

**FILTRAGE DES IMAGES RADAR
PAR DÉTECTION DES RÉGIONS HOMOGÈNES**

Jean-Marie Beaulieu

Département d'informatique

Université Laval

Pavillon Pouliot, Québec (Québec), Canada, G1K-7P4

Tél.: (418) 656-2564, Internet: beaulieu@ift.ulaval.ca

RÉSUMÉ

Nous présentons une nouvelle technique de filtrage des images radar basée sur la détection des régions homogènes. Le filtrage consiste généralement à calculer la valeur moyenne sur une fenêtre de $N \times N$ pixels. Cet estimé est valide seulement si nous sommes dans une région homogène. Le filtrage doit préserver les éléments structuraux de l'image : les lignes et frontières entre régions. Dans une fenêtre de $N \times N$, il ne faut retenir que les pixels appartenant à une même région homogène. Nous utilisons les caractéristiques des images radar pour définir ce que nous entendons par régions homogènes. La variance du signal radar σ^2 a une valeur élevée et dépendante de la valeur moyenne μ . La connaissance de la distribution du signal nous permet de définir un intervalle (min, max) ayant une probabilité α de contenir les valeurs produites par une région homogène. L'intervalle nous permet donc de sélectionner dans une fenêtre les pixels qui proviennent d'une même région. Nous ajoutons de plus une contrainte spatiale : les pixels utilisés dans le calcul de la moyenne doivent être connectés au pixel central de la fenêtre, c'est-à-dire doivent être reliés par une suite continue de pixels inclus dans l'intervalle. Ceci permet de détecter les bris dans la structure de l'image et de limiter en conséquence la portée du filtrage. Nous utilisons un algorithme de segmentation par seuillage pour déterminer les segments dans une fenêtre (les pixels connectés). Nous devons également répéter le calcul pour toutes les valeurs d'intervalles possibles pour tenir compte des différentes valeurs de la moyenne μ . Pour un pixel donné, nous retenons le résultat de l'itération où la moyenne est calculée sur le plus grand nombre de pixels (plus grande région connectée). Des résultats très intéressants ont été obtenus avec des images agricoles et forestières.

INTRODUCTION

Dans une image radar, la valeur de chacun des pixels est associée à l'amplitude du signal rétrodiffusé du terrain ou des cibles. Chaque cible est donc caractérisée par son coefficient de rétrodiffusion, σ^0 . La formation des images radar est cependant totalement différente de celles obtenues en télédétection passive. Ainsi, le coefficient de rétrodiffusion est relié à la valeur moyenne du signal mais celui-ci peut subir de grandes variations locales et aléatoires dues à la cohérence de l'onde émise et au système imageur. Cette variation se manifeste même pour des cibles homogènes et produit une granulation appelée "speckle" qui réduit la résolution radiométrique de l'image. Les techniques de traitement des images radar doivent donc tenir compte de cet aspect statistique. Les techniques usuelles d'interprétation numérique sont inefficaces ici. Il est donc indispensable de développer de nouvelles méthodes adaptées à ce type d'image.

Modèle multiplicatif des images radar

La nature cohérente du signal radar demande que l'on considère la phase du signal de retour. Pour une région homogène avec un σ^0 constant, les composantes réelles et imaginaires du signal peuvent être représentées par une distribution gaussienne circulaire. L'intensité (l'amplitude au carré) suivra alors une distribution exponentielle. Dans une distribution exponentielle, la moyenne μ et l'écart type σ sont égaux, $\mu = \sigma = \lambda$. Le paramètre λ agit donc comme un facteur d'échelle. Le signal radar suit donc un modèle de bruit multiplicatif, soit :

$$I(t) = R(t) u(t)$$

où $t = (x, y)$ représente les coordonnées spatiales de l'image et $I(t)$ est l'intensité du signal observée au point t . $R(t)$ désigne la réflectivité du terrain au point t et correspond à l'information que l'on veut mesurer. $u(t)$ correspond au bruit ou à l'élément perturbateur du signal. La composante de bruit, $u(t)$, est donc ici multipliée au signal au lieu d'être additionnée. Elle suit une distribution exponentielle avec $\lambda = 1$.

Avec un écart type égal à la moyenne, on voit que la variation du signal radar est très importante. Une manière de réduire la variance est de prendre plusieurs vues du même terrain (image) et de calculer la valeur moyenne pour chaque position dans l'image. Dans l'étape d'acquisition des données, un effet similaire peut être obtenu par un traitement approprié du signal. L'image produite est alors caractérisée par son nombre de vues (look). Pour une région homogène avec un σ^0 constant, l'image de L vues conservera la même valeur moyenne. Par contre, la variance sera divisée par L et l'écart type divisé par \sqrt{L} . L'image d'intensité L vues (la composante $u(t)$) suit une distribution Gamma. Le coefficient de variation est défini comme le rapport de l'écart type sur la moyenne, $C_1 = \sigma_1 / \mu_1$, et est égal à $1 / \sqrt{L}$.

Filtrage adaptatif

Une autre approche pour réduire la variance dans une image radar est de filtrer l'image. Un filtre simple, appelé le "box filter", consiste à calculer pour chaque pixel la valeur moyenne sur une fenêtre de $N \times N$ centrée sur le pixel. Dans une région homogène, on réduit ainsi l'écart type par N . Selon une approche probabiliste, on peut calculer la distribution de la valeur moyenne ainsi obtenue. Pour cela, il faut connaître la distribution de chacun des pixels utilisés dans le calcul de la moyenne. Généralement, on assume que les pixels proviennent d'une même population, i.e., que la distribution est la même pour tous les pixels. La fenêtre de $N \times N$ est donc entièrement incluse dans une région homogène. La valeur moyenne sur la fenêtre correspond alors au résultat recherché, soit le meilleur estimé de la vraie valeur de réflectivité du terrain, $R(t)$. Par contre, si la fenêtre est à cheval sur deux régions distinctes, les pixels vont appartenir à deux populations différentes. La valeur moyenne sera alors un estimé de la réflectivité moyenne des deux régions. Nous avons une perte de résolution spatiale. Les contours des régions sont rendus flous. Il faut éviter ce lissage des éléments structuraux de l'image qui peut atténuer ou faire disparaître les lignes ou les très petites régions (cibles isolées) et qui peut rendre les contours flous.

Des filtres adaptatifs ont été développés pour réduire la granularité de l'image tout en préservant sa structure (arêtes, lignes et cibles isolées) [Lee 1986, 1994][Lopes 1990, 1993]. Le filtre adaptatif se présente comme une somme pondérée de la valeur initiale et de la valeur moyenne sur la fenêtre :

$$I_{\text{estimé}}(t) = \beta I(t) + (1 - \beta) \text{Moyenne}_{N \times N}(t)$$

β vaut zéro pour une région homogène et augmente à mesure que la région devient hétérogène. On se sert du coefficient de variation calculé sur la fenêtre pour déterminer le degré d'homogénéité de la fenêtre. En effet, pour une région homogène, le coefficient de variation est déterminé par le nombre de vues de l'image, $C_I = 1 / \sqrt{L}$. Par contre, si la fenêtre contient des pixels venant de plusieurs régions, le coefficient va croître selon la différence de réflectivité entre les régions. Le coefficient de variation calculé sur une fenêtre, $C_{I, N \times N}(t)$, peut donc servir à évaluer l'homogénéité de la fenêtre. Cependant, $C_{I, N \times N}$ est seulement un estimé et la dispersion de l'estimé peut être très grande. La grande variance du signal se répercute dans le calcul de l'estimé du coefficient de variation.

Le filtrage d'image radar devrait nous permettre de mieux voir la structure de l'image. Cependant, le filtrage adaptatif a besoin de voir cette structure pour savoir si on est dans une région homogène ou non. On peut donc se demander si on doit commencer par filtrer ou par détecter la structure (arêtes, lignes ou cibles isolées)[Touzi 1988] [Lopes 1993]. Quelle que soit la première étape, elle sera difficile à cause de la grande variance du signal.

Nous pouvons donc voir deux limitations au filtrage adaptatif. L'estimation de l'homogénéité (ou du coefficient de variation) est très bruitée et n'est pas robuste. De plus, le filtrage prend toujours en compte la valeur moyenne calculée sur l'ensemble de la fenêtre. On ne cherche pas à éliminer une partie des pixels pour ne retenir que ceux appartenant à la même région que le pixel central.

Le filtre Sigma

Le filtre Sigma, proposé par Jong-Sen Lee [Lee 1986, 1994], vise justement à n'utiliser que les pixels les plus appropriés dans le calcul de la moyenne d'une fenêtre. On utilise le fait que pour une distribution gaussienne unidimensionnelle, 95.5% des points seront compris dans un intervalle de $\pm 2 \sigma$ autour de la moyenne. Considérant que le bruit est multiplicatif dans une image radar, nous définissons l'intervalle de 2σ autour de z comme suit : $((1 - 2 \sigma_u) z, (1 + 2 \sigma_u) z)$. Pour chaque pixel t de l'image, nous devons connaître sa moyenne a priori, $z(t)$, et définir un intervalle autour de cette moyenne. Nous calculons ensuite la valeur moyenne des pixels de la fenêtre qui appartiennent à l'intervalle. Ceci suppose qu'une partie des pixels de la fenêtre, dont le pixel central, appartient à une région homogène de moyenne z . La majorité des pixels de cette région vont être dans l'intervalle et donc contribuer au calcul de la moyenne. Par contre, la majorité des pixels des autres régions ne seront pas dans l'intervalle si les régions sont suffisamment distinctes. Nous pouvons utiliser l'algorithme de manière itérative. Nous posons initialement $z(t) = I(t)$, la valeur initiale du pixel t . Par la suite, le résultat de l'itération k détermine la valeur $z(t)$ de l'itération suivante.

Filtrage par détection des régions homogènes

Dans cet article, nous présentons une nouvelle approche pour le filtrage de l'image. Elle s'inspire du filtre Sigma et exploite l'information spatiale. Comme dans le filtre Sigma, nous définissons un intervalle (min, max) dont l'étendu varie en fonction de l'intensité du signal. Nous n'avons cependant pas besoin de définir une valeur $z(t)$ pour chaque pixel. Par contre, nous devons définir une séquence d'intervalle pour toutes les valeurs d'intensité de l'image. Si les valeurs de l'image sont codées sur 8 octets, nous avons un maximum de 256 intervalles distincts. Pour chaque pixel, nous allons considérer tous les intervalles de la séquence et retenir l'intervalle qui est le plus approprié pour la fenêtre centrée sur ce pixel. L'intervalle approprié devra inclure le plus grand nombre possible de pixels de la fenêtre connectés au pixel central et nécessairement le pixel central. D'autre part, dans le calcul de la moyenne, nous ne considérons que les pixels connectés au pixel central de la fenêtre.

Considérons par exemple un champ (région) homogène de moyenne μ et définissons un intervalle (min, max) tel que $\text{Prob}(I(t) \in (\text{min}, \text{max})) = 80\%$. Si nous avons une fenêtre de $N \times N$ incluse entièrement dans ce champ, alors 80% des pixels seront à l'intérieur de l'intervalle. Ces 80 % des pixels vont généralement être connectés. En effet, les 20% restants vont être distribués aléatoirement dans la fenêtre et formeront de petits segments (régions) qui ne pourront diviser la fenêtre en parties isolées. Nous utilisons ici le terme segment pour désigner les régions formées par la connexion des pixels et le terme champ pour désigner les régions réellement présentes dans le terrain. D'autre part, si une partie de la fenêtre contient des pixels provenant d'un champ avec une valeur moyenne différente, alors ces pixels seront généralement en dehors de l'intervalle. Si la différence entre les deux moyennes est petite, il y aura des pixels du deuxième champ à l'intérieur de l'intervalle, cependant ils formeront généralement de petits segments isolés et non reliés aux pixels du premier champ. Par contre, si le deuxième champ est en fait une ligne qui traverse la fenêtre, alors la fenêtre sera divisée en deux segments et la moyenne ne sera calculée que sur un des deux segments, soit un seul côté de la ligne.

Nous utilisons un algorithme de segmentation par seuillage pour déterminer les segments d'une fenêtre (les pixels connectés). Le filtrage consiste donc à calculer la moyenne des pixels à l'intérieur de la fenêtre et connectés au

pixel central. Cet aspect de connexion spatiale permet de détecter les bris dans la structure de l'image et de limiter en conséquence la portée du filtrage. Nous devons également répéter le calcul pour toutes les valeurs d'intervalle possibles pour tenir compte des différentes valeurs moyennes. Pour un pixel donné, nous retenons le résultat de l'itération où la moyenne est calculée sur le plus grand nombre de pixels (plus grande région connectée).

Des résultats très intéressants ont été obtenus avec des images agricoles et forestières. Nous présentons dans cet article les résultats obtenus avec une image forestière.

FILTRAGE PAR DÉTECTION DES RÉGIONS HOMOGÈNES

Nous présentons maintenant l'approche de filtrage par détection de régions homogènes. Nous expliquons d'abord l'utilisation d'un intervalle autour de la valeur moyenne pour définir une région homogène. Nous présentons ensuite l'algorithme de filtrage.

Détection des régions homogènes

L'algorithme de filtrage peut être utilisé avec des images d'intensité ou avec des images d'amplitude. Dans les deux cas, le modèle multiplicatif du bruit s'applique, soit :

$$f(x, y) = r(x, y) \times u(x, y)$$

où $f(x, y)$ désigne l'image radar à traiter. x et y sont les coordonnées d'un pixel appartenant au domaine de l'image $I = \{ (x, y) \mid x \in 1 \dots N_{\text{pixel}} \text{ et } y \in 1 \dots N_{\text{ligne}} \}$, N_{pixel} et N_{ligne} étant le nombre de colonnes et le nombre de lignes de l'image. Dans une image d'intensité, $r(x, y)$ correspond à la réflectivité du terrain, tandis que pour une image d'amplitude, $r(x, y)$ correspond à la racine carrée de la réflectivité. $u(x, y)$ sert à modéliser le comportement aléatoire du signal. Il est normalisé, c'est-à-dire qu'il a une valeur moyenne de 1. Donc, la valeur moyenne de $f(x, y)$ est égale à $r(x, y)$, $E(f(x, y)) = r(x, y)$. $r(x, y)$ est donc la valeur que l'on cherche à découvrir malgré la présence de l'élément perturbateur que représente $u(x, y)$. La distribution de $u(x, y)$ varie selon que l'on utilise une image d'intensité ou d'amplitude et aussi selon de nombre de vues. Dans tous les cas, elle aura une moyenne de 1, $\mu_u = 1$, et une variance σ_u^2 importante.

La connaissance de la distribution de $u(x, y)$ nous permet de définir un intervalle avec un coefficient de confiance désiré α [Lopes 1990, 1993]. Il faut donc déterminer l'intervalle (min, max) telle que $\text{Prob}_u(u \in (\min, \max)) = \alpha$. α correspond à la probabilité que la valeur de la variable aléatoire u soit comprise dans l'intervalle. Par exemple, pour une image d'intensité à 6 vues, nous avons $\sigma_u = 1/\sqrt{6} = 0.408$. Si nous utilisons une approximation gaussienne pour $u(x, y)$ et si nous désirons obtenir un coefficient de confiance de 68.8%, $\alpha = 0.688$, nous devons utiliser un intervalle de $\pm \sigma$, soit (0.592, 1.408).

Il faut multiplier les valeurs de l'intervalle ci-haut définies par $r(x, y)$ pour obtenir un intervalle exprimé en terme des valeurs de l'image radar $f(x, y)$. Par exemple, si dans une région homogène les valeurs de $r(x, y)$ égalent 100, alors l'intervalle précédant devient (59, 140). Donc, si nous prenons une fenêtre de $N \times N$ incluse dans cette région, 68.8% des pixels auront des valeurs comprises entre 59 et 140. Ces pixels seront généralement connectés. En effet, seulement 15.1% des pixels seront inférieurs à 59 et 15.1% seront supérieurs à 140. Ceux-ci seront distribués aléatoirement dans la fenêtre et formeront de petits segments qui ne pourront diviser la fenêtre en partie isolée. Par contre, si la même fenêtre recouvre une région ayant une valeur $r(x, y)$ de 50, alors seulement 1/3 de ses pixels auront une valeur comprise dans l'intervalle (59, 140). De plus, les segments formés par ces pixels ont peu de chance d'être connectés à l'autre région. En effet, les 2/3 des pixels sont inférieurs à 59 et formeront des segments plus gros qui entoureront et isoleront les segments des pixels supérieurs à 59. Le calcul de la moyenne pour le pixel (x, y)

prend en compte seulement les pixels qui lui sont connectés, i.e., qui appartiennent au même segment. Ainsi, le calcul de la moyenne se fait seulement sur les pixels inclus dans la fenêtre et provenant de la bonne région. Cet aspect de connexion spatiale permet de détecter les bris dans la structure de l'image (transition entre régions homogènes) et de limiter en conséquence la portée du filtrage.

Présentation de l'algorithme de filtrage

Les principales variables utilisées par l'algorithme sont :

$\{ (\min_i, \max_i) \mid i = 1 \dots N_{\text{intervalle}} \}$ la liste des intervalles,
 $f(x, y)$ l'image radar (l'image d'entrée),
 $g(x, y)$ l'image filtrée (l'image de sortie) obtenue par calcul de la valeur moyenne,
 $sz(x, y)$ le nombre de pixels utilisés dans le calcul de la moyenne $g(x, y)$,
 N la taille de la fenêtre pour le calcul de la moyenne,

où (x, y) doit appartenir au domaine de l'image $I = \{ (x, y) \mid x \in 1 \dots N_{\text{pixel}} \text{ et } y \in 1 \dots N_{\text{ligne}} \}$, N_{pixel} et N_{ligne} étant le nombre de colonnes et de lignes de l'image.

L'algorithme peut être décrit comme suit :

```

Définir la liste des intervalles  $\{ (\min_i, \max_i) \mid i = 1 \dots N_{\text{intervalle}} \}$ .
Lire l'image radar dans  $f(x, y)$ .
Initialiser  $g$  :  $g(x, y) = 0$  pour tout  $(x, y) \in I$ .
Initialiser  $sz$  :  $sz(x, y) = 0$  pour tout  $(x, y) \in I$ .

POUR  $i = 1 \dots N_{\text{intervalle}}$                                 {pour tous les intervalles}
FAIRE
    Utiliser l'intervalle  $(\min_i, \max_i)$  pour segmenter l'image.

    POUR  $(x, y) \in I$                                         {pour tous les pixels de l'image}
    FAIRE
        SI  $f(x, y) \in (\min_i, \max_i)$  ALORS
             $m =$  valeur moyenne calculée sur une fenêtre de  $N \times N$  centrée sur
                 $(x, y)$  et considérant seulement les pixels appartenant au
                même segment que  $(x, y)$ .
             $nb =$  nombre de pixels utilisés dans le calcul de  $m$ .
            SI  $nb > sz(x, y)$  ALORS
                 $g(x, y) = m$ 
                 $sz(x, y) = nb$ 
        FIN_SI
    FIN_POUR
FIN_POUR
  
```

Nous examinons plus en détail certaines parties de l'algorithme, soit la construction de la liste des intervalles, l'utilisation d'un intervalle pour segmenter l'image et le calcul de la valeur moyenne sur une fenêtre.

Sélection et génération des intervalles

Pour toute distribution de $u()$, nous pouvons déterminer un intervalle (min, max) nous donnant un coefficient de confiance α désiré. Nous pouvons exprimer cet intervalle en terme de rapport de l'étendue de l'intervalle sur la valeur moyenne :

$$\text{étendue relative} = \text{Erreur!}$$

L'utilisation de l'étendue relative nous permet de générer facilement la liste des intervalles à utiliser. De plus, l'algorithme utilise seulement les valeurs min et max de la liste sans avoir besoin de savoir quelle est la valeur moyenne correspondante. Ainsi, nous pouvons utiliser l'intervalle (59, 140) présenté plus haut sans avoir besoin de savoir qu'il a été calculé à partir d'une valeur moyenne de 100. Donc, la forme de la distribution de $u()$ nous importe peu, tout ce que nous avons besoin est l'étendue relative.

Nous devons également considérer les différentes valeurs que peuvent prendre les pixels de l'image $f(x, y)$. Si l'image comporte 8 bits par pixel, alors les pixels peuvent prendre les valeurs de 0 à 255. Nous aurons alors un maximum de 256 intervalles distincts. Pour $i \in 0 \dots 255$, le $i^{\text{ème}}$ intervalle est calculé comme suit :

$$\begin{aligned} \min_i &= i \times (1 - \frac{1}{2} \text{étendue relative}) \\ \max_i &= i \times (1 + \frac{1}{2} \text{étendue relative}) \end{aligned}$$

Pour réduire le temps de calcul, il serait préférable de réduire le nombre d'intervalles utilisés. On constate que pour des valeurs de i élevées, il y a un important recouvrement entre les intervalles i et $i+1$. Donc, on peut utiliser un pas d'incrément de i qui augmente avec i .

Segmentation de l'image

Pour un intervalle donné (min, max), nous devons identifier les segments de l'image. Nous assignons d'abord une classe à chaque pixel selon sa valeur, $f(x, y)$:

```
POUR (x, y) ∈ I           {pour tous les pixels de l'image}
FAIRE
    classe(x, y) = inférieure   SI min > f(x, y)
    classe(x, y) = inclus      SI min ≤ f(x, y) ≤ max
    classe(x, y) = supérieure  SI f(x, y) > max
FIN_POUR
```

Nous utilisons ensuite un algorithme d'étiquetage des pixels ("pixel labeling", aussi appelé algorithme de segmentation) pour assigner un même étiquette (numéro de segment) à tous les pixels connectés de la même classe. Donc, chaque pixel (x, y) recevra un numéro de segment, $seg(x, y)$. Tous les pixels ayant le même numéro appartiendront à la même classe et de plus seront connectés entre eux, i.e., formeront un segment.

Calcul de la valeur moyenne

A chaque itération de l'algorithme, i.e., pour chaque valeur d'intervalle, nous devons examiner chaque pixel de l'image $((x, y) ∈ I)$ et au besoin calculer la valeur moyenne sur une fenêtre $N \times N$. Nous calculons la moyenne seulement si la classe du pixel est égale à *inclus*, $classe(x, y) = inclus$. Nous parcourrons alors chacun des pixels d'une fenêtre $N \times N$ centré sur (x, y) et utilisons ce pixel dans le calcul de la moyenne seulement si son numéro de segment est le même que celui du pixel (x, y) .

```
m = 0
nb = 0
POUR xx = x - (N-1)/2 ... x + N/2
POUR yy = y - (N-1)/2 ... y + N/2
FAIRE
    SI seg( xx, yy ) = seg( x, y ) ALORS
        m = m + f( xx, yy )
        nb = nb + 1
    FIN_SI
FIN_POUR
m = m / nb           {calcul de la moyenne}
```

À chaque itération, nous retenons seulement la valeur moyenne qui a été calculée sur le plus grand nombre de pixels. Ceci indique que l'intervalle correspondant (min_i, max_i) est le plus approprié pour ce pixel. Cet intervalle devrait en effet produire le segment le plus grand englobant ce pixel. Plus précisément, il produit le segment donc la partie incluse dans la fenêtre est la plus grande.

FILTRAGE D'UNE IMAGE SAR

L'algorithme de filtrage a été expérimenté sur des images agricoles et forestières. Nous présentons ici les résultats pour une image aéroportée C/X SAR 580 de la Forêt Montmorency au nord de la ville de Québec. C'est une image 7 vues, de 802 x 701 pixels, de 6 mètres de résolution et d'un pas de pixel de 4 mètres. La Figure 1 montre une partie de 800 x 400 pixels de cette image. L'image contient des cours d'eau, un lac, des routes et des zones de coupes forestières. Nous utilisons l'étendue relative pour définir les valeurs des intervalles utilisés dans l'algorithme. Nous avons utilisé des étendues relatives de 0.3, de 0.4 et de 0.5. Le filtrage est peu prononcé à 0.3 et devient très visible à 0.5. La Figure 2 montre le résultat du filtrage avec une étendue relative de 0.3 pour la zone de 800 x 400 pixels. La Figure 3 présente le résultat pour une valeur de 0.4. Dans la Figure 2, on voit que le filtre préserve toute la structure de l'image tout en réduisant l'effet granulaire (de chatoiement, de "speckle"). En effet, les routes, les contours de cours d'eau et des coupes forestières sont bien préservés. Dans la Figure 3, on voit que les surfaces sont nettement plus lisses. Encore ici, les routes et les contours sont très bien préservés. Les cibles isolées de petites tailles mais bien contrastées sont très bien préservées. Ceci montre que le filtre réussit très bien à lisser les régions homogènes sans rendre les contours flous. Nous avons utilisé 131 intervalles lors du filtrage avec une étendue de 0.3. Le temps de calcul sur une station Sun a été de 13 minutes. Le temps de calcul a été de 16 et 17 minutes pour les autres images filtrées.

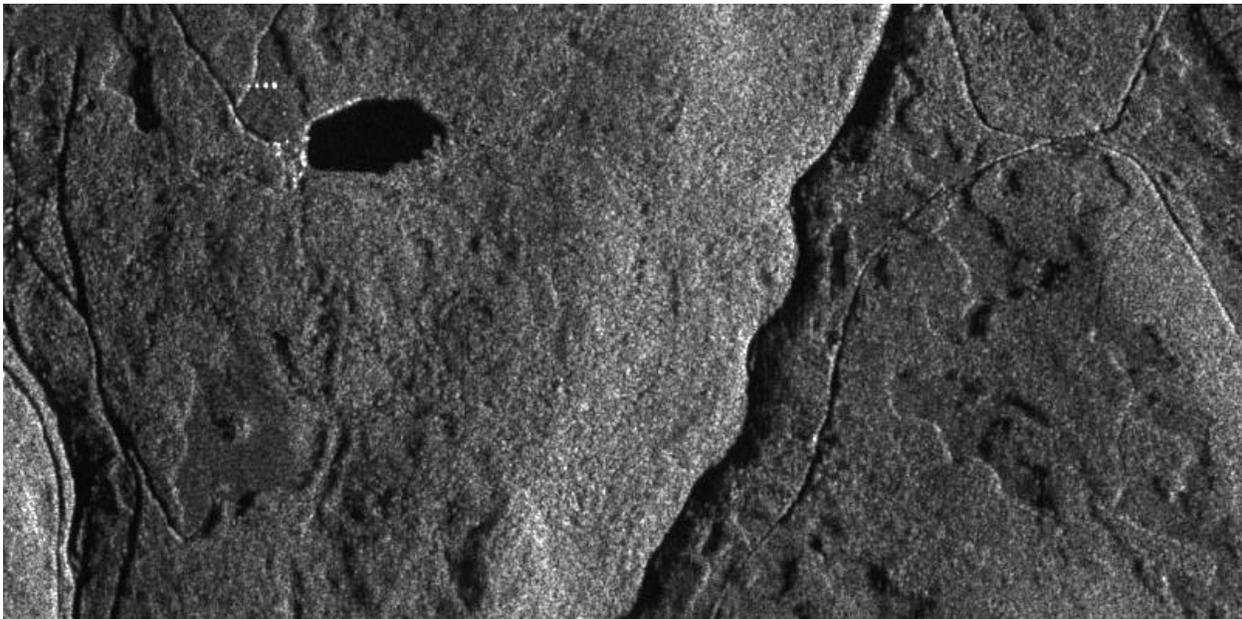


Figure 1 : Image aéroportée C/X SAR 580 de la Forêt Montmorency, 7 vues, 800 x 400 pixels.

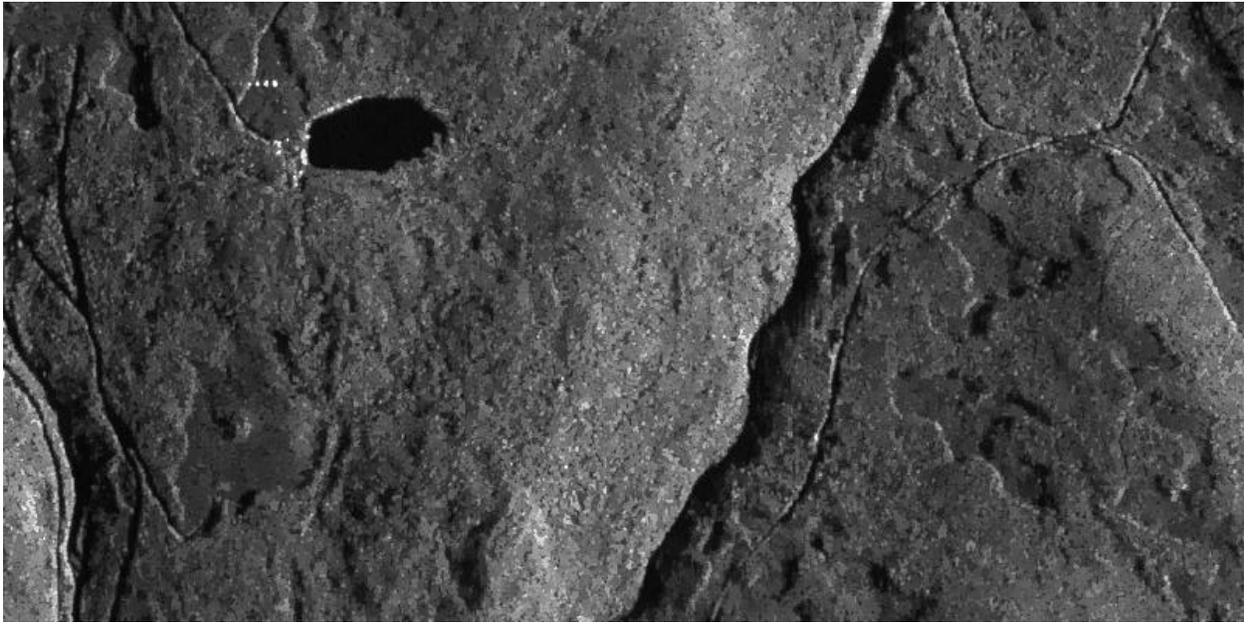


Figure 2 : Image filtrée avec une étendue relative de 0.30.

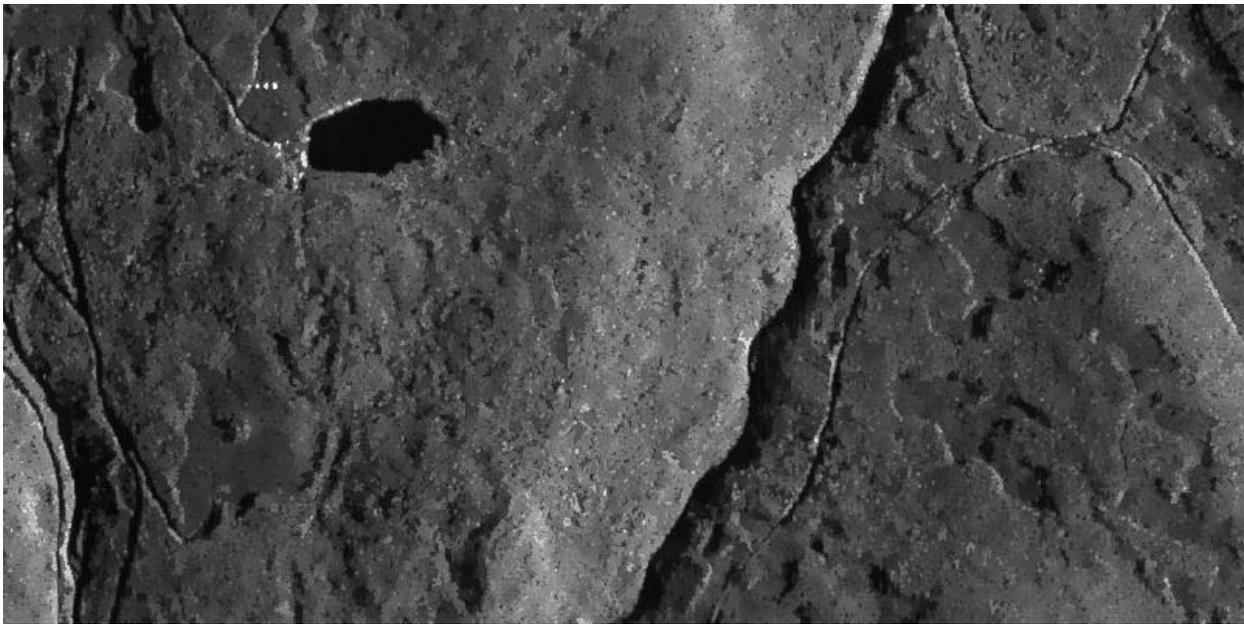


Figure 3 : Image filtrée avec une étendue relative de 0.40.

CONCLUSION

Nous avons présenté une nouvelle approche pour le filtrage des images radar. Elle exploite les caractéristiques du signal radar ainsi que des contraintes spatiales. Elle s'inspire du filtre Sigma où seulement une partie des pixels d'une fenêtre est utilisée pour le calcul de la moyenne. Cependant, elle impose de plus une contrainte de connexité entre pixels qui permet de détecter les bris dans la structure de l'image et de limiter en conséquence la portée du filtrage. Des résultats très intéressants ont été obtenus avec des images agricoles et forestières. Il faut maintenant évaluer la performance de ce filtre et le comparer aux autres filtres utilisés.

REMERCIEMENT

Nous tenons à remercier le Centre Canadien de Télédétection, Ottawa, pour son support et pour avoir fourni l'image utilisée dans cet article.

RÉFÉRENCES

- Lee, Jong-Sen, 1986, "Speckle Supression and Analysis for Synthetic Aperture Radar Images", *Optical Engineering*, 25(5):636-643.
- Lee, Jong-Sen, I. Jurkevich, P. Dewaele, P. Wambacq et A. Oosterlinck, 1994, "Speckle Filtering of Synthetic Aperture Radar Images: A Review", *Remote Sensing Reviews*, 8:313-340.
- Lopes, Armand, Ridha Touzi et E. Nezry, 1990, "Adaptive Speckle Filters and Scene Heterogeneity", *IEEE Trans. on Geoscience and Remote Sensing*, 28(6):992-1000.
- Lopes, Armand, E. Nezry, R. Touzi et H. Laur, 1993, "Structure Detection and Statistical Adaptive Speckle Filtering in SAR Images", *Int. J. Remote Sensing*, 14(9):1735-1758.
- Touzi, Ridha, Armand Lopes et Pierre Bousquet, 1988, "A Statistical and Geometrical Edge Detector for SAR Images", *IEEE Trans. on Geoscience and Remote Sensing*, 26(6):764-773.