

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

De Master Académique en Automatique,
Option : Commande des Systèmes

Thème

*Approche LBP pour la caractérisation
des textures dans les images*

Proposé et dirigé par :

Mr. MESSAR Youcef

Mme. OUSLIMANI Farida

Présenté par :

Mr. ATMAN Youcef

Soutenu le : / /2013

Promotion 2013

Remerciements

Au terme de ce mémoire, je tiens à remercier tout naturellement en premier lieu Dieu le tout puissant qui m'a donné la force, le courage et la patience de bien mener ce travail.

J'exprime mes remerciement et ma gratitude à mon promoteur Monsieur Y.MESSAR, pour l'aide et le soutien et les conseils qu'il m'apportés tout au long de ce travail.

Je tiens également à remercier Madame F.OUSLIMANI de m'avoir suivi et guider dans ce projet.

Je tiens aussi à exprimer mes plus vifs remerciements à tous ceux qui ont contribué de près ou de loin, d'une manière ou d'une autre à la réalisation de ce modeste travail.

Tout mes respects et mes remerciements vont vers les membres du jury qui vont pleinement consacrer leur temps et leur attention afin d'évaluer ce travail, que j'espère le sera à la hauteur de leur attente.

Enfin, mes remerciements les plus sincères sont adressés à tous les professeurs, l'administration et le personnel du département Automatique pour leurs contributions dans la réussite de nos études.

Dédicaces

*Je dédie ce modeste travail à
mes très chers parents, pour leurs soutient financier
et moral
en leurs espérant une longue vie
et que Dieu les garde.*

*Je dédie aussi ce travail à mes sœurs et frères
et à tous mes amis.*

TABLE DES FIGURES

Table des figures :

Figure I.1 : <i>Illustration des trois types de textures</i>	4
Figure I.2 : <i>Illustration de la variance</i>	6
Figure I.3 : <i>Illustration du moment d'ordre 3</i>	7
Figure I.4 : <i>Illustration du moment d'ordre 4</i>	7
Figure I.5 : <i>Les directions considérées dans le calcul des matrices de cooccurrence</i>	8
Figure I.6 : <i>Exemple de construction d'une matrice de cooccurrence</i>	9
Figure II.1 : <i>Illustration de calcul d'un LBP de base</i>	13
Figure II.2 : <i>Exemple d'une image d'entrée, l'image et l'histogramme LBP correspondants</i>	14
Figure II.3 : <i>Exemple d'un histogramme LBP d'une image faciale</i>	15
Figure II.4 : <i>Le calcul du code de LBP et une mesure de contraste</i>	16
Figure II.5 : <i>Robustesse du LBP aux variations d'illumination</i>	16
Figure II.6 : <i>Exemples de voisinages avec différentes valeurs de (P, R)</i>	17
Figure II.7 : <i>Calcul d'un $LBP_{P,R}$ (P=8, R=1)</i>	18
Figure II.8 : <i>Rotation d'une image</i>	20
Figure II.9 : <i>Liste des 36 formes binaires locales invariantes en rotation de $LBP_{8,R}^{ri}$</i>	21
Figure II.10 : <i>Exemple des rotations invariantes des codes LBP</i>	21
Figure II.11 : <i>Exemples de motifs uniformes et non-uniformes</i>	22
Figure II.12 : <i>Textures particulières détectées par $LBP^{\mu 2}$</i>	23
Figure II.13 : <i>Les 58 différents modèles uniformes du quartier (8, R)</i>	24
Figure II.14 : <i>Illustration de l'opérateur LTP de base</i>	25
Figure II.15 : <i>Division d'un code LTP en codes LBP positif et négatif</i>	26
Figure II.16 : <i>Illustration du CS-LBP</i>	27
Figure III.1 : <i>Exemple de segmentation par seuillage</i>	29
Figure III.2 : <i>Exemple d'histogramme d'une image</i>	29
Figure III.3 : <i>Exemple de segmentation par approche contour</i>	30
Figure III.4 : <i>Le processus d'un algorithme de croissance par région</i>	31
Figure III.5 : <i>Principe de la classification</i>	32
Figure III.6 : <i>Classifieur par Maximum de Vraisemblance dans le cas de deux classes</i>	35

TABLE DES FIGURES

Figure III.7 : <i>Exemple de classification avec la méthode de k-means</i>	37
Figure IV.1 : <i>Environnement de développement Intégré (IDE) de Code ::Blocks</i>	40
Figure IV.2 : <i>(a):image enfant,(b) et (c) : image et histogramme LBP correspondants</i>	41
Figure IV.3 : <i>Les images enfant (à gauche), images LBP correspondantes (à droite)</i>	42
Figure IV.3 : <i>Les images enfant (à gauche), images LBP correspondantes (à droite)</i>	43
Figure IV.4 : <i>Calcul d'un $LBP_{P,R}$ ($P=8, R=1$)</i>	44
Figure IV.5 : <i>Les images LBP obtenues pour chaque rotation du voisinage</i>	45
Figure IV.6 : <i>résultat de l'invariance par rotation</i>	46
Figure IV.7 : <i>(a) : image enfant, (b) : image LTP correspondante</i>	46
Figure IV.8 : <i>(a) : image enfant avec bruit, (b) : image LBP correspondante, (c) : image LTP correspondante</i>	47

Table des matières

Introduction générale	1
------------------------------------	----------

Chapitre I : la texture

I.1. Introduction.....	3
I.2. Définition de la texture.....	3
I.3. Les méthodes de caractérisation de la texture.....	4
I.3.1. Les méthodes structurelles.....	4
I.3.2. Les méthodes fréquentielles.....	5
I.3.3. Les méthodes basées sur modèles.....	5
I.3.4. Les méthodes statistiques.....	5
I.3.4.1. Attributs statistiques du premier ordre.....	5
I.3.4.2. Attributs statistiques d'ordre 2.....	8
I.4. Approche proposée.....	10
I.5. Conclusion.....	11

Chapitre II: Local Binary Pattern (LBP)

II.1. Introduction.....	12
II.2. Définition du LBP.....	12
II.3. Domaines d'application du LBP.....	12
II.4. Le LBP de base.....	13
II.5. Le $LBP_{P,R}$	17
II.6. La dérivation.....	18
II.7. Le LBP invariant par rotation.....	20
II.8. Le LBP uniforme.....	22
II.9. Le motif uniforme invariant par rotation.....	24

II.10. Les extensions plus récentes du LBP.....	25
II.11. Conclusion.....	27
Chapitre III : la segmentation	
III.1. Introduction.....	28
III.2. Définition de la segmentation.....	28
III.3. Exemple de segmentation basée sur le seuillage d’histogramme.....	28
III.4. Les approches de segmentation.....	30
III.4.1. Approche contour.....	30
III.4.2. Approche région.....	30
III.4.3. Classification.....	31
III.5. Les approches de classification.....	32
III.5.1. Approche de classification supervisée.....	33
III.5.1. Approche de classification non supervisée	33
III.6. Etude de quelques méthodes de classification.....	33
III.6.1. La méthode de maximum de vraisemblance.....	33
III.6.2. La méthode des k-means.....	35
III.7. Conclusion.....	38
Chapitre IV : Implémentation et expérimentation	
IV.1. Introduction.....	39
IV.2. Environnement matériel et logiciel de programmation.....	39
IV.3. Implémentations et discussions.....	41
IV.4. Conclusion.....	47
Conclusion générale.....	48

Une image numérique est composée d'unités élémentaires appelées pixels, contenant chacune différentes informations (intensité lumineuse, couleur, etc.). L'arrivée de ces images constitue un véritable atout dans beaucoup de domaines scientifiques. En effet, les images numériques apportent un support de représentation permettant de modéliser et de véhiculer beaucoup d'informations, les rendant ainsi accessibles à l'interprétation et ce, pour une utilisation optimale dans le domaine socio-économique.

Vers les années 70-80, la thématique du traitement d'images est devenue indispensable, afin d'apporter aux experts des outils informatiques et mathématiques pouvant leur faciliter l'étude des images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information pertinente.

En traitement d'images, l'étape d'extraction des paramètres (traitement de bas niveau) précède souvent une étape de décision (traitement de haut niveau), de manière à pouvoir répondre à des questions telles que: matériau normal ou défectueux ? Tissu biologique sain ou pathologique ? Types de défauts ? etc.. Les paramètres calculés permettant donc de décrire, de caractériser, de segmenter et d'analyser les images en question. La segmentation d'image est un traitement de bas niveau qui consiste à créer une partition de l'image en régions. Il n'y a pas de méthode unique de segmentation d'une image, le choix d'une méthode est lié à la nature de l'image et aux primitives à extraire.

Afin d'exposer plus en détail les différentes méthodes de segmentation, on doit présenter la notion de « la texture », car elle est très souvent associée aux attributs permettant de caractériser l'homogénéité d'un ensemble connexe de pixels. Donc, on doit aussi présenter les principales méthodes d'analyse permettant l'extraction des caractéristiques texturales des régions d'une image.

Au cours des dernières années, nous avons assisté à un développement d'un opérateur qui caractérise la texture dans l'image. Cet opérateur est simple, mais très efficace en terme de mesure, appelé motif binaire local (LBP). Il a été appliqué avec succès dans nombreuses applications, telles que la classification et la segmentation des textures. Dans de nombreuses études empiriques, le LBP et ses extensions montrent des excellents résultats en termes de précision et de complexité des calculs.

Dans ce mémoire de fin d'études, notre objectif est de faire une étude sur les motifs binaires locaux (LBP) dans des images en niveaux de gris, afin de caractériser et de décrire les textures.

Le présent mémoire est structuré en quatre chapitres.

Le premier chapitre, introductif, présentera la notion et les types de texture, ainsi que les méthodes de caractérisation de celle-ci. L'accent sera mis sur les méthodes statistiques, dont nous introduirons l'approche à proposer.

Le deuxième chapitre sera spécialement consacré pour la présentation du descripteur LBP, en précisant la manière de calcul de cet opérateur dans les images en niveaux de gris, et comment il joue le rôle dans l'extraction des textures. Nous allons voir aussi quelques extensions du LBP les plus récentes.

Dans le troisième chapitre, nous présenterons les principales approches de la segmentation des images, en se basant essentiellement sur l'approche classification.

Le dernier chapitre est réservé aux tests et à la présentation des résultats obtenus sur quelques images par les approches proposées.

Enfin, nous terminerons notre travail par une conclusion générale, clôturant ce mémoire et quelques perspectives.

I.1. Introduction

La texture constitue une caractéristique importante dans le domaine de l'analyse d'images, son étude est un thème qui a suscité un grand intérêt ces dernières années. Plusieurs techniques et méthodes permettant la caractérisation de ces caractéristiques ont été élaborées.

Dans ce chapitre, nous allons présenter quelques méthodes d'analyse de texture ainsi que sa définition.

I.2. Définition de la texture

Définir de manière unique la notion de texture est une tâche difficile, car il existe de nombreuses définitions dans la littérature. On peut s'en approcher en disant qu'une texture est une zone de l'image qui présente certaines caractéristiques d'homogénéité, qui la font apparaître comme une zone unique. Nous pouvons aussi la décrire comme étant un ensemble de primitives (motifs) de tailles et de formes variables, présentant une organisation spatiale particulière [1].

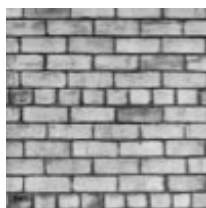
En pratique, nous distinguons deux grandes classes de textures : les macro-textures et les micro-textures.

Les macro-textures : Pour ce type, il est facile d'extraire visuellement les primitives, ainsi que les lois régissant leurs positions dans l'image. Dans ce cas, la texture est définie comme une répétition régulière et périodique de ces primitives (ex : peau de lézard, mur de brique). On parle alors de textures périodiques ou structurelles ou aussi déterministes [2].

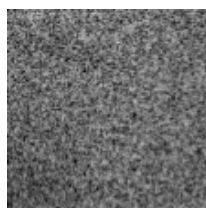
Les micro-textures : Présentent des éléments "microscopiques" distribuées d'une manière aléatoire et anarchique (ex: sable, laine tissée, herbe). La texture est alors définie à l'aide de son plus fin détail, le pixel qui a en général un caractère aléatoire. On parle alors de textures aléatoires ou statistiques ou encore stochastiques [3].

On peut distinguer aussi des textures composées de ces deux types, appelées textures quasi-déterministes, qui sont caractérisées sans qu'il y ait vraiment de répartition régulière ou complètement aléatoire d'un motif de base.

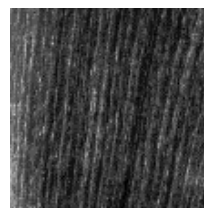
La figure suivante illustre ces trois types de textures :



Texture déterministe



Texture aléatoire



Texture quasi-déterministe

Figure I.1. : Illustration des trois types de textures.

I.3. Les méthodes de caractérisation de la texture

L'étude de la texture a fait l'objet de nombreux travaux de recherche qui ont engendré une multitude de méthodes d'analyse [4]. Ces méthodes ont pour objet de caractériser, extraire, décrire, discriminer,..., segmenter les textures. Nous nous attacherons plus loin à la présentation de certaines méthodes de segmentation.

Dans un premier temps, nous allons aborder différentes méthodes permettant l'extraction d'attributs texturaux dans les images à niveau de gris. L'accent sera mis sur les méthodes statistiques, pour enfin définir notre méthode proposée, qui est le motif binaire local (LBP) puisqu'il représente l'objectif de ce travail.

Vu la difficulté d'une définition précise de la texture, plusieurs approches ont été élaborées ces dernières années pour identifier cette opération. Nous donnerons ci-après les méthodes utilisées pour l'analyse de la texture :

- Méthodes structurelles.
- Méthodes fréquentielles.
- Méthodes basées sur un modèle.
- Méthodes statistiques.

I.3.1. Les méthodes structurelles

Elles considèrent que les textures sont formées de motifs élémentaires (primitives), qui se répètent suivant une organisation assez régulière. L'analyse d'une texture à l'aide d'une méthode structurelle est généralement composée de deux étapes : d'abord identifier les primitives, puis définir les règles de placement de celles-ci.

Cette approche reste limitée car elle n'est réellement efficace que pour le traitement de textures très régulières (textures macroscopiques) [5]. C'est pourquoi cette famille de méthodes de caractérisation de textures ne sera pas développée.

I.3.2. Les méthodes fréquentielles

Elles considèrent que la texture est une répétition d'éléments avec une certaine fréquence. Ces méthodes font généralement appel à des outils de traitement du signal et consistent à analyser la texture par la caractérisation des différentes fréquences qui la composent. Ainsi, de nombreuses méthodes initialement développées pour l'analyse fréquentielle des signaux. Parmi ces méthodes on trouve les filtres de Gabor et les ondelettes qui ont montré de bonnes performances en analyse et la description de la texture [6].

I.3.3. Les méthodes basées sur modèles

Pour ces approches, les descripteurs sont les paramètres d'un modèle, permettant de décrire ou de générer une texture. Plusieurs modèles peuvent être utilisés. Les plus connus sont les modèles Markoviens [7], les modèles autorégressifs et les modèles fractals [6].

I.3.4. Les méthodes statistiques

Elles étudient les relations entre un pixel et ses voisins (vision microscopique). Dans ce cas, la texture est décrite par les statistiques de la distribution des niveaux de gris présents dans l'image. Les attributs extraits de ces méthodes permettent de caractériser tout type de texture, même les textures fines et sans régularité apparente.

Ces attributs peuvent être divisés en plusieurs catégories selon leur ordre. Ce dernier dépend du nombre de pixels mis en jeu dans le calcul des paramètres.

I.3.4.1. Attributs statistiques du premier ordre [2] [8]

Ces attributs donnent des informations générales caractérisant une image en fonction de la variation de l'intensité des pixels individuels qui la composent.

Les paramètres tels que la moyenne, la variance,...etc. sont calculés à partir de l'histogramme des intensités.

Les attributs les plus courants sont :

- **La moyenne**

La valeur moyenne m_1 des niveaux de gris de tous les pixels de la région considérée est donnée par :

$$m1 = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N I(x, y) \quad (I.1)$$

Avec $I(x, y)$ représente la valeur du niveau de gris du pixel (x, y) , M est la taille de l'image en x et N sa taille en y .

Une image claire possède une moyenne plus élevée qu'une image foncée.

- **La variance**

Elle correspond au moment d'ordre 2. Elle mesure la répartition des niveaux de gris (luminosité) autour de la valeur moyenne. Sa valeur est donnée par :

$$Var = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N (I(x, y) - m1)^2 \quad (I.2)$$

Exemple :

- Deux images de même moyenne mais de variance différente.

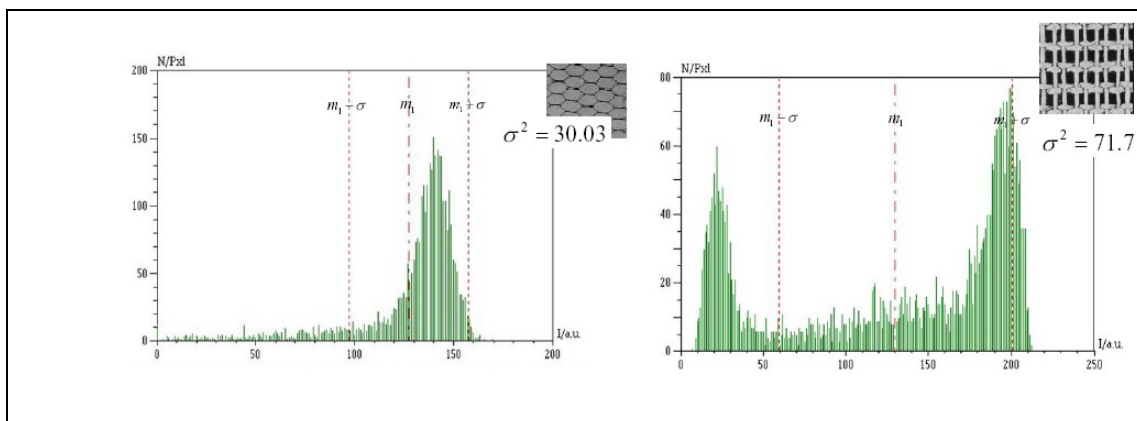


Figure I.2. : Illustration de la variance.

- **Le moment d'ordre 3**

Le moment centré d'ordre 3 (skewness en Anglais) mesure l'asymétrie de la distribution des niveaux de gris par rapport à une distribution symétrique.

$$Skew = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N (I(x, y) - m1)^3 \quad (I.3)$$

Si l'histogramme de l'image est décalé vers la droite de la moyenne, le moment centré d'ordre 3 est négatif, et s'il est décalé vers la gauche de la moyenne, le moment centré d'ordre 3 est positif.

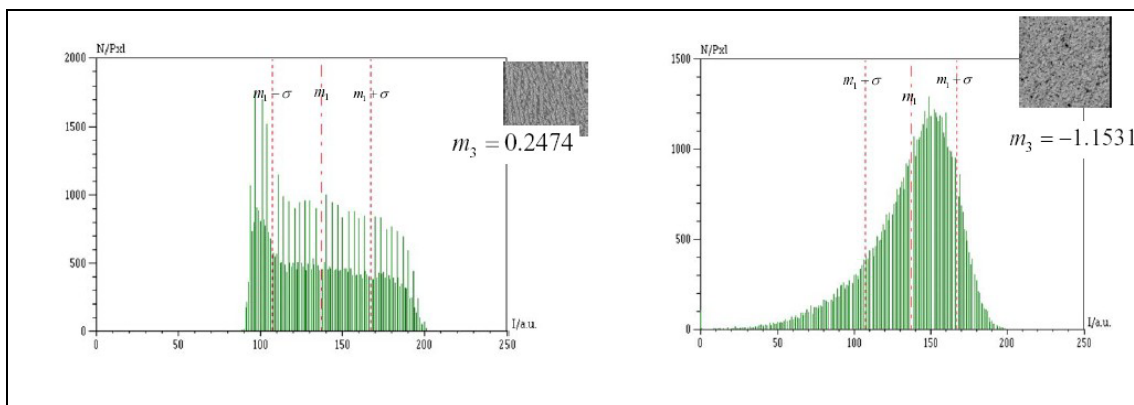


Figure I.3. : Illustration du moment d'ordre 3.

- **Le moment d'ordre 4**

Le moment centré d'ordre 4 (kurtosis en Anglais) caractérise la forme du sommet de l'histogramme.

$$Kurt = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N (I(x, y) - m_1)^4 \quad (I.4)$$

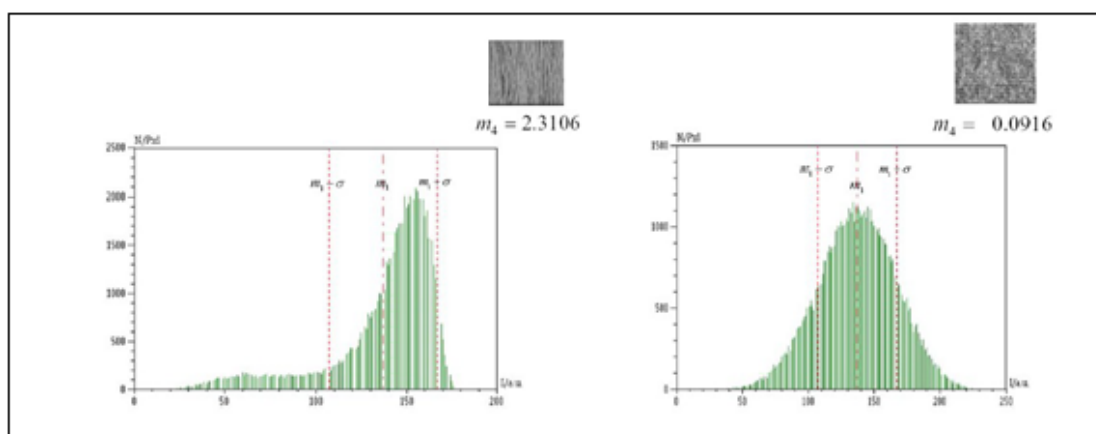


Figure I.4. : Illustration du moment d'ordre 4.

Plus le moment centré d'ordre 4 est faible, plus le sommet de l'histogramme est arrondi.

I.3.4.2. Attributs statistiques d'ordre 2

Ce type de méthodes prend en considération les interactions entre les niveaux de gris de deux pixels.

On retiendra, en particulier la méthode des matrices de cooccurrence (Haralick 1979) [4] connue sous le nom GLCM (Gray Level Co-occurrence Matrix).

Du fait de leur richesse en information de texture, les matrices de cooccurrences sont devenues les plus connues et les plus utilisées pour extraire les caractéristiques des textures. Elles servent de référence à toute nouvelle méthode de caractérisation de textures. [8]

La matrice de cooccurrence permet de déterminer la fréquence d'apparition d'un motif formé de deux pixels séparés par une certaine distance d selon une direction particulière θ . Afin de limiter le nombre de calculs, on prend généralement les valeurs 0, 45, 90 et 135 degrés pour la direction angulaire θ et la valeur 1 pour la distance d . La figure I.2 illustre les différentes directions considérées.

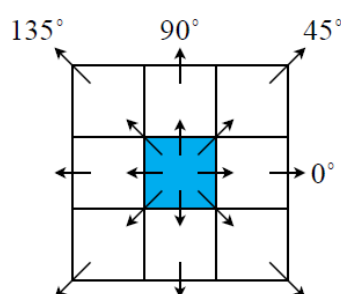
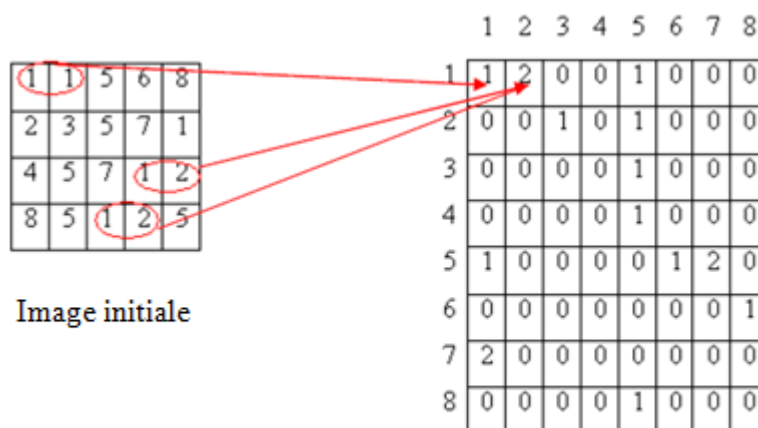


Figure I.5. : Les directions considérées dans le calcul des matrices de cooccurrence.

Le choix des vecteurs de déplacement est toujours très important pour la réussite de la méthode. En effet la difficulté de cette méthode réside dans le choix de ces vecteurs, car ils varient selon le type de textures.

A chaque direction θ et pour chaque valeur de d correspond une matrice de cooccurrence $\varphi(d, \theta)$. La figure suivante illustre un exemple de construction de cette matrice pour $\theta=0^\circ$ et $d=1$.



Matrice de cooccurrence ($d = 1, \theta = 0^\circ$)

Figure I.6. : Exemple de construction d’une matrice de cooccurrence.

Les matrices de cooccurrence contiennent une masse d’informations trop importante. Quatorze (14) descripteurs de textures qui peuvent être calculés à partir de ces matrices ont été proposés par Haralick et al [4].

Voici quelques exemples d’attributs donnés par Haralick :

- **La Variance** Ce paramètre mesure la distribution des niveaux de gris autour de la moyenne.

$$Var = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (i - Moy)^2 p(i, j) \quad (I.5)$$

$p(i, j)$: correspond aux éléments de la matrice de cooccurrence, c’est-à-dire à la probabilité de passer d’un pixel de niveaux de gris i à un pixel de niveau de gris j .

(i, j) : deux niveaux de gris.

$M \times N$: est la taille de l’image initiale.

- **Le Contraste** Il favorise les variations locales des niveaux de gris; ainsi il est élevé si on passe souvent d’un pixel très clair à un pixel très foncé ou inversement (fortes transitions).

$$Cont = \sum_{i=1}^M \sum_{j=1}^N (i - j)^2 p(i, j) \quad (I.6)$$

- **L'Entropie** Ce paramètre mesure le désordre dans l'image. L'entropie atteint de fortes valeurs lorsque la texture est complètement aléatoire (sans structure apparente).

$$Ent = - \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (\log p(i, j) p(i, j)) \quad (1.7)$$

I.4. Approche choisie

Notre approche est basée sur les caractéristiques Local Binary Pattern (LBP).

La méthode LBP permet de décrire les textures des images en utilisant des modèles locaux. Cette approche présente l'avantage d'être invariante aux changements d'orientation (translation de l'image) et robuste aux variations de luminosité.

Le principe de cette approche est de caractériser chaque pixel de l'image en fonction des valeurs du niveau de gris de ses voisins (approche statistique).

La distribution LBP permet d'extraire les primitives de la texture et les règles de leur placement (approche structurelle).

Donc, d'une certaine manière, la méthode LBP rassemble les approches statistiques et structurelles de l'analyse de texture. Pour cette raison, il est supposé que la distribution LBP peut être utilisée avec succès dans la reconnaissance d'une grande variété de types de textures.

Remarques

Cette approche peut être classée dans les méthodes statistiques du fait qu'elle s'appuie essentiellement sur la répartition spatiale des niveaux de gris dans la texture considérée.

Une comparaison de plusieurs méthodes statistiques est également réalisée dans [9] et l'approche par LBP apparaît comme la plus précise.

I.5. Conclusion

L'analyse des textures est un problème difficile. Cette difficulté est due essentiellement au fait qu'il n'existe pas de définition précise permettant de décrire exactement la notion de la texture.

Dans ce chapitre, nous avons vu quelques notions essentielles de la texture, puis les principales méthodes permettant sa caractérisation.

Les méthodes d'analyse de textures sont très nombreuses et très variées. Parmi toutes ces méthodes, nous nous sommes concentrés sur les méthodes statistiques (surtout les matrices de cooccurrence) où, on a classé l'approche LBP qui forme l'objet du chapitre suivant.

II.1. Introduction

Dans le chapitre précédent, nous avons présenté quelques méthodes permettant l'analyse des textures. Parmi toutes ces méthodes, nous avons introduit une nouvelle approche, qui est le motif binaire local (LBP). Dans ce chapitre, nous allons essayer de définir le LBP, comment calculer les codes LBP dans les images en niveaux de gris, ainsi nous présenterons les améliorations et les extensions de cet opérateur.

II.2. Définition du LBP

Le LBP(Local Binary Pattern) est défini comme une caractéristique qui résume la structure spatiale locale d'une image utilisée en vision par ordinateur, pour reconnaître des textures ou pour détecter des objets dans les images numériques.

Ce descripteur a été d'abord mentionné pour la première fois par Harwood en 1993 pour mesurer le contraste local d'une image, mais réellement popularisé en 1996 par Timo Ojala pour analyser les textures [9]. Il a montré d'excellentes performances dans de nombreuses études comparatives, tant en termes de vitesse et de discrimination.

Remarque

La méthode LBP est spécialement adaptée pour les images à forte variation d'intensité, car elle permet de bien détecter les différentes textures.

II.3. Domaines d'application du LBP

En raison de son pouvoir discriminant et sa simplicité de calcul, le motif binaire local est devenu une approche populaire dans plusieurs applications. En effet, le LBP et ses extensions ont été déjà appliqués, par exemple, pour l'analyse visuelle, la restauration de l'image, la détection du mouvement, la télédétection, l'analyse de l'image biomédicale, la biométrie, la modélisation de l'environnement et l'analyse de la scène extérieure...etc..

Il est également une technique de base pour l'authentification des visages et la reconnaissance faciale [10].

Notamment, l'utilisation du LBP dans les applications biomédicales et les systèmes de reconnaissance biométrique a connu une croissance rapide ces dernières années.

II.4. Le LBP de base

Le principe de calcul de cet opérateur est simple. Il considère une matrice de dimension 3×3 pixels, puis compare le niveau de gris du pixel central avec les niveaux de ses 8 voisins.

Ci-dessous les étapes d'exécution de l'algorithme d'extraction des LBP illustrées par la figure II.1. Notez que cette présentation a été adaptée par Bouchard (2009) [11]:

1. Les huit valeurs du voisinage du pixel à traiter sont extraites pour former une matrice de 3×3 pixels. Dans cet exemple, chaque pixel possède une valeur de niveau de gris différente. Le pixel à traiter possède la valeur radiométrique 4 ;
2. Un seuillage est effectué sur chaque valeur de gris des pixels du voisinage. Tout pixel possédant une valeur supérieure ou égale à la valeur du pixel central est attribué à la valeur 1. La valeur 0 est attribuée à toute valeur de gris inférieure à celle du pixel central ;
3. Une matrice multiplicatrice est mise en mémoire. Cette matrice est représentée par une répartition des poids donnés par des puissances de deux ;
4. Une multiplication élément par élément est effectuée entre la matrice résultante du seuillage de l'étape 2 et la matrice multiplicatrice de l'étape 3 ;
5. La sommation des valeurs de la matrice résultante de l'étape 4 est effectuée. Cette somme représente donc la valeur LBP du pixel central, qui sert à décrire les informations de la texture locale dans ce pixel ;
6. L'algorithme ré-exécute les étapes 1 à 5 jusqu'à ce que tous les pixels de l'image d'entrée soient traités.

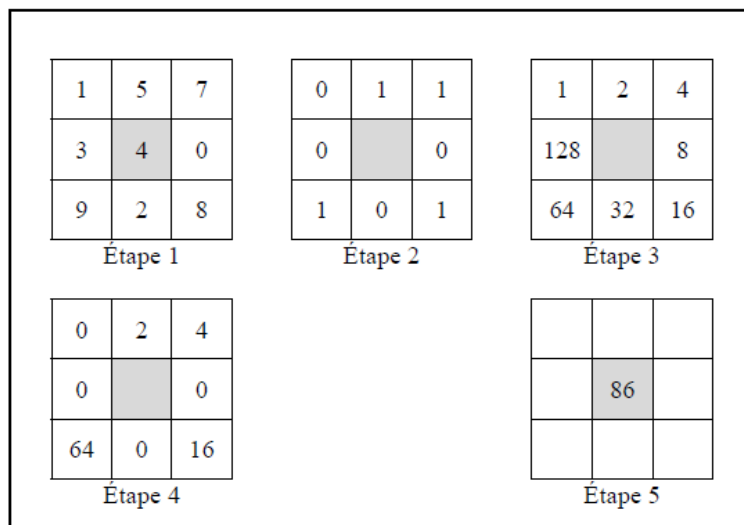


Figure II.1. : illustration de calcul d'un LBP de base.

On obtiendra donc, comme pour une image à niveaux de gris, une image contenant des pixels dont l'intensité se situe entre 0 et 255.

Remarque

On peut voir que le LBP est très rapide à calculer, et il est un bon candidat pour la description de la région locale de l'image. Lors du calcul de ce code, les bords qui n'ont pas suffisamment d'informations sont ignorés car elles pourraient produire des fausses informations.

- **La formule du LBP de base**

Le LBP de base est défini par cette formule :

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c) \times 2^p \quad (II.1)$$

Où : g_c est le niveau de gris du pixel central de coordonnées (x_c, y_c) .

g_p ($p = 0, 1, \dots, 7$) est le niveau de gris de chaque pixel environnant.

Si g_p est plus petit que g_c le résultat binaire du pixel prend la valeur 0, sinon (supérieure ou égale), il sera mis à 1.

Avec : $S(x)$ une fonction définie comme suit :

$$S(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

- **Histogramme LBP**

Souvent la distribution des codes LBP sur l'image est utilisée pour décrire la texture sous forme d'histogramme.

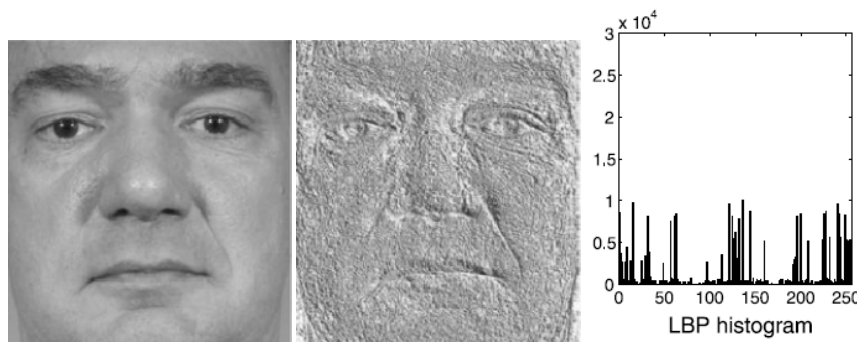


Figure II.2. : Exemple d'une image d'entrée, l'image et l'histogramme LBP correspondants.

Une fois le code LBP est calculé pour tous les pixels de l'image, on calcule l'histogramme de cette image LBP, pour former un vecteur de caractéristiques représentant l'image.

En réalité, afin d'incorporer plus d'informations spatiales au vecteur représentant l'image (par exemple le visage), on divise tout d'abord cette image codée par l'opérateur LBP en petites régions et l'histogramme sera construit pour chaque région. Finalement, on concatène tous les histogrammes des régions afin de former un grand histogramme représentant l'image des caractéristiques faciales (voir la figure ci-après). L'efficacité du code LBP comme indice facial s'explique par le fait que le LBP permet de caractériser les détails fins d'un visage.

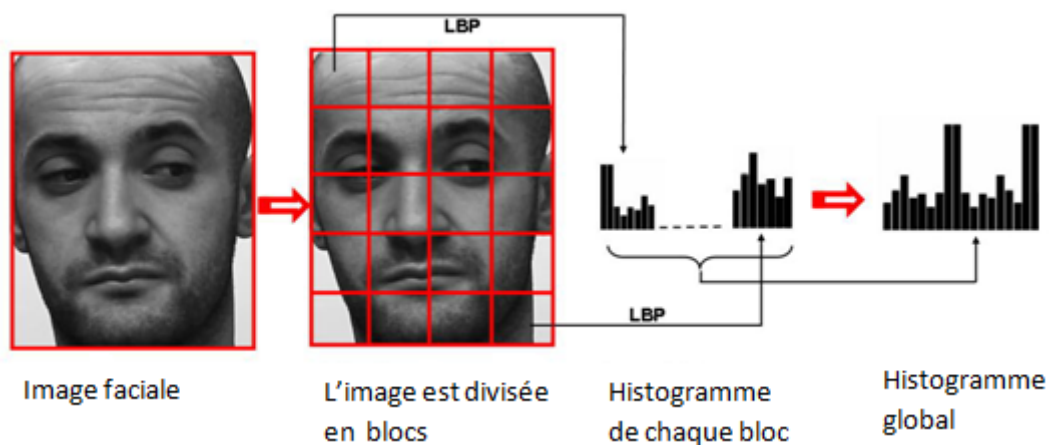


Figure II.3. : Exemple d'un histogramme LBP d'une image faciale.

L'histogramme LBP est connu sous cette forme :

$$h(i) = \sum_{x,y} B(LBP(x, y) = i) \quad (II.2)$$

Avec : $i \in [0, \dots, 2^p - 1]$ et $B(v) = \begin{cases} 1, & \text{lorsque } v \text{ est vraie} \\ 0, & \text{autrement} \end{cases}$

- **Le LBP et la Mesure du contraste**

Le LBP a été complété par une mesure du contraste local. La (Figure II.4) montre comment la mesure du contraste (C) a été dérivée. Elle correspond à la différence entre la moyenne des niveaux de gris supérieurs ou égaux à la valeur du pixel central avec celle des valeurs inférieurs.

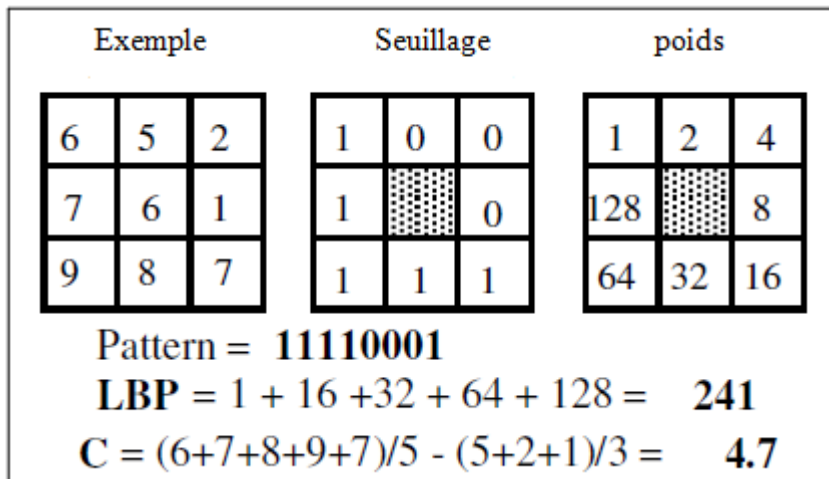


Figure II.4. : Le calcul du code de LBP et une mesure de contraste.

L'opérateur a été appelé **LBP / C**, et de très bons taux de discrimination étaient rapportés avec des textures choisies de l'album photographique Brodatz [12].

La combinaison du LBP avec la mesure de contraste augmente la robustesse aux changements d'intensité. En effet, la propriété la plus importante du code LBP est qu'il est invariant aux changements d'illumination et aux variations uniformes des niveaux de gris, parce que le LBP d'un pixel ne dépend que des différences entre son niveau de gris et celui de ses voisins. La figure suivante est un bon exemple montrant la robustesse du LBP aux changements de la lumière.

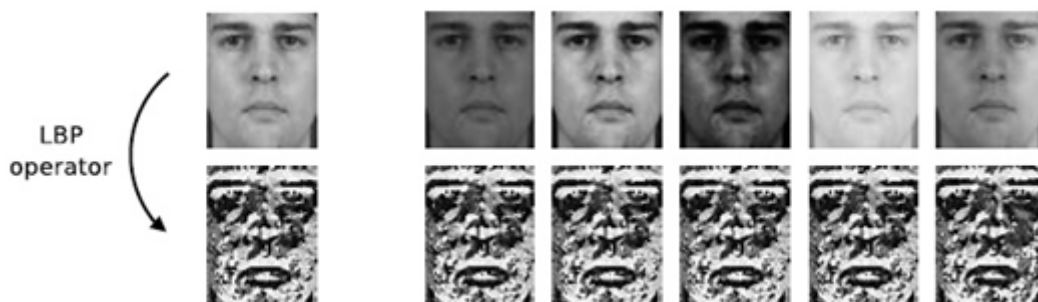


Figure II.5. : Robustesse du LBP aux variations d'illumination.

Sur la rangée du haut, l'image d'origine (à gauche), ainsi que plusieurs images (droite) obtenues en faisant varier la luminosité, contraste et l'éclairage. La rangée du bas montre les images LBP correspondantes qui sont à peu près identiques.

II.5. Le $LBP_{P,R}$

L'opérateur LBP a été étendu ultérieurement en utilisant un voisinage local qui se présente comme un ensemble de points P échantillonnés, régulièrement espacés sur un cercle centré de rayon R autour d'un pixel central (voir la figure suivante).

Les valeurs des points P sont prises et comparées avec la valeur du pixel central. Pour obtenir les valeurs de ces points échantillonnés dans le voisinage, pour tout rayon R , une interpolation est nécessaire.

Le masque de poids utilisé pour calculer les $LBP_{P,R}$ est alors lui aussi circulaire et les poids prennent des valeurs comprises entre 2^0 et 2^{P-1} .

Dans ce qui suit, la notation (P, R) sera utilisée pour les quartiers des pixels qui signifient des points de prélèvement P sur un cercle de rayon R .

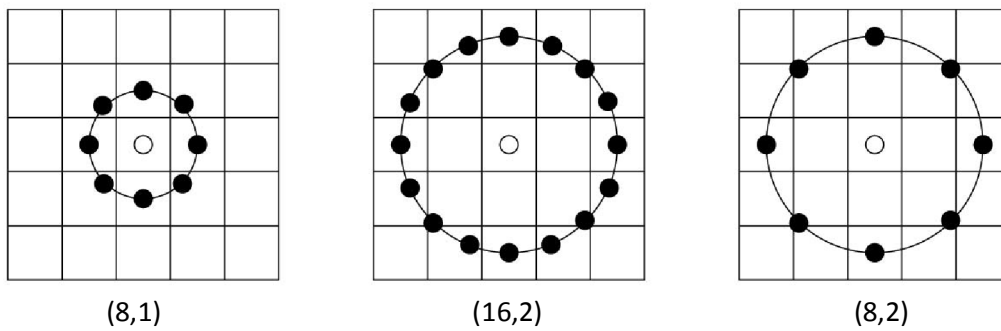


Figure II.6 : Exemples de voisinages avec différentes valeurs de (P, R) .

Le code LBP au pixel (x_c, y_c) sera décrit de la manière suivante, en ajoutant l'indice (P, R) :

$$LBP_{P,R} = \sum_{p=0}^{P-1} S(g_p - g_c) 2^p \tag{II.3}$$

L'opérateur LBP obtenu pour $P = 8$ et $R = 1$ ($LBP_{8,1}$) est très proche de l'opérateur LBP d'origine. La principale différence est que les pixels doivent d'abord être interpolés pour obtenir les valeurs des points sur le cercle (voisinage circulaire au lieu de rectangulaire).

On peut illustrer la méthode calculant le $LBP_{P,R}$ par cette figure :

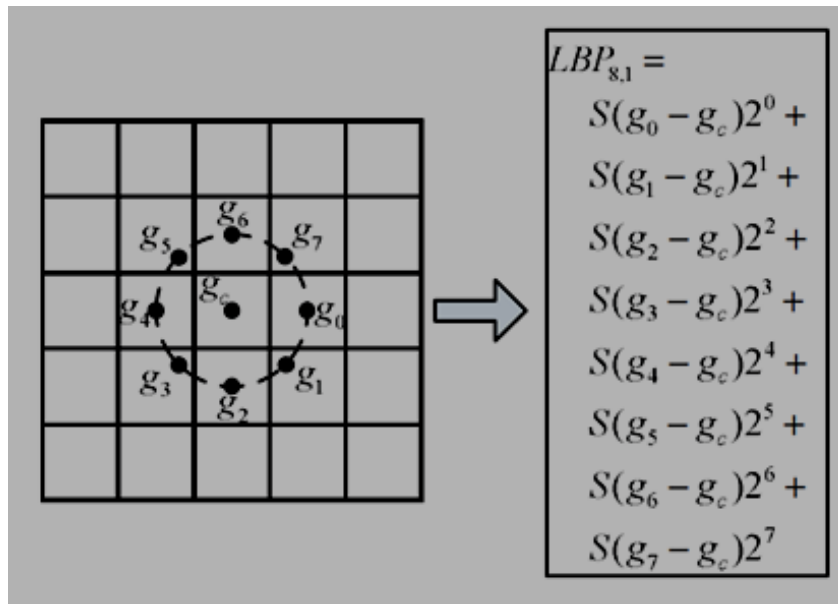


Figure II.7. : Calcul d'un $LBP_{P,R}$ ($P=8, R=1$).

II.6. La dérivation [13]

L'opérateur LBP est dérivé d'une définition générale de la texture dans un quartier local, d'où le nom « dérivation ».

La dérivation du LBP est représentée en 2002 par Ojala et al [14], en se basant sur le principe de la texture :

Une texture est définie dans le voisinage de rayon $R > 0$ d'une image (en niveaux de gris) comme la distribution conjointe de $P+1$ ($P > 0$) pixels.

Soit une fenêtre de texture T d'une image, décrite en fonction des niveaux de gris des pixels dans T :

$$T = t(g_c, g_0, \dots, g_{P-1}) \tag{II.4}$$

Avec g_c : est le niveau de gris du pixel central de T , et $g_0 \dots g_{P-1}$ les niveaux de gris des pixels voisins.

Ainsi définie, la texture peut se représenter, sans perte d'informations par les différences de niveaux de gris entre le pourtour et le centre :

$$T = t(g_c, g_0 - g_c, \dots, g_{P-1} - g_c) \tag{II.5}$$

Si on suppose que ces différences sont localement indépendantes de la valeur du pixel central, on aura :

$$T = t(g_c)t(g_0 - g_c, \dots, g_{P-1} - g_c) \tag{II.6}$$

Dans la pratique, l'hypothèse de l'indépendance n'est pas toujours vraie.

On trouve que $t(g_c)$ n'est qu'une description de luminance d'un pixel de l'image et ne fournit pas d'informations nécessaires pour analyser la texture [15]. Donc, la plupart d'informations caractéristiques de la texture sont préservées dans la distribution conjointe des différences :

$$T \approx t(g_0 - g_c, \dots, g_{P-1} - g_c) \quad (II.7)$$

Cette distribution de dimension P reflète les différences locales des niveaux de gris. Afin de renforcer l'invariance à des changements monotones des niveaux de gris, seul le signe de ces différences est pris en compte. Donc, la distribution devient :

$$T \approx t(\delta(g_0 - g_c), \dots, \delta(g_{P-1} - g_c)) \quad (II.8)$$

Où $\delta(\cdot)$ est une fonction échelon définie comme suit :

$$\delta(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Puis, des poids binomiaux 2^p sont assignés pour chaque signe $\delta(g_p - g_c)$, et cette distribution des différences est transformée à un code LBP unique. Le code caractérisant la texture locale autour d'un pixel (x_c, y_c) de l'image est :

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p \delta(g_p - g_c) \quad (II.9)$$

Ce motif est calculé en chaque point où le voisinage peut être défini, ce qui résulte en un descripteur global décrivant la texture de l'image. La distribution des codes LBP est représentée maintenant par un vecteur de caractéristiques noté T :

$$T \approx t(LBP_{P,R}(x_c, y_c)) \quad (II.10)$$

Remarque sur le voisinage

Les coordonnées des points p ($p = 0, \dots, P - 1$) du voisinage circulaire de rayon R autour du point (x_c, y_c) sont données par :

$$(x_c + R \cos(2p\pi/P), y_c + R \sin(2p\pi/P)) \quad (II.11)$$

La valeur du niveau de gris d'un point du voisinage qui ne «tombe» pas exactement sur un pixel réel est calculée par interpolation à la valeur du pixel le plus proche.

II.7. Le LBP invariant par rotation

Dans de nombreuses applications d'analyse de texture, il est souhaitable d'avoir des caractéristiques qui sont invariantes à des rotations de l'image d'entrée.

Quand l'image d'entrée tourne d'un angle α dans un sens, les pixels voisins tournent autour du pixel central comme le montre cette figure :

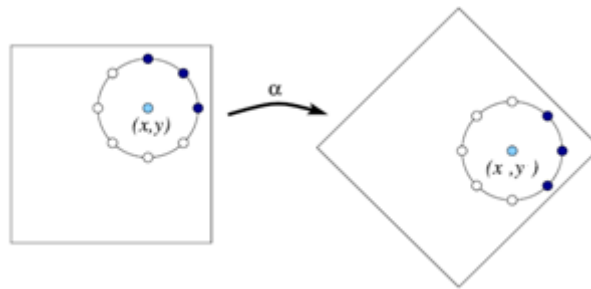


Figure II.8. : Rotation d'une image.

On désigne par $I^{\alpha}(x, y)$ la rotation de l'image $I(x, y)$ par α degrés.

L'opérateur $LBP_{P,R}$ donne 2^P valeurs de sortie différentes correspondant à 2^P formes binaires différentes qui sont formées par P pixels de l'ensemble des voisins. Quand l'image est tournée, les valeurs du niveau de gris g_p se déplacent (tournent) en conséquence selon le périmètre du cercle autour de g_c . Cet effet de rotation cause une variation du code $LBP_{P,R}$ parce que les poids binomiaux 2^p affectés aux g_p lors du calcul du code sont différents, c.à.d. au lieu d'affecter par exemple le poids 2^3 pour g_3 , on l'affecte pour g_4 à cause de la rotation.

On peut traduire aussi cet effet de rotation par le décalage à droite d'un bit du code $LBP_{P,R}$ pour chaque rotation de l'image.

Pour éviter ce changement de rotation, on définit le code LBP invariant à la rotation, noté $LBP_{P,R}^{ri}$ qui est définie en vue de retourner une valeur unique (valeur minimale du code $LBP_{P,R}$) quelle que soit la rotation appliquée [16] :

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \quad i = 0, 1, \dots, P - 1\} \quad (II.12)$$

Où $ROR(x, i)$ correspond au résultat de i décalages circulaires successifs ($|i| < P$) vers la droite (sens horaire) des bits du nombre binaire x (codé sur P bits) correspond à la valeur du $LBP_{P,R}$ et l'exposant ri signifie « invariance par rotation ».

En bref, le code invariant à la rotation se produit en tournant le code original jusqu'à ce que sa valeur minimale soit atteinte.

Avec cette méthode, seulement 36 motifs LBP sont considérés au lieu de 256 dans le cas de $P = 8$, et qui sont représentés dans la figure ci-dessus.

L'avantage est que la texture sera plus efficace en réduisant le nombre des LBP.

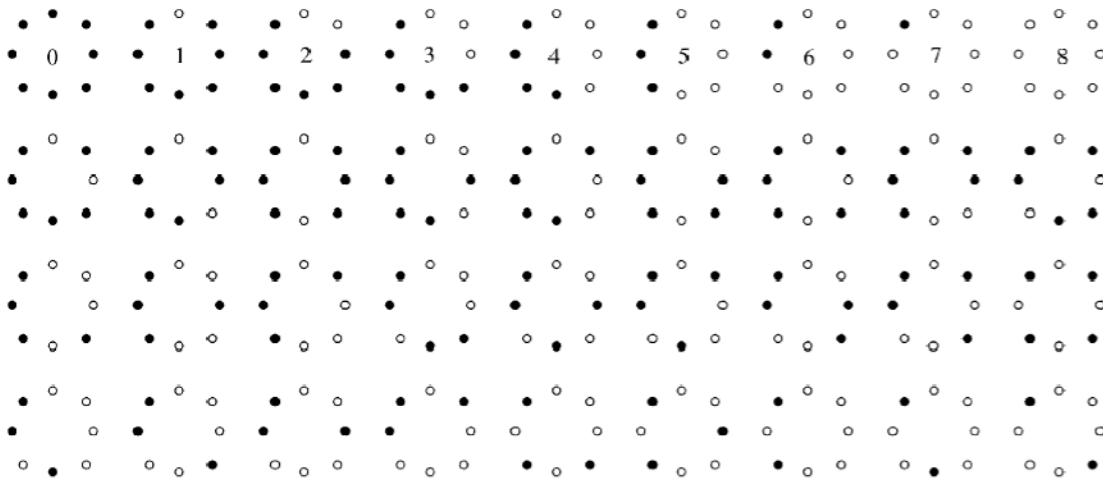


Figure II.9. : Liste des 36 formes binaires locales invariantes en rotation de $LBP_{8,R}^{r1}$.

Dans cette figure, on trouve à titre d'exemple, que le motif 0 permet de détecter les tâches claires (points lumineux), le motif numéro 8 détecte les tâches sombres et le motif 4 les contours.

En pratique, il est difficile de distinguer ces 36 formes lorsqu'une rotation est appliquée sur des images.

Remarque

Si le code est à 8 bits, l'angle de rotation se fait de 45° pour chaque rotation.

Etude d'un exemple :

La Figure suivante illustre six rotations invariantes des codes dans la rangée du haut :

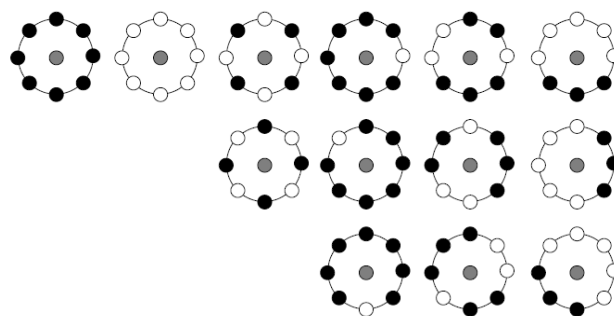


Figure II.10. : Exemple des rotations invariantes des codes LBP.

Les deux premiers quartiers sont immuables par rotation, et il n'y a pas par conséquent d'autres quartiers qui pourraient produire les mêmes codes. Le troisième est particulier, il n'a que deux versions différentes de rotation. Les trois autres quartiers ont un total de huit versions différentes de rotation, dont deux sont présentées à titre d'exemples.

II.8. Le LBP uniforme

Une autre amélioration à l'opérateur d'origine utilise des modèles dits uniformes [13]. Cette extension a été inspirée par le fait que certains motifs binaires surviennent plus fréquemment chez des images de texture que d'autres.

Un motif binaire locale, noté LBP^{u2} est dit uniforme si sa configuration binaire contient au plus deux transitions binaires de 0 à 1 ou de 1 à 0 lorsque cette configuration est considérée circulaire. Par exemple, les modèles de 00000000 (0 transitions), 01110000 (2 transitions) et 11001111 (2 transitions) sont des codes uniformes, alors que les modèles 11001001 (4 transitions) et 01010010 (6 transitions) ne sont pas.

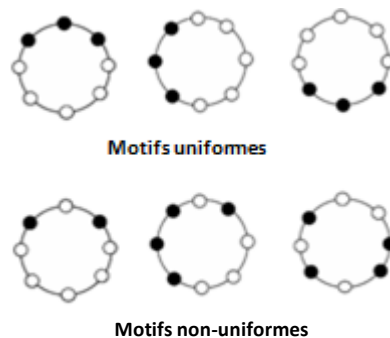


Figure II.11. : Exemples de motifs uniformes et non-uniformes.

- Avantage du LBP uniforme :

Le LBP^{u2} permet de détecter uniquement les textures locales importantes (ou les primitives de texture) comme les spots, les fins de ligne, les bords et les coins (voir la figure II.12). En effet, Ojala et al [14] ont montré que les LBPs uniformes contiennent plus de 90% de l'information d'une image.

Ainsi, pour décrire la texture des images, il est possible d'utiliser uniquement un sous-ensemble de l'ensemble des motifs binaires locaux.

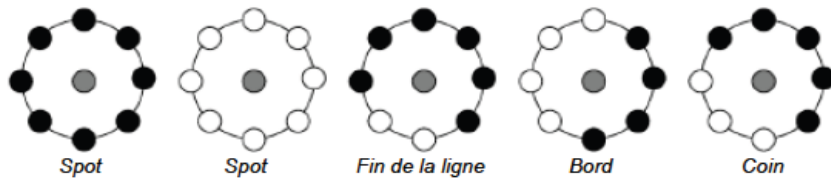


Figure II.12. : Textures particulières détectées par LBP^{u2}.

Pour définir formellement l'«uniformité» d'un quartier, une mesure de l'uniformité U est nécessaire:

$$U(\text{LBP}_{P,R}) = |S(g_{P-1} - g_c) - S(g_0 - g_c)| + \sum_{p=1}^{P-1} |S(g_p - g_c) - S(g_{p-1} - g_c)| \quad (\text{II.13})$$

Si $U(\text{LBP}_{P,R}) \leq 2$ alors le modèle est uniforme.

Dans le calcul des codes LBP, chaque motif uniforme a son propre code de sortie et tous les motifs non-uniformes sont affectés à un seul code.

Le nombre de codes possibles en utilisant seulement les codes uniformes est $P(P - 1) + 2$, où P est le nombre de points de la région avoisinante.

Par conséquent, le nombre des codes différents de sortie des modèles de P bits est $P(P - 1) + 3$.

Par exemple, lors d'utilisation du quartier (8, R), il y a un total de 256 motifs, dont 58 sont uniformes, ce qui donne en sortie 59 étiquettes différentes (58 étiquettes pour les motifs uniformes, plus une seule étiquette pour les motifs non-uniformes).

La figure suivante illustre les 58 modèles uniformes pour le quartier (8, R) :

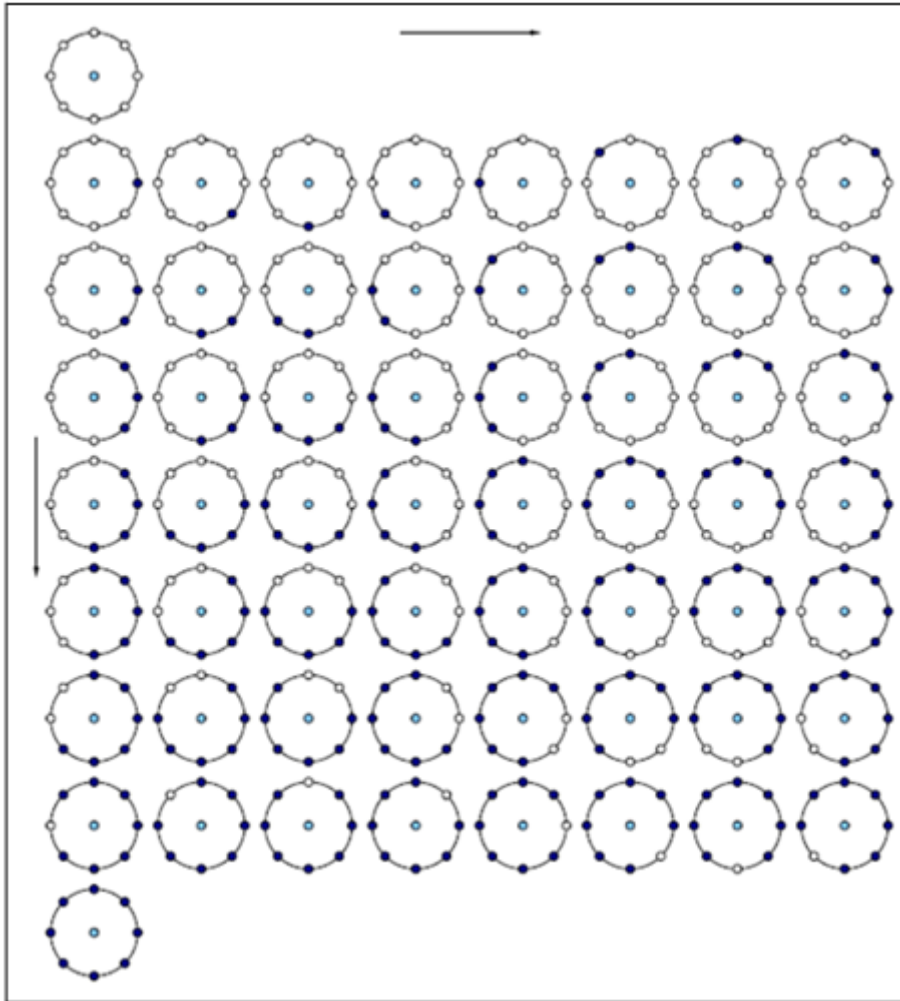


Figure II.13. : Les 58 différents modèles uniformes du quartier (8, R).

La flèche du haut indique le sens de la rotation des modèles et celle du côté représente le nombre du chiffre binaire “1” dans les motifs.

II.9. Le motif uniforme invariant par rotation

En combinant les deux notions invariance par rotation et uniformité, on obtient un opérateur LBP noté $LBP_{P,R}^{riu2}$ qui est défini par la façon suivante :

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} S(g_p - g_c) & \text{si } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{sinon} \end{cases} \quad (II.14)$$

Avec $U(LBP_{P,R})$ une mesure d’uniformité déjà définie.

La notation **riu2**, de l'anglais « rotation invariant uniform », indique que l'opérateur est uniforme et invariant à la rotation.

Le code LBP invariant par rotation correspond à la somme des bits du voisinage s'il est uniforme. Au contraire si le motif est non-uniforme, l'opérateur est affecté à la valeur P+1 [13].

II.10. Les extensions plus récentes du LBP

Plusieurs extensions et modifications du LBP ont été proposées pour augmenter sa robustesse et son pouvoir discriminant. Parmi ces extensions on trouve :

- Le motif ternaire local **LTP** (Local Ternary Pattern) : une extension introduite en 2010 par Tan et Triggs [17] pour sa plus grande résistance au bruit. Au lieu de binariser les valeurs échantillonnées, celle-ci peut prendre trois valeurs selon leur distance à la valeur du pixel central :

$$S(U, i_c, t) = \begin{cases} 1 & \text{si } U \geq i_c + t \\ 0 & \text{si } |U - i_c| < t \\ -1 & \text{si } U \leq i_c - t \end{cases} \quad (II.15)$$

Avec : U correspond aux valeurs de gris des pixels voisins.

i_c correspond à la radiométrie du pixel central.

t est la valeur du seuil fixé par l'utilisateur.

La procédure d'encodage LTP est illustrée dans la figure ci-après. Voici le seuil t a été fixé à 5, de sorte que l'intervalle de tolérance est [49, 59].

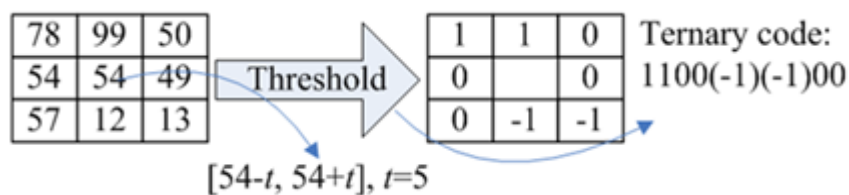


Figure II.14. : Illustration de l'opérateur LTP de base.

Par la suite, pour simplifier les calculs, le code ternaire obtenu sera divisé en deux codes LBP positif et négatif, comme l'illustre la figure II.15, et les histogrammes LBP seront construits pour chaque code, puis ils seront combinés à la fin des calculs.

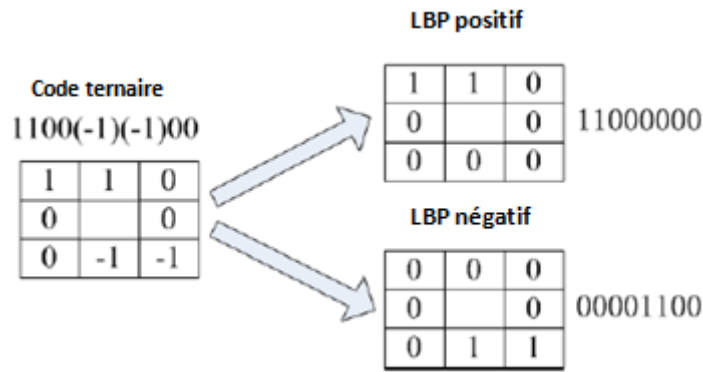


Figure II.15. : Division d'un code LTP en codes LBP positif et négatif.

- Le motif local quantifié **LQP** (Local Quantized Patterns) : Récemment Hussain et al [18] ont introduit les motifs locaux quantifiés. Ceux-ci permettent d'utiliser les LBP ou les LTP sur des voisinages plus complexes, c'est à dire des voisinages constitués de plusieurs cercles concentriques et avec un plus grand nombre d'échantillons ;
- Le centre-symétrique LBP (**CS-LBP**) : CS-LBP présente plusieurs avantages comme la tolérance aux changements d'éclairage, la robustesse des zones de l'image et l'efficacité de calcul [19].

Comme l'illustre la (Figure II.16), au lieu de comparer le niveau de gris de chaque pixel avec le pixel central, les paires symétriques du pixel central sont comparées.

Les caractéristiques CS-LBP peuvent être calculées par l'équation:

$$CS\ LBP_{P,r,T} = \sum_{i=0}^{\frac{N}{2}-1} S\left(g_i - g_{i+\frac{N}{2}}\right) 2^i \quad (II.16)$$

Avec :

$$S(x) = \begin{cases} 1 & \text{si } x \geq T. \\ 0 & \text{autrement.} \end{cases}$$

g_i et $g_{i+\frac{N}{2}}$ correspondent au niveau de gris du centre-symétrique des paires des pixels

(N au total) régulièrement espacés sur un cercle de rayon r . T est une valeur faible utilisée comme un seuil.

En [20], les auteurs ont utilisé le descripteur CS-LBP pour décrire la région autour d'un point d'intérêt et leurs expériences montrent les performances de cette méthode.

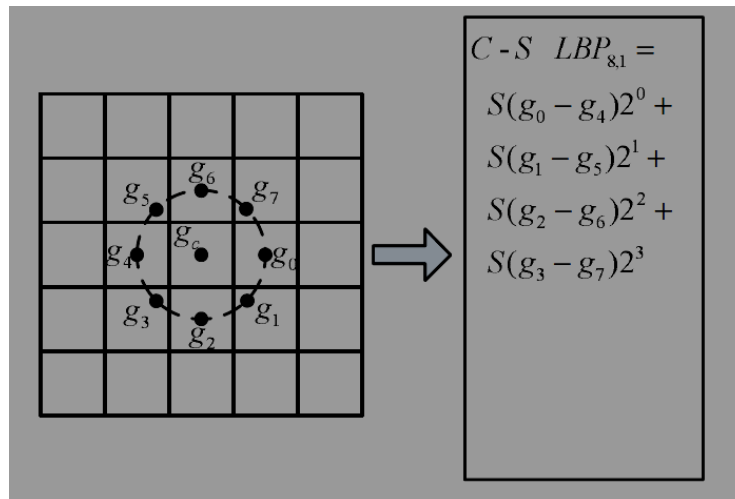


Figure II.16. : Illustration du CS-LBP.

- **MLBP** (Multivariate LBP) : Développé par Arco Lucifer [21]. Largement utilisé pour la classification et la segmentation d'images ;
- LBP variance (**LBPV**) : proposé par Zhenhua Guo [22] pour caractériser les informations du contraste dans une dimension histogramme LBP ;
- Amélioration LBP (**ILBP**) : introduit par Jin et al [23]. La principale différence entre ILBP et LBP réside dans la comparaison de tous les pixels (incluant le pixel central) avec la moyenne de tous les pixels dans le noyau pour le ILBP.

II.11. Conclusion

Le LBP est devenu aujourd'hui une approche très populaire utilisée dans de nombreux domaines d'application.

Nous avons vu dans ce chapitre, comment le LBP a été introduit à partir d'une définition de la texture dans un quartier de l'image. Nous pouvons remarquer aisément la simplicité de calcul de cet opérateur, sa robustesse à la rotation des images et aux changements d'illumination grâce à la caractéristique invariance par rotation ($LBP_{P,R}^i$) et encore plus, l'efficacité des modèles uniformes LBP^u , qui sont riches en informations dans la description des textures importantes.

Après la phase de la description des textures, il sera nécessaire d'isoler les zones texturées dans les images, afin de simplifier leur étude. La technique qui permet ceci est la segmentation qu'on va aborder dans le chapitre suivant.

III.1. Introduction

La recherche des régions qui composent une image est appelée segmentation. La segmentation est une partie très importante dans l'analyse des images et l'un des sujets les plus étudiés actuellement, car elle a des applications dans plusieurs domaines pratiques, comme dans la robotique, l'analyse d'images médicales, la détection d'objets, ou encore dans le domaine de l'imagerie spatiale et bien d'autres [24].

III.2. Définition de la segmentation

La segmentation permet de séparer des objets du fond d'une image et consiste à diviser cette même image en plusieurs régions où, chaque région est homogène au sens d'un critère (forme, texture, intensité etc.), ou d'une manière plus générale, elle consiste à rassembler les différents pixels ayant des propriétés communes. Si la segmentation sépare deux classes de pixels, elle est appelée binarisation.

Nous pouvons adopter la définition mathématique suivante pour la segmentation :

"La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image A en sous-ensembles R_i , appelés régions tels qu'aucune région ne soit vide, l'intersection entre deux régions est vide et l'ensemble des régions recouvre toute l'image " [25].

III.3. Exemple de segmentation basée sur le seuillage d'histogramme [1]

La segmentation par seuillage d'histogramme exploite les histogrammes de l'image. On citera essentiellement la méthode du seuillage des niveaux de gris.

Un seuil est une valeur numérique correspondant à un paramètre de l'image.

Le principe général du seuillage consiste à chercher une valeur appropriée du seuil, puis de classifier tous les pixels de l'image selon la valeur de leurs niveaux de gris par rapport à ce seuil et ce, afin de séparer les régions d'intérêt et l'arrière plan de l'image. Le seuillage peut être :

- Global : un seuil pour toute l'image ;
- Local : un seuil pour une portion de l'image ;
- Adaptatif : un seuil qui s'ajuste selon les images, ou les parties de l'image.

Le seuillage le plus simple est le seuillage global, il a pour principe :

Soit p un pixel et V une fonction définissant le niveau de gris et S un seuil :

$$V(p) = \begin{cases} 0 & \text{si } V(p) < S \\ 1 & \text{si } V(p) \geq S \end{cases}$$

Le résultat sera évidemment une image binaire :

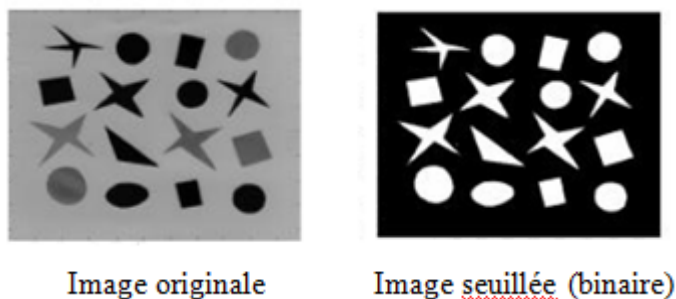


Figure III.1. : Exemple de segmentation par seuillage.

L'histogramme des niveaux de gris est utilisé afin d'en déduire le seuil. Si un seuillage simple est utilisé, on obtient une séparation de l'histogramme en deux modes uniquement.

Soit I une image composée d'un objet clair sur un fond sombre, dont l'histogramme des niveaux de gris est représenté par la figure III.2.

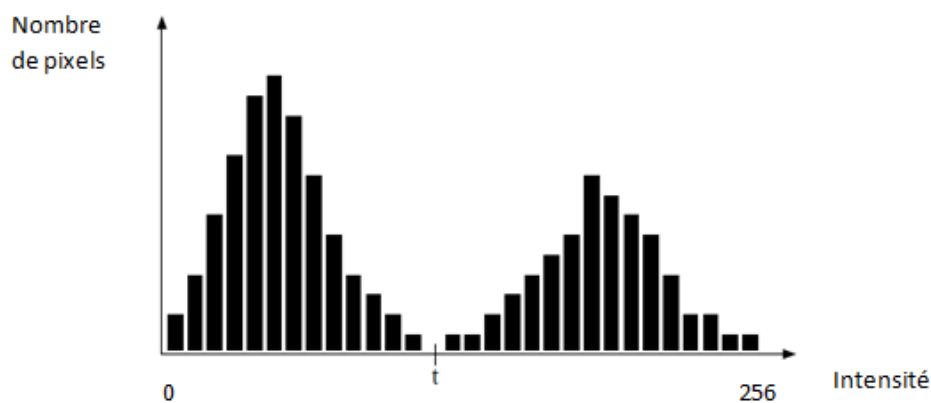


Figure III.2. : Exemple d'histogramme d'une image.

On place le seuil t entre les deux modes. Les pixels dont le niveau de gris est supérieur à t seront attribués à l'objet, les autres pixels au fond.

Pour les cas plus complexes d'histogrammes présentant plus de deux classes, un seuillage multiple devient nécessaire.

III.4. Les approches de segmentation

Une multitude de méthodes de segmentation sont proposées dans la littérature. Elles se répartissent en quatre grandes familles : la segmentation par approche contour, par approche région et par classification.

III.4.1. Approche contour

Un contour est défini comme une zone de l'image où l'intensité des pixels change brusquement. Cette variation dans l'image est le passage d'un niveau de gris à un autre.

L'approche contour consiste donc à identifier les frontières (transitions) qui séparent les différentes régions de l'image.

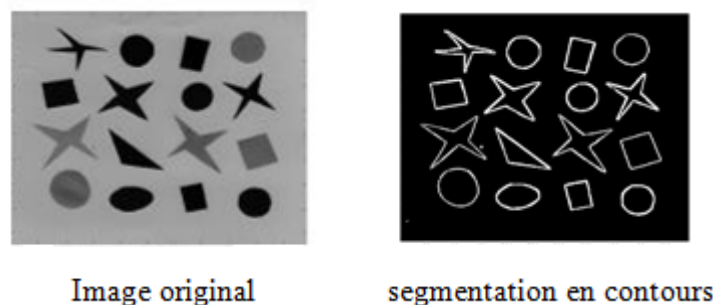


Figure III.3. : Exemple de segmentation par approche contour.

Notons que la variation peut être détectée par le maximum d'une dérivée première de la fonction niveau de gris d'un point, ou par le passage par zéro de la seconde dérivée. Les opérateurs les plus utilisés pour détecter les contours sont l'opérateur gradient, Robert, Sobel, Perwitt et le Laplacien ainsi que beaucoup d'autres [24].

III.4.2. Approche région

L'approche région a pour objectif de décomposer l'image en un ensemble de régions connexes, les plus homogènes possibles et les plus différentes par rapport aux autres régions. Chaque pixel est affecté à une région unique après une partition de l'image. Notons qu'une région est dite homogène si les caractéristiques des pixels la constituant sont sensiblement identiques. Or pour une image en niveau de gris, un pixel peut être caractérisé soit par son niveau de gris, soit par un ensemble de paramètres de texture.

La méthode par croissance des régions est l'une des méthodes de segmentation par régions les plus connues. Son principe est de réunir de façon itérative un ensemble de points connectés et adjacents en une région de plus en plus large, en fonction d'un critère de mesure de similarité. Pour définir une région, on définit un germe dans la région d'intérêt qui sert comme un point de départ pour l'agrégation. Ce germe peut être sélectionné de façon aléatoire ou automatique. Lors d'une itération du processus, les pixels adjacents à la région sont étudiés. S'ils vérifient les conditions d'homogénéité (critère défini au préalable), le pixel sera alors agrégé dans la région. Les pixels non intégrés dans la région peuvent générer eux-mêmes de nouvelles régions [1].

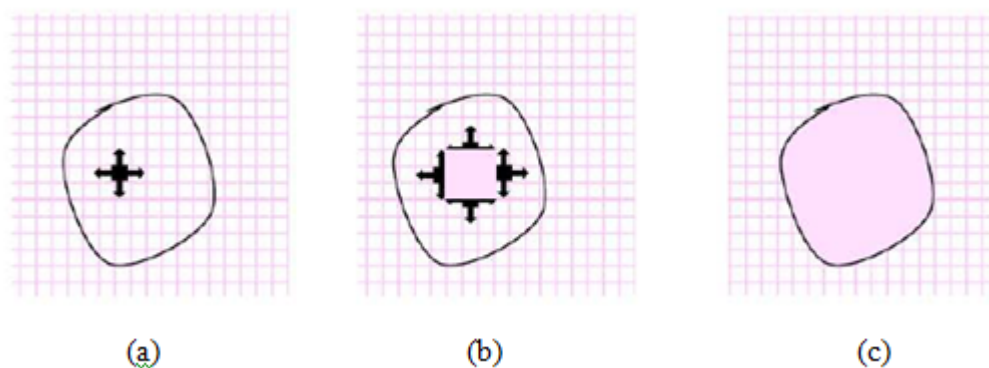


Figure III.4. : Le processus d'un algorithme de croissance par région.

(a) début de processus, le point noir présente le germe et les flèches la direction de croissance. (b) la croissance des pixels après quelques itérations. (c) le résultat de la segmentation.

A titre d'exemple, la différence entre la moyenne d'intensité d'une région et l'intensité d'un pixel adjacent non classé peut être considérée pour mesurer la similarité entre ce pixel et la région adjacente. Un seuil de décision est ensuite appliqué sur cette différence de luminosité afin d'intégrer ou non, un pixel à la région adjacente.

III.4.3. Classification [28]

C'est cette approche qui nous intéresse dans ce travail. La classification consiste à organiser un ensemble de données multidimensionnelles en un **ensemble fini de classes** selon un ou plusieurs critère(s) de classification à l'aide d'un classifieur. On parle alors de partitionnement ou de regroupement.

Il existe de nombreuses méthodes de classification de données multidimensionnelles. La figure III.5 illustre le principe de la classification.

Les données sont représentées par des points (vecteurs) dans un espace à n dimensions (dans notre exemple, $n=2$). En sortie de la classification, on obtient k classes (ou familles) de points selon les critères donnés et le classifieur choisi (dans notre exemple, $k=3$).

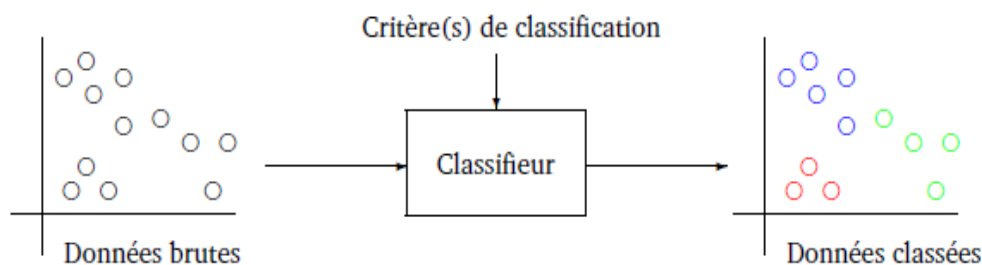


Figure III.5. : Principe de la classification.

Les pixels de l'image sont regroupés en un ensemble de k classes disjointes. A chaque pixel de l'image, on attribue une étiquette parmi les k étiquettes qui correspondent chacune à une classe.

Mathématiquement, la classification est définie de la manière suivante [2] :

Soit $X = \{X_1, X_2, \dots, X_N\}$ l'ensemble des N observations (pixels) à classer.

Chaque observation X_i est caractérisée par p paramètres $X_i = \{x_{i1}, x_{i2}, \dots, x_{ip}\}$.

Soit $C = \{c_1, c_2, \dots, c_k\}$ l'ensemble des k classes de l'image.

La classification consiste à répartir l'ensemble des N pixels en k classes tel que :

- $c_i \neq \emptyset$ pour $i = 1, 2, \dots, k$
- $c_i \cap c_j = \emptyset \quad \forall i \neq j$
- $\cup c_i = C$

III.5. Les approches de classification

On peut distinguer deux approches de classification :

- Approche supervisée.
- Approche non supervisée.

III.5.1. Approche de classification supervisée (dirigée) [2]

La classification supervisée est une technique de classification automatique où l'on cherche à produire automatiquement des règles, pour regrouper des individus en un certain nombre de classes. Dans cette approche, le nombre de classes possibles et l'appartenance de chaque individu (pixel) à l'une de ces classes sont connues à priori. L'approche supervisée nécessite l'intervention d'un expert qui dispose d'un échantillon de l'image (ensemble de pixels dont le classement est connu), dont le but d'apprendre les caractéristiques de chaque classe (phase d'apprentissage), pour ensuite, associer à chaque pixel la classe qui lui ressemble le plus (phase de classification).

Il existe plusieurs méthodes de classification supervisée comme : les réseaux de neurones artificiels, K plus proches voisins, maximum de vraisemblance etc..

III.5.2. Approche de classification non supervisée (non dirigée) [26]

Cette approche consiste à découper l'espace de représentation en zones homogènes, selon un critère de ressemblance entre les individus.

Aucune connaissance à priori n'est disponible sur la nature et le nombre de classes existant dans un ensemble d'individus. De ce fait, la classification non supervisée sert à regrouper les individus qui se ressemblent et séparer ceux qui sont dissemblables pour former l'ensemble de classes homogènes.

Parmi les méthodes de classification non supervisée on peut citer : la méthode des k-means (k-moyennes), C-means flou, champ de Markov etc..

III.6. Etude de quelques méthodes de classification

Nous proposons de décrire quelques méthodes de classification supervisée et non supervisée. Notre choix s'est porté sur la méthode de vraisemblance pour la classification supervisée et la méthode des k-means pour la classification non supervisée.

III.6.1. La méthode du maximum de vraisemblance [26]

La classification par maximum de vraisemblance découle de la théorie de probabilité et du théorème de Bayes. Pour chaque pixel de l'image, on détermine sa probabilité d'appartenir à telle ou telle classe à partir des vérités terrains.

Le pixel est affecté à la classe pour laquelle la probabilité est la plus élevée. La totalité de l'image est classée sur cette base.

Cette méthode est très utilisée en traitement d'images satellitaire et la télédétection.

La règle de Bayes est un critère fondamental dans tout problème d'estimation ou de décision. Elle permet de classer automatiquement des objets quelconques selon une approche basée sur l'estimation des probabilités conditionnelles.

On cherche à classer une image en K classes. Soient :

$P(x)$: La probabilité d'être le pixel x .

$P(C_i)$: La probabilité d'appartenir à la classe C_i .

$P(x / C_i)$: La probabilité d'être le pixel x sachant qu'on se trouve dans la classe C_i .

$P(C_i / x)$: La probabilité que le pixel x appartienne à la classe C_i .

$$\text{D'après la règle de Bayes : } P(C_i / x) = \frac{P(x/C_i) \times P(C_i)}{P(x)} \quad (III.1)$$

On affecte x à la classe C_i si et seulement si la probabilité que x appartienne à la classe C_i est supérieure à la probabilité que x appartienne à n'importe quelle autre classe j : $P(C_i / x) > P(C_j / x)$ quelque soit j .

Ou $[P(x / C_i) \times P(C_i)] > [P(x / C_j) \times P(C_j)]$ (conformément à la règle de Bayes).

Exemple :

➤ Classification entre deux catégories :

Pour deux classes C_1 et C_2 , pour une observation x :

$$P(C_1 / x) = \frac{P(x/C_1) \times P(C_1)}{P(x)}$$

$$P(C_2 / x) = \frac{P(x/C_2) \times P(C_2)}{P(x)}$$

Comment décider que x appartient à C_1 ou à C_2 ?

Une idée simple serait de calculer $P(C_1 / x)$ et $P(C_2 / x)$. Donc pour tout x :

Si $P(C_1 / x) > P(C_2 / x)$ alors $x \in C_1$ sinon $x \in C_2$.

Par la règle de Bayes ceci est :

Si $P(x / C_1) P(C_1) > P(x / C_2) P(C_2)$ alors $x \in C_1$ sinon $x \in C_2$.

On note que si $P(C_1) = P(C_2)$ (on suppose en général que les probabilités d'avoir une classe sont égales) alors ceci devient :

Si $P(x/C_1) > P(x/C_2)$ alors $x \in C_1$ sinon $x \in C_2$.

$P(x/C_1)$ est connue comme la vraisemblance de x par rapport à C_1 .

Dans le cas général on remplace les deux classes par K classes.

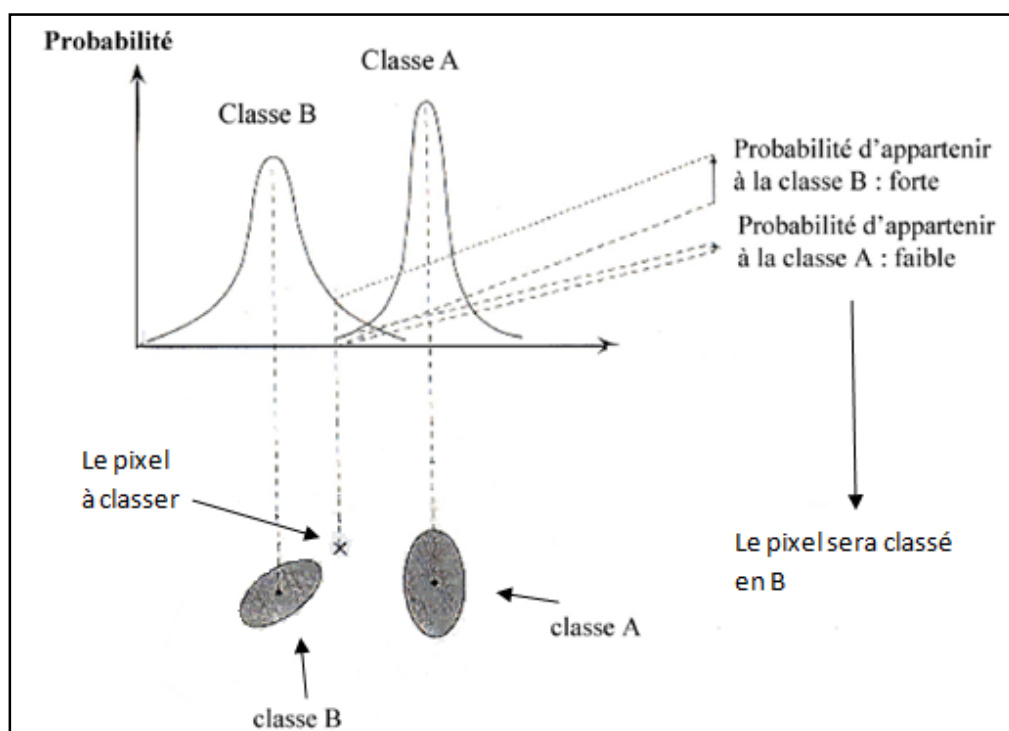


Figure III.6. : Classifieur par Maximum de Vraisemblance dans le cas de deux classes.

La classification par la méthode du maximum de vraisemblance suppose que tous les pixels appartiennent forcément à une des classes existantes. Cependant, ce n'est pas toujours une affectation pertinente, et quelques pixels peuvent être assignés de force dans une classe qui ne correspond pas réellement à ses caractéristiques.

III.6.2. La méthode des k-means

La méthode des k-means (appelée aussi méthode des centres mobiles), introduite par J. Mac Queen [27] en 1967, est un outil de classification classique, qui permet de répartir un ensemble de données (individus ou pixels de l'image) en k classes homogènes.

La valeur de k n'est pas connue, elle est choisie selon une connaissance préalable (combien de régions sont souhaitées, etc.) ou de manière aléatoire. Chaque groupe est représenté par son centre de gravité qui correspond à la moyenne des éléments (pixels) du groupe. Cette méthode de classification est basée sur le mouvement de ces centres, qui se déplacent vers une position de stabilité avec chaque itération.

Le centre de gravité d'une classe C_k est calculé comme suit :

$$\mu_k = \frac{1}{N} \sum_{x_i \in C_k} x_i \quad (III.2)$$

Avec : N est le nombre d'observations (individus ou pixels).

k est le nombre de classes souhaitées.

$i=1, \dots, k$ attribut de la classe d'un individu x .

Après avoir déterminé les paramètres (intensité par exemple) de toute région, chaque individu est affecté au centre le plus proche (généralement selon une mesure de distance euclidienne). Après l'affectation de toutes les données, les centres de gravités sont recalculés, ils constituent les nouveaux représentants des groupes. Le critère d'arrêt de cet algorithme correspond à la stabilité des classes (quand aucune donnée ne change pas de groupe), c.à.d. lorsque les centres des classes ne changent pas durant deux itérations successives :

$$\mu(n, p) = \mu(n, p+1) \quad (III.3)$$

Avec $\mu(n, p)$: centre de gravité de classe n ($n=1, \dots, k$) à la $p^{\text{ème}}$ itération.

L'algorithme de k-means minimise la fonction suivante :

$$J = \sum_i^k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (III.4)$$

Où $\|\cdot\|$ représente la norme euclidienne, qui mesure la distance entre un point et un centre.

On peut résumer cet algorithme comme suit :

Entrées

Ensemble de N données noté par x , Nombre de groupes souhaité noté par k ,

Centres μ des classes ;

Sorties Une partition finale de k groupes $\{ C_1, C_2, \dots, C_k \}$;

Début

- Initialisation aléatoire des k centres de classes ;

Répéter

- Affecter chaque individu au centre le plus proche ;

$$x_i \in C_k \text{ si } \forall j \ \|x_i - \mu_k\| = \min_j \|x_i - \mu_j\| \quad (III.5)$$

Avec μ_k le centre de la classe k

- Calculer les centres de gravités des classes obtenues ;

$$\mu_k = \frac{1}{N} \sum_{x_i \in C_k} x_i$$

Jusqu'à convergence de l'algorithme vers une partition stable ;

Fin.

La figure suivante illustre un exemple de classification par la méthode des k-means :

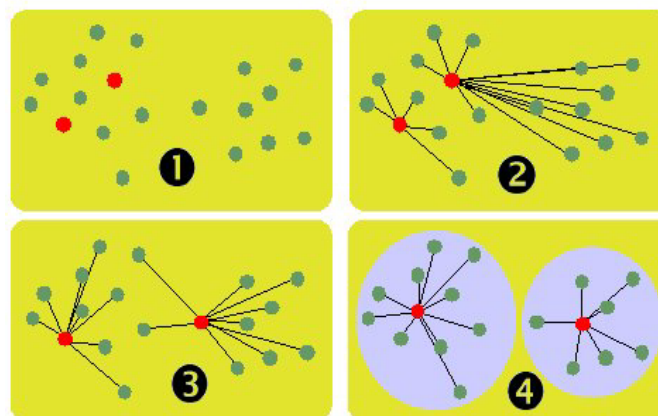


Figure III.7. : Exemple de classification avec la méthode de k-means.

Cette technique n'implique pas de longs calculs et converge rapidement. Par contre, elle est sensible aux conditions initiales du fait qu'elle impose de donner le nombre de classes désirées au début de l'algorithme. En plus, elle nécessite l'existence d'une distance.

À la fin de la classification par les k-means, on obtient des classes strictes, c'est-à-dire que chaque individu n'appartient qu'à une seule classe. Les individus situés en périphérie d'une classe, près de la frontière d'une autre classe, posent le problème d'appartenance stricte à une classe unique [28].

III.7. Conclusion

Nous avons consacré ce chapitre à l'étude de la segmentation et de la classification des images. Dans un premier temps, nous avons introduit la notion de la segmentation, puis nous avons donné un aperçu sur les différentes approches de cette dernière, en insistant particulièrement sur l'approche classification qui regroupe deux grandes parties : supervisée et non supervisée. Parmi toutes les méthodes de la classification supervisée, nous avons abordé la méthode du maximum de vraisemblance, puis nous avons choisi la méthode des k-means comme un bon exemple pour illustrer la technique de classification non supervisée.

IV.1. Introduction

Après avoir présenté un état de l'art sur quelques méthodes de caractérisation des textures (essentiellement la méthode LBP et ses extensions) dans les chapitres précédents, nous exposerons dans ce chapitre les résultats de certaines de ces méthodes.

Afin de mieux évaluer les méthodes programmées et de valider les résultats obtenus, nous avons choisi comme expérimentation, des images de taille 256×256 codées sur 8 bits.

IV.2. Environnement matériel et logiciel de programmation

Les méthodes présentées dans notre travail sont implémentées en langage C++, sous l'environnement Code::Blocks version 4.9, sur un micro ordinateur portable Lenovo, ayant une fréquence de 2.3 GHz, une mémoire vive (RAM) de 2 Go et un disque dur de 300 Go.

❖ Présentation du langage C++

Le langage de programmation C++ est une extension du langage C, en lui apportant le concept orientée objets. Il est développé par Bjarne Stroustrup dans les laboratoires Bell (AT&T) au début des années 1980. Le langage C++ est l'un des langages de programmation les plus utilisés dans le monde, il s'impose aujourd'hui, grâce à ses fonctions très puissantes, comme le langage de référence de développement des programmes, aussi bien pour les applications scientifiques et informatiques [29].

❖ L'environnement Code ::Blocks

Le Code ::Blocks est un IDE (Environnement de Développement Intégré), spécialement conçu pour la programmation en langages C et C++ . Sachant qu'un IDE est un environnement qui contient un ensemble complet d'outils, permettant au développeur d'écrire, d'éditer et de modifier des fichiers sources, de compiler et déboguer un programme avec un seul logiciel.

Nous avons choisi, pour développer nos programmes, de travailler sous le logiciel Code ::Blocks pour plusieurs raisons : il est libre et multiplateformes (version pour Windows et Linux), facilement extensible et configurable à nos besoins, complet et puissant que les autres IDE (DEV+++, Visual C++, etc.), simple d'utilisation et personnalisable, simple à installer, etc..

L'illustration ci-dessous montre l'apparence de la fenêtre de l'interface utilisateur de Code ::Blocks.

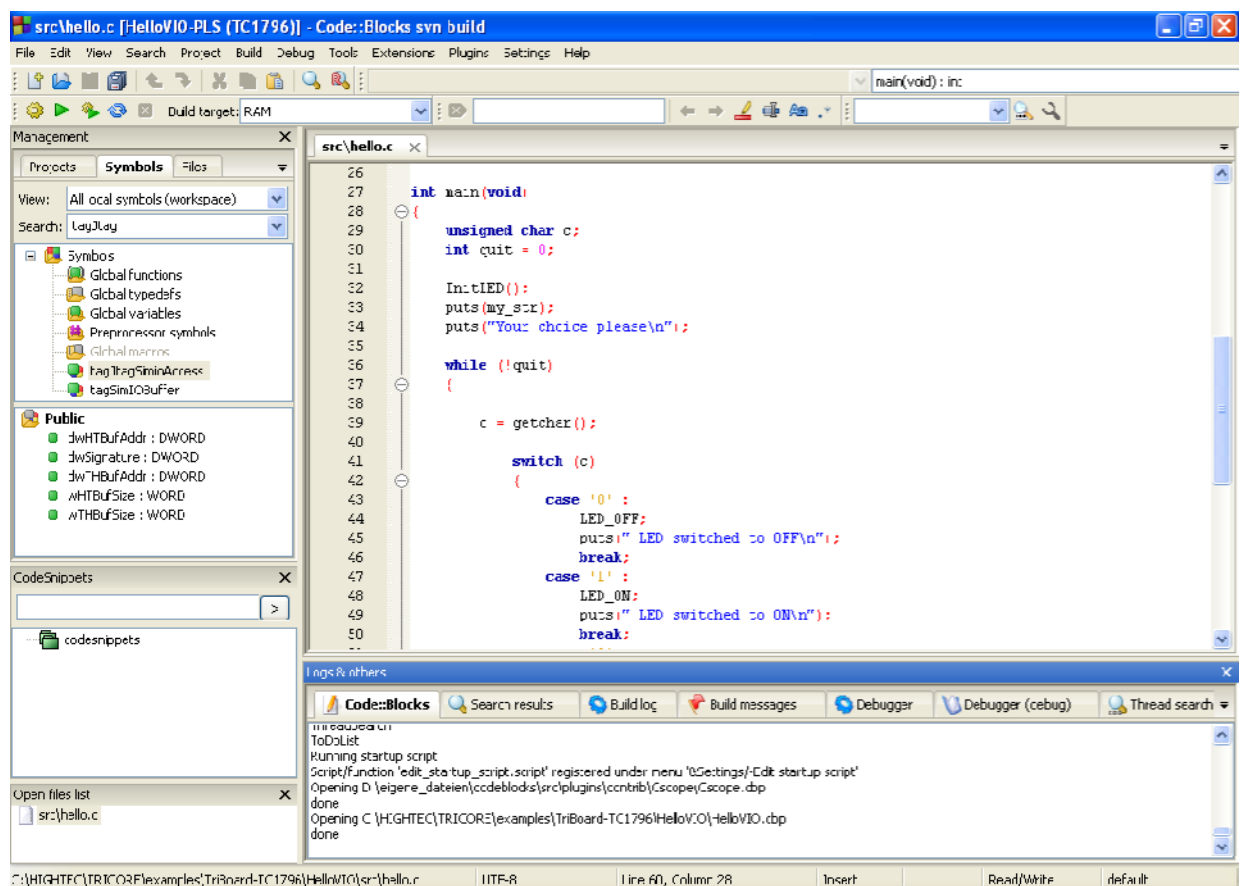


Figure IV.1. : Environnement de développement Intégré (IDE) de Code ::Blocks.

IV.3. Implémentations et discussions

➤ Application de l’approche LBP sur l’image « enfant »

On applique la formule LBP suivante sur une matrice 3*3 pixels de l’image.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c) \times 2^p \quad (IV.1)$$

g_c est le niveau de gris du pixel central et g_p ($p = 0, \dots, 7$) est celui des pixels voisins.

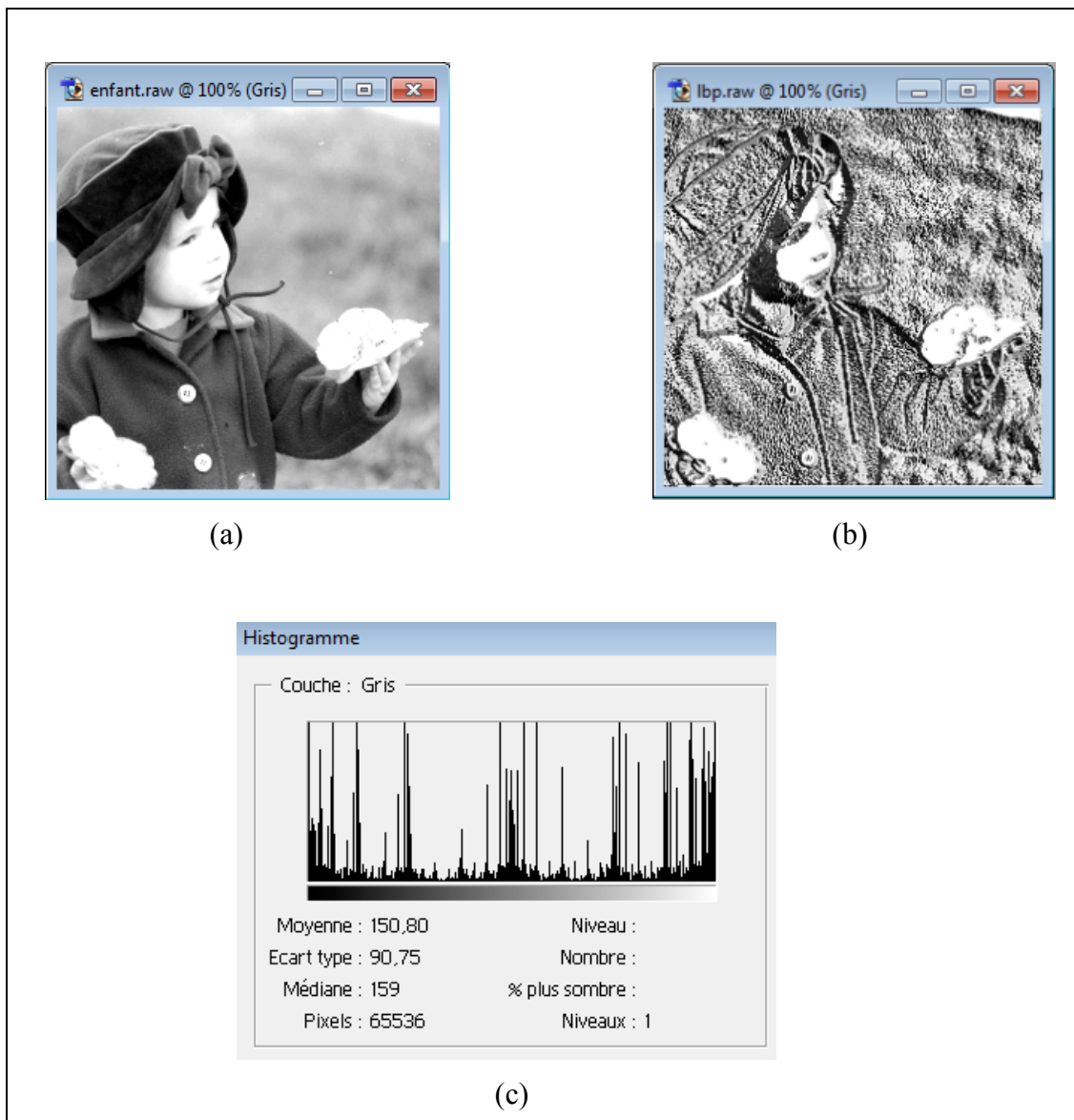


Figure IV.2. : (a) : image enfant, (b) et (c) : image et histogramme LBP correspondants.

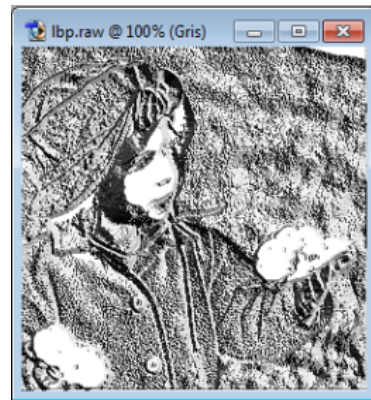
On remarque bien que l'application du descripteur LBP sur l'image (a) a augmenté les contrastes des régions sombres et a atténué celui des régions lumineuses, ainsi qu'il a bien fait apparaître la forme des textures et accentué les reliefs.

Par exemple, dans l'image originale, on remarque très mal les trous des boutons de la veste. Par contre, nous constatons aisément qu'ils sont bien mis en évidence dans l'image résultante.

La forme de l'arrière plan de l'image a l'aire d'une surface plane. Sur l'image résultante, on remarque bien les formes des courbures sur les textures.

➤ Robustesse du LBP aux changements d'illuminations

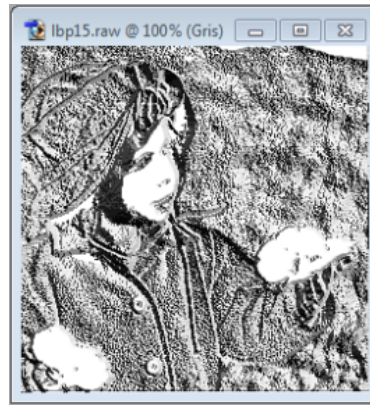
Nous prendrons l'image « enfant », puis nous ajusterons sur sa valeur de luminance à l'aide de Photoshop, de façon à avoir des images ayant différentes valeurs de luminance et ce, pour calculer le LBP de chaque image et faire une comparaison.



Luminance = 0



Luminance = 10



Luminance = 15



Luminance = 30

Figure IV.3. : Les images enfant (à gauche), images LBP correspondantes (à droite).

Les résultats obtenus nous confirment la robustesse et l'invariance de la technique LBP aux changements de la luminosité et l'éclairage, du fait que les images résultantes sont à peu près identiques.

➤ Le $LBP_{P,R}$ et l'effet de la rotation

Pour une position initiale du voisinage circulaire de la matrice 3×3 de l'image « enfant », nous calculons le code $LBP_{P,R}$ ($P = 8, R = 1$) comme suit :

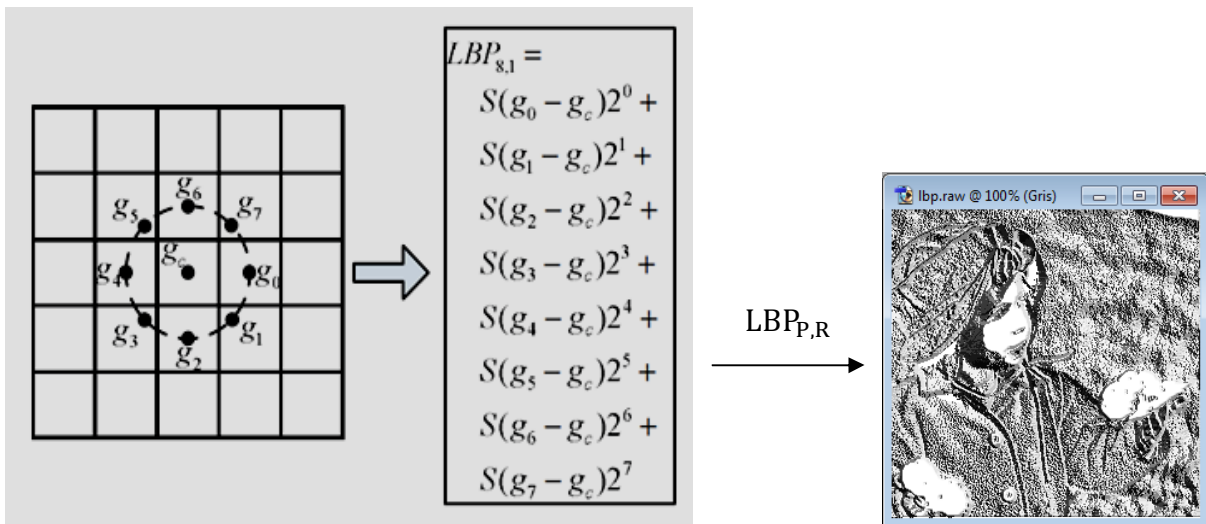


Figure IV.4. : Calcul d'un $LBP_{P,R}$ ($P=8$, $R=1$).

Comme nous l'avons déjà vu dans le deuxième chapitre, l'effet de la rotation d'une image cause la rotation du voisinage circulaire, par α degré, autour du centre. Comme les poids binomiaux 2^p associés pour chaque voisin g_p ($p = 0, \dots, 7$) lors du calcul du code LBP sont différents, ce code change alors de valeur pour chaque rotation.

Puisque on travail sur un voisinage circulaire de 8 pixels ($P = 8$), l'angle de rotation α sera de 45° pour chaque rotation.

Dans notre application, nous avons calculé le code LBP de l'image « enfant » pour toutes les rotations possibles du voisinage à partir de la position initiale de l'image. Nous avons obtenu les résultats suivants :

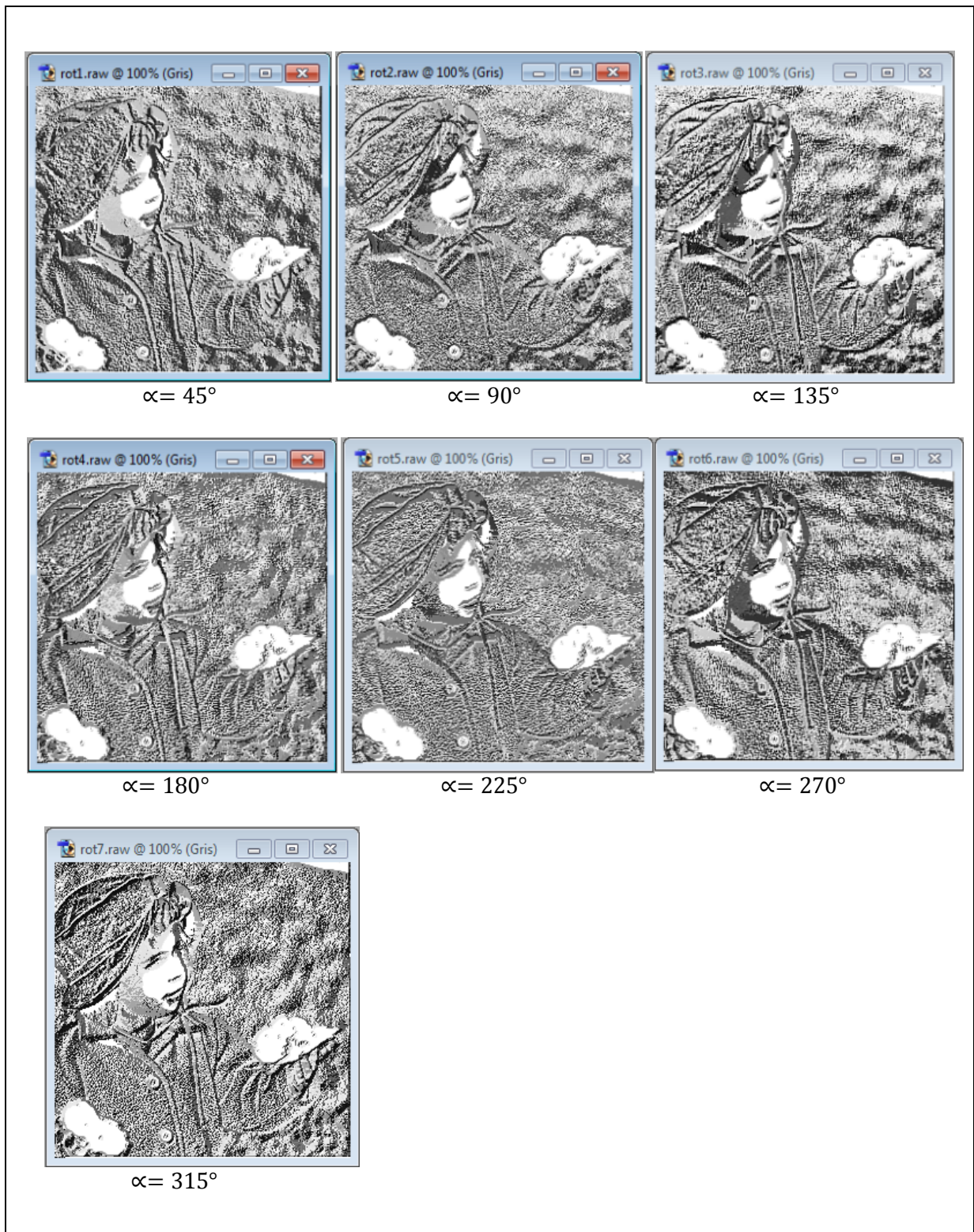


Figure IV.5. : Les images LBP obtenues pour chaque rotation du voisinage.

Les résultats obtenus nous montrent que le LBP est sensible à la rotation.

➤ Invariance par rotation

Pour éviter l'effet de la rotation présenté en haut, nous devons sélectionner la valeur LBP minimale de toutes les valeurs LBP possibles lors de la rotation de l'image, en utilisant la formule :

$$LBP_{P,R}^i = \min\{ROR(LBP_{P,R}, i) \quad i = 0, 1, \dots, P - 1\} \quad (IV.2)$$

Le résultat donne l'image suivante :

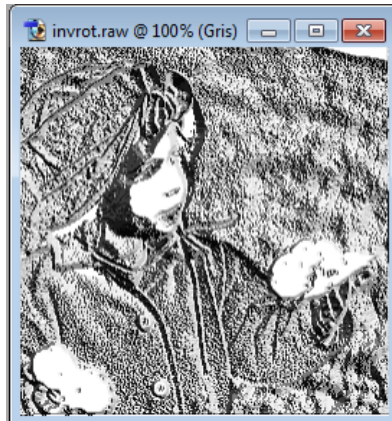


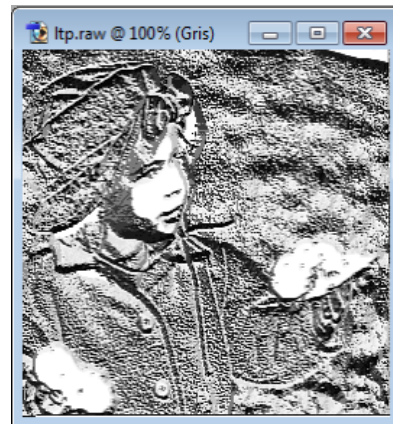
Figure IV.6. : résultat de l'invariance par rotation.

➤ Application de l'extension LTP

Parmi les extensions plus récentes du LBP, nous avons choisi d'appliquer l'extension LTP sur l'image test, en :



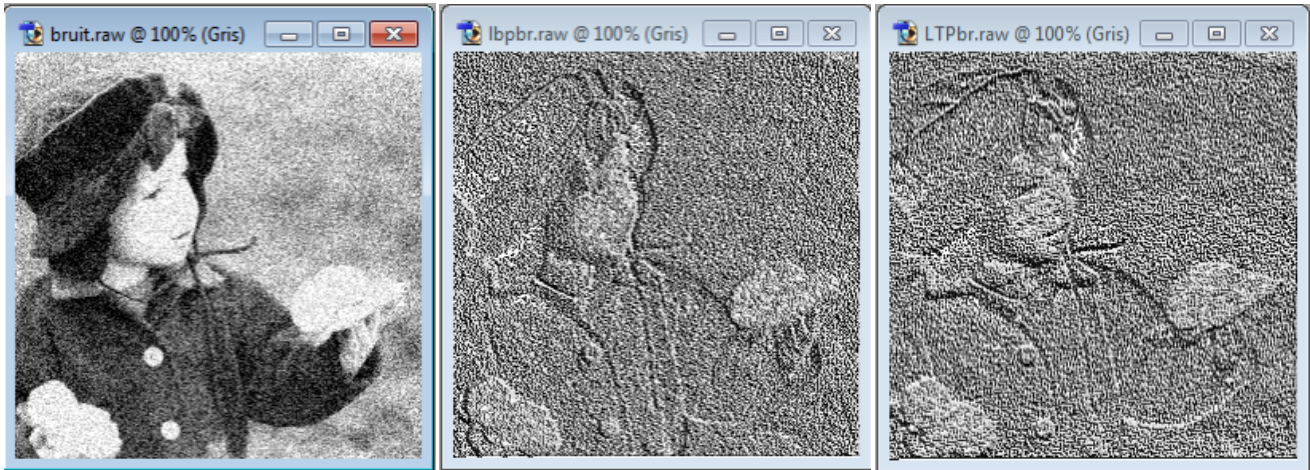
(a)



(b)

Figure IV.7. : (a) : image enfant, (b) : image LTP correspondante.

L'extension LTP est connue par sa résistance au bruit par rapport à LBP. Pour démontrer cela, nous présenterons les résultats obtenus après l'application des deux modèles LBP et LTP sur l'image « enfant sous l'effet d'un bruit ».



(a)

(b)

(c)

Figure IV.8. : (a) : image enfant avec bruit, (b) : image LBP correspondante,
(c) : image LTP correspondante.

D'après l'image (b), nous pouvons constater que la méthode LBP est sensible aux bruits, car elle n'a pas bien fait apparaître les détails de l'image originale. Par contre, nous constatons aisément qu'ils sont mis en évidence dans l'image (c) (surtout les détails du visage). Cela, nous confirme la robustesse de la technique LTP par rapport aux bruits.

IV.4. Conclusion

Dans ce chapitre, nous avons fait une description du langage C++ et l'environnement Code ::Blocks, dont nous avons implémenté nos applications.

Nous avons présenté les différents résultats obtenus en utilisant quelques méthodes et extensions de l'approche LBP afin d'extraire les textures, dans le but de les utiliser dans le processus de classification des images.

Nous avons prouvé la robustesse des modèles LBP contre les variations de la lumière, leur invariance aux rotations des images. Enfin, nous avons démontré l'efficacité des approches LTP par rapport aux bruits en les comparant aux approches LBP.

Conclusion générale

L'objectif du travail exposé dans ce mémoire, a été l'étude des motifs binaires locaux (LBP) appliqués en traitement d'images, pour la caractérisation des textures et la description des régions locales dans les images, afin d'extraire des entités significatives du point de vue thématique et ce, dans le but d'optimiser les phases de segmentation et de classification.

Dans une première partie, nous avons rappelé la notion et les types de la texture dans le cadre d'analyse d'image, puis, nous avons donné un aperçu sur les différentes méthodes permettant l'extraction des attributs texturaux. Notre choix s'est porté sur les méthodes statistiques, qui présentent l'avantage de caractériser tous types de texture. Les matrices de cooccurrence, qui sont très riches en informations texturales, est l'une des approches statistiques que nous avons développé dans cette partie.

Dans la deuxième partie, nous avons exposé l'approche LBP proposée dans cette étude. Nous avons vu la simplicité de calcul des modèles LBP et les améliorations apportées sur le LBP original, tel que le $LBP_{p,R}$ qui est introduit pour la description des textures dans des voisinages circulaires. Cette approche nous a permis de présenter le $LBP_{p,R}^i$ qui évite l'effet de rotation de ces voisinages et le LBP uniforme qui permet d'extraire une grande quantité de texture.

La troisième partie s'est porté sur l'étude des méthodes de segmentation des images, qui est une étape incontournable dans tout processus d'analyse d'image. En fait, nous avons essentiellement mis l'accent sur les approches de classification.

Dans la dernière partie, nous avons tout d'abord introduit le langage C++ qui présente toutes les conformités nécessaires pour l'implémentation de notre logiciel. Appliquées à une image 8 bits, nous avons remarqué que les approches développées, ont donné des résultats très pertinents dans **un premier temps** néanmoins, un travail de réflexion reste encore à faire dans le but de faire une étude plus exhaustive pour mieux appréhender l'utilisation du LBP dans des domaines pluridisciplinaires.

Enfin, nous espérons que nous avons apporté un appui considérable pour les étudiants des promotions futurs afin d'optimiser les processus d'analyse des images à

base des motifs binaires locaux et leurs extensions, nous souhaitons que d'autres études seront faites sur ce sujet, en prenant en considération les différents domaines d'applications des LBP, particulièrement, celui de la reconnaissance faciale et la détection des visages.

Références bibliographiques

- [1] Hamza Chekkaf et Ibrahim Touil, « Segmentation des masses dans les images mammographiques », Mémoire de Master en Informatique, Département d'Informatique, Université Abou Bakr Belkaid de Tlemcen, 2012.
- [2] Medrouk.H et Toucherift.S, « Segmentation par classification d'une partie de l'image », Mémoire d'Ingénieur d'Etat en Automatique, Département d'automatique, Université Mouloud Mammeri de Tizi Ouzou, 2010.
- [3] Belisson Andy Anderson, Le Vigouroux Swann, Mi Sis Luminy, [Http:// En.Wikipedia.Org/Wiki/Texture_Synthesis](http://En.Wikipedia.Org/Wiki/Texture_Synthesis) 2007-2008.
- [4] R.M. Haralick, « Statistical and Structural Approachs to Texture», Proceedings of The IEEE In Proceedings of The IEEE, Vol. 67, Pages 786–804, May 1979.
- [5] Julien Olivier, « Méthodes d'Accélération et Approches Supervisées pour les Contours Actifs. Applications à la Segmentation d'Images 2d, 3d et Texturées », Thèse de Doctorat en Informatique, Université François Rabelais de Tours, 2009.
- [6] Adel Hafiane, « Caractérisation De Texture Et Segmentation Pour La Recherche D'images Par Le Contenu », Thèse de Doctorat en Sciences et Technologies de l'Information des Télécommunications et des Systèmes, Université de Paris-Sud XI, Faculté des Sciences d'Orsay, 2005.
- [7] Li, S. Z, Markov, « Random Field Modeling In Image Analysis », Springer-Verlag New York, 2001.
- [8] Lehamel Malha, « Segmentation d'Images Texturées à Partir des Attributs Fractals », Mémoire de Magister en Automatique Option Traitement d'Images et Reconnaissance de Formes, Département d'Automatique, Université Mouloud Mammeri de Tizi Ouzou, 2011.
- [9] T.Ojala, M.Pietikainen and D.Harwood, « A Comparative Study of Texture Measures with Classification Based on Feature Distributions », Pattern Recognition 29(1):51–59, 1996.
- [10] Ludovic Paulhac, « Outils et Méthodes d'Analyse d'Images 3d Texturées : Application à la Segmentation des Images Echographiques », Thèse de Doctorat en Informatique, Université François Rabelais de Tours, 2009.

- [11] Jonathan Bouchard, « Méthodes de Vision et d'Intelligence Artificielles pour la reconnaissance de Spécimens Coralliens », A l'obtention de la Maîtrise en Génie de la Production Automatisée, Ecole de Technologie Supérieure, Université du Québec, 2011.
- [12] P.Brodatz, « Texture: A Photographic Album for Artists and Designers », New York: Dover, 1966.
- [13] Topi Mäenpää, « The Local Binary Pattern Approach to Texture Analysis – Extensions and Applications », InfoTech Oulu and Department of Electrical and Information Engineering, University of Oulu, Finland, 2003.
- [14] T.Ojala, M.Pietikäinen, T.Mäenpää, « Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns », IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 971–987, 2002.
- [15] T.Ojala, K.Valkealahti and M.Pietikainen, « Texture Discrimination with Multidimensional Distributions of Signed Gray Level Differences», Pattern Recognition 34: 727–739. 2001.
- [16] Thomas Penne, « Développement D'un Système De Tracking Vidéo Sur Caméra Robotisée », Thèse de Doctorat en Informatique, Université Blaise Pascal - Clermont-Ferrand II, 2011.
- [17] X. Tan And B. Triggs, « Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions », in IEEE Transactions on Image Processing, 19(6):1635–1650, 2010.
- [18] S.Hussain, B.Triggs, « Feature Sets and Dimensionality Reduction for Visual Object Detection», In: Proc. British Machine Vision Conference, PP. 112.1–112.10 2010.
- [19] K.Meena, A.Suruliandi, « Local Binary Patterns and Its Variants for Face Recognition », IEEE-International Conference on Recent Trends in Information Technology, Anna University, Chennai, 2011.
- [20] Marko Heikkila, Matti Pietikainen, Cordelia Schmid, « Description Of Interest Regions with Local Binary Pattern », June 2008.
- [21] Arko Lucieer, Alfred Stein and Peter Fisher, « Multivariate Texture Based Segmentation of Remotely Sensed Imagery for Extraction of Objects and Their Uncertainty ».

- [22] Zhenhua Guo, Lei Zhang, David Zhang, « Rotation Invariant Texture Classification Using LBP Variance (LBPV) With Global Matching », *Pattern Recognition* 43 (2010) 706–719.
- [23] H. Jin, Q. Liu, H. Lu and X. Tong, « Face Detection Using Improved LBP Under Bayesian Framework », *International Conference On Image And Graphics*, Hong Kong, China. 306–309, 2004.
- [24] Ehrler Frederic, « Segmentation Interactive Pour La Création d'Images Intelligentes Basées Sur Les Informations Cachées », Département d'Informatique Stochastic Image Processing Group, Université de Genève, 2003.
- [25] Arnaud Capri, « Caractérisation Des Objets Dans Une Image En Vue D'une Aide A L'interprétation Et D'une Compression Adaptée Au Contenu : Application Aux Images Echographiques », Thèse de Doctorat en Sciences et Technologies Industrielles, Université d'Orléans, 2007.
- [26] S.Ait Ali et F.Aiboud, « Segmentation D'images Météosat Par La Méthode Du Maximum De Vraisemblance », Mémoire d'Ingénieur d'Etat en Electronique Option Contrôle, Département d'Electronique, Université Mouloud Mammeri de Tizi Ouzou, 2004.
- [27] Macqueen, J. B. « Some Methods for Classification and Analysis of Multivariate Observations. Berkeley Symposium on Mathematical Statistics and Probability », University Of California Press, Vol.1, PP 281–297, 1967.
- [28] Jérôme Landré, « Analyse Multi résolution Pour La Recherche Et L'indexation D'images Par Le Contenu Dans Les Bases De Données Images - Application A La Base D'images Paléontologique Trans'tyfipal », Thèse de Doctorat en Instrumentation et Informatique de l'image, Université de Bourgogne, 2005.
- [29] Michaël Hauspie, « Programmation C++ », DUT Informatique, Michael.Hauspie@lifl.fr, <http://xkcd.com/371/> 2008-2009.