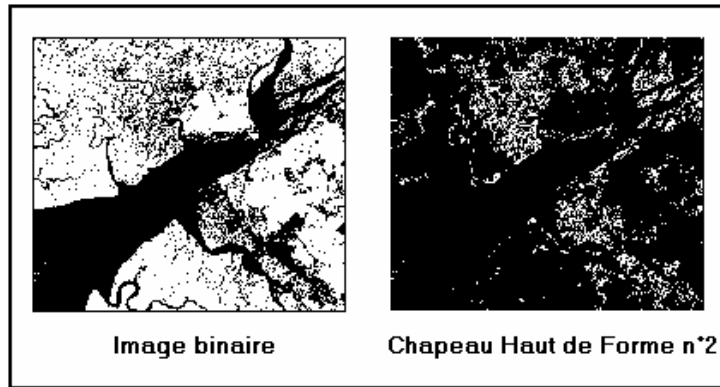


Figure 5.8 Chapeau Haut de Forme Noir d'une portion d'image ERS1 (183 x 159 pixels) de la région de Douala (source : logiciel **VOIR**)

5.8 La Squelettisation

La squelettisation consiste à réduire l'épaisseur des entités d'une image à la dimension d'un pixel. Lors du processus de squelettisation, les entités de l'image sont amincies par le biais d'un élément structurant, jusqu'à ce qu'il y ait stabilité. La squelettisation s'effectue par une succession d'opérations appelées amincissement, jusqu'à l'obtention d'une structure stable ne pouvant plus être amincie. Ceci peut être réalisé en balayant l'image avec une série de huit masques obtenus à partir d'un masque B1 par rotation de 45°:

$$\begin{aligned}
 B1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} &
 B2 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} &
 B3 &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} &
 B4 &= \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 B5 &= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} &
 B6 &= \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} &
 B7 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} &
 B8 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}
 \end{aligned}$$

L'algorithme de squelettisation est présenté ci-dessous.

Algorithme de squelettisation

1. Créer deux tableaux ima_{b1} et ima_{b2} à **valeurs booléennes**, et de mêmes dimensions que l'image binaire initiale ima .

2. **Pour** tous les pixels (i,j) de niveau de gris $ima[i,j]$ de l'image initiale **Faire**

Début

$ima_{b1}[i, j] \leftarrow$ FAUX

$ima_{b2}[i, j] \leftarrow$ FAUX

Si $(ima[i, j] = 0)$ Alors $ima_{b1}[i, j] \leftarrow$ VRAI

Fin

3. **Faire** les initialisations suivantes :

Iter \leftarrow 1;

Stable \leftarrow FAUX ("**Stable**" est une variable booléenne).

4. **Tant Que** la variable booléenne **Stable** a la valeur FAUX **Faire**

Début

Stable \leftarrow VRAI ;

Iter \leftarrow Iter+1 ;

//MASQUE Est

Pour tous les pixels (i,j) **Faire**

Début

SI (**Im** a_{b1}[i - 1, j - 1])

OU (**Im** a_{b1}[i - 1, j])

OU (**Im** a_{b1}[i - 1, j + 1])

OU(NON(**Im** a_{b1}[i, j]

ET (**Im** a_{b1}[i + 1, j - 1])

ET (**Im** a_{b1}[i + 1, j])

ET (**Im** a_{b1}[i + 1, j + 1])))

ALORS **Im** a_{b2}[i, j] \leftarrow **Im** a_{b1}[i, j]

SINON

Début

Im a_{b2}[i, j] \leftarrow FAUX;

Stable \leftarrow FAUX;

Fin

Fin

//MASQUE Sud-Est

Pour tous les pixels (i,j) **Faire**

Début

```

SI  ( Im ab2[i - 1, j])
OU  ( Im ab2[i - 1, j - 1])
OU  ( Im ab2[i, j - 1])
OU(  NON( Im ab2[i, j]
      ET  ( Im ab2[i + 1, j])
      ET  ( Im ab2[i + 1, j + 1])
      ET  ( Im ab2[i, j + 1]))))
ALORS  Im ab1[i, j] ← Im ab2[i, j]
SINON
Début
    Im ab1[i, j] ← FAUX;
    Stable ← FAUX;
Fin

```

Fin

//MASQUE Sud

Pour tous les pixels (i,j) **Faire**

Début

```

SI  ( Im ab1[i - 1, j - 1])
OU  ( Im ab1[i, j - 1])
OU  ( Im ab1[i + 1, j - 1])
OU(  NON( Im ab1[i, j]
      ET  ( Im ab1[i - 1, j + 1])
      ET  ( Im ab1[i, j + 1])
      ET  ( Im ab1[i + 1, j + 1]))))
ALORS  Im ab2[i, j] ← Im ab1[i, j]
SINON
Début
    Im ab2[i, j] ← FAUX;
    Stable ← FAUX;
Fin

```

Fin

//MASQUE Sud-Ouest

Pour tous les pixels (i,j) **Faire**

Début

```

SI  ( Im ab2[i, j - 1])
OU  ( Im ab2[i + 1, j - 1])
OU  ( Im ab2[i + 1, j])
OU(  NON( Im ab2[i, j]
      ET  ( Im ab2[i - 1, j])
      ET  ( Im ab2[i - 1, j + 1])
      ET  ( Im ab2[i, j + 1]))
ALORS  Im ab1[i, j] ← Im ab2[i, j]
SINON
Début
    Im ab1[i, j] ← FAUX;
    Stable ← FAUX;
Fin

```

Fin

//MASQUE Ouest

Pour tous les pixels (i,j) **Faire**

Début

```

SI  ( Im ab1[i + 1, j - 1])
OU  ( Im ab1[i + 1, j])
OU  ( Im ab1[i + 1, j + 1])
OU(  NON( Im ab1[i, j]
      ET  ( Im ab1[i - 1, j - 1])
      ET  ( Im ab1[i - 1, j])
      ET  ( Im ab1[i - 1, j + 1])
ALORS  Im ab2[i, j] ← Im ab1[i, j]
SINON
Début
    Im ab2[i, j] ← FAUX;
    Stable ← FAUX;
Fin

```

Fin

//MASQUE Nord-Ouest

Pour tous les pixels (i,j) **Faire**

Début


```

SI  ( Im ab2[i + 1, j])
OU  ( Im ab2[i + 1, j + 1])
OU  ( Im ab2[i, j + 1])
OU(  NON( Im ab2[i, j]
      ET  ( Im ab2[i - 1, j])
      ET  ( Im ab2[i - 1, j - 1])
      ET  ( Im ab2[i, j - 1]))))
ALORS  Im ab1[i, j] ← Im ab2[i, j]
SINON
Début
    Im ab1[i, j] ← FAUX;
    Stable ← FAUX;
Fin

```

Fin

//MASQUE Nord

Pour tous les pixels (i,j) **Faire**

Début

```

SI  ( Im ab1[i - 1, j + 1])
OU  ( Im ab1[i, j + 1])
OU  ( Im ab1[i + 1, j + 1])
OU(  NON( Im ab1[i, j]
      ET  ( Im ab1[i - 1, j - 1])
      ET  ( Im ab1[i, j - 1])
      ET  ( Im ab1[i + 1, j - 1]))))
ALORS  Im ab2[i, j] ← Im ab1[i, j]
SINON
Début
    Im ab2[i, j] ← FAUX;
    Stable ← FAUX;
Fin

```

Fin

//MASQUE Nord-Est

Pour tous les pixels (i,j) **Faire**

Début

```

SI  (  $\mathbf{Im}_{a_{b_2}}[i-1, j]$ 
OU  (  $\mathbf{Im}_{a_{b_2}}[i-1, j+1]$ 
OU  (  $\mathbf{Im}_{a_{b_2}}[i, j+1]$ 
OU(  NON(  $\mathbf{Im}_{a_{b_2}}[i, j]$ 
      ET  (  $\mathbf{Im}_{a_{b_2}}[i, j-1]$ 
      ET  (  $\mathbf{Im}_{a_{b_2}}[i+1, j-1]$ 
      ET  (  $\mathbf{Im}_{a_{b_2}}[i+1, j]$ )))
ALORS   $\mathbf{Im}_{a_{b_1}}[i, j] \leftarrow \mathbf{Im}_{a_{b_2}}[i, j]$ 
SINON
Début
     $\mathbf{Im}_{a_{b_1}}[i, j] \leftarrow \text{FAUX};$ 
    Stable  $\leftarrow \text{FAUX};$ 
Fin
    
```

Fin

Fin

Pour tous les pixels (i,j) **Faire**

Début

Si ($\mathbf{ima}_{b_1}[i, j]$) **Alors** $\mathbf{ima}[i, j] \leftarrow 255$ (attribution de la couleur blanche au pixel positionné en (i,j))

Sinon $\mathbf{ima}[i, j] \leftarrow 0$ (attribution de la couleur noire au pixel positionné en (i,j))

Fin

Afficher le tableau image ima contenant le résultat de la squelettisation.

La figure 5.7 présente un exemple de squelettisation appliquée sur l'image binaire de la figure 5.2.

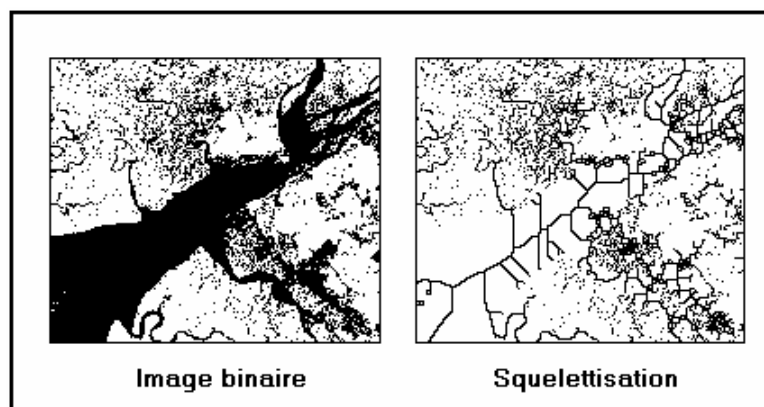


Figure 5.9 Squelettisation d'une portion d'image ERS1 (183 x 159 pixels)
De la région de Douala (source : logiciel **VOIR**)

Chapitre 6. L'analyse de texture

6.1 Introduction

Des objets distincts ayant des propriétés spectrales semblables ou identiques peuvent induire beaucoup de confusions entre les classes lorsqu'on effectue des classifications se basant uniquement sur l'information spectrale contenue dans les images. C'est pour cette raison que des méthodes de classification prenant en compte l'information spatiale ont été développées. L'analyse de texture est une méthode qui tient compte de la distribution spatiale des niveaux de gris autour d'un pixel considéré de l'image. La méthode d'analyse de texture la plus classique est celle de la matrice de cooccurrences. L'approche de spectre de texture est semblable à celle de la matrice de cooccurrence mais elle corrige quelques limitations de la matrice de cooccurrence.

6.2 Les méthodes statistiques du premier ordre

Ces méthodes sont basées sur les histogrammes de premier ordre. Un tel histogramme indique la fréquence d'apparition d'un niveau de gris dans un voisinage considéré. A partir de l'histogramme on peut extraire des paramètres avec des statistiques de degrés différents. Voici quelques uns de ces paramètres :

Moyenne :
$$S_M = \sum_{i=0}^{L-1} i \cdot P(i) \quad (1)$$

Ce paramètre représente l'emplacement de l'histogramme sur l'échelle des niveaux de gris.

Entropie :
$$\text{Ent} = - \sum_{i=0}^{L-1} P(i) \cdot \log_2 [P(i)] \quad (2)$$

L'entropie mesure la complexité de l'image.

Ecart-type :
$$S_D = \left[\sum_{i=0}^{L-1} (i - S_M)^2 \right]^{1/2} \quad (3)$$

L'écart-type est une mesure de la répartition des niveaux de gris autour de la moyenne.

Dissymétrie (Skewness) :
$$\text{Skw} = \frac{1}{S_D^3} \cdot \left[\sum_{i=0}^{L-1} (i - S_M)^3 \cdot P(i) \right] \quad (4)$$

Ce paramètre mesure la dissimilarité de l'histogramme (s'il est bien équilibré autour de sa moyenne ou bien s'il est plus orienté à gauche ou à droite par rapport à sa moyenne).

Le paramètre "Kurtosis" :
$$\text{Kurt} = \frac{1}{S_D^4} \cdot \left[\sum_{i=0}^{L-1} (i - S_M)^4 \cdot P(i) \right] \quad (5)$$

Le paramètre "Kurtosis" donne une indication sur l'aspect pointu et effilé ou plutôt aplati de l'histogramme.

Avec $P(i) = N(i)/M$, où M est le nombre de pixels de la fenêtre d'image F et $N(i)$ le nombre de pixels ayant le même niveau de gris "i" dans F . L est le nombre maximal de niveaux de gris possibles. Illustrons le calcul du paramètre moyenne, par exemple, avec une image numérique simple. Considérons l'image numérique de la Figure 6.1.

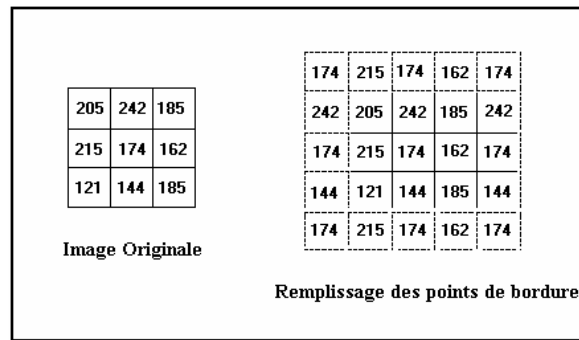


Figure 6.1. Image numérique et remplissage des points de bordure.

Calculons les valeurs $P(i)$ pour tous les niveaux de gris de l'image en utilisant une fenêtre de taille 3x3 et la méthode de symétrie pour les points de bordure. Considérons une fenêtre de taille 3x3 centrée sur le pixel de niveau de gris 205. Cette fenêtre contient les valeurs suivantes :

$$W_{205} = \begin{bmatrix} 174 & 215 & 174 \\ 242 & 205 & 242 \\ 174 & 215 & 174 \end{bmatrix}$$

Nous avons : $P(174)=N(174)/M=4/9$; $P(215)=N(215)/M=2/9$;
 $P(242)=N(242)/M=2/9$; $P(205) = N(205)/M = 1/9$;

La valeur du paramètre **moyenne** au point de niveau de gris 205 est donnée par :

$$S_M(205) = ENT[(174 \times 4/9) + (215 \times 2/9) + (242 \times 2/9) + (205 \times 1/9)] = ENT[201.66] = 201$$

Où ENT désigne la partie entière. On procède de même pour tous les autres pixels et on obtient les résultats ci-après.

$S_M(242) = 192$ $S_M(185) = 187$ $S_M(174) = 181$ $S_M(162) = 183$
 $S_M(144) = 172$ $S_M(185) = 165$ $S_M(215) = 184$ $S_M(121) = 170$

L'image originale et l'image de texture sont présentées sur la Figure 6.3.

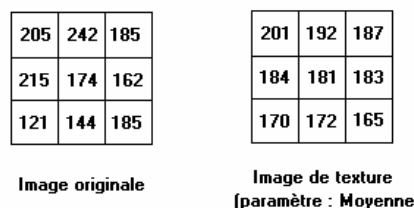


Figure 6.2. Exemple d'image de texture calculée avec le paramètre "moyenne"

6.3 Les méthodes statistiques du second ordre

Ces méthodes sont basées sur l'histogramme de deuxième ordre des niveaux de gris, communément appelé matrice de cooccurrence de deuxième ordre. La matrice de cooccurrence exprime la probabilité d'apparition d'un couple de niveaux de gris (i,j) de l'image dans une fenêtre et dans une direction données. Nous présentons ici quelques uns des paramètres les plus utilisés.

$$\text{Energie ou second moment angulaire : } E = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} [P(i, j, d, \theta)]^2 \quad (6.6)$$

$$\text{Entropie : } \text{Ent} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i, j, d, \theta) \cdot \log_2 (P(i, j, d, \theta)) \quad (6.7)$$

$$\text{Contraste (ou inertie) : } \text{Cont} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i - j)^2 P(i, j, d, \theta) \quad (6.8)$$

$$\text{Corrélation : } \text{Corr} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} [(i - \mu_x)(j - \mu_y)P(i, j, d, \theta)] / (\sigma_x \sigma_y) \quad (6.9)$$

$$\text{Le paramètre "Cluster prominence" : } C_{pr} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} [(i + j - \mu_x - \mu_y)^4 P(i, j, d, \theta)] \quad (6.10)$$

Dans ces expressions, $P(i, j, d, \theta)$ est la probabilité d'apparition du couple de niveaux de gris (i,j) dans une direction θ , d étant la distance spatiale entre les pixels. μ_x et σ_x représentent respectivement la moyenne et l'écart-type des lignes de la matrice de cooccurrence. μ_y et σ_y représentent respectivement la moyenne et l'écart-type des colonnes de la matrice de cooccurrence.

L'**énergie** mesure l'homogénéité de l'image. Elle a une valeur numérique faible quand les $p(i,j)$ ont des valeurs très proches et une valeur forte quand certaines valeurs sont grandes et d'autres petites, par exemple quand les $p(i,j)$ sont concentrées autour de la diagonale. L'**inertie** est une mesure du contraste (ou de la variation locale des niveaux de gris) dans l'image. Ce paramètre a une valeur numérique importante si les $p(i,j)$ sont concentrées hors diagonale. Si au contraire, les valeurs de la matrice de cooccurrence sont centrées sur la diagonale, le contraste est faible. La **corrélation** a une forte valeur quand les valeurs sont uniformément distribuées dans la matrice de cooccurrence, et une faible valeur dans le cas contraire. Quant à l'**entropie**, elle est grande quand les valeurs de la matrice de cooccurrence sont presque toutes égales et elle est faible dans le cas contraire.

Les algorithmes de calcul de la matrice de cooccurrence dans les quatre directions : 0° , 45° , 90° et 135° sont les suivants.

Sous-Programme MATCOOC-0°

Variables d'entrée :
 W : Masque centré autour d'un pixel de l'image (W est Une matrice de taille Tx par Ty) ;
 L : Nombre maximal de niveaux de gris de l'image ;

Variable de sortie :
 MCOOC : Matrice de cooccurrence (C'est un tableau Bidimensionnel de réels) ;

Variables locales :
 i, j, x, y : Entiers ;
 MCOOCs : Tableau de type matrice de cooccurrence ;

Début
Pour j variant de 0 à L-1 **Faire**
 Pour i variant de 0 à L-1 **Faire**
 MCOOC[i,j] ← 0 ; //Initialisation
Pour y variant de 0 à Ty-1 **Faire**
 Pour x variant de 0 à Tx-2 **Faire**
 Début
 I ← W[x,y]
 J ← W[x+1,y]
 MCOOC[I,J] ← MCOOC[I,J] + 1 ;
 Fin
 MCOOCs ← Symétrique de MCOOC ;
Pour j variant de 0 à L-1 **Faire**
 Pour i variant de 0 à L-1 **Faire**
 MCOOC[i,j] ← ((1/2)*(MCOOC[i,j]+MCOOCs[i,j]))/(2*Ty*(Tx-1));

Sous-Programme MATCOOC-45°

Variables d'entrée :
 W : Masque centré autour d'un pixel de l'image (W est Une matrice de taille Tx par Ty) ;
 L : Nombre maximal de niveaux de gris de l'image ;

Variable de sortie :
 MCOOC : Matrice de cooccurrence (C'est un tableau Bidimensionnel de réels) ;

Variables locales :
 i, j, x, y : Entiers ;
 MCOOCs : Tableau de type matrice de cooccurrence ;

Début
Pour j variant de 0 à L-1 **Faire**
 Pour i variant de 0 à L-1 **Faire**
 MCOOC[i,j] ← 0 ; //Initialisation
Pour y variant de 0 à Ty-1 **Faire**
 Pour x variant de 0 à Tx-2 **Faire**
 Début
 I ← W[x,y]
 J ← W[x+1,y-1]
 MCOOC[I,J] ← MCOOC[I,J] + 1 ;
 Fin
 MCOOCs ← Symétrique de MCOOC ;
Pour j variant de 0 à L-1 **Faire**
 Pour i variant de 0 à L-1 **Faire**
 MCOOC[i,j] ← ((1/2)*(MCOOC[i,j]+MCOOCs[i,j]))/(2*(Tx-1)*(Ty-1));

De façon similaire, on écrit les algorithmes de calcul de la matrice de cooccurrence dans les autres directions (90° et 135 °.)

Les étapes d'évaluation d'une image de texture sont décrites dans l'exemple du cas du paramètre *énergie* suivant :

1. **Reéchantillonner** les niveaux de gris de l'image originale à la valeur maximale L souhaitée : *Pour chaque pixel en (x,y) , Faire : $Ima[x,y] \leftarrow ENT[(Ima[x,y]*L)/Max\{Ima\}]$, $Max\{Ima\}$ étant le niveau de gris maximal de l'image originale Ima .*
2. Créer un tableau bidimensionnel : $\mathbf{ima}_f[]$ de réels, ayant les mêmes dimensions que l'image originale Ima .
3. **Pour** chaque pixel (x,y) de l'image originale, **Faire** :
 - a. Centrer un masque \mathbf{W} de taille $T_x * T_y$ en (x,y) . Le masque \mathbf{W} contiendra le niveau de gris du pixel en (x,y) et celui de ses voisins.
 - b. Calculer la matrice de cooccurrence associée à \mathbf{W} , selon la direction choisie.
 - c. Initialiser deux variables réelles \mathbf{ENE} et \mathbf{ENEmax} , toutes à zéro.
 - d. **Pour** j variant de 0 à $L-1$ **Faire**
 - Pour** i variant de 0 à $L-1$ **Faire**

$$\mathbf{ENE} \leftarrow \mathbf{ENE} + (\mathbf{MCOOC}[i,j] * \mathbf{MCOOC}[i,j])$$
 - e. $\mathbf{ima}_f[x,y] \leftarrow \mathbf{ENE}$;
 - f. **Si** $(\mathbf{ENE} \geq \mathbf{ENEmax})$ **Alors** $\mathbf{ENEmax} \leftarrow \mathbf{ENE}$;
4. **Pour** chaque pixel en (x,y) **Faire** :

$$\mathbf{Ima}[x,y] \leftarrow ENT((255 * \mathbf{ima}_f[x,y]) / \mathbf{ENEmax}).$$

\mathbf{ENT} représente la fonction partie entière, et le tableau \mathbf{ima} contient maintenant l'image de texture.

Il est possible de considérer toutes les directions à la fois, soit en prenant la moyenne des valeurs des matrices de cooccurrence dans les différentes directions, soit en prenant le maximum des directions.

6.3 La méthode des longueurs de plages

La méthode des longueurs de plage considère un voisinage monodimensionnel de type auto-adaptatif : un point appartient au voisinage du point courant si son niveau de gris est peu différent. La taille du voisinage est donc très variable, et elle constitue un paramètre caractéristique majeur dans cette approche.

La méthode des longueurs de plage de niveau de gris consiste à compter le nombre de plages de niveau de gris d'une certaine longueur, dans une certaine direction θ . Une matrice $R(\theta)$ de longueurs de plage est associée à une direction d'angle θ . Nous présentons ici quelques paramètres extraits de ces matrices et caractérisant la texture :

Le poids des petites plages de faible niveau de gris :
$$\text{SRLGE} = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{r(i, j)}{i^2 j^2} \quad (6.12)$$

Le poids des petites plages de fort niveau de gris :
$$\text{SRHGE} = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{i^2 r(i, j)}{j^2} \quad (6.13)$$

Le poids des longues plages de faible niveau de gris :
$$\text{LRLGE} = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{j^2 r(i, j)}{i^2} \quad (6.14)$$

Le poids des longues plages de fort niveau de gris :
$$\text{LRHGE} = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N i^2 j^2 r(i, j) \quad (6.15)$$

La constante $nr = \sum_{i=1}^M \sum_{j=1}^N r(i, j)$ est le nombre total de plages dans une fenêtre centrée sur le pixel courant. M est le nombre de niveaux de gris dans l'image et N est le nombre de longueurs de plages rencontrées dans la fenêtre. Pour obtenir l'image de texture, on remplace le niveau de gris du pixel courant par la valeur du paramètre de texture évalué dans une fenêtre centrée sur ce pixel. Pour une direction donnée, la matrice de longueurs de plages se calcule de la façon suivante :

Sous-Programme R0°
Variables d'entrée :
 W : Masque de taille $T_m * T_m$;
 I, J : Entiers ;
Variable de sortie :
 R0 : Nombre de plages (dans la direction 0°) de niveau de gris I et de longueur J dans la fenêtre W
Variables locales :
 C1, C2, x, y, jj : Entiers ;
Début
 C1 ← 0 ;
Pour y variant de 0 à T_m-1 **Faire**
 Début
 x ← 0 ;
 Tant Que (x ≤ T_m-J) **Faire**
 Début
 C2 ← 0 ;
 Pour jj variant de x à $x+J-1$ **Faire**
 Si (W[jj,y] = I) **Alors** C2 ← C2+1 ;
 Si (C2 = J) **Alors**
 Début
 C1 ← C1+1 ;
 x ← x + J ;
 Fin
 Sinon x ← x + 1 ;
 Fin
 Fin
 Fin
 R0 ← C1 ;
Fin du Sous-Programme.

Sous-Programme R45a
Variables d'entrée :
W : Masque de taille $T_m * T_m$;
I, J : Entiers ;
Variable de sortie :
R45a : Nombre de plages (dans la direction 45°) de niveau de gris I et de longueur J dans la fenêtre **W**.
Variables locales :
C1, C2, x, y, ii : Entiers ;
Début
C1 \leftarrow 0 ;
Pour y variant de J-1 à T_m-1 **Faire**
Début
x \leftarrow 0 ;
C2 \leftarrow 0 ;
Tant Que (x \leq y-J+1) **Et** (x \leq T_m-1) **Faire**
Début
Pour ii variant de x à x+J-1 **Faire**
Si (W[ii,y-ii] = I) **Alors** C2 \leftarrow C2+1;
Si (C2 = J) **Alors**
Début
C1 \leftarrow C1+1;
x \leftarrow x + J;
Fin
Sinon x \leftarrow x + 1;
Fin
Fin
R45a \leftarrow C1 ;
Fin du Sous-Programme.

Sous-Programme R45b
Variables d'entrée :
W : Masque de taille $T_m * T_m$
I, J : Entiers ;
Variable de sortie :
R45b : Nombre de plages (dans la direction 45°) de niveau de gris I et de longueur J dans la fenêtre **W**
Variables locales :
C1, C2, x, y, ii : Entiers ;
Début
C1 \leftarrow 0 ;
Pour y variant de J-1 à T_m-2 **Faire**
Début
x \leftarrow 0 ;
C2 \leftarrow 0 ;
Tant Que (x \leq y-J+1) **Et** (x \leq T_m-1) **Faire**
Début
Pour ii variant de x à x+J-1 **Faire**
Si (W[ii,y-ii] = I) **Alors** C2 \leftarrow C2+1;
Si (C2 = J) **Alors**
Début
C1 \leftarrow C1+1;
x \leftarrow x + J;
Fin
Sinon x \leftarrow x + 1;
Fin
Fin
R45b \leftarrow C1 ;
Fin du Sous-Programme.

Sous-Programme R45
Variables d'entrée :
 W : Masque de taille T_m par T_m ;
 I, J : Entiers ;
Variable de sortie :
 R45 : Nombre de plages (dans la direction 45°) de niveau de gris I et de longueur J dans la fenêtre W de taille $T_m \times T_m$
Variables locales :
 C1, C2 : Entiers ;
 Ws : Matrice de même type que W ;
Effectuer les opérations suivantes :
 a. $C1 \leftarrow R45a(I, J, W, T_m)$;
 b. Ws \leftarrow Symétrique de W ;
 c. $C2 \leftarrow R45b(I, J, Ws, T_m)$;
 d. $R45 \leftarrow C1 + C2$;
Fin du Sous-Programme.

Figure 6.3. Algorithme de calcul de la matrice des longueurs de plages dans la direction 45° dans une fenêtre centrée sur un pixel de l'image originale.

Sous-Programme nR0
Variables d'entrée :
 W : Masque de taille $T_m * T_m$;
 L : Entier (nombre maximal de niveaux de gris de l'image);
Variable de sortie :
 nR0 : Nombre total de plages (dans la direction 0°) dans la fenêtre W de taille $T_m * T_m$ (nR0 est une variable entière).
Variables locales :
 Som, i, j : Entiers ;
Début
 Som \leftarrow 0 ;
Pour i variant de 0 à L-1 **Faire**
Pour j variant de 1 à T_m-1 **Faire** Som \leftarrow Som + R0(i, j, W, T_m) ;
 nR0 \leftarrow Som ;
Fin du Sous-Programme.

Figure 6.4. Algorithme de calcul du nombre total de plages dans une fenêtre centrée sur un pixel de l'image originale, dans la direction 0° .

Sous-Programme nR45
Variables d'entrée :
 W : Masque de taille $T_m * T_m$;
 L : Entier (nombre maximal de niveaux de gris de l'image);
Variable de sortie :
 NR45 : Nombre total de plages (dans la direction 45°) dans la fenêtre W de taille $T_m \times T_m$ (nR45 est une variable entière).
Variables locales :
 Som, i, j : Entiers ;
Début
 Som \leftarrow 0 ;
Pour i variant de 0 à L-1 **Faire**
Pour j variant de 1 à T_m-1 **Faire** Som \leftarrow Som + R45(i, j, W, T_m) ;
 nR45 \leftarrow Som ;
Fin du Sous-Programme.

Figure 6.5. Algorithme de calcul du nombre total de plages dans une fenêtre centrée sur un pixel de l'image originale, dans la direction 45° .

Une fois que la matrice des longueurs de plages est calculée, il devient aisé de calculer les différents paramètres de texture. L'algorithme d'obtention d'une image de texture, avec le paramètre « **poils des petites plages de faible niveau de gris : SRLGE (Short Run Low Gray level Emphasis)** » se présente de la façon suivante :

1. Rééchantillonner les niveaux de gris de l'image originale à la valeur maximale L souhaitée ;
2. Créer un tableau bidimensionnel $\mathbf{ima}_f[]$ de réels, ayant les mêmes dimensions que l'image originale \mathbf{Ima} ;
3. Initialiser une variable réelle $\mathbf{aSRLGEmax}$ à zéro.
4. **Pour** chaque pixel (x,y) de l'image originale **Faire** :
 - a. Centrer le masque \mathbf{W} de taille $T_m * T_m$ en (x,y) . Le masque \mathbf{W} contiendra le niveau de gris du pixel en (x,y) et celui de ses voisins.
 - b. Calculer la matrice des longueurs de plages associée à \mathbf{W} , selon la direction choisie.
 - c. Initialiser une variable réelle \mathbf{aSRLGE} à zéro.
 - d. **Pour** i variant de 0 à $L-1$ **Faire**
 - Pour** j variant de 1 à T_m-1 **Faire**
 - a. $\mathbf{aSRLGE} \leftarrow \mathbf{aSRLGE} + (\mathbf{R}\emptyset(i, j, \mathbf{W}, T_m) / [(i + 1)^2 \times (j + 1)^2])$;
 - b. // \emptyset représente la direction choisie ($\emptyset = 0, 45, 90$ ou 135).
 - e. $\mathbf{aSRLGE} \leftarrow \mathbf{aSRLGE} / \mathbf{nR}\emptyset(\mathbf{W}, T_m, L)$;
 - f. $\mathbf{ima}_f[x,y] \leftarrow \mathbf{aSRLGE}$;
 - g. SI $(\mathbf{aSRLGE} \geq \mathbf{aSRLGEmax})$ ALORS $\mathbf{aSRLGEmax} \leftarrow \mathbf{aSRLGE}$;
5. **Pour** tous les pixels (x,y) modifier les valeurs du tableau \mathbf{ima} selon :

$$\mathbf{ima}[x,y] \leftarrow \text{ENT}((255 * \mathbf{ima}_f[x,y]) / \mathbf{aSRLGEmax}).$$

ENT représente la fonction partie entière, et le tableau \mathbf{ima} contient maintenant l'image de texture.

Il est possible de considérer toutes les directions à la fois, soit en prenant la moyenne des valeurs des matrices des longueurs de plages dans les différentes directions, soit en prenant le maximum des directions.

Chapitre 7. La classification des images

7.1 Introduction

Les données de télédétection peuvent être analysées pour extraire des informations thématiques utiles. La classification multispectrale est l'une des méthodes les plus souvent utilisées pour l'extraction de ces informations. Cette procédure suppose que les images d'une région géographique donnée sont réparties dans différentes régions du spectre électromagnétique. La classification multispectrale peut être effectuée en utilisant une variété d'algorithmes comprenant :

- 1) les méthodes de classification statistique ou neuronale, supervisées ou non supervisées;
- 2) les méthodes de classification floue;
- 3) les systèmes experts ou les méthodes à base de connaissances;
- 4) les méthodes de classification hybrides combinant les méthodes précédentes ou utilisant parfois des données auxiliaires.

Dans une classification supervisée, l'identité et la localisation de certains types de couverture du sol (par exemple : sol nu, broussailles arbustives, forêt, cultures) sont connues a priori à travers la combinaison des observations sur le terrain, de l'analyse de photographies aériennes ou de cartes. L'analyste tente de localiser sur l'image des sites spécifiques représentant des régions homogènes de ces types de couverture du sol connus. Ces régions homogènes sont communément appelées des sites d'entraînement, car les caractéristiques spectrales ou texturales de ces régions sont utilisées pour entraîner le classifieur que l'on utilisera pour classifier le reste de l'image. Des paramètres statistiques multivariés (moyenne, écart-type, matrices de covariance, matrices de corrélation, etc.) sont calculés pour chaque site d'entraînement. Chaque pixel se trouvant à l'intérieur ou à l'extérieur des sites d'entraînement est ensuite examiné et affecté à la classe avec laquelle il a la plus grande vraisemblance d'appartenance.

Dans une classification non supervisée, l'identité des types de couverture du sol devant constituer les classes d'information n'est pas généralement connue a priori. Cette situation résulte souvent du manque d'informations de référence sur le terrain, ou de l'incertitude quant à la définition des formes de surface à l'intérieur de la scène analysée. L'analyste procède ensuite à une interprétation de ces regroupements pour produire des classes d'information.

La logique de classification floue, qui prend en compte la nature hétérogène et imprécise des données géographiques, peut être utilisée en coopération avec des algorithmes de

classification supervisée ou non supervisée. La plupart des algorithmes de classification utilisent la théorie booléenne des ensembles pour produire des classes d'information homogènes, ignorant complètement l'imprécision naturelle des données géographiques. Plutôt que d'affecter chaque pixel de l'image à une seule des m classes d'entraînement, un algorithme de classification floue calcule m nombres réels indiquant les degrés d'appartenance de chaque pixel aux différentes classes d'intérêt. Cette information peut ensuite être utilisée par l'analyste pour extraire des informations plus précises sur la couverture du sol, telle que la composition des pixels hétérogènes.

Il est parfois nécessaire d'inclure des données auxiliaires non spectrales dans une classification supervisée ou non supervisée et floue pour extraire les informations voulues. De nombreuses méthodes existent, comprenant la stratification géographique, la classification hiérarchique et les systèmes experts ou à base de connaissance.

7.2 Elaboration d'un protocole de classification

Une bonne classification supervisée ou non supervisée des données de télédétection peut être obtenue si les étapes générales sont comprises et attentivement suivies.

1. L'analyste sélectionne d'abord une région d'intérêt appropriée sur laquelle les hypothèses seront testées. Les classes d'intérêt devant être testées dans l'hypothèse vont dicter la nature de la méthode de classification à utiliser.
2. L'analyste sélectionne ensuite des images de télédétection appropriées, en gardant à l'esprit les contraintes environnementales et celles du système d'acquisition des données. Quand les données sont finalement disponibles, elles sont habituellement corrigées radiométriquement et/ou géométriquement.
3. Un algorithme de classification approprié est ensuite choisi et les données d'entraînement initiales sont collectées.
4. Une réduction de paramètres est souvent effectuée dans le but d'extraire de nouveaux paramètres qui discriminent mieux les classes en présence.
5. Dans le cas d'une classification supervisée, les paramètres retenus à l'étape précédente sont alors utilisés pour entraîner le système de classification.
6. Des données d'entraînement additionnelles sont ensuite collectées pour évaluer les performances du classificateur. Si ses performances sont jugées acceptables, le classificateur est appliqué à l'image entière pour produire une carte de classification.

7. Une évaluation rigoureuse de l'erreur de classification est ensuite effectuée. Si les résultats sont acceptables, les cartes de classification et les statistiques associées sont communiquées aux utilisateurs.

7.2.1 Elaboration d'un système de classification

Il est nécessaire pour l'analyste de réaliser qu'il y a une différence fondamentale entre les classes d'information qu'il recherche sur le terrain et les classes spectrales observables sur les images.

Les classes d'information sont les catégories d'intérêt pour les utilisateurs des données. Elles sont, par exemple, les différents types d'unités géologiques, de forêt ou d'utilisation du sol qui apportent de l'information aux planificateurs, gestionnaires, administrateurs et aux scientifiques qui utilisent les informations dérivées des données de télédétection. Ces classes constituent l'information que nous voulons extraire de ces données. Elles sont la finalité du processus d'analyse.

Les classes spectrales sont des groupes de pixels ayant des caractéristiques spectrales semblables et directement observables sur l'image. S'il est possible de définir des liens entre les classes spectrales observables sur l'image et les classes d'information recherchées, alors l'image constitue une bonne source d'information. Ainsi, la classification par télédétection procède par une mise en correspondance entre les classes spectrales observables sur l'image et les classes d'information recherchées sur le terrain. Si cette mise en correspondance peut être effectuée de façon précise, alors la carte de classification résultante sera fiable. Si les classes spectrales et les classes d'information ne correspondent pas assez, l'image ne constitue pas une bonne source d'information pour cette forme particulière d'information. Chaque classe d'information comprend en effet des variations spectrales dues aux variations naturelles intra classe. Par exemple, une région de la classe d'information "forêt" reste forêt même si elle présente des variations en âge des arbres et en composition des espèces. Cependant, ces variations conduisent à des différences dans l'apparence spectrale de la classe d'information.

En conséquence, les classes d'information sont typiquement composées de groupes de pixels spectralement distincts. En classification numérique, on doit souvent traiter les classes spectrales comme des unités distinctes, mais on doit les afficher avec une même étiquette sur la carte thématique finale.

7.2.2 Sélection des sites d'entraînement et extraction des statistiques

Un analyste peut sélectionner sur l'image des sites d'entraînement représentatifs des classes de couverture du sol d'intérêt, après l'adoption d'un système de classification. Les données d'entraînement seront de qualité si l'environnement dans lequel elles sont obtenues est

relativement homogène. Par contre, si les conditions du sol varient à travers la région d'étude, les données d'entraînement collectées sur cette région auront une réponse spectrale non homogène, et on sera confronté à un problème d'extension des signatures spectrales. Dans ce cas, les facteurs environnementaux permettant de résoudre le problème de l'extension des signatures seront identifiés. Parmi ces facteurs on peut citer :

- 1) les différences sur les types de sol;
- 2) la turbidité des eaux;
- 3) les espèces de cultures;
- 4) les conditions d'humidité du sol non habituelles, possiblement causées par une tempête dont les précipitations ne sont pas émises de façon uniforme;
- 5) les plaques de diffusion des gaz atmosphériques, etc.

De telles conditions environnementales seront clairement annotées sur l'image, et la sélection des sites d'entraînement sera effectuée sur la base de la stratification géographique de ces données.

Une fois que les facteurs d'extension des signatures sont considérés, l'analyste sélectionne des sites d'entraînement représentatifs pour chaque classe. Il collecte ensuite les statistiques spectrales de chaque pixel contenu dans chaque site d'entraînement. Si l'analyse effectuée l'entraînement avec n bandes spectrales, alors chaque pixel de chaque site d'entraînement est représenté par un vecteur de mesures X_c tel que :

$$X_c = \begin{pmatrix} BV_{ij1} \\ \cdot \\ \cdot \\ \cdot \\ BV_{ijn} \end{pmatrix}$$

où BV_{ijk} est la valeur de luminance du pixel (i,j) dans la bande k. Les valeurs de luminance de chaque pixel dans chaque bande sont ensuite analysées statistiquement pour produire un vecteur de mesures moyen M_c pour chaque classe :

$$M_c = \begin{pmatrix} \mu_{c1} \\ \cdot \\ \cdot \\ \cdot \\ \mu_{cn} \end{pmatrix}$$

où μ_{ck} représente la valeur moyenne de la donnée obtenue pour la classe c dans la bande k . Les vecteurs de mesure brutes peuvent aussi être analysés pour produire la matrice de covariance pour chaque classe c :

$$V_c = \begin{pmatrix} \text{Cov}_{c11} & \cdot & \cdot & \cdot & \text{Cov}_{cn1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \text{Cov}_{cn1} & \cdot & \cdot & \cdot & \text{Cov}_{cnn} \end{pmatrix}$$

où Cov_{ckl} est la covariance de la classe c entre les bandes k et l .

7.2.3 La réduction des paramètres

La représentation des données d'entrée d'un système de classification statistique ou neuronal est importante et peut affecter significativement les performances du système considéré. On distingue deux approches de réduction des paramètres : l'extraction des paramètres et la sélection des paramètres.

7.2.4 L'extraction des paramètres

L'extraction des paramètres est une opération qui consiste à trouver un ensemble de vecteurs qui représentent une observation en réduisant sa dimensionalité. Bien qu'une réduction de dimensionalité soit désirable, l'erreur qu'elle induit ne doit pas sacrifier le pouvoir de discrimination du classificateur. Les méthodes d'extraction de paramètres peuvent être supervisées ou non supervisées, linéaires ou non linéaires. Comme méthode d'extraction des paramètres, on peut citer : l'analyse en composantes principales, de l'analyse discriminante et de l'extraction des paramètres à partir des frontières de décision.

Pour toutes ces méthodes, une matrice des caractéristiques est définie. Les valeurs propres de la matrice et les vecteurs propres correspondants sont calculés et classés par ordre de grandeur décroissante. La dimension des données d'entrée correspond au nombre de vecteurs propres retenus. Les données transformées sont déterminés par $Y = \phi^T X$, où ϕ est la matrice de transformation composée des vecteurs propres de la matrice des caractéristiques, X est le vecteur de données dans l'espace des caractéristiques original et Y est le vecteur de données transformé dans le nouvel espace des caractéristiques.

7.3 Les algorithmes de classification supervisée

Ce type de classification est utilisé pour convertir les données de télédétection en une carte thématique et il comprend trois étapes en général.

1. l'étape d'entraînement : des pixels de classe connue sont identifiés et caractérisés pour former les statistiques d'entraînement servant à décrire les classes.

2. l'étape d'allocation des classes : chaque pixel de l'image est affecté à la classe avec laquelle il a la plus grande vraisemblance.

3. l'étape de vérification : la qualité de la classification est évaluée.

Parmi les algorithmes de classification supervisée les plus utilisés on peut citer :

- la règle de décision des parallélépipèdes;
- la règle de décision de la distance minimale;
- la règle de décision des k plus proches voisins;
- la règle du maximum de vraisemblance.

A toutes ces méthodes classiques nous devons ajouter les méthodes de classification basées sur les réseaux de neurones artificiels, qui sont devenus depuis quelques années une alternative importante des méthodes paramétriques conventionnelles.

7.3.1 La classification par la méthode du parallélépipède

Cette méthode est largement utilisée et elle est basée sur une logique booléenne. Des données d'entraînement dans n bandes spectrales sont utilisées pour effectuer la classification. Pour

chaque classe c on calcule le vecteur $M_c = \begin{pmatrix} \mu_{c1} \\ \cdot \\ \cdot \\ \cdot \\ \mu_{cn} \end{pmatrix}$ où μ_{ck} représente le niveau de gris moyen

des pixels de la classe c dans la bande k. s_{ck} est l'écart type des données d'entraînement de la classe c dans la bande k. Si on utilise un seuil égal à s_{ck} par exemple, l'algorithme du

parallélépipède affecte un pixel de vecteur de mesures $X_c = \begin{pmatrix} BV_{ij1} \\ \cdot \\ \cdot \\ \cdot \\ BV_{ijn} \end{pmatrix}$ à la classe c, si et

seulement si l'équation ci-après est satisfaite.

$$\mu_{ck} - s_{ck} \leq BV_{ijk} \leq \mu_{ck} + s_{ck}, \quad \forall k \tag{1}$$

où $c = 1, \dots, m$ représente le nombre de classes, et $k = 1, \dots, n$ représente le nombre de bandes.

Par conséquent, si les frontières de décision inférieure et supérieure sont définies par

$L_{ck} = \mu_{ck} - s_{ck}$ et $H_{ck} = \mu_{ck} + s_{ck}$ respectivement, la règle de décision du parallélépipède devient celle qui est exprimée par la relation ci-après.

$$L_{ck} \leq BV_{ijk} \leq H_{ck}, \quad \forall k \Rightarrow BV_{ij} \in w_c \quad (2)$$

Ces règles de décision forment un parallélépipède multidimensionnel dans l'espace des caractéristiques. Si la valeur du pixel est comprise entre la borne inférieure et la borne supérieure pour toutes les n bandes évaluées pour une classe donnée, il est affecté à cette classe, et dans le cas contraire il est affecté à une catégorie non classifiée.

7.3.2 La classification par la méthode de la distance minimale

Cette méthode est simple et communément utilisée. Quand elle est correctement utilisée, elle peut produire des classifications comparables à celles produites par d'autres algorithmes plus coûteux en calculs, comme l'algorithme du maximum de vraisemblance. A l'instar de l'algorithme du parallélépipède, la méthode de la distance minimale nécessite que l'utilisateur fournisse les vecteurs moyens M_c de toutes les classes à partir des données d'apprentissage. Pour effectuer une classification par distance minimale, un programme doit calculer la distance par rapport à chaque vecteur moyen M_c , pour chaque pixel inconnu (BV_{ijk}).

$$D_{cij} = \sqrt{\sum_{k=1}^n (BV_{ijk} - \mu_{ck})^2} \quad (3)$$

La règle de décision de la distance minimale aux moyennes s'exprime alors par la relation

$$D_{cij} = \min_k \{D_{kij}\} \Rightarrow X_{ij} \in w_c \quad (4)$$

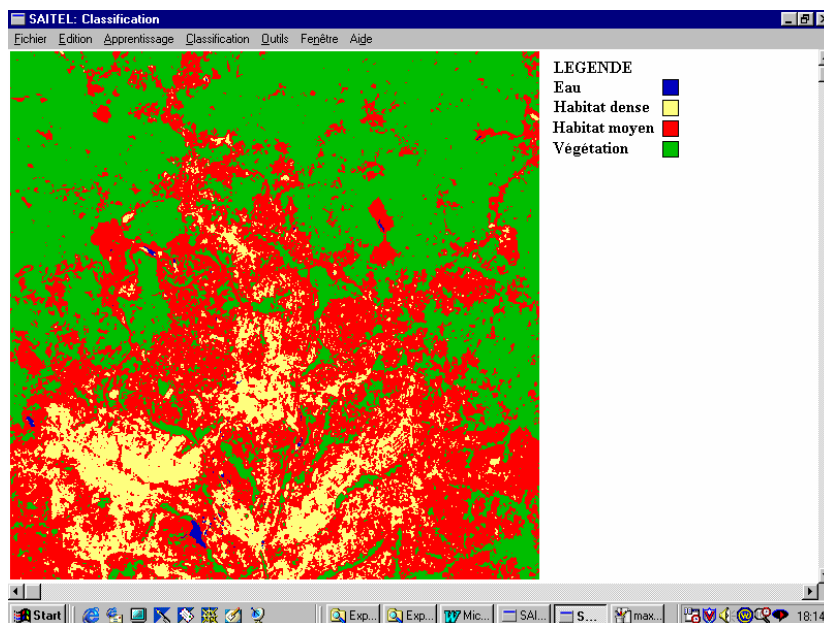


Figure 1. Exemple de classification d'une image SPOT de Yaoundé par la méthode de la distance minimale (source : logiciel SAITEL).

7.3.3 La classification par la méthode des K plus proches voisins

Cette règle de décision est simple comme les précédentes et elle est communément utilisée pour la classification des données de télédétection. Quand elle est correctement utilisée, elle peut produire des résultats comparables à ceux d'autres algorithmes classiques, comme l'algorithme du maximum de vraisemblance. L'algorithme de classification par la méthode des K plus proches voisins nécessite que l'utilisateur fournisse les vecteurs de mesures X_i pour tous les pixels d'entraînement. Pour effectuer une classification par cette méthode, l'algorithme calcule la distance euclidienne entre chaque pixel inconnu (BV_{ijk}) et chaque vecteur de référence X_i , puis il détermine la fréquence d'apparition f_{ck} des pixels de chaque classe c parmi les K plus proches voisins du pixel considéré.

La règle de décision des K plus proches voisins s'exprime alors par :

$$f_{ck} = \max_c \{f_{ck}\} \Rightarrow BV_{ij} \in w_k \quad (5)$$

L'application de cette règle de décision nécessite que la valeur de K soit définie. En pratique, on choisit $K = c\sqrt{n}$ où c est une constante appropriée (typiquement $c = 1$), et n est le nombre de données d'entraînement. Lorsque $K = 1$ cette règle de décision correspond à la règle de classification de la distance minimale.

7.3.4 La classification par maximum de vraisemblance

La règle de décision du maximum de vraisemblance affecte chaque pixel, de vecteur de mesures X , à la classe c dont les unités sont plus probables de produire le vecteur de mesures X . Elle suppose que les statistiques des données d'entraînement de chaque classe, dans chaque bande, sont distribuées selon la loi normale. Par conséquent, les données d'entraînement multimodales dans une bande ne sont pas appropriées. La classification par maximum de vraisemblance utilise les statistiques des classes déjà calculées, comprenant le vecteur moyen μ_c et la matrice de covariance V_c pour chaque classe c . La règle de décision du maximum de vraisemblance affecte le vecteur X à la classe c , si et seulement si la condition illustrée par l'ensemble d'équations ci-après est satisfaite.

$$p_c(X) = \max_j \{p_j(X)\} \Rightarrow X \in w_c \quad (6)$$

$$p_c(X) = \frac{1}{(2\pi)^{n/2} |V_c|^{1/2}} \exp\left\{-\frac{1}{2}(X - \mu_c)^T V_c^{-1}(X - \mu_c)\right\} \quad (7)$$

où $|V_c|$ représente le déterminant de la matrice de covariance V_c .

Par conséquent, pour placer un vecteur de mesures X d'un pixel inconnu dans une classe, la règle de décision du maximum de vraisemblance calcule la valeur de $p_c(X)$ pour chaque classe. Puis, elle affecte le pixel à la classe qui présente la plus grande valeur.

La figure ci-après présente un exemple de classification d'une image SPOT multispectrale de la ville de Yaoundé par la méthode du maximum de vraisemblance.

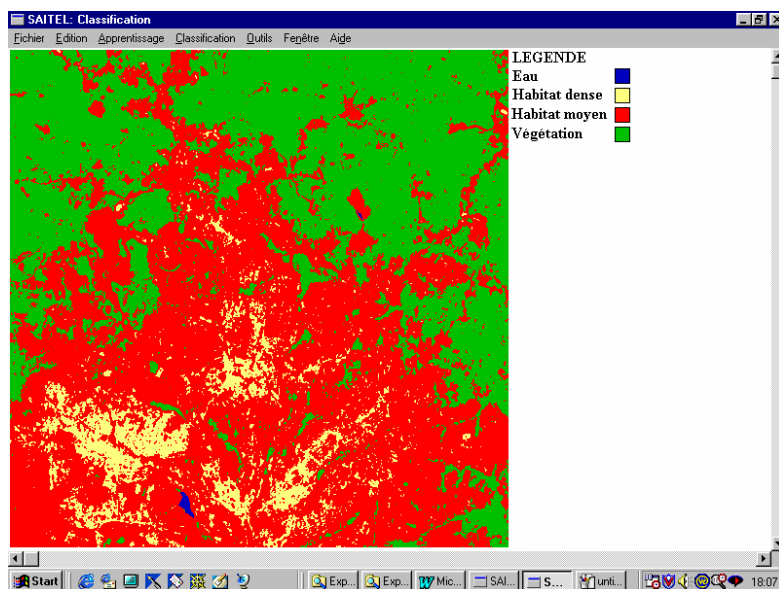


Figure 2. Exemple de classification d'une image SPOT de Yaoundé par la méthode du maximum de vraisemblance (source : logiciel SAITEL).

7.4 Les algorithmes de classification non supervisée

Contrairement à la classification supervisée, la classification non supervisée nécessite seulement une quantité minimale d'entrée initiales de la part de l'analyste. Avec cette méthode, des opérations numériques sont effectuées pour déterminer les regroupements naturels des propriétés spectrales des pixels, dans l'espace multispectral des caractéristiques. Un programme informatique est utilisé pour calculer les statistiques des segments (moyennes et matrices de covariances) qui seront utilisées dans la classification. Une fois que les données sont classifiées, l'analyste tente d'affecter ces classes naturelles ou spectrales aux classes d'information d'intérêt. Cette tâche peut être difficile. Certains segments peuvent être sans signification, parce qu'ils représentent des mélanges de classes des éléments de la surface de la terre. L'analyste devra bien comprendre les caractéristiques spectrales du terrain, pour considérer certains segments comme représentant des classes d'information. Des centaines de méthodes de segmentation ont été développées pour une grande variété d'applications, allant de la reconnaissance de formes à la télédétection. Les algorithmes de segmentation utilisés pour la classification non supervisée des données de télédétection varient généralement

suivant l'efficacité avec laquelle la segmentation s'effectue. Des critères d'efficacité différents conduisent à des approches différentes.

7.4.1 La classification par la méthode des modes de l'histogramme

Un histogramme monodimensionnel d'une image est une représentation graphique ayant en abscisses les valeurs de niveau de gris, et en ordonnées le nombre de pixels associé à chaque valeur de niveau de gris. Le **mode** d'un histogramme est un maximum local de cet histogramme, et une **vallée** est un minimum local. Le mode d'un histogramme indique un regroupement de pixels et il sert à détecter des classes. L'algorithme de classification par mode de l'histogramme monodimensionnel se résume par les étapes suivantes.

- 1) Construire l'histogramme de l'image (tableau $H[i]$);
- 2) Détecter K maximums locaux du tableau $H[i]$, K étant le nombre de classes souhaité;
Les abscisses des K maximums locaux représentent les noyaux ou centres des différentes classes à constituer.
- 3) Regrouper les pixels de l'image selon le critère de distance minimale par rapport aux différents centres de classes. Chaque pixel est placé dans la classe dont le centre lui est le plus proche.
- 4) Attribuer une même couleur aux pixels appartenant à la même classe et afficher l'image résultat.

La ci-après présente un exemple de classification par cette méthode. L'image traitée est une image ESAR de la côte atlantique camerounaise.

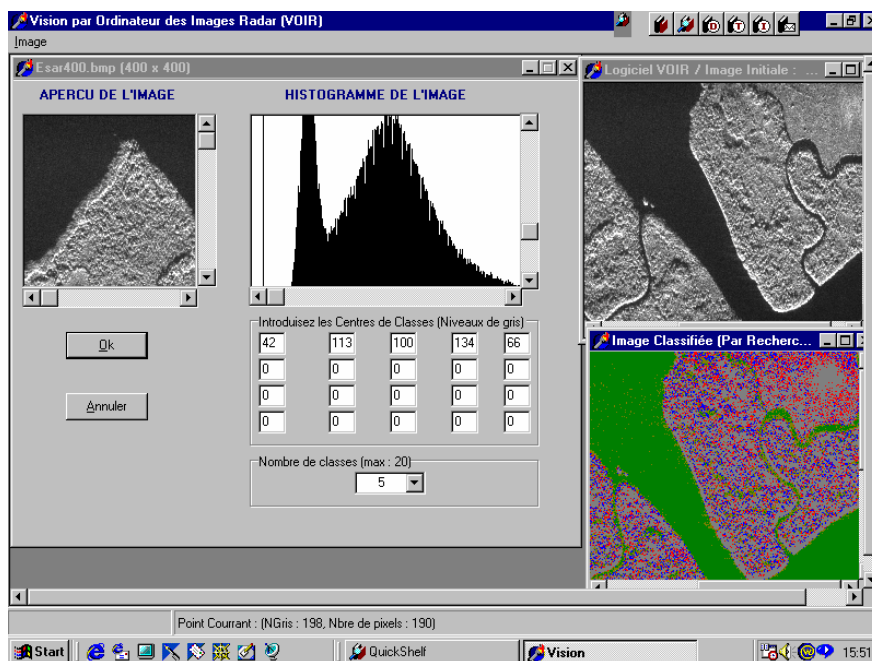


Figure 3 Image classifiée par la méthode des modes de l'histogramme (source : logiciel VOIR)

On peut aussi réaliser une classification multibande en réalisant un histogramme bidimensionnel. Dans ce cas, on compte le nombre de pixels associé à chaque paire de niveaux de gris. Lorsque ceci est fait, on utilise le même algorithme que celui des modes de l'histogramme monodimensionnel.

7.4.2 Classification non supervisée utilisant la méthode ISODATA

La méthode ISODATA (Iterative Self-Organizing Data Analysis Technique) représente un ensemble compréhensible de procédures heuristiques qui ont été incorporées dans un algorithme de classification itératif. La plupart des étapes incorporées dans l'algorithme sont le résultat des expériences accumulées à travers des expérimentations. L'algorithme ISODATA est auto-organisateur parce qu'il nécessite relativement très peu d'entrées de la part de l'analyste. Un algorithme ISODATA sophistiqué requiert normalement la spécification des critères suivants, de la part de l'analyste :

- 1) C_{\max} : le nombre maximum de segments à identifier par l'algorithme. Toutefois, il n'est pas inhabituel que moins de C_{\max} segments soient trouvés dans la carte de classification finale, après que des éclatements et des fusions aient eu lieu.
- 2) T : le pourcentage maximum de pixels dont les valeurs de classe sont autorisées à rester inchangées entre des itérations successives. L'algorithme ISODATA se termine lorsque ce nombre est atteint.
- 3) M : le nombre maximum de fois que l'algorithme ISODATA classe les pixels et recalcule les vecteurs moyens des segments. L'algorithme ISODATA se termine lorsque ce nombre est atteint.
- 4) Le nombre minimum de membres (%) dans un segment. Si un segment contient moins de membres que le minimum spécifié, il est supprimé et ses membres sont affectés à un autre segment.
- 5) L'écart-type maximum. Lorsque l'écart type d'un segment dépasse le maximum spécifié, et que le nombre de membres dans le segment est deux fois supérieur au minimum de membres spécifiés dans une classe, le segment est éclaté en deux segments.
- 6) La distance minimum entre les segments. Des segments ayant une distance pondérée inférieure à cette valeur sont fusionnés.

La figure ci-après présente un exemple de classification d'une image SPOT multispectrale de la ville de Yaoundé par la méthode ISODATA.

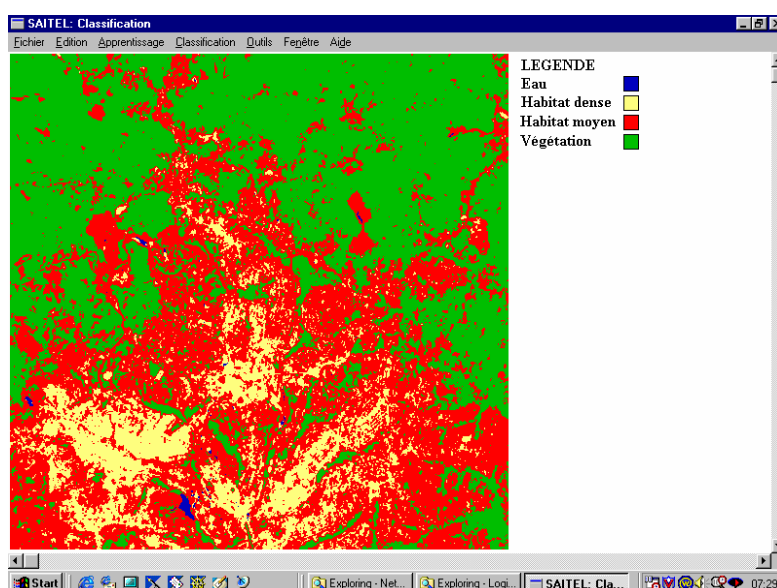


Figure 4. Exemple de classification d'une image SPOT de Yaoundé par la méthode ISODATA (source : logiciel SAITEL).

7.5 Autres méthodes de classification : la classification par réseaux de neurones

Les réseaux de neurones ont été appliqués avec succès dans plusieurs domaines. Une caractéristique des réseaux de neurones est qu'ils peuvent nécessiter de longs temps d'entraînement, mais ils sont des classificateurs relativement rapides. L'utilisation des réseaux de neurones comme outil de traitement des données de télédétection est principalement motivée par les raisons suivantes :

- 1) Ils ont la capacité d'opérer avec une plus grande précision que les autres techniques, à l'exemple des classificateurs statistiques, particulièrement lorsque l'espace des caractéristiques est complexe, et que les sources de données ont des distributions statistiques différentes.
- 2) Ils ont la capacité de traiter plus rapidement des grands ensembles de données, en comparaison aux classificateurs statistiques.
- 3) Ils ont la capacité d'incorporer des connaissances a priori et des contraintes physiques réelles dans le processus d'analyse.
- 4) Ils ont la capacité d'incorporer différents types de données (y compris des données de différents capteurs) dans l'analyse, facilitant ainsi des études de synergie.

Au regard de ces nombreux atouts, il est clair que l'une des principales opportunités offertes par les réseaux de neurones est la possibilité de traiter efficacement les grandes quantités de données de télédétection qui sont actuellement disponibles. Le modèle de réseau de neurones

le plus communément utilisé pour la classification d'images en télédétection est le *perceptron multicouche*, entraîné avec l'algorithme de *rétro-propagation*.

7.6 Evaluation de la précision d'une classification

L'évaluation de la précision d'une classification implique la comparaison de l'image classifiée avec des données de terrain. Cette comparaison est habituellement basée sur une matrice de confusion qui indique les agréments et les désagréments entre les ensembles de données. Des mesures telles que le pourcentage de classification correcte et le coefficient kappa peuvent être dérivées des éléments d'une matrice de confusion et utilisées pour exprimer la précision d'une classification.

7.6.1 Données d'entraînement et données de vérification

L'expression de la précision d'une classification est un élément essentiel pour la classification supervisée des images de télédétection. Typiquement la précision d'une classification est évaluée en comparant la classe d'appartenance trouvée par la procédure de classification avec la classe d'appartenance effective sur le terrain, pour toutes les données de vérification. Des indices quantitatifs exprimant la précision peuvent alors être calculés à partir de la matrice de confusion. Toutefois, la qualité de la classification n'est pas le seul facteur qui influence l'expression de la précision. La nature des données de vérification peut avoir un effet significatif sur l'expression de la précision de la classification. Les données de vérification doivent être représentatives des classes d'entraînement, de même que les données d'entraînement. Par conséquent, ces données doivent être collectées dans des sites de vérification, différents des sites d'entraînement. Les échantillons de données doivent être assez grands pour permettre une évaluation rigoureuse de la précision de la classification.

Il est souhaitable de collecter certaines données de vérification avant la classification, probablement au moment de la collecte des données d'entraînement. Malheureusement, les données de vérification sont en général collectées après la classification. Puisque les paysages changent souvent très rapidement, il est souhaitable de collecter les données de vérification et les données d'entraînement à des dates aussi proches que possible de la date d'acquisition des images.

7.6.2 Taille des échantillons

Le nombre effectif de pixels devant être référencés sur le terrain et utilisés pour évaluer la précision de la classification est souvent difficile à déterminer. Certains analystes utilisent des équations basées sur la distribution binomiale, ou sur l'approximation normale d'une distribution binomiale pour calculer la taille des échantillons requise. Ces techniques sont

statistiquement appropriées pour trouver la taille des échantillons nécessaire pour évaluer la précision globale d'une classification. Les équations utilisées sont basées sur la proportion des données d'entraînement correctement classifiées et sur l'erreur de classification permise. Par exemple, la taille N des données utilisées pour évaluer la précision d'une classification peut être déterminée à partir de l'expression suivante.

$$N = \frac{Z^2 p(100 - p)}{E^2} \quad (8)$$

où Z est l'écart normal au seuil de confiance de 95%. p est le pourcentage de précision attendu et E est l'erreur permise dans l'estimation de la valeur de p. Le choix de E est dans une certaine mesure arbitraire, mais on prend généralement E = 4%.

7.6.3 Méthodes d'échantillonnage

La plupart des mesures robustes d'évaluation statistique d'erreur supposent que les données de référence sont collectées de façon aléatoire. L'échantillonnage simple sans remplacement produit des estimations adéquates des paramètres de la population, à condition que la taille de l'échantillon soit grande. Toutefois, l'échantillonnage aléatoire peut sous-échantillonner de petites classes, à condition que la taille de l'échantillon soit significativement grande. Les techniques d'échantillonnage systématique seront utilisées avec une grande précaution, car elles tendent à surestimer les paramètres de la population. Pour cette raison, les analystes préfèrent l'échantillonnage aléatoire par strates, par lequel un nombre minimum d'échantillons sont sélectionnés de chaque strate (catégorie de couverture du sol). La combinaison de l'échantillonnage aléatoire et par strates produit le meilleur équilibre entre la validité statistique et l'application pratique. Un tel système peut utiliser un échantillonnage aléatoire pour collecter les données de validation au début du projet. L'échantillonnage par strates, quant à lui, sera utilisé après la classification.

7.6.4 Précision globale et coefficient KAPPA

La précision globale s'obtient en divisant le nombre total de pixels bien classifiés (somme des éléments diagonaux de la matrice de confusion) par le nombre total de pixels de vérification. La précision d'une catégorie individuelle s'obtient par division du nombre total de pixels bien classifiés dans cette catégorie par le nombre total de pixels de la catégorie. Cette statistique indique la probabilité qu'un pixel de référence soit correctement classé et elle est une mesure de l'erreur d'omission. Elle est appelée la précision du producteur parce que le producteur de la classification est intéressé par le succès avec lequel une certaine région peut être classifiée. Si le nombre total de pixels bien classifiés dans une catégorie est divisé par le nombre total de pixels qui ont été effectivement classifiés dans cette catégorie, le résultat est une mesure de

l'erreur de commission. Cette mesure, appelée la précision de l'utilisateur ou fiabilité, est la probabilité qu'un pixel classifié dans une catégorie sur la carte de classification représente effectivement cette catégorie sur le terrain.

L'analyse KAPPA est une technique discrète multivariable, utilisée pour évaluer la précision d'une classification. L'analyse KAPPA produit une statistique KHAT (une estimation du KAPPA), qui est une mesure d'accord ou de précision. La statistique KHAT se calcule selon l'expression suivante.

$$\hat{K} = \frac{N \sum_{i=1}^r x_{ii} - \sum_{i=1}^r (x_{i+} \times x_{+i})}{N^2 - \sum_{i=1}^r (x_{i+} \times x_{+i})} \quad (7.22)$$

où r est le nombre de lignes de la matrice de confusion. x_{ii} est le nombre d'observations dans la ligne i et la colonne i . x_{i+} et x_{+i} sont les totaux marginaux de la ligne i et de la colonne i , respectivement, et N est le nombre total d'observations.

La précision globale et la précision du KAPPA ne sont pas en général égales pour une classification donnée. Cette différence est due au fait que la précision globale utilise uniquement les éléments diagonaux de la matrice de confusion, ignorant ainsi les erreurs d'omission et de commission indiquées par les éléments hors diagonaux. Par contre, le KHAT intègre les éléments hors diagonaux en termes de produit des sommes marginales des colonnes et des lignes. Ainsi, selon la quantité d'erreur contenue dans la matrice, ces deux mesures peuvent ne pas correspondre.

Chapitre 8. Analyse d'images de télédétection

8.1 Introduction

Quand les photo-interprètes analysent des images de télédétection, ils prennent en compte la synergie entre le contexte, les contours, la texture et la variation des niveaux de gris. Or la plupart des algorithmes de classification numérique actuels exploitent uniquement l'information spectrale (c'est-à-dire les valeurs des niveaux de gris). Il n'est donc pas surprenant qu'il y ait une activité intense depuis quelques années pour essayer d'intégrer d'autres caractéristiques, principalement la texture, dans les procédures de classification numérique. Le but de ce chapitre consiste à présenter des informations détaillées sur l'établissement des formules concernant les notions de texture, d'analyse fractale, d'analyse granulométrique d'une image de texture et du spectre de texture.

Le développement des techniques d'analyse d'images de plus en plus sophistiquées suscite une augmentation continue du nombre de méthodes d'analyse de texture proposées. La sélection heuristique ou manuelle des paramètres devient par conséquent une activité de plus en plus difficile. Ces considérations nous conduisent à la recherche des algorithmes produisant la combinaison la plus appropriée de paramètres pour un problème donné. De tels algorithmes devront non seulement économiser le temps de traitement, mais aussi réduire la probabilité de mauvaise classification.

8.2 Les transformations de texture

8.2.1 Introduction

La texture est un concept qui traduit un aspect homogène local de la surface d'un objet. C'est un concept très important et largement utilisé dans la plupart des domaines du traitement d'images. Malgré cette importance, la notion de texture reste un concept sans définition universelle satisfaisante. De nombreuses définitions ont été proposées, mais aucune ne convient parfaitement aux différents types de texture rencontrées. L'une des plus citées a été introduite par Laws (Unser, 1984). Dans sa définition, la texture est présentée comme une structure disposant de certaines propriétés spatiales homogènes et invariantes par translation. Cette définition stipule que la texture donne la même impression à l'observateur, quelle que soit la position spatiale de la fenêtre à travers laquelle il observe cette texture. Par contre l'échelle d'observation doit être précisée. On peut le faire par exemple en précisant la taille de

la fenêtre d'observation. Une autre définition précise que la notion de texture est liée à trois concepts principaux :

1. un certain ordre local qui se répète dans une région de taille assez grande;
2. cet ordre est défini par un arrangement structuré de ses constituants élémentaires;
3. ces constituants élémentaires représentent des entités uniformes, qui se caractérisent par des dimensions semblables dans toute la région considérée.

On constate que la multiplicité des textures entraîne une multiplication des définitions. Cependant, toutes s'accordent à séparer les textures en deux classes : les textures structurées (macrotextures) et les textures aléatoires (microtextures) (Kpalma, 1992). Une texture qualifiée de structurée est constituée par la répétition d'une primitive à intervalles réguliers. On peut différencier dans cette classe les textures parfaitement périodiques (carrelage, damier, ...) et les textures dont la primitive subit des déformations ou des changements d'orientation (mur de briques, grains de café, ...). Les textures qualifiées d'aléatoires se distinguent en général par un aspect plus fin (sable, laine tissée, herbe, ...). Contrairement aux textures de type structuré, les textures aléatoires ne comportent ni primitive isolable, ni fréquence de répétition. On ne peut donc pas extraire de ces textures une primitive qui se répète dans l'image, mais plutôt un vecteur de paramètres statistiques homogènes à chaque texture. Les textures rencontrées dans les images de télédétection étant généralement de type aléatoire, les méthodes basées sur une analyse statistique de l'image sont probablement les plus appropriées. Le but de cette section est de fournir une description consistante des principales approches d'analyse de texture qui ont trouvé des applications en télédétection. Les différentes méthodes proposées dans la littérature peuvent se répartir en quatre catégories distinctes :

- 1) les modèles de processus aléatoire,
- 2) les méthodes d'analyse des caractéristiques fréquentielles,
- 3) les méthodes d'analyse des caractéristiques structurelles,
- 4) les méthodes d'analyse des caractéristiques spatiales.

Les méthodes basées sur la modélisation d'un processus aléatoire considèrent la dépendance spatiale des niveaux de gris entre pixels voisins dans l'image. Dans ce cas, le type du modèle et les valeurs des paramètres du modèle caractérisent la texture. On peut tester l'adéquation du modèle et la fidélité des paramètres en effectuant une synthèse de la texture analysée, et en la comparant avec la texture originelle. Les méthodes basées sur l'analyse des caractéristiques spatiales sont nombreuses et elles sont de loin les plus utilisées pour l'analyse des images de télédétection. Ces méthodes peuvent être groupées en quatre sous-groupes basées sur les paramètres suivants :

- 1) les histogrammes généralisés de niveaux de gris;
- 2) la mesure de l'activité du signal de texture;
- 3) la morphologie mathématique;
- 4) l'analyse fractale.

Les histogrammes généralisés mesurent les statistiques de la distribution des niveaux de gris sur différents n-uplets de pixels voisins. Ainsi, le caractère plus ou moins périodique de la texture peut être relevé. La taille et l'orientation des primitives des textures structurées régulières peuvent être obtenues par la fonction d'autocorrélation ou par la méthode des histogrammes de cooccurrence d'ordre deux ou trois. L'activité du signal de texture peut être estimée par différentes approches. On peut mesurer la densité de contours ou d'extrema locaux. On peut aussi calculer la longueur curviligne du signal sur un trajet donné. Ces méthodes ont le mérite de la simplicité. Elles se prêtent donc facilement à des mises en œuvre sur de petits processeurs.

La morphologie mathématique, développée par Serra (1982), a largement montré son efficacité sur les images biologiques ou en métallurgie (Analyseur de Texture LEITZ). Des éléments structurants de formes et de tailles variées permettent de révéler des éléments particuliers de la texture, à partir desquels on pourra extraire quelques paramètres simples (par exemple, la moyenne et variance des niveaux de gris). Cette méthode produit de bons résultats pour certains types de textures (textures structurées en particulier).

Dans l'approche fractale, l'amplitude des niveaux de gris peut être assimilée à l'altitude d'une surface géométrique. La dimension fractale de cette surface est utilisée pour caractériser la texture. En général, la dimension fractale est insuffisante pour la discrimination. Elle peut être complétée par un autre paramètre appelé lacunarité. Cette approche se révèle particulièrement fructueuse pour des textures aléatoires de type microtexture, mais aussi macrotexture (échographie des poumons, par exemple).

Les caractéristiques fréquentielles mesurent les composantes de fréquence spatiale de la texture. Depuis longtemps, la transformée de Fourier a été utilisée pour calculer ces composantes à partir du spectre de puissance. D'autres transformées plus sophistiquées comme la transformée en ondelettes et les filtres de Gabor ont été proposées ces dernières années, dans le but de contourner certaines limites de la transformée de Fourier.

8.2.2 Modélisation par des processus aléatoires

Dans les paragraphes qui suivent, nous présentons quelques modèles de base généraux.

a- Modèle de prédiction linéaire 2D

On fait l'hypothèse que la valeur du niveau de gris du pixel considéré est une combinaison linéaire des valeurs de niveau de gris des pixels voisins. On peut alors caractériser l'image de texture, notée $f = \{f(i, j); 0 \leq i, j \leq M - 1\}$, par l'expression de l'équation (8.1).

$$f(i, j) = \hat{f}(i, j) + \varepsilon(i, j) \quad (8.1)$$

Le terme $\hat{f}(i, j)$ apparaissant dans l'équation (8.1) est la prédiction et elle est définie par l'équation (8.2).

$$\hat{f}(i, j) = \sum_{(k,l) \in R} \alpha(k,l) \cdot f(i-k, j-l) \quad (8.2)$$

Dans l'expression (8.2), on a :

- $\alpha(k,l)$: coefficient de prédiction,
- R : domaine de prédiction,
- $\varepsilon(i, j)$: erreur entre prédiction et observation.

Le domaine de prédiction R est causal, semi-causal ou non-causal selon que les points appartenant au domaine R existent tous ou non avant le point considéré. Ici, la notion de temps est donnée par le balayage séquentiel (de gauche à droite et de haut en bas) de l'écran de visualisation.

b- Modèle de filtre linéaire

L'erreur de prédiction $\varepsilon(i, j)$ est un élément d'un champ aléatoire E . On considère que c'est un champ de bruit blanc (sa densité spectrale de puissance est une constante). La fonction de covariance $g_e(m, n)$ s'exprime donc par la relation (8.3) ci-après.

$$g_e(m, n) = h^2 \delta(m, n),$$

$$\text{avec } \delta(m, n) = \begin{cases} 1 & \text{si } m = n \\ 0 & \text{si } m \neq n \end{cases} \quad (8.3)$$

La texture $f(i, j)$ est considérée comme la sortie d'un système linéaire dont l'entrée est un bruit blanc (figure 8.1).

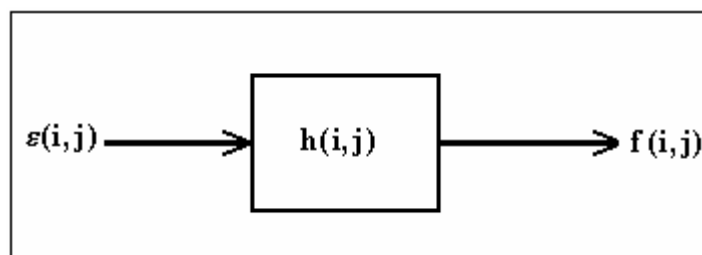


Figure 8.1 Représentation schématique d'un système linéaire dont l'entrée est un bruit blanc.

Elle est caractérisée par trois paramètres:

- 1) la valeur moyenne du signal : $m = E\{f(i,j)\}$
- 2) la fonction de transfert du filtre : $H(z)$ (transformée en z de la réponse impulsionnelle $h(i,j)$ du système).
- 3) l'histogramme du bruit.

c- Modèle de Markov

On fait l'hypothèse que la valeur de niveau de gris du pixel considéré ne dépend que d'un voisinage réduit du pixel, ceci suppose que tout le passé du processus est contenu dans ce même voisinage. Le processus markovien est complètement défini par :

- 1) un état initial;
- 2) un ensemble de probabilités entre états (matrice de transition du processus).

Chaque état du processus correspond à une distribution particulière des niveaux de gris sur le voisinage. Les champs aléatoires de Markov peuvent être utilisés pour l'analyse et la synthèse de texture, ainsi que pour la segmentation d'images texturées. Analyser la texture par modèle de Markov consiste à estimer les paramètres du modèle, qui deviennent des caractéristiques de la texture. On peut montrer que les valeurs prises par les paramètres qui modélisent la texture sont des fréquences spatiales. Généralement, les paramètres sont estimés dans les quatre directions classiques (0° , 45° , 90° , 135°) et ils renseignent sur les fréquences spatiales dans ces quatre directions. Un champ de Markov est un modèle de la dépendance qui existe entre les niveaux de gris de l'image. La principale propriété d'un champ de Markov exprime que la valeur de niveau de gris d'un pixel ne dépend pas des niveaux de gris de tous les autres pixels de l'image, mais seulement d'un certain nombre d'entre eux, qui appartiennent à un voisinage du pixel considéré. On considère qu'un pixel n'appartient pas à son propre voisinage. De plus, si un pixel p_1 appartient au voisinage d'un pixel p_2 , alors la réciproque est vraie. La figure 8.2 présente un exemple de voisinage qui peut être attribué au pixel (i,j) (ici, on considère le 4-voisinage formé des 4 plus proches pixels).

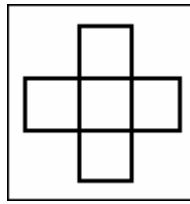


Figure 8.2 Exemple de voisinage du pixel (i,j) : le 4-voisinage.

Etant donné que le niveau de gris d'un pixel ne dépend que de ses plus proches voisins, il est intéressant d'étudier cette dépendance et de comprendre comment les paramètres du modèle markovien évoluent en fonction des relations qui existent entre pixels voisins. A ce niveau, la notion de clique doit être introduite. Une clique est un ensemble de pixels (souvent un, deux ou trois mais parfois plus) tels que chacun des pixels appartient au voisinage (qui a été défini préalablement, et qui peut être un 4-voisinage, un 8-voisinage ou un autre) des autres pixels. La figure 8.3 montre l'exemple du 4-voisinage et des cliques qui lui sont associées. On distingue une clique à un pixel (singleton) et 4 cliques à deux pixels (doubletons), chacune étant orientée dans une des directions classiques ($0^\circ, 45^\circ, 90^\circ$ et 135°).

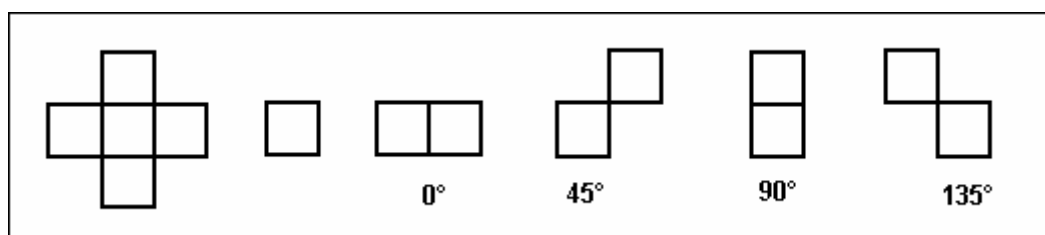


Figure 8.3 Le 4-voisinage et les cliques associées.

A chaque clique est associée une fonction appelée fonction potentiel, qui s'exprime généralement comme une fonction linéaire du paramètre correspondant à cette clique. La valeur de la fonction potentiel dépend des relations qui existent entre les niveaux de gris des pixels de la clique. Un certain nombre de méthodes sont utilisées pour estimer les paramètres du modèle. Les valeurs estimées des paramètres permettent de connaître les fréquences spatiales des niveaux de gris dans les directions choisies. Dans la plupart des modèles, plus la valeur d'un paramètre est faible (voire négative), plus les fréquences spatiales dans la direction associée sont élevées. Les modèles markoviens sont très utilisés en segmentation d'images bruitées ou texturées. Pour la caractérisation des textures, peu de résultats probants ont été obtenus à ce jour. Ceci est dû au fait que la modélisation par processus markovien suppose une bonne stationnarité du signal et qu'elle ne s'adapte bien qu'aux micro-textures. Deux modèles dérivés du modèle de Markov sont présentés dans les paragraphes qui suivent.

d- Modèle probabiliste bidimensionnel

En pratique, et pour tenir compte du caractère bidimensionnel de l'image, le voisinage est limité à trois pixels : les voisins immédiats gauche et supérieur. De plus, pour limiter le nombre N de probabilités dans la matrice de transition, la texture est quantifiée sur un nombre réduit (N_g) de niveau de gris. Pour un voisinage de taille $u*v$, possédant N_g niveaux de gris, le nombre de configurations différentes est $N = N_g^{(u*v)}$, ce qui montre l'explosion combinatoire de N avec $u*v$. Pour estimer les paramètres du modèle, on calcule l'histogramme des différentes configurations rencontrées, puis on le normalise pour obtenir les probabilités. Cependant, pour obtenir une estimation valable de ces probabilités sur une petite zone d'image, on doit avoir un nombre réduit de types différents de configurations. Il est donc nécessaire de d'abord quantifier la texture sur quelques niveaux seulement. En pratique, la quantification s'effectue localement sur trois niveaux, correspondant à la valeur moyenne et la valeur moyenne plus ou moins l'écart-type. L'intérêt de ce modèle réside dans la simplicité de sa mise en œuvre, aussi bien pour l'analyse que pour la synthèse des textures. Les inconvénients majeurs de ce modèle résident d'une part, dans la taille très limitée du voisinage, et d'autre part, dans le nombre limité des niveaux de gris. Cet inconvénient a pour conséquence le fait qu'avec ce modèle, on ne peut bien décrire que des textures de type aléatoire à grain fin (microtextures).

e- Modèle probabiliste avec quantification vectorielle

La quantification vectorielle permet de limiter les inconvénients précédents en :

- 1) agrandissant le voisinage (de $2*2$ à $4*4$ typiquement)
- 2) augmentant le nombre des niveaux de gris (de 4 à 32 typiquement)

Ces améliorations se font sans augmentation du nombre de paramètres. Le principe de la quantification vectorielle consiste à quantifier la texture en remplaçant les configurations $u*v$ par les N configurations $u*v$ les plus fréquemment rencontrées (plus exactement, par les centres des N amas principaux dans l'espace vectoriel des configurations). On obtient alors une image vectorielle de dimension $1/u*1/v$ par rapport à l'image originale, où chaque vecteur possède N valeurs possibles. L'analyse probabiliste 2D s'effectue sur cette image vectorielle avec un voisinage $U*V$, ce qui correspond à un voisinage $(U*V)*(u*v)$ sur la texture. De même, les N valeurs différentes pour les vecteurs correspondent à beaucoup plus que N niveaux de gris différents (au maximum à $N*(u*v)$ niveaux de gris). Les résultats obtenus en analyse et en synthèse de textures mettent en évidence la possibilité de modéliser des textures possédant des primitives fortement aléatoires, et de taille supérieure à quelques pixels (macrotexture aléatoire).

8.2.4 Les méthodes basées sur les caractéristiques spatiales

a- Les histogrammes généralisés des niveaux de gris

L'information texturale associée à un pixel est contenue dans les relations spatiales entre le pixel traité et ses voisins. Il s'agit ici de déterminer le réseau de pixels voisins impliqués dans l'analyse et la fonction de variation des niveaux de gris entre ces pixels. Cette information peut être caractérisée statistiquement par un histogramme N-dimensionnel, avec M niveaux de gris, dans lequel chaque entrée $p\{f(i_1), f(i_2), \dots, f(i_n)\}$ représente une estimation de la probabilité jointe du n-uplet $\{f(i_1), f(i_2), \dots, f(i_n)\}$. N est le nombre de voisins utilisés et $f(i_k)$ est une fonction du niveau de gris du pixel i_k . A partir de ces histogrammes multidimensionnels, des statistiques de différents ordres peuvent être calculées pour caractériser la texture d'une image (Wang, 1994). C'est cette approche qui soustend la plupart des méthodes probabilistes d'analyse de texture que nous présentons dans les paragraphes qui suivent.

b- La méthode de l'histogramme du premier ordre

L'histogramme du premier ordre est surtout utilisé sous le nom d'histogramme d'image. Il décrit la distribution des niveaux de gris de l'image. Un élément $p(i)$ d'un histogramme du premier ordre indique la fréquence du niveau de gris i dans l'image. L'histogramme du premier ordre exploite uniquement l'information de niveaux de gris des pixels et non les relations spatiales entre eux. Il est souvent utilisé pour la segmentation par seuillage sur le niveau de gris, dans le cas où l'image présente des zones uniformes en niveau de gris. Différentes statistiques du premier ordre peuvent être calculées à partir de l'histogramme des niveaux de gris d'une image. La moyenne et la variance des niveaux de gris sont souvent calculées pour une première caractérisation de l'image. Des paramètres de degré supérieur caractérisent la forme de l'histogramme (vitesse de décroissance, etc.). Ces statistiques ne sont réellement valables que si la distribution des niveaux de gris est assez régulière (histogramme possédant un seul mode). Quelques paramètres de texture, dérivés de l'histogramme du premier ordre sont présentés dans les lignes qui suivent (équations (8.4a-g)).

- **La moyenne des niveaux de gris (μ) :**
$$F_1 = \sum_{i=0}^{L-1} i \times P(i) \tag{8.4a}$$

La moyenne représente l'emplacement de l'histogramme sur l'échelle des niveaux de gris.

$$\text{- Entropie :} \quad F_2 = -\sum_{i=0}^{L-1} P(i) \times \ln[P(i)] \quad (8.4b)$$

Le paramètre d'entropie mesure le degré de désordre de la texture considérée.

$$\text{- Energie :} \quad F_3 = \sum_{i=0}^{L-1} [P(i)]^2 \quad (8.4c)$$

Plus douce (plus homogène) est la texture, plus grande est la valeur du paramètre d'énergie.

$$\text{- La variance des niveaux de gris } (\sigma^2) : \quad F_4 = \sum_{i=0}^{L-1} (i - \mu)^2 P(i) \quad (8.4d)$$

La variance est une mesure de la variabilité des niveaux de gris autour de la moyenne.

$$\text{- Le coefficient de symétrie :} \quad F_5 = \frac{1}{\sigma^3} \left[\sum_{i=0}^{L-1} (i - \mu)^3 P(i) \right] \quad (8.4e)$$

Le coefficient de symétrie mesure la symétrie de la distribution des niveaux de gris autour de la moyenne.

$$\text{- Le coefficient d'aplatissement :} \quad F_6 = \frac{1}{\sigma^4} \left[\sum_{i=0}^{L-1} (i - \mu)^4 P(i) \right] \quad (8.4f)$$

Le coefficient d'aplatissement donne une indication sur l'aspect pointu et effilé ou plutôt aplati de l'histogramme.

$$\text{- Le coefficient de variation :} \quad F_7 = \frac{\sigma}{\mu} \times 100\% \quad (8.4g)$$

Le coefficient de variation mesure l'hétérogénéité locale de l'image.

c- La méthode de l'histogramme local

Une utilisation de l'histogramme local a été proposée par Lowitz (1984). Cette méthode se propose de caractériser l'information spatiale locale d'une image, à travers une fenêtre glissante qui balaie toute l'image. On calcule l'histogramme h des luminances des points d'image appartenant à la fenêtre d'observation, puis on caractérise cet histogramme par son module $\|h\|$ et sa phase ϕ_h . Supposons une fenêtre de taille $n \times n$ centrée sur le pixel (x,y) dans une image quantifiée sur r niveaux de gris. Les histogrammes locaux doivent alors rester dans l'hyperplan décrit par l'expression (8.24).

$$\sum_{i=1}^r c_i = n^2 \quad (8.5)$$

Dans l'expression (8.24), c_i représente le nombre de points d'image au $i^{\text{ème}}$ niveau de gris.

Lowitz (1984) a introduit les points remarquables suivants :

- 1) l'histogramme centré : $h_0 = \left\{ \frac{n^2}{r}, \frac{n^2}{r}, \dots, \frac{n^2}{r} \right\}$
- 2) les histogrammes maximaux : $h_{j_0}^M = \{0, \dots, n^2, \dots, 0\}$

Le modèle d'un histogramme local est défini comme étant la distance entre l'histogramme centré h_0 et l'histogramme considéré h (expression (8.6)).

$$\|h\| = \sum_{i=1}^r \left(c_i - \frac{n^2}{r} \right) \left[\log(c_i) - \log\left(\frac{n^2}{r}\right) \right] \quad (8.6)$$

La phase d'un histogramme local est défini comme étant le plus petit indice j de l'histogramme maximal h_j^M (le plus grand au sens de $\|h\|$) de l'histogramme local considéré. On a donc $\phi_h = j$. Une autre utilisation de l'histogramme local en analyse de texture est celle de l'histogramme cumulé complémentaire (HCC décrite par Kpalma (1992)). Considérons une fenêtre d'observation de taille donnée. L'observation de la surface 3D de l'image dans cette fenêtre montre des structures en relief de forme et de taille variées en fonction de la texture. Si on découpe ce relief à une hauteur donnée, l'aire résultante (nombre de pixels dont la valeur est supérieure au niveau de découpe) constitue un attribut discriminant de la texture. Soient I_{\min} et I_{\max} les valeurs minimale et maximale de niveaux de gris dans cette fenêtre. La dynamique maximale de luminance caractérisant la fenêtre considérée est alors obtenue par $\Delta I = I_{\max} - I_{\min}$. L'aire de ces sections en fonction du niveau de découpe k ($k \in [0, \Delta I]$) correspond à ce qu'on appelle histogramme cumulé complémentaire (HCC). Si $H(k)$ est la valeur de l'histogramme normalisé de ce signal en une valeur k , HCC(k) s'obtient par l'expression (8.7).

$$HCC(k) = \sum_{i=k}^{\Delta I} H(i) , \forall k \in [0, \Delta I] \quad (8.7)$$

HCC(k) est la probabilité d'avoir un pixel dont la valeur de niveau de gris est supérieure à k , dans une fenêtre. C'est une mesure de la distribution des valeurs du signal de luminance sur la dynamique des niveaux de gris dans un voisinage donné. Des utilisations de l'histogramme local en analyse d'images peuvent être trouvées dans les références (Kpalma, 1992) et (Raboison, 1985).

d- La méthode des matrices de cooccurrence

Un ensemble de mesures de texture d'ordre supérieur à 1 été proposé par Haralick et Shanmugan (1973), basé sur le calcul des matrices de cooccurrence des niveaux de gris. Supposons que $c = (\Delta x, \Delta y)$ soit un vecteur du plan image (x, y) , où Δx et Δy sont des entiers. Alors, pour toute image $f(x, y)$, il est possible de calculer la distribution de probabilité jointe des paires de niveaux de gris correspondant à des paires de pixels séparés par $c = (\Delta x, \Delta y)$. Habituellement, le vecteur de séparation $c = (\Delta x, \Delta y)$ est défini par une distance interpixels d et un angle interpixels θ . Si les valeurs de niveaux de gris varient de 0 au plus grand niveau de quantification de l'image (par exemple, $\text{quant}_k = 255$), cette probabilité jointe prend la forme d'une matrice p_c , où $p_c(i, j) = p(i, j; c)$. La grandeur $p(i, j; c)$ est la probabilité que les paires de valeurs de niveaux de gris (i, j) apparaissent à la séparation $c = (\Delta x, \Delta y)$. C'est une matrice carrée d'ordre quant_k . On calcule la matrice p_c pour une image $f(x, y)$ en comptant le nombre de fois que chaque paire de niveaux de gris apparaît dans l'image à la séparation $c = (\Delta x, \Delta y)$. Il est communément admis que toute l'information texturale dérivable des matrices de cooccurrence est obtenue pour les angles 0° , 45° , 90° et 135° , avec une distance interpixels égale à 1 ou 2. En règle générale, plus grands sont les éléments diagonaux de la matrice de cooccurrence, plus la texture est homogène dans la zone d'image considérée. Pour illustrer le calcul des matrices de cooccurrence, considérons l'image présentée sur la figure 8.4. Cette image a 5 lignes et 5 colonnes, avec des niveaux de gris variant de 0 à 3.

0	1	1	2	3
0	0	2	3	3
0	1	2	2	3
1	2	3	2	2
2	2	3	3	2

Image originale

Figure 8.4 Image numérique servant à illustrer le calcul des matrices de cooccurrence.

L'application de la méthode de calcul de la matrice de cooccurrence à cette image produit les matrices de cooccurrence présentées sur la figure 8.5.

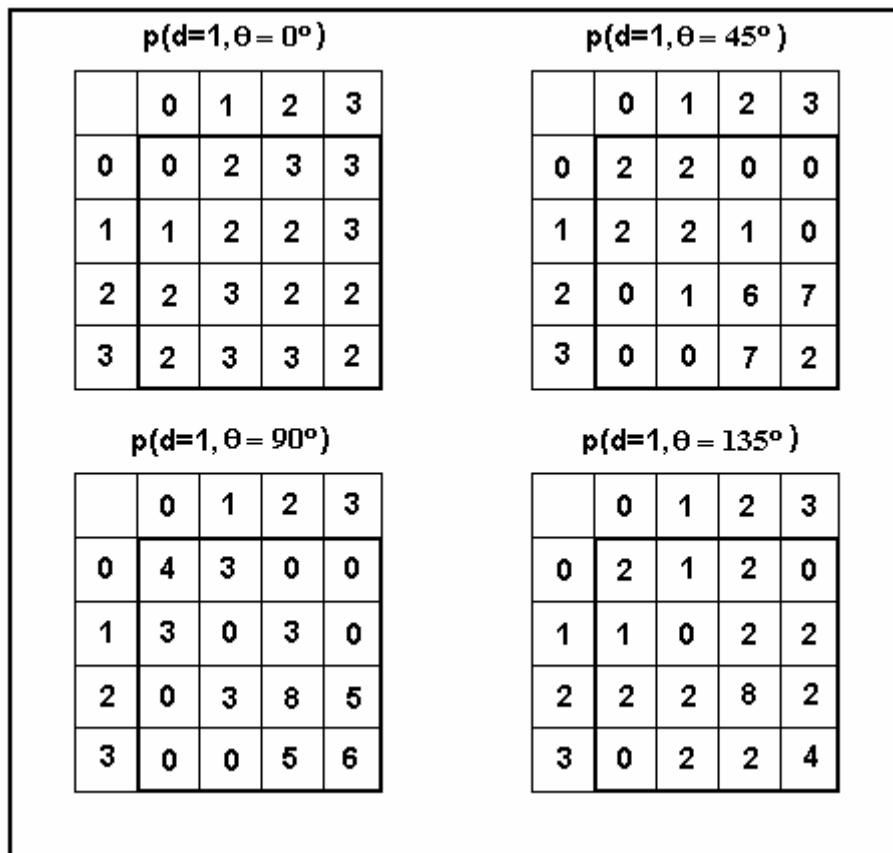


Figure 8.5 Matrices de cooccurrence issues de l'image de la figure 8.4. Ces matrices ont été évaluées pour une distance interpixels égale à 1, et pour divers angles (0°, 45°, 90° et 135°).

Haralick et Shanmugan (1973) ont proposé une variété de mesures pour caractériser l'information texturale à partir des matrices de cooccurrence. Les expressions de ces mesures sont présentées dans les lignes qui suivent (équations (8.8a-o)).

- **Second Moment Angulaire (Energie) :**
$$F_1 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} [p(i, j)]^2 \quad (8.8a)$$

Le second moment angulaire mesure l'homogénéité de l'image. Il a une valeur faible lorsque les $p(i,j)$ ont des valeurs très proches et une grande valeur lorsque certaines valeurs sont grandes et d'autres petites, par exemple quand les $p(i,j)$ sont concentrés autour de la diagonale.

- **Contraste (Inertie) :**
$$F_2 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} (i - j)^2 p(i, j) \quad (8.8b)$$

Le contraste est une mesure de la variation locale des niveaux de gris dans l'image. Ce paramètre a une valeur numérique importante si les $p(i,j)$ sont concentrés hors diagonale

(c'est-à-dire s'il existe de nombreuses transitions caractérisées par une différence de niveaux de gris (i-j) importante). Au contraire, si les valeurs de la matrice de cooccurrence sont concentrées autour de la diagonale, le contraste sera faible.

- **Corrélation:**
$$F_3 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \frac{(i - \mu_x)(j - \mu_y) \times p(i, j)}{\sigma_x \sigma_y} \quad (8.8c)$$

où μ_x , μ_y , σ_x , et σ_y sont les moyennes et les écart-types. p_x et p_y sont les distributions marginales associées à la matrice de cooccurrence $p(i, j)$ et définies par les expressions (8.8d) ci-après.

$$\begin{cases} p_x(i) = \sum_{j=0}^{N_g} p(i, j) \\ p_y(j) = \sum_{i=0}^{N_g} p(i, j) \end{cases} \quad (8.8d)$$

Ce paramètre est le coefficient de corrélation des pixels de la fenêtre avec un modèle linéaire, la droite de régression étant la diagonale de la matrice de cooccurrence. Ce paramètre a une grande valeur quand les valeurs sont uniformément distribuées dans la matrice de cooccurrence, et une faible valeur dans le cas contraire.

- **Somme des carrés des variances :**
$$F_4 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} (i - \mu_x)^2 \times p(i, j) \quad (8.8e)$$

- **Moment Différentiel Inverse :**
$$F_5 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \frac{1}{1 + (i - j)^2} \times p(i, j) \quad (8.8f)$$

- **Somme des moyennes :**
$$F_6 = \sum_{i=2}^{2N_g} i \times p_{x+y}(i) \quad (8.8g)$$

- **Somme des variances :**
$$F_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 \times p_{x+y}(i) \quad (8.8h)$$

- **Entropie :**
$$F_8 = - \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} p(i, j) \times \log(p(i, j)) \quad (8.8i)$$

- **Somme des entropies :**
$$F_9 = - \sum_{i=2}^{2N_g} p_{x+y}(i) \times \log(p_{x+y}(i)) \quad (8.8j)$$

L'entropie mesure la complexité de l'image. Elle est grande quand les valeurs de la matrice de cooccurrence sont presque toutes égales et elle est faible dans le cas contraire.

- **Variance de Différence :**
$$F_{10} = \text{variance de } p_{x-y} \quad (8.8k)$$

- **Entropie de Différence :**
$$F_{11} = - \sum_{i=2}^{2N_g} p_{x-y}(i) \times \log(p_{x-y}(i)) \quad (8.8l)$$

- **Mesures de l'information de corrélation :**

$$F_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad (8.8m)$$

$$F_{13} = [1 - \exp(-2(HXY2 - HXY))]^{1/2} \quad (8.8n)$$

où

HX et HY sont les entropies de p_x et p_y , respectivement.

$$HXY = - \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} p(i, j) \times \log(p(i, j)),$$

$$HXY1 = - \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} p(i, j) \times \log(p_x(i)p_y(j)),$$

$$HXY2 = - \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} p_x(i)p_y(j) \times \log(p_x(i)p_y(j)).$$

Coefficient de corrélation maximal :

$$F_{14} = [\text{Deuxième plus grande valeur propre de } Q]^{1/2} \quad (8.8o)$$

où

$$Q(i, j) = \sum_k \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)}$$

$$p_{x+y}(k) = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \{p(i, j); |i + j| = k\}, \quad k = 2, 3, \dots, 2N_g$$

$$p_{x-y}(k) = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \{p(i, j); |i - j| = k\}, \quad k = 0, 1, 2, 3, \dots, N_g$$

A la suite de Haralick et Shanmugan (1973), plusieurs auteurs (Pratt, 1991; Anys and He, 1995; Collorec, 1995) ont proposé d'autres paramètres de cooccurrence du deuxième ordre dont les plus connus sont présentés ci-après (équations (8.9a-i)).

- **Moment Diagonal :**
$$F_{15} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \left(\frac{1}{2} |i - j| p(i, j) \right)^{1/2} \quad (8.9a)$$

- **Moyenne :**
$$F_{16} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} i \times p(i, j) \quad (8.9b)$$

- **Paramètre «cluster shade» :**
$$F_{17} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} (i + j - 2F_{16})^3 \times p(i, j) \quad (8.9c)$$

$$\text{- Paramètre «cluster prominence» : } F_{18} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} (i + j - 2F_{16})^4 \times p(i, j) \quad (8.9d)$$

$$\text{- Différence Inverse : } F_{19} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \frac{p(i, j)}{1 + |i - j|} \quad (8.9e)$$

$$\text{- Dissimilarité ou Valeur Absolue : } F_{20} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} |i - j| \times p(i, j) \quad (8.9f)$$

$$\text{- Covariance : } F_{21} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} (i - \mu_x)(j - \mu_y) \times p(i, j) \quad (8.9g)$$

$$\text{- Probabilité Maximale : } F_{22} = \max_{0 \leq i, j \leq N_g} \{p(i, j)\} \quad (8.9h)$$

$$\text{- Troisième Moment Angulaire : } F_{23} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} [p(i, j)]^3 \quad (8.9i)$$

La plupart de ces paramètres ont une signification texturale semblable à celles des précédents. Cependant, quelques uns de ces paramètres permettent de caractériser un aspect différent de la texture d'une image. Il est donc courant que l'on procède à une sélection de paramètres pour déterminer les plus appropriés pour une application donnée.

e- Nouvelle approche dérivée de la matrice de cooccurrence des niveaux de gris

La méthode de calcul des paramètres de texture par les matrices de cooccurrence est très consommatrice en temps de calcul et en espace mémoire. Divers auteurs se sont penchés sur ce problème, lui trouvant chacun une solution propre, ainsi, Unser (1986) remplace la matrice de cooccurrence par la somme et la différence d'histogrammes qui définissent les axes principaux des probabilités de second ordre des processus stationnaires. Marceau et al. (1990) proposent une approche texturale et spectrale pour la classification de différents thèmes et adopte la réduction du niveau de quantification (16, 32 au lieu de 256), sans altérer la précision de la classification. Peckinpaugh (1991) décrit une approche efficace pour le calcul des mesures de texture basées sur la matrice de cooccurrence permettant un gain de temps appréciable. Kourgli et Belhadj-Aissa (1999) présentent un nouvel algorithme pour calculer des paramètres de texture par le biais de différents histogrammes. Cet algorithme nécessite l'allocation d'un vecteur au lieu d'une matrice, et le calcul des paramètres de texture se fait selon de nouvelles formules faisant appel à une somme et non à une double somme, ce qui réduit considérablement le temps de calcul. La méthodologie de cette méthode est présentée sur les lignes qui suivent.

Pour illustrer cette méthode, considérons l'exemple du calcul de l'élément moyenne. Pour une fenêtre F de taille $N_1 \times N_c = 3 \times 3$ et une MCNG (Matrice de Cooccurrence des Niveaux de Gris) M symétrique suivant la relation R : (d,θ), la moyenne s'exprime comme suit :

$$\mu_x = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} \frac{p_{ij}}{N}$$

Pour une normalisation appropriée chaque probabilité p_{ij} doit être divisé par le nombre total des p_{ij} . Ainsi, pour un déplacement d horizontal ou vertical, on prendra $N = 2 \times N_1 \times (N_c - d)$ et $N = 2 \times (N_1 - d) \times N_c$, respectivement. Pour un déplacement en diagonale, on prendra $N = 2 \times (N_1 - 1) \times (N_c - 1)$. Illustrons cette méthode par un exemple.

A la fenêtre F correspond la MCNG M pour R(1,0°), tel qu'illustré ci-dessous.

$$F: \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 1 \end{bmatrix} \Rightarrow M: \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 2 & 1 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \end{matrix}$$

Dans ce cas $N = 2 \times N_1 \times (N_c - 1) = 2 \times 3 \times 2 = 12$ et $\mu_x = \sum_{i=1}^3 i \sum_{j=1}^3 \frac{P_{ij}}{12}$.

$$\text{Soit } \mu_x = \frac{(1 \times (2 + 1 + 2) + 2 \times (1 + 0 + 2) + 3 \times (2 + 2 + 0))}{12} = \frac{(1 \times 5 + 2 \times 3 + 3 \times 4)}{12} = \frac{23}{12}$$

Nous noterons, en particulier, que $\sum p_{ij}$ pour un 'i' donné représente la probabilité d'occurrence du niveau de gris 'i' dans la fenêtre suivant F1 additionné à la probabilité d'occurrence du niveau de gris 'i' dans la fenêtre F2.

$$F1(d = 1, \theta = 0^\circ): \begin{bmatrix} 1 & 2 & * \\ 2 & 3 & * \\ 3 & 1 & * \end{bmatrix} \quad \text{et} \quad F2(d = 1, \theta = 180^\circ): \begin{bmatrix} * & 2 & 3 \\ * & 3 & 1 \\ * & 1 & 1 \end{bmatrix}$$

Ainsi $\text{occ}(1) = 2 + 3 = 5$, $\text{occ}(2) = 2 + 1 = 3$ et $\text{occ}(3) = 2 + 2 = 4$.

En règle générale, pour une fenêtre F, le calcul de $\text{occ}(i)$ reviendra à sommer les fréquences des occurrences de 'i' suivant R(d,θ) dans cette fenêtre.

$\text{occ}(i) = \text{fréquence d'occurrence}(i) \text{ suivant } R(d,0)$.

Evaluation de μ_x :

Evaluer μ_x revient à multiplier chaque niveau de gris de la fenêtre image par la fréquence d'occurrence correspondante :

$$\mu_x = \sum_{i=1}^3 \frac{i \times \text{occ}(i)}{12} = \frac{(1 \times 5 + 2 \times 3 + 3 \times 4)}{12} = \frac{23}{12},$$

ce qui donne la nouvelle formulation de μ_x : $\mu_x = \sum_{i=0}^{L-1} \frac{i \times \text{occ}(i)}{N} = \sum_{j=0}^{L-1} \frac{j \times \text{occ}(j)}{N}$.

Ainsi, il est possible d'estimer la moyenne sans avoir à calculer la matrice de cooccurrence et de réduire le calcul d'une double sommation à une simple sommation. De la même façon, il est possible d'estimer d'autres paramètres de texture moyennant le calcul des fréquences d'occurrence sur les fenêtres adéquates :

- la variance :

On procède de la même façon.

A l'origine, la variance est définie par :

$$\sigma_x^2 = \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} \frac{p_{ij}}{N} \quad \text{et devient} \quad \sigma_x^2 = \sum_{i=0}^{L-1} (i - \mu_x)^2 \frac{\text{occ}(i)}{N}$$

- la corrélation :

Sachant que pour une matrice symétrique $\mu_x = \mu_y$ et $\sigma_x = \sigma_y$, on a :

$$C = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{(i - \mu_x)(j - \mu_x)}{\sigma_x \sigma_y} \frac{p_{ij}}{N} = \frac{1}{\sigma_x^2} \left(\sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} j \frac{p_{ij}}{N} - \mu_x^2 \right)$$

De la même façon, reconsidérons le même exemple pour lequel nous allons

estimer $\sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} j \frac{p_{ij}}{N}$.

$$\begin{aligned} \sum_{i=1}^3 \sum_{j=1}^3 j \frac{p_{ij}}{N} &= \frac{(1 \times (1 \times 2 + 2 \times 1 + 3 \times 2)) + (2 \times (1 \times 1 + 2 \times 0 + 3 \times 2)) + (3 \times (1 \times 2 + 2 \times 2 + 3 \times 0))}{12} \\ &= \frac{(1 \times 10) + (2 \times 7) + (3 \times 6)}{12} = \frac{42}{12} = \frac{7}{2} \end{aligned}$$

$\sum j p_{ij}$ pour un 'i' donné représente la somme des 'j' pour lesquels la relation R(d,θ) est vérifiée, ce qui revient à considérer les deux fenêtres F1 et F2.

$$F1(d=1, \theta=0^\circ): \begin{matrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 1 \end{bmatrix} \\ (i,j) \end{matrix} \quad \text{et} \quad F2(d=1, \theta=180^\circ): \begin{matrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 1 \end{bmatrix} \\ (j,i) \end{matrix}$$

Ainsi,

$$\begin{cases} \text{occ}'(1) = (2+1)_{F1} + (3+3+1)_{F2} = 10 \\ \text{occ}'(2) = (3+3)_{F1} + (1)_{F2} = 7 \\ \text{occ}'(3) = (1+1)_{F1} + (2+2)_{F2} = 6 \end{cases}$$

Pour une fenêtre F, le calcul de $\text{occ}'(i)$ revient à sommer pour chaque 'i' les 'j' suivant $R(d, \theta)$ dans cette fenêtre : $\text{occ}'(i) = \sum_j$ suivant $R(d, \theta)$.

$$\sum_{i=1}^3 i \sum_{j=1}^3 j \frac{p_{ij}}{N} = \sum_{i=1}^3 \frac{i \times \text{occ}'(i)}{12} = \frac{(1 \times 10 + 2 \times 7 + 3 \times 6)}{12} = \frac{7}{2}$$

ce qui donne la nouvelle formulation de la corrélation :

$$C = \frac{1}{\sigma_x^2} \left(\sum_{i=0}^{L-1} \frac{i \times \text{occ}'(i)}{N} - \mu_x^2 \right)$$

- l'inertie :

$$\begin{cases} I = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 \frac{p_{ij}}{N} = 2 \sum_{i=0}^{L-1} i^2 \sum_{j=0}^{L-1} \frac{p_{ij}}{N} - 2 \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} j \frac{p_{ij}}{N} \\ I = 2 \sum_{i=0}^{L-1} i^2 \frac{\text{occ}(i)}{N} - 2 \sum_{i=0}^{L-1} i \frac{\text{occ}'(i)}{N} \end{cases}$$

- le groupe de nuance :

$$\begin{aligned} A &= \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ((i - \mu_x) + (j - \mu_x))^3 \frac{p_{ij}}{N} \\ &= 2 \sum_{i=0}^{L-1} (i - \mu_x)^3 \sum_{j=0}^{L-1} \frac{p_{ij}}{N} + 6 \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} j \frac{p_{ij}}{N} - 6\mu_x \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} \frac{p_{ij}}{N} \\ A &= 2 \sum_{i=0}^{L-1} (i - \mu_x)^3 \frac{\text{occ}(i)}{N} + 6 \sum_{i=0}^{L-1} (i - \mu_x)^2 \frac{\text{occ}'(i)}{N} - 6\mu_x \sum_{i=0}^{L-1} (i - \mu_x)^2 \frac{\text{occ}(i)}{N} \\ &= 2 \sum_{i=0}^{L-1} (i - \mu_x)^2 \left((i - 4\mu_x) \frac{\text{occ}(i)}{N} + 3 \frac{\text{occ}'(i)}{N} \right) \end{aligned}$$

- le groupe de prédominance :

$$\begin{aligned}
 B &= \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \left((i - \mu_x) + (j - \mu_x) \right)^4 \frac{P_{ij}}{N} \\
 &= 2 \sum_{i=0}^{L-1} (i - \mu_x)^4 \sum_{j=0}^{L-1} \frac{P_{ij}}{N} + 8 \sum_{i=0}^{L-1} (i - \mu_x)^3 \sum_{j=0}^{L-1} \frac{P_{ij}}{N} - 8 \mu_x \sum_{i=0}^{L-1} (i - \mu_x)^3 \sum_{j=0}^{L-1} \frac{P_{ij}}{N} \\
 &\quad + 6 \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} j^2 \frac{P_{ij}}{N} + 12 \mu_x \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} j \frac{P_{ij}}{N} + 6 \mu_x^2 \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} \frac{P_{ij}}{N} \\
 B &= 2 \sum_{i=0}^{L-1} (i - \mu_x)^4 \frac{\text{occ}(i)}{N} + 8 \sum_{i=0}^{L-1} (i - \mu_x)^3 \frac{\text{occ}'(i)}{N} - 8 \mu_x \sum_{i=0}^{L-1} (i - \mu_x)^3 \frac{\text{occ}(i)}{N} \\
 &\quad + 6 \sum_{i=0}^{L-1} (i - \mu_x)^2 \frac{\text{occ}''(i)}{N} - 12 \mu_x \sum_{i=0}^{L-1} (i - \mu_x)^2 \sum_{j=0}^{L-1} \frac{\text{occ}'(i)}{N} + 6 \mu_x^2 \sum_{i=0}^{L-1} (i - \mu_x)^2 \frac{\text{occ}(i)}{N} \\
 &= 2 \sum_{i=0}^{L-1} (i - \mu_x)^2 \left((i^2 - 6i\mu_x + 8\mu_x^2) \frac{\text{occ}(i)}{N} + (4i - 10\mu_x) \frac{\text{occ}'(i)}{N} + 3 \frac{\text{occ}''(i)}{N} \right)
 \end{aligned}$$

où $\text{occ}''(i) = \sum j^2$ suivant $R(d, \theta)$.

Les paramètres que l'on peut estimer grâce à cet algorithme sont des paramètres faisant intervenir des sommations directes sur les probabilités d'occurrence. Les paramètres tels que l'énergie, l'entropie et l'homogénéité locale font intervenir des calculs sur les probabilités prises individuellement. On ne peut donc pas appliquer cette méthode pour les estimer. Cette méthode a été appliquée avec succès par les auteurs, réduisant le temps de calcul des paramètres de texture dans un rapport approximatif de 1/60, par rapport à la méthode classique.

f- La méthode des différences de niveaux de gris

La méthode des différences de niveaux de gris est décrite ci-après (Weszka et al., 1976). Soit $f(x, y)$ la fonction de niveau de gris d'une image. Pour tout vecteur $\delta = (\Delta x, \Delta y)$ de l'espace image, où Δx et Δy sont des entiers (δ peut être défini par un angle interpixels θ et une distance interpixels d), on définit $f_\delta(x, y) = |f(x, y) - f(x + \Delta x, y + \Delta y)|$. Soit finalement $p(\bullet|\delta)$ l'estimation de la fonction de probabilité associée aux valeurs possibles de f_δ , c'est-à-dire $p(m|\delta) = p(m) = \Pr\{f_\delta(x, y) = m\}$. La fonction $p(\bullet|\delta)$ est souvent appelée fonction de densité des différences de niveaux de gris. Il est communément admis que toute l'information texturale dérivable de la distribution $p(\bullet|\delta)$ est obtenue pour les angles interpixels 0° , 45° , 90° et 135° , avec une distance interpixels d pouvant prendre la valeur 1 ou 2. A partir de ces

distributions de probabilité, plusieurs paramètres statistiques peuvent être calculés pour caractériser la texture locale de l'image (équations (8.10a-h)).

- **Moyenne (μ) :**
$$F_1 = \sum_m m \times p(m) \quad (8.10a)$$

- **Contraste :**
$$F_2 = \sum_m m^2 \times p(m) \quad (8.10b)$$

- **Second Moment Angulaire :**
$$F_3 = \sum_m [p(m)]^2 \quad (8.10c)$$

- **Entropie :**
$$F_4 = -\sum_m p(m) \times \ln[p(m)] \quad (8.10d)$$

- **Moment Différentiel Inverse :**
$$F_5 = \sum_m \frac{p(m)}{1+m^2} \quad (8.10e)$$

- **Variance (σ^2) :**
$$F_6 = \sum_m (m-\mu)^2 p(m) \quad (8.10f)$$

- **Coefficient de symétrie (skewness) :**
$$F_7 = \frac{1}{\sigma^3} \left[\sum_m (m-\mu)^3 p(m) \right] \quad (8.10g)$$

- **Coefficient d'aplatissement (kurtosis) :**
$$F_8 = \frac{1}{\sigma^4} \left[\sum_m (m-\mu)^4 p(m) \right] \quad (8.10h)$$

Le principal avantage de la méthode des différences de niveaux de gris est qu'elle calcule des histogrammes d'ordre un tout en contenant réellement des statistiques d'ordre deux.

g- Méthode des longueurs de plage

La méthode des longueurs de plage de niveaux de gris consiste à compter le nombre de plages de niveaux de gris d'une certaine longueur, dans une certaine orientation (Dasarathy and Holder, 1991). Une matrice $R(\theta) = [r(i, j/\theta)]$ de longueurs de plage est associée à un angle θ , où $r(i, j/\theta)$ est le nombre de plages de pixels de niveau de gris i , de longueur j et dans la direction θ . Généralement, cinq paramètres sont extraits de ces matrices (équations (8.11a-e)).

- **le poids des plages courtes (short run emphasis) :**

$$F_1 = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{r(i, j; \theta)}{j^2} \quad (8.11a)$$

- **le poids des plages longues (long run emphasis) :**

$$F_2 = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N j^2 \times r(i, j; \theta) \quad (8.11b)$$

- la distribution des niveaux de gris :

$$F_3 = \frac{1}{nr} \sum_{i=1}^M \left[\sum_{j=1}^N r(i, j; \theta) \right]^2 \quad (8.11c)$$

- la distribution des longueurs de plages :

$$F_4 = \frac{1}{nr} \sum_{j=1}^N \left[\sum_{i=1}^M r(i, j; \theta) \right]^2 \quad (8.11d)$$

- le pourcentage des niveaux de gris :

$$F_5 = \frac{1}{NB} \sum_{i=1}^M \sum_{j=1}^N r(i, j; \theta) \quad (8.11e)$$

La constante $nr = \sum_{i=1}^M \sum_{j=1}^N r(i, j; \theta)$ est le nombre total de plages dans l'image, M est le nombre de niveaux de gris dans l'image et N est le nombre de longueurs de plages rencontrées dans l'image.

Deux autres paramètres complètent la liste précédente (équations (8.12a-b)).

- le poids des longueurs de plage de petit niveau de gris :

$$F_6 = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{r(i, j; \theta)}{i^2} \quad (8.12a)$$

- le poids des longueurs de plage de grand niveau de gris :

$$F_7 = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N i^2 \times r(i, j; \theta) \quad (8.12b)$$

Quatre autres paramètres ont été proposés par Dasarathy et Holder (1991). Ces paramètres sont une combinaison de certains des paramètres précédents. Ils tiennent compte à la fois, de la distribution des niveaux de gris et des longueurs de plage (8.13a-d).

- le poids des courtes plages de petit niveau de gris :

$$F_8 = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{r(i, j; \theta)}{i^2 j^2} \quad (8.13a)$$

- le poids des courtes plages de grand niveau de gris :

$$F_9 = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{i^2 \times r(i, j; \theta)}{j^2} \quad (8.13b)$$

- **le poids des longues plages de petit niveau de gris :**

$$F_{10} = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N \frac{j^2 \times r(i, j; \theta)}{i^2} \quad (8.13c)$$

- **le poids des longues plages de grand niveau de gris :**

$$F_{11} = \frac{1}{nr} \sum_{i=1}^M \sum_{j=1}^N i^2 j^2 \times r(i, j; \theta) \quad (8.13d)$$

Les performances de cette méthode par rapport à la méthode des matrices de covariance sont controversées. la méthode des longueurs de plage fonctionne mieux que les matrices de covariance sur des textures contenant des formes plus ou moins allongées dans une direction ou dans une autre. Par contre, elle est moins efficace sur des textures constituées de micro-textures (Reuzé, 1995).

h- La méthode des histogrammes du troisième ordre

Un histogramme du troisième ordre est aussi appelé matrice de cooccurrence du troisième ordre. L'élément $p(i, j, k) = p(i, j, k; d_1, \theta_1, d_2, \theta_2)$ d'un histogramme du troisième ordre représente le nombre de fois que le niveau de gris i précède le niveau j selon une distance d_1 et un angle θ_1 , qui à son tour précède le niveau de gris k selon une distance d_2 et un angle θ_2 . Un des aspects importants de ce passage des histogrammes du second ordre aux histogrammes du troisième ordre est que les trois pixels impliqués ne se trouvent pas nécessairement sur une droite. Les couples (d_1, θ_1) et (d_2, θ_2) peuvent être totalement différents (Hevenor, 1985), ce qui résulte en une plus grande prise en compte des relations spatiales existant entre les pixels de l'image analysée. A partir de ces histogrammes du troisième ordre, plusieurs paramètres de cooccurrence définis dans le cas des matrices de cooccurrence du second ordre peuvent être généralisés (équations (8.14a-k)) (Hevenor, 1985).

- **Différence Inverse :**

$$F_1 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} \frac{p(i, j, k)}{1 + |i - j| + |i - k| + |j - k|} \quad (8.14a)$$

- **Dissimilarité ou Valeur Absolue :**

$$F_2 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} [|i - j| + |i - k| + |j - k|] \times p(i, j, k) \quad (8.14b)$$

- **Entropie :**

$$F_3 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} p(i, j, k) \times \ln[p(i, j, k)] \quad (8.14c)$$

- **Contraste :**

$$F_4 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} [(i-j)^2 + (i-k)^2 + (j-k)^2] \times p(i, j, k) \quad (8.14d)$$

- **Second Moment Angulaire :**

$$F_5 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} [p(i, j, k)]^2 \quad (8.14e)$$

- **Moment Différentiel Inverse :**

$$F_6 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} \frac{p(i, j, k)}{1 + (i-j)^2 + (i-k)^2 + (j-k)^2} \quad (8.14f)$$

- **Corrélation :**

$$F_7 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} \frac{(i - \mu_i)(j - \mu_j)(k - \mu_k)}{\sigma^2} \times p(i, j, k) \quad (8.14g)$$

- **Covariance :**

$$F_8 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} (i - \mu_i)(j - \mu_j)(k - \mu_k) \times p(i, j, k) \quad (8.14h)$$

- **Variance :**

$$F_9 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} (i - \mu_i) \times p(i, j, k) \quad (8.14i)$$

- **Probabilité maximale :**

$$F_{10} = \max_{0 \leq i, j, k \leq N_g} \{p(i, j, k)\} \quad (8.14j)$$

- **Troisième Moment Angulaire :**

$$F_{11} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} [p(i, j, k)]^3 \quad (8.14k)$$

A la suite de ces paramètres classiques, de nouveaux paramètres ont été proposés pour tirer le maximum d'information des histogrammes du troisième ordre (équations (8.15a-c)) (Pratt, 1991).

- **Importance des petits nombres :**

$$F_{12} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} \frac{p(i, j, k)}{(i^2 + j^2 + k^2)} \quad (8.15a)$$

- **Importance des grands nombres :**

$$F_{13} = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} (i^2 + j^2 + k^2) \times p(i, j, k) \quad (8.15b)$$

- **Importance de la profondeur :**

$$F_{14} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left(\sum_{k=1}^{N_g} p(i, j, k) \right)^2}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \sum_{k=1}^{N_g} p(i, j, k)} \quad (8.15c)$$

σ^2 a la valeur suivante :

$$\sigma^2 = \sum_{i=0}^{N_g} \sum_{j=0}^{N_g} \sum_{k=0}^{N_g} (i - \mu_i)(j - \mu_j)(k - \mu_k) \times p(i, j, k)$$

Les applications de cette méthode en télédétection sont plutôt rares à cause de sa très grande consommation en temps de calcul et en occupation mémoire. Anys et He (1995) ont évalué la contribution des statistiques de cooccurrence du troisième ordre dans la classification des cultures en utilisant les images radar multipolarisation. Leurs travaux ont conclu que la combinaison des mesures texturales de différents ordres avec les données multipolarisations produit des améliorations du taux de classification correcte de 98% environ.

i- Activité du signal de texture

Une méthode plus simple pour mesurer l'activité du signal de texture consiste à détecter les extréma par filtrage logique, puis à mesurer leur nombre sur un support d'une certaine taille. La densité d'extréma ainsi mesurée permet de différencier un certain nombre de textures. L'inconvénient principal de cette méthode réside dans sa grande sensibilité au bruit. Pour remédier à ce problème, la densité de contours est préférée. Ceux-ci sont extraits par des filtres gradients en horizontal et en vertical. Il est préférable de ne pas seuiller la réponse de ces filtres, ce qui donnerait une image binaire des contours (très dépendante de la valeur du seuil). Il est plutôt souhaitable de prendre en compte l'amplitude du gradient, ce qui offre les deux avantages suivants :

- 1) Il n'y a pas de seuil de binarisation à choisir. Ceci implique que la densité des contours ne varie pas avec la valeur du seuil.
- 2) Dans la mesure de la densité, on intègre l'amplitude du contour, ce qui renforce la présence des contours les plus forts.

La mesure de la densité peut s'effectuer pour chaque réponse de filtre séparément, ce qui permet de mesurer l'activité de la texture en fonction de l'orientation. La technique des

longueurs curvilignes consiste à mesurer la longueur curviligne du signal de luminance le long de certaines lignes d'analyse L_i . L'intégrale curviligne mesure la longueur d'une fonction spatiale de niveaux de gris, selon une demi-droite dans une direction donnée (Barba, 1983; Ronsin et al., 1985). Les demi-droites partent de chaque point p de l'image avec des orientations angulaires différentes $\theta_i \{i = 1, 2, 3, \dots, n\}$ couvrant la gamme $[0, 2\pi]$. Les demi-droites se terminent lorsque la valeur de l'intégrale curviligne atteint un seuil défini à priori. La mesure de la texture, le long d'une demi-droite L_i est donnée par la valeur a_i du déplacement le long de cette ligne. A chaque point p de l'image, on peut associer un vecteur d'attributs (équation (8.16)).

$$V(p) = [a_1, a_2, \dots, a_n]^T \quad (8.16)$$

Dans l'expression (8.39), on a : $a_i = \int_{L_i} ds$, avec $ds = \sqrt{\lambda^2(dx^2 + dy^2) + (df(x, y))^2}$. $f(x, y)$ est la fonction de niveau de gris du pixel situé en (x, y) et λ est la fonction d'échelle introduite entre la variation spatiale et la variation des niveaux de gris. Ce vecteur d'attributs peut être utilisé en segmentation d'images et il peut servir à l'extraction d'attributs en analyse de texture.

8.3 Les indices de végétation

Dans les programmes de recherche sur les changements à l'échelle du globe, les indices de végétation dérivés des données de télédétection constituent une information de base précieuse pour la gestion de l'environnement végétal (Bannari et al., 1995). La collecte des informations précises et opportunes sur la production agricole mondiale et la santé des cultures reste une préoccupation constante de l'humanité (Groten, 1993). La collecte de telles informations en utilisant des techniques **in situ** est très coûteuse, consommatrice en temps et souvent impossible (Eastman and Fulk, 1993). Une approche alternative consiste à mesurer la quantité et la condition de la végétation à partir d'une analyse des données multispectrales (Goel and Norman, 1992). Plusieurs études dans ce domaine ont impliqué l'analyse des données multispectrales Landsat MSS (**Multispectral Scanner**), Landsat TM (**Thematic Mapper**), SPOT-HRV (Haute Résolution Visible) ou NOAA-AVHRR (**Advanced Very High Resolution Radiometer**) en utilisant des techniques de traitement d'images. Le but visé consiste souvent à réduire les multiples bandes de données en une seule image transformée, dans laquelle la valeur d'un pixel prédit et mesure les caractéristiques du paysage. Les caractéristiques en question peuvent par exemple être : la biomasse, la productivité (photomasse), le LAI (**Leaf Area Index**), la quantité de radiation active photosynthétique

(PAR) consommée ou le pourcentage de couverture végétale du sol (Larsson, 1993). Cette section présente plusieurs algorithmes utilisés pour extraire de telles informations à partir de certaines données de télédétection appelées indices de végétation. La végétation verte saine réfléchit généralement entre 40% et 50% de l'énergie proche infrarouge incidente (0.7 à 1.1 μ m). La chlorophylle contenue dans les plantes absorbe approximativement entre 80% et 90% de l'énergie incidente dans la partie visible (0.4 à 0.7 μ m) du spectre (Jensen, 1983). La végétation morte ou sénescente réfléchit une plus grande quantité d'énergie que la végétation verte saine à travers le spectre du visible. Par contre, elle réfléchit moins que la végétation verte dans la région réfléchissante de l'infrarouge. Les sols nus ont généralement une réflectance plus élevée que la végétation verte et une réflectance plus faible que la végétation morte dans la région du visible, alors que dans le proche infrarouge, le sol nu a généralement une réflectance plus faible que la végétation verte ou sénescente. Plusieurs indices de végétation sont basés sur le fait qu'il y a des différences significatives dans la forme de ces trois courbes (courbes de réflectance dans le visible, l'infrarouge et le proche infrarouge). La télédétection ne peut offrir aucune information utile sur la condition de la végétation si ces trois courbes se superposent les unes sur les autres.

Le premier et le plus simple de ces indices de végétation est le DVI (**Difference Vegetation Index**) qui est la différence entre la réflectance du proche infrarouge et la réflectance rouge de la surface. Cet indice n'a pas été utilisé abondamment, mais le RVI (**Ratio Vegetation Index**) défini comme le rapport des réflectances dans le rouge et le proche infrarouge a été utilisé par plusieurs auteurs. Un des avantages de ce quotient par rapport au DVI est qu'il offre une normalisation partielle de certaines perturbations de mesures telles que les effets de calibration de l'instrument. Malgré le développement de plusieurs nouveaux indices adaptés aux comportements des sols, le NDVI (**Normalized Difference Vegetation Index**), proposé par Rouse et al. (1974), reste l'indice le plus utilisé. Kaufman et Tanré (1992) développent un nouvel indice de végétation, appelé ARVI (**Atmospherically Resistant Vegetation Index**). Celui-ci est une version améliorée de l'indice NDVI pour mieux raffiner la diffusion atmosphérique et il est calculé à partir des réflectances apparentes au capteur. Afin de minimiser à la fois l'effet du sol et l'effet de l'atmosphère, Pinty et Verstraete (1992) proposent un indice de végétation non linéaire pour une gestion globale de la végétation à partir d'images satellitaires : le GMI (**Global Environment Monitoring Index**). Un autre indice de végétation résistant aux effets atmosphériques et à la brillance du sol est défini par Plummer et al. (1994) : le **Angular Vegetation Index** (AVI). Cet indice utilise les trois bandes spectrales vert, rouge et proche infrarouge, ainsi que la longueur d'onde centrale de

chacune de ces bandes $(\lambda_{\text{bleu}}, \lambda_{\text{red}}, \lambda_{\text{nir}})$. La dépendance spectrale de cet indice se normalise par rapport à la bande centrale dans le rouge (λ_{red}) . Tous ces indices s'expriment par les relations ci-après (équations (8.17), (8.18), (8.19) et (8.20)).

$$\text{DVI} = \text{PIR} - \text{R} \quad (8.17)$$

$$\text{RVI} = \frac{\text{PIR}}{\text{R}} \quad (8.18)$$

$$\text{NDVI} = \frac{\text{PIR} - \text{R}}{\text{PIR} + \text{R}} \quad (8.19)$$

$$\text{PIR} = a\text{R} + b$$

$$\text{ARVI} = \frac{(\text{PIR}^* - \text{RB}^*)}{(\text{PIR}^* + \text{RB}^*)} \quad (8.20)$$

$$\text{avec } \text{RB}^* = \text{R}^* - \gamma(\text{B}^* - \text{R}^*) \text{ et } \gamma = \frac{\varphi_{a-b}}{\varphi_{a-b} - \varphi_{a-r}},$$

avec les définitions suivantes :

R : réflectance moyenne du canal rouge;

PIR : réflectance moyenne du canal proche infrarouge;

R* : réflectance apparente moyenne du canal rouge;

B* : réflectance moyenne apparente du canal bleu;

PIR* : réflectance moyenne apparente du canal proche infrarouge;

RB* : réflectance apparente combinée à partir des deux canaux bleu et rouge;

φ_{a-r} : réflectance intrinsèque de l'atmosphère dans le canal rouge;

φ_{a-b} : réflectance intrinsèque de l'atmosphère dans le canal bleu

μ : facteur d'autocorrélation atmosphérique qui dépend du type d'aérosol.

a : pente de la droite des sols nus dans l'espace R/PIR.

b : ordonnée à l'origine de la droite des sols nus dans l'espace R/PIR.

L : facteur d'ajustement égal à 0.5.

La droite des sols nus dans l'espace spectral apparent rouge-bleu et proche infrarouge s'obtient par l'expression (8.21) ci-après.

$$\text{PIR}^* = a_{\text{rb}}\text{RB}^* + b_{\text{rb}} \quad (8.21)$$

avec

a_{rb} : pente de la droite des sols nus dans l'espace RB^*/PIR^* ;

b_{rb} : ordonnée à l'origine de la droite des sols nus dans l'espace RB^*/PIR^* .

$$\text{GEMI} = \frac{\eta(1 - 0,25\eta) - (R - 0,125)}{1 - R} \quad (8.22a)$$

$$\eta = \frac{2(\text{PIR}^2 - R^2) + 1,5\text{PIR} + 0,5R}{\text{PIR} + R + 0,5}$$

$$\text{AVI} = \tan^{-1}\left(\frac{\lambda_{\text{blue}} - \lambda_{\text{red}}}{\lambda_{\text{red}}} \times [\text{PIR} - R]^{-1}\right) + \tan^{-1}\left(\frac{\lambda_{\text{blue}} - \lambda_{\text{red}}}{\lambda_{\text{red}}} \times [V - R]^{-1}\right) \quad (8.22b)$$

Le degré de résistance de l'indice ARVI dépend du succès de la détermination du paramètre d'autocorrélation atmosphérique. En se basant sur le modèle radiatif (modèle 5S), Kaufman et Tanré (1992) montrent que la valeur de l'expression (8.22b) permet un meilleur ajustement pour la plupart des applications en télédétection, à moins que le modèle des aérosols ne soit connu à priori.

8.4 L'analyse en composantes principales

Plusieurs études ont démontré l'utilité et l'importance de l'analyse en composantes principales (ACP) dans l'analyse des données multispectrales (Byrne et al., 1980; Eastman and Fulk, 1993; Gong, 1993; Wang et al., 1993).

Pour effectuer l'analyse en composantes principales, on applique une transformation linéaire à un ensemble de données multispectrales corrélées. L'application de la transformation aux données de télédétection corrélées résulte en un autre ensemble de données multispectrales non corrélées ayant certaines propriétés de variance ordonnées (Singh and Harrison, 1985). Mathématiquement, le but de l'analyse en composantes principales est de trouver une transformation linéaire $Y = \Phi^T X$ d'un espace d-dimensionnel vers un espace M-dimensionnel ($M < d$) par optimisation d'un critère d'erreur de représentation.

Soit $\{X^n; 1 \leq n \leq N\}$ un ensemble de N vecteurs de l'espace d'entrée d-dimensionnel que nous supposons euclidien. Tout vecteur X de cet espace peut s'écrire comme combinaison linéaire d'une famille de d vecteurs orthogonaux u_i (équation (8.23)).

$$X = \sum_{j=1}^d y_j u_j \quad (8.23)$$

Les vecteurs u_i vérifient la relation d'orthonormalité exprimée par l'équation (8.24).

$$u_i^T u_j = \delta_{ij} \quad (8.24)$$

Dans l'expression (8.24), δ_{ij} est le symbole de Kronecker. Les expressions explicites des coefficients y_i dans l'équation (8.23) peuvent être trouvées en utilisant l'équation (8.24), ce qui permet d'obtenir l'expression (8.25).

$$y_j = u_j^T X \quad (8.25)$$

La transformation exprimée par l'équation (8.25) peut être considérée comme une rotation du système de coordonnées des x originaux vers un nouvel ensemble de coordonnées donné par les y . Supposons maintenant qu'on ne retienne que M ($M < d$) vecteurs de base u_i , tel qu'on n'utilise que M coefficients y_i . Les coefficients restants doivent être remplacés par des constantes b_i telles que chaque vecteur X soit approché par une expression de la forme (8.26).

$$\tilde{X} = \sum_{j=1}^M y_j u_j + \sum_{j=M+1}^d b_j u_j \quad (8.26)$$

Ceci représente une forme de réduction de dimensionnalité, car le vecteur original X qui contient d degrés de liberté doit maintenant être approché par un nouveau vecteur Y ayant $M < d$ degrés de liberté. Considérons maintenant tout l'ensemble de données $\{X^n; 1 \leq n \leq N\}$. Nous voulons choisir les vecteurs de base u_i et les coefficients b_i tels que l'approximation donnée par l'équation (8.24) et les valeurs des y_i déterminées par l'équation (8.25) produisent la meilleure approximation du vecteur original X . L'erreur dans le vecteur X^n introduite par la réduction de la dimensionnalité est donnée par l'expression (8.27).

$$X^n - \tilde{X}^n = \sum_{j=M+1}^d (y_j^n - b_j) \cdot u_j \quad (8.27)$$

Nous pouvons alors définir la meilleure approximation comme celle qui minimise l'erreur quadratique sur tout l'ensemble de données. On cherche à minimiser l'erreur exprimée par l'équation (8.28) où la relation d'orthonormalité (équation (8.28)) a été utilisée.

$$E_M = \frac{1}{2} \sum_{n=1}^N \|X^n - \tilde{X}^n\|^2 = \frac{1}{2} \sum_{n=1}^N \sum_{j=M+1}^d (y_j^n - b_j)^2 \quad (8.28)$$

Si on annule la dérivée de E_M par rapport à b_i on trouve l'expression (8.29),

$$b_j = \frac{1}{N} \sum_{n=1}^N y_j^n = u_j^T \cdot \bar{X} \quad (8.29)$$

où on définit le vecteur moyen \bar{X} par l'expression (8.30).

$$\bar{X} = \frac{1}{N} \sum_{n=1}^N X^n \quad (8.30)$$

En utilisant les expressions (8.23) et (8.29), on peut exprimer l'erreur quadratique (équation (8.28)) par l'expression (8.31) ci-après,

$$E_M = \frac{1}{2} \sum_{j=M+1}^d \sum_{n=1}^N [\mathbf{u}_j^T (\mathbf{X}^n - \bar{\mathbf{X}})]^2 = \frac{1}{2} \sum_{j=M+1}^d (\mathbf{u}_j^T \cdot \Sigma \mathbf{u}_j) \quad (8.31)$$

où Σ est la matrice de covariance de l'ensemble de vecteurs \mathbf{x}^n et est donnée par l'expression (8.32).

$$\Sigma = \frac{1}{N-1} \sum_{n=1}^N (\mathbf{X}^n - \bar{\mathbf{X}})(\mathbf{X}^n - \bar{\mathbf{X}})^T \quad (8.32)$$

Il ne reste plus qu'à minimiser l'erreur quadratique E_M par rapport au choix des vecteurs de base \mathbf{u}_i . On montre que le minimum d'erreur est atteint lorsque les vecteurs de base vérifient la relation (8.33).

$$\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (8.33)$$

Ceci signifie que le minimum d'erreur est atteint lorsque les vecteurs de base sont égaux aux vecteurs propres de la matrice de covariance. On notera que la matrice de covariance étant réelle et symétrique, ses vecteurs propres peuvent être choisis orthonormaux pour satisfaire l'hypothèse de base. En substituant (8.33) dans (8.31) et en utilisant la relation d'orthonormalité (8.61), on obtient la valeur du critère d'erreur minimum sous la forme de l'expression (8.34) ci-après.

$$E_M = \frac{1}{2} \sum_{i=M+1}^d \lambda_i \quad (8.34)$$

Donc, l'erreur minimum est obtenue en choisissant les $d-M$ plus petites valeurs propres et leurs vecteurs propres comme ceux à éliminer. Chacun des vecteurs propres \mathbf{u}_i est appelé une composante principale.

Dans la pratique, l'algorithme précédent s'exécute de la manière suivante :

- 1) On calcule le vecteur moyen et la matrice de covariance des données originales, puis on détermine ses valeurs propres et les vecteurs propres correspondants;
- 2) Les vecteurs propres associés aux M plus grandes valeurs propres sont retenus et les vecteurs d'entrée \mathbf{X}^n sont projetés sur l'espace de ces vecteurs propres, pour donner les composantes des vecteurs transformés \mathbf{Y}^n dans l'espace M -dimensionnel.

L'erreur introduite par la réduction de la dimensionnalité en utilisant l'analyse en composantes principales peut être évaluée en utilisant l'expression (8.31). Dans certaines applications, les données originales ont une très grande dimensionnalité et on souhaite ne retenir que les toutes premières composantes. Dans ce cas, on peut utiliser des algorithmes plus efficaces en produisant uniquement les valeurs propres et les vecteurs propres requis.

En résumé, la transformation en composantes principales se calcule de la manière suivante, à partir des données originales :

- 1) Calculer la matrice de covariance, Cov, des données originales devant être transformées. L'utilisation directe de la matrice de covariance résulte en une analyse en composantes principales (ACP) non standardisée, tandis que l'utilisation de la matrice de corrélation résulte en une ACP standardisée (Eastman and Fulk, 1993).
- 2) Les valeurs propres $E = \{\lambda_1, \dots, \lambda_n\}$ (classées par ordre décroissant) et la matrice des vecteurs propres associés $EV = \{a_{kp}, 1 \leq k \leq n \text{ bandes}; 1 \leq p \leq n \text{ composantes}\}$ de la matrice de covariance sont calculées (équation (8.72)).

$$EV_{[n \times n]} COV_{[n \times n]} EV^T_{[n \times n]} = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \cdot & \cdot & \cdot \\ 0 & \cdot & \lambda_n \end{bmatrix} = \text{diag}\{\lambda_1, \dots, \lambda_n\} \quad (8.35)$$

Dans l'équation (8.69), EV^T est la transposée de la matrice des vecteurs propres EV et E est une matrice de covariance diagonale dont les éléments diagonaux λ_i , appelés **valeurs propres**, sont les variances des p composantes principales.

Les valeurs propres λ_i véhiculent une information importante. Par exemple, il est possible de déterminer le pourcentage $\%_p$ de la variance totale expliquée par chacune des composantes principales, en utilisant la relation (8.36) ci-après.

$$\%_p = \frac{\text{valeur propre } \lambda_p \times 100}{\sum_{k=1}^n \text{valeur propre } \lambda_k} \quad (8.36)$$

Dans l'expression (8.73), λ_p est la p-ième valeur propre.

Mais, quelle est la signification thématique de ces nouvelles composantes? Que représente par exemple la première composante? En calculant la corrélation de chaque bande k avec chaque composante p, il est possible de déterminer le degré de contribution de chaque bande dans chacune des composantes principales (voir expression (8.37)).

$$R_{kp} = \frac{a_{kp} \times \sqrt{\lambda_p}}{\sqrt{\text{Var}_k}} \quad (8.37)$$

où

a_{kp} est le poids de la bande k dans la composante p,

λ_p est la p-ième valeur propre,

Var_k est la variance de la bande k dans la matrice de covariance.

Les calculs précédents résultent en une nouvelle matrice d'ordre n dont les éléments représentent les degrés de contribution des différentes variables de base (bandes) dans chacune des composantes principales. L'analyse de cette matrice peut permettre d'identifier les variables les plus importantes permettant d'exprimer la variance totale des données à analyser. Maintenant que nous comprenons quelle information est contenue dans chaque composante principale, il est utile de voir à quoi ressemblent les images de ces composantes. Pour cela il est d'abord nécessaire d'identifier les valeurs de luminance ($\text{BV}_{i,j,k}$) associées à un pixel donné dans les différentes bandes. On applique ensuite la transformation appropriée à ces données tel qu'elles soient projetées sur les axes des premières composantes principales. De cette manière on trouvera la nouvelle valeur de luminance ($\text{newBV}_{i,j,p}$) du pixel dans la composante p en utilisant l'expression (8.38) ci-après.

$$\text{newBV}_{i,j,p} = \sum_{k=1}^n a_{kp} \text{BV}_{i,j,k} \quad (8.38)$$

où on a :

a_{kp} : vecteurs propres,

$\text{BV}_{i,j,k}$: valeur de luminance dans la bande k du pixel à la position (i,j),

n : nombre de bandes.

Cette procédure est exécutée pour chaque pixel de l'image originale pour produire progressivement les différentes images des composantes principales. Si les composantes 1, 2 et 3 comptent le plus dans la variance de l'ensemble de données, il est possible que les n bandes de données originales soient écartées et que le reste du processus d'analyse soit effectué en utilisant seulement ces trois images de composantes principales. Ceci réduit considérablement la quantité de données à analyser et simplifie le processus, souvent coûteux et consommateur en temps, de sélection des paramètres dans la classification des données de télédétection.

Eklundh et Singh (1993) et Eastman et Fulk (1993) ont démontré que l'ACP standardisée (basée sur le calcul de la matrice de corrélation) est plus appropriée que l'ACP non standardisée (calculée à partir de la matrice de covariance) dans l'analyse des changements des ensembles de données d'images multitemporelles. L'ACP standardisée force chaque bande à avoir un poids égal dans la dérivation de nouvelles composantes images. Elle est

identique à la conversion de toutes les valeurs de l'image à des valeurs standards (en soustrayant la moyenne et en divisant le résultat par l'écart-type). Eastman et Fulk (1993) ont traité 36 images mensuelles AVHRR dérivées du NDVI de l'Afrique pour les années 1986 à 1988. Ils ont trouvé que la première composante était toujours fortement corrélée avec le NDVI, indépendamment de la saison, alors que les deuxième, troisième et quatrième composantes sont reliées aux changements saisonniers du NDVI. De même, Eklundh et Singh (1993) ont calculé les composantes principales en utilisant quatre ensembles de données :

- 1) des composées mensuelles de maximum de valeur NOAA-NDVI;
- 2) des données NOAA-LAC;
- 3) des données Landsat-TM;
- 4) des données SPOT-HRV.

Les résultats obtenus révèlent des améliorations significatives du rapport signal sur bruit en utilisant la matrice de corrélation, en comparaison avec la matrice de covariance dans l'analyse en composantes principales, pour tous les ensembles de données.