

Table des matières

Introduction générale	5
Chapitre 1 : ANALYSE DE LA TEXTURE	6
I.1 Introduction :	6
I.2 Définition d'image :	6
I.3 Types d'image :	7
I.3.1 Windows BitMap : BMP.....	7
I.3.2 Joint Photographic Expert Group : JPEG.....	7
I.3.3 Graphics Interchange Format : GIF.....	7
I.3.4 Portable Network Graphic : PNG.....	7
I.4 Les caractéristiques de l'image :	8
I.4.1 Pixel :	8
I.4.2 HISTOGRAMME :	8
I.4.3 Histogramme cumulé :	8
I.4.4 Le Contraste :	8
I.4.5 L'étalement dynamique de l'image :	8
I.5 Traitement d'images	9
I.5.1 Filtrage :	9
I.6. la texture :	10
I.6.1 Perception et analyse visuelle d'une texture	11
I.6.1.1 Vision et perception de la texture par l'œil :	11
I.6.2 Notions d'attribut et de descripteur de texture :	11
I.6.3 Principales méthodes d'analyse de texture :	12
I.6.3.1 Méthodes Statistiques :	12
a) Les matrices de cooccurrence :	12
b) Spectre de texture	12

I.6.3.2 Méthodes structurelles :	12
I.6.3.3 Méthodes basées sur le filtrage :	13
a) Ondelettes :	13
b) Théorie de la morphologie mathématique :	13
I.6.3.4 Méthodes basées sur les modèles :	13
a) Méthodes markovienne :	13
b) Méthodes fractales :	14
I.7 Conclusion :	14
Chapitre 2 : Matrice de cooccurrence	15
II.1 Matrice de cooccurrence :	15
II.1.1 Définition :	15
II.1.2 Matrice de cooccurrence des niveaux de gris :	17
II.1.3 Caractérisation statistique des textures :	17
II.1.3.1 Moyenne locale :	18
II.1.3.2 Écart-type local :	18
II.1.3.3 Variance :	18
II.1.3.4 Obliquité :	18
II.1.3.5 Énergie :	18
II.1.3.6 Entropie :	18
II.1.3.7 Inertie (Contraste) :	19
II.1.3.8 Moment différentiel inverse (Homogénéité) :	19
II.1.3.9 Corrélacion :	19
II.1.3.10 Moments :	19
II.1.3.11 Probabilité maximale :	19
II.2. Les Filtres :	20
II.2.1 Filtre Moyen :	20
II.2.2 Filtre Nagao :	21
II.2.3 Filtre de Kurtosis :	22

Chapitre 3 :Application	24
III.1. Introduction :	24
III.2. Langage de programmation :.....	24
III.3. Description de l'application :	24
III.3.1 Ouvrir et Actualisé :	25
III.3.2 Menu « Fichier » :	25
III.3.3 Menu « Enregister IMG » :.....	25
III.3.4 Menu « Aide ! »	26
III.3.5 Parametre :	26
III.3.6 Filtrage :	29
III.3.7 Matrice de coocurrence :	34
III.3.4 Conclusion :	35
Conclusion générale	36
Références Bibliographiques	37

Liste des figures

Figure I.1 : Exemple de forme de masque	13
Figure I.2 : Exemples d'images de l'album de Brodatz	9
Figure II.1 : Exemples des directions de la matrice de cooccurrence	17
Figure II.2 : (a) image originale, (b) filtre moyen de l'image	20
Figure II.3 : les 9 fenêtres pour le calcul de la variance	21
Figure II.4 Image Original	23
Figure II.5 : des paramètres de filtrage sur l'image Original	23
Figure III.1 : Calcul des descripteurs statistiques d'ordre 1 d'une Image en format (.Bmp).....	28
Figure III.2 : Exemple du filtre Moyenne	29
Figure III.3 : Exemple du filtre Variance.....	29
Figure III.4 : Exemple du filtre Kurtosis	30
Figure III.5 : Exemple du filtre Nagao.....	30
Figure III.6 : Calcule les indices d'Haralick avec un niveau de gris de 256 et une direction 0°.....	34
Figure III.7 : Calcule les indices d'Haralick avec un niveau de gris 256 et une direction de 45°.....	35

Introduction générale

Le traitement d'image consiste à modifier une image et de rendre apparent ou de cacher des choses dans l'image. Dans le domaine du traitement de l'image une texture est un champ de l'image qui apparait comme un domaine cohérent et homogène, c'est à dire formant tout pour un observateur, l'analyse de texture est un domaine très important en traitement d'images, parmi les principaux éléments d'interprétation du message visuel pour un observateur humain on peut citer les contours, la couleur, la forme, etc. Ainsi que le filtrage c'est d'améliorer la qualité visuelle de l'image ou extraire des attributs de l'image, en modifiant la valeur de niveau du gris d'un pixel en fonction de la valeur de ses voisins

Parmi les différents approches de l'analyse de texture on cite les matrices de cooccurrences, le spectre de texture, les ondelettes, les modèles de markovienne, fractales et la morphologie mathématiques, on s'intéressé aux matrices de cooccurrence et aux filtrages

La structure de notre mémoire se compose comme suit :

Un premier chapitre présente les différentes approches de l'analyse de la texture

quand au deuxième chapitre expose les descripteurs statistiques du 1^{er} ordre de la matrice de cooccurrence et le filtrage.

Un troisième chapitre décrit notre application avec les différents résultats obtenus

Et enfin on termine par une conclusion et perspectives

CHAPITRE I

Analyse de la Texture

I.1 Introduction

L'analyse de texture est un domaine très important en traitement d'images. Parmi les principaux éléments d'interprétation du message visuel pour un observateur humain on peut citer les contours, la couleur, la forme, etc... L'analyse de l'image consiste souvent à extraire un certain nombre de propriétés caractéristiques et à les exprimer sous forme paramétrique. Les paramètres calculés permettent donc de décrire, de caractériser, de segmenter et d'analyser les images en question. Selon le cas, l'analyse peut être globale ou locale. Il est évident que le choix des paramètres dépend surtout de l'application considérée. Le but peut être de lier par exemple ces paramètres avec les propriétés physiques et biologiques réelles afin de les quantifier ou alors de trouver des similitudes avec des textures de référence afin de les identifier. Il faut tout de même signaler que l'interprétation des informations dans un environnement naturel n'est pas un problème simple. En effet, les textures naturelles sont très irrégulières et ne peuvent être parfaitement modélisées par les techniques mathématiques actuelles.

I.2. Définition d'image

La définition d'une image avec sa notion connexe de résolution définit le niveau de détails qui seront visibles dans l'image. Pour une taille donnée, plus il y aura de pixels, plus il y aura de détails fins visibles. On dit que plus une image a de pixels, plus elle est de grande qualité. Une image numérisée avec une définition de 640×480 pixels (donc contenant 307 200 pixels) apparaîtra très approximative et sous forme d'un pavage de petits carrés de couleur, par comparaison à une image de 1280×1024 px (1 310 720 pixels).

I.3. Types d'image

I.3.1 Windows BitMaP : BMP

Le format BMP est un des premiers formats d'image utilisé sous Windows. C'est un ancien format (.bmp) très facultatif qui crée inutilement des fichiers volumineux. Principal avantage : pas de compression ni de perte de qualité.[4]

I.3.2 Joint Photographic Expert Group : JPEG

Ce format offre des taux de compression inégalés, même si la qualité de l'image s'en ressent au fur et à mesure que vous augmentez la compression. Avec des taux de compression élevés donnant lieu à des fichiers images de petite taille. [4]

I.3.3 Graphics Interchange Format : GIF

Ce format est l'autre standard d'internet. Les fichiers gif sont de petites tailles, ce qui est dû au fait que ces images ne peuvent enregistrer que 256 couleurs. [3]

I.3.4 Portable Network Graphic : PNG

C'est le format appelé à devenir le futur standard internet. Comme le gif il permet le détourage des images, mais là où le format gif enregistre 256 couleurs, le png en retient 16.7 MILLIONS ce qui offre une image parfaite, avec un excellent rendu des nuances et des dégradés. [3]

I.4. Les caractéristiques de l'image

I.4.1 Pixel

Le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. [4]

I.4.2 HISTOGRAMME

un histogramme peut être graphique ou numérique, il indique la fréquence de répétition d'une valeur radiométrique dans une image et donne l'allure générale sur la dynamique de l'image et sert à déterminer les seuils pour future étalement de la dynamique.[5]

I.4.3 Histogramme cumulé

L'histogramme cumulé d'une image est le calcul du taux (pourcentage) de pixels ayant une valeur inférieure à un niveau de couleur donné, pour chaque niveau de couleur (ou de gris) d'une image, et ce pour chaque canal Rouge Vert Bleu de l'image.[6]

I.4.4 Le Contraste

Le contraste est une opposition entre deux choses, Le *contraste* est un caractère de la répartition lumineuse d'une image ou entre deux points d'une image (ou « pixels », en photographie numérique). [4]

I.4.5 L'étalement dynamique de l'image

L'étalement de la dynamique permet de faire étaler la dynamique entre 0 et 255 d'une image qui a une dynamique réduite (mauvaise interprétation). Il sert surtout pour une amélioration de contraste (interprétation aisée) ou pour d'autres traitements comme la composition colorée, les classifications, etc ...

Prenons comme exemple l'étalement linéaire de la dynamique, elle se résume par la formule suivante :

$$\text{ImC}(i, j) = (\text{ImB}(i, j) - \text{Min} / \text{Max} - \text{Min}) * 255$$

ImC(i, j) image étalé

ImB(i, j) image brute

Min : minimum de pixel de l'image

Max : maximum de pixel de l'image

I.5. Traitement d'images

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo. [11]

I.5.1 Filtrage

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. [7]

L'objectif principal du filtrage est d'améliorer la qualité visuelle de l'image ou extraire des attributs de l'image, en modifiant la valeur de niveau du gris d'un pixel en fonction de la valeur de ses voisins (figure I.1), les éléments significatifs de l'image doivent être préservés au mieux.

Les voisinages du pixel à traiter est fonction de la forme du masque choisi par le filtre.

La figure suivante donne quelques exemples de masques (les cases en vert : le pixel central, les cases oranges : les pixels voisins)

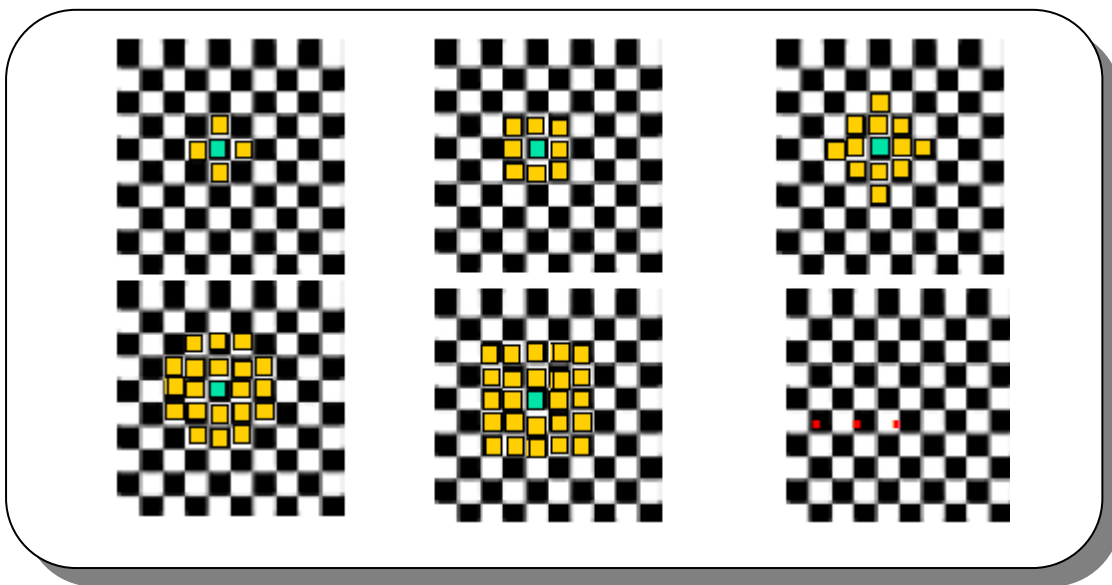


Figure I.1 : Exemple de forme de masque.

I.6. la texture

La définition littéraire de la texture est la suivante : « répétition spatiale d'un même motif dans différentes directions de l'espace ». Cette définition est limitative car elle caractérise l'objet indépendamment d'un observateur humain. La notion de texture est utilisée pour traduire un aspect homogène de la surface d'un objet sur une image.

Haralick élargit la définition en décrivant une texture comme un phénomène à deux dimensions : la première concernant la description d'éléments de base ou primitives (le motif) à partir desquels est

formée la texture; la deuxième dimension est relative à la description de l'organisation spatiale de ces primitives.

La figure (I.2) donne cinq images tirées de l'album de Brodatz représentant dans l'ordre : clôture (D6), gazon (D9), cailloux (D23), grain du bois (D68) et mur de briques (D94). Nous pouvons catégoriser chaque texture selon les qualificatifs suivants : périodique, quasi-périodique, aléatoire, grossière, fine, uniforme, contrastante, régulière, irrégulière, directionnelle, etc. [8]

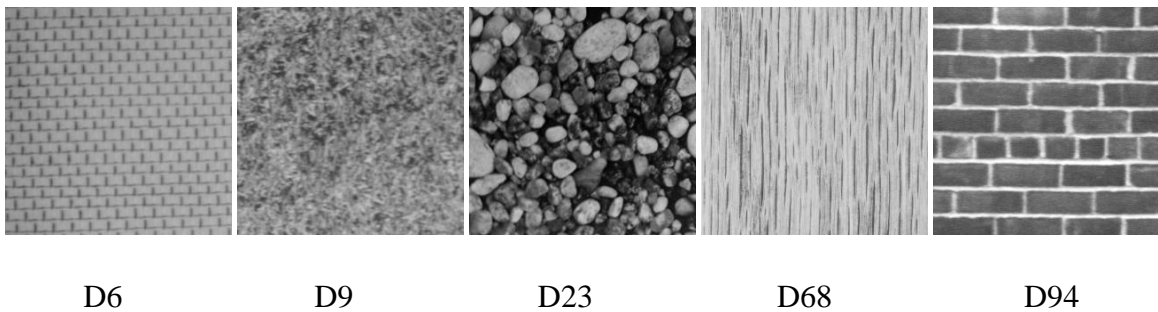


Figure I.2 : Exemples d'images de l'album de Brodatz

I.6.1 Perception et analyse visuelle d'une texture

I.6.1.1 Vision et perception de la texture par l'œil

L'œil humain distingue en moyenne 16 niveaux de gris différents du noir au blanc. Un œil peu exercé en voit moins, alors qu'un professionnel peut en distinguer jusqu'à 20. En revanche, il est capable d'apprécier des différences de niveaux voisins de 2% (contraste). La résolution de l'œil (pouvoir séparateur) est de 1 min d'angle maximum, ce qui correspond à deux points séparés de 150 μm vus à une distance de 40 cm. De plus il existe une zone de l'espace auquel l'œil n'est pas sensible. On ne peut donc parcourir une image en une seule fois; on la décompose instinctivement en plusieurs parties. [9]

I.6.2 Notions d'attribut et de descripteur de texture

L'étude de la texture des objets d'une image peut avoir des objectifs très divers : obtenir des informations sur la nature d'un objet, segmenter l'image en régions homogènes, améliorer la

qualité de l'image (restauration), identifier la texture afin de la réduire à un ensemble de paramètres (compression d'images) etc...

La discrimination et l'analyse de la texture nécessitent l'extraction d'un ou de plusieurs paramètres caractéristiques de cette texture. Nous désignerons ces paramètres sous le terme d'attributs texturaux ou indices (en anglais : textural features) et l'ensemble qu'ils constituent sous le terme de descripteur de texture. Certains de ces paramètres correspondent à une propriété visuelle de la texture.

Les attributs texturaux peuvent être obtenus à partir d'un ensemble assez vaste de différentes théories mathématiques. [9]

I.6.3 Principales méthodes d'analyse de texture

Il existe plusieurs méthodes d'analyse de la texture, qui s'appuient sur des approches statistiques ou sur des approches structurelles ou sur des modèles [10] :

I.6.3.1 Méthodes Statistiques

a) Les matrices de cooccurrence

La matrice de cooccurrence mesure la probabilité pour qu'un couple de niveaux de gris, vérifiant une loi spatiale donnée, apparaisse dans l'image, le niveau de gris d'un pixel d'une image dépend fortement des niveaux de gris des pixels voisins. La légitimité de la démarche provient d'une modélisation Markovienne des textures : la connaissance de toute l'image autour d'un pixel donné est pratiquement équivalente à la connaissance du seul voisinage autour de ce pixel. Haralick a défini 14 indices à partir de cette matrice pour résumer les diverses propriétés de la texture, dans le but d'intégrer cette information dans une classification. [10]

b) Spectre de texture

Le spectre de texture, est une approche originale d'analyse de texture développée par Wang et He . Elle se base sur la comparaison, dans une fenêtre mobile de 3 x 3 pixels, du pixel central avec ses huit pixels voisins selon trois possibilités : inférieur, égale ou supérieur. On obtient donc une transformée d'image avec une possibilité de 3^8 ou 6561 unités de texture différentes dont l'histogramme unidimensionnel est appelé spectre de texture. [10]

I.6.3.2 Méthodes structurelles

Les méthodes structurelles considèrent la texture comme un ensemble d'éléments de base. Cette approche propose des modèles qui permettent à partir d'une texture idéale de synthétiser une texture observée correspondant à une version distordue de la première. La texture idéale est composée de primitives régulières (exemple le carré) disposées régulièrement. Des transformations déterministes ou stochastiques sont appliquées pour générer la texture distordue.

I.6.3.3 Méthodes basées sur le filtrage

a) Ondelettes

Cette technique a été développée dans les années 90, testée pour l'analyse et la classification des textures des images satellitaires. L'intérêt pour les ondelettes s'est largement développé ces dernières années. Cette méthode se rapproche des filtres de Gabor. [11]

b) Théorie de la morphologie mathématique

La morphologie mathématique s'intéresse à la forme des objets. Le principe de base de cette méthode consiste à comparer les objets que l'on veut analyser à un autre objet de forme connue appelé élément structurant. On peut le représenter comme une matrice binaire affichant un objet de forme généralement très simple, auquel les objets de l'image seront comparés. [11]

I.6.3.4 Méthodes basées sur les modèles

Ces méthodes consistent à choisir un modèle pour représenter le signal. Ce modèle est défini par un certain nombre de paramètres qui, entre autre, peuvent caractériser la texture. [11]

a) Méthodes markovienne

Les modèles de Markov s'intéressent à l'aspect aléatoire des textures en considérant que même structurées, les textures ne sont "jamais" parfaitement régulières. Dans ces modèles, l'analyse des textures se fait dans un cadre stochastique lié à la notion de voisinage, où le niveau de gris d'un pixel dépend uniquement des niveaux de gris des pixels voisins. [11]

b) Méthodes fractales

La théorie fractale a été introduite par Mandelbort pour décrire la géométrie des objets naturels qui présentent une similarité statistique entre les parties constituant ces objets, c'est-à-dire que leurs formes géométriques restent statistiquement invariables par changement d'échelle. L'idée de base de cette théorie est la suivante : la géométrie euclidienne classique, avec des figures lisses et régulières, est incapable de donner une représentation fidèle du monde qui l'entoure. Beaucoup de structures naturelles sont très régulières, trop rugueuses, trop fragmentées ou, les trois à la fois, pour pouvoir être décrites par la géométrie classique. Le grand succès de cette méthode est dû au fait qu'elle est très adaptée à l'étude des phénomènes naturels. La télédétection, en tant que science de l'observation des milieux naturels, peut avoir une place privilégiée dans les études fractales.[11]

I.7 Conclusion

Les méthodes d'analyse de texture sont riches et variées. De nombreuses méthodes permettant le calcul d'attributs de texture ont été développées durant ces 30 dernières années. Leur domaine d'application est en général limité. Le choix des attributs est en effet lié à la tâche que l'on souhaite effectuer.

L'analyse de texture est utilisée dans divers domaines. On peut citer, la restauration d'une partie dégradée ou manquante dans une image en la remplaçant par une version synthétique générée à partir du modèle de texture, la segmentation des images ou la reconnaissance des formes, la classification et la compression d'images ...

CHAPITRE II

Matrice de cooccurrence et Filtrage

II.1 Matrice de cooccurrence

Les matrices de cooccurrence sont analogues à des histogrammes bidimensionnels. Elles représentent le nombre d'occurrence de couples de pixels particuliers dans l'image. La principale caractéristique des matrices de cooccurrence est de s'intéresser à des couples de pixels qui sont, par définition, séparés par une distance (1, 2, 3, 4...). [12]

II.1.1 Définition

La matrice de cooccurrence $Mc(i, j)$ est carrée et de dimension $L \times L$, où L est le nombre de niveaux de gris présents dans B . Les indices de la matrice de cooccurrence sont donc les niveaux de gris de la texture étudiée. La matrice $Mc(i, j)$ se construit comme suit :

1. Initialisation de la matrice $\forall i, j \in [0, L[: Mc(i, j) = 0$.
2. Remplissage de la matrice. Si la relation R entre deux pixels (x_1, y_1) et (x_2, y_2) est respectée:

$$Mc (f(x_1, y_1), f(x_2, y_2)) = MC (f(x_1, y_1), f(x_2, y_2)) + 1$$

Dès lors, $Mc(i, j)$ contient le nombre de fois que l'on a rencontré dans B deux pixels (x_1, y_1) , (x_2, y_2) vérifiant la relation géométrique R et tel que $f(x_1, y_1) = i$ et $f(x_2, y_2) = j$. En pratique, on utilise des relations géométriques assez simples comme celle citée comme exemple. Ainsi, on définit une matrice orientée

$$P_{90^\circ}, d(i, j) = (x_1, y_1), (x_2, y_2) \in B \quad x_1 = x_2, |y_2 - y_1| = d, f(x_1, y_1) = i \text{ et } f(x_2, y_2) = j$$

90° = correspond à une direction verticale.

d = détermine la distance entre les deux pixels.

$$P_{0^\circ}, d(i, j) = (x_1, y_1), (x_2, y_2) \in B \quad x_1 = x_2, |y_2 - y_1| = d, f(x_1, y_1) = i \text{ et } f(x_2, y_2) = j$$

0° = correspond à une direction horizontale.

d = détermine la distance entre les deux pixels.

Observation : Il est également possible de définir d'autres matrices correspondant aux directions 45° et 135° .

Un exemple sur la matrice de cooccurrence :

Soit l'image suivante comportant $L = 4$ niveaux de gris ($l = 0, 1, 2, 3$) :

$Mc(x, y)$ est la matrice de cet image :

$$Mc(x, y) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

Les matrices $P_{0,1}^0$ et $P_{90,1}^0$ sont donc de dimension 4 et valent

$$P_{0,1}^0 = \begin{bmatrix} 6 & 2 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 4 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix} \quad P_{90,1}^0 = \begin{bmatrix} 6 & 1 & 2 & 0 \\ 1 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

On peut remarquer que les matrices de cooccurrence sont bien symétriques. [13]

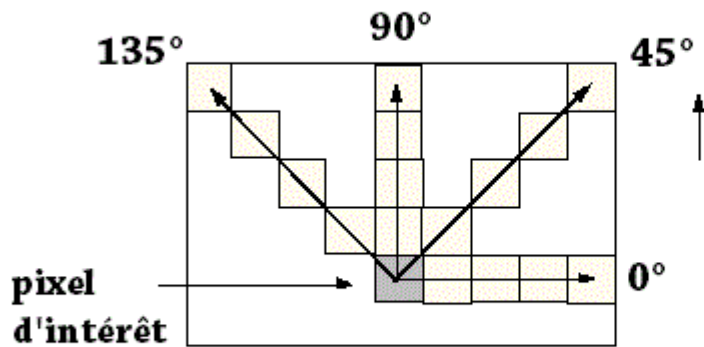


Figure II.1 : Exemples des directions de la matrice de cooccurrence

II.1.2 Matrice de cooccurrence des niveaux de gris

Les descripteurs statistiques décrits précédemment ne fournissent pas d'information concernant la nature périodique qu'une texture peut présenter.

La matrice de cooccurrence introduite ci-après va nous offrir un premier moyen de caractériser la périodicité et la directivité des textures.

L'idée consiste à étudier la texture au moyen d'une fenêtre B d'observation de la texture en comptabilisant le nombre de paires de pixels distants de d qui présentent une différence Δ_z en niveaux de gris. En plus de la distance, on tient compte de la direction définie par toute paire de pixels. [13]

II.1.3 Caractérisation statistique des textures

Les matrices de cooccurrences contiennent les moyennes d'espace du premier ordre. Plusieurs indices ont été proposés par Haralick qui correspondent à des caractères descriptifs des textures peuvent être calculés à partir de ces matrices :

II.1.3.1 Moyenne locale

$$\text{Moyenne} = \sum_{i=0}^{N_G-1} \sum_{j=0}^{N_G-1} Mc(i, j)$$

II. 1.3.2 Écart-type local

L'écart-type mesure l'énergie locale d'une texture :

$$\sigma_i^2 = \sum_{i=0}^{N_G-1} (i - \mu_i)^2 \sum_{j=0}^{N_G-1} Mc(i, j)$$

$$\sigma_j^2 = \sum_{j=0}^{N_G-1} (j - \mu_j)^2 \sum_{i=0}^{N_G-1} Mc(i, j)$$

II. 1.3.3 Variance

Donne l'écart carré moyen des niveaux de gris par rapport à la moyenne c.à.d il mesure l'ampleur des variations d'intensité par rapport à la moyenne

$$\text{Variance} = \sum_{i=0}^{N_G} \sum_{j=0}^{N_G} (i - \text{moyenne})^2 Mc(i, j)$$

II. 1.3.4 Obliquité

$$\text{Obli} = \frac{1}{\text{ecart}^3} \sum_{i=0}^{N_G} \sum_{j=0}^{N_G} (i - \text{moyenne})^2 Mc(i, j)$$

II. 1.3.5 Énergie

$$\text{Energie} = \sum_{i=0}^{N_G-1} \sum_{j=0}^{N_G-1} Mc(i, j)^2 \quad (\text{Second moment angulaire})$$

II. 1.3.6 Entropie

Faible si on a souvent les mêmes couples de pixels. C'est un indicateur de désordre : [13]

$$\text{Entropie} = - \sum_{i=0}^{N_G-1} \sum_{j=0}^{N_G-1} Mc(i, j) \log Mc(i, j)$$

II. 1.3.7 Inertie (Contraste)

plus la texture est contrastée, plus ce terme est grand : [13]

$$\text{Inertie} = \sum_{i=0}^{N_G-1} \sum_{j=0}^{N_G-1} (i-j)^2 Mc(i, j) \quad (\text{Contraste})$$

II. 1.3.8 Moment différentiel inverse (Homogénéité)

Plus on retrouve le même couple de pixels, plus cet indice est élevé, par exemple image uniforme, ou texture périodique dans le sens de la translation : [13]

$$\text{Moment différentiel inverse} = \sum_{i,j} \frac{1}{1+(i+j)^2} Mc(i, j)$$

II. 1.3.9 Corrélation

Décrit les corrélations entre les lignes et les colonnes de la matrice de cooccurrence :

$$\text{Corrélation} = \sum_{i=0}^{N_G-1} \sum_{j=0}^{N_G-1} \frac{(i-\mu_i)(j-\mu_j)}{\sigma_i \sigma_j} Mc(i, j)$$

II. 1.3.10 Moments

Ce descripteur a de faibles valeurs lorsque les éléments diagonaux de Mc sont grands. L'effet inverse peut être obtenu grâce aux moments de la différence inverse : [13]

$$\text{Moment} = \sum_{i=0}^{N_G-1} \sum_{j=0}^{N_G-1} (i-j) Mc(i, j)$$

II. 1.3.11 Probabilité maximale

C'est simplement la valeur maximale normalisée d'une des matrices ou de toutes les matrices de cooccurrence considérées. [13]

$$\text{où } \mu_i = \sum_{i=0}^{N_G-1} i \sum_{j=0}^{N_G-1} Mc(i, j) \quad (\text{moyennes des lignes})$$

$$\text{où } \mu_j = \sum_{j=0}^{N_G-1} j \sum_{i=0}^{N_G-1} Mc(i, j) \quad (\text{moyennes des colonnes})$$

où Mc est la matrice de cooccurrence

II.2. Les Filtres

II.2.1 Filtre Moyen

Ce filtre lisseur part du principe que la valeur d'un pixel est relativement similaire à son voisinage. Il fait donc en sorte que chaque pixel est remplacé par la moyenne pondérée de ses voisins. Si on applique un filtre moyenneur de taille $\lambda=3$, cela signifie qu'on additionne la valeur de tous les pixels du voisinage du pixel traité. On obtient ainsi la matrice de convolution suivante [1] :

$$h = \frac{1}{9} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

h s'appelle le masque de convolution. La somme des coefficients du masque valant 1, le lissage préservera toute zone de l'image où le niveau de gris est constant. Prenons ce koala et appliquons-lui un filtre moyenneur.



(a)



(b)

Figure II.2 : (a) image originale, (b) filtre moyen de l'image

Le résultat est que l'image devient plus floue, en particulier ses contours. On perd également le détail de la fourrure du koala. Ce filtre est isotrope.

II.2.2 Filtre Nagao

Ce filtrage doit enlever le bruit des zones homogènes tout en conservant les contours, dans l'optique d'un prétraitement pour une segmentation par régions.

Le filtrage de Nagao, qui a été présenté en 1979 par M.Nagao et T.Matsuyama, semble répondre à ces impératifs.

Le principe est le suivant : pour chaque pixel de l'image, on examine les neuf fenêtres suivantes, centrées sur ce pixel :

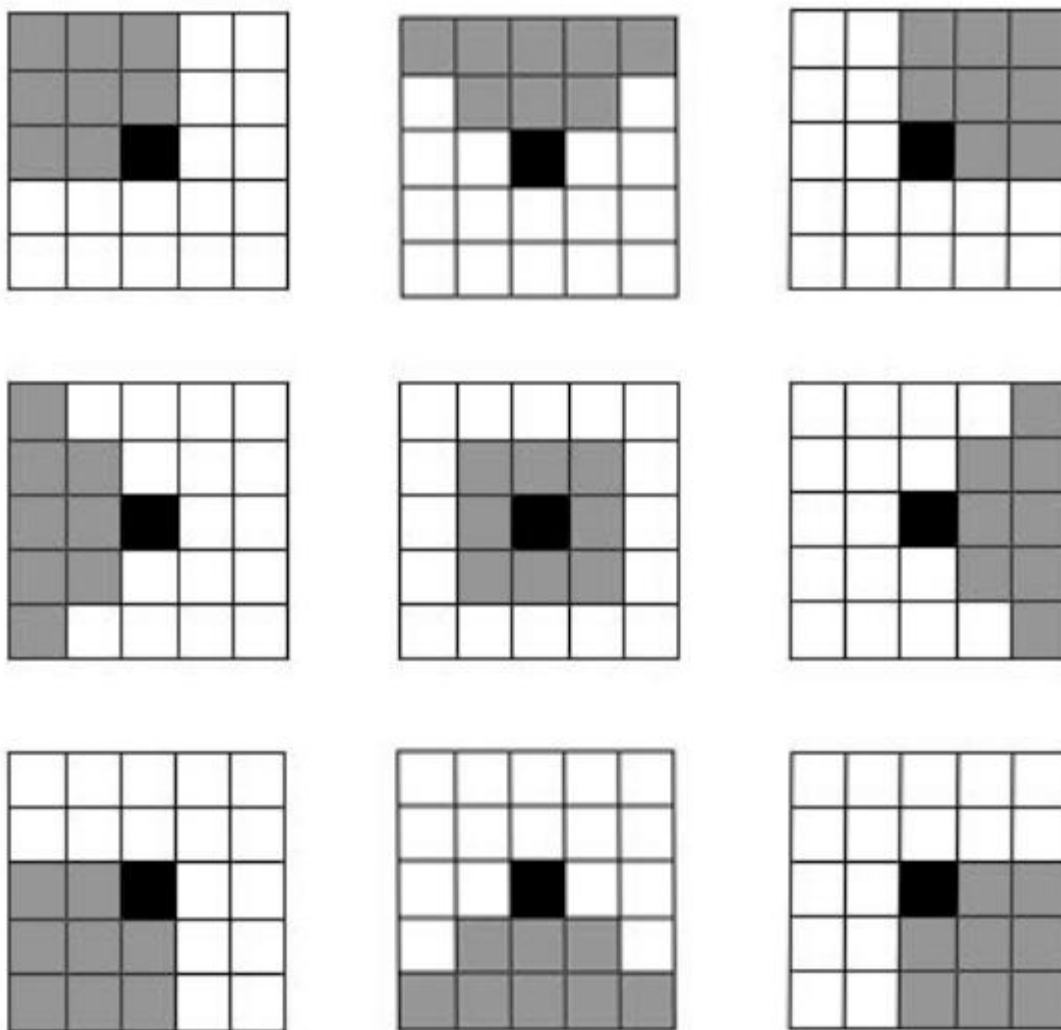


Figure II.3 : les 9 fenêtres pour le calcul de la variance

Pour chacune de ces neuf fenêtres on calcule la variance. Une variance faible indique une zone homogène, et une variance forte indique un changement important de niveau, typiquement un contour. Puisque l'on veut préserver les contours, on choisit la fenêtre de plus petite variance, puis on fait la moyenne sur la valeur des pixels de cette fenêtre. On ne fait donc plus la moyenne entre les pixels d'une zone homogène et les pixels d'un contour, ce qui permet de garder les contrastes tout en éliminant les points aberrants. [2]

II.2.3 Filtre de Kurtosis

Caractérise la forme du sommet de l'histogramme des niveaux de gris de l'image :

$$\text{Kurt} = \frac{1}{\text{ecart}^4} \sum (1 - \text{Moyenne})^4 \text{Mc}(i,j) - 3$$



Figure II.4 Image Original

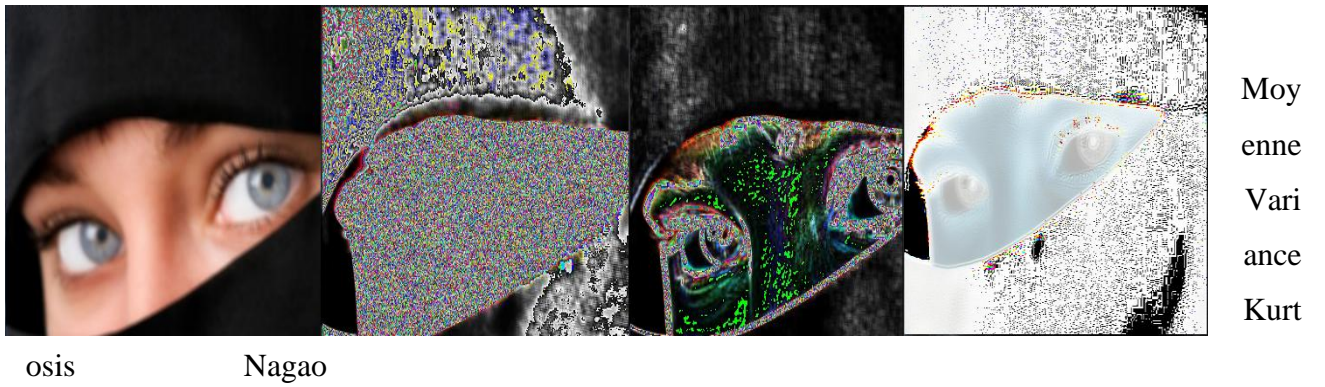


Figure II.5 : des paramètres de filtrage sur l'image Original

CAHPITRE III

Application

III. 1. Introduction

Dans ce chapitre nous présentons notre application et les résultats obtenus en appliquant les différents descripteurs de la matrice de cooccurrence et les différents filtres, ainsi que la description de l'environnement de travail

III.2. Langage de programmation

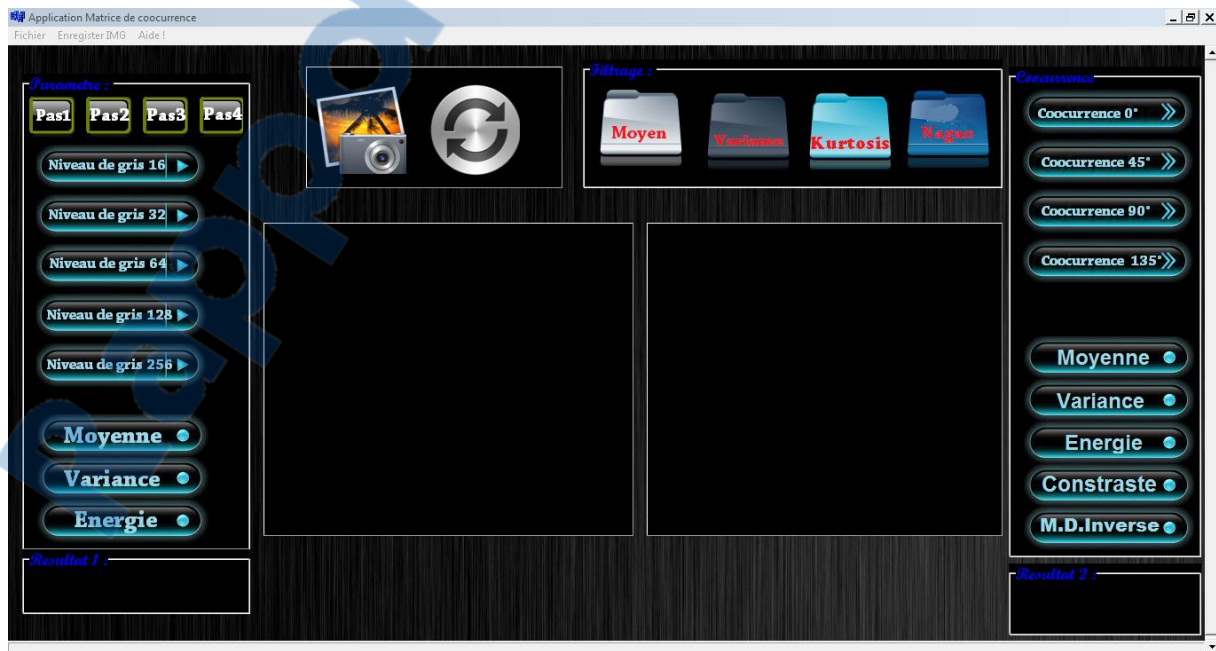
Notre logiciel est implémenté sous l'environnement de programmation C++ Builder version 6 qui produit de Borland.

Le Borland C++ est un outil RAD, qui permet le développement rapide d'application (Rapid Application Development) sous Windows, En un mot, C++ Builder permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc ...).

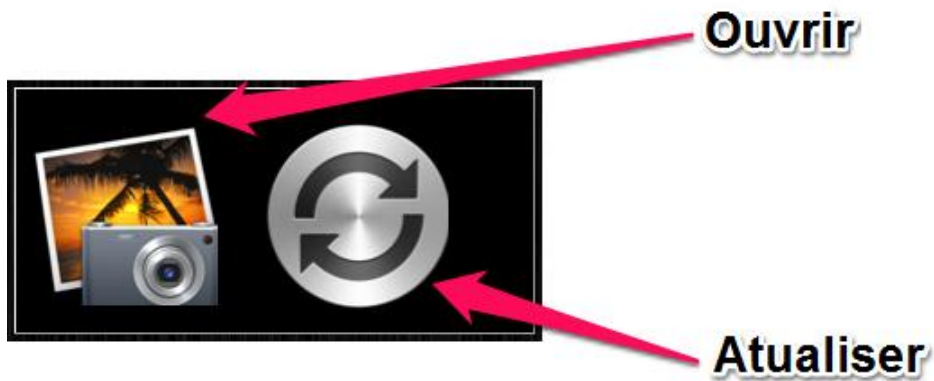
III.3. Description de l'application

Notre programme se présente comme suit :

Fenêtre principale de l'application



III.3.1 Ouvrir et Actualisé



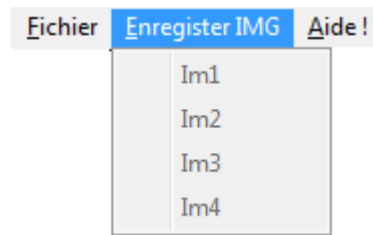
- L'icône « Ouvrir » nous permet d'ouvrir une image en format Bmp .
- L'icône « Actualiser » permet d'actualiser l'application.

III.3.2 Menu « Fichier »



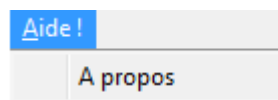
1. Le menu « ouvrir » nous permet d'ouvrir une image.
2. Le menu « Imprimer » nous permet d'imprimer une image.
3. Le menu « Quitter » permet de quitter l'application.

III.3.3 Menu « Enregister IMG »



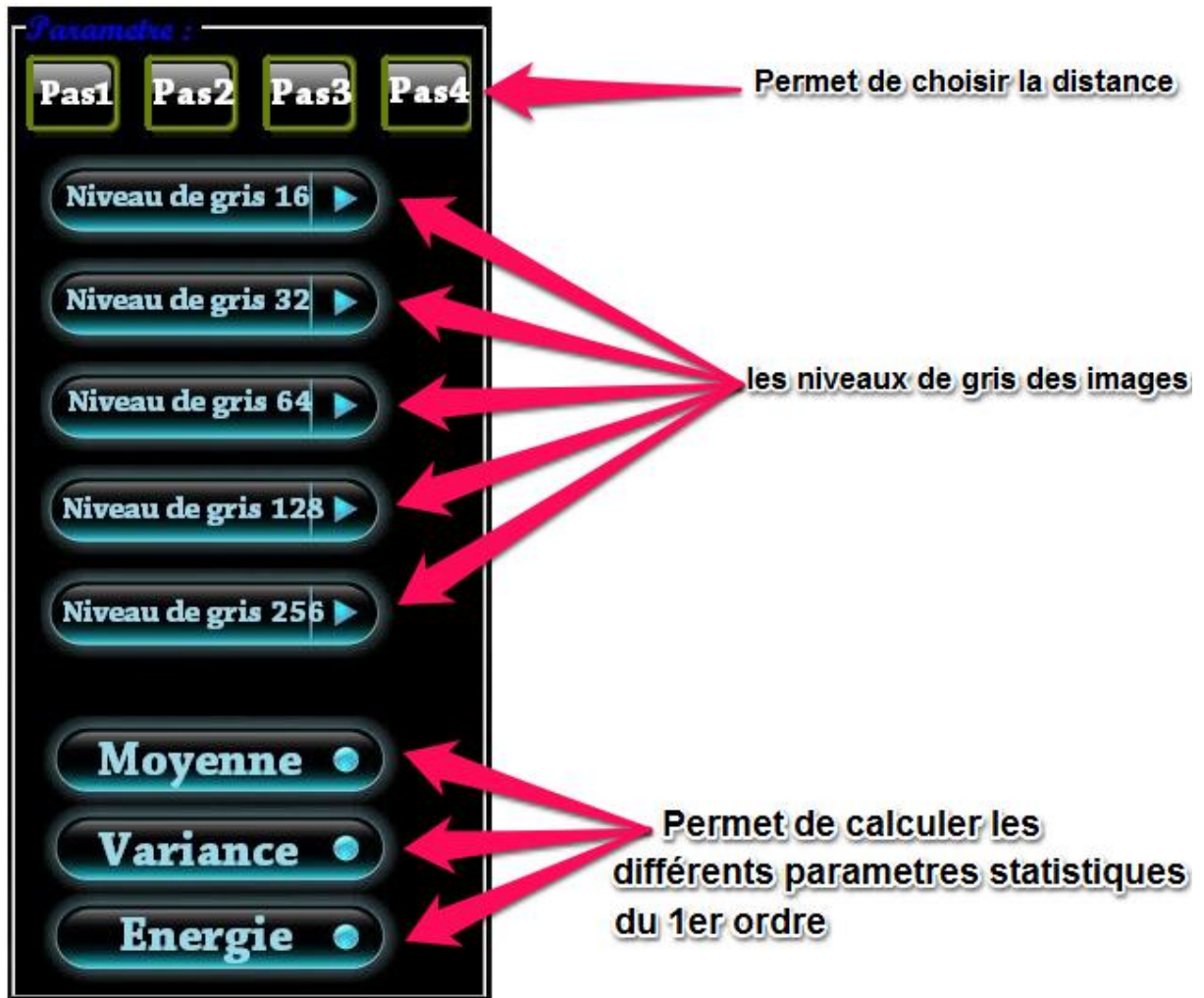
- Le menu « Enregistrer IMG » permet d'enregistrer les images résultats (Im1, Im2, Im3, Im4) selon les quatre filtres correspondant.

III.3.4 Menu « Aide ! »



- Le menu « A propos » permet d'expliqué la matrice de cooccurrence

III.3.5 Parametres



La fonction qui calcule le niveau de gris 16 :

```
niv=16;

    max=MaxBlue(width,height);

    max1=MaxRed(width,height);

    max2=MaxGreen(width,height);

    min=MINBlue(width,height);

    min1=MINRed(width,height);

    min2=MINGreen(width,height);

    for(i=0;i<width;i++)

        {

            for(j=0;j<height;j++)

                {

                    rgb[i][j].r=ICM(rgb[i][j].r,max,min);

                    rgb[i][j].g=ICM(rgb[i][j].r,max1,min1);

                    rgb[i][j].b=ICM(rgb[i][j].r,max2,min2);

                }

        }

    for(i=0;i<width;i++)

        for(j=0;j<height;j++)

            {

                red=rgb[i][j].r;

                green=rgb[i][j].g;

                blue=rgb[i][j].b;
```

<pre> for(i=0;i<width;i++) { for(j=0;j<height;j++) { a=a+rgb[i][j].r; b=b+rgb[i][j].g; c=c+rgb[i][j].b; } } moy=a/(width*height); moy1=b/(width*height); moy2=c/(width*height); moy3=((moy+moy1+moy2)/3); </pre>	<pre> for(i=0;i<width;i++) { for(j=0;j<height;j++) { a=a+(rgb[i][j].r*rgb[i][j].r); b=b+(rgb[i][j].g*rgb[i][j].g); c=c+(rgb[i][j].b*rgb[i][j].b); } } moy=a/(width*height); moy1=b/(width*height); moy2=c/(width*height); ene=((moy+moy1+moy2)/3); </pre>
---	--

« Moyenne »

« Variance »

```

m=moy3*moy3;

for(i=0;i<width;i++)
{

for(j=0;j<height;j++)
{ a=a+((rgb[i][j].r-
m)*(rgb[i][j].r-m));
b=b+((rgb[i][j].g-
m)*(rgb[i][j].g-m));
c=c+((rgb[i][j].b-

```



```
m)*(rgb[i][j].b-m));  
    }  
}  
var=a/(width*height);  
var1=b/(width*height);  
var2=c/(width*height);  
  
var3=((var+var1+var2)/3);
```

« Energie »

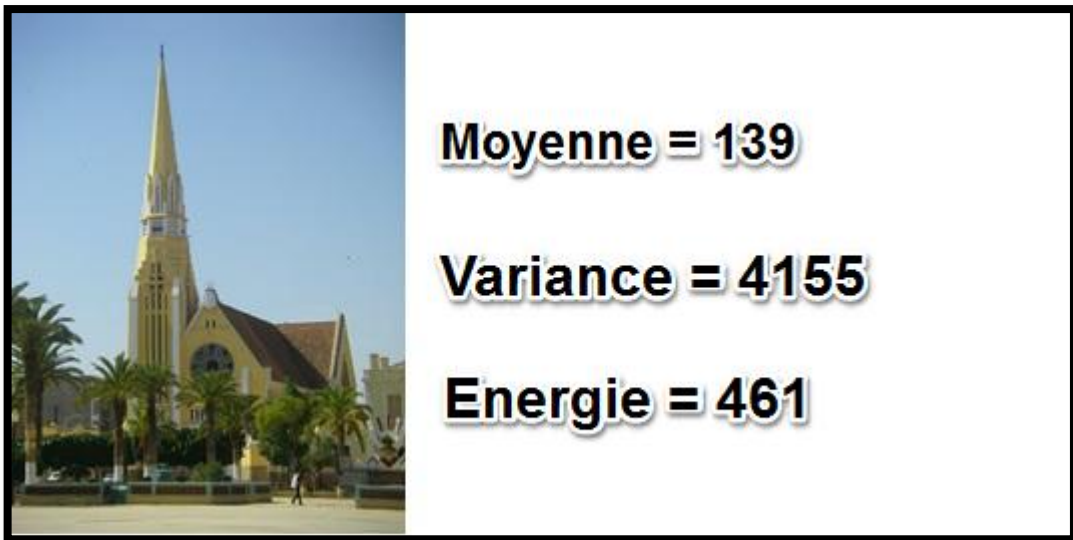


Figure III.1 : Calcul des descripteurs statistiques d'ordre 1 d'une Image en format (.Bmp)

III.3.6 Filtrage



Figure III.2 : Exemple du filtre Moyenne

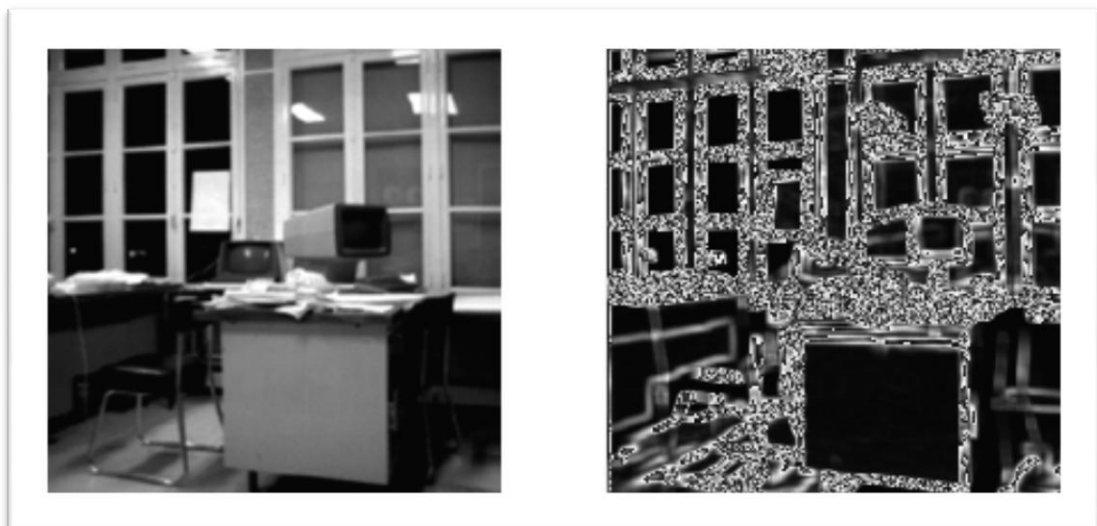


Figure III.3 : Exemple du filtre Variance

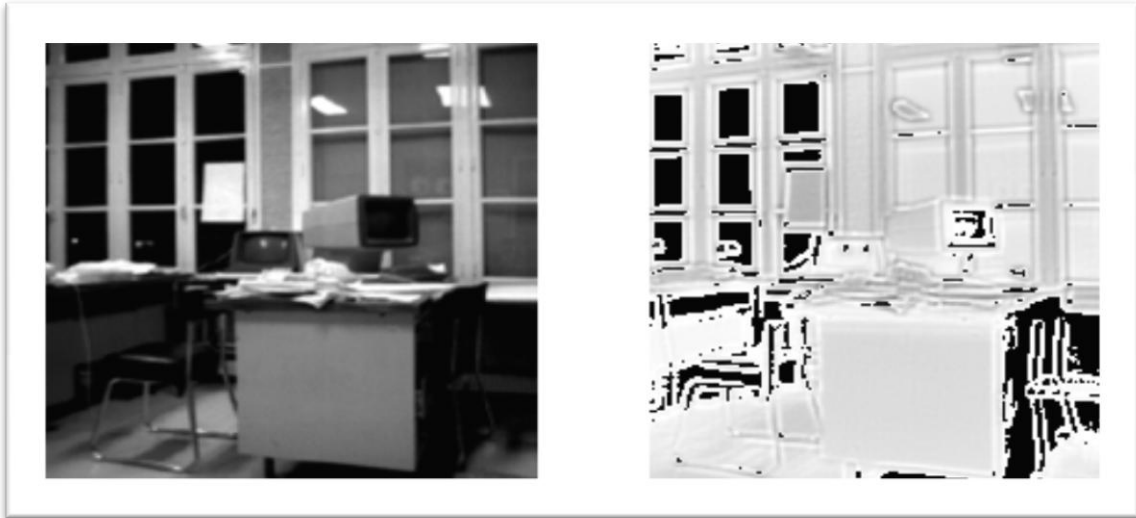


Figure III.4 : Exemple du filtre Kurtosis



Figure III.5 : Exemple du filtre Nagao

Code source du filtre de Nagao :

```
for(i=0;i<width;i++)
    for(j=0;j<height;j++)
    {
        if(n==5)
        {
            if(i==0||i==1||i==width-1||i==width-2||j==0||j==1||j==height-1||j==height-2)
            {
                red=rgb[i][j].r;
                green=rgb[i][j].g;
                blue=rgb[i][j].b;
            }
            else
            {
                d1=((rgb[i-2][j-2].r*rgb[i-2][j-2].r)+
                    +(rgb[i-1][j-2].r*rgb[i-1][j-2].r)
                    +(rgb[i-1][j-1].r*rgb[i-1][j-1].r)
                    +(rgb[i][j-2].r*rgb[i][j-2].r)+(rgb[i][j-1].r*rgb[i][j-1].r)
                    +(rgb[i][j].r*rgb[i][j].r)+(rgb[i+1][j-2].r*rgb[i+1][j-2].r)+(rgb[i+1][j-1].r*rgb[i+1][j-1].r)
                    +(rgb[i+2][j-2].r*rgb[i+2][j-2].r))/9;
                d2= ((rgb[i-2][j-2].r*rgb[i-2][j-2].r)+(rgb[i-2][j-1].r*rgb[i-2][j-1].r)+(rgb[i-2][j].r*rgb[i-2][j].r)
                    +(rgb[i-1][j-2].r*rgb[i-1][j-2].r)
                    +(rgb[i-1][j-1].r*rgb[i-1][j-1].r)+(rgb[i-1][j].r*rgb[i-1][j].r)
                    +(rgb[i][j-2].r*rgb[i][j-2].r)+(rgb[i][j-1].r*rgb[i][j-1].r)
                    +(rgb[i][j].r*rgb[i][j].r))/9;
                d3= ((rgb[i-1][j-1].r*rgb[i-1][j-1].r)+(rgb[i-1][j].r*rgb[i-1][j].r)+(rgb[i-1][j+1].r*rgb[i-1][j+1].r)
```

```

+(rgb[i][j-1].r*rgb[i][j-1].r)
+(rgb[i][j].r*rgb[i][j].r)+(rgb[i][j+1].r*rgb[i][j+1].r)
+(rgb[i+1][j-1].r*rgb[i+1][j-1].r)+(rgb[i+1][j].r*rgb[i+1][j].r)
+(rgb[i+1][j+1].r*rgb[i+1][j+1].r))/9;

if ((d1>d2)&&(d1>d3))
{red=((rgb[i-2][j-2].b)+
(rgb[i-1][j-2].b)
+(rgb[i-1][j-1].b)
+(rgb[i][j-2].b)+(rgb[i][j-1].b)
+(rgb[i][j].b)+(rgb[i+1][j-2].b)+(rgb[i+1][j-1].b)
+(rgb[i+2][j-2].b))/9;}

if ((d2>d1)&&(d2>d3))
{ red=((rgb[i-2][j-2].r)+(rgb[i-2][j-1].r)+(rgb[i-2][j].r)
+(rgb[i-1][j-2].r)
+(rgb[i-1][j-1].r)+(rgb[i-1][j].r)
+(rgb[i][j-2].r)+(rgb[i][j-1].r)
+(rgb[i][j].r))/9; }

if ((d3>d1)&&(d3>d2))
{ red=((rgb[i-1][j-1].r)+(rgb[i-1][j].r)+(rgb[i-1][j+1].r)
+(rgb[i][j-1].r)
+(rgb[i][j].r)+(rgb[i][j+1].r)
+(rgb[i+1][j-1].r)+(rgb[i+1][j].r)
+(rgb[i+1][j+1].r))/9; }

s1=((rgb[i-2][j-2].g*rgb[i-2][j-2].g)+
+(rgb[i-1][j-2].g*rgb[i-1][j-2].g)
+(rgb[i-1][j-1].g*rgb[i-1][j-1].g)
+(rgb[i][j-2].g*rgb[i][j-2].g)+(rgb[i][j-1].g*rgb[i][j-1].g)
+(rgb[i][j].g*rgb[i][j].g)+(rgb[i+1][j-2].g*rgb[i+1][j-2].g)+(rgb[i+1][j-1].g*rgb[i+1][j-1].g)

```

1].g)

$+(rgb[i+2][j-2].g*rgb[i+2][j-2].g)/9;$

$s2=((rgb[i-2][j-2].g*rgb[i-2][j-2].g)+(rgb[i-2][j-1].g*rgb[i-2][j-1].g)+(rgb[i-2][j].g*rgb[i-2][j].g)$
 $+(rgb[i-1][j-2].g*rgb[i-1][j-2].g)$
 $+(rgb[i-1][j-1].g*rgb[i-1][j-1].g)+(rgb[i-1][j].g*rgb[i-1][j].g)$
 $+(rgb[i][j-2].g*rgb[i][j-2].g)+(rgb[i][j-1].g*rgb[i][j-1].g)$
 $+(rgb[i][j].g*rgb[i][j].g))/9;$

$s3=((rgb[i-1][j-1].g*rgb[i-1][j-1].g)+(rgb[i-1][j].g*rgb[i-1][j].g)+(rgb[i-1][j+1].g*rgb[i-1][j+1].g)$
 $+(rgb[i][j-1].g*rgb[i][j-1].g)$
 $+(rgb[i][j].g*rgb[i][j].g)+(rgb[i][j+1].g*rgb[i][j+1].g)$
 $+(rgb[i+1][j-1].g*rgb[i+1][j-1].g)+(rgb[i+1][j].g*rgb[i+1][j].g)$
 $+(rgb[i+1][j+1].g*rgb[i+1][j+1].g))/9;$

if ((s1>s2)&&(s1>s3))

{ green= ((rgb[i-2][j-2].g)
+(rgb[i-1][j-2].g)
+(rgb[i-1][j-1].g)
+(rgb[i][j-2].g)+(rgb[i][j-1].g)
+(rgb[i][j].g)+(rgb[i+1][j-2].g)+(rgb[i+1][j-1].g)
+(rgb[i+2][j-2].g))/9;}

if ((s2>s1)&&(s2>s3))

{ green=((rgb[i-2][j-2].g)+(rgb[i-2][j-1].g)+(rgb[i-2][j].g)
+(rgb[i-1][j-2].g)
+(rgb[i-1][j-1].g)+(rgb[i-1][j].g)
+(rgb[i][j-2].g)+(rgb[i][j-1].g)
+(rgb[i][j].g))/9; }

if ((s3>s1)&&(s3>s2))

{ green=((rgb[i-1][j-1].g)+(rgb[i-1][j].g)+(rgb[i-1][j+1].g)

```

+ (rgb[i][j-1].g)
+ (rgb[i][j].g) + (rgb[i][j+1].g)
+ (rgb[i+1][j-1].g) + (rgb[i+1][j].g)
+ (rgb[i+1][j+1].g) / 9; }

e1 = ((rgb[i-2][j-2].b * rgb[i-2][j-2].b) +
+ (rgb[i-1][j-2].b * rgb[i-1][j-2].b)
+ (rgb[i-1][j-1].b * rgb[i-1][j-1].b)
+ (rgb[i][j-2].b * rgb[i][j-2].b) + (rgb[i][j-1].b * rgb[i][j-1].b)
+ (rgb[i][j].b * rgb[i][j].b) + (rgb[i+1][j-2].b * rgb[i+1][j-2].b) + (rgb[i+1][j-1].b * rgb[i+1][j-1].b)
+ (rgb[i+2][j-2].b * rgb[i+2][j-2].b)) / 9;

e2 = ((rgb[i-2][j-2].b * rgb[i-2][j-2].b) + (rgb[i-2][j-1].b * rgb[i-2][j-1].b) + (rgb[i-2][j].b * rgb[i-2][j].b)
+ (rgb[i-1][j-2].b * rgb[i-1][j-2].b)
+ (rgb[i-1][j-1].b * rgb[i-1][j-1].b) + (rgb[i-1][j].b * rgb[i-1][j].b)
+ (rgb[i][j-2].b * rgb[i][j-2].b) + (rgb[i][j-1].b * rgb[i][j-1].b)
+ (rgb[i][j].b * rgb[i][j].b)) / 9;

e3 = ((rgb[i-1][j-1].b * rgb[i-1][j-1].b) + (rgb[i-1][j].b * rgb[i-1][j].b) + (rgb[i-1][j+1].b * rgb[i-1][j+1].b)
+ (rgb[i][j-1].b * rgb[i][j-1].b)
+ (rgb[i][j].b * rgb[i][j].b) + (rgb[i][j+1].b * rgb[i][j+1].b)
+ (rgb[i+1][j-1].b * rgb[i+1][j-1].b) + (rgb[i+1][j].b * rgb[i+1][j].b)
+ (rgb[i+1][j+1].b * rgb[i+1][j+1].b)) / 9;

if ((e1 > e2) && (e1 > e3))

{ blue = ((rgb[i-2][j-2].b) +
+ (rgb[i-1][j-2].b)
+ (rgb[i-1][j-1].b)

```

```

+(rgb[i][j-2].b)+(rgb[i][j-1].b)
+(rgb[i][j].b)+(rgb[i+1][j-2].b)+(rgb[i+1][j-1].b)
+(rgb[i+2][j-2].b))/9; }
if ((e1>e2)&&(e2>e3))

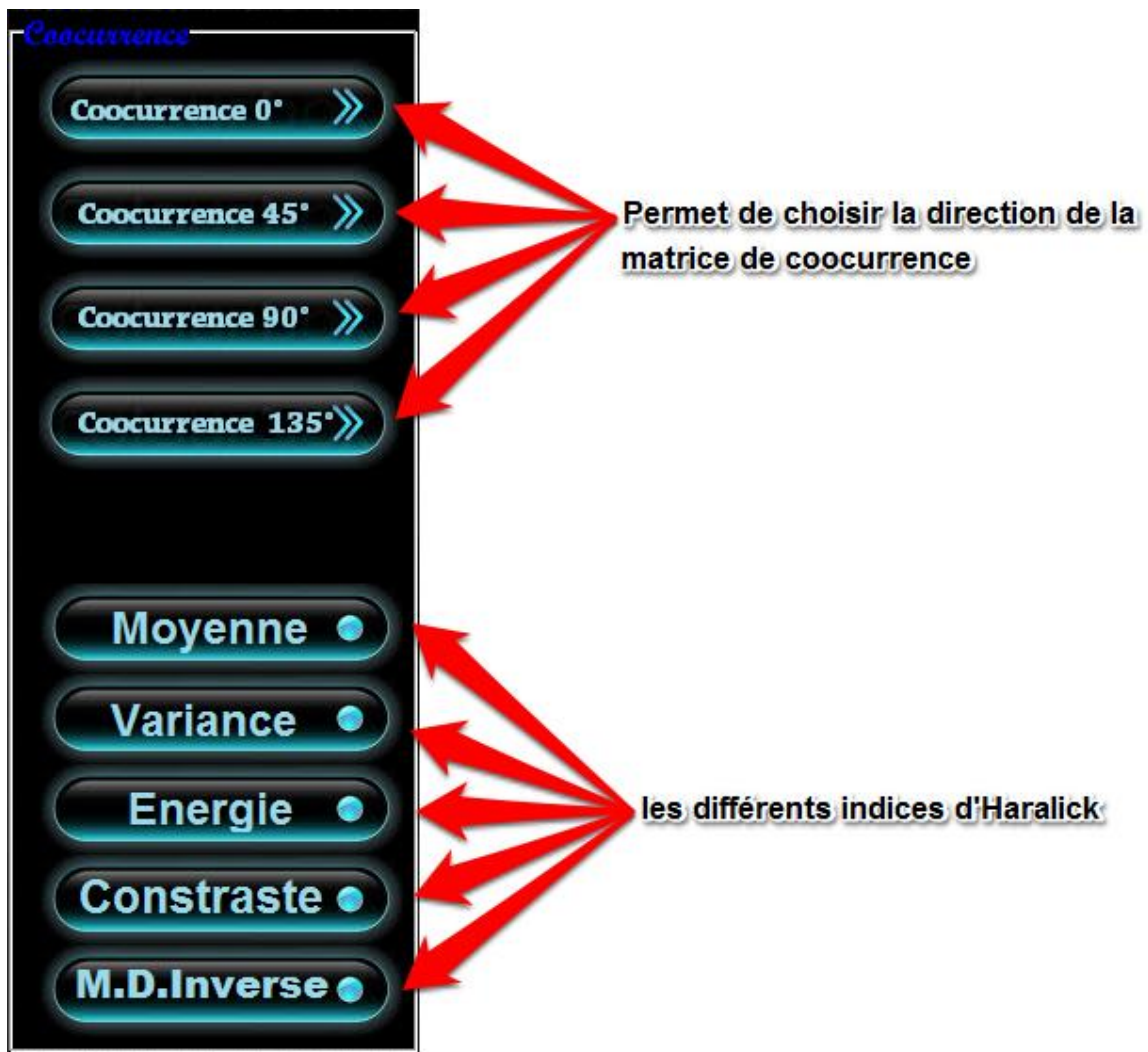
{ blue=((rgb[i-2][j-2].b)+(rgb[i-2][j-1].b)+(rgb[i-2][j].b)
+(rgb[i-1][j-2].b)
+(rgb[i-1][j-1].b)+(rgb[i-1][j].b)
+(rgb[i][j-2].b)+(rgb[i][j-1].b)
+(rgb[i][j].b))/9; }
if ((e3>e1)&&(e3>e2))
{ blue=((rgb[i-1][j-1].b)+(rgb[i-1][j].b)+(rgb[i-1][j+1].b)
+(rgb[i][j-1].b)
+(rgb[i][j].b)+(rgb[i][j+1].b)
+(rgb[i+1][j-1].b)+(rgb[i+1][j].b)
+(rgb[i+1][j+1].b))/9; }

d1=0;
d2=0;
d3=0;
s1=0;
s2=0;
s3=0;
e1=0;
e2=0;
e3=0;

}

```


III.3.7 Matrice de cooccurrence



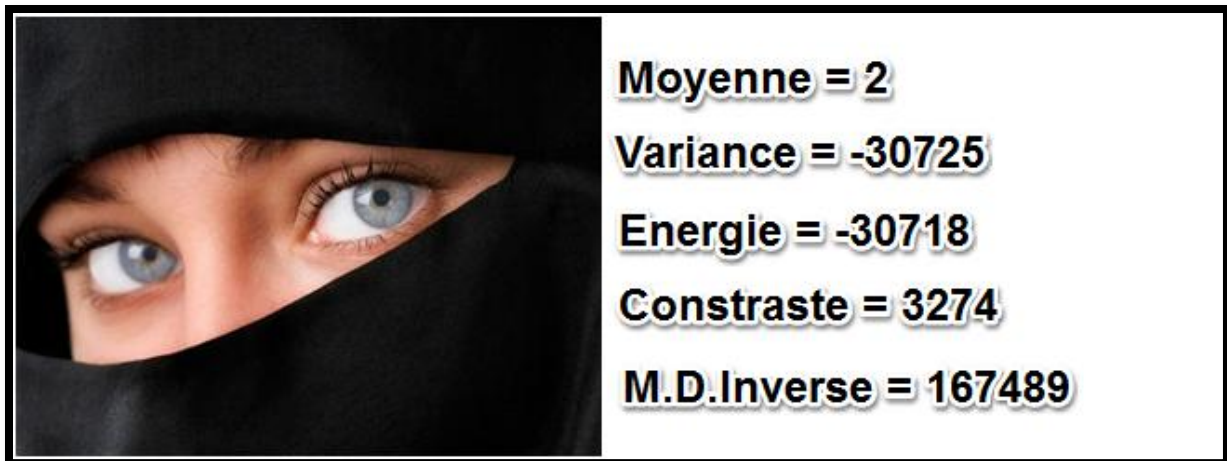


Figure III.6 : Calcule les indices d’Haralick avec un niveau de gris de 256 et une direction 0°

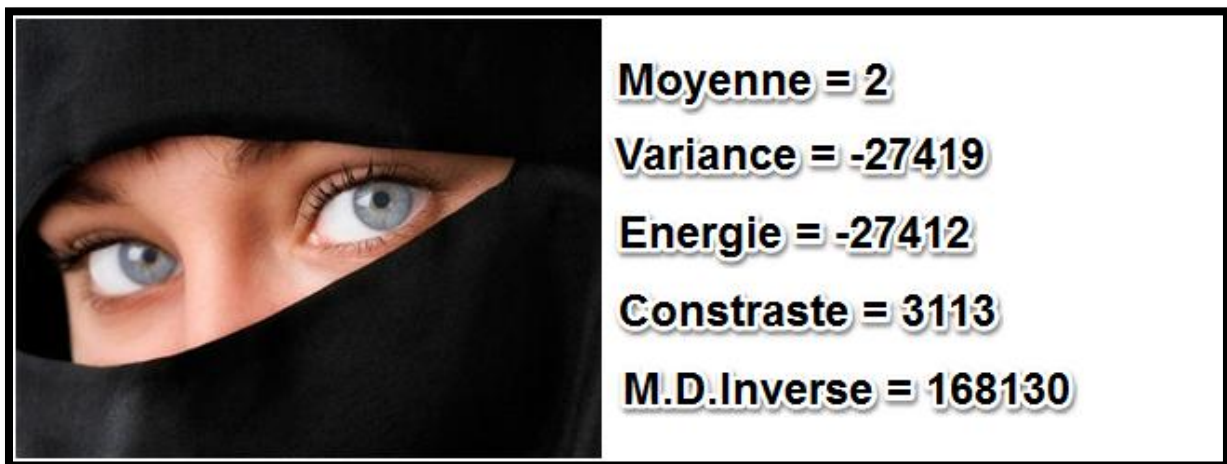


Figure III.7 : Calcule les indices d’Haralick avec un niveau de gris 256 et une direction de 45

III.3.4 Conclusion

Dans ce chapitre nous avons décrit les différentes composantes de notre application qui concerne le filtrage des images et l’extraction des descripteurs statistiques du premier ordre, Nous avons à travers ce travail pu s’initier au domaine de traitement d’images et nous avons élargi nos connaissances aussi en programmation.

Conclusion

générale

Conclusion générale

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo.

L'analyse de la texture des images est un domaine de recherche qui, aujourd'hui, donne lieu à de nombreuses études. Analyser la texture, revient à exploiter les relations de voisinage des pixels. En ce sens, elle cherche à rehausser les structures significatives contenues dans l'image.

Différentes approches ont été proposées dans la littérature. On trouve les matrices de cooccurrences, la morphologie mathématique, le variogramme, les ondelettes, etc ...

Notre but de notre mémoire est de pouvoir extraire les différents descripteurs statistiques du premier ordre de la matrice de cooccurrence, nous avons aussi appliqué certains filtres sur les images à savoir le filtre Moyen, Variance, Kurtosis et Nagao.

Nous avons remarqué que la variance mesure l'ampleur des variations d'intensité par rapport à la moyenne, une faible entropie pour une image d'intensité homogène et le filtre Nagao qui permet d'éliminer le bruit tout en préservant les contours.

Comme perspective on peut élargir notre travail sur les matrices de cooccurrence du second ordre, travailler sur d'autres méthodes d'analyse de la texture.

Références Bibliographiques

- [1] Christophe Coroyer. "Apports des Corrélations d'Ordre Elevé à l'Analyse de Textures Non Gaussiennes". Thèse de Doctorat, Université Cergy-Pontoise, France, 1996.
- [2] Adrien Janvier. « Débruitage de données tomographiques RX 3D » Travaux d'études et de recherches. 2010
- [3] www.nicolasleclerc.com/wp-content/uploads/2013/03/formats-image13.pdf
- [4] http://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner4.html
- [5] maths.cnam.fr/IMG/pdf/poly_R_deuxiemepartie.pdf
- [6] http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.OutilsProcessingImageHistogrammeCumul
- [7] http://xphilipp.developpez.com/articles/filtres/?page=page_3
- [8] http://www.tsi.telecom-paristech.fr/pages/enseignement/ressources/beti/textures_Fourier/texture.html
- [9] www.ummt0.dz/IMG/pdf/GADA_Saliha.pdf
- [10] www.ummt0.dz/IMG/pdf/AMROUN_Fazia.pdf
- [11] www.modulad.fr/archives/numero-35/Ladiray-35/Ladiray-35.pdf
- [12] <http://documents.irevues.inist.fr/bitstream/handle/2042/1744/002.PDF%20TEXTE.pdf>
- [13] http://www2.ulg.ac.be/telecom/teaching/notes/totali/elen016/node113_mn.html

Résumé :

L'analyse de la texture des images est un domaine de recherche qui, aujourd'hui, donne lieu à de nombreuses études. Analyser la texture, revient à exploiter les relations de voisinage des pixels. En ce sens, elle cherche à rehausser les structures significatives contenues dans l'image. Dans ce document, on s'est intéressé à l'une des approches d'analyse de texture à savoir la matrice de cooccurrence et au filtrage des images.

Ces approches sont des outils qui peuvent permettre d'extraire à partir des images des composants utilisées pour leur présentation et leur description et faire des pré ou post-traitement, ou dans un processus classificatoire.

Abstract:

The analysis of texture images is a research field which today gives rise to numerous studies. Analyze the texture, is to exploit the relations of neighborhood pixels. In this sense, it seeks to enhance significant structures contained in the image. In this paper, we became interested in one of the texture analysis namely the co-occurrence matrix and image filtering approaches.

These approaches are tools that can help to extract from the images of the components used for their presentation and description and make pre-or post-treatment, or in a classificatory process.

ملخص:

تحليل بنية الصور هو حقل البحوث الذي أصبح اليوم يثير العديد من الدراسات. تحليل ا لبنية هو استغلال علاقات جوانب بكسل. في هذا المعنى فهو يسعى إلى تعزيز هياكل هامة الواردة في الصورة. في هذا البحث ، أصبحنا مهتمين بواحدة من تحليل الملمس المعروفة بمصفوفة التساوق وأساليب تصفية الصور.

هذه الأساليب هي الأدوات التي يمكن أن تساعد على الاستخراج من الصور بعض المكونات المستخدمة في العرض والوصف وما قبل أو بعد المعالجة، أو في العملية التصنيفية.