



Université Lille1 - Sciences et Technologies
Master 2 Informatique spécialité E-Services
Rapport de stage de recherche

Analyse et régularisation spatio-temporelle : application à l'écriture manuscrite

Erika Pillu

Maître de stage : Youssef Chahir <youssef.chahir@unicaen.fr>

Tuteur universitaire : Luigi Lancieri <luigi.lancieri@univ-lille1.fr>

GREYC - Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen - UMR 6072

Campus Côte de Nacre, Boulevard du Maréchal Juin

BP 5186 - 14032 Caen CEDEX

Tél : +33 (0)2 31 56 74 86

Fax : +33 (0)2 31 56 73 30

Site web : www.greyc.fr

Contact : <contact@greyc.fr>

Caen, le 28 août 2011

Résumé

Les images et les vidéos numériques sont des données de grande dimension. Extraire de l'information à partir de ces données est une problématique importante lorsque l'on souhaite traiter automatiquement leur contenu. La méthode que nous utilisons s'applique sur des images (2d) et des séquences d'images (2d+t). Dans un premier temps, nous cherchons à réduire la dimension des données en les représentant par des descripteurs de forme (1d). Ensuite, nous calculons la distance entre ces vecteurs afin de trouver des similarités entre les images. Pour cela, nous utilisons plusieurs caractéristiques spatio-temporelles comme les moments géométriques 2D et 3D. Les résultats que nous avons obtenus, basés sur les moments invariants de Krawtchouk et de Zernike, sont très prometteurs. Dans un cadre plus général, cette approche peut permettre d'utiliser un graphe de similarités modélisant l'ensemble des images puis d'exploiter les propriétés spectrales du graphe.

Mots-clés : moments géométriques, catégorisation, réduction de dimension, analyse spatio-temporelle

Table des matières

| | | |
|----------|-------------------------------------|-----------|
| 1 | Introduction | 1 |
| 1.1 | Contexte | 1 |
| 1.2 | Problématique | 2 |
| 1.3 | Cadre applicatif | 3 |
| 1.4 | Objectifs du stage | 4 |
| 1.5 | Organisation du mémoire | 4 |
| 2 | Le traitement d'images | 5 |
| 2.1 | Introduction | 5 |
| 2.2 | Images binaires | 8 |
| 2.3 | Images en niveaux de gris | 10 |
| 2.4 | De l'image à la vidéo | 11 |
| 2.5 | Conclusion | 12 |
| 3 | Approche proposée | 13 |
| 3.1 | Descripteurs de forme | 13 |
| 3.2 | Les moments de Zernike | 15 |
| 3.3 | Les moments de Krawtchouk | 18 |
| 3.4 | Classification | 21 |
| 3.5 | Conclusion | 21 |
| 4 | Implémentation et Résultats | 23 |
| 4.1 | Outils utilisés | 23 |
| 4.2 | Traitements basiques | 25 |
| 4.3 | Les moments de Zernike | 30 |
| 4.4 | Les moments de Krawtchouk | 33 |
| 4.5 | Conclusion | 33 |
| 5 | Conclusion et perspectives | 34 |

Chapitre 1

Introduction

1.1 Contexte

Dans le cadre de mon cursus universitaire en Master 2 Informatique spécialité E-Services de l'Université de Lille1 Sciences et Technologies, j'ai décidé d'effectuer un stage orienté recherche couvrant la période du 21 mars au 21 septembre 2011 au sein du laboratoire GREYC¹ (Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen) de l'Université de Caen Basse-Normandie (14).

Le GREYC est un laboratoire de recherche associé au Centre national de la recherche scientifique (CNRS). Ses activités sont centrées sur le domaine des Sciences et technologies de l'information et de la communication (STIC). Les thématiques étudiées sont l'informatique, l'intelligence artificielle, le traitement du signal, le traitement d'image, la sécurité informatique, l'électronique et les dispositifs. L'équipe dans laquelle se déroule mon stage est l'équipe Image qui travaille sur le développement de nouvelles méthodes de traitement et d'analyse d'images, confrontées à la résolution de problèmes concrets. Ses deux champs d'action sont la mise en oeuvre de modèles de représentation d'images d'une part et l'extraction et la gestion des connaissances en traitement et analyse d'images d'autre part.

De plus en plus de disciplines de recherche travaillent ensemble pour résoudre des problèmes complexes et c'est le cas tout particulièrement du traitement des images et des vidéos numériques. Ce domaine intervient dans de très nombreuses branches scientifiques et d'ingénieurs avec par exemple le traitement des images spatiales ou l'imagerie médicale. Le traitement d'images suscite également l'intérêt d'autres communautés scientifiques telles que la psychologie, la biologie, la cognition ou les sciences humaines en général.

1. www.greyc.fr

Elle permet de résoudre de nombreux problèmes et de réaliser des applications informatiques dans ce contexte. C'est dans cette optique que s'inscrit mon travail de stage encadré par Y. Chahir, enseignant-chercheur de l'équipe Image du laboratoire GREYC, avec la collaboration des laboratoires PALM² (Psychologie des Actions Langagières et Motrices) et CHART³ (Cognitions Humaine & ARTificielle).

1.2 Problématique

Avec le développement des nouvelles technologies, il est désormais possible d'informatiser et de traiter un grand nombre de données. Ces données peuvent être de nature très variée et sont utilisées dans de nombreux domaines comme la biologie, la médecine, la finance, le marketing, les sciences, les ressources humaines, etc. Par exemple dans la branche de la gestion de la relation client, les habitudes et le comportement des consommateurs sont collectés et analysés afin d'établir des profils types et de personnaliser les services à leur proposer. Une démarche d'exploration et d'analyse classique d'une grande quantité de données consiste à en extraire de l'information en utilisant des méthodes automatiques ou semi-automatiques. Ce procédé permet non seulement de constater et d'expliquer des faits, mais aussi de les classer et de les prévoir.

L'image et la vidéo numérique sont des sources très abondantes en informations. Si l'image reflète notre sens de la vue, la vidéo reproduit notre expérience quotidienne puisqu'elle capture les changements et les mouvements des objets. Elle contient donc encore davantage d'informations que l'image. Toutefois, l'image et la vidéo sont des signaux multi-dimensionnels (généralement 2d pour l'image et 2d+t, c'est-à-dire 3d, pour la vidéo). Ils sont donc rapidement coûteux en ressources, en calculs et en temps, particulièrement dans le cas de la vidéo, alors que la plupart des applications qui les utilisent doivent être exécutées en temps réel. Cela veut dire que les résultats des traitements apparaissent instantanément (ou perçu comme tel) à l'entrée des données. Même si les performances des machines et des ordinateurs ne cessent d'être améliorées, il est nécessaire de prendre en compte ces contraintes qui vont influencer les structures mathématiques et algorithmiques.

La problématique qui nous intéresse appartient au domaine de la reconnaissance, de la vision par ordinateur et de l'intelligence artificielle. Elle consiste dans un premier temps à sélectionner les caractéristiques les plus pertinentes parmi toutes les mesures que l'on peut effectuer sur les formes.

2. www.unicaen.fr/recherche/mrsh/palm

3. www.cognition-usages.org/chart/

Ces descripteurs de formes, ou vecteurs caractéristiques, permettent de réduire de façon significative les données de grande dimension en minimisant la perte d'information. Ils autorisent aussi des calculs ou des traitements complexes par la machine qui ne seraient pas possibles sur les données d'origine. Dans un second temps, on peut établir des comparaisons entre plusieurs descripteurs de formes. Cela permet de regrouper ou de discriminer des objets entre eux et de les classer en plusieurs catégories. Cette méthode est largement utilisée dans le domaine de l'imagerie médicale mais également dans celui de la biométrie, qui permet de reconnaître une personne en fonction d'une ou de plusieurs caractéristiques physiologiques ou comportementales, sur lesquelles reposent par exemple la reconnaissance humaine faciale. Parallèlement à la recherche en biologie et en psychologie qui continue d'étudier la façon dont le système visuel humain fonctionne et la façon dont on reconnaît les objets ou les personnes, la recherche et l'application de la vision par ordinateur vise à reproduire ce principe.

1.3 Cadre applicatif

Dans le cadre de la problématique annoncée dans 1.2, les applications possibles de cette recherche intéressent particulièrement M. Molina et F. Jouen, des laboratoires PALM et CHART. En effet, l'un des projets de recherche en cours est de réaliser une interface multimodale d'aide à la production de l'écriture. Le système, destiné à des enfants en phase d'apprentissage de l'écriture, permettrait de dépister des dysfonctionnements neuromoteurs⁴ et d'y remédier via une interface informatisée. L'idée est que l'application donne un retour en temps réel à l'apprenant sur la qualité de son tracé des lettres alphanumériques. L'évaluation de l'écriture peut par exemple s'appuyer sur le test du BHK [8] dans lequel les critères sont basés sur la vitesse de l'écriture, les tremblements et les hésitations, la grandeur de l'écriture, la retouche des lettres, la non-planarité des lignes d'écriture, l'ambiguïté de la forme des lettres, etc.

On peut considérer ce projet comme le cas d'étude qui va nous guider à travers ce travail de stage. Les exemples et les expérimentations seront donc généralement appliqués à des images de caractères manuscrits.

4. Par exemple : la dysgraphie (trouble dans l'apprentissage de l'écriture) et la dyspraxie (trouble dans l'acquisition de la coordination du geste).

1.4 Objectifs du stage

Les objectifs du stage sont multiples. Il s'agit dans un premier temps de s'imprégner des connaissances actuelles et des techniques liées au traitement d'images. Ensuite, en vue d'une initiation à la recherche, on réfléchira aux problématiques actuelles de recherche dans ce domaine et on expérimentera certaines idées qui émergent des publications récentes. Parallèlement à ce travail, une réflexion de fond sera menée pour concevoir le cadre applicatif abordé dans 1.3.

1.5 Organisation du mémoire

Dans le Chapitre 2, nous introduisons les notions importantes liées au traitement d'images. Cette partie dresse un rapide état de l'art en matière de traitement d'images et met l'accent sur les concepts utiles pour traiter de notre problématique. Le Chapitre 3 développe l'approche que nous avons utilisée pour résoudre le problème et qui repose principalement sur les moments invariants de Zernike et de Krawtchouk en 2d et 3d. L'implémentation de la solution et les résultats que nous avons obtenus seront discutés dans le Chapitre 4. Enfin, la conclusion donnera le bilan et les perspectives que l'on peut envisager suite à cette étude et ce travail de stage.

Chapitre 2

Le traitement d'images

2.1 Introduction

2.1.1 Le système visuel

L'être humain possède cinq sens - la vue, l'ouïe, l'odorat, le toucher et le goût - qui lui permettent de percevoir consciemment et d'analyser des objets ou des phénomènes extérieures. Les différents systèmes sensoriels de l'homme sont fortement liés à sa capacité de motricité, c'est-à-dire "l'ensemble des fonctions biologiques qui assurent le mouvement chez l'être humain et les animaux"¹. En effet, le système sensori-moteur est ce qui confère à l'être humain la faculté de réaliser (phase de préparation) et de contrôler (phase d'exécution) une succession d'actions motrices planifiées [33].

Si le système auditif est considéré comme le plus discriminant [29], la vision reste la source d'informations la plus riche et la plus importante que nous recevons du monde extérieur. Au niveau perceptif, la vision permet d'interpréter des images et d'identifier des formes. À un niveau plus complexe, la vision implique la catégorisation, la mémorisation et l'attention. "L'image qui se forme au fond de l'œil n'est que le point de départ de la perception visuelle. Elle est codée sous forme d'impulsions électriques, transmises jusqu'aux aires visuelles cérébrales où elles sont analysées en termes de reconnaissance des formes, des couleurs, des mouvements, etc."²

2.1.2 Types d'images

Il est possible de différencier les images selon les interactions physiques entre la source de rayonnements et l'objet. Bovik distingue ainsi trois types

1. Extrait du Petit Larousse 2006.

2. Imbert, M. (1983) et (1984).

d'images [5] : les *images de réflexion*, les *images d'émission* et les *images d'absorption*. **Figure 2.1** illustre ces différents types d'images. Les images de réflexion proviennent de la réflexion des rayonnements sur la surface des objets. La majorité de ce que nous voyons provient de ce phénomène (voir **Fig. 2.1 (a)**). La lumière visible se réfléchit sur les objets avant d'atteindre notre œil. Parmi les exemples de réflexion de lumière non visible, on peut citer les images radar, sonar (**b**), laser ou issues des microscopes électroniques (**c**). Les images d'émission signifient que l'image provient d'objets qui sont eux-mêmes source de lumière, comme les étoiles ou bien les ampoules électriques. On peut ainsi obtenir des images thermiques ou infrarouges (**d**), très utilisées dans les applications médicales, spatiales ou militaires. Quant aux images d'absorption, elles indiquent que les rayonnements traversent l'objet qui en absorbe une certaine partie. On les utilise par exemple en radiographie médicale avec les rayons X (**e**). L'intérêt d'utiliser des images d'un type particulier est qu'elles diffusent des informations différentes sur les propriétés de l'objet. Tous ces exemples montrent bien l'importance et le potentiel du traitement d'images dans de nombreux domaines.

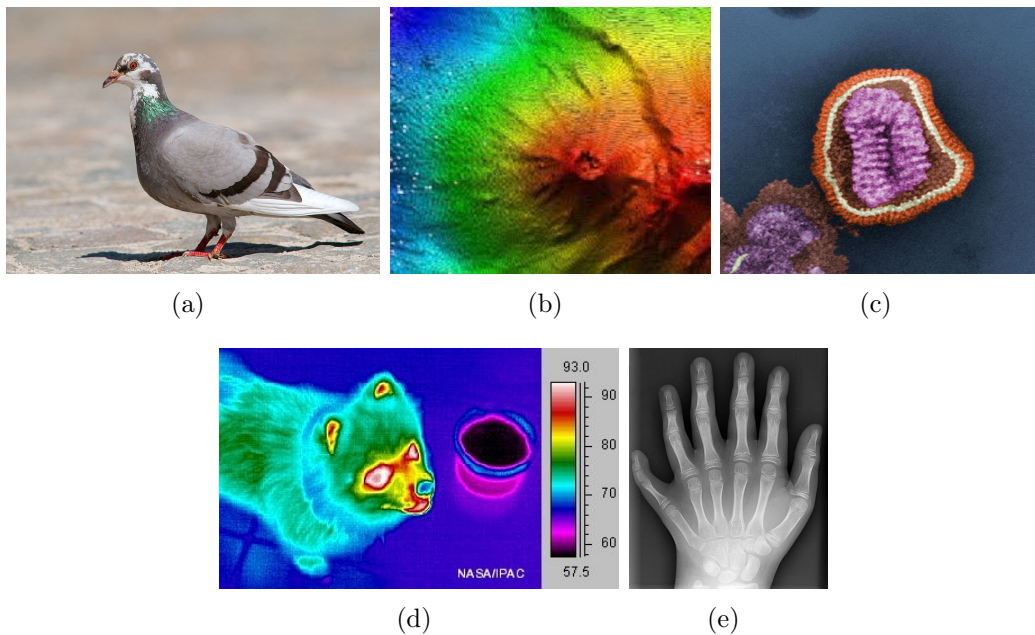


FIGURE 2.1 – Différents types d'images. (a) photographie (réflexion); (b) image sonar (réflexion); (c) image par microscopie électronique (réflexion); (d) image thermique ou infrarouge (émission); (e) radiographie (absorption).

2.1.3 Numérisation de l'image

Dans le domaine du traitement d'images, un aspect important à considérer est l'acquisition, ou plus précisément la *numérisation* de l'image. C'est le processus par lequel on transforme un signal (ou une observation d'un phénomène) du monde réel en une représentation accessible pour l'ordinateur, c'est-à-dire binaire. Dans le monde réel, les variables définissant le signal sont continues dans l'espace et dans le temps et peuvent prendre une infinité de valeurs, alors que dans le monde numérique il est nécessaire de discrétiser ces variables afin de leur accorder un nombre fini de valeurs possibles. C'est de façon plus générale le principe de conversion d'un signal analogique en signal numérique. Cela implique deux étapes : l'*échantillonnage* des variables spatio-temporelles et la *quantification* de la valeur que peuvent prendre ces variables. La dimension d'un signal est le nombre de variables à prendre en compte pour le décrire. En effet, un signal physique peut être monodimensionnel lorsque l'on a besoin uniquement d'une seule coordonnée pour indexer ses valeurs, par exemple la mesure de la température dans le temps. Dans le cas de l'image numérique, on retiendra deux ou trois coordonnées pour indexer la valeur d'un pixel selon que l'image est en 2d ou en 3d. Et dans le cas de la vidéo numérique, on ajoute une troisième (ou quatrième) coordonnée qui n'est autre que le temps. [Figure 2.2](#) illustre les trois types de signaux que nous venons de mentionner.

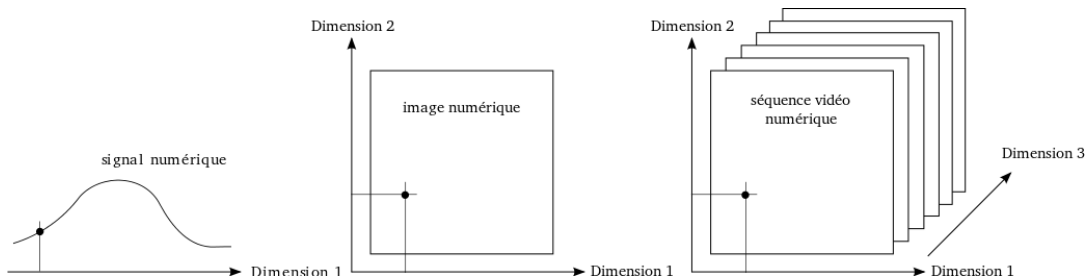


FIGURE 2.2 – Exemples de dimensions de différents signaux.

2.1.4 Espaces de couleur

Nous avons vu (dans [2.1.3](#)) que la valeur des pixels ne peut prendre qu'un nombre fini de possibilités. Si K est le nombre de valeurs autorisées, l'ensemble des possibilités est $\{0, \dots, K - 1\}$ où K est un entier positif puissance de 2 tel que $K = 2^B$ et B est le nombre de bits utilisés pour stocker sa valeur. Pour les images binaires, on a $K = 1$, c'est-à-dire que chaque pixel ne peut

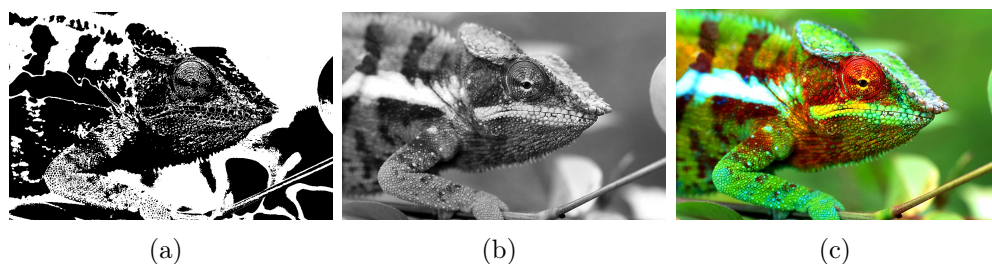


FIGURE 2.3 – Exemples de différents espaces de couleur. (a) image binaire; (b) image en niveaux de gris; (c) image en couleur.

être qu'à 0 ou à 1, par exemple noir ou blanc. Pour les images en niveaux de gris, on a $K = 256$, ce qui signifie que l'ensemble des valeurs de pixels s'étale sur $\{0, \dots, 255\}$ et qu'il faut $B = 8$ bits (ou 1 byte) pour stocker un pixel. Pour les images en couleurs, il est usuel de coder les pixels de chaque composante, comme rouge, vert et bleu, ce qui nécessite 24 bits par pixel. [Figure 2.3](#) illustre les trois configurations précédemment mentionnées.

2.2 Images binaires

De nombreux traitements peuvent être effectués sur les images numériques. On peut distinguer ces traitements selon qu'ils opèrent sur des images binaires, sur des images en niveaux de gris ou sur des images couleurs ou multispectrales. Nous n'aborderons que peu le cas des images en niveaux de gris ainsi que celui des images couleurs ou multispectrales dans ce document, mais on pourra trouver de nombreuses sources d'informations dans la littérature [5, 15, 30, 32]. En effet, les images qui nous intéressent pour notre problématique sont essentiellement binaires.

2.2.1 Définition

Comme nous l'avons vu dans la [Section 2.1.4](#), les images binaires sont composées de pixels qui ne peuvent prendre que la valeur 0 ou 1. On peut donc leur appliquer des opérations logiques booléennes (vrai ou faux), qui associées au fait que l'on n'ait besoin que d'un seul bit pour stocker chaque valeur, signifient que les traitements sur les images binaires sont particulièrement rapides. De plus, les appareils de type tablette électronique (voir par exemple [Fig. 2.4](#)), qui capturent les coordonnées du stylet en contact avec la surface, produisent déjà des images binaires. Par exemple, l'image est initialement

composée de pixels blancs que l'on assigne à 0 pour toute coordonnée n de l'image :

$$g(n) = 0 \quad (2.1)$$

Lorsqu'un changement (pression, résistance ou lumière différente) est détecté en un point n_0 , la valeur passe à 1 et on l'y associe un pixel noir.

$$g(n_0) = 1 \quad (2.2)$$

L'assignation de la couleur noire à la valeur 1 et de la couleur blanche à la valeur 0 n'est qu'une question de convention et la configuration inverse pourrait convenir indifféremment.

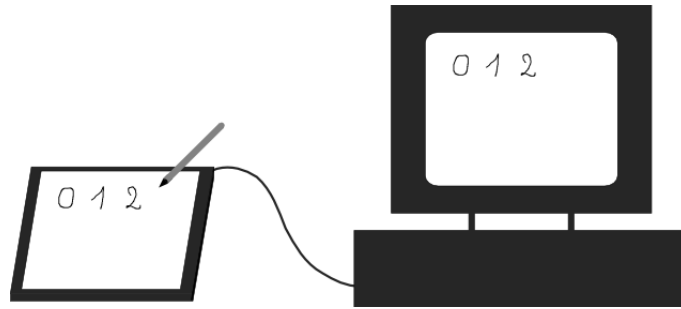


FIGURE 2.4 – Appareil d'acquisition d'image binaire.

2.2.2 Étiquetage de régions

Parmi les traitements intéressants sur les images binaires, on peut citer l'*étiquetage de régions* ou *identification de composantes connexes*. Cette méthode vise à segmenter l'image binaire f afin d'extraire et d'identifier des groupes de pixels connectés et de même valeur. La segmentation est le processus par lequel on partitionne l'image en plusieurs régions en rassemblant des pixels entre eux. Les différents objets de l'image peuvent ensuite être manipulés séparément. Une méthode classique consiste à segmenter l'image en fonction de son histogramme. Un algorithme de cette méthode est disponible dans [5] et une implémentation en C dans [25]. Il existe d'autres techniques de segmentation que l'on peut trouver par exemple dans l'article de Haralick et al. [13]. Figure 2.5 montre le résultat de l'algorithme d'identification des composantes connexes où chaque couleur identifie une composante connexe.



FIGURE 2.5 – Résultat de l’application de l’étiquetage de région. (a) image binarisée ; (b) image étiquetée.

2.2.3 Transformations morphologiques

Les images binaires permettent également d’appliquer des transformations morphologiques sur les formes. Il peut s’agir par exemple de les épaissir ou des les amincir, de les lisser, d’éliminer des objets trop petits, etc. Ce genre de traitements utilise des filtres morphologiques, tels que la dilatation, l’érosion, le filtre médian. Ces opérations permettent par exemple de remplir des trous laissés vacants dans une forme lors de la binarisation. Dans la même idée, on peut aussi détecter les frontières des formes puis les squelettiser.

2.3 Images en niveaux de gris

2.3.1 Moyenne et réduction de bruit

Du bruit peut apparaître sur les images numériques. On appelle bruit l’ensemble des éléments qui ne sont pas désirés dans une image. Il peut provenir par exemple du processus de numérisation. Pour traiter le bruit, on peut décomposer une image f en deux sous-images g et h , où g représente l’idéal (sans bruit) et h représente le bruit. Une décomposition fréquente pour le bruit est additive (on parle aussi de bruit gaussien) :

$$f = g + h \quad (2.3)$$

Dans ce cas, on peut réduire ce bruit en faisant la moyenne de plusieurs images à la condition qu’il ne doit pas y avoir de mouvement entre les images. On peut trouver la démonstration mathématique pour la réduction du bruit additif dans [5]. La théorie indique que plus on utilise d’images pour faire la moyenne, plus le bruit diminuera. Figure 2.6 montre le résultat de la moyenne

entre plusieurs échantillons représentant la lettre manuscrite 0 (le résultat apparaît à la droite de la séparation rouge). Si on considère la variabilité intra-classe³ comme du bruit, alors on voit que l'image obtenue est moins bruitée et fait émerger une lettre plus régulière. Ce résultat peut être intéressant pour construire des images de référence ou des modèles en considérant une grande quantité d'échantillons.



FIGURE 2.6 – Moyenne d'images représentant le chiffre manuscrit 0.

Cependant, pour débruiter une image, le modèle additif n'est pas toujours adéquat. Il existe d'autres types de bruits : les dégradations de type multiplicatif, les dégradations de type "poivre et sel" et les dégradations de type Poisson. Le choix d'un modèle spécifique de débruitage dépend de la configuration que l'on veut donner à l'image g et reste aujourd'hui encore un problème complexe. Pour plus d'informations, une étude des principaux algorithmes de débruitage d'images peut être trouvée dans [7].

2.3.2 Différence et détection du mouvement

On peut s'intéresser au mouvement qui existe dans une séquence d'images d'une même scène. De nombreuses applications découlent de la détection du mouvement, par exemple les algorithmes de compression de vidéos, de suivi de cible, de reconnaissance d'objets par le mouvement, etc [5]. Une façon classique de détecter le mouvement consiste à faire la différence de deux images de sorte que soient mis en évidence les pixels qui ont une valeur différente d'une image à l'autre. Figure 2.7 illustre la différence entre deux images représentant la lettre 1. De nombreux algorithmes de détection de mouvement découlent de ce principe et sont la base pour effectuer ensuite une estimation et un suivi (ou tracking) des objets en mouvement.

2.4 De l'image à la vidéo

Nous venons de voir qu'à partir de deux images on peut obtenir des informations comme l'estimation du mouvement. Les données qui nous intéressent

3. Dans la reconnaissance de formes, on appelle la *variabilité intra-classe* les différences qui apparaissent dans une même catégorie de formes, et la *variabilité inter-classe* les différences entre plusieurs catégories de formes.

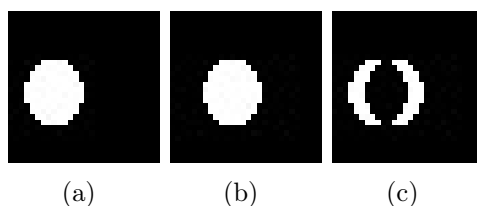


FIGURE 2.7 – Différence de deux images. Les pixels blancs dans le résultat (c) indiquent les changements entre l'image (a) et l'image (b).

pour le projet d'interface informatisée d'aide à l'apprentissage peuvent être vues comme des images qui varient dans le temps, ou plus simplement des vidéos. Ce sont des données spatio-temporelles en trois dimensions ($2d+t$) qui prennent la forme de la fonction $f(x, y, t)$. Dans notre cas d'étude, le signal f est échantillonné dans les trois dimensions (signal numérique). On peut donc appliquer sur ces données des algorithmes de traitement de vidéos pour traiter notre problématique. Les principaux algorithmes actuellement utilisés pour la détection, l'estimation et le suivi du mouvement sont basés sur le flot optique ou le block matching [6].

2.5 Conclusion

Dans ce chapitre nous avons vu les bases des techniques de traitement d'images numériques. Elles peuvent nous permettre de résoudre notre problématique en traitant automatiquement les images statiques d'écriture : analyser la qualité du tracé, comparer le modèle et l'échantillon de l'élève, établir des classifications de lettres, détecter des anomalies, etc. Tandis que le traitement d'image permet de juger globalement l'aspect de la lettre, le traitement de vidéo permet de prendre en compte le mouvement d'écriture, de suivre la réalisation du tracé dans le temps et d'analyser le geste de l'élève.

Chapitre 3

Approche proposée

3.1 Descripteurs de forme

Nous cherchons à obtenir des résultats en temps réel ce qui implique d'utiliser des algorithmes légers en temps de calculs. Il est indispensable d'effectuer une réduction des données pour ne garder que les points caractéristiques extraits des formes. L'ensemble des points caractéristiques d'une image est représenté par un vecteur scalaire avec lequel on peut calculer la similarité entre plusieurs formes. Pour mesurer la similarité entre deux vecteurs caractéristiques, nous utilisons la distance euclidienne.

Les descripteurs de forme permettent de donner une signature à un objet ou à une classe d'objets en réduisant significativement des données de grande dimension. La signature doit permettre de caractériser un objet quelque soit sa configuration dans l'espace. L'efficacité d'un descripteur se mesure en fonction de la robustesse par rapport aux faibles variations de forme mais également en terme de temps de calcul.

Il existe deux types de descripteurs : les descripteurs globaux et les descripteurs locaux. Les descripteurs locaux sont extraits à partir de points d'intérêts et sont plus sensibles au bruit [9]. Les descripteurs globaux portent sur l'image entière et ils sont plus performants que les descripteurs locaux dans un contexte perturbé et bruité [4].

Certains descripteurs globaux sont invariants à des transformations géométriques telles que les translations, les rotations et les changements d'échelle. C'est le cas par exemple des moments de Hu [14], de la transformée de Fourier-Mellin [2] ou des moments de Zernike [28, 16].

3.1.1 Les moments invariants

Introduction aux moments

Dans cette section, nous introduisons la théorie des moments, que l'on peut retrouver dans [22]. Les moments peuvent être utilisés pour décrire une forme de façon globale. Les moments cartésiens 2d d'ordre p et q d'une fonction $f(x, y)$ sont représentés par :

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (3.1)$$

Ce qui donne pour une image discrète :

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (3.2)$$

On peut ainsi calculer le moment d'ordre 0 qui représente la masse ou l'aire d'une fonction :

$$m_{00} = \sum_x \sum_y f(x, y) \quad (3.3)$$

les deux moments d'ordre 1 qui correspondent sont relatifs aux coordonnées du centre de la fonction :

$$m_{10} = \sum_x \sum_y x f(x, y) \quad (3.4)$$

$$m_{01} = \sum_x \sum_y y f(x, y) \quad (3.5)$$

À partir des deux moments d'ordre 1, on peut retrouver le centre de masse de la forme (\bar{x}, \bar{y}) :

$$(\bar{x}, \bar{y}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3.6)$$

3.1.2 Les moments de Hu

On cherche généralement à obtenir des moments qui soient invariants à la translation, la taille et la rotation. Il existe par exemple les moments centraux qui sont invariants à la translation :

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (3.7)$$

Les moments centraux d'ordre 1 μ_{01} et μ_{10} ne donnent pas d'information descriptive car ils sont toujours égaux à 0. Par contre, les moments de second ordre (i.e. μ_{20} et μ_{02}) ont une capacité descriptive. Les moments centraux étant uniquement invariants à la translation, on peut les normaliser pour les rendre invariants à la taille de la forme. On calcule alors les moments centraux normalisés :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (3.8)$$

avec

$$\gamma = \frac{p+q}{2} + 1, \quad \forall p+q \geq 2 \quad (3.9)$$

À partir de là, Hu a défini sept moments invariants à la rotation [14] :

$$\left\{ \begin{array}{l} M_1 = \eta_{20} + \eta_{02} \\ M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ M_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ M_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ M_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + 3\eta_{12}) + ((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2) + \\ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \\ M_6 = (\eta_{20} - \eta_{02})((\eta_{30} + 3\eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ M_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) + \\ (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{12} + \eta_{03})^2 - (\eta_{21} + \eta_{03})^2) \end{array} \right. \quad (3.10)$$

3.2 Les moments de Zernike

Introduits par Teague en 1980 pour la première fois en vision par ordinateur [28], les moments de Zernike forment un descripteur robuste utilisé pour décrire les caractéristiques globales d'une forme en reconnaissance et analyse d'images. Les moments de Zernike ont été largement utilisés dans le domaine de l'image en 2d [16, 19, 27] et ils connaissent depuis peu un regain d'intérêt que ce soit pour l'évolution ou l'optimisation des calculs [1], pour son application dans la biométrie pour l'analyse des empreintes digitales [12, 3], dans la bio-informatique avec l'étude des molécules ou des protéines [31], ou encore pour la classification d'actions humaines [17].

Les moments de Zernike sont basés sur des fonctions orthogonales définies sur le disque unité : les polynômes de Zernike.

3.2.1 Les polynômes de Zernike

Les polynômes de Zernike peuvent être décomposés en fonctions paires et impaires [34]. Les fonctions paires sont :

$$Z_n^m(\rho, \phi) = R_n^m(\rho) \cos(m\phi) \quad (3.11)$$

et les fonctions impaires sont :

$$Z_n^{-m}(\rho, \phi) = R_n^m(\rho) \sin(m\phi) \quad (3.12)$$

où m et n sont des nombres entiers naturels, avec $n \geq m$ et $n - m$ pair, n est le degré du polynôme, ϕ est l'angle d'azimut¹ en radians et ρ la distance radiale normalisée. Les polynômes radiaux R_n^m sont définis par :

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} \frac{(-1)^k (n-k)!}{k! \left(\frac{n+m}{2} - k\right)! \left(\frac{n-m}{2} - k\right)!} \rho^{n-2k} \quad (3.13)$$

3.2.2 Les moments 2D de Zernike

Mathématiquement, les moments de Zernike sont définis [26] avec un ordre p et une répétition q sur $D = \{(p, q) \mid 0 \leq p \leq \infty, |q| \leq p, |p - q| \text{ pair}\}$:

$$Z_{pq} = \frac{p+1}{\pi} \iint_{x^2+y^2 \leq 1} V_{pq}^*(x, y) f(x, y) dx dy \quad (3.14)$$

soit pour une image :

$$Z_{pq} = \frac{p+1}{\pi} \sum_x \sum_y V_{pq}^*(x, y) f(x, y) \quad (3.15)$$

où $x^2 + y^2 \leq 1$ et V_{pq}^* est le conjugué complexe de V_{pq} , lui-même défini comme :

$$V_{pq}(\rho, \theta) = R_{pq}(\rho) e^{iq\theta} \quad (3.16)$$

et R_{pq} représente le polynôme radial de Zernike :

$$R_{pq}(\rho) = \sum_{k=|q|, |p-k| \text{ pair}}^p \frac{(-1)^{\frac{p-k}{2}} \frac{p+k}{2}!}{\frac{p-k}{2}! \frac{k-q}{2}! \frac{k+q}{2}!} \rho^k \quad (3.17)$$

1. L'azimut est l'angle dans le plan horizontal entre la direction d'un objet et une direction de référence.

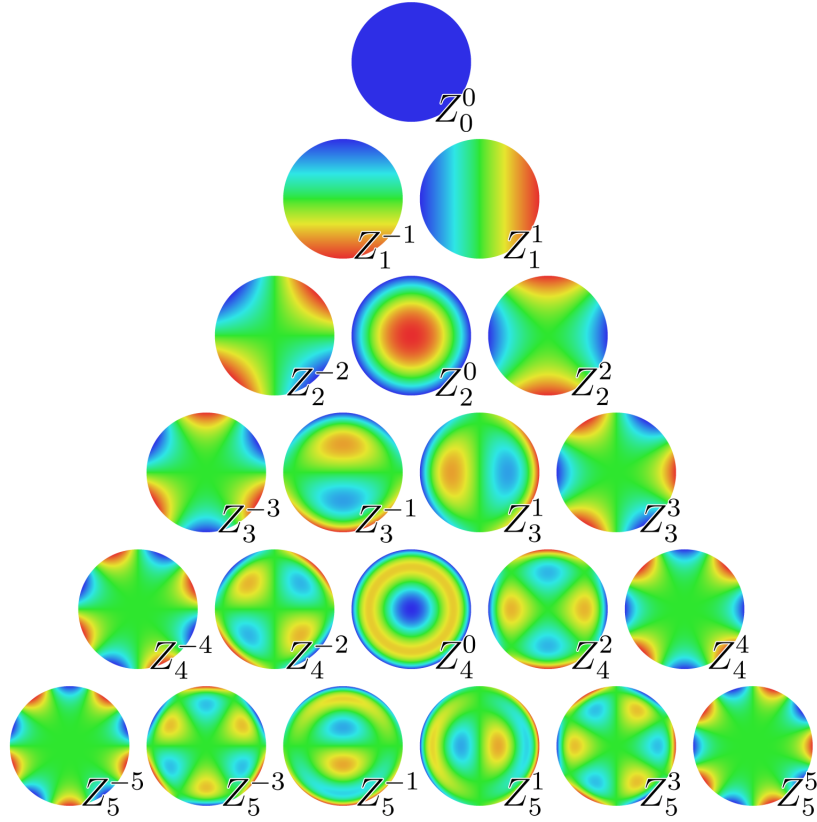


FIGURE 3.1 – Tracés des polynômes de Zernike sur le disque unité.

Les équations (3.14) et (3.16) permettent d'exprimer les moments de Zernike d'une image tournée d'un angle α autour de son origine en coordonnées polaires :

$$Z_{pq}^\alpha = Z_{pq} e^{iq\alpha} \quad (3.18)$$

La reconstruction d'une image s'exprime ainsi :

$$\tilde{f}(x, y) = \sum_{(p,q) \in D} \sum Z_{pq} V_{pq}(x, y) \quad (3.19)$$

3.2.3 Les moments 3D de Zernike

Les moments 2d de Zernike peuvent être étendus en 3d. La description des calculs et des algorithmes est déjà détaillée dans [23]. Les principales étapes comprennent la normalisation de l'image sur le disque unité, le calcul

des moments géométriques, le calcul des moments et des descripteurs 3d de Zernike. L'implémentation en C++ est disponible à l'adresse : cg.cs.uni-bonn.de/project-pages/3dsearch/downloads.html

3.3 Les moments de Krawtchouk

La théorie des moments de Krawtchouk a de nombreuses applications. Les moments invariants de Krawtchouk ont par exemple été utilisé par El affar et al. [10] comme caractéristique discriminante pour la reconnaissance de l'écriture manuscrite arabe ou encore par Yap et al. pour la reconstruction d'image [35]. Les moments de Krawtchouk reposent sur un ensemble de polynômes orthogonaux appelés les polynômes de Krawtchouk. Ces polynômes ont été introduits pour la première fois par M. Krawtchouk en 1929 [20].

3.3.1 Les moments 2D de Krawtchouk

Dans cette section nous introduisons les moments 2d de Krawtchouk, d'après la définition que l'on peut retrouver dans [36]. L'idée est que l'on peut construire des moments invariants en extrayant des caractéristiques locales d'une image 2d avec les polynômes de Krawtchouk. Les polynômes classiques de Krawtchouk d'ordre n s'écrivent :

$$k_n(x; p, N) = \sum_{k=0}^N a_{k,n,p} x^k = {}_2F_1(-n, -x; -N; \frac{1}{p}) \quad (3.20)$$

où $x, n = 0, 1, 2, \dots, N$, $N > 0$, $0 \leq p \leq 1$ et ${}_2F_1$ est la fonction hypergéométrique définie par :

$${}_2F_1(a, b; c, z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \quad (3.21)$$

où le symbole $(a)_k$ représente le symbole de Pochhammer :

$$(a)_k = a(a+1)(a+2)\dots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)} \quad (3.22)$$

L'ensemble des $N+1$ polynômes de Krawtchouk obtenus peuvent être pondérés par la fonction de poids $w(x; p, N)$:

$$w(x; p, N) = \binom{N}{x} p^x (1-p)^{N-x} \quad (3.23)$$

pour satisfaire la condition d'orthogonalité :

$$\sum_{x=0}^N w(x; p, N) k_n(x; p, N) k_m(x; p, N) = \rho(x; p, N) \delta_{nm} \quad (3.24)$$

avec $n, m = 1, 2, \dots, N$ et

$$\rho(n; p, N) = (-1)^n \left(\frac{1-p}{p} \right)^n \frac{n!}{(-N)_n} \quad (3.25)$$

On normalise ensuite les polynômes de Krawtchouk de cette façon :

$$\bar{K}_n(x; p, N) = \frac{K_n(x; p, N)}{\sqrt{\rho(n; p, N)}} \quad (3.26)$$

et la condition d'orthogonalité devient :

$$\sum_{x=0}^N \bar{K}_n(x; p, N) \bar{K}_m(x; p, N) = \delta_{nm} \quad (3.27)$$

On peut maintenant calculer les moments invariants de Krawtchouk d'ordre $n + m$ d'une fonction image $f(x, y)$:

$$\bar{Q}_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \bar{K}_n(x; p_x, N-1) * \bar{K}_m(y; p_y, M-1) * f(x, y) \quad (3.28)$$

L'image f peut ensuite être entièrement reconstruite en fonction de ces moments :

$$f(x, y) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \bar{K}_n(x; p_x, N-1) * \bar{K}_m(y; p_y, M-1) * Q_{nm} \quad (3.29)$$

3.3.2 Les moments 3D de Krawtchouk

Les moments 2d de Krawtchouk peuvent être étendus en 3D comme l'ont montré Mademlis et al. [21]. Soit $f(x, y, z)$ une fonction de trois dimensions définie sur un ensemble discret $A = \{(x, y, z) \mid x, y, z \in \mathbb{N}, x \in [0..N-1], y \in [0..M-1], z \in [0..L-1]\}$. Les moments 3D de Krawtchouk d'ordre $(n+m+l)$ sont définis par :

$$\bar{Q}_{nml} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{L-1} \bar{K}_n(x; p_x, N-1) * \bar{K}_m(y; p_y, M-1) * \bar{K}_l(z; p_z, L-1) * f(x, y, z) \quad (3.30)$$

où \bar{K}_n est le polynôme de Krawtchouk normalisé (orthonormé) :

$$\bar{K}_n(x; p, N) = \frac{K_n(x; p, N)}{\sqrt{\rho(n; p, N)}} \quad (3.31)$$

avec K_n le polynôme classique de Krawtchouk de degré n :

$$K_n(x; p, N) = \sum_{k=0}^n a_{k,n,p} x^k = {}_2F_1(-n, -x; -N; \frac{1}{p}) \quad (3.32)$$

où ${}_2F_1$ est la fonction hypergéométrique définie par :

$${}_2F_1(a, b; c, z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \quad (3.33)$$

où le symbole $(a)_k$ représente le symbole de Pochhammer :

$$(a)_k = a(a+1)(a+2)\dots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)} \quad (3.34)$$

$$\rho(n; p, N) = (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{(-N)_n} \quad (3.35)$$

Pour calculer les moments de Krawtchouk, nous pouvons utiliser la relation de récurrence suivante :

$$K_{n+1}(x; p, N) = \left(1 + \frac{n - np - x}{pN - pn}\right) * K_n(x; p, N) - \left(\frac{n - np}{pN - pn}\right) * K_{n-1}(x; p, N) \quad (3.36)$$

avec les conditions initiales :

$$\begin{cases} K_0(x; p, N) = 1 \\ K_1(x; p, N) = 1 - \frac{1}{Np}x \end{cases} \quad (3.37)$$

Connaissant les moments 3D de Krawtchouk, la fonction f peut ensuite être reconstruite :

$$f(x, y, z) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} \bar{K}_n(x; p_x, N-1) * \bar{K}_m(y; p_y, M-1) * \bar{K}_l(z; p_z, L-1) * \bar{Q}_{nml} \quad (3.38)$$

3.4 Classification

La classification permet de catégoriser un ensemble d'objets en fonction des similarités qu'ils présentent entre eux. La méthode de catégorisation que l'on utilise est basée sur les graphes de similarités.

3.4.1 Graphe de similarités

Nous utilisons la théorie des graphes pour modéliser les données dont on dispose. L'idée est de représenter un ensemble fini d'objets par un graphe $G = (V, E)$ où V est un ensemble fini de sommets $V = \{v_1, v_2, \dots, v_n\}$ et E un ensemble fini d'arêtes $E \subset V * V$. Deux sommets sont dits adjacents si l'arête $(u, v) \in E$. Le graphe est pondéré par une fonction de poids p qui reflète la similarité entre deux sommets du graphe et donc entre les deux objets considérés. La fonction p est donc une fonction inverse à la distance entre deux sommets. La distance utilisée peut par exemple être la distance euclidienne entre les vecteurs caractéristiques de u et de v . Le graphe peut être représenté par la matrice de similarité $P : P(u, v) = p(v, u)$ si les sommets u et v sont adjacents, $P(u, v) = 0$ sinon et $P(u, u) = 1$ pour la diagonale principale. Après normalisation des valeurs, la matrice de similarités obtenue permet de mettre en évidence les relations entre les sommets : plus $P(u, v)$ est proche de 1, plus les objets sont similaires et plus $P(u, v)$ tend vers 0, plus les objets sont différents.

3.4.2 Autres méthodes

D'autres algorithmes de classification plus classique d'objets peuvent être envisagés. Ces algorithmes utilisent en entrée un vecteur de caractéristiques de même type extraites des objets. Par exemple, la classification selon les K plus proches voisins qui attribue à l'individu la classe majoritaire parmi les K plus proches voisins dans l'espace d'apprentissage. De même, la classification selon les K -moyennes effectue une classification automatique des individus en K classes avec l'algorithme des K -moyennes. Un dernier modèle à titre d'exemple est la classification utilisant un modèle gaussien pour représenter la distribution de chaque classe.

3.5 Conclusion

Nous avons vu dans ce chapitre que les moments géométriques invariants permettaient de représenter une image en une métrique autorisant les calculs

de similarité. Cette méthode a l'avantage de préserver la signature de l'image (la variabilité intra-classe) tout en gardant la similarité entre des images différentes (la variabilité inter-classe). En outre, la réduction des données ainsi effectuée présente l'avantage de réduire la complexité des calculs sur les images et les vidéos et autorise son implémentation pour une utilisation en temps réel.

Chapitre 4

Implémentation et Résultats

4.1 Outils utilisés

4.1.1 La librairie Pandore

La librairie Pandore [24] est une bibliothèque d'opérateurs de traitement d'images développée par le GREYC. Conçue pour réaliser des opérations de traitement sur des images 1d, 2d et 3d, Pandore s'utilise en lignes de commandes et autorise le développement de nouveaux opérateurs en C++. Elle est disponible publiquement et gratuitement à l'adresse : <http://www.greyc.ensicaen.fr/~regis/Pandore/index-fr.html>.

Les opérateurs de traitement d'images disponibles avec Pandore rassemblent les opérations basiques de conversion, d'arithmétique et de logique, de transformation géométrique sur les images, ainsi que des opérations plus complexes comme la segmentation de contour ou de région, l'extraction de caractéristiques, la classification d'objets, etc. Pandore utilise son propre format d'image, reconnaissable par l'extension de fichier `.pan`.

Listing 4.1 présente un exemple simple de scénario classique d'utilisation de Pandore (sous Linux). Cette séquence convertit une image bitmap en image pandore, binarise l'image obtenue et permet de visualiser le résultat final. Les paramètres appliqués à l'opérateur `pbinarization` permettent de situer le seuillage des niveaux de gris et ses valeurs (88 et 255) sont obtenus expérimentalement d'après l'histogramme de l'image. Figure 4.1 montre l'image originale et l'image binarisée obtenue.

```
$ pbmp2pan tangram.bmp tangram.pan
$ pbinarization 88 255 tangram.pan | pvisu
```

Listing 4.1 – Opérateurs simples de Pandore

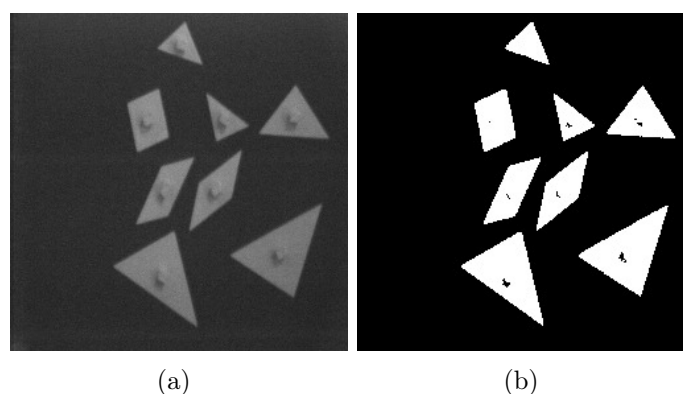


FIGURE 4.1 – Exemple de traitement d’image avec Pandore. (a) image originale ; (b) image binarisée.

4.1.2 La base MNIST

MNIST [18] est une banque de données d’images de chiffres manuscrits initiée par Y. LeCun et C. Cortes afin de produire un grand nombre de données réelles et variées dans le domaine de l’apprentissage et de la reconnaissance de lettres manuscrites. Elle contient un ensemble de 60 000 images utilisées pour l’apprentissage et un ensemble de 10 000 images pour les tests. L’intérêt d’utiliser MNIST est que les images ont toutes la même taille et ont été normalisées et centrées, ce qui évite avantageusement les étapes de prétraitement et de formattage. La base MNIST est disponible publiquement et gratuitement à cette adresse <http://yann.lecun.com/exdb/mnist/>.

Un ensemble d’images MNIST est composé de deux fichiers : l’un contenant les matrices de pixels de chaque image et l’autre contenant les étiquettes correspondant à chaque image. Par exemple on a les fichiers `train-images-idx3-ubyte` et `train-labels-idx1-ubyte` pour la base d’apprentissage. Ces fichiers utilisent un format spécial qu’il faut décoder pour pouvoir extraire les données. Un simple programme, comme par exemple celui de l’UCLA (University of California) disponible à cette adresse : abel.ee.ucla.edu/cvxopt/_downloads/mnist.py, peut être utilisé pour effectuer cette opération. Lorsque les matrices d’images sont extraites, il est ensuite possible de les convertir dans un format d’image standard comme bmp, tiff, jpg, etc. ou dans le format pandore pour pouvoir les manipuler avec Pandore. Un code Python est disponible en [Annexe ??](#) pour effectuer l’opération de construction des images bitmap. On obtient ensuite une collection de 60 000 images en niveaux de gris et de taille 28x28 pixels, nommées en fonction de leur étiquette et d’un numéro de comptage, par exemple `1.12.bmp` pour la douzième image du chiffre

1. Il est facile ensuite de convertir ces images bmp dans le format pandore en utilisant l'opérateur `pbmp2pan` de Pandore. [Figure 4.2](#) montre des images de dix chiffres issus de la base MNIST.



FIGURE 4.2 – Exemples d'images extraites de MNIST.

4.1.3 La base écriture

La base écriture est une base de données d'écriture qui rassemble des lettres manuscrites (ou chiffres) collectées lors d'examens passés par des enfants. Un examen constitue à faire écrire à un enfant un ensemble de lettres, par exemple des chiffres, des minuscules ou des majuscules. La base contient environ 330 examens. Chaque examen contient 3 conditions d'écriture. Une condition d'écriture est un ensemble de lettres écrites sur un même calque, qui peut être soit les 10 chiffres, soit les 13 premières lettres de l'alphabet, soit les 13 dernières lettres de l'alphabet. En particulier l'alphabet peut représenter l'alphabet cursif minuscule, l'alphabet cursif majuscule, l'alphabet bâton minuscule ou l'alphabet bâton majuscule. [Figure 4.3](#) montre un exemple d'examen représenté par les trois images reconstituant les traces d'écriture. Ces données ont été capturées avec un Tablet-PC et un stylet, ce qui permet de reconstituer le tracé de l'écriture dans le temps. Nous disposons donc des informations visuelles des traces d'écriture, ou images 2d, mais aussi des informations temporelles, avec lesquelles on peut reconstituer des animations (images 2d+t). En plus des traces d'écriture, des informations telles que l'âge et la classe scolaire de l'enfant sont recueillies et sont associées à chaque examen.

4.2 Traitements basiques

4.2.1 Segmentation d'image 2d

L'organisation et la structure des images provenant de la base écriture (voir [Fig. 4.3](#)) permettent de les manipuler facilement par Pandore. Quelques résultats d'opérations basiques de traitement d'images sont illustrés en [Fig. 4.4](#) : conversion en niveaux de gris (a), binarisation (b), étiquetage des régions homogènes (c). Les résultats des deux premières images correspondent aux

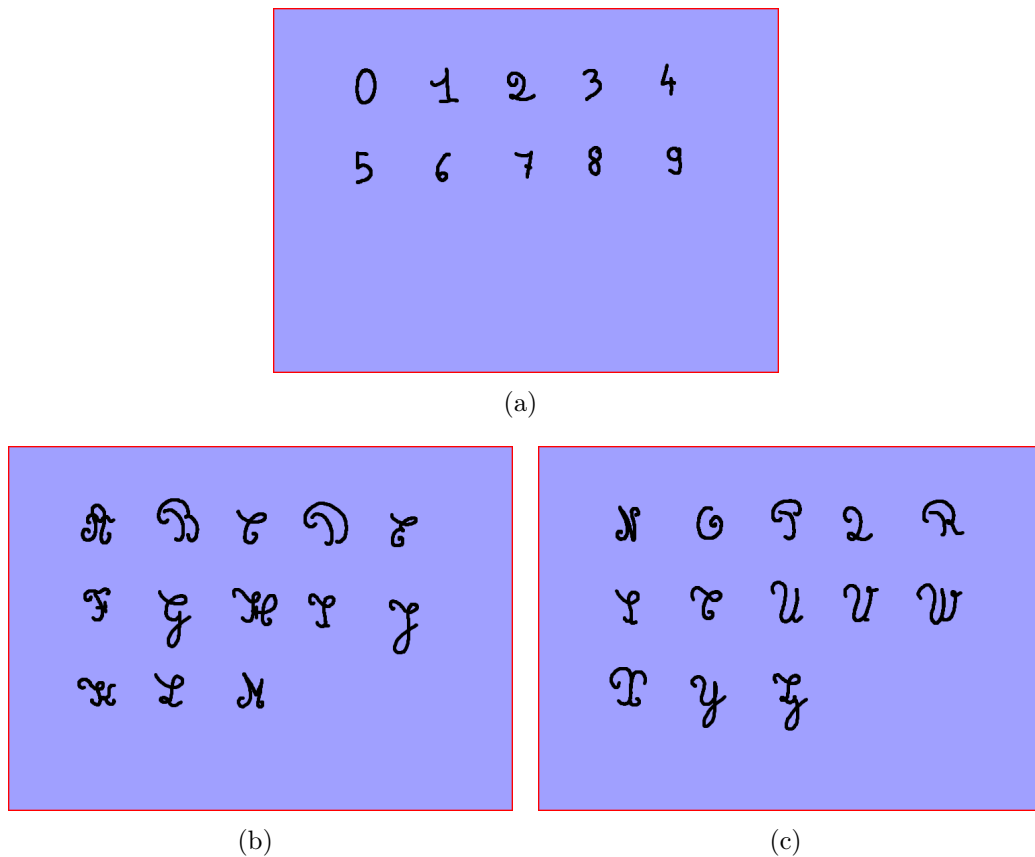


FIGURE 4.3 – Exemple d’examen. (a) les dix chiffres ; (b) les treize premières lettres de l’alphabet cursif majuscule ; (c) les treize dernières lettres de l’alphabet cursif majuscule.

opérations classiques de conversion en niveaux de gris et de binarisation en noir et blanc. La troisième image nécessite d’avantage d’interprétation : l’image a été segmentée en régions étiquetées et colorées avec une couleur unique. On parle également d’identification des composantes connexes. Les résultats sur l’image (c) montrent que la segmentation a bien détecté et séparé les différentes lettres. Par contre, on voit sur l’image (d) que l’algorithme considère le point et la base des lettres *i* et *j* comme deux objets distincts car ils ne sont pas connexes.

Les commandes associées à ces opérations sont explicitées par [Listing 4.2](#). La première ligne permet de convertir une image couleur en image de niveaux de gris en combinant les trois composantes (rouge,vert,bleu) avec les ratios de la méthode NTSC (National Television System Committee) passés

en paramètre, c'est-à-dire 0.299, 0.587 et 0.114. La deuxième ligne permet d'effectuer le seuillage binaire de l'image selon une valeur de seuil sélectionnée automatiquement d'après l'histogramme de l'image, ce qui permet de séparer les traces d'écriture de l'arrière-plan mais aussi d'éliminer le cadre de l'image. Cette opération permet donc déjà de distinguer deux groupes dans l'image : les objets et le fond. La troisième ligne permet d'aller plus loin dans la segmentation puisqu'elle permet de repérer et d'étiqueter les régions connexes de l'image. La valeur 8 du paramètre indique le voisinage utilisé pour la recherche de la connexité.

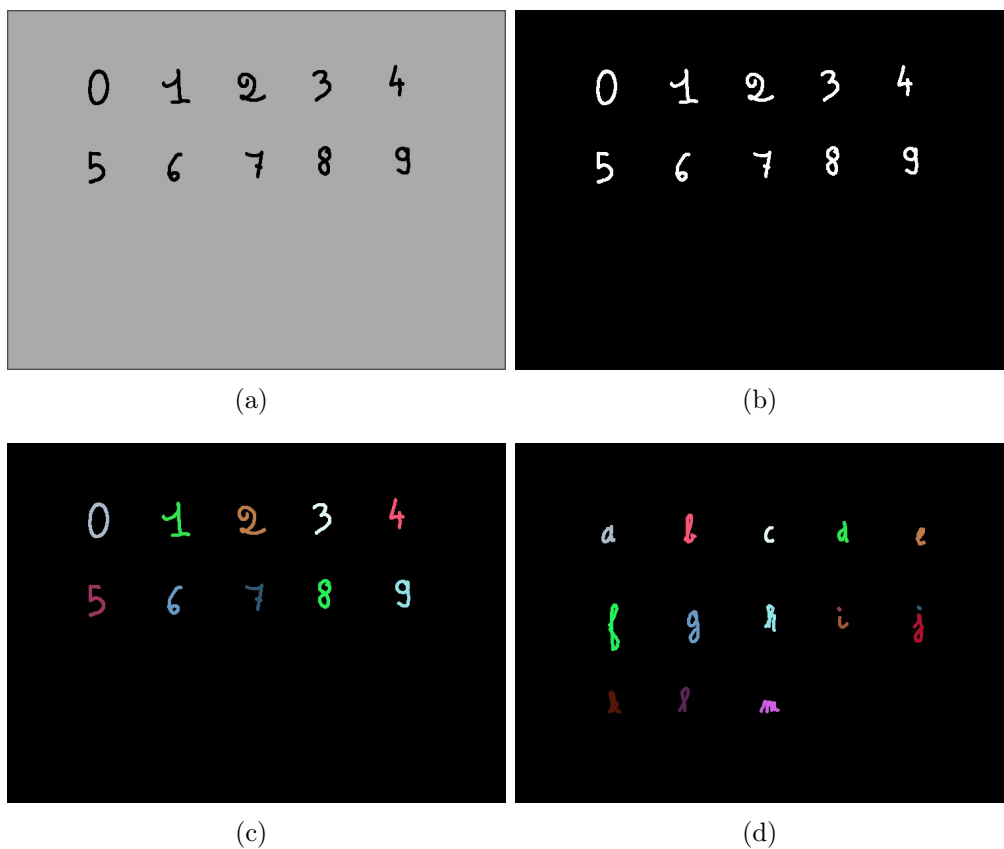


FIGURE 4.4 – Exemples de traitements d'images d'un examen. (a) image en niveaux de gris ; (b) image binarisée ; (c) et (d) images segmentées en régions étiquetées.

```
$ prgb2gray 0.299 0.587 0.114 condition.pan conditiongray.pan
$ pvariancebinarization conditiongray.pan conditionbw.pan
$ plabeling 8 conditionbw.pan conditionsegm.pan
```

Listing 4.2 – Opérateurs de traitements d’image d’examen en Pandore

4.2.2 Conversion d’image 2d en 3d

En plus des images, nous disposons des données spatio-temporelles contenues dans des fichiers. L’idée est que la dimension apportée par le temps apporte davantage d’informations sur les traces d’écriture que les images en 2d vues dans [Section 4.2.1](#). Les fichiers disponibles regroupent les coordonnées des points du stylet capturés dans le temps. [Listing 4.3](#) montre un exemple de ce type de fichiers.

```
Fichier: S051BEWA.txt – Examens
Modele caracteres: Ecolier
Sujet S051BEWA Examen: condition1 Dictee
Nombre de traces: 14
Nombre de pauses: 13
Temps total des traces: 15.398 sec
Temps total des pauses: 129.199 sec
Temps total essai: 144.597 sec

trace 1: 0.234 sec

Pause 1: 119.387 sec

149x89
148x89
147x89
...
148x89
147x89
146x90
trace 2: 2.059 sec

Pause 2: 0.921 sec

269x102
270x102
270x103
...
```

Listing 4.3 – Exemple type de fichier examen.

On peut convertir ce type de données en images 3d sous Pandore, dans lesquelles chaque plan représente une unité de temps. La conversion peut se faire via l’opérateur `ptxt2pan` qui prend transforme un fichier texte en fichier

image pandore. Dans notre cas, le fichier texte en entrée comprend les valeurs et les coordonnées des points non nuls. Listing 4.4 est un exemple du format de fichier utilisé. Le premier nombre représente la valeur du pixel de coordonnée cartésienne indiquée par les deux nombres suivants, et la quatrième valeur est le plan dans l'image, c'est-à-dire le repérage dans le temps.

```
255 149 89 0
255 148 89 1
255 147 89 2
255 146 89 3
255 146 90 4
255 145 90 5
...
```

Listing 4.4 – Exemple type de fichier examen.

4.2.3 Construction des modèles

Nous avons vu dans Section 2.3.1 que faire la moyenne de plusieurs images permettait de réduire le bruit. Si nous appliquons ce principe à l'ensemble des échantillons d'une même lettre, il est possible de construire un modèle qui n'est autre que la moyenne de tous les échantillons d'une lettre. Par exemple sur la base MNIST, la moyenne de chaque classe de chiffre a été calculée (voir Fig. 4.5).

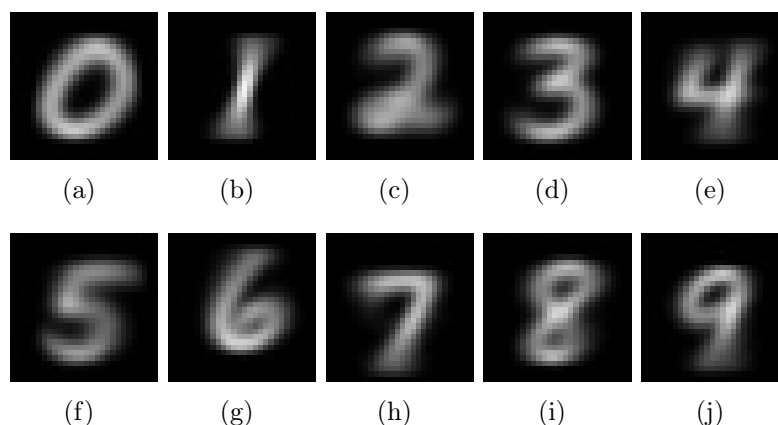


FIGURE 4.5 – Modèles des chiffres par la moyenne.

La construction des modèles en 3d peut être envisagée de la même manière. Elle n'a pas encore été testée à cet instant car l'automatisation de la conversion des fichiers de la base écriture dans le format pandore peut être

longue. Une amélioration consisterait à utiliser le filtrage médian temporel sur les images. De plus, nous pensons que la méthode basée sur la régularisation non-locale discrète sur graphes pondérés apporterait de meilleurs résultats [11].

4.3 Les moments de Zernike

4.3.1 Résultats 2d

Dans cette partie sont présentés les résultats des tests effectués sur des images de chiffres de 28×28 px (voir Fig. 4.6). Le Tableau 4.1 donne les moments de Zernike 2d calculés à partir des images. Les nombres obtenus ont été arrondis pour des raisons de lisibilité du tableau. $r(45^\circ)$ signifie que l'image a subi une rotation de 45° dans le sens horaire, $s(x2)$ signifie que l'image a subi un agrandissement de facteur 2 et $t(20,20)$ signifie que la forme a subi une translation de 20 pixels vers la droite et de 20 pixels vers le bas.

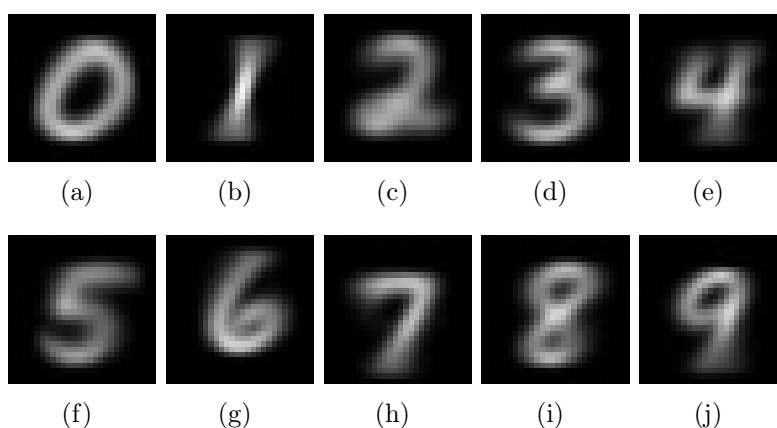


FIGURE 4.6 – Images utilisées pour les tests.

On voit que le premier moment calculé $M1$ est toujours égal à 0 ce qui n'est pas intéressant pour discriminer les vecteurs caractéristiques. On constate aussi que les moments restent relativement insensibles à la rotation et à l'échelle mais les valeurs changent de façon significative lorsque l'on effectue une translation. Ce problème peut être partiellement résolu par une étape de normalisation, très souvent présente lors de prétraitements sur l'image. Les résultats laissent déjà apparaître que les classes peuvent être discriminées par ces moments car les valeurs sont suffisamment distinctes pour la variabilité

inter-classes et suffisamment proches pour la variabilité intra-classe. L'utilisation d'un classifieur automatique est cependant nécessaire pour valider et calculer le taux d'efficacité de cette méthode.

TABLE 4.1 – Résultats moments de Zernike 2d sur la rotation, l'échelle et la translation.

| | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 |
|----------|----|--------|--------|--------|---------|----------|---------|---------|--------|
| (a) | 0 | 0.0057 | 0.6587 | 0.0172 | 0.4303 | 0.0230 | 0.23531 | 0.0437 | 0.4568 |
| r(45 °) | 0 | 0.0065 | 0.6466 | 0.0146 | 0.3947 | 0.01536 | 0.2560 | 0.0418 | 0.4296 |
| r(90 °) | 0 | 0.0057 | 0.6587 | 0.0172 | 0.4303 | 0.02300 | 0.2353 | 0.0437 | 0.4568 |
| r(135 °) | 0 | 0.0065 | 0.6466 | 0.0146 | 0.3947 | 0.01536 | 0.2560 | 0.0418 | 0.4296 |
| s(x2) | 0 | 0.0117 | 0.6572 | 0.0294 | 0.4282 | 0.0250 | 0.2364 | 0.0509 | 0.4483 |
| s(x4) | 0 | 0.0169 | 0.6560 | 0.0399 | 0.4260 | 0.0275 | 0.2366 | 0.0628 | 0.4410 |
| t(20,20) | 0 | 0.2646 | 0.6738 | 0.6961 | 0.5525 | 0.7427 | 0.8964 | 0.4064 | 0.5366 |
| (b) | 0 | 0.0024 | 0.7680 | 0.0080 | 0.8304 | 0.0142 | 0.6330 | 0.0302 | 0.6210 |
| r(45 °) | 0 | 0.0035 | 0.7608 | 0.0113 | 0.81001 | 0.02454 | 0.6124 | 0.03323 | 0.5792 |
| r(90 °) | 0 | 0.0024 | 0.7680 | 0.0080 | 0.83045 | 0.01421 | 0.6330 | 0.03028 | 0.6210 |
| r(135 °) | 0 | 0.0035 | 0.7608 | 0.0113 | 0.81001 | 0.02454 | 0.6124 | 0.03323 | 0.5792 |
| s(x2) | 0 | 0.0126 | 0.7666 | 0.0366 | 0.8261 | 0.050434 | 0.62804 | 0.0740 | 0.6079 |
| s(x4) | 0 | 0.0182 | 0.7654 | 0.0542 | 0.8223 | 0.079128 | 0.62274 | 0.1115 | 0.5982 |
| t(20,20) | 0 | 0.2658 | 0.6819 | 0.7367 | 0.5628 | 0.87643 | 1.2634 | 0.7039 | 1.3053 |
| (c) | 0 | 0.0090 | 0.6087 | 0.0380 | 0.3160 | 0.0816 | 0.2999 | 0.1302 | 0.3809 |
| r(45 °) | 0 | 0.0122 | 0.6137 | 0.0365 | 0.3261 | 0.0673 | 0.2926 | 0.1387 | 0.3801 |
| r(90 °) | 0 | 0.0090 | 0.6087 | 0.0380 | 0.3160 | 0.0816 | 0.2999 | 0.1302 | 0.3809 |
| r(135 °) | 0 | 0.0090 | 0.6114 | 0.0361 | 0.3220 | 0.0731 | 0.2951 | 0.13057 | 0.3857 |
| s(x2) | 0 | 0.0059 | 0.6080 | 0.0347 | 0.3172 | 0.0775 | 0.3082 | 0.1007 | 0.3884 |
| s(x4) | 0 | 0.0104 | 0.6074 | 0.0416 | 0.3175 | 0.0758 | 0.3116 | 0.1018 | 0.3919 |
| t(20,20) | 0 | 0.2607 | 0.6737 | 0.6720 | 0.5779 | 0.7134 | 0.7413 | 0.4425 | 0.4267 |
| (d) | 0 | 0.0327 | 0.6306 | 0.0691 | 0.3896 | 0.0490 | 0.3212 | 0.1395 | 0.4309 |
| r(45 °) | 0 | 0.0248 | 0.6389 | 0.0525 | 0.4025 | 0.0310 | 0.3134 | 0.1079 | 0.4072 |
| r(90 °) | 0 | 0.0327 | 0.6306 | 0.0691 | 0.3896 | 0.0490 | 0.32120 | 0.1395 | 0.4309 |
| r(135 °) | 0 | 0.0215 | 0.6460 | 0.0471 | 0.4147 | 0.0288 | 0.2994 | 0.0947 | 0.4111 |
| s(x2) | 0 | 0.0278 | 0.6330 | 0.0566 | 0.3931 | 0.04772 | 0.3223 | 0.1166 | 0.4367 |
| s(x4) | 0 | 0.0269 | 0.6337 | 0.0535 | 0.3942 | 0.0474 | 0.3222 | 0.1100 | 0.4374 |
| t(20,20) | 0 | 0.2548 | 0.6820 | 0.6711 | 0.5838 | 0.7241 | 0.7753 | 0.4340 | 0.4661 |
| (e) | 0 | 0.0056 | 0.6430 | 0.0254 | 0.3975 | 0.0816 | 0.2314 | 0.1151 | 0.4144 |
| r(45 °) | 0 | 0.0071 | 0.6372 | 0.0293 | 0.3804 | 0.0829 | 0.2383 | 0.1224 | 0.4034 |

| | | | | | | | | | |
|----------|---|---------|--------|---------|---------|---------|---------|---------|---------|
| r(90 °) | 0 | 0.0056 | 0.6430 | 0.0254 | 0.3975 | 0.0816 | 0.2314 | 0.1151 | 0.4144 |
| r(135 °) | 0 | 0.0071 | 0.6372 | 0.0293 | 0.3804 | 0.0829 | 0.2383 | 0.1224 | 0.4034 |
| s(x2) | 0 | 0.0144 | 0.6411 | 0.0465 | 0.3931 | 0.0812 | 0.2220 | 0.1392 | 0.4041 |
| s(x4) | 0 | 0.0198 | 0.6398 | 0.0580 | 0.3903 | 0.08183 | 0.2169 | 0.1532 | 0.3969 |
| t(20,20) | 0 | 0.2665 | 0.6744 | 0.6924 | 0.5767 | 0.7398 | 0.8432 | 0.4581 | 0.4842 |
| (f) | 0 | 0.0134 | 0.6429 | 0.0301 | 0.4068 | 0.05127 | 0.3081 | 0.0682 | 0.4778 |
| r(45 °) | 0 | 0.0128 | 0.6447 | 0.0299 | 0.41155 | 0.05678 | 0.3053 | 0.0770 | 0.4725 |
| r(90 °) | 0 | 0.0134 | 0.6429 | 0.0301 | 0.4068 | 0.05127 | 0.3081 | 0.0682 | 0.47786 |
| r(135 °) | 0 | 0.0128 | 0.6447 | 0.02998 | 0.4115 | 0.05678 | 0.3053 | 0.07708 | 0.4725 |
| s(x2) | 0 | 0.0223 | 0.6399 | 0.05318 | 0.4011 | 0.05628 | 0.3023 | 0.0988 | 0.4598 |
| s(x4) | 0 | 0.0274 | 0.6379 | 0.06524 | 0.3973 | 0.05989 | 0.2988 | 0.1164 | 0.4480 |
| t(20,20) | 0 | 0.2701 | 0.6539 | 0.6850 | 0.4995 | 0.67761 | 0.8502 | 0.3308 | 0.4202 |
| (g) | 0 | 0.05292 | 0.6566 | 0.1104 | 0.4337 | 0.0593 | 0.1476 | 0.2269 | 0.3849 |
| r(45 °) | 0 | 0.0564 | 0.6553 | 0.1188 | 0.4309 | 0.07406 | 0.1490 | 0.2411 | 0.3734 |
| r(90 °) | 0 | 0.0529 | 0.6566 | 0.1104 | 0.4337 | 0.0593 | 0.14768 | 0.2269 | 0.3849 |
| r(135 °) | 0 | 0.0564 | 0.6553 | 0.1188 | 0.43093 | 0.07406 | 0.14904 | 0.24115 | 0.3734 |
| s(x2) | 0 | 0.0483 | 0.6563 | 0.1013 | 0.4342 | 0.0601 | 0.1620 | 0.2125 | 0.3847 |
| s(x4) | 0 | 0.0469 | 0.6559 | 0.09893 | 0.4337 | 0.0613 | 0.1687 | 0.2062 | 0.3842 |
| t(20,20) | 0 | 0.2515 | 0.6982 | 0.684 | 0.6264 | 0.7891 | 0.8391 | 0.5207 | 0.6178 |
| (h) | 0 | 0.0495 | 0.6382 | 0.1237 | 0.4066 | 0.1591 | 0.1565 | 0.3317 | 0.3182 |
| r(45 °) | 0 | 0.0371 | 0.6525 | 0.1069 | 0.4315 | 0.1485 | 0.1371 | 0.2994 | 0.3242 |
| r(90 °) | 0 | 0.0433 | 0.6411 | 0.1100 | 0.4155 | 0.1382 | 0.1569 | 0.3257 | 0.3360 |
| r(135 °) | 0 | 0.0371 | 0.6525 | 0.1069 | 0.4315 | 0.1485 | 0.1371 | 0.2994 | 0.3242 |
| s(x2) | 0 | 0.0530 | 0.6384 | 0.1293 | 0.4021 | 0.1622 | 0.1349 | 0.3329 | 0.3185 |
| s(x4) | 0 | 0.0555 | 0.6382 | 0.1338 | 0.3997 | 0.1641 | 0.1274 | 0.3333 | 0.3182 |
| t(20,20) | 0 | 0.2725 | 0.6698 | 0.6966 | 0.5870 | 0.7367 | 0.8322 | 0.5009 | 0.4475 |
| (i) | 0 | 0.0163 | 0.6572 | 0.04613 | 0.4429 | 0.05735 | 0.2716 | 0.1101 | 0.4753 |
| r(45 °) | 0 | 0.01621 | 0.6545 | 0.04467 | 0.4339 | 0.0540 | 0.2832 | 0.1018 | 0.4536 |
| r(90 °) | 0 | 0.01635 | 0.6572 | 0.04613 | 0.4429 | 0.05735 | 0.2716 | 0.1101 | 0.4753 |
| r(135 °) | 0 | 0.01621 | 0.6545 | 0.04467 | 0.4339 | 0.05401 | 0.2832 | 0.1018 | 0.4536 |
| s(x2) | 0 | 0.0269 | 0.6534 | 0.0707 | 0.4339 | 0.0734 | 0.2607 | 0.1477 | 0.4482 |
| s(x4) | 0 | 0.0324 | 0.6509 | 0.0832 | 0.42851 | 0.0812 | 0.2547 | 0.1663 | 0.4318 |
| t(20,20) | 0 | 0.2737 | 0.6598 | 0.7013 | 0.5238 | 0.7138 | 0.9096 | 0.3879 | 0.4718 |
| (j) | 0 | 0.0462 | 0.6750 | 0.1014 | 0.4917 | 0.0874 | 0.1452 | 0.2585 | 0.3518 |
| r(45 °) | 0 | 0.0424 | 0.6759 | 0.1021 | 0.4860 | 0.1143 | 0.1271 | 0.2599 | 0.3125 |
| r(90 °) | 0 | 0.0462 | 0.6750 | 0.1014 | 0.4917 | 0.0874 | 0.1452 | 0.2585 | 0.3518 |
| r(135 °) | 0 | 0.0455 | 0.6717 | 0.1062 | 0.4783 | 0.1214 | 0.1411 | 0.2723 | 0.3230 |

| | | | | | | | | | |
|----------|---|--------|--------|--------|--------|--------|--------|--------|--------|
| s(x2) | 0 | 0.0544 | 0.6724 | 0.1182 | 0.4861 | 0.0847 | 0.1242 | 0.2747 | 0.3443 |
| s(x4) | 0 | 0.0590 | 0.6709 | 0.1275 | 0.4830 | 0.0890 | 0.1141 | 0.2819 | 0.3405 |
| t(20,20) | 0 | 0.2840 | 0.6641 | 0.7278 | 0.5777 | 0.7879 | 0.9689 | 0.5583 | 0.6545 |

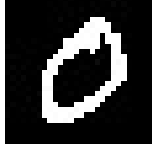

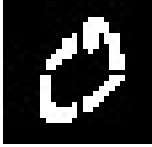

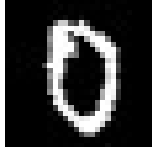
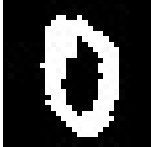
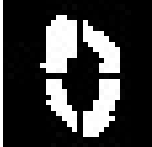
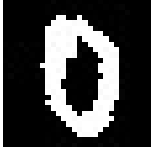
4.3.2 Résultats 3d

4.4 Les moments de Krawtchouk

4.4.1 Résultats 2d

Le [Tableau 4.2](#) présente quelques tests des moments de Krawtchouk calculés sur des images 2d avec Pandore. Les résultats obtenus montrent que le moment d'ordre 0 est déjà suffisant pour la reconstruction de l'image.

TABLE 4.2 – Moments de Krawtchouk et reconstruction d'images 2d.

| Image originale | Ordre 0 | Ordre 1 | Ordre 2 |
|---|--|---|--|
|  |  $Q_0 = 32895$ |  $Q_1 = -355470$ |  $Q_2 = 1.36138e + 06$ |
| Rotation 45° | Ordre 0 | Ordre 1 | Ordre 2 |
|  |  $Q_0 = 32859$ |  $Q_1 = 178305$ |  $Q_2 = 679505$ |

4.4.2 Résultats 3d

4.5 Conclusion

Les résultats qui sont encore en cours de vérification et de tests, permettent cependant d'affirmer que la dimension temporelle donne un indice prometteur dans l'analyse des données d'écriture. D'autres algorithmes de calculs de moments géométriques 2d et 3d peuvent être utilisés pour comparer leur efficacité et leur robustesse et de choisir quelle méthode est la plus adaptée pour une problématique donnée.

Chapitre 5

Conclusion et perspectives

En conclusion, nous avons montré que la réduction de la dimension des images et des séquences d'images numériques par les moments géométriques offrent une signature efficace des formes et des mouvements. Les moments invariants 2D et 3D de Zernike et de Krawtchouk donnent de meilleurs résultats que les moments classiques tels que les moments de Hu. Les perspectives intéressantes pour la suite de cette approche pourraient être par exemple de poursuivre la classification en utilisant les propriétés spectrales du graphe de similarités, de modéliser chaque classe de la base des images 2d ou 3d en utilisant la régularisation non-locale sur graphes pondérés ou encore de détecter et d'estimer le mouvement de l'écriture en se basant sur les algorithmes de block matching ou de flot optique.

L'objectif de ce stage était principalement d'expérimenter des résultats très récents, voire innovants, dans le domaine du traitement d'images, de la reconnaissance de formes ou de la vision par ordinateur, qui suscitera l'intérêt de la recherche pour encore de nombreuses années à venir. L'approche que nous avons utilisée, basée sur les moments invariants de Zernike et de Krawtchouk sont des solutions efficaces pour la problématique de reconnaissance de formes 2d et 3d. Leur utilisation suscite depuis peu l'engouement dans le domaine et est fortement d'actualité. Non seulement cette approche résout le problème d'invariance par transformation géométrique (translation, taille et rotation) mais elle peut prendre également en compte la dimension du temps, ce qui est particulièrement intéressant pour notre cas d'étude. Les résultats obtenus montrent clairement l'efficacité et la robustesse apportée par cette troisième dimension pour la reconnaissance. De plus, cette approche permet également de réduire de façon significative les données et autorise son utilisation en temps réel. L'implémentation d'un prototype utilisant cette approche pour le projet d'interface informatisée d'aide à l'apprentissage de l'écriture est en cours.

L'utilisation des invariants dans le temps ouvre également d'importantes applications dans de nombreux domaines. Parmi elles, on peut citer l'indexation d'images ou de vidéos par le contenu, qui permet entre autre de trouver de classifier des mouvements ou des actions et qui a besoin d'extraire des caractéristiques pertinentes des vidéos, de réduire leur dimensionnalité et d'être organisée par classification. Le choix des invariants se fera alors en fonction de leur robustesse et de leur efficacité.

D'un point de vue plus personnel, ce stage m'a particulièrement intéressé et m'a apporté un grand nombre de compétences. Cette expérience a été l'occasion de faire une synthèse de toutes les connaissances acquises lors de mes cinq années d'études universitaires en sciences informatiques et c'est avec une grande satisfaction qu'elles se soient ainsi terminées. En plus d'avoir progressé, amélioré ma propre autonomie et perfectionné mes méthodes de travail que ce soit pour la recherche d'informations, la réflexion, ou la technique, ce stage m'a permis de découvrir et de m'initier au monde de la recherche. J'ai pu me former aux techniques de traitements d'images et aux outils utilisés tels que Pandore. Mais la théorie découverte par la lecture d'articles et de livres m'a permis d'avoir une vision encore plus globale des principes de cette discipline. Les nombreux domaines abordés durant ce stage, que ce soit évidemment en traitement d'images, mais aussi tout ce qui approche la fouille de données, le traitement du signal, les mathématiques, et bien d'autres encore, m'ont ouverte à des problématiques de recherche passionnantes. L'aspect le plus enrichissant de ce stage est encore l'interdisciplinarité qui existe entre la science informatique et les autres communautés de recherche telles que les sciences humaines ou naturelles. En conclusion, cette expérience m'a permis d'orienter et de confirmer la suite de mon parcours professionnel vers la recherche en informatique, à la fois dans son aspect théorique et abstrait mais également dans son aspect pratique lorsqu'elle peut aider à la résolution de problèmes concrets.

Glossaire

- **apprentissage automatique** : ou *machine learning*, méthode qui permet à une machine d'acquérir des connaissances. L'apprentissage peut être supervisé (si les classes et les exemples sont prédéterminés et connus), non-supervisé (lorsque les exemples ne sont pas étiquetés et le nombre de classes non prédéterminées), ou semi-supervisé (si certains exemples ne sont pas étiquetés).
- **bruit numérique** : présence d'informations parasites qui s'ajoutent de façon aléatoire aux détails de la scène photographiée numériquement. (Source : Wikipédia)
- **classification automatique** : catégorisation algorithmique d'objets permettant d'attribuer une classe ou catégorie à chaque objet en fonction de données représentatives.
- **composante connexe** : ou région, groupe de pixels connectés selon voisinage donné.
- **descripteur** : ensemble de signes donnant une description d'un fichier, d'une variable, etc. (Source : Le Petit Larousse 2006)
- **image** : ensemble plan de points ou d'éléments (pixels) représentatifs de l'apparence d'un objet, formés à partir du rayonnement émis, réfléchi, diffusé ou transmis par cet objet. (Source : Le Petit Larousse 2006)
- **image numérique** : image (dessin, icône, photographie, etc.) acquise, créée, traitée ou stockée sous forme binaire. (Source : Wikipédia)
- **moment (d'ordre n)** : pour une variable statistique, moyenne des puissance nièmes de ces valeurs pondérée par les effectifs de leurs classes respectives. (Source : Le Petit Larousse 2006)
- **moment géométrique invariant** : moment dont la valeur ne change

pas ou peu lors de transformations géométriques telles que la rotation, la translation ou le changement d'échelle.

- **numérisation** : action de représenter sous forme numérique une information analogique.
- **signal** : en télécommunication, variation d'une grandeur physique d'une grandeur quelconque porteuse d'information. (Source : Le Petit Larousse 2006)
- **signature** : en télédétection, ensemble des éléments qui permettent de caractériser la présence, le type, ou l'identité d'un objet par un senseur. (Source : Wikipédia)
- **spatio-temporel** : se dit du continuum espace-temps qui comporte quatre dimensions : trois dimensions x,y,z pour l'espace et une pour le temps, t.
- **traitement de la vidéo** : branche du traitement du signal qui s'applique à un signal vidéo, dans le but notamment d'en améliorer la qualité, de le compresser, ou d'en extraire de l'information. (Source : Wikipédia)
- **traitement d'image** : discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. (Source : Wikipédia)
- **vidéo** : l'ensemble des techniques concernant la formation, l'enregistrement, le traitement ou la transmission d'images ou de signaux de type télévision. (Source : Le Petit Larousse 2006)

Table des figures

| | | |
|-----|--|----|
| 2.1 | Différents types d'images. (a) photographie (réflexion); (b) image sonar (réflexion); (c) image par microscopie électronique (réflexion); (d) image thermique ou infrarouge (émission); (e) radiographie (absorption). (a) Crédits : Luc Viatour; (c) extrait de <i>www.kidsgeo.com</i> ; (d) Crédits : Cynthia Goldsmith. | 6 |
| 2.2 | Exemples de dimensions de différents signaux - illustration librement inspirée du livre <i>The Essential Guide to Image Processing</i> de Al Bovik. | 7 |
| 2.3 | Exemples de différents espaces de couleur. (a) image binaire; (b) image en niveaux de gris; (c) image en couleur - Crédits photographie originale : Jean-Louis Vandevivère. | 8 |
| 2.4 | Appareil d'acquisition d'image binaire. | 9 |
| 2.5 | Résultat de l'application de l'étiquetage de région. (a) image binarisée; (b) image étiquetée. | 10 |
| 2.6 | Moyenne d'images représentant le chiffre manuscrit 0. | 11 |
| 2.7 | Différence de deux images. Les pixels blancs dans le résultat (c) indiquent les changements entre l'image (a) et l'image (b). | 12 |
| 3.1 | Tracés des polynômes de Zernike sur le disque unité. | 17 |
| 4.1 | Exemple de traitement d'image avec Pandore. (a) image originale; (b) image binarisée. | 24 |
| 4.2 | Exemples d'images extraites de MNIST. | 25 |
| 4.3 | Exemple d'examen. (a) les dix chiffres; (b) les treize premières lettres de l'alphabet cursif majuscule; (c) les treize dernières lettres de l'alphabet cursif majuscule. | 26 |
| 4.4 | Exemples de traitements d'images d'un examen. (a) image en niveaux de gris; (b) image binarisée; (c) et (d) images segmentées en régions étiquetées. | 27 |
| 4.5 | Modèles des chiffres par la moyenne. | 29 |

| | |
|--|----|
| <i>TABLE DES FIGURES</i> | 39 |
| 4.6 Images utilisées pour les tests. | 30 |

Liste des tableaux

| | | |
|-----|---|----|
| 4.1 | Résultats moments de Zernike 2d sur la rotation, l'échelle et la translation. | 31 |
| 4.2 | Moments de Krawtchouk et reconstruction d'images 2d. | 33 |

Bibliographie

- [1] N. Abu Bakar, S. Shamsuddin, and A. Ali. An integrated formulation of zernike representation in character images. In Nicolás García-Pedrajas, Francisco Herrera, Colin Fyfe, José Benítez, and Moonis Ali, editors, *Trends in Applied Intelligent Systems*, volume 6098 of *Lecture Notes in Computer Science*, pages 359–368. Springer Berlin / Heidelberg, 2010.
- [2] R. A. Altes. The fourier mellin transform and mammalian hearing. *The Journal of the Acoustical Society of America*, 63(1) :174–183, 1978.
- [3] G. Badrinath, Naresh Kachhi, and Phalguni Gupta. Verification system robust to occlusion using low-order zernike moments of palmprint sub-images. *Telecommunication Systems*, 47 :275–290, 2011.
- [4] Y. Benezeth, B. Emile, H. Laurent, and A. Hafiane. Reconnaissance par blocs : une alternative aux descripteurs locaux. In *GRETSI*, 2009.
- [5] A. C. Bovik. *The Essential Guide to Image Processing*. Academic Press, 2009.
- [6] Al Bovik. *The Essential Guide to Video Processing*. Academic Press, 2009.
- [7] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Simul*, 4 :490–530, 2005.
- [8] M. Charles, R. Soppelsa, and J.-M. Albaret. *BHK - Échelle d'évaluation rapide de l'écriture chez l'enfant*. 2003.
- [9] V. Chen, Szabo A., and M. Roussel. Recherche d'images iconique utilisant les moments de zernike. In *Actes de CORESA'03*, 2003.
- [10] A. El affar, K. Ferdous, A. Cherkaoui, H. El fadili, and H. Qjidaa. Krawtchouk moment feature extraction for neural arabic handwritten words recognition. *IJCSNS International Journal of Computer Science and Network Security*, 9(1) :417–423, January 2009.
- [11] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Boughleux. Non-local Discrete Regularization on Weighted Graphs : a framework for

- Image and Manifold Processing. *IEEE Transactions on Image Processing*, 17(7) :1047–1060, 2008. 14 pages This research work was partially supported by the ANR foundation under grant ANR-06-MDCA-008-01/FOGRIMMI.
- [12] P. Esther Rani and R. Shanmuga Lakshmi. Palmprint recognition system using zernike moments feature extraction. In Vinu V Das and R. Vijaykumar, editors, *Information and Communication Technologies*, volume 101 of *Communications in Computer and Information Science*, pages 449–454. Springer Berlin Heidelberg, 2010.
- [13] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1) :100 – 132, 1985.
- [14] M. K. Hu. Visual pattern recognition by moments invariants. *IRE Transactions on Information Theory*, 8 :179–187, 1962.
- [15] Anil K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
- [16] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12 :489–497, May 1990.
- [17] I. Lassoued, E. Zagrouba, and Y. Chahir. An efficient approach for video action classification based on 3d zernike moments. In James J. Park, Laurence T. Yang, and Changhoon Lee, editors, *Future Information Technology*, volume 185 of *Communications in Computer and Information Science*, pages 196–205. Springer Berlin Heidelberg, 2011.
- [18] Y. Lecun and C. Cortes. The MNIST database of handwritten digits.
- [19] S. Liao and M. Pawlak. A study of zernike moment computing. In Roland Chin and Ting-Chuen Pong, editors, *Computer Vision — ACCV’98*, volume 1351 of *Lecture Notes in Computer Science*, pages 394–401. Springer Berlin / Heidelberg, 1997.
- [20] Krawtchouk M. On interpolation by means of orthogonal polynomials. *Memoirs Agricultural Inst. Kyiv*, 4 :21–28, 1929.
- [21] A. Mademlis, A. Axenopoulos, P. Daras, D. Tzovaras, and M. G. Strintzis. 3d content-based search based on 3d krawtchouk moments. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06)*, 3DPVT ’06, pages 743–749, Washington, DC, USA, 2006. IEEE Computer Society.
- [22] M. Nixon and A. S. Aguado. *Feature Extraction & Image Processing, Second Edition*. Academic Press, 2 edition, January 2008.

- [23] Marcin Novotni and Reinhard Klein. Shape retrieval using 3d zernike descriptors. *Computer Aided Design*, 36(11) :1047–1062, 2004.
- [24] Pandore : Une bibliothèque d’opérateurs de traitement d’images (Version 6.4). [Logiciel]. Laboratoire Greyc. <http://www.greyc.ensicaen.fr/regis/Pandore>, 2011. [consulté le 18/04/2011].
- [25] D. Phillips. *Image processing in C*. Technical books. R & D Publications, Inc., 1994.
- [26] J. Revaud, G. Lavoué, and A. Baskurt. Une nouvelle mesure de distance entre descripteurs de moments de zernike pour une similarité optimale et un angle de rotation entre les images. In *CORESA*, March 2009.
- [27] R. Swiniarski. Rough sets and neural networks application to handwritten character recognition by complex zernike moments. In Lech Polkowski and Andrzej Skowron, editors, *Rough Sets and Current Trends in Computing*, volume 1424 of *Lecture Notes in Computer Science*, pages 617–624. Springer Berlin / Heidelberg, 1998.
- [28] M. R. Teague. Image analysis via the general theory of moments*. *J. Opt. Soc. Am.*, 70(8) :920–930, Aug 1980.
- [29] H. Trocmé-Fabre. *J’apprends, donc je suis : Introduction à la neuropédagogie*. Editions d’Organisation, 1993.
- [30] H. Joel Trussell and Manish S. Kulkarni. Sampling and processing of color signals. *IEEE Transactions on Image Processing*, 5(4) :677–681, 1996.
- [31] V. Venkatraman, P. Chakravarthy, and D. Kihara. Application of 3d zernike descriptors to shape-based ligand similarity searching. *Journal of Cheminformatics*, 1(1) :19, 2009.
- [32] B. A. Wandell. The synthesis and analysis of color images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9 :2–13, January 1987.
- [33] Wikipédia. Circuit sensori-moteur — wikipédia, l’encyclopédie libre, 2010. [En ligne ; Page disponible le 29-juillet-2011].
- [34] Wikipédia. Polynômes de zernike — wikipédia, l’encyclopédie libre, 2010. [En ligne ; Page disponible le 14-juillet-2011].
- [35] P. T. Yap, P. Raveendran, and S. H. Ong. Krawtchouk moments as a new set of moments for image reconstruction. In *IJCNN’02*, volume 1, pages 908–912, 2002.
- [36] L. Zhang, W.W. Xiao, and Z. Ji. Local affine transform invariant image watermarking by krawtchouk moment invariants. *IET Inf. Secur.*, 1(3), 2007.

Index

- échelle, 13, 30
- écriture, 3, 25, 27
- apprentissage, 24
- block matching, 12
- bruit, 10, 13, 29
- caractéristique, 2, 13, 15
- caractère, 3
- classe, 13
- classification, 3, 21
- composante connexe, 9, 26
- decripteur, 3
- descripteur, 13, 15
- dimension, 7, 12, 13, 28, 33
- examen, 25
- flot optique, 12
- graphe, 21
- histogramme, 9, 23
- Hu, 15
- image, 2, 5–7, 15, 23, 25
- invariant, 13
- Krawtchouk, 18, 33
- modèle, 11, 29
- moment, 13, 15, 17, 30, 33
- morphologie, 10
- mouvement, 11
- moyenne, 10, 29
- numérisation, 7
- pixel, 7
- réduction, 13
- région, 9, 26
- régularisation, 30
- reconnaissance, 11, 15, 24
- rotation, 13, 30
- segmentation, 9, 25
- signal, 2, 7
- signature, 13
- similarité, 13, 21
- spatio-temporel, 12, 28
- temps réel, 3, 13
- tracking, 11
- traitement d'image, 1, 5, 6, 8, 9, 23
- traitement de vidéo, 12
- transformation, 13
- translation, 13, 30
- variabilité, 11, 30
- vecteur, 13
- vidéo, 2, 7, 11, 12
- vision, 2, 5, 15
- Zernike, 15, 30