INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Noyaux texturaux pour les problèmes de classification par SVM en télédétection

Florent Lafarge — Xavier Descombes — Josiane Zerubia

N° 5370

Novembre 2004

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5370--FR

Noyaux texturaux pour les problèmes de classification par SVM en télédétection

Florent Lafarge , Xavier Descombes , Josiane Zerubia

Thème COG — Systèmes cognitifs
Projet Ariana

Rapport de recherche n° 5370 — Novembre 2004 — 39 pages

Résumé : Nous détaillons dans ce rapport la construction de deux noyaux texturaux s'utilisant dans les problèmes de classification par "Support Vector Machines" en télédétection. Les SVM constituent une méthode de classification supervisée particulièrement bien adaptée pour traiter des données de grande dimension telles que les images satellitaires. Par cette méthode, nous souhaitons réaliser l'apprentissage de paramètres qui permettent la différenciation entre deux ensembles de pixels connexes non-identiques. Nous travaillons pour cela sur des fonctions noyaux, fonctions caractérisant une certaine similarité entre deux données. Dans notre cas, cette similarité sera fondée à la fois sur une notion radiométrique et sur une notion texturale. La principale difficulté rencontrée dans cette étude réside dans l'élaboration de paramètres texturaux pertinents qui modélisent au mieux l'homogénéité d'un ensemble de pixels connexes. Nous appliquons les noyaux proposés à deux problèmes de télédétection : la détection de feux de forêt et la détection de zones urbaines à partir d'images satellitaires haute résolution.

Mots-clés : base d'apprentissage, Support Vector Machines, classification, noyau, texture, feux de forêt, zones urbaines

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Textural kernels for SVM classification applied to remote sensing problems

Abstract: We present in this report two textural kernels for "Support Vector Machines" classification applied to remote sensing problems. SVMs constitute a method of supervised classification well adapted to deal with data of high dimension, such as images. We would like to learn parameters which allow the differentiation between two sets of connected pixels. We also introduce kernel functions which characterize a notion of similarity between two pieces of data. In our case this similarity is based on a radiometric charateristic and a textural characteristic. The main difficulty is to elaborate textural parameters which are pertinent and characterize as well as possible the homogeneity of a set of connected pixels. We apply this method to remote sensing problems : the detection of forest fires and the extraction of urban areas in high resolution satellite images.

Key-words: training set, Support Vector Machines, classification, kernel, texture, forest fire, urban area

Table des matières

1 Introduction

Un des grands domaines de recherche en télédétection spatiale est la reconnaissance d'objets. La nature de ces objets est aussi variée que les méthodologies utilisées. On peut citer notamment les problèmes de détection de zones urbaines, de réseaux linéiques, de batiments, de véhicules ou bien encore de feux de forêt par exemple. Il s'agit, dans tous ces cas, de reconnaitre des types d'objets particuliers par des critères tels que la radiométrie, la texture ou bien encore la forme géométrique.

Dans le milieu des années 90, une méthode de classification supervisée appelée "Support Vector Machines" ("Séparateur à Vaste Marge" en français) fut introduite par Vapnik [Vapnik, 1996] . Cette méthode est particulièrement adaptée aux problèmes de grande dimension tels que l'imagerie 2D. Les SVM ont ainsi été exploités dans les problèmes de reconnaissance d'objets en télédétection spatiale. Pour pouvoir reconnaitre les objets de manière pertinente, il faut faire l'apprentissage de paramètres qui puissent différencier un ensemble de pixels connexes représentant l'objet recherché du reste de l'image. Les noyaux sont des fonctions utilisées afin de rendre le problème de classification par SVM possible, en spécifiant les critéres de similarité recherchés dans le problème de reconnaissance.

Dans ce rapport, nous proposons deux noyaux fondés à la fois sur un critère radiométrique et sur un critère textural. Deux ensembles de pixels connexes seront ainsi différenciés grace à une information radiométrique (c'est-à-dire une information liée à la teinte de la zone considérée) et grace à une information texturale (liée à l'homogénéité de l'intensité des pixels de la zone considérée). Un des intérêts de ces noyaux résidera dans la possibilité de pondérer l'influence du critère textural sur le critère radiométrique via un paramètre. Cela permettra aux noyaux d'être adaptatifs et de pouvoir considérer des objets nombreux et variés.

Nous avons testé ces deux noyaux sur des problèmes relativement classiques de télédétection. Ainsi, nous avons cherché à détecter les feux de forêt sur des images satellitaires haute résolution (Images SPOT5 de la région PACA durant les feux de forêt du mois d'août 2003). Nous avons également appliqué ces deux noyaux au problème de détection de zones urbaines sur des images issues du même capteur.

2 Support Vector Machines

2.1 Classification supervisée

Les SVM sont une méthode de classification supervisée, introduite par Vapnik en 1996 [Vapnik, 1996]. Cette méthode consiste à séparer les données en trouvant le séparateur dont la marge est la plus grande possible, d'où la seconde signification des initiales SVM : "Séparateur à Vaste Marge". Dans un premier temps, nous allons rappeler le principe du SVM dans le cas où le problème de classification est linéairement séparable. Nous considérons dans ce chapitre que la classification est binaire, c'est-à-dire que l'on cherche à séparer les données en deux classes.

Définissons tout d'abord quelques **notations** :

- **E**, un ensemble de données à classifier
- **F**, un sous-ensemble de E constituant une classe d'intérêt à distinguer.
- **Y**, l'ensemble des classes : ici $Y = \{-1, 1\}$
- $\mathcal{A}$, une base d'exemples, c'est-à-dire un sous-ensemble de $E \times Y$ (appelé aussi base d'apprentissage).
- **A**, l'ensemble d'apprentissage, est la première projection de $\mathcal{A}$ sur E, ainsi que $A_1 = A \cap F$ et $A_{-1} = A \cap F^c$. Alors $\mathcal{A} = [A_1 \times \{1\}] \bigcup [A_{-1} \times \{-1\}]$.
- $\overline{\omega}$=$sign_F$ est une application de E dans $\{-1, 1\}$ appelée classification cible.Nous définons l'application $sign_F$ qui à $x \in E$ associe 1 si $x \in F$ et -1 sinon.
- $\mathcal{W} \in \{-1, 1\}^E$ est la famille des classifieurs du problème.

Nous cherchons alors à construire une règle de classification, c'est-à-dire à reconstruire selon la base d'exemple $\mathcal{A}$ la fonction indicatrice $\overline{\omega} = sign_F \in E^{[-1,1]}$ de F.

Définition 1 *Un problème de classification binaire supervisé est définie par le quintuplet* $(E, \overline{\omega}, A, \mathcal{A}, \mathcal{W})$.

Définition 2 *Soit* $(E, \overline{\omega}, A, \mathcal{A}, \mathcal{W})$, *un problème de classification binaire supervisée. La base d'apprentissage* $\mathcal{A}$ *est dit* ***consistante*** *avec la cible* $\overline{\omega}$ *si*

$$\forall (x, y) \in \mathcal{A}, y = \overline{\omega}(x)$$

La consistance de la base d'apprentissage signifie que la base d'exemples a bien été apprise et donc que le classifieur trouvé sépare correctement les données de la base d'apprentissage.

Définition 3 *Si E est un sous-ensemble d'un espace vectoriel réel de dimension finie n, on dit que le problème $(E, \varpi, A, \mathcal{A}, \mathcal{W})$ est un problème de classification binaire* ***numérique****. n est appelé le nombre d'attributs.*

Définition 4 *Soit $(E, \varpi, A, \mathcal{A}, \mathcal{W})$, un problème de classification binaire numérique où le nombre d'attributs est n. Ce problème est dit* ***linéaire*** *si $\mathcal{W} = W_{\mathfrak{L}}$ est l'ensemble des* ***classifieurs linéaires*** *qui s'identifie à l'espace vectoriel $\mathbb{R}^{n+1}$ par l'identification qui à $(\theta, w) \in \mathbb{R} \times \mathbb{R}^n$ associe la fonction binaire définie sur $E \subset \mathbb{R}^n$ par $\omega(x) = sign_{\mathbb{R}^+}(w.x - \theta)$ où . représente le produit scalaire euclidien dans $\mathbb{R}^n$.*

Rechercher un séparateur parmi les classifieurs linéaires revient à chercher, nous le verrons lors de la formulation du problème, un hyperplan de E qui sépare l'espace E en deux demi-espaces dans lequels se trouveront les deux classes d'exemples.

Définition 5 *Soit $(E, \varpi, A, \mathcal{A}, \mathcal{W})$, un problème de classification binaire numérique linéaire. On dit que le problème est* ***linéairement séparable*** *si la cible est linéaire $\varpi \in W_{\mathfrak{L}}$ et si la base d'apprentissage $\mathcal{A}$ est consistante avec la cible.*

Une notion importante dans les problèmes de classification supervisée est la complexité d'apprentissage. De manière elliptique, elle représente le cardinal minimal que la base d'apprentissage doit avoir pour être consistante. Elle caractérise donc la taille optimale de la base d'apprentissage permettant de ne pas être en situation de sous-apprentissage tout en n'étant pas en sur-apprentissage. Cette complexité n'est pas calculable ; la solution consistant alors à la borner.

2.2 Formulation du problème

Nous considèrons un problème de classification numérique binaire linéaire $(E, \varpi, A, \mathcal{A}, \mathcal{W})$ où le nombre d'attributs est n. Nous cherchons alors une solution consistante avec la base d'apprentissage (ce problème est aussi appelé problème du perceptron, présenté par Rosenblatt [Rosenblatt, 1962]).
Nous formulons ce problème en associant à chaque exemple x de A, le vecteur

$\widetilde{x} = (x, -1) \in \mathbb{R}^{n+1}$ si $x \in A_1$ et $\widetilde{x} = (-x, 1) \in \mathbb{R}^{n+1}$ si $x \in A_{-1}$. Nous appelons $\widetilde{\mathcal{A}}$, le sous-ensemble de $\mathbb{R}^{n+1}$ ainsi construit.
Nous cherchons alors $\widetilde{w} = (w, \theta) = (w_1, ..., w_n, \theta) \in \mathbb{R}^{n+1}$ tel que

$$\forall \widetilde{x} \in \widetilde{\mathcal{A}},\ \widetilde{w}.\widetilde{x} > 0 \tag{1}$$

Nous cherchons donc un hyperplan de E de la forme $w.x - \theta = 0$ qui sépare les exemples de la base d'apprentissage de manière stricte (les exemples ne doivent pas appartenir à l'hyperplan) et unanime (tous les exemples doivent être bien classés selon leur base d'apprentissage).

Si l'équation 1 admet un vecteur solution, alors nous pouvons voir que les vecteurs homotétiques dans un rapport positif à ce vecteur solution sont aussi solutions de l'équation. Cela nous permet alors de normaliser le problème en imposant

$$\widetilde{w}.\widetilde{x} \geq 1 \tag{2}$$

L'ensemble d'apprentissage est ainsi séparé par deux hyperplans parallèles. En effet, pour $x \in A_1$, $w.x - \theta \geq 1$ et pour $x \in A_{-1}$, $w.x - \theta \leq -1$.
Les équations des deux hyperplans sont donc :

$$\begin{cases} w.x - \theta & = 1 \\ w.x - \theta & = -1 \end{cases} \tag{3}$$

Les contraintes imposées par l'équation 2 s'écrivent, en utilisant la réponse désirée y du séparateur linéaire à l'exemple x, par :

$$\forall (x, y) \in \mathcal{A},\ y(w.x - \theta) \geq 1 \tag{4}$$

L'espace entre ces deux hyperplans définit donc une zone de sécurité entre les deux classes d'exemples. La distance euclidienne entre les deux hyperplans , appelée

aussi la marge du classifieur, est $\frac{2}{\|w\|}$.
La marge représente donc le double de la distance de l'hyperplan séparateur $w.x-\theta = 0$ aux exemples.
Le but de la classification par SVM est alors de trouver le séparateur (c'est-à-dire le couple (w, θ)) dont la marge est la plus large possible, et cela sous les contraintes de séparation par les deux hyperplans (contraintes données par l'équation (4)). Nous formulons cela dans la proposition suivante.

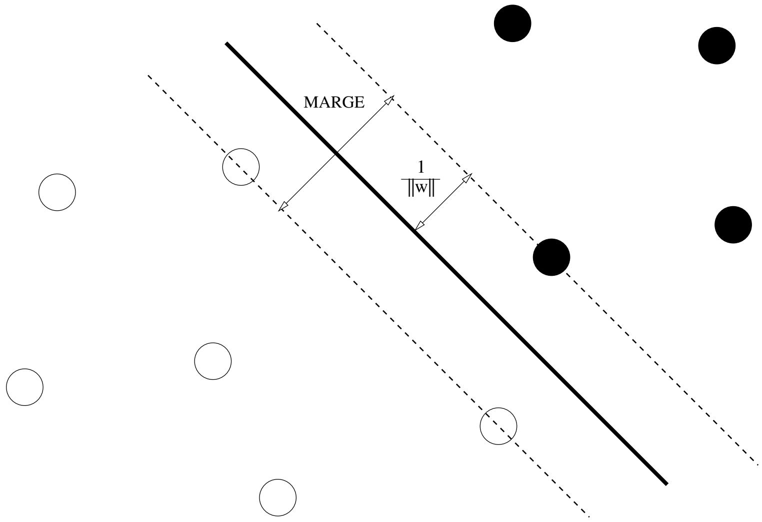


FIG. 1 – Marge d'un classifieur linéaire

Propriété 1 *Soit* $(E, \varpi, A, \mathcal{A}, \mathcal{W}_{\mathcal{L}})$, *un problème de classification supervisée binaire linéaire et linéairement séparable. Le séparateur linéaire de marge maximale est solution du problème d'optimisation quadratique sous contraintes*

$$\begin{cases} \min_{w,\theta} \frac{\|w\|}{2} = \frac{1}{2}(w_1^2 + ... + w_n^2) \\ sous\, \forall (x_i, y_i) \in \mathcal{A},\ y_i(w.x_i - \theta) \geq 1 \end{cases} \tag{5}$$

Nous sommes en présence d'un problème d'optimisation quadratique sous contraintes inégalités linéaires. Le problème de classification étant linéairement séparable, il admet une unique solution donnée par le théorème de Kuhn-Tucker. En introduisant des multiplicateurs de Lagrange pour chaque contrainte, nous établissons la formulation duale de ce problème (démonstration dans [Samuelides, 2004]).

Le problème d'optimisation 5 est équivalent au problème d'optimisation suivant :

$$\left\{ \begin{array}{l} \max_\alpha \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i|x_j) \\ sous \left\{ \begin{array}{l} \forall i,\, \alpha_i \geq 0 \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{array} \right. \end{array} \right. \tag{6}$$

Où $(.|.)$ représente le produit scalaire euclidien associé à E
N est le nombre de paramètres de Lagrange-Kuhn-Tucker.

La résolution de ce problème d'optimisation convexe n'est pas évidente. Il existe cependant plusieurs méthodes telle que la méthode du gradient conjugué sous contraintes. Dans notre étude, nous utiliserons un algorithme d'optimition utilisé en imagerie, appelé ICM, avec une réduction de contraintes. Nous développerons cette méthode dans la suite de ce chapitre.

En appelant $\alpha^\star$, la famille de paramètres de Lagrange-Kuhn-Tucker optimale (c'est-à-dire l'unique solution du problème 6), les paramètres du classifieur optimal $(w^\star, \theta^\star)$ sont alors obtenus par :

$$w^\star = \sum_{i=1}^N \alpha_i^\star y_i x_i \tag{7}$$

$$\theta^\star = \sum_{i=1}^N \alpha_i^\star y_i (x_i|x_j) - 1\,, pour\; x_j \in A_1 \tag{8}$$

Le classifieur optimal $\omega^\star$ s'exprime alors par :

$$\omega^\star(x) = sign_{\mathbb{R}^+}\left(\sum_{i=1}^N \alpha_i^\star y_i (x_i|x) - \theta^\star\right) \tag{9}$$

Nous proposons de résumer ce résultat dans la proposition suivante.

Propriété 2 *Soit $(E, \overline{\omega}, A, \mathcal{A}, \mathcal{W}_{\mathcal{L}})$, un problème de classification supervisée binaire linéaire et linéairement séparable. Le séparateur linéaire de marge maximale $\omega^\star$ est donné par :*

$$\omega^\star(x) = sign_{\mathbb{R}^+}\left(\sum_{i=1}^{N} \alpha_i^\star y_i (x_i|x) - \theta^\star\right)$$

où les $\alpha_i^\star$ sont les paramètres de Lagrange-Kuhn-Tucker solutions du problème d'optimisation quadratique sous contraintes inégalités 6.

Les $\alpha_i^\star \neq 0$ sont des vecteurs supports. Ce sont eux qui vont définir les hyperplans séparateurs. On remarque d'ailleurs que les $\alpha_i^\star = 0$ n'interviennent pas dans la formulation du classifieur optimal donnée par l'équation (9). L'exemple i dont le paramètre de Lagrange-Kuhn-Tucker associé est non nul se trouvera ainsi sur un des deux hyperplans séparateurs.

2.3 Fonction noyau

Nous venons de voir la classification par SVM dans le cas où le problème est linéairement séparable. Cependant, dans une grande majorité de cas, le problème n'est pas linéairement séparable. Il existe alors plusieurs alternatives. L'une d'elles consiste à changer d'espace de représentation. Il s'agit ainsi de transporter, par une application non linéaire, les données $x \in E$ dans un espace $\mathcal{E}$, de dimension plus grande que l'espace E où il sera alors possible de les séparer linéairement par un hyperplan de $\mathcal{E}$. Cela revient en fait à utiliser une fonction de décision non-linéaire, en ayant, d'un point de vu géométrique, une hypercourbe qui marque la frontière entre les deux classes d'exemples dans l'espace d'origine.

2.3.1 Définition

Nous appelons Φ, l'application permettant ce changement d'espace de représentation, c'est-à-dire l'application définie par :

$$\begin{aligned} \Phi : \quad & E \rightarrow \mathcal{E} \\ & x \rightarrow \Phi(x) \end{aligned}$$

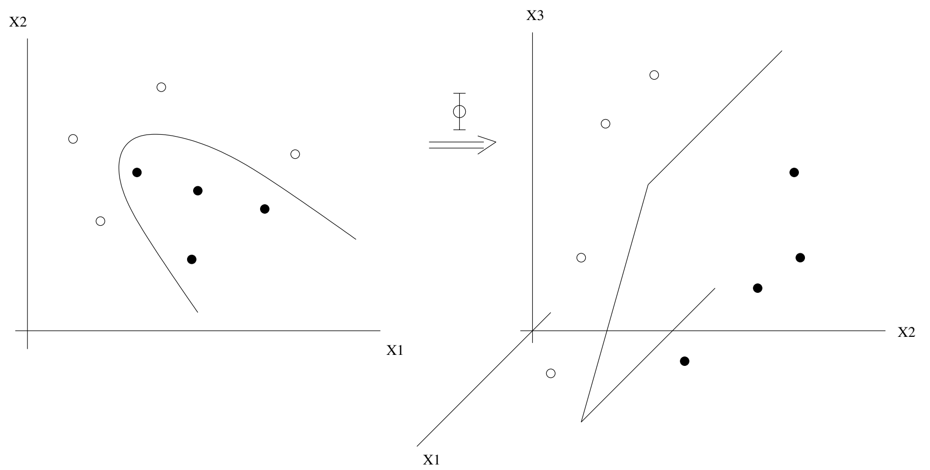


FIG. 2 – Changement d'espace de représentation par la fonction Φ afin de rendre le problème linéairement séparable

Comme nous pouvons le voir sur la figure 2, le nouvel espace des données $\mathcal{E}$ est de dimension supérieure à celle de l'espace de départ E. Les nouveaux axes contiennent une surgénération d'information par rapport aux précédents, ce qui permet idéalement d'effectuer une discrimination linéaire là où auparavant ce n'était pas possible.

Nous formulons ci-dessous le nouveau problème de classification par SVM résultant de ce changement d'espace de représentation.

Propriété 3 *Soient* $(E, \overline{\omega}, A, \mathcal{A}, \mathcal{W}_{\mathcal{L}})$*, un problème de classification supervisée binaire linéaire et* Φ*, une application de changement d'espace de représentation telle que le nouveau problème de classification supervisée binaire linéaire* $(\mathcal{E}, \widehat{\overline{\omega}}, \widehat{A}, \widehat{\mathcal{A}}, \widehat{\mathcal{W}_{\mathcal{L}}})$ *soit linéairement séparable. Le séparateur linéaire de marge maximale* $\widehat{\omega^\star}$ *est donnée par :*

$$\widehat{\omega^\star}(x) = sign_{I\!R^+}\left(\sum_{i=1}^{N} \alpha_i^\star y_i(\Phi(x_i)|\Phi(x)) - \widehat{\theta^\star}\right) \tag{10}$$

où les $\alpha_i^\star$ sont les paramètres de Lagrange-Kuhn-Tucker solutions du problème d'optimisation quadratique sous contraintes inégalités suivant :

$$\left\{ \begin{array}{l} \max_\alpha \sum_{i=1}^N \alpha_i \; - \; \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\Phi(x_i)|\Phi(x_j)) \\ sous \left\{ \begin{array}{l} \forall i, \; \alpha_i \geq 0 \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{array} \right. \end{array} \right. \tag{11}$$

Où $(.|.)$ *représente le produit scalaire euclidien associé à* $\mathcal{E}$
N est le nombre de paramètres de Lagrange-Kuhn-Tucker.

Il est à noter que le fait de se projetter dans un espace de représentation plus grand ne complexifie pas la résolution algorithmique du problème puisque la famille de paramètres de Lagrange-Kuhn-Tucker reste de taille N.
De plus, le nouveau problème, ainsi que sa solution, ne dépendent que des produits scalaires $(\Phi(x_i)|\Phi(x_j))$ pour $(x_i, x_j) \in E \times E$. Ainsi, bien souvent, plutôt que de chercher une application Φ, il est préférable de chercher directement une fonction K : $E \times E \rightarrow \mathbb{R}$, appelée **fonction noyau**, et définie par :

$$K(x_i, x_j) = (\Phi(x_i)|\Phi(x_j)) \text{ pour } (x_i, x_j) \in E \times E$$

La fonction noyau représente un produit scalaire dans le nouvel espace de représentation. Elle traduit donc la répartition des exemples dans cet espace. La fonction noyau matérialise une notion de proximité adaptée au problème, c'est-à-dire qu'elle caractérise une certaine similarité entre deux données de l'espace initial [Vert, 2003]. Un noyau peut donc se définir de deux manières : par une forme directe (en définissant directement la fonction K) ou par une forme indirecte (en utilisant la fonction de changement d'espace Φ et le produit scalaire associé au nouvel espace de représentation). Quand on le définit par sa forme directe, c'est-à-dire sans utiliser l'application Φ, un tel noyau doit répondre à certaines conditions : ce doit être une fonction symétrique définie positive comme nous le montre le théorème suivant.

Théorème 1 (Condition de Mercer) *Une fonction $K : E \times E \rightarrow \mathbb{R}$, symétrique, est un noyau si $\forall (x_i, x_j) \in E^2$, $(K(x_i, x_j))_{i,j}$ est une matrice définie positive.*

Dans ce cas, il existe un espace $\mathcal{E}$ et une fonction Φ tels que :

$$\forall (x_i, x_j) \in E^2,\ K(x_i, x_j) = (\Phi(x_i)|\Phi(x_j))$$

où $(.|.)$ représente le produit scalaire euclidien associé à $\mathcal{E}$.

La preuve de ce théorème peut être trouvée dans [Shawe-Taylor et Critianini, 2002]. La condition de Mercer permet ainsi de changer d'espace de représentation sans être obligé de définir la fonction Φ de changement d'espace. Il est à noter que la convexité du problème d'optimisation 11 est conserver (la fonction K étant définie positive, cela suffit à rendre le problème convexe).

Notons qu'un tel noyau K possède les trois propriétés fondamentales du produit scalaire :
$\forall (x_i, x_j) \in E^2$,
• $K(x_i, x_i) \geq 0$ (positivité)
• $K(x_i, x_j) = K(x_j, x_i)$ (symétrie)
• $|K(x_i, x_j)| \leq \|x_i\|.\|x_j\|$ (inégalité de Cauchy-Shwartz)

2.3.2 Noyau gaussien

Dans ce paragraphe, nous nous penchons sur un noyau très utilisé : le noyau gaussien. La forme générique de ce noyau est :

$$\forall (x_i, x_j) \in E^2,\ K(x_i, x_i) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

La fonction de changement d'espace Φ induit par ce noyau est un peu spéciale. En effet, l'exemple va être transporté sur une fonction gaussienne représentant la similarité de l'exemple avec tous les éléments de E. Le nouvel espace de representation $\mathcal{E}$ de la fonction Φ est de dimension infinie étant donné que Φ fait correspondre une fonction continue à chaque exemple. Le noyau gaussien permet donc de calculer des similarités dans un espace de dimension infinie ; pour plus d'informations, on consultera [Pontil et Verri, 1998].
Le noyau gaussien possède aussi la particularité d'être normé. En effet, $K(x, x) = 1$.

Le paramètre σ permet de régler la largeur de la gaussienne. En prenant σ grand, la similarité d'un exemple par rapport à ceux qui l'entourrent sera assez élevée ; à l'inverse, en prenant $\sigma \to 0$, l'exemple ne sera similaire à aucun autre. En resserrant la gaussienne, un classifieur pourra donc apprendre n'importe quelle base d'apprentissage sans commettre d'erreur. Il faut cependant se méfier du sur-apprentissage.

Il existe, par ailleurs, un noyau relativement similaire au noyau gaussien : le noyau laplacien. Il possède, en effet, les principales caractéristiques du noyau gaussien. La forme générique de ce noyau est :

$$\forall (x_i, x_j) \in E^2,\ K(x_i, x_i) = \exp\left(-\frac{\|x_i - x_j\|_1}{\sigma}\right)$$

Ce noyau utilise, contrairement au noyau gaussien, la norme N_1 pour comparer deux exemples. Cela assure une distinction moins brutale que pour un noyau gaussien (la décroissance exponentielle est, en effet, moins forte pour deux exemples non semblables).

2.3.3 Composition de noyaux

Un des principaux intérêts des noyaux est de pouvoir les combiner. En composant des noyaux, nous nous projettons d'espace de représentation en espace de représentation. Dans la pratique, nous combinons souvent des noyaux simples pour en obtenir des plus complexes, s'adaptant mieux aux problèmes de classification. Cela permet, en effet, d'améliorer les paramètres discriminants de séparation des classes. Nous allons voir plusieurs opérations appliquées aux noyaux, notamment la composée de noyaux dans la propriété qui suit.

Propriété 4 *Soient deux noyaux* $K_1 : E \times E \to I\!R$ *et* $K_2 : \mathcal{E} \times \mathcal{E} \to I\!R$ *associés respectivemment aux fonctions de changement d'espace* $\Phi_1 : E \to \mathcal{E}$ *et* $\Phi_2 : \mathcal{E} \to \widetilde{\mathcal{E}}$, *c'est-à-dire tels que* $\forall (x_i, x_j) \in E^2$ *et* $\forall (z_i, z_j) \in \mathcal{E}^2$, $K_1(x_i, x_j) = (\Phi_1(x_i)|\Phi(x_j))$ *et* $K_2(z_i, z_j) = (\Phi_2(z_i)|\Phi(z_j))$.
Alors, La fonction $K : E \times E \to I\!R$ *définie par* $\forall (x_i, x_j) \in E^2$, $K(x_i, x_j) = (\Phi(x_i)|\Phi(x_j))$ *où*

$$\Phi : \begin{array}{l} E \rightarrow \widetilde{\mathcal{E}} \\ x \rightarrow \Phi(x) = \Phi_2 \circ \Phi_1(x) \end{array}$$

est un noyau.
Dans ce cas, $\forall (x_i, x_j),\ K(x_i, x_j) = K_2(\Phi_1(x_i)|\Phi_1(x_j))$.

Cette propriété est très intéressante puisqu'elle permet de composer deux noyaux sans pour autant connaitre, à la fois la forme directe et la forme indirecte de ces deux noyaux. En effet, la forme indirecte du premier noyau (K_1) et la forme directe du deuxième (K_2) permettent de définir, sous sa forme directe, le noyau résultant de leur composée. Cela peut s'avérer utile car de nombreux noyaux ne sont facilement exprimables que sous une seule de leurs formes.
Il existe, par ailleurs, d'autres opérations sur les noyaux comme nous le montre la propriété suivante.

Propriété 5 *Soient* $K_1 : E \times E \rightarrow I\!R$ *et* $K_2 : E \times E \rightarrow I\!R$*, deux noyaux, c'est-à-dire deux fonctions vérifiant la condition de Mercer.*
soient $a \in I\!R^+$ *et* B *une matrice définie positive.*
Alors, les fonctions suivantes définies par, $\forall (x_i, x_j) \in E^2$
$\bullet K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j)$
$\bullet K(x_i, x_j) = a.K_1(x_i, x_j)$
$\bullet K(x_i, x_j) = K_1(x_i, x_j).K_2(x_i, x_j)$
$\bullet K(x_i, x_j) =^T x_i.B.x_j$
sont des ***noyaux****.*

Les deux propriétés précédentes se démontrent facilement par calculs matriciels. En résumé, l'introduction d'un noyau dans une classification par SVM a bien souvent un objectif double : il permet, premièrement, de rendre le problème linéairement séparable en se projetant dans un espace de représentation de dimension plus grande. Il permet aussi de faciliter la classification en lui associant des caractéristiques pertinentes de différenciation des classes, le noyau est ainsi le coeur du mécanisme puisqu'il caractérise la similarité entre deux données. Dans la suite de notre étude, nous utiliserons des compositions de noyaux afin de définir des caractéristiques texturales pertinentes dans notre problème de classification du feu et du non-feu.

2.4 Méthode des marges relaxées

L'introduction d'un noyau dans une classification par SVM ne rend pas pour autant le problème forcément linéairement séparable. Il est cependant possible d'utiliser une méthode consistant à relaxer la contrainte de la marge du classifieur par l'introduction d'une variable de relaxation. Cela signifie que l'on tolère un certain manque de consistance avec la base d'apprentissage. On admet donc que certains exemples soient mal classés. Il est à noter que même dans une situation où le problème est linéairement séparable, il peut être préférable d'utiliser cette méthode : on viole quelques contraintes pour élargir les marges de l'ensemble des séparateurs linéaires et gagner en robustesse ce que l'on perd en erreur d'apprentissage.

Du point de vue de la formulation de notre problème, la méthode des marges relaxées ("Soft Margin" en anglais) revient à introduire ce que l'on appelle des variables d'écart dans les contraintes de séparations par deux hyperplans. L'équation 4 s'écrit alors sous la forme :

$$\left\{ \begin{array}{l} \forall (x_i, y_i) \in \mathcal{A},\ y_i(w.x_i - \theta) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{array} \right. \tag{12}$$

Où les ξ_i sont les variables d'écart.

Le problème d'optimisation 5 devient alors :

$$\left\{ \begin{array}{l} \min_{w,\xi,\theta}\ \frac{1}{2} \sum_{k=1}^{n} w_k^2 + C \sum_{k=1}^{n} \xi_k \\ sous\, \forall (x_i, y_i) \in \mathcal{A},\ y_i(w.x_i - \theta) \geq 1 - \xi_i\ ,\ \xi_i \geq 0 \end{array} \right. \tag{13}$$

Comme précédemment, on écrit le problème sous la formulation duale de Wolfe. Des informations sur les calculs de passage peuvent être consultés dans [Samuelides, 2004]. On peut naturellement appliquer la méthode des marges relaxées à la classification par noyau. Le problème d'optimisation 11 va alors s'écrire sous la forme :

$$\left\{ \begin{array}{l} \max_{\alpha} \sum_{i=1}^{N} \alpha_i\ -\ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\Phi(x_i)|\Phi(x_j)) \\ sous \left\{ \begin{array}{l} \forall i,\ 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{array} \right. \end{array} \right. \tag{14}$$

Le problème est quasiment inchangé mis à part que l'on a maintenant deux contraintes inégalités pour chaque α_i. Les α_i sont en effet maintenant majorés par la constante C. La saturation de $\alpha_i \geq 0$ correspond toujours au fait que l'exemple i est un vecteur support (il se trouve sur un des deux hyperplans). La saturation de $\alpha_i \leq C$ correspond à un exemple i en dehors de la marge.
Dans la suite de notre étude, cette méthode des marges relaxées sera automatiquement appliquée avec une classification par SVM.

2.5 Problème de classification par SVM en imagerie 2D

Les SVM sont bien adaptés aux problèmes de classification en imagerie 2D. Dans ces problèmes, on cherche à classifier les pixels d'une image selon certaines caractéristiques. Quand celles-ci sont d'ordre purement radiométriques (c'est-à-dire ne dépendant que de l'intensité du pixel en question), on prend comme ensemble E des données à classifier (c'est-à-dire comme espace de représentation) l'ensemble des pixels de l'image : une donnée ou un exemple correspond alors à l'observation d'un pixel. Ce cas correspond à l'espace de représentation initial le plus simple que l'on puisse trouver. En appelant S, l'ensemble des sites d'une image et x_s, l'observation associée au site s, l'ensemble des données E s'exprime par $E = \{x_s / s \in S\}$.
Quand ces caractéristiques sont plus élaborées, notamment quand on se place dans un cadre markovien, il faut tenir compte de l'interaction entre pixels voisins. Dans ce cas, nous ne pouvons plus utiliser l'espace de représentation initial définit précédemment car un pixel n'est plus une entité indépendante, mais dépend de son voisinnage. Une donnée ou un exemple de E correspond alors à l'observation d'un pixel ainsi qu'à celle de ses voisins. En notant ν_s, le voisinage du site s, l'espace de représentation E s'exprime alors par $E = \{\{x_s\} \cup \{x_r / r \in \nu_s\} / s \in S\}$.

Le but de notre étude est de classifier une image selon, en autres, des caractéristiques texturales. Le cadre markovien semble donc bien adapté. Il nous faut pour cela définir un système de voisinage.

Définition 6 $V = \{V(s)/s \in S\}$ *est un* ***système de voisinage*** *si*

- $s \notin V(s)$
- $s \in V(t) \iff t \in V(s)$

De plus, on appelle ***clique****, un sous-ensemble c de S dont les sites sont voisins deux à deux. On note C, l'ensemble de toutes les cliques asociées à (S, V).*

Définition 7 *On définit le* ***voisinage*** ν_s^k *du site s et de taille $k \geq 1$, comme l'ensemble des sites présents dans la fenêtre carrée de longeur $2k+1$ et centrée sur le site s.*

Nous utiliserons le voisinage ν_s^k dans notre étude sur les modèles primaires de noyaux à texture.

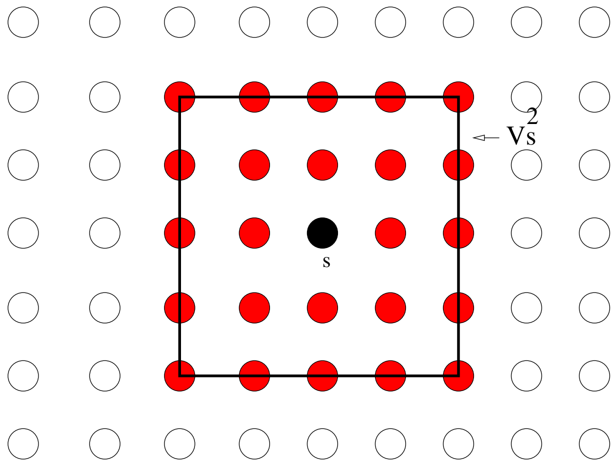


FIG. 3 – Voisinage ν_s^k dans le cas $k = 2$

Définition 8 *On définit également $\mathcal{C}_{s,t}^k$, l'ensemble des couples de sites (r, q) où $r \in \nu_s^k$ et $q \in \nu_t^k$ tels que r et q occupent des places similaires dans leur fenêtre respective.*

Définition 9 *On définit, enfin, la* ***moyenne d'un voisinage*** μ_s *par la moyenne des observations du voisinage ν_s^k, c'est-à-dire par :*

$$\mu_s = \frac{1}{card\, \nu_s^k} \sum_{r \in \nu_s^k} x_r$$

Le voisinage que nous définissons ainsi dépend du paramètre k. L'espace de représentation que nous utiliserons dans notre étude dépendra donc lui aussi de ce paramètre. Finalement, l'espace de représentation retenu est :

$$E_k = \{\{x_s\} \cup \{x_r / r \in \nu_s^k\} / s \in S\} \tag{15}$$

Dans notre cas où l'observation x_s d'un site s est une valeur réelle (cas des images panchromatiques), la dimension de l'espace de représentation E_k est $(2k+1)^2$. Notre problème de classification binaire est donc numérique et le nombre d'attributs est $(2k+1)^2$. Nous considérons, de plus, qu'une observation x est contenue dans C, un compact de $\mathbb{R}$. Nous avons alors $E_k = C^{(2k+1)^2}$.

Un élément X_s de E_k représentera un vecteur de $(2k+1)^2$ éléments de C. Il contiendra l'observation x_s ainsi que toutes ses observations voisines.

La base d'apprentissage $\mathcal{A}$ du problème est constituée d'une série d'imagettes extraites de l'image originale, auxquelles sont associées les imagettes de la classification binaire correspondante. Le nombre et la taille de ces imagettes sont très importants : ils définissent le cardinal de $\mathcal{A}$, déterminant à la fois dans la complexité du problème, mais aussi dans le niveau de difficulté de résolution du problème d'optimisation 6 (le nombre de paramètres de Lagrange-Kuhn-Tucker étant donné par le cardinal de $\mathcal{A}$).

L'algorithme utilisé afin de résoudre le problème d'optimisation (11) est un algorithme de type ICM ("Iterated Conditionnal Modes"), présenté dans [Besag, 1986], auquel nous ajoutons des contraintes de tirage. En effet, lors de chaque itération de cet algorithme, nous réduisons l'espace des tirages possibles pour tendre vers l'hyperplan que définit la contrainte du problème 11. Par ailleurs, nous appliquerons la méthode des marges relaxées à ce problème d'optimisation afin d'obtenir une meilleur optimisation et des résultats plus souples.

3 Noyaux texturaux

La classification par SVM est bien adaptée pour traiter des données de grande dimension. Nous allons donc utiliser cette méthode afin de détecter des objets sur des images de grande taille et à haute résolusion. Le but est de faire apprendre par SVM sur une imagette (i.e. la base d'exemple) ce qu'est l'objet et plus précisément, nous souhaitons mettre en place un noyau capable de le distinguer du reste de l'image par l'utilisation d'un critère radiométrique et d'un critère textural. Le paragraphe suivant nous apporte des informations sur cette notion de texture.

3.1 Définition de la texture

La texture est un concept intuitif. Il n'en existe pas de définition formelle ou universelle. Nous pouvons cependant en dégager deux idées. Dans la première, la texture s'apparente à une variation des niveaux de gris entre pixels voisins (cela caractérise en fait l'hétérogénéité à l'échelle de la résolution de l'image). La deuxième consiste à caractériser la texture par une notion d'homogénéité à une echelle plus grande que la résolution.

Une texture présente, à une échelle donnée, le même aspect quelque soit la zone observée. Dans ces conditions, on considère l'image comme la réalisation d'un processus stochastique stationnaire : c'est-à-dire que les statistiques calculées sont invariantes en translation.

Revenons sur les deux types de définition de la texture :

• La première est déterministe et fait référence à une répétition spatiale d'un motif de base dans différentes directions. Cette approche structurelle correspond à une vision macroscopique des textures. C'est le cas, par exemple, des tissus ou de la peau des reptiles. Ainsi, B. Julesz, dès 1962, considère la texture comme l'agencement d'un motif de base qu'il appelle "texton" [Julesz, 1962].

• La deuxième est probabiliste et cherche à caractériser l'aspect anarchique et homogène qui ne comprend ni de motif localisable, ni de fréquence de répétition principale (cas par exemple de l'herbe). Elle correspond à une vision microscopique.

Gagalowicz propose une synthèse des deux approches en considérant la texture comme "une structure spatiale constituée de l'organisation de primitives (ou motifs de base) ayant chacune un aspect aléatoire" [Gagalowicz, 1983]. C'est donc une structure hiérarchique à deux niveaux.

Il existe plusieurs types de méthodes d'analyse de la texture, notamment les méthodes de filtrage ou les méthodes structurelles-géométriques. Les outils les plus utilisés sont la transformée de Fourrier ou les matrices de cooccurence.

3.2 Noyau textural : modèle primaire

Dans ce paragraphe, nous proposons un premier modèle de noyau textural : le modèle primaire. Nous nous plaçons dans l'espace de représentation défini par 15, c'est-à-dire dans $E_k = \{\{x_s\} \cup \{x_r/r \in \nu_s^k\}/s \in S\}$.
L'idée consiste à faire ressortir une information de texture en comparant les observations du voisinage d'un site avec l'observation de ce site. Pour ce faire, nous nous projettons dans un espace de représentation $\mathcal{E}_k$ dans lequel les dimensions ajoutées font ressortir cette information.

$$\mathcal{E}_k = E_k \times \Sigma_k$$

avec

$$\Sigma_k = \{\{\gamma|x_r - x_s|/r \in \nu_s^k\}/s \in S\}$$

où γ est un paramètre réel positif pondérant l'importance des nouvelles dimensions introduites par rapport aux anciennes.

En notant X_s le vecteur constitué des $(2k+1)^2$ observations du voisinage ν_s^k, nous définisons Φ_1, fonction de changement d'espace de représentation par

$$\begin{aligned} \Phi_1 : \quad & E_k \rightarrow \mathcal{E}_k \\ & X_s \rightarrow \Phi_1(X_s) = \begin{pmatrix} X_s \\ \gamma|X_s - x_s.Id| \end{pmatrix} \end{aligned}$$

Le nouvel espace de représentation a une dimension deux fois plus grande que l'espace de représentation initial. les nouvelles dimensions introduites sont beaucoup plus pertinentes et font ressortir une certaine notion de texture que nous recherchons. Cependant, ce noyau est défini par un simple produit scalaire entre deux éléments du nouvel espace : cela n'est pas pertinent pour définir la similarité de ces deux éléments.

En nous appuyant sur la propriété (4) concernant la composée de noyaux, nous établissons alors le noyau K par la composée d'un noyau gaussien K_g avec le noyau K_1 précédemment défini :

$K(X_s,X_t) = K_g(\Phi_1(X_s),\Phi_1(X_t)) = \exp\left(-\frac{\|\Phi_1(X_s)-\Phi_1(X_t)\|^2}{2\sigma^2}\right)$ pour $(X_s,X_t) \in E_k^2$

le noyau K s'exprime de manière plus explicite sous la forme :

$\forall (X_s,X_t) \in E_k^2$,

$$K(X_s,X_t) = \exp\left(-\frac{1}{2\sigma^2}\sum_{(r,q)\in\mathcal{C}_{s,t}^k}(x_r-x_q)^2+\gamma^2(|x_r-x_s|-|x_q-x_t|)^2\right) \quad (16)$$

Le premier terme présent dans l'exponentielle provient de l'espace de représentation initial E_k : ce terme ne caractérise aucune notion de texture, mais définit cependant une information radiométrique. Le deuxième terme, provenant de l'espace Σ_k, caractérise une certaine notion de texture : il quantifie, en effet, les variations de niveaux de gris dans deux voisinages, puis les compare. Nous voyons dans cette formulation l'importance du facteur pondérateur γ.

Un tel noyau correspond, a priori, à nos attentes : il renvoie une valeur proche de 1 quand deux voisinages sont proches selon des caractéristiques de texture et de radiométrie pré-définies, et une valeur proche de 0 quand ces deux voisinages sont différents. Ce noyau permet, de plus, de calculer des similarités dans un espace de dimension infinie.

3.3 Noyau textural : modèle avancé

Nous proposons dans cette partie un noyau textural fondé sur les champs de Markov et plus précisement sur l'estimation de paramètres de texture dans un cadre markovien. Pour cela, nous nous appuyons sur les travaux de Xavier Descombes [Descombes, 1993] et d'Anne Lorette [Lorette, 1999]. Rappelons tout d'abord quelques informations utiles sur les champs de Markov. Pour des informations complémentaires, le lecteur est invité à consulter [Winkler, 1995] et [Li, 1995].

3.3.1 Champs de Markov

Définition 10 *Soit* $X = (X_s)_{s\in S}$ *un champ aléatoire associé à l'espace probabilisé complet* $(\Omega, \mathcal{F}, P)$.

X est un **champ de Markov** *relativement au système de voisinage V si*

$$\forall x \in \Omega, \forall s \in S, P(X_s = x_s | X_r = x_r, r \in S - \{s\}) = P(X_s = x_s | X_r = x_r, r \in V(s))$$

Cela signifie que l'observation x_s d'un élément du champ X ne dépend que des observations voisines de s. En imagerie 2D, un champ de Markov est donc un champ à mémoire spatiale locale.
Le théorème suivant permet de caractériser la distribution d'un tel champ : c'est le théorème d'Hammersley-Clifford.

Théorème 2 (Hammersley-Clifford) *Soit X un champ de Markov associé à l'espace probabilisé* $(\Omega, \mathcal{F}, P)$*, et tel que* $\forall x \in \Omega, P(X = x) > 0$*.*
Alors, la distribution $P(X)$ *de ce champ est une distribution de Gibbs :*

$$P(X) = \frac{\exp -U(X)}{Z}$$

où $U(X) = \sum_{c \in C} V_c(X_s, s \in c)$ *et* $Z = \sum_{X \in \Omega} \exp -U(X)$

3.3.2 Paramètre de texture avancé

[Descombes *et al.*, 1999] proposent un paramètre de texture défini en modélisant l'image par un modèle markovien gaussien ; La distribution conditionnelle $P(X_s | X_r, r \in V(s))$ s'exprimant sous la forme :

$$P(X_s | X_r, r \in V(s)) = \frac{1}{Z_{V(s)}} \exp \left(-\beta(\sum_{r \in V(s)} (X_s - X_r)^2 + \lambda (X_s - \mu)^2) \right)$$

où
λ et β sont les paramètres du modèle,
μ, la moyenne locale,
$Z_{V(s)}$, la fonction de partition locale.

[Descombes, 1993] a montré que cette distribution conditionnelle dépend uniquement de la moyenne m_s des voisins de s, ce qui rend les estimateurs plus robustes.

Nous avons entre autre :

$$P(X_s|X_r, r \in V(s)) = \frac{1}{Z_{V(s)}} \exp\left(-\beta(4+\lambda)\left(X_s - \frac{4m_s + \lambda\mu}{4+\lambda}\right)^2\right) = P(X_s|m_s)$$

En se plaçant dans un cadre stationnaire (c'est-à-dire que la moyenne locale μ est indépendante de s), la probabilité conditionnelle $P(X_s|X_r, r \in V(s))$ suit alors une loi normale de moyenne $\frac{4m_s+\lambda\mu}{4+\lambda}$ et de variance $\frac{1}{2\beta(4+\lambda)}$.

$$P(X_s|X_r, r \in V(s)) \equiv \mathcal{N}\left(\frac{4m_s + \lambda\mu}{4+\lambda}, \frac{1}{2\beta(4+\lambda)}\right)$$

Les paramètres de texture nous intéressant vont alors être les variances conditionnelles locales $\sigma^2_{P(X_s|m_s)}$. Elles caractérisent, en effet, une certaine homogénéité dans le voisinage du site en question. Cette variance conditionnelle s'exprime par les paramètres du modèle. [Descombes, 1993] propose une estimation des variances conditionnelles locales par la méthode des "queues de comètes". Cette méthode consiste à estimer les matrices des probabilités conditionnelles dans une fenêtre glissante. La taille de cette fenêtre doit être suffisamment grande pour que les statistiques soient robustes et aussi suffisamment petite pour éviter les mélanges de texture. Pour chaque position de la fenêtre, on estime alors $\sigma^2_{P(X_s|m_s)}$ à partir de la matrice de probabilités conditionnelles associée.
[Descombes, 1993] a proposé un modèle isotrope en prennant un voisinage multi-directionnel, c'est-à-dire sans favoriser certaines directions dans le voisinage. Un tel modèle ne semble pas a priori répondre à nos besoins puisque nous cherchons un paramètre de texture qui ne soit pas réactif à des structures linéiques tels ques les réseaux routiers : nous voulons donc un paramètre de texture anisotrope, c'est-à-dire qui tienne compte des directions. [Lorette, 1999] a repris les travaux de X. Descombes en introduisant ce caractère anisotrope dans son modèle. Nous n'avons plus alors une variance conditionnelle associée à chaque site mais huit, correspondant à une direction établie via un des huit voisinages. A. Lorette définit ainsi huit voisinages V_d associés à huit directions d comme nous le montre la figure 4.

Pour chacune des huit directions d, nous aurons une probabilité contitionnelle $P(X_s|X_r, r \in V_d(s))$ s'exprimant par :

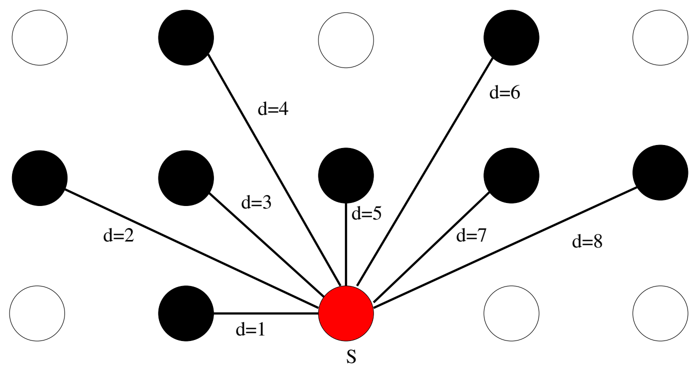


FIG. 4 – Représentation des 8 directions prises en compte

$$P(X_s|X_r, r \in V_d(s)) = \frac{1}{Z_{V_d(s)}} \exp\left(-\beta_d(\sum_{r \in V_d(s)} (X_s - X_r)^2 + \lambda_d(X_s - \mu)^2)\right)$$

Comme dans le cas isotrope, cette probabilité conditionnelle de direction d suit une loi normale :

$$P(X_s|X_r, r \in V_d(s)) \equiv \mathcal{N}\left(\frac{4m_s^d + \mu\lambda_d}{4 + \lambda_d}, \frac{1}{2\beta_d(4 + \lambda_d)}\right)$$

Pour chacun de ces huit modèles, il est établi un paramètre de texture suivant la direction d qui correspond à la variance conditionnelle $\sigma_d^2 = \frac{1}{2\beta_d(4+\lambda_d)}$. Cette variance est estimée par la même méthode que pour le modèle isotrope (c'est-à-dire par la méthode des queues de comète). Il a cependant fallu normaliser les paramètres : les sites des huit voisinages du site s se situent, en effet, à des distances différentes de s. Dans la suite de notre étude, nous appélerons $\sigma^2_{(d),s}$, le paramètre de texture correspondant à cette variance conditionnelle associée au site s, de direction d et dont l'estimation a été calculée dans une fenêtre carrée de taille k. Par ailleurs, nous noterons D, l'ensemble des huit directions.

3.3.3 Noyau à texture "avancé"

De la même manière que pour les modèles précédents, nous allons définir un noyau basé sur ce paramètre de texture. Nous nous projetons, tout d'abord, dans l'espace de représentation $\mathcal{E}_k = E_k \times \Sigma$ où Σ est défini par :

$$\Sigma = \{\{\gamma\sigma^2_{(d),s} \,/ d \in D\} \,/ s \in S\}$$

Où γ est un paramètre réel positif pondérant l'importance des nouvelles dimensions introduites par rapport aux anciennes.

En composant le noyau qui en résulte par un noyau gaussien, nous obtenons le noyau K s'exprimant par :

$\forall (X_s, X_t) \in E_k^2,$

$$K(X_s, X_t) = \exp\left(-\frac{1}{2\sigma^2}\sum_{(r,q)\in \mathcal{C}^k_{s,t}} (x_r - x_q)^2 + \gamma^2 . \sum_{d \in D} (\sigma^2_{(d),s} - \sigma^2_{(d),t})^2\right) \qquad (17)$$

4 Application à la détection de feux de forêt

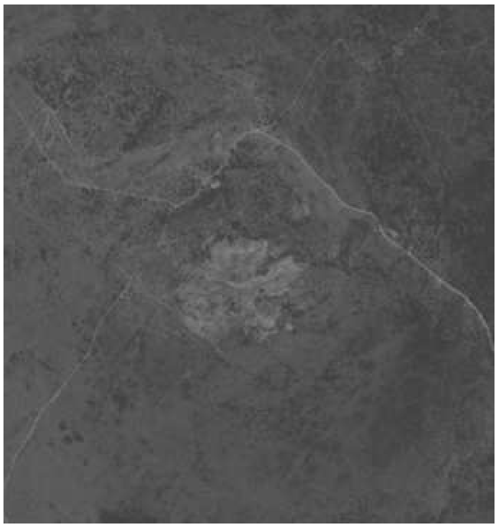
L'idée consiste à utiliser une classification par SVM avec les noyaux précédemment établis afin de détecter la fumée associée aux feux. Ces noyaux possèdent deux termes : l'un est lié à une certaine notion de texture (l'homogénéité dans un voisinage ; terme utilisé dans la détection des zones urbaines), l'autre correspond à une information radiométrique (teinte générale de la zone). La seule information texturale ne suffit pas à détecter la fumée des incendies. Les deux informations de texture et de radiométrie sont primordiales pour caractériser la fumée associée aux feux de forêt. En effet, il est nécessaire de caractériser la fumée par sa texture mais aussi par sa teinte, et ce afin de ne pas confondre la fumée associée à un feu de forêt et un nuage par exemple. Il s'agit alors notamment de fixer de manière pertinente le paramètre pondérateur γ.

Les données utilisées pour cette application sont des images SPOT5 (résolution 5 mètres) de la région PACA (Maures et Esterel) durant les feux de forêt du mois d'août 2003.

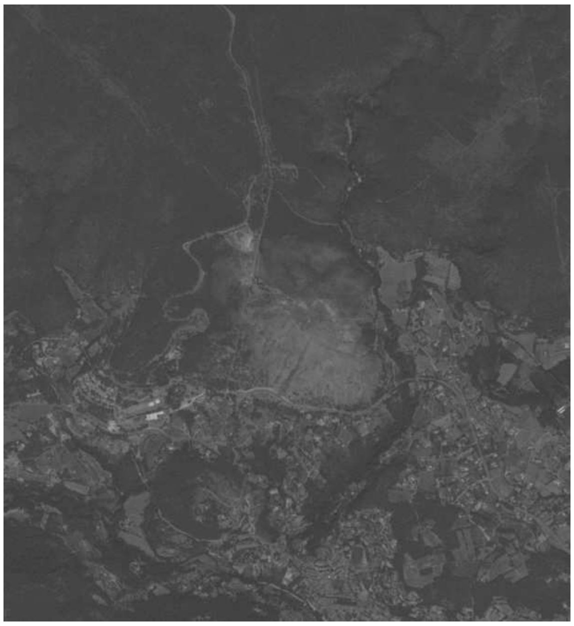
Les résultats suivants ont été obtenus avec une base d'apprentissage constituée de 2 imagettes (figure 5 - c et d). L'une représente une portion de fumée d'un feu ainsi qu'une portion de plaine ; cette imagette est extraite de l'extrait 1 (figure 5 - a). L'autre imagette représente une portion de nuage : elle a été introduite afin de ne pas confondre les nuages avec la fumée dégagée par les feux.

Les résultats obtenus sur l'extrait 1 sont très satisfaisants, aussi bien avec le modèle local qu'avec le modèle avancé (figure 6). Les contours trouvés montrent une très bonne précision de la détection. De plus, le nombre de fausses alarmes est nul. La base d'apprentissage a donc été bien apprise et les deux noyaux établis semblent pertinents pour ce problème puisqu'à partir d'une petite zone de fumée de l'extrait 1, nous sommes capables de retrouver entièrement la fumée des feux sur l'extrait 1.

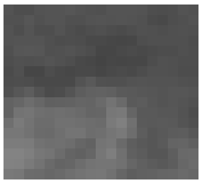
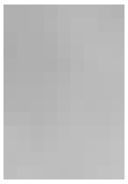
Les résultats obtenus sur l'extrait 2 (figure 7) sont beaucoup plus révélateurs de l' efficacité de la méthode proposée. En effet, il s'agit alors de détecter la fumée associée aux feux sur une image ne contenant pas la base d'apprentissage : c'est la véritable phase de test. Les résultats obtenus sont différents selon les deux modèles. Le modèle local donne des résultats très moyens : nous détectons, certes, la zone de fumée mais avec une précision moyenne au niveau des contours. Il y a, de plus, un nombre de fausses alarmes particulièrement élevé (6 fausses détections). Le modèle avancé fournit, quant-à lui, de très bons résultats : la zone de fumée est détectée avec beaucoup de précision sur les contours. Le nombre de fausses alarmes est, par ailleurs, relativement faible (2 fausses alarmes).
Nous avons, de plus, tester la détection de nuages : le résultat a été convaincant puisqu'ils n'ont été détectés par aucun des deux modèles. Cela permet de confirmer le bon apprentissage de la base.



(a)



(b)

(c) (d)

FIG. 5 – **(a)** extrait 1 contenant un feu (image SPOT5 - programme ISIS). **(b)** extrait 2 contenant un feu (image SPOT5 - programme ISIS). **(c)** et **(d)** base d'apprentissage considérée (c-feux et plaine / d-nuage)

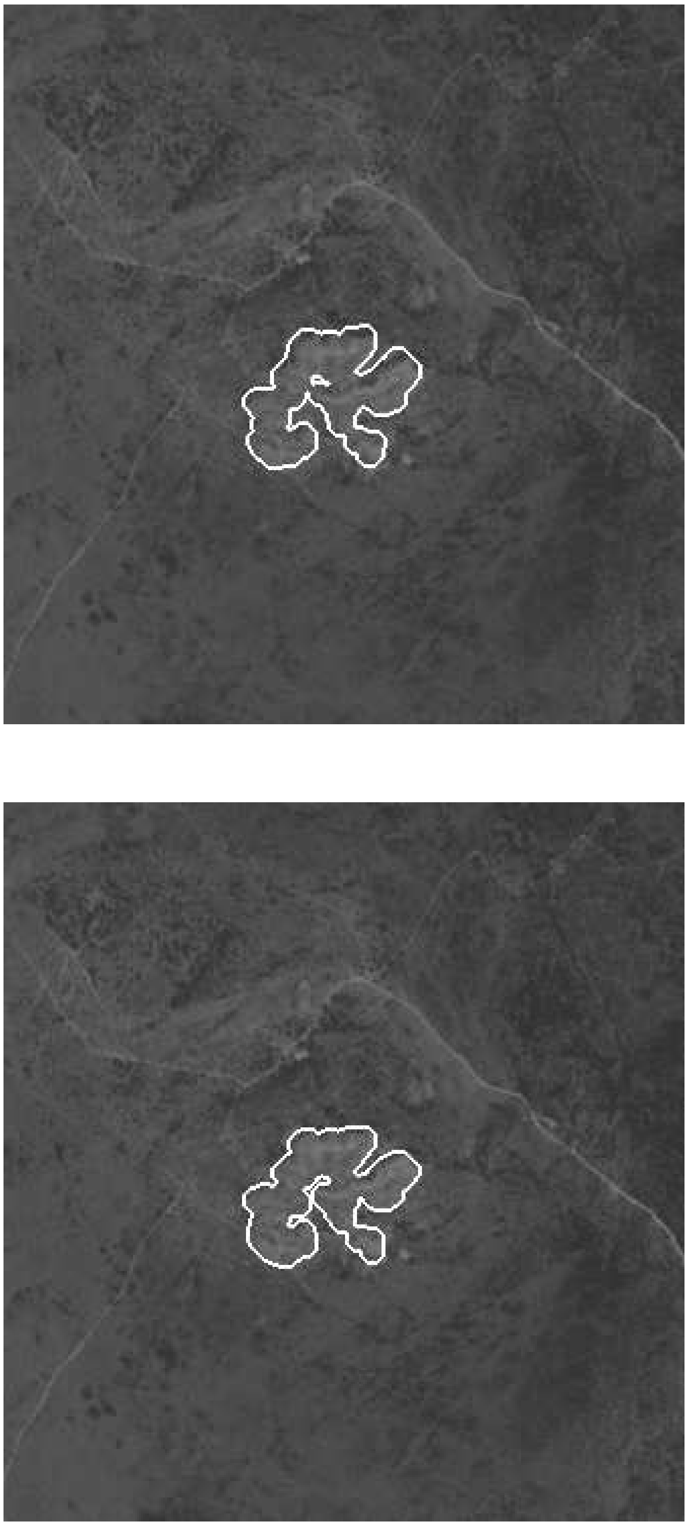
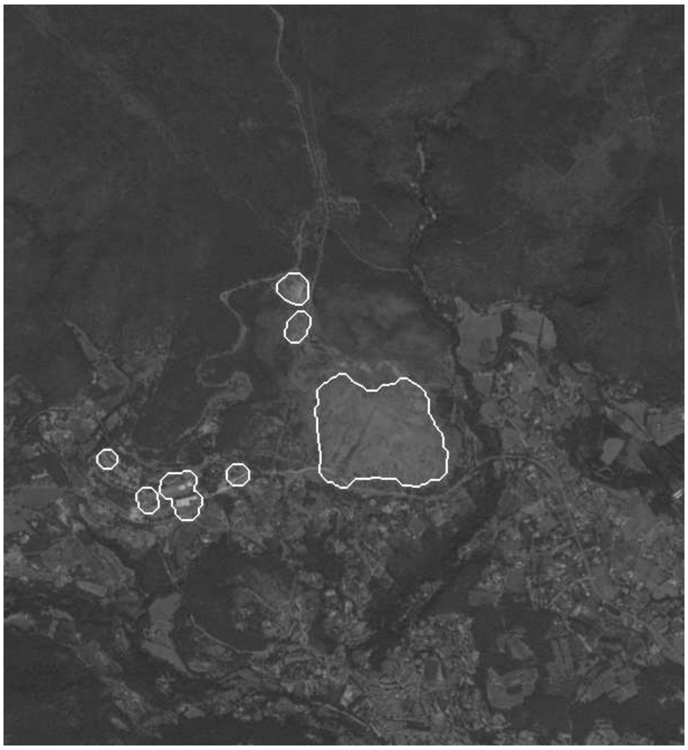


FIG. 6 – **En haut** : résultat obtenu sur l'extrait 1 par le modèle local. **En bas** : résultat obtenu sur l'extrait 1 par le modèle avancé



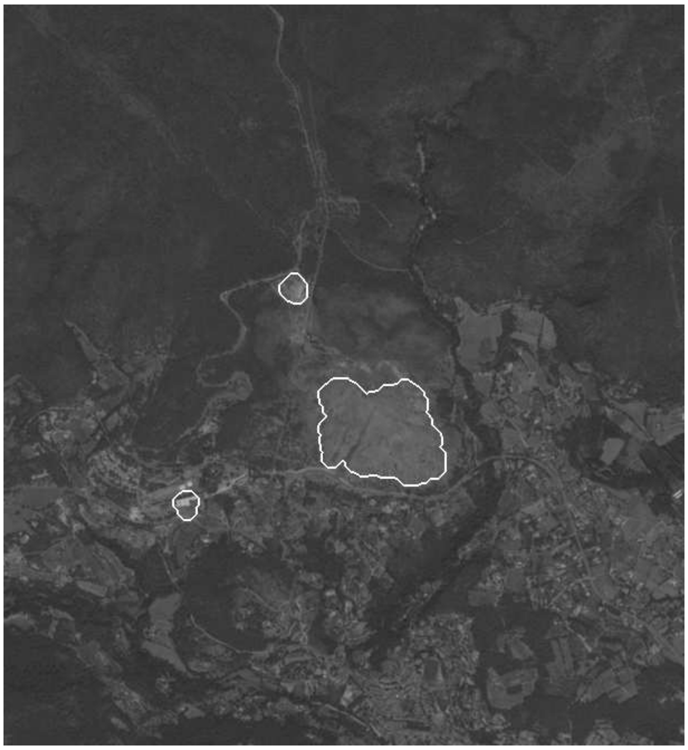
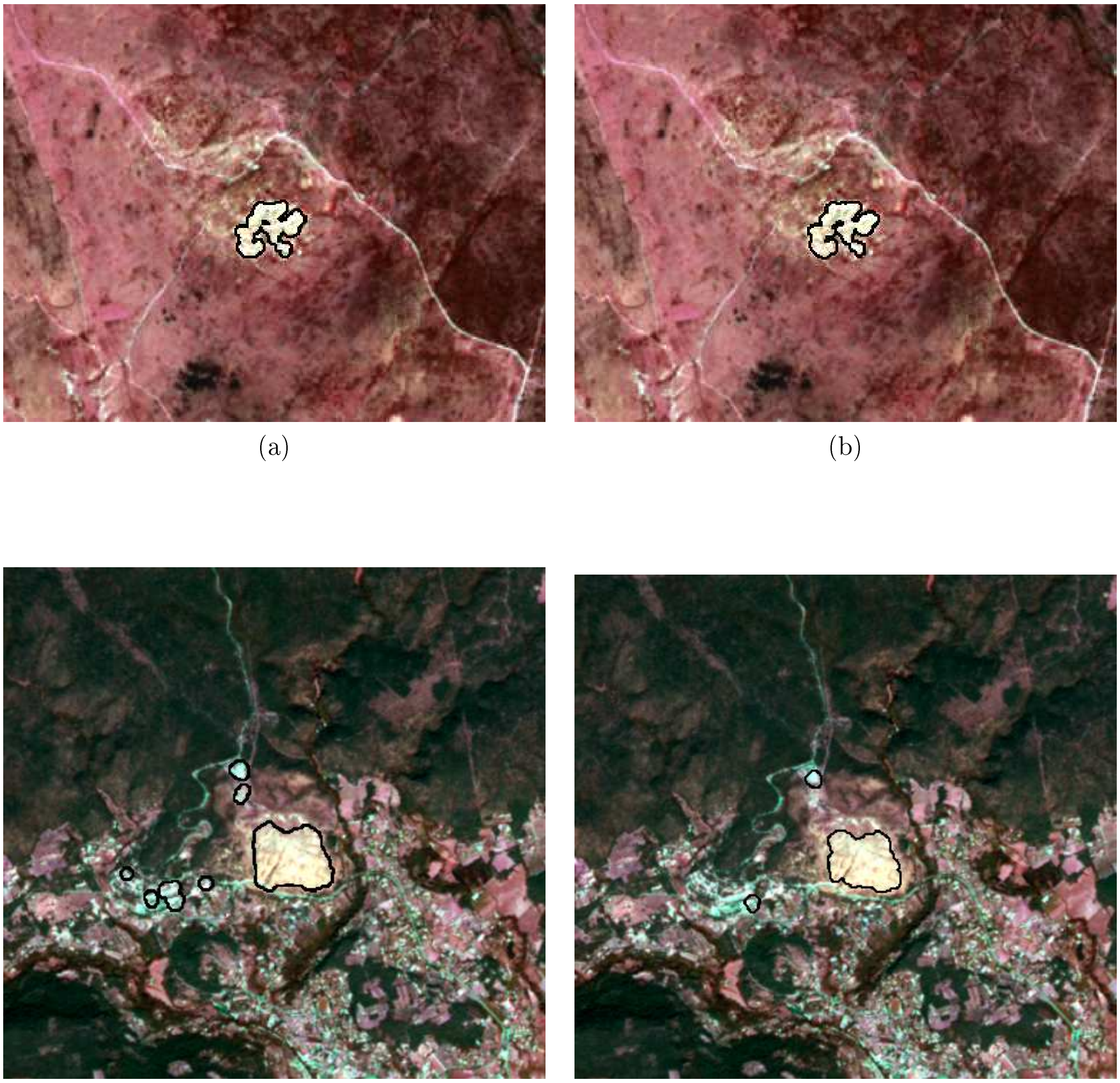


FIG. 7 – **En haut** : résultat obtenu sur l'extrait 2 par le modèle local. **En bas** : résultat obtenu sur l'extrait 2 par le modèle avancé



(a) (b)

(c) (d)

FIG. 8 – Résultats couleur obtenus **(a)** sur l'extrait 1 par le modèle local. **(b)** sur l'extrait 2 par le modèle local. **(c)** sur l'extrait 1 par le modèle avancé. **(d)** sur l'extrait 2 par le modèle avancé

5 Application à la détection de zones urbaines

Nous avons aussi testé la pertinence de ces noyaux sur un problème de classification de zones urbaines. Dans les villes, il y a une forte hétérogénéité de l'intensité des pixels. A l'inverse, les zones non-urbaines telles que les champs sont relativement homogènes. Dans ce type de problème, le critère radiométrique n'est pas utile : seul compte le critère textural.
Il est donc essentiel que le paramètre pondérateur γ soit très grand devant 1 afin que le terme radiométrique soit négligeable . Pour ce faire, nous posons $\sigma = \gamma.\sigma'$ et faisons tendre γ vers l'infini. Le noyau primaire s'exprime alors par :

$\forall (X_s, X_t) \in E_k^2$,

$$K(X_s, X_t) \approx \exp\left(-\frac{1}{2\sigma'^2} \sum_{(r,q)\in \mathcal{C}_{s,t}^k} (|x_r - x_s| - |x_q - x_t|)^2\right) \tag{18}$$

Le noyau avancé s'exprime, quant-à lui, par :

$\forall (X_s, X_t) \in E_k^2$,

$$K(X_s, X_t) \approx \exp\left(-\frac{1}{2\sigma'^2} \sum_{d\in D} (\sigma^2_{(d),s} - \sigma^2_{(d),t})^2\right) \tag{19}$$

Les images utilisées sont des simulations SPOT5 de Toreilles et d'Eppeville fournies par le CNES (résolution 5m). La base d'apprentissage utilisée est une petite partie des images de taille 20×23, soient 460 exemples ; elle est constituée d'un bout de zone urbaine et d'un bout de zone non-urbaine (figure 9).

Le résultat obtenu par le modèle primaire fait apparaître de manière correcte la structure urbaine de l'image. Nous avons amélioré les résultats en faisant valoir une notion de connexité dans la classification obtenue. Par des ouvertures et fermetures morphologiques, nous avons, en effet, favoriser les classes connexes et denses. Nous éliminons pour cela les pixels isolés du point de vue de la classe. Une ouverture et fermeture morphologiques [Soille, 2003] permet donc de corriger les erreurs de classification du SVM, c'est-à-dire les données mal classées. La figure 10 présente le résultat obtenu suite à l'utilisation de morphologie mathématique à partir du modèle local. Le résultat obtenu est relativement satisfaisant puisque la zone urbaine est correctement détectée.

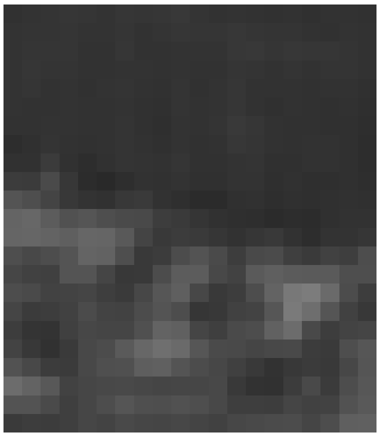
Le résultat obtenu par l'utilisation du noyau avancé est bon (figure 11). Les variances conditionnelles sont calculées dans une fenêtre de taille 11×11. Ce noyau détecte de manière très satisfaisante la zone urbaine de Toreilles : les contours trouvés de la ville sont, en effet, très précis comme nous le montre la figure 11. Les réseaux linéiques ne sont pas détectés. Le défaut que l'on peut cependant attribué à ce noyau est la détection erronée d'un champ ayant une répétition spatiale directionnelle (le vignoble en haut à droite de l'image). Ce défaut existait déjà sur le modèle primaire. Dans une optique de pure détection de zones urbaines, nous pouvons corriger ce défaut en travaillant sur les combinaisons des variances conditionnelles directionnelles. Nous ne pousserons cependant pas les recherches dans ce domaine. De manière générale, le modèle avancé fournit de meilleurs résultats, notamment grace à une meilleur précision des contours urbains.
Les résultats obtenus peuvent encore être améliorés. En effet, une étude sur l'estimation des deux principaux paramètres intervenant : la constante de majoration des paramètres de Lagrange-Kuhn-Tucker et la variance provenant du noyau gaussien dans le cas du noyau primaire peut être réalisée.

6 Conclusion

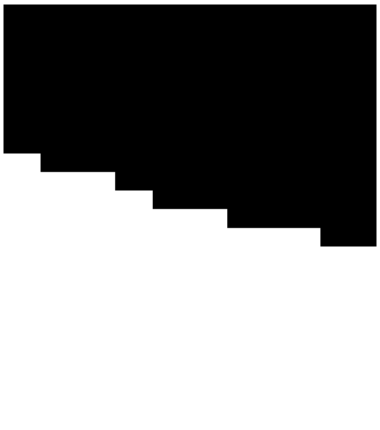
La classification par SVM utilisant les deux noyaux texturaux que nous avons développés, et en particulier le modèle avancé, a fourni de bons résultats sur des problèmes de reconnaissance de structures dans des images satellitaires. En effet, les applications à la détection de feux de forêt et de zones urbaines ont donné des résultats satisfaisants : en définissant une base d'apprentissage de manière pertinente, cette méthode est capable de retrouver des structures texturalement et radiométriquement reconnaissables.

Cette méthode n'est cependant pas automatique puisque la classification est supervisée et fait intervenir dans les noyaux des paramètres régulateurs. Ces résultats sont loin d'être optimaux tant du point de vue du choix de la base d'apprentissage que du règlage des paramètres du noyau : des réglages précis sont nécessaires sur

(a)



(b)



(c)

FIG. 9 – **(a)** Simulation SPOT5 de Toreilles fournie par le CNES. **(b)** et **(c)** base d'apprentissage associée aux résultats.

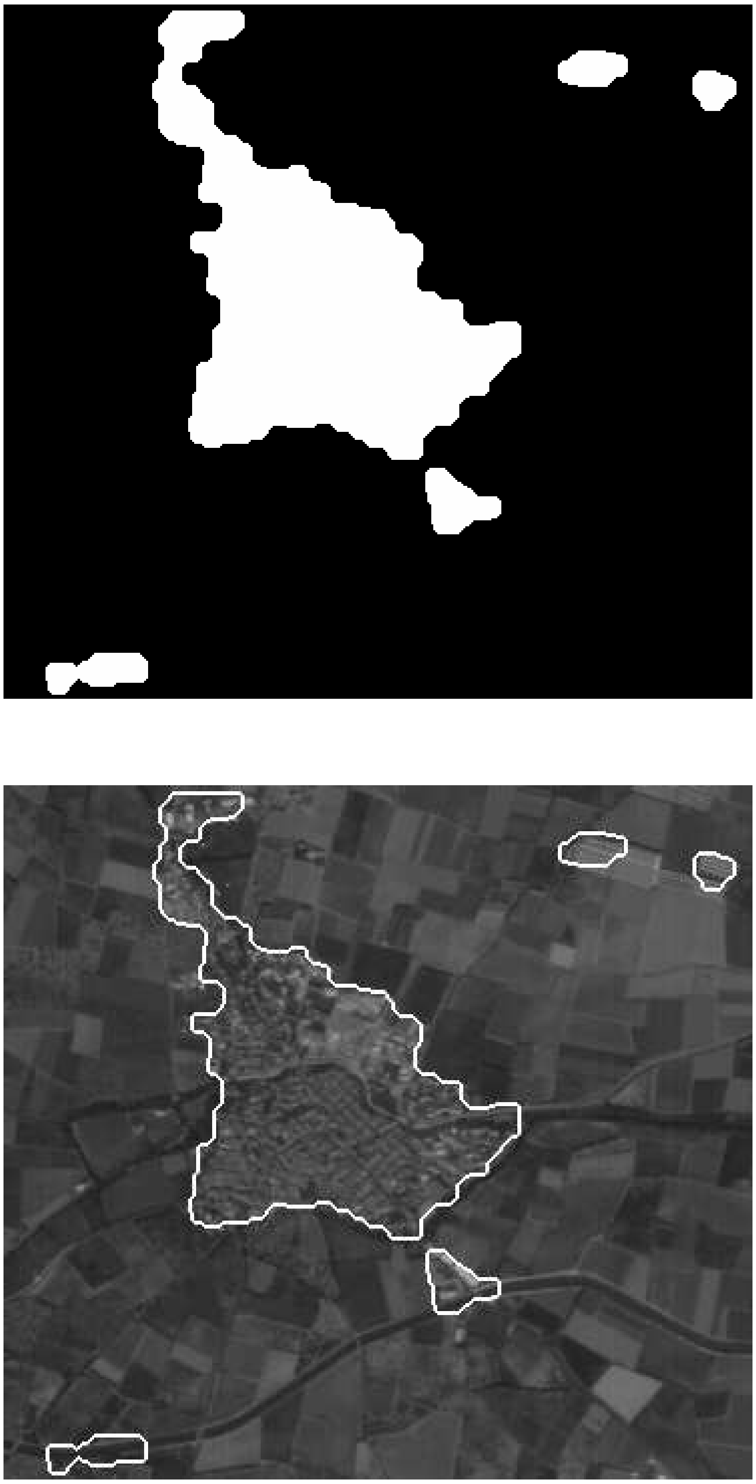


FIG. 10 – **En haut** : Classification obtenue par le modèle primaire (suite à une correction par une ouverture et fermeture morphologiques). **En bas** : résultat juxtaposé sur l'image d'origine

FIG. 11 – **En haut** : Classification obtenue par le modèle avancée (suite à une correction par une ouverture et fermeture morphologiques). **En bas** : résultat juxtaposé sur l'image d'origine

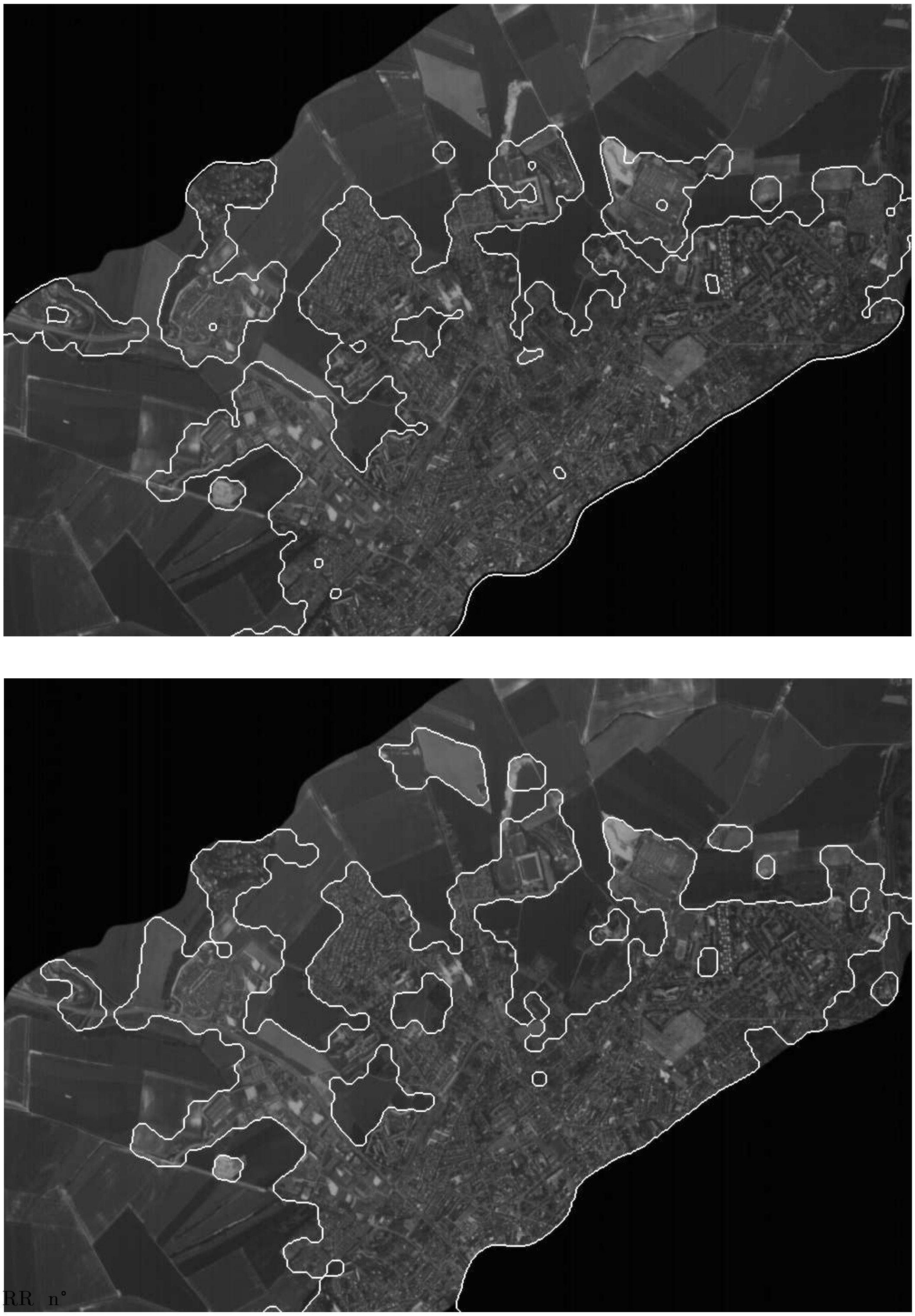


FIG. 12 – **En haut** : Résultat obtenu avec le modèle primaire sur une simulation SPOT5 d'Eppeville fournie par le CNES. **En bas** : Résultat obtenu avec le modèle avancé

une famille donnée d'images. De plus, une version multispectale de cette méthode pourrait apporter de meilleurs résultats.

Références

[Besag, 1986] J. Besag. On the statistical analysis of dirty pictures. *J. Roy. Statist. Soc.*, 48 :259–302, 1986.

[Descombes *et al.*, 1999] X. Descombes, M. Sigelle et F. Prêteux. Estimating Gaussian Markov random field parameters in a non-stationary framework : Application to remote sensing imaging. *IEEE Trans. on Image Processing*, 1999.

[Descombes, 1993] X. Descombes. *Champs markoviens en analyse d'images*. Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1993.

[Gagalowicz, 1983] A. Gagalowicz. *Vers un modèle de texture*. Thèse d'état, Université Pierre et Marie Curie, ParisVI, 1983.

[Julesz, 1962] B. Julesz. Visuel pattern recognition. *IEEE Trans. on Information Theory*, 8, 1962.

[Li, 1995] S. Li. *Markov Random Field modeling in computer vision*. Springer-Verlag, 1995.

[Lorette, 1999] A. Lorette. *Analyse de texture par méthodes markoviennes et par morphologie mathématique : application à l'analyse des zones urbaines sur des images satellitales*. Thèse de Doctorat, Université de Nice-Sophia Antipolis, 1999.

[Pontil et Verri, 1998] M. Pontil et A. Verri. Properties of support vector machines. Rapport de recherche, MIT, 1998.

[Rosenblatt, 1962] F. Rosenblatt. *Principles of Neurodynamics*. Spartan, 1962.

[Samuelides, 2004] M. Samuelides. *Réseaux neuronaux et Apprentissage*. cours SUPAERO, 2004.

[Shawe-Taylor et Critianini, 2002] J. Shawe-Taylor et N. Critianini. *An Introduction to Support Vector Machines*. The Press Syndicate of the University of Cambridge, 2002.

[Soille, 2003] P. Soille. *Morphological Image Analysis.* Springer-Verlag, Heidelberg, 2nd edition, 2003.

[Vapnik, 1996] V. Vapnik. *The nature of statistical learning theory.* Springer Verlag, 1996.

[Vert, 2003] J.P. Vert. *Noyaux définis positifs.* cours Ecole des Mines de Paris, 2003.

[Winkler, 1995] G. Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods a mathematical introduction.* Springer, 1995.

Remerciements

Les auteurs remercient Alcatel Space Cannes pour le soutien financier partiel de ce travail de recherche, ainsi que le CNES et SPOTIMAGE pour la fourniture des données SPOT5 via le programme ISIS.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399