

**RÉSUMÉ**

Les travaux présentés dans ce projet de recherche sont dirigés vers la reconnaissance de formes et la classification des bases de données d'images de grandes tailles qui coulent dans l'amélioration des systèmes biométriques. Dans ce cadre, nous apportons des améliorations inédites aux différents composants des systèmes biométriques. Les systèmes résultants procèdent en trois phases : filtrage, extraction des primitives et classification. Dans la phase de filtrage, nous utilisons la méthode des éléments finis mixte pour résoudre le modèle de Perona-Malik qui permet de réduire le bruit de l'image sans supprimer les parties significatives du contenu de l'image, généralement les bords, les contours, les textures ou d'autres détails importants pour l'interprétation de l'image. Dans la deuxième phase, nous nous intéressons à l'extraction des vecteurs descripteurs des images en utilisant la théorie des moments invariants. Les sept moments de Hu sont largement utilisés dans le domaine de la classification d'images et la reconnaissance de formes. Ces moments sont d'ordre fini, par conséquent, ils n'arrivent pas à extraire un ensemble complet de caractéristiques de l'image. Pour cette raison, nous proposons dans ce travail trois nouvelles approches. Dans la première approche, nous introduisons un ensemble de moments invariants en fonction des moments centraux normalisés d'ordre infini permettant de remédier au manque de la description complète de l'image posée par les moments invariants de Hu. Ces moments invariants ne sont pas orthogonaux, ce qui implique ainsi une redondance d'information. Pour remédier à ce problème, nous introduisons une série de moments invariants orthogonaux basés sur un ensemble de polynômes orthogonaux non séparables à deux variables, qui sont construits à l'aide d'une nouvelle technique axée sur une relation de récurrence. Dans la deuxième approche, nous introduisons un nouvel ensemble de polynômes orthogonaux basés sur les polynômes de Legendre, que nous appelons les polynômes de Legendre adaptés. A partir de ces polynômes nous construisons une famille de moments invariants pour les trois transformations géométriques (la translation, le changement d'échelle et la rotation) d'une image. Dans la troisième approche, nous proposons un ensemble de moments invariants orthogonaux centrés sur les polynômes orthogonaux de Jacobi en utilisant les coordonnées polaires. Pour la phase finale, nous appliquons des algorithmes de classification supervisée : les k plus proches voisins (k-PP), les réseaux de neurones à fonctions de base radiales (RNRBF) et les réseaux de neurones perceptron multicouche (RN-MLP) et des algorithmes de classification non supervisée tels que les méthodes de k-means (KM), de fuzzy k-means (FKM) et de possibilistic fuzzy k-means (PFKM). Nous introduisons aussi des nouveaux systèmes biométriques basés sur les vecteurs descripteurs proposés. Pour évaluer la performance des systèmes de classification et des systèmes biométriques proposés, nous effectuons quelques tests expérimentaux. Les résultats obtenus montrent l'efficacité et la supériorité des approches proposées.

**Mots clés :** Méthodes de machine Learning, Reconnaissance biométrique, Classification d'images, Méthodes des éléments finis mixtes, Moments orthogonaux invariants, Modèle de Perona-Malik.



UNIVERSITE SULTAN MOULAY SLIMANE  
Faculté des Sciences et Techniques  
Béni-Mellal



**Centre d'Études Doctorales : Sciences et Techniques**

Formation Doctorale : **Mathématiques et Physique Appliquée**

**THÈSE**

Présentée par

**HJOUJI AMAL**

Pour l'obtention du grade de

**DOCTEUR**Spécialité : **INFORMATIQUE**Option : **INFORMATIQUE**


---

**CONTRIBUTION AUX SYSTEMES BIOMETRIQUES BASEE SUR  
DES NOUVEAUX MOMENTS INVARIANTS, LES ELEMENTS FINIS  
MIXTES ET LES METHODES DU MACHINE LEARNING**

---

**Soutenu le Samedi 20 Octobre 2020 à 12h devant la commission d'examen :**

<b>Mohamed FAKIR</b>	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni-Mellal, Maroc	Président
<b>Charki DAOUI</b>	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni-Mellal, Maroc	Rapporteur
<b>Karim EL MOUTAOUAKIL</b>	Professeur, Université Sidi Mohamed Ben Abdellah, F.P. Taza, Maroc	Rapporteur
<b>Rachid EL AYACHI</b>	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni-Mellal, Maroc	Rapporteur
<b>Belaid BOUIKHALENE</b>	Professeur, Université Sultan Moulay Slimane, F.P. Béni-Mellal, Maroc	Co-Directeur de thèse
<b>Mostafa JOURHMANE</b>	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni-Mellal, Maroc	Directeur de thèse
<b>Hicham Moncif</b>	Professeur, Université Sultan Moulay Slimane, F.P. Béni-Mellal, Maroc	Invité

**CONTRIBUTION AUX SYSTEMES BIOMETRIQUES BASEE SUR DES NOUVEAUX MOMENTS INVARIANTS, LES ELEMENTS FINIS MIXTES ET LES METHODES DU MACHINE LEARNING**

**HJOUJI AMAL**

**N° d'ordre : 283/20**

THÈSE  
POUR OBTENIR LE GRADE DE

**DOCTEUR DE**

**Université Sultan Moulay Slimane**

Faculté des Sciences et Techniques-Beni Mellal

SPÉCIALITÉ : Informatique

Par : Amal Hjouji

**Titre du sujet de thèse**

**Contribution aux systèmes biométriques basée sur des nouveaux moments invariants, les éléments finis mixtes et les méthodes du Machine Learning**

Directeur de thèse : Mostafa JOURHMANE

Co-directeur de thèse : Belaid BOUIKHALENE

Soutenue le ...

Membres du jury :

Mme/M. NOM, Prénom	Titre Établissement	Président
Mme/M. NOM, Prénom	Titre Établissement	rapporteur
Mme/M. NOM, Prénom	Titre Établissement	rapporteur
Mme/M. NOM, Prénom	Titre Établissement	Examineur
Mme/M. NOM, Prénom	Titre Établissement	Examineur
Mme/M. NOM, Prénom	Titre Établissement	Invité



**Résumé :**

Les travaux présentés dans ce projet de recherche sont dirigés vers la reconnaissance de formes et la classification des bases de données d'images de grandes tailles qui coulent dans l'amélioration des systèmes biométriques. Dans ce cadre, nous apportons des améliorations inédites aux différents composants des systèmes biométriques. Les systèmes résultants procèdent en trois phases : filtrage, extraction des primitives et classification.

Dans la phase de filtrage, nous utilisons la méthode des éléments finis mixte pour résoudre le modèle de Perona-Malik qui permet de réduire le bruit de l'image sans supprimer les parties significatives du contenu de l'image, généralement les bords, les contours, les textures ou d'autres détails importants pour l'interprétation de l'image.

Dans la deuxième phase, nous nous intéressons à l'extraction des vecteurs descripteurs des images en utilisant la théorie des moments invariants. Les sept moments de Hu sont largement utilisés dans le domaine de la classification d'images et la reconnaissance de formes. Ces moments sont d'ordre fini, par conséquent, ils n'arrivent pas à extraire un ensemble complet de caractéristiques de l'image. Pour cette raison, nous proposons dans ce travail trois nouvelles approches. Dans la première approche, nous introduisons un ensemble de moments invariants en fonction des moments centraux normalisés d'ordre infini permettant de remédier au manque de la description complète de l'image posée par les moments invariants de Hu. Ces moments invariants ne sont pas orthogonaux, ce qui implique ainsi une redondance d'information. Pour remédier à ce problème, nous introduisons une série de moments invariants orthogonaux basés sur un ensemble de polynômes orthogonaux non séparables à deux variables, qui sont construits à l'aide d'une nouvelle technique axée sur une relation de récurrence. Dans la deuxième approche, nous introduisons un nouvel ensemble de polynômes orthogonaux basés sur les polynômes de Legendre, que nous appelons les polynômes de Legendre adaptés. A partir de ces polynômes nous construisons une famille de moments invariants pour les trois transformations géométriques (la translation, le changement d'échelle et la rotation) d'une image. Dans la troisième approche, nous proposons un ensemble de moments invariants orthogonaux centrés sur les polynômes orthogonaux de Jacobi en utilisant les coordonnées polaires.

Pour la phase finale, nous appliquons des algorithmes de classification supervisée : les  $k$  plus proches voisins ( $k$ -PP), les réseaux de neurones à fonctions de base radiales (RN-RBF) et les réseaux de neurones perceptron multicouche (RN-MLP) et des algorithmes de classification non supervisée tels que les méthodes de  $k$ -means (KM), de fuzzy  $k$ -means (FKM) et de possibilistic fuzzy  $k$ -means (PFKM).

Nous introduisons aussi des nouveaux systèmes biométriques basés sur les vecteurs descripteurs proposés. Pour évaluer la performance des systèmes de classification et des

systemes biométriques proposés, nous effectuons quelques tests expérimentaux. Les résultats obtenus montrent l'efficacité et la supériorité des approches proposées.

---

**Mots clés : Méthodes de machine learning, Reconnaissance biométrique, Classification d'images, Méthodes des éléments finis mixtes, Moments orthogonaux invariants, Modèle de Permpna-Malik**

---

**Abstract :**

The works presented in this research project are directed towards pattern recognition and classification of the large image databases. In this work, we propose a new image classification system, particularly, valid for the databases containing noisy or geometrically distorted images. This system consists of three steps. In the first step, we use the mixed finite element method to solve the Perona-Malik model allowing to reduce the noise of the image without removing the significant parts of the image content, usually the edge, the contours, the textures or other details important for the interpretation of the image. In the second step, we are interested in the extraction of descriptor vectors of the images using the theory of invariant moments. The seven invariant moments of Hu are widely used in the field of image classification and pattern recognition. These moments have finite order, therefore, they can not extract a complete set of features from the image. For this reason, we propose in this work three new approaches : In the first approach, we introduce a set of invariant moments in terms of the normalized central moments have infinite order, which make it possible to remedy the incomplete description of the image which poses by the invariant moments of Hu. These invariant moments are not orthogonal, which implies a redundancy of information. To remedy this problem, we introduce a series of orthogonal invariant moments based on a set of non-separable bivariate orthogonal polynomials, which are constructed using a new technique based on a recurrence relation. In the second approach, we introduce a new set of orthogonal polynomials based on the Legendre polynomials, which we call "the adapted Legendre polynomials". From these polynomials we construct a family of invariant moments under the three geometric transformations (the translation, the scaling and the rotation) of the image. In the third approach, we propose a set of orthogonal invariant moments centered on the Jacobi orthogonal polynomials using the polar coordinates. In the last step, we apply supervised and unsupervised classification algorithms such as k-means clustering (KM), fuzzy k-means clustering (FKM), possibilistic fuzzy k-means clustering (PFKM), k-nearest neighbor (K-NN) and the artificial neural networks RBF and MLP. We also introduce an indexing and image search system based on the proposed descriptor vectors. To evaluate the performance of our classification and indexation systems and image search, we perform experimental tests, the results obtained show the effectiveness and superiority of the proposed approaches.

---

**Keywords :** Image classification, Finite element methods, Machine learning, Pattern recognition, Orthogonal invariant moments, Perona-Malik model, Image retrieval

---



## Remerciement

Je tiens à remercier le professeur Mohamed FAKIR pour l'honneur qu'il me fait pour présider le jury. Je tiens à remercier sincèrement mon directeur de thèse le professeur Mostafa JOURHMANE et le co-directeur Belaid BOUIKHALENE qui m'ont apporté une aide très efficace grâce à ces nombreuses remarques pertinentes et leurs précieux conseils qui m'étaient utiles tout au long de la réalisation de ce travail. Je remercie également les professeurs Cherki DAOUI, Karim EL MOUTAOUAKIL et Rachid EL AYACHI pour avoir accepté de rapporter sur mon travail. Je remercie très vivement le professeur Hicham MOUNCIF qui m'a honoré par sa présence parmi les membres du jury. Je le prie d'accepter mes vifs remerciements. Je tiens à remercier également les professeurs Hassan QJIDAA et Jaouad EL-MEKKAOUI qui m'ont apporté un grand appui dans l'avancement de mes travaux. Je leur exprime ma gratitude pour leur patience et le temps qu'ils m'ont consacré tout au long de la réalisation de ce travail. Je tiens à exprimer ma vive reconnaissance à ceux qui m'ont donné le sens de la vie : mes parents, à ceux qui m'ont encouragé tout au long de mes études : mes frères, mes soeurs et mon mari. Je n'oublie pas le personnel du laboratoire de TIAD pour leur encouragement sympathique qui a contribué à ma motivation pour ce travail. Je les remercie pour leurs conseils et l'intérêt qu'ils m'ont témoigné pour mes recherches. Mes vifs remerciements vont également à tous les collègues doctorants pour leur soutien, leur encouragement et leur bonne humeur, ce qui m'a parfois permis de surmonter des moments difficiles.

# Table des matières

<b>Introduction Générale</b>	<b>11</b>
0.1 Contexte et problématique . . . . .	11
0.2 Approches proposées . . . . .	13
0.3 Plan du manuscrit . . . . .	14
<b>1 Filtrage d'image en résolvant le modèle de Perona-Malik par la méthode des éléments finis mixtes</b>	<b>17</b>
1.1 Modèle de Perona-Malik à une seule variable . . . . .	18
1.2 Modèle de Perona-Malik à deux variables . . . . .	19
1.3 Discrétisation en temps . . . . .	19
1.4 Formulation variationnelle . . . . .	20
1.5 Approximation par éléments finis mixtes . . . . .	21
1.6 Construction géométrique des éléments de bases des espaces d'approximations	22
1.7 Résultats expérimentaux . . . . .	26
1.8 Conclusion . . . . .	29
<b>2 Extraction des vecteurs descripteurs des images en utilisant la théorie des moments orthogonaux</b>	<b>37</b>
2.1 Types de descripteurs d'images . . . . .	37
2.1.1 La couleur . . . . .	38
2.1.2 La texture . . . . .	39
2.1.3 La forme . . . . .	39
2.2 Théorie des moments . . . . .	40
2.3 Moments orthogonaux . . . . .	43
2.3.1 Moments orthogonaux de Legendre . . . . .	44
2.3.2 Moments orthogonaux de Legendre adaptés . . . . .	46
2.3.3 Nouvel ensemble des moments orthogonaux . . . . .	57
2.3.4 Moments non orthogonaux basés sur les moments centraux normalisés	57

2.3.5	Moments orthogonaux basés sur les moments centraux normalisés . . . . .	60
2.4	Moments orthogonaux radiaux de Jacobi invariants . . . . .	67
2.4.1	Moments orthogonaux radiaux de Jacobi . . . . .	67
2.4.2	La dérivation des invariants de moments orthogonaux radiaux de Jacobi . . . . .	69
2.4.3	L'invariance à la rotation . . . . .	69
2.4.4	L'invariance à la translation . . . . .	70
2.4.5	L'invariance au changement d'échelle . . . . .	70
2.4.6	Tests expérimentaux sur les moments radiaux de Jacobi . . . . .	71
2.5	Conclusion . . . . .	73
<b>3</b>	<b>Classification d'images par la méthode des éléments finis mixtes et la théorie des moments invariants</b>	<b>75</b>
3.1	Méthodes de classification non-supervisée . . . . .	76
3.1.1	Méthode de k-means clustering : KM . . . . .	76
3.1.2	Méthode fuzzy k-means clustering : FKM . . . . .	78
3.1.3	Méthode k-moyennes floues possibilistes . . . . .	81
3.2	Méthodes de classification supervisée . . . . .	81
3.2.1	Classification par k-plus proches voisins : k-PPV . . . . .	83
3.2.2	Classification par réseaux de neurones à fonctions radiales : RBF . . . . .	85
3.2.3	Classification par réseaux de neurones perceptron multicouche : MLP . . . . .	87
3.3	Résultats expérimentaux sur la classification . . . . .	89
3.3.1	Bases de données d'images utilisées . . . . .	91
3.3.2	Classifications des bases de données d'images . . . . .	93
3.4	Conclusion . . . . .	100
<b>4</b>	<b>Systèmes biométriques basés sur les vecteurs descripteurs proposés</b>	<b>102</b>
4.1	Architecture d'un système biométrique . . . . .	102
4.2	Les catégories biométriques utilisées . . . . .	103
4.2.1	Système biométrique basé sur les empreintes digitales . . . . .	105
4.2.2	Système biométrique basé sur le visage . . . . .	105
4.2.3	Système biométrique basé sur l'iris . . . . .	106
4.2.4	Système biométrique basé sur la signature . . . . .	106
4.3	Architecture du système d'indexation et recherche d'images . . . . .	107
4.4	Mesures pour évaluer un système . . . . .	108
4.4.1	Rappel . . . . .	109
4.4.2	Précision . . . . .	109

4.4.3	Courbe rappel-précision . . . . .	110
4.5	Résultats expérimentaux . . . . .	111
4.5.1	Bases d'images utilisées . . . . .	111
4.5.2	Nos résultats . . . . .	113
4.6	Conclusion . . . . .	116
	<b>Conclusion et perspectives</b>	<b>121</b>

## Introduction Générale

Les travaux présentés, dans le présent rapport de thèse, s'inscrivent dans le domaine de reconnaissance biométrique et de classification d'images. Plus précisément, la classification des bases de données et la reconnaissance de formes des images 2D qui pourraient être bruitées et déformées géométriquement. L'approche proposée dans cette thèse donne des solutions aux problèmes de classification et de reconnaissance d'images assez bruitées par la résolution du modèle de Perona-Malik à l'aide de la méthode des éléments finis mixte et par l'utilisation des moments invariants orthogonaux. Dans cette introduction, nous présentons d'abord le contexte dans lequel nous nous plaçons, suivi d'une description de l'approche proposée. Ensuite, on expose les contributions de cette thèse et on aboutit à un plan de ce document.

### 0.1 Contexte et problématique

Aujourd'hui, on est confronté à plusieurs bases de données contenant des milliers d'images, que ce soit de bases ciblées pour un domaine d'activité professionnelle ( médecine, presse, tourisme, éducation, musées, audio-visuel ...) ou pour les particuliers qui accumulent d'immenses bases de photographies numériques des événements marquants comme les fêtes, les voyages, les souvenirs d'enfance, les enquêtes policières etc. Pour gérer et utiliser efficacement ces bases de données, un système de classification automatique s'avère nécessaire et inévitable. Au fil des années, de nombreuses méthodes de machine learning ont été développées pour la classification d'images. Chaque méthode peut avoir sa propre technique de partitionnement, à savoir : la méthode hiérarchique, k-means (KM), fuzzy k-means (FKM), possibilistic fuzzy k-means (PFKM), réseaux de neurones convolutifs (CNN), k-plus proches voisins (K-PPV) et les réseaux de neurones artificiels (ANN) etc. Compte tenu de leur limite d'efficacité pour les bases de données d'images contenant des images bruitées ou géométriquement déformées, plusieurs chercheurs se sont attachés à les améliorer ou à inventer de nouvelles méthodes pour satisfaire la demande actuelle.

Pour que tout système de classification d'images soit efficace, il est impératif d'extraire des vecteurs descripteurs invariants pour les trois types de transformations géométriques :

la translation, la rotation et le changement d'échelle. Au cours des dernières années, plusieurs groupes de recherche ont proposé les moments invariants [2,3] pour extraire les vecteurs descripteurs des images. Les descripteurs basés sur la théorie des moments et des polynômes orthogonaux sont des concepts fondamentaux très utilisés dans l'analyse d'image, non seulement en raison de leur simplicité, mais aussi pour l'extra-ordinaire variété des sujets et des applications où ils sont éclairés. Les moments et les fonctions moments sont appliquées aux différents types d'images définies en 2D ou en 3D pour la reconnaissance de formes avec des applications telles que la détection des contours, la classification et la segmentation d'image, l'analyse de texture, l'identification des invariants, la classification d'objets, le codage d'image, la reconstruction d'image, l'analyse de la scène et l'analyse d'objets 3D...

L'idée fondamentale de la théorie des moments est la projection de l'espace des données d'images sur une base polynomiale complète souvent orthogonale pour extraire les informations utiles. Ces dernières sont extraites sous forme des coefficients de projection appelés moments. Chaque famille de moments est directement liée à la base de projection utilisée. On peut distinguer deux types de moments : les moments non orthogonaux et les moments orthogonaux, qui sont basés sur les polynômes orthogonaux séparables  $\psi_{n,m}(x, y) = P_n(x)P_m(y)$ . Les descripteurs extraits à l'aide des moments invariants ont été introduits pour la première fois par Hu en 1962. Ce dernier a généré un ensemble de moments géométriques invariants. En effet, Hu a présenté en [1] sept moments invariants pour les trois transformations géométriques : la rotation, le changement d'échelle et la translation. Ces moments invariants sont largement utilisés dans le domaine de classification et reconnaissance d'images : Y. S. Abu-Mostafa [57, 58] a utilisé les moments de Hu dans le domaine de reconnaissance de formes. Cependant l'inconvénient des moments de Hu est que la base de projection n'est pas orthogonale, impliquant ainsi une redondance d'information [13]. Afin de garantir relativement les propriétés précédentes, Teague a suggéré en 1980 d'utiliser des moments orthogonaux, basés sur les polynômes orthogonaux séparables, pour représenter l'image à partir des moments orthogonaux [25]. Il a été démontré que ces moments orthogonaux [2-25-3] sont meilleurs que d'autres types de moments en termes de redondance d'informations et sont plus robustes au bruit. Comme noté par Teh et Chin [17-18], les moments invariants sont considérés comme des caractéristiques fiables dans la reconnaissance de formes s'ils sont insensibles à la présence de bruit dans l'image. Mais cela ne se produit pas toujours car la plupart des moments invariants sont affectés relativement par le bruit.

Lors de la classification supervisée ou non supervisée des bases de données d'images, autant que nous avons besoin des vecteurs descripteurs robustes, nous avons besoin aussi

d'un algorithme convenable. La complexité croissante des algorithmes traités dans ce domaine a donné lieu à des approches très diverses et a amené à envisager des outils plus innovants. Parmi les algorithmes introduits dans cette discipline, les algorithmes de k-means (KM), fuzzy k-means (FKM), k-plus proche voisin (k-PPV), les réseaux de neurones convolutifs et les réseaux de neurones artificiels qui ont prouvé leur efficacité dans plusieurs applications. Ces méthodes sont largement utilisées, mais elles ne peuvent pas être efficaces pour la classification des bases de données d'images déformées et très bruitées si elles ne sont pas accouplées avec des vecteurs descripteurs robustes.

## 0.2 Approches proposées

Dans cette thèse, nous proposons un nouveau système de classification des bases de données d'images, notamment bruitées. Ce système s'articule en trois étapes comme illustré dans la figure 0.1.

**Étape 1 :** Dans cette étape, nous utilisons la méthode des éléments finis mixtes pour résoudre le modèle de Perona-Malik qui permet de réduire le bruit de l'image sans supprimer les parties significatives du contenu de l'image, généralement les bords, les contours, les textures ou des d'autres détails importants pour l'interprétation de l'image.

**Étape 2 :** Dans cette étape, nous utilisons la théorie des moments pour extraire les vecteurs descripteurs d'images qui sont invariants aux trois transformations géométriques : translation, rotation et changement d'échelle. Comme nous avons déjà évoqué, les sept moments de Hu  $\{\varphi_{i=1,\dots,7}\}$  sont largement utilisés dans le domaine de classification d'images et la reconnaissance de formes. Ces moments sont d'ordre fini, par conséquent, ils n'arrivent pas à donner un ensemble complet de caractéristiques de l'image. Pour cette raison, nous proposons de nouvelles approches.

Dans la première approche, nous introduisons un ensemble de moments invariants en fonction des moments centraux normalisés d'ordre infini permettant de remédier au manque de la description complète de l'image. Malheureusement, ces moments invariants ne sont pas orthogonaux, ce qui implique ainsi une redondance d'information [13]. Pour remédier à ce problème, nous introduisons dans ce projet une série de moments invariants orthogonaux basés sur un ensemble de polynômes orthogonaux non séparables à deux variables. Ces polynômes sont construits à l'aide d'une nouvelle technique basée sur une relation de récurrence.

Dans la seconde approche, nous introduisons un nouvel ensemble de polynômes orthogonaux basés sur les polynômes de Legendre, que nous appelons "les polynômes de Legendre adaptés". A partir de ces polynômes nous proposons une famille de moments inva-

riants pour les trois transformations géométriques (la translation, le changement d'échelle et la rotation) d'une image.

Dans la dernière approche, nous proposons un ensemble de moments invariants orthogonaux basés sur les polynômes orthogonaux de Jacobi en utilisant les coordonnées polaires  $(r, \theta)$ .

**Etape 3 :** Dans cette étape, en se basant sur les vecteurs descripteurs extraits à l'aide des étapes précédentes, nous appliquons un algorithme de classification pour classifier une base de données d'images comme la méthode k-means (KM), fuzzy k-means (FKM), possibilistic fuzzy k-means (PFKM), k-plus proches voisins (K-PP), réseaux de neurones artificiels (ANN) etc, tels que les descripteurs calculés à l'aide de la méthode proposée considérées comme des vecteurs entrés.

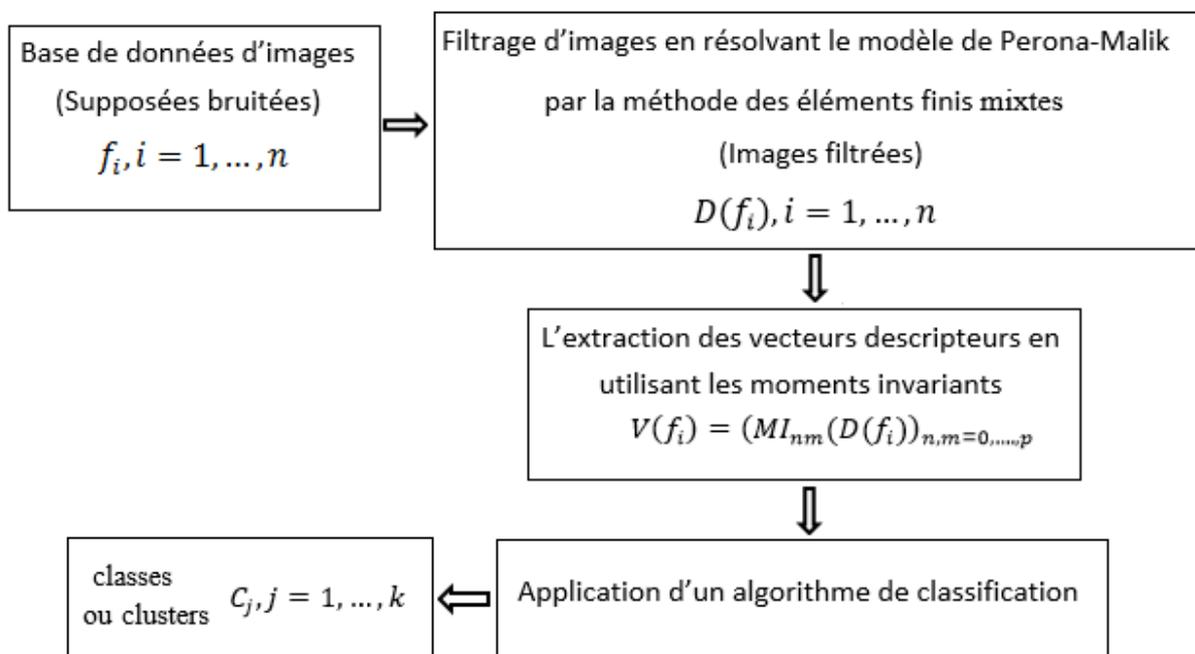


FIGURE 1 – Architecture de notre système de classification.

### 0.3 Plan du manuscrit

La présente thèse est composée de quatre chapitres. Dans le premier chapitre, nous exposons les étapes de la résolution numérique du modèle de Perona-Malik par la méthode des éléments finis mixtes. Dans ce cadre, nous établissons la formulation variationnelle et les espaces associés. Nous montrons l'existence et l'unicité de la solution. Puis, nous

construisons un algorithme de filtrage d'image basé sur les éléments finis de Taylor-Hood  $P_1 - P_2$  et  $Q_1 - Q_2$ . Et pour évaluer la performance de cet algorithme nous présentons des tests expérimentaux sur quelques images 2D et 3D. Puis, nous effectuons une étude comparative avec d'autres méthodes comme les méthodes de différence finie, les éléments finis classiques et la méthode de volume fini, les résultats obtenus montrent l'efficacité de l'algorithme proposé.

Dans le deuxième chapitre, nous présentons quelques types de moments non orthogonaux et orthogonaux que nous utiliserons dans ce projet à savoir les moments géométriques, les moments centraux, les moments centraux normalisés, les sept moments invariants de Hu, les moments de Legendre. Puis, nous introduisons une nouvelle série de moments invariants non orthogonaux d'ordre infini basés sur les moments géométriques centraux normalisés. Nous présentons aussi une suite de polynômes orthogonaux non séparables à deux variables permettant de construire une nouvelle série de moments orthogonaux invariants. A l'aide des polynômes de Legendre, nous présentons un nouvel ensemble de moments orthogonaux invariants pour les trois transformations affines, la translation, la rotation et le changement d'échelle. Ces moments sont appelés les moments invariants orthogonaux de Legendre adaptés. Pour des raisons de précision et de clarification, nous présentons les définitions de certains termes et de propositions que nous utilisons tout au long de cette thèse.

Dans le troisième chapitre, nous présentons de nouveaux systèmes de classification des bases de données d'images. Dans ce cadre, nous nous intéressons aux trois algorithmes de classification supervisée et trois algorithmes de classification non-supervisée. En se basant sur les vecteurs descripteurs extraits à l'aide des étapes présentés dans les chapitres précédents. Pour la classification supervisée, nous utilisons la méthode de k-plus proches voisins (K-PPV) et deux méthodes de réseaux de neurons artificiels : la méthode perceptron multicouche (MLP) et la méthode à base radiale (RBF). Et pour la classification non-supervisée nous appliquons la méthode k-means (KM), la méthode fuzzy k-means (FKM) et la méthode possibilistic fuzzy k-means (PFKM).

Dans le dernier chapitre, nous proposons un système d'indexation et recherche d'image par le contenu. Pour indexer les images, nous utilisons les vecteurs descripteurs présentés dans les deux premiers chapitres, qui sont basées sur le débruitage d'images en utilisant la méthode des éléments finis mixtes et le modèle de Perona-Malik ainsi la théorie des moments orthogonaux. Pour évaluer les systèmes de classification proposés, nous présentons quelques tests expérimentaux sur des bases de données d'images bien connues comme : la base de données MPEG-7-CE Shape [143], Columbia Object Image Library (COIL-20), Columbia Object Image Library (COIL-100)[144], la base de données de visages ORL

[145], la base MNIST fashion [146] et la base MNIST handwritten [147]. Les résultats obtenus montrent l'efficacité et la supériorité des approches proposées.

Enfin, la conclusion générale présente une synthèse de différentes contributions apportées ainsi que les pistes qui définissent des perspectives possibles pour des travaux futurs.

# Chapitre 1

## Filtrage d'image en résolvant le modèle de Perona-Malik par la méthode des éléments finis mixtes

Le filtrage d'image est un problème essentiel en traitements d'images. Effectivement, il arrive continuellement que lors d'une capture d'image des phénomènes comme le bruit, le flou, la mauvaise qualité... apparaissent. Le modèle de diffusion non linéaire, introduit par Perona et Malik [30, 31] est une technique de filtrage non linéaire d'images. Lors d'un filtrage linéaire (comme la convolution avec une Gaussienne), l'image est lissée uniformément, ce qui crée du flou aux bords des discontinuités. L'idée des méthodes non linéaires est de paramétrer le filtrage par la régularité locale de l'image. Dans ce cadre Perona et Malik ont proposé l'utilisation d'une équation aux dérivées partielles (EDP) (1) qui permet de supprimer le bruit de l'image tout en préservant les contours, les bords et les textures qui jouent un rôle très important en reconnaissance des formes des objets. C'est un modèle puissant est largement utilisé dans le débruitage d'image. Ce modèle est basé sur l'équation de la chaleur améliorée [1, 25] en introduisant le processus de détection de bords et de contours. Le débruitage d'une image est obtenu en résolvant l'équation de diffusion anisotrope (1.1) avec une condition aux limites de Neumann (1.2). Plusieurs méthodes ont été utilisées pour résoudre ce problème comme les éléments finis classique (MEFC) , la méthode de différence finie (MDF) et la méthode de volumes finis (MVF). Grâce à leur efficacité dans les domaines de génie industriel et génie mécanique nous avons utilisé dans ce travail les méthodes des éléments finis mixtes (MEFM) basée sur les éléments de Taylor-Hood.

## 1.1 Modèle de Perona-Malik à une seule variable

En nous plaçant dans le cadre de l'amélioration d'une image en niveaux de gris, la formulation par equation aux dérivées partielles (EDP) implique classiquement la modélisation de l'image par un ensemble discret de points (pixels). En considérant la luminance comme une fonction des coordonnées spatiales  $(x, y)$  et du temps  $t$ . Les propriétés de l'image filtrée sont obtenues à travers une EDP ayant comme arguments la fonction luminance et ses dérivées partielles, la solution de cette EDP à un instant  $t$  représente l'image restaurée (améliorée).

Soit  $u_0 : \Omega \rightarrow R$  l'image à traiter où  $u_0(x, y)$  est le niveau de gris associé au pixel de coordonnées  $(x, y)$ . Le modèle d'évolution de l'image proposé par Perona et Malik est le suivant :

$$\frac{\partial u}{\partial t}(x, y, t) - \text{div}(g_\lambda(|\nabla u(x, y, t)|)\nabla u(x, y, t)) = 0 \text{ dans } Q_T \quad (1.1)$$

$$\nabla u \cdot \vec{n} = 0 \text{ sur } \partial\Omega \times [0, T] \quad (1.2)$$

$$u(x, y, 0) = u_0 \text{ dans } \Omega \quad (1.3)$$

$u(x, y, t) : \Omega \times [0, T] \rightarrow R$  représente l'état de l'image résultat à un instant  $t$ . Finalement, la solution de l'équation (1.1) n'est autre qu'une version débruité de l'image originale correspondant à un temps d'évolution donné . Au fur et à mesure de l'évolution temporelle, la transformation de l'image  $u_0$  conduit à la réduction du bruit tout en conservant les bords et les contours. Le domaine de l'image  $\Omega$  est une partie rectangulaire bornée dans  $R^2$  et  $\vec{n}$  est le vecteur normal unitaire au bord de l'image  $\partial\Omega$ .

Pour obtenir le résultat recherché, la conductivité  $g_\lambda$  est définie comme une fonction décroissante valant 1 en 0 et tendant vers 0 en  $+\infty$ , dépendante de la norme du vecteur gradient  $|\nabla u|$  et vérifie la condition suivante :

$$g_\lambda(|\nabla u(x, y, t)|) \geq \alpha_0, \forall (x, y, t) \in \Omega \times [0, T] \quad (1.4)$$

où  $\alpha_0$  est une constante strictement positive.

Perona et Malik ont proposé pour la première fois deux types de fonctions de  $g_\lambda$  :

$$g_{1,\lambda}(s) = \exp \left[ -\left(\frac{s}{\lambda}\right)^2 \right] \quad (1.5)$$

$$g_{2,\lambda}(s) = \left[ 1 + \left(\frac{s}{\lambda}\right)^2 \right]^{-1} \quad (1.6)$$

Le paramètre  $\lambda$  appelé, seuil ou barrière de diffusion, permet de paramétrer la valeur du gradient à partir de laquelle le comportement de l'EDP varie entre diffusion et éventuel

rehaussement.

## 1.2 Modèle de Perona-Malik à deux variables

En introduisant une nouvelle fonction  $c_\lambda$  telle que  $\frac{1}{g_\lambda(s)} = 1 + c_\lambda(s^2)$ . Les conditions sur la fonction de conductivité  $g_\lambda$  permettent de conclure que  $c_\lambda$  est une fonction croissante vérifie les conditions suivantes :

$$c_\lambda(0) = 0 \tag{1.7}$$

$$\lim_{s \rightarrow +\infty} c_\lambda(s) = +\infty \tag{1.8}$$

$$c_\lambda(|\nabla u(x, y, t)|) \leq \beta_0, \forall (x, y, t) \in \Omega \times [0, T] \tag{1.9}$$

En ajoutant à l'équation (1.1) une nouvelle variable  $p = g_\lambda(|\nabla u(x, y, t)|)\nabla u(x, y, t)$ , le problème de Perona-Malik s'écrit sous la forme :

$$\frac{\partial u}{\partial t}(x, y, t) - \operatorname{div}(p) = 0 \text{ dans } Q_T \tag{1.10}$$

$$\left( 1 + c_\lambda(|\nabla u(x, y, t)|^2) \right) p - \nabla u(x, y, t) = 0 \text{ dans } Q_T \tag{1.11}$$

$$p \cdot \vec{n} = 0 \text{ sur } Q_T \tag{1.12}$$

$$u(x, y, 0) = u_0 \text{ dans } \Omega \tag{1.13}$$

Ce problème est un système de deux équations aux dérivées partielles et deux inconnus  $u$  et  $p$ , pour le résoudre nous utilisons la méthode des éléments finis mixtes basé sur l'approximation de Taylor-Hood [46].

## 1.3 Discrétisation en temps

Dans cette section, on se propose de résoudre numériquement le problème (1) sur l'intervalle  $[0, T]$ . On considère pour cela un pas de temps discretisation  $\tau = \frac{T}{N} > 0$ , avec  $N$  est un entier strictement positif, et on subdivise l'intervalle du temps  $[0, T]$  en  $N$  sous-intervalles  $[t_i, t_{i+1}]$ ,  $i = 0, \dots, N - 1$  tels que :

$$0 = t_0 < t_1 < \dots < t_N = T, t_i = i\tau, i \in \{0, 1, \dots, N\} \tag{1.14}$$

en utilisant l'approximation de la dérivée  $\frac{\partial u(t)}{\partial t} \simeq \frac{u(t+\tau) - u(t)}{\tau}$ . Les termes non linéaires des équations (1.10), (1.11) sont traités à partir de l'étape précédente tandis que les termes linéaires sont traités à l'échelle actuelle. Le problème approché s'écrit sous la forme :

trouver  $u_{n+1} : \Omega \rightarrow R$  et  $p_{n+1} : \Omega \rightarrow R$  tels que

$$\frac{u_{n+1} - u_n}{\tau} - \operatorname{div}(p_{n+1}) = 0 \text{ dans } \Omega \quad (1.15)$$

$$\left( 1 + c_\lambda(|\nabla u_n|^2) \right) p_{n+1} - \nabla u_{n+1} = 0 \text{ dans } \Omega \quad (1.16)$$

$$p_{n+1} \cdot \vec{n} = 0 \text{ sur } \partial\Omega \quad (1.17)$$

$$u_0 = u^0 \text{ dans } \Omega \quad (1.18)$$

La résolution de ce problème par la méthode des éléments finis mixtes est une démarche à suivre. En commençant par la formulation variationnelle, nous faisons les calculs nécessaires afin de concevoir un algorithme et un code qui visualisent la solution numérique du problème de Perona-Malik. Ainsi, les résultats expérimentaux de débruitage d'images montrent que notre méthode est plus efficace que les autres méthodes utilisées.

## 1.4 Formulation variationnelle

On multiplie les deux membres de l'équation (1.16) par une fonction test  $q \in V = H^1(\Omega)$  et les deux membres de l'équation (1.15) par une fonction  $v \in Q = h_0^1(\Omega)$ , puis on intègre sur  $\Omega$ , on applique la formule de Green [48] et la condition aux limites  $\frac{\partial u}{\partial n} = 0$  sur  $\partial\Omega$ , on obtient alors la formulation variationnelle suivante : Pour  $n = 0, \dots, N$  trouver  $(p_{n+1}, u_{n+1}) \in V \times Q$  tel que,

$$a_n(p_{n+1}, q) + b(q, u_{n+1}) = 0 \quad (1.19)$$

$$b(p_{n+1}, v) - c(u_{n+1}, v) = l_n(v) \quad (1.20)$$

$$u_{t=0} = u^0 \text{ dans } \Omega \quad (1.21)$$

pour tout  $(p, q) \in V \times V$  et  $(u, v) \in Q \times Q$ . Où  $a_n(p, q)$  et  $c(u, v)$  sont des formes bilinéaires symétriques définies respectivement sur  $V \times V$  et  $Q \times Q$ ,  $b(q, u)$  est une forme bilinéaire définie sur  $V \times Q$  et  $l_n(v)$  est une forme linéaire définie sur  $V$  tels que :

$$a_n(p, q) = \int_{\Omega} \left( [1 + c_\lambda(|\nabla u|^2)] \right) p \cdot q dx dy \quad (1.22)$$

$$b(p, v) = \tau \int_{\Omega} v \operatorname{div}(p) dx dy \quad (1.23)$$

$$c(u, v) = \int_{\Omega} u v dx dy \quad (1.24)$$

$$l_n(v) = \frac{1}{\tau} \int_{\Omega} u_n \cdot v dx dy. \quad (1.25)$$

Toutes les conditions de l'existence et l'unicité de la solution faible du système (1.19),(1.20) sont bien établies par la vérification de la continuité et de la coercivité pour les formes bilinéaires  $a_n(p, q) : V \times V \rightarrow R$ ,  $c(u, v) : Q \times Q \rightarrow R$ ,  $b(p, v) : V \times Q \rightarrow R$  et la continuité de la forme linéaire  $l_n(v) : V \rightarrow R$  définies ci-dessus dans (1.22)-(1.25). Pour la preuve voir l'article [Hajouji et al (2018)].

## 1.5 Approximation par éléments finis mixtes

La méthode des éléments finis mixtes a été utilisée pour la première fois dans les années soixante, afin d'écrire les méthodes par éléments finis dans lesquelles les champs de contrainte et de déplacement sont approchés en tant que variables primaires. Nous utilisons la méthode des éléments finis mixtes pour résoudre le modèle ainsi obtenue. La dite méthode consiste à convertir le modèle proposé en un système linéaire. Pour ce faire, nous passons par les trois étapes : Formulation faible, choix de maillage du domaine et discrétisation des équations étudiées. Le système linéaire ainsi obtenu peut être résolu par des méthodes itératives; nous citons dans ce cadre, la méthode de minimum résidu (MINRES) pour les systèmes symétriques, la méthode de minimum résidu généralisée (GMRES) [121] et la méthode stabilisée du gradient bi-conjugué (BICGSTAB) [122] pour les systèmes non symétriques. On s'intéresse maintenant à une approximation du problème (1.19),(1.20) par la méthode des éléments finis mixtes dans le cadre de la méthode de Galerkin standard. Cette méthode consiste à utiliser un maillage du domaine d'étude, et considérer la restriction de la fonction solution sur chaque élément du maillage. Les espaces  $V$  et  $Q$  sont généralement de dimensions infinies, on construit deux espaces d'approximation  $V_h \subset V$  et  $Q_h \subset Q$  avec  $dim V_h$  et  $dim Q_h$  sont finies, et on réécrit le problème (1.19), (1.20) de la façon approchée suivante : Pour  $n = 0, \dots, N$ , trouver  $(p_{n+1}^h, u_{n+1}^h) \in V_h \times Q_h$  telles que,

$$a_n(p_{n+1}^h, q^h) + b(q^h, u_{n+1}^h) = 0 \quad (1.26)$$

$$b(p_{n+1}^h, v^h) - c(u_{n+1}^h, v^h) = l_n(v^h) \quad (1.27)$$

$$u_{t=0}^h = u^0 \text{ dans } \Omega \quad (1.28)$$

pour tout  $(p^h, q^h) \in V_h \times V_h$  et  $(u^h, v^h) \in Q_h \times Q_h$ .

Pour plus d'explications voir [123].

Pour résoudre le problème (2.23) nous construisons une base  $\{\vec{\phi}_i, i = 1, \dots, N\}$  de l'espace vectoriel  $V_h$  et une base  $\{\varphi_i, i = 1, \dots, M\}$  de l'espace  $Q_h$ , tels que  $N$  et  $M$  sont respectivement les dimensions de  $V_h$  et  $Q_h$ . Comme  $p_h \in V_h$  et  $u_h \in Q_h$ , il existe un unique

## 1.6. CONSTRUCTION GÉOMÉTRIQUE DES ÉLÉMENTS DE BASES DES ESPACES D'APPROXIMATIONS

$(x_1, x_2, \dots, x_N) \in R^N$  et unique  $(z_1, z_2, \dots, z_M) \in R^M$  tels que

$$p_{n+1}^h = \sum_{i=1}^N x_i \vec{\phi}_i \quad (1.29)$$

$$u_{n+1}^h = \sum_{i=1}^M z_i \varphi_i \quad (1.30)$$

Nous trouvons alors le système linéaire suivant :

$$\begin{pmatrix} A_n & B \\ B^t & C \end{pmatrix} \begin{pmatrix} X_{n+1} \\ Z_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ L_n \end{pmatrix} \quad (1.31)$$

Où  $A_n$ ,  $B$  et  $C$  sont, respectivement les matrices associées aux formes bilinéaires  $a_n(.,.)$ ,  $b(.,.)$  et  $c(.,.)$ , et  $L_n$  est le vecteur discret associé à la forme linéaire  $l_n(.,.)$ . Avec

$$A_n = (a_n(i, j)); a_n(i, j) = a_n(\vec{\phi}_j, \vec{\phi}_i), \text{ pour } i, j = 1, \dots, N \quad (1.32)$$

$$B = (b_{jk}); b_{jk} = b(\vec{\phi}_j, \varphi_k), \text{ pour } k = 1, \dots, M, j = 1, \dots, N \quad (1.33)$$

$$C = (c_{kl}); c_{kl} = -c(\varphi_k, \varphi_l), \text{ pour } k, l = 1, \dots, M \quad (1.34)$$

$$l_n = \begin{pmatrix} l_n(\varphi_1) \\ l_n(\varphi_2) \\ \vdots \\ l_n(\varphi_M) \end{pmatrix}; X_{n+1} = \begin{pmatrix} x_1^{n+1} \\ x_2^{n+1} \\ \vdots \\ x_N^{n+1} \end{pmatrix} \text{ et } Z_{n+1} = \begin{pmatrix} z_1^{n+1} \\ z_2^{n+1} \\ \vdots \\ z_M^{n+1} \end{pmatrix} \quad (1.35)$$

et les inconnus  $p_{n+1}^h$  et  $u_{n+1}^h$  sont obtenus en substituant les coordonnées des vecteurs de la solution  $X_{n+1}$  et  $Z_{n+1}$  dans les équations (1.29) et (1.30).

## 1.6 Construction géométrique des éléments de bases des espaces d'approximations

Dans le cadre de la construction géométrique des éléments des bases des espaces d'approximations  $V_h$  et  $Q_h$ , nous considérons une discrétisation du domaine  $\Omega$  qui est basée sur la notion de pixels en 2D (voxels en 3D) de forme rectangulaire en général. Nous utilisons cette structure d'image pour subdiviser le domaine  $\Omega$  en  $p$ -rectangles ou  $p$ -triangles uniformes  $T_h = \{K_i, i = 1, \dots, n_T\}$  avec  $T_h$  se compose de triangles ou de rectangles en 2D ( $p = 2$ ) et de tétraèdres ou de parallélépipèdes rectangles en 3D ( $p = 3$ ) (figures 1.1 et 1.2). Cette subdivision est appelée dans la méthode des éléments finis maillage du

### 1.6. CONSTRUCTION GÉOMÉTRIQUE DES ÉLÉMENTS DE BASES DES ESPACES D'APPROXIMATIONS

domaine. Les valeurs des fonctions initialement données, ainsi que celles calculées, sont considérées comme des approximations des valeurs des fonctions au centre des pixels. Les centres des pixels correspondent alors aux noeuds du maillage par éléments finis. Nous pouvons obtenir une telle p-triangulation en reliant les centres des pixels par un nouveau maillage p-rectangulaire, puis en divisant chaque p-rectangle en deux p-triangles : Cela signifie également que, dans ces deux méthodes, le domaine de calcul est donné sous la forme de l'union de tous les p-éléments  $K_i, i = 1, \dots, n_T$  (figures 1.3, 1.4).

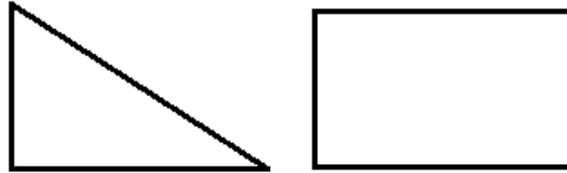


FIGURE 1.1 – un p-triangle (gauche) et un p-rectangle (droite) en 2D ( $p = 2$ ).

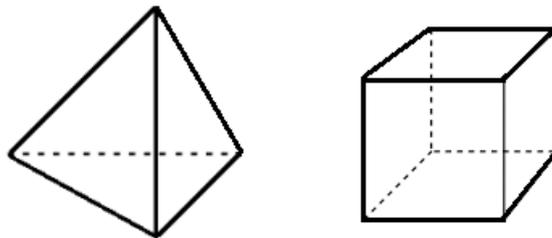


FIGURE 1.2 – un p-triangle (gauche) et p-rectangle (droite) en 3D ( $p = 3$ ).

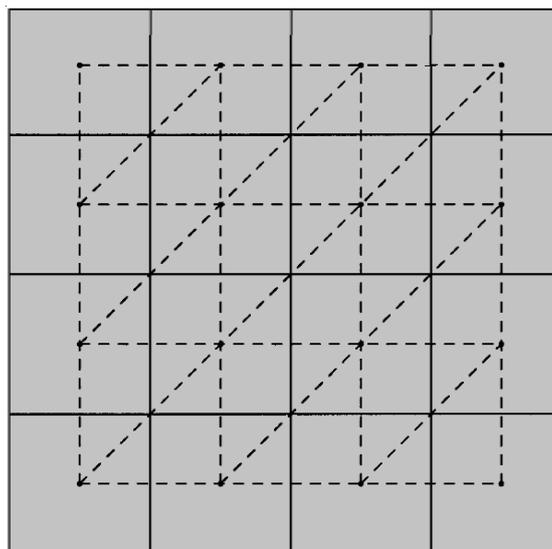


FIGURE 1.3 – Discretisation du domaine  $\Omega$  d'une image 2D.

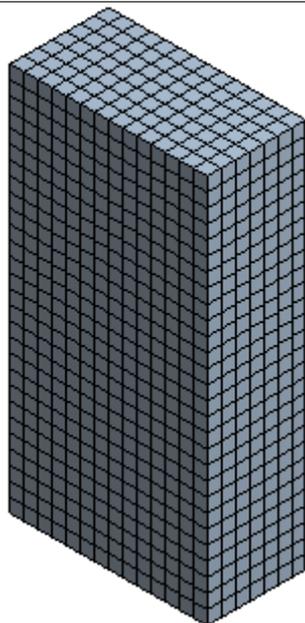


FIGURE 1.4 – Discretisation du domaine  $\Omega$  d'une image 3D.

Le formulation discrète (1.26), (1.27) prend la forme du problème (1.19), (1.20), les espaces d'approximation  $V_h$  et  $Q_h$  sont inclus dans  $V$  et  $Q$  respectivement, et puis toutes les conditions de l'existence et de l'unicité sont vérifiées à l'exception de l'obstacle de la propriété inf-sup [Hjouji et all 2019] qui dépend de la construction géométrique des éléments des bases des espaces d'approximations. Plusieurs couples d'éléments finis ont franchi cet obstacle comme les éléments finis de Raviart-Thomas Nedelec d'indice  $k$ , les éléments finis de Brezzi-Douglas Marini d'indice  $k$ ; ou les éléments finis de Brezzi-Douglas-Fortin-Marini d'indice  $k + 1$ ; correspondant à une triangulation affine d'éléments finis, voir [123] pour la définition spécifique de ces espaces. Nous nous interessons ici à celui des éléments finis de Taylor-Hood  $P_2 - P_1$  et  $Q_2 - Q_1$  (figures 1.5, 1.8) et la technique des macroéléments présentée dans [35].

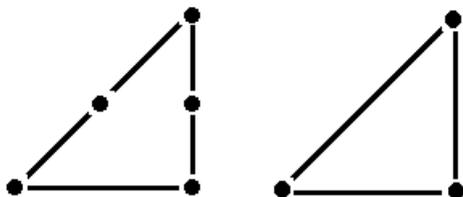


FIGURE 1.5 – élément fini de Taylor-Hood  $P_2 - P_1$  en dimension 2.

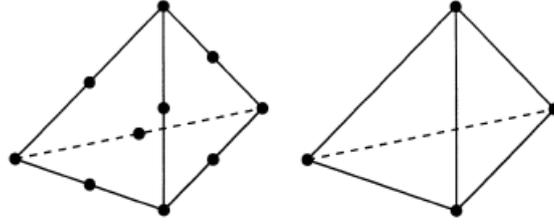


FIGURE 1.6 – élément fini de Taylor-Hood  $P_2 - P_1$  en dimension 3.

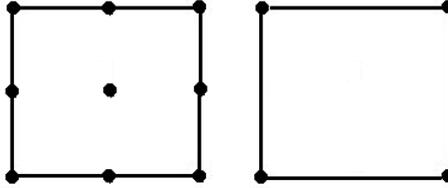


FIGURE 1.7 – élément fini de Taylor-Hood  $Q_2 - Q_1$  en dimension 2.

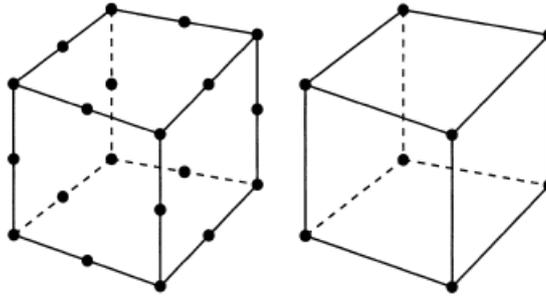


FIGURE 1.8 – élément fini de Taylor-Hood  $Q_2 - Q_1$  en dimension 3.

En se basant sur la technique mentionnée, si le maillage  $T_h$  sous forme de p-triangles, nous proposons les espaces d'approximations suivants :

$$V_h = \{ \vec{q} \in V; \vec{q}|_K \in [P_2(K)]^p, K \in R_h \} \quad (1.36)$$

$$Q_h = \{ q \in C(\Omega) \cap L_0^2(\Omega); v|_K \in Q_1(K), K \in T_h \} \quad (1.37)$$

où  $P_n(K) = \{ p : K \rightarrow R; p(x, y) = \sum_{i,j} \alpha_{ij} x^i y^j, i + j \leq n \}$  désigne l'ensemble des polynômes de deux variables de degré total inférieur ou égal à  $n$ . Et si  $T_h$  est une p-triangulation, nous considérons les espaces suivants :

$$V_h = \{ \vec{q} \in V; \vec{q}|_K \in [Q_2(K)]^p, K \in R_h \} \quad (1.38)$$

$$Q_h = \{ q \in C(\Omega) \cap L_0^2(\Omega); v|_K \in Q_1(K), K \in T_h \} \quad (1.39)$$

où  $Q_n(K) = \{ p : K \rightarrow R; p(x, y) = \sum_{i,j} \alpha_{ij} x^i y^j, i \leq n \text{ et } j \leq n \}$  désigne l'ensemble des polynômes de deux variables de degré partiel inférieur ou égal à  $n$ .

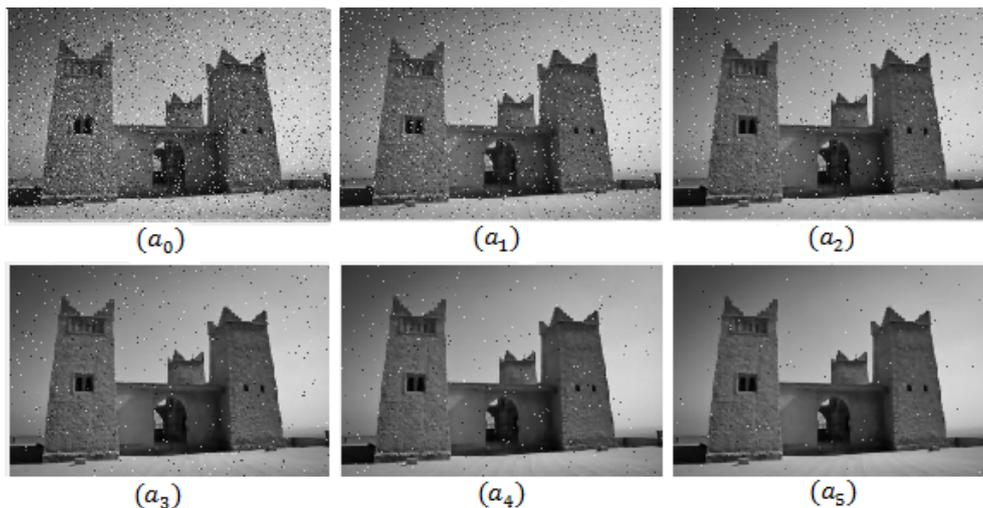


FIGURE 1.9 – L'image originale ( $a_0$ ) et les images débruitées après les itérations 5 ( $a_1$ ), 10 ( $a_2$ ), 15 ( $a_3$ ), 20 ( $a_4$ ) et 30 ( $a_5$ )

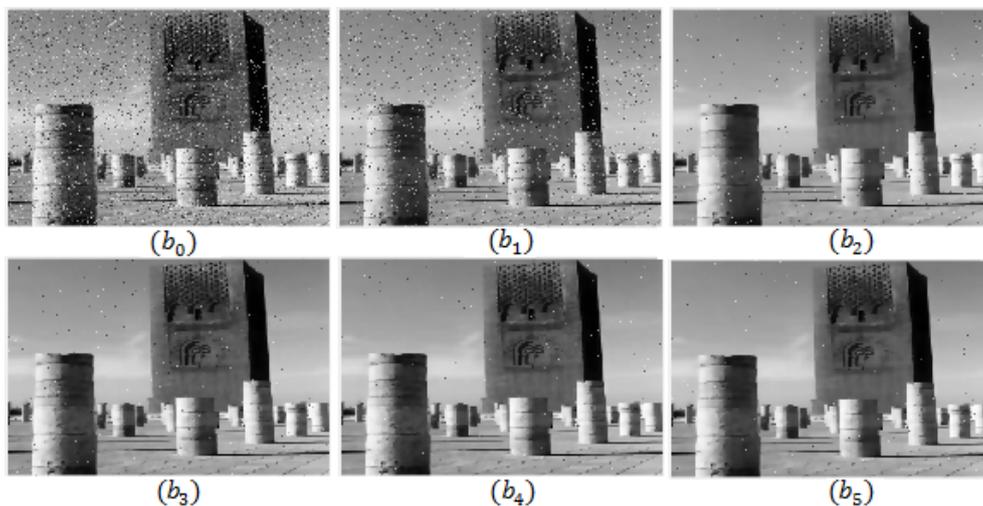


FIGURE 1.10 – l'image originale ( $b_0$ ) et les images débruitées après les itérations 5 ( $b_1$ ), 10 ( $b_2$ ), 15 ( $b_3$ ), 20 ( $b_4$ ) et 30 ( $b_5$ )

## 1.7 Résultats expérimentaux

Cette section est consacrée aux nombreuses expérimentations numériques. Le programme s'arrêtera lorsqu'il atteindra notre objectif. La plupart des paramètres d'algorithme sont choisis de manière heuristique pour que les algorithmes fonctionnent de manière optimale. Nous prenons  $h = 1$ ,  $\tau = 10^{-3}s$  et  $c_\lambda(s) = (\lambda s)^2$ . Pour les testes d'améliorations d'images. Nous présentons les résultats de débruitage d'images de notre algorithme en choisissant le paramètre  $\lambda = 20$ , nous appliquons la méthode des éléments finis mixte pour résoudre la modéle de Perona-Malik. Nous donnons ici quelques exemples des images 2D et 3D pour tester la méthode de débruitage proposée. Toutes les sections

précédentes me permet d'écrire l'algorithme 1 permettant de filtré une image bruitée ou au moins améliorer sa qualité.

---

**Algorithm 1** : Calculer  $u_{n+1}^h$

---

1. Initialisation de  $Z_0$ , et  $A_0$  et  $L_0$

- $Z_0 = \left( Z_0(i) \right)_{i=1, \dots, M}$ , avec  $Z_0(i) = u_0(H_i)$  et  $u_0$  est l'image bruitée.
- $A_0 = \left( A_0(i, j) \right)_{i, j=1, \dots, m}$  est la matrice définie par ses coefficients  $A_0(i, j)$ , où
 
$$A_0(i, j) = \int_{\Omega} \left( 1 + c_{\lambda} (| \sum_{k=1}^M Z_0(k) \nabla \varphi_k |) \right) \vec{\phi}_j \cdot \vec{\phi}_j d\mu .$$
- $L_0 = \left( L_0(i) \right)_{i=1, \dots, M}$  est le vecteur définie par
 
$$L_0(k) = \sum_{i=1}^M Z_0(i) \int_{\Omega} \varphi_i \varphi_k d\mu .$$
- $Z \leftarrow Z_0$
- $A \leftarrow A_0$
- $L \leftarrow L_0$
- $X \leftarrow 0$

2. pour  $i = 1$  à  $N$  faire

- résoudre le système par la méthode GMRES  $\begin{pmatrix} A & B \\ B^t & C \end{pmatrix} \begin{pmatrix} X' \\ Z' \end{pmatrix} = \begin{pmatrix} 0 \\ L \end{pmatrix}$
- $A' = \left( A'(i, j) \right)_{i, j=1, \dots, m}$ , avec
 
$$A'(i, j) = \int_{\Omega} \left( 1 + c_{\lambda} (| \sum_{k=1}^M Z'(k) \nabla \varphi_k |) \right) \vec{\phi}_j \cdot \vec{\phi}_j d\mu .$$
- $L' = \left( L'(i) \right)_{i=1, \dots, M}$ , avec
 
$$L'(k) = \sum_{i=1}^M Z'(i) \int_{\Omega} \varphi_i \varphi_k d\mu .$$
- $A \leftarrow A'$
- $L \leftarrow L'$
- $Z \leftarrow Z'$
- $X \leftarrow X'$

3. Afficher l'image obtenue  $u_h(x, y) = \sum_{i=1}^M Z(i) \varphi_i(x, y)$ .

---

Nous construisons le vecteur initial  $Z_0$  mentionné dans l'algorithme 1 de la manière suivante :

$$Z_0 = \left( u_0(H_1), u_0(H_2), \dots, u_0(H_M), \right)^t \quad (1.40)$$

où  $u_0$  est l'image initiale (l'image bruitée) de taille  $M = n \times m$  et  $u_0(H_i)$  est la valeur de niveau de gris au pixel numéro  $i$ .

Pour montrer l'efficacité de notre méthode, nous avons effectué deux expériences expérimentaux. Dans la première, nous testons l'efficacité de notre algorithme pour filtrer des images 2D (figures 1.9-1.12) et des images 3D (figures 1.13-1.15). Ces images sont corrompues par des bruits de sel et de poivre de densité  $d$  varie de 0.01 à 0.05. Dans la seconde expérience, nous conserverons les mêmes valeurs que celles des paramètres précédents et nous présentons une étude comparative de la performance de notre technique de débruitage qui est basée sur la méthode des éléments finis mixtes (MFEM) avec d'autres techniques qui sont basées sur la méthode de différence finis (MDF), la méthode de volumes finis (MVF) et la méthode des éléments finis classique (MEFC). Pour mesurer la performance du débruitage, nous utilisons l'erreur absolue moyenne (MAE) et le rapport signal sur bruit de pointe (PSNR) qui sont définies par :

**Pour les images 2D**

$$MAE = \frac{\|I(x, y) - I_0(x, y)\|_1}{M \times N} \quad (1.41)$$

$$PSNR = 10 \log_{10} \left( \frac{255^2}{\frac{1}{MN} \|I(x, y) - I_0(x, y)\|_2^2} \right) \quad (1.42)$$

Où  $I$  et  $I_0$  sont respectivement l'image originale 2D (image nette, avant d'ajouter le bruit) et l'image débruité de taille  $M \times N$ .

**Pour les images 3D**

$$MAE = \frac{\|I(x, y) - I_0(x, y)\|_1}{M \times N \times P} \quad (1.43)$$

$$PSNR = 10 \log_{10} \left( \frac{255^3}{\frac{1}{MNP} \|I(x, y, z) - I_0(x, y, z)\|_2^2} \right) \quad (1.44)$$

Où  $I$  et  $I_0$  sont respectivement l'image originale 3D (image nette, avant d'ajouter le bruit) et l'image débruité de taille  $M \times N \times P$ , et les normes  $\|\cdot\|_1$  et  $\|\cdot\|_2$  sont définies par les

relations suivantes :

$$\|I(x, y) - I_0(x, y)\|_1 = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |I(x, y) - I_0(x, y)| \quad (1.45)$$

$$\|I(x, y) - I_0(x, y)\|_2 = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [I(x, y) - I_0(x, y)]^2 \quad (1.46)$$

$$\|I(x, y, z) - I_0(x, y, z)\|_1 = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{P-1} |I(x, y, z) - I_0(x, y, z)| \quad (1.47)$$

$$\|I(x, y, z) - I_0(x, y, z)\|_2 = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{P-1} [I(x, y, z) - I_0(x, y, z)]^2 \quad (1.48)$$

Pour mèsurer la capacité de restauration de la méthode proposée, le rapport signal sur bruit (SNR) est appliqué et noté :

$$SNR = 10 \log_{10} \left( \frac{\sigma_u}{\sigma_n} \right) \quad (1.49)$$

où  $\sigma_u$  et  $\sigma_n$  sont respectivement les écarts-types de signal et de bruit. Les valeurs de ces mesures statistiques indiquent la force du signal dans les images restaurées. Par conséquent, les valeurs  $PSNR$  et  $SNR$  augmentent et la valeur de  $MAE$  diminue lorsque la version restaurée de l'image se rapproche de celle d'originale. Les résultats obtenus sont présentés dans les tableaux des figures (1.16-1.19). Les résultats visuels de la comparaison montrent l'efficacité et la supériorité de la méthode proposée pour le débruitage d'images 2D et 3D comparée à celles de MEFM et de MVF. Lors des tests précédents, nous nous sommes concentrés uniquement sur le bruit du sel et du poivre. Pour montrer que notre méthode est valable pour les différents types de bruit, nous avons traité un autre test pour les bruits "gaussien", de "poisson" et de "speckle" (Fig. 1.20). À partir de ce test, nous pouvons conclure que notre algorithme est très efficace pour tous les types de bruit.

## 1.8 Conclusion

Le modèle Perona est un outil efficace pour le débruitage d'image tout en préservant les contours, les régions, les limites et les textures qui jouent un rôle important en reconnaissance de formes des objets. Le filtrage non linéaire de Perona-Malik a l'avantage de construire des zones plates dans l'image, même quand le contraste est faible. La fonction d'arrêt du processus contient le paramètre  $\lambda$ . Ce paramètre correspond à la hauteur des

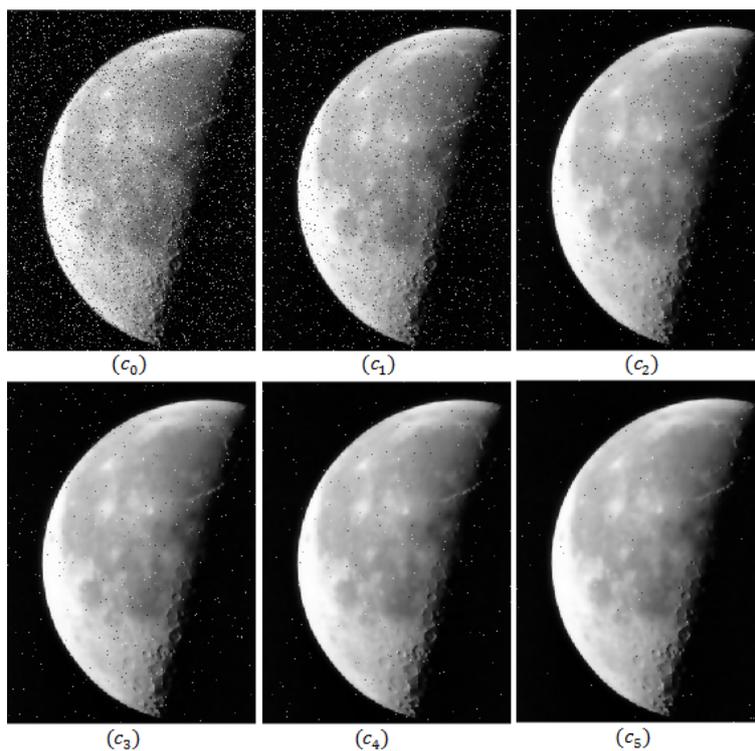


FIGURE 1.11 – l'image originale ( $c_0$ ) et les images débruitées après les itérations 5 ( $c_1$ ), 10 ( $c_2$ ), 15 ( $c_3$ ), 20 ( $c_4$ ) et 30 ( $c_5$ )

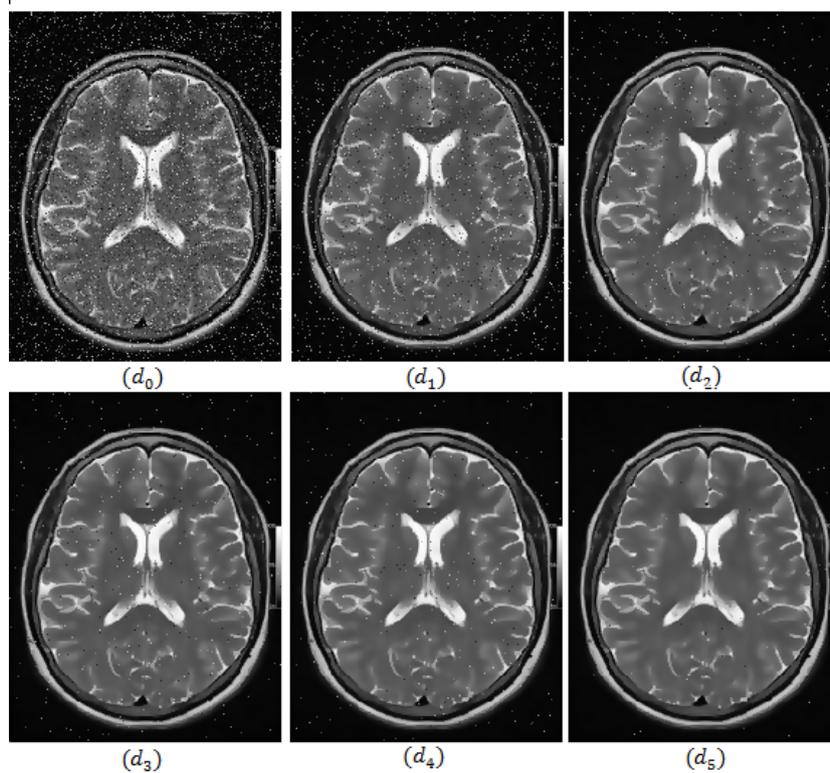


FIGURE 1.12 – l'image originale ( $d_0$ ) et les images débruitées après les itérations 5 ( $d_1$ ), 10 ( $d_2$ ), 15 ( $d_3$ ), 20 ( $d_4$ ) et 30 ( $d_5$ )

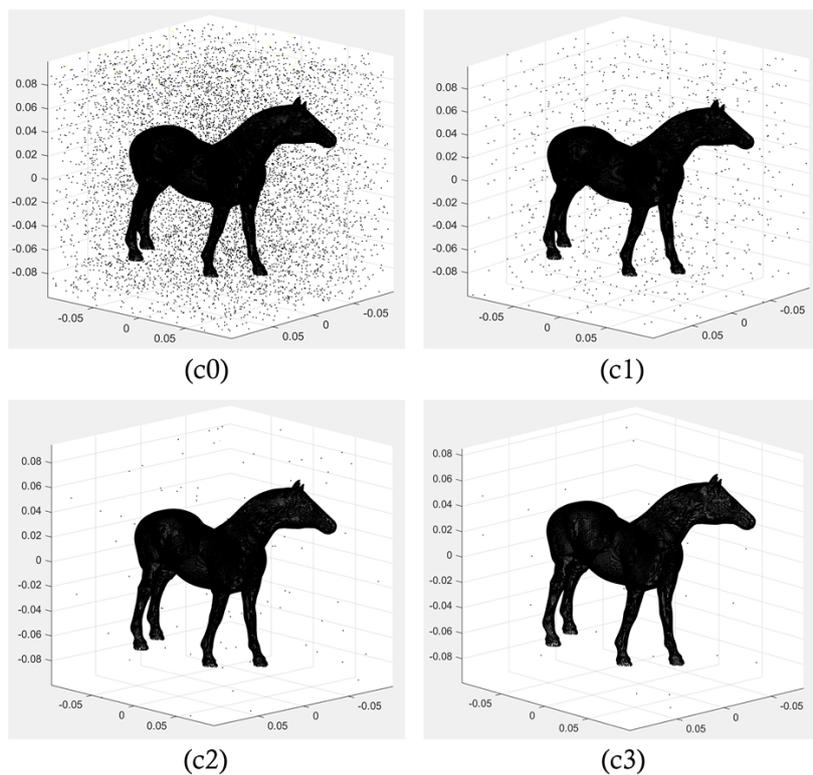


FIGURE 1.13 – l'image originale ( $c_0$ ) et les images débruitées après les itérations 5 ( $c_1$ ), 20 ( $c_2$ ) et 30 ( $c_3$ )

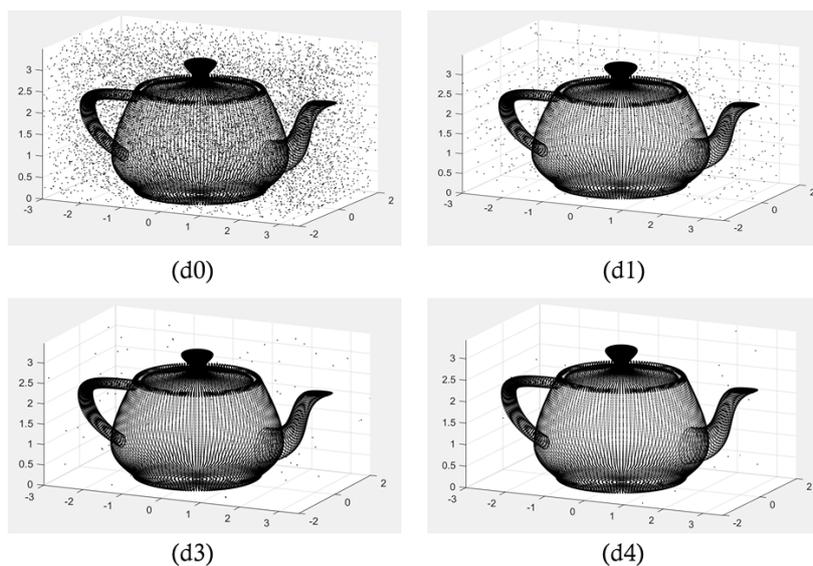


FIGURE 1.14 – l'image originale ( $d_0$ ) et les images débruitées après les itérations 5 ( $d_1$ ), 20 ( $d_2$ ) et 30 ( $d_3$ )

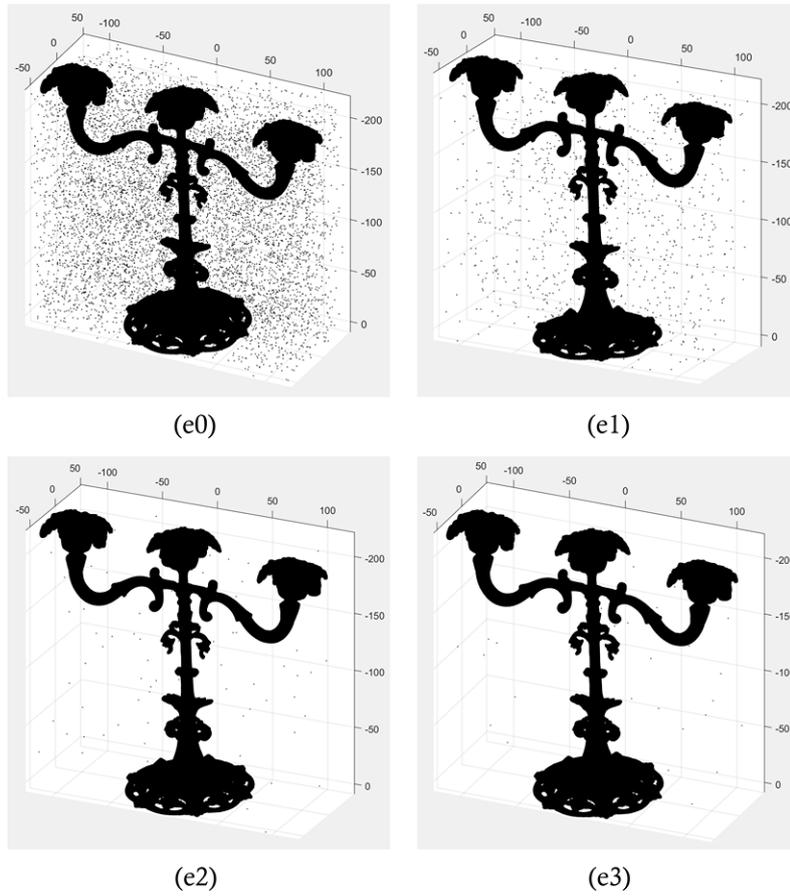


FIGURE 1.15 – l'image originale ( $e_0$ ) et les images débruitées après les itérations 5 ( $e_1$ ), 20 ( $e_2$ ) et 30 ( $e_3$ )

## 1.8. CONCLUSION

FIGURE 1.16 – Comparaisons de MAE et PSNR pour la résolution du modèle de Perona-Malik en utilisant la méthode des éléments finis mixtes (MEFM), la méthode des éléments finis (MEF) classique et la méthode des volumes finis (MVF) pour l'image (a)

Noise density $d$ and parameter $\lambda$	Scale steps	FEM			FVM			MFEM		
		MAE	PSNR	SNR	MAE	PSNR	SNR	MAE	PSNR	SNR
$d = 0.01 \lambda = 20$	5	5.7509	21.7215	09.6120	6.8907	20.3105	09.6427	4.8801	22.4210	10.6120
	10	3.9704	24.3171	11.0018	4.1521	23.3410	12.7914	3.8165	26.8271	12.8364
	20	2.0129	37.1114	15.1242	3.3915	36.5167	17.1795	1.4304	41.1204	16.6018
	30	1.7312	42.0541	21.1100	2.0026	39.1523	20.0730	0.9834	47.3091	27.8461
$d = 0.01 \lambda = 30$	5	5.6130	17.3105	08.6120	5.5781	18.0459	08.3251	4.8130	20.3105	10.6120
	10	4.4417	19.1923	09.1123	4.9931	21.1123	08.9178	4.0021	28.5127	14.6612
	20	3.0427	22.0249	12.1212	2.7452	24.7532	12.07512	2.0311	30.1218	15.1217
	30	1.8113	23.3491	12.4791	0.9045	29.1531	12.0018	0.4441	31.4207	16.2546
$d = 0.05 \lambda = 20$	5	7.0212	31.1245	15.1123	8.9901	32.1211	14.1558	4.6820	34.1414	16.9147
	10	5.1620	36.0211	17.7703	7.0912	33.1012	17.0119	4.0901	39.0125	19.2154
	20	3.4445	39.1425	17.8019	4.0012	38.0016	18.0145	2.0517	42.2530	21.4587
	30	3.0096	42.1142	22.0117	3.0112	46.1699	21.1889	1.1704	46.2257	23.1110
$d = 0.05 \lambda = 30$	5	8.9232	26.1874	11.1013	7.0062	22.1180	10.0224	4.7082	30.0215	16.1222
	10	5.0008	29.1961	13.827	6.1232	23.2531	12.1405	2.7264	30.7102	17.1751
	20	4.1113	28.8463	15.1407	4.9426	26.0236	12.8684	2.0821	32.9954	17.4101
	30	4.0012	27.9701	16.0034	4.6015	30.1543	16.1221	1.8244	34.8011	18.1125

FIGURE 1.17 – Comparaisons de MAE et PSNR pour la résolution du modèle de Perona-Malik en utilisant la méthode des éléments finis mixtes (MEFM), la méthode des éléments finis (MEF) classique et la méthode des volumes finis (MVF) pour l'image (b)

Noise density $d$ and parameter $\lambda$	Scale steps	FEM			FVM			MFEM		
		MAE	PSNR	SNR	MAE	PSNR	SNR	MAE	PSNR	SNR
$d = 0.01 \lambda = 20$	5	4.5096	25.2157	11.1207	6.1208	19.1103	10.4271	3.018	28.2101	14.1207
	10	2.7047	26.1710	12.8124	5.5214	21.4114	11.9148	2.0169	30.2718	15.3641
	20	2.1297	26.1145	15.2428	4.9158	24.7767	14.7914	1.5342	32.2049	17.0081
	30	1.3128	31.5987	19.1564	3.2147	28.5212	15.7309	0.8817	37.0012	17.4965
$d = 0.01 \lambda = 30$	5	6.0154	18.1061	10.1205	7.7842	16.4591	09.2513	4.1306	24.1058	12.1209
	10	6.0652	21.9232	09.1232	5.9310	19.1238	10.1781	4.0212	28.1271	15.1244
	20	3.1112	25.2508	11.2140	4.4513	23.5329	11.5512	1.3115	29.2145	18.1747
	30	2.1145	26.4110	12.7812	2.9475	24.5318	11.8718	0.7481	31.0214	18.6541
$d = 0.05 \lambda = 20$	5	5.2124	29.1245	14.2314	8.0199	23.2117	11.5751	4.8128	31.1414	16.1479
	10	3.6120	28.0211	15.0317	5.9214	23.0129	12.1259	2.0008	37.0125	21.1543
	20	2.3641	28.4255	17.0197	4.1204	26.8164	14.4501	1.0517	48.5306	21.9507
	30	1.9614	32.0044	22.1970	3.1562	30.6899	17.8852	0.1744	51.2587	28.2118
$d = 0.05 \lambda = 30$	5	5.1124	26.8874	12.0138	7.2145	22.2217	10.2284	3.0827	31.0215	17.2247
	10	5.0142	30.9617	13.279	6.202	23.5312	12.4051	1.9261	31.1028	18.1410
	20	4.1523	33.4613	15.4072	4.4261	26.2361	12.9604	1.4822	34.9541	18.0121
	30	3.2145	33.9811	16.3447	3.5014	28.5432	16.2210	0.7249	38.0119	19.1250

## 1.8. CONCLUSION

FIGURE 1.18 – Comparaisons de MAE et PSNR pour la résolution du modèle de Perona-Malik en utilisant la méthode des éléments finis mixtes (MEFM), la méthode des éléments finis (MEF) classique et la méthode des volumes finis (MVF) pour l'image (c)

Noise density $d$ and parameter $\lambda$	Scale steps	FEM			FVM			MFEM		
		MAE	PSNR	SNR	MAE	PSNR	SNR	MAE	PSNR	SNR
$d = 0.01; \lambda = 20$	5	6.0112	23.1112	10.4025	7.0021	21.2154	07.6427	4.1458	29.4502	14.4025
	10	4.2345	28.0128	11.5189	5.5620	22.4142	08.7582	2.8142	33.5231	17.1450
	20	3.1241	32.5123	17.3247	5.0214	29.5131	10.2541	1.0015	33.9150	17.6062
	30	2.9235	33.9101	21.8107	4.0147	30.3159	12.4425	0.5209	41.1221	21.0457
$d = 0.01 \lambda = 30$	5	10.4502	18.3005	08.4561	10.6451	17.4402	08.3251	5.2301	19.0114	9.9017
	10	7.1254	22.0226	10.1214	9.4401	19.0658	9.0550	4.6257	25.0554	14.0146
	20	7.0142	22.7510	10.7899	8.1887	19.7114	9.7744	2.1235	28.1204	18.7500
	30	2.0991	24.8251	15.0121	2.9400	24.0599	12.1428	0.2550	38.5901	19.1450
$d = 0.05 \lambda = 20$	5	6.8012	29.8213	14.2541	9.1102	22.1425	11.2215	3.0125	31.4152	17.0326
	10	4.7561	34.0091	17.0717	7.3331	24.0147	13.0152	3.0007	39.4450	19.5567
	20	4.1008	35.4255	18.1221	5.1471	30.1592	15.2157	1.4581	43.0048	25.0583
	30	3.6142	40.2213	23.2310	4.0011	31.2215	21.4512	0.0708	44.1975	29.8124
$d = 0.05 \lambda = 30$	5	7.8324	28.1874	14.2015	7.62540	27.9115	26.6217	5.7112	32.1247	17.1204
	10	6.2520	31.4561	14.8701	7.0224	28.0587	28.0124	3.0128	34.1952	17.7518
	20	6.0124	32.3862	16.4950	6.2214	31.1322	32.1684	2.1025	36.1258	17.1015
	30	3.8112	35.2714	16.8931	3.8112	32.2235	34.2503	0.5402	39.0014	18.1259

FIGURE 1.19 – Comparaisons de MAE et PSNR pour la résolution du modèle de Perona-Malik en utilisant la méthode des éléments finis mixtes (MEFM), la méthode des éléments finis (MEF) classique et la méthode des volumes finis (MVF) pour l'image 3D (b)

Noise density $d$ and parameter $\lambda$	Scale steps	FEM			FVM			MFEM		
		MAE	PSNR	SNR	MAE	PSNR	SNR	MAE	PSNR	SNR
$d = 0.01 \lambda = 20$	5	4.5096	25.2157	11.1207	6.1208	19.1103	10.4271	3.018	28.2101	14.1207
	10	2.7047	26.1710	12.8124	5.5214	21.4114	11.9148	2.0169	30.2718	15.3641
	20	2.1297	26.1145	15.2428	4.9158	24.7767	14.7914	1.5342	32.2049	17.0081
	30	1.3128	31.5987	19.1564	3.2147	28.5212	15.7309	0.8817	37.0012	17.4965
$d = 0.01 \lambda = 30$	5	6.0154	18.1061	10.1205	7.7842	16.4591	09.2513	4.1306	24.1058	12.1209
	10	6.0652	21.9232	09.1232	5.9310	19.1238	10.1781	4.0212	28.1271	15.1244
	20	3.1112	25.2508	11.2140	4.4513	23.5329	11.5512	1.3115	29.2145	18.1747
	30	2.1145	26.4110	12.7812	2.9475	24.5318	11.8718	0.7481	31.0214	18.6541
$d = 0.05 \lambda = 20$	5	5.2124	29.1245	14.2314	8.0199	23.2117	11.5751	4.8128	31.1414	16.1479
	10	3.6120	28.0211	15.0317	5.9214	23.0129	12.1259	2.0008	37.0125	21.1543
	20	2.3641	28.4255	17.0197	4.1204	26.8164	14.4501	1.0517	48.5306	21.9507
	30	1.9614	32.0044	22.1970	3.1562	30.6899	17.8852	0.1744	51.2587	28.2118
$d = 0.05 \lambda = 30$	5	5.1124	26.8874	12.0138	7.2145	22.2217	10.2284	3.0827	31.0215	17.2247
	10	5.0142	30.9617	13.279	6.202	23.5312	12.4051	1.9261	31.1028	18.1410
	20	4.1523	33.4613	15.4072	4.4261	26.2361	12.9604	1.4822	34.9541	18.0121
	30	3.2145	33.9811	16.3447	3.5014	28.5432	16.2210	0.7249	38.0119	19.1250

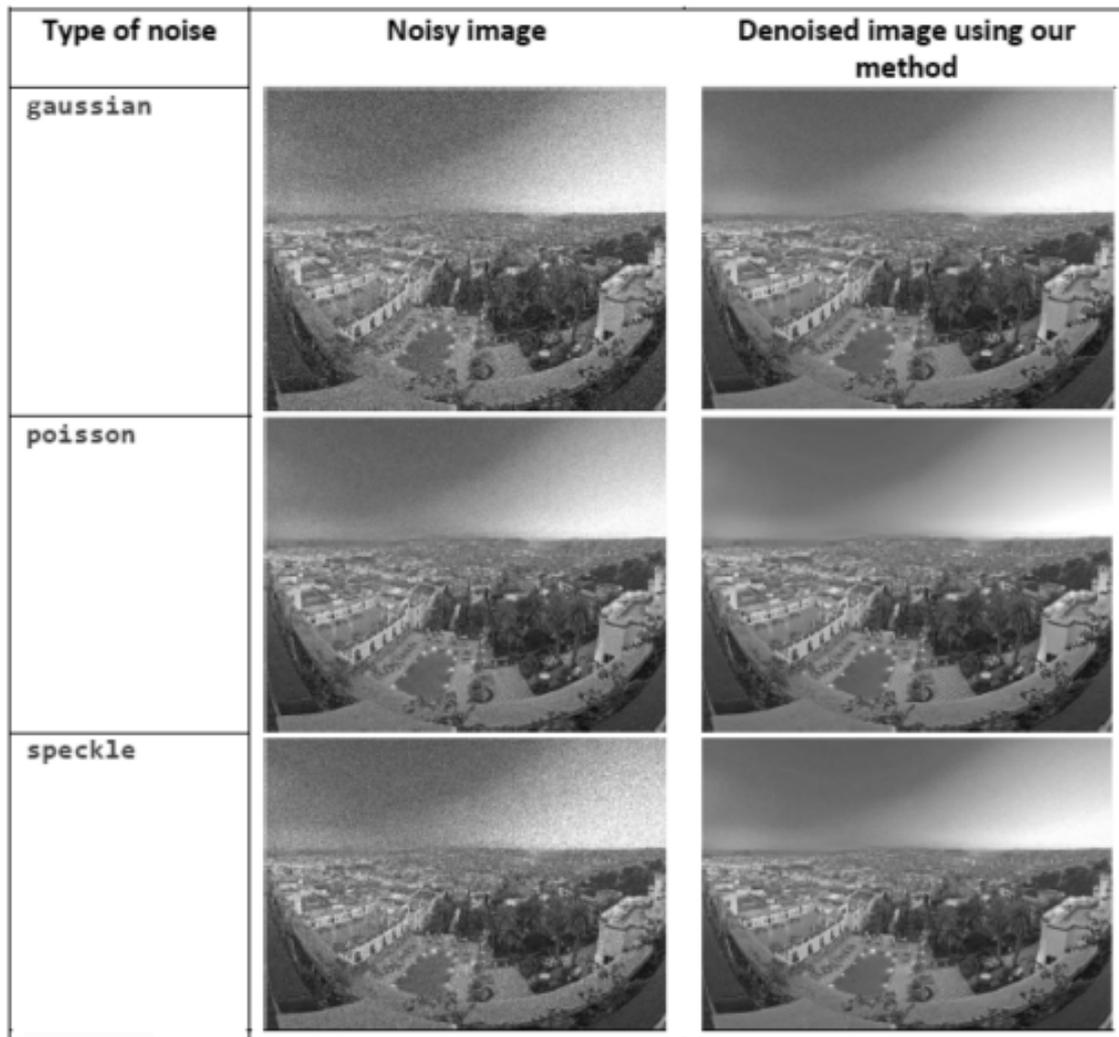


FIGURE 1.20 – Débruitage d’image pour les trois types de bruit (gaussien, poisson et speckle) en utilisant notre méthode

contours à préserver. Ce filtre diminue la sur-segmentation résultante de l’application de méthodes de détection de contours. Ce type de filtrage est particulièrement intéressant lorsque les images à analyser sont bruitées. En effet le bruit induit de multiples bassins versants non-significatifs. Un moyen aisé de les supprimer est d’effectuer un filtrage passe-bas. Ce filtre a l’inconvénient d’atténuer les contours. Pour résoudre les problèmes issus d’une diffusion isotrope, la première idée de diffusion anisotrope a été proposée par Malik et Perona . La motivation essentielle de l’utilisation de modèles basés sur la diffusion est la construction d’un opérateur de diffusion dépendant des propriétés locales de l’image. L’idée est de créer un filtre qui conserve les régions locales et les contours qui floutent les zones à faible gradient. Pour résoudre ce modèle, plusieurs auteurs ont appliqué la méthode des différences finies, les méthodes des volumes finis, les méthodes des éléments finis classiques et les méthode des volumes complémentaires [3, 20, 21, 8, 22, 23, 24, 25].

Dans ce travail nous avons utilisé les méthodes des éléments finis mixtes. Pour montrer l'efficacité de notre algorithme nous avons effectué plusieurs tests expérimentaux qui montre la supériorité et l'efficacité de la méthode proposée. Enfin, nous avons effectué la première étape de l'extraction de notre vecteur descripteur d'image pour la reconnaissance de formes et la classification des bases de données d'images bruitées ou déformées géométriquement. Dans le second chapitre nous expliquons la deuxième étape de notre approche de classification et de reconnaissance de formes qui est basée sur l'extraction des vecteurs descripteurs des images en utilisant la théorie des moments orthogonaux.

# Chapitre 2

## Extraction des vecteurs descripteurs des images en utilisant la théorie des moments orthogonaux

L'extraction de vecteurs descripteurs des images est l'une des étapes les plus importantes en reconnaissance de formes, elle vise l'extraction de l'information la plus pertinente qui caractérise chaque image. Dans ce processus, les caractéristiques appropriées sont extraites des valeurs pour former des vecteurs descripteurs. Ces vecteurs permettent généralement de mieux rendre compte de certaines propriétés visuelles de l'image utilisées pour des traitements ultérieurs entrant dans le cadre d'applications telles que la détection d'objets, la classification, l'indexation et recherche d'images par le contenu. Ils sont utilisés alors par les classificateurs pour reconnaître l'élément d'entrée c.à.d. l'image. L'objectif principal de l'extraction de vecteurs caractéristiques des images est d'extraire un ensemble de caractéristiques qui maximise le taux de reconnaissance d'image avec le minimum d'éléments et qui génère un ensemble de caractéristiques similaires pour divers instances du même symbole. Dans la littérature il y a plusieurs vecteurs descripteurs qui ont été utilisés pour la représentation et la reconnaissance d'images, on peut les classer en trois grands types : Les descripteurs de couleurs, de textures et de formes.

### 2.1 Types de descripteurs d'images

Les images sont des objets numériques très riches en termes d'informations. En plus de l'espace mémoire gigantesque exigé, la manipulation directe de ces images dans un système de reconnaissance d'images par le contenu ne permet pas d'obtenir des temps de réponse réalistes. Il convient donc d'utiliser une représentation de dimension réduite

pour caractériser le contenu de ces images. L'objectif principal de l'extraction de descripteurs est de déterminer pour chaque image, une représentation (signature) qui soit, d'une part compacte, pour être rapidement accessible et facilement comparable, et d'autre part suffisamment complète pour bien caractériser l'image. Il est recommandé d'employer des descripteurs invariants aux transformations colorimétriques tel le changement d'éclairage et aux transformations géométriques tels la translation, la rotation et le changement d'échelle. Ceci permet de pallier aux différentes transformations que peut subir une image. Cependant, la caractérisation robuste et discriminante des images reste un grand défi en traitement d'images. Généralement, les descripteurs dits de bas niveau sont souvent les plus utilisés pour la description d'images par le contenu. Ces attributs décrivent les principales caractéristiques visuelles existant dans une image, à savoir la couleur, la texture et la forme.

### 2.1.1 La couleur

Le physicien James Clerk Maxwell [138] a prouvé que toute couleur, étant initialement une sensation provoquée par la lumière avec l'oeil, est une synthèse de trois couleurs seulement : le rouge, le vert et le bleu. Grâce cette découverte en 1865, la colorimétrie (la science qui étudie la couleur) a vu le jour. Les attributs couleurs sont les premiers utilisés dans les systèmes de reconnaissance d'images par le contenu et ils sont toujours les plus utilisés grâce à leur simplicité d'extraction, leur richesse de description et leur efficacité de reconnaissance. Bimbo et all [139] ont fourni une étude étendue des différentes méthodes employées pour l'extraction de descripteurs couleurs. Un ensemble de descripteurs couleurs a été testé pour être inclus dans le standard MPEG-7. Ces attributs couleurs dépendent directement de l'espace couleur utilisé pour la représentation de la couleur de l'image. Dans la littérature, plusieurs espaces couleurs sont étudiés. Défini en 1931 par la compagnie internationale de l'éclairage (C.I.E), l'espace couleur RVB (Rouge Vert et Bleu) représente les couleurs par synthèse sous tractive. Chaque couleur est représentée par trois composantes : rouge, vert et bleu. L'espace RVB est généralement l'espace couleur de base dans lequel les images sont capturées. Les autres espaces couleurs représentent des transformations directes de l'espace RVB. Ces transformations ont pour objectif d'améliorer la représentativité colorimétrique de l'image et d'assurer des mesures de distance plus linéaires. Les espaces couleurs HSI, HSV, HSB et HLS (appelés communément HSx) sont plus proches de la perception humaine de couleur que l'espace couleur RVB. Pour plus de détails concernant les espaces couleurs, nous recommandons au lecteur intéressé de consulter la référence suivante [138].

### 2.1.2 La texture

La texture est liée aux apparences de surface des images ce qui est d'une grande importance dans tout domaine relatif à la perception visuelle. Dans la littérature, nous trouvons plusieurs définitions de la texture. Nous en citons parmi lesquelles : « la texture est l'attribut qui représente la distribution spatiale des niveaux de gris des pixels d'une région » . Ou encore : « la texture est l'ensemble des propriétés de voisinage locales des niveaux de gris d'une région de l'image » [141]. Bien qu'il n'y ait pas jusqu'à présent une définition précise de la texture, toutes les définitions possibles affirment que la texture est l'information générée par les variations des niveaux de gris entre des pixels voisins au sein d'une image. Les descripteurs textures sont de plus en plus utilisés dans la description d'images car ils atténuent certains problèmes liés à la description couleur. En effet, la description texture est très efficace, surtout dans le cas d'une distribution couleur très proche. Les attributs textures sont divisés en deux catégories : La première est déterministe et fait référence à une répétition spatiale d'un motif de base dans différentes directions. Cette approche structurelle correspond à une vision macroscopique des textures. La seconde approche, dite microscopique, est probabiliste et cherche à caractériser l'aspect chaotique qui ne comprend ni motif localisable, ni fréquence de répétition principale.

### 2.1.3 La forme

Contrairement aux descripteurs couleurs et textures qui s'intéressent à la description du contenu général de l'image. Les descripteurs de formes sont capables de caractériser les différents objets contenus dans l'image. Généralement, ce type de descripteur indique l'aspect général d'un objet, comme son contour, donc une segmentation sous forme de traitement préliminaire de l'image est souvent nécessaire. Deux catégories de descripteurs de formes peuvent être extraites : La première catégorie est basée sur la géométrie des régions de l'image, la deuxième est basée sur les statistiques des intensités de pixels des différentes régions dans l'image. Une image peut subir des transformations géométriques comme la rotation, la translation et le changement d'échelle. Pour assurer une description assez robuste et efficace, les descripteurs de formes couvrent, généralement, toutes les échelles de représentation que comporte un objet. De plus, ils sont souvent insensibles aux différentes variations causées par des transformations géométriques.

Les descripteurs de formes sont généralement basés sur les caractéristiques de régions et de contours de l'image qu'ils présentent une description très riche d'un objet. De nombreuses méthodes ont été proposées pour représenter une forme. A cause de l'invariance aux trois transformations géométriques, la translation, la rotation et le changement

d'échelle, nous utilisons dans ce projet de recherche la théorie des moments invariants.

## 2.2 Théorie des moments

Les moments sont des mesures quantitatives utilisées pour caractériser une fonction et pour capturer ses caractéristiques importantes. L'idée fondamentale de la théorie des moments est la projection de chaque élément d'un ensemble de données sur une base de fonctions pour extraire les caractéristiques utiles, ces dernières sont extraites sous forme des scalaires appelés moments. Les moments sont largement utilisés dans le domaine du traitement d'images et de la reconnaissance de formes à savoir : la classification des bases de données d'images, l'indexation et recherche d'images, la détection de contour, l'analyse d'images, la reconnaissance de l'action, la classification des navires , la reconnaissance de caractères , la compression des données , l'enregistrement des images satellites, etc. Hu a été le premier à utiliser les moments géométriques dans la reconnaissance des formes [1] et à montrer le pouvoir de ces caractéristiques dans le cas de la reconnaissance des caractères majuscules imprimés. En général le moment géométrique d'ordre  $(p + q)$  de l'image  $f(x, y)$  en dimension 2D est défini par :

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) d\mu, \quad p, q = 0, 1, \dots \quad (2.1)$$

La notion de théorie des moments était introduite par M. K. Hu en 1962. A l'aide des moments géométriques présentés dans l'équation (2.1) ce dernier a construit set moments invariants aux trois transformations affines : la translation, la rotation et le changement d'échelle.

Les moments géométriques d'ordre faible représentent différentes caractéristiques spatiales de l'image. Le moment géométrique d'ordre 0,  $m_{00}$  s'appelle la masse de l'image. Les moments du premier ordre  $m_{10}$  et  $m_{01}$  sont respectivement le moment d'intensité sur l'axe des (ox) et l'axe des (oy).  $(\bar{x}, \bar{y})$  est le centre de gravité ou le barycentre de l'image  $f(x, y)$ , où

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2.2)$$

IL faut noter que les moments géométriques présentés dans l'équation (2.1) n'assurent pas la propriété de l'invariance aux trois transformations géométriques (translation, rotation et changement d'échelle ). Les moments invariants à la translation seront obtenus en

proposant les moments centraux qui sont définis comme suit :

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) d\mu, \quad p, q = 0, 1, \dots \quad (2.3)$$

Les moment centraux calculés en utilisant le centre de l'image  $(\bar{x}, \bar{y})$  sont équivalents aux moments géométriques  $m_{pq}$  dont le centre du repère cartésien est transformé vers le centre de l'image  $f(x, y)$ . Par conséquent les moments centraux sont invariants à la translation. L'invariance au changement d'échelle peut être obtenu par la normalisation des moments centraux comme suit :

$$\mu_{p,q} = \frac{M_{pq}}{M_{00}^{\frac{p+q+2}{2}}} \quad (2.4)$$

Ces moments sont appelés les moments centraux normalisés de l'image  $f(x, y)$ . A l'aide des moments centraux normalisés présentés dans l'équation (2.4), Hu a proposé les sept moments invariants aux trois transformations géométriques (la translation, la rotation et le changement d'échelle) définis par :

$$\varphi_1 = \mu_{0,2} + \mu_{2,0} \quad (2.5)$$

$$\varphi_2 = (\mu_{0,2} - \mu_{2,0})^2 + 4\mu_{2,0}^2 \quad (2.6)$$

$$\varphi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \quad (2.7)$$

$$\varphi_4 = (\mu_{3,0} - \mu_{1,2})^2 + (\mu_{2,1} - \mu_{0,3})^2 \quad (2.8)$$

$$\begin{aligned} \varphi_5 = & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ & (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (2.9)$$

$$\begin{aligned} \varphi_6 = & (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] + \\ & 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{2,1} + \mu_{0,3}) \end{aligned} \quad (2.10)$$

$$\begin{aligned} \varphi_7 = & (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & + (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (2.11)$$

A l'aide de ces moments invariants, on peut construire un vecteur descripteur à sept composantes de la forme suivante :

$$V(f) = \left( \varphi_1(f), \varphi_2(f), \dots, \varphi_7(f) \right) \quad (2.12)$$

Les six premiers moments sont invariants à la translation, la rotation et le changement d'échelle ainsi qu'aux réflexions. Le septième moment  $\varphi_7$  n'étant pas invariant aux réflexions. Les moments invariants de Hu sont d'ordre fini, par conséquent, ils ne comprennent pas un ensemble complet de caractéristiques de l'image. Pour cette raison, nous

introduisons [Hjouji et al. (2020)]. Dans ce travail nous introduisons un autre ensemble de moments invariants en fonction des moments centraux normalisés d'ordre infini présentés comme suit.

$$\Phi_n(f) = \sum_{k=0}^n \binom{n}{k} \mu_{2k, 2n-2k}, \quad n = 0, 1, \dots \quad (2.13)$$

où  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ .

Pour bien comprendre la nature des moments invariants proposés, nous présentons les huit premiers moments invariants

$$\Phi_1 = \mu_{0,2} + \mu_{2,0} \quad (2.14)$$

$$\Phi_2 = \mu_{0,4} + 2\mu_{2,2} + \mu_{4,0} \quad (2.15)$$

$$\Phi_3 = \mu_{0,6} + 3\mu_{2,4} + 3\mu_{4,2} + \mu_{6,0} \quad (2.16)$$

$$\Phi_4 = \mu_{0,8} + 4\mu_{2,6} + 6\mu_{4,4} + 4\mu_{6,2} + \mu_{8,0} \quad (2.17)$$

$$\Phi_5 = \mu_{0,10} + 5\mu_{2,8} + 10\mu_{4,6} + 10\mu_{6,4} + 5\mu_{8,2} + \mu_{10,0} \quad (2.18)$$

$$\Phi_6 = \mu_{0,12} + 6\mu_{2,10} + 15\mu_{4,8} + 20\mu_{6,6} + 15\mu_{8,4} + 6\mu_{10,2} + \mu_{12,0} \quad (2.19)$$

$$\Phi_7 = \mu_{0,14} + 7\mu_{2,12} + 21\mu_{4,10} + 35\mu_{6,8} + 35\mu_{8,6} + 21\mu_{10,4} + 7\mu_{12,2} + \mu_{14,0} \quad (2.20)$$

$$\Phi_8 = \mu_{0,16} + 8\mu_{2,14} + 28\mu_{4,12} + 56\mu_{6,10} + 70\mu_{8,8} + 56\mu_{10,6} + 28\mu_{12,4} + 8\mu_{14,2} + \mu_{16,0} \quad (2.21)$$

Ces moments permettent d'extraire des caractéristiques globales d'une image. Nous avons testé ces moments invariants dans les domaines de reconnaissance de formes, de classification d'images et de récupération d'information. Ils ont donné des bons résultats. Cependant l'inconvénient de ces moments et les sept moments de Hu est que la base de projection n'est pas orthogonale, impliquant ainsi une redondance d'information et une sensibilité au bruit [57, 58]. Pour remédier à ce problème, Teague [7] a proposé la théorie des moments orthogonaux. Ces moments sont calculés à l'aide de la projection de l'image sur les éléments de la base des polynômes orthogonaux. Dans cette approche, Teague a utilisé aussi les moments pour la reconstruction des images en utilisant la propriété de non redondance qui est due à l'orthogonalité de la base polynômiale utilisée. Ensuite, Teh et al. proposent dans [126] une étude détaillée sur les moments géométrique, Legendre, Zernike, Pseudo-Zernike, etc. Ils ont présenté une étude comparative en terme de qualité de représentation, redondance de l'information et sensibilité au bruit. Ils ont exposé des résultats analytiques et expérimentaux. Les résultats obtenus ont assuré que les moments orthogonaux sont bien meilleurs que les autres moments en terme de qualité de reconstruction et de redondance d'information. Le problème qui se pose lors de l'utilisation des

moments orthogonaux pour extraire les vecteurs descripteurs des images est la difficulté d'assurer la propriété de l'invariance par rapport aux trois transformations géométriques : translation, rotation et le changement d'échelle. Pour surmonter ce problème, nous avons présenté dans [Hjouji et al. 2020] une nouvelle technique de construction de polynômes orthogonaux non séparables à deux variables basé sur une relation de récurrence et un nouvel ensemble de moments invariants orthogonaux (OIM). Ces moments orthogonaux ont été appliqués avec succès dans le domaine de la reconnaissance de formes.

## 2.3 Moments orthogonaux

Les moments orthogonaux constituent des quantités statistiques obtenues par la projection d'une image sur les éléments d'une base polynômiale orthogonale  $\{P_n(x)P_m(y), n, m = 0, 1, \dots\}$  (au lieu d'une base quelconque  $\{x^n y^m, n, m = 0, 1, \dots\}$  utilisées pour les moments géométriques). Par conséquent, les moments orthogonaux d'ordre  $(n + m)$  d'une image  $f(x, y)$  sont définis par :

$$MO_{nm} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P_n(x)P_m(y)f(x, y)d\mu, \quad n, m = 0, 1, \dots \quad (2.22)$$

Avec  $\{P_n, n = 0, 1, \dots\}$  vérifient la condition d'orthogonalité

$$\int_{-\infty}^{+\infty} P_n(x)P_m(x)d\mu = \alpha\delta_{nm}, \quad (2.23)$$

où  $\alpha$  est un réel strictement positif et  $\delta_{nm}$  est le symbole de Kronecker, qui est égale 1 si  $n = m$ , et 0 sinon.

A cause de leur orthogonalité, ils offrent la possibilité d'extraction des caractéristiques d'une image dont le but de pouvoir les utiliser au domaine de reconnaissance de formes, compression de données et la reconstruction des images. A l'aide de l'équation (2.22), il est claire que les moments orthogonaux sont liées directement à la base orthogonale des polynômes considérée. Dans les sections qui suivent nous allons introduire un ensemble des moments orthogonaux que nous utilisons dans ce projet : Legendre, Jacobi, Tchebyshev, Krawtchouk, Hahn et Dual Hahn et nous allons introduire notre méthode de construction des polynomes non-séparables à deux variables  $P_n(x, y), n = 0, 1, \dots$  et les moments orthogonaux proposés dans [Hjouji et al. 2020].

### 2.3.1 Moments orthogonaux de Legendre

Les moments de Legendre basés sur les polynômes de même nom peuvent être efficacement utilisés en tant que caractéristiques dans l'analyse des images bidimensionnelles, ils sont introduits par Legendre (1785), donc avant l'introduction des polynômes de Jacobi. Le polynôme de Legendre de degré  $n$  est défini par :

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n], \quad -1 \leq x \leq 1. \quad (2.24)$$

A l'aide de cette formule de Rodriguez, nous pouvons déterminer les dix premiers polynômes de Legendre comme suit :

$$L_0(x) = 1 \quad (2.25)$$

$$L_1(x) = x \quad (2.26)$$

$$L_2(x) = \frac{1}{2}(3x^2 - 1) \quad (2.27)$$

$$L_3(x) = \frac{1}{2}(5x^3 - 3x) \quad (2.28)$$

$$L_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3) \quad (2.29)$$

$$L_5(x) = \frac{1}{8}(63x^5 - 70x^3 + 15x) \quad (2.30)$$

$$L_6(x) = \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5) \quad (2.31)$$

$$L_7(x) = \frac{1}{16}(429x^7 - 639x^5 + 315x^3 - 35x) \quad (2.32)$$

$$L_8(x) = \frac{1}{128}(6435x^8 - 12012x^6 + 6930x^4 - 1260x^2 + 35) \quad (2.33)$$

$$L_9(x) = \frac{1}{128}(12155x^9 - 25740x^7 + 18018x^5 - 4620x^3 + 315x) \quad (2.34)$$

$$L_{10}(x) = \frac{1}{256}(46189x^{10} - 109395x^8 + 90090x^6 - 30030x^4 + 3465x^2 - 63) \quad (2.35)$$

La figure (2.1) montre les graphes de six premiers polynomes de Legendre. Les polynômes de Legendre sont orthogonaux sur  $[-1, 1]$  pour le produit scalaire continue, en plus, nous avons la relation d'orthogonalité suivante :

$$\int_{-1}^1 L_n(x)L_m(x)dx = \frac{2}{2n+1}\delta_{n,m}. \quad (2.36)$$

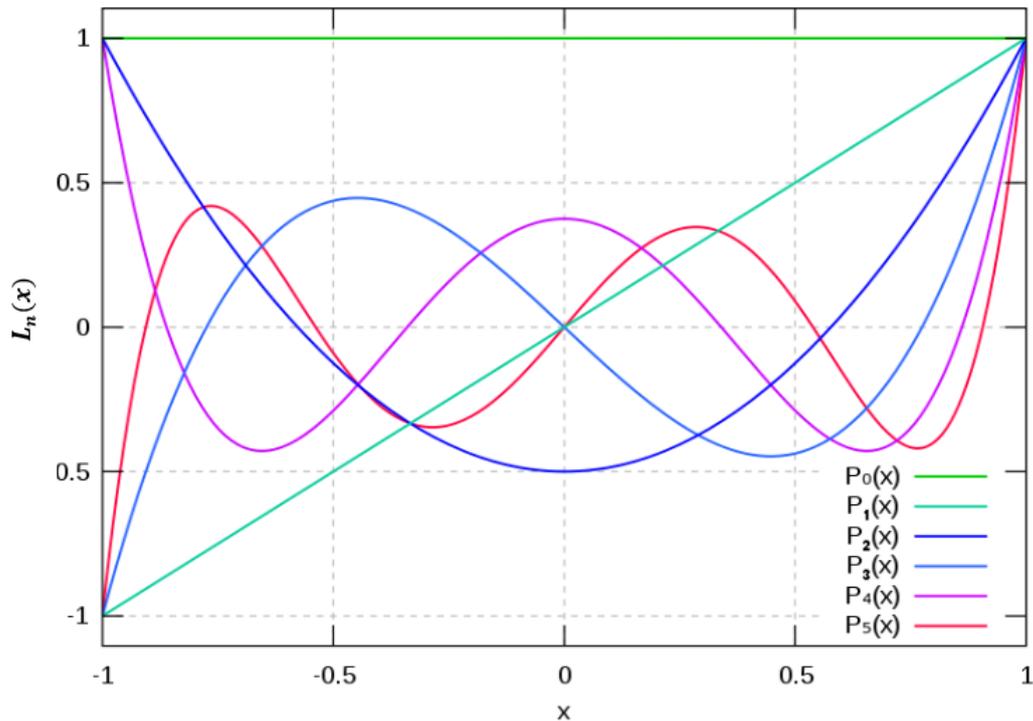


FIGURE 2.1 – Les courbes de six premiers polynômes de Legendre

Comme tout ensemble de polynômes orthogonaux, les polynômes de Legendre vérifient la relation de récurrence suivante :

$$L_{n+1}(x) = \frac{2n+1}{n+1}xL_n(x) - \frac{n}{n+1}L_{n-1}(x), \quad -1 \leq x \leq 1. \quad (2.37)$$

Par ailleurs, les polynômes  $L_n$  vérifient l'équation différentielle de Legendre :

$$\frac{d}{dx} \left[ (1-x^2) \frac{d(L_n(x))}{dx} \right] + n(n+1)L_n(x) = 0, \quad -1 \leq x \leq 1. \quad (2.38)$$

A l'aide de la formule de Leibnitz appliqué sur (2.24), la forme explicite de ces polynômes peut être écrire

$$L_n(x) = \frac{1}{2^n} \sum_{k=0}^{E(n/2)} (-1)^k \binom{n}{k} \binom{2n-2k}{n} x^{n-2k} \quad (2.39)$$

Les moments continus orthogonaux de Legendre d'ordre  $(n+m)$  d'une image  $f(x, y)$  sont donnés par la relation suivante :

$$ML_{nm} = \frac{(2n+1)(2m+1)}{4} \int_{-1}^1 \int_{-1}^1 L_n(x)L_m(y)f(x, y)dx dy, \quad n, m = 0, 1, 2, \dots \quad (2.40)$$

Dans le pratique, la double intégration dans (2.40) doit être remplacée par doubles sommations. En supposant qu'une image numérique  $f(x, y)$  de taille  $N \times M$ , le moment de Legendre s'écrit sous la forme :

$$ML_{nm} = \frac{(2n+1)(2m+1)}{4} \sum_{i=1}^N \sum_{j=1}^M L_n(x_i) L_m(y_j) f(i, j) \quad n, m = 0, 1, 2, \dots \quad (2.41)$$

avec  $x_i = -1 + \frac{2i}{N}$  et  $y_j = -1 + \frac{2j}{M}$ . Alors nous pouvons approximer l'image reconstruite, à un ordre  $max$ , par la relation suivante :

$$\hat{f}(x, y) = \sum_{m=0}^{max} \sum_{n=0}^m ML_{m-n, n} L_{m-n}(x) L_n(y). \quad (2.42)$$

Notez que si l'image reconstruite s'approche de l'image originale lorsque  $max$  assez grand, on dit que les moments orthogonaux sont de bons caractéristiques de l'image.

### 2.3.2 Moments orthogonaux de Legendre adaptés

Pour utiliser les moments de Legendre dans le domaine de la classification d'images, nous devons construire des vecteurs descripteurs invariants pour les trois types de transformations : la translation, la rotation et le changement d'échelle. Il y a plusieurs méthodes qui ont été utilisées dans la littérature, à savoir les moments présentés par Xiao et al [20] qui sont basés sur les polynômes radiaux de Legendre substitués (PRLS) et pondérés (PRLP), les moments invariants rotationnels de Legendre (MIRLs) présentés par Hosny et al [16] et les moments invariants de Legendre proposées par Raveendran et al. [75] pour la reconnaissance de formes. Dans ce travail, nous présentons un nouvel ensemble de moments invariants de Legendre en suivant la même stratégie que celle utilisée par Papakostas et al. [127] pour les moments de Krawtchouk, appelés les moments invariants de Legendre adaptés (MILAs), qui sont basés sur le nouveau ensemble de polynômes de Legendre adaptés orthogonaux.

Pour  $N \geq 2$ , le polynôme de Legendre adapté  $A_n$  de taille  $N$  et de degré  $n$  est obtenu directement du polynôme de Legendre  $L_n$  en posant

$$A_n(t) = L_n \left( \frac{2}{N}t - 1 \right), \quad 0 \leq t \leq N. \quad (2.43)$$

Par conséquent, nous pouvons établir facilement que l'ensemble des polynômes de Legendre adaptés  $A_n(t)$  vérifient les quatre formulations :

*Formulation différentielle* : Les polynômes de Legendre adaptés sont les solutions de

l'équation différentielle suivante

$$\frac{d}{dt} \left[ (Nt - t^2) \frac{d(A_n(t))}{dt} \right] + n(n+1)A_n(t) = 0, \quad 0 \leq t \leq N. \quad (2.44)$$

**Preuve.** En utilisant le changement de variable  $x = \frac{2}{N}t - 1$ , nous avons

$$\begin{aligned} (1-x^2) \frac{d(P_n(x))}{dx} &= \frac{4}{N^2} [Nt - t^2] \frac{d(P_n(\frac{2}{N}t - 1))}{dt} \times \frac{dt}{dx} \\ &= \frac{4}{N^2} [Nt - t^2] \frac{d(A_n(t))}{dt} \times \frac{N}{2} = \frac{2}{N} [Nt - t^2] \frac{d(A_n(t))}{dt} \end{aligned} \quad (2.45)$$

Par conséquent,

$$\begin{aligned} \frac{d}{dx} \left[ (1-x^2) \frac{d(P_n(x))}{dx} \right] &= \frac{d}{dt} \left[ (1-x^2) \frac{d(P_n(x))}{dx} \right] \times \frac{dt}{dx} \\ &= \frac{d}{dt} \left[ (1-x^2) \frac{d(P_n(x))}{dx} \right] \times \frac{N}{2} = \frac{d}{dt} \left[ (Nt - t^2) \frac{d(A_n(t))}{dt} \right] \end{aligned} \quad (2.46)$$

alors, les formules (2.45), (2.46) et (2.38) donnent

$$\begin{aligned} 0 &= \frac{d}{dx} \left[ (1-x^2) \frac{d(P_n(x))}{dx} \right] + n(n+1)P_n(x) \\ &= \frac{d}{dt} \left[ (Nt - t^2) \frac{d(A_n(t))}{dt} \right] + n(n+1)A_n(t). \end{aligned} \quad (2.47)$$

*Formule de récurrence :* Les polynômes de Legendre adaptés vérifient la formule de récurrence suivante :

$$A_{n+1}(t) = \frac{2n+1}{n+1} \left( \frac{2}{N}t - 1 \right) A_n(t) - \frac{n}{n+1} A_{n-1}(t), \quad 0 \leq t \leq N. \quad (2.48)$$

*Formule de Rodriguez*

Les polynômes de Legendre adaptés peuvent être écrits sous la forme suivante

$$A_n(t) = \frac{1}{n!N^n} \frac{d^n}{dt^n} \left[ (t^2 - Nt)^n \right] \quad (2.49)$$

**Preuve.** D'après la formule (2.43), on peut en déduire qu'il existe une constante  $\alpha_n$  telle que

$$A_n(t) = \alpha_n \frac{d^n}{dt^n} \left[ (t^2 - Nt)^n \right] = \alpha_n \frac{d^n}{dt^n} \left[ (t - N)^n \times t^n \right] \quad (2.50)$$

Nous appliquons la règle générale de Leibniz, nous obtenons

$$A_n(t) = \alpha_n \sum_{i=0}^n \binom{n}{k} \left[ (t - N)^n \right]^{(k)} \left[ t^n \right]^{(n-k)} \quad (2.51)$$

As  $\left[ [t - N]^n \right]^{(k)} = \frac{n!}{(n-k)!} [t - N]^{n-k}$  et  $(t^n)^{(n-k)} = \frac{n!}{k!} t^k$  donc

$$\begin{aligned} A_n(t) &= \alpha_n \sum_{i=0}^n \binom{n}{k} \frac{(n!)^2}{(n-k)!k!} t^k [t - N]^{n-k} \\ &= \alpha_n \left[ \sum_{i=0}^{n-1} \binom{n}{k} \frac{(n!)^2}{(n-k)!k!} t^k [t - N]^{n-k} + n! t^n \right]. \end{aligned} \quad (2.52)$$

A l'aide de cette formule, on peut voir que  $A_n(N) = \alpha_n n! N^n$ . On sait que l'ensemble des polynômes de Legendre  $\{P_n, n \geq 0\}$  vérifie la condition  $P_n(1) = 1$ . Alors la formule (2.43) donne  $A_n(N) = P_n(1) = 1$ . Ce qui implique  $\alpha_n = \frac{1}{n!N^n}$  et

$$A_n(t) = \frac{1}{n!N^n} \frac{d^n}{dt^n} \left[ (t^2 - Nt)^n \right] \quad (2.53)$$

*Formule explicite*

**Lemma 3.** L'ensemble des polynômes de Legendre adaptés peuvent être écrits sous la forme explicite suivante :

$$A_n(t) = \frac{1}{N^n} \sum_{i=0}^n (-N)^{n-i} \binom{n+i}{2i} \binom{2i}{i} t^i \quad (2.54)$$

**Preuve.** Nous pouvons utiliser l'équation (2.49) et la formule de Newton, nous obtenons

$$A_n(t) = \frac{1}{n!N^n} \frac{d^n}{dt^n} \left[ (t^2 - Nt)^n \right] = \frac{1}{n!N^n} \frac{d^n}{dt^n} \left[ \sum_{i=0}^n \binom{n}{i} (-N)^{n-i} t^{n+i} \right] \quad (2.55)$$

Comme  $\frac{d^n}{dt^n} (t^{n+i}) = \frac{(n+i)!}{i!} t^i$ , alors

$$A_n(t) = \frac{1}{N^n} \sum_{i=0}^n (-N)^{n-i} \binom{n+i}{2i} \binom{2i}{i} t^i. \quad (2.56)$$

Les quatre premiers polynômes de Legendre adaptés de taille  $N$  peuvent être exprimés

comme suit :

$$A_0(t) = 1 \quad (2.57)$$

$$A_1(t) = \frac{2}{N}t - 1 \quad (2.58)$$

$$A_2(t) = \frac{6}{N^2}t^2 - \frac{6}{N}t + 1 \quad (2.59)$$

$$A_3(t) = \frac{20}{N^3}t^3 - \frac{30}{N^2}t^2 + \frac{12}{N}t - 1 \quad (2.60)$$

L'ensemble des polynômes de Legendre adaptés vérifie la condition d'orthogonalité définie sur l'intervalle  $[0, N]$  par :

$$\int_0^N A_n(x)A_m(t)dt = \frac{N}{2n+1}\delta_{n,m}. \quad (2.61)$$

Où  $\delta_{n,m}$  est le symbole de Kronecker.

**Preuve.** Nous utilisons les relation (2.36) et (2.43) et le changement de variable  $x = \frac{2}{N}t - 1$  nous obtenos,

$$\begin{aligned} \int_0^N A_n(t)A_m(t)dt &= \int_0^N P_n\left(\frac{2}{N}t - 1\right)P_m\left(\frac{2}{N}t - 1\right)dt \\ &= \frac{N}{2} \int_{-1}^1 P_n(x)P_m(x)dx \\ &= \frac{N}{2} \times \frac{2}{2n+1}\delta_{n,m} = \frac{N}{2n+1}\delta_{n,m} \end{aligned} \quad (2.62)$$

Dans la suite, le polynôme de Legendre adapté de taille  $N$  et de degré  $n$  noté par  $A_n^{(N)}$  et le polynôme de Legendre adapté normalisé noté par  $\bar{A}_n^{(N)}$  et définie par

$$\bar{A}_n^{(N)} = \sqrt{\frac{2n+1}{N}}A_n^{(N)}. \quad (2.63)$$

De sorte que les polynômes de Legendre adaptés  $\bar{A}_n^{(N)}$  sont orthonormales, i.e.,

$$\int_0^1 \bar{A}_n^{(N)}(x)\bar{A}_m^{(M)}(x)dx = \delta_{n,m}. \quad (2.64)$$

Si  $f(x, y)$  est une image de taille  $N \times M$ ,  $\bar{A}_n^{(N)}$  et  $\bar{A}_m^{(M)}$  sont respectivement les polynômes de Legendre adaptés normalisés de taille  $N$  et  $M$ , alors  $f(x, y)$  et  $\bar{A}_n^{(N)}(x) \times \bar{A}_m^{(M)}(y)$  possèdent le même domaine de définition  $[0, N] \times [0, M]$ . Celui montre l'avantage de ces polynômes.

Le moment orthogonal de Legendre adapté continu (MOLAC) d'ordre  $(n + m)$  d'une

image  $f(x, y)$  de taille  $N \times M$  est définie par :

$$MLA_{nm} = \int_0^N \int_0^M \bar{A}_n^{(N)}(x) \bar{A}_m^{(M)}(y) f(x, y) dx dy, \quad n, m = 0, 1, 2, \dots \quad (2.65)$$

Le moment orthogonal de Legendre adapté discret (MOLAD) d'ordre  $(n+m)$  d'une image  $f(x, y)$  de taille  $N \times M$  est définie par :

$$MLA_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \bar{A}_n^{(N)}(x) \bar{A}_m^{(M)}(y) f(x, y). \quad n, m = 0, 1, 2, \dots \quad (2.66)$$

Les moments géométriques discrets d'une image  $f(x, y)$  de taille  $N \times M$  sont définis par

$$MG_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^n y^m f(x, y). \quad n, m = 0, 1, 2, \dots \quad (2.67)$$

A partir de ces moments Hu a construit un ensemble des moments invariants aux trois transformations affines, la translation, la rotation et le changement d'échelle, qui sont définis sous la forme suivante

$$GMI_{nm} = m_{00}^{-\gamma} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [(x - \bar{x}) \cos \theta + (y - \bar{y}) \sin \theta]^n \times [(x - \bar{x}) \cos \theta - (y - \bar{y}) \sin \theta]^m f(x, y) \quad (2.68)$$

Où

$$\gamma = \frac{n+m}{2} + 1, \quad \bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2.69)$$

$$\theta = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (2.70)$$

et  $\mu_{nm}$  est le moment central d'ordre  $(n+m)$  définie par :

$$\mu_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (x - \bar{x})^n ((y - \bar{y})^m f(x, y), \quad n, m = 0, 1, 2, \dots \quad (2.71)$$

Une technique courante pour construire des invariants de moment orthogonaux de Legendre adaptés en fonction des invariants de moment géométriques (GMI) comme nous allons détailler dans la suite de cette section :

Le moment de Legendre adaptés d'une image  $f(x, y)$  peut être écrit en fonction des mo-

ments invariants géométriques

$$MLA_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \bar{A}_n^{(N)}(x) \bar{A}_m^{(M)}(y) f(x, y) \quad (2.72)$$

les relations (2.54) et (2.63) donnent,

$$\bar{A}_n^{(N)}(t) = \sqrt{\frac{2n+1}{N}} \times \frac{1}{N^n} \sum_{i=0}^n (-N)^{n-i} \binom{n+i}{2i} \binom{2i}{i} t^i. \quad (2.73)$$

$$\bar{A}_m^{(M)}(t) = \sqrt{\frac{2m+1}{M}} \times \frac{1}{M^m} \sum_{j=0}^m (-M)^{m-j} \binom{m+j}{2j} \binom{2j}{j} t^j. \quad (2.74)$$

les relations (2.72),(2.73) et (2.74), donnent

$$MLA_{nm} = \sqrt{\frac{(2n+1)(2m+1)}{N^{2n+1}M^{2m+1}}} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{i=0}^n \sum_{j=0}^m (-N)^{n-i} (-M)^{m-j} \times \binom{n+i}{2i} \binom{2i}{i} \binom{n+j}{2j} \binom{2j}{j} x^i y^j f(x, y) \quad (2.75)$$

en utilisant (2.66), nous obtenons

$$MLA_{nm} = \sqrt{\frac{(2n+1)(2m+1)}{N^{2n+1}M^{2m+1}}} \sum_{i=0}^n \sum_{j=0}^m (-N)^{n-i} (-M)^{m-j} \binom{n+i}{2i} \times \binom{2i}{i} \binom{n+j}{2j} \binom{2j}{j} GMI_{ij} \quad (2.76)$$

Les moments invariants de Legendre adaptés peuvent être étendu en termes de GMI présenté dans l'équation (2.68) comme suit :

$$MLAI_{nm} = \sum_{i=0}^n \sum_{j=0}^m (-N)^{n-i} (-M)^{m-j} \binom{n+i}{2i} \binom{2i}{i} \binom{n+j}{2j} \binom{2j}{j} V_{ij} \quad (2.77)$$

où  $V_{ij}$  est le paramètre définie par :

$$V_{nm} = \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} \binom{m}{j} \left( \frac{N \times M}{2} \right)^{\frac{(i+j)}{2}+1} \left( \frac{N}{2} \right)^{n-i} \left( \frac{M}{2} \right)^{m-j} GMI_{ij} \quad (2.78)$$

En utilisant la formule binomial, le moment invariant géométrique  $GMI$  présenté dans

l'équation (2.68) peut être calculé comme suit

$$\begin{aligned}
 GMI_{nm} &= \\
 GM_{00}^{-\gamma} \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} \binom{m}{j} (\cos\theta)^{i+j} (\sin\theta)^{n+m-i-j} \times (-1)^{m-j} \mu_{m+i-j, n+j-i} \\
 &= \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} \binom{m}{j} (\cos\theta)^{i+j} (\sin\theta)^{n+m-i-j} \times (-1)^{m-j} \eta_{m+i-j, n+j-i} \quad (2.79)
 \end{aligned}$$

Où

$$\eta_{nm} = \frac{\mu_{nm}}{GM_{00}^{\gamma}} \quad (2.80)$$

Par conséquent, les moments de Legendre adaptés invariants aux trois transformations : rotation, translation et changement d'échelle d'une image peuvent être obtenus à l'aide des formules (2.77), (2.78) et (2.79).

Maintenant, nous allons présenter des tests numériques pour valider le cadre théorique présenté précédemment. Dans ce contexte, nous présenterons deux expériences. Dans la première, nous testons l'invariance des moments invariants proposés (MLAIs) pour les trois transformations, translation, rotation et changement d'échelle. Dans la seconde partie, nous testons la capacité des moments de Legendre adaptés orthogonaux (MLAO) pour la reconstruction des images. Nous utilisons l'image Cat (Fig.2.2) pour tester l'invariance de (MLAIs) par rapport à la translation, la rotation et le changement d'échelle.



FIGURE 2.2 – image de Cat.

L'image "Cat" est transformée par des translations de vecteurs  $\{\vec{u}_k\}_{k=0,\dots,20}$ , des changements d'échelles de facteurs  $\{\lambda_k\}_{k=0,\dots,20}$  et des rotations d'angles  $\{\theta_k\}_{k=0,\dots,36}$  tels que

$$\lambda_k = 0.5 + (0.05)k, k = 0, \dots, 30, \quad (2.81)$$

$$\theta_k = 10k, k = 0, \dots, 36 \quad (2.82)$$

$$\vec{u}_k = (-5 + 0.5k, -5 + 0.5k), k = 0, \dots, 40. \quad (2.83)$$

Pour mesurer le degré de l'invariance des moments invariants de Legendre adaptés (MO-

LAI) sous différentes transformations d'image, nous utilisons l'erreur relative entre les deux ensembles de moments invariants correspondant à l'image originale  $f(x, y)$  et l'image transformée  $f_k(x, y)$  comme suit :

$$E(f, f_k) = \frac{\|MLAI(f) - MLAI(f_k)\|}{\|MLAI(f)\|}, k = 0, \dots, p \quad (2.84)$$

où  $p$  est le nombre des images transformées,  $\|\cdot\|$  la norme Euclidienne  $MLAI(f)$ ,  $MLAI(f_k)$  sont respectivement les moments invariants de Legendre adaptés de l'image originale et l'image transformée. Nous présentons également une étude comparative de cette erreur de nos moments MLAI avec d'autres moments invariants bien connus comme les moments invariants de Hu, les moments invariants de Legendre classique MLCs et les moments invariants de Tchebichef (MTs).

La figure (2.3) montre les graphes de l'erreur relative pour la rotation  $E(f, f_k)$  des moments invariants de Legendre adaptés, les moments invariants de Hu, les moments invariants de Legendre classique et les moments de Tchebychev. Selon les résultats présentés dans cette figure, nous pouvons dire que les moments invariants (MLAI) ont de meilleures performances que les autres moments invariants testés quel que soit l'angle de rotation. La figure (2.4) montre les graphes de l'erreur relative pour le changement d'échelle  $E(f, f_k)$  des moments invariants de Legendre adaptés, moments invariants de Hu, moments invariants de Legendre classique MLCs et les moments de Tchebychev MTs. Selon les résultats présentés dans cette figure, nous pouvons également voir que l'erreur relative pour les moments proposés est inférieure à l'erreur pour les autres moments invariants, quel que soit le facteur de l'échelle. La figure (2.5) présente les courbes de l'erreur pour les images translées. Les résultats de cette figure montrent que l'erreur pour les moments (MLAI) est inférieure à l'erreur pour les moments invariants testés, quel que soit le vecteur de translation. D'après les résultats présentés dans ces figures, nous déduisons que les moments invariants de Legendre adaptés sont très stables pour les trois transformations d'image. Par conséquent, les moments invariants orthogonaux proposés pourraient être des outils importants pour les tâches de reconnaissance de formes qui nécessitent la propriété d'invariabilité sous la translation, la rotation et le changement d'échelle de l'image. Le tableau 1 présente quelques valeurs des moments invariants de Legendre adaptés à MLAI pour les 10 images transformées et bruitées de l'image de Lena (Fig.2.6) calculées à l'aide de la relation (2.67). Selon les résultats présentés dans le tableau 1, nous pouvons voir que les moments du même ordre sont presque égaux, ce qui montre l'invariabilité des moments proposés pour les trois transformations, translation, rotation et changement d'échelle. Maintenant, nous présentons des tests pour évaluer la capacité des moments orthogonaux

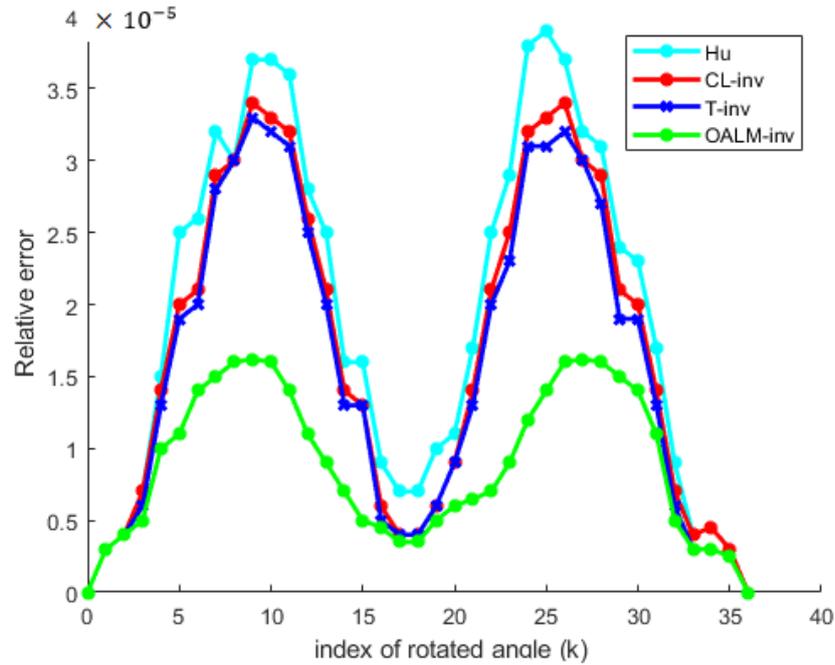


FIGURE 2.3 – L’erreur  $E(f, f_k)$  pour la rotation en utilisant les moments invariants de Legendre adaptés OALM-inv, les moments invariants de Legendre classique CL-inv et les moments invariants de Tchebichef T-inv .

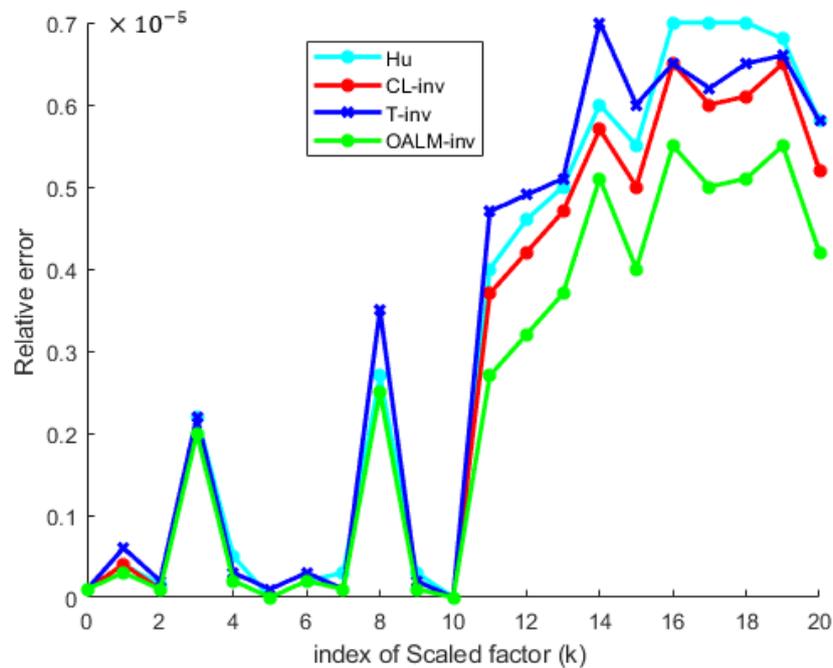


FIGURE 2.4 – L’erreur  $E(f, f_k)$  pour le changement d’échelle en utilisant les moments invariants de Legendre adaptés OALM-inv, les moments invariants de Legendre classique CL-inv et les moments invariants de Tchebichef T-inv .

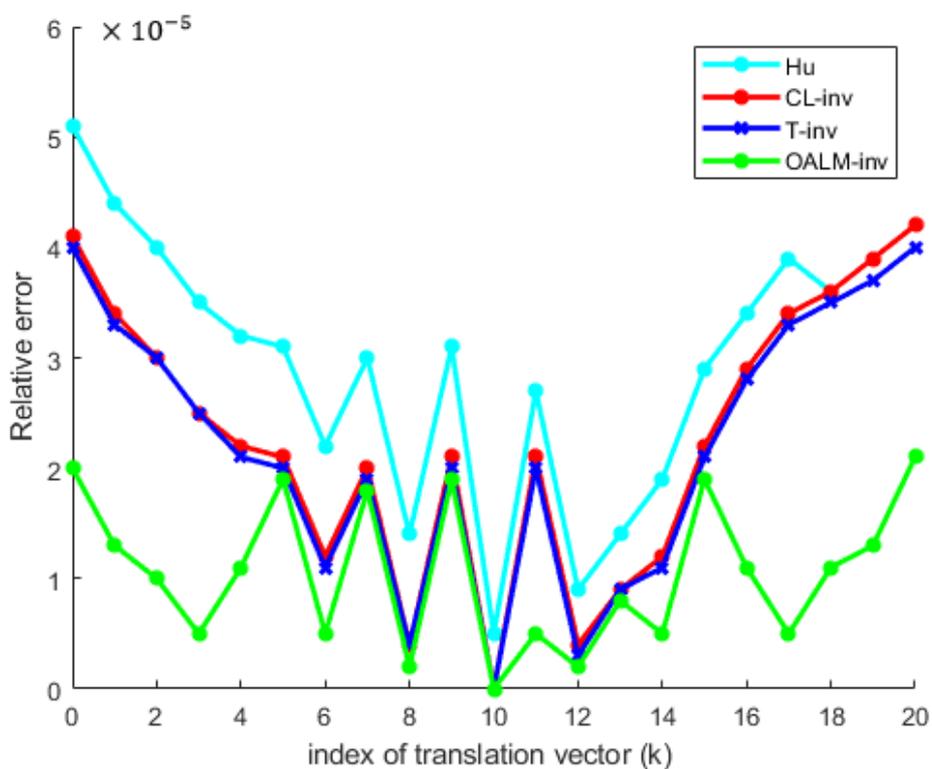


FIGURE 2.5 – L’erreur  $E(f, f_k)$  pour la translation en utilisant les moments invariants de Legendre adaptés OALM-inv, les moments invariants de Hu, les moments invariants de Legendre classique CL-inv et les moments invariants de Tchebychev T-inv .

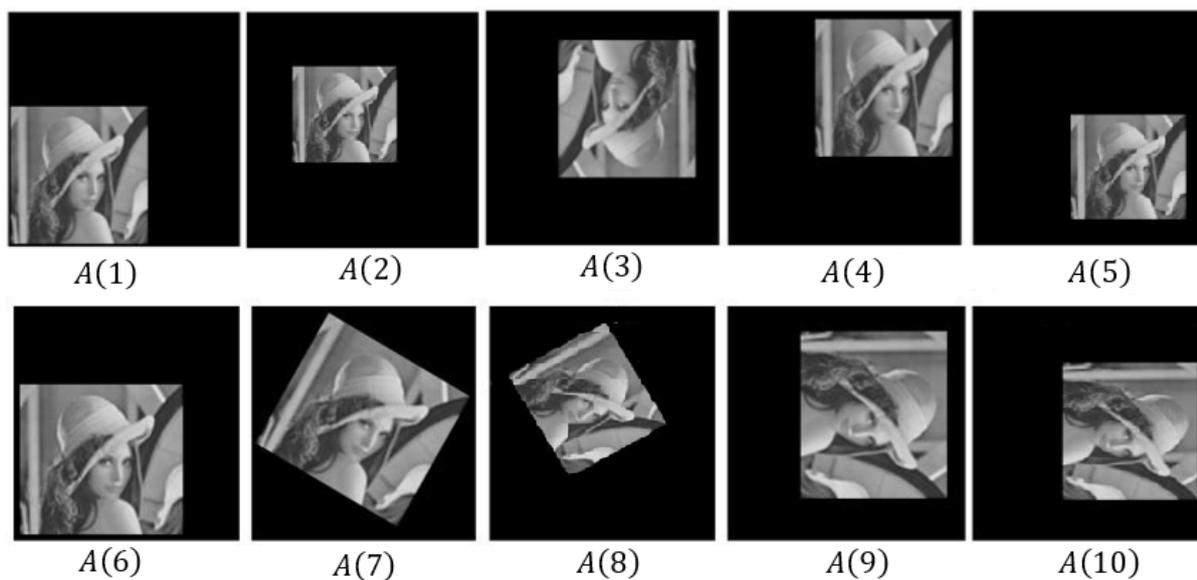


FIGURE 2.6 – 10 images transformées  $A(i), i = 1, \dots, 10$  de Lena.

### 2.3. MOMENTS ORTHOGONAUX

TABLE 2.1 – Valeurs des moments invariants de Legendre adaptés pour les 10 images transformées de l'image "Lena".

	image A(1)	image A(2)	image A(3)	image A(4)	image A(5)
$in\xi_{00}$	4081.62	4081.63	4082.62	4085.13	4084.83
$in\xi_{01}$	312.51	312.48	312.42	311.39	312.02
$in\xi_{02}$	-7421.28	-7422.31	-7421.18	-7423.14	-7423.77
$in\xi_{03}$	1381.27	1382.68	1381.13	1378.87	1380.19
$in\xi_{12}$	527.19	527.23	525.01	530.05	527.15
$in\xi_{13}$	-2554.25	-2555.11	-2554.78	-2550.72	-2550.67
	image A(6)	image A(7)	image A(8)	image A(9)	image A(10)
$in\xi_{00}$	4080.68	4082.01	4079.93	4084.10	4084.83
$in\xi_{01}$	313.01	311.	312.42	311.39	311.19
$in\xi_{02}$	-7422.00	-7423.05	-7422.16	-7423.60	-7423.82
$in\xi_{03}$	1380.90	1381.95	1382.01	1379.15	1381.20
$in\xi_{12}$	527.10	528.11	526.17	531.12	527.14
$in\xi_{13}$	-2555.05	-2555.40	-2553.95	-2551.07	-2551.55



FIGURE 2.7 – image fille (à gauche) et image Barbara (à droite)

de Legendre adaptés pour la reconstruction d'images en utilisant la formule

$$\hat{f}(x, y) = \sum_{m=0}^{\max} \sum_{n=0}^m MLA_{m-n, n} \bar{A}_{m-n}^{(N)}(x) \bar{A}_n^{(M)}(y). \quad (2.85)$$

Pour mesurer la capacité de la reconstruction d'image, nous calculons l'erreur quadratique moyenne (MSE) entre l'image originale  $f(x, y)$  de taille  $N \times M$  et l'image reconstruite  $\hat{f}(x, y)$  présenté dans l'équation (2.85).  $MLA_{ij}$  sont les moments orthogonaux de Legendre adaptés calculés par la formule (2.66). Nous réalisons des tests expérimentaux sur les deux images "fille" et "Barbara" présentées dans la figure (2.7). Toutes les images sont de taille  $200 \times 200$ . Sachant que l'ordre des moments maximum est compris entre 0 et 200. Les figures (2.8) et (2.9) montrent respectivement, les images reconstruites de l'image "fille et de l'image "Barbara" et les valeurs correspondantes de MSE des moments orthogonaux proposés (OALM), des moments de Legendre classiques (CLM) et des moments Tchebychev (TM). Il est évident que le MSE diminue lorsque l'ordre du moment

Moment	Ordre max		
	50	100	150
Moment propose MLA	 $MSE = 0.35$	 $MSE = 0.017$	 $MSE = 0.004$
MLC	 $MSE = 0.53$	 $MSE = 0.030$	 $MSE = 0.014$
MT	 $MSE = 0.41$	 $MSE = 0.022$	 $MSE = 0.007$

FIGURE 2.8 – images reconstruites de l'image « fille » et l'erreur correspondante MSE en utilisant les moments de Legendre adaptés (MLA), les moments orthogonaux de Legendre classique (CLM) et les moments orthogonaux de Tchebichef (MT) pour l'ordre max 50, 100 et 150

augmente et le MSE se rapproche de zéro lorsque l'ordre du moments assez grand. Nous avons effectué les reconstructions d'images testées à partir de différents ordres maximums  $max = 50, 100, 150$  de moments Legendre adaptés. Les images reconstruites pour les deux images testées sont présentées dans les figures (2.8), (2.9). L'erreur MSE et les résultats visuels pourraient facilement mesurer le degré de similitude entre les images originales et les images reconstruites.

### 2.3.3 Nouvel ensemble des moments orthogonaux

### 2.3.4 Moments non orthogonaux basés sur les moments centraux normalisés

Les moments de Hu sont des descripteurs qui ont montré leur efficacité dans plusieurs domaines de la reconnaissance des formes. Cependant, ces moments sont d'ordre fini,

Méthode	Ordre max		
	50	100	150
Moment Proposé MLA	 $MSE = 0.32$	 $MSE = 0.012$	 $MSE = 0.003$
MLC	 $MSE = 0.57$	 $MSE = 0.071$	 $MSE = 0.014$
MT	 $MSE = 0.49$	 $MSE = 0.083$	 $MSE = 0.011$

FIGURE 2.9 – images reconstruites de l’image « Barbara » et l’erreur correspondante MSE en utilisant les moments de Legendre adaptés (MLA), les moments orthogonaux de Legendre classiques (CLM) et les moments orthogonaux de Tchebichef (MT) pour l’ordre max 50, 100 et 150

par conséquent nous ne pouvons pas extraire un ensemble suffisant de caractéristiques d’image. Pour cette raison, nous avons introduit un autre ensemble de moments invariants en fonction des moments centraux normalisés d’ordre infini de façon différente que les moments invariants de Hu, qui sont définis comme suit.

$$\Phi_n(f) = \sum_{k=0}^n \binom{n}{k} \mu_{2k, 2k-2n}, \quad n = 0, 1, \dots \quad (2.86)$$

**Théorème 1.** Pour tout  $n \geq 0$  le moment  $\Phi_n(f)$  est invariant à la translation, rotation et changement d’échelle .

**Preuve.**  $\Phi_n(f)$  est une combinaison linéaire des moments centraux normalisés  $\mu_{2k, 2k-2n}$ , qui sont aussi invariants à la translation et le changement d’échelle. Maintenant, nous allons démontrer l’invariabilité à la rotation. Si l’image  $f(x, y)$  est transformée par une rotation d’angle  $\theta$ , alors la matrice de rotation est définie par  $M_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$  dont

la matrice inverse  $M_{-\theta}$ , l'image transformée est définie par  $f^r(x, y) = f\left(M_{\theta}\begin{pmatrix} x \\ y \end{pmatrix}\right)$  et le moment après la rotation est le suivant

$$\Phi_n(f^r) = \varphi'_n(f(x\cos\theta - y\sin\theta, x\cos\theta + y\sin\theta)) \quad (2.87)$$

$$= \sum_{k=0}^n \binom{n}{k} \mu_{2k, 2k-2n}(f(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)) \quad (2.88)$$

$$= \frac{1}{M_{00}^{n+1}} \sum_{k=0}^n \binom{n}{k} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^{2k} (y - \bar{y})^{n-2k} f(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta) d\mu$$

En posant  ${}^t(x, y) = M_{\theta}^t(x, y)$ , nous avons  $d\mu = dx dy = |M_{-\theta}| dx' dy' = dx' dy'$  et  ${}^t(\bar{x}, \bar{y}) = M_{-\theta}^t(\bar{x}', \bar{y}')$ , le moment  $\Phi_n(f^r)$  peut être écrit comme suit

$$\begin{aligned} \Phi_n(f^r) &= \frac{1}{M_{00}^{n+1}} \sum_{k=0}^n \binom{n}{k} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [(x' - \bar{x}')\cos\theta + (y' - \bar{y}')\sin\theta]^{2k} \\ &\quad [(y' - \bar{y}')\cos\theta - (x' - \bar{x}')\sin\theta]^{2n-2k} f(x', y') d\mu \\ &= \frac{1}{M_{00}^{n+1}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sum_{k=0}^n \binom{n}{k} \alpha^k \beta^{n-k} f(x', y') d\mu \end{aligned} \quad (2.89)$$

où  $\alpha = [(x' - \bar{x}')\cos\theta + (y' - \bar{y}')\sin\theta]^2$  et  $\beta = [(y' - \bar{y}')\cos\theta - (x' - \bar{x}')\sin\theta]^2$ .

Nous utilisons la formule de Newton sur  $(\alpha + \beta)^n$ , nous obtenus

$$\Phi_n(f^r) = \frac{1}{M_{00}^{n+1}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (\alpha + \beta)^n f(x', y') d\mu. \quad (2.90)$$

Nous avons

$$\begin{aligned} \alpha + \beta &= [(x' - \bar{x}')\cos\theta + (y' - \bar{y}')\sin\theta]^2 + [(y' - \bar{y}')\cos\theta - (x' - \bar{x}')\sin\theta]^2 \\ &= (x' - \bar{x}')^2 + (y' - \bar{y}')^2 \end{aligned} \quad (2.91)$$

Par conséquent,

$$\begin{aligned} (\alpha + \beta)^n &= \sum_{k=0}^n \binom{n}{k} (x' - \bar{x}')^{2k} (y' - \bar{y}')^{2n-2k} \\ \Phi_n(f^r) &= \frac{1}{M_{00}^{n+1}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sum_{k=0}^n \binom{n}{k} (x' - \bar{x}')^{2k} (y' - \bar{y}')^{2n-2k} f(x', y') d\mu = \varphi'_n(f) \end{aligned} \quad (2.92)$$

En fin, le moment  $\Phi_n$  est invariant par rapport à la translation, la rotation et le changement d'échelle.

### 2.3.5 Moments orthogonaux basés sur les moments centraux normalisés

#### A. Polynomes orthogonaux non-séparables à deux variables

Soit  $d\mu = w(x, y)dx dy$ , avec  $w$  est une fonction strictement positive qui s'appelle fonction du poids et  $\langle f, g \rangle = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)g(x, y)d\mu$  est un produit scalaire

**Definition :** On dit qu'un ensemble de fonctions  $\{f_i : i = 1, 2, 3, \dots\}$  est orthogonal avec le produit scalaire  $\langle \cdot, \cdot \rangle$  si  $\langle f_i, f_j \rangle = \alpha \delta_{ij}$ , où  $\alpha$  est réel  $\delta_{ij}$  est le symbole de Kronecker définie par :

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

Nous considérons l'ensemble des polynomes non-séparables à deux variables  $P_n(x, y)$ ,  $n \in N$  définie par la relation de récurrence suivante :

$$P_0(x, y) = 1 \tag{2.93}$$

$$a_n = \frac{\langle (x^2 + y^2)P_{n-1}, P_{n-1} \rangle}{\langle P_{n-1}, P_{n-1} \rangle}, \quad n \geq 1 \tag{2.94}$$

$$P_1(x, y) = x^2 + y^2 - a_1 \tag{2.95}$$

$$b_n = \frac{\langle (x^2 + y^2)P_{n-1}, P_{n-2} \rangle}{\langle P_{n-2}, P_{n-2} \rangle}, \quad n \geq 2 \tag{2.96}$$

$$P_n(x, y) = (x^2 + y^2 - a_n)P_{n-1}(x, y) - b_n P_{n-2}(x, y), \quad n \geq 2 \tag{2.97}$$

**Theorem 2.** L'ensemble des polynomes  $P_n(x, y)$ ,  $n \in N$  est orthogonal.

**Preuve.** Nous utilisons la démonstration par récurrence forte sur  $n$  que :

Pour tout  $n \in N$ ,  $\langle P_n, P_i \rangle = 0$  pour tout  $i < n$ .

• Nous prouvons que la propriété est vérifiée pour  $n = 1$ . Nous avons

$$\langle P_1, P_0 \rangle = \langle (x^2 + y^2)P_0 - a_1 P_0, P_0 \rangle = \langle (x^2 + y^2)P_0, P_0 \rangle - a_1 \langle P_0, P_0 \rangle \tag{2.98}$$

L'équation (2.84) donne  $a_1 = \frac{\langle (x^2+y^2)P_0, P_0 \rangle}{\langle P_0, P_0 \rangle}$ , alors  $\langle P_1, P_0 \rangle = 0$ .

- Supposons que la propriété est vérifiée pour tout  $k = 1, \dots, n-1$ , i.e.

$$\langle P_k, P_i \rangle = 0 \text{ pour tout } k = 1, \dots, n-1 \text{ et } i < k. \quad (2.99)$$

- Nous prouvons que la propriété est vérifiée pour  $n$ , i.e,

$$\langle P_n, P_i \rangle = 0 \text{ pour tout } i < n. \quad (2.100)$$

Nous avons

$$\begin{aligned} \langle P_n, P_{n-1} \rangle &= \langle (x^2 + y^2 - a_n)P_{n-1} - b_n P_{n-2}, P_{n-1} \rangle \\ &= \langle (x^2 + y^2)P_{n-1}, P_{n-1} \rangle - a_n \langle P_{n-1}, P_{n-1} \rangle - b_n \langle P_{n-2}, P_{n-1} \rangle \end{aligned} \quad (2.101)$$

L'équation (2.84) donne  $\langle (x^2 + y^2)P_{n-1}, P_{n-1} \rangle - a_n \langle P_{n-1}, P_{n-1} \rangle = 0$  et l'hypothèse d'induction donne  $\langle P_{n-2}, P_{n-1} \rangle = 0$ , alors  $\langle P_n, P_{n-1} \rangle = 0$ . Nous avons aussi

$$\begin{aligned} \langle P_n, P_{n-2} \rangle &= \langle (x^2 + y^2 - a_n)P_{n-1} - b_n P_{n-2}, P_{n-2} \rangle \\ &= \langle (x^2 + y^2)P_{n-1}, P_{n-2} \rangle - a_n \langle P_{n-1}, P_{n-2} \rangle - b_n \langle P_{n-2}, P_{n-2} \rangle \end{aligned} \quad (2.102)$$

Eq. (2.86) donne  $\langle (x^2 + y^2)P_{n-1}, P_{n-2} \rangle - b_n \langle P_{n-2}, P_{n-2} \rangle = 0$  l'hypothèse d'induction donne  $\langle P_{n-1}, P_{n-2} \rangle = 0$  donc  $\langle P_n, P_{n-2} \rangle = 0$ .

Pour  $k < n-2$ , nous avons

$$\begin{aligned} \langle P_n, P_k \rangle &= \langle (x^2 + y^2 - a_n)P_{n-1} - b_n P_{n-2}, P_k \rangle \\ &= \langle (x^2 + y^2)P_{n-1}, P_k \rangle - a_n \langle P_{n-1}, P_k \rangle - b_n \langle P_{n-2}, P_k \rangle \end{aligned} \quad (2.103)$$

L'hypothèse d'induction donne  $\langle P_{n-1}, P_k \rangle = 0$  and  $\langle P_{n-2}, P_k \rangle = 0$ . Alors

$$\langle P_n, P_k \rangle = \langle (x^2 + y^2)P_{n-1}, P_k \rangle = \langle P_{n-1}, (x^2 + y^2)P_k \rangle \quad (2.104)$$

l'équation (2.87) donne  $(x^2 + y^2)P_k = P_{k+1} + a_{k+1}P_k + b_{k+1}P_{k-1}$ . Cette relation et (2.94) donnent,

$$\begin{aligned} \langle P_n, P_k \rangle &= \langle P_{n-1}, P_{k+1} + a_{k+1}P_k + b_{k+1}P_{k-1} \rangle \\ &= \langle P_{n-1}, P_{k+1} \rangle + a_{k+1} \langle P_{n-1}, P_k \rangle + b_{k+1} \langle P_{n-1}, P_{k-1} \rangle \end{aligned} \quad (2.105)$$

et l'hypothèse d'induction donne  $\langle P_{n-1}, P_{k+1} \rangle = 0$ ,  $\langle P_{n-1}, P_k \rangle = 0$  et  $\langle P_{n-1}, P_{k-1} \rangle = 0$ . Par conséquent  $\langle P_n, P_k \rangle = 0$ .

**Proposition 3.** Pour tout  $n \geq 2$

$$\langle P_n, P_n \rangle = \langle P_1, P_1 \rangle \prod_{i=2}^n b_{i+1} \quad (2.106)$$

**Preuve.**

$$\begin{aligned} \langle P_2, P_2 \rangle &= \langle (x^2 + y^2 - a_2)P_1 - b_2P_0, P_2 \rangle = \langle (x^2 + y^2)P_1, P_2 \rangle \\ &= \langle P_1, (x^2 + y^2)P_2 \rangle = \langle P_1, P_3 + a_3P_2 + b_3P_1 \rangle = b_3 \langle P_1, P_1 \rangle \end{aligned} \quad (2.107)$$

Supposons que le résultat est vérifié pour  $n-1$ . Nous devons montrer qu'il est vérifié pour  $n$ .

$$\begin{aligned} \langle P_n, P_n \rangle &= \langle (x^2 + y^2 - a_n)P_{n-1} - b_nP_{n-2}, P_n \rangle \\ &= \langle (x^2 + y^2)P_{n-1}, P_n \rangle = \langle P_{n-1}, (x^2 + y^2)P_n \rangle \\ &= \langle P_{n-1}, P_{n+1} + a_{n+1}P_n + b_{n+1}P_{n-1} \rangle = b_{n+1} \langle P_{n-1}, P_{n-1} \rangle \end{aligned} \quad (2.108)$$

En utilisant l'hypothèse de récurrence, la relation (2.94) peut être écrit

$$\langle P_n, P_n \rangle = \langle P_1, P_1 \rangle \prod_{i=2}^{n-1} b_{i+1} \times b_{n+1} = \langle P_1, P_1 \rangle \prod_{i=2}^n b_{i+1}. \quad (2.109)$$

**Remarque.** Les polynomes  $\{K_n = \rho_n P_n, n \in N\}$  sont orthonormaux i.e.

$$\langle K_n, K_m \rangle = \delta_{nm}. \quad (2.110)$$

où

$$\rho_0 = \frac{1}{\sqrt{\langle P_0, P_0 \rangle}}, \quad \rho_1 = \frac{1}{\sqrt{\langle P_1, P_1 \rangle}} \quad (2.111)$$

$$\rho_n = \frac{1}{\sqrt{\langle P_1, P_1 \rangle \prod_{i=2}^n b_{i+1}}}, \quad n \geq 2 \quad (2.112)$$

## B. Moments invariants orthogonaux proposés

Nous définissons un nouvel ensemble des moments orthogonaux comme suit

$$OIM_n = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} K_n(x - \bar{x}, y - \bar{y}) f(x, y) d\mu, \quad n = 0, 1, \dots \quad (2.113)$$

**Lemme 4.** Pour tout  $n \geq 0$  il exist un unique polynome d'une seule varaible  $Q_n \in R_n[X]$  de degré  $n$  tel que :

$$P_n(x, y) = Q_n(x^2 + y^2), \quad \forall (x, y) \in R^2. \quad (2.114)$$

**Preuve.** Nous utilisons aussi la démonstration par récurrence sur  $n$ .

$P_0(x, y) = 1$  et  $P_1(x, y) = x^2 + y^2 - a_1$  alors  $Q_0(X) = 1$  et  $Q_1(X) = X - a_1$ . Soit  $n \geq 2$  supposons que la propriété est vraie pour  $k \leq n - 1$ .

$$\begin{aligned} P_n(x, y) &= (x^2 + y^2 - a_n)P_{n-1}(x, y) - b_n P_{n-2}(x, y) \\ &= (x^2 + y^2 - a_n)Q_{n-1}(x^2 + y^2) - b_n Q_{n-2}(x^2 + y^2) = Q_n(x^2 + y^2) \end{aligned} \quad (2.115)$$

où  $Q_n(X) = (X - a_n)Q_{n-1}(X) - b_n Q_{n-2}(X)$  est un polynome de degré  $n$ .

**Théorème 5.** Pour tout  $n \geq 0$  le moment  $OIM_n$  est invariant à la translation, rotation et changement d'échelle. En plus

$$OIM_n = \sum_{i=1}^n \beta_i \varphi'_i(f) = \sum_{i=1}^n \sum_{k=0}^i \beta_i \binom{i}{k} \mu_{2k, 2i-2k}(f) \quad (2.116)$$

**Preuve.** En utilisant (2.103) et la remarque précédente, nous obtenons

$$OIM_n = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} K_n(x - \bar{x}, y - \bar{y}) f(x, y) d\mu = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \rho_n P_n(x - \bar{x}, y - \bar{y}) f(x, y) d\mu$$

D'après le lemme 4, il existe  $(\alpha_i)_{i=1, \dots, n}$ ,  $\alpha_n \neq 0$  tel que

$$Q_n(X) = \sum_{i=0}^n \alpha_i X^i \text{ et } P_n(x, y) = Q_n(x^2 + y^2). \quad (2.117)$$

Donc la formule (2.106) peut être écrit comme

$$\begin{aligned} OIM_n &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \rho_n Q_n((x - \bar{x})^2 + (y - \bar{y})^2) f(x, y) d\mu \\ &= \rho_n \sum_{i=1}^n \alpha_i \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [(x - \bar{x})^2 + (y - \bar{y})^2]^i f(x, y) d\mu \\ &= \rho_n \sum_{i=1}^n \alpha_i \sum_{k=0}^i \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^{2k} (y - \bar{y})^{2i-2k} f(x, y) d\mu \\ &= \sum_{i=1}^n \sum_{k=0}^i \beta_i \binom{i}{k} \mu_{2k, 2i-2k}(f) \end{aligned} \quad (2.118)$$

Où  $\beta_i = \rho_n \alpha_i$ .

Pour une image  $f(x, y)$  de taille  $N \times M$  l'algorithme 1 permet de calculer l'ensemble des polynômes  $P_n(x, y)$  et  $K_n(x, y)$  pour tout  $x = 0, \dots, N - 1$  et tout  $y = 0, \dots, M - 1$ .

---

**Algorithme 1 :** Calcule les polynômes  $P_n(x, y)$  et  $K_n(x, y)$  pour tout  $x = 0 : N - 1$  et  $y = 0 : M - 1$

---

- Input : les entiers N, M et n.
- Output : les nombres réels  $P_n(x, y)$  pour tous  $x = 0 : N - 1$  et  $y = 0 : M - 1$
- **Etape 1**(initialisation)
  1.  $a_1 = \frac{\langle x^2+y^2, 1 \rangle}{\langle 1, 1 \rangle} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (x^2+y^2)}{N \times M}$
  2.  $P_0(x, y) = 1$
  3.  $\rho_0 = \frac{1}{\sqrt{\langle 1, 1 \rangle}}$
  4.  $P_1(x, y) = x^2 + y^2 - a_1$
  5.  $\rho_1 = \frac{1}{\sqrt{\langle P_1, P_1 \rangle}}$
- **Etape 2 :** pour  $i = 2 : n$ 
  1.  $a_i = \frac{\langle (x^2+y^2)P_{i-1}, P_{i-1} \rangle}{\langle P_{i-1}, P_{i-1} \rangle}$
  2.  $b_i = \frac{\langle (x^2+y^2)P_{i-1}, P_{i-2} \rangle}{\langle P_{i-2}, P_{i-2} \rangle}$
  3.  $P_i(x, y) = (x^2 + y^2 - a_i)P_{i-1}(x, y) - b_iP_{i-2}(x, y)$
  4.  $\rho_i = \frac{1}{\sqrt{\langle P_i, P_i \rangle}}$
- **Etape 3 :**  
Afficher le polynôme  $P_n$  sous forme d'une matrice  $(P_n(x, y))_{x=0:N-1; y=0:M-1}$
- **Etape 4 :**  
Afficher le polynôme  $K_n$  sous forme d'une matrice  $(K_n(x, y))_{x=0:N-1; y=0:M-1}$

$$K_n(x, y) = \rho_n P_n(x, y)$$


---

Il est clair que l'ensemble des polynômes  $\{P_n, n = 0, 1, \dots\}$  présenté dans la relation de récurrence et les moments orthogonaux invariants  $OIM_n, n = 0, 1, \dots$  définis dans (34) ne dependent que de la fonction du poids  $w(x, y)$ . Dans cette section, nous allons aborder

les trois exemples suivants de fonctions du poids.

$$w_1(x, y) = 1 \text{ si } (x, y) \in B(0, 1), \quad w_1(x, y) = 0 \text{ sinon} \quad (2.119)$$

$$w_2(x, y) = e^{-(x^2+y^2)} \text{ pour tout } (x, y) \in \mathbb{R}^2 \quad (2.120)$$

$$w_3(x, y) = -2(x^2 + y^2) + \sqrt{x^2 + y^2} \text{ si } (x, y) \in B\left(0, \frac{1}{\sqrt{2}}\right), \quad w_3(x, y) = 0 \text{ sinon} \quad (2.121)$$

où  $B(0, r)$  est la boule fermée de centre  $O$  et de rayon  $r$  :

$$B(0, r) = \{(x, y) \in \mathbb{R}^2, x^2 + y^2 \leq r^2\}$$

On note  $OIM^{(i)}$  le moment orthogonal invariant associé à la fonction du poids  $w_i, i = 1, 2, 3$  pour les trois exemples précédents. Maintenant, nous présenterons des simulations numériques pour valider le cadre théorique précédent. Dans ce contexte, nous présenterons deux expériences. Dans la première, nous testerons l'invariabilité des moments proposés pour les trois transformations, translation, rotation et changement d'échelle. Dans la seconde expérience nous présentons les résultats d'une étude comparative avec d'autres moments invariants. Dans cette expérience, nous utilisons l'image de Lena présentée dans la figure (2.7, T1) pour tester l'invariabilité des moments invariants orthogonaux proposés  $OIM^{(i)}, i = 1, 2, 3$  par rapport à trois transformations affines : la translation, la rotation et le changement d'échelle. L'image de Lena est transformée avec des changements d'échelles de facteurs  $\lambda_k = 0.5 + (0.05)k, k = 0, \dots, 20$ , des rotations d'angles  $\theta_k = 10k, k = 0, \dots, 36$  et des translations des vecteurs  $u_k = (-5 + 0.5k, -5 + 0.5k), k = 0, \dots, 20$ . Pour mesurer le degré d'invariabilité des trois moments invariants  $OIM^{(i)}, i = 1, 2, 3$  nous utilisons l'erreur relative entre l'image originale  $f$  et les images transformées  $f_k$  défini par :

$$E^{(i)}(f, f_k) = \frac{\|OIM^{(i)}(f) - OIM^{(i)}(f_k)\|}{\|OIM^{(i)}(f)\|}, k = 0, \dots, p \quad (2.122)$$

Où  $p$  est le nombre d'images transformées,  $\|\cdot\|$  est la norme euclidienne et  $OIM^{(i)}(f); OIM^{(i)}(f_k)$  sont respectivement les moments invariants de l'image originale  $f$  et les images transformées  $f_k$ . Les résultats obtenus sont illustrés dans les figures 2.10, 2.11 et 2.12.

Les résultats obtenus montrent que les trois types de moments invariants proposés  $OIM^{(i)}, i = 1, 2, 3$  sont très stables pour les trois transformations d'image avec un peu de supériorité du moment invariant orthogonal  $OIM^{(3)}$ . Par conséquent, les moments invariants orthogonaux présentés pourraient être des outils utiles pour les tâches de reconnaissance de formes qui nécessitent la propriété d'invariabilité sous la translation, la rotation et le changement d'échelle de l'image.

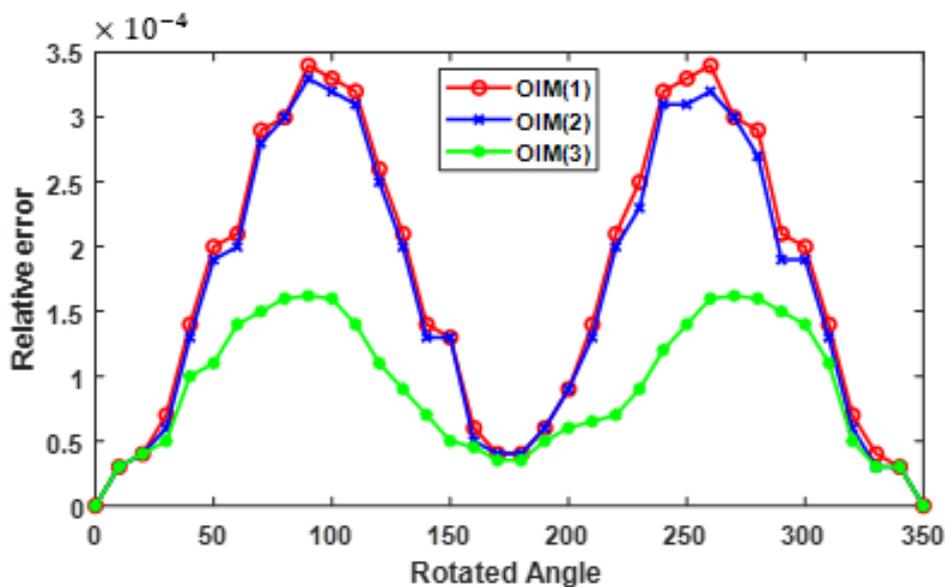


FIGURE 2.10 – l'erreur relative  $E(i)(f, f_k)$  pour les images transformées par des rotations en utilisant les trois moments orthogonaux invariants  $MOI(i), i = 1, 2, 3$ .

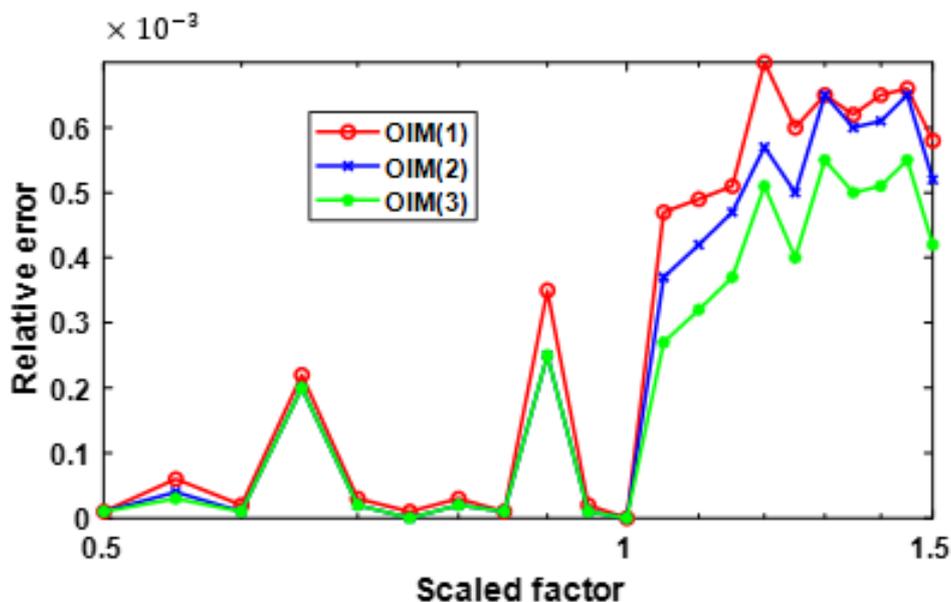


FIGURE 2.11 – l'erreur relative  $E(i)(f, f_k)$  pour les images mises à l'échelle en utilisant les trois moments orthogonaux invariants  $MOI(i), i = 1, 2, 3$ .

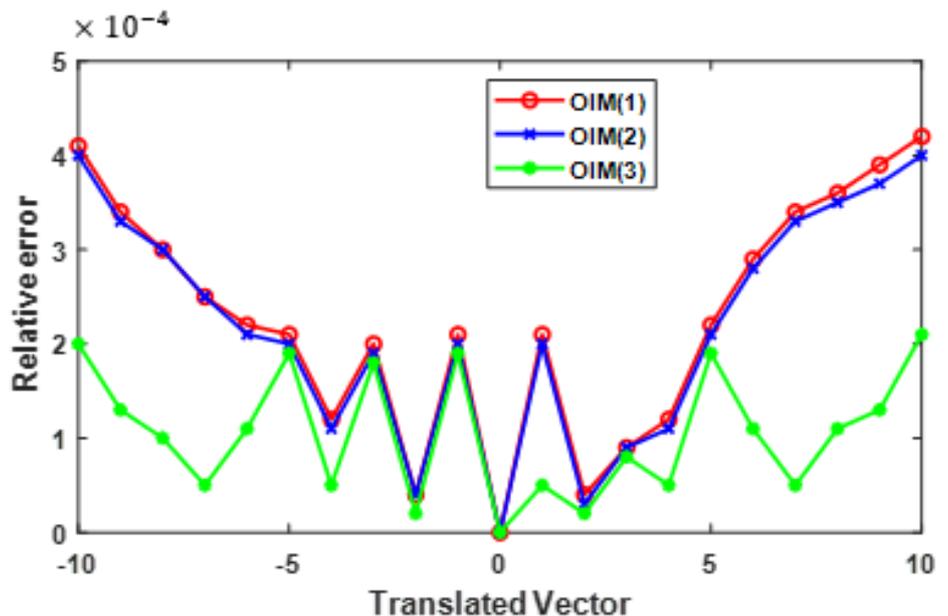


FIGURE 2.12 – l'erreur relative  $E(i)(f, f_k)$  pour les images traduites en utilisant les trois moments orthogonaux invariants  $MOI(i)$ ,  $i = 1, 2, 3$ .

## 2.4 Moments orthogonaux radiaux de Jacobi invariants

### 2.4.1 Moments orthogonaux radiaux de Jacobi

Ping et all [128] sont les premiers qui ont utilisé les polynômes de Jacobi orthonormés décalés comme noyau pour les moments radiaux, qui sont définis comme suit

$$S_n^{\alpha, \beta}(r) = \sqrt{\frac{w(\alpha, \beta, r)}{b_n(\alpha, \beta)r}} J_n^{\alpha, \beta}(r) \quad (2.123)$$

Où  $J_n^{\alpha, \beta}(r)$  sont les polynômes de Jacobi décalés,  $b_n(\alpha, \beta)$  est la constante de normalisation et  $w(\alpha, \beta, r)$  est la fonction de poids définis par

$$J_n^{\alpha, \beta}(r) = \frac{n! \Gamma(\beta)}{\Gamma(\alpha + n)} \sum_{s=0}^n (-1)^s \frac{\Gamma(\alpha + n + s)}{(n - s)! s! \Gamma(\beta + s)} r^s \quad (2.124)$$

$$b_n(\alpha, \beta) = \frac{n! \Gamma^2(\beta) \Gamma(\alpha - \beta + n + 1)}{\Gamma(\beta + n) \Gamma(\alpha + n) \Gamma(\alpha + 2n)} \quad (2.125)$$

$$w(\alpha, \beta, n) = (1 - r)^{\alpha - \beta} r^{\beta - 1} \quad (2.126)$$

où  $\Gamma(\cdot)$  est la fonction Gamma,  $\alpha - \beta \geq -1$ ,  $\alpha, \beta > 0$ . Les polynômes de Jacobi orthonormés décalés radiaux  $S_n^{\alpha, \beta}$  sont orthogonaux sur l'intervalle  $[0, 1]$  et la condition

d'orthogonalité donnée par

$$\int_0^1 S_n^{\alpha,\beta}(r) S_m^{\alpha,\beta}(r) r dr = \delta_{nm} \quad (2.127)$$

avec  $\delta_{nm}$  est le symbole de kronecker. Les calculs des factorielles et de la fonction Gamma dans les équations (2.124) et (2.125) augmentent le temps de calcul et ne sont précis que pour les factorielles inférieures à 21. De plus, le calcul de  $r^n$  dans des ordres supérieurs à 21 provoque une instabilité numérique des valeurs autour de  $r = 1$  à l'intérieur du disque unitaire. Récemment, Camacho et al. [129] ont proposé une relation de récurrence par rapport à  $n$  pour augmenter la stabilité numérique pour le calcul des polynômes orthonormaux de Jacobi décalés. La relation récursive est donnée par

$$a_n S_n^{\alpha,\beta}(r) = (2r - 1 - b_n) S_{n-1}^{\alpha,\beta}(r) - a_{n-1} S_{n-2}^{\alpha,\beta}(r) \quad (2.128)$$

où  $r \in [0, 1]$ ,  $\alpha - \beta > -1$ ,  $\alpha, \beta > 0$  et les coefficients sont calculés comme suit

$$a_n = \sqrt{\frac{4n(n + \alpha - \beta)(n + \beta - 1)(n + \alpha - 1)}{(2n + \alpha - 1)^2(2n + \alpha)(2n + \alpha - 2)}} \quad (2.129)$$

$$b_n = \frac{(\alpha - 1)(2\beta - \alpha - 1)}{(2n + \alpha - 1)(2(n - 1) + \alpha - 1)}. \quad (2.130)$$

Pour commencer le calcul numérique, les polynômes de Jacobi orthonormés décalés d'ordres 0 et 1 sont donnés par

$$S_0^{\alpha,\beta}(r) = \sqrt{\frac{w(\alpha, \beta, r)}{b_0(\alpha, \beta)}} \quad (2.131)$$

$$S_1^{\alpha,\beta}(r) = S_0^{\alpha,\beta}(r) \sqrt{\frac{(\alpha + 2)\beta}{\alpha - \beta + 1}} \left( \frac{\alpha + 1}{\beta} r - 1 \right). \quad (2.132)$$

L'expression générale de moment radial de Jacobi d'ordre  $n + m$ , pour une fonction d'image  $f(r, \theta)$  donnée en coordonnées polaires est donnée par

$$MRJ_{nm} = \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 S_n^{\alpha,\beta}(r) S_m^{\alpha,\beta}(r) f(r, \theta) e^{-im\theta} r dr d\theta \quad (2.133)$$

En se basant sur la propriété de l'orthogonalité de polynômes de Jacobi orthonormés décalés, l'image  $f(r, \theta)$  peut être reconstruite à l'aide d'un nombre fini des moments radiaux de Jacobi MORJs. Pour  $0 \leq p \leq pmax$  et  $-qmax \leq q \leq qmax$  l'image reconstruite

peut être écrite sous la forme

$$f(r, \theta) = \sum_{p=0}^{+\infty} \sum_{q=-\infty}^{+\infty} MRJ_{pq} S_p^{\alpha, \beta}(r) e^{iq\theta} \quad (2.134)$$

$$\approx \sum_{p=0}^{pmax} \sum_{q=-qmax}^{+qmax} MRJ_{pq} S_p^{\alpha, \beta}(r) e^{iq\theta} \quad (2.135)$$

## 2.4.2 La dérivation des invariants de moments orthogonaux radiaux de Jacobi

### 2.4.3 L'invariance à la rotation

Pour une image originale  $f(r, \theta)$ , et sa version pivotée,  $f^\alpha(r, \theta)$ , avec un angle  $\alpha$  : les moments orthogonaux sont appelés invariants à la rotation si les valeurs de magnitude pour les deux moments sont identiques. Les moments radiaux de Jacobi (MRJs) de l'image pivotée  $f^\alpha(r, \theta)$  sont calculés comme suit

$$MRJ_{nm}(f^\alpha) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 S_n^{\alpha, \beta}(r) S_m^{\alpha, \beta}(r) f^\alpha(r, \theta) e^{-im\theta} r dr d\theta \quad (2.136)$$

$$= \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 S_n^{\alpha, \beta}(r) S_m^{\alpha, \beta}(r) f(r, \theta + \alpha) e^{-im\theta} r dr d\theta \quad (2.137)$$

Nous posons,  $\theta' = \theta + \alpha$ , nous obtenons

$$MRJ_{nm}(f^\alpha) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 S_n^{\alpha, \beta}(r) S_m^{\alpha, \beta}(r) f(r, \theta + \alpha) e^{-im\theta} r dr d\theta \quad (2.138)$$

$$= e^{im\alpha} \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 S_n^{\alpha, \beta}(r) S_m^{\alpha, \beta}(r) f(r, \theta') e^{-im\theta'} r dr d\theta' \quad (2.139)$$

$$= e^{im\alpha} MRJ_{nm}(f) \quad (2.140)$$

avec  $MRJ_{nm}(f)$  et  $MRJ_{nm}(f^\alpha)$  sont respectivement les moments radiaux de Jacobi de l'image originale et de l'image pivotée par l'angle  $\alpha$ . D'après cette dernière relation, on déduit que

$$MRJ_{nm}(f^\alpha) = e^{im\alpha} MRJ_{nm}(f) \quad (2.141)$$

Comme  $|e^{im\alpha}| = 1$ , quelque soit la valeur de  $\alpha$ , donc

$$|MRJ_{nm}(f^\alpha)| = |MRJ_{nm}(f)|, \quad (2.142)$$

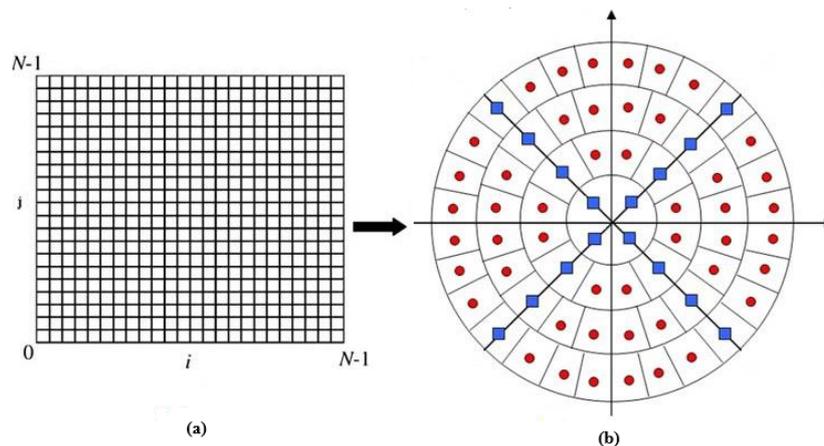


FIGURE 2.13 – image rectangulaire (a), image circulaire (b).

Cela signifie que  $|MRJ_{nm}|$  sont invariants à la rotation.

#### 2.4.4 L'invariance à la translation

Selon Suck et Flusser [130], l'invariance à la translation s'obtient en effectuant un changement de repère. En considérant le nouveau repère tel que son point d'origine est le barycentre  $C(\bar{x}, \bar{y})$  de l'image. On obtient alors ce qu'on appelle des moments centraux radiaux de Jacobi qui s'écrivent en fonction de  $\bar{r}$  sous la forme suivante :

$$MRJ_{nm}(f) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 S_n^{\alpha,\beta}(\bar{r}) S_m^{\alpha,\beta}(\bar{r}) f(\bar{r}, \theta) e^{-im\theta} \bar{r} d\bar{r} d\theta \quad (2.143)$$

$$\bar{r} = \sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} \quad (2.144)$$

Ces moments sont invariants à la translation.

#### 2.4.5 L'invariance au changement d'échelle

Selon Wang et all 2015 [131], les moments orthogonaux radiaux sont invariants au changement d'échelle si la zone de calcul et le domaine de l'image sont identiques. Cette condition n'est pas remplie pour les images rectangulaires. Pour cela nous utilisons la technique *rectangle-au-cercle* (Figure 2.13) qui permet de convertir une image sous forme rectangulaire à une image circulaire définie sur le disque unité  $D(O, 1)$ . Par conséquent, l'image de taille  $2N \times 2N$  définie en coordonnées cartésiennes  $(x, y)$  peut être convertie à une image circulaire en coordonnées polaires  $(r, \theta)$  définie sur le disque unité divisé en  $N$



FIGURE 2.14 – image Barbara, image Lena.

anneaux au long de la direction radiale, qui sont séparés par des cercles de rayons :

$$r_k = \frac{k}{N}, k = 0, 1, \dots, N - 1 \quad (2.145)$$

et chaque anneau d'ordre  $j$  peut être divisé en  $8j + 4$  parties égales déterminées par les angles

$$\theta_{jk} = \frac{2\pi(k + 0.5)}{4 + 8j}, k = 0, 1, \dots, 3 + 8j \quad (2.146)$$

En fin, le moment radial de Jacobi peut être écrit en sa forme discrete suivante

$$MRJ_{nm}(f) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{3+8j} \frac{1}{(4 + 8j)} S_n^{\alpha,\beta}(r_j) f(r_j, \theta_{jk}) r_j e^{-im\theta_{jk}} \quad (2.147)$$

Avec cette technique, la condition de wang et al. est remplie. Par conséquent les moments radiaux de Jacobi calculés par (2.147) sont invariants au changement d'échelle.

### 2.4.6 Tests expérimentaux sur les moments radiaux de Jacobi

Pour valider le cadre théorique des moments radiaux de Jacobi, nous présentons deux tests expérimentaux. Dans le premier, nous testons l'invariance des moments invariants proposés en translation, rotation et changement d'échelle. Dans le second test, nous évaluons la capacité des moments radiaux de Jacobi pour la reconstruction d'images. Dans ce contexte, nous avons travaillé sur l'image "Barbara" et l'image "Lena" présentées dans la figure 2.14, et dans chaque expérience nous avons présenté une étude comparative avec d'autres moments bien connus comme les moments rationnels de Chebyshev (MRCb) [132], les moments radiaux de Legendre décalés (MRLD)[133], les moments fractionnaires de Chebyshev (MFrCb) [134], les moments fractionnaires de Legendre décalés

## 2.4. MOMENTS ORTHOGONAUX RADIAUX DE JACOBI INVARIANTS

TABLE 2.2 – Les valeurs de l’erreur relative MSE pour la rotation en utilisant les moments proposés MRJ, et les autres moments testés pour l’image ”Barbara”

Rotation d’angle	Notre MRJ	MRCb	MLD	MFrCb	MFrLD	MFrCh
0°	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
30°	$5.167e^{-08}$	$2.930e^{-02}$	$3.069e^{-03}$	$2.853e^{-04}$	$4.058e^{-04}$	$2.057e^{-03}$
60°	$9.249e^{-08}$	$2.521e^{-02}$	$2.628e^{-03}$	$2.443e^{-04}$	$3.619e^{-04}$	$3.617e^{-03}$
90°	$9.046e^{-09}$	$4.147e^{-03}$	$8.025e^{-04}$	$4.759e^{-05}$	$8.127e^{-05}$	$9.129e^{-04}$
120°	$9.388e^{-08}$	$2.987e^{-02}$	$2.158e^{-03}$	$2.404e^{-04}$	$3.643e^{-04}$	$4.643e^{-03}$
150°	$9.257e^{-08}$	$2.413e^{-02}$	$2.103e^{-03}$	$2.012e^{-04}$	$3.003e^{-04}$	$6.014e^{-03}$
180°	$4.214e^{-12}$	$7.009e^{-04}$	$4.312e^{-05}$	$8.711e^{-06}$	$6.258e^{-06}$	$8.257e^{-05}$

TABLE 2.3 – Les valeurs de l’erreur relative MSE pour la translation en utilisant les moments proposés MRJ, et les autres moments testés pour l’image ”Barbara”

Translation de ve	Notre MRJ	MRCb	MLD	MFrCb	MFrLD	MFrCh
(-5, -5)	$8.923e^{-08}$	$4.212e^{-05}$	$7.598e^{-05}$	$6.624e^{-04}$	$8.568e^{-04}$	$7.557e^{-05}$
(-3, -3)	$1.039e^{-08}$	$7.817e^{-06}$	$2.608e^{-05}$	$9.847e^{-06}$	$8.884e^{-05}$	$7.873e^{-06}$
(-1, -1)	$1.023e^{-08}$	$7.418e^{-06}$	$2.121e^{-05}$	$3.578e^{-06}$	$5.232e^{-05}$	$3.221e^{-06}$
(1, 1)	$1.023e^{-08}$	$7.419e^{-06}$	$2.121e^{-05}$	$3.578e^{-06}$	$5.232e^{-05}$	$3.223e^{-06}$
(3, 3)	$1.038e^{-08}$	$7.817e^{-06}$	$2.608e^{-05}$	$9.847e^{-06}$	$8.885e^{-05}$	$7.874e^{-06}$
(5, 5)	$8.924e^{-08}$	$4.213e^{-05}$	$7.598e^{-05}$	$6.624e^{-04}$	$8.565e^{-04}$	$7.561e^{-05}$

(MFrLD) [135] et les moments fractionnaires de Charlier (MFrCh) [136]. Dans la première expérience l’image de Barbara est transformée avec des changements d’échelles de facteurs 0.5, 0.75, 1, 1.25, 1.5, 1.75 et 2 des rotations d’angles 0°, 30°, 60°, 90°, 120°, 150°, 180° et des translations des vecteurs (-5, -5), (-3, -3), (-1, -1), (1, 1), (3, 3), (5, 5). Pour mesurer le degré d’invariance des invariants nous utilisons l’erreur relative entre l’image originale  $f$  et les images transformées défini précédemment. Les résultats obtenus sont représentés dans les tableaux 2.2-2.4. D’après ces résultats, nous remarquons que l’erreur relative entre l’image originale et l’image à mise à l’échelle pour notre moment (MRJ) est inférieure à l’erreur des autres moments testés quelque soient l’angle de rotation, le facteur de changement d’échelle et le vecteur de translation. Cela montre la supériorité et l’efficacité de notre moment par rapports aux moments testés pour l’invariance aux trois transformations géométriques. Dans la seconde expérience, nous présentons un test pour évaluer la capacité des moments orthogonaux radiaux de Jacobi pour la reconstruction d’images en utilisant la formule

$$\hat{f}(x, y) = \sum_{m=0}^{max} \sum_{n=0}^m MRJ_{m-n,n} S_{m-n}^{\alpha,\beta}(x) S_n^{\alpha,\beta}(y). \quad (2.148)$$

## 2.5. CONCLUSION

TABLE 2.4 – Les valeurs de l’erreur relative MSE pour le changement d’échelle en utilisant les moments proposés MRJ, et les autres moments testés pour l’image ”Barbara”

Echelle de change	Notre MRJ	MRCb	MLD	MFrCb	MFrLD	MFrCh
0.5	$8.923e^{-08}$	$4.212e^{-05}$	$7.598e^{-05}$	$6.624e^{-04}$	$8.568e^{-04}$	$7.557e^{-05}$
0.75	$2.067e^{-07}$	$2.546e^{-02}$	$2.822e^{-02}$	$2.577e^{-04}$	$3.574e^{-03}$	$7.574e^{-03}$
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.25	$5.986e^{-08}$	$2.762e^{-03}$	$5.654e^{-03}$	$7.731e^{-05}$	$8.728e^{-04}$	$1.728e^{-04}$
1.5	$9.911e^{-08}$	$5.814e^{-03}$	$7.063e^{-03}$	$9.930e^{-05}$	$1.927e^{-03}$	$5.927e^{-03}$
1.75	$7.104e^{-08}$	$5.109e^{-03}$	$9.263e^{-03}$	$9.971e^{-05}$	$1.968e^{-03}$	$5.968e^{-03}$
2	$7.714e^{-09}$	$9.245e^{-04}$	$2.215e^{-03}$	$7.877e^{-06}$	$5.874e^{-05}$	$1.874e^{-04}$

TABLE 2.5 – L’erreur de la reconstruction d’image ”Lena” en fonction d’ordre maximum de moment varie de 30 jusqu’à 140, en utilisant les moments proposés MRJ et les autres moments testés.

Ordre Max	30	50	70	90	100	120	140
Notre MRJ	0.0280	0.0190	0.0131	0.0044	0.0039	0.0039	0.0008
MFrCb	0.0411	0.0207	0.0179	0.0078	0.0073	0.0073	0.0073
MFrLD	0.0364	0.0234	0.0210	0.0122	0.0117	0.0117	0.0098
MFrCh	0.0314	0.0172	0.0111	0.0050	0.0045	0.0045	0.0046
MRCb	0.0338	0.0175	0.0113	0.0051	0.0046	0.0046	0.0049
MLD	0.0349	0.0177	0.0114	0.0052	0.0048	0.0048	0.0052

Pour mesurer la capacité de la reconstruction d’image, nous calculons l’erreur quadratique moyenne (MSE) entre l’image originale  $f(x, y)$  et l’image reconstruite  $\hat{f}(x, y)$  présentée dans l’équation précédente. Où  $MRJ_{ij}$  sont les moments orthogonaux radiaux de Jacobi. Nous réalisons le test expérimental sur l’image Lena présenté dans la figure (2.14). Sachant que l’ordre des moments maximum est compris entre 30 et 140. Le tableau (2.5) montre les valeurs de MSE des moments orthogonaux proposés (MRJ) et des moments testés Il est évident que le MSE diminue lorsque le moment augmente et le MSE se rapproche de zéro lorsque l’ordre du moments assez grand. L’erreur de la reconstruction d’image en utilisant les moments radiaux de Jacobi MRJ est très petit par rapport aux autres moments testés quel que soit l’ordre maximum.

## 2.5 Conclusion

Hu est le premier qui a utilisé les moments invariants géométriques dans la reconnaissance de formes. Basé sur la théorie des invariants algébriques, il a dérivé sept invariants des moments géométriques, qui sont indépendants à la position, à la taille et à l’orientation. Ces moments sont d’ordre fini, ils ne comprennent pas toutes les caractéristiques

d'image. Pour cette raison, nous avons introduit dans ce travail une autre suite de moments invariants en fonction des moments centraux normalisés d'ordre infini. La propriété de non-orthogonalité de ces moments provoque la redondance de l'information. Pour surmonter ce problème, nous avons proposé une nouvelle technique de construction de polynômes orthogonaux non-séparables à deux variables, qui est basée sur une formule de récurrence. Nous avons présenté aussi deux autres ensembles des moments orthogonaux : l'ensemble de moments de Legendre adaptés (MLAs) et l'ensemble de moments radiaux de Jacobi (MRJs). Pour montrer l'efficacité et la supériorité des approches proposées nous avons effectué plusieurs tests expérimentaux de vision par ordinateur. La performance de ces moments invariants est comparée avec des moments orthogonaux très récents, tels que les moments rationnels de Chebyshev (MRCb) [132], les moments radiaux de Legendre décalés (MRLD)[133], les moments fractionnaires de Chebyshev (MFrCb) [134], les moments fractionnaires de Legendre décalés (MFrLD) [135], les moments fractionnaires de Charlier (MFrCh)[136], etc. Les résultats expérimentaux obtenus montrent la supériorité et l'efficacité des moments proposés, et surtout les moments  $OIM^{(3)}$  liés à la fonction du poids  $w_3$ , sont mieux que les autres moments invariants testés. Tout cela nous incite, dans les travaux futurs, à chercher des méthodes d'optimisation afin de déterminer la fonction du poids qui donne les meilleurs résultats dans la reconnaissance des formes. En fin, nous sommes arrivé à exposer la seconde étape d'extraction des vecteurs descripteurs en se basant sur les moments invariants orthogonaux. Dans les deux derniers chapitres nous allons tester nos vecteurs descripteurs dans le domaine de la classification des bases de données bruitées et de la récupération d'une image en utilisant quelques méthodes de machine Learning.

# Chapitre 3

## Classification d'images par la méthode des éléments finis mixtes et la théorie des moments invariants

Dans les chapitres précédents nous avons exposé les deux étapes essentielles d'extraction des vecteurs descripteurs en se basant sur le filtrage d'images et en résolvant le modèle de Perona-Malik par la méthode des éléments finis mixtes et sur la théorie des moments orthogonaux. Pour tester la fiabilité de cette approche, nous présentons dans ce chapitre quelques systèmes de classifications des bases de données d'images. La classification des images représente une discipline vaste, qui regroupe plusieurs techniques et méthodes visant à identifier une forme au sens informatique à partir des données brutes. Les méthodes de classification les plus connues peuvent être séparées en deux grandes catégories : les méthodes de classification supervisée et les méthodes de classification non supervisée. Pour notre étude, en se basant sur les vecteurs descripteurs extraits à l'aide des étapes présentés dans les chapitres précédents, nous nous intéressons aux trois algorithmes de classification supervisée et trois algorithmes de classification non-supervisée. Pour la classification supervisée, nous avons amélioré la méthode k-plus proches voisins (k-PP) et les deux méthodes de réseaux de neurons artificiels : Le perceptron multicouche (multilayer perceptron MLP) et la méthode à base radiale (radial basis function RBF). Et pour la classification non-supervisée nous avons appliqué la méthode de regroupement k-moyennes (k-means clustering KM), la méthode de regroupement k-moyennes floue (fuzzy k-means clustering FKM) et la méthode possibiliste k-moyennes floue (possibilistic fuzzy k-means clustering PFKM ). Dans ce cadre, nous utilisons des bases de données d'images bien connues comme : la base de données MPEG-7-CE Shape [143], Columbia Object Image Library (COIL-20), Columbia Object Image Library (COIL-100) [144], ORL [145],

MNIST fassion [146] et la base de données "MNIST hanwritten" [147].

## 3.1 Méthodes de clasification non-supérvisée

L'objectif d'une méthode de classification déborde le cadre strictement exploratoire. C'est la recherche d'une typologie, ou segmentation, c'est-à-dire d'une partition, ou répartition des individus en classes homogènes, ou catégories. Ceci est fait en optimisant un critère visant à regrouper les individus dans des classes, chacune le plus homogène possible et, entre elles, les plus distinctes possible. Cet objectif est à distinguer des procédures de discrimination, ou encore de classification pour lesquelles une typologie est a priori connue, au moins pour un échantillon d'apprentissage. Nous sommes dans une situation d'apprentissage non-supervisé, ou en anglais de clustering. Il existe de très nombreuses méthodes de classification non supervisées, nous nous intéressons à la méthode de K-moyennes (K-means clustering : KM), la méthode de k-moyennes floue (fuzzy k-means clustering : FKM) et la méthode possibiliste k-moyennes floue (possibilistic fuzzy k-means clustering : PFKM).

### 3.1.1 Méthode de k-means clustering : KM

L'algorithme k-means a été utilisé pour la première fois par James MacQueen en 1967, un des plus simples algorithmes d'apprentissage non supervisé, appelée algorithme des centres mobiles, il attribue chaque point dans un cluster dont le centre (centroïde) est le plus proche. Le centre est la moyenne de tous les points dans le cluster, ses coordonnées sont la moyenne arithmétique pour chaque dimension séparément de tous les points dans le cluster c'est à dire chaque cluster est représentée par son centre de gravité. Cet algorithme fut longtemps utilisé sur les grands bases de données en raison de sa rapidité. Elle consiste à supposer qu'il existe  $k$  classes distinctes. On commence par désigner  $k$  centres de classes  $c_1, \dots, c_k$  parmi les individus. Ces centres peuvent être soit choisis par l'utilisateur pour leur "représentativité", soit désignés aléatoirement. On réalise ensuite itérativement les deux étapes suivantes :

- Pour chaque individu qui n'est pas un centre de classe, on regarde quel est le centre de classe le plus proche. On définit

$$C_i = \{\text{ensemble des points les plus proches du centre } c_i\}.$$

- Dans chaque nouvelle classe  $C_i$ , on définit le nouveau centre de classe  $c_i$  comme étant le barycentre des points de  $C_i$ . L'algorithme s'arrête suivant un critère d'arrêt fixé par

l'utilisateur qui peut être choisi parmi les suivants : soit le nombre limite d'itérations est atteint, c'est-à-dire qu'entre deux itérations les classes formées restent les mêmes, soit l'algorithme a convergé, c'est-à-dire que l'inertie intra-classe ne s'améliore quasiment plus entre deux itérations.

La principale limite de cette méthode est la dépendance des résultats des valeurs de départ (centres initiaux). À chaque initialisation correspond une solution différente (optimum local) qui peut dans certain cas être très loin de la solution optimale (optimum global).

**Algorithme k-means :**

**Entrée**

- Ensemble de  $N$  données, noté par  $X = \{x_1, x_2, \dots, x_N\}$
- Nombre de groupes souhaité, noté par  $k$

**Sortie**

- Une partition de  $k$  groupes

**Début**

- 1) Initialisation aléatoire des centres  $C_k$

**Répéter**

- 2) Affectation : générer une nouvelle partition en assignant chaque objet au groupe dont le centre est le plus proche :

$$x_i \in C_k \Leftrightarrow \min_{j=1, \dots, K} d(x_i, c_j) = d(x_i, c_k)$$

- 3) Représentation : Calculer les centres associés à la nouvelle partition :

$$c_k = \frac{1}{N_k} \sum_{x_i \in C_k} x_i$$

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

**Fin.**

#### Méthode global k-means clustering

Il y a une autre version de l'algorithme k-means s'appelle "Algorithme global k-means" [15]. Cet algorithme est une solution au problème d'initialisation du k-means, elle est fondé sur les données et vise à atteindre une solution globalement optimale. Elle consiste à effectuer un clustering incrémental et à ajouter dynamiquement un nouveau centre suivi par l'application du k-means jusqu'à la convergence. Les centres sont choisis un par un de la façon suivante : le premier centre est le centre de gravité de l'ensemble des données (résultat de l'application du k-means avec  $k=1$ ), les autres centres sont tirés de l'ensemble

de données ou chaque donnée est une candidate pour devenir un centre, cette dernière sera testée avec le reste de l'ensemble, le meilleur candidat est celui qui minimise la fonction objectif

$$J(x_1, x_2, \dots, x_N, c_1, \dots, c_k) = \sum_{i=1}^K \sum_{x_j \in C_i} \|x_j - c_i\|^2. \quad (3.1)$$

### 3.1.2 Méthode fuzzy k-means clustering : FKM

La plupart des algorithmes analytiques de classification floue (ainsi que tous les algorithmes présentés dans ce chapitre) sont basés sur l'optimisation de la fonction objectif de base de k-means, ou sur une certaine modification de celle-ci. Soient  $X = \{X_j; j = 1, \dots, N\}$  un ensemble des vecteurs de  $R^p$ , avec  $X_j = (x_{j1}, x_{j2}, \dots, x_{jp})^t$  et  $C = \{C_i; i = 1, \dots, c\}$  un ensemble de prototypes de vecteurs inconnus, où  $C_i$  caractérise la  $i$ ème classe et  $C_i = (c_{i1}, c_{i2}, \dots, c_{ip})^t$ . Dans cette méthode un élément de  $X$  est affecté à une classe et un seul des candidats proposés  $C$ . Dans ce cas, la fonction à minimiser est la suivante :

$$\begin{aligned} J(C, U, X) &= \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m d^2(X_j, C_i) \\ &= \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m \|X_j - C_i\|^2 \end{aligned} \quad (3.2)$$

où  $m$  est un nombre réel supérieur à 1  $X_i$  est le  $i$ -ème élément,  $C_j$  est le center de la classe. Et la matrice de partition floue  $U = (u_{ij})_{c \times N}$  vérifie les conditions suivantes :

$$u_{ik} \in [0, 1], \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \quad (3.3)$$

$$\sum_{i=1}^c u_{ik} = 1, \quad 1 \leq k \leq N \quad (3.4)$$

$$0 < \sum_{k=1}^N u_{ik} < N, \quad 1 \leq i \leq c. \quad (3.5)$$

La minimisation de la fonction k-means (3.2) représente un problème d'optimisation non linéaire qui peut être résolu en utilisant diverses méthodes, notamment la minimisation itérative, le recuit simulé ou des algorithmes génétiques. La méthode la plus populaire est la méthode itération de Picard simple dans les conditions de premier ordre pour les points stationnaires de (3.2), connue sous le nom d'algorithme fuzzy K-means (FKM). Les points stationnaires de la fonction objectif (3.2) peuvent être trouvés en joignant les

contraintes (3.4) à la fonction objective  $J$  au moyen de multiplicateurs de Lagrange :

$$J(C, U, X, \lambda) = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m d_A^2(X_j, C_i) + \sum_{j=1}^N \lambda_j \left( \sum_{i=1}^c u_{ij} - 1 \right). \quad (3.6)$$

où  $\lambda = (\lambda_j)$ ,  $j = 1, \dots, N$ , sont les multiplicateurs de Lagrange pour les  $N$  contraintes (3.4). En utilisant la différentielle de  $J(U, C_1, \dots, C_c, \lambda)$  à toutes les variables, les conditions nécessaires pour que  $J$  atteint son minimum sont

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{(X_j - C_i)^t A (X_j - C_i)}{(X_j - C_k)^t A (X_j - C_k)} \right)^{\frac{1}{m-1}}} \quad (3.7)$$

$$C_i = \frac{\sum_{j=1}^N (u_{ij})^m X_j}{\sum_{j=1}^N (u_{ij})^m} \quad (3.8)$$

Cette itération s'arrêtera quand

$$\|U^{(k+1)} - U^{(k)}\| = \max_{ij} |u_{ij}^{(k+1)} - u_{ij}^{(k)}| < \varepsilon. \quad (3.9)$$

Où  $\varepsilon$  est compris entre 0 et 1.

**Algorithm 2.** Calcul du degré d'appartenance  $u_{ij}$  et les centres de cluster  $C_i$ .

– Initialisez la matrice de partition au hasard, de sorte que  $U^{(0)} = (u_{ij}^{(0)})$  satisfait les conditions (3.3)-(3.4)-(3.5).

– Répéter pour  $l = 1, 2, \dots$

– **Etape 1 :** Calculez les prototypes des clusters (moyennes) :

$$C_i^{(l)} = \frac{\sum_{j=1}^N (u_{ij}^{(l-1)})^m X_j}{\sum_{j=1}^N (u_{ij}^{(l-1)})^m}, \quad 1 \leq i \leq c.$$

– **Etape 2 :** Calculer les distances :

$$d_{ij}^2 = (z_j - c_i)^t A (z_j - c_i) \quad 1 \leq i \leq c, \quad 1 \leq j \leq N.$$

– **Etape 3 :** Mettez à jour la matrice de partition :

pour  $1 \leq j \leq N$

si  $d_{ij} > 0$  pour tout  $i = 1, 2, \dots, c$

$$u_{ij}^{(l)} = \frac{1}{\sum_{k=1}^c \left( \frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}}$$

autrement

$$u_{ij}^{(l)} = 0 \text{ si } d_{ij} > 0, \text{ et } u_{ij}^{(l)} \in [0, 1] \text{ avec } \sum_{k=1}^c u_{kj}^{(l)} = 1$$

– **Jusqu'à**  $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$

**Exemple :** Considérons un ensemble de données synthétiques dans  $R^2$ , qui contient deux groupes bien séparés de formes différentes, comme illustré à la figure (3.1). L'algorithme FKM a été appliquée à cet ensemble de données. L'exposant de pondération sur  $m = 2$  et le critère de terminaison sur 0.01. L'algorithme a été initialisé avec une matrice de partitions aléatoires et convergé après 4 itérations. Dans les courbes de niveau de la figure (3.1), on peut voir que l'algorithme FKM impose une forme circulaire aux deux groupes, même si la grappe inférieure est plutôt allongée.

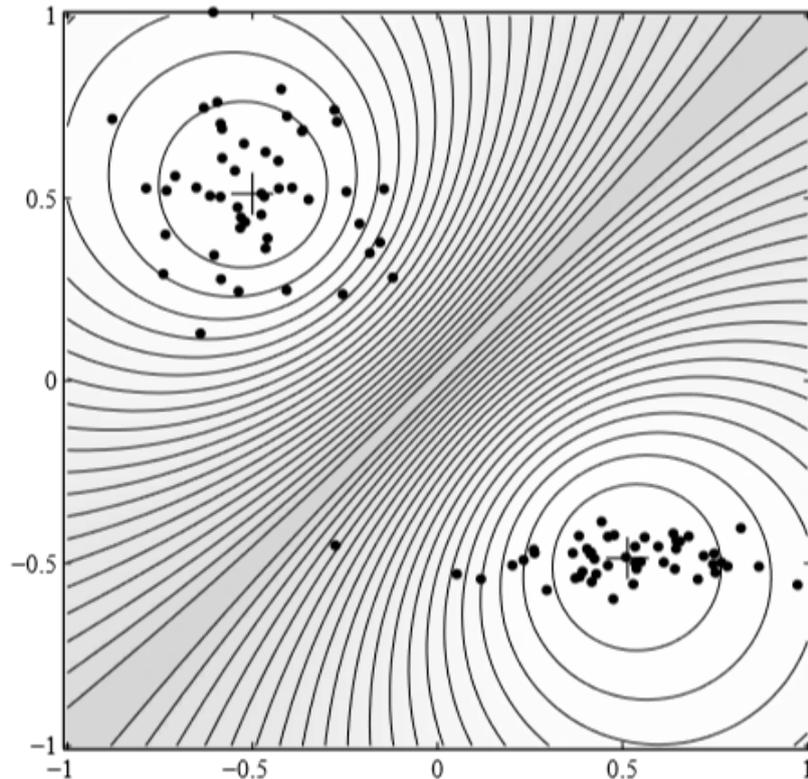


FIGURE 3.1 – Un exemple de l’algorithme fuzzy k-means appliqué à un ensemble de données en dimension 2.

### 3.1.3 Méthode k-moyennes floues possibilistes

Krishnapuram et Keller [137] ont proposé une version possibiliste des k-moyennes appelée k-moyennes floues possibilistes (en anglais possibilistic fuzzy k-means :PFKM). Cette méthode permet d’éviter le problème de contrainte de normalisation posé par la méthode fuzzy k-means FKM en rajoutant un terme dans la fonction objective comme suit :

$$J(C, U, X, \eta) = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m d^2(X_j, C_i) + \sum_{i=1}^c \eta_i \sum_{j=1}^N (1 - u_{ij})^m d^2(X_j, C_i) \quad (3.10)$$

où  $\eta_i$  est un paramètre permettant de contrôler la décroissance de la fonction d’appartenance  $C_i$  et  $m$  est un paramètre permettant de contrôler le degré de flou..

## 3.2 Méthodes de classification supervisée

En bref, la classification supervisée est la tâche qui consiste à discriminer des données, de manière que les objets d’un même groupe (i.e classe) sont plus proches les uns des

### 3.2. MÉTHODES DE CLASSIFICATION SUPÉRVISÉE

autres par rapport à celles des autres groupes. En général, on passe par une phase d'apprentissage dont l'objectif est d'apprendre une règle de classification à partir des données par l'expert (i.e classifieur), afin de déterminer les classes appropriées pour chaque donnée d'apprentissage, ensuite on passe à la phase de prédiction de la classe des nouvelles données qui sera basée sur les classes obtenues en phase d'apprentissage. Comme nous avons énoncé au début, notre objectif est d'évaluer la validité de ces moments obtenus en domaine de reconnaissance de forme, pour cela nous avons opté pour la classification supervisée des images 2D en niveau de gris comme exemple de reconnaissance de forme, les méthodes suivies pour atteindre cet objectif sont : la méthodes de réseaux de neurons (RBF) et (MLP) et la méthode de k-plus proches voisins, afin de mieux éclaircir le fonctionnement du système de classification réalisé dans notre travail, le schéma en figure (3.2) résume ce fonctionnement :

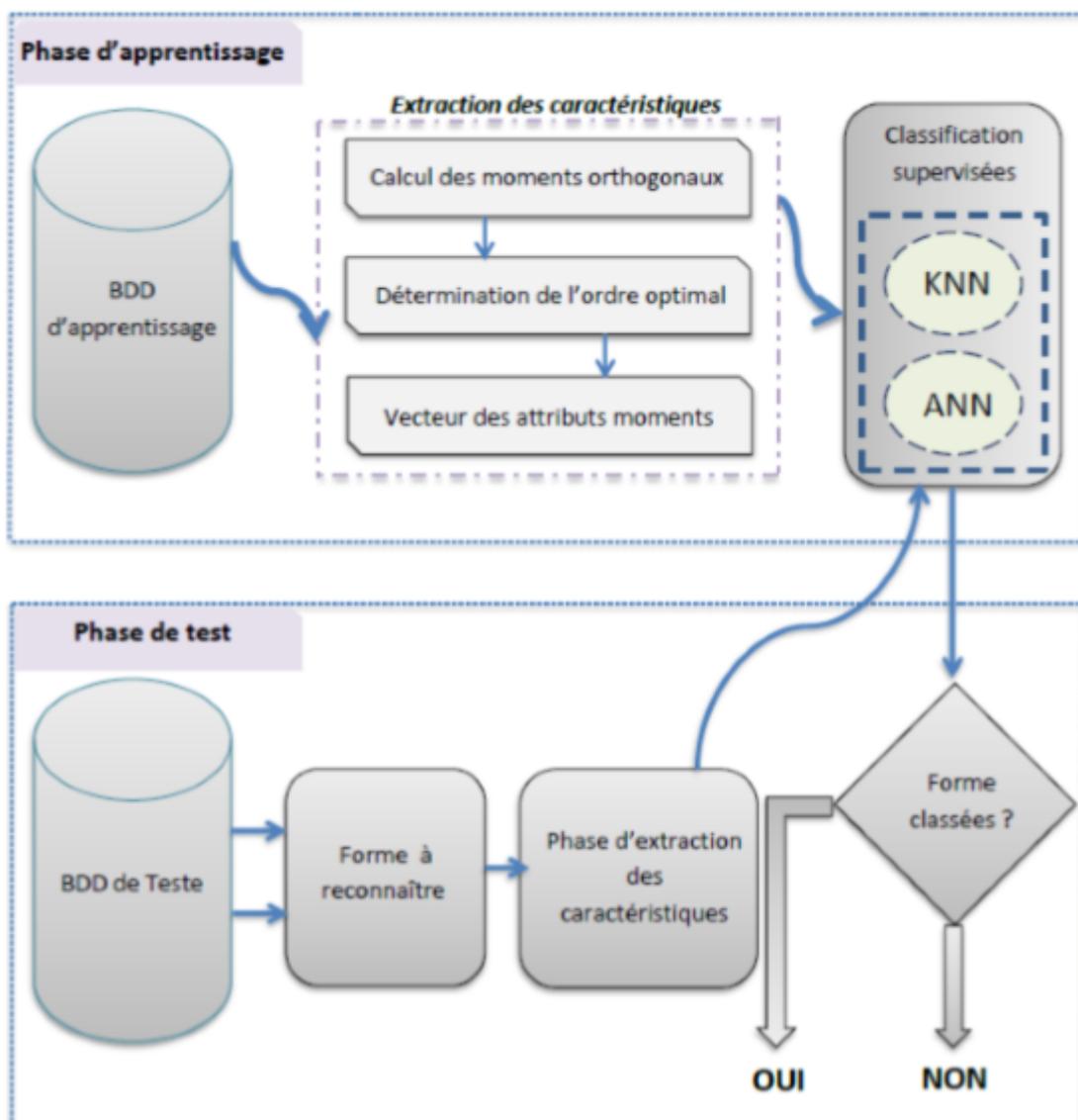


FIGURE 3.2 – Système de classification.

L'architecture de système de classification proposé au-dessus, est composée de deux étapes importantes, à savoir l'extraction des vecteurs caractéristiques à l'aide des approches présentés dans les chapitres précédents et les méthodes de classification. La phase d'extraction des caractéristiques est déjà présentée dans les deux premiers chapitres. Ainsi que les méthodes de classification supervisée seront détaillées dans la section suivante

### 3.2.1 Classification par k-plus proches voisins : k-PPV

L'algorithme k-plus proches voisins (k-PPV) est l'un des algorithmes les plus utilisés dans la classification supervisée et le plus connu dans les algorithmes d'instance. Cet algorithme ne nécessite aucune phase de formation. L'ensemble d'entraînement associé à une fonction de distance et d'une fonction de sélection de la classe en fonction des classes des plus proches voisins, qui constitue le modèle. Cet Algorithme a prouvé son efficacité face au traitement de données textuelles (Yang, 1997). La phase d'apprentissage consiste à stocker les exemples étiquetés. Le classement de nouvelles entrées s'opère en calculant la distance entre la représentation vectorielle de la donnée et celle des exemples du corpus, les k éléments les plus proches sont sélectionnés et l'entrée à classer est assigné à la classe majoritaire. Parmi les avantages de la méthode des k plus proches voisins on cite :

- Facile à comprendre et facile à mettre en oeuvre.
- Plus rapide dans l'entraînement que les méthodes d'apprentissage avides.
- Performant pour les applications dont on trouve qu'un exemple peut avoir plusieurs étiquettes de classe.
- Ils sont sensibles à la sélection de la fonction de similarité qui est utilisée pour comparer des instances

Parmi les inconvénients de la méthode des k-plus proches voisins on cite :

- Sensible à la structure locale des données.
- Ils ont de grands besoins de stockage.

Comme cette méthode est basée sur la notion de plus proche voisin, cette notion sera évaluée en termes de distance. En particulier en  $R^n$ , Minkowski a présenté une distance générale entre deux points de  $R^n$ ,  $P_1 = (x_1, \dots, x_n)$  et  $P_2 = (y_1, \dots, y_n)$ , la distance est donnée par :

$$d(X_1, Y_1) = \sqrt[p]{\sum_{k=1}^n |x_k - y_k|^p}. \quad (3.11)$$

Comme un cas particulier pour  $p = 1$  on obtient la distance de Manhattan, et pour  $p = 2$  on a la distance euclidienne. Dans notre étude la règle de classification admise est la distance euclidienne. L'algorithme de k-PPV revient à implémenter le 1-PPV qui est la base

de K-PPV.

**Algorithme 1-PPV** La méthode de plus proche voisin permet de classer une observation dans la classe d'appartenance de l'observation d'échantillon d'apprentissage qui lui est la plus proche, La détermination de similarité est basée sur la mesure de distance, formellement l'algorithme de 1-plus proches voisins est donné comme suit :

**Algorithm.** Algorithme du plus proches voisins (1-ppv)

- 
- **Input :** Données d'apprentissage  $X^{train} = (x_1^{train}, \dots, x_n^{train})$ , classes des données d'apprentissage  $Z^{train} = (z_1^{train}, \dots, z_n^{train})$ ,  $X^{test} = (x_1^{test}, \dots, x_m^{test})$ ,
  - **Output**  $Z^{test} = (z_1^{test}, \dots, z_n^{test})$ ,
  - for  $i \leftarrow 1$  to  $m$  do
    - for  $j \leftarrow 1$  to  $n$  do
    - calculer la distance entre  $x_i^{test}$  et  $x_j^{train}$
    - $d_j \leftarrow d(x_j^{train}, x_i^{test})$
    - Calculer la classe  $z_i^{test}$  du ième exemple qui vaut la classe de son ppv :
    - trouvez l'indice de ppv de  $x_i^{test}$  :
    - $ind - ppv_i \leftarrow argmin_{j=1}^n d_j$
    - trouvez la classe de ppv de  $x_i^{test}$  (qui est  $x_{ind-ppv_i}^{test}$  )
    - $z_i^{test} = x_{ind-ppv_i}^{train}$
  - **Return** classe des données de test  $Z^{test} = (z_1^{test}, \dots, z_n^{test})$ .

**Algorithme k-PPV** Cette méthode est largement utilisée en classification, c'est une extension d'algorithme 1-ppv. La plus proche observation n'est plus la seule utilisée pour la classification. Nous utilisons les k plus proches observations. Ainsi la décision est en faveur de la classe majoritaire représentée par les k voisins. Soit  $k_r$ , le nombre d'observation issu de groupe des plus proches voisins appartenant à la classe  $r$  :

$$\sum_{r=1}^c k_r = k. \quad (3.12)$$

Ainsi une nouvelle observation est prédite dans la classe résultante est obtenue par :

$$l = max_r(k_r) \quad (3.13)$$

Le choix de paramètre K réside le problème majeur en algorithme KNN, pour notre cas le choix est réalisé par la variation de  $k$  jusqu'à l'obtention d'erreur minimal. Ceci évite que la classe prédite ne soit déterminée qu'à partir d'une seule observation, donc le degré

de prédiction est obtenu par le paramètre  $k$ . L'algorithme qui implémente cette méthode est donné comme suit :

**Algorithm.** Algorithme des  $k$ -plus proches voisins (k-PPV)

- 
- **Input** : Données d'apprentissage  $X^{train} = (x_1^{train}, \dots, x_n^{train})$ , classes des données d'apprentissage  $Z^{train} = (z_1^{train}, \dots, z_n^{train})$ ,  $X^{test} = (x_1^{test}, \dots, x_m^{test})$ , nombre de ppv  $k$
  - **Output**  $Z^{test} = (z_1^{test}, \dots, z_n^{test})$ ,
  - for  $i \leftarrow 1$  to  $m$  do
    - for  $j \leftarrow 1$  to  $n$  do
    - Calculer la distance entre  $x_i^{test}$  et  $x_j^{train}$
    - $d_j \leftarrow d(x_j^{train}, x_i^{test})$
    - Calculer la classe  $z_i^{test}$  du ième exemple qui vaut la classe de son ppv :
    - // trouvez les  $k$ -ppv de  $x_i^{test}$  //
    - Trier les distances  $d_j$  selon un ordre croissant pour  $j = 1, \dots, n$
    - Récupérer en même temps les indices  $IndVoisins$  avant le tri des  $d_j$
    - Récupérer les classes des  $K$  premiers ppv à partir des indices  $IndVoisins$  et trouvez la classe majoritaire.
    - $C_k \leftarrow 0 (k = 1, \dots, n)$
    - for  $j \leftarrow 1$  to  $n$  do
    - $ind - voisin_k \leftarrow IndVoisins_k$
    - $h \leftarrow z_n^{ind-voisin_k}$
    - $C_h \leftarrow C_{h+1}$
    - // trouvez la classe du ppv de  $x_i^{test}$  //
    - (la classe majoritaire de ses  $k$ -ppv)
    - $z_i^{test} = argmax_{k=1}^k C_k$
  - **Return** classe des données de test  $Z^{test} = (z_1^{test}, \dots, z_n^{test})$ .

### 3.2.2 Classification par réseaux de neurones à fonctions radiales : RBF

#### Architecture de RBF

Introduit par Powell et Broomhead, le réseau RBF (Radial Basis Functions) fait partie des réseaux de neurones supervisés. Il est constitué de trois couches (figure 3.3) : une couche d'entrée qui retransmet les entrées sans distorsion, une seule couche cachée qui contient les neurones RBF qui sont généralement des gaussiennes et une couche de sortie

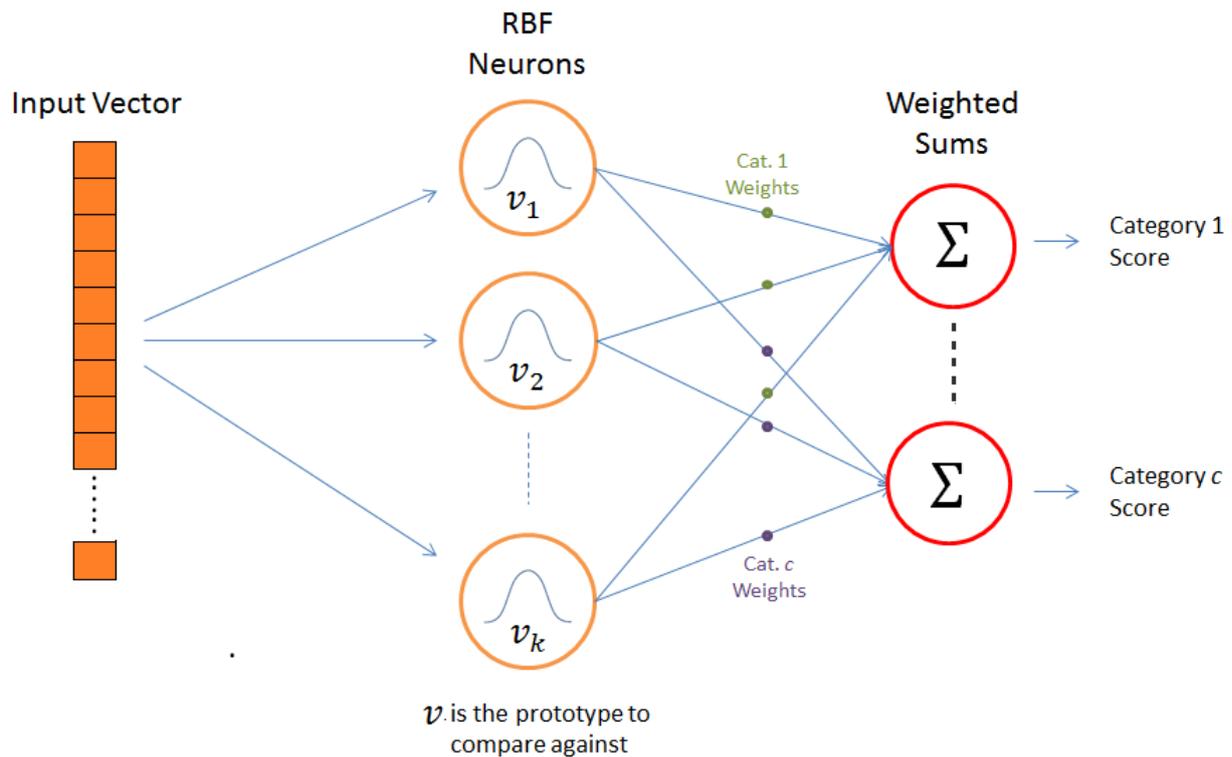


FIGURE 3.3 – Présentation schématique d'un réseau RBF

dont les neurones sont généralement animés par une fonction d'activation linéaire. Chaque couche est complètement connectée à la suivante et il n'y a pas de connexions à l'intérieur d'une même couche. Ce réseau est constitué de  $N$  neurones d'entrée,  $K$  neurones cachés et  $c$  neurones de sortie. La sortie du  $m$ -ème neurone de la couche cachée est donnée par :

$$y_m^{(q)} = \exp\left(-\frac{\|x^{(q)} - v_m\|^2}{2\sigma_m^2}\right), \quad (3.14)$$

où  $x^{(q)} = \left( MI_{nm} \right)_{n,m=0,\dots,p}$  sont les vecteurs descripteurs des images de la base de données considérée,  $v_m$  est le centre du  $m$ -ème neurone de la couche cachée ou du  $m$ -ème neurone Gaussien et  $\sigma_m$  est la largeur du  $m$ -ième gaussienne. La sortie du  $j$ -ième neurone de la couche de sortie est donnée par :

$$z_j^q = \frac{1}{K} \left( \sum_{m=0}^K w_{mj} y_m^{(q)} \right), \quad (3.15)$$

où  $m = 1, \dots, K$ ;  $j = 1, \dots, c$  et  $w_{mj}$  sont les poids reliant la couche cachée à celle de la sortie.

### Algorithme d'apprentissage du réseau RBF

L'apprentissage du réseau RBF a été présenté la première fois par Moody et Darken. Il consiste à régler quatre paramètres principaux : le nombre de neurones dans l'unique couche cachée ou le nombre des gaussiennes, la position des centres de ces gaussiennes, la largeur de ces gaussiennes et les poids des connexions entre les neurones cachés et le(s) neurone(s) de sortie. Le réseau RBF consiste à minimiser l'erreur quadratique totale  $E$  calculée entre les sorties obtenues du réseau et celles désirées

$$E = \sum_{q=1}^M \sum_{m=0}^K (t_j^{(q)} - z_j^{(q)})^2, \quad (3.16)$$

Pour le réseau RBF, l'ajustement des poids  $w_{mj}$  reliant la couche cachée à celle de la sortie est réalisé par la règle de Widrow-Hoff. Il se fait comme suit :

$$w_{mj}^{(i+1)} = w_{mj}^{(i)} + \eta(t_j - z_j)y_m, \quad (3.17)$$

$t_j$  est la sortie du j-ième neurone désirée,  $z_j$  est la sortie du j-ième neurone calculée,  $y_m$  est la sortie du m-ième neurone de la couche cachée et  $\eta$  est le pas d'apprentissage dont la valeur est comprise entre 0 et 1. Dans les tâches de classification, le nombre et les centres des gaussiennes sont choisis par des techniques de regroupement de données de type k-means. Les largeurs des gaussiennes sont calculées par la valeur moyenne des distances séparant tous les exemples au centre correspondant. Les k-means sont des méthodes itératives permettant de séparer une série de vecteurs en différents clusters et chaque cluster est représenté par un centre.

### 3.2.3 Classification par réseaux de neurones perceptron multi-couche : MLP

#### Architecture de MLP

Un MLP (Multi Layer Perceptron) est composé d'une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées. Les éléments de deux couches adjacentes sont interconnectés par des poids assurant la liaison des différentes couches comme indiqué sur la figure 2.  $w$  représente les poids reliant la couche d'entrée avec la couche cachée ;  $u_{mj}$  représente les poids reliant la couche cachée avec celle de la sortie. Les composants du vecteur d'entrée  $X$  du MLP seront pondérés par les poids  $w_{nm}$  et ensuite transmis à la

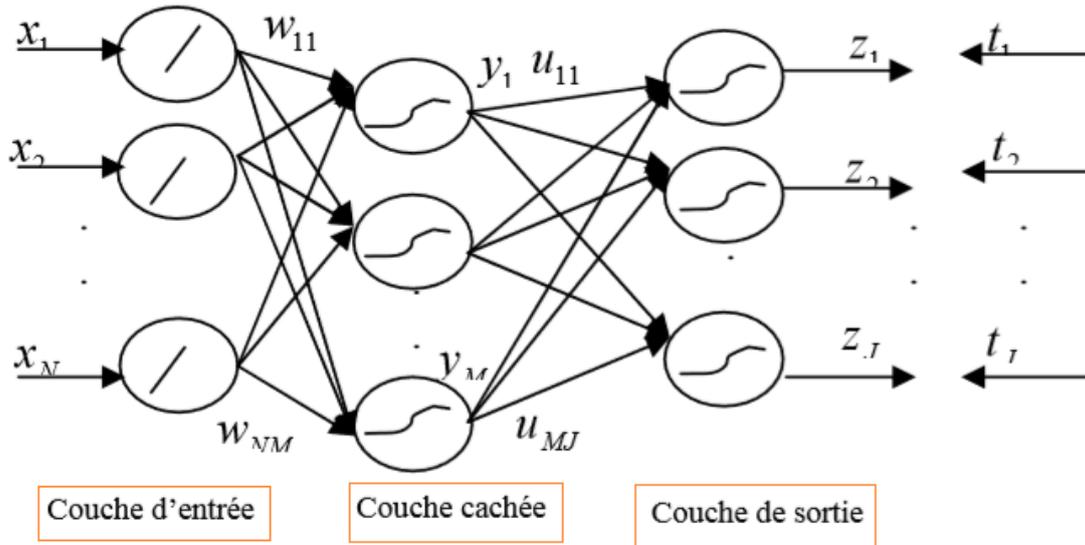


FIGURE 3.4 – Présentation schématique d'un réseau MLP

couche cachée selon les équations suivantes :

$$c_m = \sum_{n=1}^N x_n w_{nm} \quad (3.18)$$

$$y_m = f(c_m) \quad (3.19)$$

Les sorties de la couche cachée seront aussi pondérées par les poids  $u_{mj}$  et transmis ensuite à la couche de sortie selon les équations suivantes :

$$o_j = \sum_{m=1}^M y_m u_{mj} \quad (3.20)$$

$$t_j = g(o_j) \quad (3.21)$$

$f$  et  $g$  représentent respectivement les fonctions d'activations des neurones de la couche cachée et celle de la de sortie. Dans ce cas  $f$  est la fonction sigmoïde définie par

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (3.22)$$

Sa dérivée est :

$$f'(x) = \alpha \cdot f(x) [1 - f(x)] \quad (3.23)$$

### Algorithme d'apprentissage du réseau MLP

L'apprentissage consiste à adapter les valeurs des poids afin d'obtenir le comportement désiré du MLP. Pour cela, on se crée une base d'exemples où chaque exemple est constitué d'un vecteur d'entrée et d'un vecteur de sortie approprié. Un algorithme d'apprentissage connu sous le nom d'algorithme de rétropropagation du gradient a été mis en place en 1985 par Rumelhart et al. [6]. Ainsi l'adaptation des poids  $u_{mj}$  qui relie la couche cachée à celle de la sortie est comme suite :

$$u_{mj}^{(i+1)} = u_{mj}^{(i)} + \eta_1(t_j - z_j)g'(o_j)y_m \quad (3.24)$$

La mise à jour des poids  $w_{nm}$  reliant la couche d'entrée à la couche cachée est comme suite :

$$w_{nm}^{(i+1)} = w_{nm}^{(i)} + \eta_2 \left[ \sum_{j=1}^J (t_j - z_j)g'(o_j)u_{mj}f'(c_m)x_n \right] \quad (3.25)$$

$\eta_1$  et  $\eta_2$  sont les pas d'apprentissage.

### 3.3 Résultats expérimentaux sur la classification

Dans cette section, nous allons fournir des expériences pour valider la précision de la classification des objets en utilisant l'ensemble proposé des vecteurs descripteurs et les méthodes de classification présentées dans les sections précédentes. Pour cela, nous allons utiliser les vecteurs descripteurs suivants

$$V(f) = \left( MI_{ij}(D(f)) \right)_{i,j=0,\dots,p} \quad (3.26)$$

Où  $D(f)$  est l'image débruitée à l'aide du modèle de Perona-Malik, et  $MI_{ij}(D(f))$  sont les moments orthogonaux invariants utilisés de l'image  $D(f)$ . Nous allons utiliser les méthodes de classification par réseaux de neurones PLM et RBF et les méthodes de k-plus proche voisins (K-PPV), (les méthode de k-moyennes (KM) et de k-moyennes floue (FKM) basés sur des distances simples. Le choix de la distance est primordial au bon fonctionnement de la méthode. Il est à noter qu'une distance doit avoir quatre propriétés [Mukundan -

Ramakrishnan, 1998]. Pour tout vecteur  $a$ ,  $b$  et  $c$  :

$$d(a, b) \geq 0 \text{ (} d \text{ est positive)} \quad (3.27)$$

$$d(a, b) = 0 \text{ si et seulement si } a = b. \quad (3.28)$$

$$d(a, b) + d(b, c) \geq d(a, c) \quad (3.29)$$

$$d(a, b) = d(b, a) \quad (3.30)$$

Le choix de la distance s'effectue en fonction des connaissances préalables du problème. Il est possible de choisir la distance en faisant varier cette distance et, pour chacun des choix, estimer l'erreur réelle. On choisit alors la distance donnant la meilleure erreur réelle estimée. Dans la littérature, il existe plusieurs types de distances à savoir : la distance Euclidienne, la distance de grandeur logarithmique, la distance de coefficient de corrélation, la distance « city bloc » ou "Manhattan" (somme des valeurs absolues), la distance de Tchebycheff, la distance de Mahalanobis, etc. Dans notre cas, nous avons utilisé trois distances : la distance Euclidienne (la distance la plus populaire), la distance de grandeur logarithmique et la distance de coefficient de corrélation. Ces distances sont définies par les relations suivantes [Mukundan -Ramakrishnan, 1998] :

- La distance Euclidienne

$$d(X, Y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \quad (3.31)$$

- La distance de grandeur logarithmique

$$d(X, Y) = \sqrt{\sum_{i=1}^p (\log(|x_i|) - \log(|y_i|))^2} \quad (3.32)$$

- La distance de coefficient de corrélation

$$d(X, Y) = \sqrt{{}^t X A Y} = \sqrt{\sum_{i=1}^p a_{ij} (x_i - y_i)(x_j - y_j)} \quad (3.33)$$

avec  $A = (a_{ij})$  est une matrice symétrique définie et positive.

Les formules ci-dessus mesurent la distance entre deux vecteurs  $X = (x_1, \dots, x_p)$  et  $Y = (y_1, \dots, y_p)$ , qui sont définis dans l'espace  $R^p$ . Ces distances sont rapides à calculer et facile à implémenter. L'idée sous-jacente de l'utilisation de ces distances est que chaque image à  $p$  caractéristiques est un point dans un espace de dimension  $p$ . Chaque caractéristique est

un vecteur de cet espace. La distance entre deux images est la distance entre deux points dans cet espace. Si les deux vecteurs  $X$  et  $Y$  sont égaux, alors la distance tend vers 0. Par conséquent, pour classer les images par les méthodes K-PPV, KM, FKM , on prend les valeurs minimales pour la distance  $d$ .

La précision de la reconnaissance est définie comme étant :

$$\eta = \frac{A}{B} \times 100\%. \quad (3.34)$$

Où  $A$  est le nombre d'images bien classées et  $B$  est le nombre d'images utilisées dans le test.

#### 3.3.1 Bases de données d'images utilisées

Pour tester la performance des systèmes de classification proposés nous avons utilisé sept bases de données d'images bien connues. Ces bases de données sont :

**La base MPEG-7 CE-shape-1 Part B :** est une base de données d'images qui contient 72 classes et chaque classe contient 20 images binaires de taille  $128 \times 128$  pixels. Un aperçu de cette base est représenté dans la figure 3.5.

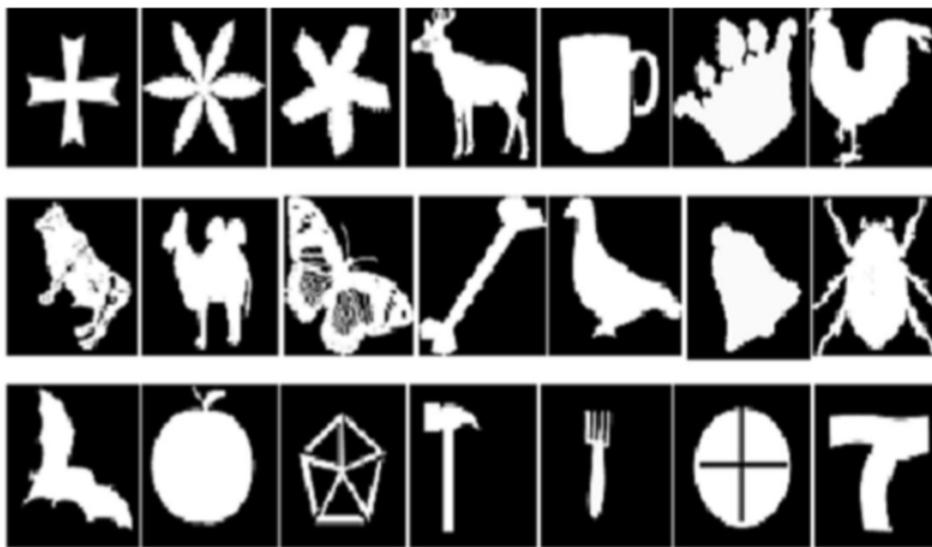


FIGURE 3.5 – Echantillons des images de la base d'images MPEG-7

**La base d'images COLUMBIA :** est une base de données contient des images tirées d'objets 3D pris sous différentes positions [COIL-20, 2014]. Le nombre total d'images est 1440 répartis comme 72 images pour chaque objet.

**La base de données MNIST handwritten digit :**



FIGURE 3.6 – Echantillons des 20 objets de la base d’images COIL-20 de COLUMBIA

La base de données de MNIST handwritten digit, disponible à partir de <http://yann.lecun.com/exdb/mnist/>, dispose d’un ensemble de d’apprentissage de 60 000 chiffres manuscrits, et un ensemble de 10 000 exemples de test. Il s’agit d’un sous-ensemble d’un ensemble plus vaste disponible à partir du NIST. Les chiffres sont de taille normalisée et centrée sur une image de taille fixe. C’est une bonne base de données pour les personnes qui veulent essayer des techniques d’apprentissage et les méthodes de reconnaissance des formes sur des données du monde réel tout en dépensant un minimum d’efforts sur le prétraitement. Cette base de données est celle qui sera admise dans notre évaluation de classification basée sur la notion des moments discrets. La figures (3.7) présente des exemples d’apprentissage et de test, pris de BD-Mnist.

**La base ImageNet :** est une base de données d’images annotées produit par l’organisation du même nom, à destination des travaux de recherche en vision par ordinateur. En 2016, plus de dix millions d’URLs ont été annotées à la main pour indiquer quels objets sont représentés dans l’image, plus d’un million d’images bénéficient en plus de boîtes englobantes autour des objets. La base de données d’annotations sur des URL d’images tierces est disponible librement, ImageNet ne possédant cependant pas les images elles-mêmes. Depuis 2010, le projet ImageNet organise un concours annuel : ImageNet Large Scale Visual Recognition Challenge (ILSVRC), ou ”Compétition ImageNet de Reconnaissance Visuelle à Grande Échelle”. Elle consiste en une compétition logicielle dont le but est de détecter et classifier précisément des objets et des scènes dans les images naturelles.

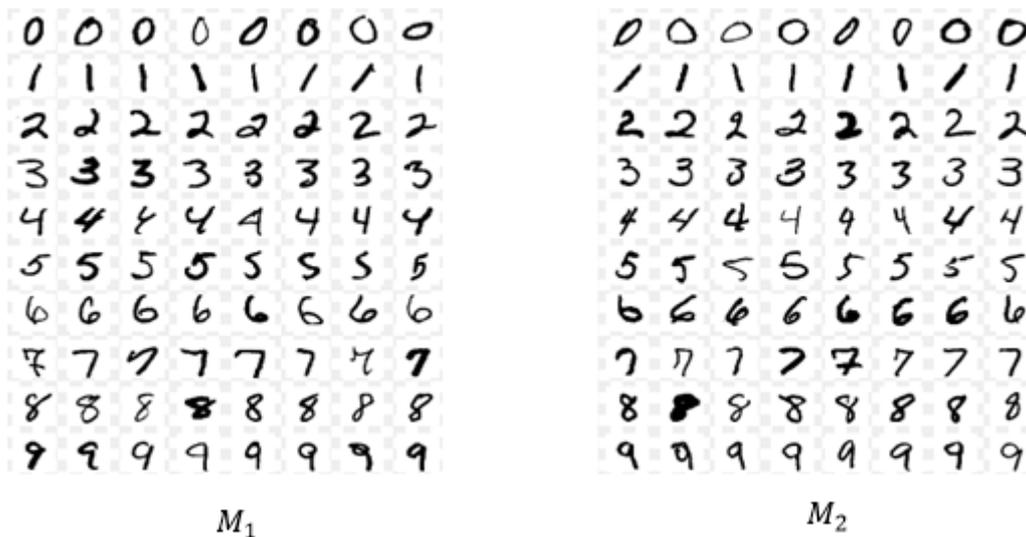


FIGURE 3.7 – Echantillons des chiffres de l'ensemble d'apprentissage ( $M_1$ ) et l'ensemble de test ( $M_2$ ).

Cette base de données contient plus de 15 millions d'images appartenant à environ 22 000 catégories. Il existe environ 1,2 million d'images d'apprentissage, 50 000 images de validation et 150 000 images de test. ImageNet consiste en des images à résolution variable. Par conséquent, les images ont été sous-échantillonnées à une résolution fixe de  $256 \times 256$ . Dans le cas d'une image rectangulaire, l'image est redimensionnée et découpée du patch central  $256 \times 256$  de l'image résultante. La figure 3.8 présente un exemple des images de cette base de données.

**La base Columbia Object Image Library COIL-100 :** est une base de données d'images couleur de 100 objets. Les objets ont été placés sur une platine motorisée sur un fond noir. Le plateau tournant a été pivoté de 360 degrés pour faire varier la pose de l'objet par rapport à une caméra couleur fixe. Les images des objets ont été prises à des intervalles de pose de 5 degrés . Cela correspond à 72 poses par objet. Les images ont été normalisées en taille. COIL-100 est disponible en ligne via : [http : //www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php](http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php). La figure (3.9) montre les 100 objets de cette base de données.

### 3.3.2 Classifications des bases de données d'images

Afin de valider la précision de la reconnaissance et la classification des objets en utilisant les moments invariants proposés et les différentes méthode de classification présentées dans la section précédentes, nous allons utiliser les sept bases de bases de données, qui sont

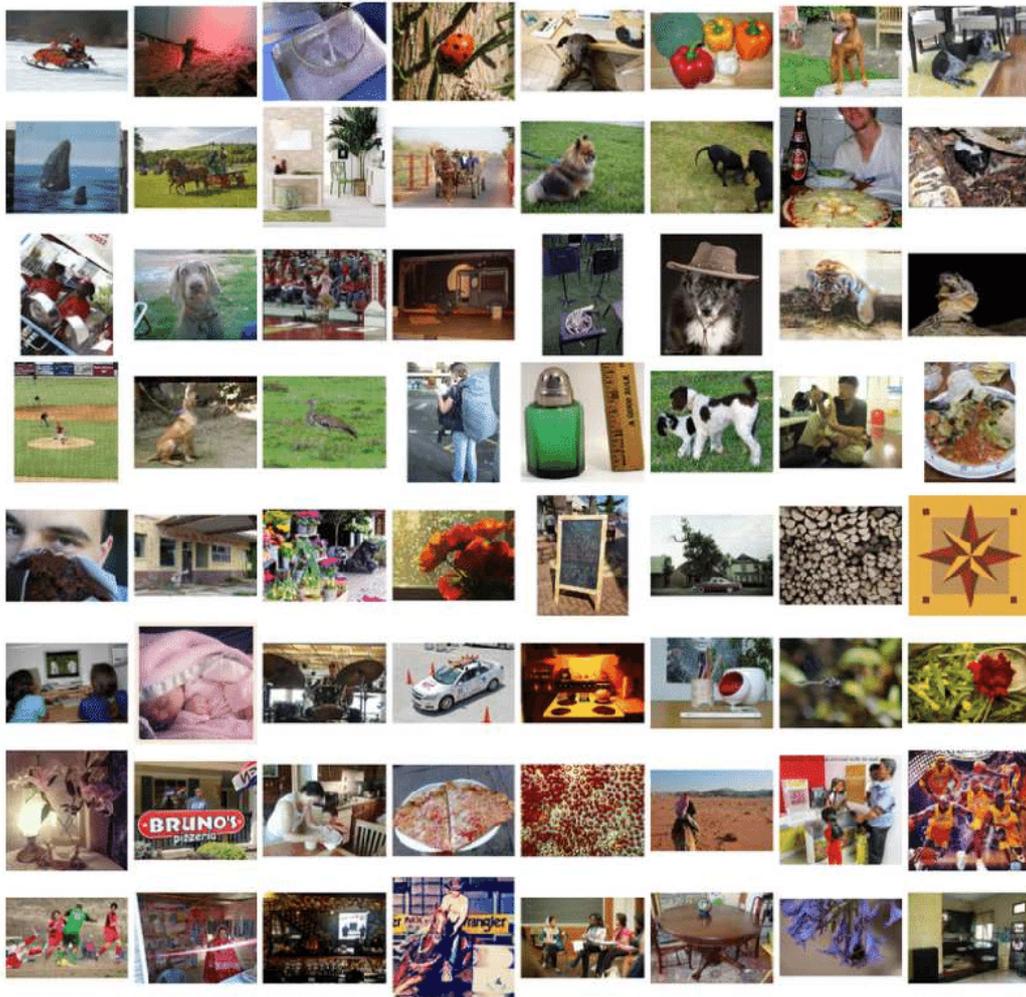


FIGURE 3.8 – Un exemple d’images dans la base de données ImageNet

disponibles gratuitement sur Internet. Chaque base d’images contient plusieurs classes, où chaque image appartient à une classe. Le test de validation de la précision de classification des images est effectué sur les des les sept bases en utilisant les moments invariants de Legendre adaptés et les moments invariants (OIM). Les résultats de classification obtenus sont comparés avec d’autres méthodes bien connues comme Hu [1], Legendre classique [11], Tchebichef-Krawtchouk (TKM), Tchebichef-Hahn (THM) et Krawtchouk-Hahn (KHM)[142]. Le test de classification est suivi ensuite par l’addition d’un bruit « sel et poivre » de différentes densités. Les résultats de classification sont présentés dans les tableaux (3.1-3.5). Ces résultats montrent l’efficacité des approches proposées en termes de précision de la reconnaissance des images bruitées et non bruitées par rapport à ceux de Hu, de Legendre, de TKM, de THM et de KHM.

### 3.3. RÉSULTATS EXPÉRIMENTAUX SUR LA CLASSIFICATION

TABLE 3.1 – Résultats de classification par la méthode de réseaux de neurons RBF

Database	Invariant moments	Noise free	1 (%)	2(%)	3(%)	4(%)
MPEG7-CE database	Hu	97.65%	92.49%	86.89%	76.73%	75.14%
	CL-inv	98.71%	93.19%	90.11%	79.12%	75.22%
	Kr-inv	99.63%	94.07%	90.77%	81.08%	76.33%
	Tc-inv	99.81%	94.43%	91.85%	83.12%	78.42%
	OALM	100%	96.84%	94.73%	91.54%	89.11%
	THIM	98,15%	93.11%	92.45%	91.08%	88.17%
	KHIM	98,14%	93.06%	92.02%	91.01%	88.13%
	OIM(1)	100%	98.14%	97.45%	95.17%	93.79%
	OIM(2)	100%	98.21%	97.52%	95.19%	93.83%
OIM(3)	100%	98.45%	97.63%	95.31%	93.94%	
COIL-20 database	Hu	97.54%	92.38%	86.78%	76.62%	75.03%
	CL-inv	98.60%	93.08%	90.01%	79.02%	75.11%
	Kr-inv	99.52%	93.97%	90.66%	81.02%	76.22%
	Tc-inv	99.70%	94.32%	91.74%	83.01%	78.31%
	OALM	100%	96.73%	94.62%	91.43%	89.03%
	THIM	98,04%	93.09%	92.33%	90.95%	88.06%
	KHIM	98,03%	93.01%	91.69%	90.88%	88.02%
	OIM(1)	100%	98.13%	97.04%	95.06%	93.68%
	OIM(2)	100%	98.10%	97.41%	95.08%	93.72%
OIM(3)	100%	98.34%	97.52%	95.20%	93.83%	
MNIST database	Hu	96.43%	91.27%	85.67%	75.51%	74.01%
	CL-inv	97.59%	92.07%	89.08%	78.56%	74.28%
	Kr-inv	98.51%	92.58%	89.85%	80.58%	75.33%
	Tc-inv	98.69%	93.31%	90.76%	82.03%	77.33%
	OALM	99.97%	95.87%	93.72%	90.45%	88.77%
	THIM	97.62%	92.15%	91.45%	89.97%	87.58%
	KHIM	97.17%	92.14%	91.40%	90.05%	87.42%
	OIM(1)	99.94%	97.88%	96.75%	94.78%	92.95%
	OIM(2)	99.96%	97.91%	96.78%	94.81%	92.98%
OIM(3)	100%	97.94%	96.82%	94.83%	92.98%	
ImageNet database	Hu	96.51%	91.44%	85.58%	75.58%	74.43%
	CL-inv	97.63%	92.22%	89.38%	78.21%	74.14%
	Kr-inv	98.42%	92.63%	89.93%	80.62%	75.47%
	Tc-inv	98.75%	93.45%	90.82%	82.14%	77.41%
	OALM	99.92%	95.82%	93.68%	90.40%	88.72%
	THIM	97.55%	92.10%	91.40%	89.92%	87.53%
	KHIM	97.35%	92.13%	91.47%	90.12%	87.47%
	OIM(1)	99.47%	97.55%	96.47%	94.58%	92.44%
	OIM(2)	99.49%	97.73%	96.51%	94.62%	93.02%
OIM(3)	99.63%	97.81%	96.65%	94.78%	93.18%	

TABLE 3.2 – Résultats de classification par la méthode de réseaux de neurons MLP

Database	Invariant moments	Noise free	1 (%)	2(%)	3(%)	4(%)
MPEG7-CE database	Hu	97.43%	92.37%	86.67%	76.51%	74.98%
	CL-inv	98.63%	92.21%	89.13%	78.14%	74.26%
	Kr-inv	98.65%	93.09%	89.79%	80.10%	75.35%
	Tc-inv	98.83%	93.45%	90.87%	82.14%	77.44%
	OALM	99.92%	95.86%	93.76%	90.56%	88.13%
	THIM	97.17%	92.13%	91.47%	90.11%	87.19%
	KHIM	97.16%	92.08%	91.04%	90.03%	87.15%
	OIM(1)	99.95%	97.16%	95.43%	94.19%	92.81%
	OIM(2)	99.98%	97.23%	96.54%	94.21%	92.85%
	OIM(3)	100%	97.47%	96.65%	94.33%	92.96%
COIL-20 database	Hu	96.57%	91.40%	85.81%	75.64%	74.05%
	CL-inv	97.62%	92.11%	89.03%	78.04%	74.13%
	Kr-inv	98.54%	92.99%	89.68%	80.04%	75.24%
	Tc-inv	98.72%	93.34%	90.76%	82.03%	77.33%
	OALM	99.16%	95.75%	93.64%	90.45%	88.05%
	THIM	98.07%	93.12%	92.36%	90.98%	88.09%
	KHIM	98,03%	93.01%	91.69%	90.88%	88.02%
	OIM(1)	99.91%	97.15%	96.06%	94.08%	92.70%
	OIM(2)	99.96%	97.12%	96.43%	94.10%	92.74%
	OIM(3)	100%	97.36%	96.54%	94.22%	92.85%
MNIST database	Hu	96.45%	91.29%	85.70%	75.53%	74.04%
	CL-inv	97.61%	92.10%	89.12%	78.59%	74.30%
	Kr-inv	98.51%	92.58%	89.85%	80.58%	75.33%
	Tc-inv	98.71%	93.34%	90.78%	82.06%	77.35%
	OALM	99.98%	95.89%	93.75%	90.49%	88.78%
	THIM	97,62%	92.15%	91.45%	89.97%	87.58%
	KHIM	97,19%	92.17%	91.43%	90.07%	87.43%
	OIM(1)	99.93%	97.90%	96.76%	94.81%	92.97%
	OIM(2)	99.97%	97.93%	96.79%	94.85%	93.02%
	OIM(3)	100%	97.96%	96.81%	94.87%	93.18%
ImageNet database	Hu	96.56%	91.49%	85.62%	75.61%	74.46%
	CL-inv	97.65%	92.23%	89.42%	78.26%	74.17%
	Kr-inv	98.44%	92.67%	89.97%	80.69%	75.52%
	Tc-inv	98.76%	93.47%	90.85%	82.18%	77.46%
	OALM	99.96%	95.86%	93.69%	90.45%	88.77%
	THIM	97.57%	92.13%	91.44%	89.96%	87.57%
	KHIM	97.41%	92.18%	91.53%	90.17%	87.54%
	OIM(1)	99.48%	97.57%	96.49%	94.60%	92.54%
	OIM(2)	99.53%	97.69%	96.58%	94.68%	93.25%
	OIM(3)	99.69%	97.88%	96.67%	94.72%	93.58%

TABLE 3.3 – Résultats de classification par la méthode K-PPV

Database	moments invariants	0 (%)	1 (%)	2(%)	3(%)	4(%)
MPEG7-CE database	Hu	96.12%	92.06%	86.36%	76.40%	74.67%
	CL-inv	98.32%	92.02%	89.04%	78.24%	74.05%
	Kr-inv	98.35%	93.11%	89.48%	80.00%	75.04%
	Tc-inv	98.52%	93.14%	90.56%	82.07%	77.13%
	OALM	99.81%	95.75%	93.45%	90.27%	88.17%
	THIM	97.21%	92.44%	91.48%	90.17%	87.25%
	KHIM	97.01%	92.01%	90.63%	89.85%	87.03%
	OIM(1)	99.64%	97.11%	95.12%	94.02%	92.73%
	OIM(2)	99.65%	97.01%	96.23%	94.07%	92.52%
	OIM(3)	99.77%	97.14%	96.34%	94.21%	92.85%
COIL-20 database	Hu	96.42%	91.32%	85.51%	75.15%	73.23%
	CL-inv	97.15%	92.00%	88.63%	77.87%	74.01%
	Kr-inv	98.23%	92.66%	89.35%	79.89%	75.01%
	Tc-inv	98.48%	93.08%	90.43%	82.00	74.87%
	OALM	99.01%	95.44%	93.32%	90.14%	88.00%
	THIM	97.87%	93.05%	92.09%	90.68%	87.99%
	KHIM	97,73%	92.91%	91.36%	90.58%	87.79%
	OIM(1)	99.68%	97.02%	95.76%	93.98%	92.40%
	OIM(2)	99.63%	97.01%	96.14%	93.92%	92.47%
	OIM(3)	99.82%	97.08%	96.25%	94.08%	92.58%
MNIST database	Hu	96.12%	91.02%	85.47%	75.14%	73.84%
	CL-inv	97.35%	91.90%	89.02%	78.25%	74.01%
	Kr-inv	98.51%	92.58%	89.85%	80.58%	75.33%
	Tc-inv	98.42%	93.01%	90.45%	81.76%	77.05%
	OALM	99.28%	95.59%	93.45%	90.19%	88.48%
	THIM	97.32%	92.05%	91.15%	89.68%	87.24%
	KHIM	97.01%	92.00%	91.18%	89.87%	87.15%
	OIM(1)	99.61%	97.60%	96.46%	94.51%	92.67%
	OIM(2)	99.67%	97.63%	96.49%	94.55%	93.11%
	OIM(3)	99.88%	97.66%	96.51%	94.57%	93.19%
ImageNet database	Hu	96.26%	91.14%	85.35%	75.35%	74.17%
	CL-inv	97.38%	92.01%	89.11%	78.01%	74.00%
	Kr-inv	98.15%	92.38%	89.68%	80.39%	75.22%
	Tc-inv	98.45%	93.17%	90.55%	82.08%	77.18%
	OALM	99.65%	95.58%	93.32%	90.14%	88.47%
	THIM	97.24%	92.07%	91.15%	89.65%	87.28%
	KHIM	97.17%	92.00%	91.28%	90.01%	87.29%
	OIM(1)	99.21%	97.25%	96.17%	94.35%	92.21%
	OIM(2)	99.32%	97.35%	96.28%	94.38%	93.01%
	OIM(3)	99.39%	97.58%	96.37%	94.42%	93.28%

### 3.3. RÉSULTATS EXPÉRIMENTAUX SUR LA CLASSIFICATION

TABLE 3.4 – Résultats de classification par la méthode k-means

Database	Moments invariants	0 (%)	1 (%)	2(%)	3(%)	4(%)
MPEG7-CE database	Hu	94.53%	91.57%	85.81%	75.52%	73.58%
	CL-inv	97.83%	91.53%	88.55%	77.75%	73.56%
	Kr-inv	97.56%	91.62%	88.99%	79.55%	74.56%
	Tc-inv	98.52%	93.14%	90.56%	82.07%	77.13%
	OALM	98.89%	94.91%	92.96%	89.78%	87.68%
	THIM	96.72%	91.65%	90.97%	89.68%	86.76%
	KHIM	96.53%	91.55%	89.97%	88.09%	86.17%
	OIM(1)	98.25%	96.14%	94.57%	93.19%	91.24%
	OIM(2)	98.29%	96.18%	94.68%	93.28%	91.33%
	OIM(3)	98.45%	96.31%	94.78%	93.43%	91.56%
COIL-20 database	Hu	94.72%	90.62%	84.77%	74.87%	72.74%
	CL-inv	96.20%	91.31%	87.57%	76.51%	73.14%
	Kr-inv	97.35%	91.87%	88.42%	78.91%	74.57%
	Tc-inv	97.54%	92.18%	89.84%	81.12	73.76%
	OALM	98.41%	94.25%	92.24%	89.42%	87.12%
	THIM	96.72%	92.24%	91.35%	89.57%	86.57%
	KHIM	96.74%	91.95%	90.40%	89.65%	86.77%
	OIM(1)	97.14%	95.25%	93.78%	91.78%	90.48%
	OIM(2)	97.25%	95.33%	93.85%	91.94%	90.75%
	OIM(3)	97.45%	95.47%	93.91%	91.98%	91.14%
MNIST database	Hu	94.10%	89.09%	83.49%	73.16%	71.94%
	CL-inv	95.31%	89.47%	87.14%	76.27%	72.81%
	Kr-inv	96.58%	90.68%	87.87%	78.61%	73.35%
	Tc-inv	96.65%	91.08%	88.47%	79.87%	75.88%
	OALM	97.47%	93.42%	91.41%	89.23%	86.54%
	THIM	95.34%	90.85%	90.75%	87.47%	85.14%
	KHIM	97.01%	92.00%	91.18%	89.87%	87.15%
	OIM(1)	98.17%	96.15%	95.35%	93.62%	91.72%
	OIM(2)	98.24%	96.24%	95.42%	93.71%	91.73%
	OIM(3)	98.35%	96.38%	95.58%	94.89%	92.01%
ImageNet database	Hu	94.14%	89.11%	83.28%	72.31%	72.11%
	CL-inv	94.41%	89.28%	86.15%	75.49%	71.25%
	Kr-inv	95.19%	90.17%	87.17%	79.72%	71.88%
	Tc-inv	96.34%	91.24%	90.55%	82.08%	77.18%
	OALM	97.84%	93.28%	91.37%	89.14%	87.28%
	THIM	95.98%	91.47%	90.15%	88.85%	86.44%
	KHIM	96.18%	91.08%	90.26%	89.41%	86.11%
	OIM(1)	97.22%	96.84%	94.28%	92.38%	90.27%
	OIM(2)	97.42%	96.97%	94.35%	93.04%	91.11%
	OIM(3)	97.49%	97.14%	95.05%	94.24%	91.84%

TABLE 3.5 – Résultats de classification par la méthode fuzzy k-means

Database	Invariant moments	Noise free	1 (%)	2(%)	3(%)	4(%)
MPEG7-CE database	Hu	95.68%	91.41%	85.87%	75.71%	74.13%
	CL-inv	96.75%	91.11%	90.01%	78.11%	74.28%
	Kr-inv	97.12%	93.11%	91.75%	80.06%	75.14%
	Tc-inv	98.02%	92.48%	90.21%	81.77%	77.87%
	OALM	99.20%	94.81%	92.13%	90.14%	88.14%
	THIM	98.11%	93.11%	92.45%	91.08%	88.17%
	KHIM	98.12%	92.08%	91.82%	90.51%	86.35%
	OIM(1)	99.10%	96.11%	95.48%	94.19%	91.71%
	OIM(2)	99.20%	96.21%	95.52%	94.79%	91.83%
OIM(3)	99.41%	96.45%	96.11%	95.22%	91.54%	
COIL-20 database	Hu	95.51%	90.37%	85.74%	75.28%	73.83%
	CL-inv	96.61%	91.78%	89.61%	78.42%	74.24%
	Kr-inv	97.02%	91.72%	89.68%	79.72%	77.28%
	Tc-inv	96.71%	92.34%	89.14%	81.71%	76.82%
	OALM	99.14%	94.71%	92.68%	90.48%	87.03%
	THIM	98.04%	93.09%	92.33%	90.95%	88.06%
	KHIM	98.03%	93.01%	91.69%	90.88%	88.02%
	OIM(1)	99.12%	98.25%	97.33%	95.12%	93.28%
	OIM(2)	99.24%	98.32%	97.45%	95.78%	93.82%
OIM(3)	99.42%	98.36%	97.59%	95.84%	93.89%	
MNIST database	Hu	96.43%	91.27%	85.67%	75.51%	74.01%
	CL-inv	97.59%	92.07%	89.08%	78.56%	74.28%
	Kr-inv	98.51%	92.58%	89.85%	80.58%	75.33%
	Tc-inv	98.69%	93.31%	90.76%	82.03%	77.33%
	OALM	99.97%	95.87%	93.72%	90.45%	88.77%
	THIM	97.62%	92.15%	91.45%	89.97%	87.58%
	KHIM	97.17%	92.14%	91.40%	90.05%	87.42%
	OIM(1)	99.94%	97.88%	96.75%	94.78%	92.95%
	OIM(2)	99.96%	97.91%	96.78%	94.81%	92.98%
OIM(3)	100%	97.94%	96.82%	94.83%	92.98%	
ImageNet database	Hu	96.51%	91.44%	85.58%	75.58%	74.43%
	CL-inv	97.63%	92.22%	89.38%	78.21%	74.14%
	Kr-inv	98.42%	92.63%	89.93%	80.62%	75.47%
	Tc-inv	98.75%	93.45%	90.82%	82.14%	77.41%
	OALM	99.92%	95.82%	93.68%	90.40%	88.72%
	THIM	97.55%	92.10%	91.40%	89.92%	87.53%
	KHIM	97.35%	92.13%	91.47%	90.12%	87.47%
	OIM(1)	99.47%	97.55%	96.47%	94.58%	92.44%
	OIM(2)	99.49%	97.73%	96.51%	94.62%	93.02%
OIM(3)	99.63%	97.81%	96.65%	94.78%	93.18%	



### 3.4. CONCLUSION

---

montrent l'efficacité des approches proposées.

# Chapitre 4

## Systemes biométriques basés sur les vecteurs descripteurs proposés

La sécurité des systèmes d'information est devenue un domaine de recherche d'une très grande importance dont l'identification de l'individu est essentielle pour assurer la sécurité des systèmes des organisations. Ainsi la conception d'un système d'identification fiable, efficace et puissant est une étape nécessaire. Dans ce sens, la biométrie est un exemple pratique parce qu'elle est de plus en plus présente dans la vie quotidienne : au travail des opérations bancaires, l'accès à certains endroits militaires ou industriels. Un système biométrique est un système de reconnaissance d'individus qui permet d'identifier une personne en fonction de ses caractéristiques physiologiques ou comportementales telles que : l'iris, la voix, les empreintes digitales, le visage, la signature, l'empreinte palmaire ... etc. (Fig.4.1). Dans les applications de contrôle d'accès, la biométrie permet d'apporter un niveau de sécurité supérieur en ce qui concerne des accès logiques (ordinateurs, comptes bancaires, etc.) ou des accès physiques (bâtiments sécurisés, aéroports, laboratoires etc.). La biométrie regroupe deux axes principaux : une identification (reconnaissance) et une authentification. Dans le cas d'identification, le système biométrique demande une information biométrique et la compare avec chaque information stockée dans la base de données. Alors que pour l'authentification l'utilisateur annonce son identité par une information biométrique, et le système compare les données obtenues à partir de l'information entrée avec la donnée enregistrée.

### 4.1 Architecture d'un système biométrique

Tout système biométrique se compose en deux opérations qui se chargent à réaliser les phases d'enregistrement et de test :

L'enregistrement : est une opération qui permet d'enregistrer les caractéristiques des utilisateurs dans la base de données.

Le test : est une opération qui permet de réaliser l'identification ou la vérification d'une personne.

Dans chacune des deux opérations mentionnées, le système exécute quatre processus :

**L'acquisition** : consiste à utiliser un capteur pour acquérir une caractéristique de l'individu, plusieurs processus peuvent être utilisés pour l'acquisition comme l'appareil photo, un lecteur d'empreintes digitales, une caméra de sécurité, le scanner laser pour le visage, etc.

**L'extraction des vecteurs descripteurs** : Après l'acquisition d'une image, nous réalisons l'extraction du vecteur descripteur dont le processus d'authentification a besoin. Donc, ce module sert à traiter l'image afin d'extraire uniquement les caractéristiques biométriques, sous forme d'un vecteur, qui peuvent être ensuite utilisées pour reconnaître les personnes.

**La classification** : En examinant les vecteurs descripteurs stockés dans la base de données, les caractéristiques biométriques extraites sont comparées avec ce vecteur en utilisant la distance Euclidienne.

**La décision** : la stratégie de décision nous permet de vérifier l'identité affirmée par un utilisateur ou déterminer l'identité d'une personne basée sur le degré de similitude entre les caractéristiques extraites et les vecteurs stockés. La figure 4.2 présente une architecture d'un système de reconnaissance biométrique. Dans ce travail, nous proposons un système biométrique basé sur le modèle de Perona-Malik et la méthode des éléments finis mixte pour le prétraitement, et les moments orthogonaux invariants proposés précédemment pour l'extraction des caractéristiques.

## 4.2 Les catégories biométriques utilisées

Les caractéristiques biométriques de l'individu peuvent être encore classées en trois grandes catégories : l'analyse physique ou morphologique comme les empreintes digitales, la forme de la main, les traits du visage, les traces biologiques comme l'odeur, l'ADN, et l'analyse comportementale comme la dynamique du tracé de la signature, la frappe sur un clavier, etc. Dans cette partie, en se basant sur la méthode des éléments finis mixte et les moments invariants orthogonaux proposés précédemment, nous introduisons les quatre systèmes biométriques suivants

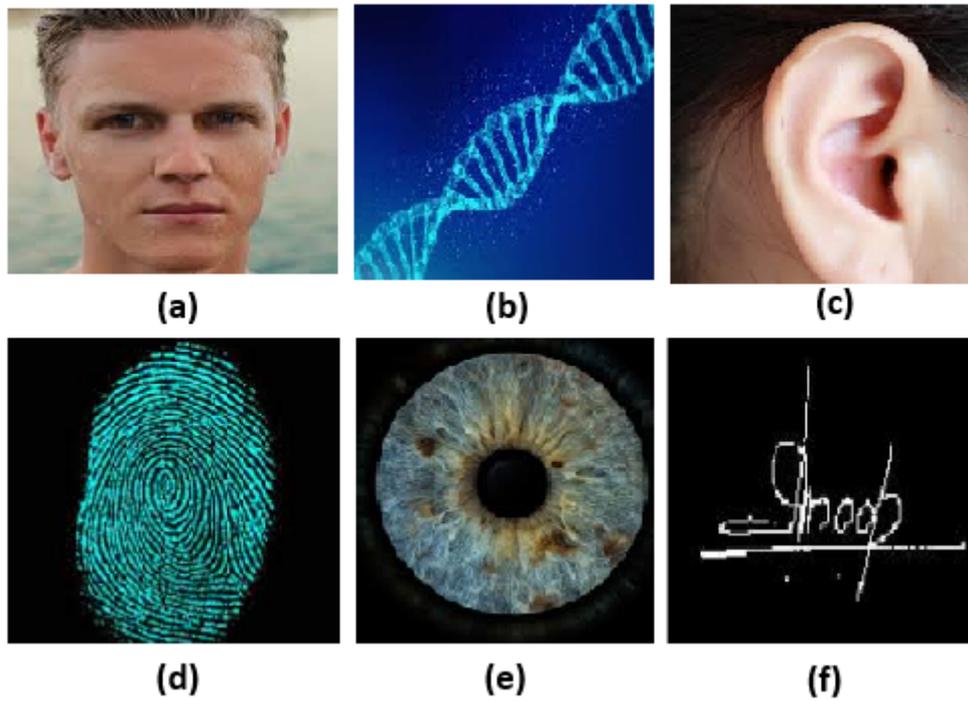


FIGURE 4.1 – Caractéristiques biométriques : (a) visage, (b) ADN, (c) oreille, (d) empreintes digitales, (e) iris, (f) signature.

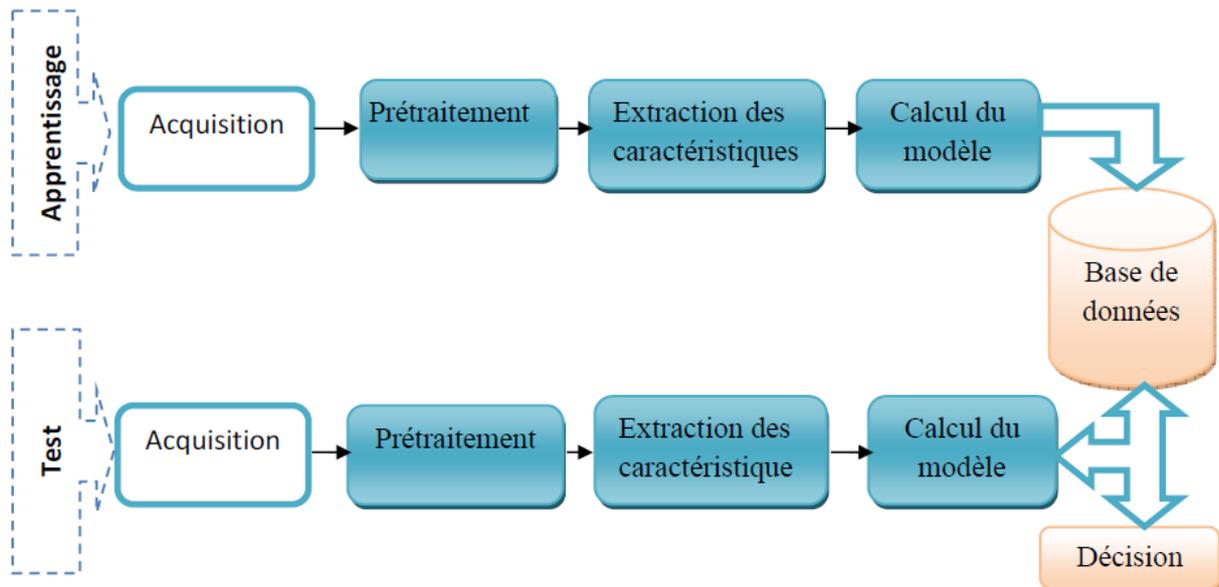


FIGURE 4.2 – Architecture d'un système de reconnaissance biométrique

### 4.2.1 Système biométrique basé sur les empreintes digitales

Les empreintes digitales sont les caractéristiques biométriques les plus communément utilisées pour la reconnaissance de criminels. Le premier système automatique d'authentification utilisant les empreintes digitales a été commercialisé au début des années soixante. D'autres parts, plusieurs études ont démontré que l'iris est la caractéristique la plus fiable car la texture de l'iris reste stable au cours de la vie. Comme montré dans la Figure 4.6, à l'aide d'un lecteur spécifique, nous scannons les empreintes digitales d'un individu, nous utilisons le modèle de Perona-Malik et la méthode des éléments finis mixte pour le prétraitement et les moments invariants orthogonaux pour l'extraction du vecteur caractéristique.

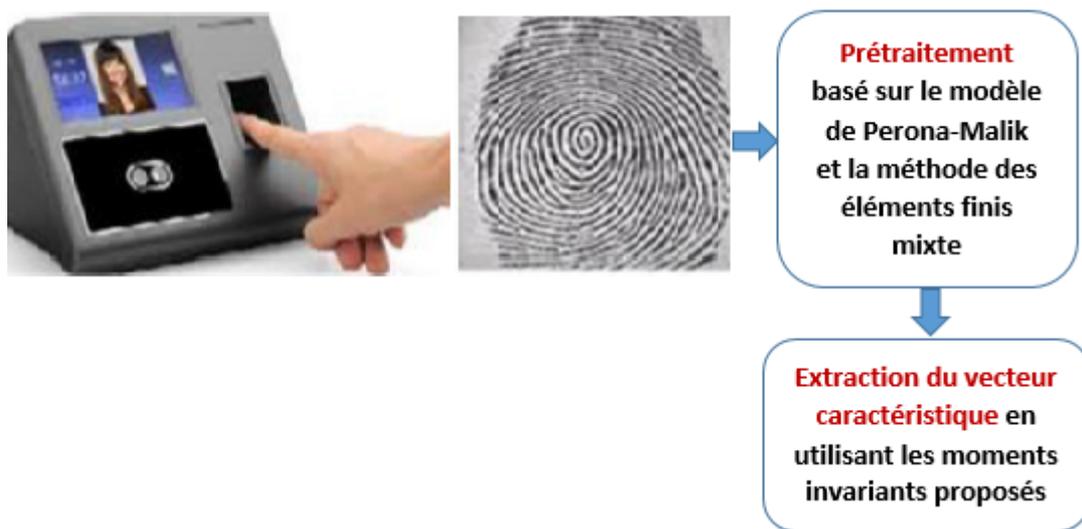


FIGURE 4.3 – Système biométrique basé sur les empreintes digitales

### 4.2.2 Système biométrique basé sur le visage

La reconnaissance faciale fait partie des techniques biométriques. On remarque que dans la vie quotidienne chacun de nous identifie tout au long de la journée différents visages. Ainsi lorsque nous rencontrons une personne, notre cerveau va chercher dans notre mémoire et vérifier si cette personne est répertoriée ou non. La difficulté de la reconnaissance de visage par ordinateur varie énormément suivant que les conditions d'acquisition. Dans un environnement contrôlé, des paramètres tels que l'arrière plan, la direction et l'intensité des sources lumineuses, l'angle de la prise de vue, la distance de la caméra au sujet sont des paramètres maîtrisés par le système. Plusieurs méthodes ont été développées pour la reconnaissance de visage 2D. Dans ce travail, nous présentons un système basé

sur les approches proposées précédemment (Fig.4.6).

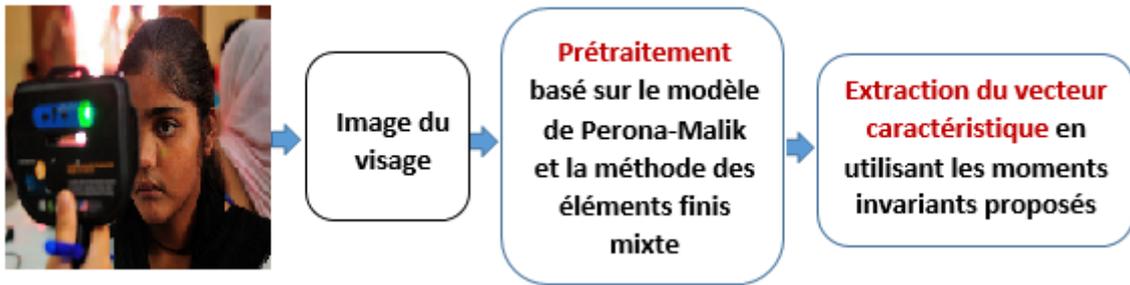


FIGURE 4.4 – Système biométrique basé sur le visage

### 4.2.3 Système biométrique basé sur l'iris

L'iris est une région sous forme d'anneau, située entre la pupille et le blanc de l'oeil, elle est unique. L'iris a une structure extraordinaire et offre de nombreuses caractéristiques de texture qui sont uniques pour chaque individu. La reconnaissance de l'iris a été développée dans les années 80, elle est donc considérée comme une technologie récente. L'image de l'iris est capturée par un appareil qui contient une caméra infrarouge, lorsque la personne se place à une courte distance de l'appareil.

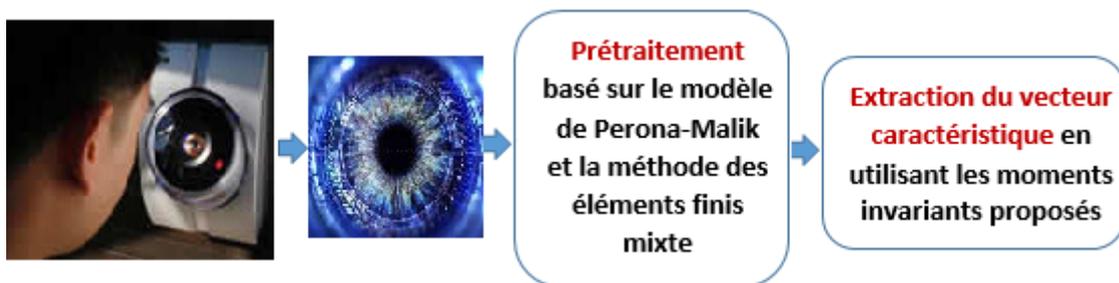


FIGURE 4.5 – Système biométrique basé sur l'iris

### 4.2.4 Système biométrique basé sur la signature

C'est une écriture personnelle d'un individu, la vérification de la signature met l'accent sur les formes géométriques de la signature, dans ce mode, en générale, la signature est normalisée à une taille connue ensuite décomposer en élément simple.

Après l'acquisition des images, nous utilisons un système d'indexation et de recherches d'images par le contenu. Ce type d'indexation consiste à caractériser le contenu visuel des

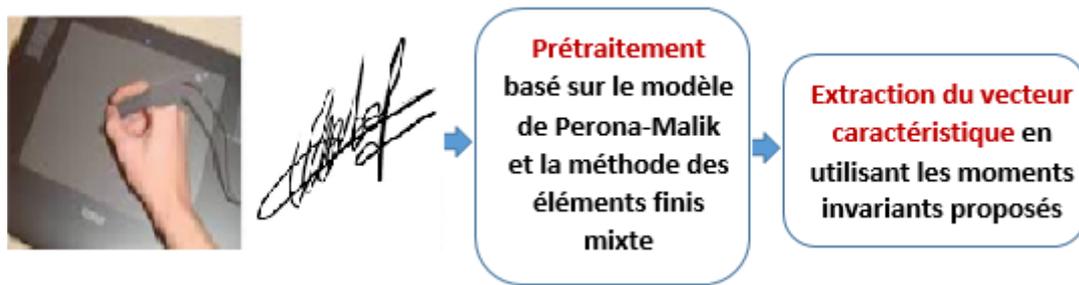


FIGURE 4.6 – Système biométrique basé sur la signature

images par des caractéristiques (descripteurs), à effectuer des recherches par similarité visuels à partir de ces descripteurs. L'indexation par le contenu a donc pour but d'extraire et de représenter le sens d'une image de manière à ce qu'elle puisse être retrouvée par l'utilisateur. Bien que plusieurs années de travaux scientifiques en recherche d'images par le contenu contemporains, ont d'ores et déjà permis la réalisation d'outils performants, permettant plusieurs formes d'indexation par analyse du contenu mais ce domaine demeure encore aujourd'hui un problème ouvert et très actif. Les images généralement sont caractérisées à l'aide de leur forme, couleur et texture. IL consiste à récupérer les images les plus similaires sur le plan visuel à une image de requête donnée à partir d'une base de données d'images. La fonction (CBIR) est basée sur l'application de technique de vision par ordinateur au problème de récupération d'image dans les grandes bases de données. Cette technique s'oppose à la recherche d'images par mots-clés ou tags, qui fut historiquement proposée par les moteurs de recherche tels que Google , Yahoo, etc. Dans ce travail, nous proposons un nouveau système de recherche d'image par le contenu et un nouveau système biométrique basés sur les approches proposées pour extraire les vecteurs descripteurs des images.

### 4.3 Architecture du système d'indexation et recherche d'images

Le système de recherche d'image s'exécute en deux étapes : l'étape d'indexation et l'étape de recherche. Dans l'étape d'indexation, des caractéristiques sont automatiquement extraites à partir de l'image et stockées dans un vecteur descripteur visuel. Grâce aux techniques de la base de données, on peut stocker ces caractéristiques et les récupérer rapidement et efficacement.

Dans l'étape de l'indexation, nous utilisons la méthode présentée dans les deux pre-

miers chapitres pour extraire les vecteurs descripteurs,

$$V(f) = \left( MI_{ij}(D(f)) \right)_{i,j=0,\dots,p} \quad (4.1)$$

Ce vecteur descripteur est basé sur le débruitage d'image à l'aide du modèle de Perona-Malik et la théorie des moments orthogonaux invariants. Dans l'étape de recherche, le système prend une requête à l'utilisateur et lui donne le résultat qui correspond à une liste d'images ordonnées en fonction de la similarité entre leur descripteur visuel et celui de l'image requête en utilisant une mesure de distance. Néanmoins, nous nous concentrons sur l'étape de recherche.

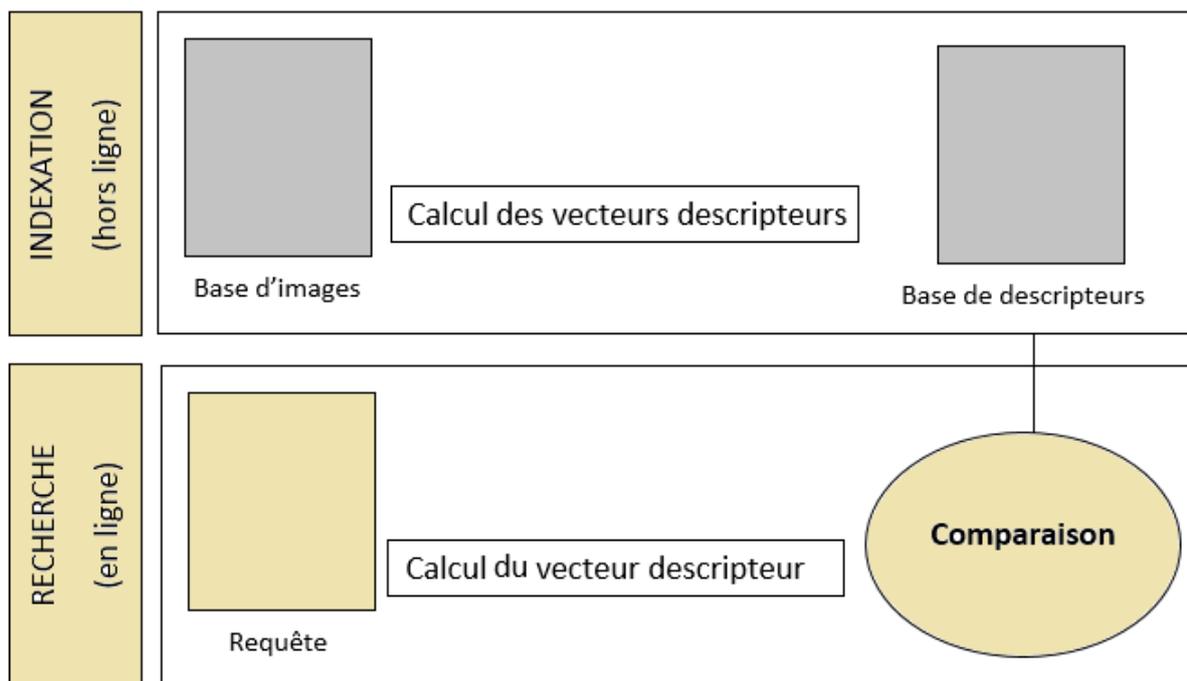


FIGURE 4.7 – Principe général d'indexation et de recherche d'images

## 4.4 Mesures pour évaluer un système

Avant l'exécution d'un système de recherche d'informations, une évaluation qui permet de mesurer la performance de ce système est nécessaire. Les mesures les plus courantes pour évaluer un système sont le temps de réponse et l'espace utilisé. Plus le temps de réponse est court, plus l'espace utilisé est petit, et plus le système est considéré bon. Mais avec des systèmes qui ont été faits pour la recherche d'informations, en plus de ces deux mesures, on s'intéresse à d'autres mesures. Dans le système de recherche d'infor-

mations, l'utilisateur s'intéresse aux réponses pertinentes du système. Donc les systèmes de recherche d'informations exigent l'évaluation de la précision de la réponse. Ce type d'évaluation est considéré comme l'évaluation des performances de recherche. Le système d'indexation et de recherche d'images est un système de recherche d'informations. Dans les systèmes de recherche d'images, les auteurs ont souvent utilisé les mesures d'évaluation que l'on a utilisé pour évaluer des systèmes de recherche d'informations. Dans cette section, nous allons décrire les deux mesures les plus courantes : le rappel et la précision. Ces mesures sont reliées entre elles. Donc on décrit souvent cette relation par une courbe de rappel et précision. Ensuite nous présentons d'autres mesures que l'on utilise aussi pour évaluer des systèmes de recherche d'informations.

##### 4.4.1 Rappel

Le rappel est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images pertinentes dans la base d'images.

$$Rappel = \frac{|Ra|}{|R|} \quad (4.2)$$

##### 4.4.2 Précision

La précision est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images trouvées.

$$Precision = \frac{|Ra|}{|A|} \quad (4.3)$$

Où

- $R$  : l'ensemble d'images pertinentes dans la base d'images utilisée pour évaluer.
- $|R|$  : le nombre d'images pertinentes dans la base d'images.
- $A$  : l'ensemble des réponses.
- $|A|$  : le nombre d'images dans l'ensemble des réponses.
- $|Ra|$  : le nombre d'images pertinentes dans l'ensemble des réponses.

Des définitions sont montrées dans la figure 4.1 : Dans les systèmes de recherche d'informations, afin de définir si une information est pertinente ou non, on a besoin d'experts dans le domaine. Dans les systèmes de recherche d'images, une image est pertinente pour une requête si les deux images sont dans la même classe. C'est pourquoi dans l'étape de préparation de la base d'images à évaluer, on doit faire des annotations. L'annotation est un processus qui permet aux utilisateurs de choisir des mots clés correspondants à

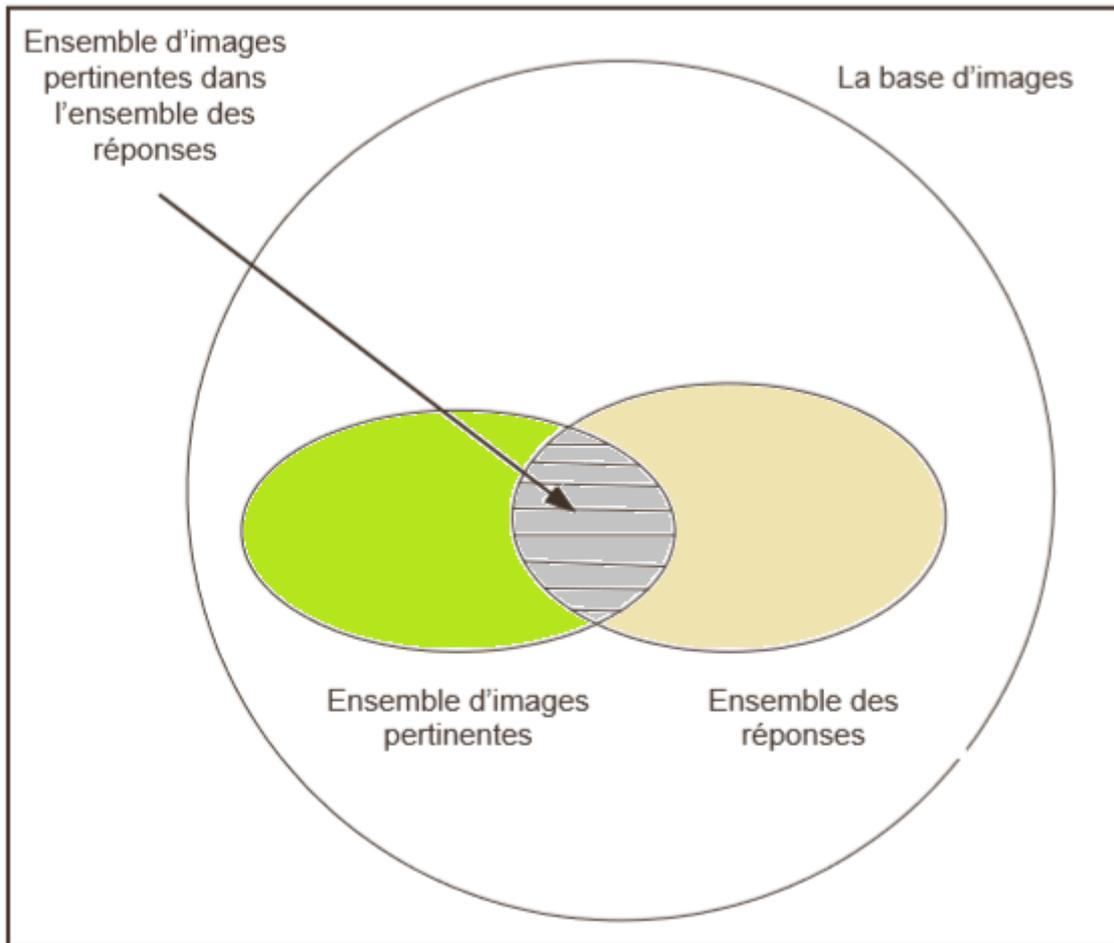


FIGURE 4.8 – Le rappel et la précision pour une requête (Yates, 1999)

chaque image. Après l'annotation, on va classifier les images en classes appropriées. Si des images ne contiennent pas beaucoup d'objets, c'est facile de les classifier dans ces classes. Mais si les images contiennent beaucoup d'objets, la tâche de classification devient de plus difficile. Dans ce cas, chaque image appartient à plusieurs classes.

### 4.4.3 Courbe rappel-précision

Le rappel et la précision sont les mesures importantes, mais si on voit seulement une paire de valeurs de rappel et précision, cette paire de valeurs ne peut pas indiquer la performance du système. C'est pourquoi on donne souvent une distribution de rappel et précision sous en forme de courbe. La figure 4.3 donne un exemple de courbe de rappel et précision. Pour dessiner cette courbe, on doit calculer plusieurs paires de rappel et précision et les interpoler. En pratique, on utilise plusieurs requêtes. Dans ces cas pour évaluer un système, on calcule la précision moyenne pour toutes les requêtes correspondant

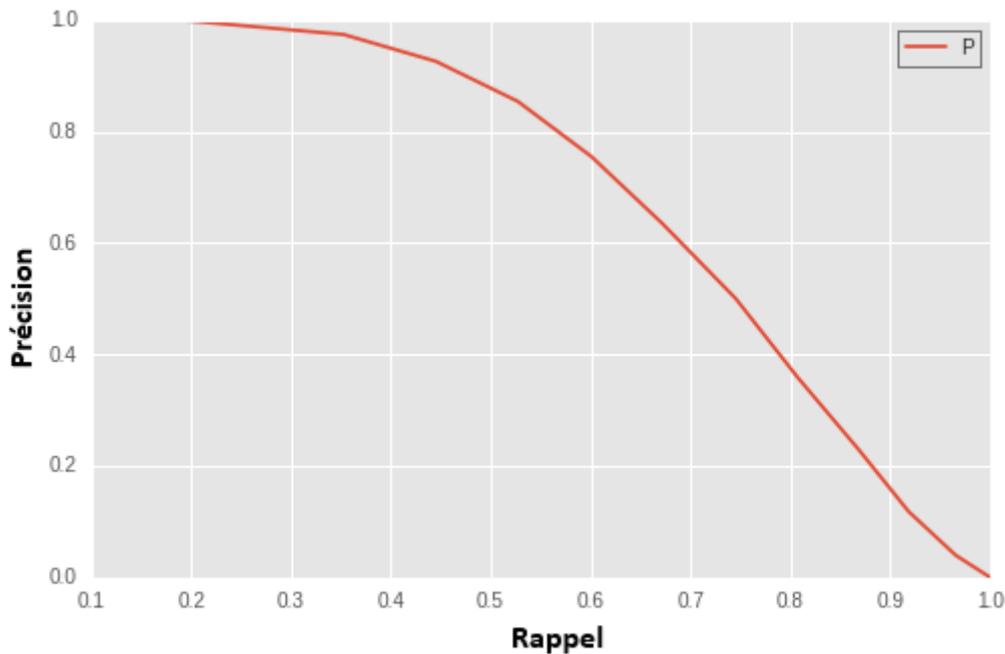


FIGURE 4.9 – Une courbe de rappel et précision

à chaque niveau de rappel.

$$\overline{P(r)} = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q} \quad (4.4)$$

Où  $N_q$  est le nombre de requêtes et  $P_i(r)$  est la précision pour le rappel  $r$  avec la requête  $i$ .

## 4.5 Résultats expérimentaux

### 4.5.1 Bases d'images utilisées

Pour évaluer et valider notre système, nous avons utilisé trois bases de données d'images, qui sont disponibles sur Internet librement. Ces bases d'images possèdent déjà des classes définies où chaque image n'appartient qu'à une seule classe.

**La base de données de visages ORL** qui contient un ensemble d'images de visages prises entre avril 1992 et avril 1994 au laboratoire. La base de données a été utilisée dans le cadre d'un projet de reconnaissance de visage réalisé en collaboration avec le groupe de discours, vision et robotique du département d'ingénierie de l'université de Cambridge. Il existe dix images différentes de chacun des 40 sujets distincts. Pour certains sujets, les

images ont été prises à des moments différents, en modifiant l'éclairage, les expressions du visage (yeux ouverts / fermés, souriant / pas souriant) et les détails du visage (lunettes / pas de lunettes). Toutes les images ont été prises sur un fond sombre et homogène, les sujets étant dans une position verticale et frontale (avec une tolérance pour certains mouvements latéraux). Un aperçu de la base de données de visages est présenté dans la figure 3.10.



FIGURE 4.10 – Quelques images de la base de données ORL.

**La base de données XM2VTS** [Mes99] qui contient 295 individus avec 4 sessions et 2 images par session pour chaque individu soit un total de 8 images par personne. Ces images contiennent des variations de luminosité ainsi que de petites variations de poses avec un hochement horizontal et vertical de la tête. Dans notre protocole d'expérimentation nous avons pris pour chaque personne quatre images de différentes sessions ce qui fait une base de travail de 1180 images.

**La base de données FERET** La base de données FERET a été collectée dans le cadre du programme « Facial Recognition Technology » [PMRR00, Nist.gov] mené par le National Institute of Standards and Technology (NIST) Américain. Il s'agit de la plus grande base disponible pour les chercheurs qui a été acquise avec des poses différentes et durant 15 sessions entre 1993 et 1996. Les images, initialement collectées depuis un appareil photographique de 35mm ont ensuite été digitalisées. Une première version de cette base de données a été réalisée en 2001 et contient 14051 images faciales en niveaux de gris avec une résolution de  $256 \times 384$  pixels. La version la plus récente, réalisée en 2003, renferme des images numériques couleurs de plus grande qualité avec une résolution de  $512 \times 768$



FIGURE 4.11 – Exemples des images de la base de données XM2VTS.

pixels et une compression sans perte de données à la différence des premières images en niveaux de gris. En plus, de multiples erreurs de noms d'images, d'identifiants, et dates de capture, qui figurent sur la première base à niveau de gris ont été corrigées. Cette dernière base contient 11338 images représentant 994 personnes différentes. Pour chaque individu, on dispose d'une vue faciale régulière *fa* et une vue faciale alternative *fb* prise un peu après *fa*. D'autres poses ont été acquises pour la majorité de ces individus allant du profil gauche au profil droit avec des rotations de  $15^\circ$ ,  $22^\circ$ ,  $45^\circ$ ,  $67^\circ$ , et  $90^\circ$  en profondeur de la tête. Pour quelques personnes de la base, on dispose d'autres vues « duplicate » collectées dans des conditions similaires à *fa* et *fb* mais dans des sessions ultérieures. Aucune contrainte n'est imposée sur la date de la prise de vue de l'image duplicate I. par contre, la vue duplicate II a été collectée au moins 540 jours après la première prise de vue.

### 4.5.2 Nos résultats

La première expérience est réalisée sur la base de données ORL. Nous avons considéré l'image requête "face (b)" (Fig.4.13), et nous avons utilisé notre système précédent pour récupérer les 8 meilleures images. Les deux phases de notre système sont répétées pour cette image de requête afin de récupérer les 10 premières images. Les résultats obtenus sont représentés sur les figures 4.14, 4.15 et 4.16. D'après ces résultats, nous remarquons que le taux de précision est de 87,5% lorsque nous utilisons les moments invariants orthogonaux

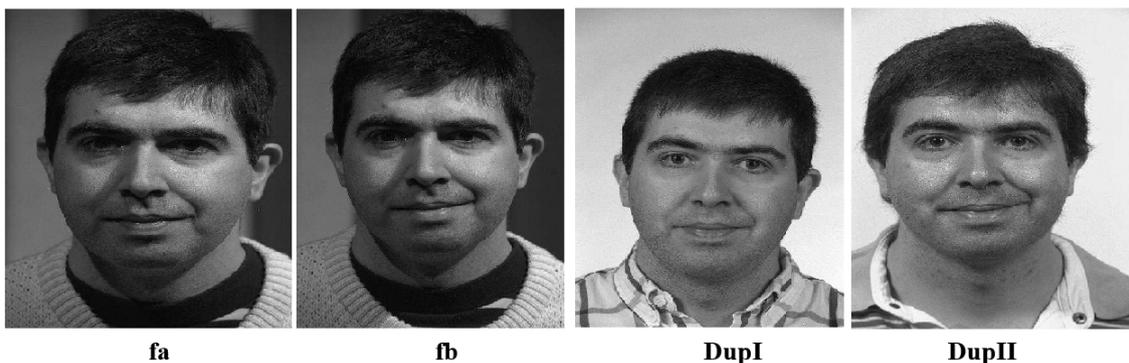


FIGURE 4.12 – Extrait de la base Color FERET

$OIM^{(1)}$  pour extraire les vecteurs descripteurs et de 100% si nous utilisons les moments invariants orthogonaux  $OIM^{(2)}$  ou  $OIM^{(3)}$ . Nous allons maintenant présenter une étude comparative de notre méthode de récupération d'image basée sur les moments invariants orthogonaux  $OIM^{(i)}$ ,  $i = 1, 2, 3$  avec d'autres méthodes bien connues, également basées sur la forme. des descripteurs tels que AHDH [16] et HOG [17]. Les descripteurs de couleur en tant que descripteurs calculés dans les espaces colorimétriques : RVB, HSV, YCrCb et HMMD [18] et les descripteurs de couleur d'histogramme (Hist) [19]. Et les descripteurs de texture en tant que descripteurs de statistiques de premier ordre (FOS) calculés à partir de l'histogramme des images en niveaux de gris [20] et des descripteurs de la dépendance des niveaux de gris (SGLD) [21].



FIGURE 4.13 – Images requêtes.

Pour évaluer le pouvoir de récupération de chaque descripteur, nous avons effectué les tests sur les bases de données d'images précédentes en utilisant les images requêtes présentées dans la figure 4.13, et les résultats expérimentaux obtenus sont illustrés dans les tableaux 4.1-4.6. La performance est également testée avec les mesures de précision et de rappel définies dans les équations (4.2) et (4.3). Les tests sont effectués avec les images des bases de données précédentes en utilisant nos moments invariants orthogonaux  $OIM^{(i)}$ ,  $i = 1, 2, 3$  et les autres descripteurs telles que AHDH, HOG, Hu, Hist, HSV, FOS et SGLD. Sachant que les conditions de ces expériences sont satisfaites, la précision et le rappel sont



FIGURE 4.14 – les 8 meilleures images récupérées de l'image requête "face (b)" de la base de données ORL en utilisant les moments invariants  $OIM^{(1)}$ .



FIGURE 4.15 – les 8 meilleures images récupérées de l'image requête "face (b)" de la base de données ORL en utilisant les moments invariants  $OIM^{(2)}$ .



FIGURE 4.16 – les 8 meilleures images récupérées de l'image requête "face (b)" de la base de données ORL en utilisant les moments invariants  $OIM^{(3)}$ .

## 4.6. CONCLUSION

---

TABLE 4.1 – Taux de récupération moyen des 15 meilleures images récupérées de la base de données FERET en utilisant nos moments  $OIM^{(i)}$ ,  $i = 1, 2, 3$ , et les méthodes AHDH et HOG .

	$OIM^{(1)}$	$OIM^{(2)}$	$OIM^{(3)}$	AHDH	HOG
TRM	75.55%	81.23%	85.57%	65.72%	73.25%

TABLE 4.2 – Taux de récupération moyen des 20 meilleures images récupérées de la base de données FERET en utilisant nos moments  $OIM^{(i)}$ ,  $i = 1, 2, 3$ , et les méthodes AHDH et HOG.

	$OIM^{(1)}$	$OIM^{(2)}$	$OIM^{(3)}$	AHDH	HOG
TRM	72.85%	79.27%	80.55%	61.25%	70.16%

calculés et les résultats obtenus sont représentés sous forme de graphes présentés dans les figures 4.17-4.23 Les résultats de la comparaison montrent la supériorité de moments invariants orthogonaux proposés. Enfin, les  $OIM^{(i)}$ ,  $i = 1, 2, 3$  proposés sont robustes pour les transformations d'images.

## 4.6 Conclusion

Dans ce chapitre, nous avons proposé des nouveaux systèmes biométriques. Pour représenter les images acquises, nous avons utilisé la méthode des éléments finis mixte et le modèle de Perona-Malik pour le prétraitement, et des nouveaux moments orthogonaux invariants pour extraire les vecteurs descripteurs des images. Nous avons utilisé un système de recherche d'image basé sur la distance euclidienne pour récupérer les images similaires à l'image requête. Pour évaluer la performance des approches proposées, nous avons effectué des expérimentations numériques sur des bases de données d'images bien connues comme la base d'image XM2VTS, la base FERET et la base de visages ORL. Nous avons présenté une étude comparative avec les méthodes existantes. Les résultats obtenus montrent l'efficacité et la supériorité des systèmes biométriques proposés.

TABLE 4.3 – Taux de récupération moyen des 9 meilleures images de récupérées de la base de données "ORL" en utilisant nos moments  $OIM^{(i)}$ ,  $i = 1, 2, 3$ , et les méthodes Hu et HOG.

	$OIM^{(1)}$	$OIM^{(2)}$	$OIM^{(3)}$	Hu	HOG
TRM	77.77%	89.00%	89.00%	55.56%	66.70%

## 4.6. CONCLUSION

TABLE 4.4 – Taux de récupération moyen des 20 meilleures images de récupérées de la base de données XM2VTS en utilisant nos moments  $OIM^{(i)}$ ,  $i = 1, 2, 3$ , et les méthodes Hu et HOG.

	$OIM^{(1)}$	$OIM^{(2)}$	$OIM^{(3)}$	Hu	HOG
TRM	85.51%	90.00%	95.01%	75.20%	80.25%

TABLE 4.5 – Taux de récupération moyen des 23 meilleures images de récupérées de la base de données XM2VTS en utilisant nos moments  $OIM^{(i)}$ ,  $i = 1, 2, 3$ , et et les caractéristiques couleur et texture Hist, HSV, F.O.S et S.G.L.D

	$OIM^{(1)}$	$OIM^{(2)}$	$OIM^{(3)}$	Hist	HSV	F.O.S	S.G.L.D
TRM	73.15%	88.16%	92.20%	63.33%	69.12%	70.13	67.15

TABLE 4.6 – Taux de récupération moyen des 30 meilleures images de récupérées de la base de données XM2VTS en utilisant nos moments  $OIM^{(i)}$ ,  $i = 1, 2, 3$ , et et les caractéristiques couleur et texture Hist, HSV, F.O.S et S.G.L.D

	$OIM^{(1)}$	$OIM^{(2)}$	$OIM^{(3)}$	Hist	HSV	F.O.S	S.G.L.D
TRM	65.23%	73.17%	83.13%	59.21%	63.27%	64.73%	60.43%

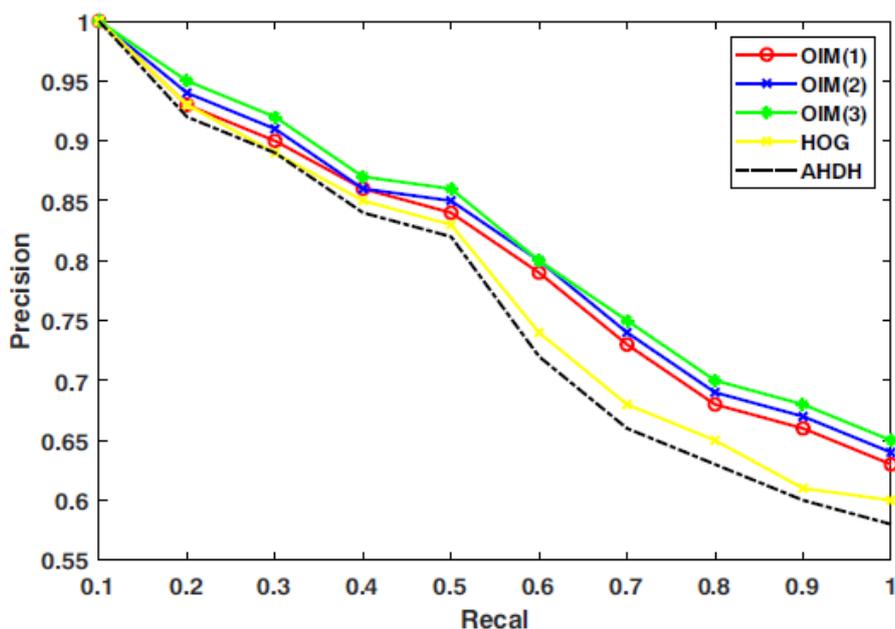


FIGURE 4.17 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , et AHDH, HOG et Hu pour un ensemble d'images de la base de données FERET

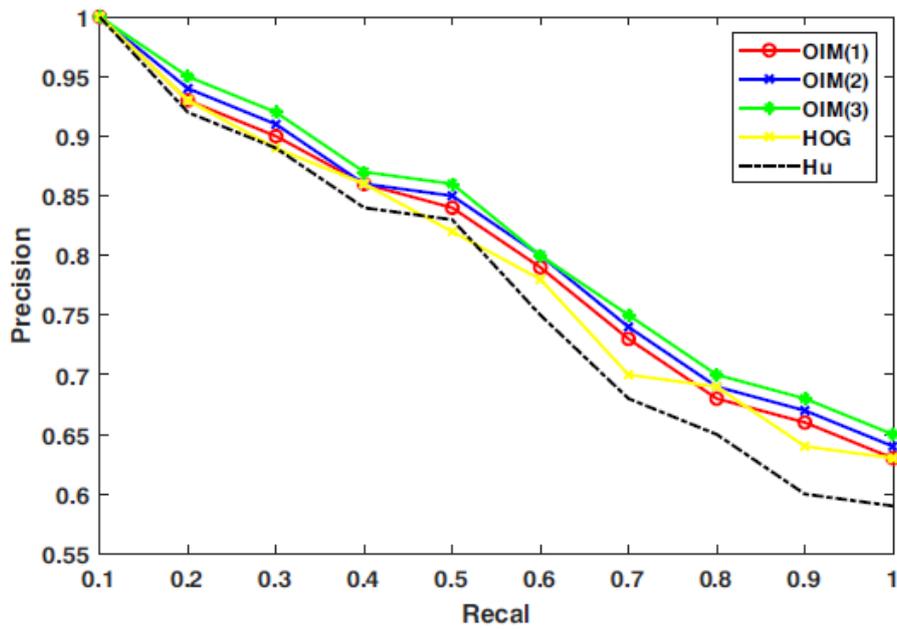


FIGURE 4.18 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , et AHDH, HOG et les moments de Hu pour un ensemble d'images de la base de données ORL

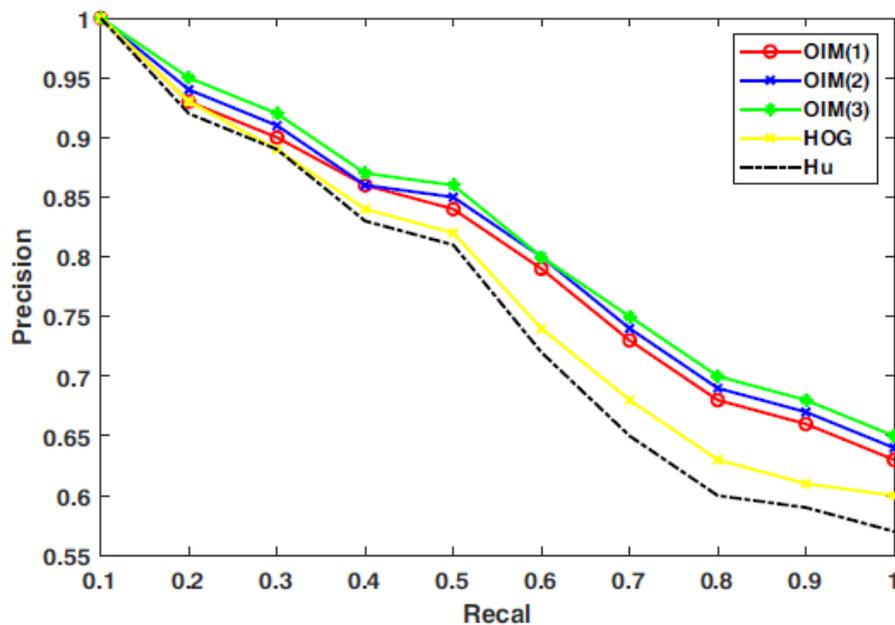


FIGURE 4.19 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , HOG, Hu, Hist, HSV, F.O.S et S.G.L.D pour un ensemble d'images de la base de données XM2VTS .

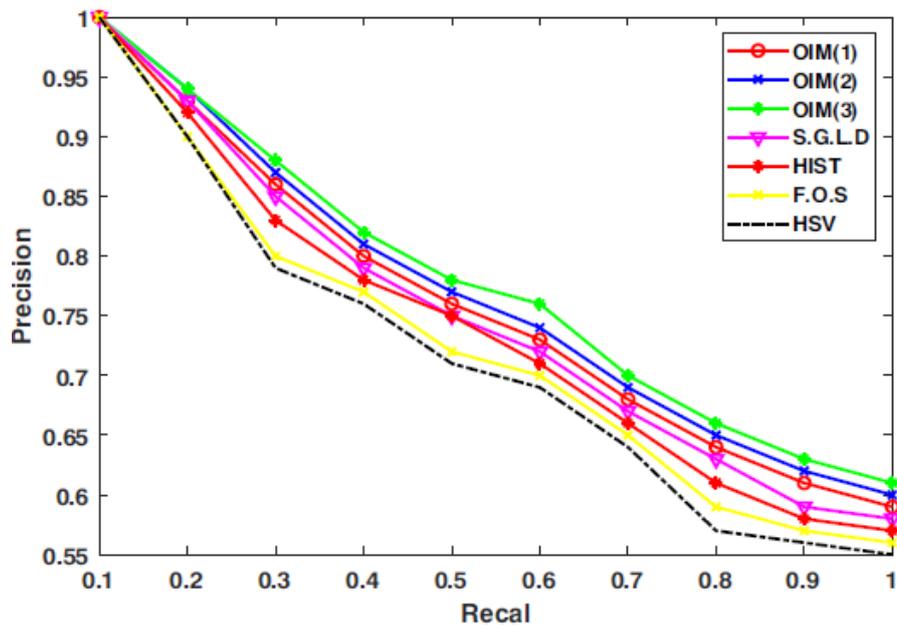


FIGURE 4.20 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , HOG, Hu, Hist, HSV, F.O.S et S.G.L.D pour la base de données XM2VTS.

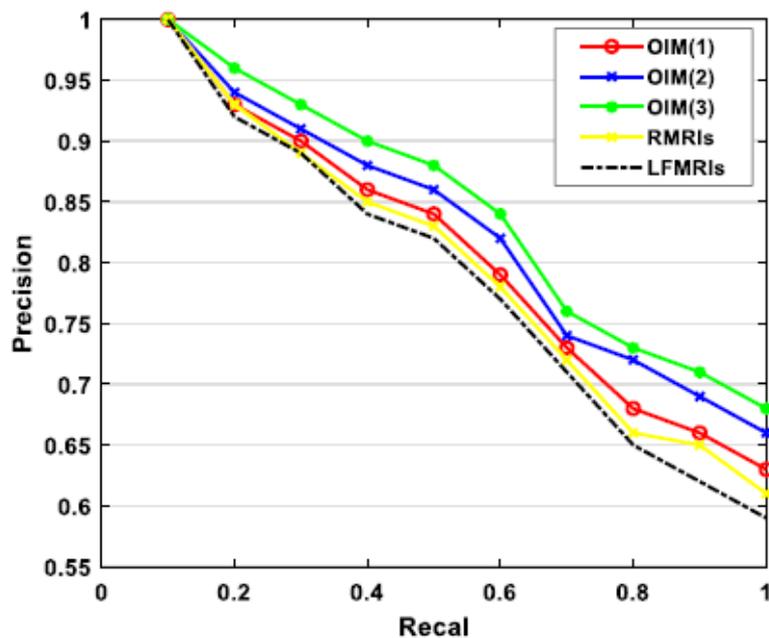


FIGURE 4.21 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , les moments invariants RMRIs [16] et LFMRI [20] pour la base de données FERET.

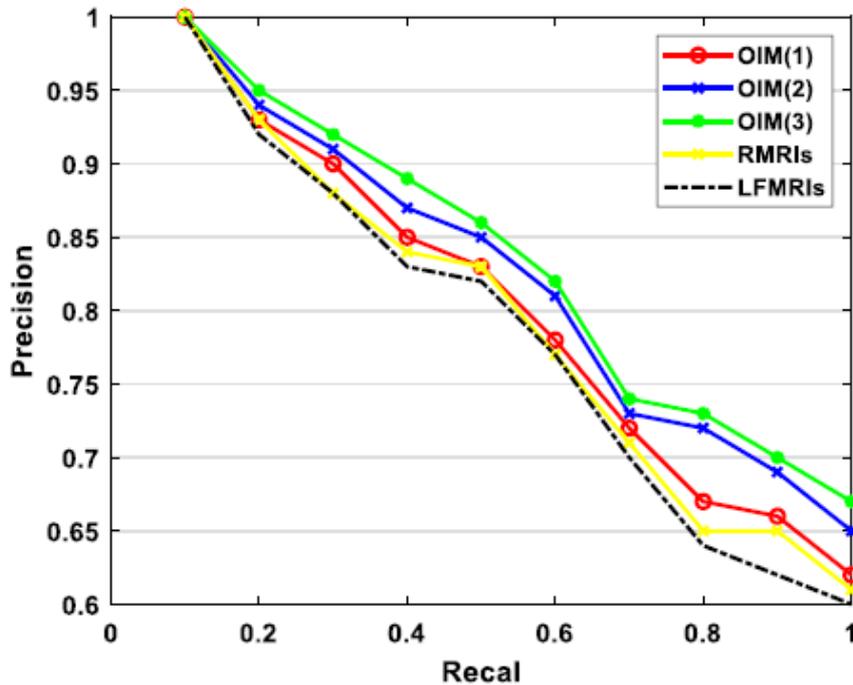


FIGURE 4.22 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , les invariants RMRIs [16] et LFMRI [20] pour la base de données ORL.

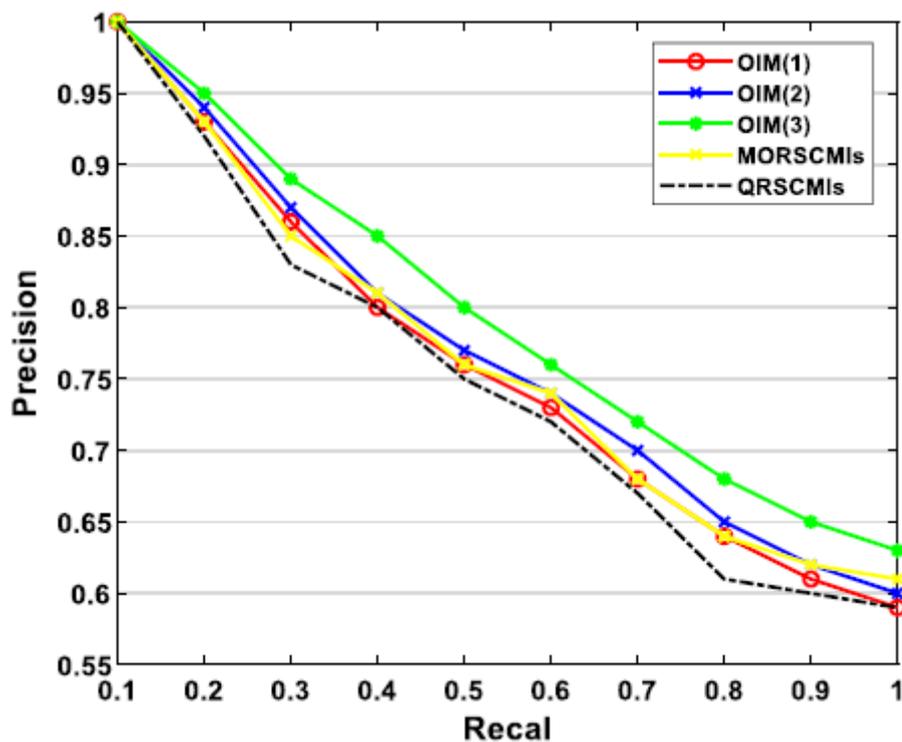


FIGURE 4.23 – Courbes rappel-précision obtenues en utilisant les moments invariants proposés  $OIM^{(i)}$ , les moments invariants MORSCMIs et QRSCMIs [16] pour la base de données XM2VTS

## Conclusion et perspectives

Le but essentiel de ce travail était de proposer un nouveau système de classification d'images, particulièrement, valable pour la reconnaissance biométrique. Ce système se compose de trois étapes. Dans la première étape, nous avons utilisé la méthode des éléments finis mixtes pour résoudre le modèle de Perona-Malik qui permet de réduire le bruit de l'image sans supprimer les parties significatives du contenu de l'image, généralement, les bords, les contours, les textures ou d'autres détails importants pour l'interprétation de l'image. Nous nous sommes intéressés dans la deuxième étape à l'extraction des vecteurs descripteurs des images en utilisant la théorie des moments invariants. Les sept moments introduit par Hu sont largement utilisés dans le domaine de la classification d'images et la reconnaissance de formes. Ces moments sont d'ordre fini, par conséquent, ils n'arrivent pas à extraire un ensemble complet de caractéristiques de l'image. Pour cette raison, nous avons proposé dans ce travail de nouvelles approches : Dans la première approche, nous avons introduit un ensemble de moments invariants en fonction des moments centraux normalisés d'ordre infini permettant de remédier au problème la description incomplète de l'image qui pose par les moments invariants de Hu. Ces moments invariants ne sont pas orthogonaux, ce qui implique ainsi une redondance d'information. Pour remédier à ce problème, nous avons introduit une série de moments invariants orthogonaux basés sur un ensemble des polynômes orthogonaux non séparables à deux variables, qui sont construits à l'aide d'une nouvelle technique axée sur une relation de récurrence. Dans la deuxième approche, nous avons introduit un nouvel ensemble de polynômes orthogonaux basés sur les polynômes de Legendre, que nous appelons, les polynômes de Legendre adaptés. A partir de ces polynômes nous avons construit une famille de moments invariants pour les trois transformations géométriques (la translation, le changement d'échelle et la rotation) d'une image. Dans la troisième approche, nous avons proposé un ensemble de moments invariants orthogonaux centrés sur les polynômes orthogonaux de Jacobi en utilisant les coordonnées polaires. Dans la dernière étape, nous avons utilisé les algorithmes de classification supervisée et non supervisée comme les méthodes de k-means, fuzzy k-means, possibilistic fuzzy k-means, k-plus proche voisin, les réseaux de neurones artificiels RBF et MLP. Nous avons aussi proposé un système d'indexation et recherche d'images basé sur les vecteurs descripteurs proposés. Pour évaluer la performance de nos systèmes de classification, d'indexation et de recherche d'images, nous avons effectué des tests expérimentaux sur des bases de données d'images bien connues et les résultats obtenus montrent l'efficacité et la supériorité des approches proposées.

En perspective nous allons étendre notre approche basée sur le filtrage des images par la méthode des éléments finis mixte pour résoudre le modèle de Perona-Malik et sur la théorie des moments invariants orthogonaux dans le domaine de classification des images couleurs 2D et 3D. Nous allons aussi explorer d'autres bases des polynômes tels que les polynômes de Meixner.

# Bibliographie

- [1] M.K Hu, Visual pattern recognition by invariant moment invariants, IRE Trans. Inform. Theory 8, 179-187 (1962).
- [2] A. Khotanzad, Y. Hong, Invariant image recognition by zernike moments, IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 489-497 (1990).
- [3] S.X. Liao, M. Pawlak, On image analysis by moments, IEEE Trans. Pattern Anal. Mach. Intell. 18, 254-266 (1996).
- [4] T. Asano, N. Yokoya, Image segmentation schema for low-level computer vision, Patterrn Recognition 14, 267-273 (1981).
- [5] H.C Shen, C.Y.C, Bie, D.K.Y Chiu, A texture-based distance measure for classification, Pattern recognition 26, 1429-1437 (1993).
- [6] P. Srivastava, N.T. Binh, A. Khare, Content-based image retrieval using moments of local ternary pattern, Mobile Networks and Applications 19, 618-625 (2014).
- [7] M. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Amer. 70, 920-930 (1980).
- [8] J.C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well separated clusters, Journal of Cybernetics 3, 32-57 (1974).
- [9] <http://www.dabi.temple.edu/shape/mpeg7/dataset.html>.
- [10] <http://www.cs.columbia.edu/cave/software/softlib/coil-20.php>.
- [11] H. Zhang, H.Z. Shu, G.N. Han, G. Coatrieux, L.M. Luo, J.L. Coatrieux, Blurred image recognition by legendre moment invariants, Image Processing, IEEE Transactions on Image Processing 19, 596-611 (2010).
- [12] H. Zhu, Image representation using separable two-dimensional continuous and discrete orthogonal moments, Pattern Recognition 45, 1540-1558 (2012).
- [13] C. Chong, R. Caramesran, R. Cukundan, Translation and scale invariants of legendre moments, Pattern Recognition 37, 119-129 (2004).
- [14] <http://www.cs.columbia.edu/cave/software/softlib/coil-100.php>.

- [15] <http://www.cl.cam.ac.uk/research/dtg/attarchive:pub/data/att-faces.tar.z>.
- [16] M.K. Hosny, M.M. Darwish, New Set of Multi-Channel Orthogonal Moments for Color Image Representation and Recognition, *Pattern Recognition*, 88, 153-173 (2019).
- [17] K.M. Hosny, M.M. Darwish, New Set of Quaternion Moments for Color Images Representation and Recognition, *Journal of Mathematical Imaging and Vision* 60, 717-736 (2018).
- [18] K.M. Hosny, M.M. Darwish, Invariant color images representation using accurate quaternion Legendre-Fourier moments, *Pattern Anal Applic* 22, 1105-1122 (2019).
- [19] K.M. Hosny, M.M. Darwish, Robust color image watermarking using invariant quaternion Legendre-Fourier moments, *Multimed Tools Appl* 77, 24727-24750 (2018).
- [20] B. Xiao, J. Cui, H. Qin, W. Li, G. Wang, Moments and moment invariants in the Radon space. *Pattern Recognition* 48, 2772-2784 (2015).
- [21] B. Xiao, L. Li, Y. Li, W. Li, G. Wang, Image analysis by fractional-order orthogonal moments. *Information Sciences* 382, 135-149 (2017).
- [22] B. Chen, M. Yu, Q. Su, Fractional quaternion Zernike moments for robust color image copy-move forgery detection, *IEEE Access* 6, 56637-56646 (2018).
- [23] B. Xiao G. Wang W. Li, Radial shifted Legendre moments for image analysis and invariant image recognition, *Image and Vision Computing* 32, 994-1006 (2014).
- [24] S.O. Belkasim, M. Shridhar, M. Ahmadi, Pattern recognition with moment invariants : a comparative study and new results, *Pattern Recognition* 24, 1117-1138 (1991).
- [25] J. Flusser, T. Suk, A moment-based approach to registration of images with affine geometric distortion, *IEEE Trans. Geosci. Remote Sensing* 32, 382-387 (1994).
- [26] F. El-Khaly, M.A. Sid-Ahmed, Machine recognition of optically captured machine printed arabic text, *Pattern Recognition* 23, 1207-1214 (1990).
- [27] K. Tsirikolias, B.G. Mertzios, Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition, *Pattern Recognition* 26, 877-882 (1993).
- [28] A. Khotanzad, Y.H. Hong, Invariant image recognition by Zernike moments, *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 489-497 (1990).
- [29] J. Flusser, T. Suk, Affine moment invariants : A new tool for character recognition, *Pattern Recognition Lett.* 15, 433-436 (1994).
- [30] P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 629-639 (1990).

- [31] F. Catte, P.L. Lions, J.M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, *SIAM J. Numer. Anal.* 29, 182-193 (1992).
- [32] A. Handlovicova, K. Mikula, F. Sgallari, Variational Numerical Methods for Solving Nonlinear Diffusion Equations Arising in Image Processing, *Journal of Visual Communication and Image Representation* 13, 217-237 (2002).
- [33] L. Alvarez, P.L. Lions, J.M. Morel, Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numer. Anal.* 29, 845-866 (1992).
- [34] E. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, (1991).
- [35] R. Stenberg, A technique for analysing finite element methods for viscous incompressible flow, *international journal for numerical methods in fluids* 11, 935-948 (1990).
- [36] R. Aboulaich, D. Meskine and A. Souissi, New diffusion models in image processing, *Computers and Mathematics with Applications* 56, 874-882 (2008).
- [37] K. Mikula, N. Ramarosy, Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing, *Numer. Math.* 89, 561-590 (2001).
- [38] L. Alvarez, J.M. Morel, Formalization and computational aspects of image analysis, *Acta Numer.* 3, 1-59 (1994).
- [39] T. Preusser, M. Rumpf, An Adaptive Finite Element Method for Large Scale Image Processing, *Journal of Visual Communication and Image Representation* 11, 183-195 (2000).
- [40] S. Osher and J. Sethian, Front propagating with curvature dependent speed : Algorithms based on the Hamilton- Jacobi formulation, *J. Comput. Phys.* 79, 12-49 (1988).
- [41] J.A. Sethian, Numerical algorithm for propagating interfaces : Hamilton-Jacobi equations and conservation laws, *J. Differential Geom.* 31, 131-161 (1990).
- [42] G. Dziuk, Convergence of a semidiscrete scheme for the curve shortening flow, *Math. Models Methods Appl. Sci.* 4, 589-606 (1994).
- [43] G. Dziuk, Algorithm for evolutionary surfaces, *Numer. Math.* 58, 603-611 (1991).
- [44] G. Sapiro, A. Tannenbaum, On affine plane curve evolution, *J. Funct. Anal.* 119, 79-120 (1994).
- [45] N.J. Walkington, Algorithms for computing motion by mean curvature, *SIAM J. Numer. Anal.* 33, 2215-2238 (1996).
- [46] F. Brezzi and R. Falk, Stability of higher order Taylor-Hood methods, *SIAM J. Numer. Anal.* 28, 581-590 (1991).

- [47] S.C. Brenner, L.R. Scott, The mathematical theory of finite element methods, Texts in Applied Mathematics. Springer-Verlag, New York, second edition, 15, (2002).
- [48] C. Wieners, Taylor-Hood elements in 3D , Analysis and Simulation of Multifield Problems , Springer-Verlag Berlin Heidelberg, 12, 189-196 (2003).
- [49] A. Handlovičová, K. Mikula, A. Sarti, Numerical solution of parabolic equations related to level set formulation of mean curvature flow, Comput. Visual. Sci. 1, 179-182 (1999).
- [50] J. Kačur, K. Mikula, Solution of nonlinear diffusion appearing in image smoothing and edge detection, Appl. Numer. Math. 17, 47-59 (1995).
- [51] R. Eymard, T. Gallouet, R. Herbin, Finite volume method, Handbook for Numerical Analysis 7, 713-1018 (2000).
- [52] E. Bansch and K. Mikula, Adaptivity in 3D image processing, Comput Visual Sci. 4, 21-30 (2001).
- [53] T. Preußer and M. Rumpf, An Adaptive Finite Element Method for Large Scale Image Processing, Journal of Visual Communication and Image Representation 11, 183-195 (2000)
- [54] A. Sarti, K. Mikula, F. Sgallari, Nonlinear multiscale analysis of 3D echocardiographic sequences, IEEE Trans. Medical Imaging 18, 453-466 (1999).
- [55] S.C Kothari, Oh Heekuck, Neural Networks for Pattern Recognition, Advances in Computers 37, 119-166 (1993)
- [56] M. Abramowitz, I.A. Stegun, I. Stegun, Handbook of mathematical functions (with formulas, graphs, mathematical tables), Dover Publications, New York, (1970).
- [57] Y.S. Abu-Mostafa, D. Psaltis, Recognitive aspects of moment invariants, IEEE Transactions on Pattern Analysis-Machine Intelligence 6, 698-706 (1984).
- [58] Y.S. Abu-Mostafa, D. Psaltis, Image normalization by complex moments, IEEE Transactions on Pattern Analysis, Machine Intelligence 7, 46-55 (1985).
- [59] S. Adam, J.M. Ogier, C. Cariou, J. Gardes, Y. Lecourtier, Combination of Invariant Pattern Recognition Primitive on Technical Documents, Graphics Recognition (GREC), (1999).
- [60] M. Aly, P. Welinder, M. Munich, P. Perona, Automatic discovery of image families : Global vs. local features, International Conference on Image Processing ICIP'09, (2009).
- [61] G. Amayeh, A. Erol, G. Bebis, M. Nicolescu, Accurate and efficient computation of high order Zernike moments, Dans ISVC, 3804, 462-469 (2005).

- [62] V. Aquilanti, S. Cavalli, C. Coletti, Angular, hyperangular momentum recoupling, harmonic superposition, Racah polynomial : a recursive algorithm, *Chem. Phys. Lett.* 344, 587-600 (2001).
- [63] J.P. Asselin, F.Z. Kettaf, Bases théoriques pour l'apprentissage et la décision en reconnaissance des formes, Cepaduès, (2005).
- [64] J. Bach, C.Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, C. Shu, The Virage image search engine : an open framework for image management. Number 2670. *Storage, Retrieval for Image, Video Databases*, (1996).
- [65] S.O. Belkasim, M. Shridhar, et M. Ahmadi. Pattern recognition with moment invariants : A comparative study, new results. *Pattern Recognition*, 24, 1117-1138 (1991).
- [66] S.O. Belkasim, M. Ahmadi, M. Shridhar, Efficient algorithm for fast computation of Zernike moments, *J. Franklin Institute Eng. Appl. Math.* 33, 577-581 (1996).
- [67] A. Del Bimbo, W-X. He, E. Vicario, Using weighted spatial relationships in retrieval by visual contents. In : Vicario E. (eds) *Image Description and Retrieval. Advances in Computer Vision and Machine Intelligence*. Springer, Boston, MA, 161-192 (1998).
- [68] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Saux, H. Sahbi, Interactive generic and specific image retrieval, In *International Workshop on Multimedia Content-Based Indexing, Retrieval MMCBIR'01*, (2001).
- [69] A.M. Bronstein, M.M. Bronstein, R. Kimmel, Three-Dimensional Face Recognition. *International Journal of Computer Vision* 64, 5-30 (2005).
- [70] R. Brunelli, O. Mich, Compass : an image retrieval system for distributed databases, In *IEEE International Conference on Multimedia, Expo ICME'00*, (2000).
- [71] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, J. Malik, Blobworld : A system for region-based image indexing and retrieval, In *International Conference on Visual Information System VIS'99*, (1999).
- [72] M.E. Celebi, Y.A. Aslandogan, A comparative study of tree moment-based shape description, proceeding of *ITCC, Las Vegas, Nevada, USA* 1, 788-793 (2005).
- [73] N. Chaikla, Y. Qi, Genetic algorithms in feature selection, *Systems, Man, Cybematics, IEEE SMC Conference Proceedings* 5, 538-540, (1999).
- [74] W.A. Al-Salam, Characterization theorems for orthogonal polynomials, in : *Orthogonal polynomials : theory and practice*, P. Nevai (ed.), *NATO ASI Series C*, Kluwer, 294, 1-24 (1990).
- [75] C.W. Chong, P. Raveendran, R. Mukundan, Translation scale invariants of Legendre moments, *Pattern Recognition* 37, 119-129 (2004).

- [76] M. Delalandre, E. Trupin, J.M. Ogier, Système de Reconnaissance Structurale de Symboles, Basé sur une Multi Représentation en Graphes de Régions, Conférence Internationales Francophone sur l'Ecrit et le Document (CIFED), (2004).
- [77] C. Deng, X. Gao, X. Li, D. Tao, A local Tchebichef Moments-based robust image watermarking, *Signal Process* 89, 1531-1539 (2009).
- [78] C.F. Dunkl, Y. Xu, *Orthogonal Polynomials of Several Variables*, Cambridge University Press, (2001).
- [79] C.W. Chong, P. Raveendran, R. Mukundan, Translation invariants of Zernike moments, *Pattern Recognition* 36, 1765-1773 (2003).
- [80] L. Fernandez, T.E. Perez, M.A. Pinar, Orthogonal polynomials in two variables as solutions of higher order partial differential equations, *Journal of Approximation Theory* 163, 84-97 (2011).
- [81] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image video content : the qbic system. *IEEE Computer* 28, 23-32 (1995).
- [82] J. Flusser, T. Suk, Pattern recognition by affine moment invariants. *Pattern Recognition* 26, 167-174 (1993).
- [83] J. Freeman, D. Skapura, *Neural networks algorithms applications and programming techniques*. Reading, MA : Addison-Wesley, (1991).
- [84] T. Gevers, A. Smeulders, The pictoseek www image search system, *International Conference on Mathematics Computer Science* 1, 264-269 (1999).
- [85] L. Heutte, T. Paquet, J. V. Moreau, Y. Lecourtier, C. Olivier, A structural/statistical feature based vector for handwritten character recognition, *Pattern recognition letters* 19, 629-641 (1998).
- [86] G.B. Wang, S.G. Wang, Recursive computation of Tchebichef moment and its inverse transform, *Pattern Recognit.* 39, 47-56 (2006).
- [87] K.M. Hosny, Fast computation of accurate Gaussian-Hermite moments for image processing applications, *Digital Signal Processing* 22, 476-485 (2012).
- [88] H. Shu, H. Zhang, C. Beijing, P. Haigron, L. Luo, Fast computation of Tchebichef moments for binary and grayscale images, *IEEE Trans Image Process.* 19, 3171-3180 (2010).
- [89] A. Hmimid, M. Sayyouri, H. Qjidaa, Image classification using a new set of separable two-dimensional discrete orthogonal invariant moments, *J. Electron Imaging.* 23, 013-026 (2014).

- [90] A. Hmimid, M. Sayyouri, H. Qjidaa, Image classification using novel set of charlier moment invariants, *WSEAS Transactions on Signal Processing*, 10, 156-167 (2014).
- [91] K.M. Hosny, Exact fast computation of geometric moments for gray level images, *Appl. Math. Comput.* 189, 1214-1222 (2007).
- [92] K.M. Hosny, Exact Legendre moment computation for gray level images. *Pattern Recognition* 40, 3597-3605 (2007).
- [93] K.M. Hosny, Image representation using accurate orthogonal Gegenbauer moments, *Pattern Recognition Letters*, 32, 795-804 (2011).
- [94] K.M. Hosny, Fast computation of accurate Gaussian-Hermite moments for image processing applications, *Digital Signal Processing*, 22, 476-485 (2012).
- [95] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition : A review, *IEEE Trans. on PAMI*, 22, 4-37 (2000).
- [96] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall Information Series, (1989).
- [97] R. Jerome, Improving Zernike Moments Comparison for Optimal Similarity Rotation Angle Retrieval, *IEEE Transactions on Pattern Analysis Machine Intelligence* 31, 627-636 (2009).
- [98] A. Wallin, O. Kubler, Complete sets of complex Zernike moment invariants the role of the pseudo invariants, *IEEE Transactions Pattern Analysis Machine Intelligence*, 17, 1106-1110 (1995).
- [99] C-Y. Wee R. Paramesran. Efficient computation of radial moment functions using symmetrical property. *Pattern Recognition* 39, 2036-2046 (2006).
- [100] C-Y. Wee, R. Paramesran, On the computational aspects of Zernike moments, *Image Vision Comput.* 25, 967-980 (2007).
- [101] S. Weiss, Small sample error rate estimation for k-NN classifiers, *IEEE Transactions on Pattern Analysis Machine Intelligence*, 13, 285-289 (1991).
- [102] J. Weston, C. Watkins, Multi-class support vector machines, In *European Symposium on Artificial Neural Networks ESANN'99*, (1999).
- [103] Y. Wu, J. Shen, Properties of orthogonal Gaussian-Hermite moments their applications, *EURASIP Journal on Applied Signal Processing*, 4, 588-599 (2005).
- [104] P. Xu, A. Chan, Support vector machine for multi-class signal classification with unbalanced samples, In *International Joint Conference on Neural Networks*, 1116-1119, (2003).

- [105] Y. Xu, Second order difference equations and discrete orthogonal polynomials of two variables, *International Mathematics Research Notices*, 449-475 (2005).
- [106] B. Yang, M. Dai, Image analysis by Gaussian-Hermite moments. *Signal Processing* 91, 2290-2303 (2011).
- [107] B. Yang, M. Dai, Image reconstruction from continuous Gaussian-Hermite moments implemented by discrete algorithm. *Pattern Recognition* 45, 1602-1616 (2012).
- [108] J. Yang, V. Honavar, *Feature Subset Selection Using A Genetic Algorithm*, the Springer International Series in Engineering and Computer Science, 453, 117-136 (1998).
- [109] B. Yang, G. Li, H. Zhang, M. Dai, Rotation translation invariants of Gaussian-Hermite moments, *Pattern Recognition Letters* 32, 1283-1298 (2011).
- [110] P.T. Yap, R. Paramesran, S.H. Ong, Image analysis by Krawtchouk moments *IEEE Trans Image Processing* 12, 1367-1377 (2003).
- [111] P.T. Yap, P. Raveendran, S.H. Ong, Image analysis using Hahn moments, *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 2057-2062, (2007).
- [112] N. Yokoya, M.D. Levine, Range image segmentation based on differential geometry : a hybrid approach, *IEEE Transactions on Pattern Analysis Machine Intelligence* 11, 643-649 (1989).
- [113] F. Zernike, Beugungs theorie des Schneiden verfahrens und seiner verbesserten Form, der Phasenkontrastmethode. *Physica* 1, 689-704 (1934).
- [114] H. Zhang, H.Z. Shu, P. Haignon, B.S. Li, L.M. Luo, Construction of a complete set of orthogonal Fourier-Mellin moment invariants for pattern recognition applications, *Image Vision Computing*, 28, 38-44, (2010).
- [115] A. Hjouji, J. EL-Mekkaoui, M. Jourhmane, B. Bouikhalene, New Set of Non-separable Orthogonal Invariant Moments for Image Recognition, *J Math Imaging Vis.* 62, 606-624 (2020).
- [116] A. Hjouji, J. EL-Mekkaoui, M. Jourhmane, Rotation scaling and translation invariants by a remediation of Hu's invariant moments, *Multimed Tools Appl.* 79, 14225-14263 (2020).
- [117] A. Hjouji, M. Jourhmane, J. EL-Mekkaoui, M. Es-sabry, Mixed Finite Element Approximation for Bivariate Perona-Malik Model Arising in 2D and 3D Image Denoising, *3D Res* 9, 36 (2018).
- [118] A.Hjouji, J.EL-Mekkaoui, M.Jourhmane, Mixed Finite Element Method for Nonlinear Diffusion Equation in Image Processing, *Pattern Recognition and Image Analysis*, 29, 296-308 (2019).

- [119] A. Hjouji, J. EL-Mekkaoui, M. Jourhmane, H. Qjidaa, B. Bouikhalene, Image retrieval and classification using shifted Legendre invariant moments and Radial Basis Functions Neural Networks, *Procedia Computer Science* 148, 154-163 (2019).
- [120] A. Hjouji, M. Jourhmane, J. EL-Mekkaoui, H. Qjidaa, B. Bouikhalene, A. ELKhalfi, Image Clustering Based On Hermetian positive definite matrix And Radial Jacobi moments, 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), 2-4, (2018).
- [121] Y. Saad, M.H. Schultz, GMRES : a generalized minimal residual algorithm for solving non symmetric linear systems, *SIAM J. Sci. and Stat. Comput.*, 7, 856-869 (1986).
- [122] H.A. Vander Vorst. Bi-CGStab : Bi-CGSTAB : A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems *SIAM J. Sci. and Stat. Comput.*, 13, 631-644 (1992) .
- [123] V. Girault, P.A. Raviart, *Finite Element Methods for Navier-Stokes Equations*. Springer-Verlag, Berlin, (1986).
- [124] V. Girault, P.A. Raviart, *Finite Element Method for Navier-Stokes Equations : Theory and Algorithms*. Springer-Verlag, Berlin and Heidelberg, (1987).
- [125] R. Stenberg, A technique for analysing finite element methods for viscous incompressible flow, *Int. J. Numer. Methods Fluids* 11, 935-948 (1990).
- [126] C.H. Teh, R.T. Chin, On Image Analysis by the Methods of Moments. *IEEE Transactions on Pattern Analysis ; Machine Intelligence* 10, 496-513 (1988).
- [127] G.A. Papakostas, E.G Karakasis, D.E. Koulouriotis , Novel moment invariants for improved classification performance in computer vision applications, *Pattern Recognition* 43, 58-68 (2010).
- [128] Z. Ping, H. Ren, J. Zou, Y. Sheng, and W. Bo. Generic orthogonal moments : Jacobi-Fourier moments for invariant image description. *Pattern Recognition* 40, 1245-1254 (2007).
- [129] C. Camacho-Bello, C. Toxqui-Quitl, A. Padilla-Vivanco, J. Báez-Rojas, Highprecision and fast computation of Jacobi-Fourier moments for image description, *J. Opt. Soc. Am.* 31, 124-134 (2014).
- [130] T. Suk, J. Flusser, Affine moment invariants of color images, *The 13th International Conference on Computer Analysis of Images and Patterns, Lecture Notes Computer Science*, 5702, Münste, Germany, 334-341 (2009).

- [131] X.Y Wang, W.Y Li, H. Yang, P. Wang, Y-W. Li, Quaternion polar complex exponential transform for invariant color image description, *Applied Mathematics and Computation*, 256, 951-967 (2015).
- [132] K.M. Hosny, M.M. Darwish, Comments on robust circularly orthogonal moment based on Chebyshev rational function, *Digital Signal Processing*, 62, 249-258, vol. 2017., *Digital Signal Processing*, 68, 152-153 (2017).
- [133] B. Xiao, G. Wang, W. Li, Radial Shifted Legendre Moments for Image Analysis and Invariant Image Recognition, *Image and Vision Computing* 32, 994-1006 (2014).
- [134] R. Benouini, I. Batioua, K. Zenkouar, A. Zahi, S. Najah, H. Qjidaa, Fractional-order orthogonal Chebyshev Moments and Moment Invariants for image representation and pattern recognition, *Pattern Recognition*, 86, 332-343 (2019).
- [135] B. Xiao, J. Luo, X. Bi, W. Li, B. Chen, Fractional discrete Tchebyshev moments and their applications in image encryption and watermarking, *Information Sciences*, 516, 545-559 (2020).
- [136] M. Yamni, A. Daoui, O. El ogri, H. Karmouni, M. Sayyouri, H. Qjidaaa, J. Flusser, Fractional Charlier moments for image reconstruction and image watermarking, *Signal Processing*, 171, Article No. 107509 (2020).
- [137] R. Krishnapuram, J.M Keller, The possibilistic c-means algorithm : insights and recommendations, *IEEE transactions on Fuzzy Systems*, 4, 385-393 (1996).
- [138] D. Pascale, A review of rgb color spaces from xyy to rgb, the BabelColor company, (2003).
- [139] A. Bimbo, E. Vicario, Using weighted spatial relationships in retrieval by visual contents, In *IEEE Workshop on Image Video Libraries*, 1998.
- [140] B. Manjunath, J. Ohm, V. Vasudevan, A. Yamada, Color texture descriptors, In *IEEE Transactions on circuits systems for video technology*, 11, 703-715 (2001).
- [141] S. Livens, P. Scheunders, G. Van-Wouwer, D. Van-Dyck, Wavelets for texture analysis, an overview. In *Image Processing Its Applications*, 2, 581-585 (1997).
- [142] H. Zhu, Image representation using separable two- dimensional continuous and discrete orthogonal moments, *Pattern Recogn* 45, 1540-1558 (2012).
- [143] <http://www.dabi.temple.edu/shape/mpeg7/dataset.html>
- [144] <http://www.cs.columbia.edu/cave/software/softlib/coil-100.php>
- [145] <http://www.cl.cam.ac.uk/research/dtg/attarchive:pub/data/att-faces.tar.z>
- [146] <https://www.kaggle.com/zalandoresearch/fashionmnist> fashion-mnist test.csv

- [147] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc IEEE 86, 2278-2344 (1998).