

**UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS**

**Moments de Zernike pour la recherche d'images par le contenu**

MÉMOIRE PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DU PROGRAMME DE MAÎTRISE EN SCIENCES ET TECHNOLOGIES DE

L'INFORMATION

PAR

**Youssef AKALAL**

Juillet 2019

## DÉDICACE

*A ma femme et mes deux filles,*

*A mes chers parents*

*A ceux qui n'ont jamais cessé de m'encourager et me conseiller.*

*A ceux qui n'ont jamais été avares ni de leur temps ni de leurs connaissances pour satisfaire mes interrogations.*

*A ces éducateurs bienveillants, je dédie le fruit de ma carrière estudiantine.*

*A mes frères et ma sœur.*

*En témoignage de l'amour et de l'affection qui nous lient. A tous mes amis.*

## REMERCIEMENTS

*De prime abord je tiens à remercier très vivement toutes les personnes qui ont participé au bon déroulement de ce mémoire, en particulier ma directrice de recherche à l'UQO, la professeure docteure Nadia BAAZIZ pour son soutien, son bon coaching, l'aide et le soutien qu'elle m'a apportés.*

*Enfin, je formule ma gratitude aux membres du jury et au corps professoral de l'université du Québec en Outaouais.*

## **Jury d'évaluation**

Président du Jury : Dr. Ahmed Lakhssassi

Membre du Jury : Dr. Marek Zaremba

Directrice de recherche : Dr. Nadia BAAZIZ

## Table des matières

<b>DÉDICACE .....</b>	<b>II</b>
<b>REMERCIEMENTS .....</b>	<b>III</b>
<b>JURY D'ÉVALUATION .....</b>	<b>IV</b>
<b>TABLE DES MATIÈRES .....</b>	<b>I</b>
<b>LISTE DES FIGURES .....</b>	<b>IV</b>
<b>LISTE DES TABLEAUX .....</b>	<b>5</b>
<b>LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES.....</b>	<b>6</b>
<b>RÉSUMÉ.....</b>	<b>7</b>
<b>ABSTRACT .....</b>	<b>8</b>
<b>INTRODUCTION.....</b>	<b>9</b>
<b>1 PROBLEME .....</b>	<b>11</b>
<b>2 ETAT DE L'ART .....</b>	<b>12</b>
2.1. L'architecture d'un système de recherche d'images par le contenu .....	12
2.2. Les types de requêtes .....	14
2.2.1. Requête par mots-clés .....	14
2.2.2. Requête par description.....	14
2.2.3. Requête par exemple ou esquisse .....	14
2.2.4. C'est quoi la meilleure méthode de requête ?.....	15
2.3. Les descripteurs d'une image .....	15
2.3.1. La couleur .....	16
2.3.2. La texture .....	18
2.3.3. La forme .....	20
2.3.4. La mesure de similarité .....	21

2.4.	Les mesures de performance d'un système CBIR.....	24
2.5.	Les systèmes de recherche d'images .....	25
2.6.	Les moments de Zernike.....	27
2.6.1.	Les polynômes de Zernike.....	27
2.6.2.	Le calcul des moments .....	31
2.6.3.	L'invariance à la rotation des moments de Zernike .....	32
2.7	Les moments de Zernike et la recherche par le contenu .....	34
2.8	La transformée en contourlettes redondantes.....	36
<b>3</b>	<b>PROPOSITION DE RECHERCHE .....</b>	<b>40</b>
3.1	Objectifs .....	40
3.2	Méthodologie .....	41
<b>4</b>	<b>DÉVELOPPEMENTS .....</b>	<b>42</b>
4.1	Configuration de l'environnement de test .....	45
4.2	Reconstruction d'image à partir des moments de Zernike .....	45
4.3	Extraction des descripteurs de Zernike .....	48
4.4	Recherche d'images par les descripteurs de Zernike .....	52
4.4.1.	Impact de l'ordre du descripteur sur la précision .....	53
4.4.2.	Évaluation de la performance globale du CBIR .....	63
4.4.3.	La normalisation des images préalablement à l'extraction des descripteurs .....	69
4.4.4.	L'incorporation de la transformée en contourlettes redondantes (RCT).....	72
4.4.5.	Descripteur hybride Zernike et RCT .....	74
<b>5</b>	<b>CONCLUSION.....</b>	<b>79</b>
<b>ANNEXE A RECONSTRUCTION D'IMAGE À PARTIR DES MOMENTS DE ZERNIKE.....</b>		<b>80</b>
<b>ANNEXE B TAUX DE REUSSITE DE LA RECHERCHE (RR%) AVEC L'APPLICATION DE LA NORMALISATION AUX IMAGES PRÉALABLEMENT À L'EXTRACTION DES DESCRIPTEURS DE ZERNIKE SUR LA BASE MPEG7 .....</b>		<b>82</b>

<b>ANNEXE C DÉCOMPOSITION MULTI-RÉSOLUTION ET DIRECTIONNELLE RCT (3 NIVEAUX DE RÉOLUTION, 4 DIRECTIONS).....</b>	<b>88</b>
<b>ANNEXE D TAUX DE RESUSSITE (RR%) SUR UN TOP-16 DE LA RCT SUR LA BASE MPEG7 ET VISTEX .....</b>	<b>89</b>
<b>ANNEXE E TAUX DE REUSSITE (RR%) SUR TOP-16 POUR LA COMBINAISON DE ZERNIKE ET RCT SUR LA BASE MPEG7 .....</b>	<b>92</b>
<b>ANNEXE F TAUX DE REUSSITE TOP-16 POUR LA COMBINAISON DE ZERNIKE ET RCT SUR LA BASE VISTEX .....</b>	<b>95</b>

## Liste des figures

Figure 1 : Architecture d'un système de recherche d'images par le contenu.....	13
Figure 2: Classification des descripteurs de couleurs [10].....	17
Figure 3 : L'histogramme RVB d'une image [11]. .....	18
Figure 4 : L'amplitude (Module) de quelque polynômes $V_{nm}(x, y)$ [20], $aV_{1,1}(x, y), bV_{2,2}(x, y), cV_{3,1}(x, y), dV_{4,2}(x, y)$ .....	30
Figure 5 : Schéma bloc de décomposition en contourlettes redondantes d'une image. .....	38
Figure 6 : Partition fréquentielle d'une RCT avec 2 niveaux de résolution et 4 directions fréquentielles. L'approximation de l'image (C3) n'est pas décomposée par le banc de filtres directionnels [3].....	39
Figure 7 : La base de données MPEG7-CE Shape-1 utilisée contient 70 familles d'images. Chaque image ci-dessus fait partie d'une classe distincte contenant 16 images. ....	43
Figure 8 : La base de données VisTex contient 40 familles d'images. Chaque image ci-dessus fait partie d'une classe distincte contenant 16 images.....	44
Figure 9 : Résultat de reconstruction d'images d'objets (binaires) à partir des moments de Zernike à ordres croissants. ....	46
Figure 10 : Résultat de reconstruction d'images de textures à partir des moments de Zernike à ordres croissants.....	47
Figure 11 : Résultat de reconstruction d'une image RGB à partir des moments de Zernike. ....	48
Figure 12 : Descripteur hybride obtenu par concaténation des descripteurs RCT et Zernike. ....	75
Figure 13 : Méthode de recherche hybride exploitant l'union des résultats basés sur chacun descripteurs RCT et Zernike. ....	76

## Liste des tableaux

Tableau 1 : Nombre de polynômes de Zernike. Les valeurs $n$ et $m$ indiquent respectivement l'ordre et l'itération du polynôme de Zernike.....	29
Tableau 2 : Amplitude des Moments de Zernike de l'image <i>apple-1</i> pour un ordre $n=10$ . Le descripteur résultant est de dimension=36.....	50
Tableau 3 : Amplitude de moments de Zernike $Z_{nm}$ pour 4 images appartenant à des classes distinctes.....	51
Tableau 4 : Amplitude des moments de Zernike $Z_{nm}$ pour 5 images de la même classe 'Device3' et différentes paires de valeurs $(n,m)$ .....	52
Tableau 5 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 0. La <i>Précision</i> est de 20%. .....	56
Tableau 6 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 1. La <i>Précision</i> est de 20%. .....	57
Tableau 7 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 5. La <i>Précision</i> est de 100%. .....	58
Tableau 8 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 10. La <i>Précision</i> est de 90%. .....	59
Tableau 9 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 15. La <i>Précision</i> est de 90%. .....	60
Tableau 10 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 20. La <i>Précision</i> est de 90%. .....	61
Tableau 11 : Top-10 images retrouvées pour la requête <i>apple-2</i> et leur distance Euclidienne entre descripteurs de Zernike d'ordre 22. La <i>Précision</i> est de 100%. ...	62
Tableau 12 : Performance du CBIR ( $RR\%$ ) pour une recherche Top-16 effectuée pour les classes d'images MPEG7-CE Shape-1. Quatre ordres $n$ distincts sont utilisés pour les descripteurs de Zernike. ....	66
Tableau 13 : Performance du CBIR ( $RR\%$ ) pour une recherche Top-16 effectuée pour les 40 classes d'images VisTex. Six ordres $n$ sont utilisés pour les descripteurs de Zernike. ....	69
Tableau 14 : Étapes du processus de normalisation des images. ....	70
Tableau 15 : Taux de réussite de la recherche ( $RR\%$ ) en appliquant la normalisation pour quelques images de la base de données MPEG7-CE Shape-1. ....	71
Tableau 16: Tableau comparatif de la performance de systèmes CBIR basés sur 4 méthodes distinctes, Zernike, RCT, Concaténation Zernike+RCT et Union Zernike-RCT. La recherche est effectuée sur les bases de données MPEG7-CE Shape-1 et VisTex. Les taux de réussite ( $RR\%$ ) correspondent à trois ordre $n$ distincts 5, 10 et 15.....	77

## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

CBIR : Content Based Image Retrieval

RIC : Recherche d'images par le contenu

RGB : Red, Green and Blue

SGBD : Système de gestion de base de données.

ZMs : Zernike Moments. Moments de Zernike

DWT : Discrete Wavelet Transform. Transformée en ondelettes discrète.

SWT : Stationary Wavelet Transform. Transformée en ondelettes stationnaires.

CT : Contourlet Transform. Transformée en contourlettes.

RCT : Redundant Contourlet Transform. Transformée en contourlettes redondantes.

## Résumé

Le traitement d'images numériques, la production et le stockage de données numériques à grandes échelles connaissent de nos jours une véritable expansion dans divers domaines d'applications. De ce fait, une gestion efficace de grandes bases de données d'images est requise et passe par le recours à des méthodes d'indexation automatiques et innovantes. Une des approches adoptées, à cette fin, est la recherche d'images par le contenu ou Content-Based Image Retrieval (CBIR) qui vise l'exploitation de caractéristiques visuelles de l'image plutôt que ses caractéristiques sémantiques pour l'indexation et l'interrogation de la base de données d'images.

Dans ce mémoire de recherche, nous proposons d'explorer l'utilisation des moments de Zernike en conjonction avec des représentations d'images directionnelles, telles que la transformée en contourlettes, pour extraire des caractéristiques visuelles d'image et construire des descripteurs pertinents en vue de la recherche d'images par le contenu. Les images de textures seront particulièrement considérées. De plus, l'apport d'une telle approche à la performance d'un système CBIR permettra la validation de la pertinence des descripteurs extraits. La méthodologie que nous proposons est motivée par le fait que la transformée en contourlettes permet l'extraction de primitives de textures et que les moments de Zernike sont connus pour leur pouvoir à caractériser les formes d'objets dans une image de manière globale. Comme preuve de concept, les résultats expérimentaux montrent la reconstruction d'images à partir des moments de Zernike et évalue la précision et le taux de réussite de la recherche d'un système CBIR prototype en utilisant des bases de données d'images de test connues.

## **Abstract**

Digital image processing, the production and storage of large-scale digital data are now experiencing a real expansion in various fields of application. As a result, efficient management of large image databases is required and requires the use of automatic and innovative indexing methods. One of the approaches adopted, for this purpose, is Content-Based Image Retrieval (CBIR), which aims to exploit visual features of the image rather than its semantic features for indexing and querying the image database.

In this research paper, we propose to explore the use of Zernike moments in conjunction with representations of directional images, such as the RCT (Redundant Contourlet Transform), to extract visual image features and construct relevant descriptors for image search by content. The images of textures will be particularly considered. Moreover, the contribution of such an approach to the performance of a CBIR system will allow the validation of the relevance of the extracted descriptors. The methodology we propose is motivated by the fact that the RCT allows the extraction of texture primitives and that the Zernike moments are known for their ability to characterize the object shapes in an image in a global way. As proof of concept, the experimental results show the reconstruction of images from the Zernike moments and evaluate the accuracy and success rate of our CBIR prototype system using databases of known test images.

## INTRODUCTION

Dans les deux dernières décennies les domaines de l'image numérique, de stockage de données et d'internet ont connu une vraie expansion. On peut maintenant prendre des centaines d'images de haute résolution en utilisant seulement son téléphone portable et les partager sur des réseaux sociaux en deux clics. Sur Facebook par exemple le nombre d'images partagé quotidiennement sur ce réseau social atteint les 300 millions [1], cela crée des bases de données gigantesques que ce soit pour des particuliers mais aussi pour les professionnels (musée, agence web, sécurité, etc..).

Pour gérer et retrouver l'information sur ces bases de données les premiers systèmes ont proposé la technique de recherches d'images par mots clés ou tags dans les années 1970 [2] mais cette technique s'est avérée limitée et subjective car elle dépend en grande partie de la langue et de la compréhension de la personne chargée de classer et indexer les images [3].

Le besoin d'avoir une méthode d'indexation indépendante et automatique a poussé les chercheurs à se pencher sur les caractéristiques visuelles de l'image plutôt que les caractéristiques sémantiques et cela a donné naissance dans le début des années 1990 [4] à une deuxième génération de méthodes désignées par CBIR [5].

La recherche d'images par le contenu (CBIR: *Content Based Image Retrieval* ou RIC en français) consiste comme son nom l'indique à rechercher une image ou plusieurs sur une base de données qui est préalablement indexée en se basant sur des descripteurs du contenu des images (tests et requête) dits bas niveau comme la couleur, la texture ou la forme [3].

Dans ce projet de recherche, nous nous intéressons à la recherche d'images par le contenu, les différents descripteurs et les comparateurs utilisés dans ce domaine pour calculer la similarité entre les descripteurs d'images. Dans la première section, nous définissons le problème qui sera traité dans le cadre de ce mémoire de maîtrise. La deuxième section synthétise les différents travaux de la littérature effectués ces dernières années dans le domaine de la recherche par le contenu et les moments de Zernike. L'architecture d'un système de recherche d'images par le contenu y est présentée ainsi que les différents types de descripteurs utilisés pour l'extraction de caractéristiques pertinentes dans les images ainsi que les techniques usuelles de mesure de similarité entre descripteurs. Dans la troisième section, nous définissons les objectifs de recherche poursuivis et nous proposons la méthodologie de travail adoptée. Les développements effectués et les résultats d'expérimentation atteints sont discutés dans la section 5. Finalement, la section 6 résume le travail réalisé et présente les perspectives futures.

## **1 PROBLEME**

Dans ce projet, nous explorons l'utilisation des moments de Zernike en conjonction avec des représentations d'images directionnelles, telles que la transformée en contourlettes, pour extraire des descripteurs d'images pertinents. Les images de textures sont particulièrement considérées. De plus, l'apport d'une telle approche à la performance d'un système de recherche d'images par le contenu (CBIR) permet la validation de la pertinence des descripteurs extraits. L'approche que nous avons étudiée est motivée par le fait que la transformée en contourlettes permet l'extraction de primitives de textures et que les moments de Zernike sont connus pour leur pouvoir à caractériser les formes d'objets dans une image de manière globale. Un bon descripteur de forme répond à trois exigences : 1) le descripteur doit être capable de caractériser le mieux possible les informations des formes ; 2) la taille des vecteurs ne doit pas être trop grande, et 3) le temps de calcul de similarité doit être le plus court possible.

## **2 ETAT DE L'ART**

La recherche d'images par le contenu (ou CBIR) est née du besoin de remplacer les systèmes d'indexation et de recherche d'images par sémantique ou mots-clés. Cette méthode efficace d'association de sens à une image en utilisant un descripteur textuel est cependant subjectif et dépend de la compréhension et de la langue de l'iconographe qui associe manuellement sa perception de l'image à une catégorie ou un thème. En effet la probabilité que deux personnes recherchent une image avec le même terme est moins de 20% et lorsqu'on utilise un thésaurus contraint avec une liste de mots acceptés la probabilité reste inférieure à 70% [6]. Ajoutant à cela la difficulté voire l'impossibilité d'indexer manuellement les images quand la base de données est massive. Cela a poussé les chercheurs à aller au-delà de la recherche sémantique et à explorer les caractéristiques visuelles représentées dans les données de l'image.

Dans ce chapitre, nous décrivons les composantes essentielles d'un système de recherche d'images par le contenu.

### **2.1. L'architecture d'un système de recherche d'images par le contenu**

Le système comme le représente la Figure 1 se compose de trois parties :

1. L'interface utilisateur, qui permet à l'utilisateur de rentrer son image requête ou ses images (Il y a des systèmes qui peuvent rechercher une combinaison d'images requêtes en même temps) puis affiche le résultat de la comparaison avec les images indexées de la base de données. En général, l'affichage du résultat se fait avec un ordre décroissant du taux de similarité calculée. Ce module fonctionne en mode en ligne.

2. La deuxième partie est l'indexation des images de la base de données en mode hors-ligne pour éviter lors d'une recherche sur la base par un utilisateur la charge élevée de calcul et temps lors de l'analyse et l'extraction des descripteurs visuels.
3. La dernière partie correspond au traitement en ligne de la requête. Les descripteurs sont extraits de l'image requête, le calcul de similarité par rapport à la base des index est effectué et les résultats triés par ordre de pertinence sont renvoyés.

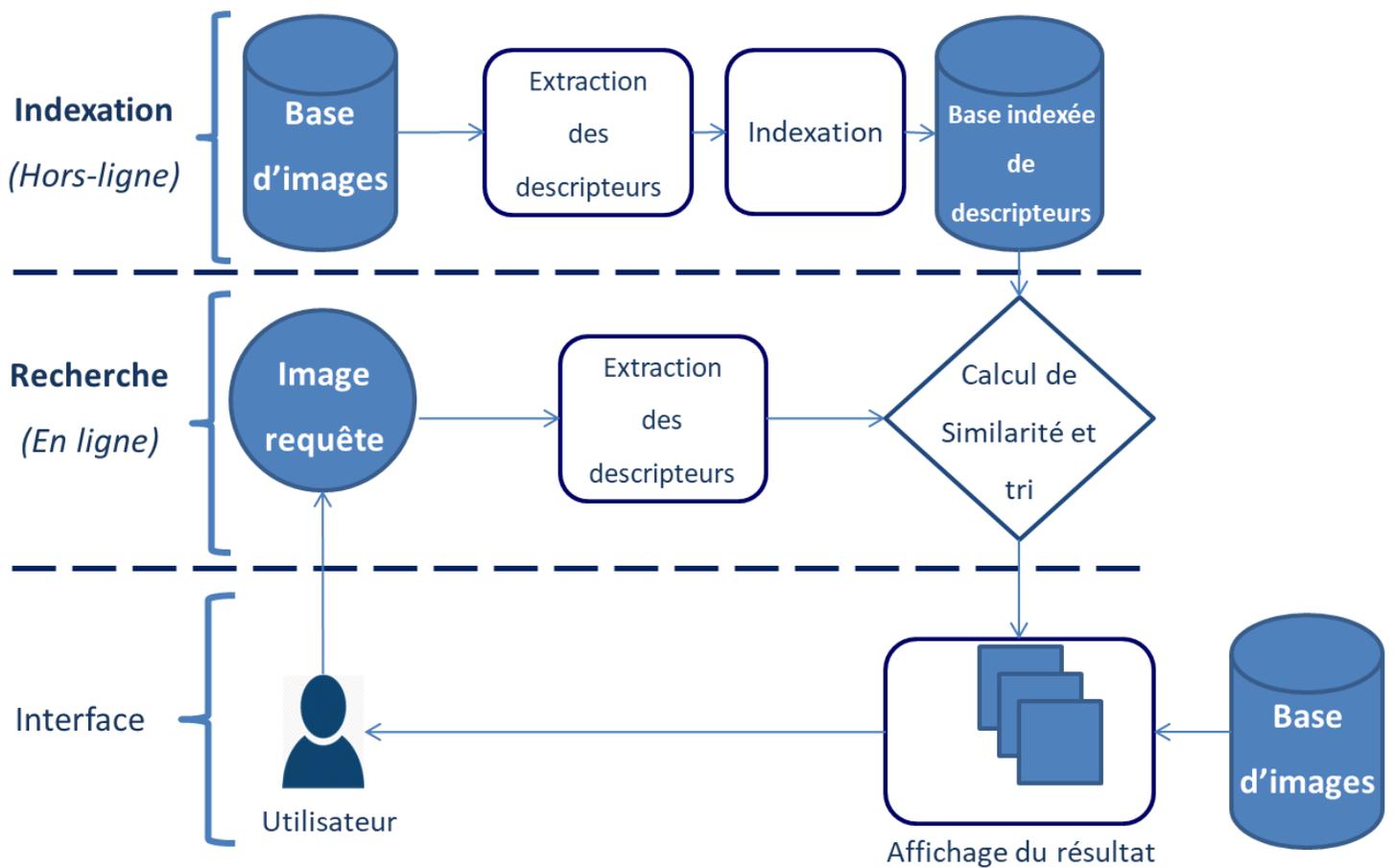


Figure 1 : Architecture d'un système de recherche d'images par le contenu.

## **2.2. Les types de requêtes**

Pour exprimer son besoin au système de recherche d'images et retrouver les images qu'on veut, on dispose de trois types de requêtes ; la requête par mots clés, par esquisse ou exemple ou bien la requête par caractéristiques. Il y a aussi des systèmes qui proposent la combinaison de plusieurs types de requêtes [3].

Dans la section suivante, les différents types de requêtes sont parcourus pour montrer leurs avantages et leurs inconvénients.

### **2.2.1. Requête par mots-clés**

Le système se base sur l'annotation manuelle des images de la base de données préalablement faite pour retrouver les images qui correspondent le plus possible aux mots-clés recherchés par l'utilisateur. Cependant une image peut contenir un grand nombre d'informations, et il sera donc difficile de les représenter en quelques mots-clés [7].

### **2.2.2. Requête par description**

Cette méthode consiste à décrire des caractéristiques visuelles qu'on recherche dans une image. L'utilisateur dispose d'une interface pour spécifier des caractéristiques descriptives de son image recherchée. Par exemple il peut rechercher les images dont le quadrant gauche contient 30% de pixels rouges [7].

### **2.2.3. Requête par exemple ou esquisse**

Dans ce cas l'utilisateur fournit au système une image ou choisit dans la même base de données une image exemple qui sera utilisée pour trouver les images similaires à l'image requête. Il peut aussi disposer d'une interface pour dessiner un croquis qui représentera le mieux possible les images recherchées sur la base. La méthode de l'esquisse ou croquis

reste très limitée car il est parfois difficile de représenter notre besoin sous forme d'un dessin [3].

#### **2.2.4. C'est quoi la meilleure méthode de requête ?**

Pour un utilisateur normal, la recherche par mots-clés ou par un exemple restent probablement les plus naturelles. C'est plus facile de donner un exemple d'image ou d'une partie d'image ou bien écrire des mots clés qui peuvent être associés aux images recherchées.

### **2.3. Les descripteurs d'une image**

L'extraction de descripteurs à partir d'une image et leur représentation sous forme d'un vecteur de caractéristiques sont au cœur d'un système CBIR. L'indexation d'une image se fait en deux phases. La première phase dite logique consiste à extraire des attributs (couleur, texture, forme, ...) qui caractérisent le mieux possible l'image. Ainsi, durant la recherche, l'analyse va se limiter à ces descripteurs. Dans la deuxième phase qu'on appelle indexation physique, on détermine la structure d'accès aux données la plus efficace pour mieux organiser la base de données et optimiser la recherche au maximum (Parmi les différentes structures d'indexation on retrouve (arbres-B, les arbres K-d-tree, les arbres quadrants, le clustering) [8], [9].

Dans ce projet, nous nous intéressons à l'indexation logique. Dans ce qui va suivre, nous décrivons les types de descripteurs d'images les plus populaires dans le contexte de la recherche d'images par le contenu.

### 2.3.1. La couleur

La couleur joue un grand rôle dans le mécanisme de perception visuel humain. Vu son invariance à la taille, à l'orientation et à la variation d'échelle, les descripteurs basés sur la couleur figurent parmi les plus populaires et largement utilisés pour la caractérisation visuelle d'images [3]. Il y a deux aspects déterminant dans la structuration de ces descripteurs ; l'espace de représentation des couleurs et le modèle de caractérisation des données de couleur.

#### a) L'espace de représentation des couleurs

Un espace de couleur est une représentation des couleurs dans un système de synthèse des couleurs compte tenu des limites de la vision humaine. Ces espaces sont généralement à 3 ou à 4 dimensions. Chaque couleur de lumière peut donc être caractérisée par un point (triplet) dans un espace à trois dimensions ou bien par un point (quadruplet) dans un espace à quatre dimensions. L'espace de couleur le plus utilisé reste le RVB (ou *RGB* : *Red, Green, Blue*) car il s'est avéré efficace pour la recherche d'images [3] mais d'autres systèmes tel que HSV (*Hue, Saturation, Value*), CMY (*Cyan, Magenta, Yellow*), CMYK (*Cyan, Magenta, Yellow, Black*) sont aussi utilisés [10].

#### b) Les modèles de caractérisation de la couleur

La Figure 2 représente les deux catégories de modèles utilisés pour caractériser les attributs des données de l'image relatives à la couleur ; le modèle de l'histogramme et le modèle statistique [10]. L'histogramme associe à chaque valeur d'intensité lumineuse le nombre d'échantillons (pixels) ayant cette valeur. La Figure 3 montre un exemple d'une image et ses différents histogrammes associés à chacune des composantes couleurs R, V et B. Les histogrammes d'images sont faciles à calculer mais présentent trois inconvénients majeurs [7]. Premièrement l'histogramme ne nous fournit pas l'emplacement des objets (l'information spatiale) du contenu dans une image. Deuxièmement, L'histogramme est sensible aux changements de luminosité, contraste et

artefacts de compression. Puis finalement le calcul de la distribution complète des couleurs engendre la construction de grands vecteurs ce qui complique l'indexation.

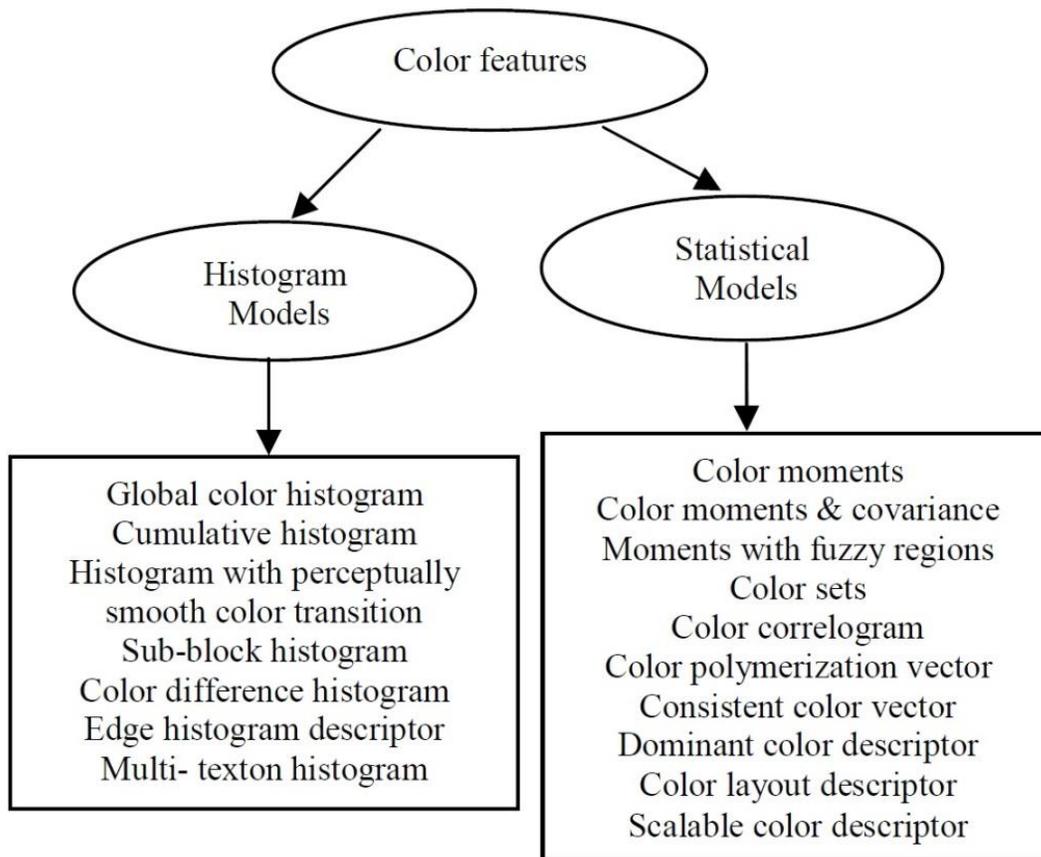


Figure 2: Classification des descripteurs de couleurs [10].

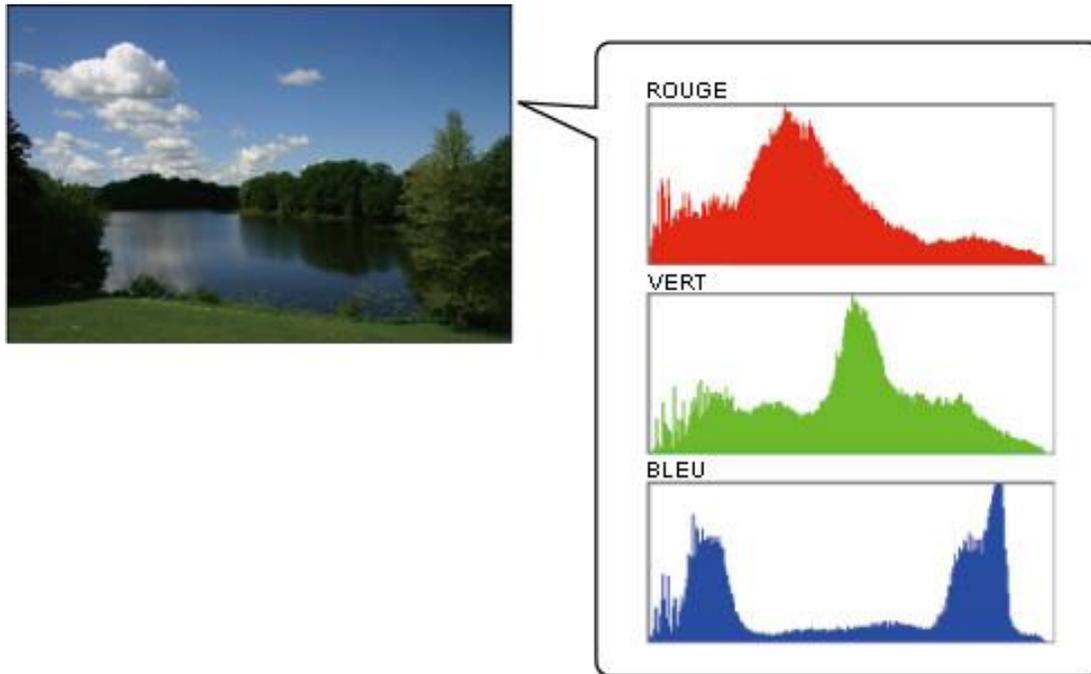


Figure 3 : L'histogramme RVB d'une image [11].

Afin de remédier à ces inconvénients et simplifier le processus d'indexation d'images basée sur la couleur, on a eu recours aux moments statistiques [2]. Ainsi, la distribution des couleurs dans l'image est caractérisée par quelques moments dominants comme l'espérance et la variance [2].

### 2.3.2. La texture

On retrouve la texture dans presque chaque surface naturelle (feuilles des arbres, bois) ou artificielle (tissu, murs en brique). Elle est d'ailleurs utilisée pour compléter le descripteur de couleur, si deux régions ont la même couleur ou une distribution très proche, on peut recourir au calcul de la similarité de leur texture pour les distinguer. La texture est le motif visuel homogène qui représente l'agencement structurel de la couleur ou l'intensité

dans une image [2]. Elle dépend de la distribution des intensités dans l'image. En effet, le contraste et la netteté sont des paramètres utilisés pour l'analyse de la texture dans une image [10]. De nombreuses méthodes ont été proposées par les chercheurs, dont ces deux approches bien connues : l'approche statistique et l'approche structurale.

#### ***a) L'approche statistique***

Selon N. Idrissi [12], cette approche traite la texture comme un processus stochastique déterministe. Elle modélise les notions qualitatives usuelles de texture (granularité, contraste, homogénéité, répétitivité, fragmentation, orientation, etc.).

La granularité est le motif dominant de la texture et parfois le synonyme de texture. Le grain est constitué de pixels voisins qui possèdent le même niveau de gris. La taille et la densité des grains déterminent le niveau de finesse de la texture. Le contraste est basé sur le nombre de niveaux de gris et leur taux de variation. Le changement du contraste entraîne une modification dans la qualité de l'image mais pas dans sa structure. L'orientation est une propriété globale pour une région et traduit la direction générale prise par les motifs ou grains d'une texture [12].

#### ***b) L'approche structurale***

L'approche structurale est particulièrement bien adaptée aux macros textures car elle tient compte de l'information structurelle et contextuelle de la texture [12]. Les étapes d'analyse sont d'abord l'identification des éléments primitifs puis la caractérisation des règles de déplacement entre ces primitives [12]. Autrement dit, on effectue une croissance de région sur un ou plusieurs attributs (intensité lumineuse, gradient, etc.) pour identifier les primitives qu'on caractérise par leur forme et la valeur moyenne de l'attribut utilisé. Puis on calcule les histogrammes du premier ou du second ordre de ces paramètres pour caractériser leur répartition dans l'image. Les deux structures les plus importantes sont les structures de graphe et les structures syntaxiques [12].

### 2.3.3. La forme

La forme est un attribut très important lorsqu'on veut analyser le contenu d'une image. Les premiers algorithmes ont été développés vers les années 1970 [13] mais le domaine de recherche par la forme est devenu très dynamique ces dernières années. On peut classer les descripteurs de forme en deux grandes catégories basées sur les contours ou bien sur les régions respectivement. Dans, l'approche basée sur les contours, les formes sont décomposées en segments limités par leur contour qu'on appelle les primitives [10]. Pour le calcul des contours, les méthodes les plus utilisées sont les approximations polygonales, décomposition selon la courbure ou bien l'ajustement de courbe [10]. Dans l'approche basée sur les régions, la caractérisation porte sur l'intégralité de la forme d'une région. Tous les pixels d'une région de forme sont pris en compte pour la description de la forme. Cette approche utilise les moments invariants (comme les moments de Hu, les moments de Zernike, etc....) [10].

La différence entre les deux approches est que l'approche basée sur les contours est facilement calculée puisque seuls les contours sont pris en compte. Par contre, elle ne peut pas décrire des objets qui ont des trous ou des objets complexes qui sont constitués de multiples régions déconnectées [14]. La deuxième méthode basée sur les régions prend en compte et la région et son contour. C'est pourquoi elle nécessite plus de calcul mais elle est plus robuste face aux bruits et aux distorsions de l'image [14]. Dans ce qui suit, trois méthodes de description de la forme sont présentées.

#### *a) Les descripteurs de Fourier*

Le descripteur de Fourier est un des plus populaires descripteurs de forme vu sa performance et sa simplicité [15]. Le principe est de présenter le contour de l'objet par un signal 1D puis de le décomposer en séries de Fourier [15].

***b) Les moments géométriques***

Les moments géométriques se basent sur les propriétés statistiques pour décrire une forme. Ils représentent les propriétés spatiales de la distribution des pixels dans l'image [15]. Ils sont simples à calculer, invariants à la rotation mais sensibles aux bruits et ont une redondance d'informations puisque la base n'est pas orthogonale. Les invariants de Hu restent les moments géométriques les plus utilisés dans la littérature à cause de leur invariance aux rotations, translations et changement d'échelle [14].

***c) Les moments orthogonaux***

Ces moments ont l'avantage d'être définis dans une base orthogonale ce qui conduit à une réduction au niveau de la redondance d'informations en comparaison avec les moments géométriques [14]. Cependant, la complexité de calcul est plus élevée. Parmi les moments orthogonaux les plus utilisés, on trouve les moments de Zernike et les moments de Legendre [15].

**2.3.4. La mesure de similarité**

Dans un CBIR, le résultat d'une recherche n'est pas une seule image qui ressemble exactement à ce qu'on recherche mais un ensemble de plusieurs images qui ont des similarités avec l'image requête. La mesure de similarité est souvent opérée par un calcul de distance entre le descripteur de l'image requête et celui de l'image de la base de données pour déterminer combien ils sont similaires ou différents. Une distance nulle signifie que les deux descripteurs sont similaires. Plusieurs méthodes permettent de faire cette mesure mais on va citer quelques-unes des plus utilisées et proposées dans la littérature du domaine.

### ***La distance de Minkowski***

La distance classique (*Minkowski*) est une distance basée sur la norme  $L_p$  qui se définit comme suit :

$$L_p(A, B) = \left( \sum_{i=0}^{N-1} |a_i - b_i|^p \right)^{\frac{1}{p}} \quad (2.1)$$

Où  $p$  est le paramètre de l'ordre,  $N$  la dimension de l'espace et  $A, B$  deux vecteurs à comparer. Pour  $p=1$ , on obtient la distance de norme-1 ( $L_{Man}$ ), connue comme la distance de *Manhattan* ou *City-block*.

$$L_{Man}(A, B) = \sum_{i=0}^{N-1} |a_i - b_i| \quad (2.2)$$

Pour  $p=2$ , on aura la distance *Euclidienne*  $L_{Euc}$ .

$$L_{Euc}(A, B) = \sqrt{\sum_{i=0}^{N-1} (a_i - b_i)^2} \quad (2.3)$$

Quand ( $p \rightarrow \infty$ ), on obtient la distance maximum  $L_{\infty}$  appelée aussi la distance de *Chebyshev*.

$$L_{\infty}(A, B) = \max_{0 \leq i \leq N-1} \{|a_i - b_i|\} \quad (2.4)$$

Les distances de *Minkowski* sont simples à implémenter et rapides à calculer. Cependant, dans leur calcul, on considère que chaque composante du vecteur apporte la même contribution à la distance.

### ***La distance quadratique***

Alors que les distances de *Minkowski* traitent les composantes des vecteurs une par une, la distance quadratique privilégie les éléments les plus ressemblants [15]. Elle est donc définie comme suit :

$$D_{QF}(A, B) = \sqrt{(A - B).S.(A - B)^T} \quad (2.5)$$

Où  $S = [S_{ij}]$  est la matrice de similarité,  $S_{ij}$  décrit la similarité (distance basique) entre les éléments  $a_i$  et  $b_i$ .

$$S_{ij} = 1 - \frac{d_{ij}}{d_{max}} \quad (2.6)$$

Où  $d_{ij}$  est la distance entre  $a_i$  ( $i^{\text{ème}}$  élément de  $A$ ) et  $b_i$  ( $i^{\text{ème}}$  élément de  $B$ ),  $d_{max}$  est le maximum des  $d_{ij}$ .

### ***Distance de Mahalanobis***

Cette distance prend en considération la covariance des variables dans le calcul et reste indépendante de l'échelle de données. Elle est donc définie comme suit :

$$d_{Mah} = \sqrt{(A - B).C^{-1}.(A - B)^T} \quad (2.7)$$

Où  $C^{-1}$  est la matrice de covariance.

### ***Intersection d'histogrammes***

Cette mesure, proposée par *Swain* et *Ballard*, est l'une des premières distances utilisées dans les CBIR même si elle n'est pas invariante aux changements d'illuminations [15]. Elle permet d'évaluer le recouvrement de deux histogrammes normalisés  $H^1$  et  $H^2$  et s'exprime comme suit :

$$d(H^1, H^2) = 1 - \frac{\sum_{j=1}^n \min(H_j^1, H_j^2)}{\sum_{j=1}^n H_j^2} \quad (2.8)$$

Où  $n$  est le nombre de valeurs de chaque histogramme.

#### 2.4. Les mesures de performance d'un système CBIR

La recherche d'images par le contenu a repris les mesures d'évaluation utilisées dans le domaine de la recherche d'information. Les mesures de performance *Rappel et Précision* sont probablement les plus utilisées dans la littérature. La paire permet d'évaluer la pertinence d'un système de recherche d'information en mesurant la concordance des informations retournées par rapport à la requête.

Dans un CBIR, la *Précision* est mesurée par le ratio entre le nombre d'images pertinentes (à la requête) retournées par le système et le nombre total d'images retournées par le système pour une requête donnée.

$$Précision = \frac{R_a}{A} \in [0,1] \quad (2.9)$$

Avec  $R_a$ : Le nombre d'images pertinentes retournées par le système

$A$  : Le nombre total d'images retournées par le système

Le *Rappel* quant à lui est calculé en divisant le nombre d'images pertinentes retournées par le système sur le nombre d'images pertinentes disponibles dans la base de données. Cette mesure requiert la vérité terrain (connaissance de toutes les images pertinentes à la requête et disponibles dans la base de données).

$$Rappel = \frac{R_a}{R} \in [0,1] \quad (2.10)$$

Avec  $R_a$  : Le nombre d'images pertinentes retournées par le système

$R$  : Le nombre total d'images pertinente dans la base de données

Une autre mesure de performance dérivée du *Rappel* est désignée par le «*Retrieval Rate (RR%)*» ou bien par le *Taux de Réussite/Succès de la recherche (RR%)*. On lance  $M$  requêtes et le système retourne, pour chacune, les  $N$  meilleures images (*Top-N*) trouvées. Les valeurs de *Rappel* pour chaque requête sont alors moyennées et exprimées en pourcentage pour donner la mesure *RR%* (pour une recherche *Top-N*). Pour le choix de la valeur  $M$ , il est d'usage de lancer autant de requêtes que possible pourvu que la vérité terrain soit disponible (connaissance de toutes les images pertinentes à chaque requête).

## 2.5. Les systèmes de recherche d'images

Dans les dernières années, le besoin d'organiser et de rechercher des images dans des grandes bases de données a donné le jour à plusieurs systèmes dont ceux présentés ci-dessous.

- **QBIC (Query By Image Content)** : est le premier CBIR commercial. Il a été développé par IBM et supporte plusieurs types de requêtes (exemple, croquis, mots-clés...). Le système combine trois descripteurs d'images (le contour, la couleur et la texture) mais n'est plus disponible sur le marché et on n'a pas pu avoir aucun lien pour le télécharger ou l'essayer. [16]
- **Photobook** : est un système d'indexation d'images développé par MIT (*Massachusetts Institute of Technology, USA*). Le système requiert une image exemple et se base sur la couleur, la texture et la forme pour effectuer sa recherche. On peut télécharger gratuitement une copie de Photobook depuis un

serveur FTP du MIT Media Lab en utilisant le lien suivant (<http://vismod.media.mit.edu/pub/photobook/>)

- **Virage** : a été développé par Virage Inc. Il est similaire à QBIC sauf qu'il offre à l'utilisateur le choix de pondérer sa recherche en ce qui concerne les descripteurs (couleur, forme, texture). Comme QBIC on n'est pas arrivée à avoir un lien pour le télécharger ou l'acheter. [17]

## 2.6. Les moments de Zernike

Les moments de Zernike sont utilisés pour capturer les détails globaux d'une image. D'ailleurs en raison de leur efficacité, ils sont considérés comme le meilleur choix pour extraire un descripteur global (basé sur la région) d'une image [18].

Les moments de Zernike ont les propriétés suivantes : le module (amplitude) des moments de Zernike est invariant à la rotation ; ils sont robustes au bruit et aux légères variations des formes ; il n'y a pas de redondance d'information car leurs bases sont orthogonales. Les moments de Zernike d'ordre faible représentent la forme globale, alors que les ordres plus élevés représentent les détails de la forme caractérisée.

### 2.6.1. Les polynômes de Zernike

Les moments de Zernike se basent sur les polynômes de Zernike qui sont un ensemble de polynômes complexes orthogonaux dans le cercle unité  $x^2 + y^2 = 1$  [19]. Les polynômes de Zernike d'ordre  $n$  et de répétition  $m$  sont définis en coordonnées polaires ( $x = r \cos(\theta)$ ,  $y = r \sin(\theta)$ ) comme suit :

$$V_{nm}(x, y) = V_{nm}(r, \theta) = R_{nm}(r) \exp(jm\theta) \quad (2.11)$$

Où :

- $n \in Z^+$  (ordre du polynôme)
- $m \in Z$  (répétition), doit vérifier les deux conditions suivantes :  
 $n - |m| = \text{pair}$  et  $|m| \leq n$
- $r$  est la longueur du vecteur entre l'origine et les coordonnées du point  $(x, y)$  :  $r = \sqrt{(x^2 + y^2)}$

- $\theta = \angle(r, x) = \arctan\left(\frac{y}{x}\right)$  l'angle d'Azimuth entre le vecteur  $r$  et l'axe des  $x$  dans le sens anti-horaire

$R_{nm}(r)$  représente le polynôme radial de Zernike, de degré  $n$  et à valeurs réelles. Il est défini comme suit :

$$R_{nm}(r) = \sum_{s=0}^{\frac{(n-|m|)}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} r^{n-2s} \quad (2.12)$$

Avec  $R_{n,-m}(r) = R_{n,m}(r)$

Grâce à la condition  $n - |m| = \text{pair}$  et  $|m| \leq n$  de l'équation ( 2.11 ), l'ensemble des polynômes de Zernike contient  $\frac{1}{2}(n+1)(n+2)$  polynômes linéairement indépendants si le degré maximal donné est  $n$ . Par exemple si le degré maximal est  $n = 5$ , la base orthogonale sera formée de 21 polynômes.

Ordre $n$	Répétition $m$	Nb de polynômes	Polynômes d'ordre $n$ et répétition $ m $
0	0	1	$V_{00}$
1	$\pm 1$	2	$V_{11}$
2	0, $\pm 2$	3	$V_{20}, V_{22}$
3	$\pm 1, \pm 3$	4	$V_{31}, V_{33}$
4	0, $\pm 2, \pm 4$	5	$V_{40}, V_{42}, V_{44}$
5	$\pm 1, \pm 3, \pm 5$	6	$V_{51}, V_{53}, V_{55}$
6	0, $\pm 2, \pm 4, \pm 6$	7	$V_{60}, V_{62}, V_{64}, V_{66}$
7	$\pm 1, \pm 3, \pm 5, \pm 7$	8	$V_{71}, V_{73}, V_{75}, V_{77}$
8	0, $\pm 2, \pm 4, \pm 6, \pm 8$	9	$V_{80}, V_{82}, V_{84}, V_{86}, V_{88}$
9	$\pm 1, \pm 3, \pm 5, \pm 7, \pm 9$	10	$V_{91}, V_{93}, V_{95}, V_{97}, V_{99}$
10	0, $\pm 2, \pm 4, \pm 6, \pm 8, \pm 10$	11	$V_{100}, V_{102}, V_{104}, V_{106}, V_{108}, V_{1010}$

Tableau 1 : Nombre de polynômes de Zernike. Les valeurs  $n$  et  $m$  indiquent respectivement l'ordre et l'itération du polynôme de Zernike.

La précision lors du calcul des polynômes de Zernike est affectée par deux paramètres ; l'ordre  $n$  et la répétition  $m$ . Plus l'ordre  $n$  est élevé meilleure est la précision du calcul. En outre, plus la répétition  $m$  est élevée plus on a de points caractéristiques d'image [20].

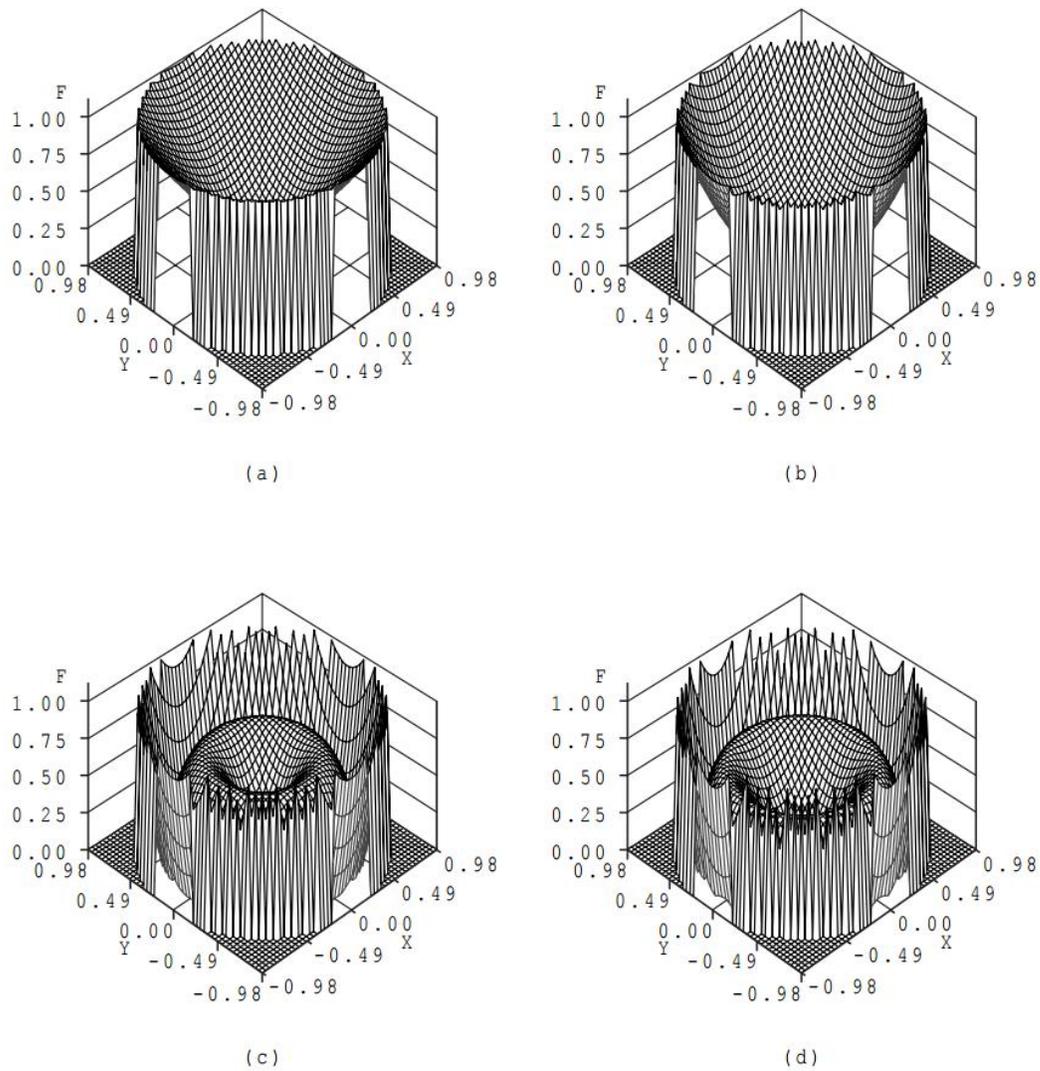


Figure 4 : L'amplitude (Module) de quelque polynômes  $V_{nm}(x, y)$  [20],  
 (a)  $|V_{1,1}(x, y)|$ , (b)  $|V_{2,2}(x, y)|$ , (c)  $|V_{3,1}(x, y)|$ , (d)  $|V_{4,2}(x, y)|$ .

Comme on peut le constater sur la Figure 4, le contour est un seul cercle unité. Si on augmente l'ordre  $n$  et la répétition  $m$  ce n'est pas seulement un lobe principal qu'on a mais plusieurs lobes secondaires. Cela veut dire qu'on obtient plus de points caractéristiques en augmentant  $n$  et  $m$ .

### 2.6.2. Le calcul des moments

Les moments de Zernike d'ordre  $n$  et de répétition  $m$  d'un signal bidimensionnel  $f(x, y)$  défini sur le cercle unité ( $x^2 + y^2 = 1$ ), sont les projections de  $f$  sur la base orthogonale formée par les polynômes  $V_{nm}(x, y)$  comme suit :

$$A_{nm} = \frac{n+1}{\pi} \int \int_{x^2+y^2 \leq 1} f(x, y) V_{nm}^*(x, y) dx dy \quad (2.3)$$

Ou bien, en coordonnées polaires ( $x = r \cos(\theta)$ ,  $y = r \sin(\theta)$ ), comme suit :

$$A_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta) R_{nm}(r) \exp(-jm\theta) r dr d\theta \quad (2.4)$$

Il est à noter que les valeurs sont  $A_{nm}$  complexes. Dans le cas d'un signal 2-D discret, les intégrales dans (3.4) sont remplacées par des sommations :

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V_{nm}^*(x, y), \quad x^2 + y^2 \leq 1 \quad (2.5)$$

Inversement, la reconstruction de  $f(x, y)$  à partir de ses moments de Zernike, s'exprime comme suit :

$$f'(x, y) = \sum_n \sum_m A_{nm} V_{nm}(x, y) \quad (2.6)$$

Pour une image numérique  $F(i, j)$  à niveaux de gris et de taille  $N \times N$ ,  $i \in [1, N]$  et  $j \in [1, N]$ , le calcul de ses moments de Zernike requiert une normalisation préalable de ses coordonnées à  $[-1, 1]$ . En effet  $F(i, j)$  doit être mappée vers une fonction  $G(x, y)$  définie dans  $[-1, 1] \times [-1, 1]$  avec  $(x, y) = F(i, j)$ ,  $x = \frac{(2i-N+1)}{(N-1)}$  et  $y = \frac{(2j-N+1)}{(N-1)}$ . Puis  $f(x, y)$  est définie à partir de  $G(x, y)$  comme suit :

$$f(x, y) = \{G(x, y) | x \in [-1, 1] \wedge y \in [-1, 1] \wedge x^2 + y^2 \leq 1\} \quad (2.7)$$

Grace à la condition  $n - |m| = \text{pair}$  et  $|m| \leq n$  de l'équation ( 2.11 ), le nombre maximal de moments de Zernike est  $\frac{1}{2}(n+1)(n+2)$  si l'ordre maximal demandé est  $n$ . Par exemple si le degré maximal est  $n = 5$ , l'image aura 21 moments de Zernike. En général, on retient le module des moments de Zernike  $Z_{nm} = |A_{nm}|$  comme descripteur d'image. Puisque  $A_{nm}^* = A_{n,-m}$  alors  $|A_{nm}| = |A_{n,-m}|$ , par conséquent on considère seulement  $A_{nm}$  avec  $m \geq 0$ . La taille du descripteur d'image calculé pour  $n = 5$ , sera alors de 12.

### 2.6.3. L'invariance à la rotation des moments de Zernike

Pour une transformation de type rotation, l'angle de rotation des moments de Zernike est simplement un facteur de phase. On peut alors considérer que les amplitudes des moments de Zernike sont invariantes à la rotation de l'image. Si l'image originale et l'image pivotée dans les mêmes coordonnées polaires sont respectivement dénotées  $f(r, \theta)$  et  $f^r(r, \theta)$ , on peut alors écrire :

$$f^r(r, \theta) = f(r, \theta - \alpha) \quad (2.13)$$

Avec  $\alpha$  : l'angle avec lequel l'image originale a été pivotée.

En utilisant l'équation (2.4), le moment de Zernike d'une image pivotée est défini comme suit :

$$\begin{aligned} A_{nm}^r &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta - \alpha) R_{nm}(r) \exp(-jm\theta) r dr d\theta \quad (2.14) \\ &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta - \alpha) R_{nm}(r) \exp(-jm(\theta - \alpha + \alpha)) r dr d\theta \\ &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta - \alpha) R_{nm}(r) \exp(-jm(\theta - \alpha)) \exp(-jm\alpha) r dr d\theta \end{aligned}$$

Donc la relation entre  $A_{nm}^r$  et  $A_{nm}$  peut s'écrire comme suit :

$$A_{nm}^r = A_{nm} \exp(-jm\alpha) \quad (2.15)$$

Cette équation indique que les moments de Zernike ont une propriété de transformation simple en présence de la rotation. Les amplitudes des moments de Zernike d'une image pivotée restent identiques à celles de l'image originale

$|A_{nm}^r| = |A_{nm}|$ . Ainsi, l'amplitude du moment de Zernike peut être utilisée comme un descripteur invariant à la rotation de l'image.

## 2.7 Les moments de Zernike et la recherche par le contenu

Dans la partie qui suit, nous allons dresser ce qui s'est fait dans les cinq dernières années concernant l'utilisation des moments de Zernike en recherche d'images par le contenu. Plusieurs travaux dans la recherche d'images par le contenu ont été proposés durant la dernière décennie. Cependant, les articles qui traitent spécifiquement des moments de Zernike et CBIR depuis 2013 ne sont pas nombreux. Di Yang et al. [21] ont proposé une architecture de CBIR sur un environnement distribué tels que les centres de données. Ils montrent qu'en intégrant des moments de Zernike comme descripteurs d'images dans des tables de hachage distribuées (*DHT*), et en exploitant un hachage avec une propriété de préservation de la localité (*Locality-preserving property LSH*), les images similaires sont plus probablement rassemblées dans le même nœud sans la connaissance d'une information globale. Ainsi, on arrive à générer un taux de rappel élevé, un petit nombre de nœuds tout en maintenant l'équilibre de la charge. Les auteurs dans [22] ont procédé à une investigation des propriétés des moments de Zernike comme descripteurs d'un système CBIR. Suite aux expérimentations effectuées sur des images d'objets binaires, ils ont conclu que les moments de Zernike étaient invariants à la rotation, à la translation mais non invariants au changement d'échelle. En 2014, J. Esther et M. Sathik [23] ont utilisé les moments de Zernike bidimensionnels pour extraire les descripteurs à partir d'images médicales prétraitées. La recherche d'images basée sur le contenu proposée utilise la technique d'apprentissage *Extreme Learning Machine (ELM)* en conjonction avec plusieurs métriques de distance. Ils ont conclu que le résultat est prometteur en combinant la méthode avec la distance de Hamming.

Les auteurs dans [24] et [25] ont eu recours aux moments de Zernike pour caractériser les formes dans les images. Les descripteurs ainsi générés sont conjointement exploités, au sein d'un même système CBIR, avec d'autres types de descripteurs basés sur la couleur HSV et la texture par représentation en ondelettes ou Gabor. Bien que le traitement adopté au niveau du module de mesures de similarité manque de clarté dans les deux références,

cette approche hybride semble permettre l'amélioration des performances de la recherche en termes de taux de précision.

En 2016, Pewint et al. [26] ont combiné dans un même système CBIR les moments de Zernike d'ordre 4 avec l'histogramme radial angulaire, horizontal et vertical extraits d'une image binaire pré-traitée. La mesure de similarité entre deux images est obtenue en sommant les quatre distances Euclidiennes calculées pour les quatre paires de primitives comparées. Les résultats obtenus montrent que l'apport du descripteur de Zernike à la précision de la recherche surpasse ceux des 3 autres descripteurs.

Dans [27], Aggarwal et Singh ont préféré les moments de Zernike pour la capture des caractéristiques globales de l'image médicale de type MRI ou CT et pour prendre avantage des autres propriétés comme l'invariance à la rotation, la robustesse au bruit et la réduction de la redondance ou de la dimensionnalité. Cette préférence est corroborée par les performances atteintes par le système CBIR. En effet, les résultats montrent des améliorations significatives des taux de succès de la recherche par rapport à d'autres descripteurs locaux comme *LBP (Local Binary Pattern)* et ses variantes.

Et finalement, les auteurs dans [28] proposent l'utilisation des moments de Zernike dans une application très originale des systèmes CBIR. Il s'agit en effet de la recherche d'images de faces humaines en lien avec le vieillissement subi. La requête soumise comprend une image de visage ainsi que l'âge de l'individu. Le système retourne l'image faciale la plus similaire à la requête et dans la tranche d'âge sélectionnée. La méthode proposée utilise les moments de Zernike (*ZM*) pour l'extraction des caractéristiques faciales (locales et globales) et un réseau de neurones *MLP (Multi-Layer Perceptron)* comme méthode d'apprentissage sur la base d'images. Selon les conclusions des auteurs, cette combinaison *ZM-MLP* a des résultats intéressants et se compare favorablement à l'état de l'art.

## 2.8 La transformée en contourlettes redondantes

La transformée en contourlettes redondantes (RCT) a été introduite par N. Baaziz [29] comme une version améliorée de la transformée en contourlettes (CT) [30]. Il s'agit d'une transformation qui permet la représentation du signal bidimensionnel (comme l'image) dans le domaine espace-fréquence selon une variété de résolutions spatiales et directionnelles. La décomposition se fait en deux étapes : une décomposition multi-résolution suivie d'une décomposition directionnelle (orientée). Les opérations de base utilisées sont principalement du filtrage linéaire et du sous-échantillonnage à cadence 2.

En utilisant la décomposition en pyramide Laplacienne classique, la première étape consiste à transformer l'image en une sous-image d'approximation qui contient les basses fréquences en plus de sous-images désignées par sous-bandes ou niveaux Laplaciens et contenant les détails de l'image (hautes fréquences). Le sous-échantillonnage inhérent à la pyramide Laplacienne résulte en un taux de redondance qui atteint les 4:3.

La deuxième étape de la décomposition utilise un banc de filtres directionnels (DFB) en formes de losange et de damier suivis d'un échantillonnage critique pour subdiviser chaque niveau Laplacien en sous-bandes directionnelles. Ainsi, les structures géométriques orientées de l'image sont capturées par les coefficients contourlettes. Cette méthode de représentation d'image en contourlettes permet d'avoir une reconstruction parfaite de l'image d'entrée.

Dans la variante redondante (RCT), la décomposition en pyramide Laplacienne utilise des filtres linéaires pseudo-Gaussiens. Aucun sous-échantillonnage n'est opéré [29]. Ainsi, on obtient une décomposition sur-échantillonné (redondante). L'image d'approximation ainsi que tous les niveaux Laplaciens ont tous la même taille que celle de l'image originale  $T$ . Dans l'étape DFB, le nombre de décompositions est limité à deux niveaux. Ainsi, chaque

sous-bande obtenue est de taille  $T/4$  pour un niveau Laplacien en entrée, de taille  $T$  (voir la Figure 5 et la Figure 6).

Le taux de redondance total de la RCT est supérieur à 4:3, mais ses propriétés sont bénéfiques à l'extraction de caractéristiques d'images de texture et la recherche par le contenu [31]. Une image originale et sa décomposition en sous-bandes RCT sont montrées à l'ANNEXE C.

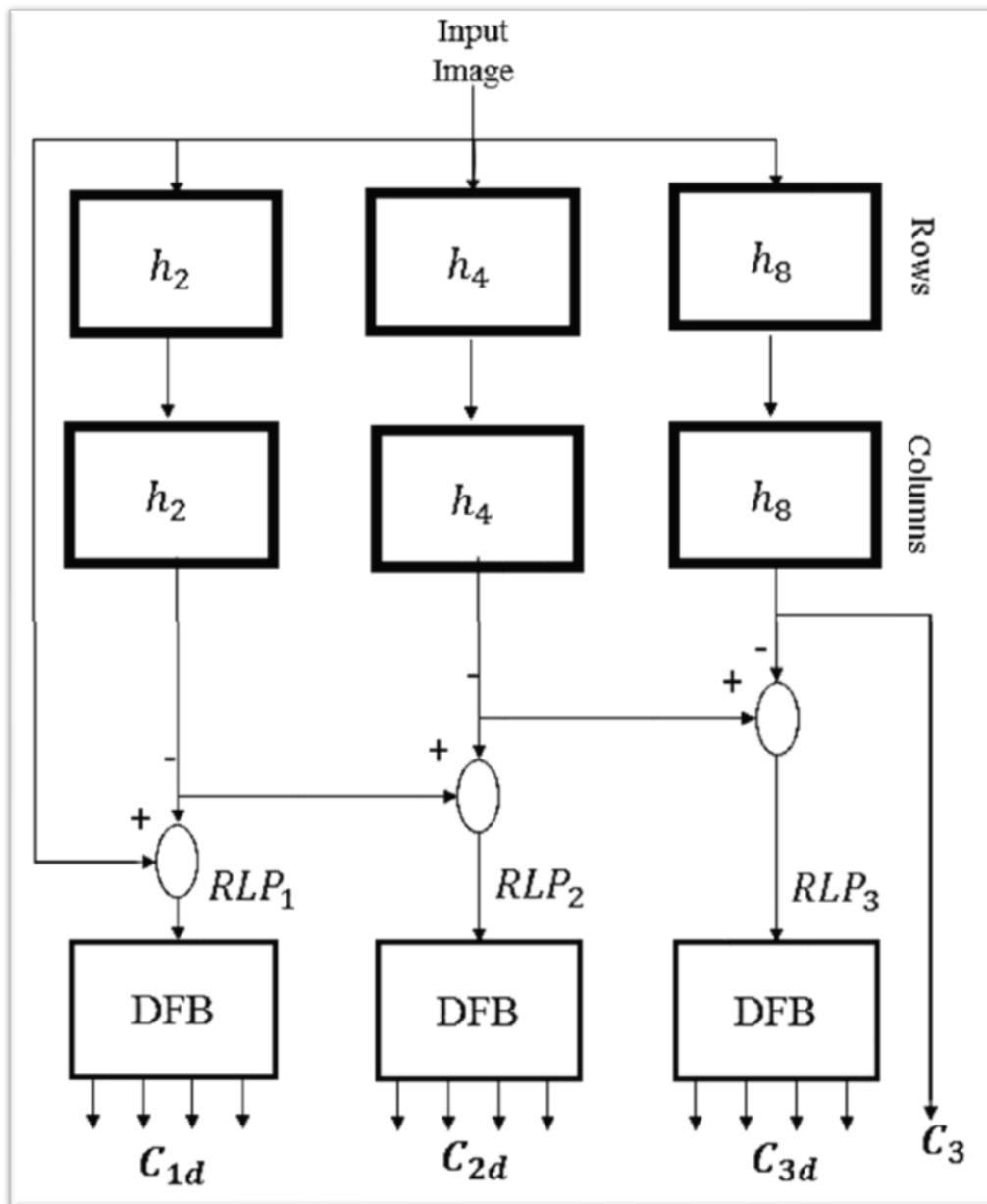


Figure 5 : Schéma bloc de décomposition en contourlettes redondantes d'une image.

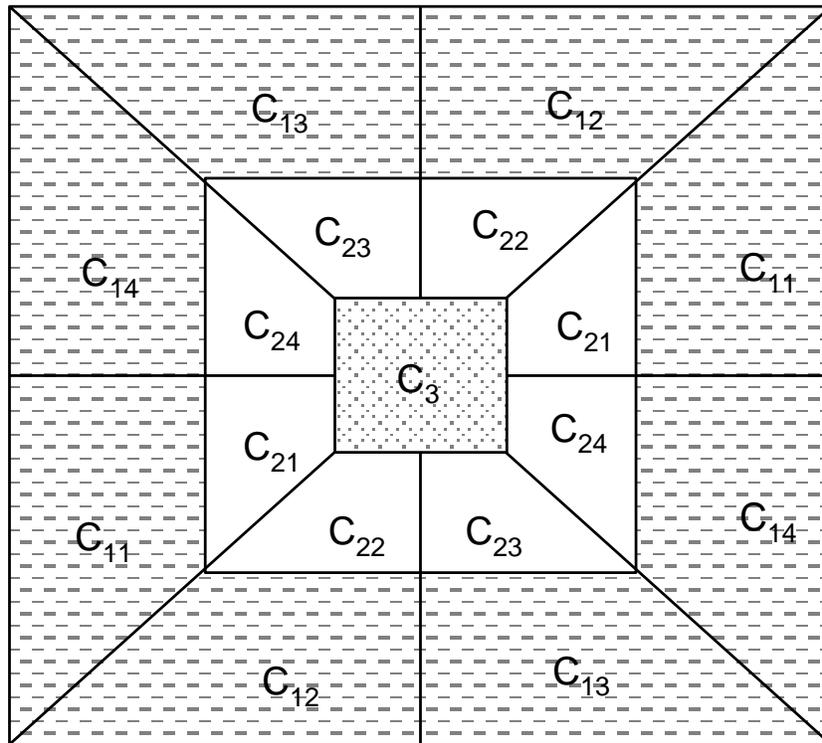


Figure 6 : Partition fréquentielle d'une RCT avec 2 niveaux de résolution et 4 directions fréquentielles.  
L'approximation de l'image ( $C_3$ ) n'est pas décomposée par le banc de filtres directionnels [3].

### 3 PROPOSITION DE RECHERCHE

#### 3.1 Objectifs

Dans ce projet de recherche, nous explorons l'apport des moments de Zernike combinés à la représentation en contourlettes en vue de l'extraction de descripteurs pertinents pour la recherche d'images par le contenu. Les images de textures sont particulièrement considérées. Les objectifs spécifiques qu'on poursuit sont :

- Valider la génération des moments de Zernike d'images binaires et en niveaux de gris. La reconstruction d'une image à partir de ses moments de Zernike fait partie de cet objectif.
- Intégrer l'extraction de descripteurs de Zernike dans un système CBIR complet et en évaluer les performances.
- Étudier les schémas d'incorporation de la transformée en contourlettes pour l'extraction de descripteurs d'images et analyser leur impact sur les performances du CBIR.
- Analyser les résultats obtenus sur des images de formes géométriques et de textures.
- Évaluer la performance des méthodes proposées.

### 3.2 Méthodologie

Il s'agit d'abord de valider la génération des moments de Zernike sur des exemples d'images, de comprendre le rôle des paramètres de génération comme l'ordre  $n$ . La reconstruction d'images à partir des moments de Zernike et divers ordres  $n$  fait partie de cette validation. Notre point de départ est la librairie LANS Pattern Recognition Toolbox développée par l'équipe de Kui-yu Chang [32]. La construction du descripteur de Zernike est guidée par des critères de choix de moments qui vont fournir un descripteur riche en information, discriminant entre les images et de dimensionnalité réduite (compact). L'approche proposée pour l'évaluation de la qualité des descripteurs de Zernike extraits consiste à les exploiter à des fins de recherche d'images par le contenu. Le prototype CBIR développé dans [31] est modifié et adapté à cette fin. Des schémas sont proposés et mis-en-œuvre en vue de l'incorporation d'une décomposition multirésolution de l'image telle que la transformée en contourlettes redondantes (RCT) en conjonction avec les moments de Zernike pour l'extraction de descripteurs pertinents. Une des approches considérées consiste à générer un descripteur global mettant en concaténation les moments de Zernike avec des primitives extraites du domaine RCT. Ces approches hybrides sont analysées et validées en mesurant la performance de la recherche par le contenu opérée sur des bases d'images connues dont des images de textures.

## 4 DÉVELOPPEMENTS

Afin de valider le bon fonctionnement de notre module utilisé pour l'extraction des descripteurs de Zernike et pour mieux choisir le nombre de moments, on a décidé de commencer par le test de reconstruction des images à partir des vecteurs de descripteurs sur deux types de base de données d'images :

**MPEG7-CE Shape-1** : est une base de données de 1400 images binaires de formes organisées en 70 classes distinctes contenant 20 images chacune (voir la figure 7). Les formes incluent des variations dans la rotation, la position et la taille. La résolution de l'image n'est pas constante pour des classes (comme dans les classes *hat*, *butterfly*). L'objet contenu peut-être centré ou pas. Une classe peut contenir un même objet mais montrant des formes différentes perceptibles (comme dans les classes *bird* et *car*). Pour ce projet on a utilisé les 16 premières images de chacune des 70 classes, ce qui fait un total de 1120 images de notre base de tests. La base de données MPEG7- CE Shape-1 peut être téléchargée du lien suivant :

[http://www.imageprocessingplace.com/downloads\\_V3/root\\_downloads/image\\_databases/MPEG7\\_CE-Shape-1\\_Part\\_B.zip](http://www.imageprocessingplace.com/downloads_V3/root_downloads/image_databases/MPEG7_CE-Shape-1_Part_B.zip)

**VisTex** : est une base de données de 640 images de textures organisées en 40 classes distinctes (16 images par classe) tel que montré dans la figure 8. Toutes les images sont en niveaux de gris et de même taille (128×128) pixels. La base de données VisTex peut être téléchargée du lien suivant : <http://vismod.media.mit.edu/pub/VisTex/VisTex.tar.gz>



Figure 7 : La base de données MPEG7-CE Shape-1 utilisée contient 70 familles d'images. Chaque image ci-dessus fait partie d'une classe distincte contenant 16 images.

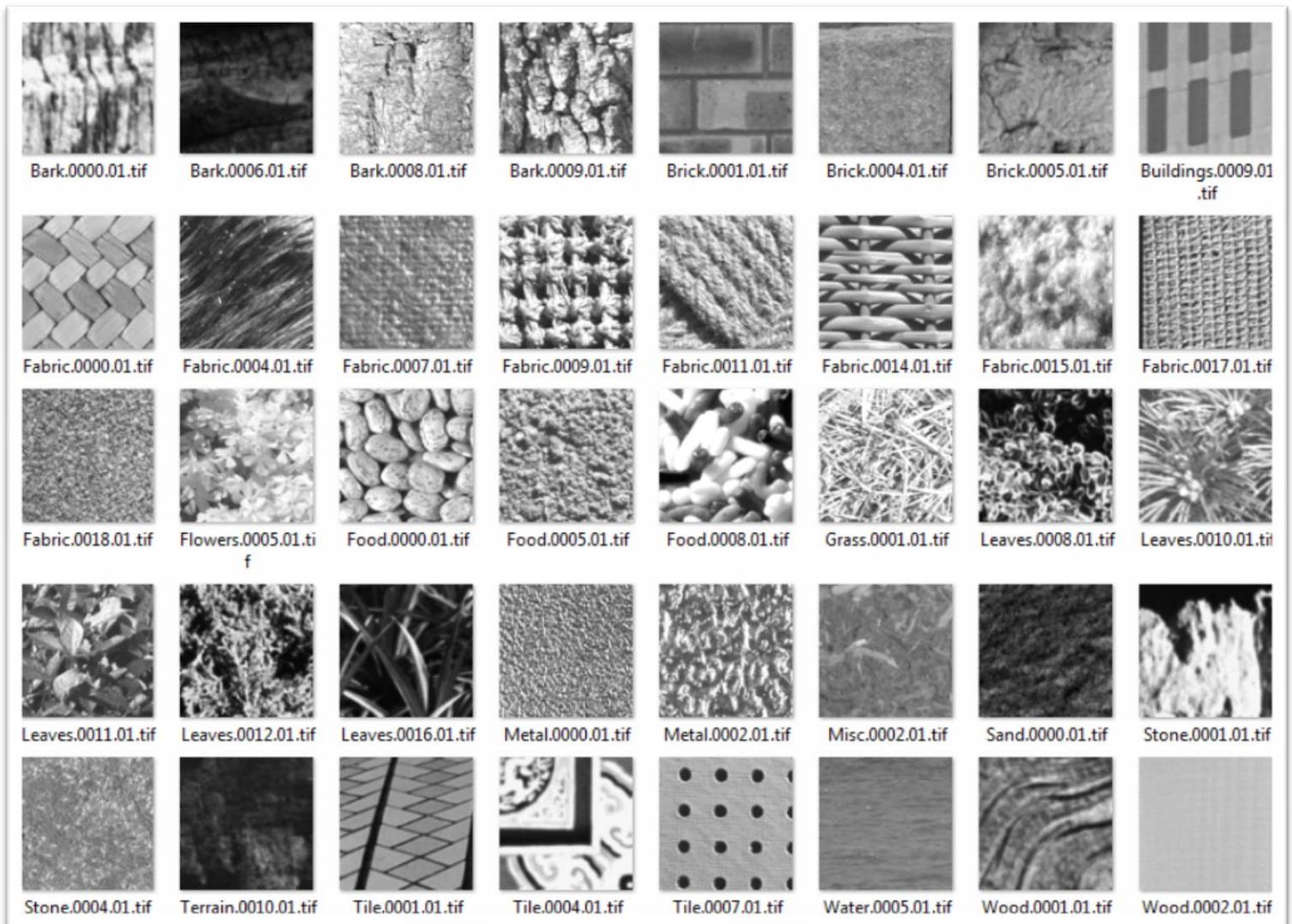


Figure 8 : La base de données VisTex contient 40 familles d'images. Chaque image ci-dessus fait partie d'une classe distincte contenant 16 images.

## 4.1 Configuration de l'environnement de test

Pour le calcul des moments de Zernike d'une image et sa reconstruction à partir de ses moments de Zernike, nous avons fait usage des fonctions de la librairie *LANS Pattern Recognition Toolbox* [32], qui offre plusieurs routines de visualisation, de régression, de reconnaissance et d'extraction de descripteurs. Pour la simulation et les tests, la configuration matérielle et logicielle utilisée est la suivante:

### Configuration logicielle :

- Matlab R2015a
- LANS Pattern Recognition Toolbox Version 2000.11.27
- Windows 10 Enterprise, Version:1703, OS Build: 15063.1387

### Configuration matérielle :

- Lenovo T470s i7-7600U, CPU@2.8Ghz, RAM: 16 Go

## 4.2 Reconstruction d'image à partir des moments de Zernike

Cette étape de reconstruction nous a aidé à valider le bon fonctionnement du code utilisé pour le calcul des moments de Zernike et comparer les résultats pour plusieurs ordres  $n$ . La librairie *LANS toolbox* propose deux fonctions principales : la fonction "*lans\_zmoment( )*" qui, à partir d'une image exemple et d'un ordre  $n$  défini, retourne le vecteur des moments de Zernike calculés, la matrice des ordres et répétitions utilisés pour le calcul, un vecteur d'indices 1-D de la région du cercle unité retenue et un vecteur des polynômes de Zernike. Il est à noter que cette fonction s'applique aussi bien pour des images en niveaux de gris ou binaires. La fonction "*lans\_zmrecon( )*" permet la

reconstruction de l'image à partir des 4 résultats retournés par "*lans\_zmoment( )*" plus les dimensions de l'image de test et un seuil pour l'élimination des pixels (voir l'Annexe 1).

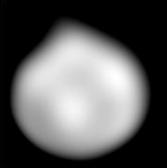
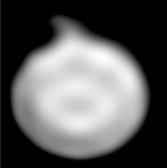
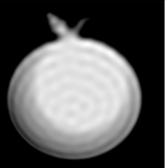
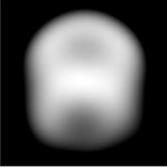
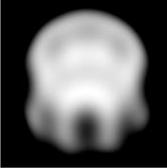
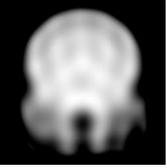
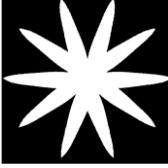
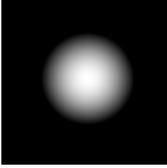
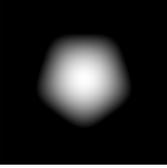
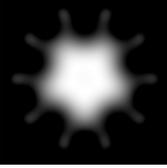
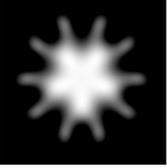
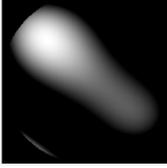
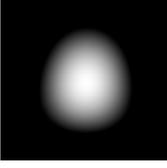
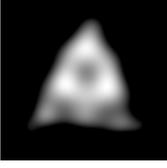
Base d'images : MPEG7-CE Shape-1						
Image Originale	Ordre 5	Ordre 10	Ordre 15	Ordre 20	Ordre 30	Ordre 40
						
						
						
						
						

Figure 9 : Résultat de reconstruction d'images d'objets (binaires) à partir des moments de Zernike à ordres croissants.

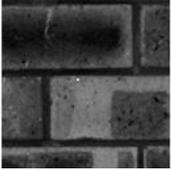
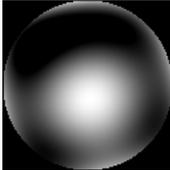
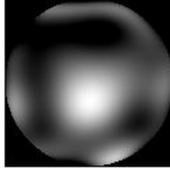
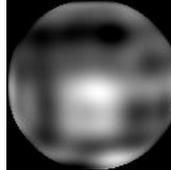
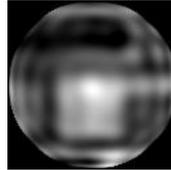
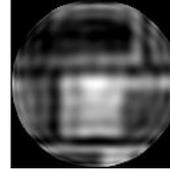
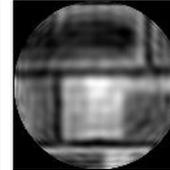
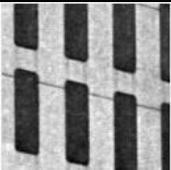
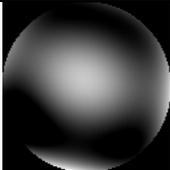
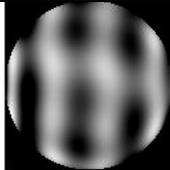
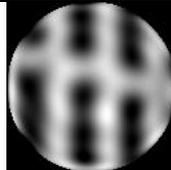
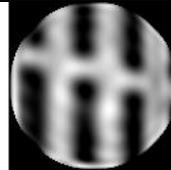
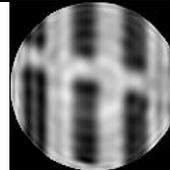
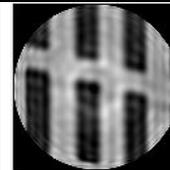
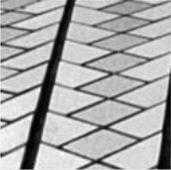
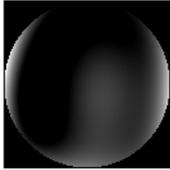
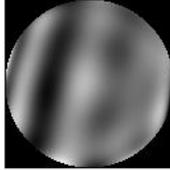
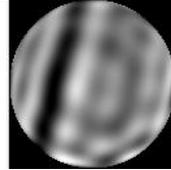
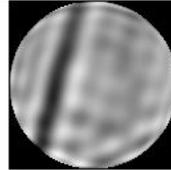
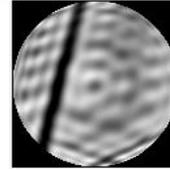
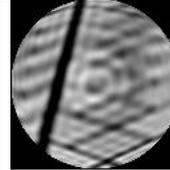
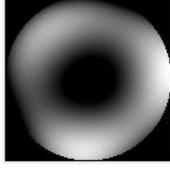
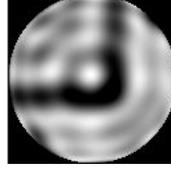
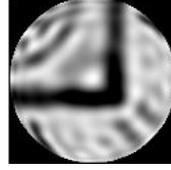
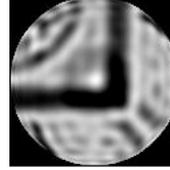
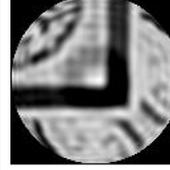
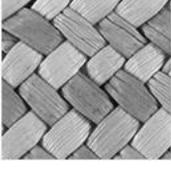
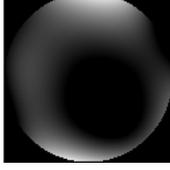
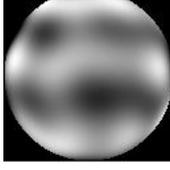
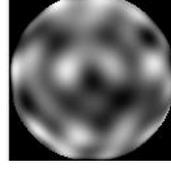
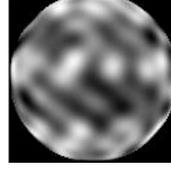
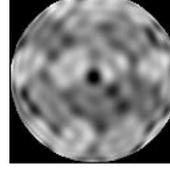
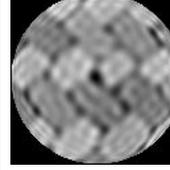
Base d'images : VisTex						
Image Originale	Ordre 5	Ordre 10	Ordre 15	Ordre 20	Ordre 30	Ordre 40
						
						
						
						
						

Figure 10 : Résultat de reconstruction d'images de textures à partir des moments de Zernike à ordres croissants.

Les figures 9 et 10 illustrent les résultats de reconstruction d'image à partir des moments de Zernike avec six valeurs d'ordre : 5, 10, 15, 20, 30 et 40. Cinq différentes images extraites de chacune des bases de données MPEG7-CE Shape-1 et VisTex sont utilisées. On remarque que l'ordre 15 permet déjà une distinction de la forme de l'objet ou bien du

motif contenu dans l'image. Cette distinction s'améliore au fur et à mesure qu'on augmente la valeur de l'ordre.

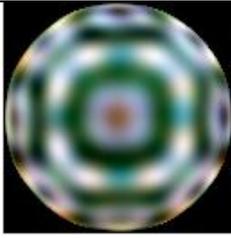
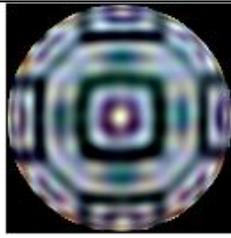
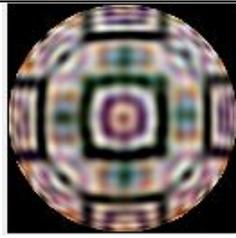
Image RGB				
Image Originale 128x128	Ordre 20	Ordre 25	Ordre 30	Ordre 40
				

Figure 11 : Résultat de reconstruction d'une image RGB à partir des moments de Zernike.

Finalement nous avons expérimenté la méthode sur une image en couleur RGB. Chaque composante R, G et B est traitée séparément pour le calcul des moments de Zernike ainsi que pour la reconstruction. À la fin, les 3 composantes reconstruites sont concaténées et l'image en couleur est affichée. Cependant pour une image complexe comme celle utilisée dans la figure 11, l'image reconstruite est reconnaissable à partir d'un ordre 20 et le résultat s'améliore en augmentant l'ordre.

### 4.3 Extraction des descripteurs de Zernike

Pour extraire un descripteur de Zernike d'ordre  $n$  (par exemple:  $n=10$ ) sur une image donnée, on calcule tous les moments de Zernike  $A_{nm}$  de cette image en commençant par un ordre 0 et une répétition 0 et en augmentant  $n$  et  $m$  tout en respectant les deux conditions suivantes:  $(n-|m|)$  pair et  $|m| \leq n$ . Au final, on obtient 36 moments à valeurs complexes. Le descripteur d'ordre 10 sera alors un vecteur constitué des amplitudes  $Z_{nm} = |A_{nm}|$  et sera de dimension 36. Le tableau 2 montre un exemple de descripteur de Zernike d'ordre 10 extrait de l'image binaire *apple-1* de la base d'images MPEG7-CE Shape-1.

Descripteur de Zernike	Ordre $n$	Répétition $m$	Valeurs $Z_{nm} =  A_{nm} $ arrondies à 2 chiffres décimaux
$Z_{00}$	0	0	9001.49
$Z_{11}$	1	1	1990
$Z_{20}$	2	0	11127.56
$Z_{22}$	2	2	232.22
$Z_{31}$	3	1	1118.24
$Z_{33}$	3	3	559.66
$Z_{40}$	4	0	1287.02
$Z_{42}$	4	2	386.12
$Z_{44}$	4	4	495.32
$Z_{51}$	5	1	3026.28
$Z_{53}$	5	3	230.18
$Z_{55}$	5	5	555.25
$Z_{60}$	6	0	4442.8
$Z_{62}$	6	2	797.61
$Z_{64}$	6	4	111.1
$Z_{66}$	6	6	497.95
$Z_{71}$	7	1	802.01
$Z_{73}$	7	3	525.62
$Z_{75}$	7	5	497.84
$Z_{77}$	7	7	423.66
$Z_{80}$	8	0	289.99
$Z_{82}$	8	2	294.41
$Z_{84}$	8	4	766.06
$Z_{86}$	8	6	548.96
$Z_{88}$	8	8	337.5
$Z_{91}$	9	1	3161.53

$Z_{93}$	9	3	321.06
$Z_{95}$	9	5	389.18
$Z_{97}$	9	7	644.7
$Z_{99}$	9	9	255.39
$Z_{100}$	10	0	2093.41
$Z_{102}$	10	2	1576.52
$Z_{104}$	10	4	527.59
$Z_{106}$	10	6	519.89
$Z_{108}$	10	8	665.67
$Z_{1010}$	10	10	190.77

Tableau 2 : Amplitude des Moments de Zernike de l'image *apple-1* pour un ordre  $n=10$ . Le descripteur résultant est de dimension=36.

Il est intéressant de constater le pouvoir de discrimination entre images que peuvent offrir les descripteurs de Zernike ainsi que l'influence de l'ordre maximal  $n$  sur cet aspect. En effet, les résultats du Tableau 4 correspondent à 4 images différentes et issues de 4 classes distinctes de MPEG7-CE Shape-1. On peut bien voir que l'ordre de grandeur du premier moment  $Z_{00}$  permet déjà de distinguer les formes plutôt circulaires des autres formes (formes quelconques dans *spoon-1* et *tree-1*). Lorsqu'on augmente l'ordre du moment de Zernike, les valeurs obtenues permettent de différencier les 4 images.

Par ailleurs, le Tableau 3 donne un aperçu de quelques valeurs  $Z_{nm}$  pour 5 images de même classe supposées contenir la même forme d'objet. On peut alors constater que l'ordre 0 permet une caractérisation grossière de la forme dans l'image, alors qu'aux ordres supérieurs ( $n > 15$ ), les valeurs des moments accusent des écarts importants permettant ainsi de distinguer les détails que peuvent abriter chacune des formes contenues dans les images.

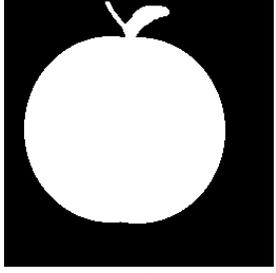
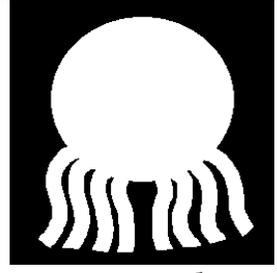
Image utilisée					
		<i>apple-1</i>	<i>octopus-1</i>	<i>spoon-1</i>	<i>tree-1</i>
Taille de l'image		256×256	256×256	256×256	256×256
Les Moments de Zernike	$Z_{0\ 0}$	9001.48	9972.01	2234.22	3754.78
	$Z_{5\ 1}$	3026.28	557.71	1549.81	1227.57
	$Z_{5\ 5}$	555.25	1215.01	1536.16	180.98
	$Z_{10\ 0}$	2093.41	800.2	1107.54	764.48
	$Z_{10\ 10}$	190.77	203.98	513.25	47.99
	$Z_{15\ 1}$	327.39	1176.85	328.8	438.75
	$Z_{15\ 15}$	297.12	554.13	974.53	15.31
	$Z_{20\ 0}$	2119.64	1122.12	273.79	264.2
	$Z_{20\ 20}$	230.57	447.37	425.92	4.5
	$Z_{23\ 23}$	136.25	877.05	459.07	2.26
$Z_{25\ 25}$	136.96	370.75	503.73	1.47	

Tableau 3 : Amplitude de moments de Zernike  $Z_{nm}$  pour 4 images appartenant à des classes distinctes

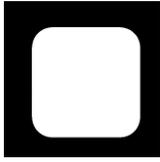
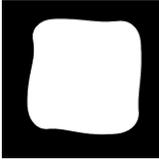
Image utilisée						
		<i>device3-1</i>	<i>device3-3</i>	<i>device3-5</i>	<i>device3-13</i>	<i>device3-16</i>
Taille de l'image		558 × 558	558×558	558×558	558×558	558×558
Les Moments de Zernike	$Z_{00}$	47023.92	47090.76	38804.2	38763.14	47047.79
	$Z_{51}$	8466.34	8577.17	706.69	340.02	8519.26
	$Z_{55}$	4272.82	4313.9	1524.13	1315.3	4250.83
	$Z_{100}$	6836.99	6651.94	5405.08	5410.25	6691.91
	$Z_{1010}$	813.03	861.81	767.28	769.51	735.91
	$Z_{151}$	1259.4	1317.3	1910.42	2149.21	1104.4
	$Z_{1515}$	191.17	205.82	542.59	524.42	166.68
	$Z_{200}$	318.38	231.32	1735.24	1824.05	430.4
	$Z_{2020}$	101.22	111.69	1167.42	1188.03	108.19
	$Z_{2323}$	43.4	39.89	285.35	277.05	81.29
$Z_{2525}$	25.69	35.73	248.71	236.59	97.99	

Tableau 4 : Amplitude des moments de Zernike  $Z_{nm}$  pour 5 images de la même classe 'Device3' et différentes paires de valeurs  $(n,m)$ .

#### 4.4 Recherche d'images par les descripteurs de Zernike

Pour chaque base de données, chaque image est traitée afin d'extraire son descripteur de Zernike d'ordre  $n$  et le sauvegarder dans l'index du CBIR. Il s'agit de l'étape 'offline' d'un CBIR. Pour les besoins expérimentaux, on génère plusieurs index, chacun

correspondant à un ordre  $n$  distinct. Dans les expérimentations qui vont suivre, la similarité entre deux images est mesurée en calculant la distance Euclidienne entre les deux descripteurs de Zernike correspondants puis en normalisant par leur dimension. La phase de recherche ‘en ligne’ s’exécute comme suit :

- 1- À chaque fois qu’une requête est soumise au CBIR, son descripteur de Zernike d’ordre  $n$  est extrait  $Q_{nm}$ .
- 2-  $Q_{nm}$  est comparé à chaque descripteur de l’index. Les similarités (distances) sont mesurées puis ordonnées par ordre croissant.
- 3- Les  $N$  plus petites distances sont utilisées pour récupérer et retourner les  $N$  images correspondantes comme étant les Top- $N$  images les plus similaires à la requête.

#### 4.4.1. Impact de l’ordre du descripteur sur la précision

Nous présentons les résultats de la recherche d’images basée sur les descripteurs de Zernike, et appliquée à des images requêtes de la base de données MPEG7-CE Shape-1. On considère l’image requête *apple-2* et on affiche le résultat du CBIR pour une recherche du Top-10. Pour chaque recherche, les descripteurs utilisés sont d’un ordre  $n$  distinct.

Les 7 tableaux qui vont suivre (

Image requête	Résultat de recherche pour des descripteurs de Zernike d’ordre 0				
	1	2	3	4	5

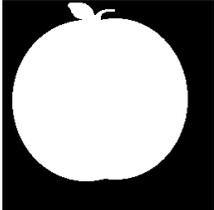
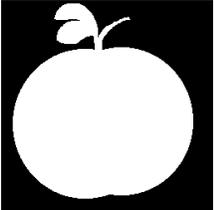
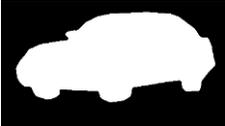
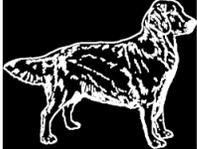
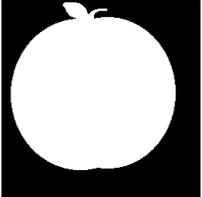
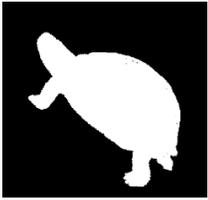
	 <i>apple-2</i>	 <i>apple-5</i>	 <i>personal_car-9</i>	 <i>dog-6</i>	 <i>lmfish-12</i>
	<b>Distance Euclidienne</b>				
	0	27.1	31.5	49.7	51.2
	6	7	8	9	10
 <i>apple-2</i>	 <i>turtle-5</i>	 <i>personal_car-1</i>	 <i>personal_car-14</i>	 <i>face-13</i>	 <i>HCircle-6</i>
	<b>Distance Euclidienne</b>				
	92	93.9	105.7	113.6	121.3

Tableau 5, Tableau 6,

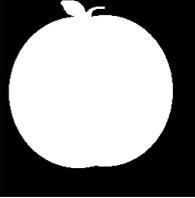
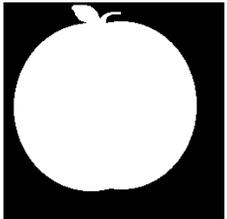
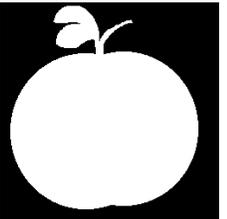
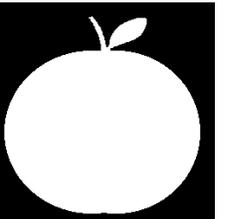
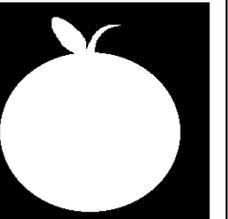
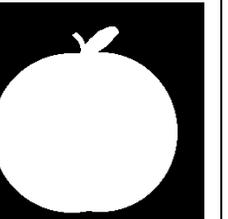
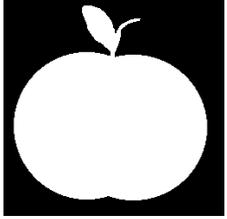
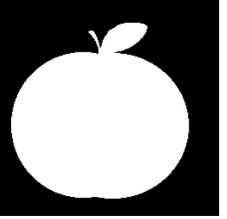
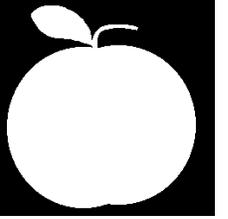
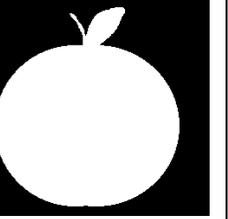
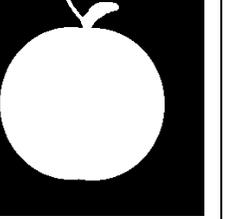
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 5				
	1	2	3	4	5
 apple-2	 apple-2	 apple-5	 apple-13	 apple-7	 apple-6
	<b>Distance Euclidienne</b>				
	0	245.3	245.5	264.2	286.8
	6	7	8	9	10
	 apple-11	 apple-16	 apple-3	 apple-12	 apple-1
	<b>Distance Euclidienne</b>				
	288.8	307.3	318.1	343.7	350.1

Tableau 7, Tableau 8, Tableau 9, Tableau 10 et Tableau 11) présentent les Top-10 images retournées par le CBIR ainsi que les mesures de similarité avec l'image requête. On peut voir que la *Précision* de la recherche du CBIR s'améliore au fur et à mesure que l'ordre des descripteurs de Zernike augmente. La *Précision* atteint sa valeur maximale (100%) à l'ordre  $n=22$  (10 images de la même classe que la requête sont retrouvées). Aussi, on constate qu'en augmentant l'ordre  $n$ , les mesures de distance diminuent.

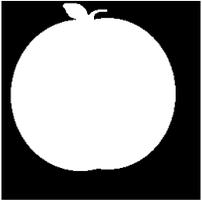
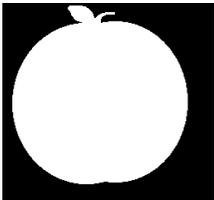
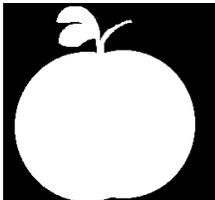
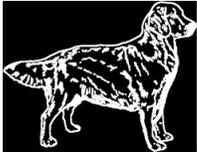
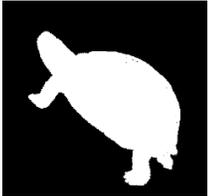
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 0				
	1	2	3	4	5
 <i>apple-2</i>	 <i>apple-2</i>	 <i>apple-5</i>	 <i>personal_car-9</i>	 <i>dog-6</i>	 <i>lmfish-12</i>
	<b>Distance Euclidienne</b>				
	0	27.1	31.5	49.7	51.2
	6	7	8	9	10
	 <i>turtle-5</i>	 <i>personal_car-1</i>	 <i>personal_car-14</i>	 <i>face-13</i>	 <i>HCircle-6</i>
	<b>Distance Euclidienne</b>				
	92	93.9	105.7	113.6	121.3

Tableau 5 : Top-10 images retrouvées pour la requête *apple-2* et leur distance Euclidienne entre descripteurs de Zernike d'ordre 0. La *Précision* est de 20%.

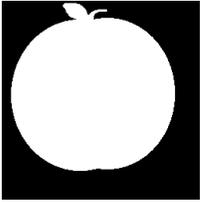
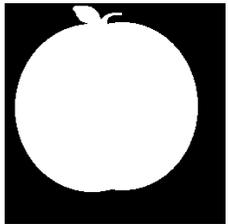
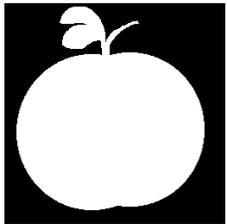
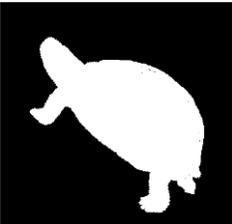
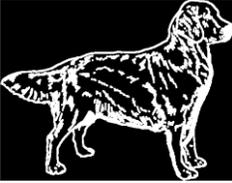
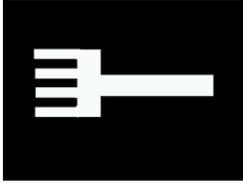
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 1				
	1	2	3	4	5
 apple-2	 apple-2	 apple-5	 turtle-5	 lizzard-1	 fly-11
	<b>Distance Euclidienne</b>				
	0	89.2	101.5	150.6	244.9
	6	7	8	9	10
	 fly-10	 HCircle-3	 dog-6	 fork-16	 Camel-8
	<b>Distance Euclidienne</b>				
	247.3	247.9	273.3	295.7	360.2

Tableau 6 : Top-10 images retrouvées pour la requête apple-2 et leur distance Euclidienne entre descripteurs de Zernike d'ordre 1. La *Précision* est de 20%.

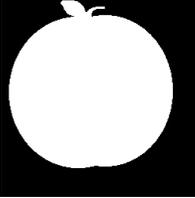
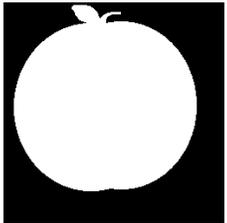
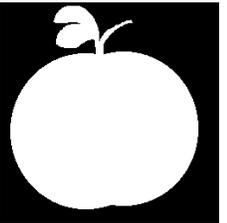
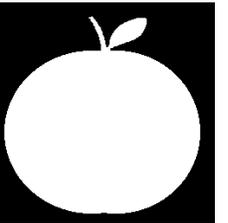
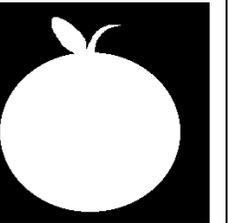
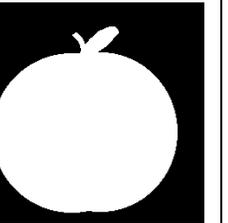
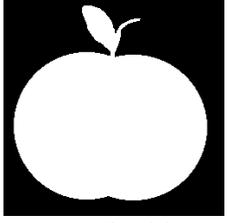
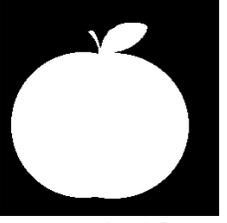
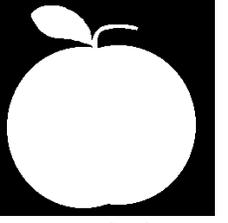
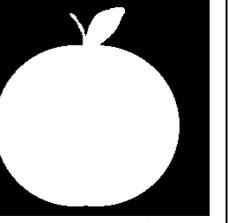
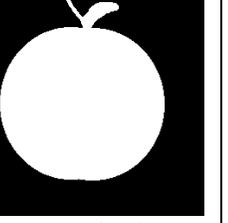
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 5				
	1	2	3	4	5
 apple-2	 apple-2	 apple-5	 apple-13	 apple-7	 apple-6
	<b>Distance Euclidienne</b>				
	0	245.3	245.5	264.2	286.8
	6	7	8	9	10
	 apple-11	 apple-16	 apple-3	 apple-12	 apple-1
	<b>Distance Euclidienne</b>				
	288.8	307.3	318.1	343.7	350.1

Tableau 7 : Top-10 images retrouvées pour la requête *apple-2* et leur distance Euclidienne entre descripteurs de Zernike d'ordre 5. La *Précision* est de 100%.

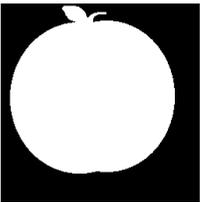
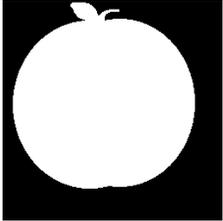
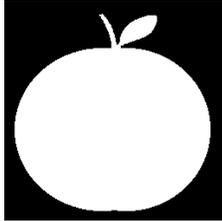
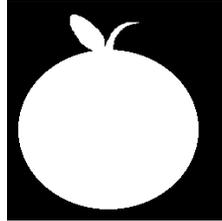
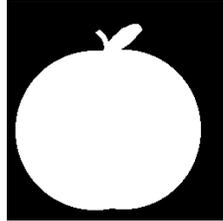
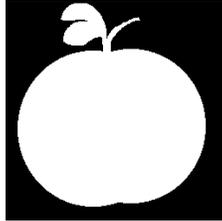
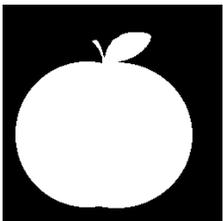
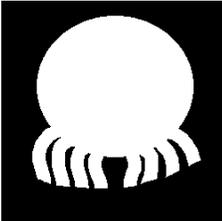
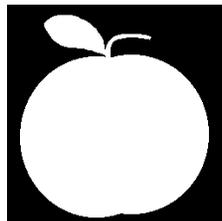
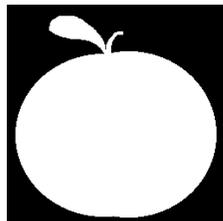
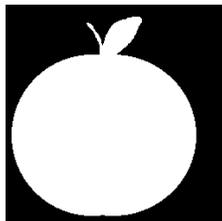
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 10				
	1	2	3	4	5
 <i>apple-2</i>	 <i>apple-2</i>	 <i>apple-13</i>	 <i>apple-7</i>	 <i>apple-6</i>	 <i>apple-5</i>
	<b>Distance Euclidienne</b>				
	0	131.7	132.5	134.3	140.5
	6	7	8	9	10
	 <i>apple-16</i>	 <i>Octopus-2</i>	 <i>apple-3</i>	 <i>apple-15</i>	 <i>apple-12</i>
	<b>Distance Euclidienne</b>				
	150.3	151.3	158	160.1	162.6

Tableau 8 : Top-10 images retrouvées pour la requête *apple-2* et leur distance Euclidienne entre descripteurs de Zernike d'ordre 10. La *Précision* est de 90%.

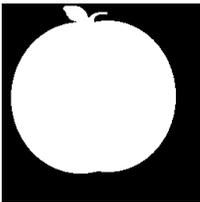
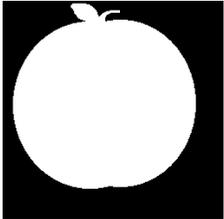
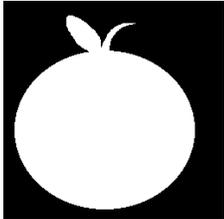
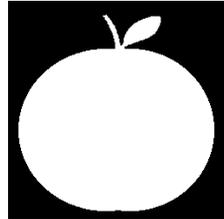
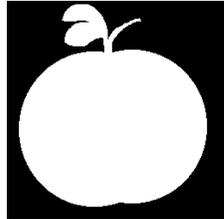
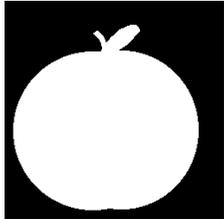
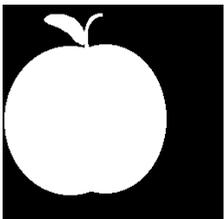
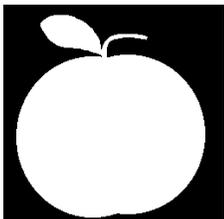
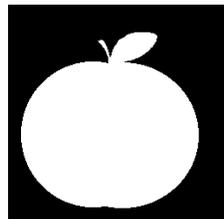
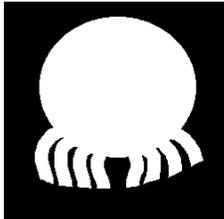
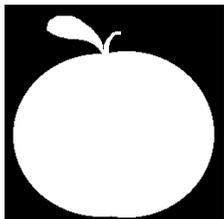
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 15				
	1	2	3	4	5
 <i>apple-2</i>	 <i>apple-2</i>	 <i>apple-7</i>	 <i>apple-13</i>	 <i>apple-5</i>	 <i>apple-6</i>
	<b>Distance Euclidienne</b>				
	0	82.9	82.9	83.1	86.9
	6	7	8	9	10
	 <i>apple-10</i>	 <i>apple-3</i>	 <i>apple-16</i>	 <i>Octopus-2</i>	 <i>apple-15</i>
	<b>Distance Euclidienne</b>				
	88	93.8	94.1	94.6	95.7

Tableau 9 : Top-10 images retrouvées pour la requête *apple-2* et leur distance Euclidienne entre descripteurs de Zernike d'ordre 15. La *Précision* est de 90%.

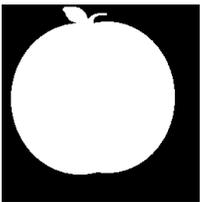
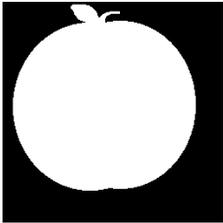
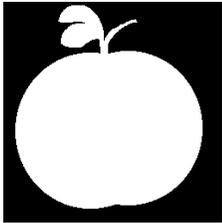
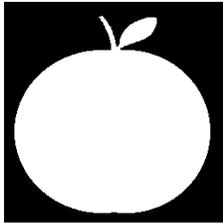
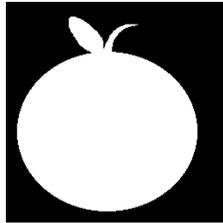
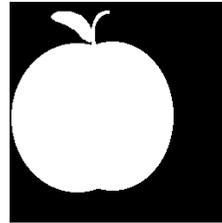
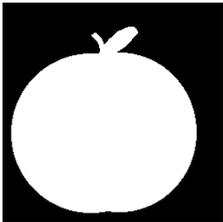
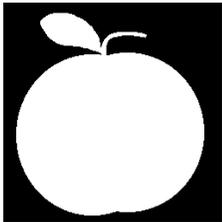
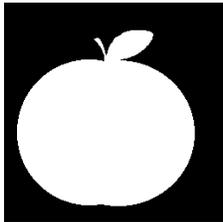
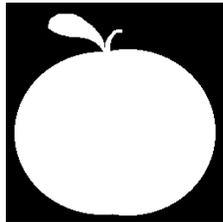
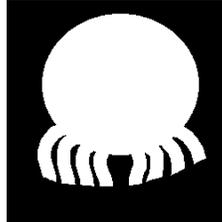
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 20				
	1	2	3	4	5
 apple-2	 apple-2	 apple-5	 apple-13	 apple-7	 apple-10
	<b>Distance Euclidienne</b>				
	0	54.9	55.3	55.3	56.7
	6	7	8	9	10
	 apple-6	 apple-3	 apple-16	 apple-15	 Octopus-2
	<b>Distance Euclidienne</b>				
	57.6	60.8	61.2	61.7	62.1

Tableau 10 : Top-10 images retrouvées pour la requête *apple-2* et leur distance Euclidienne entre descripteurs de Zernike d'ordre 20. La *Précision* est de 90%.

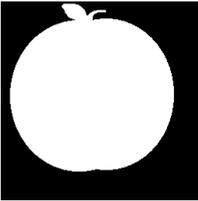
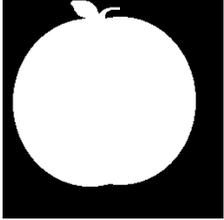
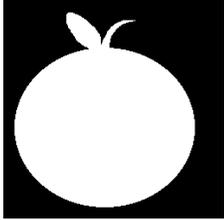
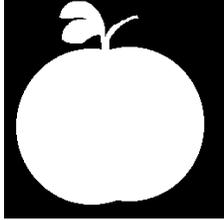
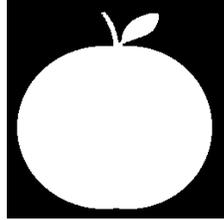
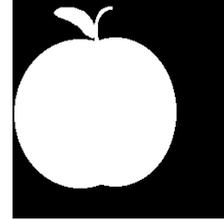
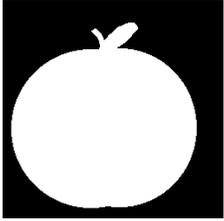
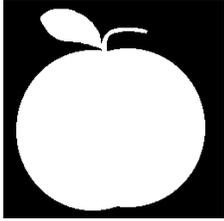
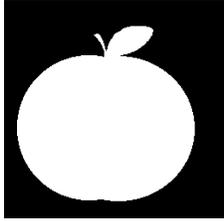
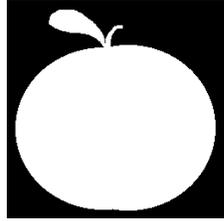
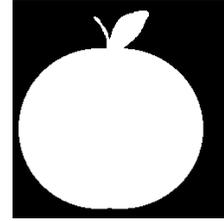
Image requête	Résultat de recherche pour des descripteurs de Zernike d'ordre 22				
	1	2	3	4	5
 apple-2	 apple-2	 apple-7	 apple-5	 apple-13	 apple-10
	<b>Distance Euclidienne</b>				
	0	47.3	47.4	47.8	48.1
	6	7	8	9	10
	 apple-6	 apple-3	 apple-16	 apple-15	 apple-12
	<b>Distance Euclidienne</b>				
	49.4	51.8	52.5	52.8	53.4

Tableau 11 : Top-10 images retrouvées pour la requête *apple-2* et leur distance Euclidienne entre descripteurs de Zernike d'ordre 22. La *Précision* est de 100%.

#### 4.4.2. Évaluation de la performance globale du CBIR

L'expérimentation décrite ci-après permet une évaluation globale de la performance de la recherche du CBIR basée sur les moments de Zernike d'ordre  $n$ . Pour une base de données contenant  $M$  images, chaque image va jouer le rôle de requête. Ainsi, on lance  $M$  requêtes et le système CBIR retourne, pour chacune, les  $N$  meilleures images (Top-N) trouvées. Les valeurs de *Rappel* (pour  $M$  requêtes) sont alors moyennées et exprimées en pourcentage pour donner la mesure de performance *RR%* du CBIR: *Taux de Réussite de la Recherche du Top-N*. On suppose que la vérité terrain est accessible (connaissance de toutes les images pertinentes à chaque requête) et que les index basés sur les descripteurs de Zernike d'ordre  $n$  sont préalablement générés en phase *offline*.

		<b>Taux de réussite de la recherche (RR%)</b>			
		Ordre 5	Ordre 10	Ordre 15	Ordre 20
<b>Classe d'image</b>	<b>device7</b>	63	70	100	100
	<b>children</b>	100	100	100	100
	<b>teddy</b>	100	100	100	100
	<b>fountain</b>	89	91	99	99
	<b>bottle</b>	89	96	93	93
	<b>chopper</b>	71	70	89	90
	<b>device1</b>	25	44	88	89
	<b>car</b>	84	87	89	89
	<b>rat</b>	88	88	88	88
	<b>device8</b>	77	77	86	87
	<b>HCircle</b>	85	87	87	86
	<b>device3</b>	70	79	84	85
	<b>device4</b>	78	80	84	84
	<b>Heart</b>	84	85	84	84
	<b>device0</b>	43	58	79	81
	<b>stef</b>	90	84	79	81
	<b>apple</b>	82	80	79	79
	<b>device9</b>	77	79	78	78
	<b>truck</b>	73	75	78	78
	<b>device2</b>	29	31	69	73
	<b>carriage</b>	68	68	71	72
	<b>device5</b>	39	58	68	68
<b>hammer</b>	74	81	69	68	
<b>Comma</b>	64	64	66	68	

	<b>Misk</b>	62	66	69	68
	<b>Glas</b>	52	63	66	67
	<b>face</b>	63	63	63	63
	<b>tree</b>	53	52	52	52
	<b>brick</b>	50	52	52	52
	<b>device6</b>	37	39	47	50
	<b>jar</b>	40	41	48	48
	<b>Bone</b>	46	44	47	47
	<b>key</b>	55	54	48	46
	<b>cellular_phone</b>	39	43	46	45
	<b>shoe</b>	41	45	45	45
	<b>octopus</b>	39	38	39	39
	<b>cup</b>	30	36	39	39
	<b>fork</b>	34	36	38	38
	<b>spoon</b>	39	41	41	36
	<b>elephant</b>	39	38	35	36
	<b>flatfish</b>	39	39	39	36
	<b>camel</b>	33	38	34	34
	<b>crown</b>	28	29	28	28
	<b>watch</b>	25	28	28	28
	<b>frog</b>	23	23	27	27
	<b>dog</b>	27	27	26	26
	<b>personal_car</b>	27	25	25	25
	<b>fish</b>	26	23	24	24
	<b>horseshoe</b>	29	26	24	24
	<b>spring</b>	23	23	23	23
	<b>cattle</b>	24	24	22	22
	<b>pencil</b>	21	23	22	22

<b>classic</b>	18	20	21	21
<b>deer</b>	18	20	21	21
<b>ray</b>	21	19	20	19
<b>chicken</b>	17	17	17	17
<b>bat</b>	15	16	17	16
<b>bell</b>	14	14	16	16
<b>pocket</b>	18	17	15	15
<b>lmfish</b>	13	14	14	14
<b>fly</b>	20	16	13	14
<b>guitar</b>	17	16	13	13
<b>horse</b>	14	14	13	13
<b>beetle</b>	14	13	13	12
<b>lizzard</b>	13	11	12	12
<b>sea_snake</b>	14	14	12	12
<b>turtle</b>	11	12	13	12
<b>bird</b>	11	11	11	11
<b>butterfly</b>	12	12	11	11
<b>hat</b>	12	11	11	11
<b>RR% (global)</b>	<b>43.7</b>	<b>45.4</b>	<b>48.1</b>	<b>48.1</b>

Tableau 12 : Performance du CBIR ( $RR\%$ ) pour une recherche Top-16 effectuée pour les classes d'images MPEG7-CE Shape-1. Quatre ordres  $n$  distincts sont utilisés pour les descripteurs de Zernike.

En analysant les résultats du Tableau 12 et relatif à la base de données MPEG7-CE Shape-1, on remarque que globalement, les taux de réussite de la recherche ( $RR\%$ ) s'améliorent avec l'augmentation de l'ordre  $n$  du descripteur de Zernike. La valeur  $n=15$  semble être un bon compromis entre la performance de la recherche  $RR\%$  et la dimension du descripteur. Par ailleurs, il est à noter que les plus faibles taux  $RR\%$  correspondent aux classes contenant des images avec des dimensions variées (Exemples: *hat*, *butterfly*, *bird*,

*turtle...*). Le CBIR arrive à un taux  $RR\%$  d'environ  $50\%$  lorsque les objets ne sont pas centrés ou n'ont pas les mêmes dimensions entre les images mais les dimensions d'images sont à peu près égales (Exemples: *spoon, octopus, bone, jar...*). Lorsque l'objet est bien centré et les images ont la même dimension, le CBIR réussit à atteindre un taux  $RR\%$  de  $93\%$ , ce qui est le cas pour les dix premières classes du Tableau 12 (Comme *device7, children, teddy* et *fountain*). On conclut de cette expérimentation que les moments de Zernike sont uniquement invariants par rapport à la rotation. Pour obtenir une invariance par rapport à l'échelle et à la translation, l'image doit subir une étape de normalisation avant de calculer les moments.

Pour la base de données d'images VisTex, on a décidé d'extraire les descripteurs de Zernike sur des images préalablement transformées en binaires par le recours à la méthode d'Otsu. Cette méthode effectue un seuillage sur l'image pour convertir l'image à niveaux de gris en une image binaire. Ce prétraitement a permis d'augmenter le Taux de Réussite de la recherche ( $RR\%$ ) de  $10\%$  environ.

Le Tableau 13 récapitule le Taux de Réussite de la Recherche ( $RR\%$ ) obtenu pour toutes les classes de la base de données VisTex et cela en utilisant des descripteurs de Zernike correspondant à six différents ordres  $n$ , soit 5, 10, 15, 20, 30 et 40. On remarque, que Zernike arrive à avoir de bons résultats quand la texture exhibe des formes d'objets et des motifs (Exemple : *Buildings.0009*) au lieu de textures non prononcées (*Tile.0007*) mais on remarque aussi que le meilleur taux de succès de la recherche ( $RR\%$ ) global est de  $36.17\%$  pour l'ordre 10 ce qui reste faible par rapport à nos résultats obtenus pour la base de données MPEG7-CE Shape-1.

		Taux de réussite de la recherche (RR%)					
		Ordre 5	Ordre 10	Ordre 15	Ordre 20	Ordre 30	Ordre 40
<b>Classe d'image</b>	<b>Metal.0000</b>	93	100	100	100	100	100
	<b>Fabric.0017</b>	76	86	97	83	66	52
	<b>Fabric.0018</b>	78	85	93	83	59	35
	<b>Tile.0007</b>	100	78	69	70	73	88
	<b>Metal.0002</b>	68	64	61	54	35	36
	<b>Buildings.0009</b>	43	57	58	66	59	73
	<b>Food.0005</b>	50	49	57	52	38	35
	<b>Fabric.0014</b>	37	48	47	47	44	40
	<b>Fabric.0009</b>	36	44	46	52	53	46
	<b>Misc.0002</b>	35	48	46	40	22	25
	<b>Tile.0001</b>	52	45	45	33	32	30
	<b>Water.0005</b>	46	46	40	39	34	34
	<b>Terrain.0010</b>	40	40	38	42	48	53
	<b>Wood.0002</b>	25	35	36	33	33	29
	<b>Fabric.0007</b>	42	45	36	41	34	20
	<b>Stone.0004</b>	24	27	31	32	29	32
	<b>Leaves.0016</b>	45	43	31	28	31	30
	<b>Tile.0004</b>	37	34	30	30	29	33
	<b>Fabric.0000</b>	25	27	29	37	44	46
	<b>Food.0000</b>	39	31	29	31	29	28
	<b>Brick.0005</b>	23	27	29	30	27	24
	<b>Grass.0001</b>	16	22	28	25	20	19
	<b>Leaves.0010</b>	30	21	27	39	23	21
	<b>Sand.0000</b>	20	25	26	22	26	25
	<b>Brick.0004</b>	16	21	25	28	38	39
	<b>Bark.0008</b>	25	26	25	25	24	23
	<b>Stone.0001</b>	26	24	24	21	21	25
	<b>Leaves.0008</b>	19	25	24	20	20	18
<b>Bark.0006</b>	23	25	23	20	21	21	

	<b>Leaves.0012</b>	23	22	23	20	17	17
	<b>Fabric.0004</b>	15	19	21	21	20	21
	<b>Fabric.0011</b>	23	18	20	20	24	18
	<b>Bark.0009</b>	33	23	18	18	20	19
	<b>Brick.0001</b>	20	18	18	16	16	17
	<b>Wood.0001</b>	17	17	18	17	16	16
	<b>Bark.0000</b>	22	20	17	17	16	18
	<b>Flowers.0005</b>	19	16	17	17	18	16
	<b>Leaves.0011</b>	25	21	16	21	21	21
	<b>Food.0008</b>	14	14	12	13	13	18
	<b>Fabric.0015</b>	13	11	11	12	14	16
	<b>RR% (global)</b>	35.32	36.17	36.02	35.37	32.67	31.92

Tableau 13 : Performance du CBIR (*RR%*) pour une recherche Top-16 effectuée pour les 40 classes d'images VisTex. Six ordres  $n$  sont utilisés pour les descripteurs de Zernike.

#### 4.4.3. La normalisation des images préalablement à l'extraction des descripteurs

Comme on a pu constater lors de l'évaluation de la performance du CBIR, la taille de l'objet et sa position dans l'image jouent un grand rôle dans le calcul des descripteurs de Zernike. Par conséquent, une opération de normalisation de l'image s'avère nécessaire avant de calculer ses moments de Zernike. La normalisation comme le montre le tableau ci-dessous s'effectue en quatre étapes :

1. Mettre tous les objets au centre de l'image en enlevant toutes les bordures
2. Redimensionner en respectant le rapport de format (aspect ratio)
3. Mettre l'objet dans le centre d'une image 128×128
4. Rajouter une bordure suffisante dans les quatre cotés afin que tous les objets soient à l'intérieur du cercle unitaire et que l'ensemble de l'objet soit entièrement pris en compte lors du calcul des moments de Zernike.

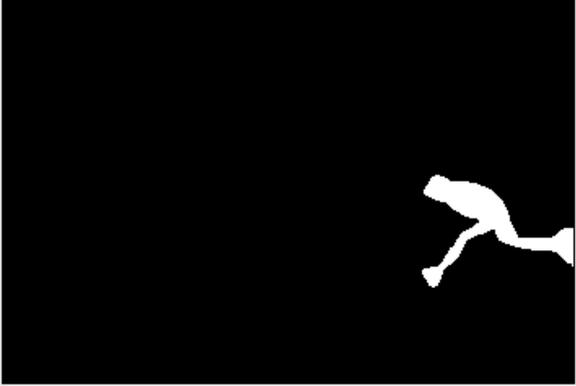
Étape	Image	Dimensions en pixel
<b>Image Originale</b>		352x240
<b>1</b>		93x70
<b>2</b>		
<b>3</b>		128x128
<b>4</b>		184x184

Tableau 14 : Étapes du processus de normalisation des images.

		Taux de réussite de la recherche ( <i>RR%</i> )					
		Zernike			Zernike + normalisation		
Dimensions	Classe d'image	Ordre 5	Ordre 10	Ordre 15	Ordre 5	Ordre 10	Ordre 15
different 130-613 × 201-660	<b>beetle</b>	14	9	9	35	43	40
different 277-704 × 182-705	<b>butterfly</b>	12	10	9	44	49	41
different 137-392 × 161-436	<b>sea_snake</b>	14	11	10	46	43	39
different 317-324 × 289-361	<b>Glas</b>	52	55	60	72	72	71
different 366-520 × 146-213	<b>cellular_phone</b>	39	58	45	95	91	93
558×558	<b>device9</b>	77	78	78	100	99	99
512×512	<b>device1</b>	25	90	83	70	98	93
558×558	<b>device8</b>	77	95	87	55	56	63
352×288	<b>rat</b>	88	96	94	89	88	88
352×240	<b>children</b>	100	100	100	80	82	83
360×243	<b>teddy</b>	100	100	100	100	100	100

Tableau 15 : Taux de réussite de la recherche (*RR%*) en appliquant la normalisation pour quelques images de la base de données MPEG7-CE Shape-1.

Le tableau de l'annexe B compare le taux de réussite de la recherche (*RR%*) de deux CBIR basés sur les moments de Zernike pour l'extraction des descripteurs tout en appliquant la normalisation des images dans le cas du deuxième. Vu que toutes les images

ont maintenant la même taille et que l'objet est centré au milieu, le taux de réussite (global) est passé de 43.0% à 55.8% pour l'ordre 10 et de 40.6% à 53.9% pour l'ordre 15. On remarque aussi, dans les résultats du tableau 15, que pour des classes comme *cellular\_phone* ou *pocket*, le taux de réussite a doublé vu que toutes les images ont maintenant la même taille et l'objet recherché a les mêmes dimensions. Cependant la normalisation a engendré une baisse du taux de recherche pour quelques classes (comme *device3* et *device4*) mais en général on peut conclure que la normalisation a significativement amélioré le taux de réussite de la recherche effectuée par le CBIR.

#### 4.4.4. L'incorporation de la transformée en contourlettes redondantes (RCT)

La construction de descripteurs RCT caractéristiques à partir de la décomposition en contourlettes redondantes d'une image peut se faire par l'extraction de primitives d'énergie dans les sous-bandes. Ceci consiste à calculer l'énergie (norme  $L_1$ , norme  $L_2$  ou leur combinaison) et à en caractériser la distribution à travers les sous-bandes multi-résolution de l'image décomposée. L'approche basée sur l'énergie suppose que différentes images ont une distribution d'énergie différente dans le domaine fréquentiel. Utilisant essentiellement le calcul des moments de premier et second ordre des coefficients contourlettes, cette approche est très intéressante en raison de sa faible complexité de calcul [31].

Étant donné une décomposition RCT d'une image avec  $L$  niveaux de résolution et  $D$  sous-bandes directionnelles  $C_{ld}$  à chaque niveau, un vecteur descripteur  $E$  est formé pour contenir les caractéristiques d'énergies de l'image, calculées à partir des coefficients des sous-bandes  $C_{ld}$  de taille  $K \times M$  [31]. Il est défini selon les formules suivantes :

$$F(l, d) = \sqrt{\frac{1}{KM} \sum_{i=1}^K \sum_{j=1}^M [C_{ld}(i, j)]^2} \quad (4.1)$$

$$E_2(l, d) = (E_1(l, d), F(l, d)), \quad (4.2)$$

$$E_1(l, d) = \frac{1}{KM} \sum_{i=1}^K \sum_{j=1}^M |C_{ld}(i, j)|, \quad (4.3)$$

$$E = \{E_2(l, d); \quad l = 1 \dots L; d = 1 \dots D\} \quad (4.4)$$

Il est à noter que pour une image décomposée en RCT à 3 niveaux ( $L=3$  et  $D=4$ ), la taille du descripteur RCT désigné par  $E$  est de 24 composantes à valeurs réelles. Avec des descripteurs RCT, deux images peuvent être comparées à l'aide d'une mesure de similarité entre ces descripteurs. La mesure de similarité entre deux vecteurs descripteurs RCT peut être effectuée en calculant la distance Euclidienne entre les deux vecteurs  $E$  correspondants.

L'application de la normalisation d'image préalablement à l'extraction des descripteurs de Zernike a permis d'améliorer la performance du CBIR en termes de taux de réussite de la recherche. Avant de développer un descripteur hybride, on se propose de mesurer l'efficacité d'un descripteur RCT directement sur les images de la base MPEG7-CE Shape-1 et VisTex. La RCT (Annexe D) réussit bien à rechercher et récupérer les images de textures dans la base VisTex avec un taux de réussite de 70% pour un Top-16 et un taux de réussite de 40% dans la base MPEG7-CE Shape-1. Ce qui est un taux plus faible comparé au CBIR basé sur Zernike mais reste un bon résultat par rapport au temps de calcul nécessaire et à la taille des descripteurs (Un descripteur extrait d'une RCT à 3 niveaux comprend 24 composantes d'énergie ou valeurs réelles). Il est à noter que la normalisation des images n'a pas eu d'impact important sur les descripteurs RCT et le

CBIR résultant contrairement au cas Zernike. Cela peut s'expliquer par la différence entre les deux méthodes face au changement d'échelle et translation.

#### 4.4.5. Descripteur hybride Zernike et RCT

Comme on a testé l'approche RCT à part puis on a amélioré l'approche Zernike en ajoutant une étape de normalisation, l'objectif poursuivi est de mettre en coopération les deux approches, dériver des méthodes hybrides et mesurer le taux de réussite pour différents schémas CBIR. La première méthode (voir la Figure 12) consiste à construire un vecteur descripteur hybride par concaténation  $frz=[fz; frct]$  où  $frct$  représente le vecteur descripteur RCT et  $fz$  le vecteur descripteur Zernike correspondant à un ordre  $n$  (5, 10 ou 15). La deuxième méthode (voir la Figure 13) consiste à effectuer une recherche d'images basée sur chacun des descripteurs RCT et Zernike, de manière séparée, puis de procéder à une union des résultats de recherche obtenus.

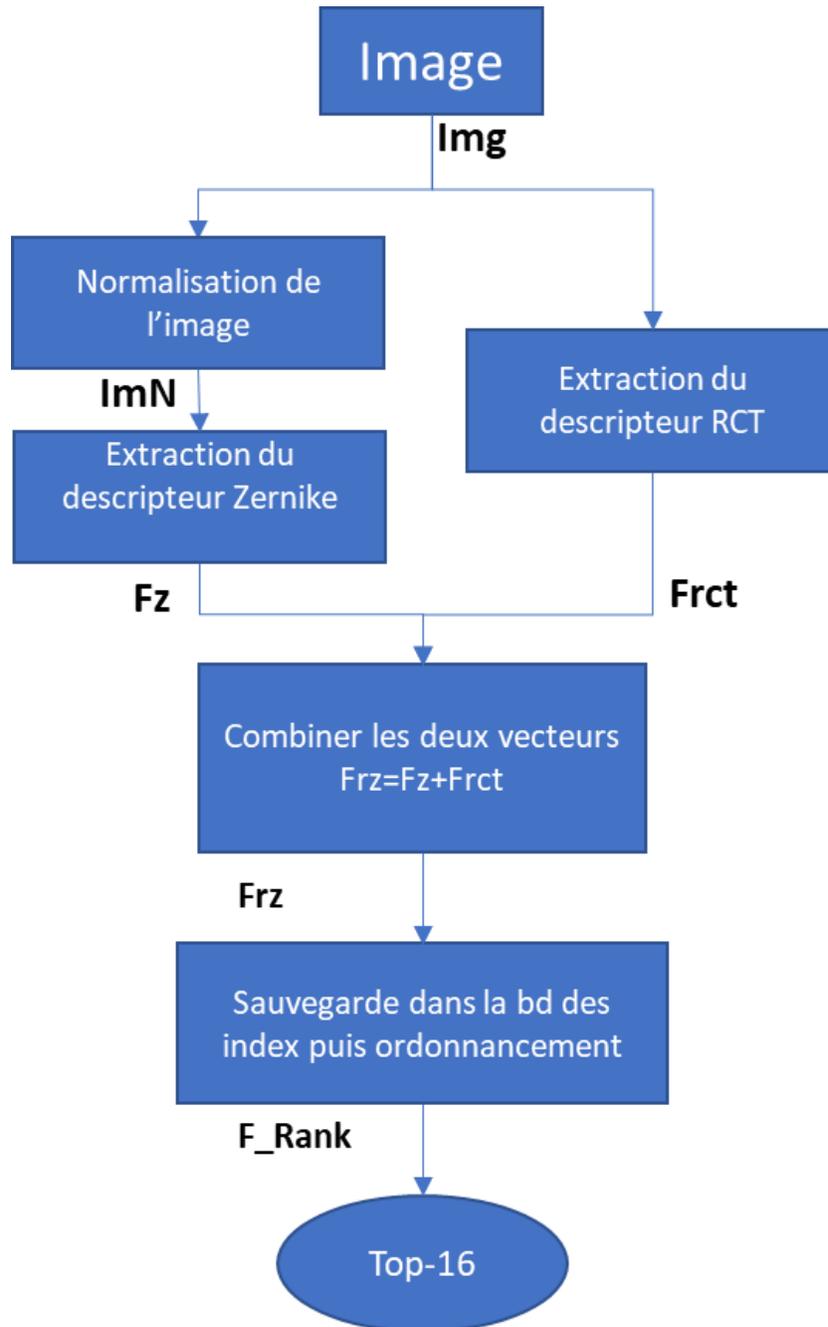


Figure 12 : Descripteur hybride obtenu par concaténation des descripteurs RCT et Zernike.

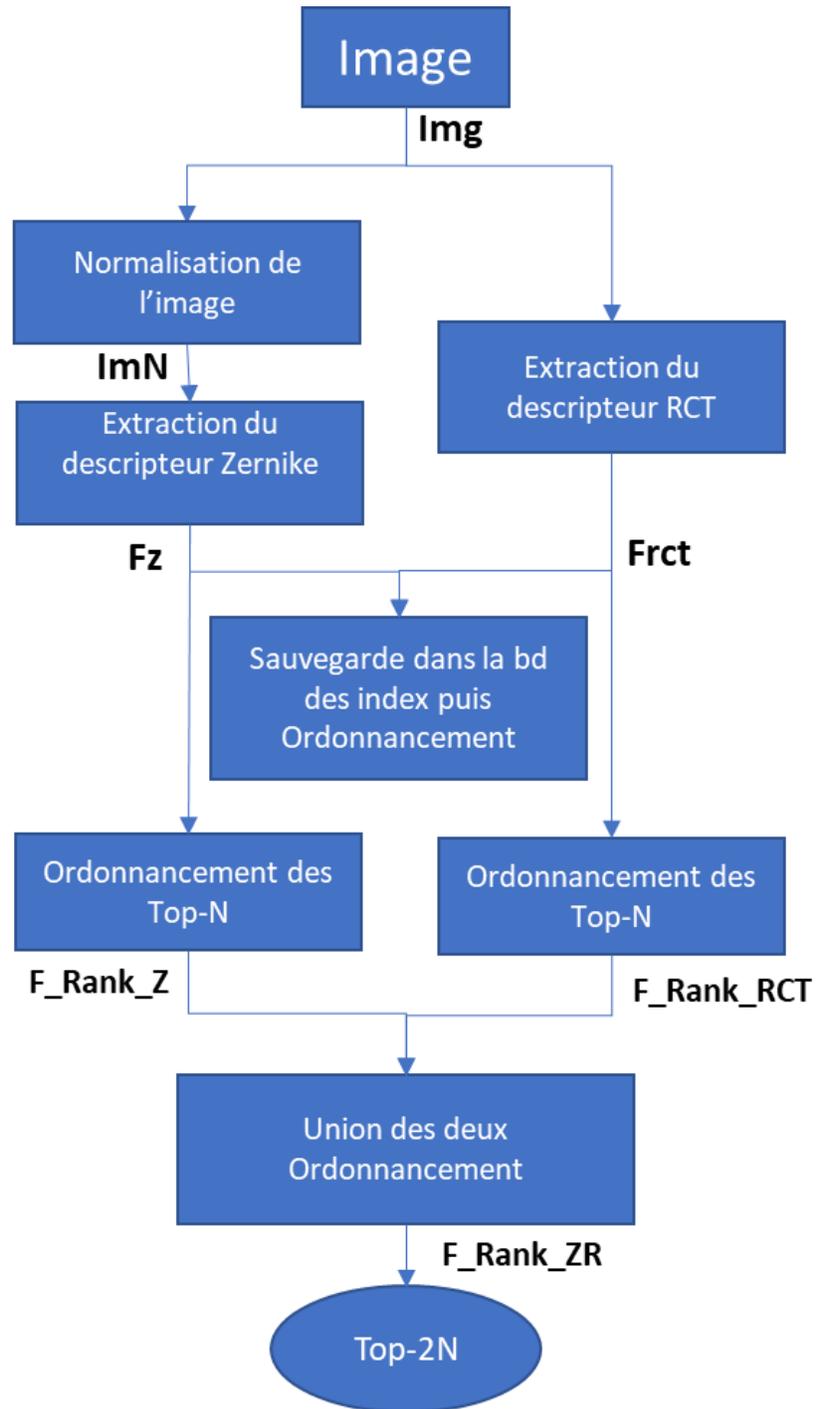


Figure 13 : Méthode de recherche hybride exploitant l'union des résultats basés sur chacun descripteurs RCT et Zernike.

Méthode	Taux de réussite de la recherche $RR\%$ (global)					
	BD: MPEG7-CE Shape-1			BD: VisTex		
	Ordre 5	Ordre 10	Ordre 15	Ordre 5	Ordre 10	Ordre 15
Zernike	53.7	55.8	53.9	35.32	36.17	36.02
RCT	40.2			70.4		
Zernike+RCT (1)	53.76	55.90	54.07	66.08	58.53	53.83
Union Zernike-RCT (2)	55.84	62.28	63.93	74.80	74.78	74.98

Tableau 16: Tableau comparatif de la performance de systèmes CBIR basés sur 4 méthodes distinctes, Zernike, RCT, Concaténation Zernike+RCT et Union Zernike-RCT. La recherche est effectuée sur les bases de données MPEG7-CE Shape-1 et VisTex. Les taux de réussite ( $RR\%$ ) correspondent à trois ordre  $n$  distincts 5, 10 et 15.

Les deux méthodes hybrides présentent des améliorations par rapport à l'approche Zernike toute seule. En effet, toutes les deux arrivent à un taux de réussite supérieur à 55% (Annexes D et E). La deuxième méthode hybride qui consiste à assembler et unir les résultats de recherche RCT et Zernike constitue le meilleur choix car elle rassemble les forces de chaque approche RCT et Zernike. Ainsi, plus d'images d'une même classe sont retournées et le taux de réussite global  $RR\%$  se trouve significativement augmenté (atteint les 64% dans le cas de la base MPEG7-CE Shape-1 et 75% pour VisTex). Le tableau 16 dresse un récapitulatif des résultats obtenus facilitant ainsi la comparaison de la performance en termes des valeurs  $RR\%$ . Il est aussi pertinent de remarquer que les classes pour lesquelles le taux de réussite a énormément augmenté (*jar, camel, fork, stef,..*) sont des classes où les formes d'objet contenu sont complexes. Pour la base VisTex, Les méthodes hybrides ont permis de bien identifier les images de textures pertinentes que Zernike ne pouvait pas retourner (*Wood.0002, Water.005, Metal.0002, Fabric.0014, ...*). En utilisant la deuxième méthode, le résultat s'améliore encore plus et on arrive à un taux de réussite qui avoisine les 75% dès l'ordre 5. On remarque aussi que

l'approche RCT apporte la puissance de détection des textures (*Wood.0002*, *Water.005*, *Metal.0002*, ..) alors que Zernike performe plus sur les images avec des formes ou textures géométriques (*Fabric.0000*, *Food.0000*, *Tile.0007*).

On peut alors conclure de ces expérimentations que l'incorporation de l'approche par descripteur RCT améliore de manière significative les performances d'un CBIR initialement basé sur les descripteurs Zernike.

## 5 CONCLUSION

Dans le cadre de ce mémoire, la recherche bibliographique nous a permis de comprendre les concepts de base des moments de Zernike et de prendre connaissance de l'état de l'art en matière de recherche d'images par le contenu basée sur les moments de Zernike. À cela, s'est ajouté tout un travail de développement et de simulation qui a permis de tester le calcul des moments de Zernike sur une image et sa reconstruction, l'extraction de descripteurs de Zernike à différents ordres ainsi que la mise au point d'un prototype de simulation CBIR basé sur les moments de Zernike. Les résultats de recherche d'images basée sur les moments de Zernike sont obtenus par rapport à deux bases de données : MPEG7-CE Shape-1 et VisTex et ont montré que le moment de Zernike n'est pas invariant au changement de position ou de l'échelle et n'arrive pas à performer sur des textures non géométriques. Pour y remédier, une étape de normalisation est appliquée aux images de la base MPEG7-CE Shape-1 afin que les objets soient centrés et les images aient toutes les mêmes dimensions. Ainsi, la normalisation a contribué à améliorer la performance du CBIR et a donné lieu à des taux de réussite de la recherche qui dépassent les 50%. La deuxième contribution a consisté en l'incorporation de la transformée en contourlettes redondante (RCT) et a donné lieu à deux méthodes hybrides distinctes qui exploitent conjointement les moments de Zernike et des primitives d'énergie extraites de la RCT. Les expérimentations effectuées ont permis de montrer, de manière quantitative, des améliorations significatives au niveau de la performance de recherche du CBIR notamment lorsque la méthode d'union des résultats de recherche Zernike-RCT est utilisées. Comme perspective de recherche, d'autres schémas de coopération des deux approches Zernike et RCT peuvent être élaborés.

## Annexe A

### RECONSTRUCTION D'IMAGE À PARTIR DES MOMENTS DE ZERNIKE

```
close all;

% Image test
im=imread('./MPEG7dataset/apple-7.gif');
close all;

imshow(im, []); % Afficher l'image originale

dim = size(im); % Dimensions originales requises pour la reconstruction

% Ordre Zernike utilisé

order = 0:30;

disp('Calcul des moments de Zernike');

% Différent calcul si l'image est RGB ou gray
if ndims(im) == 3
    imR = im(:,:,1);
    imG = im(:,:,2);
    imB = im(:,:,3);

    % Calcul des moments de Zernike
    [A_nmR, zmlistR, cidxB, V_nmR] = lans_zmoment(imR, order);
    [A_nmG, zmlistG, cidxB, V_nmG] = lans_zmoment(imG, order);
    [A_nmB, zmlistB, cidxB, V_nmB] = lans_zmoment(imB, order);

    figure,
```

```

%---Reconstruction à partir des moments de Zernike calculés

disp('Reconstruction de l''image basée sur les moments de Zernike');

[reimgR] = lans_zmrecon(A_nmR,V_nmR,zmlistR,cidxR,dim(1:2),50,1);
[reimgG] = lans_zmrecon(A_nmG,V_nmG,zmlistG,cidxG,dim(1:2),50,1);
[reimgB] = lans_zmrecon(A_nmB,V_nmB,zmlistB,cidxB,dim(1:2),50,1);

% Afficher l'image reconstruite

reimgRi=uint8(reimgR);

reimgGi=uint8(reimgG);

reimgBi=uint8(reimgB);

REIMG=cat(3,reimgRi, reimgGi, reimgBi);

imshow(REIMG, []);

im = double(im);

% Calcul des moments de Zernike

[A_nm,zmlist,cidx,V_nm] = lans_zmoment(im,order);

figure,

%---Reconstruction à partir des moments de Zernike calculés

disp('Reconstruction de l''image basée sur les moments de Zernike');

[reimg] = lans_zmrecon(A_nm,V_nm,zmlist,cidx,dim,50,1);

% Afficher l'image reconstruite

imshow(reimg, []);

end

```

## Annexe B

### TAUX DE REUSSITE DE LA RECHERCHE (RR%) AVEC L'APPLICATION DE LA NORMALISATION AUX IMAGES PRÉALABLEMENT À L'EXTRACTION DES DESCRIPTEURS DE ZERNIKE SUR LA BASE MPEG7

		Taux de réussite de la recherche (RR%) pour un Top-16					
		Zernike			Zernike + normalisation de l'image		
Dimensions	Classe d'image	Ordre 5	Ordre 10	Ordre 15	Ordre 5	Ordre 10	Ordre 15
different 138-715 × 142-739	<i>bird</i>	11	12	9	17	17	18
different 98-412 × 98-396	<i>chicken</i>	17	13	13	25	25	24
different 264-684 × 248-479	<i>lizzard</i>	13	14	13	20	25	24
different 230-284 × 359-412	<i>Comma</i>	64	52	48	26	25	27
different 184-346 × 197-311	<i>fly</i>	20	13	12	31	27	24

different 211-884 × 144-875	<i>horse</i>	14	12	12	28	28	29
different 395-851 × 168-819	<i>bat</i>	15	15	15	26	28	24
different 150-686 × 130-535	<i>dog</i>	27	20	20	31	29	29
different 71-633 × 38-521	<i>crown</i>	28	33	30	24	29	32
different 198-428 × 175-530	<i>frog</i>	23	39	37	27	29	31
different 303-545 × 149-427	<i>turtle</i>	11	12	12	27	30	30
different 174-600 × 56-459	<i>fish</i>	26	18	20	26	30	28
different 239-522 × 170-359	<i>pencil</i>	21	17	13	34	31	28
256×256	<i>spoon</i>	39	43	34	26	32	31
different 324-437 × 307-363	<i>camel</i>	33	22	21	40	35	29
256×256 & 7320×371	<i>hammer</i>	74	62	66	39	35	37

different 91-437 × 146-509	<i>guitar</i>	17	16	13	34	36	38
different 333-984 × 249-986	<i>deer</i>	18	16	15	28	37	38
different 228-654 × 309-670	<i>ray</i>	21	21	18	42	37	41
256×256	<i>octopus</i>	39	25	27	34	37	36
different 422-802 × 272-765	<i>cattle</i>	24	22	20	35	38	34
different 147-1116 × 111-658	<i>lmfish</i>	13	16	16	37	39	52
different 59-360 × 61-334	<i>horseshoe</i>	29	21	17	27	39	37
different 317-351 × 355-405	<i>Misk</i>	62	48	38	26	39	37
different 50-477 × 48-286	<i>hat</i>	12	14	14	43	41	35
different 130-613 × 201-660	<i>beetle</i>	14	9	9	35	43	40

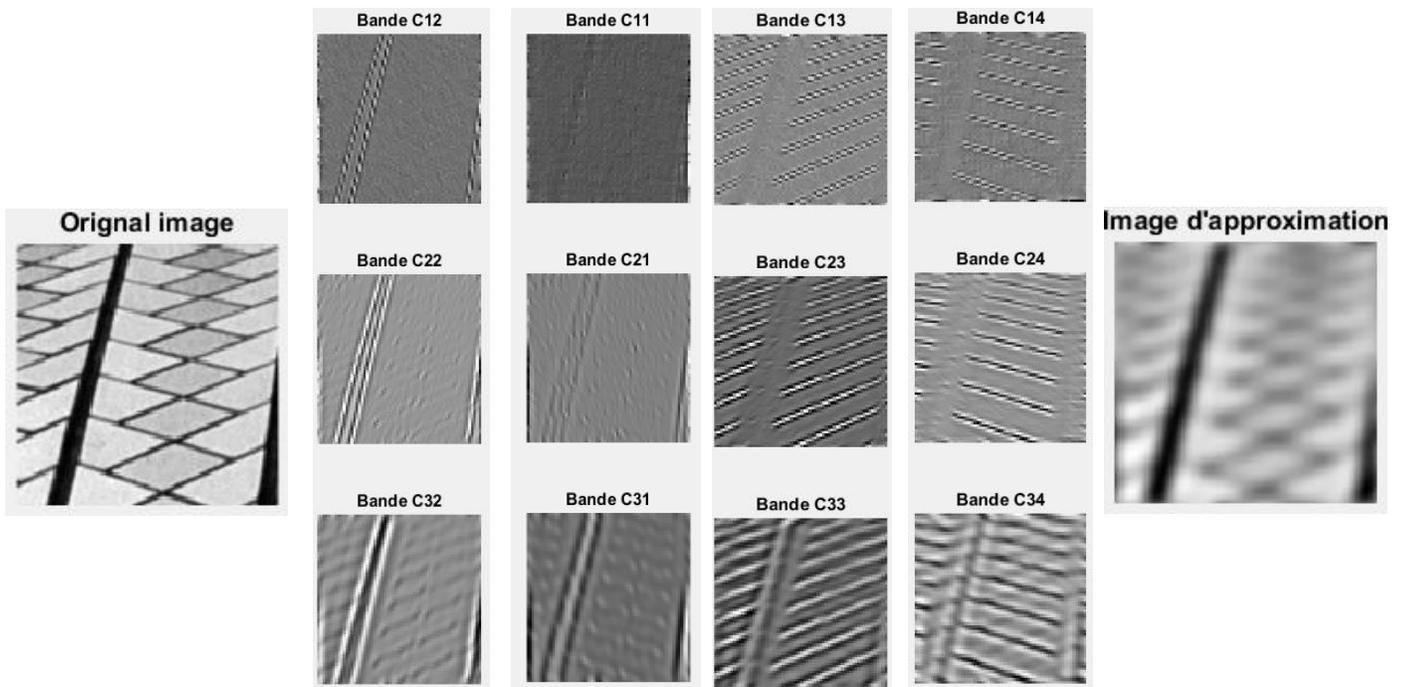
different 137-392 × 161-436	<i>sea_snake</i>	14	11	10	46	43	39
different 176-640 × 121-582	<i>spring</i>	23	15	18	52	43	41
different 298-998 × 289-981	<i>elephant</i>	39	20	17	38	43	27
different 335-357 × 313-439	<i>Bone</i>	46	37	33	51	43	45
different 208-425 × 256-414	<i>tree</i>	53	50	38	41	45	36
352×240	<i>stef</i>	90	60	51	68	46	39
different 277-704 × 182-705	<i>butterfly</i>	12	10	9	44	49	41
different 309-1110 × 145-488	<i>classic</i>	18	14	15	38	49	48
512×512	<i>jar</i>	40	52	41	56	49	39
640×480 & 297×650	<i>fork</i>	34	31	30	45	50	46
different 470-990 × 160-378	<i>watch</i>	25	21	20	39	52	47
352×288	<i>brick</i>	50	55	51	57	54	50
352×288	<i>chopper</i>	71	66	79	55	55	50
558×558	<i>device4</i>	78	82	75	44	56	50

558×558	<i>device8</i>	77	95	87	55	56	63
558×558	<i>device3</i>	70	80	73	57	62	46
different 400-559 × 203-261	<i>personal_ca r</i>	27	21	20	66	64	55
different 404-551 × 237-307	<i>shoe</i>	41	35	21	62	64	65
different 379-479 × 209-276	<i>key</i>	55	43	31	81	65	57
different 59-320 × 64-420	<i>bell</i>	14	23	23	51	66	73
558×558	<i>device6</i>	37	50	55	57	68	71
different 420-613 × 166-492	<i>flatfish</i>	39	38	38	74	69	66
different 317-324 × 289-361	<i>Glas</i>	52	55	60	72	72	71
different 360-464 × 318-384	<i>Heart</i>	84	57	49	65	75	74
different 64-364 × 90-534	<i>pocket</i>	18	17	15	62	77	71
different 166-221 × 315-362	<i>HCircle</i>	85	76	73	72	80	79

352×240	<i>children</i>	100	100	100	80	82	83
352×288	<i>carriage</i>	68	42	50	88	86	81
256×256	<i>apple</i>	82	66	55	88	86	79
558×558	<i>device5</i>	39	66	68	72	88	83
352×288	<i>rat</i>	88	96	94	89	88	88
352×288	<i>car</i>	84	79	72	86	89	89
352×288	<i>bottle</i>	89	86	74	97	89	94
different 315-443 × 296-392	<i>cup</i>	30	50	32	83	90	82
512×512	<i>device0</i>	43	70	80	75	90	96
different 366-520 × 146-213	<i>cellular_phone</i>	39	58	45	95	91	93
352×288	<i>truck</i>	73	69	71	95	92	84
269×337 204×331	<i>face</i>	63	63	63	96	96	100
512×512	<i>device1</i>	25	90	83	70	98	93
360×243	<i>fountain</i>	89	93	100	99	98	100
558×558	<i>device9</i>	77	78	78	100	99	99
360×243	<i>teddy</i>	100	100	100	100	100	100
	<b>RR%</b> (global)	43.6	43.0	40.6	53.7	55.8	53.9

## Annexe C

### DÉCOMPOSITION MULTI-RÉSOLUTION ET DIRECTIONNELLE RCT (3 NIVEAUX DE RÉOLUTION, 4 DIRECTIONS)



## Annexe D

### TAUX DE RESUSSITE (RR%) SUR UN TOP-16 DE LA RCT SUR LA BASE MPEG7 ET VISTEX

Classe d'image	RCT-3 Niveaux	Classe d'image	RCT-3 Niveaux
<i>Comma</i>	49	<i>Buildings.0009</i>	82
<i>Misk</i>	12	<i>Bark.0000</i>	45
<i>hammer</i>	23	<i>Bark.0006</i>	35
<i>device4</i>	21	<i>Bark.0008</i>	47
<i>stef</i>	9	<i>Bark.0009</i>	31
<i>device8</i>	15	<i>Brick.0001</i>	79
<i>children</i>	16	<i>Brick.0004</i>	93
<i>Heart</i>	36	<i>Brick.0005</i>	65
<i>chopper</i>	24	<i>Fabric.0000</i>	51
<i>spoon</i>	32	<i>Fabric.0004</i>	68
<i>device3</i>	51	<i>Fabric.0007</i>	79
<i>HCircle</i>	29	<i>Fabric.0009</i>	89
<i>tree</i>	15	<i>Fabric.0011</i>	88
<i>octopus</i>	32	<i>Fabric.0014</i>	100
<i>crown</i>	13	<i>Fabric.0015</i>	59
<i>horseshoe</i>	25	<i>Fabric.0017</i>	100
<i>elephant</i>	24	<i>Fabric.0018</i>	89
<i>teddy</i>	17	<i>Flowers.0005</i>	69
<i>fish</i>	27	<i>Food.0000</i>	41
<i>rat</i>	22	<i>Food.0005</i>	48
<i>car</i>	11	<i>Food.0008</i>	73
<i>dog</i>	76	<i>Grass.0001</i>	80
<i>frog</i>	59	<i>Leaves.0008</i>	66
<i>Bone</i>	74	<i>Leaves.0010</i>	29
<i>apple</i>	34	<i>Leaves.0011</i>	57
<i>bird</i>	77	<i>Leaves.0012</i>	61
<i>brick</i>	86	<i>Leaves.0016</i>	59

<i>lizzard</i>	26
<i>camel</i>	80
<i>bottle</i>	100
<i>chicken</i>	80
<i>fountain</i>	21
<i>deer</i>	89
<i>fork</i>	25
<i>fly</i>	45
<i>bat</i>	13
<i>cattle</i>	27
<i>pencil</i>	11
<i>horse</i>	63
<i>watch</i>	49
<i>jar</i>	52
<i>turtle</i>	24
<i>guitar</i>	46
<i>carriage</i>	94
<i>device6</i>	17
<i>classic</i>	74
<i>Glas</i>	18
<i>beetle</i>	16
<i>ray</i>	86
<i>shoe</i>	56
<i>truck</i>	11
<i>device9</i>	21
<i>lmfish</i>	28
<i>key</i>	13
<i>spring</i>	68
<i>hat</i>	37
<i>device0</i>	33
<i>sea_snake</i>	34
<i>butterfly</i>	90
<i>face</i>	17
<i>device5</i>	13
<i>flatfish</i>	34
<i>bell</i>	23
<i>personal_car</i>	46

<i>Metal.0000</i>	86
<i>Metal.0002</i>	100
<i>Misc.0002</i>	89
<i>Sand.0000</i>	69
<i>Stone.0001</i>	71
<i>Stone.0004</i>	97
<i>Terrain.0010</i>	38
<i>Tile.0001</i>	80
<i>Tile.0004</i>	61
<i>Tile.0007</i>	99
<i>Water.0005</i>	100
<i>Wood.0001</i>	46
<i>Wood.0002</i>	95
<b>RR% (global)</b>	<b>70.4</b>

<i>pocket</i>	95
<i>device1</i>	99
<i>cup</i>	16
<i>cellular_phone</i>	36
<b>RR% (Global)</b>	40.2

## Annexe E

### TAUX DE REUSSITE (RR%) SUR TOP-16 POUR LA COMBINAISON DE ZERNIKE ET RCT SUR LA BASE MPEG7

Classe d'Image	Taux de réussite de la recherche (RR%)					
	Descripteur hybride Zernike+RCT			UNION Zernike-RCT		
	Ordre 5	Ordre 10	Ordre 15	Ordre 5	Ordre 10	Ordre 15
<i>apple</i>	88	86	80	93	93	93
<i>bat</i>	26	28	24	25	30	34
<i>beetle</i>	35	42	41	28	43	43
<i>bell</i>	52	67	72	63	73	71
<i>bird</i>	17	16	18	31	27	30
<i>butterfly</i>	43	48	42	21	35	38
<i>chicken</i>	25	25	26	36	33	33
<i>device0</i>	75	90	96	57	87	95
<i>device1</i>	70	98	93	65	79	95
<i>device3</i>	58	61	46	96	98	96
<i>device4</i>	43	56	51	38	46	49
<i>device5</i>	72	88	83	58	99	98
<i>device6</i>	56	68	70	44	47	48
<i>device8</i>	55	56	63	55	57	57
<i>device9</i>	100	99	99	94	97	97
<i>hammer</i>	38	36	37	53	61	66
<i>jar</i>	56	49	39	51	50	52
<i>lmfish</i>	36	39	52	24	34	35
<i>octopus</i>	34	37	36	32	36	36
<i>spoon</i>	26	32	32	34	39	38
<i>tree</i>	41	45	36	30	32	32
<i>Bone</i>	51	45	44	71	88	91
<i>bottle</i>	97	89	94	85	95	94

<i>brick</i>	57	54	49	70	76	81
<i>camel</i>	39	35	29	47	56	53
<i>car</i>	86	89	89	91	94	96
<i>carriage</i>	88	86	81	100	100	100
<i>cattle</i>	35	38	35	34	43	47
<i>cellular_phone</i>	95	91	93	94	96	97
<i>children</i>	80	82	83	100	100	100
<i>chopper</i>	55	55	50	75	79	83
<i>classic</i>	38	48	48	42	54	55
<i>Comma</i>	25	25	28	42	50	53
<i>crown</i>	23	27	30	43	52	54
<i>cup</i>	83	89	83	69	75	75
<i>deer</i>	27	38	38	39	38	39
<i>dog</i>	31	29	29	36	39	39
<i>elephant</i>	39	43	27	39	48	54
<i>face</i>	96	96	100	99	98	100
<i>fish</i>	27	29	28	35	50	48
<i>flatfish</i>	75	70	68	58	63	70
<i>fly</i>	30	27	25	36	44	43
<i>fork</i>	45	50	47	62	63	66
<i>fountain</i>	99	98	100	100	100	100
<i>frog</i>	28	29	31	36	48	46
<i>Glas</i>	73	72	71	90	88	89
<i>guitar</i>	34	36	38	38	52	51
<i>hat</i>	46	43	34	29	29	29
<i>HCircle</i>	70	79	79	57	55	61
<i>Heart</i>	66	74	75	63	73	75
<i>horse</i>	28	27	29	30	35	46
<i>horseshoe</i>	27	38	34	33	66	64
<i>key</i>	81	65	57	62	75	72
<i>lizzard</i>	20	25	24	32	37	37
<i>Misk</i>	25	38	36	50	51	58
<i>pencil</i>	34	30	28	56	60	63
<i>personal_car</i>	66	64	56	52	61	59
<i>pocket</i>	62	77	71	38	52	59
<i>rat</i>	89	88	88	88	88	88
<i>ray</i>	42	37	40	40	34	36
<i>sea_snake</i>	46	43	39	36	41	41

<i>shoe</i>	68	72	73	58	60	64
<i>spring</i>	53	43	41	63	63	65
<i>stef</i>	68	46	39	66	69	69
<i>teddy</i>	100	100	100	100	100	100
<i>truck</i>	95	93	84	100	100	100
<i>turtle</i>	28	30	30	22	33	34
<i>watch</i>	40	53	46	63	68	67
<b>RR% (global)</b>	53.76	55.90	54.07	55.84	62.28	63.93

## Annexe F

### TAUX DE REUSSITE TOP-16 POUR LA COMBINAISON DE ZERNIKE ET RCT SUR LA BASE VISTEX

Classe d'image	Taux de réussite de la recherche (RR%)					
	Descripteur hybride Zernike+RCT			Union Zernike-RCT		
	Ordre 5	Ordre 10	Ordre 15	Ordre 5	Ordre 10	Ordre 15
<i>Buildings.0009</i>	71	61	68	89	89	89
<i>Bark.0000</i>	46	30	28	46	46	45
<i>Bark.0006</i>	49	41	40	43	44	42
<i>Bark.0008</i>	41	39	39	50	51	51
<i>Bark.0009</i>	25	21	21	48	38	36
<i>Brick.0001</i>	48	43	45	81	80	80
<i>Brick.0004</i>	64	54	47	93	93	94
<i>Brick.0005</i>	61	51	38	69	69	70
<i>Fabric.0000</i>	46	46	45	58	59	61
<i>Fabric.0004</i>	67	67	65	68	68	69
<i>Fabric.0007</i>	93	94	89	81	81	83
<i>Fabric.0009</i>	83	65	58	91	91	93
<i>Fabric.0011</i>	75	54	44	88	88	88
<i>Fabric.0014</i>	100	84	58	100	100	100
<i>Fabric.0015</i>	32	24	25	59	60	60
<i>Fabric.0017</i>	100	100	100	100	100	100
<i>Fabric.0018</i>	95	94	92	95	97	97
<i>Flowers.0005</i>	53	34	34	70	70	70
<i>Food.0000</i>	55	44	34	46	50	52
<i>Food.0005</i>	52	57	52	62	65	72
<i>Food.0008</i>	27	25	21	76	77	75
<i>Grass.0001</i>	80	79	67	81	83	82

<i>Leaves.0008</i>	57	49	50	67	70	68
<i>Leaves.0010</i>	38	43	48	46	38	45
<i>Leaves.0011</i>	45	34	25	66	63	61
<i>Leaves.0012</i>	52	50	42	64	63	63
<i>Leaves.0016</i>	75	57	44	70	70	67
<i>Metal.0000</i>	85	85	85	98	100	100
<i>Metal.0002</i>	100	98	88	100	100	100
<i>Misc.0002</i>	93	91	79	90	97	95
<i>Sand.0000</i>	79	66	54	69	71	73
<i>Stone.0001</i>	39	32	30	73	73	73
<i>Stone.0004</i>	95	89	91	98	98	98
<i>Terrain.0010</i>	57	53	54	57	56	56
<i>Tile.0001</i>	68	42	40	86	85	85
<i>Tile.0004</i>	67	44	32	71	67	66
<i>Tile.0007</i>	98	71	54	100	99	99
<i>Water.0005</i>	100	100	97	100	100	100
<i>Wood.0001</i>	32	30	30	48	47	46
<i>Wood.0002</i>	100	100	100	95	95	95
<b>RR% (Global)</b>	<b>66.08</b>	<b>58.53</b>	<b>53.83</b>	<b>74.80</b>	<b>74.78</b>	<b>74.98</b>

## BIBLIOGRAPHIE

- [1] D. Noyes, "The Top 20 Valuable Facebook Statistics," Zephoria Internet Marketing Solutions, 28 11 2018. [Online]. Available: <https://zephoria.com/top-15-valuable-facebook-statistics/>. [Accessed 01 12 2018].
- [2] S. S. Hiwale and D. Dhotre, "Content-Based Image Retrieval: Concept and Current Practices," in *Electrical, Electronics, Signals, Communications and Optimization (EESCO) 2015*, Visakhapatnam, 2015.
- [3] N. Baaziz, O. Abahmane and R. Missaoui, "Texture feature extraction in the spatial-frequency domain for content-based image retrieval," in *Corr Information Retrieval and Multimedia*, arXiv:1012.5208, 2010.
- [4] K. M. Shriwas and V. R. Raut, "Content Based Image Retrieval : A past, Present and New Feature Descriptor," in *International Conference on Circuit, Power and Computing Technologies*, 2015.
- [5] A. Mumtaz, A. Awais, S. Muhammd and u. Zahid, "Content Based Image Retrieval (CBIR) Using Materialized Views," in *International Conference on Computer Science and Information Technology (ICCSIT)*, Pattaya, 2011.
- [6] A. Ventresque, "Une mesure de similarité sémantique utilisant des résultats de psychologie," in *Conférence en Recherche d'Informations et Applications-CORIA 2006*, Lyon, Mars 2006.
- [7] J. M. Zachary and S. S. Iyengar, "Content Based Image Retrieval Systems," in *Application-Specific Systems and Software Engineering and Technology*, Richardson, TX, 1999.
- [8] H. Abed and L. Zaoui, "Système d'indexation et de recherche d'images par le contenu," in *Proceedings of the 2nd Conférence Internationale sur l'Informatique et ses Applications (CIIA'09)*, Algeria, 2009.
- [9] M. Bouet, "Vers une recherche intelligente du média image?," in *Congrès Inforsid : INFormatique des ORganisations et Systèmes d'Information et de Décision*, Nantes, France, 2002.
- [10] C. A. Hussain, D. Rao and S. Mastani, "Low level feature extraction methods for content based image retrieval," in *Electrical, Electronics, Signals, Communications and Optimization (EESCO) 2015*, Visakhapatnam, 2015.
- [11] "Comment lire un histogramme? (EOS 7D)," 21 10 2009. [Online]. Available: [http://canoncanadafr.custhelp.com/app/answers/detail/a\\_id/16743/~/-comment-lire-un-histogramme%3F-\(eos-7d\)](http://canoncanadafr.custhelp.com/app/answers/detail/a_id/16743/~/-comment-lire-un-histogramme%3F-(eos-7d)). [Accessed 31 10 2018].
- [12] N. Idrissi, "La navigation dans les bases d'images : prise en compte des attributs de

- texture," Doctoral thesis Human-Computer Interaction [cs.HC]. Université de Nantes; Faculté des sciences de Rabat, 2008.
- [13] M. Ashutosh Gupta, "Image Retrieval Based on Color, Shape and Texture," in *Computing for Sustainable Global Development (INDIACom)*, New Delhi, 2015.
- [14] C. E. Mehemmed and Y. A. Aslandogan, "A comparative study of three moment-based shape descriptors," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, Texas USA, April 2005.
- [15] S. BEDOUHENE, "Recherche d'images par le contenu," 02 07 2011. [Online]. Available: <http://www.ummtto.dz/IMG/pdf/these-2.pdf>. [Accessed 02 11 2015].
- [16] IBM Almaden Research Center , "Query by image and video content: the QBIC system," in *IEEE Computer*, 1995.
- [17] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain and C.-F. Shu, "Virage image search engine: an open framework for image management," in *Proceedings of the SPIE, Volume 2670*, p. 76-87, 1996.
- [18] W. Ekta, G. Anjali and Y. S. Brar, "Zernike moments and LDP-weighted patches for content-based image retrieval," Springer-Verlag, London, 2013.
- [19] C.-Y. Wee and R. Paramesran, "On the computational aspects of Zernike moments," *Image and Vision Computing*, no. 25, pp. 967-980, 2007.
- [20] T. Le-Tien, T. Huynh-Ngoc, H.-K. Tu and M. Luong, "Zernike Moment-Based Approach for Detecting Duplicated Image Regions by a Modified Method to Reduce Geometrical and Numerical Errors," Springer International Publishing, Switzerland, 2015.
- [21] D. Yang, L. Zhang, J. Wang and J. Liao, "PZMLIR: Pseudo-Zernike Moments and LSH-based Image Retrieval in P2P datacenter," in *National Doctoral Academic Forum on Information and Communications Technology*, Beijing, China, 2013.
- [22] S. Abu Bakar, M. Hitam and W. Yussof, "Investigating the properties of Zernike moments for robust content based image retrieval," in *International Conference on Computer Applications Technology (ICCAT)*, 2013.
- [23] J. Esther and M. Sathik, "Retrieval of Brain Image using Soft Computing Technique," in *2014 International Conference on Intelligent Computing Applications*, Coimbatore, India, 2014.
- [24] S. Shrivastava, B. Gupta and M. Gupta, "Optimization of image retrieval by using HSV color space, Zernike moment & DWT technique," in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India, 2015.
- [25] D. Sudarvizhi, "Feature based Image Retrieval System using Zernike Moments and Daubechies Wavelet Transform," in *IEEE International Conference on Recent Trends in Information Technology (ICRTIT)*, 2016.
- [26] M. Z. Pwint, T. T. Zin, M. YOKOTA and M. M. Tin, "Shape Descriptor for Binary

- Image Retrieval," in *IEEE 5th Global Conference on Consumer Electronics*, Kyoto, Japan, 2016.
- [27] A. Aggarwal and K. Singh, "Zernike Moments-based Retrieval of CT and MR Images," in *IEEE India Conference (INDICON)*, 2015.
- [28] M. E. Malek, Z. Azimifar and R. Boostani, "Age-based Human Face Image Retrieval using Zernike Moments," in *IEEE Artificial Intelligence and Signal Processing Conference (AISP)*, 2017.
- [29] N. Baaziz, "Adaptive watermarking schemes based on a redundant contourlet transform," in *IEEE International Conference on Image Processing, (ICIP) 221-224*, 2005.
- [30] M. Vetterli and M. N. DO, "The contourlet transform: an efficient directional multiresolution image representation," in *IEEE Transactions on Image Processing*, 14/12:2091-2106, 2005.
- [31] N. Baaziz and M. Diop, "Impact of redundancy and Gaussian filtering on Contourlet-based texture retrieval," in *International Conference on Advances in Signal, Image and Video Processing*, Lisbon, 2016.
- [32] K.-y. Chang, "LANS Pattern Recognition Toolbox," 13 08 2000. [Online]. Available: <http://lans.ece.utexas.edu/~kuiyu>.