

# DÉTECTION DE DÉFAUTS TEMPS RÉEL SUR DES OBJETS A GÉOMÉTRIE COMPLEXE : ÉTUDE PAR SVM, BOOSTING ET HYPERRECTANGLES

## REAL-TIME FLAW DETECTION ON COMPLEX PART: STUDY OF SVM, BOOSTING AND HYPERRECTANGLE BASED METHOD

S. Bouillant, J. Mitéran, M. Paindavoine

Le2i

Aile des Sciences de l'ingénieur

Université de Bourgogne

BP 47870 21087 Dijon

[miteranj@u-bourgogne.fr](mailto:miteranj@u-bourgogne.fr)

Tel/fax : 03 80 39 68 55/59 10

### Résumé

Nous présentons dans cet article une application complète des « Support Vector Machine » au contrôle qualité par vision artificielle de pièces à géométrie complexe. Nous précisons le cadre pratique dans lequel s'effectuent les opérations, la nature des défauts à détecter ainsi que les techniques d'extraction des paramètres discriminants. Nous présentons ensuite les trois méthodes de classification utilisées. Nous définissons le protocole d'apprentissage, ainsi que la méthode de recherche des paramètres optimum du classifieur. Nous comparons les résultats obtenus à partir d'un espace de description défini *a priori* ainsi que ceux issus d'une sélection de paramètres via un algorithme séquentiel.

Mots Clef : Classification, SVM, Hyperrectangle, Boosting, analyse de formes, contrôle qualité, image.

### Abstract

We present in this paper our works on the classification of industrial parts based on "Support Vector Machine" method. We present the practical frame in which are made the operations, flaws types to detect as well as feature extraction techniques. Then we introduce the three classification techniques we

implemented. We explain our learning method and how we obtain optimum classifier parameters. We compare the results obtained using feature space based on *a priori* knowledge and on space extracted from sequential selection algorithm.

Keywords: Classification, SVM, Hyperrectangle, Boosting, quality control, image processing, pattern analysis.

## 1 Introduction

Nous présentons dans cet article une application complète des « Support Vector Machine » (SVM) au contrôle qualité par vision artificielle de pièces à géométrie complexe. La détection de défauts d'aspect et de dimension sur des objets en défilement à cadence élevée (10 pièces par seconde) demande la mise en place d'un système de contrôle robuste et rapide. Nous proposons ici l'étude d'une machine de vision permettant l'analyse par classification d'une forme relativement complexe (spiralage) afin d'améliorer la production, en détectant les défauts d'aspect et de géométrie qui peuvent l'affecter.

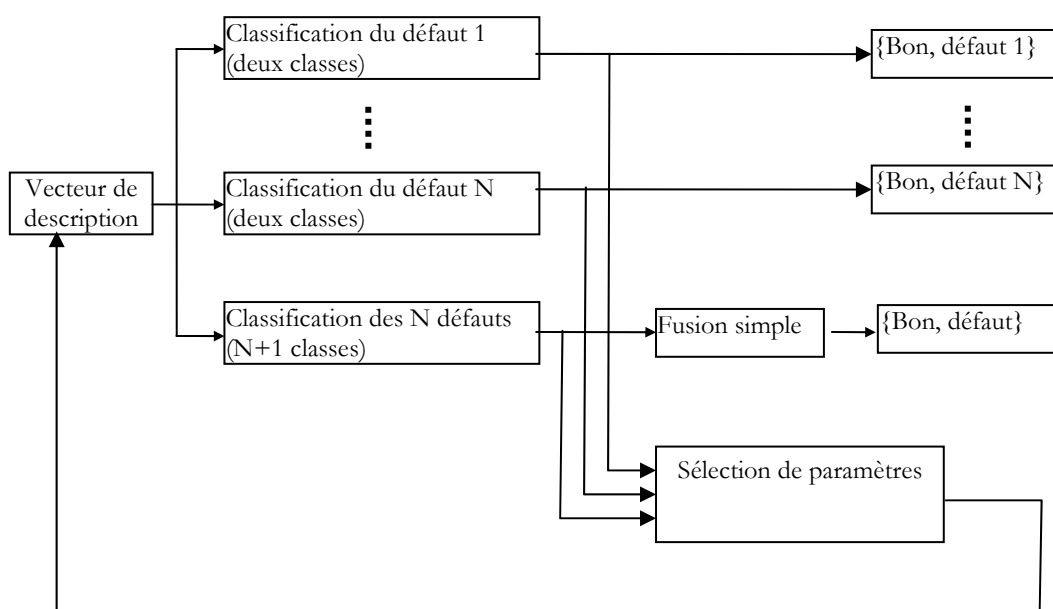
Actuellement, l'inspection est réalisée manuellement par des opérateurs, à un coût élevé. L'augmentation de productivité est donc le but initial des travaux présentés ici. Les spécifications industrielles demandent une vitesse de traitement minimale de 10 pièces par seconde. De plus, la qualité des pièces produites doit être au moins maintenue au niveau actuel. Ce problème peut être résolu en utilisant un système de classification basé sur une machine de vision. De nombreux articles [1], [2], [3] décrivent la détection de défauts à l'aide de méthodes de classification. La méthode retenue ici consiste à extraire de chaque point de vue un ensemble de mesures modélisant la partie spiralée dimensionnelles ou d'aspect [4], [5], puis d'extraire des paramètres internes à chaque spire. Ces mesures alimentent le classifieur qui permet de distinguer une pièce bonne d'une pièce défectueuse.

Afin de répondre à ces exigences de qualité tout en respectant les impératifs de traitement en temps réel, trois méthodes de classification, de complexités algorithmiques très différentes, ont été envisagées.

La première est basée sur un découpage de l'espace de description par des hyperrectangles [6], [7], la seconde sur la méthode du Boosting (Adaboost) [8], [9] et la dernière sur les Support Vector Machine [10]. La première méthode est rapide, que ce soit au niveau de l'apprentissage ou au niveau de la décision, et elle a prouvé son efficacité dans plusieurs applications concrètes [11], [12]. Elle présente toutefois des faiblesses au niveau de la généralisation et une grande sensibilité à la présence d'éléments mal déterminés dans la base d'apprentissage. La deuxième méthode est plus robuste, tout en conservant une complexité en phase de décision compatible avec les contraintes de temps réel. Elle présente également l'avantage de pouvoir réaliser automatiquement une sélection de paramètres. L'avantage de la dernière méthode est de présenter un fort pouvoir de généralisation et de très bonnes performances de classification [10]. La fonction de décision est relativement coûteuse en temps de calcul mais peut tout de même être utilisée en

décision haut niveau (10 décisions par seconde). Dans le cas d'une décision au niveau pixel, la frontière peut être modélisée en utilisant une combinaison d'apprentissages entre les SVM et la méthode à base d'hyperrectangles, pour réaliser par exemple une segmentation d'image par classification des pixels [13].

Dans la première partie de cet article, nous présentons la pièce à contrôler, l'ensemble d'acquisition, et les prétraitements qui permettent l'extraction des mesures nécessaires à la caractérisation de l'objet. Dans la deuxième partie, nous rappelons les principes des trois méthodes de classification étudiées. Dans la troisième partie, après avoir défini le protocole d'apprentissage, nous présentons des résultats préliminaires de classification obtenus en recherchant les paramètres optimums du classifieur SVM. Nous montrons ensuite qu'il est encore possible d'améliorer les résultats, tant en terme de vitesse que de performances de classification, en regroupant les classes et en sélectionnant les paramètres de classification les plus discriminants.



**Figure 1 : Méthodologie utilisée.**

L'approche menée nous permet de comparer les performances des trois méthodes de classification en envisageant deux approches applicables au contrôle qualité :

Dans le premier cas, les pièces sont réparties simplement en deux classes {défaut, bon}, conduisant à une classification globale,

Dans le deuxième cas, les pièces sont réparties en N classes {défaut\_1, défaut\_2, ..., défaut\_N, bon}. L'ensemble des stratégies utilisées lors de cette étude est ainsi résumé Figure 1.

## 2 Présentation du problème et extraction des paramètres

### 2.1 Système d'acquisition et prétraitements

La pièce est composée d'une partie spiralée et d'une partie longiligne appelée « pattes » (Figure 2). Des travaux précédents nous ont conduits à choisir un système d'éclairage à diodes rouges en régime pulsé, commandé en courant, éclairant la partie externe du spiralage [14]. Cet éclairage a été réglé de manière à mettre en évidence les défauts tout en permettant des mesures dimensionnelles relativement aisées (pièce claire sur fond noir). La partie optique du système d'acquisition est composée d'une caméra 1300x1030 pixels et d'un objectif télécentrique.

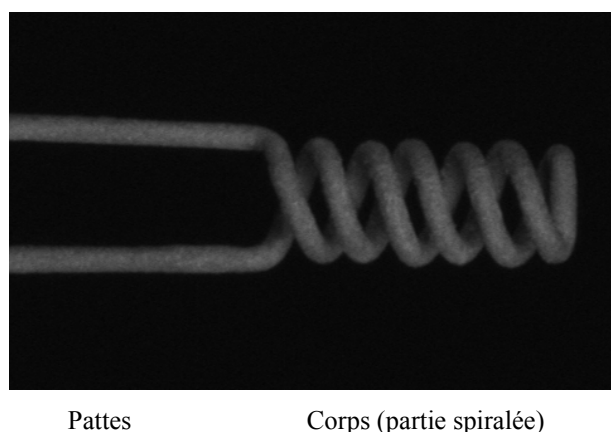


Figure 2: Partie à contrôler.

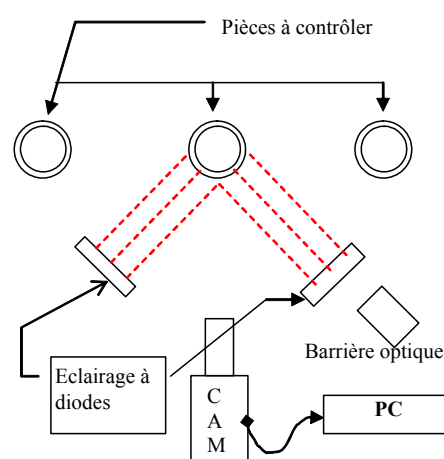
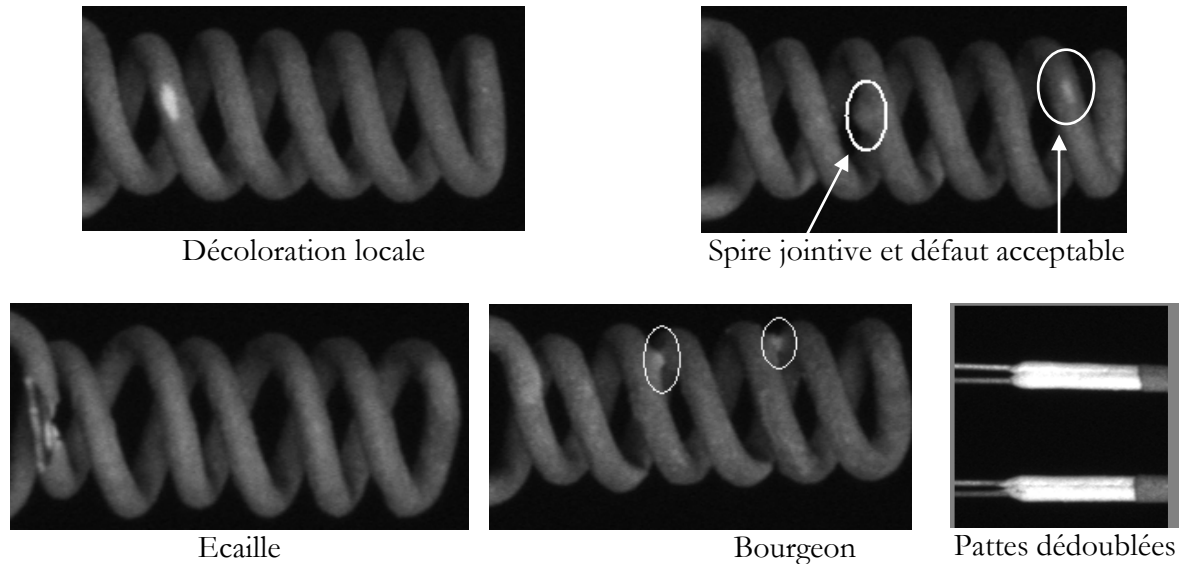


Figure 3 : Chaîne d'acquisition.

L'éclairage et la prise d'image sont synchronisés avec le passage d'une pièce dans le champ via une barrière optique. Ce système permet l'acquisition de 10 images 1300x1030 à la seconde. (Figure 3). L'ordinateur utilisé est un biprocesseur Pentium III. Un seul point de vue a été utilisé pour cette étude. La mise en place finale du système de contrôle demandera plusieurs prises de vues (3 à 4) afin de couvrir l'ensemble du filament.

### 2.2 Défauts à détecter

Les défauts à détecter sont au nombre de 50, regroupés en deux grandes familles : dimensionnelle et d'aspect. La détection des défauts dimensionnels débordant du cadre de cette étude ne sera pas présentée ici. En effet, ces mesures seront simplement comparées à des seuils de référence fournis par l'industriel partenaire de ce projet et ne nécessitent pas l'utilisation d'un classifieur. La seconde famille de défauts, dits d'aspect, est composée de pièces ayant des irrégularités de textures, des manques ou des surplus de matière (pouvant induire des modifications dimensionnelles), des fissures... Ces défauts sont au nombre de 20 répartis sur la totalité de la pièce (pattes, corps). Parmi ces défauts, 5 ont été classés prioritaires par l'industriel.



**Figure 4 : Défauts d'aspect**

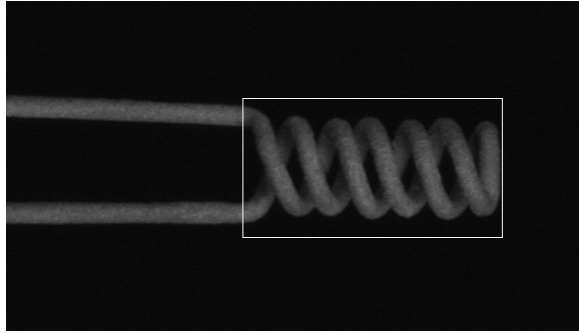
Seuls ces 5 défauts seront présentés dans cet article. Afin d'effectuer une classification de ces pièces, des ensembles d'apprentissage et de tests ont été construits à partir de 1606 pièces bonnes et d'un total de 245 pièces défectueuses réparties en 5 types de défauts (Tableau 1). On peut noter d'emblée qu'une des difficultés principales de ce projet provient de la forte variabilité des pièces à contrôler, et surtout de la présence de nombreux défauts dits « acceptables » parmi les pièces bonnes.

**Tableau 1: Nombre d'éléments**

<i>Classe</i>	<i>Nb. d'éléments disponibles</i>
Bon (Classe B)	1606
Décoloration locale (Classe D1)	18
Bourgeon (Classe D2)	42
Spires jointives (Classe D3)	24
Badigeon écaillé (Classe D4)	78
Pattes dédoublées (Classe D5)	83

### 2.3 Extraction des paramètres de classification

La vitesse de défilement étant de 10 pièces par seconde, il est nécessaire de limiter les traitements permettant l'extraction des paramètres de classification. La chaîne complète de traitement (de la détection à l'extraction des paramètres) est composée de 4 étapes. La première étape est la localisation de la pièce dans l'image (Figure 5).



**Figure 5: Localisation de la pièce dans la scène.**

Le but de cette étape est double : elle permet de limiter les traitements ultérieurs à une région d'intérêt rectangulaire, centrée sur le corps, de dimension réduite par rapport à l'image, réduisant ainsi les temps de calcul. De plus elle permet de détecter un défaut de type particulier se traduisant par une déformation de la partie spiralée. Les dimensions du corps étant connues, la présence de ce type de défaut est diagnostiquée par le dépassement du corps de la « région d'intérêt ». Cette localisation est effectuée grâce au calcul du centre de gravité de l'image, complété par des connaissances *a priori* sur les dimensions du spirilage.

La deuxième étape consiste à segmenter l'image afin de différencier les spirilages externe et interne ainsi que le fond. Du fait d'une très grande variation de textures, de teintes générales des pièces, un simple seuillage ne permet pas d'obtenir une segmentation robuste, et ce, malgré la stabilité de l'éclairage, composé de diodes en régime pulsé alimentées par une source stabilisée. Nous utilisons donc l'algorithme de Wen [15] appliqué d'une part sur l'image de luminance et d'autre part sur l'image résultant d'un gradient de Sobel. Cet algorithme a été choisi pour son adaptabilité intrinsèque aux variations d'éclairage ou de teinte globale du filament, ainsi que pour sa complexité algorithmique compatible avec la cadence de traitement imposée. La segmentation doit permettre la différenciation entre le spirilage externe et interne. Etant donné la petite taille de la pièce et le grandissement utilisé, la profondeur de champ est réduite, ce qui permet de dissocier ces deux régions par une mesure de gradient (la partie externe de la spire est plus nette et partie interne de la spire est plus floue). Une combinaison logique entre les différentes populations de haute intensité de ces deux images (luminance et gradient) permet donc de segmenter l'image.

Toutefois, comme on peut le voir sur la **Figure 6**, cette segmentation n'est pas parfaite au sens où elle ne décrit pas la totalité des spires supérieures. Ces régions segmentées sont alors utilisées comme « graines » pour un algorithme itératif de croissance, basé sur un opérateur morphologique simple de dilatation. A chaque étape, un OU exclusif est réalisé avec une image binaire de l'ensemble du filament (**Figure 7**). Du fait de la forme quasi convexe de la spire, la surface résultant de cette opération logique décroît, atteint un minimum, puis recommence à croître lorsque la dilatation est trop importante. La détection de ce minimum de surface permet donc d'arrêter la croissance de la région.



Figure 6: Combinaison des images Sobel et Luminance (**G**).



Figure 7: Image binaire complète (**B**).



Figure 8: Segmentation finale (**R**).

Cette troisième étape de segmentation permet d'augmenter la surface détectée à l'intérieur de chaque spire (Figure 8).

Si **O** représente l'image originale, la phase de segmentation peut se résumer sous la forme suivante

Calculer **B**, binarisation de **O**, avec un seuil permettant de supprimer uniquement le fond,

Calculer **O<sub>w</sub>**, transformation de **O** à l'aide de l'algorithme de Wen,

Calculer **L**, gradient de Sobel de **O**, et **L<sub>w</sub>**, transformation de **L** à l'aide de l'algorithme de Wen,

Calculer **G**, combinaison logique de **L<sub>w</sub>** et **O<sub>w</sub>** tel que

$$G(i,j)=1 \text{ si } A_w(i,j)>s_1 \text{ et } L_w(i,j)>s_2 \text{ et } 0 \text{ sinon, pour tout pixel de coordonnées } (i,j), s_1 \text{ et } s_2 \text{ étant des seuils déterminés empiriquement}$$

Initialiser  $t$  à 1, définir le nombre d'itérations maximum  $t_{\max}$ , copier **G** dans **D** et dans **R**,

Faire tant que  $t < t_{\max}$

Calculer **X**, combinaison logique de **D** et **B** tel que

$$X(i,j) = D(i,j) \text{ XOR } B(i,j)$$

Calculer  $k_t$ , nombre de pixels dont la valeur est différente de 0 de **X**,

Si  $k_t > k_{t-1}$  : arrêt

Calculer **D**, dilatée de **D**,

$k_{t-1} = k_t$  et copier **D** dans **R**,

Fin tant que

Le résultat est dans l'image binaire **R**.

En pratique, la méthode de croissance de région est appliquée séparément pour chacune des spires trouvées dans G, afin que le nombre d'itérations s'adapte automatiquement aux variations locales du filament.

La stabilité et la répétabilité de l'algorithme ont été démontrées grâce à de nombreuses mesures effectuées sur l'ensemble d'apprentissage, et ce aussi bien en acquisition statique (mesures géométriques répétées sur 1000 prises de vues d'un même filament) qu'en dynamique (mesures géométriques effectuées sur une centaine de passage d'un même filament). Les filaments trop déformés pour que l'algorithme soit applicable sont détectés en amont par un simple test de présence à l'intérieur d'une boîte englobante dont les dimensions ont été définies par le partenaire industriel.



Figure 9 : Etiquetage des spires.

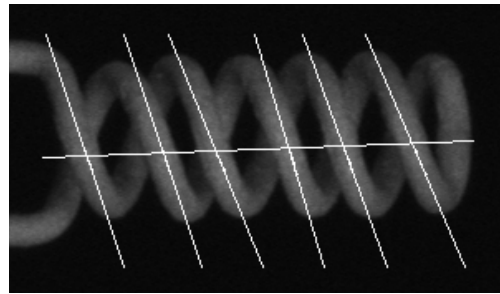


Figure 10 : Exemple d'extraction de paramètres géométriques.

Pour identifier individuellement chaque spire, il est ensuite nécessaire d'effectuer la quatrième étape, un étiquetage en composantes connexes. La différenciation des spires est effectuée en les classant selon l'ordre décroissant de l'abscisse de leur centre de gravité (ici de la spire la plus foncée à la spire la plus claire - Figure 9). Cette étape permettra d'attribuer les paramètres extraits non plus à la totalité du corps mais à une spire déterminée. Durant cette étape, il est nécessaire de filtrer le résultat de l'étiquetage, en fonction de la surface des zones détectées. En effet, le seuillage adaptatif génère des zones non représentatives. Afin de les éliminer, les zones de surfaces inférieures à un seuil donné (de l'ordre de 50) sont attribuées au fond.

La dernière étape est l'extraction des paramètres sur cette image segmentée. Des mesures dimensionnelles sont effectuées. Ces paramètres sont directement issus du cahier des charges dimensionnel fournis par l'industriel. Des modélisations linéaires sont menées pour chaque spire et pour la totalité du corps. De ces modélisations nous extrayons des paramètres décrivant la géométrie du filament :

- Nombre de spires détectées ;
- Nombre de pattes détectées ;



- Inclusion de la partie spiralée dans un rectangle de dimensions connues ;
- Coordonnées des centres de gravité de chaque spire;
- Distance entre les centres de gravité de chaque spire (distance inter centre de gravité) ;
- Coefficient directeur de la modélisation linéaire de la partie spiralée (ligne horizontale Figure 10) ;
- Coefficients directeurs de la modélisation linéaire de chaque spire, du centre et des bords (lignes obliques Figure 10 - ici le centre de chaque spire) ;
- Moyenne des distances euclidiennes inter spires (Calculée à partir des mesures de distance entre le bord droit de la spire  $i$  et le bord gauche de la spire  $i+1$ ) ;
- Ecart types des distances euclidiennes inter spires.

L'utilisation de paramètres géométriques est justifiée par la présence de défauts tels que la géométrie même de la pièce est modifiée. Le défaut de type spires jointives en est un exemple (Figure 4).

À ces paramètres purement géométriques viennent s'ajouter des paramètres décrivant des défauts dits « d'aspect » (décoloration locale, manque de matière sur le spiralage, Figure 4). Parmi ces paramètres, on compte :

- L'aire de chaque spire –  $A_s$  ;
- Niveau de gris moyen de la partie spiralée (incluant toutes les spires) –  $\mu_s$  ;
- Niveau de gris moyen des spires –  $\mu'_s$  ;
- Population de pixels dont la luminance est supérieure à un seuil fonction de la moyenne –  $P_\mu$  ;
- Calcul des moments statistiques d'ordre 1 et 2 des niveaux de gris des pixels de chaque spire, etc.

De plus, face à la spécificité de certains défauts, les décolorations notamment, des paramètres plus spécifiques ont été pris en compte. Parmi ceux-ci on compte des paramètres d'analyse des luminances dans des régions spécifiques des spires. Ainsi, pour chaque spire segmentée  $S_i$ , sont définies des régions  $P_i$  (resp.  $Q_i$ ) de pixels de luminance  $L_{ij}$  telles que :

$$L_{ij} > n \cdot \mu_i \text{ (resp. } L_{ij} > m \cdot \mu_i \text{)}.$$

Le couple  $\{n, m\}$  a été défini empiriquement avec  $n < m$ . L'utilisation de ce couple  $\{n, m\}$  permet d'adapter les seuils de détection en fonction de la production. En effet, il arrive que la luminance des pièces varie d'une façon importante et ces paramètres permettent l'adaptabilité de ces paramètres à ces variations. L'utilisation de paramètres géométriques peut paraître inapproprié dans le cas de la détection de défauts de textures, mais l'expérience a montré qu'ils étaient nécessaires à la bonne détection de pièces très détériorées.

Ces paramètres ont été validés tout au long de cette étude, en collaboration avec le partenaire industriel afin de garantir une description des défauts la plus fiable et répétable possible. Des détails concernant ces paramètres sont donnés Tableau 2.

**Tableau 2 : Quelques paramètres de description.**

<i>Description du paramètre</i>	<i>Taille du paramètre</i>
Nombre de spires détectées - N	1
$A = \text{Max}(A'_i)$ avec $A'_i$ l'aire de la région $P_i$	1
$\text{Max}(S'_i)$ avec $S'_i$ la somme des luminances des pixel appartenants à $P_i$	1
$C = \text{Max}(A''_i)$ avec $A''_i$ l'aire de la région $Q_i$	1
Nb. de pattes détectées - P	1
Aire de chaque spire - $A_{\text{spire}}$	7
Luminance moyenne par spire - $\mu_{\text{spire}}$	7
Ecart type de la luminance par spires - $\sigma_{\text{spire}}$	7
Dépassement d'une boite de dimensions données - D	1
Abscisse à l'origine des droites modélisant le corps et les pattes - $B_{\text{corps}} - B_{\text{pattes}}$	3
Coeff. Dir. de la modélisation linéaire des bords et du centre de chaque spire - $\alpha_{G_{\text{spire}}} - \alpha_{D_{\text{spire}}} - \alpha_{C_{\text{spire}}}$	3x7
Moyenne des distances inter spire - $\mu_{\text{DIS}}$	6
Ecart type des distances inter spire - $\sigma_{\text{DIS}}$	6
Coeff. directeur des droites modélisant les bords sup. et inf. des pattes - $\alpha_{\text{Spatte}} - \alpha_{\text{Ipatte}}$	2x2
Distance inter centre de gravité de chaque spire - $\Delta_{\text{CDG}}$	6
Différence d'abscisse des points supérieurs de chaque spire - $\Delta x_{\text{spire}}$	6
Différence d'ordonnée des points supérieurs de chaque spire - $\Delta y_{\text{spire}}$	6
Différence d'abscisse des points inférieur de chaque spire - $\Delta x_{\text{spire}}$	6
Différence d'ordonnée des points inférieur de chaque spire - $\Delta y_{\text{spire}}$	6
Différence des coefficients directeurs de chaque spire - $\Delta \alpha_{\text{spire}}$	6

## 2.4 Analyse des paramètres

Afin d'étudier la pertinence des paramètres choisis, nous avons mis en œuvre une analyse en composantes principales (ACP) donc le but est de condenser les données originelles en de nouveaux groupements de façon à ce qu'ils ne présentent pas de corrélation entre eux et soient ordonnés en terme de pourcentage de variance apportée par chaque composante [16], [17]. Ainsi, la première composante principale contient les informations relatives à la variance maximale, la deuxième contient les informations relatives à la variance suivante. Le processus est répété jusqu'à l'obtention de la n-ième et dernière composante principale. Les pertes d'informations diminuent d'une étape à la suivante.

En sélectionnant les p premières valeurs propres déterminées, il est possible de réduire le nombre de paramètres. Toutefois, dans le cas de cette étude, cette analyse n'a donné aucun résultat probant du fait d'une très forte corrélation entre certains paramètres (la structure « élastique » du filament étant à l'origine de cette corrélation). De plus, il apparaît qu'il existe des groupes de paramètres qui devraient être considéré comme une entité propre et non une somme de paramètres. Il est nécessaire, par exemple, de ne pas dissocier le paramètre « aire de la spire 1 » des paramètres « aire de la spire i » et ce pour toutes les autre

spires i. C'est pour cette raison que d'autres méthodes de sélection de paramètres ont été mises en oeuvre et décrites dans le paragraphe 4.

### 3 Méthodes de classification étudiées

#### 3.1 Généralités

Comme nous l'avons précisé dans l'introduction, seules trois méthodes de classification ont été abordées dans cette étude. Cette limitation, par rapport aux très nombreuses méthodes présentes dans la littérature que l'on pourra trouver résumées dans [5], provient essentiellement des contraintes de temps (de décision mais aussi de développement de l'application) liées au projet industriel. Les critères qui ont guidé notre choix sont essentiellement :

- la complexité algorithmique de la phase de décision, qui doit être compatible avec le temps réel,
- la vitesse d'apprentissage (importante pour la mise au point de l'application, notamment la phase de sélection de paramètres)
- le nombre de paramètres à régler,
- les performances intrinsèques (pouvoir de généralisation),
- la solidité des bases théoriques.

Les trois premiers critères sont sans aucun doute remplis par la méthode basée sur les Hyperrectangles, que nous rappellerons ci-dessous. Les méthodes de type Boosting ou basées sur les SVM répondent mieux encore à tous ces critères. Cependant, leur complexité est aussi plus importante. Les SVM présentent notamment comme avantage, par rapport aux réseaux de neurones par exemple, de n'avoir que peu de paramètres à choisir pour obtenir une bonne classification, et ont été utilisés avec succès dans de nombreux cas réel [18], [19], [20]. Considérées par de nombreux auteurs comme faisant partie des méthodes connexionnistes, bien que n'ayant aucun lien avec le modèle biologique du neurone, les SVM présentent un certain nombre d'avantages par rapport aux réseaux de neurones classiques et à l'algorithme de la rétropropagation du gradient, difficile à paramétrer et parfois instable [21]. Le lecteur pourra trouver une bibliographie très complète sur <http://www.kernel-machines.org/>. Le boosting, qui comme les SVM peut être vu comme une méthode basée sur la maximisation des marges entre classes et frontières de décision, a souvent été appliqué dans le cadre de la recherche de documents, de textes [22], [23], mais encore relativement peu en contrôle qualité. Le lecteur pourra également se référer à <http://www.boosting.org/> pour une bibliographie très large sur le sujet.

### 3.2 Classification par hyperrectangles

Cette méthode consiste à déterminer les enveloppes des différentes classes par méthode géométrique [6]. À partir d'un élément appartenant à une classe  $C_i$ , on recherche le premier voisin appartenant à une classe différente et ce dans toutes les directions parallèles aux axes de l'espace de descriptions (s'il est de dimension  $n$ , il y a  $2.n$  directions). La frontière entre les deux classes est déterminée en fonction d'une contrainte donnée par la présence des éléments de classes opposées à la classe en cours, ce qui d'une certaine manière est similaire à la contrainte permettant de résoudre le problème des SVM. C'est la distance du « max » qui est utilisée pour déterminer le voisin le plus proche, ce qui permet d'obtenir des frontières parallèles aux axes de l'espace de description. Un hyperrectangle ou polytope de contrainte est ainsi généré pour chaque échantillon de la base d'apprentissage. Ces hyperrectangles sont ensuite fusionnés afin d'en optimiser le nombre. La fusion respecte la contrainte de non recouvrement entre hyperrectangles de classes différentes. La phase de décision se limite donc à la vérification de la fonction de décision suivante :

$$C(P) = \sum_{i=1}^{nh} \prod_{k=1}^N (x_k > a_{ik}) \cdot (x_k < b_{ik}) \quad \text{Eq. [1]}$$

où  $a_{ik}$  et  $b_{ik}$  sont les limites des hyperrectangles déterminés pendant la phase d'apprentissage. Le produit et la somme sont égaux au « ET » et au « OU » logiques.  $N$  est la dimension de l'espace de description et  $nh$  est le nombre d'hyperrectangles obtenus pour la classe considérée.

Cette méthode s'applique particulièrement bien aux distributions multimodales. On peut la comparer à un réseau de type RBF pour lequel on aurait utilisé la distance du « max ». Elle a notamment été étudiée dans [24]. Elle présente l'avantage de pouvoir être implantée sur support logiciel aussi bien que sur support matériel, permettant une décision parallélisée extrêmement rapide (de l'ordre de 10 à 20ns par décision) [25] .

### 3.3 Boosting (Adaboost)

Le principe général du Boosting est de combiner linéairement des résultats de classifieurs dits « faibles » afin de construire un classifieur « fort ». Il a été formulé de la manière suivante par Y. Freund en 1995 [8]:

«Tout algorithme prédictif capable, pour toute distribution quelconque, d'apprendre avec une certaine confiance et une erreur de classification inférieure à  $1/2$  peut être transformé en un algorithme d'apprentissage avec une confiance aussi grande et une erreur de classification aussi petite que désirée. ».

En d'autres termes, un modèle prédictif (dit classifieur faible) donnant de meilleurs résultats qu'un simple pile ou face (une chance sur deux de se tromper en prédisant le résultat du tirage) peut être combiné de façon à être aussi performant que nécessaire. Même si le classifieur original est linéaire, de part la

construction de la fonction de décision, il sera à même, après plusieurs itérations, de modéliser une fonction non linéaire. La complexité globale de ce type de méthode découle directement de celle du classifieur initial. Parmi les algorithmes dérivés de ce concept, nous nous sommes intéressés plus particulièrement celui d'AdaBoost, rappelé ci-dessous :

Base d'apprentissage de  $p$  échantillons  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_p, y_p)\}$ , nombre maximal d'itérations  $T$

Initialise les poids de chaque échantillon :  $w_i^{(t)} = 1/p$  pour  $i=1, \dots, p$

Faire pour  $t=1, \dots, T$

Entraîner le classifieur faible à partir des échantillons pondérés et obtenir l'hypothèse

$$h_t : \mathcal{X} \rightarrow \{-1, +1\}$$

Calculer l'erreur pondérée  $\varepsilon_t$  de  $h_t$  :

$$\varepsilon_t = \sum_{i=1}^p w_i^{(t)} I(y_i \neq h_t(\mathbf{x}_i))$$

Calculer les coefficients  $\alpha_t$

$$\alpha_t = \frac{1}{2} \log \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

Mettre à jour les poids

$$w_i^{(t+1)} = \frac{w_i^{(t)}}{Z_t} \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}$$

Où  $Z_t$  est une constante de normalisation  $Z_t = 2\sqrt{\varepsilon_t(1 - \varepsilon_t)}$

Arrêt si  $\varepsilon_t = 0$  ou  $\varepsilon_t \geq \frac{1}{2}$  et choisir  $T=t-1$

Fin pour

Résultat :  $y(\mathbf{x}) = \text{signe} \left( \sum_{t=1}^T \alpha_t h_t(\mathbf{x}) \right)$

Le principe est donc, à chaque itération, d'augmenter le poids des éléments mal classés et de diminuer celui des éléments bien classés. Le classifieur se focalise ainsi progressivement vers les éléments les plus difficiles à classer.

L'erreur d'apprentissage  $\varepsilon_t$  d'un classifieur à l'itération  $t$  vaut :

$$\varepsilon_t = \frac{1}{2} - \zeta_t$$

où  $\zeta_t$  tend vers 0 dans le cas d'un classifieur faible. L'erreur finale d'apprentissage est non seulement bornée mais décroît de façon exponentielle [5].

En ce qui concerne notre application, l'intérêt de cette méthode réside essentiellement dans la relative faible complexité de la fonction de décision associée à ce classifieur, notamment lorsque le classifieur faible est un simple seuil (hyperplan) parallèle à un des axes de l'espace des paramètres. Elle présente également dans ce cas l'avantage d'effectuer automatiquement une sélection des paramètres efficaces, puisqu'un seul paramètre est alors pris en compte dans chaque terme de la somme.

### 3.4 Classification par SVM

Les SVM ont été introduits par Vladimir Vapnik en 1979 [26], et c'est seulement depuis une petite dizaine d'années qu'une large communauté s'est axée sur ce domaine. Il existe différentes familles de SVM en fonction du noyau utilisé.

L'idée de base des SVM est la détermination d'un hyperplan séparateur entre des échantillons appartenant à deux classes distinctes en maximisant la marge interclasse et ce, soit dans l'espace de description dans le cas de l'utilisation d'un noyau linéaire, soit dans un espace de dimension supérieure dans le cas de noyaux non linéaires.

Soit une base d'apprentissage définis par  $n$  couples  $\{x_i; y_i\}$  où  $x_i \in \mathbb{R}^d, y_i = \pm 1$ ,  $x_i$  est le vecteur décrivant le prototype  $i$  dans l'espace de description de dimension  $d$ , et  $y_i$  l'étiquette de la classe de ce prototype.

La détermination de l'équation d'un hyperplan séparant ces deux classes, dans l'espace de description passe par la maximisation de la marge interclasse (si les classes sont linéairement séparables) sous la contrainte traduisant la séparabilité de l'hyperplan, c'est-à-dire, si  $w$  est la normale à cet hyperplan:

$$w \cdot x_i + b \geq 1 \quad \text{pour } y_i = +1 \quad \text{Eq. [2]}$$

$$w \cdot x_i + b \leq -1 \quad \text{pour } y_i = -1 \quad \text{Eq. [3]}$$

soit

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad \text{Eq. [4]}$$

pour tous les points de toutes les classes. Il s'agit de maximiser la marge interclasse,  $\frac{2}{\|w\|}$  sous la contrainte

Eq. [4]. L'utilisation des multiplicateurs de Lagrange ( $\alpha_i$ ) nous conduit à minimiser :

$$L_p(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^n \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^n \alpha_i \quad \text{Eq. [5]}$$

La résolution de ce problème peut aussi s'effectuer à l'aide du dual de  $L_p$ , le dual de Wolfe,  $L_D$ :

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad \text{Eq. [6]}$$

et en s'appuyant sur les conditions de Karush-Kuhn-Tucker [27], il vient :

$$\frac{\partial L_p(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \quad \text{Eq. [7]}$$

$$\frac{\partial L_p(w, b, \alpha)}{\partial b} = -\sum_{i=1}^n \alpha_i y_i = 0 \quad \text{Eq. [8]}$$

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad \text{Eq. [9]}$$

$$\alpha_i \geq 0 \quad \text{Eq. [10]}$$

$$\alpha_i (y_i (w \cdot x_i + b) - 1) \geq 0 \quad \text{Eq. [11]}$$

pour tous les éléments de  $\mathcal{B}$ .

Ces conditions sont nécessaires et suffisantes pour déterminer  $w$ ,  $b$  et  $\alpha$  à partir de Eq. [5]. La détermination de  $w$  est inhérente à la connaissance de tous les couples  $\{x_i; y_i\}$  qui sont par définition connus (ce sont les vecteurs description des prototypes composant la base d'apprentissage). La constante  $b$ , n'est quant à elle, définie qu'implicitement mais son évaluation est rendue possible par l'Eq. [11], car seuls les  $\alpha_i$  correspondant aux vecteurs supports sont non nuls ; ainsi le calcul de  $b$  s'effectue soit en prenant un  $i$  particulier soit en calculant la moyenne de tous les  $b$  obtenus pour chaque vecteur support.  $w$  est défini par :

$$w = \sum_{i=1}^{N_s} \alpha_i y_i s_i \quad \text{Eq. [12]}$$

où  $N_s$  est le nombre de vecteurs supports (au plus égal au nombre de prototypes composants la base d'apprentissage),  $s_i$  est le vecteur support numéro  $i$  de classe d'étiquette  $y_i$ . Après cette phase d'apprentissage, l'équation de l'hyperplan séparateur connue, tout nouveau prototype peut être classé selon sa position relative de son vecteur descripteur par rapport à ce plan. La fonction de décision est donc :

$$y = \text{signe} \left( \sum_{i=1}^{N_s} \alpha_i y_i s_i x + b \right) \quad \text{Eq. [13]}$$

où  $y$  est la prédiction d'appartenance à une classe du vecteur description  $x$  du prototype à classer. Cette fonction de décision se résume au calcul d'une somme de produits scalaires des vecteurs supports et du vecteur décrivant le prototype à classer, et ce dans l'espace de description initial. Sa complexité algorithmique est en  $o(3N_s d)$ .

À partir de cette fonction de décision, basée sur une séparabilité linéaire des données, il est possible d'étendre le champ d'application de cette méthode au cas où les données ne sont pas linéairement séparables dans l'espace de description et de passer dans un espace de dimension supérieure.

Il est possible de résoudre ce problème d'une manière simple et efficace [28]. Les équations Eq. [6], Eq. [7], Eq. [8] font intervenir les données (les prototypes  $x_i$  et leurs étiquettes de classe  $y_i$ ) sous la forme d'un produit scalaire. Si on projette l'espace de description  $\mathbb{R}^d$  dans un espace euclidien  $\mathbb{H} = \mathbb{R}^e$  (avec  $e > d$  fini ou non) à l'aide d'une fonction telle que :

$$\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^e, \varphi(x) = \tilde{x} \quad \text{Eq. [14]}$$

Il est possible que dans cet espace les données initiales soient linéairement séparables [29]. Si ces espaces sont munis d'un produit scalaire, noté  $(\cdot)$  ; on peut définir sur  $\mathbb{R}^d$  une fonction noyau  $K$  telle que :

$$K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^e \times \mathbb{R}^e, K(x, y) = \varphi(x) \cdot \varphi(y) = \tilde{x} \cdot \tilde{y} \quad \text{Eq. [15]}$$

La connaissance explicite de la fonction  $\phi$  n'est plus nécessaire et la fonction de décision devient :

$$y = \text{signe} \left( \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b \right) \quad \text{Eq. [16]}$$

où les  $s_i$  sont les vecteurs supports, au nombre de  $N_s$ , dans l'espace de description de dimension  $d$ , d'étiquette de classe  $y_i$ , issus de la phase d'apprentissage et sont déterminés de la même façon que précédemment. Le produit scalaire réalisé par l'intermédiaire de la fonction noyau est pondéré par les  $\alpha_k$  (multiplicateurs de Lagrange) déterminés eux aussi durant la phase d'apprentissage. En revanche, le choix de la fonction noyau  $K$ , doit répondre au critère de Mercer, et il a été montré dans [10] et [30] que toute fonction de la forme :

$$K(x, y) = \sum_{i=1}^{\infty} a_i (x, y)^i, \forall a_i \in \mathbb{R}^+ \quad \text{Eq. [17]}$$

satisfont ce critère. Il vient naturellement que toutes les fonctions à exponentielle bornée et tous les polynômes peuvent être utilisés comme fonctions  $K(., .)$ . Les plus fréquemment utilisées sont :

Le noyau polynomial :

$$K(x, y) = (ax, y + c)^p \quad \text{Eq. [18]}$$

Le noyau exponentiel ou Radial Basis Function (RBF) :

$$K(x, y) = \text{Exp} \left\{ -\frac{\|x - y\|^2}{\gamma} \right\} \quad \text{Eq. [19]}$$

Le noyau tangente hyperbolique :

$$K(x, y) = \text{Tanh} \{ a(xy) - \theta \} \quad \text{Eq. [20]}$$

Malgré l'apparente complexité de la méthode, sa mise en œuvre est relativement simple. La prédiction d'appartenance à une classe d'un élément s'effectue par simple calcul du signe du produit scalaire entre son vecteur descripteur et les vecteurs supports déterminés durant la phase d'apprentissage. La fonction noyau pouvant être assimilée à un coefficient multiplicatif non linéaire. De plus il existe un grand nombre de bibliothèques disponibles permettant la résolution de divers problèmes par le biais de cette méthode, et ce sur un grand nombre d'architectures et sous un grand nombre de langages (MathLab [31], C sous UNIX/Windows [32], applet Java...).

## 4 Recherche des paramètres de classification optimums

### 4.1 Protocole d'apprentissage et de test

En regard du faible nombre d'éléments disponibles dans les classes défauts, nous avons choisi de valider les différentes méthodes utilisées grâce à une Validation Croisée d'ordre 10 (CV 10) [33]. À partir de l'ensemble  $\mathcal{D}$  de cardinalité  $n$ , regroupant la totalité des éléments mis à notre disposition par le partenaire industriel de ce projet, nous avons généré de manière itérative deux bases d'apprentissage et de test. A



chaque itération, l'apprentissage est mené sur une base  $\mathcal{B}$  de cardinalité  $\frac{9n}{10}$ , et le test est mené sur une base de test  $\mathcal{T}$  (cardinalité  $\frac{n}{10}$ ). L'opération est itérée 10 fois. Les différentes erreurs obtenues à chaque itération sont sommées afin d'obtenir des taux de fausses alarme et de non détection nous permettant de comparer les performances de classification pour chacun des types de défaut.

## 4.2 Méthode d'optimisation des paramètres de l'algorithme SVM à noyau RBF

Le paramétrage du classifieur par hyperrectangles s'appuie sur les études menées dans [34]. Pour le cas du classifieur par SVM, nous nous sommes limités à un classifieur basé sur un noyau RBF équation Eq. [19] qui présente l'avantage de n'avoir qu'un paramètre à fixer contre trois pour les noyaux polynomiaux. Nous l'appellerons par la suite SVM RBF.

Selon V. Vapnik [35], le choix de la valeur de  $\gamma$  dépend de la dimension de l'espace de description auquel les SVM sont appliqués, et celui-ci préconise de choisir  $\gamma = \frac{1}{d}$ .

Toutefois, comme on peut le constater sur la Figure 11 cette valeur n'est pas toujours optimale (ici la valeur préconisée par Vapnik correspond à l'abscisse 5, pour laquelle l'erreur n'est pas minimum et les contraintes industrielles ne sont pas respectées). C'est pourquoi nous avons utilisé un parcours d'un ensemble de valeurs de  $\gamma$  discrètes réelles définies par :

$$\gamma_i = \frac{5i+1}{5d}$$

avec  $i = 0, \dots, n$ , tel que  $n = \lfloor 5d - 1 \rfloor$  où  $\lfloor X \rfloor$  est la partie entière de  $X$  et  $d$  la dimension de l'espace de description, dans ce cas on a :

$$\frac{1}{5d} \leq \gamma_i \leq 5, \forall i$$

Les bornes supérieures et inférieures de cet intervalle ont été définies de façon empirique afin de couvrir une plage de valeur suffisamment large autour de la valeur recommandée  $\gamma = \frac{1}{d}$ . Le choix de  $\gamma$  est fait alors selon la règle suivante :

$$\gamma_{opt} = \text{Arg min}(\lambda(\gamma)) \text{ avec } \lambda(\gamma) = \text{Max} \left\{ \frac{\text{FAR}(\gamma)}{1}; \frac{\text{NDR}(\gamma)}{20} \right\} \text{ et } \gamma \in \left[ \frac{1}{5d}; 5 \right] \quad \text{Eq. [21]}$$

Où TFA est le taux de fausses alarmes et TND celui de non détections obtenus après la phase de test pour un  $\gamma$  donné. La normalisation utilisée ici nous assure, pour  $\lambda < 1$  des taux satisfaisant les exigences de l'industriel pour cette étude (TFA < 1% et TND < 20%). Ces exigences représentent les spécifications

demandées pour le contrôle du processus de fabrication, mais ne sont, bien sûr, pas représentatives de la production finale.

Comme on peut le constater sur la Figure 11, la courbe présente une assez large plage pour laquelle  $\lambda$  varie peu, ce qui nous assure une assez bonne stabilité de l'erreur. Le choix de  $\gamma$  dans cette plage n'est donc pas critique.

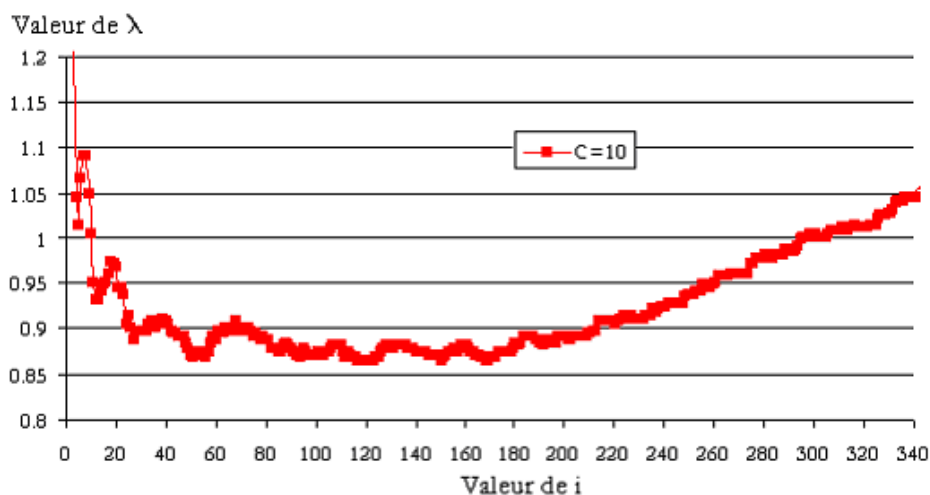


Figure 11 Exemple d'évolution de la valeur de  $\lambda$  en fonction de  $\gamma$  ( $\lambda$  minimum pour  $i=120$ ).

### 4.3 Résultats initiaux

Le Tableau 3 consigne les performances de classification obtenues en cherchant à classer les défauts particuliers et les pièces sans défaut, en utilisant l'ensemble des paramètres définis précédemment. Nous pouvons conclure qu'ici la détection n'est pas réalisée dans de bonnes conditions, quel que soit le classifieur utilisé. Ces chiffres nous indiquent que dans la majeure partie des cas, les classifieurs considèrent toutes les pièces comme « bonnes » mises à part quelques unes (Taux de Fausses Alarmes proche de 0 et celui de Non Détection proche de 100) qui peuvent être considérées comme très défectueuses. Les résultats les moins bons sont obtenus par les Hyperrectangles, qui sont incapables ici de classer les décolorations, et à peine mieux les autres défauts. Les erreurs importantes de cette méthode proviennent du trop faible nombre d'éléments d'apprentissage, en regard de la taille de l'espace. Les hyperrectangles ne généralisent pas correctement. Les résultats issus d'Adaboost sont un peu meilleurs, mais sont encore loin de respecter la contrainte  $\lambda < 1$ . Les SVM donnent les résultats les plus prometteurs, sans toutefois être satisfaisants. La conclusion de ces expériences préliminaires est que les paramètres extraits sur l'image, pris dans leur globalité, sont incapables d'apporter suffisamment d'informations pour compenser le faible nombre d'éléments d'apprentissage et le problème induit par la présence de nombreux défauts acceptables

dans les pièces bonnes. Dans ce cas, même une méthode robuste comme les SVM ne peut donner de résultats exploitables.

**Tableau 3 : Performances de classification défaut par défaut à  $\lambda$  minimum**

	<i>Décolorations locales</i>			<i>Bourgeons</i>			<i>Spires jointives</i>			<i>Ecailles</i>			<i>Pattes dédoublées</i>		
	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR
<b>SVM</b>	<b>3.00</b>	0.15	60.00	<b>1.93</b>	0.53	38.57	<b>3.75</b>	0.09	75.00	<b>1.50</b>	0.29	30.00	<b>0.49</b>	0.19	9.86
<b>Adaboost</b>	<b>3.00</b>	0,00	60,00	<b>3.33</b>	0.25	66.67	<b>2.92</b>	0.06	58.33	<b>2.63</b>	0.56	52.56	<b>0.71</b>	0.06	14.29
<b>Hyperrectangle</b>	<b>5.00</b>	0.00	100.0	<b>4.49</b>	0.00	87.45	<b>4.37</b>	0.18	87.50	<b>3.21</b>	0.62	64.12	<b>3.51</b>	0.44	70.27

**Tableau 4 Classification à 2 classes**

<i>Méthode</i>	<i>Pièces bonnes contre tous les défauts</i>		
	$\lambda_{opt}$	FAR	NDR
SVM RBF	1,36	1,12	27,20
Adaboost	1,81	1,81	34,15
Hyperrectangles	3,25	1,43	65,04

Les deux voies d'amélioration possibles sont donc d'une part le regroupement en une seule classe des défauts, et d'autre part la réduction de la taille de l'espace par sélection de paramètres. La première voie est acceptable car la première des exigences de l'industriel est la détection d'un défaut, bien avant de connaître la nature de ce défaut. Les performances de détection sont consignées dans le Tableau 4, et montrent une amélioration sensible des résultats : les performances obtenues pour la détection de pièces défectueuses sont bien meilleures que pour la détection d'un type de défaut particulier. L'espace de description étant le même, les algorithmes identiques, cette amélioration ne peut être qu'attribuée à une plus grande densité de points dans l'espace de description. En effet, dans le premier cas le nombre d'éléments disponibles pour l'apprentissage (au cours d'une itération de la validation croisée) est au plus de 75 (pour le défaut pattes dédoublées) alors que dans la seconde méthode (regroupement de toutes les pièces défectueuses) il est de 221. La hiérarchie entre les méthodes reste toutefois la même, et les spécifications industrielles ne sont toujours pas respectées.

#### 4.4 Sélection de paramètres

Le but d'un algorithme de sélection de paramètres est d'améliorer les performances de classification et de réduire les temps de calculs en sélectionnant les paramètres ou combinaisons de paramètres les plus pertinents. Pour cela il est nécessaire de fixer deux éléments principaux :

- un critère de mesure afin de comparer les différents sous-espaces extraits de l'espace de description total,
- une méthode de recherche de la meilleure combinaison possible de paramètres.

Dans un souci de cohérence avec le reste de l'étude, nous avons utilisé comme critère de minimisation les résultats de classification via l'algorithme SVM RBF [36]. Chaque phase de la recherche consiste donc à bâtir un nouvel espace de représentation, et réaliser un apprentissage, et à mesurer les performances de classification. Etant donné le nombre d'échantillons, la dimension de l'espace initial et la complexité des SVM, chaque itération demande donc un temps de calcul important, ce qui nous a guidé dans le choix de la méthode de parcours des différentes combinaisons possibles. En effet, de très nombreuses méthodes d'optimisation sont décrites dans la littérature. Le lecteur pourra se référer à [37] pour un état de l'art complet. Nous avons éliminé d'emblée les méthodes exhaustives étant donné la dimension de l'espace de départ. Parmi les méthodes non exhaustives, certaines sont déterministes, basées sur des parcours particuliers des différentes combinaisons possibles. D'autres sont basées par exemple sur des algorithmes génétiques ou encore des méthodes classiques comme le recuit simulé [38], [39]. Les travaux de Ferri [40] ayant montré que les performances des algorithmes génétiques diminuent lorsque la sélection s'effectue sur des espaces de dimension supérieure à trente paramètres, nous avons finalement choisi d'utiliser un algorithme de sélection déterministe éprouvé dit « Sequential Backward Floating Selection » décrit précisément dans [41], [42] ou encore [37] et qui présente l'avantage de donner un résultat en un petit nombre d'itérations.

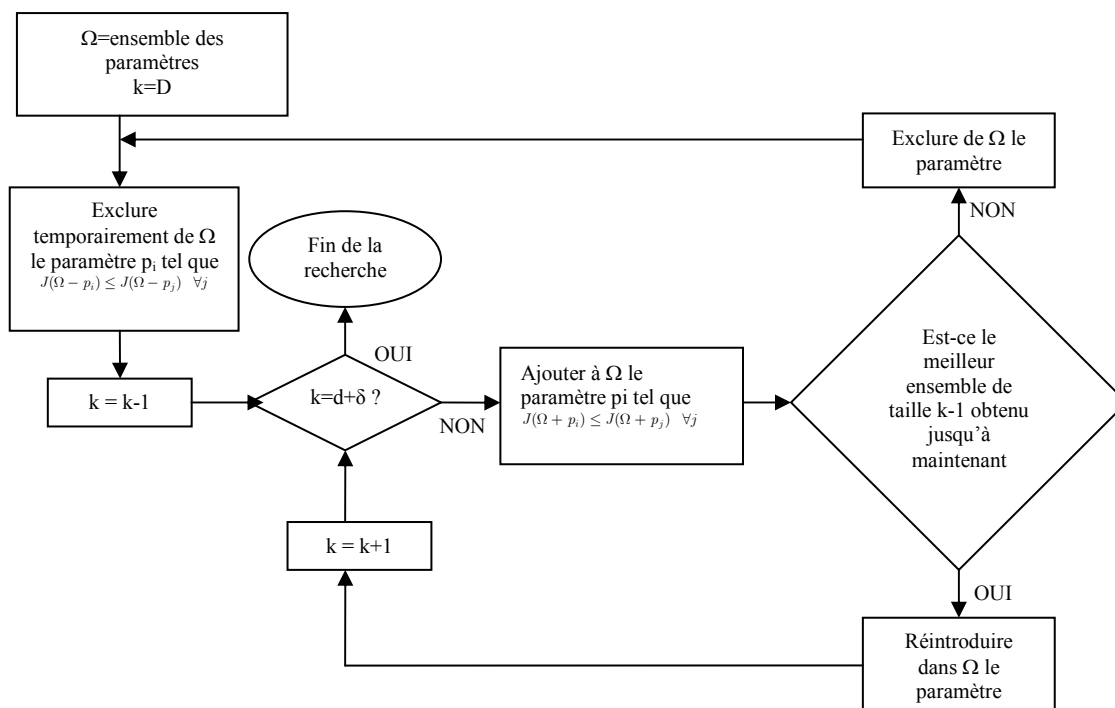


Figure 12 Sequential Backward Floating Selection (SBFS)

Cette méthode consiste à parcourir des combinaisons de paramètres, depuis l'espace total initial  $\Omega$  de dimension  $D$  jusqu'au un espace de dimension  $d+\delta < D$ , avec  $d$  et  $\delta$  fixés par l'utilisateur. L'utilisation

d'une boucle de « back tracking » permet de s'affranchir des minima locaux lors de ce parcours. Elle permet de tester l'ajout d'un paramètre à une combinaison d'où il avait été exclu précédemment. Le sous espace optimal a été défini comme étant celui de dimension la plus faible pour le  $\lambda$  le plus petit. La stratégie générale est représentée Figure 12. La fonction de critère, c'est-à-dire issue de l'erreur du classifieur SVM, est notée ici J.

Il est à noter que la sélection n'a pas porté sur les paramètres pris individuellement mais sur des groupes de paramètres. En effet, il ne serait pas pertinent de sélectionner l'aire de la spire n°2 sans sélectionner l'aire de toutes les autres, ce qui peut arriver si aucun défaut n'est présent sur la spire n°3 par exemple dans l'ensemble d'apprentissage. Tous les paramètres calculés auparavant par spire ayant été regroupés, restent 30 groupes de paramètres parmi lesquels la sélection est effectuée.

Afin de réaliser cette sélection dans des conditions optimums pour le classifieur, une phase de test a été menée pour définir le  $\gamma$  optimal à partir de la courbe  $\lambda=f(\gamma)$ , où  $\gamma$  est le paramètre du noyau gaussien des SVM RBF. Ce paramètre est fixé lors de la première itération de la méthode de validation croisée d'ordre 10. Les neuf dernières itérations complètent le processus permettant d'obtenir un taux d'erreur  $\lambda_p$  pour un espace donné de paramètres. Cette valeur  $\lambda_p$  permet de comparer les performances obtenues sur les différents sous espaces sélectionnés. La phase de sélection est donc très gourmande temps de calcul, mais elle ne s'effectue qu'une seule fois afin de sélectionner le sous espace minimisant  $\lambda$ .

Après cette sélection, on remarque une très nette amélioration, tant au niveau des performances qu'au niveau du nombre de paramètres à extraire (la dimension finale, après élimination des redondances entre défauts, est de  $d=100$ ). Le défaut « spire jointives » pose le plus de problèmes, ceci étant dû à la variabilité intrinsèque du filament. On peut remarquer que la méthode du Boosting est d'ailleurs légèrement meilleure que les SVM dans ce cas, et que la différence avec les Hyperrectangles est beaucoup moins importante que précédemment.

**Tableau 5 : Performance maximale par défaut après sélection de paramètres**

	<i>Décolorations locales (20)</i>			<i>Bourgeons (16)</i>			<i>Spires jointives (22)</i>			<i>Ecailles (39)</i>			<i>Pattes dédoublées (13)</i>		
	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR	$\lambda_{opt}$	FAR	NDR
<b>SVM</b>	<b>1,25</b>	0,39	25	<b>1,50</b>	0,24	30	<b>3,38</b>	0,24	67,5	<b>1,75</b>	0,58	35	<b>0,37</b>	0,12	7,5
<b>Adaboost</b>	<b>3,00</b>	0,00	60,00	<b>3,33</b>	0,25	66,67	<b>2,92</b>	0,06	58,33	<b>2,63</b>	0,56	52,56	<b>1,07</b>	0,17	21,4
<b>Hyperrectangle</b>	<b>3,38</b>	3,38	21,50	<b>1,78</b>	1,78	31,10	<b>4,28</b>	1,47	85,62	<b>1,85</b>	1,85	27,36	<b>1,84</b>	0,43	36,9

**Tableau 6 Performances en classification à deux classes après sélection de paramètres**

<i>Méthode</i>	<i>Pièces bonnes contre tous les défauts</i>		
	$\lambda_{\text{opt}}$	FAR	NDR
SVM	0,98	0,73	19,59
Adaboost	1,11	1,11	22,15
Hyperrectangles	1,17	1,08	23,36

Nous avons également regroupé les différents défauts en une classe unique. Les résultats obtenus dans ce cas sont reportés Tableau 6.

On peut remarquer que les spécifications industrielles sont respectées grâce à la méthode SVM, après sélection de paramètres et regroupement des classes. On remarque également que les différences entre les méthodes sont beaucoup moins importantes que dans le cas multiclassé et dans l'espace d'origine.

Il est donc clair ici que les SVM donnent les meilleures performances, et résistent mieux à la présence de paramètres non discriminants et au faible taux de remplissage de l'espace par les échantillons d'apprentissage que les autres méthodes.

## 5 Conclusion et perspectives

Nous avons montré au cours de cette étude le bien fondé de l'utilisation des SVM dans le cadre du contrôle qualité par vision artificielle. Une étude complète a été menée, depuis la définition de l'acquisition (un banc prototype a été réalisé, il permet la prise d'images des pièces à une cadence de 10 pièces.s<sup>-1</sup>), jusqu'à la prise de décision, en passant par l'extraction de nombreux paramètres faisant parfois appel à des algorithmes de bas niveau assez complexes.

Nous avons étudié trois méthodes de classification choisies sur des critères de performance et de vitesse, et nous avons appliqué ces méthodes à l'ensemble d'apprentissage défini dans les phases précédentes. Les résultats préliminaires, décevants, nous ont permis d'explorer deux voies d'amélioration que sont le regroupement des classes et la diminution du nombre de paramètres de classification à l'aide d'une méthode de sélection éprouvée, se basant sur la fonction de décision des SVM.

Nous avons de plus proposé ici une méthode permettant de déterminer automatiquement le paramètre du noyau RBF que nous avons choisi au départ de cette étude.

Ce travail nous a permis de montrer que les SVM peuvent permettre de résoudre des problèmes concrets comme ceux issus du contrôle qualité par vision artificielle, et ce même lorsque les cadences de traitement sont élevées. Nous avons pu également montrer que, bien que présentant des avantages certains par rapport aux réseaux de neurones simples comme le perceptron multicouche (convergence extrêmement rapide, faible nombre de paramètres à choisir et à régler), les SVM doivent, comme les autres méthodes,

être alimentés par des paramètres pertinents et des ensembles d'apprentissage conséquent, sous peine de voir leurs performances dégradées.

En ce qui concerne l'implantation matérielle des algorithmes, il est à noter que 95% du temps de traitement est utilisé par la phase de segmentation et d'extraction des paramètres, contre 5% seulement pour la phase de décision haut niveau (une décision par filament). Nous avons montré par ailleurs qu'il est également possible d'utiliser les performances des SVM dans le cas d'une décision bas niveau (pour réaliser la segmentation par classification des pixels, par exemple), grâce à une méthode de combinaison des phases d'apprentissage des SVM et des hyperrectangles [13]. Dans ce cas l'apprentissage est d'abord réalisé par les SVM sur une partie des échantillons disponibles. L'ensemble des éléments d'apprentissage (après ajout d'une perturbation qui permet d'améliorer la généralisation) est alors classé par les SVM, avant d'être présenté comme ensemble d'apprentissage à la méthode basée sur les hyperrectangles. Les SVM ont ainsi filtré les éventuelles ambiguïtés présentes dans l'apprentissage. En phase de décision, seul le modèle issu des hyperrectangles est utilisé, ce qui permet d'obtenir des cadences de décision de l'ordre de 10ns. Une autre voie possible d'optimisation consiste à employer une stratégie d'optimisation du nombre de vecteurs support.

Les résultats issus du Boosting ont montré qu'une méthode de complexité bien moindre que celle des SVM pouvait donner des résultats satisfaisants dans le contexte de ce projet. Une étude est en cours pour réaliser une implantation matérielle automatisée sur architecture reconfigurable de la phase de décision du Boosting, en vue également de prises de décision très rapides.

Enfin, nous pouvons noter que les premiers objectifs en terme de Fausses Alarmes et Non Détections sont atteints, même s'il reste à augmenter le nombre de points de vues afin d'améliorer encore les performances globales.

## 6 Bibliographie

- [1] M.M.Ratnam and C.P.Lim, "Classification using shadow moiré and neural network" in Proc. of QCAV 2001 - Vol. 2, pp. 355-358, 2001.
- [2] C.M.Bishop, Neural network for pattern recognition, Oxford university press, 1995.
- [3] C.Firmin, D.Hamad, J.-G.Postaire and R.D.Zhuang, "Gaussian neural network for glass bottle production : a learning procedure" in International Journal on Neural Systems - Vol. 8, pp. 41 - 60, 1996.
- [4] B.Dubuisson, Diagnostic, intelligence artificielle et reconnaissance des formes (Traité IC2 série productique), 2001.
- [5] R.O.Duda, P.E.Hart and D.G.Stork, Pattern classification, John Wiley & Son, INC, 2001.
- [6] S.Salzberg, "A nearest hyperrectangle learning method" in Machine Learning - Vol. 6, pp. 251 - 276, 1991.
- [7] J.Miteran, P.Gorria and M.Robert, "Classification géométrique par polytopes de contraintes. Performances et intégration" in Traitement du Signal - Vol. 11(:5), pp. 393 - 408, 1994.

- [8] Y.Freund, "Boosting a weak learning algorithm by majority" in Information and Computation - Vol. 121(:2), pp. 256 - 285, 1995.
- [9] R.E.Schapire, "A brief introduction to boosting" in Proc. of 16<sup>th</sup> International Joint Conference on Artificial Intelligence, pp. 1401-1405, 1999.
- [10] V.N.Vapnik, The nature of statistical learning theory, Springer-Verlag, New-York, 1995.
- [11] J.Miteran, J-P.Zimmer, F.Yang and M.Paindavoine, "Access control : adaptation and real time implantation of a face recognition method" in Optical Engineering - Vol. 40(:4), pp. 586 - 593, 1-4-2001.
- [12] P.Geveaux, S.Kohler, J.Miteran, F.Truchetet. and F.Meriodaudeau, "Comparison between two classification methods, Application to defects detection by artificial vision in industrial field" in Proc. of SPIE - Vol. 3966A, pp. 154-161, 1-1-2000.
- [13] J.Miteran, S.Bouillant and E.Bourennane, "SVM approximation for real-time image segmentation by using an improved Hyperrectangles-based methods" in Real Time Imaging, Academic Press - Vol. 9(:3), pp. 179 - 188, 2003.
- [14] J.Miteran, P.Geveaux, R.Bailly and P.Gorria, "Real time defect detection using image segmentation" in Proc. of ISIE 97, pp. 713-716, 1-7-1997.
- [15] W.H.Tsai, "Moment preserving thresholding : a new approach" in Computer vision, graphics and image processing - Vol. 29, pp. 377 - 393, 1-12-1985.
- [16] G.J.McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley & Sons, New York, 1992.
- [17] R.A.Fisher, "The use of multiple measurements in taxonomic problem" in Annales of Eugenics - Vol. 7, Part II, pp. 179 - 188, 1936.
- [18] B.Schölkopf, A.Smola, K.R.Müller, C.J.C.Burges and V.N.Vapnik, "Support vector methods in learning and feature extraction" in Australian Journal of Intelligent Information Processing Systems - Vol. 1(:1), pp. 3 - 9, 1998.
- [19] Jonsson K., Smola A., Müller K.R. and Burges C., "Support Vector Machines for Face Authentication" in Proc. of British Machine Vision Conference, pp. 543-553, 1999.
- [20] M.A.Hearst, B.Schölkopf, S.Dumais, E.Osuna and J.Platt, "Trends and Controversies - Support Vector Machines" in IEEE Intelligent systems - Vol. 13(:4), pp. 18 - 28, 1998.
- [21] S.Canu, "Modèles connexionnistes et machines à vecteurs supports pour la décision" in *Décision de reconnaissance des formes en Signal*, Hermes, 2002, pp. 181-217.
- [22] K.Tieu and P.Viola, "Boosting Image Retrieval" in International Journal of Computer Vision - 561-3, pp. 17 - 36, 2004.
- [23] R.E.Schapire and Y.Singer, "A boosting-based system for text categorization" in Machine Learning - 392/3, pp. 135 - 168, 2000.
- [24] D.Wettschereck and T.Dietterich, "An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms" in Machine Learning - Vol. 19(:6), pp. 5 - 27, 1995.



- [25] J.Miteran, R.Bailly and P.Gorria, "Classification board for real time image segmentation" in Proc. of ICASSP 97, pp. 4069-4072, 1-4-1997.
- [26] V.N.Vapnik, Statistical learning theory, John Wiley & sons, INC, 1998.
- [27] R.Fletcher, Pratical methods of optimisation, John Wiley & Son, INC, 1987.
- [28] B.E.Boser, I.M.Guyon and V.N.Vapnik, "A training algorithm for optimal margin classifiers" in Proc. of Fifth annual workshop on computational learning theory, pp. 144-152, 1992.
- [29] R.O.Duda, P.E.Hart and D.G.Stork, Chap. n° 5 "Linear Discriminant functions" in *Pattern classification*, John Wiley & Sons, 2001, pp. 264-265.
- [30] R.Courant and D.Hilbert, Methods of mathematical physics, InterScience, New York, 1953.
- [31] S.Gunn, *MatLab SVM toolbox*, <http://www.isis.ecs.soton.ac.uk/resources/svminfo/download.php> - v 1.0 - 2001.
- [32] C.C.Chang and C.J.Li, *LIBSVM - A library for support vector machines*, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html> - v. 2.33 - 2002.
- [33] C.M.Bishop, Chap. n° 9 "Learning and generalization" in *Neural network for pattern recognition*, Oxford university press, 1995, pp. 372-375.
- [34] S.Bouillant, J.Miteran, F.Yang and M.Paindavoine, "Real-time flaw detection on complex part: study of SVM and Hyperrectangle based method" in Proc. of ICASSP 2002, pp. 3596-3599, 1-1-2002.
- [35] V.N.Vapnik, Chap. n° 12 "SV machines for pattern recognition" in *Statistical learning theory*, John Wiley & Sons, 1998, pp. 496-498.
- [36] M.Dash and H.Liu, "Feature selection for classification" in *Intelligent data analysis - Vol. 1(:3)*, pp. 131 - 156, 1997.
- [37] A.Jain and D.Zongker, "Feature selection: evaluation, application, and small sample performance" in *IEEE Transactions on pattern analysis and machine intelligence - Vol. 19(:2)*, pp. 153 - 158, 1997.
- [38] D.E.Goldberg, Genetic algorithms in search, optimization and machine learning, Addison Wesley, 1989.
- [39] J.R.Koza, Genetic Programming, MIT Press, 1992.
- [40] F.Ferri, P.Pudil, M.Hatef and J.Kittler, "Comparative study of techniques for large-scale feature selection" in *Pattern Recognition in Practice IV*, pp. 403 - 413, 1994.
- [41] J.Kittler, "Feature set algorithms" in *Pattern recognition and signal processing*, pp. 41 - 60, 1978.
- [42] P.Somol, Pudil P., J.Novovicova and P.Paclik, "Adaptative floating search methods in feature selection" in *Pattern recognition letters - Vol. 10*, pp. 335 - 347, 1989.