

République Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique
DEPARTEMENT D'ELECTRONIQUE

Mémoire De Fin d'Etude De
MASTER PROFESSIONNEL
Spécialité : Electronique industriel

Thème

**Classification non supervisée de pixels
d'images couleur par analyse
d'histogrammes tridimensionnels**

Mémoire soutenu publiquement le : 26 /09 /2017 devant le jury composé de :

Présenté par : KAIDI DALIA.

Dirigé par: Mr. HAMAGUE.

Devant le jury:

PRESIDENT: Mr. MOHIA.

EXAMINATEUR: Mr. IDJERI.

Année universitaire 2016/2017.

Remerciements

Que Monsieur HAMEGUE, maitre assistant De l'Electronique à l'université « MOULOUD MAMERI de TIZI OUZOU », trouve ici le témoignage de ma profonde Reconnaissance. Ses encouragements, et surtout ses critiques, Sa sensibilisation, Largement contribué à l'accomplissement de mes travaux. Je le remercie infiniment De m'avoir toujours poussé vers l'avant. Je tiens également à remercier « TOUS » Les Messieurs et dames, mes professeurs qui m'ont enseigné durant deux ans de Formation master en Electronique industrielles, pour leurs précieux conseils et ses Orientations, Mes remerciements vont également aux membres du jury d'avoir Accepté d'évaluer mon travail.

Sans oublier de remercier mes amis et mes collègues (de l'université ou dans le Monde Virtual « internet ») qui, tous d'une manière différente, ont contribué à ce que Je puisse aboutir à la réalisation de ce mémoire. Enfin, merci à ma famille pour leur Soutien et l'encouragement qu'ils m'ont apporté tout au long de mon travail.

Sommaire

Introduction générale.....	01
----------------------------	----

Chapitre I : Généralités sur le traitement d'images

Introduction.....	02
1.1 - Définition de l'image.....	03
1.2 - Acquisition d'une image.....	03
1.3 - Caractéristiques d'une image numérique	04
1.3.1- Dimension.....	04
1.3.2- Résolution.....	04
1.3.3- Bruit.....	04
1.3.4- Histogramme.....	04
1.3.5 Luminance	05
1.3.4- Images en couleurs.....	06
1.4- Système de traitement d'images.....	06
1.5- Pré-traitement d'images	07
1.5.1- Modification d'histogramme.....	07
1.5.2- Réduction du bruit par filtrage.....	08
Conclusion.....	24

Chapitre II : Les images couleur

Introduction.....	25
2.1- Définition d'une image.....	25
2.2- Définition d'une image numérique.....	25
2.3- Les différents types d'images numériques.....	26
2.3.1- Les images noir et blanc (monochrome).....	26
2.3.2- Les images couleur.....	27
2.4- Caractérisation des images numériques.....	29

2.4.1- L’histogramme classique.....	29
2.4.2- L’histogramme compact.....	31
2.4.3- connexité dans les images numériques.....	32
Conclusion.....	33

Chapitre III : La segmentation d’images couleurs

Introduction.....	34
3.1- les différentes méthodes de segmentation.....	34
3.1.1- L’approche contour.....	35
3.1.2- L’approche région.....	36
3.1.2.1- la classification pixellaire.....	37
3.1.2.2- Croissance de régions.....	41
3.1.2.3- séparation.....	41
3.1.2.4- fusion.....	41
Conclusion.....	42

Chapitre IV :étiquetage en composants connexes d histogrammes multidimensionnels

Introduction	43
4.1- principe.....	43
4.1.1- cas d une image 2D.....	43
i- etiquetage séquentiel.....	43
ii-etiquetage récursif.....	45
4.1.2-cas d une image 3D.....	45
4.2-application des algorithmes d étiquetage en composantes connexes sur les histogrammes 2D et 3D.....	46
4.2.1-la plateforme Isaac	47
4.2.2-un module Isaac pour l étiquetage en composantes connexes	49

Conclusion	51
Conclusion générale	52
References bibliographies	

Introduction

A titre indicatif notons que près de 90% de l'information reçue par l'homme est visuelle. La production d'images de qualité, de même que leur traitement numérique (et si possible) automatique a donc une importance considérable. La plupart des appareils scientifiques fournissent des images (microscopes, télescopes, radiographies, Résonance magnéto-nucléaire, ...) et de nombreux domaines d'applications utilisent l'image comme source d'information et /ou de visualisation.

L'image est en fait une collection d'informations qui, dans un premier temps, se présentait sur un support photographique qui permettait le traitement en différé du phénomène fugace, une analyse fine des phénomènes enregistrés et bien sûr l'archivage et l'illustration. Le *traitement d'image* est né de l'idée et de la nécessité de remplacer l'observateur humain par la machine.

L'image ou les signaux provenant de capteurs ont alors été numérisés pour pouvoir être traités par l'ordinateur. Dans un deuxième temps, l'image a été codée et mémorisée sur différents supports.

L'une des étapes les plus importantes en traitement d'images est la segmentation qui est souvent décrite comme séparation fond / objets. Elle consiste à diviser l'image en plusieurs régions disjointes présentant des caractéristiques uniformes.

Plusieurs méthodes de segmentation d'images couleurs ont été proposées durant ces dix dernières années dont certaines sont basées sur l'analyse d'histogrammes unidimensionnels et multidimensionnels de l'image.

C'est dans ce cadre que se situe ce travail, dans lequel nous proposons une approche pour la segmentation des histogrammes multidimensionnels basée sur un étiquetage en composantes connexes.

Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un Développement important depuis quelques dizaines d'années.

On désigne par traitement d'images numériques l'ensemble des techniques permettant De modifier une image numérique afin d'améliorer ou d'en extraire des informations.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant Sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus Agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées Pertinentes.

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des Techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différents techniques connues dans ce domaine.

1. Définition de l'image

Une image est plutôt difficile à décrire d'une façon générale. Une image est une représentation du monde. En traitement d'image, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de $R \times R$ dans R où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique. Il arrive cependant que l'image soit dite "3D" donc la fonction est de $R \times R \times R$ dans R . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de $R \times R$ dans $R \times R \times R$ [wikipedia 04].

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter[chakib 99].

2. Acquisition d'une image

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales [chakib 99] [Bénallal 03]:

- les caméras numériques ,
- et les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en

informations binaires qui seront stockées dans un fichier[chakib 99]. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'intercorrélation de deux éléments voisins : plus cette intercorrélation est faible, meilleure est l'image[wikipedia 04].

3 Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

3.1 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image[chakib 99].

3.2 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution [chakib 99].

3.3 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [chakib 99].

3.4 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant [chakib 99] [Pillou 04].

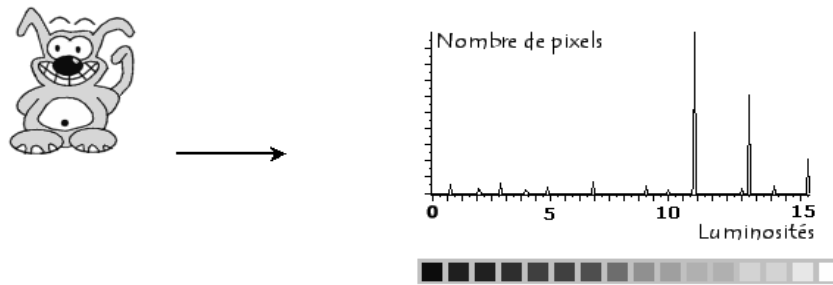


Figure1.1 Histogramme et palette associés à une image [Pillou 04]

3.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par [chakib 99] [Nicolas 01] :

- Des images lumineuses (brillantes);
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

3.6 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport [chakib 99] :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

3.7 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [chakib 99].

3.8 Images en couleurs

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.) [chakib 99].

4 Système de traitement d'images

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

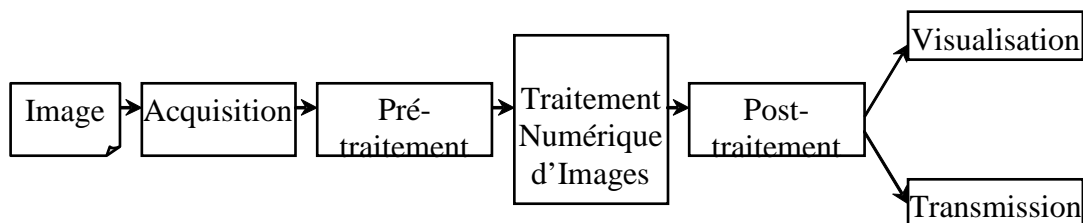


Figure1.2 Schéma d'un système de traitement d'images[chakib 99]

5 Pré-traitement d'images

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation.

Le temps de traitement (temps CPU) est très important. C'est un facteur décisif, il doit être le plus petit possible. Ceci implique que les opérateurs doivent être locaux, c'est-à-dire qu'ils doivent intervenir sur un nombre limité de pixels et essentiellement sur des pixels dans le voisinage proche du pixel courant.

Les techniques de pré-traitements les plus courantes qu'on va présenter sont [Bensrhair02,a] [Cocquerez 95,+]:

- ↳ La modification d'histogramme,
- ↳ La réduction du bruit par filtrage,

5.1 Modification d'histogramme

On cherche à améliorer l'image en appliquant une transformation ponctuelle d'intensité; à tout pixel $a(x,y)$ on associe une intensité par la transformation T croissante de façon à conserver des contrastes entre régions (claire sur fond sombre, apparaît plus claire après transformation). Le pixel après transformation est le pixel $a'_s(x,y)$:

$$a'_s(x,y) = T(a_s(x,y))$$

Dans ce qui suit, on va présenter deux types de modification d'histogramme:

- Expansion de dynamique
- Egalisation d'histogramme

Expansion de dynamique

Cette méthode consiste à utiliser au mieux l'échelle des niveaux de gris disponible sur le système d'acquisition d'images.

Soit $A[i,j]$ l'image de départ, $A'[i,j]$ image après transformation et $[a_0, a_1]$ l'intervalle des intensités présentes dans l'image et soit $[a_{min}, a_{max}]$ l'intervalle disponible.

L'expansion du dynamique correspond à la transformation: $a'_s = \alpha + \beta a_s$ telle que:

$$\forall a \in [a_0, a_1] \quad a_s \xrightarrow{T} a'_s \in [a_{min}, a_{max}]$$

$$\alpha = \frac{a_{min} - a_1 - a_{max} a_0}{a_1 - a_0}, \quad \beta = \frac{a_{max} - a_{min}}{a_1 - a_0}$$

7

Egalisation d'histogramme

Cette transformation consiste à rendre le plus plat possible l'histogramme de niveaux de gris de l'image. L'image $A[i,j]$ est considérée comme un ensemble de réalisation d'une variable aléatoire A , admettant une densité de probabilité $f(a)$ et une fonction de répartition $F(a)$, f représente l'histogramme normalisé de $A[i,j]$.

On cherche une transformation T continue dérivable par morceaux et strictement croissante telle que la variable aléatoire $B=T(A)$ soit uniformément répartie dans l'intervalle $[b_{\min}, b_{\max}]$.

Soit $g(b)$ la densité de probabilité de B , et T' la transformation dérivée de T :

$$g(b) = \begin{cases} f(a) \cdot \frac{1}{T'(a)} & \text{si } b_{\min} < b < b_{\max} \\ 0 & \text{ailleurs} \end{cases}$$

$$T'(a) = (b_{\max} - b_{\min}) \cdot f(a) \quad , \quad b = T(a) \quad , \quad a \in [a_{\min}, a_{\max}]$$

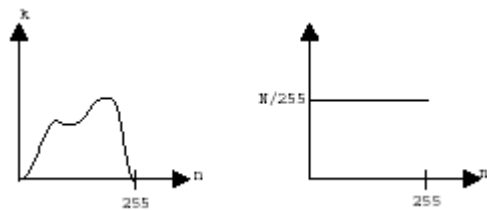


Figure1.3 Egalisation d'histogramme [Bensrhair 02,a]

5.2 Réduction du bruit par filtrage

Le bruit peut provenir de différentes causes :

- Environnement lors de l'acquisition,
- Qualité du capteur,
- Qualité de l'échantillonnage.

En général, le bruit d'images est considéré comme un champ aléatoire caractérisé par sa densité de probabilité f et sa fonction de répartition F . La nature plus ou moins impulsionnelle du bruit peut être décrite à l'aide de densité de probabilités de la forme [Cocquerez 95]:

$$f(a) = C \cdot \exp(-K|a|^{\alpha})$$

Plusieurs filtres sont utilisés pour la réduction de bruit, ils sont divisés en deux catégories :

- Filtres linéaires stationnaires ,
- Filtres non linéaires stationnaires.

La première catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).

Voici quelques exemples de filtres réducteurs de bruit :

- ☞ Filtre moyenneur
- ☞ Filtre gaussien
- ☞ Filtre exponentiel de Shen

Le principale inconvénient des filtres linéaire est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Ce problème peut être surmonté par l'utilisation des filtres non linéaires.

Le domaine de filtrage non linéaire est extrêmement vaste. Les opérateurs les plus courants en pré-traitement d'images sont :

- Les filtres d'ordre,
- Les filtres homomorphique,
- Les filtres morphologique.

6 Extraction de contours

La détection de contour est une étape préliminaire à de nombreuses applications de l'analyse d'images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- discontinuités de la fonction de réflectance (texture, ombre),
- discontinuités de profondeur (bords de l'objet),

et sont caractérisés par des discontinuités de la fonction d'intensité dans les images.



Figure 1.4 Différents types de contours : marche, toit et pointe [Bensrhair 02,b]

Le principe classique de la détection de contours repose sur l'étude des dérivées de la fonction d'intensité dans l'image : les extréma locaux du gradient de la fonction d'intensité et les passages par zéro du laplacien.

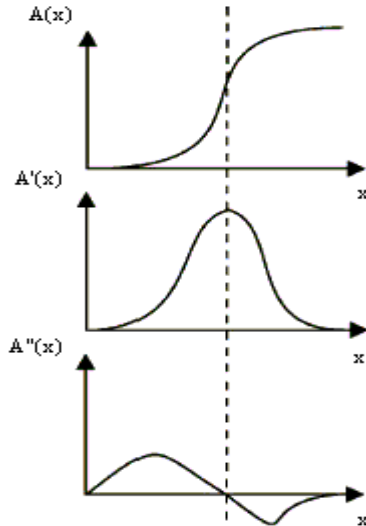


Figure 1.5 Cas bidimensionnel [Bensrhair 02,b]

6.1 Approche Gradient

La recherche de la transition peut être réalisée en recherchant les maxima locaux du gradient. Dans le cas bidimensionnel, le vecteur gradient est défini au point M(x, y) par :

$$\vec{\nabla} A(x, y) = \left(\frac{\partial A}{\partial x}, \frac{\partial A}{\partial y} \right)$$

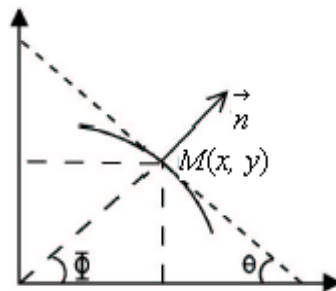


Figure 3.6 Orientation du contour [Bensrhair 02,b]

$\vec{\nabla} A(x, y)$ est normal à la courbe de niveau donnée par $A(x, y) = cste$ qui passe par M.

Au point M(x, y) l'orientation Φ du gradient est donnée par : $\Phi = \arctan \left(\frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}} \right)$

et le vecteur unitaire \vec{n} normal au contour : $\vec{n} = (\cos \Phi, \sin \Phi)$

L'orientation du contour a pour expression : $\theta = \frac{\pi}{2} - \Phi$

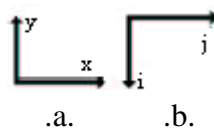


Figure 1.7 Orientation des axes. a) cas continu. b) cas discret

Pour une image numérique, on a :

$$\frac{\partial A}{\partial y} \approx \frac{\Delta A}{\Delta i} = \frac{A[i+1, j] - A[i, j]}{\Delta i}$$

$$\frac{\partial A}{\partial x} \approx \frac{\Delta A}{\Delta j} = \frac{A[i, j+1] - A[i, j]}{\Delta j}$$

La norme du gradient est :

$$|\nabla A[i, j]| = \sqrt{A_j^2[i, j] + A_i^2[i, j]}$$

ou :

$$|\nabla A[i, j]| = \max\{|A_j[i, j]|, |A_i[i, j]|\}$$

ou bien

$$|\nabla A[i, j]| = |A_j[j, j]| + |A_i[i, j]|$$

Opérateurs Prewitt et Sobel

$$A_j[i, j] = h_j * A[i, j]$$

$$A_i[i, j] = h_i * A[i, j]$$

$$h_j = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix}$$

$$h_i = \begin{pmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{pmatrix}$$

Les matrices h_j et h_i sont appelées masques. Elles sont les noyaux de convolution de filtres à réponse impulsionnelle finie. Pour Prewitt, on prend $c=1$. Pour Sobel, on prend $c=2$.

Opérateur de Kirsh

C'est un opérateur à 8 masques correspondant chacun à une direction préférentielle et obtenue par rotation de $\frac{\pi}{4}$ de l'opérateur de base h . (la numérotation des masques est faite dans l'ordre des directions de Freeman en 8-connexité).

$$h_0 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

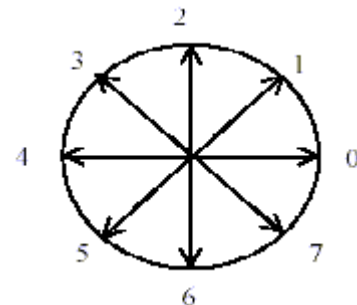


Figure 1.8 Les directions de Freeman d'après [Cocquerez 95]

6.2 Approche Laplacien

La recherche des contours dans l'image $A(x, y)$ peut aussi être réalisée en recherchant les passages par zéro du laplacien. On dispose donc, de la valeur du laplacien en chaque point de l'image soit la fonction Δ .

$$\Delta A(x, y) = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2}$$

On considère que les points de contours sont localisés aux passages par zéro de laplacien. Si le calcul du laplacien était exact il suffirait de sélectionner les points M tels que $\Delta(M)=0$.

3.6.3 Approche par filtrage optimal : Approche de Canny

Filtre optimal de réponse impulsionnelle $h(x)$ qui satisfait les trois contraintes suivantes :

- ↪ Bonne détection
- ↪ Bonne localisation
- ↪ Réponse unique

Soit $A(x)$ un signal mono-dimensionnel représentant un saut d'amplitude U_0 noyé dans un bruit blanc stationnaire $N(x)$ de moyenne nulle et de densité spectrale N_0 .

$$A(x) = U_0 U(x) + N(x)$$

Le signal de sortie est :

$$C(x) = A * h(x) = \int_{-\infty}^{\infty} A(t)h(x-t)dt$$

Bonne détection : Le rapport RSB (Rapport Signal/Bruit) doit être maximisé.

$$RSB = \frac{U_0 \int_0^{\infty} h(x-t)dt}{\left[E \left(\left| \int_{-\infty}^{\infty} N(t)h(x-t)dt \right|^2 \right) \right]^{1/2}} = \frac{U_0 \int_0^{\infty} h(x-t)dt}{N_0 \left[\int_{-\infty}^{\infty} h^2(t)dt \right]^{1/2}}$$

Bonne localisation : Les points détectés doivent être aussi près que possible du centre du contour véritable (à maximiser)

$$\frac{U_0 |h'(0)|}{N_0 \left[\int_{-\infty}^{\infty} h^2(t)dt \right]^{1/2}}$$

Unicité de la réponse : On utilise le critère de Canny. On veut minimiser la densité d_0 des passages par 0 de la réponse du bruit.

Critère généralisé :

$$d_G = \frac{1}{\pi} \left[\frac{\int_{-\infty}^{\infty} h'^2(t) dt}{\int_{-\infty}^{\infty} h^2(t) dt} \right]^{\frac{1}{2}}$$

Critère de Canny :

$$d_G = \frac{1}{\pi} \left[\frac{\int_{-M}^M h'^2(t) dt}{\int_{-M}^M h^2(t) dt} \right]^{\frac{1}{2}}$$

$h(x)$ est un RIF défini sur l'intervalle $[-M, M]$ Canny a défini les conditions aux limites suivantes :

- $h(0) = 0$
- $h(M) = 0$
- $h'(0) = S$ (pente à l'origine)
- $h'(M) = 0$

1.7 Segmentation en régions

Fondamentalement, la segmentation est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur. L'union de ces régions doit redonner l'image initiale

On regroupe généralement les algorithmes de segmentation en trois grandes classes (voir [Baillie 03]) :

- Segmentation basée sur les pixels
- Segmentation basée sur les régions
- Segmentation basée sur les contours

La première catégorie travaille sur des histogrammes de l'image. Par seuillage, clustering ou clustering flou, l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image. La segmentation est implicite puisqu'on suppose que chaque cluster de l'histogramme correspond à une région dans l'image. En pratique, ce n'est pas forcément le cas et il faut séparer les régions de l'image qui sont disjointes bien qu'appartenant au même cluster de couleur. Ces algorithmes sont assez proches des algorithmes de réduction de couleur.

La deuxième catégorie correspond aux algorithmes d'accroissement ou de découpage de région. L'accroissement de région est une méthode bottom-up : on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et on regroupe les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus

possible. Le découpage de région est le pendant top-down des méthodes d'accroissement : on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes.

La troisième catégorie s'intéresse aux contours des objets dans l'image. La plupart de ces algorithmes sont locaux, c'est à dire fonctionnent au niveau du pixel.

Les algorithmes que nous allons présenter sont :

- Accroissement de région fonctionnant au niveau du pixel.
- Split and merge classique.

Des études détaillées de ces algorithmes et d'autres sont données dans [Baillie 03].

7.1 Accroissement de région

Les méthodes d'accroissement de régions sont les méthodes de segmentation les plus simples. Le principe est basé sur une approche bottom-up : l'algorithme part de petits éléments de l'image qu'il va tenter de regrouper en éléments plus importants.

Nous présentons ici la version de base de l'algorithme d'accroissement de région qui fonctionne en agrégeant des pixels. Supposons une région de couleur homogène R . Initialement, $R = 1$ pixel. On va étendre la région R en incluant les pixels situés sur la frontière et dont la couleur est proche de celle de R (la variation de couleur est inférieure à un seuil δ caractéristique de ce type d'algorithmes). En répétant cette procédure jusqu'à ce qu'il n'y ait plus de pixels de couleur assez proche sur la frontière, on obtient une région de couleur homogène maximale autour du pixel de départ. La région initiale "gonfle" en absorbant des pixels de la frontière, jusqu'à stabilité par rapport à la propriété d'homogénéité. Afin d'assurer un critère de connexité, on utilise les voisins en 4-connexités pour l'accroissement de région et la 8-connexités pour l'accroissement de contour (cas de détection de contours).

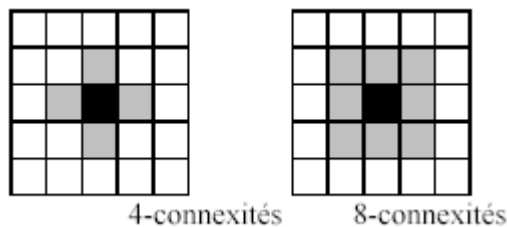


Figure 1.9 Voisinage d'un pixel [Achard 03]

Cette méthode présente deux limitations:

- Les régions obtenues dépendent fortement des pixels d'amorçage choisis et de l'ordre dans lequel les pixels de la frontière sont examinés.
- Le résultat final est très sensible à la valeur du seuil δ .

Cet algorithme fait parti de la classe d'algorithmes de segmentation dit "locaux". L'opération élémentaire consiste à manipuler des pixels adjacents et l'algorithme n'a aucune vision globale du résultat qu'il obtient.

3.7.2 Méthodes du type "split and merge"[Baillie 03] [Duchesnay 01,a]

L'algorithme "split and merge" a été présenté la première fois en 1974 par Pavlidis et Horowitz [Baillie 03]. Cet algorithme s'apparente dans son principe à l'algorithme d'accroissement de région que nous venons de présenter. La différence principale provient de la nature des régions élémentaires agrégées. Dans l'algorithme "split and merge", les régions agrégées proviennent d'une première phase (split) de traitement de l'image qui construit de manière récursive des régions carrées de taille variable mais homogènes.

☞ Split

La méthode de découpage de l'image utilisée dans cet algorithme est basée sur la notion de "quadtree". Cette structure de données est un arbre quaternaire qui permet de stocker l'image à plusieurs niveaux de résolution. On part d'une région initiale qui est l'image tout entière. Si cette image vérifie un critère d'homogénéité de couleur, l'algorithme s'arrête. Sinon, on découpe cette région en quatre parties de même taille et on lance la procédure récursivement dans ces quatre parties. La région initiale va être stockée comme un nœud dans un graphe et les sous parties comme des fils de ce nœud. La figure 3.10 montre une image en noir et blanc 8x8 et le découpage correspondant à chaque niveau, ainsi que la structure d'arbre associée à ce découpage.

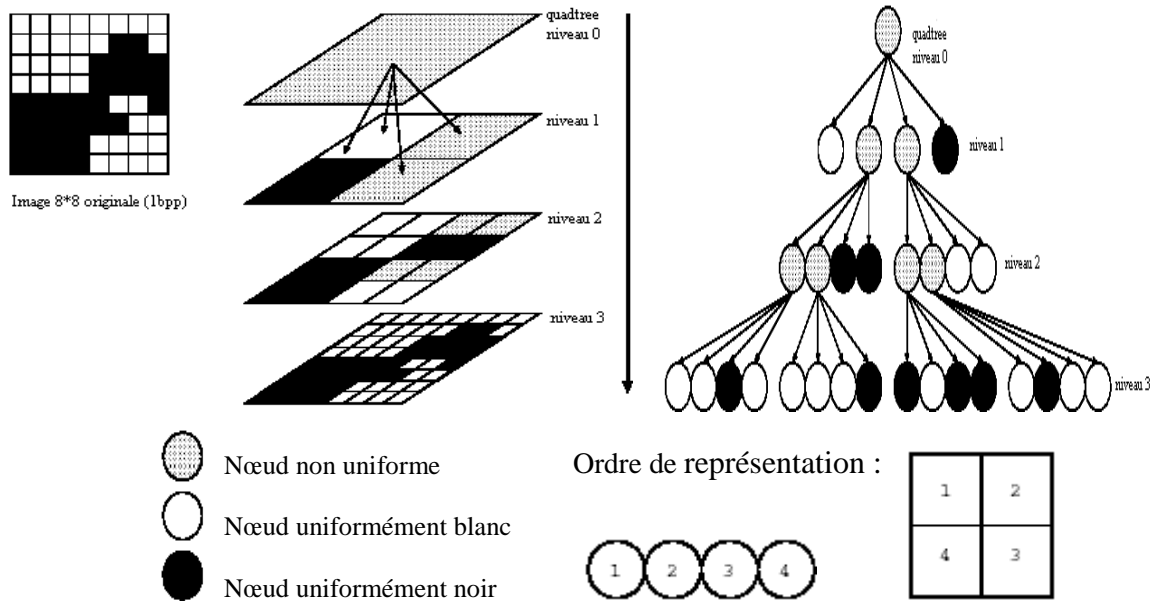


Figure 1.10 Découpage par quadtree d'une image 8x8. Traitement avec seuil=100% [Baillie 03]

☞ Merge

La procédure de découpage décrite précédemment aboutit à un nombre de régions trop élevé. La cause fondamentale de cette sur-segmentation est que l'algorithme découpe les régions de manière arbitraire. Il se peut qu'il coupe de cette façon une zone homogène en deux ou quatre parties (Figure 3.11).

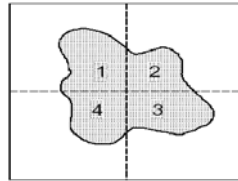


Figure 1.11 Problème de découpage arbitraire de régions dans la phase de "split"[Baillie 03]

Les parties 1,2,3 et 4 de la (Figure 3.11) par exemple, appartiennent à des branches différentes du quadtree. Elles sont donc considérées comme des régions différentes bien que leur couleur soit identique.

La solution, qui correspond à la phase "merge" de l'algorithme, est de procéder à une fusion de régions après le découpage. L'implémentation la plus simple de cette fusion cherche tous les couples de régions adjacentes dans l'arbre issu du découpage et cherche à les fusionner si leur couleur est assez proche.

1.8 Approches coopératives en traitement d'images

La résolution collective de problème se préoccupe de développer des systèmes d'agents simples capables de résoudre collectivement un problème complexe ; par exemple construire un chemin reliant tous les nœuds d'un graphe et dont la longueur est minimale (problème dit du voyageur de commerce ou TSP).

La question principale qui se pose alors est de déterminer le système de représentation de l'environnement, les comportements individuels, et la dynamique qui va régir le fonctionnement du système afin qu'il produise une réponse au problème qui lui est posé.

Plusieurs modèles inspirés des insectes sociaux ont déjà été élaborés et utilisés pour effectuer de la résolution de problème. L'une des applications de ces modèles est le traitement d'images.

Dans ce type d'approches, l'image y est vue comme un environnement ou territoire analysé par une société d'agents (ou processus) distribués spatialement. Cette approche est novatrice car elle va à l'encontre de la conception classique centralisée et planifiée des tâches d'analyse d'images. Ces architectures mettent en relief la distribution de calcul, l'adaptation

locale des traitements et la richesse des coopérations possibles entre les aspects contour et région.

Dans ce qui suit, on va présenter quelques approches appliquées aux traitement d'images.

8.1 Système de Bellet

Bellet [Bellet 98] a développé un système coopératif pour la segmentation de bas niveau des images. Ce système est multi-processus, puisqu'une image est segmentée par une multitude de processus indépendants, qui sont basés sur des modèles de croissance de région ou de suivi de contour. Chaque processus est initialisé à un endroit précis de l'image, avec un germe, et il fait croître sa primitive à partir de cet endroit. La croissance (région ou contour) s'effectue par une évaluation multi-critères de pixels avoisinants et sélection des meilleurs candidats (figure 3.12). (le lecteur intéressé peut se référer à [Bellet98]).

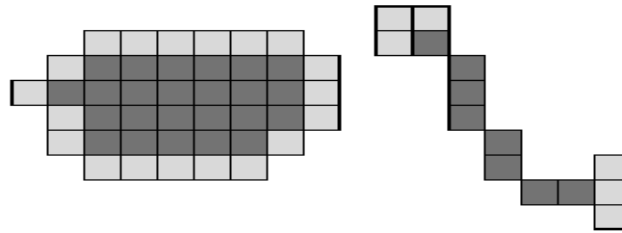


Figure 1.12 Principes de la croissance de région et du suivi de contour. Pour faire croître la primitive (pixels foncés), différents candidats sont possibles (pixels clairs). Le pixel ayant la meilleure évaluation sera ajouté à la primitive.

Ce système est basé sur une nouvelle forme de coopération entre différents processus de segmentation de natures différentes. Lorsqu'un processus manque d'informations pour mener à bien les décisions qu'il doit prendre, il crée de nouveaux processus, qualifiés de fils pour segmenter de nouvelles primitives et ainsi, récolter plus d'informations sur l'environnement local. Un processus peut également s'appuyer sur les informations récoltées par ses voisins pour mener à bien sa tâche. C'est sur ce modèle de coopération que s'effectue toute la segmentation de l'image.

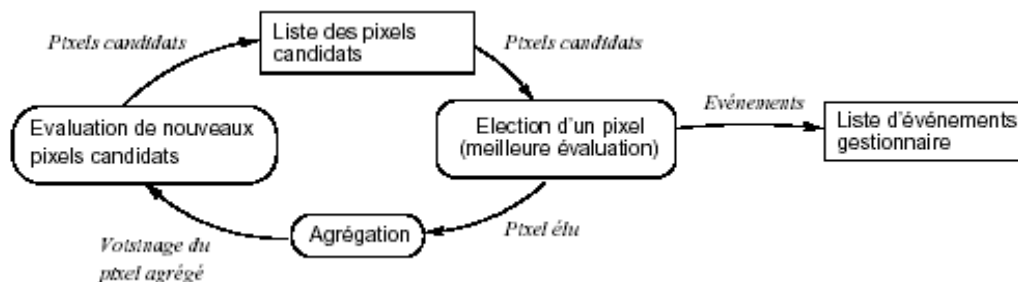


Figure 1.13 Boucle de contrôle du comportement de perception. Ce comportement boucle sur une liste de pixels tant que celle-ci n'est pas vide. [Boucher 99]

Le système est généraliste et ne bénéficie d'aucune connaissance sur les applications et les images qu'il segmente. Il a été utilisé pour différents types d'images.

8.2 Approche de Boucher

Alain Boucher [Boucher 99] étend les processus de Bellet [Bellet 98] à des agents permettant de bien identifier les structures internes aux agents et leurs interactions. Les agents sont spécialisés dans la segmentation et l'interprétation d'images cytologiques; un des aspects les plus intéressants concerne le traitement de séquences d'images et la manière de lancer des agents dans l'image suivante de la séquence, en fonction de ce qui est en train de se dérouler dans l'image courante. Les agents disposent de quatre comportements :

- Le comportement de perception explore son environnement et sélectionne des pixels candidats à l'agrégation.
- Le comportement d'interaction gère les fusions et les échanges de pixels entre agents.
- Le comportement de différenciation correspond à une interprétation à valeur sémantique. Les agents représentent au départ des primitives région ou contour ; les informations accumulées sur eux-même et dans un environnement commun vont leur permettre de déterminer la nature sémantique de la primitive traitée.
- Le comportement de reproduction, permet de créer au besoin des agents à certains endroits de l'image. Il conditionne l'exploration de cette image et donc le contrôle de système.

8.3 Approche de Duchesnay

Duchesnay [Duchesnay 01] s'appuie sur la structure de pyramide irrégulière (voir la figure 3.14) pour gérer le processus de fusion de régions et assurer la convergence de la segmentation; une coopération région-région assez sophistiquée est mise en œuvre, mais qui ne tire pas suffisamment parti de l'information contour. Un des aspects intéressants de son approche est l'utilisation d'une procédure de décimation (récursive) pour le passage du niveau k au niveau $k+1$. La pyramide se construit en partant de la base qui représente l'image pré segmentée (par exemple avec l'algorithme des quadrees) jusqu'au dernier niveau de la pyramide comportant le minimum d'information. Les niveaux de cette pyramide sont des graphes d'adjacence de régions.

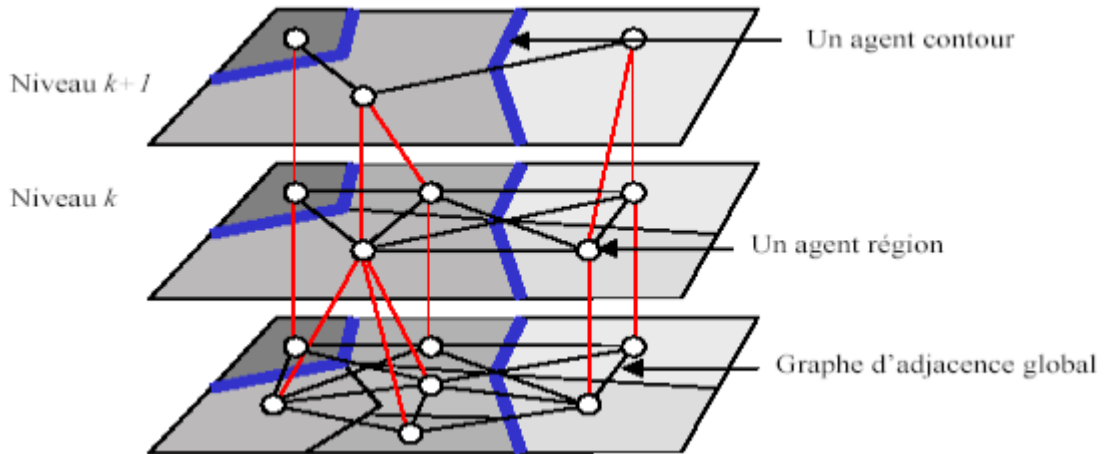


Figure1.14 La pyramide de graphe d'adjacence des régions d'après [Duchesnay 01]

8.4 Approche de Liu & Tang : Segmentation par « agents migrants »

Dans cette approche [Liu 99,+], des agents répartis aléatoirement dans l'image vont migrer (se diffuser) de pixel en pixel jusqu'à ce qu'ils trouvent dans leur environnement local un motif recherché. Une fois fixé sur le pixel, l'agent l'étiquette puis se reproduit en initialisant de nouveaux agents dans son voisinage et devient inactif.

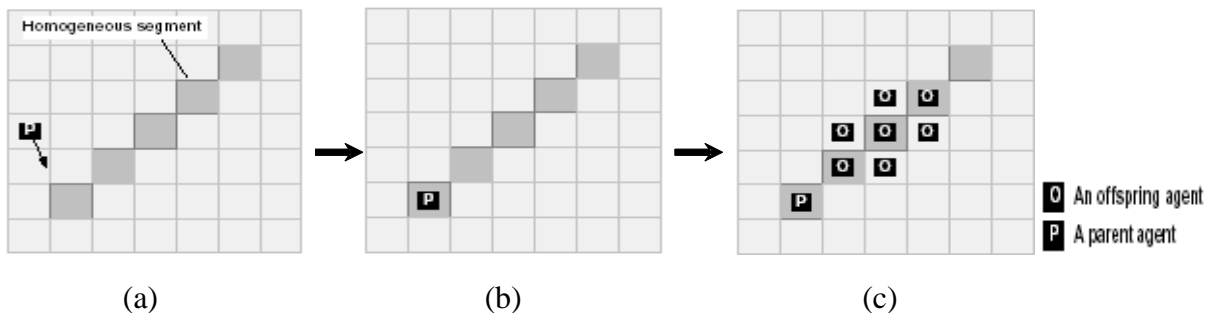


Figure1.15 Comportement de reproduction. (a) Au temps t , un agent, indiqué par le symbole 'P', est prêt à déplacer selon une direction. (b) Au temps $t+1$, l'agent est déplacé à un pixel de segment homogène. (c) Au temps $t+2$, l'agent s'auto-produit un nombre d'agents descendants, indiqués par le symbole 'O', en une direction spécifique calculée à partir de son vecteur propre. [Liu 99,+]

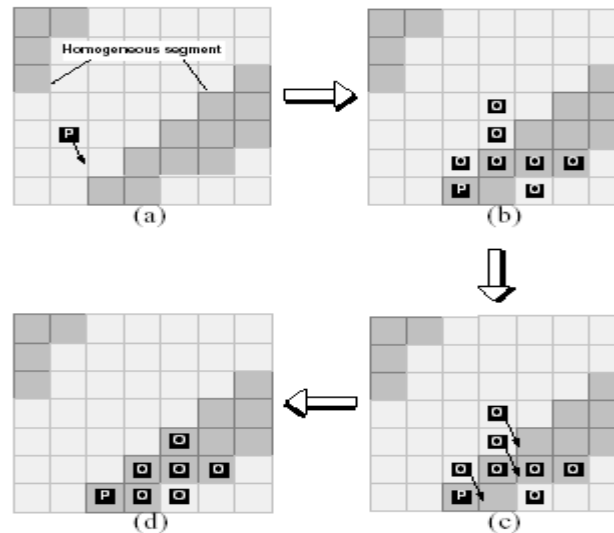


Figure 1.16 Comportement de diffusion (a) Au temps t , un agent 'P', est prêt à se déplacer à un emplacement voisin dans une direction comme indiqué. (b) Dans le temps $t+1$ et $t+2$, l'agent rencontre un pixel de segment homogène et par la suite, il s'auto-reproduit un nombre d'agents descendants 'O'. (c) Les agents descendants 'O' dans des régions non homogènes sont prêts à entreprendre des mouvements de diffusion dans leurs régions voisines, dans une direction comme héritée de leur parent et les agents de mêmes parents. Les agents qui se trouvent aux pixels de segment homogène, produiront plus d'agents descendants (bien qu'il ne soit pas montré dans la figure). (d) Au temps suivant, certains des agents se répandant comme indiqué dans (c) rencontreront plus de pixels de segment homogène. [Liu 99 ,+]

Au cours de sa migration, l'agent va vieillir et peut mourir si le nombre de cycles de recherche excède son temps de vie. Les auteurs ont aussi introduit la notion d'évolution associée aux directions de diffusion et de reproduction. En effet, en cas de succès d'un des comportements dans une direction, la pondération associée à cette direction sera renforcée. Ainsi les comportements de migration et reproduction de ses descendants favoriseront cette direction.

Les auteurs appliquent leur approche à la détection de segment de contour ; les agents migrent jusqu'à trouver un contour, ils s'y fixent, l'étiquettent, et initialisent de nouveaux agents qui auront de fortes chances de trouver un contour recherché dans un voisinage proche.

8.5 Une plate-forme pour la segmentation des images IRM

L'objectif de cette plate-forme [Settache 02] multi-agent est la segmentation d'images appliquée aux IRM cérébrales 2D. Dans ce travail, le parallélisme de segmentation permet d'envisager de réaliser la fusion simultanée de couples de régions voisines quelle que soit leur position dans l'image, et quels que soient les critères de candidature à la fusion utilisés.

Dans [Settache 02], Il existe différents protocoles de coopération :

- Coopération région-région pour la fusion de régions.
- Coopération contour-région pour la fusion de régions : il ne doit pas exister de contour entre les deux régions qui souhaitent fusionner
- Coopération contour-région pour la division de région : un contour qui traverse une région peut inciter cette région à se diviser.
- Coopération contour-contour pour le raccordement de contours.
- Coopération région-contour pour le raccordement de contours.
- Coopération région-contour pour le prolongement de contours : on peut prolonger un contour à condition que les régions voisines ne s'y opposent pas. Par exemple, si le contour suit un morceau de la frontière entre deux régions, on pourra le prolonger le long de la frontière.

La figure 1.17 illustre la coopération région-région et région contour pour la fusion de régions. Au départ toutes les régions calculent leur désir de fusion avec chacun de leur voisins et ce désir sont communiqués au meilleur voisin par un message (par exemple une flèche sur la figure 3.17a entre R10 et R110). Une opération de fusion est exécutée quand le désir de fusion est mutuel (figure 3.17.b entre R160 et R122) et la région se met alors dans un état d'attente jusqu'à exécution de la fusion de son meilleur voisin (par exemple la région R110 dans la figure 3.17.b). Comme on peut voir sur la figure 3.17.d la fusion des deux régions R210-R130 est stoppée par l'agent contour qui se situe aux frontières de ces deux régions.

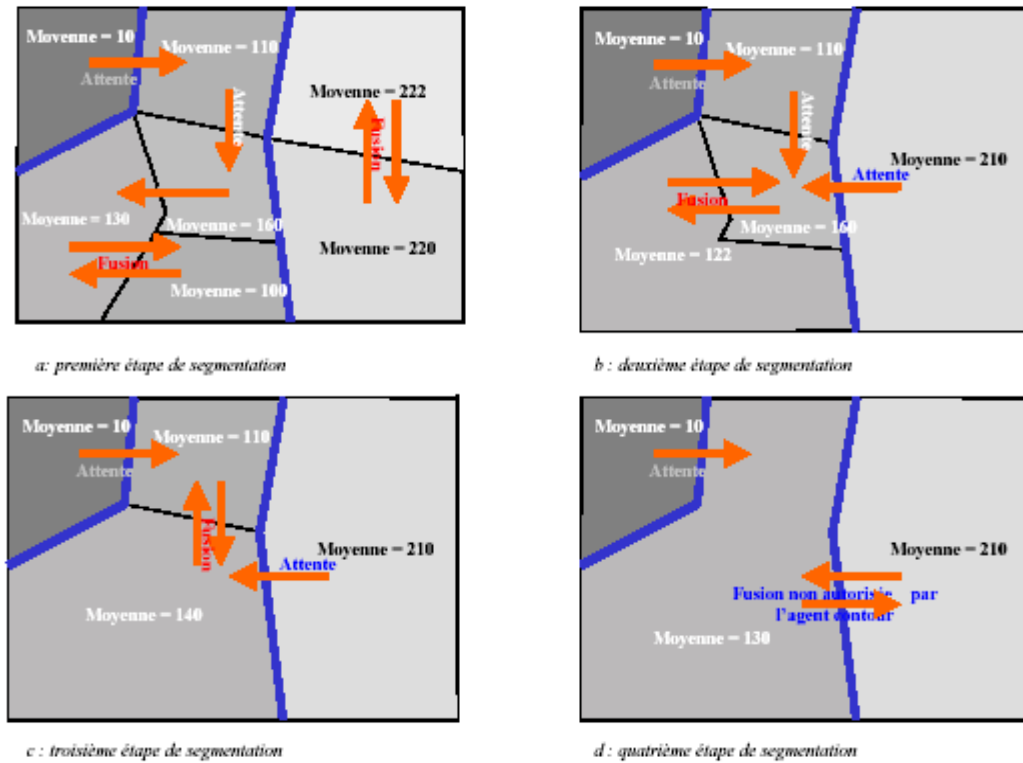


Figure 1.17 Exemple de coopération région-contour et région-région pour la fusion des régions [Settache 02]

8.6 Détection de régions dans des images à niveaux de gris : l'inspiration des araignées sociales

Ce travail [Bourjot 01,+] a exposé une démarche de transposition pour passer d'un modèle de simulation à un modèle de résolution. Le cadre biologique initial concerne un processus de construction collective de toile chez les araignées sociales. Sa transposition concerne la détection de régions dans des images à niveau de gris.

L'environnement est un carré de $N \times N$ piquets dans lequel évoluent les araignées et correspond à l'image à analyser (figure 3.18.a). Un piquet correspond à un ou plusieurs pixels de l'image d'origine, selon l'échantillonnage réalisé. La hauteur d'un piquet correspond au niveau de gris du ou des pixels correspondants (figure 3.18.b). Il peut donc y avoir 256 hauteurs différentes. Initialement, il n'y a pas de fils dans l'environnement, ils apparaîtront au fur et à mesure du déplacement des araignées.

Les items comportementaux d'une araignée sont le déplacement, la pose d'un fil et, en plus, un comportement de retour sur la toile. L'araignée est dotée d'un état interne pour permettre une pose contextuelle.

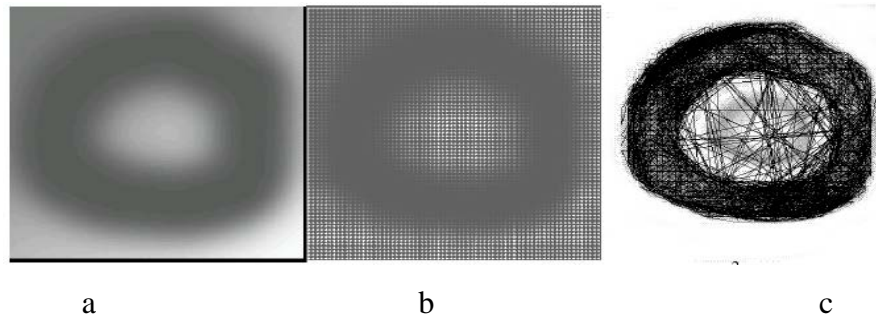


Figure 1.18 Exemple de résultat, d'après [Bourjot01+]

La dynamique du système est la même que dans le cas de la simulation et est de type stigmergique. Ce processus construit des toiles qui sont interprétées comme des régions de l'image.

8.7 Traitement d'images par vie artificielle

Carden [Carden 02] propose une approche de détection des contours des régions dans une image simple en niveaux de gris par vie artificielle. Ce travail est basé sur les travaux de Reynolds et MacGill [MacGill 00] où les agents représentent les « géoboids », qui permettent de calculer le gradient.

Les agents ont différents comportements de direction trois inspirés de Boids de Reynolds : (voir figure 3.19) cohésion, alignement et séparation plus deux autres inspirés de Géoboids de MacGill : chercher et fuir.

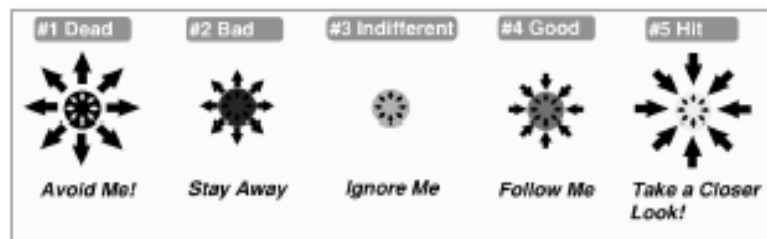


Figure 1.19 Cinq états de geoBoid, utilisés pour les boids de traitement d'images [MacGill 00]

Ces deux derniers comportements visent à chercher : se diriger vers des agents satisfaits (qui sont dans une position proche du contour), et à fuir : s'éloigner des agents insatisfaits (position plus loin du contour).

Un résultat d'application de cette méthode est indiqué dans la (figure 1.20).

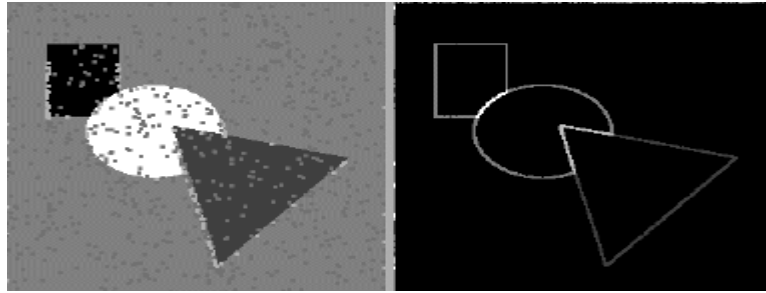


Figure1.20 Exemple de résultat (image de synthèse) d'après [Carden 02]

Conclusion

Nous avons introduit dans ce rapport les notions de base qui servent de fondement à la compréhension de différentes techniques de traitement d'images. Plusieurs méthodes classiques de traitement ont été proposées dans la littérature, nous avons présenté quelques unes qui nous semblent les plus courantes dans le processus du traitement et analyse d'image.

Les pré-traitements d'images permettent d'améliorer la qualité de l'image en vue de traitements ultérieurs. Deux grandes approches peuvent être envisagées pour extraire les zones pertinentes des images :

On cherche les discontinuités dans la scène, c'est l'approche contour.

On cherche des zones de niveaux de gris homogènes, c'est l'approche région

De nouvelles recherches s'intéressent aux systèmes complexes et leurs caractéristiques ainsi l'application de ces approches sur le traitement d'images.

Les phénomènes collectifs naturels sont une source d'inspiration pour proposer des méthodes multi-agents de résolution de problèmes. Ils permettent l'élaboration de comportements individuels simples produisant collectivement des phénomènes complexes.

A la fin de ce chapitre, nous avons présenté les différentes approches qui ont été élaborées pour le traitement d'images en niveaux de gris soit par agents simples situés dans l'image, ou bien, par inspiration des systèmes naturels.

Introduction

Une image est la représentation d'une scène acquise à l'aide de systèmes de production d'images (appareils photographique, caméra, radiographes, scanner, sonar,.....). Sa forme peut être analogique (ex: négatif, photographie, vidéo..) ou numérique (images numérisées suivant divers formats (images compressées ou non...) ou obtenues par des capteurs fournissant des images numérisées) et dans ce cas un traitement par ordinateur est possible.

2.1-Définition d'une image

Un signal q-modal et n-dimensionnel est défini dans un espace E de dimension $n \in \mathbb{N}^*$. C'est une image pour $n \geq 2$. au point P de l'espace E, l'image est décrite par le vecteur couleur au point P ayant q composantes représentant les composantes colorimétriques.

Une image peut donc être considérée comme une fonction $I(X)$ définie sur un espace multidimensionnel.

- X est un vecteur de coordonnées définissant une position dans un espace multidimensionnel (ex. $X = (x_1, x_2)$ dans le 2D)

- $I(X)$ est une valeur scalaire dans le cas d'une image unimodale et un vecteur à q composantes pour une image q-modale. Les images les plus courantes sont définies dans un espace de dimension 2 (images 2D) : photographies noir et blanc ou couleur, radiographies,... ou de dimension 3 (images 3D) : images médicales tomographiques par exemple.

2.2-Définition d'une image numérique

Pour une image numérique, les fonctions image sont échantillonnées pour former des ensembles de points (**pixels**, PICTURE ELEMENTS). En général le maillage d'échantillonnage est rectangulaire (mais il peut être aussi triangulaire ou plus complexe). D'où la représentation matricielle suivante :

$$\begin{array}{c}
 \updownarrow \\
 \begin{array}{cccc}
 I(0,0) & I(0,1) & \dots\dots\dots & I(0,n-1) \\
 I(1,0) & I(1,1) & \dots\dots\dots & I(1,n-1) \\
 \dots\dots\dots & & & \\
 \dots\dots\dots & & & \\
 \dots\dots\dots & & & \\
 I(n-1,0) & \dots\dots\dots & & I(n-1,n-1)
 \end{array} \\
 \updownarrow
 \end{array}$$

Cette image peut être stockée ligne par ligne par exemple, ce qui signifie que les traitements sur cette image seront réalisés si nécessaire plutôt ligne par ligne. L'échantillonnage de l'image peut être plus ou moins fin (64 X 64 ou 128 X 128 ou 4092 X 4092 , ou 523 X 765) mais chaque pixel va représenter une certaine partie de la scène réelle, plus ou moins grande mesurée par ce qu'on appelle la résolution (spatiale) de l'image [1].

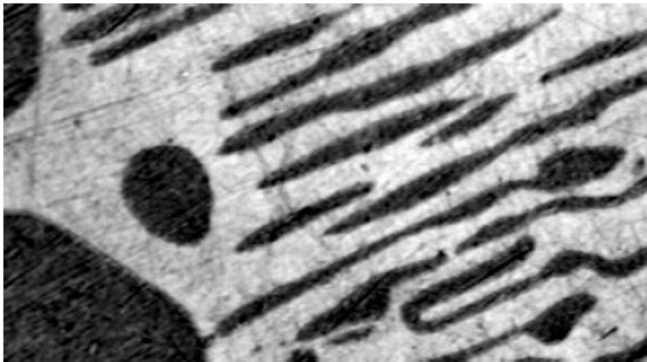


Figure 2.1: image 225x409



Figure 2.2: image 45x82 de la même scène (zoom X5)

2.3-Les différents types d'images numériques:

2.3.1- Les images noir et blanc (monochrome) :

Pour ces images on ne prend pas en compte ici la couleur mais seulement l'intensité lumineuse (l'exemple classique correspond aux photographies noir et blanc). Parmi ces images on peut trouver:

- **Les images en niveaux de gris** : dans ce cas on dispose d'une échelle de teintes de gris, et la plupart du temps on dispose de 256 niveaux de gris avec:

0 ———> noir.....127 ———> gris moyen.....255 ———> blanc

et ceci est commode car l'unité d'information est l'octet. Il est à noter que l'humain standard ne reconnaît au plus que 64 niveaux de gris.

- **Les images binaires** où chaque pixel est représenté par un bit (0/1) avec en général le 0 correspondant au noir (intensité minimale) et le 1 correspondant au blanc (intensité maximale).

Notons que la plupart des systèmes de traitement d'images placeront chaque pixel dans un octet (code 0 ou 255 (pour coder le 1 de l'image binaire)) pour des facilités d'accès et d'écriture des algorithmes. Ces images peuvent être générés facilement à partir des images en niveau de gris et

cela en fixant un seuil S et en l'utilisant pour partitionner l'image en niveau de gris en des pixels avec deux valeurs d'intensité tel que :

Si $I(x,y) \geq S$ alors $I_{BIN} = 1$

Si $I(x,y) < S$ alors $I_{BIN} = 0$

Avec I_{BIN} est la version binarisé de l'image en niveau de gris I .

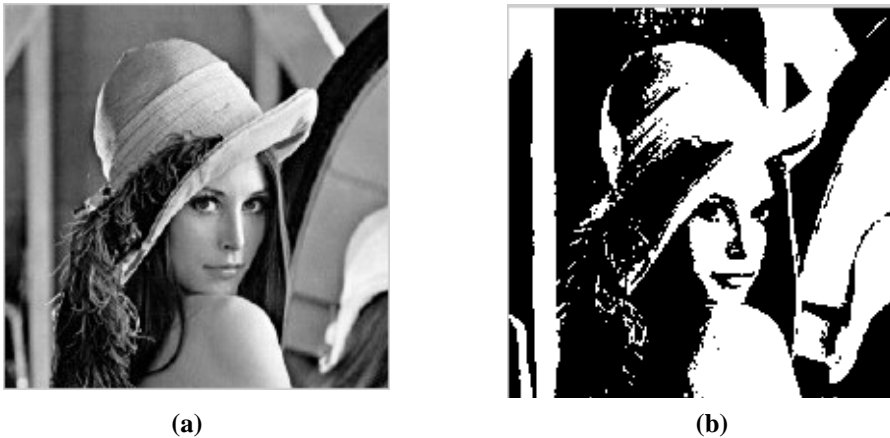


Figure 1.3 : image Lena en niveaux de gris(a), image binarisé(seuil 125) (b)

2.3.2- Les images couleur :

Ces images sont en général codées en utilisant le codage des trois couleurs fondamentales (rouge, vert, bleu), on parle alors d'images RVB. (Cela correspond au codage des téléviseurs couleur Français). Ce sont donc des images tri-modales.

Chaque couleur est codée sous forme d'un octet, d'où :

Image RVB $\left\{ \begin{array}{l} \text{composante rouge, intensité de 0 à 255} \\ \text{composante verte, intensité de 0 à 255} \\ \text{composante bleu, intensité de 0 à 255} \end{array} \right.$

On code ainsi $2^{24} = 16\,777\,216$ couleurs différentes.

Il existe d'autres techniques de représentation de la couleur pour les images couleur en passant d'un espace 3D, l'espace RVB, à un autre espace 3D défini par une autre base telle que les espaces colorimétriques HSV, $I_1 I_2 I_3 \dots$

Pour le modèle RVB, chaque couleur apparaît dans ses composantes spectrales primaire de rouge, vert et bleu. Ce modèle est basé sur un système de coordonnées cartésiennes. L'espace couleur est un cube dans lequel les couleurs rouge, vert et bleu occupent trois coins, les couleurs

cyan, magenta et jaunes occupent trois autres coins, le noir est l'origine et le blanc occupe le coin le plus loin de l'origine (Figure 1.4). Dans ce modèle, le niveau de gris s'étend du noir au blanc tout au long de la ligne qui relie ces deux points et les couleurs sont définies par des vecteurs à trois composantes [6].

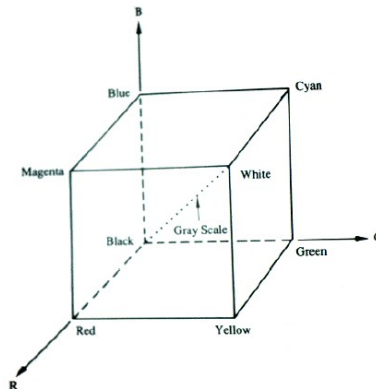


Figure 2.4 : La représentation spatiale du modèle RVB

Les images dans le modèle RVB sont constituées de trois plans indépendants, une par couleur primaire.

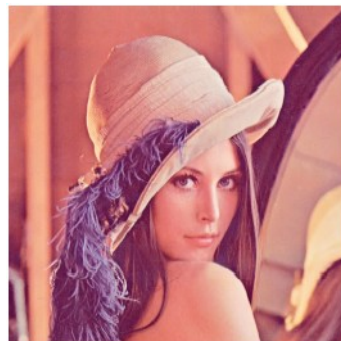


Figure 2.5 : image Lena en codage RVB



(a) (b) (c)
Figure 2.6: composante R (a), composante V (b), composante B (c)

2.4- Caractérisation des images numériques :

2.4.1- L'histogramme classique:

L'histogramme d'une image est la représentation de la fonction discrète $v(C)$ qui à chaque couleur \mathbf{r}_C de l'image associe le nombre $v(\mathbf{r}_C)$ de pixels (population) ayant cette couleur. Il peut être vu comme la densité de probabilité d'une variable aléatoire dont l'image est l'ensemble de réalisations. Pour les images couleurs, étant donné que les valeurs de l'image sont des vecteurs, on parle alors d'histogramme multidimensionnel (tridimensionnel).

En analyse d'images, les calculs d'histogrammes sont fréquents, depuis la mise au point de la chaîne d'acquisition jusqu'à l'indexation d'images dans une base, de nombreuses méthodes de prétraitement, de segmentation ou de mesures statistique nécessitent le calcul d'histogrammes. Cependant les histogrammes multidimensionnels ont l'inconvénient d'occuper en mémoire un volume gigantesque. Ce volume est de quelques dizaines ou centaines de mégaoctets pour les images couleur à trois composantes.

Partant du fait qu'une image couleur peut être décomposée en trois images unimodales indépendantes, certaines méthodes de traitement d'images utilisent les histogrammes monodimensionnels R, V et B de ces trois images.

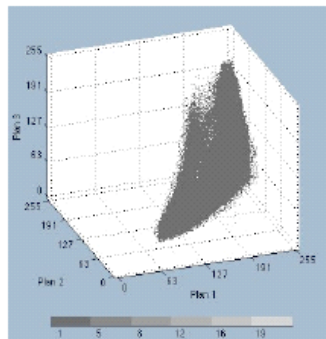
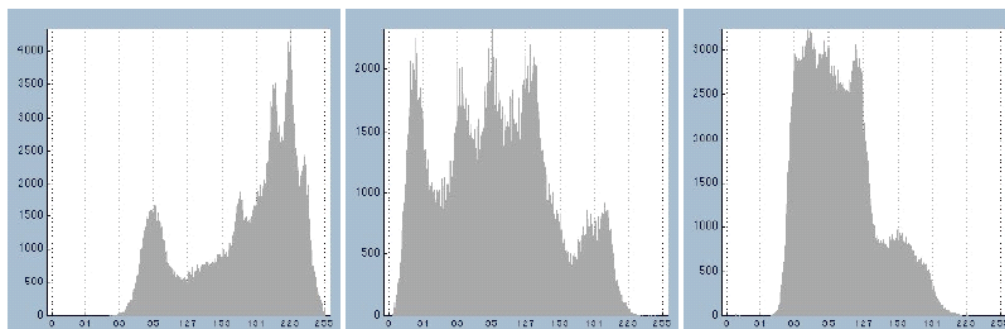


Figure 2.7 : histogramme 3-dimensionnel de l'image Lena



(a) (b) (c)
Figure 2.8 : histogramme R (a) , Histogramme V (b) , Histogramme B (c)

D'autres méthodes utilisent les histogrammes bidimensionnels obtenus par projection de l'histogramme tridimensionnel dans les plans RV, RB et VB.

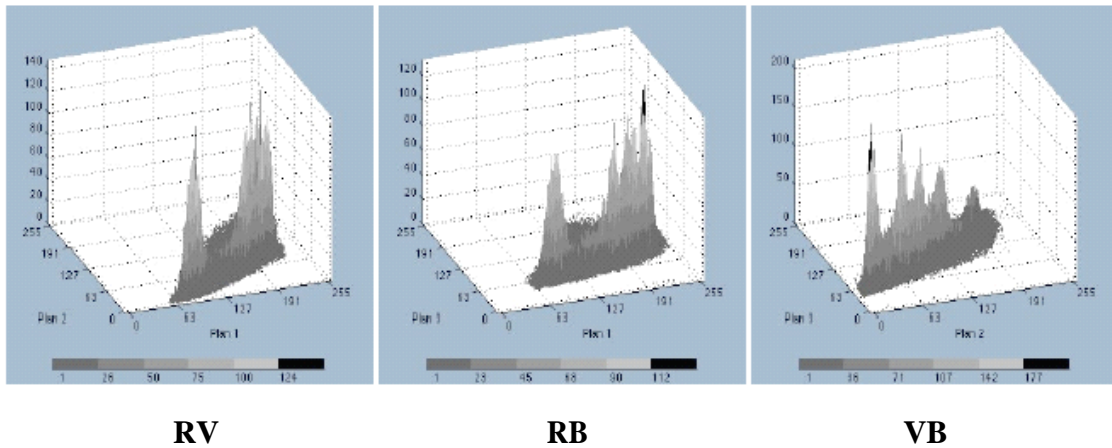


Figure 2.9 : Histogrammes 2-dimensionnels de l'image Lena

L'histogramme multidimensionnel d'une image de résolution $M \times N$ à q composantes colorimétriques codées chacune sur p bits est constitué par un tableau q -dimensionnel comportant $2^{q \cdot p}$ cellules. Chaque cellule de ce tableau est codée sur E bits tel que $E > \log_2(M \times N)$, ce qui fait que l'histogramme occupe $2^{p \cdot q} \cdot E$ bits.

2.4.2- L'histogramme compact :

Soit C le nombre de cellules occupées dans l'histogramme multidimensionnel d'une image de résolution $M \times N$ à q composantes colorimétriques codées chacune sur p bits. En pratique, C est toujours très inférieur au nombre total $2^{q \cdot p}$ de cellules disponibles. À nombre de pixels $M \times N$ constant, plus le nombre p de composantes de l'image est important, plus le nombre de cellules inoccupées augmente. Le principe de l'histogramme multidimensionnel compact consiste à ne coder que les C cellules effectivement occupées. L'histogramme compact sera représenté par un tableau de dimensions $C \times q$ pour stocker les valeurs colorimétriques, et un tableau de

dimensions $C \times 1$ pour les effectifs correspondants. L'espace mémoire occupé est de $C.E + C.p.q$.
 par rapport au codage classique énoncé précédemment [3].

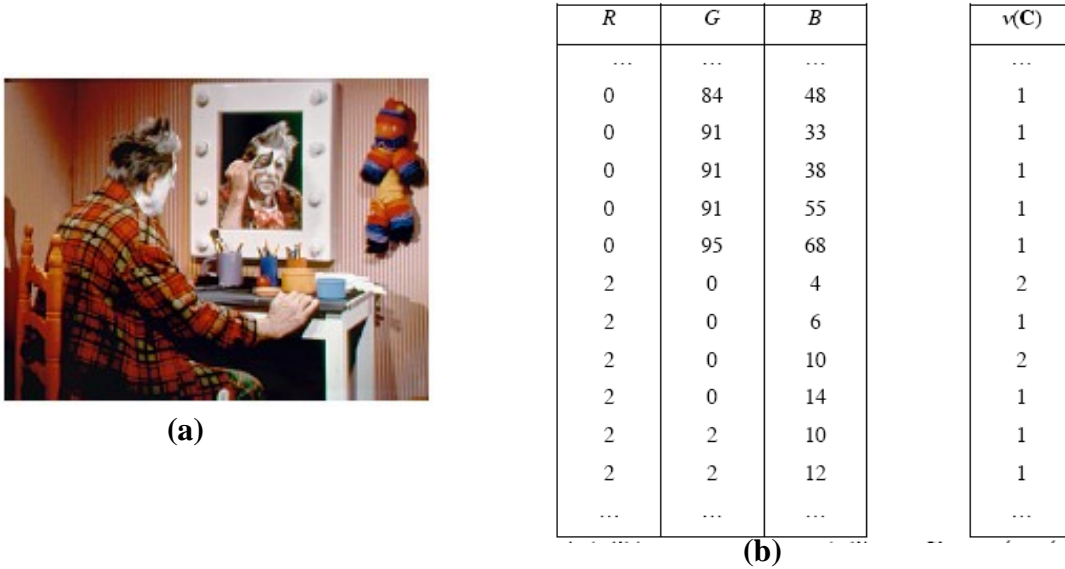


Figure 2.10 : l'image Clown en codage RVB (a) et un extrait de son histogramme compact

2.4.3- connexité dans les images numériques :

a) Voisinage d'un pixel :

Un pixel P de coordonnées (x, y) a quatre voisins horizontaux et verticaux dont les coordonnées sont $(x+1,y)$, $(x-1,y)$, $(x,y+1)$ et $(x,y-1)$. Cet ensemble représente le voisinage d'ordre 4 de P. Les quatre voisins diagonaux de P ont pour coordonnées $(x+1,y+1)$, $(x+1,y-1)$, $(x-1,y+1)$ et $(x-1,y-1)$. La réunion de ces deux ensembles constitue l'ensemble des voisins d'ordre 8.

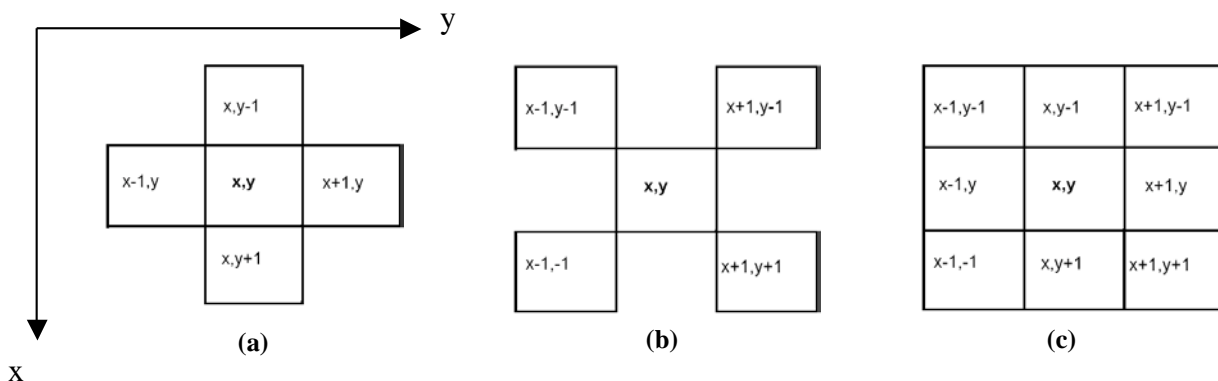


Figure 2.11 : voisinage d'ordre 4 (a), diagonal (b), d'ordre 8 (c)

b) Connexité :

La connexité est une propriété de liaison entre deux pixels qui fait qu'on les considère comme faisant partie de la même région dans une image. Pour établir que deux pixels sont connexes, il faut mettre en œuvre deux critères l'un relatif au voisinage, l'autre à la luminance (pour les images 1-modales) ou à la couleur (pour les images q -modales).

- Le critère de voisinage est un critère d'appartenance à l'un des ensembles de voisins (d'ordre 4, diagonal ou d'ordre 8)

- Le critère de luminance (ou de couleur) est un critère d'appartenance à un sous-ensemble $\{V\}$ de luminances ou de couleurs.

Deux pixels sont dits connexes si l'un appartient à l'ensemble des voisins de l'autre et si leurs luminances (ou couleurs) appartiennent à $\{V\}$. Selon le type de voisinage on parle de connexité d'ordre 4, connexité diagonale ou connexité d'ordre 8.

c) Sous-ensembles connexes :

Deux sous-ensembles I_1 et I_2 d'une image I sont dits connexes si l'un quelconque des pixels de I_1 est connexe de l'un quelconque de pixels de I_2 . La connexité des sous-ensembles peut être d'ordre 4, diagonale, ou d'ordre 8.

d) Composantes connexes :

Soit S un sous-ensemble de l'image I , et soient p et q deux pixels de S . On dit que p et q sont connexes dans S s'il existe un chemin de p à q contenu entièrement dans S . L'ensemble des pixels connexes de S est appelé composante connexe S_c de S .

En conséquence :

- deux pixels quelconques d'une composante connexe S_c sont connexes dans S_c ;
- deux composantes connexes distinctes sont disjointes (elles n'ont pas de pixel commun).

Conclusion :

Les caractéristiques énoncées dans cette partie sont à la base de la plupart des procédés de traitement d'images, notamment les méthodes de segmentation d'images qui feront l'objet de la partie qui suit.

Chapitre III

La segmentation d'images couleurs

Introduction :

A partir d'une image numérique, il convient de d'*extraire* les informations pertinentes en regard de l'application concernée, les *traiter* puis les *interpréter*. Le terme générique d'analyse d'images désigne l'ensemble de ces opérations.

En analyse, d'images on distingue les traitements de bas-niveau et ceux de haut-niveau. Cette distinction est liée au continu sémantique des entités traitées et extraites de l'image. Les traitements de bas-niveau opèrent, en général, sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire nécessairement la liaison avec la réalité qu'elles représentent. A l'opposé, les traitements de haut-niveau s'appliquent à des entités de nature symboliques associées à une représentation de la réalité extraite de l'image ; ils sont relatifs à l'interprétation et à la compréhension de l'image.

La segmentation d'image est un traitement de bas-niveau qui consiste à créer une partition d'une image A en sous ensembles R_i appelés régions, tel que :

$$\forall i R_i \neq \emptyset$$

$$\forall i, j \ i \neq j \quad R_i \cap R_j = \emptyset$$

$$A = \cup R_i$$

Une région est un ensemble connexe de pixels ayant des caractéristiques communes (intensité, texture,...) qui les différencient des pixels des régions voisines. Dans le cas des images couleur ces caractéristiques sont déterminées à partir des composantes colorimétriques des pixels. Les connaissances utilisées sont les plus souvent du domaine de l'image numérique et du traitement du signal, donc sémantiquement assez pauvres [1].

3.1- les différentes méthodes de segmentation :

La segmentation peut être abordée de deux points de vue dans la mesure où une région peut être définie par l'ensemble des pixels la composant (approche région de la segmentation) ou bien par les contours de la région (approche contour de la segmentation). Ces deux approches sont duales du fait que chaque région possède un contour et qu'un contour délimite forcément une région [4].

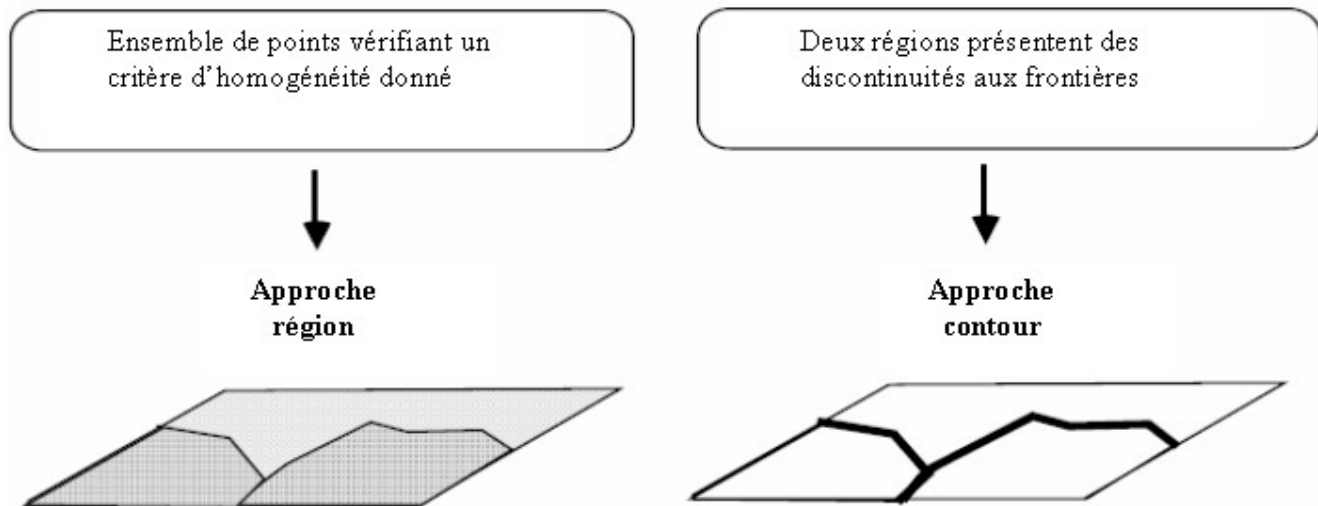


Figure 3.1 : approche région et approche contour

3.1.1- L'approche contour :

L'approche contour est fondée sur la recherche de discontinuités locales. Un moyen efficace pour détecter ces discontinuités dans le cas d'une image analogique est d'utiliser une méthode dérivative. Le contour est détecté comme le maximum du gradient de luminance $C(x,y)$ de l'image. Ce raisonnement peut être étendu aux images numériques bidimensionnelles en mettant en œuvre des calculs approchés des dérivées sous forme de convolution de l'image par des masques appropriés. Pour obtenir le gradient de l'image selon l'axe Ox (respectivement Oy) il suffit de la convoluer avec un masque M_x approprié (respectivement Oy). Les gradients les plus utilisés sont :

-1	0	1
----	---	---

M_y

-1
0
1

M_x

- es gradients de Perwitt :

-1	-1	-1
0	0	0
1	1	1

M_x

-1	0	1
-1	0	1
-1	0	1

M_y

Figure 3.3 : Masques de convolution pour le gradient de Perwitt

- les gradients de Sobel :

-1	-2	-1
0	0	0
1	2	1

M_x

-1	0	1
-2	0	2
-1	0	1

M_y

Figure 2.4: Masques de convolution pour le gradient de Sobel

3.1.2- L'approche région :

Dans l'approche région deux stratégies sont possibles :

- la classification pixellaire qui consiste à associer à chaque pixel de l'image un point

couleur dans l'espace colorimétrique q -dimensionnel. L'inconvénient de cette méthode est qu'elle ne prend pas en compte la disposition spatiale des pixels dans l'image, elle se contente de construire des classes colorimétriques, et d'affecter à chaque pixel une étiquette correspondant à sa classe d'appartenance, la formation des régions arrive en fin du traitement, par une analyse de la connexité des pixels dans l'image étiquetée ; cela suppose que les pixels appartenant à une région R_i de l'image possèdent des couleurs suffisamment voisines pour former un nuage de point (une classe) dans l'espace colorimétrique.

- La deuxième stratégie privilégie l'interaction spatio-colorimétrique entre les pixels ; elle opère directement dans l'espace image, soit de façon ascendante en regroupant des

divisant l'image en quadtree jusqu'à obtention de régions homogènes (procédure de séparation) ; dans le dernier cas, on aboutit la plupart du temps à une sur-segmentation, en rendant nécessaire un regroupement de certaines des régions ; c'est la procédure de fusion.

3.1.2.1- la classification pixellaire :

Elle opère en deux temps :

- i) la phase d'apprentissage consiste à construire les classes de pixels en déterminant leurs caractéristiques.
- ii) La phase de décision lors de laquelle chaque pixel est affecté à la classe dont le centre de gravité correspond à la couleur la plus proche de celle du pixel à classer. L'opération nécessite donc la mesure d'une distance dans l'espace colorimétrique.

Selon la nature de la phase d'apprentissage, on distingue deux familles de classification :

- La classification avec *apprentissage non supervisé* consiste à découper l'espace colorimétrique en zones homogènes selon un critère exprimé dans l'espace des composantes. Cette approche permet une classification *en aveugle* et réalise une segmentation sans connaissance *a priori* de l'image. Selon qu'il est nécessaire ou non de fixer au préalable le nombre de classes dans l'image, on distingue les méthodes fondées sur *l'analyse d'histogrammes* qui déterminent automatiquement le nombre de classes et celles de regroupement héritées de la classification de données multidimensionnelles.
- La classification avec *apprentissage supervisé* nécessite de fournir pour chaque classe un ensemble de pixels représentatifs de la segmentation que l'on souhaite obtenir. Le nombre de classes doit donc être connu au préalable et repose, de même que le choix de l'ensemble d'apprentissage, sur une connaissance *a priori* des images à segmenter.

a) l'analyse d'histogrammes :

La classification par détection de modes (pics) dans l'histogramme des couleurs ne nécessite pas la connaissance à priori du nombre de classes. Elle se heurte cependant à une difficulté qui est l'étendue de l'espace colorimétrique à explorer. Devant la difficulté à manipuler un tel volume de données, trois stratégies sont possibles :

- manipuler séparément les trois histogrammes monodimensionnels correspondant à chacun des axes colorimétriques
- considérer l'histogramme tridimensionnel mais requantifier les composantes sur γ bits

- manipuler les histogrammes k-dimensionnels tel que $k < 3$ obtenus par projection de l'histogramme tridimensionnel sur k axes de l'espace colorimétrique.

Dans tous les cas, la méthode tend à sélectionner les modes principaux de l'histogramme qu'on suppose caractéristiques des régions cherchées puisqu'une région est considérée comme un ensemble de pixels ayant des niveaux de gris voisins et ce à l'aide d'un seuil fixé à l'avance. La valeur du seuil ne dépend pas de la nature des images mais du nombre de classe souhaitées : plus l'effectif nécessaire pour qu'un mode de l'histogramme soit jugé significatif sera faible, plus le nombre de classe sera élevé.

Plusieurs méthodes de détection des seuils sont distinguées :

Méthode par recherche de sommets (ou de vallées) :

Prenons comme exemple l'histogramme monodimensionnel suivant :

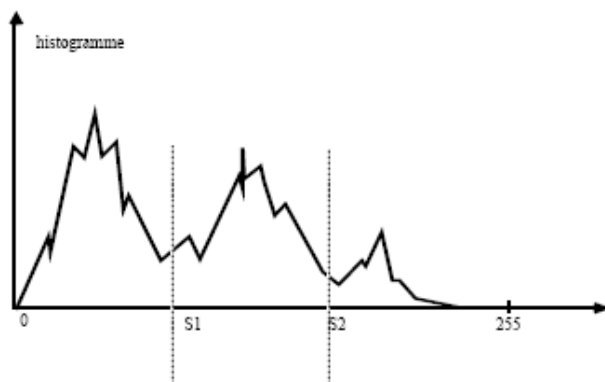


Figure 3.5 : exemple d'histogramme monodimensionnel

On voit ici que le problème de détection automatique des seuils est difficile car il n'y a pas de véritables vallées. Il peut y avoir une multitude de petites vallées. Une approche intéressante consiste à mettre en évidence les grandes vallées, à les creuser et à « supprimer » les petites vallées sans signification par un traitement directement sur l'histogramme.

Pour réaliser ce type de transformation de l'histogramme et mettre en évidence les vallées susceptibles de correspondre à des seuils on peut faire l'hypothèse raisonnable suivante:

- les sommets significatifs de l'histogramme correspondent à priori aux pixels les plus intérieurs de la région (à priori les plus nombreux)
- les vallées significatives de l'histogramme correspondent plutôt aux pixels de la frontière des régions. On obtient à la fin du traitement l'histogramme suivant sur lequel on distingue clairement les deux seuils S_1 et S_2 :

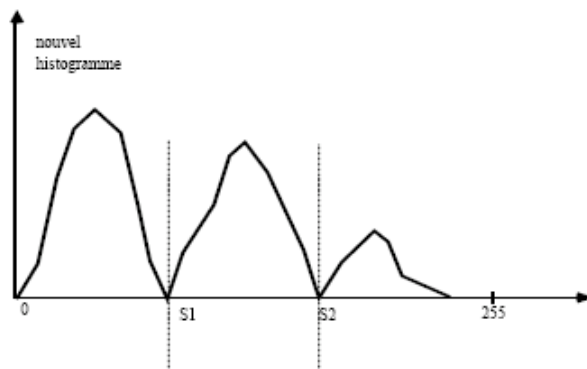


Figure 2.6: histogramme transformé

Méthode de détection de seuil par segmentation de l'histogramme :

Cette méthode (développée par Otsu) ne s'applique que dans le cas de la segmentation d'image en deux catégories (le fonds et les objets). On suppose être dans le cas d'un histogramme où les deux populations de pixels se recouvrent partiellement. L'idée va être alors de chercher un seuil permettant d'obtenir les deux populations en minimisant une fonction de coût.

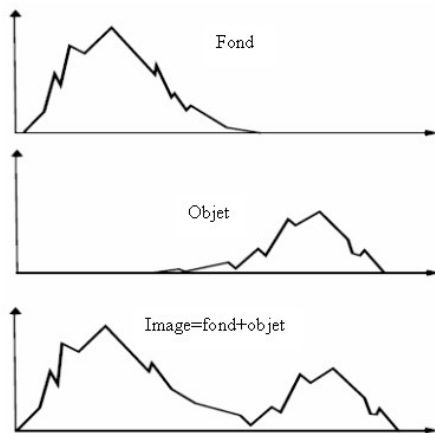


Figure 3.7 : exemple de segmentation d'histogramme

D'un point de vue théorique, la classification par analyse d'histogrammes multidimensionnels donne de meilleurs résultats que l'analyse marginale. Elle se heurte cependant au problème de l'espace mémoire nécessaire au codage des histogrammes et il y a en pratique peu de méthodes de segmentation qui relèvent de cette approche.

b) les méthodes de regroupement (clustering) :

Ces méthodes analysent les nuages de points formés par les pixels de l'image dans l'espace

préalablement fixé). Les méthodes les plus connues sont la méthode des *nuées dynamiques* et sa variante *isodata*.

c) Classification avec apprentissage supervisé :

La classification est dite supervisée lorsque la phase d'apprentissage nécessite de présenter au système de classification un ensemble de pixels dont la classe et les composantes colorimétriques sont connues. Deux approches sont distinguées :

- ***L'approche bayésienne :***

Comme indiqué au paragraphe 3.3.1, l'histogramme d'une image q -modale 2-dimensionnelle de résolution $M \times N$ est la représentation d'une fonction discrète $v(C)$ qui, à chaque couleur C présente dans l'image, associe le nombre $v(C)$ de pixels ayant cette couleur C .

L'approche neuromimétique :

Le perceptron multi-couche est le plus connu des réseaux de neurones. Il est utilisé pour affecter à N_c classes des données vectorielles de dimension L . Le perceptron est constitué d'une couche d'entrée comportant L neurones et d'une couche de sortie comprenant $\log_2(N_c)$ neurones, séparées par une ou plusieurs couches cachées. Le nombre N_s de neurones de la couche de sortie est fixé par le nombre de classes recherché (par exemple $N_s = 1$ lorsqu'il s'agit de différencier deux classes). Pour une application donnée, il faut définir la structure idéale du perceptron (celle qui donne la meilleure classification) par un ajustement du nombre.

3.1.2.2- Croissance de régions :

La méthode consiste à faire croître progressivement des régions distinctes à partir de pixels initiaux appelés germes, et notés g_i . Le nombre de régions ainsi obtenu est au plus égal au nombre de germes. A chaque région sont agrégés les pixels connexes qui obéissent au prédicat de segmentation.

La croissance d'une région, à partir de g_1 , s'arrête lorsque aucun pixel ne peut lui être agrégé. On fait croître alors une autre région autour de g_2 , etc., jusqu'à ce que tous les pixels de l'image aient été traités, en rajoutant des germes si nécessaire.

Le choix du nombre de germes, qui détermine le nombre maximum de régions segmentées, peut être guidé par l'histogramme de l'image.

3.1.2.3- séparation :

Elle consiste à séparer l'image initiale en quatre quadrants, et à tester un critère d'homogénéité des couleurs d'une région sur chacun de ces quadrants. Si un quadrant vérifie ce critère, il est retenu comme constituant une région. Sinon il est à nouveau divisé en quatre quadrants, et la procédure recommence, jusqu'à ce que tous les quadrants obtenus vérifient le critère d'homogénéité.

3.1.2.4- Fusion :

La fusion fournit un nombre minimum de régions connexes, mais ne donne aucune relation de proximité entre régions ; la séparation fournit au contraire une structure hiérarchisée (quadtree) représentative des relations de proximité entre régions, ce qui est très utile pour analyser une scène. En contrepartie, cette procédure aboutit à une multitude de régions différentes : l'image est sur-segmentée. Pour bénéficier des avantages de ces deux procédures a été élaborée la procédure hybride dite de séparation-fusion. Après avoir réalisé la séparation, et analysé la structure de la scène par le biais du quadtree, il est possible de fusionner les régions connexes qui satisfont au prédicat de séparation. Il est à noter que le prédicat utilisé pour la fusion peut différer de celui utilisé pour la séparation.

Conclusion :

Dans cette section nous nous sommes intéressés au problème de la segmentation d'images couleurs. Nous avons présenté les principales méthodes que l'on peut trouver dans la littérature et nous avons détaillé plus précisément les méthodes basées sur l'analyse d'histogrammes étant donné que la méthode faisant objet du chapitre qui suit en dérive.

Chapitre IV

Etiquetage en composantes connexes d'histogrammes multidimensionnels

Introduction :

Le concept de d'étiquetage en composantes connexes s'applique généralement sur les images binaires, et consiste à trouver les différentes composantes connexes de l'image et à les étiqueter. L'étiquetage consiste à affecter une étiquette identique à tous les pixels d'une même composante connexe.

Dans le cadre de ce travail, nous allons appliquer ce concept sur les histogrammes 2D et 3D des images couleur à fin de les segmenter [8].

4.1- Principe :

4.1.1-Cas d'une image 2D :

L'étiquetage en composantes connexes d'une image binaire attribue à tous les pixels d'une composante connexe la même valeur entière. Chaque composante connexe est alors identifiée par son étiquette. Les algorithmes d'étiquetage détectent les adjacences entre les pixels et définissent l'étiquette du point courant en fonction de celle de ses points voisins.

Ils opèrent en balayant l'intégralité de l'image. Selon le type de balayage on distingue deux approches d'étiquetage en composantes connexes :

i) Etiquetage séquentiel :

L'idée de l'algorithme repose sur un balayage séquentiel de l'image. On considère pour chaque point P ses prédécesseurs déjà traités. Les prédécesseurs d'un point dépendent du type de connexité considéré. (Figure 3.1).

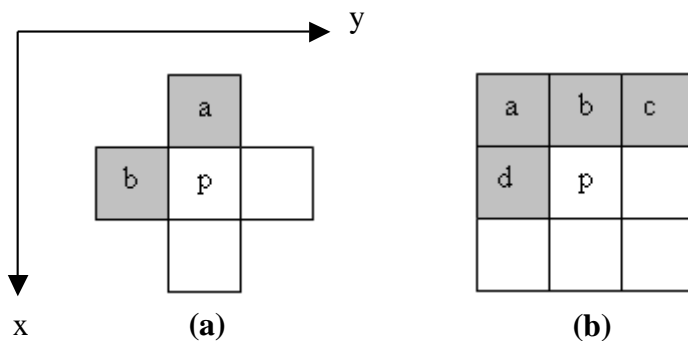


Figure 4.1: prédécesseurs d'ordre 4 (a) , prédécesseurs d'ordre 8 (b)

En voisinage d'ordre 4, P a deux prédécesseurs de coordonnées $(x-1,y)$ et $(x,y-1)$

En voisinage d'ordre 8 il en a quatre de coordonnées $(x,y-1)$, $(x-1,y-1)$, $(x-1,y)$ et $(x-1,y+1)$

Le principe est le suivant (en voisinage 4 sur lequel est basé notre algorithme) :

1. balayer l'image ligne par ligne de haut en bas et de gauche à droite pour détecter les pixels qui sont à 1.
2. Si P appartient au fond, passer à la position d'après
sinon;
 - Si a et b appartiennent au fond affecter une nouvelle étiquette à P
 - Si a ou b n'appartient pas au fond, affecter son étiquette à P
 - Si a et b n'appartiennent pas au fond, mais qu'ils ont des étiquettes différentes, affecter la plus petite étiquette à P et considérer les deux étiquettes comme équivalentes.
3. Après ce balayage, les pixels n'appartenant pas au fond sont étiquetés. Certaines étiquettes sont équivalentes, on leur affecte la même étiquette.
4. Faire un 2ème balayage pour affecter aux pixels leur étiquette finale.

Nous allons appliquer cet algorithme sur des structures 3D en effectuant les modifications nécessaires.

ii) Etiquetage récursif :

Contrairement à l'étiquetage séquentiel, l'étiquetage récursif prend en considération tous les voisins du pixel à traiter. Le principe est le suivant :

- 1- scanner l'image de gauche à droite et de haut en bas pour trouver les pixels qui sont à 1 indiquant la présence d'une région.
- 2- S'il n'y a pas de pixels non étiquetés, alors le programme s'arrête,
Sinon ;
 - Si un pixel à 1 est rencontré, on lui affecte une nouvelle étiquette L.
 - On affecte la même étiquette L à tous les pixels qui lui sont connexes.
 - La connexité est localisée par une fonction récursive qui fait appel à elle même.

4.1.2- Cas d'une image 3D :

L'algorithme énoncé précédemment pour les images 2D reste applicable sur des structures

passer à une notion de voisinage 3D vu la structure cubique de l'image.

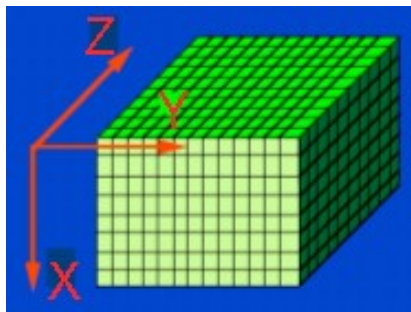


Figure 4.2: structure cubique d'une image 3D

Les types de voisinage les plus utilisés en traitement d'images 3D sont les suivants ;

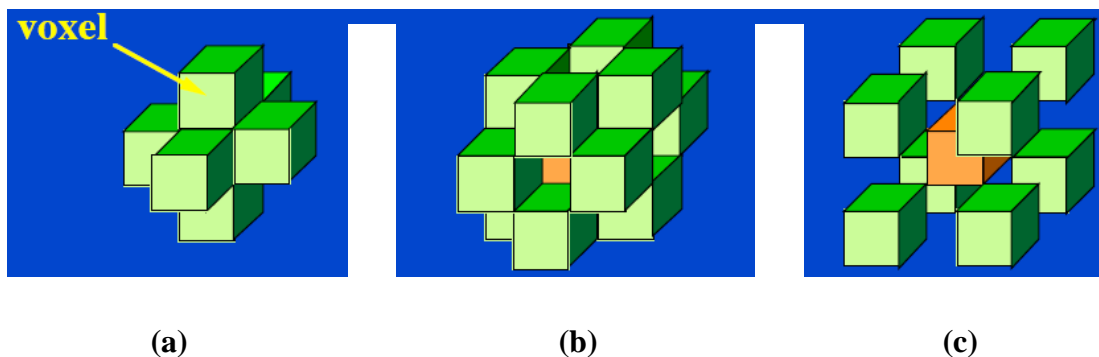


Figure 4.3: voisinage facette commune ou d'ordre 6 (a), arête commune (b), sommet commun (c)

En voisinage d'ordre 6 sur lequel est basé notre algorithme un pixel $P(x,y,z)$ a six voisins de coordonnées $(x,y-1,z)$, $(x,y,z-1)$, $(x-1,y,z)$, $(x+1,y,z)$, $(x,y+1,z)$ et $(x,y,z+1)$ et trois prédécesseurs dont les coordonnées sont $(x,y-1,z)$, $(x,y,z-1)$ et $(x-1,y,z)$.

4.2- Application des algorithmes d'étiquetage en composantes connexes sur les histogrammes 2D et 3D :

L'histogramme 2D d'une image I de résolution $M \times N$ peut être représenté par une image J dont les dimensions spatiales sont celles des deux axes choisis dans l'espace colorimétrique associé à I . Si les composantes colorimétriques de I sont codées sur Q bits, J sera de dimension $2^Q \times 2^Q$. chaque pixel P_j de J a pour valeur l'effectif de l'histogramme au point de coordonnées colorimétrique défini par les coordonnées spatiales de P_j . une image binaire sera constituée en effectuant un seuillage sur les effectifs. De même, l'histogramme 3D peut être représenté par un cube de dimension $2^Q \times 2^Q \times 2^Q$ et un seuillage adéquat nous permet d'obtenir un cube binaire. Pour le seuillage des histogrammes on procède comme suit :

- fixer deux seuils S_1 et S_2
- à chaque valeur d'effectif v de l'histogramme tel que $v \in [0, \text{effmax}]$ ($\text{effmax} \leq M \times N$) on associe la valeur $f(v) \in \{0,1\}$ tel que :

$$f(v) = \begin{cases} 1 & \text{si } v \in [S_1, S_2] \\ 0 & \text{sinon} \end{cases}$$

L'histogramme binarisé sera donc l'image de l'histogramme original par la fonction f .

Grâce à cette représentation, l'analyse de l'histogramme pourra être effectuée en utilisant les outils d'analyse d'images binaires. Nous allons implémenter l'algorithme d'étiquetage en composantes connexes sur Une plate-forme pour le développement et la mise en routine d'algorithmes en analyse d'images couleur qui est *Isaac*.

4.2.1- La plateforme Isaac:

Isaac est une plate-forme de développement modulaire conçue pour à la fois accélérer la mise en œuvre de nouveaux algorithmes et permettre la création de véritables applications utilisables en routine. Il offre à l'utilisateur toute la puissance et la facilité de mise au point du langage MATLAB enrichi par des objets de haut niveau qui assurent automatiquement l'interface avec la plateforme et grâce auxquels tout nouveau développement est immédiatement opérationnel [3].

Utilisation de la plateforme :

Une fois lancé par la commande MATLAB ''Isaac'', le logiciel se représente à l'utilisateur comme un ensemble de palettes d'outils et de menus entièrement pilotés par une interface graphique. Environnement sécurisé, Isaac génère des messages d'erreurs, demande confirmation avant d'effectuer des opérations irréversibles, dispose d'une aide en ligne de et d'une documentation au format HTML.

Développement de modules sous Isaac :

Les modules sont des fonctions constituées d'un seul fichier de type ''plug and play''. Le développement d'un module permet de supprimer totalement l'écriture des interactions avec


modules, assurer le contrôle et le passage des paramètres, la gestion des erreurs ainsi que la modification des données et objets en sortie. Les règles de développement sont les suivantes :

- Appelé sans paramètre, un module doit retourner un tableau $T=\{P_1, P_2, \dots, P_n\}$ décrivant les paramètres d'entrée demandés en plus de l'image cible qui sera toujours le premier paramètre passé au module. S'il n'y a pas d'autres paramètres que l'image cible ($n=0$), le tableau retourné est vide. Les valeurs possibles de P_i sont :

1	Matrice image couleur
2	Matrice image niveau de gris
3	Matrice image binaire
10	Nom complet de répertoire
21	Valeur numérique réelle
22	Valeur texte
23	Matrice liste des pixels
24	Matrice Région d'intérêt
25	Matrice histogramme
[M,N,P]	Matrice M lignes, N colonnes et P plans de réels

- appelé avec $n+1$ paramètres, un module doit retourner I et M tel que :
 - I : matrice image résultat de l'action du module sur l'image cible. Si $I=[]$, l'image cible ne sera pas modifiée.
 - M : matrice $[M \times N \times P]$ de résultats numériques qui sera affichée dans l'éditeur matriciel de la boîte à outil si $M \neq []$.
- un retour en erreur sera provoqué par l'appel de fonction *error*.
- Un commentaire d'information précisant (i) l'action du module et son usage, (ii) le nom de l'auteur et la date de dernière modification, doit figurer en tête du fichier. Ces informations seront affichées par la commande *<information module>*

Pour exemple, voici le code d'un module effectuant la conversion d'une image couleur RGB en une image en niveau de gris. Seules les parties figurant en bleu sont spécifiques à ce module. Tous ce qui est écrit en noir est invariant d'un module à l'autre [4].

<pre>function varargout = RGBtoGRAY(varargin)</pre>	}	Déclaration du module RGBtoGRAY
<pre>% conversion RGB → GRAYSCALE % ENTREE : % - image % SORTIE % - image % % Isaac Macintosh – Alain Clément 040101</pre>	}	Commentaire d'information
<pre>switch nargin case 0 varargout{1} = {}; return case 1 img1 = varargin {1}; otherwise error('invalid number of output argument') end</pre>	}	Entrée sans paramètres
<pre>%-----  <pre>%-----</pre> </pre>	}	Action du module en Langage MATLAB
<pre>if nargin == 2 varargout{1} = img2; varargout{2} = []; else error (invalid number of output argument) end</pre>	}	Sortie avec 2 paramètres
<pre></pre>	}	Erreur sinon

Le nombre des paramètres acceptés en entrée par un module est illimité. Grâce à la structure très simple des modules et aux fonctions spécialisées du langage MATLAB, de nombreuses opérations d'analyse d'images peuvent être écrites en quelques lignes seulement.

4.2.2- Un module Isaac pour l'étiquetage en composantes connexes :

Le module que nous avons développé sous *Isaac* agit sur une image couleur codée en RVB sur Q bits (intensité de 0 à 2^Q) en plusieurs étapes pour donner en sortie les histogrammes 2D et 3D étiquetés en composantes connexes :

–*les entrées du module* sont :

- une image couleur
- une valeur de l'ensemble des valeurs suivant {12, 13, 23, 123} représentant les plans sur lesquels l'histogramme 3D va être projeté ; 12 pour les plans RV, 13 pour RB, 23 pour VB et 123 pour les trois plans RVB.
- un entier positif N ;
- deux valeurs de seuils S_1 et S_2

–*binarisation des histogrammes* ; à chaque point de l'histogramme, si la valeur de l'effectif en ce point est dans l'intervalle $[S_1, S_2]$, on lui affecte la valeur 1, sinon il prend la valeur 0. On aboutit à la fin de cette étape à des histogrammes sous forme de structures binaires.

–*Étiquetage en composantes connexes* des l'histogrammes 2D et 3D classiques selon un algorithme séquentiel en considérant le voisinage d'ordre 4 pour les histogrammes 2D et le voisinage d'ordre 6 pour les histogrammes 3D.

–*en sortie*, le module affiche l'image cible contractée et la matrice de l'histogramme considéré étiqueté.

Conclusion :

Nous avons appliqué dans cette partie un étiquetage en composantes connexes sur les histogrammes multidimensionnels dans le but de les segmenter. C'est en fait une étape de la segmentation de l'image cible. Cette étape nous permet en fait de détecter les pics dans les histogrammes qui correspondent aux régions dans l'image à segmenter. Certains traitements ont été effectués sur ces histogrammes à fin de les adapter aux algorithmes d'étiquetage en composantes.

Conclusion générale

Dans ce travail, nous avons étudié un thème très abordé en traitement d'images couleur qui est la segmentation d'image. L'intérêt de ce thème est de transformer l'image en informations interprétables.

La première partie présente la caractérisation de l'image couleur et ces éléments de géométrie discrète. Ces caractéristiques sont à la base de la majorité des traitements qu'on peut appliquer à une image couleur.

La deuxième partie présente les différentes méthodes classiques utilisées en segmentation d'images couleurs. Appliquées à une même image, ces méthodes ne donnent pas toujours le même résultat. Chaque méthode a ses propres caractéristiques et fonctionne mieux sur un type d'images que d'autre.

La troisième partie présente l'application du concept d'étiquetage en composantes connexes sur les histogrammes multidimensionnel (bidimensionnels et tridimensionnels) des images couleurs. Pour cela il a fallu effectuer quelques traitements sur ces histogrammes étant donné que les algorithmes d'étiquetages en composantes connexes ne sont applicables que sur des structures binaires.

Notre objectif est d'essayer d'adapter l'algorithme d'étiquetage en composantes connexes à l'histogramme compact vu les avantages que présente ce dernier par rapport à l'histogramme classique.

Conclusion générale

Dans ce travail, nous avons étudié un thème très abordé en traitement d'images couleur qui est la segmentation d'image. L'intérêt de ce thème est de transformer l'image en informations interprétables.

La deuxième partie présente la caractérisation de l'image couleur et ces éléments de géométrie discrète. Ces caractéristiques sont à la base de la majorité des traitements qu'on peut appliquer à une image couleur.

La troisième partie présente les différentes méthodes classiques utilisées en segmentation d'images couleurs. Appliquées à une même image, ces méthodes ne donnent pas toujours le même résultat. Chaque méthode a ses propres caractéristiques et fonctionne mieux sur un type d'images que d'autre.

La quatrième partie présente l'application du concept d'étiquetage en composantes .

connexes sur les histogrammes multidimensionnel (bidimensionnels et tridimensionnels) des images couleurs. Pour cela il a fallu effectuer quelques traitements sur ces histogrammes étant donné que les algorithmes d'étiquetages en composantes connexes ne sont applicables que sur des structures binaires.

Notre objectif est d'essayer d'adapter l'algorithme d'étiquetage en composantes connexes à l'histogramme compact vu les avantages que présente ce dernier par rapport à l'histogramme classique.

Bibliographie

- [1] Cocquerez J-P., Philippe S., (1995). *Analyse d'images : filtrage et segmentation*. Edition Masson, Paris, France.
- [2] Clément A., Vigouroux B. (2001). *Un histogramme compact pour l'analyse d'images multicomposantes*. Acte du 18^e colloque GRETSI sur le traitement du signal et des images. Toulouse, France, Vol.1 p.305-307.
- [3] Clément A. (2001). Isaac: une plateforme pour le développement et la mise en routine de méthodes d'analyse et de traitement d'images en couleurs. Acte du colloque national de la recherche dans les IUT, Roanne, France, vol.1, p. 413-422.
- [4] Clément A., (2002) *Algorithmes et outils informatiques pour l'analyse d'images couleur application à l'études de coupes histologiques de baies de raisin en microscopie optique*. Thèse 2002, université d'Angers, France.
- [5] Lambert P., Macaire L. (2000) *Filtering and segmentation : the specificity of color images*. CGIP 2000, saint Etienne, France, P.65-75.
- [6] Lezoray O. *segmentation d'image couleur par morphologie mathématique et classification de données par réseaux de neurones : application à la classification de cellules en cytologie des séreuses*. Thèse 2000 Université de caen, France.
- [7] R. Gonzales., R. E Woods.,(1993) *Digital image processing*. Edition Addison-Wesley. USA. P. 226-227.
- [8] Plataniotis K-N., Venetsanopoulos., (2000). *Color image processing and applications*. Edition Springer, Berlin, Allemagne.