Université de Nancy I — U.E.R. Sciences de la matière

Centre de Recherche en Automatique de Nancy (URA 821)
Laboratoire d'Electricité et d'Automatique

THESE

PRESENTEE A
L'UNIVERSITE DE NANCY I

pour obtenir

LE TITRE DE DOCTEUR D'UNIVERSITE
Spécialité : AUTOMATIQUE
par

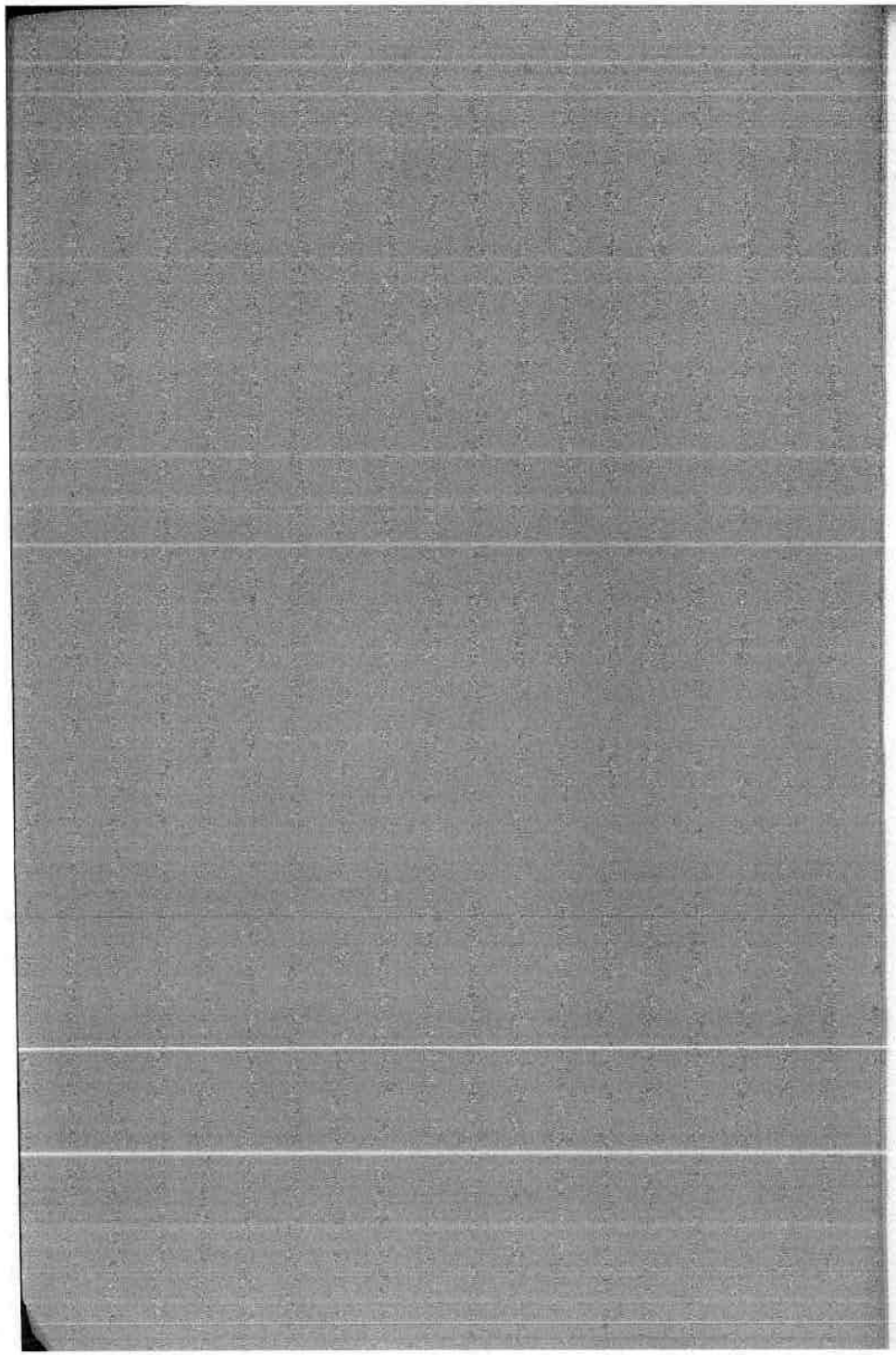
Mme Souâd MIRZAQ-BENZIDIA

DETERMINATION AUTOMATIQUE DES PARAMETRES PERTINENTS D'UNE IMAGE BINAIRE EN VUE D'UNE RECONNAISSANCE DE FORME

Présentée et soutenue le 25 Juin 1990

JURY:

Président	:	**C.HUMBERT**
Rapporteurs	:	**M.DAROUACH**
	:	**J.RAGOT**
Examinateurs	:	**J.BREMONT**
		M.LAMOTTE



THESE

PRESENTEE A
L'UNIVERSITE DE NANCY I

pour obtenir

LE TITRE DE DOCTEUR D'UNIVERSITE
Spécialité : AUTOMATIQUE

par

Mme Souâd MIRZAQ-BENZIDIA

DETERMINATION AUTOMATIQUE DES PARAMETRES PERTINENTS D'UNE IMAGE BINAIRE EN VUE D'UNE RECONNAISSANCE DE FORME

Présentée et soutenue le 25 Juin 1990

JURY:

Président	:	**C.HUMBERT**
Rapporteurs	:	**M.DAROUACH**
	:	**J.RAGOT**
Examinateurs	:	**J.BREMONT**
		M.LAMOTTE

Je dédie ce travail

A mon père, à ma mère

A ma petite fleur Ilham

A mon mari Aziz

A ma famille

Remerciements

Je tiens à remercier le Professeur ***M. LAMOTTE*** *pour le temps qu'il a consacré à mes travaux, ainsi que pour le suivi de cette thèse.*

Je tiens à remercier le Professeur ***J. BREMONT*** *pour les idées apportées lors de la correction ainsi que pour toutes les démarches administratives.*

Je remercie le Professeur ***C. HUMBERT*** *de nous avoir fait l'honneur d'être le Président du jury de cette thèse.*

Je remercie le Professeur ***M. DAROUACH*** *et le Professeur* ***J. RAGOT*** *pour avoir accepté d'être les rapporteurs de ma thèse.*

Je remercie ***F. SMIEJ****, pour sa gentillesse, ses conseils et pour le temps précieux qu'il m'a accordé.*

Mes sincères remerciements à ***M. DENIS****, pour l'aide apportée au niveau de la présentation de ce mémoire.*

TABLE DES MATIERES

Chapitre II : LES DIFFERENTES METHODES DE CLASSIFICATION ET DE SELECTION

Chapitre III : EXPERIMENTATION

INTRODUCTION

La sélection de paramètres, la classification et la reconnaissance de forme sont des idées et des techniques déjà anciennes, comme peut en témoigner la littérature. Le but de cette thèse est la création d'un produit (logiciel enchaînant ces différentes techniques), avec une mise en œuvre légère en vue d'une application dans un atelier de production.

Ce logiciel doit réaliser de façon automatique le choix des paramètres les plus pertinents sur des images binaires, en fonction des différents lots d'apprentissage proposés et du type de classification souhaitée. Le caractère automatique de ce logiciel lui permettra ainsi de pouvoir s'adapter à n'importe quel type de problème de reconnaissance posé.

Un processus de reconnaissance peut être considéré comme la succession de plusieurs étapes.

- Acquisition des données
- Extraction des paramètres (mesures faites sur les données)
- Apprentissage
- Classification
- Reconnaissance

Nous commencerons donc par définir ces différentes étapes.

Acquisition des données

Selon la nature des données (parole, image, son, musique....), l'acquisition est réalisée par différents capteurs (caméra, magnétophone...). Les capteurs fournissent un signal analogique, qui sera par la suite digitalisé.

Ces données sont bien sûr entachées d'erreurs, dues à différentes causes: mauvais éclairage, résolution limitée du capteur, dérive et bruit de fond des circuits électroniques, imperfection de la traduction analogique-numérique, variations géométriques apparentes des contours selon leurs positions dans le champ image, etc. Pour cette raison les données ne sont pas toujours exploitables sous leurs formes brutes et nécessitent ***des traitements de mise en forme.*** Par exemple, pour limiter le bruit, on pourra réaliser un filtrage, ou encore pour une image peu contrastée, réaliser un rehaussement de contraste.

Extraction des paramètres

Des méthodes numériques sont alors mises en œuvre pour extraire les paramètres caractéristiques des formes. On pourra par exemple dans le cas d'une image, réaliser des mesures géométriques telles que: le périmètre, la surface, les moments, le rayon moyen, etc....Dans le cas du signal de la parole, les mesures sont pour la plupart obtenues après passage du domaine temporel au domaine spectral grâce à la transformée de Fourier (formants, fréquence fondamentale...etc).

Apprentissage

Pour qu'un système puisse réaliser une reconnaissance de forme, il est nécessaire de lui fournir un certain nombre d'informations. La plus importante étant la description des classes. Cette dernière peut être réalisée soit en précisant les limites, les frontières séparant ces classes, soit en fournissant un ou plusieurs représentants pour chacune des classes. Ces informations sont données dans une étape dite d'apprentissage.

On distingue deux grands types d'apprentissage: l'apprentissage supervisé ou "avec moniteur" et l'apprentissage non supervisé ou "sans moniteur".

L'apprentissage supervisé nécessite la connaissance "a priori" des différentes classes, donc du nombre de classes. Par exemple, si l'on veut réaliser un tri de pièces mécaniques, le nombre et la forme des différentes pièces sont connus au départ. Il suffira donc d'effectuer des mesures initiales sur chacune des pièces connues pour obtenir le représentant de la classe. La tâche du "moniteur" est donc réduite, elle consiste à fournir à la machine un critère, qui permettra à cette dernière de classer une forme inconnue dans l'un des groupes (classe, cluster) préalablement définis.

Dans le cas d'apprentissage non-supervisé (ou sans moniteur), le dispositif automatique établit de lui-même les différentes classes et détermine les paramètres du "classifieur" de façon simultanée. Le rôle du " moniteur " consiste alors à fournir les critères de décisions retenus pour séparer les différentes classes. L'apprentissage non-supervisé est une opération beaucoup plus délicate à réaliser que la procédure supervisée. D'une part le nombre de classes n'est pas connu à l'avance et d'autre part les caractéristiques de chacune d'elles sont inconnues dans la plupart des cas. Une méthode de reconnaissance, dans ce cas, consiste à rechercher les groupements possibles au sens d'une meilleure compacité.

Dans la plupart des cas c'est au cours de l'apprentissage, que la phase de classification est réalisée. La confusion entre les deux étapes de classification et de reconnaissance de forme existe dans de nombreux articles.

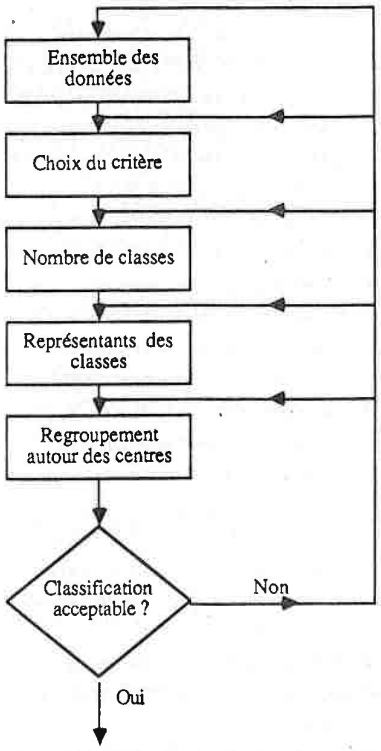


fig.1 : Organigramme d'un Apprentissage non-supervisé

La classification

Encore appelée "clustering" est l'étape durant laquelle les différentes classes de formes sont définies. Ces classes seront alors représentées soit par un point forme (centre de gravité des formes de cette classe), ou encore par un ensemble de points[DIDA-70].

La reconnaissance de forme

Les classes et leurs représentants étant définis correctement à l'étape précédente, ainsi que les paramètres pertinents, la reconnaissance peut commencer. Elle constitue la dernière étape, étape durant laquelle chaque forme va être affectée définitivement à l'une des classes.

I) Position du problème

Quel que soit le domaine, le nombre de données à traiter est souvent très important. Aussi pour obtenir une classification relativement rapide de ces données il faut dans un premier temps, minimiser ce nombre. Pour limiter le nombre de données à traiter, il suffira de réduire le nombre des paramètres représentant une forme; et pour cela soit supprimer les paramètres, qui contribuent faiblement, voire pas du tout à la classification, soit conserver uniquement les paramètres les plus pertinents de sorte à maintenir une classification correcte des formes.

En effet, lors de la conception d'un système de reconnaissance de forme, le critère de bonne classification ne peut être le seul à être pris en compte, le coût de la classification doit également être considéré. Ce coût fait intervenir le coût de la mesure des paramètres, ainsi que le temps de calcul. Il est souhaitable alors de réduire la dimension de l'espace des formes en éliminant les paramètres, qui n'apportent pas ou presque pas d'informations utiles pour la discrimination des classes.

Suivant la nature des données initiales, un paramètre peut être très pertinent pour un problème et pas du tout pour un autre. Par exemple, supposons que l'on dispose de deux lots de formes à classer : le premier constitué de quatre formes très dissemblables: un carré, un cercle, un triangle et une croix, ces quatre formes ayant même périmètre, le deuxième lot constitué de quatre cercles de rayon différent. Il est évident que le paramètre "périmètre" contribuera fortement à la classification du deuxième lot et pas du tout à celle du premier lot. D'où l'idée d'automatiser la procédure de sélection de paramètre, c'est-à-dire d'établir une procédure où le choix des paramètres les plus pertinents serait réalisé d'une manière automatique en fonction des données initiales et de l'objectif de la reconnaissance.

II) Exposé de la méthode proposée

Notre travail, basée sur la reconnaissance de forme est donc de réaliser un programme capable d'opérer d'une manière autonome la sélection des paramètres pertinents, parmi un catalogue de paramètres, puis d'effectuer la classification des formes à partir des paramètres sélectionnés, et ceci quel que soit le lot d'images proposé. La méthode sera appliquée à la classification d'images binaires.

Bien que des images à niveaux de gris soient plus proches de la réalité, nous avons opté pour des images binaires et ceci pour deux raisons :

- la première est que des images à niveaux de gris peuvent se ramener rapidement par simple seuillage à des images binaires;

- la deuxième et la plus importante, réside dans le but que nous nous sommes fixés. En effet, nous nous sommes basés sur une reconnaissance de formes de pièces mécaniques. Seul dans ce cas l'aspect géométrique de la forme intervient dans la classification, donc des images à niveaux de gris n'apportent aucune contribution au sujet de notre thèse.

Cependant, il est à souligner que dans le cas d'images à niveaux de gris, d'autres paramètres peuvent être calculés et ajoutés aux précédents, sans remettre en question la suite de notre travail (sélection, classification, reconnaissance de forme).

Dans une première phase nous avons donc créé des lots d'images synthétiques binaires 40x40. Chaque image constitue un fichier. Puis nous avons réalisé plusieurs sous-programmes de mesures géométriques de manière à obtenir un catalogue relativement important de paramètres. Nous avons programmé quelques méthodes de sélection de paramètres, de classification et de reconnaissance de forme, que nous avons testées.

III) Historique

III.1) Introduction aux différentes méthodes de reconnaissance de forme

La reconnaissance de forme, par la diversité et le nombre de ses applications est un domaine, qui a passionné de nombreux scientifiques. La diversité des méthodes rencontrées dans la littérature en témoigne.

D'une façon très simple les problèmes de reconnaissance de forme peuvent être considérés comme des problèmes de partage de données en classes, dans le but de satisfaire à un certain critère, en général celui d'approcher le partage subjectif de ces données par un être humain. Elle est en général constituée de deux étapes:

- la description des formes,
- la reconnaissance de ces formes.

La description consiste à extraire des paramètres caractéristiques des formes, qui donneront une représentation simplifiée de la forme. Un exemple simple, dans le domaine de l'image est la reconnaissance de forme de pièces mécaniques. On peut par exemple mesurer la longueur du contour, la surface; on peut également compter le nombre de trous, etc.

La reconnaissance de forme consiste à affecter chaque forme à une classe à partir des valeurs des différents paramètres obtenues à l'étape précédente. Différentes méthodes ont été proposées.

C'est au début des années 60, que les méthodes de classification par hypersurfaces font leur apparition. Un espace forme est introduit, espace rapporté aux

axes correspondant aux différents paramètres. Toute forme X est alors représentée par un point dans cet espace à n dimensions. La classification des données est réalisée par la recherche d'hypersurfaces, qui vont scinder, séparer cet espace de dimension n en régions constituant les différentes classes.

Durant cette même période TZYPKIN, s'appuyant sur la théorie des probabilités et tout particulièrement sur la théorie de Bayes, les notions de coût et de probabilité d'erreur, introduit les méthodes probabilistes.

Vers 1964, les idées de CHOMSKY sur les grammaires formelles sont appliquées à la reconnaissance de forme. K.S. FU relie les idées de Chomsky aux théories probabilistes et propose la notion de grammaire stochastique.

Les méthodes séquentielles font leur apparition dans le domaine de la reconnaissance de forme grâce à K.S. FU[FU-67].

WATANABE et COOPER introduisent à la même époque les notions de classificateur sans-professeur (non-supervisé).

On assiste aussi au développement de la reconnaissance syntaxique; Alain Faure dans son livre " Perception et reconnaissance de forme", la définit de la sorte:

"Une toute autre approche de la reconnaissance peut être effectuée en considérant la notion de structure et de relation entre les formes examinées, ou leur représentation. Par postulat, on admet que la construction de ces formes est le fruit de l'application de règles bien définies, qu'il faut naturellement déterminer dans un premier temps. La procédure de classification et d'identification en découle alors simplement et se décompose en deux étapes principales.

La première étape est la détermination des règles supposées de construction; c'est-à-dire la recherche d'une sorte de grammaire au sens linguistique du terme. La grammaire étant terminée, la seconde étape de reconnaissance à proprement dite, consiste à rechercher si une forme appartient à l'ensemble de toutes les formes générées par cette grammaire....." [FAUR-85]

C'est aussi la période de développement de la théorie floue en reconnaissance de forme [BEZD-71] [BEZD-83].

Il est évident que quelle que soit la méthode utilisée, la classification ne peut être correctement réalisée qu'à la condition que les paramètres extraits durant la phase de description soient représentatifs de la forme. Pour cette raison, à cette étape, le nombre de paramètres extraits est souvent très important. Cette étape est alors suivie d'une étape de recherche des paramètres pertinents: c'est la phase de *sélection de paramètres*.

III.2) Sélection de paramètres

Lors de la classification des formes, le nombre de données à traiter est souvent très important. Ceci pour deux raisons, d'abord parce que le nombre des formes à traiter est grand, mais aussi parce que le nombre de paramètres utilisés pour la représentation de chaque forme l'est aussi. Ceci entraîne un nombre de traitements, donc un temps de calcul, mais aussi un encombrement mémoire important, voire quelques fois expérimentalement impossible.

Or les paramètres représentant les formes peuvent être regroupés en trois catégories.

- Les paramètres pertinents :

Ce sont ceux qui caractérisent correctement les classes souhaitées, et donc permettront une classification et une reconnaissance aisées.

- Les paramètres peu "informatifs" :

Certains sont peu "informatifs" pour la classification, parce que leurs valeurs diffèrent très peu d'une classe à une autre, ou encore parce qu'ils sont redondants par rapport à d'autres paramètres déjà utilisés.

- Les paramètres "perturbateurs" :

Nous entendons par paramètres "perturbateurs", ceux qui vont nous donner une partition totalement différente de celle attendue. Ce terme est absent dans la littérature, pour la simple raison que lors d'une sélection de paramètres la classification souhaitée est connue. L'opérateur aura effectué lui même une sélection de paramètres. En effet s'il souhaite classer des formes d'après leur aspect géométrique (cercle, carré, triangle....), il prendra des paramètres tels que le cœfficient de compacité, les moments invariants...et non pas la surface, le périmètre, la couleur...Par contre s'il veut une classification d'après leur taille (petit, moyen, grand) l'opérateur choisira des paramètres tels que le périmètre, la surface. L'opérateur en fonction de la classification souhaitée va réaliser inconsciemment une première sélection de paramètres, et va supprimer les paramètres perturbateurs. Or lors de la conception d'un processus automatique, le type de classification n'est pas précisé a priori. L'ensemble initial des paramètres potentiels doit contenir un nombre important et divers de paramètres. Pour chaque type de classification, le processus doit alors retrouver parmi ces paramètres ceux qui correspondent à ce type de classification et donc supprimer ceux correspondant aux autres types de classification. Par exemple supprimer le paramètre couleur pour une classification se basant uniquement sur l'aspect géométrique (carré, cercle, triangle). Le paramètre couleur peut donc être considéré ici comme un paramètre perturbateur (dans le cas où un carré, un cercle et un triangle sont de la même couleur) .

L'idée principale des méthodes de sélection de paramètres est de réduire l'ensemble initial des paramètres, soit en supprimant les paramètres perturbateurs et les

paramètres les moins "informatifs", soit en sélectionnant individuellement ou par groupes les paramètres les plus informatifs, ou encore soit en réalisant des combinaisons linéaires des paramètres initiaux. Les deux premières alternatives peuvent sembler être identiques, il n'en est rien. En effet, la suppression des "mauvais" paramètres n'entraîne pas forcément que l'ensemble des paramètres restants soit le meilleur ensemble, puisque la combinaisons de deux paramètres dits "mauvais" peut donner un "bon" sous-ensemble. Ces variantes seront discutées dans le chapitre "Méthodes de sélection".

Le sous-ensemble sélectionné devra bien entendu réaliser une classification très voisine de la classification souhaitée par l'opérateur.

Un paramètre " informatif " est ici défini comme un paramètre possédant à la fois

- un grand pouvoir de discrimination de classes,
- une faible corrélation avec les autres paramètres sélectionnés.

Pour exprimer ces deux propriétés, les scientifiques ont défini des mesures aussi nombreuses que variées (dispersion interclasse-dispersion intraclasse).

Les méthodes de sélection de paramètres peuvent être considérées comme une transformation d'un premier espace forme de dimension n (où chaque forme est représentée par le vecteur paramètre) à un espace de dimension m (m<n), souvent appelé espace des formes réduit (des paramètres).

Le problème de la sélection de paramètres a été abordé par plusieurs auteurs et de différentes manières. En 71 Mucciardi et Gose propose un résumé et une comparaison intéressante des différentes méthodes existantes[MUCC-71].

K.S. FU en 1967, propose une solution au problème: les méthodes séquentielles [FU-66]. Ces méthodes ne peuvent s'appliquer que dans le cas d'une reconnaissance séquentielle. Elles consistent à affecter à chaque paramètre une fonction "qualité ", d'ordonner les paramètres par ordre décroissant de qualité et de les sélectionner un par un et dans cet ordre au cours de la phase de reconnaissance jusqu'à obtenir une probabilité d'erreur inférieure à un seuil prescrit (d'où le nom de séquentiel). L'inconvénient de ces méthodes est qu'elles sélectionnent les paramètres individuellement. En effet le sous ensemble des m meilleurs paramètres ne donne pas forcément une bonne classification, car on ne tient compte alors que du pouvoir discriminatoire et pas de la corrélation entre les paramètres.

L'idée est alors de rechercher le meilleur sous-ensemble de m paramètres et non pas les m meilleurs paramètres. La solution à ce problème serait évidemment de considérer toutes les combinaisons possibles de m paramètres parmi n et de prendre celle qui fournirait la bonne classification. Mais cette solution, optimale certes, est au niveau du temps de calcul et de l'emplacement mémoire pratiquement irréalisable dans de nombreux cas. Par exemple pour extraire, 6 paramètres parmi 12, il faudrait tester la

reconnaissance sur C^6_{12} = 924 combinaisons possibles, pour extraire 12 paramètres parmi 24 il faudrait tester C^{12}_{24} =2 704 156 combinaisons possibles.

Narendra et Fukunaga ont proposé un algorithme optimal: " *The branch and bound algorithm* " , qui réduit considérablement le nombre des combinaisons à évaluer [NARE-77]. Le principe de la méthode est de rechercher les paramètres à supprimer pour obtenir le meilleur sous-ensemble. Cette opération est réalisée à l'aide d'un arbre. Chaque nœud représente le paramètre à supprimer. Ainsi les différentes branches représentent toutes les combinaisons possibles de (n-m) paramètres à supprimer. Le point essentiel de cette méthode, est la position de chaque paramètre dans l'arbre, position déterminée en fonction de la valeur du critère de bonne classification. Cette méthode nécessite un critère qui décroît lorsqu'on retire un paramètre du sous-ensemble donné et qui croit lorsqu'on y ajoute un paramètre. Malheureusement ce n'est pas le cas de tous les critères. La méthode est décrite dans le chapitre II paragraphe 3.3.

D'autres auteurs ont proposé des méthodes basées sur la programmation dynamique [NELS-68] ; [CHEU-78] ; [CHAN-73] ; [FORO-86]. Ces méthodes permettent aussi de diminuer considérablement le nombre de combinaisons à évaluer. La programmation dynamique est une technique d'optimisation à plusieurs étapes. A la première étape, on considère tous les paramètres (n), chaque paramètre est alors combiné aux (n-1) paramètres et l'on retient pour chaque paramètre le couple le plus performant. A la deuxième étape, à chaque couple sélectionné on combine les (n-2) paramètres et on retient pour chaque couple le triplet le plus performant. Le processus est réitéré jusqu'à l'obtention du nombre souhaité de paramètres. Le sous-ensemble sélectionné sera celui qui, à la dernière étape aura la valeur maximale pour le critère choisi.

Une autre catégorie de méthodes, regroupées sous le nom de méthodes linéaires, ou encore méthodes d'analyse en composantes principales, a bénéficié d'un grand intérêt pour les problèmes de sélection de paramètres [DECE-79] [YOUNG-83] [YOUNG-85] [BIDA-86]. La réduction de la dimension de l'espace forme est réalisée par la détermination de la transformation linéaire, à appliquer aux vecteurs paramètres initiaux. En résumé ces méthodes considèrent le problème comme le passage d'une base de dimension n à une base de dimension m. Les vecteurs de cette nouvelle base correspondent aux vecteurs propres associés aux plus grandes valeurs propres dans la plupart des cas de la matrice de covariance des données initiales.

La méthode de Karhunen-Lœve est l'une des plus connues et des plus répandues dans la littérature [FUKU-70] ; [MARW-83]. En 83 P.C.Mali, B.B.Chauduri et D.Dutta proposent un algorithme utilisant la transformée de Walsh-Hadamard et comparent leur méthode à la méthode de Karhunen-Lœve [MALI-82]. Toujours dans cette catégorie, la méthode du discriminant linéaire de Fisher a été largement reprise et développée [MALI-87]. Cependant, pour les méthodes linéaires, un point important à noter est le fait que la

réduction du nombre de paramètres n'est effective que pour la phase de classification proprement dite et qu'il demeure nécessaire de mesurer tous les paramètres pour pouvoir calculer les nouveaux. Par conséquent, de telles méthodes ne seront pas envisagées dans notre étude.

D'autres techniques plus complexes ont été appliquées au problème de la sélection; Bezdeck utilise la théorie des ensembles flous [BEZD-71]. Segen propose une solution en se basant sur la théorie des méthodes syntaxiques [SEGE-84].

III.3) Présentation des différents chapitres

Supposons que l'on souhaite résoudre le problème suivant: On souhaite trier différentes pièces mécaniques, défilant sur un convoyeur. On dispose d'une caméra pour réaliser l'acquisition des images.

Pour faciliter les différents traitements, nous allons donc commencer par essayer d'améliorer au maximum la qualité des images; par exemple réduire le bruit ou encore rendre les contours des formes le plus net possible, on va donc appliquer aux images **"des traitements de mises en forme".**

Sur ces images "améliorées", " corrigées", nous réalisons alors différentes mesures géométriques telles que périmètre, surface, variance du rayon, cœfficient de compacité , etc....C'est l'étape **"d'extraction des paramètres"**. Ces différentes mesures seront les caractéristiques de chaque image. Des pièces de caractéristiques très voisines ont une grande probabilité d'appartenir à la même classe. Nous allons essayer maintenant à partir de ces mesures de trouver les différents types de pièces existantes (les différentes classes). Nous aurons alors réalisé une **"classification"**.

Les différentes classes étant établies, on peut alors se poser la question suivante : Toutes les mesures réalisées sont-elles vraiment bien nécessaires ?, ne pourrait-on pas réduire le nombre de paramètres, en ne conservant que les paramètres les plus "utiles" ?. Nous abordons alors le problème de **"la sélection de paramètres"**.

Pour vérifier notre choix de paramètres, nous pouvons maintenant essayer de reconnaître un grand nombre de pièces en n'utilisant uniquement que les paramètres sélectionnés lors de la phase précédente et voir si la " **reconnaissance**" obtenue est acceptable ou pas.

L'étude de cet exemple simple fournit les différents chapitres de notre mémoire. En effet le premier chapitre est consacré aux traitements d'images. Nous commencerons par donner quelques exemples de traitements de mises en forme, tels que la segmentation, le lissage, le suivi de contour.....; puis nous aborderons les méthodes de calculs de paramètres géométriques. Dans cette partie nous présenterons essentiellement les

paramètres que nous avons utilisés durant la phase expérimentale (périmètre, surface, moments, variance, rayon moyen.....).

Le deuxième chapitre est divisé en deux parties; la première décrit quelques méthodes de classification classiques; la deuxième concerne les problèmes de la sélection de paramètres.

Dans le troisième chapitre, nous présentons le travail réalisé, au laboratoire CRAN-LEA, au terme de cette étude. Nous avons donc simulé le problème précédent à l'aide de lots d'images binaires. Pour se rapprocher au maximum du contexte réel, nous avons également écrit un programme de simulation de bruit, pouvant faire intervenir ce dernier par différents pourcentages. Puis nous avons écrit un ensemble de sous-programmes permettant de calculer les différents paramètres de chaque image du lot considéré, que l'on aura préalablement bruitée ou pas. Ces mesures achevées, sont alors normalisées(nous proposons deux méthodes de normalisation).

La sélection des paramètres pertinents peut commencer. Nous expérimenterons trois méthodes dynamiques. Pour vérifier et comparer ces méthodes, nous avons écrit deux programmes :

- un programme de classification, pour vérifier si les classes obtenues avec uniquement les paramètres sélectionnés correspondaient aux classes initiales;
- un programme de reconnaissance de forme, pour vérifier si le taux de bonne reconnaissance avec les paramètres sélectionnés est acceptable.

Chaque programme fait l'objet de commentaires quant à la description et aux différents résultats.

CHAPITRE I

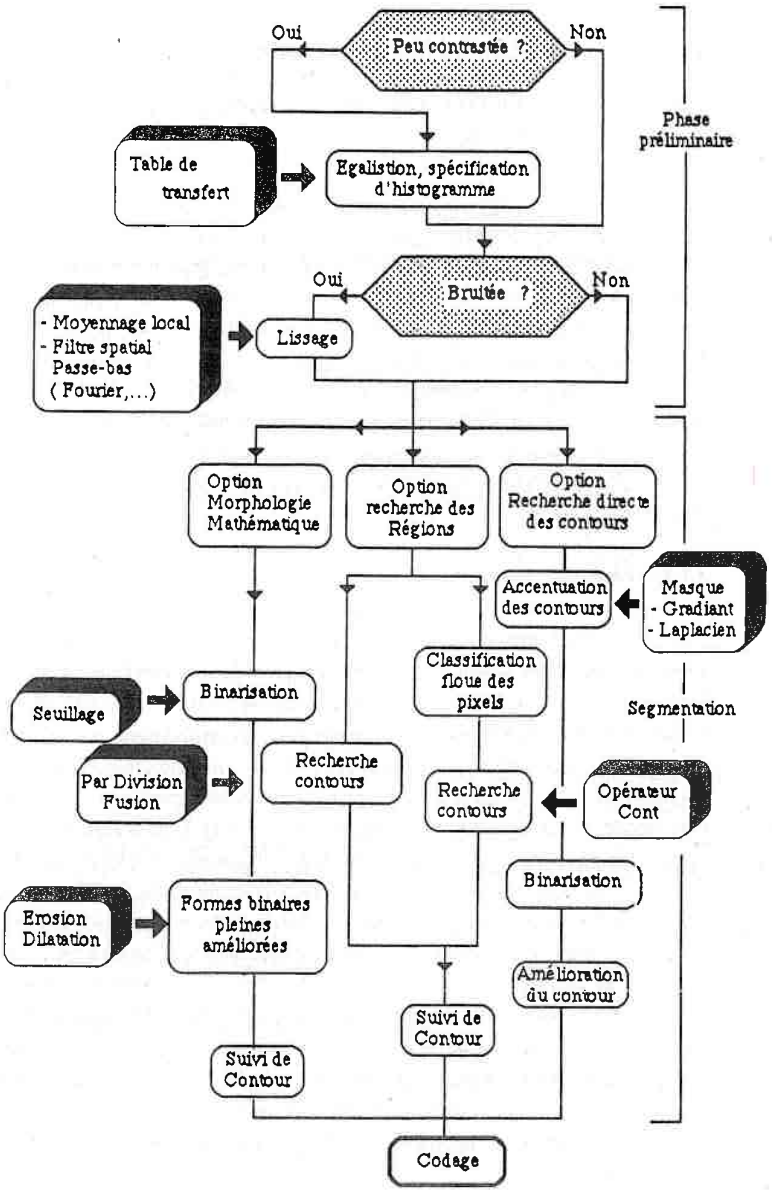
LES DIFFERENTS TRAITEMENTS D'IMAGES

Pour extraire les valeurs des paramètres géométriques ou autres, il faut dans un premier temps faire une série de traitements préliminaires dits " traitements de mise en forme ". Par exemple, si l'on souhaite obtenir la valeur du périmètre d'un objet, il est nécessaire que l'objet et le fond soient bien différenciés (segmentation), que l'image ne soit pas trop bruitée (masque pour supprimer le bruit). Mais l'objectif essentiel est l'obtention d'une image binaire dans laquelle les objets sont bien délimités par leur contour; d'où un problème de détection et de suivi de contour. Pour cette raison, nous avons scindé ce chapitre en deux parties. La première sera consacrée aux différents traitements de mise en forme, alors que la deuxième décrit les différents traitements de calcul des paramètres géométriques.

Comme nous le signalions dans notre introduction, d'autres paramètres peuvent être intégrés aux précédents. Par exemple, dans le cas d'images à niveaux de gris, on peut calculer des paramètres liés à ce dernier caractère tels que la moyenne, la variance, l'étendue ou l'intervalle interquartile des niveaux de gris.

I) Traitements de mise en forme

Nous proposons ci-dessous un schéma directeur selon lequel une séquence de traitements permet de passer d'une image à niveaux de gris à une image binaire. C'est alors que l'on pourra mesurer les paramètres géométriques. Ce schéma ne prétend pas résumer toutes les méthodes (et leurs variantes) qui sont souvent utiles pour s'adapter au mieux à l'extrême diversité des images.



I.1) Modification de contraste

Le principal objectif du rehaussement de contraste est de bien mettre en évidence la forme vis-à-vis du fond. Cette opération peut être réalisée par deux méthodes:

- la méthode spatiale,
- la méthode fréquentielle.

I.1.1) La méthode spatiale

La méthode consiste à modifier le niveau de gris de chaque pixel de l'image au moyen d'une table de transfert (qui traduit un lien entre les anciens et les nouveaux niveaux de gris). Il en résulte une modification de l'histogramme des niveaux de gris. La littérature [GONZ-77] mentionne deux méthodes voisines :

- l'égalisation d'histogramme, où l'on cherche à répartir équitablement les niveaux de gris sur toute la plage disponible, par exemple pour améliorer une image peu contrastée.
- la spécification d'histogramme, pour laquelle la répartition des niveaux de gris est imposée par l'opérateur, par exemple pour mettre en évidence des détails dans telle ou telle partie de l'image.

I.1.2) Méthodes fréquentielles

Des transformations orthogonales comme celle de Fourier ou de Walsh-Hadamard fournissent une décomposition spectrale de l'image sous forme de cœfficients qui tendent à isoler certains traits caractéristiques de l'image. Par exemple, le premier cœfficient a une valeur proportionnelle à la moyenne de l'intensité lumineuse de l'image et les cœfficients correspondant aux fréquences ou séquences élevées sont une mesure des frontières contenues dans l'image. On peut ainsi exploiter ces propriétés pour accentuer ou atténuer ces caractéristiques. En effet, ces transformations jumelées à des opérations de filtrage permettent de sélectionner, d'isoler les hautes fréquences ou basses fréquences correspondant respectivement à des variations brutales ou lentes de la luminosité de l'image. Un filtre passe-haut nous permettra d'obtenir les contours des différentes formes de l'image. Par contre, un filtre passe-bas va adoucir les transitions importantes.

Ces transformations sont riches potentiellement, mais lourdes d'un point de vue pratique. Avec les méthodes classiques, le calcul de telles transformations nécessite un temps de calcul prohibitif et un encombrement mémoire important. Cependant, la mise en œuvre d'algorithmes de calcul rapide a permis de réduire considérablement ce temps de calcul. Par exemple, pour la transformée de Fourier, on dispose de l'algorithme rapide FFT (en anglais : Fast Fourier Transform). Cet algorithme est aussi valable pour la transformée de Walsh-Hadamard.

Malgré cet effort et les possibilités croissantes des ordinateurs, la plupart de ces méthodes conservent le désavantage de ne pouvoir s'effectuer rapidement, c'est-à-dire en temps réel.

I.2 Segmentation

Ce paragraphe est consacré à la segmentation, en raison de son rôle capital et "évident" en reconnaissance de forme. En effet, avant de réaliser différentes mesures sur un objet, afin de le reconnaître, il est nécessaire de l'isoler, de le séparer de son environnement.

Segmenter une image, c'est effectuer une partition de cette image en régions telles que chacune d'entre elles possède au moins une propriété que ne possèdent pas les autres régions. Le but de la segmentation est de fournir une description de l'image sous forme d'une liste de régions caractérisées par des propriétés qui les différencient.

Mais sa mise en œuvre sur des images réelles est très délicate, ce qui explique le nombre important des méthodes et des variantes. Les différents algorithmes employés pour faire la segmentation des images en régions diffèrent par le type de propriétés recherchées pour les régions et par la manière d'opérer des regroupements de pixels pour former les régions. On distingue trois approches:

I.2.1) la classification des pixels,
I.2.2) la détection de contour,
I.2.3) l'approche par les régions.

I.2.1) Classification des pixels

La classification des pixels se fait sur la base d'une ressemblance entre pixels considérés individuellement. Ces techniques relèvent de méthodes utilisées en reconnaissance de formes. On définit un ensemble fini de classes de pixels et on cherche à affecter chaque pixel de l'image à une de ces classes. On forme les régions en regroupant les pixels connectés par la propriété d'appartenance à une même classe.

Récemment, E.LEVRAT s'est intéressé à ce problème et l'a résolu en utilisant la théorie des ensembles flous[LEVR-89].

I.2.2) La détection de contour

La formation des régions va se faire en déterminant leurs frontières. Les méthodes de détection de contours sont nombreuses et variées. La recherche se décompose en deux phases:

- la détection des pixels susceptibles d'appartenir à un contour, qui elle-même peut être réalisée de deux manières:

* Détection des points de contour sur toute l'image,
* Détection de droites et de courbes (dans ce cas , on recherche directement un ensemble de pixels constituant des droites ou des courbes du contour).

- la formation du contour, c'est-à-dire le choix des séquences de pixels constituant des courbes fermées.

I.2.2.a) Détection des pixels susceptibles d'appartenir à un contour

***Détection des points de contour sur toute l'image*

Un contour est défini par une variation plus ou moins brutale des niveaux de gris des pixels consécutifs. Cette variation peut être détectée par le maximum d'une dérivée première ou par le passage par zéro d'une dérivée seconde. Les opérateurs les plus fréquemment utilisés pour détecter ces variations sont les opérateurs: Gradient, Robert-Cross, Sobel, Prewitt, Kirsh, Laplacien etc....(voir paragraphe "masques").

***Détection de droites et de courbes*

Il existe un grand nombre d'algorithmes permettant de détecter droites et courbes dans une image à niveaux de gris ou binaire. Certains de ces algorithmes utilisent des opérateurs locaux travaillant sur une petite fenêtre, d'autres utilisent des transformations globales telles que la transformée de Hough généralisée, qui transpose l'image en une représentation dans l'espace de ses paramètres.

I.2.2.b) Formation de contour

La formation du contour à partir des points susceptibles d'appartenir au contour a fait aussi l'objet de nombreux travaux. Des algorithmes simples ont été développés, mais dès que l'objet ne se différencie pas suffisamment du fond, il est parfois très difficile de fermer le contour ou de faire correspondre deux segments de droites.

I.2.3) Approche par les régions

Cette approche est très similaire à la précédente. En effet, au lieu de rechercher des points de discontinuité, on recherche la discontinuité de certaines caractéristiques sur des groupes de pixels adjacents. La segmentation est généralement réalisée par étape, le processus est initialisé par la définition d'une partition de l'image en régions de base. Cette partition est réalisée en fonction d'un certain nombre d'hypothèses ou de connaissances a priori, comme dans le cas de la classification des formes. Puis par étapes successives, ces régions sont remodelées. Suivant le type du remodelage, on distingue trois types de méthodes:

- division, fractionnement (split)
- fusion, croissance des régions (merge)
- division, fusion.

I.2.3.a) Méthodes par division

On divise l'image globale en grandes régions, elles-mêmes divisées et ainsi de suite jusqu'à ce que les régions produites vérifient un certain critère, L'exemple le plus connu est la méthode des Quad-Trees [CLIF-86].

I.2.3.b) Méthodes par fusion

On part de petites régions constituées de quelques pixels adjacents et à chaque étape on fusionne entre elles certaines régions adjacentes. Il existe de nombreuses méthodes qui diffèrent, par la manière dont est effectuée la partition initiale et par les critères utilisés pour décider de la fusion des régions.

I.2.3.c) Méthodes par division-fusion

On utilise alternativement les méthodes précédentes, ce qui permet d'affiner davantage le découpage.

I.2.4) Observation particulière

L'approche par les régions est d'un coût élevé en temps de calcul et en espace mémoire. Les tests d'arrêt sont relativement difficiles à trouver.

I.3) Seuillage

Cette opération est utilisée pour obtenir, à partir d'une image à plusieurs niveaux de gris, une image binaire. Le principe est simple: tous les pixels de l'image, dont le niveau de gris est supérieur à un seuil sont mis à la valeur 1. Ceux dont le niveau de gris est inférieur à ce seuil sont mis à 0.

Le seuillage peut être manuel ou automatique :

- soit le seuil est choisi par l'opérateur,
- soit le seuil est déterminé par analyse statistique (moyenne, densité de probabilité, corrélation ....).

Dans une image, le seuillage est utilisé par exemple pour séparer, un objet d'un fond. Il est aussi utilisé, après une accentuation de contraste pour l'obtention du contour.

I.4) Les traitements réalisés à l'aide de masques

Soit une image continue f(x,y) soumise à une transformation par un opérateur linéaire : chaque point de f va donner une tache dans l'image résultante g. Inversement, chaque point de g reçoit par l'intermédiaire de l'opérateur une contribution émanant de tous les points de f. Le résultat pour le point (u,v) de g est donné par le produit de convolution ,

$$g(u,v) = \iint f(x,y) \ . \ h(u-x, v-y) \ dx \ dy$$

l'intégrant représentant la contribution du point f(x,y).
Dans le cas discret :

$$I_c(u,v) = \sum\sum I(x,y) \ . \ h(u-x, v-y)$$

Pour une mise en œuvre efficace, plusieurs hypothèses et commodités simplificatrices sont introduites.

- La tache "h" est supposée de taille limitée; la double sommation est donc limitée à la zone où h est différent de zéro (en pratique, souvent de taille 3x3).

- Dans le produit de convolution et par rapport aux arguments x et y, I et h sont parcourus dans des sens opposés. Il est alors plus commode de considérer un "masque" obtenu à partir de h(.,.) par symétrie horizontale et verticale. Le résultat global est alors la somme des éléments de I(x,y) pondérés par les éléments du masque.

- Les éléments du masque (poids) sont choisis parmi les entiers relatifs simples; le calcul numérique s'en trouve grandement accéléré (on rencontre souvent les poids 0, ± 1, ± 2).

De plus, chaque opérateur impose la valeur des poids. Ainsi la recherche de l'évolution des niveaux selon un axe implique le calcul d'une dérivée première, sous forme de différence : d'où les poids +1 et -1.
Dans un problème de lissage gaussien, le masque est une approche en entier d'une distribution gaussienne.

- En général, la somme des poids est nulle, traduisant le fait que si une propriété est absente, le résultat est nul.

Conclusion

Les masques font partie des traitements les plus simples utilisés dans le domaine de l'image. En effet, leur utilisation fréquente s'explique par :

- leur simplicité de mise en œuvre (logicielle et aussi matérielle),
- leur temps d'exécution relativement réduit,
- l'utilisation de cœfficients simples : entiers relatifs,
- leurs tailles réduites .

Les algorithmes utilisant un masque sous une forme ou une autre sont nombreux et souvent puissants.
Ils permettent la réalisation d'opérations, telles que le filtrage d'une image trop bruitée, l'accentuation du contraste en vue d'une détection de contour, le lissage d'une image trop contrastée. Nous présentons ci-dessous un tableau donnant quelques exemples de masques couramment utilisés, classés d'après leur fonction.

FONCTION	NOMS	TAILLE	NOMBRE	CARACTERISTIQUES
GRADIENT	Sobel	2x2 ou 3x3	2	Dérivée première
	Prewit	2x2 ou 3x3	2	Dérivée première
	Laplacien	3x3 ou 4x4	1	Dérivée seconde opé. isotrope
RECHERHE DE DIRECTION	Boussole	3x3	8	
	3-niveaux	3x3	4	
	5-niveaux	3x3	4	
	Kirsh	3x3	8	
LISSAGE	Gaussien	3x3	1	Moyenne des points voisins

NOMBRE : représente le nombre de masques nécessaires à la réalisation de la fonction.

I.5) Méthodes utilisant la morphologie mathématique

I.5.1) Principe [COST-85]

Les opérateurs morphologiques sont conçus pour des images binaires. Ces opérateurs reposent sur le concept de transformation géométrique d'une image par un élément structurant.

Un élément structurant est un masque de forme quelconque dont les éléments forment un motif. L'application de l'opérateur consiste à balayer l'image avec ce masque et à effectuer pour chaque pixel une mise en correspondance du pixel et de ses voisins avec le motif du masque, puis d'effectuer une opération. Chaque pixel est ainsi l'objet d'une transformation par une fonction plus ou moins complexe.

La morphologie mathématique utilise des opérations ensemblistes pour transformer l'image. On rencontre essentiellement deux types de transformations :

- Les transformations ensemblistes classiques,
- Les transformations en tout ou rien (utilisant un élément structurant).

I.5.2) Transformations ensemblistes classiques

Considérons deux ensembles X et Y les opérations classiques sont :

- l'union $X \cup Y$,
- l'intersection $X \cap Y$,
- la complémentation $(X^c)_Z$.

Le complémentaire est ainsi défini :
un point X appartient au complémentaire X^c s'il n'appartient pas à X.

- la différence symétrique X/Y

$$X/Y = X \cup Y - X \cap Y$$

Exemples d'utilisation de ces transformations en traitement d'image:

Ces transformations n'ont pas d'utilisations directes. En fait ce sont des outils mathématiques nécessaires à l'élaboration d'opérateurs plus complexes, telles que :

- la dilatation
- l'érosion.etc

I.5.3) Transformation en tout ou rien

Définition:

Pour réaliser une telle transformation, il faut tout d'abord choisir un élément B, de géométrie connue, appelé élément structurant, puis le déplacer de façon à ce que son centre passe par toutes les positions de l'espace. Pour chaque position, on pose une question relative à l'union, à l'intersection ou à l'inclusion de B avec ou dans X (X étant l'image à analyser). La réponse sera positive ou négative, d'où le nom de transformation en tout ou rien. Il existe différentes formes d'éléments structurants (cercles, carrés, hexagones, etc...) de différentes tailles (4, 8 voisins). La forme de l'élément structurant et sa taille sont bien sûr liées à la recherche d'une propriété particulière.

1) Transformation par érosion

La transformation par érosion est la première transformation en tout ou rien à avoir été utilisée. C'est d'ailleurs, avec la dilatation, celle qui est la plus importante. Son origine remonte à H.Hadwiger (1957). Ce concept fut repris par G.Matheron (1965-1967), puis développé par J.Serra (1969).

Définition

Soit un espace IR^2 partiellement occupé par un ensemble X, et soit un élément structurant B représentant une forme géométrique simple. Cet élément est repéré par son centre et placé en x dans l'espace IR^2. Il est ensuite déplacé de telle sorte que son centre occupe successivement toutes les positions x de l'espace. Pour chaque position, on pose la question suivante : B est-il entièrement inclus dans X ?. L'ensemble des positions x correspondant à une réponse positive forme un nouvel ensemble Y appelé érodé de X par B, la transformation s'écrit :

$$Y = E^B(X)$$

2) Transformation par dilatation

L'opération de dilatation due à H.Minkowski (1903) se définit d'une manière analogue. Dans ce deuxième cas, la question posée est la suivante : l'intersection de X et B est-elle non vide ?. L'ensemble des positions x correspondant à une réponse positive forme un nouvel ensemble Y, dont la frontière est le lieu des centres géométriques de B lorsque B touche X. Cet ensemble Y est appelé dilaté de X par B, la transformation s'écrit :

$$Y = D^B(X)$$

L'érosion et la dilatation sont deux opérations duales. C'est-à-dire que l'érodé d'un ensemble X et le dilaté de son complémentaire donnent des résultats complémentaires. L'érosion et la dilatation sont des opérations que l'on utilise rarement seules. Elles sont très souvent combinées, pour constituer de nouvelles opérations telles que l'ouverture , la fermeture ou encore le squelette.

3) Ouverture et fermeture

a) Notion d'ouverture

Appliquons à une forme X une érosion, suivie d'une dilatation; la transformation obtenue s'appellera ouverture .

En général après une ouverture, on ne retrouve pas l'ensemble de départ. L'ouvert de X est plus régulier que l'ensemble X initial .

La transformation par ouverture adoucit les contours, coupe les isthmes étroits, supprime les petites îles. L'ouverture joue donc un rôle de filtre passe-bas.

b) Notion de fermeture

Si l'opération précédente est effectuée en sens inverse, c'est-à-dire dilatation suivie d'érosion, on obtient le fermé de X .

Cette transformation bouche les canaux étroits, supprime les petits lacs et les golfes étroits.

4) Squelette

La squelettisation d'une image binaire est une technique qui extrait ce que l'on appelle le squelette de l'image. La notion de squelette a été introduite par Motzkin en 1935. Le squelette représente en fait la quantité d'informations minimale qui décrit complètement l'image. Ce qui veut dire qu'il est théoriquement possible de restituer une image originale à partir de son squelette. La squelettisation s'effectue par une succession d'opérations appelées amincissements jusqu'à obtention d'une structure stable ne pouvant plus être amincie, c'est-à-dire dont les éléments sont des lignes d'une épaisseur d'un pixel.

Définition :

Considérons un ensemble X et son contour C. Un point s appartiendra au squelette, si la distance de s à C est atteinte en au moins deux points distincts de X.

$[s \in Sq(X)] \Leftrightarrow [\exists\, y_1, y_2 \in C \;\; y_1 \neq y_2$ et tel que $d(s,C) = d(s, y_1) = d(s, y_2)]$

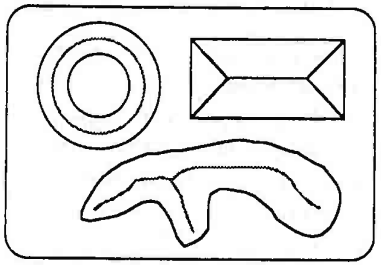


fig 1 : Exemples de squelettes de différents objets

Le squelette Sq(X) peut également être défini comme étant l'ensemble des centres des boules maximales B contenues dans X. Cette définition est bien entendue équivalente à la précédente.

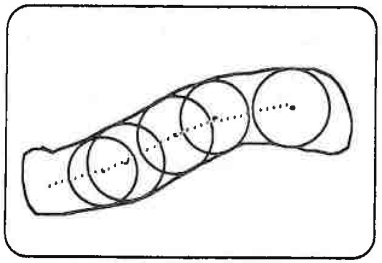


fig 2 : Illustration de la définition du squelette

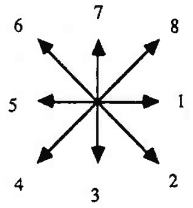
L'élément structurant cercle ne peut être appliqué à une image digitalisée. On utilise alors un élément structurant hexagonal (ou carré).

L'intérêt de cette transformation est la mise en évidence de l'information essentielle à la description de la forme. Cette transformation est réalisée dans le but de diminuer la quantité de données à analyser ou à stocker. C'est en quelque sorte un moyen de codage des images.

I.6) SUIVI DE CONTOUR

I.6.a) Introduction

La méthode que nous utilisons est celle basée sur le codage de Freeman [BOMB-87]. Ce codage est une représentation exacte de chaque pixel du contour et utilise la position relative d'un pixel du contour par rapport au précédent. Cette position sera codée par un chiffre de un à huit, représentant une des huit directions possibles.



Codage de Freeman

Cette procédure fournit une description du contour sous forme d'une liste de vecteurs, ainsi que des coordonnées des points du contour. L'hypothèse de départ est que le contour soit fermé, ce qui est toujours le cas d'une forme pleine.

Le programme commence par la recherche du premier point de contour.

I.6.b) Recherche du premier point de contour

L'image est parcourue pixel après pixel jusqu'à en rencontrer un dont le niveau est 1. Si aucun point n'est rencontré, l'image est considérée comme vide et le programme est arrêté. Si par contre un point est repéré, ses coordonnées sont mémorisées et le contour sera bouclé lorsqu'on sera revenu en cette position. Une fois que le point de départ est repéré, on recherche le point suivant en décrivant successivement, dans le sens horaire, les huit directions données par le code de Freeman.

Si on ne trouve aucun pixel dont le niveau est 1, le point de départ trouvé est un point isolé. Ces derniers points ne sont pas considérés comme des objets et sont codés au niveau 0 comme le fond de l'image. Le programme passe à la recherche d'un autre point de départ.

I.6.c) Recherche des points du contour

Une fois un point repéré, on privilégie toujours la direction située à $\pi/2$ par rapport à la direction précédemment suivie. Pour le premier point, la recherche étant effectuée ligne par ligne, on considèrera comme direction précédente la direction 1.

Ainsi, le premier pixel rencontré dont le niveau est 1 est considéré comme appartenant au contour de l'objet.

Cette méthode permet de décrire des objets et évite d'une façon générale d'aller à l'intérieur de l'objet, particulièrement dans le cas de formes concaves. C'est pourquoi l'on peut appliquer l'algorithme sans traitement initial, comme la détection de contour, qui engendre comme nous l'avons vu, des contours fragmentés (dans le cas d'un seuil trop élevé).

Dans tous les cas, avant de suivre une direction et de prendre en compte le point rencontré,

- on teste si on est dans l'image (on interdit toute direction qui sortirait du cadre).

- on teste si le point suivant a déjà été parcouru. Si oui, on regarde si on est revenu au point de départ. Dans le cas contraire, on valide tout de même la direction suivie. En effet, en aucun cas, l'algorithme ne doit bloquer. Il doit pouvoir en particulier parcourir des segments de droites. Il faut donc lui autoriser le parcours des points déjà traités dans certains cas uniquement. (Ces points parcourus deux fois ne seront pris en compte qu'une seule fois pour le calcul du périmètre et génèreront des surfaces nulles (voir périmètre et surface)).

Le programme est conçu pour des images bruitées. En partant de l'hypothèse que la forme à étudier est de périmètre supérieur à 8, le programme supprime définitivement toute tache dont le périmètre est inférieur à 8.

II) Différents traitements géométriques

Pour réaliser la recherche des paramètres les plus pertinents, nous avons choisi un ensemble de 19 paramètres. Dans ce chapitre, nous présenterons ces différents paramètres.

- la surface (2)
- le périmètre (1)
- le coefficient de compacité (1)
- le rayon moyen (1)
- la différence entre le plus grand et le plus petit rayon (1)
- la variance du rayon moyen (1)
- le nombre de changements d'orientation du contour (1)
- la valeur de quelques moments (3)
- la largeur du rectangle circonscrit (1)
- la longueur du rectangle circonscrit (1)
- la surface du rectangle circonscrit (1)
- le pourcentage de symétrie par rapport à l'axe i (1)
- le pourcentage de symétrie par rapport à l'axe j (1)
- les coordonnées du centre de gravité (2)
- le nombre d'érosions nécessaires à la disparition totale de la forme (1)

Les nombres entre parenthèses représentent le nombre de paramètres calculés lors de notre phase expérimentale. Par exemple, pour le paramètre surface, nous avons utilisé deux méthodes différentes; donc, nous obtenons deux valeurs. Le total nous donne 19.

Certains de ces paramètres sont corrélés, comme par exemple la variance du rayon et le cœfficient de compacité, d'autres pratiquement pas.

Ces paramètres sont des mesures géométriques; l'ajout d'autres types de paramètres ne modifierait en rien la suite de notre recherche et ne présenterait aucun intérêt pour la suite.

II.1) Le périmètre

Le périmètre d'un objet est donné par la somme des vecteurs élémentaires représentant une direction suivie, en prenant l'unité pour les directions codées impaires et $\sqrt{2}$ pour les directions codées paires, d'après le codage de Freeman.

II.2) La surface

Deux méthodes sont proposées.

Première méthode

Elle consiste à seuiller l'image de manière à séparer l'objet du fond et à fournir une image binaire. La surface sera obtenue en comptant le nombre de pixels au niveau 1. Cette méthode simple ne sera valable qu'à la condition que l'image ne soit pas trop bruitée, et qu'il n'existe pas trop de points isolés. Par contre, elle est très pratique pour des objets comportant des trous.

Deuxième méthode

Cette deuxième méthode utilise aussi le codage de Freeman de l'image, obtenu lors du suivi de contour .

En effet, l'aire entourée par un contour est mesurée en sommant les aires comprises entre chaque vecteur élémentaire et la droite des abscisses. Le contour de cette surface est orienté dans le sens horaire. De ce fait, la surface obtenue aura une valeur positive si l'on adopte la convention suivante:

- les vecteurs dont l'abscisse décroît (4,5,6) engendrent une surface positive,
- ceux dont l'abscisse croît (8,1,2) engendrent une surface négative
- et les autres (3,7) engendrent une surface nulle.

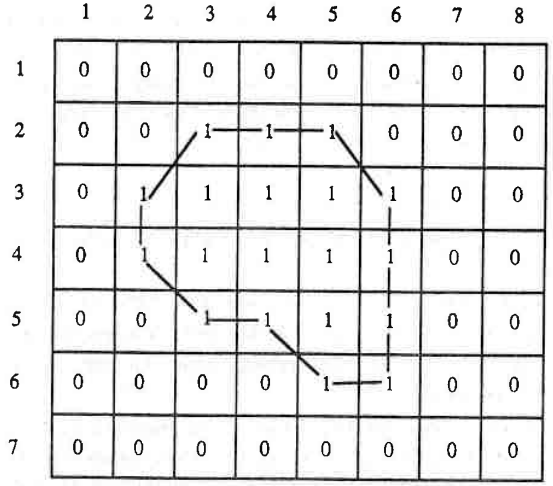
Contribution de chaque vecteur pour le calcul de l'aire:

Vecteur	Aire
1	-y
2	-y-1/2
3	0
4	y+1/2
5	y
6	y-1/2
7	0
8	-y+1/2

Remarque 1: Le programme calcule le double de la surface afin d'éviter la manipulation du facteur1/2.

Remarque 2: Si l'objet contient des trous, il suffira d'isoler la forme, d'inverser le contenu de la forme et de calculer la surface des trous de la même façon. La surface de l'objet sera donc simplement la différence entre la surface calculée à l'aide du contour extérieur et la surface des trous.

Exemple:



Coordonnées	Vecteur	contribution	2*Aire
(2,3)	1	-4	-4
(2,4)	1	-4	-8
(2,5)	2	-5	-13
(3,6)	3	0	-13
(4,6)	3	0	-13
(5,6)	3	0	-13
(6,6)	5	12	-1
(6,5)	6	11	10
(5,4)	5	10	20

(5,3)	6	9	29
(4,2)	7	0	29
(3,2)	8	-5	24
(2,3) Retour point de départ			

donc la surface réelle est 24/2=12

II.3) Centre de gravité

Dans la littérature, on peut rencontrer deux types de centre de gravité:

- le centre de gravité de la forme,
- le centre de gravité du contour.

Dans le cas où les formes sont pleines (sans trous), ces deux définitions donnent le même point. Dans le cas contraire, les deux centres obtenus sont plus ou moins éloignés en fonction de la surface et de la position des "trous". Pour notre étude, nous nous sommes contentés de calculer le centre de gravité du contour. Cependant, nous soulignons le fait que le centre de gravité de la forme est implicitement calculé lors des calculs des différents moments que nous présentons un peu plus loin.

II.4) Le coefficient de compacité

Ce coefficient est défini à partir de la surface (A) et du périmètre (P) de la manière suivante :

$$C = \frac{P^2}{A}$$

L'élément le plus compact est bien sûr le cercle pour lequel :

$$C = \frac{(2\pi R)^2}{A} = 4\pi$$

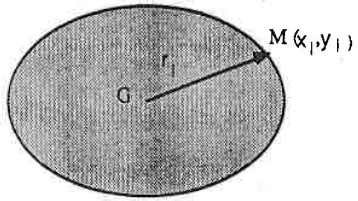
Pour cette raison, nous avons normalisé ce coefficient de la manière suivante :

$$C_n = \frac{C}{4\pi}$$

Ce qui donne la valeur 1 pour le cercle et des nombres supérieurs pour les autres formes.

II.5) Le rayon moyen

Pour chaque point du contour, nous pouvons calculer la distance de ce point au centre de gravité.



$$r_i = [(x_i - x_G)^2 + (y_i - y_G)^2]^{1/2}$$

Nous définirons le rayon moyen comme la moyenne de toutes ces distances.

$$\bar{r} = \frac{1}{N} \sum_{i=1}^{N} r_i$$

où N représente le nombre de pixels constituant le contour.

II.6) Variance du rayon moyen

Dans le cas d'un cercle, les rayons calculés sont tous égaux, donc égaux au rayon moyen. Il n'en est pas de même dans le cas de formes autres que le cercle. Pour caractériser cette "dispersion" , nous pouvons faire appel à plusieurs mesures :

- La différence entre le plus grand rayon et le plus petit rayon.

  $Dif = R_{max} - R_{min}$

- La variance.

$$V = \frac{1}{N} \sum_{i=1}^{N} (r_i - \bar{r})^2$$

- L'écart moyen.

$$E_m = \frac{1}{N} \sum_{i=1}^{N} |r_i - \bar{r}|$$

Ces trois grandeurs représentent la "dispersion absolue" . Pour caractériser une forme, il serait plus astucieux de calculer "la dispersion relative", encore appelée en statistique coefficient de variation.

$$\text{Dispersion relative} = \frac{\text{dispersion absolue}}{\text{moyenne}}$$

II.7) Rectangle circonscrit

Définition :

Le rectangle circonscrit est le plus petit rectangle contenant la forme à étudier.

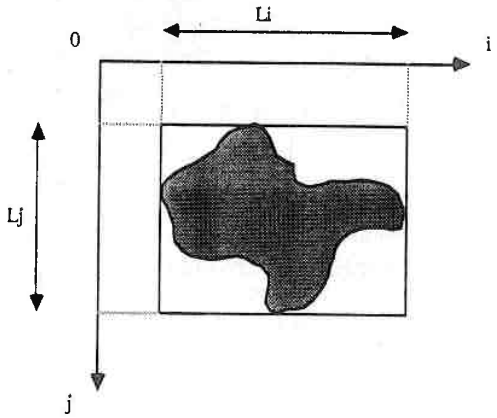


fig 1 : rectangle circonscrit

Li et Lj représentent respectivement la longueur (associée à l'axe i) et la largeur (associée à l'axe j). A partir de ces deux distances, le calcul de la surface du rectangle circonscrit donne :

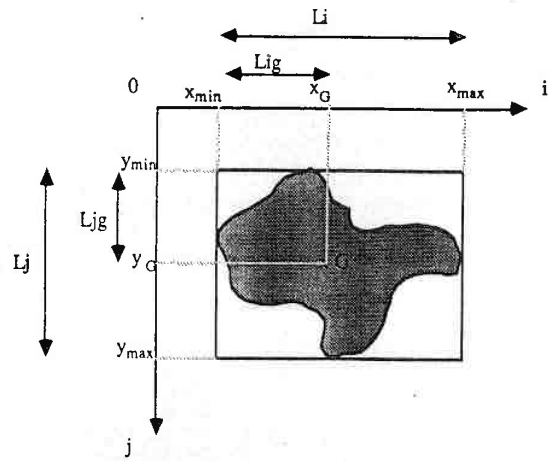
$$S = Li \times Lj$$

Pourcentage de symétrie

Nous avons défini le pourcentage de symétrie par rapport à l'axe i de la manière suivante :

$P_i = 100.[(2.Lig / Li) - 1]$ d'où $-100 \leq P_i \leq 100$

où Lig représente la différence entre x_{min} et x_G.



De la même manière, nous avons le pourcentage de symétrie par rapport à l'axe j :

$P_j = 100.[(2.Ljg / Lj) - 1]$ d'où $-100 \leq P_j \leq 100$

II.8) Nombre de changements d'orientations

Dans le paragraphe consacré au suivi de contour, nous avons vu que le contour était codé sous forme d'une liste de vecteurs. Il est aisé de compter le nombre de fois où la valeur de ce vecteur change.

Exemple : Pour un carré

1 1 1 .....1 1 1 3 3 3 .....3 3 3 5 5 5 .....5 5 5 7 7 7 .....7 7 7

1 2 3 4

donc pour un carré on a 4 changements d'orientations

Pour un cercle le nombre de changements d'orientations est beaucoup plus important.

Remarque: Il est clair que ce paramètre s'avèrera peu discriminant pour des images à contour bruité.

II.9) Calcul des moments

Les moments d'ordre p+q d'une image, calculés à partir de la fonction intensité de l'image continue, sont définis de la sorte:

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

et les moments centrés:

$$\mu_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\, (x-\bar{x})^p\, (y-\bar{y})^q\, dx\, dy$$

Pour une image digitalisée, la double intégrale peut être approximée par une double sommation.

$$m_{pq} = \sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)\, i^p\, j^q$$

$$\mu_{pq} = \sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)\, (i-\bar{i})^p (j-\bar{j})^q$$

Les limites M et N sont les dimensions de la matrice intensité f(i,j), dans laquelle i et j sont les coordonnées du pixel considéré dans l'image. Dans le cas d'image binaire, la fonction intensité ne peut prendre que deux valeurs 0 et 1, ce qui simplifie considérablement les calculs. Nous nous sommes limités aux calculs des moments d'ordre inférieur ou égal à 3 ($p+q \leq 3$)

m_{00} m_{01} m_{02} m_{03} m_{10} m_{11} m_{12} m_{20} m_{21} m_{30}

exemple : $m_{12} = \sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)\, i^1\, j^2$

Pour ces calculs, nous avons utilisé la méthode Delta proposée par Zakaria [ZAKA-86].

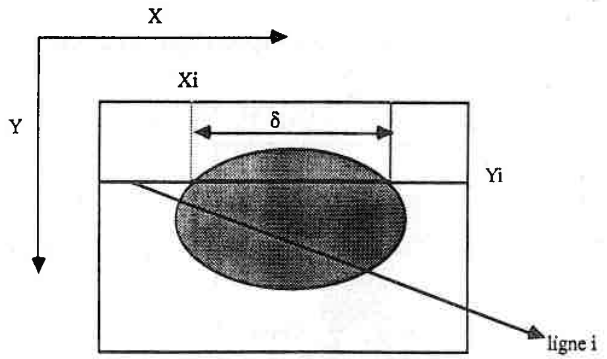
Méthode Delta :

Définition des variables:

- δ : nombre de pixels à 1 connexes dans la ligne i.
- X_i : abscisse du premier pixel à 1 dans la ligne i
- Y_i : ordonnée du premier pixel à 1 dans la ligne i
- m_{pqi} : contribution de la ligne i au calcul du moment m_{pq}

$$m_{pq} = \sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)\, i^p\, j^q = \sum_{i=0}^{M} m_{pqi}$$

$m_{pq} = m_{pq0} + m_{pq1} + m_{pq2} + \ldots\ldots\ldots + m_{pqi} \ldots\ldots\ldots + m_{pqM}$



On en déduit les valeurs de ces contributions (voir démonstration en annexe A dém 1)

$m_{00i} = \delta$

$m_{01i} = \delta\ Y_i$

$m_{02i} = \delta\ Y_i^2$

$m_{03i} = \delta\ Y_i^3$

$m_{10i} = \delta\ X_i + S_1$

$m_{11i} = Y_i (\delta\ X_i + S_1) = Y_i\ m_{10i}$

$m_{12i} = Y_i^2 (\delta\ X_i + S_1) = Y_i^2\ m_{10i}$

$m_{20i} = \delta\ X_i^2 + 2\ S_1\ X_i + S_2$

$m_{21i} = Y_i (\delta\ X_i^2 + 2\ S_1\ X_i + S_2) = Y_i\ m_{20i}$

$m_{30i} = \delta\ X_i + 3\ S_1\ X_i^3 + 3\ S_2\ X_i + S_3$

où

$$S_1 = \sum_{n=0}^{\delta-1} n = (\delta^2 - \delta) / 2$$

$$S_2 = \sum_{n=0}^{\delta-1} n^2 = \delta^3/3 - \delta^2/2 + \delta/6$$

$$S_3 = \sum_{n=0}^{\delta-1} n^3 = \delta^4/4 - \delta^3/2 + \delta^2/4$$

Comme nous le montrons en annexe A (démonstration 2), les moments centrés μ_{pq} ne sont que des combinaisons des moments m_{pq}.
Les moments centrés normalisés η_{pq} peuvent être définis de la manière suivante:

$$\eta_{pq} = \frac{\mu_{pq}}{1 + \mu_{00}^{(p+q)/2}}$$

De ces moments, HU en a déduit 7 moments invariants à la fois à la translatation, à la rotation et aux changements d'échelle.

$\varphi_1 = \eta_{20} + \eta_{02}$

$\varphi_2 = (\eta_{20} + \eta_{02})^2 + 4\ .\ \eta_{11}^2$

$\varphi_3 = (\eta_{30} - 3\ .\ \eta_{12})^2 + (3\ .\eta_{21} - \eta_{03})^2$

$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$\varphi_5 = (\eta_{30} - 3\ .\ \eta_{12})\ .\ (\eta_{30} + \eta_{12})\ .\ [\ (\eta_{30} + \eta_{12})^2 - 3\ .\ (\eta_{21} + \eta_{03})^2\ ]$
$+ (3\ .\eta_{21} - \eta_{03})\ .\ (\eta_{21} + \eta_{03})\ .\ [\ 3.\ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\ ]$

$\varphi_6 = (\eta_{20} - \eta_{02})\ .\ [\ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\ ]$
$+ 4\ .\ \eta_{11}\ .\ (\eta_{30} - 3\ .\ \eta_{12})\ .\ (\eta_{21} + \eta_{03})$

$\varphi_7 = (3\ .\eta_{12} - \eta_{30})\ .\ (\eta_{30} + \eta_{12})\ .\ [\ (\eta_{30} + \eta_{12})^2 - 3\ .\ (\eta_{21} + \eta_{03})^2\ ]$
$+ (3\ .\eta_{21} - \eta_{03})\ .\ (\eta_{21} + \eta_{03})\ .\ [\ 3.\ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\ ]$

CHAPITRE II

DIFFERENTES METHODES DE CLASSIFICATION ET DE SELECTION DE PARAMETRES

I) LES METHODES DE CLASSIFICATION

Les problèmes de classification tiennent un rôle fondamental dans le domaine du traitement d'images. Plusieurs scientifiques ont abordé ce problème et proposé différentes solutions. Dans la littérature, classification et reconnaissance de forme sont souvent confondues. Pour cette raison, nous commencerons par en donner une définition.

La classification est l'étape durant laquelle, on va s'attacher à retrouver parmi les objets les différentes classes (cluster) existantes, d'après un critère bien précis. En effet pour un ensemble d'objets donné, plusieurs classifications peuvent être possibles. On pourra par exemple vouloir classer ces objets d'après leur forme géométrique ou encore d'après leur couleur. Pour cette raison, le critère de classification doit être judicieusement choisi. Le critère étant toujours calculé en fonction des différentes mesures (paramètres) effectuées sur l'image, il en résulte que la classification obtenue dépendra essentiellement du choix des paramètres.

Pour définir chaque classe, on introduit alors la notion de représentant. Ce dernier peut être tout simplement le centre de gravité de toutes les formes de la classe.

Les classes étant bien définies, la reconnaissance de forme constitue l'étape ultime durant laquelle chaque forme va être affectée définitivement à l'une de ces classes.

Nous présentons dans ce qui suit, deux familles de méthodes de classification:

I.1) les méthodes de réallocation,

I.2) les méthodes hiérarchiques.

On rencontre de nombreux algorithmes basés sur ce principe. Ils ne diffèrent que par le choix de la partition initiale, par la méthode ou le critère utilisé pour déplacer une forme et par la définition de l'expression " la meilleure partition ".

I.1.2) Les variantes

Parmi les méthodes les plus rencontrées, on trouve les méthodes des centroïdes . Elles ne diffèrent entre elles que par le choix de la partition initiale ou des représentants initiaux.

Choix des différents représentants initiaux.

Il est évident que le choix des représentants initiaux des différentes classes joue un rôle important pour la minimisation du nombre d'itérations. Plusieurs choix sont envisageables:

1) Choix des NR premiers objets

On choisit les NR premiers objets de l'échantillon. Cette méthode est la plus simple mais aussi la plus discutable car rien n'indique que les NR premiers objets soient représentatifs de l'échantillon. De plus, la convergence vers la partition finale risque d'être très longue.

2) Tirage aléatoire

On choisit au hasard les NR objets. Ainsi, chaque groupe sera représenté proportionnellement à sa taille. Mais cette méthode a l'inconvénient de faire disparaître les petits groupes.

3) Choix effectué par l'opérateur

Si l'opérateur dispose d'une connaissance a priori des formes à classer, il peut fournir lui-même les NR centres initiaux.

4) Calcul automatique des NR centres

- La méthode du quantificateur uniforme :

Cette méthode consiste à découper le cube de dimension NP contenant tous ou la plupart des points de la série d'entraînement, en NR parties.

- Utilisation d'un algorithme hiérarchique

LANCE et WILLIAMS (1967) suggèrent d'utiliser les algorithmes hiérarchiques et de calculer le centre de gravité des groupes résultants pour constituer les centres des groupes initiaux. Ces méthodes ont l'inconvénient d'être coûteuses en place mémoire et en temps, mais elles fournissent une partition initiale qui va très vite converger vers la partition finale. L'exemple le plus connu est la méthode de splitting.

Cette technique consiste à diviser récursivement l'ensemble d'apprentissage afin d'obtenir la partition initiale (cette méthode est détaillée dans le chapitre III).

Une variante de ces méthodes réside dans la représentation des différentes classes. Dans la catégorie des méthodes des centroïdes, chaque classe est représentée par un point. Dans le cas de la méthode de Diday, les classes sont représentées non pas par un point mais par un ensemble de points. Un point important à souligner est le fait que dans le cas des méthodes des centroïdes, le représentant (centre de gravité) est en fait un point fictif (n'appartenant pas à la série d'apprentissage), alors que dans la méthode de Diday, les représentants appartiennent à la série d'apprentissage.

Nous pouvons cependant, malgré ces quelques variantes, donner un organigramme général des méthodes de réallocation (fig.1).

Nous présentons dans ce qui suit quelques exemples basés sur ces méthodes. Nous commencerons par la méthode des nuées dynamiques connues encore sous le nom de méthode de Diday. Puis nous présenterons la méthode Isodata, ainsi que celle du maximin.

I.1.3) Méthode des nuées dynamiques [DIDA-70]

Introduction

Cette méthode a été proposée en 70 par E .Diday. Comme pour les autres méthodes de réallocation, la classification est obtenue par un processus itératif, au cours duquel les formes vont être déplacées d'une classe à une autre. Ainsi la partition va s'affiner au cours des différentes étapes. Cependant cette méthode se distingue des autres par le choix des représentants des différentes classes. En général, les classes sont représentées par un seul point (centre de gravité), la méthode des nuées dynamiques caractérise chaque classe par un ensemble de points formes dont l'utilisateur aura fixé le nombre d'éléments.

Principe:

L'utilisateur choisit aléatoirement un certain nombre n_i de représentants pour chaque classe C_i parmi les formes existantes. Ces représentants seront regroupés dans un sous-ensemble E_i. La partition sera obtenue en affectant la forme x_k à la classe C_i, si la forme est plus proche de l'ensemble des représentants (E_i) de la classe C_i que des autres ensembles de représentants des autres classes; c'est à dire si

$$D(x_k,E_i) < D(x_k,E_j) \text{ pour } j \neq i$$

D est une distance entre un point et un ensemble de points.

On obtient alors la première partition. A partir de cette dernière et d'un critère R (fonction d'Agrégation-Ecartement), qu'il faudra minimiser, nous allons déterminer les nouveaux sous-ensembles E_i, de manière à avoir comme éléments des E_i les formes les plus représentatives de chaque classe. Diday appelle les éléments de E_i , "étalons" de la classe C_i.

Diday propose comme critère :

Soit $L = (E_1, E_2, E_3 \ldots\ldots E_{NR})$

où E_i est le sous-ensemble des représentants (étalons) de la classe C_i

$$R(x,i,L) = \frac{D(x,E_i)\ D(x,C_i)}{\left[\sum_{j=1}^{NR} D(x,E_j)\right]^2}$$

- les éléments de E_i sont les représentants de la i-ième classe
- $D(x,E_i)$ aura pour effet d'agréger les points représentants de la classe
- $D(x,C_i)$ aura pour effet de ramener les représentants vers le centre de leur classe.

- $\sum_{j=1}^{NR} D(x,E_j)$ aura pour effet d'écarter les E_j entre eux

d'où le nom de fonction d'Agrégation-Ecartement.

A partir de ces nouveaux étalons, nous pouvons recalculer la nouvelle partition C. Le processus est alors réitéré, jusqu'à l'obtention d'une partition stationnaire.

Nous présentons dans ce qui suit l'algorithme, puis nous donnerons un exemple simple, qui nous permettra une compréhension plus rapide.

Notation: X : $\{x_1,x_2...x_k.....\}$ensemble des objets à classer (série d'apprentissage)

P : ensemble des parties de X

D : fonction distance définie sur XxP et prenant ses valeurs dans R^+

P_i: ensemble des parties de X ayant n_i éléments

$E_1, E_2, E_3..... E_{NR}$: NR parties de E avec E_i éléments de P_i (les éléments de E_i sont les représentants de la classe C_i)

n_i: nombre d'éléments de la partie E_i (spécifié par l'utilisateur)

$C = \{ C_1 , C_2 .....C_{NR}\}$ ensembles des différentes classes (partition)

$L_{NR} = (P_1 \times P_2 \times P_3 \times .....\times P_{NR})$

T : ensemble des NR premiers entiers

R : application de $ExTxL_{NR}$ vers R^+ fonction d'agrégation-écartement

Algorithme

Soit $L^{(n)} = (E_1^{(n)}, E_2^{(n)}, E_3^{(n)}..... E_{NR}^{(n)})$

où $E_i^{(n)}$ est élément de P_i ainsi $L^{(n)}$ est élément de L_{NR} (l'exposant indique le nombre d'étapes)

Etape 1: Connaissant $L^{(n)}$, le calcul de $L^{(n+1)}$ se fait comme suit: On calcule la distance de chaque forme aux différents sous-ensembles E_i.

$D(x, E_i^{(n)})$ pour tout x de X

pour tout i de T (i=1.....NR)

Etape 2: On réalise une partition de X en NR classes $C_i^{(n)}$ pour tout i=1,2.....NR, comme suit :

$C_i^{(n)} = \{ x / D(x,E_i^{(n)}) < D(x,E_j^{(n)}) \ \forall j \in \{1,2...NR\}$ et $j \neq i \}$

Nous supposerons que $D(x, E_i^{(n)}) \neq D(x, E_j^{(n)})$ pour $i \neq j$

Etape 3: On définit $L^{(n+1)}$ à partir du critère d'agrégation-écartement $R(x,i,L^{(n)})$.

$L^{(n+1)} = (E_1^{(n+1)}, E_2^{(n+1)}, E_3^{(n+1)}..... E_{NR}^{(n+1)})$

$E_i^{(n+1)}$ = les n_i éléments de E qui minimisent la fonction critère $R(x,i,L^{(n)})$ pour i=1,2.....NR

Etape 4: Si la partition reste inchangée, le programme est terminé sinon, aller à l'étape 1.

Diday propose également un deuxième critère beaucoup plus simple que le premier.

$$R(x,i,L) = D(x,C_i)$$

Exemple simple:

Considérons la série d'apprentissage suivante:

$X = \{ x_1 ; x_2 ; x_3 ; x_4 ; x_5 ; x_6 ; x_7 ; x_8 \}$

Pour la clarté de l'exemple, chaque forme sera représentée par un seul paramètre:

$x_1 = 0$ $x_2 = 4$ $x_3 = 5$ $x_4 = 22$ $x_5 = 25$ $x_6 = 50$ $x_7 = 55$ $x_8 = 56$

Le nombre de classes recherchées est NR =3

Le critère utilisé est $R(x,i,L) = D(x,C_i)$

Nous prendrons n_i =2 pour i=1,2,3

Les différentes parties E_i auront donc deux éléments de l'ensemble initial X.

Ces éléments sont choisis aléatoirement. Nous prendrons donc:

$E_1 = \{x_1 ; x_5 \} = \{ 0 ; 25 \}$

$E_2 = \{x_3 ; x_4 \} = \{ 5 ; 22 \}$

$E_3 = \{x_4 ; x_8 \} = \{ 22 ; 56 \}$

donc $L^{(1)}$ $= \{ E_1 ; E_2 ; E_3 \}$
$= \{ \{x_1 ; x_5 \} ; \{x_3 ; x_4 \} ; \{x_4 ; x_8 \} \}$
$= \{ \{ 0 ; 25 \} ; \{ 5 ; 22 \} ; \{ 22 ; 56 \} \}$

Etape 1

Calculons les distances entre chaque forme x_i et les sous-ensembles E_k.

Par exemple:

$D(x_1 ; E_1) = (|x_1-x_1| + |x_1-x_5|) / 2 = (|0-0| + |0 - 25|) / 2 = 25/2 = 12,5$

La distance entre un point et un ensemble de points correspond ici à la distance moyenne des distances entre la forme considérée et les formes de l'ensemble.

Les résultats sont donnés dans le tableau suivant :

Ensemble des formes	x1	x2	x3	x4	x5	x6	x7	x8
Valeur des formes	0	4	5	22	25	50	55	56
E1= (x1;x5)=(0;25)	12,5	12,5	12,5	12,5	12,5	37,5	42,5	43,5
E2= (x3;x4)=(5;22)	13,5	9,5	8,5	8,5	11,5	36,5	41,5	42,5
E3= (x4;x8) = (22;56)	39	35	34	17	17	17	17	17

Etape 2 On détermine la première partition de la façon suivante: Une forme x_i va être affectée à la classe C_k si $D(x_i ; E_k) < D(x_i ; E_p)$ pour p=1...3 et p≠k

On obtient donc la partition suivante :

$C_1 = \{x_1 \} = \{ 0 \}$
$C_2 = \{ x_2 ; x_3 ; x_4 ; x_5 \} = \{ 4 ; 5 ; 22 ; 25 \}$
$C_3 = \{x_6 ; x_7 ; x_8 \} = \{ 50 ; 55 ; 56 \}$

Etape 3 Connaissant $L^{(1)}$, on peut calculer $L^{(2)} = \{ E_1^{(2)} ; E_2^{(2)} ; E_3^{(2)} \}$

$E_i^{(2)}$ = les 2 éléments de X qui minimisent le critère :

$R (x , i , L^{(1)}) = D (x , C_i)$

Pour i=1, E_1 sera donc constitué des deux éléments qui minimiseront la valeur $D(x,C_1)$.

Pour cela, nous allons calculer la distance entre chaque forme x_i et la classe C_1.

Les résultats sont donnés dans le tableau suivant:

Numéro / Valeurs des Formes	C1=(x1) (0)	C2=(x2;x3;x4;x5) (4,5,22,25)	C3=(x6;x7;x8) (50;55;56)
x1	0	14	53,67
x2	4	10	49,67
x3	5	9,5	48,67
x4	22	9,5	31,67
x5	25	11	28,67
x6	50	36	3,67
x7	55	41	2,00
x8	56	42	2,33

On peut donc remarquer que les deux formes les plus proches de la classe C_1 sont les formes x_1 et x_2 donc $E_1 = \{ x_1 ; x_2 \}$. On obtient de la même manière E_2 et E_3.

$E_1^{(2)} = \{x_1 ; x_2 \} = \{ 0 ; 4 \}$
$E_2^{(2)} = \{x_3 ; x_4 \} = \{ 5 ; 22 \}$
$E_3^{(2)} = \{x_7 ; x_8 \} = \{ 55 ; 56 \}$
donc $L^{(2)} = \{ E_1^{(2)} ; E_2^{(2)} ; E_3^{(2)} \}$
$= \{ \{x_1 ; x_2 \} \} ; \{x_3 ; x_4 \} ; \{x_7 ; x_8 \} \}$
$= \{ \{ 0 ; 4 \} ; \{ 5 ; 22 \} ; \{ 55 ; 56 \} \}$

On remonte à l'étape 1.

Etape 1 On calcule la distance entre chaque forme et les nouveaux sous-ensembles E_i. Les résultats sont regroupés dans le tableau suivant:

Ensemble des formes	x1	x2	x3	x4	x5	x6	x7	x8
Valeur des formes	0	4	5	22	25	50	55	56
E1= (x1;x2)=(0;4)	2	2	3	20	23	48	53	54
E2= (x3;x4)=(5;22)	13,5	9,5	8,5	8,5	11,5	36,5	41,5	42,5
E3= (x7;x8) = (55;56)	55,5	51,5	50,5	33,5	30,5	5,5	0,5	0,5

Etape 2 On en déduit la partition:

$C_1 = \{x_1 ; x_2 ; x_3 \} = \{ 0 ; 4 ; 5 \}$
$C_2 = \{ x_4 ; x_5 \} = \{ 22 ; 25 \}$
$C_3 = \{x_6 ; x_7 ; x_8 \} = \{ 50 ; 55 ; 56 \}$

Etape 3 On calcule les distances $D (x_i ; E_j)$ pour i=1,2.....8 et pour j=1,2,3

Numéro / Valeurs des Formes		C1=(x1;x2;x3) (0;4;5)	C2=(x4;x5) (22;25)	C3=(x6;x7;x8) (50;55;56)
x1	0	3,00	23,5	53,67
x2	4	1,67	19,5	49,67
x3	5	2,00	18,5	48,67
x4	22	19,00	1,5	31,67
x5	25	22,00	1,5	28,67
x6	50	47,00	26,5	3,67
x7	55	52,00	31,5	2,00
x8	56	53,00	32,5	2,33

On en déduit $E_1^{(3)} = \{x_2 ; x_3\} = \{4 ; 5\}$

$E_2^{(3)} = \{x_4 ; x_5\} = \{22 ; 25\}$

$E_3^{(3)} = \{x_7 ; x_8\} = \{55 ; 56\}$

d'où $L^{(3)} = \{ \{x_2 ; x_3\} ; \{x_4 ; x_5\} ; \{x_7 ; x_8\} \}$

$= \{ \{4 ; 5\} ; \{22 ; 25\} ; \{55 ; 56\} \}$

On remonte à l'étape 1.

Etape 1 On calcule les distances entre les formes x_i et les sous-ensembles E_j.

Ensemble des formes	x1	x2	x3	x4	x5	x6	x7	x8
Valeur des formes	0	4	5	22	25	50	55	56
E1= (x2;x3)=(4;5)	4,5	0,5	0,5	17,5	20,5	45,5	50,5	51,5
E2= (x4;x5)=(22;25)	23,5	19,5	18,5	1,5	1,5	26,5	31,5	32,5
E3= (x7;x8)= (55;56)	55,5	51,5	50,5	33,5	30,5	5,5	0,5	0,5

Etape 2 On en déduit la partition:

$C_1 = \{x_1 ; x_2 ; x_3\} = \{0 ; 4 ; 5\}$

$C_2 = \{x_4 ; x_5\} = \{22 ; 25\}$

$C_3 = \{x_6 ; x_7 ; x_8\} = \{50 ; 55 ; 56\}$

On s'aperçoit qu'à la troisième itération la partition reste inchangée, donc le programme s'arrête.

Conclusion : Ces méthodes donnent de bons résultats, à la condition que le choix initial des représentants des différentes classes ne soit pas trop "mauvais". En effet, le programme ne converge pas vers la solution souhaitée, si parmi les représentants initiaux n'apparaît pas au moins un représentant de chaque classe. On obtient alors, au bout de quelques étapes, une classe et un

sous-ensemble d'étalons vides. Cependant ce problème peut être contourné si l'on prend un nombre d'étalons important; mais ceci augmente évidemment le temps de calcul. Il faut alors trouver un bon compromis entre ces deux facteurs.

I.1.4) Algorithme Isodata

Principe : A partir d'une partition initiale choisie par l'utilisateur, on calcule les centres de gravité de chaque classe. Pour chaque classe, on calcule l'écart-type intragroupe. Ces écarts sont comparés à un premier seuil σ_1 choisi aussi par l'opérateur:

- Si l'écart-type intragroupe pour une classe donnée est supérieure à ce seuil, alors on perturbe légèrement le centre de gravité, de manière à obtenir deux nouveaux centres très proches. Les formes de la classe sont alors réparties en deux nouvelles classes, dont on recalculera les centres de gravité.
- Si l'écart-type intragroupe est inférieur à ce seuil, on décide de ne pas toucher à cette classe.

A partir de cette nouvelle partition, on calcule la distance entre tous les centres de gravité.

On regroupe les classes associées aux centres de gravité, dont la distance est inférieure à un second seuil σ_2 choisi par l'opérateur. La procédure est réitérée jusqu'à l'obtention d'une partition stable.

Algorithme:

- On définit une partition initiale fondée sur un choix arbitraire (mais judicieux si possible) de centres initiaux pour chaque classe.
- On définit les régions en associant à chaque représentant les formes les plus proches. La proximité peut être évaluée au sens géométrique d'Euclide.
- On scinde en deux points chaque mesure initiale si l'écart-type intragroupe dépasse un seuil σ_1 fixé par l'opérateur.
- On définit une nouvelle partition en utilisant ces nouveaux points (moyennes des points considérés) pour effectuer les groupements et déterminer le nouveau point "moyen" de chaque région.
- On calcule la distance entre tous les points moyens pris deux à deux.
- On regroupe les régions associées aux points moyens dont la distance est inférieure à un second seuil σ_2.

- On recommence la procédure jusqu'à l'obtention d'une partition stable.

Conclusion : Le nombre de classes pour cette méthode n'est pas à définir au départ. Cependant le choix des différents seuils (seuils qu'il est relativement difficile de choisir) contribue fortement à délimiter ce nombre. Si le seuil σ_1 est choisi trop grand, le nombre de classes sera relativement réduit. Par contre si ce seuil est choisi trop petit, le nombre de classes peut devenir très grand, voire très proche du nombre de formes. De même, si l'on choisit le seuil σ_2 trop grand, des classes relativement proches seront systématiquement regroupées.

I.1.5) Algorithme du Maximin

Cette méthode est différente des deux précédentes. En effet, l'opérateur n'a à fournir ni partition initiale, ni nombre de classes.

Principe:

On considère le premier point de la série comme le représentant N_1 de la première classe. Puis on recherche le deuxième représentant N_2 de la classe suivante. Il correspond au point forme de la série, le plus éloigné (au sens de la distance euclidienne par exemple) du premier représentant. Le troisième représentant sera le point forme pour lequel la distance par rapport aux deux autres représentants sera la plus grande. La distance d'un point par rapport à un ensemble de points est obtenue ici de la manière suivante :

- on calcule les distances du point considéré aux différents points de cet ensemble,
- la distance retenue alors est la plus petite de ces distances.

Le processus est alors réitéré jusqu'à ce que l'on obtienne une distance (du nouveau représentant à l'ensemble des représentants), inférieure à la moitié de la distance moyenne des représentants précédents.

Algorithme :

- Choix du premier représentant N_1 : On prend le premier élément de l'ensemble des données.
- On recherche l'élément le plus éloigné de N_1, au sens des distances. Pour cela, on calcule :

$$d_{1i}(N_1, X_i) \quad \text{pour tout i} \quad \text{et} \quad i \neq 1$$

- On recherche la distance maximale d_{1u}, ce qui permet de trouver le deuxième représentant N_2.
- On calcule ensuite les distances entre les représentants présents et les points forme restants, soit

$$d_{ki}(N_k, X_i) \qquad k = 1 \text{ ou } 2$$
$$i = 2 \ldots\ldots NI$$

et l'on conserve la distance δ_{ki} minimale

$$\delta_{ki} = \min_i (d_{ki}) \qquad k = 1 \text{ ou } 2$$

- On recherche la distance maximale $\delta_{ki} = D_{kp}$. Si cette distance est supérieure à la moitié de la distance séparant N_1 et N_2, on crée un représentant supplémentaire, soit

$$N3 = Xp \quad \text{tel que} \quad D_{kp} = \max_i (\delta_{ki}) \qquad k = 1 \text{ et } 2$$
$$\text{avec } D_{kp} \geq \frac{1}{2} D_{12}$$

- On recommence la procédure en considérant ce nouveau représentant. L'étape précédente est toutefois modifiée par la comparaison de D_{kp} avec la moitié de la valeur moyenne des distances entre représentants. Si cette valeur maximale est inférieure à ce seuil, la procédure est terminée. On dispose alors du nombre de classes données par k et de leurs représentants N_1 N_2 ..... N_k.

Conclusion

Cette méthode donne de bons résultats à la condition que les classes que l'on souhaite trouver soient uniformément réparties dans l'espace des formes considéré.

I.2) METHODES HIERARCHISEES OU METHODES D'ANALYSE DE GRAPPES

I.2.1) Introduction

Les méthodes hiérarchisées sont essentiellement basées sur la recherche de similarité ou de dissimilarité entre objets ou groupes d'objets. Grâce à une mesure de similarité ou de dissimilarité, les formes ou groupes de formes vont être regroupés ou divisés au cours des différentes étapes.

Les méthodes hiérarchisées ont été habituellement utilisées en statistique pour analyser les matrices de grandes dimensions. Ces principes ont alors été appliqués aux matrices de similarité des formes.

Les différentes méthodes hiérarchisées

Les méthodes hiérarchisées sont de deux types:

-Les méthodes basées sur des algorithmes ascendants ou agglomératifs, qui procèdent par fusions successives d'objets ou de groupes d'objets deux à deux.

- Les méthodes basées sur des algorithmes descendants ou divisifs, qui procèdent par dichotomies successives de l'ensemble des objets.

Les premières sont beaucoup plus utilisées , en raison de leur simplicité de mise en œuvre.

Matrice de similarité

Cette matrice est obtenue de façon simple. Soit E un ensemble constitué de N objets $X_1, X_2 \ldots\ldots X_N$, les éléments d_{ij} de la matrice correspondent aux mesures de distance (ou de similarité ou d'association) entre deux formes X_i et X_j.

$$\begin{bmatrix} d_{11} & d_{12} & \ldots\ldots d_{1N} \\ d_{21} & d_{22} & \ldots\ldots d_{2N} \\ . & . & . \\ . & . & . \\ d_{N1} & d_{N2} & \ldots\ldots d_{NN} \end{bmatrix}$$

Matrice de similarité

où d_{ij} représente la distance (la similarité) entre les formes (ou groupes) X_i et X_j.

Comme $d_{ij} = d_{ji}$ et que $d_{ii} = 0$, cette matrice est symétrique et sa diagonale ne comporte que des 0.

I.2.2) Méthodes hiérarchisées ascendantes

Principe:

Supposons que l'on dispose d'un ensemble E de N objets que l'on souhaite grouper. L'idée est de fusionner récursivement les objets les plus similaires. Ce fusionnement s'accompagne d'une transformation de la matrice de similarité des formes en matrice de similarité des groupes de formes, matrice dont la taille va décroître au cours des étapes.

Pour réaliser cette fusion, il faut préalablement définir une règle de calcul entre groupements disjoints de formes. Cette distance est en général une fonction simple des distances entre les différentes formes impliquées dans cette fusion.

Les différentes méthodes ne diffèrent que par le choix du critère d'agrégation.

a) Méthode du simple lien ou du plus proche voisin

Algorithme

- Rechercher dans la matrice de similarité les deux formes ou groupes de formes, A et B les plus similaires correspondants donc à la distance d_{ij} minimale: d_{AB} .
- Fusionner A et B en un groupe C .
- Calculer la distance du groupe C à tous les autres formes ou groupes de formes R d'après le critère d'agrégation du saut minimal, c'est-à-dire pour chaque groupe R, prendre comme nouvelle valeur de distance la plus petite des deux distances d_{AR} et d_{BR}.

$d_{CR} = \text{Min}\ (d_{AR}, d_{BR})$ pour tout groupe R.

- Modifier la matrice de similarité.
- Fin du programme lorsqu'on a atteint le nombre de groupes souhaités.

Remarque :

L'inconvénient de cette méthode est la possibilité de regroupement en chaînes, qui conduit à regrouper des formes très dissemblables.

b) Méthodes du lien complet ou du voisin le plus éloigné

La méthode est identique à la précédente, mais on utilisera le critère d'agrégation du saut maximal. La distance du groupe C à tous les autres formes ou groupes de formes R sera obtenue en prenant comme valeur de distance la plus grande des deux distances d_{AR} et d_{BR}.

$d_{CR} = \text{Max}\ (d_{AR}, d_{BR})$ pour tout groupe R

c) Méthodes du lien moyen à l'intérieur du nouveau groupe

La différence essentielle de cette méthode par rapport aux deux méthodes précédentes est que la distance entre deux groupes ne tient pas compte d'un seul point (le plus proche, le plus loin), mais de tous les points du groupe.

II) LES DIFFERENTES METHODES DE SELECTION DE PARAMETRES

II.1) INTRODUCTION

Dans la plupart des problèmes de sélection de paramètres, on rencontre deux sous-problèmes fondamentaux.

- Le choix du critère de sélection du sous-ensemble de paramètres,
- La procédure de recherche du sous-ensemble de paramètres.

Le premier sous-problème traduit le besoin d'un critère qui va évaluer "l'efficacité", la "qualité" des sous-ensembles de paramètres. Les critères les plus utilisés sont:

* la probabilité de mauvaise classification,
* la distance de Bhattacharyya,
* la divergence,
* la fonction d'entropie,
* la mesure de séparabilité.

Le deuxième sous-problème est le choix de la procédure à utiliser pour la recherche du meilleur sous-ensemble de m paramètres; "meilleur" étant défini ici par le critère choisi à l'étape précédente. De ce choix va dépendre la qualité du résultat.

Nous présentons dans ce chapitre quelques méthodes classiques de sélection de paramètres. Nous commencerons par les méthodes séquentielles, largement développées par K.S FU [FU-67]. Pour ce groupe, la sélection des paramètres est réalisée parallèlement à la classification et il est difficile de séparer ces deux notions.

Puis nous aborderons les méthodes dynamiques. Nous présenterons trois de ces méthodes :

- La méthode du "Branch and Bound" proposée par Narendra et Fukunaga [NARE-77],
- La méthode proposée par Cheung [CHEU-78],
- La méthode proposée par Foroutan et Sklansky [FORO-86].

La dernière catégorie concerne les méthodes d'analyse en composantes principales. Deux exemples sont donnés :

- La méthode de Karhunen -Lœve [FUKU-70], [MARW-83],
- La méthode de Fisher [MALI-87].

II.2) LES METHODES SEQUENTIELLES

Les méthodes séquentielles sont définies comme étant des procédures utilisant un à un les paramètres qui servent à caractériser les formes.

Principe

Considérons la forme X à classer, représentée par le vecteur forme $X=\{x_1,x_2,.....x_{NP}\}$. La première étape consiste à choisir le paramètre qui discrimine le mieux les NR classes possibles. Pour cela, il faut chercher à évaluer la "qualité" de chaque paramètre. La divergence a été proposée comme critère de "qualité" pour les paramètres. Lewis propose aussi une fonction sous forme d'entropie [LEWI;62]. Supposons que chaque paramètre P_j pour j = 1 à NP (NP étant le nombre de paramètres) peut prendre NI valeurs (NI observations), $P_j(k)$ représentera la valeur du j-ième paramètre pour la k-ième forme (observation). On introduit alors un nombre G_j, associé au paramètre P_j, mesurant sa " qualité ".
On sélectionne alors le paramètre P_j correspondant au plus grand G_j. La reconnaissance est alors effectuée avec ce paramètre uniquement. Si la probabilité d'erreur est trop grande, on sélectionne un deuxième paramètre (correspondant au plus grand G_j des paramètres restants) et on recalcule la probabilité d'erreur liée à ces deux paramètres. Le nombre de paramètres augmente jusqu'à obtention d'une probabilité d'erreur convenable (acceptable). La sélection et la reconnaissance sont donc réalisées simultanément.
Les méthodes séquentielles sont donc constituées de 2 grandes étapes :

1) Le choix séquentiel des paramètres.

2) La détermination d'un critère d'arrêt de la procédure. En effet, la procédure doit s'arrêter lorsque la décision de classification est optimale au sens de la probabilité d'erreur et qu'elle ne peut être améliorée par la mesure de paramètres supplémentaires.

1) Choix séquentiel des paramètres

Comme nous l'avons fait remarquer un peu plus haut, à chaque paramètre P_j est donc associé un nombre G_j mesurant la "qualité" du paramètre. Ce nombre G_j est en général une statistique obtenue par évaluation de P_j parmi un grand échantillon de formes à reconnaître.

K.S. Fu suggère les trois hypothèses suivantes liant le nombre G_j au pourcentage de bonne reconnaissance, afin de guider le choix de G_j.

1) si $G_j > G_q$, alors le pourcentage de bonne reconnaissance en utilisant uniquement P_j doit être supérieur au pourcentage de bonne reconnaissance en utilisant uniquement P_q.

2) si $G_j > G_q$, alors pour n'importe quel ensemble de paramètres F, le pourcentage de bonne reconnaissance en utilisant P_j et F doit être plus grand que le pourcentage de bonne reconnaissance en utilisant P_q et F.

3) le pourcentage de bonne reconnaissance en utilisant F est une fonction linéaire de la somme des valeurs de G_j pour les paramètres de F.
Il est difficile de trouver un nombre G_j qui vérifie les deux dernières conditions à la fois. K.S.Fu propose alors un nombre G_j qui satisfait ces deux dernières conditions dans un grand nombre de cas.

$$G_j = \sum_{i=1}^{NR} \sum_{k=1}^{NI} p(C_i , P_j(k)) \; . \ln \varphi\{ p(C_i , P_j(k)) \}$$

(sous la condition que les mesures de paramètres soient statistiquement indépendants). La fonction logarithme est utilisée en raison de la propriété d'additivité de G_j exigée dans la troisième condition. Et d'après la condition 1, φ doit être une mesure de corrélation entre P_j et C_i.

$$\varphi \{ p [C_i , P_j(k)]\} = \frac{p [C_i, P_j(k)]}{p(C_i)\, p[P_j(k)]} = \frac{p [C_i/P_j(k)]}{p(C_i)}$$

d'où

$$G_j = \sum_{i=1}^{NR} \sum_{k=1}^{NI} p(C_i, P_j(k)) \ln \left[\frac{p[C_i / P_j(k)}{p(C_i)} \right]$$

$$= \sum_{i=1}^{NR} \sum_{k=1}^{NI} p(C_i, P_j(k)) \left[\frac{p[C_i / P_j(k)}{p(C_i)} - 1 \right]$$

G_j peut donc être interprété comme l'information mutuelle entre le paramètre P_j et les classes $C_1 .... C_{NR}$.
Supposons que pour une classe C_i, X soit distribué selon une densité gaussienne multivariable avec un vecteur espérance μ_i et une matrice de covariance Σ :

$$p(X/C_i) = \{ (2\pi)^{NR/2} | \Sigma |^{1/2}\}^{-1} \exp\left[-\frac{1}{2} (X-\mu_i)^T \Sigma^{-1} (X-\mu_i) \right]$$

Soit λ le taux de vraisemblance:

$$\lambda = \frac{p (X/C_i)}{p (X/C_j)}$$

et soit

$$L = \ln(\lambda) = \ln\{p(X/C_i)\} - \ln\{p(X/C_j)\}$$

On montre alors que:

$$L = X^T\Sigma^{-1}(\mu_i-\mu_j) - \frac{1}{2}(\mu_i+\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j)$$

et que:

$$E(L/C_i) = \frac{1}{2}(\mu_i-\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j)$$

et:

$$E(L/C_j) = -\frac{1}{2}(\mu_i-\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j)$$

Soit $J(C_i, C_j)$, la divergence entre les classes C_i et C_j définie de la manière suivante (dans le cas de 2 classes):

$$J(C_i,C_j) = E(L/C_i) - E(L/C_j)$$

$$= (\mu_i-\mu_j)^T \Sigma^{-1} (\mu_i-\mu_j)$$

K.S.Fu montre que la probabilité de mauvaise classification est une fonction décroissante de $J(C_i,C_j)$, donc qu'il est possible d'ordonner les paramètres d'après la valeur de $J(C_i, C_j)$.

Dans le cas de plusieurs classes, J sera calculé de la façon suivante:

$$J = \sum_{i=1}^{NR}\sum_{j=1}^{NR} p(C_i)\ p(C_j)\ J(C_i,C_j)$$

et si les distributions sont gaussiennes

$$J = \sum_{i=1}^{NR}\sum_{j=1}^{NR} p(C_i)\ p(C_j)\ (\mu_i-\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j)$$

2) critère d'arrêt

Comme nous l'avons déjà mentionné, l'intérêt principal des méthodes séquentielles est de prendre une décision sans avoir à mesurer toutes les variables. Pour cela, il faut pouvoir disposer de tests indiquant s'il y a lieu ou non d'arrêter la procédure.

Dans le cas de deux classes C_1et C_2, Wald a proposé un rapport séquentiel des probabilités (SPRT: Sequential Probability Ratio Test). Soit λ^α ce rapport considéré à la α-ième étape :

$$\lambda^\alpha = \frac{p(X^\alpha/C_1)}{p(X^\alpha/C_2)} = \prod_{i=1}^{\alpha}\frac{p(x_i/C_1)}{p(x_i/C_2)}$$

la procédure du test est la suivante :

** continuer à prendre des paramètres aussi longtemps que :

$$B < \lambda^\alpha < A$$

** arrêter de prendre de nouveaux paramètres et décider d'accepter C_1 dès que:

$$\lambda^\alpha \geq A$$

** arrêter de prendre de nouveaux paramètres et décider d'accepter C_2 dès que:

$$\lambda^\alpha \leq B$$

Les constantes A et B sont appelées respectivement la limite supérieure et la limite inférieure d'arrêt. Elles peuvent être choisies pour obtenir approximativement les probabilités d'erreur e_{12} et e_{21} demandées.

e_{12} : probabilité de choisir $X \in C_1$ alors que réellement $X \in C_2$

e_{21} : probabilité de choisir $X \in C_2$ alors que réellement $X \in C_1$

Wald démontre alors que pour:

$$A = \frac{1-e_{21}}{e_{12}}$$

et $$B = \frac{e_{21}}{1-e_{12}}$$

on obtient une solution optimale, c'est-à-dire qu'il n'existe pas d'autre procédure donnant des probabilités d'erreurs plus faibles avec un nombre de mesures plus petit.Wald montre que dans certains cas le SPRT peut être non satisfaisant du fait que:

- un test peut être plus long que toléré,
- le nombre de mesures devient extrêmement long (si l' on choisi e_{12} et e_{21} trop petit).

Il suggère une troncature de la procédure après un certain nombre de mesures N_m.

La nouvelle règle de Wald est la suivante :

Conserver le SPRT jusqu'à ce que:

- soit une décision d'appartenance est validée,
- soit l'étape N_m du test est atteinte.

Si aucune décision n'a été prise à l'étape N_m , prendre comme solution $X \in C_1$ si $\lambda > 1$, sinon prendre comme solution $X \in C_2$. D'après cette nouvelle règle, le test doit se terminer au plus tard à la N_m-ième étape.

Pour le cas de plusieurs classes, Reed a proposé un test utilisant un rapport séquentiel généralisé de probabilité (GSPRT : Generalised Sequential Probability Ratio Test).

Supposons que l'on dispose de NR classes, à la α-ième étape.

$$U^{\alpha}(X/C_i) = \frac{p(X^{\alpha}/C_i)}{\left[\prod_{q=1}^{NR} p^{\alpha}(X/C_q)\right]^{1/NR}} \qquad \text{pour } i = 1 \text{ à } NR$$

La règle de décision du GSPRT est la suivante:

- Comparer $U^{\alpha}(X/C_i)$ avec la limite d'arrêt, $A(C_i)$ et supprimer la solution C_i si

$$U^{\alpha}(X/C_i) < A(C_i) \qquad \text{pour } i = 1 \text{ à } NR$$

la limite d'arrêt est déterminée par la relation suivante:

$$A(C_i) = \frac{1-e_{ii}}{\left[\prod_{q=1}^{NR} (1-e_{iq})\right]^{1/NR}}$$

où e_{iq} est la probabilité de décider $X \in C_i$ lorsque réellement $X \in C_q$.

Les classes sont ainsi rejetées séquentiellement jusqu'à ce qu'il ne reste plus qu'une seule classe.

Conclusion

Ces méthodes sont souvent appliquées quand le nombre des paramètres est élevé et que le critère temps joue un rôle important, ce qui est le cas dans des applications robotiques. On les rencontre aussi dans des domaines où la mesure nécessite l'interruption ou l'arrêt total d'un processus industriel, ou comporte des risques comme dans certaines applications biomédicales.

Mais l'inconvénient majeur de ces méthodes est qu'elles supposent au départ que les mesures des paramètres soient statistiquement indépendantes. Cette condition est rarement respectée et, dans ce cas les résultats obtenus lors de la classification sont peu fiables. Cabrera Quintero s'affranchit de ce problème en faisant l'hypothèse d'une dépendance markovienne, et obtient ainsi de bons résultats [CABR-81].

II.3) SELECTION DE PARAMETRES PAR PROGRAMMATION DYNAMIQUE

II.3.1) Introduction

Ces méthodes sont nées de l'idée que le sous-ensemble des m meilleurs paramètres ne fournit pas nécessairement le meilleur sous-ensemble. Comme nous l'avons déjà dit au début de cette thèse, la solution optimale serait de considérer toutes les combinaisons possibles de m paramètres parmi n, puis de rechercher celle qui fournirait la meilleure classification. Malheureusement, ce type de méthodes entraîne une explosion combinatoire du nombre de calculs, et tout particulièrement quand n est grand et m proche de la valeur n/2. La programmation dynamique est une des solutions à ce problème.

La programmation dynamique est une technique d'optimisation à plusieurs niveaux. Le principe d'optimisation est le suivant : quelle que soit la décision initiale, les décisions doivent constituer une politique optimale, en tenant compte de l'état résultant des décisions précédentes. Pour mieux comprendre ce processus de décision à plusieurs niveaux, nous détaillons dans ce qui suit trois exemples.

Le premier exemple est la solution apportée par Cheung en 1973 [CHEU;73], le deuxième est la méthode du " branch and bound ", proposée par Fukunaga et Narendra en 1986 [FUKU;86], le troisième est la méthode de Foroutan et Sklansky [FORO;86]. D'autres auteurs se sont intéressés aux méthodes basées sur la programmation dynamique, mais les variantes diffèrent très peu.

II.3.2) Méthode de Cheung

Cette méthode est basée sur une procédure à plusieurs étapes [CHEU-78]. A la première étape, on recherche pour chaque paramètre P_i, le paramètre P_j avec lequel il constituera le meilleur sous-ensemble de deux paramètres. Pour cela, le paramètre P_i est combiné successivement à tous les autres paramètres, et on retiendra le paramètre pour lequel le sous-ensemble constitué de ces deux paramètres prendra la plus grande valeur du critère. La procédure est réitérée pour chaque paramètre. A la fin de la première étape, on dispose de NP sous-ensembles de deux paramètres.

A la deuxième étape, chaque sous-ensemble obtenu à l'étape précédente est combiné aux (NP-2) paramètres restants, et pour chaque sous-ensemble, on retiendra le paramètre pour lequel le sous-ensemble des trois paramètres prend la plus grande valeur du critère. A la fin de la deuxième étape, on dispose donc de NP sous-ensembles de trois paramètres.

Le processus est répété jusqu'à obtention des sous-ensembles de dimension souhaitée.

A la dernière étape, on dispose de NP sous-ensembles de NK paramètres. Le sous-ensemble final sera sélectionné à l'aide du même critère que précédemment.

Algorithme:

Soit P = { p_1 , p_2 ..... p_{NP} } l'ensemble des NP paramètres disponibles.

Soit P_n^j = { p_j, p_x ..... p_z } l'un des NP sous-ensembles possibles sélectionnés après n-1 étapes et p_n^j représente un paramètre de P. Ce sous-ensemble est donc constitué de n paramètres et provient du sous-ensemble initialement constitué uniquement du paramètre j. On distinguera donc les différents sous-ensembles grâce au numéro du premier paramètre.

Pour chaque x_j, à la n-ième étape, le sous-ensemble P_n^j est choisi de la manière suivante:

$\lambda_n (P_n^j) = \max Dn (P_{n-1}^i , x_j)$ $\quad \forall$ i=1 à N

$x_j \notin P_{n-1}^i$

où (P_{n-1}^i , x_j) représente un sous-ensemble de paramètres formé par l'ajout au sous-ensemble P_{n-1}^i, de l'élément x_j.

$(P_{n-1}^i , x_j) = (P_1^i , P_2^i , \ldots\ldots\ldots\ldots P_{n-1}^i , x_j)$

D_n est défini comme une mesure " d'efficacité ","de qualité" du paramètre et λ_n est la plus grande de toutes ces mesures. Puisqu'il y a NP paramètres à chaque niveau, NP sous-ensembles sont générés.

Le sous-ensemble optimal P_{NK} à la fin des NK-1 étapes peut être obtenu à partir des P_n^j de la manière suivante:

$\lambda_{NK} (P_{NK}) = \max \{ \lambda_{NK}(P_{NK}^j) \}$ pour j =1 ,2 .....NP

Cet algorithme est résumé par le diagramme bloc de la figure 1. La figure 2 donne un exemple de représentation sous forme d'arbre de la méthode de Cheung.

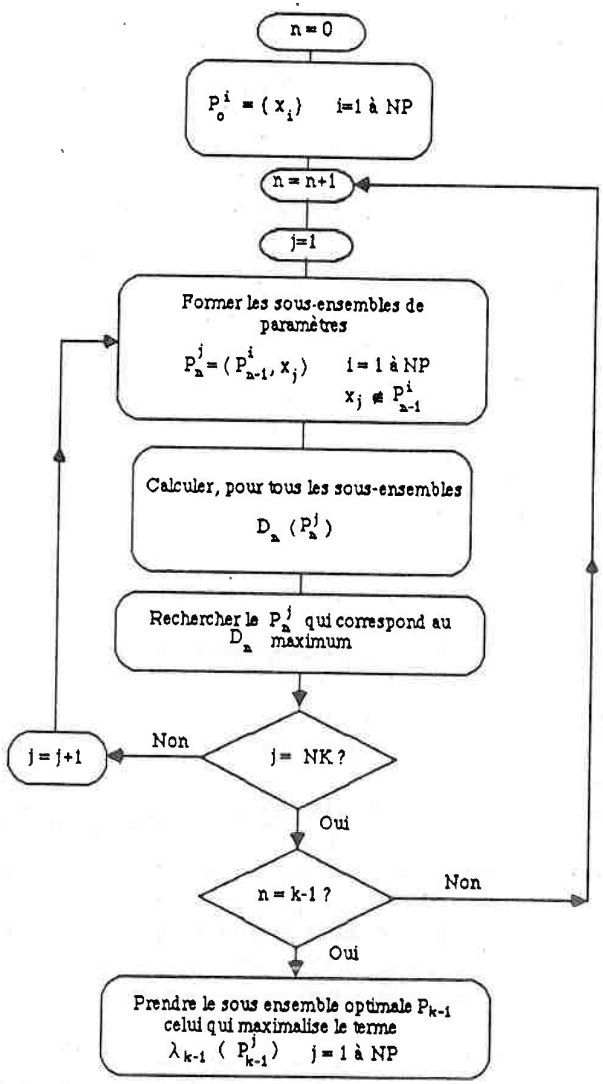


fig 1: Algorithme de sélection de paramètres par programmation dynamique: Méthode de Cheung

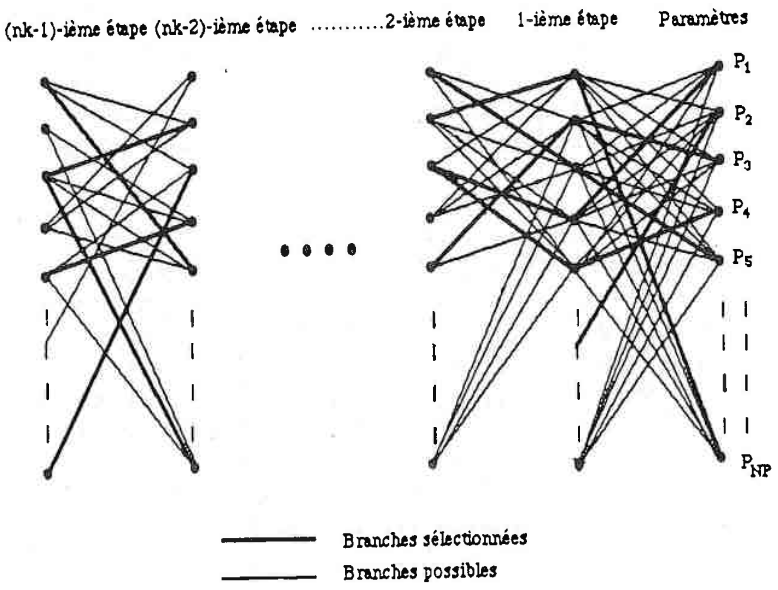


fig 2: Représentation sous forme d'arbre de la méthode de Cheung

II.3.3) L'algorithme du " Branch and Bound"

Principe :[NARE-77],[NARE-78]

Supposons que l'on souhaite trouver le meilleur sous-ensemble de m paramètres parmi n paramètres.

Pour obtenir la solution optimale, la méthode consiste à trouver l'ensemble des paramètres à supprimer. Pour cela, la méthode utilise "un arbre" avec autant de branches que de combinaisons possibles. Chaque branche est elle-même constituée de m nœuds (m=n-m nombre de paramètres à supprimer = nombre de niveaux de l'arbre). Le nœud représente le numéro du paramètre à supprimer. Un nœud à un niveau k sera associé à l'ensemble des nœuds le reliant à la racine. Ainsi, l'ensemble des nœuds d'une branche donne les numéros des m paramètres que l'on supprime, c'est-à-dire une combinaison possible.

Supposons que l'on dispose de 6 paramètres et que l'on souhaite en sélectionner 2 (m=2), il faudra en supprimer 4 (m=4). L'arbre aura donc 4 niveaux et C^4_6 =15 branches.

Exemple: (fig 3) Pour le chemin (la branche) à droite les 4 paramètres rejetés seront (P_3,P_4,P_5,P_6).

Exemple: Au niveau 4, le premier nœud à gauche (4) sera associé aux nœuds 1,2 et 3.

On définit alors un critère $J_k(P_i, P_j....P_k)$ permettant de mesurer la qualité du sous-ensemble des paramètres privé des k paramètres P_i, P_j....P_k (si k<4 on parlera de critère partiel). On impose à ce critère une seule condition :

(C1): $J_1(P_1) \geq J_2(P_1, P_2) \geq \ldots\ldots\ldots\ldots\geq J_m(P_1, P_2, \ldots\ldots\ldots P_m)$

En fait, ce n'est pas une restriction puisque beaucoup de critères la vérifient. En effet, un sous-ensemble de paramètres donne rarement, voire jamais, une meilleure classification qu'un sous-ensemble plus étendu le contenant. Les fonctions discriminantes et les mesures de distances telles que la distance de Bhattacharyya sont des exemples de critère où la propriété est vérifiée.

Cette condition est nécessaire pour la construction de l'arbre relative à l'algorithme.

Supposons que les paramètres de notre exemple soient ordonnés de la sorte :

$$J_1(P_1) < J_1(P_2) < J_1(P_3)\ldots\ldots\ldots\ldots< J_1(P_6)$$

C'est-à-dire que le sous-ensemble privé du paramètre 1 est moins bon que le sous-ensemble privé du paramètre 2, qui est lui-même moins bon que le sous-ensemble privé du paramètre 3 etc... Si l'on considère les quatre derniers paramètres (les plus mauvais P_3 P_4 P_5 P_6), la probabilité pour que le sous-ensemble constitué en supprimant ces quatre paramètres soit la solution optimale est grande. Ce qui explique le fait que la première branche (celle de droite ; la première créée et testée par le programme) soit constituée de ces paramètres.

En effet le programme ne crée pas toutes les branches possibles. Il commence par celle de droite puis construit les suivantes de manière que les branches ayant de fortes chances d'être la solution (constituée des plus mauvais paramètres) soient le plus à droite possible.

Remarque : Quelque soit le nœud considéré, tous ses successeurs sont forcément des paramètres plus mauvais que lui. Ce qui explique le fait que certains paramètres possèdent plus de successeurs que d'autres.

Le programme construit l'arbre branche par branche, le test d'arrêt s'effectue en chaque nouveau nœud créé, à l'aide du critère partiel J_k. On cherche en effet le sous-ensemble de paramètres privé de quatre paramètres ayant la plus grande valeur du critère $J_4(.)$. Après création de la première branche, on dispose de la première valeur du critère $J_4(.)$, que l'on considérera comme la valeur test B. On prend alors comme solution optimale:

$$(P_1^*, P_2^*, \ldots\ldots\ldots P_m^*) = (P_3, P_4, P_5, P_6)$$

et $$B = J_4(P_3, P_4, P_5, P_6)$$

A chaque nœud, la valeur du critère partiel J_k est calculée et comparée à B. Si cette valeur est supérieure à B, la construction de l'arbre continue si par contre, elle est inférieure à B la construction de cette branche et donc de toutes ses ramifications est stoppée et on remonte

d'un niveau dans l'arbre. En effet, si à un nœud donné la valeur du critère partiel est telle que :

(T1): $J_k(P_1, P_2 \ldots\ldots P_k) \leq B$ alors

$J_k(P_1, P_2 \ldots\ldots P_k) \leq J_m(P_1^*, P_2^*, \ldots\ldots\ldots\ldots P_m^*)$

$(P_1^*, P_2^*, \ldots\ldots\ldots\ldots P_m^*)$ représente la solution optimale, donc la dernière solution trouvée.

Or d'après la condition (C1)

$J_m(P_1, P_2 \ldots\ldots P_k, P_{k+1}, \ldots\ldots\ldots P_m) \leq J_k(P_1, P_2 \ldots\ldots P_k)$ donc

$J_m(P_1, P_2 \ldots\ldots P_k, P_{k+1}, \ldots\ldots\ldots P_m) \leq J_m(P_1^*, P_2^*, \ldots\ldots\ldots\ldots P_m^*)$

Cela signifie que si la valeur du critère partiel $J_k(P_1, P_2 \ldots\ldots P_k)$ relative à un nœud est inférieure à B, alors tous les nœuds, qui succèdent à ce nœud auront une valeur du critère partiel inférieure à B . Donc toute la branche et ses ramifications sont rejetées car elles ne fourniraient pas une solution optimale.

Exemple *:* fig 4

Lors de la création de la deuxième branche,

- le programme crée le nœud 2 et calcule $J_1(P_2) = 30$ la valeur du critère partiel est supérieure à B =25, donc passe au niveau suivant (2) .

- Il crée alors le nœud 4 et calcule $J_2(P_2, P_4) = 26$ la valeur du critère partiel est supérieure à B =25, donc passe au niveau suivant (3).

- Le programme crée le nœud 5 et calcule $J_3(P_2, P_4, P_5) = 22$, la valeur du critère partiel est inférieure à B =25 ,donc la construction de cette branche est stoppée.

- Le programme remonte d'un niveau (2), le nœud 4 n'ayant pas d'autre ramification, le programme remonte encore d'un niveau (1), le nœud 2 possédant d'autres ramifications , le programme va donc étudier cette troisième branche potentielle.

Si une nouvelle branche est créée entièrement, alors elle sera considérée comme la nouvelle solution optimale et la valeur du critère correspondante sera la nouvelle valeur test B.

Si l'algorithme parvient au niveau 0 , le programme est terminé.

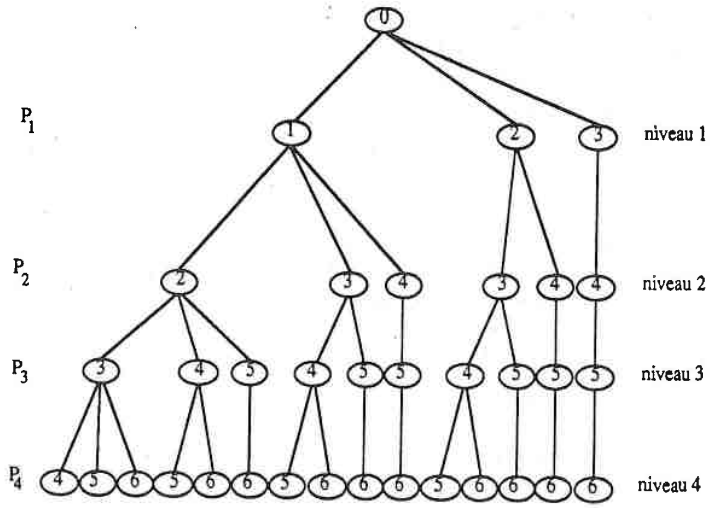


fig 3: Exemple de construction d'un arbre avec n=6 ; m=2 ; m=4

Algorithme

Notation :

***liste(i):** liste ordonnée des paramètres au niveau i.

***pointeur(i):** pointeur qui indique l'élément de **liste (i)** qui est étudié: par exemple si le k-ième élément de liste(i) est en train d'être étudié alors **pointeur(i)=k.**

***successeur(i,k):** nombre de successeurs que le k-ième élément de **liste(i)** possède.

***disponible:** liste des valeurs des paramètres disponibles que **liste(i)** peut prendre.

Etape n°0

B=0

disponible ={1,2,.....n}

i=1

liste (0)=0

successeur (0,1)=m+1

pointeur(0)=1

Etape n°1 **nœud=pointeur**(i-1)

calculer $J_i(P_1, P_2, \ldots\ldots\ldots k)$ pour tous les k dans **Disponible**

ranger les paramètres dans **Disponible** dans l'ordre croissant des J_i

p=**successeur**(i-1,**nœud**)

ranger les p premiers numéros de paramètre dans **liste**(i)

successeur(i,j)=p-j+1 pour j=1 à p

supprimer **liste**(i) de **Disponible**

Etape n°2 Si **liste**(i) est vide aller à l'étape 4

sinon

P_i=k k étant le dernier élément de **liste**(i)

pointeur(i)=nombre d'éléments dans **liste**(i)

supprimer k de **liste**(i)

Etape n°3 Si $J_i(P_1,P_2,.....P_i) < B$ remettre P_i dans **Disponible**, aller à l'étape 4

Sinon

Si i=m aller à l'étape 5

Sinon i=i+1 aller à l'étape 1

Etape n°4 i=i-1

Si i=0 fin de l'algorithme

Sinon remettre P_i dans **Disponible** et aller à l'étape 2

Etape n°5 $B= J_m(P_1,P_2,.....P_m)$

$(P_1^*,P_2^*.....P_m^*) =(P_1,P_2,.....P_m)$

remettre P_m dans **Disponible** et aller à l'étape 4

Déroulement de l'algorithme

Le fonctionnement de l'algorithme est le suivant:

Démarrer à partir de la racine de l'arbre . Les successeurs du nœud courant sont contenus dans **liste**(i). Le successeur, pour lequel le critère partiel $J_k(P_1,P_2,.....P_k)$ est le plus grand (le successeur le plus à droite), est pris comme nouveau nœud courant et l'algorithme va au niveau supérieur suivant. Les variables **successeur** déterminent le nombre de nœuds successeurs, que le nœud courant aura au niveau suivant. **Disponible** contient les paramètres qui peuvent être sélectionnés et placés dans **liste**(i) , à n'importe quel niveau. Si à un niveau quelconque, le critère partiel est inférieur à la limite B, l'algorithme remonte la branche jusqu'au niveau précédent et sélectionne un nœud non exploré.

Si l'algorithme parvient au niveau m, alors la valeur courante $J_m(P_1,P_2,.....P_m)$ constitue la nouvelle limite B et la séquence $(P_1,P_2,.....P_m)$ est sauvegardée comme la nouvelle solution optimale $(P_1^*,P_2^*.....P_m^*)$.

Lorsque tous les nœuds dans **liste**(i), pour un niveau donné ont été lus, l'algorithme remonte au niveau précédent. Si l'algorithme parvient au niveau 0 , le programme est terminé.

Exemple:

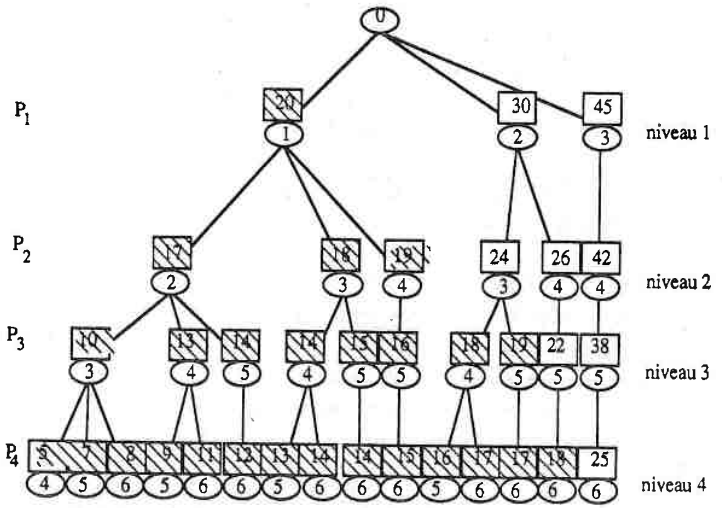


fig 4: Exemple de branches non lues

Les valeurs dans les carrés représentent la valeur du critère. Toute la partie hachurée n'est pas créée par le programme.

Chemin suivi:

paramètres	Valeurs du critère partiel	Nombre de paramètres supprimés
3	45	1
3;4	42	2
3;4;5	38	3
3;4;5;6	25 B=25	4

2	30 >B	1
2;4	26>B	2
2;4;5	22<B fin de lecture de cette branche	3
2;3	24<B fin de lecture de cette branche,	

Et on ne lira pas la branche suivante car le paramètre 1 est meilleur que le paramètre 2. Donc, si on recommence les mêmes combinaisons avec le paramètre 1 qu'avec le paramètre 2, on obtiendra forcément une valeur de critère partiel inférieure à B. On remonte d'un niveau, donc au niveau 0 c'est la fin du programme.
Le résultat final est donc la suppression des paramètres numéros 3 ; 4 ; 5 ; 6

II.3.4) La méthode de Foroutan et Sklansky

But: Trouver le plus petit sous-ensemble de paramètres qui donne une classification avec une erreur inférieure à une limite donnée.

Approche de la méthode:
Soit le vecteur à valeur binaire $\underline{a} = (a_1, a_2,.....a_{np})^T$
np étant le nombre initial de paramètres.
Si a_i=1, alors le i-ième paramètre est sélectionné.
Si a_i=0, alors le i-ième paramètre n'est pas sélectionné.
Le problème est de trouver un vecteur solution $\underline{a}$ qui minimise la fonction $f(\underline{a})$:
$f(\underline{a})=a_1+a_2+.....+a_{np}$ sous la contrainte

$$J(X/\underline{a}) \leq J_0 \qquad (1)$$

$J(X/\underline{a})$ est la fonction critère calculant l'erreur de classification faite pour un ensemble échantillon X, lorsque seuls les paramètres identifiés par $\underline{a}$ sont sélectionnés.
Donc, la minimisation de $f(\underline{a})$ sous la contrainte (1), est équivalente à la recherche du plus petit sous-ensemble de paramètres pour lequel la valeur du critère $J(X/\underline{a})$ resterait inférieure à une valeur imposée au départ J_0 (valeur limite).

Condition imposée au critère
Le critère doit être une fonction monotone croissante du nombre de paramètres (Comme pour la méthode "Branch and Bound").

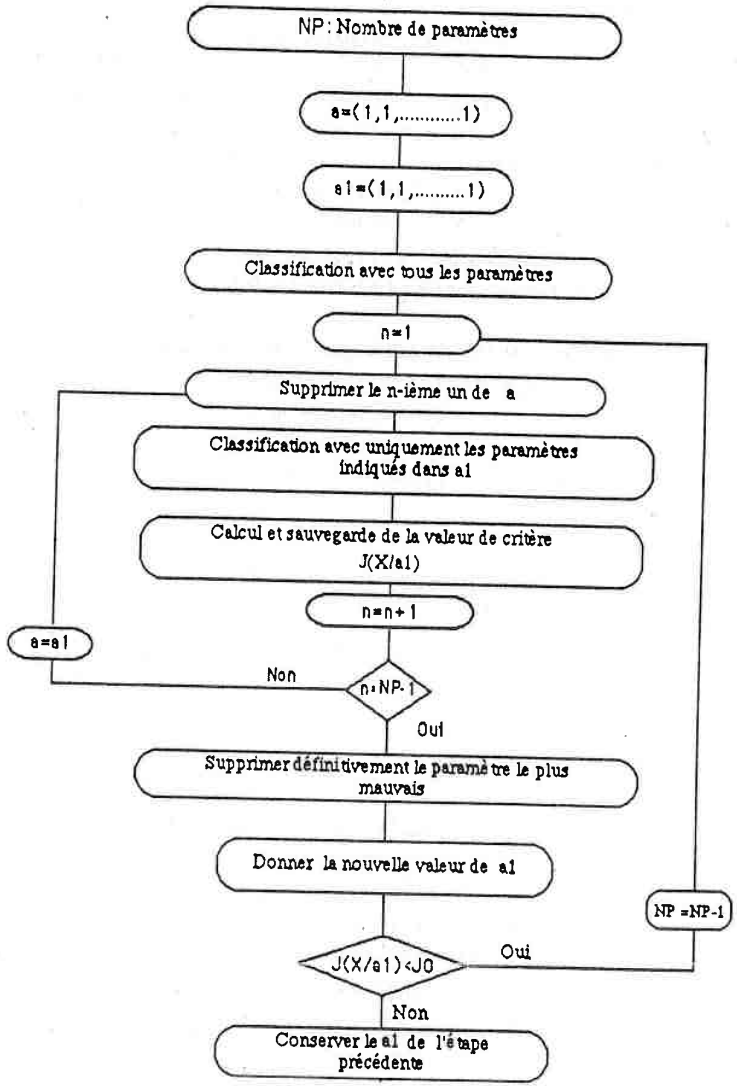


fig 5: Organigramme de la méthode de Foroutan & Sklansky

Algorithme

L'algorithme commence par classer toutes les formes en utilisant tous les paramètres. La méthode que nous avons utilisée pour la classification est une méthode de réallocation. Puis à chaque étape, l'algorithme supprime un paramètre, le plus "mauvais". Pour cela, il calcule le critère en supprimant d'abord un premier paramètre, puis en supprimant uniquement le second ,etc ....., et décide de supprimer définitivement le paramètre relatif au plus grand J(X/$\underline{a}$). On obtient alors un nouveau $\underline{a}$.
Il teste la valeur de J(X/$\underline{a}$). Si cette dernière est inférieure à la limite fixée J_0, l'algorithme recherche le nouveau plus "mauvais paramètre" parmi les paramètres restants. Sinon le programme est terminé et l'on conserve le vecteur $\underline{a}$ de l'étape précédente. L'organigramme de la méthode est donné figure 5.

L'avantage de cette méthode est que le nombre de paramètres à sélectionner n'est pas fixé au départ; le programme s'arrête lorsque le test d'arrêt est vérifié.

II.4) METHODES D'ANALYSE EN COMPOSANTES PRINCIPALES

II.4.1) Introduction

Ces méthodes diffèrent par leur principe des autres méthodes. En effet, la sélection de paramètres est ici considérée comme un problème de réduction de dimension de l'espace des formes E^n. Cette réduction est réalisée par la recherche de l'espace optimal E^m, où va s'effectuer la classification, mais surtout par celle de la transformation à appliquer aux vecteurs formes avant la classification.

Les transformations utilisées pour cette réduction sont linéaires. Ainsi, les nouveaux paramètres sont toujours des combinaisons linéaires des mesures initiales. La méthode de Karhunen-Lœve et celle de Fisher sont aujourd'hui des méthodes très utilisées pour résoudre ce problème. Pour mieux comprendre ces deux méthodes, nous rappelons brièvement le principe de l'analyse en composantes principales, puis nous développerons ces deux méthodes.

II.4.2) Principe de l'analyse en composantes principales

Dans l'espace des formes E^n, chaque forme est représentée par un point X_k de coordonnées:

$$\{x_{k1}, x_{k2}, \ldots x_{ki}, \ldots x_{kn}\}$$

La distance euclidienne entre deux formes X_k et X_l notée $d(X_k, X_l)$ est définie par :

$$d^2(X_k, X_l) = \sum_{i=1}^{n} (x_{ki} - x_{li})^2$$

On se propose alors de représenter ces formes dans un espace X^m (m<n) telles qu'en moyenne, les distances entre tous les couples de points dans E^m soient une bonne représentation des distances dans E^n.
Cherchons un sous-espace E^1 de E^n, de dimension 1, et remplaçons les points X_k de E^n par leurs projections orthogonales sur E^1, que l'on notera $P_1(X_k)$. Le problème est donc de trouver le sous-espace E^1, qui maximalise le terme :

$$\sum_{k=1}^{N} \sum_{l=1}^{N} \| P_1(X_k) - P_1(X_l) \|^2$$

où N représente le nombre de formes.

En effet, en projetant les formes sur un axe u_1, on cherche à les étaler au maximum pour les rendre bien séparables.

La solution est le sous-espace engendré par le vecteur propre V_1, associé à la plus grande valeur propre de la matrice de covariance. E^1 est appelé premier axe factoriel de l'ensemble des points et les projetés des points sont appelés premières composantes principales.

Plus généralement, chercher un espace E^m satisfaisant au même critère revient à chercher les vecteurs propres unitaires V_1, V_2..... V_m, associés aux plus grandes valeurs propres de la matrice de covariance.

II.4.3) La méthode de Karhunen -Lœve

Comme nous l'avons déjà vu, une forme est représentée dans l'espace des formes E^n de dimension n par un point X_k.

Réduire le sous ensemble des paramètres, c'est donc passer de l'espace forme E^n de dimension n à un espace forme E^m moins étendu de dimension m (m < n).

La méthode de Karhunen - Loeve nous propose un opérateur K d'extraction de paramètres. Cet opérateur transforme un vecteur X à n composantes dans la base Ω_X de E^n en un vecteur Y à m composantes dans la base Ω_y de E^m.

$$Y = K X \quad \text{et} \quad K: \text{matrice m x n}$$

La validité de l'extraction de paramètre réside dans le choix optimal de la nouvelle base représentative Ω_y. La base de Karhunen-Lœve minimise l'erreur moyenne commise dans l'approximation du vecteur X par le vecteur Y.

Développement de la méthode de Karhunen-Lœve

- soit ψ une base orthonormée de E_X composée de n vecteurs ψ_i, i = 1,2..... n

Tout vecteur de forme X peut être décomposé dans cette base:

$$X = \sum_{i=1}^{n} x_i \psi_i$$

- soit φ la nouvelle base de l'espace E_y composée de m vecteurs:

$$\varphi_j \quad j = 1, 2, \ldots, m$$

un vecteur X devient un vecteur Y défini par :

$$Y = \sum_{i=1}^{m} y_i \varphi_i$$

K étant la matrice de transformation, les composantes y_j se déduisent des composantes x_i par la relation:

$$y_j = \sum_{i=1}^{n} K(i,j)\, x_i$$

La méthode de Karhunen - Loeve montre que la base optimale est formée des vecteurs propres φ_j de la matrice de covariance A des formes considérées donc :

$$A(i,j) = E(X, X^T) = \sum_{\alpha=1}^{n} p(\alpha).E(x_i^{\alpha}.x_j^{\alpha})$$

Ces vecteurs peuvent être normalisés; en effet si Ψ est un vecteur propre de la matrice de covariance A , alors c. Ψ (c: constante arbitraire) est aussi vecteur propre, donc on peut aisément s'imposer $\Psi^T . \Psi = 1$.

Le but étant de réduire le nombre des paramètres, la méthode de Karhunen-Lœve ne conservera que m vecteurs de la base choisie parmi les n vecteurs propres Ψ_j de la matrice de covariance A.

Le choix des m vecteurs propres parmi les n vecteurs propres est réalisé de la manière suivante :

- on range suivant un ordre décroissant les valeurs propres.

$$\lambda_1 \geq \lambda_2 \geq \ldots\ldots \geq \lambda_n$$

La nouvelle base sera alors constituée des m vecteurs propres associés aux m premières valeurs propres.

Karhunen - Lœve montre que cette base est la solution optimale et que l'erreur est minimale.

(La démonstration de l'optimalité du développement de Karhunen-Lœve est donnée en annexe B)

II.4.4) Méthode du discriminant linéaire de Fisher

Cette méthode détermine la transformation à appliquer au vecteur forme initial X en recherchant la direction de l'axe pour lequel la mesure généralisée de Fisher prend sa valeur maximale. Cette mesure se présente sous forme d'un quotient. Ce dernier est proportionnel à la matrice de dispersion interclasse S_B ("Beetween scatter matrix ") et inversement proportionnelle à la matrice de dispersion intraclasse S_W ("Within scatter

matrix ") .S_W mesure la dispersion des formes à l'intérieur d'une classe. S_B est utilisée comme une mesure de distance entre les classes.

Principe

Supposons données des formes de N_c classes dans un espace de dimension n. Supposons que la distribution des formes dans chaque classe soit normale N (μ_i, Σ_i), où μ_i est le vecteur espérance et Σ_i la matrice de covariance des observations dans la classe C_i pour i= 1.....N_c. Les grandeurs μ_i, Σ_i ainsi que la probabilité a priori P (C_i) sont supposées connues.

La mesure de la distance de Fisher entre deux classes C_i et C_j est définie par : (d représente le vecteur colonne sur lequel les données vont être projetées)

$$F_{ij}(d) = \frac{d^T S_{Bij}\, d}{d^T S_{Wij}\, d}$$

$$\text{Où} \quad S_{Wij} = \frac{1}{N_i} \sum_{X \in C_i} (X-\mu_i)(X-\mu_i)^T + \frac{1}{N_j} \sum_{X \in C_j} (X-\mu_j)(X-\mu_j)^T$$

$$= \Sigma_i + \Sigma_j$$

$$\text{et} \quad S_{Bij} = (\mu_i - \mu_j)(\mu_i - \mu_j)^T$$

Dans les équations précédentes

$$\mu_i = \frac{1}{N_i} \sum_{X \in C_i} X$$

La mesure généralisée de Fisher pour N_c classes est donnée par :

$$F(d) = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} P(C_i)\, P(C_j)\, .\, F_{ij}(d)$$

Le but est donc de trouver le vecteur d sur lequel nous allons projeter les formes, de manière à ce que la dispersion interclasse soit maximale et que la dispersion intraclasse soit minimale. Donc, on recherche le vecteur d, pour lequel la valeur de F(d) atteint sa valeur maximale.

La résolution de ce problème n'est pas simple, Malina, sous des hypothèses simplificatrices, montre que d est le vecteur propre associé à la plus grande valeur propre de la matrice $A^{-1}S$.

$$A = \sum_{i=1}^{N_c} P(C_i)\, \Sigma_i$$

$$S = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} P(C_i)\ P(C_j)\ S_{Wij}$$

Malina montre aussi que pour obtenir les m nouveaux paramètres (les plus discriminants), il suffit de prendre les m vecteurs propres associés aux m plus grandes valeurs propres de la matrice $A^{-1}S$.

Ces m paramètres ne seront pas corrélés entre eux à la condition que le rang de la matrice $A^{-1}S$ soit supérieur ou égal à m.

II.4.5) Conclusion

L'inconvénient de ces méthodes est que lorsque le nombre de données est trop important, il devient très compliqué et très long de calculer l'inverse de la matrice A. De plus, les hypothèses de départ ne sont pas toujours vérifiées.

Aussi, du fait de la nature de la transformation, les nouvelles variables sont des combinaisons linéaires des variables initiales. De ce fait, la réduction du nombre de paramètres n'est effective que pour la phase de classification proprement dite et il demeure nécessaire de mesurer toutes les variables pour pouvoir calculer les nouvelles. Par conséquent, ces méthodes ne peuvent convenir pour résoudre le problème que nous nous posons. Cependant, le critère de sélection proposé dans la méthode de Fisher nous a semblé très intéressant. En effet, ce critère tient compte à la fois de la dispersion interclasse et de la dispersion intraclasse. Or, il est clair que le choix des paramètres doit se faire de manière que la représentation des points formes dans l'espace réduit soit telle

- que les points appartenant à une même classe soient regroupés au maximum autour de leur représentants(s) (dispersion intraclasse faible)
- et que les représentants des différentes classes soient le plus espacés possible (dispersion interclasse grande).

Pour cette raison, nous avons choisi comme critère de sélection dans notre partie expérimentale un critère très proche de celui de Fisher.

CHAPITRE III

EXPERIMENTATION

I) Introduction

Ce chapitre est consacré à la description des différents programmes et en particulier du programme général. Nous présenterons les différentes fonctions de ce dernier, que nous pouvons résumer de la façon suivante (fig 1):

- Lecture des différentes images d'un lot considéré.
- Bruitage des images du lot avec pourcentage variable et fixé par l'opérateur.
- Suivi et codage du contour de chaque image.
- Obtention des 19 paramètres pour chaque image.
- Sélection de paramètres dont le nombre est fixé par l'utilisateur
- Classification des formes du lot en se basant sur les paramètres sélectionnés. A ce stade, les classes obtenues doivent correspondre aux classes prédéfinies du lot d'apprentissage.
- Reconnaissance de forme sur des images identiques à celles du lot d'apprentissage, mais affectées d'un bruit additif plus ou moins important. Le taux de reconnaissance qualifie alors la totalité du processus.

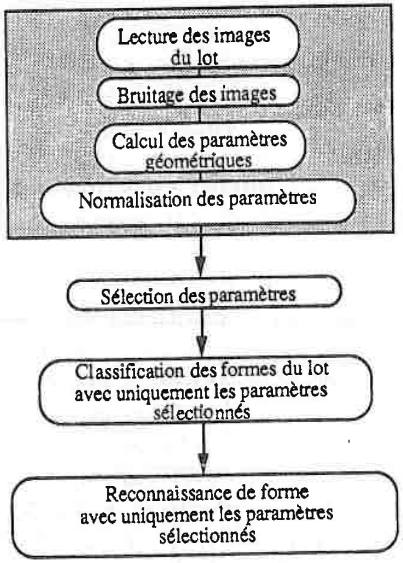


fig 1 : Organigramme du programme général

Ce chapitre comportera donc cinq parties.

. Dans la première, nous commencerons par décrire les lots d'images créés, support de notre recherche. Nous présenterons alors dans un ordre chronologique les différentes tâches effectuées par le programme principal, précédant la sélection des paramètres (bruitage des images, suivi de contour, calculs des paramètres géométriques, normalisation de ces valeurs).

. La deuxième partie est consacrée aux différents algorithmes de sélection de paramètres. Les algorithmes utilisés pour la sélection de paramètres sont basés sur trois méthodes dynamiques, vues précédemment, et qui interviennent préalablement à la classification :

- La méthode de Foroutan et Sklansky,
- La méthode de Cheung,
- La méthode du Branch and Bound.

. La troisième partie concerne les algorithmes de classification. Deux méthodes sont présentées :

- Méthode des K-means,
- Méthode de Splitting.

Comme nous le signalions un peu plus haut, la classification n'est ici réalisée que dans le but de vérifier que les étapes précédentes (et tout particulièrement l'étape de sélection) fournissent des résultats cohérents.

. La quatrième partie est consacrée au programme de reconnaissance de forme. Comme la précédente, cette partie nous permettra de mesurer, d'estimer la qualité des différentes méthodes de sélection. En effet, pour une image donnée, affectée d'un pourcentage de bruit donné, ce programme fournit en fin d'exécution le pourcentage de bonne reconnaissance.

. Dans la dernière partie, plusieurs types de résultats sont donnés puis discutés.

II) APPRENTISSAGE

II.1) Différents programmes et sous-programmes

II.1.1) Création d'images binaires

Pour tester les algorithmes écrits, et particulièrement les algorithmes de calcul de paramètres géométriques, nous avons créé des images synthétiques binaires (les images à niveaux de gris pouvant se ramener à ce cas par simple seuillage, précédé éventuellement d'un prétraitement d'amélioration d'image).

Nous nous sommes limités dans un premier temps à des images 40x40, en raison du nombre important de traitements. Des images plus grandes auraient nécessité un temps de calcul trop grand et une place mémoire trop importante, sans pour cela apporter quelque chose de nouveau à notre recherche.En effet les paramètres retenus pour notre expérimentation sont essentiellement des paramètres géométriques, qui tiennent plutôt compte de la forme que de la taille.

Nous avons donc créé un catalogue de plusieurs lots d'environ quinze images chacun, les lots étant bien entendu différents les uns des autres, pour tester la fiabilité de nos programmes.

Le premier lot, le plus simple, est constitué de trois cercles, de trois carrés, de trois triangles et d'une croix, donc de quatre classes de formes. Le deuxième lot est constitué de onze images réparties en cinq classes : trois cercles, deux cercles troués, deux carrés, deux triangles et deux triangles troués.

Chaque image se trouve dans un fichier, préalablement constitué, qui porte le nom de la forme.

exemples : CA2020 représente un carré de côté 20 pixels,
CL102020 représente un cercle de centre (20,20) et de rayon 10,
TR1026 représente un triangle de base 10 et de hauteur 26.

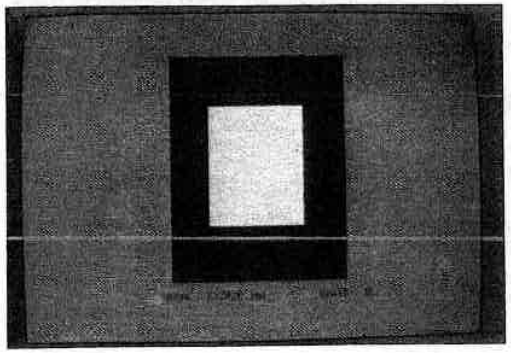


Photo n° 1 : carré de coté 20 pixels image non bruitée

Photo n°2 :
cercle de rayon 10

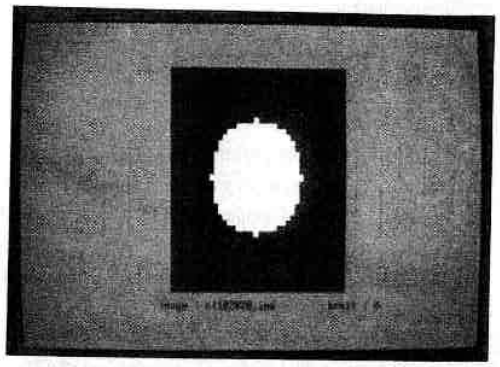


Photo n°3 :
triangle

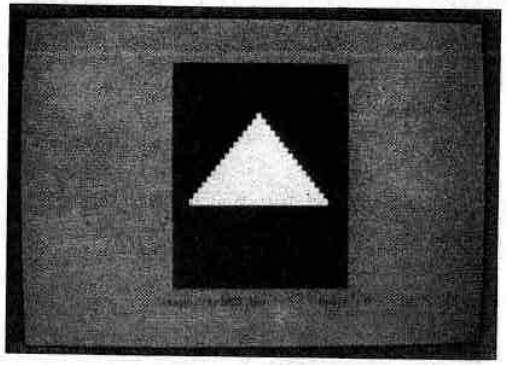
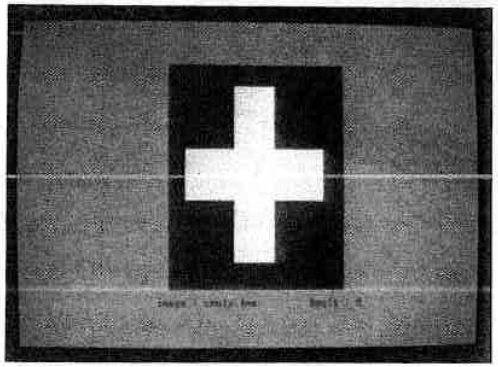


Photo n° 4 :
croix



II.1.2) Présentation du programme Général

Lors du lancement du programme principal, ce dernier demande à l'opérateur avec quel lot il désire travailler, et s'il souhaite bruiter le contour. Lorsqu'un lot a été choisi par l'opérateur, le programme va lire un à un les fichiers images constituant le lot, ainsi que le nombre d'images NI du lot. La lecture d'un fichier image ne se fait qu'une fois les traitements de mesures géométriques terminés pour l'image en cours, et une fois que ces mesures sont sauvegardées. Pour chaque image du lot, la ou les valeurs des mesures (paramètres) sont systématiquement rangées dans un tableau **PARAM1** de dimension NI x NP, qui sera lui-même sauvegardé dans le fichier **PARAIMA** à la fin de tous les calculs de paramètres géométriques (fig 2).

NI : représente le nombre d'images du lot considéré.

NP : représente le nombre de mesures effectuées sur chaque image.

Avant de sélectionner les paramètres les plus pertinents, les valeurs du fichier **PARAIMA** sont normalisées et rangées dans le fichier **PARANORM** et cela, quelle que soit la méthode de normalisation.

L'étape de normalisation étant franchie, le programme demande à l'utilisateur le nombre de paramètres qu'il souhaite sélectionner ainsi que la méthode de sélection qu'il souhaite appliquer. Cette dernière recherche alors les paramètres pertinents et range les numéros ordonnés des paramètres sélectionnés dans le fichier **SELECT** (fig 3).
Cette opération est alors suivie d'une classification puis d'une reconnaissance de forme, que nous détaillerons dans les chapitres suivants.

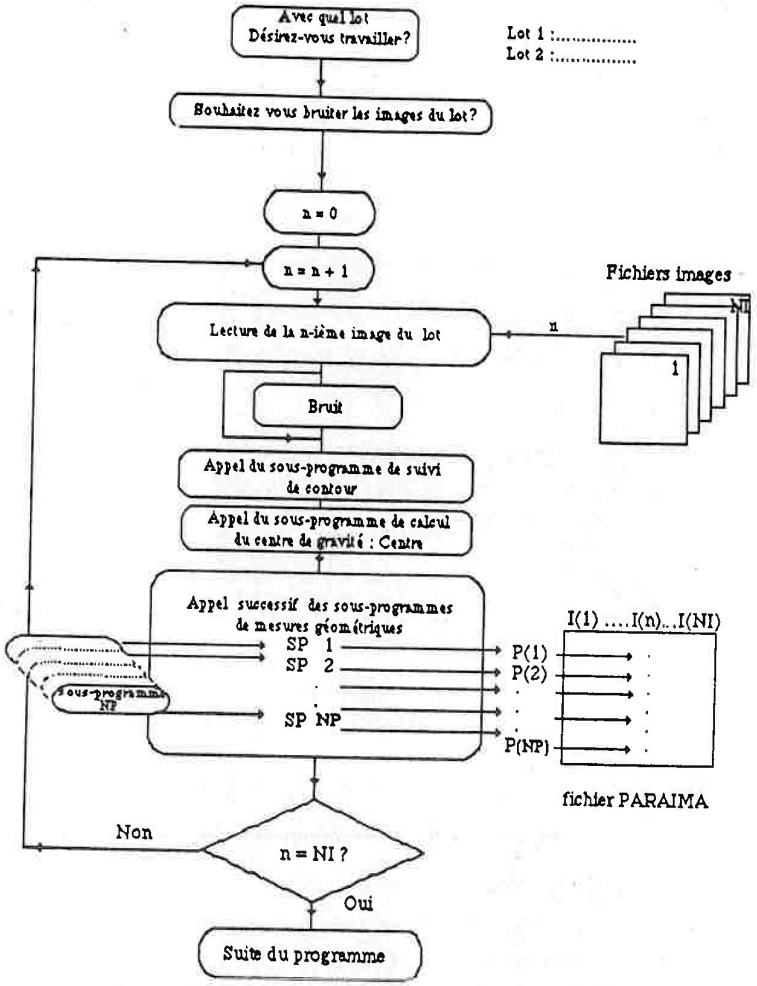
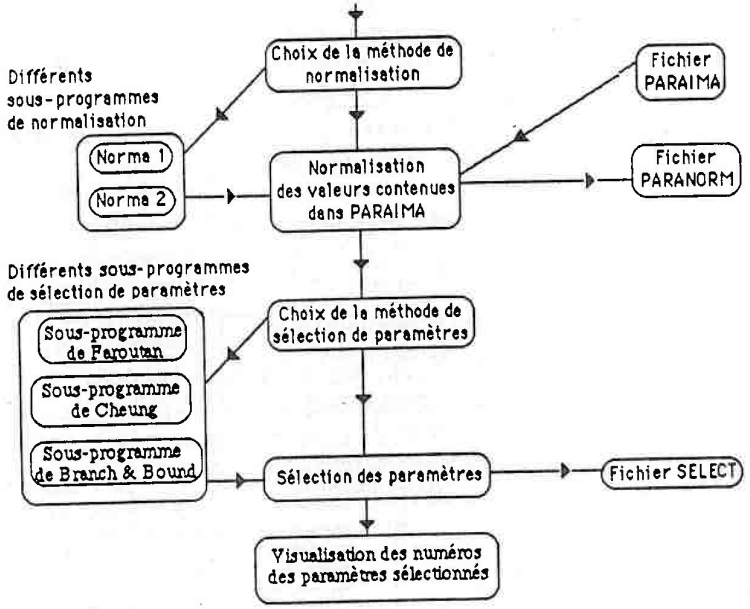


fig 2 : Algorithme de la première partie du programme principal :
Lecture des images d'un lot et calcul des paramètres géométriques

fig 3 : Suite du programme principal :
Normalisation et sélection des paramètres



II.1.3) Sous-programmes de traitements préliminaires

II.1.3.a) Bruit : Rando

Le premier prétraitement est un prétraitement optionnel qui consiste à bruiter le contour des images, avant de calculer les différents paramètres géométriques. En fait, ce n'est pas à proprement dit un prétraitement, puisque par définition un prétraitement est en général réalisé pour améliorer l'image et non pour la détériorer.

Ce prétraitement est réalisé par le sous-programme Rando. Lors de son exécution, un certain nombre de points du contour sont supprimés aléatoirement. Ce nombre est calculé à partir de la valeur du périmètre et du pourcentage souhaité par l'utilisateur.

Nombre = pourcentage x périmètre

Nous présentons ci-dessous quelques exemples d'images bruitées à 50%, 75% et 100%, sur lesquelles nous avons travaillées (fig 4). Nous avons bouclé le programme de manière à pouvoir creuser davantage le contour, introduisant ainsi la notion de profondeur de bruit.

Exemples d'images bruitées

Profondeur 1

Photo 5 :
Croix
bruitée à 50%

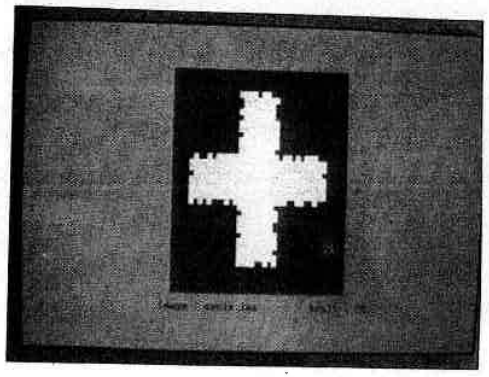


Photo 6 :
Carré de 20 pixels
de côté
bruité à 50%

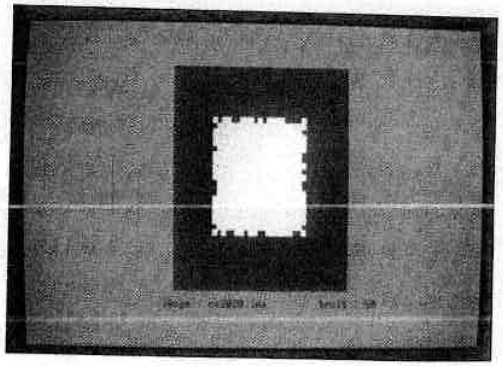


Photo 7 :
Carré de 20 pixels de côté
bruité à 100 %

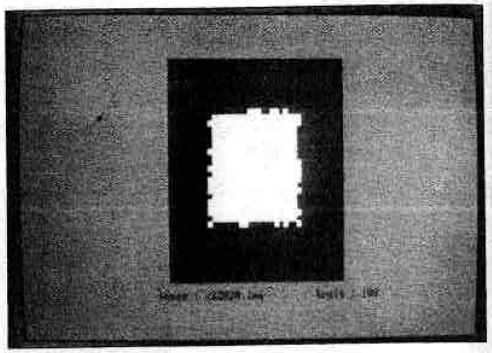
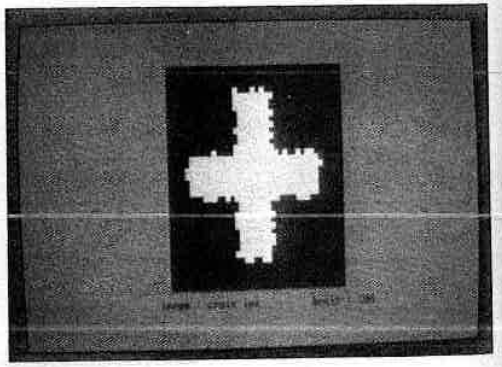


Photo 8 :
Croix
bruité à 100%



Profondeur 2

Photo 9:
Cercle
bruité à 75 %
prof : 2

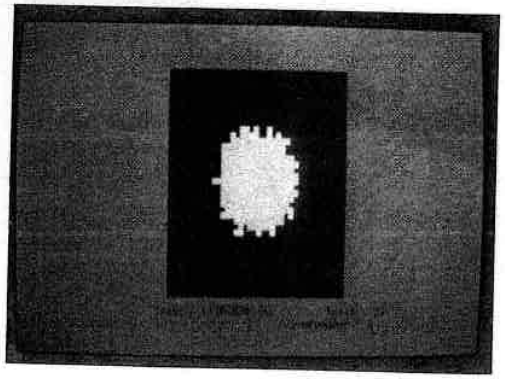
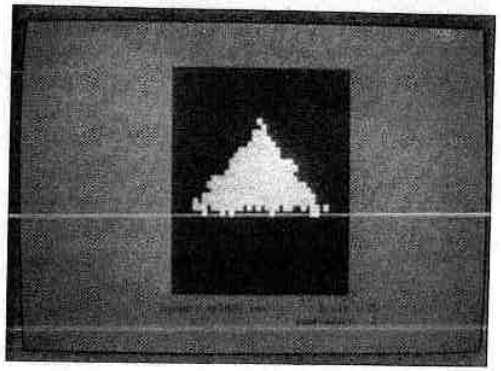


Photo 10:
Triangle
bruité à 75%
prof : 2



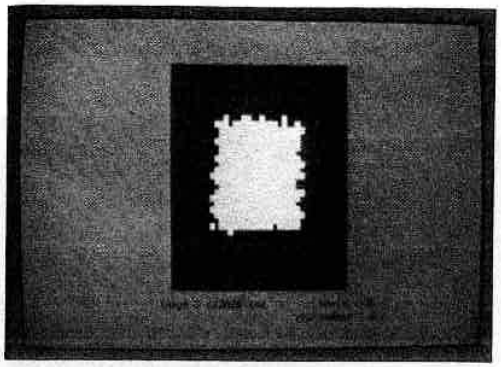


Photo 11:
Carré
bruité à 75%
prof : 2

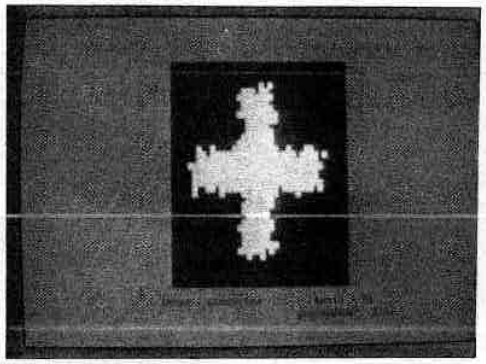


Photo 12:
Croix
bruité à 75%
prof : 2

Profondeur 3

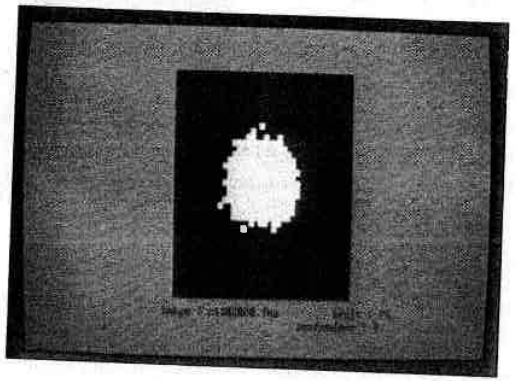


Photo 13:
Cercle
bruité à 75%
prof : 3

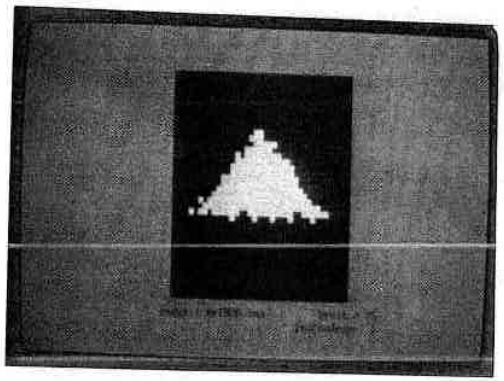
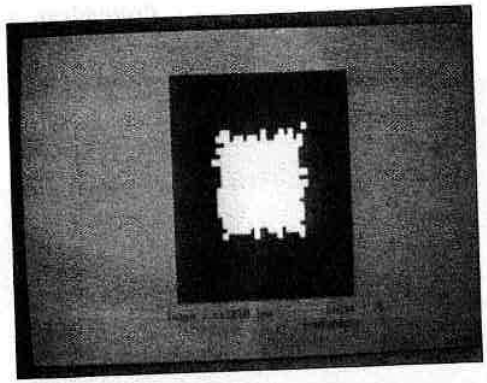


Photo 14:
Triangle
bruité à 75%
prof : 3

Photo 15:
Carré
bruité à 100%
prof : 3



II.1.3.b) Suivi de contour : Freeman

Le deuxième sous-programme à être appelé est le sous-programme ***FREEMAN***. Comme son nom l'indique, il réalise le suivi de contour de la forme et le codage de Freeman . Durant l'exécution du programme, chaque point du contour est mémorisé dans un tableau appelé **RES ()**, à trois colonnes. La première colonne contient les abscisses de ces différents points, la deuxième les ordonnées et la troisième un entier compris entre 1 et 8, indiquant d'après le codage de Freeman la direction prise par le contour après ce point. Ce tableau nous permettra entre autre de calculer le périmètre et la surface de la forme.

Ce n'est pas la seule fonction de ce sous-programme; en effet, lors de la recherche du contour, le programme supprime les points isolés (dus au bruit), mais aussi toute forme dont le périmètre est inférieur à 8 (Remarque : le programme ne supprime pas tous les points isolés de l'image mais uniquement ceux rencontrés lors de la recherche du contour).

II.1.3.c) Centre de gravité : Centre

Le troisième prétraitement est le calcul des coordonnées du centre de gravité du contour G (X_G, Y_G). Ce prétraitement est réalisé par le sous-programme ***CENTRE***. Comme dans le cas précédent, les résultats de ce sous-programme seront utilisés par d'autres sous-programmes (calcul du rayon moyen, de la variance du rayon).

II.1.4) Les différents sous-programmes de calcul de paramètres géométriques

Les sous-programmes de calcul de mesures géométriques ont été comme les autres écrits en quick-basic sur un IBM AT. Leurs principes de calcul ont été détaillés au chapitre II. Nous présentons ci-dessous la liste des différents sous-programmes ainsi que leur fonction.

PERIMETR	: Calcule le périmètre de la forme à l'aide du suivi de contour.
SURFACI	: Calcule la surface de la forme à l'aide du suivi de contour.
SURFAC2	: Calcule la surface en comptant le nombre de pixels à un.
COMPACIT	: Calcule le coefficient de compacité.
EROSTOT	: Calcule le nombre d'érosions nécessaire à la disparition de la forme.
MOMENTS	: Calcule les 7 moments invariants de HU.**
CHANGDIR	: Calcule le nombre de changements d'orientation du contour.
VARIANCE	: Calcule le rayon moyen et la variance du rayon.

RECCIR : Recherche le rectangle circonscrit , calcule sa longueur, sa largeur, sa surface et les pourcentages de symétrie par rapport aux axes.

DIFRAY : Recherche le rayon minimum et le rayon maximum et calcule la différence.

*** Nous n'avons conservé que les trois premiers moments, car les plus significatifs lors d'essais sur différentes images.*

Correspondance entre les numéros de paramètres et les sous-programmes

Cette liste sera utile pour interpréter les résultats de la sélection de paramètres, en particulier lors d'une corrélation assez forte entre deux paramètres.

N° du paramètre		Nature du paramètre	Nom du sous-programme
Paramètre n °1	:	Coefficient de compacité	Compacit
Paramètre n °2	:	Surface (FREEMAN)	Surfac1
Paramètre n °3	:	périmètre	Périmetr
Paramètre n °4	:	abscisse du centre de gravité	Centre
Paramètre n° 5	:	ordonnée du centre de gravité	Centre
Paramètre n °6	:	nombre de changements de direction	Changdir
Paramètre n °7	:	surface (comptage de un)	Surfac2
Paramètre n °8	:	différence entre le rayon maximum et le rayon minimum	Difray
Paramètre n °9	:	rayon moyen	Variance
Paramètre n °10	:	variance du rayon	Variance
Paramètre n °11	:	nombre d'érosions nécessaires à la disparition de la forme	Erostot
Paramètre n °12	:	largeur du rectangle circonscrit	Reccir
Paramètre n °13	:	longueur du rectangle circonscrit	Reccir
Paramètre n °14	:	surface du rectangle circonscrit	Reccir
Paramètre n °15	:	pourcentage de symétrie par rapport à l'axe des abscisses	Reccir
Paramètre n °16	:	pourcentage de symétrie par rapport à l'axe des ordonnées	Reccir
Paramètre n °17	:	premier moment invariant de HU	Moments
Paramètre n °18	:	deuxième moment invariant de HU	Moments
Paramètre n °19	:	troisième moment invariant de HU	Moments

II.1.5) Sous-programmes de normalisation

Les éléments du tableau PARAIMA sont donc les valeurs des différentes mesures des paramètres pour un lot donné de NI images. (fig 5)

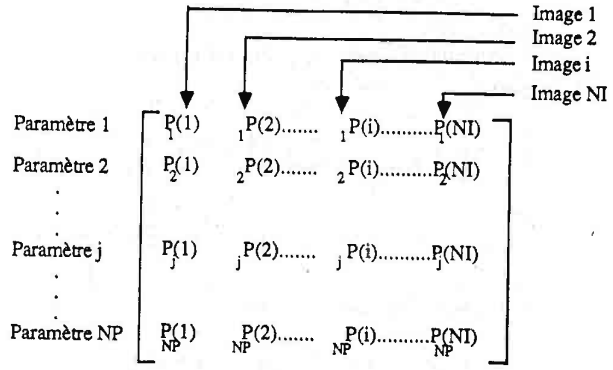


fig 5 : Tableau PARAIMA

Le tableau 2 donne un exemple de résultats du programme de calcul des différents paramètres. Le lot considéré ici se compose de dix images (trois cercles, trois carrés, trois triangles et une croix), pour lesquelles on a calculé les 19 paramètres précédemment cités. Certains paramètres prennent des valeurs très grandes d'autres de très petites. Pour éviter que les paramètres à grandes valeurs prédominent, lors de la sélection de paramètres, nous avons normalisé les valeurs de ce tableau. Pour ceci, nous avons utilisé deux méthodes.

Tableau 2

ima 1	ima 2	ima 3	ima 4	ima 5	ima 6	ima 7	ima 8	ima9	ima10
2,40	2,40	2,17	2,55	2,55	2,55	3,71	3,71	3,71	5,27
288	288	266	400	361	441	256	225	289	418
65,94	65,94	60,28	80,00	76,00	84,00	77,25	72,43	82,08	117,66
20	19	20	20	20,5	19,5	22	21,25	22,75	20
20	19	20	20	20,5	19,5	20	20	20	20
36	36	24	4	4	4	3	3	3	16
317	317	293	441	400	484	289	256	324	477
0,94	0,94	0,54	4,14	3,92	4,34	12,49	11,71	13,27	9,12
9,57	9,57	9,2	11,48	10,91	12,06	10,08	9,45	10,71	11,88
0,12	0,12	0,04	1,62	1,47	1,78	10,66	9,38	12,02	9,76
8	8	7	11	10	11	6	6	6	5
20	20	18	20	19	21	16	15	17	30
20	20	18	20	19	21	32	30	34	30

400	400	324	400	361	441	512	450	578	900
50	50	50	50	50	50	75	75	75	50
50	50	50	50	50	50	50	50	50	50
0,17	0,17	0,18	0,17	0,17	0,17	0,44	0,44	0,44	0,33
0,03	0,03	0,03	0,03	0,03	0,03	0,25	0,25	0,25	0,12
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Première méthode

Nous avons normalisé les valeurs que peut prendre un paramètre en calculant le pourcentage par rapport à la valeur maximale :

$$P_j(i)^* = \left(\frac{P_j(i)}{P_{jmax}} \right).100$$

Les résultats de la normalisation du tableau précédent sont donnés dans le tableau 3.

Exécution du programme Norma2 (pourcentage)

paramètre n°	valeur maximale du paramètre
P1	5.27081
P2	441
P3	117.6569
P4	22.75
P5	20.5
P6	36
P7	484
P8	13.2732
P9	12.05745
P10	12.01871
P11	11
P12	30
P13	34
P14	900
P15	75
P16	50
P17	0.44349
P18	0.2459177
P19	5.791165E-05

Tableau 3

ima 1	ima 2	ima 3	ima 4	ima 5	ima 6	ima 7	ima 8	ima 9	ima10
46	46	41	48	48	48	70	70	70	100
65	65	60	91	82	100	58	51	66	95
56	56	51	68	65	71	66	62	70	100
88	84	88	88	90	86	97	93	100	88
98	93	98	98	100	95	98	98	98	98
100	100	67	11	11	11	8	8	8	44
65	65	61	91	83	100	60	53	67	99
7	7	4	31	30	33	94	88	100	69
79	79	76	95	90	100	84	78	89	99
1	1	0	13	12	15	89	78	100	81
73	73	64	100	91	100	55	55	55	45
67	67	60	67	63	70	53	50	57	100
59	59	53	59	56	62	94	88	100	88
44	44	36	44	40	49	57	50	64	100
67	67	67	67	67	67	100	100	100	67
100	100	100	100	100	100	100	100	100	100
39	39	40	37	37	38	100	100	100	75
13	13	13	11	11	11	100	100	100	51
2	2	3	0	0	0	89	100	79	8

On peut voir sur cet exemple que pour le dernier paramètre (le 19-ième) les valeurs brutes sont de l'ordre de 10^{-5}; il serait complètement "écrasé" par les autres. Alors que normalisé ce paramètre prend des valeurs très intéressantes. En effet, il discrimine bien les différentes classes (3 premières formes : cercle ; 3 suivantes :carré; 3 suivantes : triangle; et la dernière : croix)

Deuxième méthode :

La deuxième méthode consiste à normaliser les valeurs des paramètres avec une moyenne nulle et une variance de 1. Les nouvelles valeurs sont calculées de la manière suivante:

$$P_j(i)^* = \frac{P_j(i) - P_j}{\sigma_j}$$

$P_j(i)$: représente l'ancienne valeur du paramètre j pour la i-ième image

$P_j(i)^*$: représente la nouvelle valeur du paramètre j pour la i-ième image

P_j : représente la valeur moyenne du paramètre j

σ_j : représente l'écart-type du paramètre j

Le tableau 4 donne les valeurs des paramètres normalisés à l'aide de la deuxième méthode.

paramètre n°	valeur moyenne du paramètre	variance du paramètre
P1	3,1021	0,8578
P2	323,2	5126,96
P3	78,1588	226,9389
P4	20,5	1,2125
P5	19,9	0,14
P6	13,3	173,01
P7	359,8	6256,56
P8	6,14314	22,86421
P9	10,4919	1,01778
P10	4,695	22,86
P11	7,8	4,36
P12	19,6	15,44
P13	24,4	35,24
P14	476,6	24703,04
P15	57,5	131,25
P16	50	0
P17	0,2687	1,528E-02
P18	0,10385	9,414E-03
P19	1,6421E-05	5,419E-10

Tableau 4

ima 1	ima 2	ima 3	ima 4	ima 5	ima 6	ima 7	ima 8	ima 9	ima10
-0,76	-0,76	-1	-0,6	-0,6	0,66	0,66	0,66	-0,66	2,34
-0,49	-0,49	-0,8	1,07	0,53	1,65	-0,94	-1,37	-0,48	1,38
-0,81	-0,81	-1,19	0,12	-0,14	0,39	-0,06	-0,38	0,26	2,62
-0,45	-1,36	-0,45	-0,45	0,00	-0,91	1,36	0,68	2,04	-0,45
0,27	-2,41	0,27	0,27	1,60	-1,07	0,27	0,27	0,27	0,27
1,73	1,73	0,81	-0,71	-0,71	-0,71	-0,78	-0,78	-0,78	0,21
-0,54	-0,54	-0,84	1,03	0,51	1,57	-0,90	-1,31	-0,45	1,48
-1,09	-1,09	-1,17	-0,42	-0,46	-0,38	1,33	1,16	1,49	0,62
-0,91	-0,91	-1,27	0,98	0,41	1,55	-0,41	-1,03	0,21	1,37
-0,96	-0,96	-0,97	-0,64	-0,68	-0,61	1,25	0,98	1,53	1,06
0,10	0,10	-0,38	1,53	1,05	1,53	-0,86	-0,86	-0,86	-1,34
0,10	0,10	-0,41	0,10	-0,15	0,36	-0,92	-1,17	-0,66	2,65
-0,74	-0,74	-1,08	-0,74	-0,91	-0,57	1,28	0,94	1,62	0,94
-0,49	-0,49	-0,97	-0,49	-0,74	-0,23	-0,23	-0,17	0,65	2,69
-0,65	-0,65	-0,65	-0,65	-0,65	-0,65	1,53	1,53	1,53	-0,65
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
-0,76	-0,76	-0,74	-0,83	-0,83	-0,83	1,41	1,41	1,41	0,50
-0,75	-0,75	-0,74	-0,79	-0,79	-0,79	1,46	1,46	1,46	0,22
-0,64	-0,64	-0,63	-0,71	-0,71	-0,71	1,50	1,78	1,26	-0,52

II.1.6) Différents sous-programmes de sélection de paramètres
II.1.6.a) Critère utilisé pour les trois méthodes.

Le critère que nous avons retenu pour notre expérimentation est un critère qui tient à la fois compte de la distance interclasse et de la distance intraclasse. Il semble évident qu'une classification ou une reconnaissance sera d'autant plus facile que la distance interclasse est grande et que la distance intraclasse est faible. En effet, nous allons rechercher les paramètres qui nous permettront de bien séparer les différentes classes (distances interclasses grandes), et pour lesquels les formes d'une classe sont très proches de leur représentant, dans notre cas le centre de gravité (distances intraclasses faibles). Donc, le critère doit être proportionnel à la distance interclasse et inversement proportionnel à la distance intraclasse. Nous avons donc choisi de calculer le critère de la façon suivante :

critère = d(inter) / d1 + d2 / d(intra) où d1 et d2 sont des valeurs constantes de distances (par exemple: distance maximale ou minimale des distances inter ou intraclasses).

II.1.6.b) Méthode Cheung

La procédure utilisée est une procédure à plusieurs étapes. A la première étape, on sélectionne des sous-ensembles de deux paramètres, à la deuxième étape des sous-ensembles de trois paramètres, jusqu'à la (k-1)-ième étape où l'on sélectionne des sous-ensembles de k paramètres. A la dernière étape, parmi tous les sous-ensembles de paramètres, on retiendra comme solution optimale le sous-ensemble correspondant à la plus grande valeur du critère considéré.

Cette méthode suppose connus au départ le nombre de classes, ainsi que la répartition des formes du lot dans les différentes classes.

II.1.6.c) Méthode du "Branch and Bound"

La méthode a été développée en détail dans le chapitre II.

II.1.6.d) Méthode de Foroutan et Sklansky

Le sous-programme que nous avons écrit est basé sur la méthode de Foroutan et Sklansky, que nous avons développée dans le chapitre II. Nous y avons apporté quelques modifications. En effet, la méthode de Foroutan et Sklansky ne fixait pas le nombre de

paramètres à sélectionner. Le principe était de supprimer un paramètre à chaque étape, tant que la valeur du critère restait supérieure à un certain seuil fixé par l'opérateur, seuil qu'il est relativement difficile d'apprécier.

Dans notre cas, le nombre de paramètres à sélectionner est fixé dès le départ. Le principe de notre méthode est de supprimer à chaque étape un paramètre. Le choix de ce paramètre est réalisé de la façon suivante : au départ, on dispose de NP paramètres comme dans le cas précédent. On appelera $E_1(1)$ le sous-ensemble constitué des NP paramètres privé du paramètre n°1, et d'une manière plus générale $E_1(j)$ le sous-ensemble des NP paramètres privé du paramètre n°j. Pour chacun de ces sous-ensembles, on calculera la valeur de la fonction critère, qui caractérisera la qualité du sous-ensemble de paramètres restants après élimination du paramètre. Si à la première étape, $E_1(k)$ est le sous-ensemble correspondant à la plus grande valeur du critère, le processus supprime définitivement le paramètre n°k. Le processus est réitéré jusqu'à l'obtention d'un sous-ensemble de NK paramètres (NK étant le nombre de paramètres à sélectionner). L'organigramme de la méthode est donné fig.6.

Supposons que l'on dispose de 19 paramètres, et que l'on souhaite sélectionner 10 paramètres, il faudra donc supprimer 9 paramètres; le processus sera donc constitué de 8 étapes. A la première étape, le nombre de sous-ensembles à tester est donc 19, à la deuxième 18 .....à la j-ième étape 19-(j-1).....à la dixième étape :11 sous-ensembles. Ce qui nous donne un nombre total de sous-ensembles à tester de 19+18+17+.....11=135.

Le but est donc de trouver le sous-ensemble de NK paramètres pour lequel le critère prend la plus grande valeur.

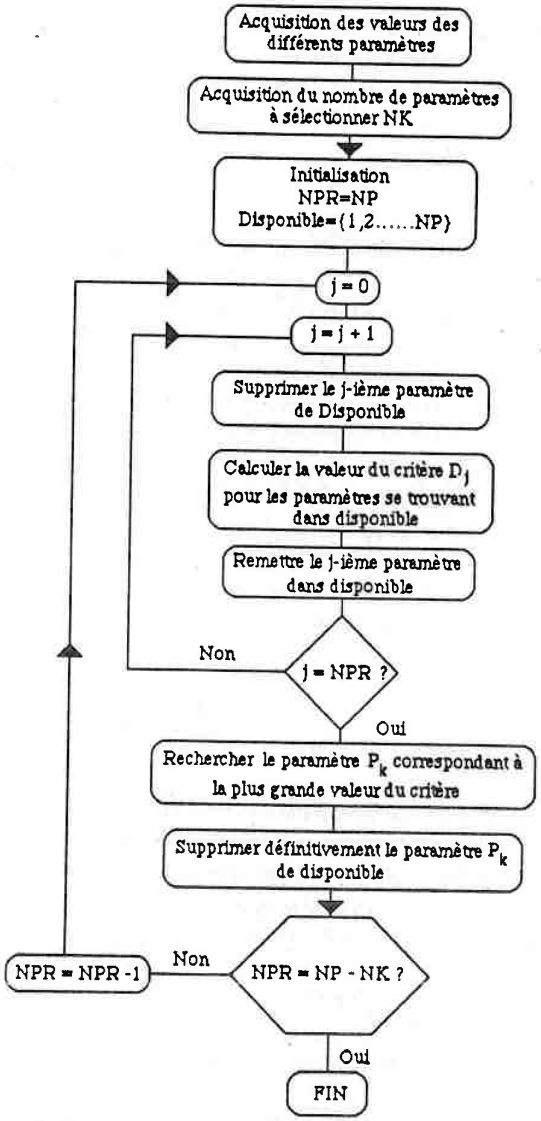


fig 6 : Organigramme du sous-programme Foroutan

NPR: représente le nombre de paramètres contenus dans "disponible", à l'étape en cours.

NP : nombre de paramètres initiaux.
NK : nombre de paramètres à sélectionner.
Disponible () : tableau contenant les numéros des paramètres restants.

Remarque :

Pour ces différentes méthodes de sélection de paramètres basées sur la programmation dynamique, le critère est supposé être une fonction monotone du nombre de paramètres, ce qui évite de calculer toutes les combinaisons possibles. En fait, au premier abord, il est presque "normal" de penser qu'un sous-ensemble de paramètres donne une meilleure classification ou reconnaissance qu'un sous-ensemble plus petit contenu dans le premier. Mais cette condition n'est vraie que si au départ l'utilisateur a lui-même inconsciemment déjà réalisé une sélection de paramètres (le choix des paramètres initiaux), supprimant ainsi les paramètres "nuisibles". Par exemple, pour faire de la classification de formes, l'utilisateur va inconsciemment supprimer ou ne pas prendre le paramètre couleur, il choisira des paramètres du type surface périmètre..... Il réalise donc ce que l'on pourrait appeler une première sélection de paramètres. On peut donc dire que les méthodes travaillant sur des paramètres quelconques ne nous fournissent pas toujours une solution optimale, mais très proche. Les résultats obtenus sont dans de nombreux cas très satisfaisants.

II.2) Les différents fichiers

Pour le passage d'un grand nombre d'informations d'un programme à un autre, nous avons choisi de créer un certain nombre de fichiers. Leur rôle est fondamental pour l'enchaînement des différents programmes et sous-programmes, c'est pour cette raison que nous leur avons consacré ce paragraphe. Nous en donnons dans ce qui suit une présentation détaillée sous forme graphique.

- Constitution du fichier RESUL

Il contient trois grandeurs

* le nombre de paramètres sélectionnés NK,
* le nombre initial de paramètres NP,
* le nombre de classes NR

NK	NP	NR

- Constitution du fichier PARAIMA (*PARANORM*)

Il contient

* le nombre d'images dans le lot : NI
* les valeurs de tous les paramètres (*normalisés*) pour toutes les images.

NI	$P_1(1)$	$P_1(2)$	.....	$P_1(NI)$	$P_2(1)$	$P_2(2)$	.....	$P_2(NI)$	.....	$P_{NP}(1)$	$P_{NP}(2)$	.....	$P_{NP}(NI)$

P_j (i) représente la valeur du j-ième paramètre (*normalisé*) pour la i-ième image.

- Constitution du fichier PARTITION

Il contient les partitions initiales.

- Constitution du fichier CENGRA :

Il contient les coordonnées du centre de gravité de chaque classe.

al(1,1)	al(1,2)	.....	al(1,np) ,	al(2,1)	al(2,2)	.....	al(2,np).....
.....	al(nrr,1)	al(nrr,2)	.....	al(nrr,np)			

al(i,j) : j-ième coordonnée du représentant (centre de gravité) de la i-ième classe.

- Constitution du fichier " NORM.RES" :

La configuration de ce fichier dépend du type de normalisation que l'opérateur a choisi:

Si la normalisation est réalisée à l'aide de la moyenne et de l'écart-type, le fichier est constitué de la manière suivante :

CodNor	M(1)	M(2).....	M(np)	V(1)	V(2).....	V(np)

CodNor : Grandeur pouvant prendre deux valeurs :

* 1 : correspondant au premier type de normalisation
* 2 : correspondant au deuxième type de normalisation

M(j) : moyenne du j-ième paramètre, calculée pour la normalisation
V(j) : Variance du j-ième paramètre

Si maintenant la normalisation est réalisée à l'aide du pourcentage, le fichier est constitué de la manière suivante :

CodNor Max(1) Max(2) ..... Max(np)

Max(j): Valeur maximale prise par le j-ième paramètre pour la normalisation (%)

- Constitution du fichier SELECT

Il contient les numéros ordonnés par ordre croissant des paramètres sélectionnés.

P(1) P(2) .....P(nkk)

P(k): Numéro du k-ième paramètre sélectionné

III) CLASSIFICATION

III.1) Introduction

Ce troisième chapitre a été écrit dans le but de vérifier que les résultats obtenus lors de la sélection de paramètres sont cohérents. En effet, avant de passer à la dernière étape qui consiste à attribuer une forme à l'une des classes préétablies durant la première phase, il nous a paru intéressant de voir si en ne considérant que les paramètres sélectionnés, les classes obtenues à l'aide d'une méthode de classification correspondaient bien aux mêmes partitions, aux mêmes classes qu'à l'étape initiale.

Nous avons vu qu'il existait deux sortes de méthode de classification (les méthodes de réallocation et les méthodes hiérarchiques). Pour notre expérimentation nous avons donc pris une méthode de chaque groupe.

III.2) Description du programme de Classification: CLASSI

III.2.1) Description générale du programme

Ce programme est essentiellement constitué de deux sous-programmes de classification:

- sous-programme de classification par la méthode des K-means: K-M
- sous-programme de classification par la méthode de splitting : SPLI

Ces deux sous-programmes seront détaillés dans les chapitres suivants.

L'organigramme du programme est donné fig.1. La seule question que le programme pose à l'utilisateur est le choix de la méthode de classification. En effet, toutes les variables nécessaires à son exécution ont été rangées durant les phases précédentes dans les fichiers suivants:

- Fichier **RESUL** : contient trois grandeurs:
  - * le nombre de classes,
  - * le nombre de paramètres sélectionnés,
  - * le nombre initial de paramètres.
- Fichier **PARANORM**: contient
  - * le nombre d'images dans le lot,
  - * les valeurs de tous les paramètres normalisés pour toutes les images.
- Fichier **CENGRA**: contient les coordonnées du centre de gravité de chaque classe.
- Fichier **PARTITION** : contient les partitions initiales.
- Fichier **SELECT** : contient les numéros ordonnés par ordre croissant des paramètres sélectionnés.

Ces fichiers ne sont pas ouverts dans n'importe quel ordre. Par exemple, on ne peut ouvrir le fichier CENGRA que si on connaît le nombre de classes, donc que si on a déjà ouvert le fichier RESUL.

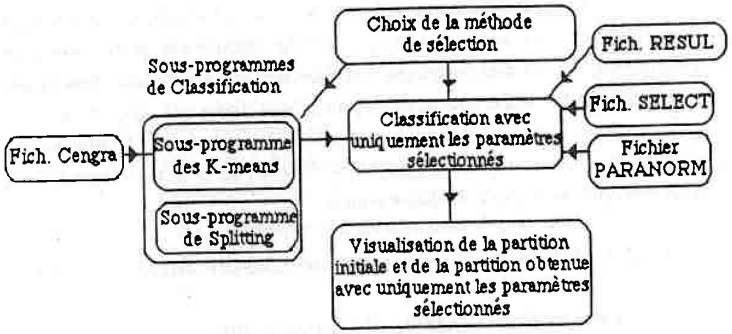


fig 1 : Organigramme du programme de classification

III.2.2) Méthode des K-means :K-M

Cette méthode fait partie des méthodes de réallocation, que nous avons développées dans le chapitre II [LIND-80], [SHIR-86].

Principe de la méthode:

Soit $X=\{X_1,X_2,.....X_{NI}\}$ l'ensemble des formes d'apprentissage. Chaque point est représenté par un vecteur de dimension NP (NP paramètres). On souhaite répartir les NI formes dans NR classes. Chaque classe sera représentée par son centre de gravité, et l'ensemble de tous les centres de gravité constitueront l'alphabet de reproduction.

$$\underline{A}=\{y_1,y_2.....y_{NR}\}.$$

$\underline{A}_0$ est l'alphabet de reproduction initial, choisi de façon aléatoire. (Le choix de $\underline{A}_0$ a été discuté dans le chapitre II)). Pour chaque point forme, on calcule la distance par rapport à tous les centres de gravité et on attribue la forme à la classe pour laquelle cette distance est la plus faible (distance euclidienne par exemple). Lorsque tous les points sont affectés à une classe, on obtient la première partition $S(1)=\{S_1,S_2.....S_{NR}\}$. On recalcule alors pour chaque classe le nouveau centre de gravité. Ces nouveaux centres constitueront alors le nouvel alphabet de reproduction $\underline{A}_i$. A chaque étape la distorsion totale est

donnée. Cette dernière est obtenue simplement en calculant la moyenne des distances entre chaque forme de la série d'apprentissage et du centre de gravité de la classe à laquelle elle appartient. L'opération est renouvelée tant que la distorsion totale décroît rapidement. La partition s'affine au cours de ces différentes étapes. Le test d'arrêt est simple:

- l'utilisateur fixe un seuil de distorsion E,
- dès que la différence entre les distorsions totales
  - * de l'étape en cours
  - * et de l'étape précédente

divisée par la distorsion totale de l'étape en cours est inférieure au seuil, le programme s'arrête et conserve la dernière partition obtenue.

Algorithme des K-means :

Etape 0 Initialisation
NR : nombre de classes
E seuil de distorsion
$\underline{A}_0$ alphabet initial de reproduction
$X=\{x_j ; j=1 .....NI\}$ série d'apprentissage
m=0
$D_{-1}= +\infty$ Distorsion totale initiale

Etape 1 $\underline{A}_m=\{ y_i ; i=1.....NR \}$
Trouver à partir de $\underline{A}_m$, la partition qui possède la distorsion minimale $P(\underline{A}_m)=\{S_i$ pour i=1 à NR$\}$ de la série d'apprentissage.
x_j appartient à S_i si $d(x_j,y_i) \leq d(x_j,y_k)$ pour tout k=1 à NR
Calculer la distorsion totale:

$$D_m = D(\{\underline{A}_m , P(\underline{A}_m)\}) = \frac{1}{n} \sum_{j=0}^{n-1} [\min d(x_j,y)]_{y\in \underline{A}_m}$$

Etape 2 Si $(D_{m-1} - D_m) / D_m \leq E$
Prendre $\underline{A}_m$ comme alphabet de reproduction final.
Fin du programme
Sinon continuer.

Etape 3 Trouver l'alphabet de reproduction optimal
Pour chaque classe S_i calculer le nouveau centre de gravité y_i
m=m+1
$\underline{A}_m = \{ y_i ; i=1 à NR \}$
aller à l'étape 1

L'inconvénient de cette méthode est le fait que la vitesse de convergence et la convergence de l'algorithme dépendent énormément du choix de l'alphabet initial.

Notamment lorsque deux classes sont très proches l'une de l'autre, la méthode est incapable de séparer les deux classes si l'alphabet initial ne donne pas un représentant pour chaque classe. Dans notre cas, nous avons choisi comme alphabet initial les centres de gravité des classes initiales privés des paramètres non sélectionnés. Nous avons résolu le problème des classes trop proches par une division de la classe la plus importante en fin de programme, dans le cas où l'une des classes ne contiendrait aucune forme.

C'est pour cette raison que les résultats obtenus par la deuxième méthode nous ont parus plus intéressants et plus exacts.

III.2.3) Méthode de splitting: sous-programme SPLI

La méthode de splitting est une méthode hiérarchique. Le processus utilisé est très simple. L'algorithme (en ne conservant que les paramètres sélectionnés) calcule le centre de gravité, de toutes les formes du lot. Puis il introduit une petite perturbation, de façon à obtenir deux nouveaux centres de gravité. Il calcule alors pour chaque forme la distance euclidienne de cette forme à ces centres de gravité et attribue la forme à la classe la plus proche. On obtient ainsi deux partitions dont on recalcule les nouveaux centres. Ces centres sont perturbés et chaque classe est divisée en deux (d'où le nom de splitting). Le processus est répété jusqu'à obtention du bon nombre de classes. La méthode ne peut donc être utilisée que pour un nombre de classes égal à 2,4,8,16...2^n. En fait, on peut contourner le problème très simplement si le nombre de classes n'est pas une puissance de deux. Il suffit de partager en un nombre de classes supérieur, puis de regrouper les classes les plus proches de manière à retrouver le bon nombre de classes. Ou encore, comme nous allons le voir, de ne diviser qu'une seule classe à la fois, celle dont la distorsion est la plus grande.

Principe de la méthode de Splitting:

1) Calcul du centre de gravité X_0 de l'ensemble d'apprentissage et définition d'un alphabet de reproduction initial.

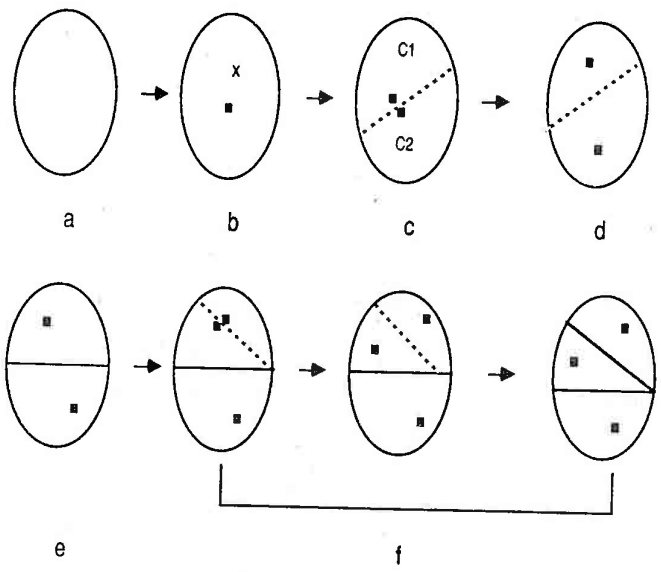
$D_n^0 = \{ X_0-x, X_0+x \}$ n=2 à partir d'une perturbation.

2) Rechercher l'alphabet optimal à partir de D_n^0. Une distorsion moyenne est obtenue pour chaque nouvelle classe.

3) Définition d'un nouvel alphabet D_{n+1}^0, à partir de D_n^0, en perturbant la classe de distorsion maximale.

4) itération de 2) et 3) jusqu'à l'obtention de la taille finale du dictionnaire recherché.

Exemple de la méthode de splitting:



a) ensemble d'apprentissage.
b) X: centre de gravité.
c) Perturbation autour de X qui partage l'espace en deux classes C_1 et C_2.
d) Calcul du nouveau centre de gravité de chaque classe, D_n.
e) Partage de l'ensemble en deux sous-ensembles.
f) Refaire c) d) et e) pour le sous-ensemble dont la distorsion est la plus élevée.
g) répéter f) jusqu'à l'obtention de l'alphabet final.

Dans le chapitre suivant, nous présentons les différents résultats obtenus avec ces deux méthodes, en fonction du nombre de paramètres sélectionnés.
Nous signalons que bien qu'utilisant quelques fichiers de résultats du programme principal, le programme de classification est indépendant de ce dernier. Il n'a été réalisé que dans le but de vérification.

IV) RECONNAISSANCE DE FORME

Pour tester l'efficacité de nos programmes de sélection de paramètres, nous avons créé un programme de reconnaissance de forme, basé sur les résultats de l'apprentissage.

En effet, lors de l'apprentissage, le programme a recherché les paramètres les plus pertinents pour un lot d'images. Il nous a aussi fourni les représentants de chaque classe possible pour ce lot. Ces résultats ont alors été classés dans différents fichiers.

IV.1) But du programme

On présente une forme quelconque et le programme de reconnaissance, doit en utilisant uniquement les paramètres sélectionnés, attribuer cette forme à l'une des classes possibles. L'expérience est répétée un grand nombre de fois. On pourra ainsi, pour chacune des configurations possibles du programme de sélection, vérifier le pourcentage de bonne reconnaissance.

IV.2) Principe de l'algorithme

Le programme demande à l'opérateur le nom de l'image à classer. Puis comme pour le programme initial il demande à l'opérateur s'il souhaite bruiter ou non l'image. Dans le cas affirmatif, le sous-programme Rando est appelé et demande à l'opérateur le pourcentage et la profondeur de bruitage souhaité. Les sous-programmes de suivi de contour et de centre de gravité sont systématiquement appelés. Le programme ouvre plusieurs fichiers de résultats mentionnés précédemment. Le fichier RESUL contient les numéros des paramètres sélectionnés. Après lecture de ce dernier, le programme appelle uniquement les sous-programmes correspondant aux paramètres sélectionnés, et range les résultats dans un tableau appelé PARAIMA. Les valeurs sont normalisées à l'aide des grandeurs mémorisées dans le fichier PARANORM au cours de la phase d'apprentissage.

Cette étape franchie, la phase de reconnaissance peut commencer.Comme nous l'avons déjà signalé, le fichier CENGRA contient les centres de gravité de chacune des classes. Le programme calcule alors la distance (euclidienne) entre la forme considérée et les différents centres de gravité, et attribue la forme à la classe la plus proche.

Le programme demande alors à l'opérateur s'il souhaite classer une autre forme.

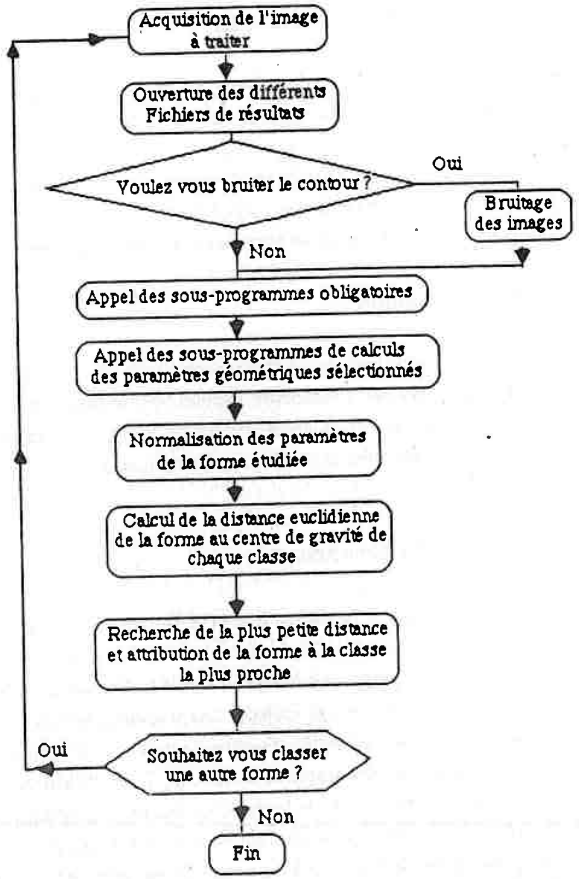
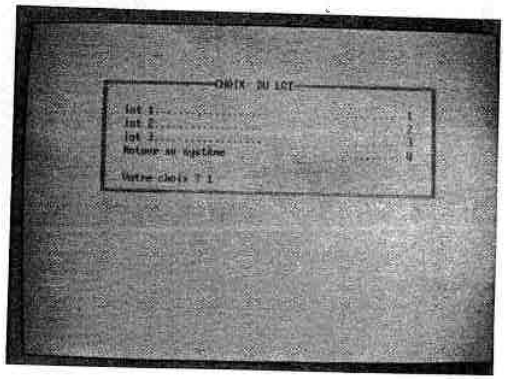


fig 1: Organigramme du programme de reconnaissance de forme.

EXEMPLE D'EXECUTION

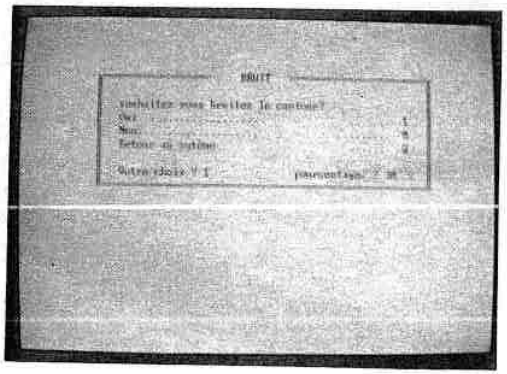
CHOIX DU LOT

Photo 1 : Choix du lot
lot 1
classe 1 : 3 cercles
classe 2 : 3 carrés
classe 3 : 3 triangles
classe 4 : 1 croix



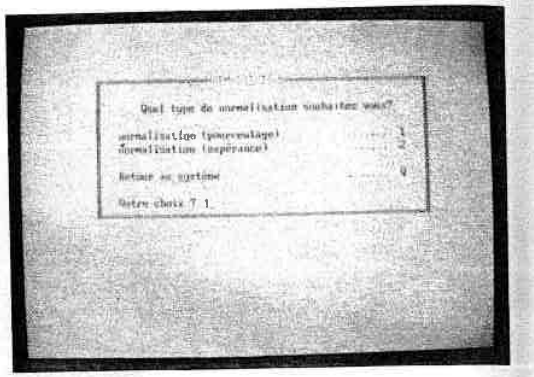
BRUITAGE

Photo 2 : Choix du pourcentage de bruit



NORMALISATION

Photo 3 :
choix de la méthode de normalisation



SELECTION DE PARAMETRE

Photo 4 :
choix de la méthode de sélection

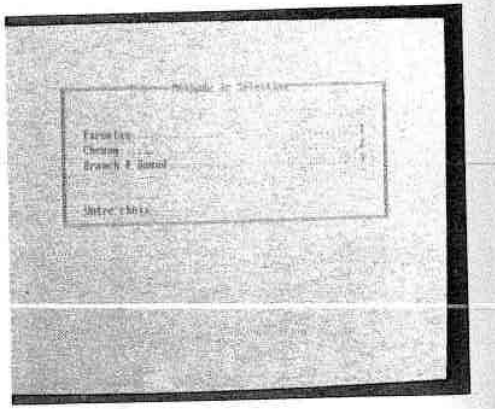


Photo 5 :
choix du nombre de paramètres à sélectionnés

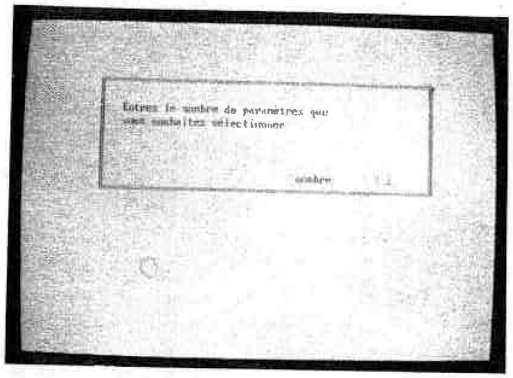
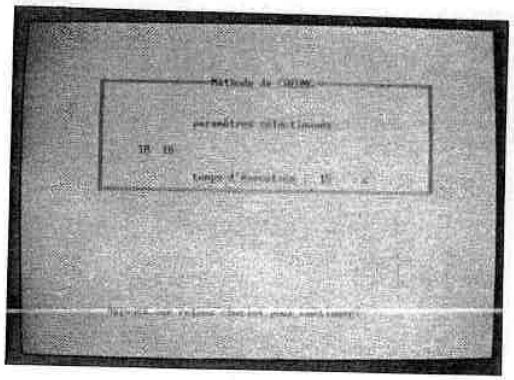


Photo 6 :
Affichage des numéros des paramètres sélectionnés



CLASSIFICATION

Photo 7 :
choix de la méthode de classification (Splitting)

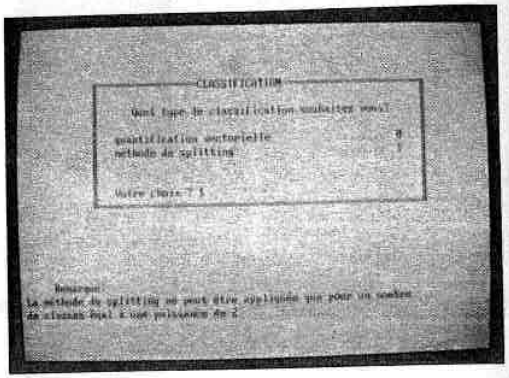
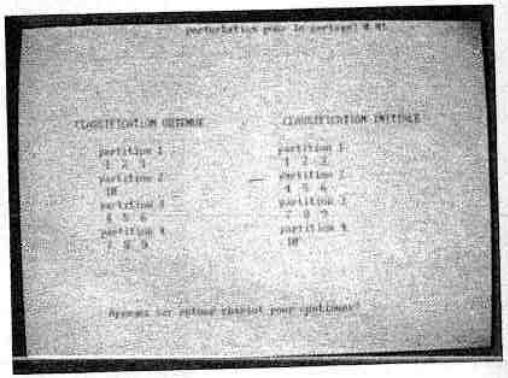


Photo 8 :
Affichage de la partition obtenue avec ces 2 paramètres



RECONNAISSANCE DE FORME

Photo 9 :
choix de l'image à reconnaître (carré)

Photo 10 :
choix du pourcentage de bruit

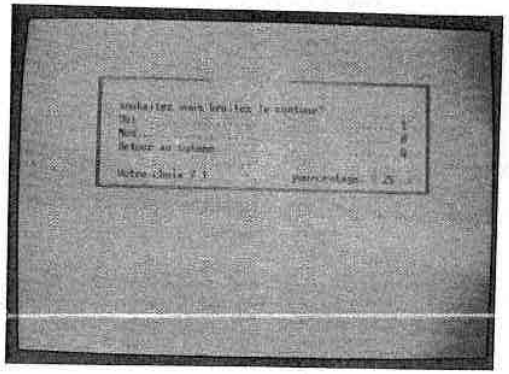
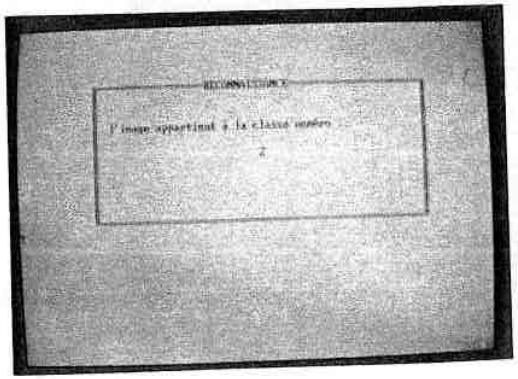


Photo 11 :
Affichage du numéro de la classe (2: classe des carrés)



V) DISCUSSION DES RESULTATS

V.1) Introduction

En raison du nombre important de combinaisons possibles de traitements, il est impossible de présenter et de discuter les résultats obtenus pour toutes les combinaisons. Nous présentons ci-dessous d'une manière très synthétique les différentes possibilités de notre programme.

Différentes possibilités du programme:

- Choix du lot ..... 2 lots
- Pourcentage de bruit..... de 0 à 100%
- Profondeur du bruit..... de 0 à 3
- Type de normalisation..... - pourcentage
  - moyenne et écart type
- Nombre de paramètres à sélectionner..... - de 1 à 18
- Méthode utilisée pour la sélection..... - Foroutan & Sklansky
  - Cheung
  - Branch and Bound
- Choix de la méthode de classification..... -Quantification vectorielle
  - Méthode de Splitting
- Choix des images sur lesquelles on va opérer la reconnaissance
  - lot 1 : 10 images
  - lot 2 : 11 images
- Pourcentage de bruit pour l'image à reconnaître.. de 0 à 100%
- Profondeur du bruit pour l'image à reconnaître.. de 0 à 3

Nous nous sommes efforcés de présenter dans ce qui suit les résultats qui nous ont parus les plus intéressants (d'autres tableaux de résultats sont donnés en annexe C).

V.2) Résultats obtenus au niveau de la sélection

V.2.1) Les paramètres

Notre but étant avant tout de comparer les méthodes de sélection de paramètres, nous présentons ci-dessous les résultats obtenus par ces trois méthodes appliquées dans les mêmes conditions.

Nombre de Paramètres ↓	Méthode →	Foroutan	Cheung	Branch & Bound
1	Numéros des paramètres	10	19	10
	Durée de la sélection	01min 55s	3s	3 min 51s
2	Numéros des paramètres	10-19	10-19	10-19
	Durée de la sélection	01min 28s	46s	5min 29s
4	Numéros des paramètres	6-10-18-19	6-10-18-19	6-10-8-19
	Durée de la sélection	1min 23s	2min 13s	24min 32s
6	Numéros des paramètres	6-8-10-14-18-19	6-8-10-14-18-19	6-8-10-14-18-19
	Durée de la sélection	1min 18s	3min 37s	2min 11s
8	Numéros des paramètres	1-6-8-10-14-17-18 19	1-6-8-10-14-17-18 19	1-6-8-10-14-17-18 19
	Durée de la sélection	1min 10s	5min	9min 04s
10	Numéros des paramètres	1-6-8-10-11-12-14 17-18-19	1-6-8-10-11-12-14 17-18-19	1-6-8-10-11-12-14 17-18-19
	Durée de la sélection	1min 01s	6min 14s	5min 06s
12	Numéros des paramètres	1-3-6-7-8-10-11-12 14-17-18-19	1-3-6-7-8-10-11-12 14-17-18-19	1-3-6-7-8-10-11-12 14-17-18-19
	Durée de la sélection	51s	7min 21s	1min 36s
14	Numéros des paramètres	-(16-5-4-9-15)	-(4-5-9-15-16)	-(16-5-4-9-15)
	Durée de la sélection	39s	8min 18s	17s
16	Numéros des paramètres	-(4-5-16)	-(4-5-16)	-(4-5-16)
	Durée de la sélection	26s	8min 58s	12s
18	Numéros des paramètres	-(16)	-(5)	-(5)
	Durée de la sélection	11s	9min 24s	10s

Tableau 1: Numéros des paramètres et durée de la sélection pour les trois méthodes en fonction du nombre de paramètres.

Le premier tableau a été obtenu à partir du lot n°1, non bruité, normalisé à l'aide de la méthode de pourcentage. Le critère de sélection utilisé est $D_B + 1/D_W$.

Les nombres de la colonne de gauche indiquent le nombre des paramètres sélectionnés. Pour chacune des méthodes, les numéros des paramètres sélectionnés, ainsi que le temps d'exécution de la méthode sont indiqués. *(Les numéros entre parenthèses et précédés d'un signe moins représentent les paramètres que l'on supprime pour obtenir le sous-ensemble optimal).*

A l'exception de la première et de la dernière lignes les paramètres sélectionnés sont les mêmes pour les trois méthodes. C'est un point positif car on comprendrait mal que la sélection dépende de la méthode.

Pour un paramètre, on s'aperçoit que les méthodes de "Foroutan and Sklansky" et du "Branch and Bound", nous ont fourni le paramètre n°10 (variance du rayon), alors que la méthode de Cheung nous a fourni le paramètre n°19 (troisième moment invariant de Hu). Pour 18 paramètres sélectionnés, la méthode de "Foroutan and Sklansky" supprime le paramètre 16, correspondant au pourcentage de symétrie par rapport à l'axe des x. Ce paramètre avait été choisi en raison de son pouvoir discriminatoire nul. En effet, ce paramètre prend la même valeur pour toutes les formes du lot n°1. Les deux autres méthodes ont supprimé le paramètre n°5, correspondant à l'ordonnée du centre de gravité.

V.2.2) Le temps d'exécution

Pour la méthode de Foroutan, le temps d'exécution est une fonction décroissante du nombre de paramètres sélectionnés ou encore une fonction croissante du nombre de paramètres supprimés. On peut aussi remarquer que pour un nombre donné de paramètres à sélectionner, le temps est pratiquement indépendant de la qualité des paramètres (très redondants ou pas). Il ne varie qu'en fonction du nombre d'images dans le lot. Ce qui n'est pas le cas de la méthode du Branch and Bound, où le temps dépend considérablement de la différence de qualité des paramètres, qui se traduit par un parcours plus ou moins long dans l'arbre. En effet, pour un même nombre de paramètres à sélectionner, le temps peut varier de 1min 19s à 1h 45min 19s voire plus, soit un rapport d'environ 60.

Pour la méthode de Cheung, le temps d'exécution est une fonction croissante du nombre de paramètres sélectionnés et comme pour la méthode de Foroutan, le temps ne dépend pratiquement pas de la qualité des paramètres. En fait, ceci s'explique par le fait que pour la méthode de Foroutan and Sklansky, ainsi que pour la méthode de

Cheung, le nombre de sous-ensembles à tester pour un nombre donné de paramètres à sélectionner est toujours le même.

Par exemple, supposons que l'on souhaite sélectionner 4 paramètres parmi 19.

- Pour la méthode de Foroutan and Sklansky, il faudra tester :

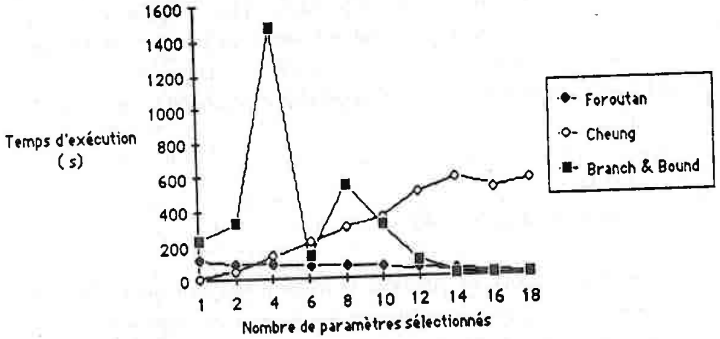
$19 + 18 + 17 + 16 \ldots\ldots\ldots + 5 = (24 \times 15):2 = 180$ sous-ensembles.

- Pour la méthode de Cheung, il faudra tester :

$(19\times18) + (19\times17) + (19\times16) + (19\times15) = 19\times(18 + 17 + 16 + 15) = 1254$ sous-ensembles

- Pour la méthode du "Branch and Bound" le nombre peut varier de 4 à C_{np}^{nk}.

Evolution du temps d'exécution des différentes méthodes en fonction du nombre de paramètres sélectionnés



V.2.3) Variation du critère

Nous avons développé dans le chapitre précédent le critère utilisé pour les différentes méthodes. Nous avions opté pour un critère proportionnel à la distance interclasse et inversement proportionnel à la distance intraclasse.

critère = inter + k / intra

Dans ce qui suit nous noterons D_W la distance intraclasse (de l'anglais Within) et D_B la distance interclasse (Between)

Le coefficient k est un moyen d'équilibrer, selon les lots, la contribution des deux grandeurs D_B et D_W. Nous présentons dans les tableaux ci-dessous les résultats obtenus en faisant varier ce coefficient.

Si le critère ne tient compte que de la valeur de D_B, les centres de gravité seront bien séparés, mais les formes d'une classe peuvent être dispersées autour de leur centre. Si par contre le critère ne tient compte que de la valeur de D_W, les formes seront bien regroupées autour des différents centres, mais le risque est d'obtenir des centres très proches, voire confondus dans certains cas. D'où l'importance du choix du coefficient k, qui nous permettra d'obtenir un compromis entre les effets opposés de l'utilisation de distances interclasses ou intraclasses.

Le tableau 2 donne les paramètres obtenus dans les conditions suivantes:

- méthode de Cheung
- lot n°2
- images du lot non bruitées
- valeurs des paramètres normalisées à l'aide de la méthode des pourcentages

On s'aperçoit que suivant la valeur donnée à la constante k du critère (1,10,100), les paramètres sélectionnés ne sont pas les mêmes dans tous les cas.

Si l'on considère le cas de deux paramètres sélectionnés pour k=1 et k=10, les paramètres sélectionnés sont les paramètres 6.10 (6: nombre de changements d'orientation, 10 variance du rayon), alors que pour k=10 les paramètres sont: 1.16 (1: cœfficient de compacité, 16: pourcentage de symétrie par rapport à l'axe des x).

Or le paramètre 16 est un paramètre qui prend la même valeur pour toutes les images, donc qui possède un D_B grand et un D_W très faible (nulle), en prenant k=100, on favorise D_W par rapport à D_B. Le paramètre 16 sera considéré comme un bon paramètre puisque les formes sont bien regroupées autour de leur centre de gravité; mais en fait, D_B, dont on ne tient pas entièrement compte, nous donne des centres très regroupés (ici confondus).

Le tableau 3 donne les paramètres sélectionnés en fonction de k, dans les conditions suivantes :

- méthode du Branch and Bound
- lot 2
- images du lot non bruitées
- valeurs des paramètres normalisées à l'aide de la méthode des pourcentages.

On constate encore une fois que les paramètres sélectionnés varient en fonction de la constante donnée à la variable k du critère. Si l'on considère cette fois le cas de 18

paramètres, on s'aperçoit que pour k=1, c'est le paramètre 16 qui est supprimé en raison de sa mauvaise valeur de D_B, alors que pour k=10 c'est le paramètre 2, et pour k=100 c'est le paramètre 1.
Pour k=10 ou k=100, on favorise D_W par rapport à D_B; 16 est alors considéré comme un bon paramètre.

On peut donc conclure que pour le lot 2, le bon "équilibre" entre la valeur de D_W et de D_B est obtenu pour la valeur k=1.

Remarquons aussi que le critère est surtout très sensible dans le cas où le nombre de paramètres sélectionnés est très grand ou très petit.

Nombre de Paramètres ↓	Critère →	inter + 1/intra	inter + 10/intra	inter + 100/intra
2	Numéros des paramètres	6-10	6-10	1-16
	Durée de la sélection	54s	53s	54s
4	Numéros des paramètres	6-8-10-11	6-8-10-11	1-6-11-16
	Durée de la sélection	2min 36s	2min 35s	2min 35s
6	Numéros des paramètres	6-8-10-11-18-19	6-8-10-11-18-19	6-8-10-11-18-19
	Durée de la sélection	4min 28s	4min 17s	4min 17s
8	Numéros des paramètres	6-7-8-10-11-17-18 19	6-7-8-10-11-17-18 19	1-6-7-8-10-11-18 19
	Durée de la sélection	5min 55s	5min 55s	5min 54s
10	Numéros des paramètres	1-6-7-8-10-11-13 17-18-19	1-6-7-8-10-11-13 17-18-19	1-6-7-8-10-11-16 17-18-19
	Durée de la sélection	7min 27s	7min 26s	7min 26s
12	Numéros des paramètres	- (2-3-4-5-9-12-16)	- (2-3-4-5-9-12-16)	- (2-3-5-9-12-13-14)
	Durée de la sélection	8min 48s	8min 47s	8min 47s
14	Numéros des paramètres	- (2-4-5-9-16)	- (2-3-5-9-16)	- (2-3-9-12-14)
	Durée de la sélection	9min 56s	9min 56s	9min 55s
16	Numéros des paramètres	- (2-9-16)	- (2-9-16)	- (2-9-4)
	Durée de la sélection	10min 48s	10min 47s	10min 56s
18	Numéros des paramètres	- (16)	- (2)	- (2)
	Durée de la sélection	11min 18s	11min 19s	11min 18s

Tableau 2: Numéros des paramètres et durée de la sélection pour la méthode de Cheung en fonction du critère choisi (lot 2).

Nombre de Paramètres ↓	Critère →	inter + 1/intra	inter + 10/intra	inter + 100/intra
2	Numéros des paramètres	6-10	6-10	6-11
	Durée de la sélection	12min 34s	12min 39s	5min 21s
4	Numéros des paramètres	6-8-10-11	6-8-10-11	6-8-10-11
	Durée de la sélection	13min 37s	12min 05s	5min 51s
6	Numéros des paramètres	6-8-10-11-18-19	6-8-10-11-18-19	6-8-10-11-18-19
	Durée de la sélection	4min 57s	3min 53s	1min 19s
8	Numéros des paramètres	6-7-8-10-11-17-18 19	6-7-8-10-11-17-18 19	6-7-8-10-11-17-18 19
	Durée de la sélection	1min 24s	1min 04s	40s
10	Numéros des paramètres	1-6-7-8-10-11-13 17-18-19	6-7-8-10-11-13-15 17-18-19	6-7-8-10-11-13-15 17-18-19
	Durée de la sélection	1min 27s	59s	30s
12	Numéros des paramètres	- (16-9-2-5-4-3-12)	- (2-9-16-5-3-4-12)	- (2-14-9-12-3-1-4)
	Durée de la sélection	24s	36s	24s
14	Numéros des paramètres	- (16-9-2-5-4)	- (2-9-16-3-5)	- (2-14-3-1-9)
	Durée de la sélection	32s	31s	18s
16	Numéros des paramètres	- (16-9-2)	- (16-9-2)	- (2-1-14)
	Durée de la sélection	16s	14s	14s
18	Numéros des paramètres	- (16)	- (2)	- (1)
	Durée de la sélection	12s	16s	16s

Tableau 3: Numéros des paramètres et durée de la sélection pour la méthode du Branch and Bound en fonction du critère choisi (lot 2).

V.2.4) Qualité de la sélection

Pour avoir un aperçu de la qualité des différentes méthodes dans le cas de deux paramètres sélectionnés, nous présentons ci-dessous quelques exemples de la distribution des formes dans un espace réduit.

La figure n°1 correspond à la représentation des formes du lot n°1, non bruité et à partir des paramètres 10 et 19 uniquement, paramètres sélectionnés par les trois méthodes.

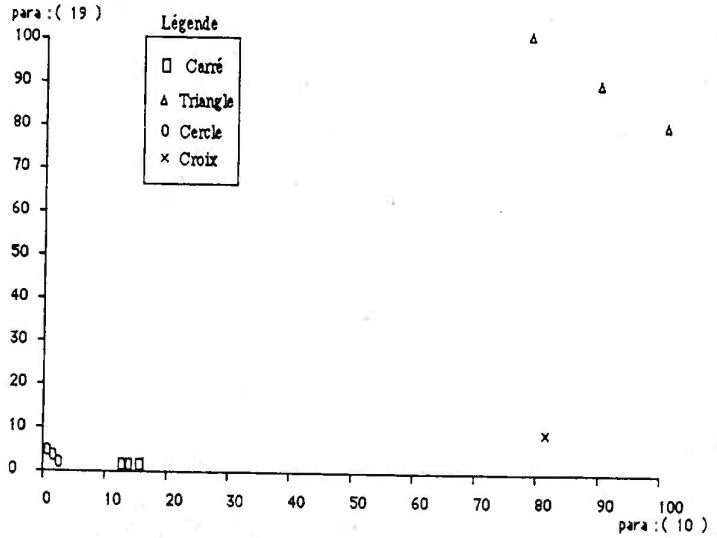


fig.1 Distribution des formes dans l'espace réduit aux paramètres (10;19)

On s'aperçoit que les formes sont bien regroupées (fig.1), mais que les classes cercle et carré sont très proches l'une de l'autre, ce qui peut prêter à confusion lors de la reconnaissance de forme. On s'aperçoit que les paramètres 6 et 10 (fig.2) donnent un espace réduit, où les classes sont mieux séparées mais où les formes ne sont pas très groupées. On peut constater dans le tableau 1 que le paramètre n°6 sera le prochain paramètre sélectionné. On peut déjà dire que les méthodes que nous avons utilisées ne sont pas des méthodes optimales, mais que les résultats obtenus sont relativement bons.

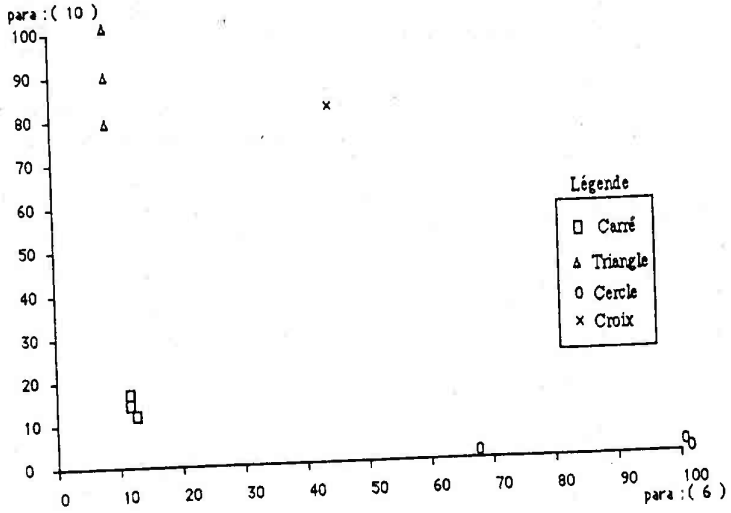


fig.2 : Exemples de paramètres pertinents

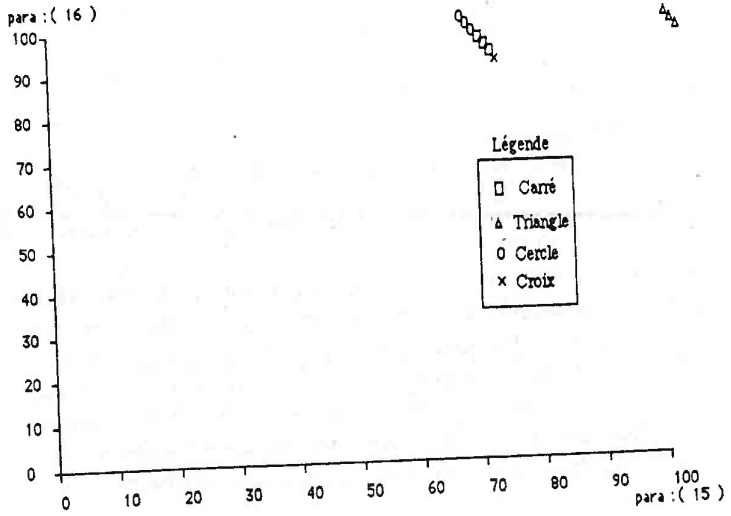


fig.3 : Exemples de paramètres non pertinents.

Pour le dernier cas (fig.3), on peut remarquer que les paramètres 15 et 16 (pris parmi les plus mauvais) donnent une représentation des formes, pratiquement inexploitable. Les formes de plusieurs classes sont regroupées dans une petite région, ce qui rend impossible toute classification ou reconnaissance.

V.3) Résultats obtenus au niveau de la classification

Comme nous l'avons déjà signalé, le but de la classification est de vérifier que la partition des formes obtenue avec uniquement les paramètres sélectionnés est bien la partition souhaitée. Ceci nous permettra de pouvoir comparer les différentes méthodes de sélection.

Les résultats obtenus sont très satisfaisants. En effet, nous avons recherché dans un premier temps, pour chacune des méthodes de sélection, le meilleur sous-ensemble de deux paramètres (Les résultats étant les mêmes pour un nombre de paramètres supérieur ou égal à 2). Pour le lot 1(non bruité initialement et normalisé à l'aide de la méthode des pourcentages), les trois méthodes nous ont fourni les mêmes paramètres:

- le paramètre n°10 correspondant à la variance du rayon,
- le paramètre n° 19 correspondant au troisième moment invariant de Hu.

En utilisant uniquement ces deux paramètres, nous obtenons la même partition que celle fixée initialement, quelque soit la méthode de classification utilisée (Quantification vectorielle ou méthode de Splitting).

Nous avons donc recommencé, en ne conservant cette fois qu'un seul paramètre. Pour la méthode de Foroutan, ainsi que pour la méthode du Branch and Bound, le paramètre retenu fut le paramètre n°10, alors que pour la méthode de Cheung, ce fut le paramètre n°19. Ceci s'explique par le fait que les deux premières méthodes suppriment les plus mauvais paramètres, alors que la méthode de Cheung recherche le meilleur sous-ensemble de paramètres. Les classifications obtenues avec ces trois méthodes sont données ci dessous.

Classification par la méthode de Quantification vectorielle:

Partition : initiale	Partition avec paramètre 10	Partition avec paramètre 19:
Partition 1 : 1 2 3	Partition 1 : 1 2 3	Partition 1 : 1 2 3
Partition 2 : 4 5 6	Partition 2 : 4 5 6	Partition 2 : 4 5 6
Partition 3 : 7 8 9	Partition 3 : 7 9	Partition 3 : 7 8 9
Partition 4 : 10	Partition 4 : 8 10	Partition 4 : 10

Classification par la méthode de Splitting:

Partition initiale	Partition avec paramètre 10:	Partition avec paramètre 19:
Partition 1 : 1 2 3	Partition 1 : 1 2 3	Partition 1 : 4 5 6
Partition 2 : 4 5 6	Partition 2 : 8 10	Partition 2 : 7 9
Partition 3 : 7 8 9	Partition 3 : 4 5 6	Partition 3 : 1 2 3 10
Partition 4 : 10	Partition 4 : 7 9	Partition 4 : 8

On peut remarquer que pour une classification par quantification vectorielle, le paramètre 19 (sélectionné par la méthode de Cheung) donne la même partition que l'initiale, alors que le paramètre 10 (sélectionné par les deux autres méthodes) a mal classé la forme 8, c'est-à-dire qu'il a classé un triangle dans la classe des croix. Pour la classification par Splitting, les deux paramètres ne redonnent pas les bonnes partitions. Si nous reprenons la figure 1, et que nous projetons les différents points formes sur les deux axes Ox et Oy correspondant respectivement au paramètre 10 et au paramètre 19 (fig.4), on peut remarquer que les différents points de projection se regroupent et nous fournissent la même partition que celle obtenue avec la méthode de Splitting. En effet il est clair que si l'on ne tient compte que du paramètre 10, la forme 10 (croix) sera assimilée à un triangle. Par contre si l'on ne tient compte que du paramètre 19, cette forme va être affectée à la classe des cercles. Si la méthode de quantification ne redonne pas la même partition, cela s'explique simplement par le fait qu'au cours des différentes étapes, la méthode parvient à une classe vide. Comme nous l'avons précisé précédemment, l'algorithme divise alors la plus grande classe.

Les résultats obtenus avec la méthode de Splitting sont plus corrects que ceux obtenus avec la méthode de quantification vectorielle.

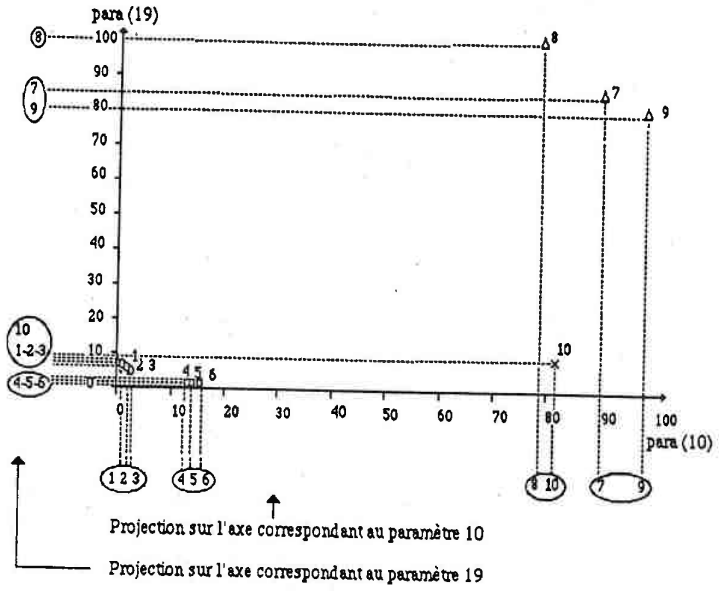


fig.4 : Projection des points formes sur les deux axes.

Choix du nombre de paramètres

Au niveau de la sélection de paramètres, c'est l'opérateur qui fait le choix du nombre de paramètres.

L'exemple précédent montre qu'en classification, le nombre de paramètres (fig.4) doit être au moins de deux.

Conclusion : Le nombre minimal de paramètres à sélectionner sera celui pour lequel la classification souhaitée sera vérifiée.

V.4) Résultats obtenus au niveau de la Reconnaissance de forme

V.4.1) Tableaux de résultats

Comme la classification, la reconnaissance est réalisée afin de pouvoir évaluer la qualité de la sélection de paramètres et donc de comparer les différentes méthodes proposées au cours de cette thèse.

Nous avons testé, pour quelques images et pour différents pourcentages de bruit donné, le programme de reconnaissance. Ce dernier calcule donc, pour l'image, les valeurs des différents paramètres sélectionnés, les normalise, puis attribue l'image à l'une des classes préétablies. L'expérience est répétée 100 fois pour chaque image et pour chaque pourcentage de bruit (en effet le bruit pour un même taux ne se répartit pas de la même façon sur le contour), et permet ainsi d'avoir un pourcentage de bonne reconnaissance. Pour chaque expérience les conditions d'apprentissage sont indiquées en dessous de chaque tableau:

- numéro du lot d'apprentissage suivi du pourcentage et de la profondeur de bruit appliqués aux images de ce lot (0%, 25%, 50%)
- nombre de paramètres sélectionnés
- numéros des paramètres sélectionnés
- méthode de normalisation
- méthode de sélection

Eventuellement, la profondeur du bruit appliqué aux images à reconnaître est également indiquée au-dessus du tableau.
Le pourcentage de bruit des images à reconnaître est indiqué dans le tableau.

Lot n° 1

IMAGES INITIALES NON BRUITEES

NORMALISATION PAR LA METHODE DES

POURCENTAGES

Tableau 1

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0
Nombre de paramètres sélectionnés : 1
Numéro du paramètre sélectionné : 19
Normalisation par pourcentage
Méthode de sélection : Cheung

Tableau 1: Taux de bonne reconnaissance d'une image de chaque classe en fonction du bruit.

Tableau 2

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	90	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	90	0	100	0	0
TR1026	0	0	0	100	0
	10	0	0	64	36
	25	0	0	27	73
	50	0	0	16	84
	75	0	0	12	88
CROIX	0	0	0	0	100
	25	0	0	26	74
	50	0	0	14	86
	75	0	0	26	74
	99	0	0	14	86

Lot : 1 bruit : 0%

Nombre de paramètres sélectionnés : 1

Numéro des paramètres sélectionnés : 10

Normalisation par pourcentage

Méthode - de Foroutan

- du Branch and Bound

Tableau 3

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0

Nombre de paramètres sélectionnés : 2

Numéro des paramètres sélectionnés : 10 19

Normalisation par pourcentage

Méthode : mêmes résultats pour les trois méthodes

Tableau 4

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	90	100	0	0	0
CA2020	0	0	100	0	0
	5	0	100	0	0
	10	95	5	0	0
	15	50	50	0	0
	25	100	0	0	0
	50	100	0	0	0
TR1026	0	0	0	100	0
	15	0	0	95	5
	25	0	0	30	70
	75	0	0	15	85
CROIX	0	0	0	0	100
	15	0	0	0	100
	20	60	0	0	40
	25	90	0	0	10
	50	95	0	0	5
	75	96	0	0	4

Lot : 1 bruit : 0%

Nombre de paramètres sélectionnés : 4

Numéro des paramètres sélectionnés: 6 10 18 19

Normalisation par pourcentage

Méthode : mêmes résultats pour les trois méthodes

Tableau 5

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	79	21	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	75	25
	75	0	0	89	11
	100	0	0	97	3
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit : 0%

Nombre de paramètres sélectionnés : 10

Numéro des paramètres sélectionnés 1-6-8-10-11-12-14-17-18-19

Normalisation par pourcentage

Méthode : mêmes résultats pour les trois méthodes

Tableau 6

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0
Nombre de paramètres sélectionnés : 16
Numéro des paramètres sélectionnés : -(4, 5 16)
Normalisation par pourcentage
Méthode : mêmes résultats pour les trois méthodes

LOT n° 1

IMAGES INITIALES BRUITEES à 25%

PROFONDEUR DU BRUIT 1

NORMALISATION PAR LA METHODE DES POURCENTAGES

Tableau 7

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	76	24
	50	0	0	31	69
	75	0	0	17	83
	100	0	0	8	92
CROIX	0	0	0	0	100
	25	0	0	42	58
	50	0	0	50	50
	75	0	0	41	59
	100	0	0	36	84

Lot : 1 bruit 25%
Nombre de paramètres sélectionnés : 1
Numéro des paramètres sélectionnés : 10
Normalisation par pourcentage
Méthode de Foroutan et du Branch and Bound

Tableau 8

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25%
Nombre de paramètres sélectionnés : 1
Numéro du paramètre sélectionné : 19
Normalisation par pourcentage
Méthode de Cheung

Tableau 9

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	10	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	90	100	0	0	0
CA2020	0	0	100	0	0
	10	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	90	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25 %
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés: 10 19
Méthode : mêmes résultats pour les trois méthodes
Normalisation pourcentage

Tableau 10

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25 %
Nombre de paramètres sélectionnés : 4
Numéro des paramètres sélectionnés: 8 10 18 19
Méthode : mêmes résultats pour les trois méthodes
Normalisation pourcentage

LOT n° 1

IMAGES INITIALES BRUITEES à 50 %

PROFONDEUR DU BRUIT 1

NORMALISATION PAR LA METHODE DES POURCENTAGES

Tableau 11

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	99	1	0	0
	50	100	0	0	0
	75	99	1	0	0
	100	99	1	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	2	0	0	98

Lot : 1 bruit 50%
Nombre de paramètres sélectionnés : 1
Numéro du paramètre sélectionné : 19
Normalisation par la méthode des pourcentages
Méthode de Cheung

Tableau 12

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	10	0	0	14	86
	25	0	0	39	61
	50	0	0	67	33
	75	0	0	77	23
CROIX	0	0	0	0	100
	10	0	0	26	74
	25	0	0	10	90
	50	0	0	15	85
	99	0	0	14	86

Lot : 1 bruit 50%
Nombre de paramètres sélectionnés : 1
Numéro des paramètres sélectionnés : 10
Normalisation par la méthode des pourcentages
Méthode : de Foroutan
de Branch & Bound

Tableau 13

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	0	100
	100	0	0	0	100
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50%
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés : 10 18
Normalisation par la méthode des pourcentages
Méthode : mêmes résultats pour les trois méthodes

Tableau 14

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	10	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50%
Nombre de paramètres sélectionnés : 4
Numéro des paramètres sélectionnés : 8 10 18 19
Normalisation par la méthode des pourcentages
Méthode :mêmes résultats pour les trois méthodes

Tableau 15

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	10	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50%
Nombre de paramètres sélectionnés : 10
Numéro des paramètres sélectionnés 1-6-7-8-10-11-14-17-18-19
Normalisation par la méthode des pourcentages
Méthode: mêmes résultats pour les trois méthodes

Tableau 16

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
		100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50 %
Nombre de paramètres sélectionnés : 16
Numéro des paramètres sélectionnés : -(4, 5 16)
Normalisation par la méthode des pourcentages
Méthode : mêmes résultats pour les trois méthodes

Différents commentaires sur les résultats.

V.4.2) Paramètres perturbateurs

Dans le cas de lots non bruités initialement, on peut constater que la reconnaissance avec deux paramètres (tableau 3) est de 100%, malgré un pourcentage de bruit important, alors que pour quatre paramètres (tableau 4), on constate que les formes de la classe 2 sont rangées dans la classe 1 à partir d'un pourcentage de bruit de 10%.

Les résultats montrent qu'une reconnaissance avec un grand nombre de paramètres n'est pas forcément meilleure qu'une reconnaissance avec un petit nombre de paramètres, car certains paramètres sont "nuisibles". Pour mieux comprendre l'expression " paramètre nuisible", nous donnons un exemple simple :

Exemple

Considérons 6 formes réparties dans trois classes comme suit :

		paramètre n°1	paramètre n°2
Classe n° 1:	Forme 1	(100 ,	50)
	Forme 2	(120 ,	300)
Classe n° 2:	Forme 3	(200 ,	60)
	Forme 4	(210 ,	270)
Classe n° 3:	Forme 5	(300 ,	55)
	Forme 6	(310 ,	280)

On s'aperçoit ici, (fig.1) que si l'on considère uniquement le paramètre n°1, les formes sont bien regroupées et nous permettront une classification et une reconnaissance aisées. Par contre, si l'on fait intervenir les deux paramètres (fig.2), les formes d'une même classe se dispersent, allant jusqu'à donner une classification et une reconnaissance totalement différentes de celles initialement considérées. Le paramètre 2 est donc un paramètre nuisible, pour la classification souhaitée.

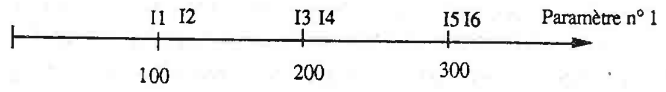


fig.1:

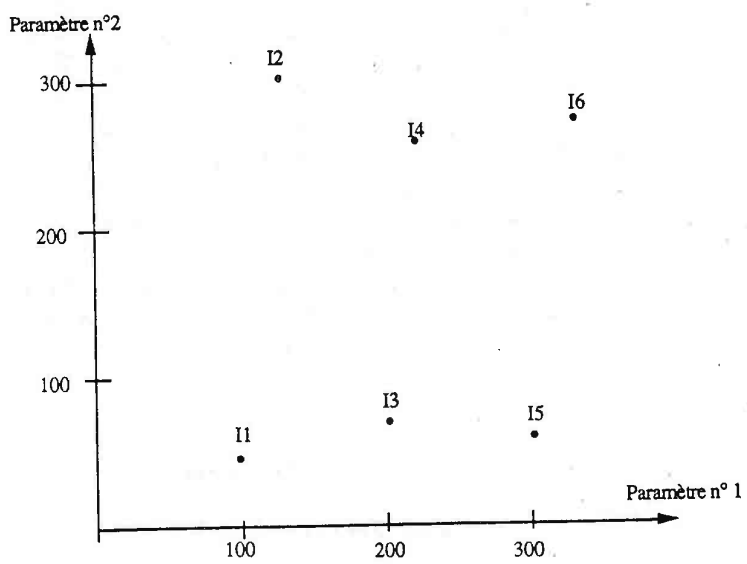


fig.2: Exemple de paramètre perturbateur : Paramètre n° 2.

V.4.3) Influence du bruit appliqué aux images d'apprentissage sur la sélection de paramètres

On constate que si l'on souhaite reconnaître à la dernière étape des images bruitées, il faut avoir sélectionné les paramètres sur des images bruitées et non pas sur des images non bruitées et ceci pour deux raisons. La première raison est que les méthodes de sélection ne donnent pas les mêmes paramètres pour des images bruitées que pour des images non bruitées. Considérons le cas de quatre paramètres à sélectionner. Les trois méthodes de sélection nous ont fourni les mêmes numéros de paramètres pour le lot d'images n°1, non bruité : 6-10-18-19, alors que pour des images bruitées, les trois méthodes nous ont fourni les paramètres 8-10-18-19. Les résultats obtenus dans le deuxième cas sont nettement meilleurs que dans le premier cas. Les méthodes de sélection ont donc remplacé le paramètre 6, correspondant au nombre de changements d'orientation du contour, par le paramètre 8 correspondant à la différence entre le rayon max et le rayon min. Pour des images non bruitées, le nombre de changements d'orientation est toujours de 4 pour un carré, de 3 pour un triangle, de 12 pour une croix et très important pour un cercle. Ce paramètre est donc un bon paramètre

pour ce type d'images. Mais lorsque le contour des images est bruité, le nombre de changements d'orientation augmente très rapidement et ceci, quelque soit la forme considérée. Le paramètre 6 perd alors sa qualité de paramètre pertinent, ce qui explique le taux important de mauvaise classification dans le cas où le lot est constitué d'images non bruitées et que les images à classer sont bruitées. Par contre, le rayon max et le rayon min d'une image bruitée ont une grande probabilité de garder les mêmes valeurs que les images non bruitées. Le paramètre 8 est donc un paramètre peu sensible au bruitage du contour.

La deuxième raison est due aux calculs des centres de gravité. En effet si on tient compte du bruit, les centres de gravité des classes vont être modifiés, respectant mieux la fluctuation des valeurs des paramètres.

V.4.4) Comparaison des méthodes de sélection

Les trois méthodes donnent dans de très nombreux cas les mêmes résultats (le temps d'exécution mis à part). Pour le cas où le nombre de paramètres à sélectionner est égal à 1, les trois méthodes donnent un résultat différent ; la méthode de Foroutan et du Branch and Bound ont sélectionné le paramètre 10 (variance du rayon) alors que la méthode de Cheung a sélectionné le paramètre 19 (troisième moment invariant de Hu).

Les résultats obtenus avec le paramètre 19 sont très satisfaisants, la reconnaissance est de 100%, alors que pour le paramètre 10 la reconnaissance est nettement moins bonne, surtout pour les classes triangles et croix.

Le programme de reconnaissance nous permet de confirmer l'hypothèse que nous avions avancée dans le chapitre sur la classification, à savoir que la méthode de sélection de Cheung semble donner de meilleurs résultats que la méthode de Foroutan and Sklansky ou la méthode du Branch and Bound.

V.4.5) Normalisation par la méthode de la moyenne et de variance

Nous avons recommencé la même série de mesures, mais en utilisant la deuxième méthode de normalisation (moyenne = 0 et variance=1). Les tableaux des différents résultats sont donnés en annexe (Tableaux 21- 31).

- A la différence du premier type de normalisation, les trois méthodes de sélection ne donnent pas les mêmes paramètres dans tous les cas. Par exemple, pour le lot 1 bruité initialement à 25%, on obtient :

  - pour la méthode de Cheung, les paramètres : 18-19 (2 moments invariants),

  - pour la méthode de Foroutan, les paramètres : 17-18 (2 moments invariants).

- pour la méthode du Branch and Bound les paramètres : 12-14 (largeur et surface du rectangle circonscrit)

- Les résultats obtenus pour le lot n°1 non bruité à partir de la méthode du Branch and Bound montrent que le taux de reconnaissance est de 100% avec les paramètres 17 et 18, alors qu'il chute légèrement avec les paramètres 15 17 18 19.

- Le taux de bonne reconnaissance est meilleur lorsqu'on travaille sur un lot d'apprentissage bruité que sur un lot non bruité. Dans le cas de 4 paramètres sélectionnés à l'aide de la méthode de Foroutan, on constate que le taux de bonne reconnaissance est de 100% pour le lot 1 bruité initialement à 50%, alors qu'il est nettement moins bon pour le lot 1 non bruité.

- Pour un nombre donné de paramètres à sélectionner, les méthodes de sélection donnent des paramètres différents, suivant le pourcentage de bruit du lot initial. Par exemple :

- La méthode du Branch and Bound, pour le lot 1 non bruité supprime les paramètres 1-2-5, alors que pour le même lot bruité à 50% elle supprime les paramètres 1-5-16.

- La méthode du Cheung pour le lot 1 non bruité, donne les paramètres 15-18, alors que pour 25% donne les paramètres 17-18 et pour 50% elle donne les paramètres 18-19.

- La méthode de Cheung donne dans la plupart des cas de meilleurs résultats que les deux autres méthodes, surtout quand le nombre de paramètres est réduit.

Les observations faites sur cette deuxième série de mesures confirment les résultats obtenus pour la première série.

V.4.6) Influence de l'amplitude du bruit (profondeur de 2 et 3 pixels sur les contours)

Nous présentons dans les deux tableaux suivant, les résultats obtenus pour le lot 1 bruité initialement à 50% à la profondeur 2, lors d'une sélection de 2 paramètres à l'aide de la méthode de Cheung. Le tableau 17 correspond à une reconnaissance d'images bruitées de 0 % à 100% avec une profondeur de 2; le tableau 18 correspond à une reconnaissance d'images bruitées de 0 % à 100% avec une profondeur de 3.

Tableau 17

Profondeur du bruit des images à reconnaître : 2

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	84	16	0	0
	50	88	12	0	0
	75	93	7	0	0
	100	90	10	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	1	99	0	0
	100	2	98	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	3	97

Lot : 1 bruit 50% profondeur 2
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés : 17-19
Normalisation par moyenne et écart type
Méthode de Cheung

Tableau 18

Profondeur du bruit des images à reconnaître : 3

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	93	7	0	0
	50	94	6	0	0
	75	99	1	0	0
	100	97	3	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	3	97	0	0
	75	15	85	0	0
	100	25	75	0	0
TR1026	0	0	0	100	0
	25	0	0	99	1
	50	0	0	96	4
	75	0	0	96	4
	100	0	0	98	2
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	10	90
	75	0	0	49	51
	100	0	0	68	62

Lot : 1 bruit 50% profondeur 2
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés : 17-19
Normalisation par moyenne et écart type
Méthode de Cheung

On constate

- que dans les deux cas, les paramètres sélectionnés sont les mêmes : 17-19 (moments invariants de Hu).

- que la reconnaissance de l'image CL102020 (cercle de rayon 10), contrairement aux autres images, est meilleure avec la profondeur 3, qu'avec la profondeur 2. On aurait pu penser que la reconnaissance allait se dégrader au fur et à mesure que l'on augmentait la profondeur du bruit, ceci est vrai pour les images à l'exception du cercle. En fait ceci s'explique par le fait que les deux paramètres sélectionnés sont tous deux des moments invariants à l'homothétie et que le bruit ne modifie pas la forme générale du cercle. Ce qui n'est pas le cas d'un carré où les coins risquent d'être biseautés lors du bruitage et de ressembler alors à des cercles

- que le taux de mauvaise classification dans le premier cas est de 51 pour 2 000 images (2,55 %) et de 198 pour 2 000 (9,8 %) dans le deuxième cas. Donc, le taux de bonne classification se dégrade lorsque le bruit devient important, à l'exception du cercle comme on vient de le voir.

Les tableaux 19 et 20 présentent les résultats obtenus pour le lot 1 bruité initialement à 50% à la profondeur 2, lors d'une sélection de 10 paramètres à l'aide de la méthode de Cheung. Le tableau 19 correspond à une reconnaissance d'images bruitées de 0% à 100% avec une profondeur de 2; le tableau 20 correspond à une reconnaissance d'images bruitées de 0% à 100% avec une profondeur de 3.

tableau 19

Profondeur du bruit des images à reconnaître : 2

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	99	1	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25 % profondeur 2
Nombre de paramètres sélectionnés : 10
Numéro des paramètres sélectionnés : 1 8 10 12 13 14 15 17 18 19
Méthode de Cheung
Normalisation par moyenne et écart-type

tableau 20

Pourcentage de bruit des images à reconnaître : 3

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	98	2	0	0
	50	97	3	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	100	0	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	1	99	0	0
	100	20	80	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25 % profondeur 2
Nombre de paramètres sélectionnés : 10
Numéro des paramètres sélectionnés : 1 8 10 12 13 14 15 17 18 19
Méthode de Cheung
Normalisation par moyenne et écart-type

On constate

- que dans les deux cas, les paramètres sélectionnés sont les mêmes :

  1-8-10-12-13-14-15-17-18-19

- que la reconnaissance du cercle CL102020 n'est pas meilleure cette fois pour la profondeur 3 que pour la profondeur 2. Ceci s'explique par le fait que les paramètres sélectionnés, autres que les moments invariants de Hu, sont plus sensibles au bruit.

- que le taux de mauvaise classification dans le premier cas est de 1 pour 2 000 images (0,05%) et de 126 pour 2 000 (6,3%) dans le deuxième cas, c'est-à-dire une nette dégradation avec le bruit.

V.5) CONCLUSION

V.5.1) Discussion à propos des 3 méthodes de sélection

Les méthodes à programmation dynamique sont en général basées sur une hypothèse simplificatrice ; un sous-ensemble de paramètres donne de meilleurs résultats qu'un sous-ensemble plus petit contenu dans le premier. C'est-à-dire que lorsqu'on augmente le nombre de paramètres, le pourcentage de bonne reconnaissance est plus grand, ou encore le lot initial ne doit pas contenir de paramètres "nuisibles". Cette hypothèse permet ainsi d'éviter de vérifier un certain nombre de combinaisons possibles. Si cette hypothèse est vérifiée, alors ces méthodes fourniraient des solutions optimales. En pratique, on ne peut assurer que cette condition est toujours satisfaite, et donc assurer que ces méthodes donnent des solutions optimales. Cependant, les résultats obtenus sont très satisfaisants et tout particulièrement ceux de la méthode de Cheung, comme peut en témoigner la partie expérimentale de cette thèse.

Lorsque le nombre de paramètres est réduit, la méthode du Branch and Bound ne donne pas de bons résultats. De plus, l'inconvénient majeur de la méthode du Branch and Bound est son temps d'exécution trop sensible à la qualité des paramètres. En effet comme nous le signalions, la sélection de 4 paramètres parmi 19 peut prendre quelques secondes pour un lot d'images et 2 heures pour un autre lot de même taille. On ne peut donc utiliser cette méthode pour un traitement en temps réel.

La méthode de Foroutan est intéressante lorsque le nombre de paramètres à sélectionner est important. Dans le cas contraire, la méthode de Cheung semble plus appropriée.

V.5.2) Discussion à propos de la pertinence des paramètres

Au niveau des paramètres géométriques, certains semblent cependant plus appropriés que d'autres *(pour les classifications que nous avons réalisées)*.

On a pu constater que les moments invariants de Hu (paramètres 17-18-19) faisaient souvent partie des paramètres sélectionnés, et ceci en raison de leur invariance par translation, rotation et échelle. La variance du rayon moyen (paramètre 10) est aussi un paramètre que nous avons souvent retrouvé dans le sous-ensemble solution et ceci est dû à sa faible sensibilité au bruit. Le paramètre nombre de changements d'orientation (paramètre 6) est sélectionné lorsque les images du lot initial ne sont pas bruitées. Dans le cas d'images bruitées ce paramètre est supprimé, trop sensible au bruit.
Les paramètres les plus souvent éliminés sont les paramètres 4, 5 et 16, les deux premiers étant les coordonnées du centre de gravité. Pour une reconnaissance de forme, l'objet

pouvant se trouver n'importe où dans l'image, les coordonnées de leurs centres de gravité peuvent être considérés comme des paramètres perturbateurs. En effet, il est possible qu'un carré et qu'un cercle possèdent le même centre de gravité. .
Le troisième paramètre correspond au pourcentage de symétrie par rapport à l'axe des ordonnées. Ce dernier avait été intentionnellement choisi, de manière à ce qu'il prenne la même valeur pour toutes les images du lot 1. Ce paramètre est donc un paramètre non informatif, voire perturbateur.

Les paramètres périmètre, surface, nombre d'érosions nécessaires à la disparition totale de la forme, longueur, largeur et surface du rectangle circonscrit ne sont pas des paramètres pertinents pour la classification réalisée. Il est à souligner que ces paramètres auraient été des paramètres pertinents si la classification avait été réalisée en fonction de la taille et non de la forme.

Ces résultats confirment la qualité des méthodes à programmation dynamique.

CONCLUSION

Cette étude a donc abouti à la production d'un logiciel, réalisant une sélection automatique de paramètres pertinents, et ceci quel que soit le lot d'images considéré et la classification souhaitée. Pour la réalisation d'un tel programme (sélection automatique), aucune hypothèse ne doit être faite sur l'ensemble d'apprentissage, limitant ainsi le choix des méthodes de sélection. En effet, le nombre de méthodes de sélection de paramètres est très important et très varié dans la littérature. Cependant, la plupart de ces méthodes restent des méthodes très théoriques qu'il est souvent difficile d'appliquer à la pratique. Les méthodes à programmation dynamique que nous avons présentées dans cette thèse fournissent de bons résultats, comme peut en témoigner la partie expérimentale. Ce choix s'explique aussi par leur simplicité de mise en œuvre.

A ce logiciel viennent se greffer plusieurs programmes:

** programmes de calculs des mesures géométriques,

** programmes de classification,

** programme de reconnaissance de forme.

Les premiers nous ont fourni le lot initial de paramètres sur lequel nous avons réalisé la sélection. Les deux dernières catégories de programmes nous ont permis de vérifier et d'estimer la qualité du sous-ensemble des paramètres sélectionnés.

De cette étude nous avons tiré plusieurs conclusions.

Au niveau de la sélection des paramètres: On a pu constater qu'un sous-ensemble de paramètres ne donne pas forcément une meilleure classification qu'un sous-ensemble plus petit contenu dans le premier sous-ensemble. En effet, certains paramètres sont nuisibles à la classification; leur présence appauvrit la puissance de la classification. C'est pour cette raison que parfois la classification ou la reconnaissance obtenue avec peu de paramètres est meilleure que la classification ou la reconnaissance obtenue avec tous les paramètres, car plus proches du type de classification que l'on souhaite. En effet, la machine ne peut discerner si l'utilisateur souhaite par exemple classer les formes selon leur forme (cercle, carré, triangle...) ou selon leur taille (petite, moyenne, grande).

Le caractère automatique de ce logiciel introduit nécessairement la notion de paramètres perturbateurs, que l'on rencontre rarement dans la littérature. En effet dans un cas particulier, la classification souhaitée est connue et l'opérateur réalise inconsciemment une première sélection. Il ne choisira pas parmi les paramètres initiaux le paramètre surface si la classification qu'il souhaite réaliser est une classification d'après les formes (cercles, carrés, triangles...). Dans le cadre d'une sélection automatique, le nombre de paramètres initiaux doit être élevé et divers, afin de pouvoir répondre à un grand nombre de types de classification, classification qui n'est donc pas connue lors de la création du logiciel. Ceci explique le fait qu'un paramètre sera tantôt un paramètre perturbateur, tantôt

un paramètre pertinent pour un même lot d'images, en fonction du type de classification attendue.

Ce traitement est "extensible": nous entendons par "extensible", le fait que d'autres programmes aussi bien de calculs de paramètres géométriques, de prétraitements, que de sélection de paramètres peuvent être ajoutés, améliorant ainsi l'efficacité du programme.

Il serait intéressant dans un deuxième temps d'appliquer ce traitement sur de vraies images, à niveaux de gris, moyennant l'ajout de quelques programmes de prétraitements (voir annexe D: Traitements appliqués à des images réelles).

La sélection des paramètres a toujours été considérée comme un problème d'optimisation, en fait ceci est vrai dans la majorité des cas, mais elle peut aussi être considérée comme un moyen de suppression des paramètres perturbateurs, permettant ainsi une meilleure classification et reconnaissance. Donc la sélection n'est pas uniquement un moyen de minimiser le temps de la classification et de la reconnaissance mais aussi un moyen d'améliorer ces dernières.

ANNEXE

ANNEXE A

Démonstration du calcul des moments invariants de HU

Démonstration 1

$$m_{00i} = 1 + 1 + 1 \ldots\ldots\ldots + 1 = \delta$$

$$m_{01i} = Y_i + Y_i + Y_i + \ldots\ldots\ldots Y_i = \delta \cdot Y_i$$

$$m_{02i} = Y_i^2 + Y_i^2 + Y_i^2 + \ldots\ldots\ldots Y_i^2 = \delta \cdot Y_i^2$$

$$m_{03i} = Y_i^3 + Y_i^3 + Y_i^3 + \ldots\ldots\ldots Y_i^3 = \delta \cdot Y_i^3$$

$$m_{10i} = X_i + (X_i + 1) + (X_i + 2) + (X_i + 3) + (X_i + 4) \ldots\ldots\ldots + (X_i + \delta - 1)$$

$$= \delta \cdot X_i + (0 + 1 + 2 + 3 \ldots\ldots\ldots + \delta - 1)$$

$$= \delta \cdot X_i + (\delta^2 - \delta)/2$$

$$m_{11i} = X_i \cdot Y_i + (X_i + 1) \cdot Y_i + (X_i + 2) \cdot Y_i + (X_i + 3) \cdot Y_i + \ldots\ldots\ldots + (X_i + \delta - 1) \cdot Y_i$$

$$= Y_i \cdot [X_i + (X_i + 1) + (X_i + 2) + (X_i + 3) + (X_i + 4) \ldots\ldots\ldots + (X_i + \delta - 1)]$$

$$= Y_i \cdot (\delta \cdot X_i + (\delta^2 - \delta)/2)$$

$$= Y_i \cdot m_{10i}$$

$$m_{12i} = Y_i^2 \cdot X_i + Y_i^2 \cdot (X_i + 1) + Y_i^2 \cdot (X_i + 2) + Y_i^2 \cdot (X_i + 3) + \ldots\ldots\ldots + Y_i^2 \cdot (X_i + \delta - 1)]$$

$$= Y_i^2 \cdot [X_i + (X_i + 1) + (X_i + 2) + (X_i + 3) + \ldots\ldots\ldots + (X_i + \delta - 1)]$$

$$= Y_i^2 \cdot [\delta \cdot X_i + (\delta^2 - \delta)/2]$$

$$= Y_i^2 \cdot m_{10i}$$

$$m_{20i} = X_i^2 + (X_i + 1)^2 + (X_i + 2)^2 + (X_i + 3)^2 + \ldots\ldots\ldots + (X_i + \delta - 1)^2$$

$$= \delta \cdot X_i^2 + 2 \cdot X_i \cdot (0 + 1 + 2 \ldots\ldots\ldots + \delta - 1) + (0 + 1 + 2^2 \ldots\ldots\ldots + (\delta - 1)^2)$$

$$= \delta \cdot X_i^2 + X_i \cdot (\delta^2 - \delta) + \sum_{n=0}^{\delta-1} n^2$$

où

$$\sum_{n=0}^{\delta-1} n^2 = \frac{\delta^3}{3} - \frac{\delta^2}{2} + \frac{\delta}{6}$$

$$= \delta \cdot X_i^2 + X_i \cdot (\delta^2 - \delta) + \frac{\delta^3}{3} - \frac{\delta^2}{2} + \frac{\delta}{6}$$

$$m_{30i} = X_i^3 + (X_i + 2)^3 + (X_i + 3)^3 + \ldots\ldots\ldots + (X_i + \delta - 1)^3$$

$$= \delta \cdot X_i^3 + 3 \cdot X_i^2 \cdot (0 + 1 + 2 + \ldots\ldots\ldots \delta - 1)$$

$$+ 3 \cdot X_i \cdot (0 + 1^2 + 2^2 + \ldots\ldots\ldots (\delta - 1)^2)$$

$$+ (0 + 1^3 + 2^3 + \ldots\ldots\ldots (\delta - 1)^3)$$

$$= \delta \cdot X_i^3 + 3 X_i^2 \cdot (\delta^2 - \delta)/2 + 3 \cdot \left[\frac{\delta^3}{3} - \frac{\delta^2}{2} + \frac{\delta}{6}\right] \cdot X_i + \sum_{n=0}^{\delta-1} n^3$$

où

$$\sum_{n=0}^{\delta-1} n^3 = \frac{\delta^4}{4} - \frac{\delta^3}{2} + \frac{\delta^2}{4}$$

$$= \delta \cdot X_i^3 + 3 X_i^2 \cdot (\delta^2 - \delta)/2 + 3 \cdot \left[\frac{\delta^3}{3} - \frac{\delta^2}{2} + \frac{\delta}{6}\right] \cdot X_i + \frac{\delta^4}{4} - \frac{\delta^3}{2} + \frac{\delta^2}{4}$$

Démonstration 2

$$\mu_{00} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 \cdot (j - \bar{j})^0 = \sum_{i=0}^{M} \sum_{j=0}^{N} i^0 \cdot j^0 = m_{00}$$

$$\mu_{01} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 \cdot (j - \bar{j})^1 = \sum_{i=0}^{M} \sum_{j=0}^{N} i^0 \cdot j^1 - \sum_{i=0}^{M} \sum_{j=0}^{N} i^0 \cdot \bar{j}$$

$$= m_{01} - m_{01} \cdot (m_{01}/m_{00}) = 0$$

$$\mu_{02} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 \cdot (j - \bar{j})^2$$

$$= \sum_{i=0}^{M} \sum_{j=0}^{N} i^0 \cdot j^2 - 2 \cdot \sum_{i=0}^{M} \sum_{j=0}^{N} i^0 \cdot \bar{j} \cdot j + \sum_{i=0}^{M} \sum_{j=0}^{N} i^0 \cdot \bar{j}^2$$

$$= m_{02} - 2 \cdot m_{01} \cdot (m_{01}/m_{00}) + m_{00} \cdot (m_{01}/m_{00})^2$$

$$= m_{02} - m_{01}^2/m_{00}$$

De la même manière nous obtenons les moments suivants :

$$\mu_{03} = m_{03} - 3 . (m_{01} / m_{00}) . m_{02} + 2 . (m_{01} / m_{00})^2 . m_{01}$$

$$\mu_{10} = m_{10} - (m_{10} / m_{00}) . m_{00} = 0$$

$$\mu_{11} = m_{11} - (m_{01} / m_{00}) . m_{10}$$

$$\mu_{12} = m_{12} - 2 . (m_{01} / m_{00}) . m_{11} - (m_{10} / m_{00}) . m_{02} + 2 . (m_{01} / m_{00}) . m_{10}$$

$$\mu_{20} = m_{20} - m_{10}^2 / m_{00}$$

$$\mu_{21} = m_{21} - 2 . (m_{10} / m_{00}) . m_{11} - (m_{01} / m_{00}) . m_{20} + 2 . (m_{10} / m_{00})^2 . m_{01}$$

$$\mu_{30} = m_{30} - 3 . (m_{10} / m_{00}) . m_{20} + 2 . (m_{10} / m_{00})^2 . m_{10}$$

ANNEXE B

Démonstration de l'optimalité du développement de Karhunen-Lœve

Soit $V = \{ V_1, V_2, \ldots\ldots V_n \}$ une base orthonormée

Tout vecteur X est défini par la relation:

$$X = \sum_{i=1}^{n} c_i \; V_i \qquad X (c_1 , c_2 , \ldots\ldots c_n) \text{ dans la base } V$$

$$\text{avec } c_j = X^T V_j \qquad (1)$$

soit $T^T = \{ V_1, V_2, \ldots\ldots V_m \}$
$S^T = \{ V_{m+1}, V_{m+2}, \ldots\ldots V_n \}$

Le vecteur transformé devient :

$$Y^T = X^T T^T = \left[\sum_{j=1}^{n} c_j \; V_j^T \right] [V_1 , V_2 \ldots\ldots V_m] = [c_1 , c_2 \ldots\ldots c_m]$$

La transformation optimale est celle qui maximise la valeur de $E [| Y^2 |]$
Or

$$E [| Y^2 |] = E (Y^T Y) = \sum_{j=1}^{m} E (| c_j |^2)$$

Or d'après (1)

$$\sum_{j=1}^{m} E [(X^T V_j)^2] = \sum_{j=1}^{m} E [\; (X^T V_j)^T (X^T V_j) \;]$$

$$\sum_{j=1}^{m} E [V_j^T X X^T V_j] = \sum_{j=1}^{m} V_j^T E [X X^T] \; V_j = \sum_{j=1}^{m} V_j^T A \; V_j$$

La base étant orthonormée ($V_j^T V_j = 1$) et ∂_j étant des multiplicateurs de Lagrange, on doit maximiser:

$$E [\Omega Y^2 \Omega] - \sum_{j=1}^{m} \partial_j V_j^T \; V_j = \sum_{j=1}^{m} (V_j^T A V_j - \partial_j V_j^T \; V_j)$$

En remplaçant V_j par $V_j + dV_j$ et en égalant à zéro les termes du premier ordre on obtient:

$$E[\Omega Y^2 \Omega] = \sum_{j=1}^{m} \partial_j$$

Donc pour maximiser $E[\Omega Y^2 \Omega]$, c'est-à-dire minimiser l'erreur moyenne commise dans la transformation d'un vecteur X à n composantes en un vecteur Y à m composantes, il faut maximiser

$$\sum_{j=1}^{m} \partial_j$$

c'est-à-dire prendre les m vecteurs propres de A associés aux m plus grandes valeurs propres.

ANNEXE C

Tableaux de résultats

Pourcentage de Reconnaissance obtenu à partir des résultats d'un apprentissage effectué sur le lot 1, normalisé à l'aide de la méthode de la moyenne et de la variance.

Tableau 21

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	96	4	0	0
	50	98	2	0	0
	75	94	6	0	0
	100	92	8	0	0
CA2020	0	0	100	0	0
	25	1	99	0	0
	50	4	96	0	0
	75	11	89	0	0
	100	21	79	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0% Normalisation : moyenne et écart type
Nombre de paramètres sélectionnés : 2 méthode de Foroutan en 53s
Numéro des paramètres sélectionnés : 15 18 méthode de Cheung en 12s

Tableau 22

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0% Normalisation : moyenne et écart type
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés : 17 18 méthode de Branch & Bound

Tableau 23

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	99	1	0	0
	50	99	1	0	0
	75	98	2	0	0
	100	95	5	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	1	99	0	0
	75	1	99	0	0
	100	4	96	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0% Normalisation : moyenne et écart type
Nombre de paramètres sélectionnés : 4
Numéro des paramètres sélectionnés : 15 17 18 19
méthode du Branch & Bound

Tableau 24

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	99	0	1
	100	0	99	0	1
TR1026	0	0	0	100	0
	25	0	0	98	2
	50	0	0	91	9
	75	0	0	97	3
	100	0	0	97	3
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0% Normalisation : moyenne et écart type
Nombre de paramètres sélectionnés : 4
Numéro des paramètres séletionnés : 1 15 17 18
méthode de Foroutan

Tableau 25

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 0% Normalisation : moyenne et écart type
Nombre de paramètres sélectionnés : 16
Numéro des paramètres sélectionnés : - (1,2,5)
méthode de Branch & Bound

Tableau 26

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25% Normalisation: Moyenne et écart type
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés 18-19
méthode de Cheung en 12s

Tableau 27

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	0	100	0	0
	25	43	57	0	0
	50	74	26	0	0
	75	91	9	0	0
	100	92	8	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25% Normalisation: Moyenne et écart type
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés 12-14
méthode de Branch and Bound en 788s

Tableau 28

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	92	8	0	0
	50	86	14	0	0
	75	88	12	0	0
	100	88	12	0	0
CA2020	0	0	100	0	0
	25	1	99	0	0
	50	2	98	0	0
	75	6	94	0	0
	100	10	90	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 25% Normalisation: Moyenne et écart type
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés : 17-18
méthode de Foroutan en 55s

Tableau 29

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50 % Normalisation: Moyenne et écart type
Nombre de paramètres sélectionnés : 2
Numéro des paramètres sélectionnés : 18 19
méthode de Cheung

Tableau 30

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50 % Normalisation: Moyenne et écart type
Nombre de paramètres sélectionnés : 4
Numéro des paramètres sélectionnés 8 15 17 18
méthode de Foroutan

Tableau 31

Nom de l'image	Pourcentage de bruit :	Classe :1	Classe : 2	Classe :3	Classe :4
CL102020	0	100	0	0	0
	25	100	0	0	0
	50	100	0	0	0
	75	100	0	0	0
	100	100	0	0	0
CA2020	0	0	100	0	0
	25	0	100	0	0
	50	0	100	0	0
	75	0	100	0	0
	100	0	100	0	0
TR1026	0	0	0	100	0
	25	0	0	100	0
	50	0	0	100	0
	75	0	0	100	0
	100	0	0	100	0
CROIX	0	0	0	0	100
	25	0	0	0	100
	50	0	0	0	100
	75	0	0	0	100
	100	0	0	0	100

Lot : 1 bruit 50 % Normalisation: Moyenne et écart type
Nombre de paramètres sélectionnés : 16
Numéro des paramètres sélectionnés : - (1,5,16)
méthode de Branch and Bound en 9s

ANNEXE D

Traitements sur des images réelles

Pour créditer notre recherche, nous avons dans un deuxième temps, réalisé l'acquisition d'images monochromes. Notre test a été effectué sur six images (objets divers). Comme on peut le constater sur les photos, ces images sont réparties en trois classes de deux éléments chacune.

Classe 1 : Image 1 : Cosse
Image 2 : Cosse orientée différemment
Classe 2 : Image 3 : Rondelle
Image 4 : Rondelle (de $\emptyset_{ext}$ et de $\emptyset_{int}$ différents)
Classe 3 : Image 5 : Petit écrou
Image 6 : Grand écrou

Après avoir binarisé ces images, nous avons appliqué les différents traitements suivants :

- calcul de paramètres géométriques (19),
- normalisation des valeurs de ces paramètres,
- sélection de paramètres (2 et 4)
- classification
- reconnaissance de forme.

Les résultats obtenus à l'aide de ces images sont identiques à ceux obtenus sur les images synthétiques.

Par exemple pour une sélection de deux paramètres se sont encore une fois les paramètres 10 (variance du rayon) et 19 (troisième moment de Hu), qui ont été sélectionnés.

Au niveau de la classification on retrouve bien, à l'aide uniquement de ces deux paramètres la partition souhaitée.

Au niveau de la reconnaissance de forme, les images sont bien classées. Nous avons testé la reconnaissance d'un cercle provenant d'une image synthétique, toujours dans le cas de deux paramètres sélectionnés, alors que l'apprentissage avait été effectué sur des images réelles. Le programme l'a bien rangé dans la classe des cercles.

CLASSE 1 : (cosses)

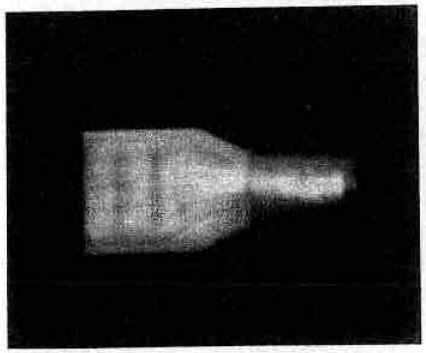


Image 1 : Cosse

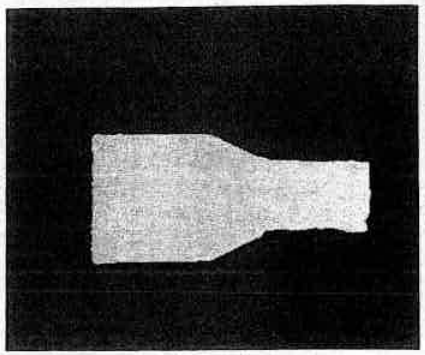


Image 1 binarisée

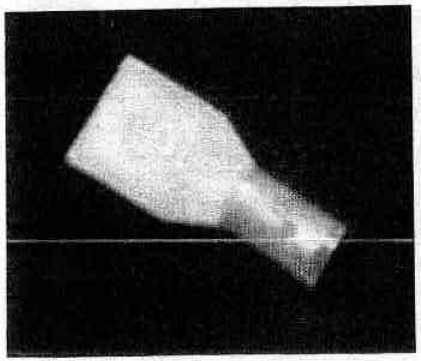


Image 2 : Cosse orientée différemment

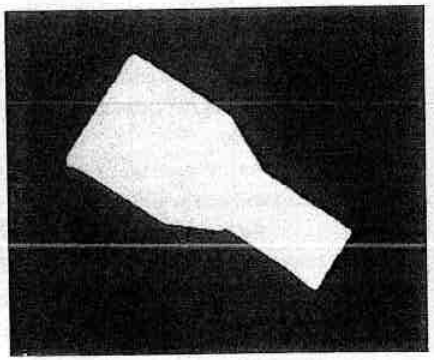


Image 2 binarisée

CLASSE 2 : (rondelles)

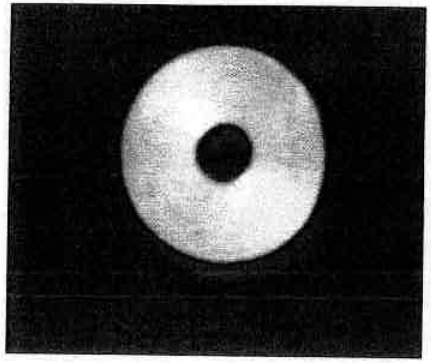


Image 3 : Rondelle 1

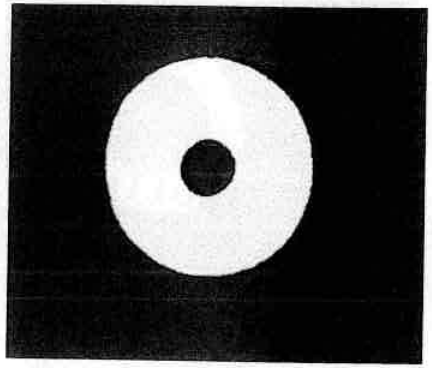


Image 3 binarisée

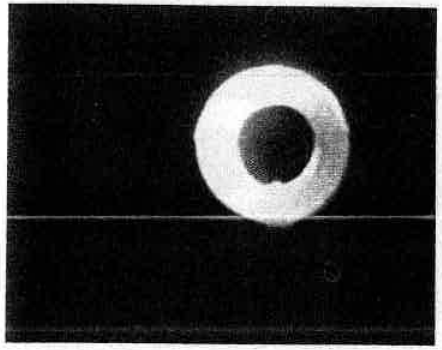


Image 4 : Rondelle 2

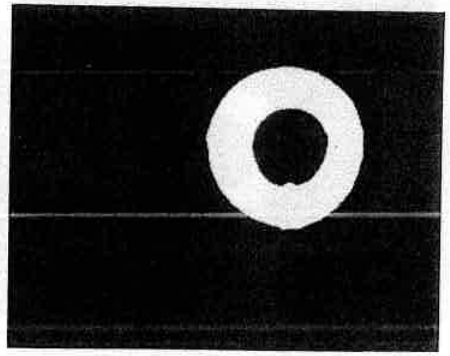


Image 4 binarisée

CLASSE 3 : (Ecrou)

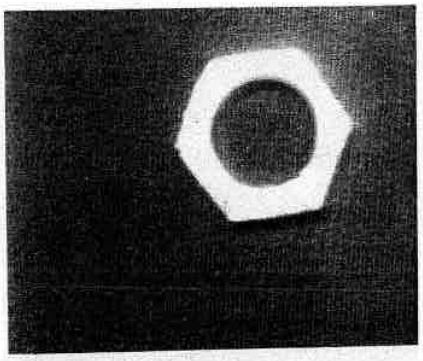


Image 5 : Ecrou 1

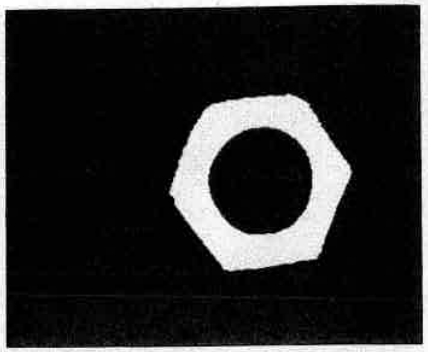


Image 5 binarisée

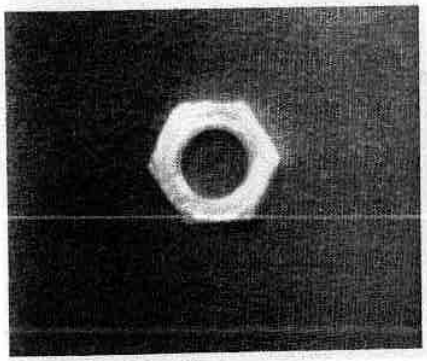


Image 6 : Ecrou 2

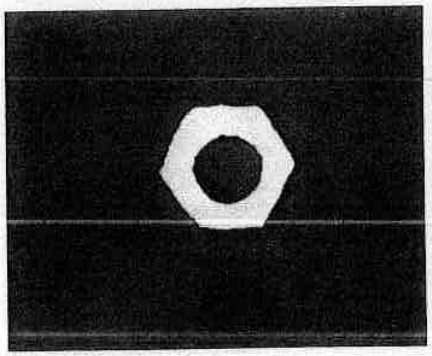


Image 6 binarisée

BIBLIOGRAPHIE

- [BEZD-71] J.C. BEZDEK and P.F CASTELAZ *"Prototype classification and feature selection with Fuzzy Sets"*. **IEEE Transaction.on Systems, Man, and Cybernetics** (1971), Vol: 7, 87-92.

- [BEZD-83] J C BEZDEK *"Some recent applications of fuzzy C-Means in pattern recognition and image processing"*. **IEEE** (1983).

- [BHAR-86] B.BHARATHI D and V.V.S. SARMA *"Binary tree design using fuzzy isodata"*. **Pattern Recognition Letters,** (1986), Vol:4,13-18.

- [BIDA-86] H.B. BIDASARIA *"Least desirable feature elimination in a general pattern recognition problem"*.**Pattern Recognition,** (1986), Vol 20 n°3; 365-370.

- [BOMB-87] V. BOMBARDIER *"Développement de programmes de traitement d'image pour le logiciel d'Aide à la Conception et à l'Application de traitements d'Images"*.**D.E.A à l'Université de Nancy I,** (1987).

- [CABR-81] CABRERA QUINTERO *"Méthodes pratiques pour la reconnaissance de forme."* **Thèse Université de Clermont II,** (1981).

- [CHAN-73] CHANG *" Dynamic programming as applied to Feature Subset selection in a Pattern recognition System "*. **IEEE Transaction.on Systems, Man, and Cybernetics,** (Mars 73), Vol SMC-37,N°2.

- [CHAT-84] J.P. CHATEAU *Systéme d'acquisition et de traitement d'image à vocation industrielle."* **Thèse de troisième cycle de Nancy I,** (1984).

- [CHAU-85] B.B.CHAUDHURI *"An efficient hierarchical clustering technique"*. **Pattern Recognition letters 3,** (1985), 179-183.

- [CHEN-75] C.H. CHEN *"On a class of computationally efficient feature selection criteria "*. **Pattern Recognition** ,(1975), Vol: 7 N° 1-2, 87-94.

- [CHEU-78] R.S CHEUNG and B.A EINSTEIN *"Feature Selection via Dynamic Programming for Text Independant Speaker Identification"*. , **IEEE Trans.Acoustic Speech and Signal Processing** (1978), Vol:26 ASSP, 397-403,

- [CLIF-86] A.CLIFFORD SHAFFER and HANAN SAMET.*"An optimal quadtree construction algorithm"*. **I.E.E.E,** (1986).

- [COST-85] M. COSTER , J-L. CHERMANT *"Précis d'Analyse d'images"*. **Editions du Centre National de la Recherche scientifique,** Paris, (1985).

-[DECE-76] DECEL and MARANI *" Feature combinaisons and the Bhattacharyya criterion"*, **Communications in statistics Theory and Methods,** (1976), Vol A5, 1143-1152.

-[DECE-77] DECEL and MAYEKAR *" Feature combinaisons and the divergence criterion"* , **Computers and Mathematics with applications** (1977), Vol 6, 71-76.

-[DECE-79] DECEL and GUSEMAN *"Linear feature selection with application,* (1979), **Pattern Recognition,**(1979), Vol:11; 55-63.

- [DIDA-70] E. DIDAY *" Une nouvelle méthode de classification automatique et reconnaissance de formes"* **Revue de statistique Appliquée,**(1970), Vol: 19 ; n°2.

- [DURO-78] M. DUROC *" Autoapprentissage et reconnaissance en temps réel, des segments de la parole continue, autoadaptation aux locuteurs"*, **Thèse de Nancy I,** (9 Juin 78).

- [FATO-87] **FATOS YARMAN-VURAL** *"Noise, Histogram and Cluster validity for Gaussian-Mixtured Data"*., **Pattern Recognition,** (1987), Vol:20; N°4 pg: 385-401.

- [FAUR-85] **A. FAURE** *"Perception et reconnaissance de formes"*. **edit: Editest,** (1985).

- [FORO-86] **FOROUTAN and J. SKLANSKY** *"Automatic Feature selection for classification of non- gaussian data"* **IEEE,**(1986) 327-329.

- [FU-67] **K.S FU** *" Sequential Methods in Pattern Recognition and Machine Learning "* , (1967).

- [FU-67] **K.S. FU and CHIEN G. CARDILLO** *" A dynamic programme approch to sequential pattern recognition , IEEE* **Transactions on Pattern Analysis and Machine intelligence,**(1967), Vol:8, n°3, 313-26, USA.

-[FUKU-70] **FUKUNAGA KOONTZ** *"Application of K.L.E. to feature selection and ordering, IEEE* **Transactions on Computers,** (Avril 70), Vol C-19, n°4.

- [GONZA-77] **R.GONZALEZ and P. WINTZ**" *Digital Image Processing"*, **edit: ADDISON WESLEY Compagny,** Collection : Advanced Book Program World Science Division, (1977).

- [GOOD-78] **D.G. GOOGENOUGH P.M NARENDRA and K.O'NEILL** *"Feature subset selection in remote sensing"* **Canadian Journal of Remote sensing,** (1978).

- [HANA-73] **HANAKATA** *" Feature selection for compact pattern description""*, **1st International Joint Conference on Pattern Recognition, Washington,**(30 Oct - 1 Nov 73), 416-22.

- [INOU-87] **K.INOUE; K.KIMURA** *"A method for calculating the perimeter of objets for automatic recognition of circular defects"*. **N.D.T. Internationa,** (1987), Vol 20, 225-230.

- [JAIN-87] **JAIN and J.V. MOREAU** *"Bootstrap Technique in Cluster Analysis"* **Pattern Recognition** , (1987), Vol:20 ; N°5, 547-568 A.K.

- [KITTL-88] **J.KITTLER** *"Optimality of reassignement rules in dynamic clustering"*., **Pattern Recognition,** (1988), Vol:21 ; N°2, 169-174 .

- [KOU-86] **Weidong KOU and Zheng HU** *"Fast search algoritms for vector quantization"*., I.E.E.E, (1986)

- [KUSI-87] **A. KUSIAK and W.S. CHOW** *"An efficient cluster identification algorithm"*., **IEEE Transaction. Systems Man and Cybernetics,** (Juillet-Août 87),Vol: SMC 17; Fasc 4, 696-699

- [LACH-77] **G.J McLACHLAN** *"A note on the choise of a weighting function to give n efficient method for estimating the probability of mis--classification"*. **Pattern Recognition,** (Oct 77), Vol 9 n°3.

- [LEVR-89] **E.LEVRAT***"Applications de la théorie des ensembles flous à l'amélioration et à la segmentation d'images monochromes"*. **Thèse de l'Université de Nancy I,** (10 Juillet 89).

- [LEVR-89] **E.LEVRAT, BOMBARDIER, SIMON, BREMONT.** *"Applications de la théorie des ensembles flous en rehaussement de contraste"*. **Cercle Français de Microscopie Quantitative : Etude de la prolifération, Méthodologie, Analyse d'imageen pathologie cancéreuse, M.R.E.S., Paris,** (14-15 Décembre 89).

- [LIND-80] **Y. LINDE, A. BUZO and R.M.GRAY.** *"An algorithm for vector quantizer desing"*. **IEEE Transaction ou communication,** (1980), Vol COM-28, n°1, 84-95.

- [LOPE-85] **LOPEZ DE MANTARAS,** *"Self-learning pattern classification using a sequential clustering technique"*., **Pattern Recognition,**(1985), Vol 18 n°3-4 R.

- [LOPE-88] **LOPEZ DE MANTARAS and L. VALVERDE** *"New Result in Fuzzy Clustering Based on the Concept of Indistinguiishability Relation"*.,**IEEE Transaction on pattern analysis and machine intelligence,** (Sept 88), Vol: 10 n°5.R.

- [LUCI-84] **LUCIANO V. DUTRA and NELSON D.A.MASCARENHAS** *" Some experiments with spatial feature extration methods in multispectral classification"*., **International Journal of Remote Sensing,** (1984), Vol: 5 n°2, 303-313.

- [MALI-83] **MALI, B.B CHAUDHURI & D. DUTTA MAJJUMDER** *"Perfomance bound of Walsh-Hadamard transform for feature selection and compression and some related fast algorithms"*.,**Pattern Recognition Letters,** (Oct 83), Vol 2 .P.C, 5-12.

- [MALI-87] **W.MALINA** *"Some multiclass Fisher feature selection algorithms and their comparaison with Karhunen -Lœve algorithm"*., **Pattern Recognition Letters,** (Dec 87), Vol:6, Fasc:5; 279-285.

- [MARW-83] **MARWAN JAMIL MUASHER and D.LANDGREBE,** *"The K.L Expansion as an effective feature ordering technique for limited training sample size"*., **IEEE Transactions on Geoscience and Remote Sensing** , (Oct 83) Vol: GE 21, n°4, 438-441.

- [MARW-83] **MARWAN JAMIL MUASHER and D.LANDGREBE** *"Feature selection with limited training samples"*, **Pattern Recognition,** (1983).

- [MUCC-71] **MUCCIARDI and GOSE** *" A comparaison of seven techniques for choosing subsets of pattern recogniyion properties"*.,**IEEE Transaction on Computers,** (Sept 71), Vol:C-20, 1023-31.

- [NARE-77] **P.M. NARENDRA and K.FUKUNAGA** *" A Branch and Bound algorithm for feature subjet selection"*. **I.E.E.E. Transaction on Computers,** (1977), Vol C. 26, 917-922.

- [NELS-68] **NELSON LEVY** *" A dynamic programme approach to the selection of pattern feature "*. **I.E.E.E. Transaction Systems Science and Cybernetics,** (Juillet 1968), Vol SSC4, 145-151.

- [NISH-87] **N.NISHIYAMA** *"Extraction of Nth-order feature of line patterns considering redundant information"*.,**Systems & Computers Japan (USA),** (Avril 87), Vol: 18 Fasc:4, 76-85.

- [PASC-85] **PASCAR and COHEN** *" Comparaison of then feature selection"*, **14th Convention of electrical and Image Electronic Engineers in Israël Proceeding,** (26-28 Mars 85), 4-4-2/1-5,Tel Aviv .

- [PAU-77] **L.F.PAU** *"An adaptive signal classification procedure. Application to aircraft engine condition monitoring "* , **Pattern Recognition,** (Oct 77), Vol 9 n°3.

- [PEDR-85] **W. PEDRYCZ** *"Algorithms of fuzzy clustering with partial supervision"*., **Pattern Recognition Letters,** (1985), Vol: 3, 13-20.

- [SEGE-84] SEGEN *" Feature selection and constructive inference"*, **7th International Conference on Pattern Recognition,** (30 Juill-2 Août 84), Vol 2, 1344-46, Montréal .

- [SONG-88] SONG SHUWU and QIN JIAMEI *"Picture geometrical parameter measurement with computer"*., **Kexue Tongbao,** (Jan 88), Vol 33; Fasc:2, 159-165.

- [SU HU-86] SU HUAN-YU*"Utilisation de la quantification vectorielle en reconnaissance de la parole continue"*.. **I.R.I.S.A ,Campus de Beaulieu 35042 RENNES Cedex,** (1986).

- [SWAI-73] P.H. SWAIN and R.C KING *"Two effective feature selection criteria for multispectral remote sensing"* .LARS **Information note 042673 Purdue University West Lafayette. Indiana,** (1973).

- [SWAI-77] P.H. SWAIN and H.HSUKA *"The decision tree classifier : design and potential"*., **IEEE Trans. Geoscience and Electronics,** (1977), Vol 15, fasc 3,142-147.

- [TOUM-87] J.J. TOUMAZET *" Traitement de l'image sur micro-ordinateur"* **édition Sybex,** (1987).

- [TUNS-75] K.W. TUNSTALL *"Recognizing patterns: are there processes that precede feature analysis?"* , **Pattern Recognition,** (Juin 75), Vol: 7, N°1-2.

- [WANG-86] P.S.P. WANG and X.W.DAI*"Un algorithme d'inférence de Grammaires de Tableaux libres de contextes"* , **Deuxième Colloque image,** (Avril 86), 804-809, Nice.

- [WANG-87] Z.Y.HE , G. Q. WEI and T.J. WANG *"A new algorithm for fast computation of moments of binary picture"* , **IEEE Pacific Rim Conference on Communication Computers and Signal Proccedings (Cat n° 87CH2482-8) Victoria .BC Canada ,** (4-5 Juin 87), 179-182.

- [WRIG-77] W.E. WRIGHT *"Gravitational clustering"* **Pattern Recognition** , (Oct 77), Vol 9 n°3.

- [YOUN-83] D.YOUNG & P.L.ODELL *"A formulation and comparaison of two linear feature selection techniques applicables to statistical classification"*., **Pattern Recognition,** (1983), Vol:17,331-37.

- [YOUN-85] D.YOUNG & P.L.ODELL & V.R.MARCO *"Optimal linear feature selection for a general class of statistical pattern recognition models"*. , **Pattern Recognition Letters,** (Mai 1985), Vol: 3,161-165.

- [ZAKA-86] M.F ZAKARIA L.J.A. ZSOMBOR-MURRAY and J.M.H.M VAN KESSEL *" Fast algorithm for the computation of moment invariant."* , **Pattern Recognition,** (1986), Vol: 20, 639-647.

- [ZHENG-86] ZHENG HUANG and ZUYIN CHEN *"A new clustering by bipartite graph theory"*. **I.E.E.E,** University China , (1986).

NOM DE L'ETUDIANT : Mme BENZIDIA née MIRZAQ Souad

NATURE DE LA THESE : Doctorat de l'Université de NANCY I en Automatique

VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le 15 JUIN 1990 n° 1213

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

UNIVERSITE NANCY I
LE PRESIDENT
M. BOULANGE

RESUME

La sélection de paramètres est l'une des étapes fondamentales d'un processus de reconnaissance de forme. En effet le choix des paramètres contribue fortement à la qualité du processus de reconnaissance.

Dans ce mémoire, après avoir retracé les différentes étapes d'un processus de reconnaissance de forme d'images, la première partie donne un résumé des méthodes de traitements de mise en forme, ainsi que la description de mesures géométriques d'objets dans l'image.

Dans la deuxième partie, différentes méthodes de classification sont décrites: méthodes de réallocation, méthodes hiérarchiques. Trois grandes familles de méthodes de sélection de paramètres sont ensuite présentées : les méthodes séquentielles, les méthodes d'analyse en composantes principales et les méthodes à programmation dynamique.

La troisième partie de ce mémoire est consacrée à la description de la partie expérimentale. Nous présentons en détail l'ensemble des différents programmes : création d'images binaires, mesures de paramètres géométriques, normalisation des valeurs, sélection, classification et reconnaissance. Pour la partie sélection de paramètres, trois méthodes à programmation dynamique répondent le mieux au problème et aux objectifs que nous nous sommes posés: la méthode du Branch and Bound, la méthode de Cheung et la méthode de Foroutan et Sklansky.

Dans la dernière partie, différents résultats obtenus au niveau de la classification et de la reconnaissance sont donnés puis discutés.

Les résultats obtenus confirment la qualité des méthodes à programmation dynamiques.

Mots clés : Reconnaissance de forme, Traitement d'images, Classification, Sélection de paramètres, Programmation dynamique