

Aplicación de un Filtro Stack en Imágenes con Ruido Speckle

María E. Buemi, Juliana Gambini, Marta Mejail y Julio Jacobo Berllés

Departamento de Computación, FCEyN, UBA, Buenos Aires, Argentina
{mebuemi; jgambini; marta; jacobob}@dc.uba.ar

Resumen

Las imágenes de radar de apertura sintética (SAR) o las generadas con sistemas de ultrasonido, son muy difíciles de analizar e interpretar debido a que están contaminadas por el ruido speckle. En este trabajo se presenta la aplicación de un filtro adaptativo no lineal, que tiene el objetivo de atenuar el ruido speckle y mejorar la imagen para facilitar la búsqueda de características. Está basado en el entrenamiento con dos imágenes, una que no tiene ruido y otra contaminada con un ruido de distribución similar al de la imagen que se quiere tratar. Con el fin de hallar bordes de objetos se aplica un algoritmo de extracción de contornos basado en curvas B-spline deformables. Se han hecho pruebas con imágenes sintéticas y reales, obteniéndose muy buenos resultados.

1. Introducción

Los *stack filters* son un caso especial de filtros no lineales, por lo tanto tienen buena performance para filtrado de imágenes con distintos tipos de ruido preservando bordes y detalles. Estos filtros se basan en la descomposición por umbrales de la señal de entrada, obteniéndose una señal binaria para cada uno de ellos, y en la aplicación de un filtro a cada una de esas señales por medio de una ventana deslizante. Este filtro se genera por medio de un algoritmo adaptativo, de manera que cumpla la propiedad de *stacking*. El método propuesto en este trabajo está basado en el algoritmo de Yoo *et al.* [6]. La ventaja de utilizar la descomposición por umbrales es que las señales binarias resultantes son fáciles de manipular y de analizar. Por otro lado, el análisis de imágenes con ruido *speckle* es un problema difícil de resolver, tratado en la literatura con diversos métodos estadísticos ([4], [5], [3]). Los métodos clásicos de filtrado de imágenes resultan poco satisfactorios en imágenes con ruido *speckle*, dado que este ruido proviene del sistema de captura y se modela como el producto de dos variables aleatorias independientes, en lugar de ser aditivo. En este trabajo se aplica el filtro diseñado a imágenes con ruido *speckle*, se atenúa el ruido para luego encontrar contornos de regiones utilizando un algoritmo de curvas B-spline deformables [1]. Este artículo está organizado de la siguiente manera, en la sección 2 se introduce un resumen de las herramientas necesarias para el diseño de filtros stack. En la sección 3 se presenta un resumen del algoritmo utilizado para la detección de bordes de objetos de interés. En la sección 4 se muestran los resultados obtenidos, y en la sección 5 se presentan las conclusiones.

2. Filtros no lineales

Un filtro es una función que mapea señales de entrada en señales de salida con características determinadas. El objetivo del filtrado de una señal es atenuar o eliminar el ruido contenido en la misma. Los filtros pueden ser lineales o no lineales al satisfacer o no las condiciones de linealidad:

$$h(ax + by) = ah(x) + bh(y)$$

donde h es la función de filtrado, x, y son las variables de función y a y b son valores constantes. Los filtros no lineales poseen la ventaja de que atenúan el ruido de la imagen, preservando bordes y detalles.

Filtros Stack

Los *filtros stack* son un caso especial de filtros no lineales, por lo que comparten con ellos la propiedad de poseer una buena performance para filtrado de imágenes con distintos tipos de ruido, preservando bordes y detalles. Se basan en la descomposición por umbrales y la propiedad de *stacking*. El proceso comienza con la descomposición por umbrales de la imagen de entrada, obteniéndose una imagen binaria para cada uno de los umbrales utilizados. Luego se aplica una función de filtrado S_f a cada una de esas imágenes por medio de una ventana deslizante, donde S_f es construido a partir de una función f que cumple con la propiedad de *stacking*. Las salidas de los filtros correspondientes a cada umbral se suman para dar el resultado final. La ventaja de este método es que se aplica el filtro a imágenes binarias, además de que el procesamiento de la imagen correspondiente a cada umbral, puede realizarse en paralelo.

El operador de umbralización T^m obtiene una imagen binaria de la siguiente manera:

$T^m : \{0, 1, \dots, M\} \rightarrow \{0, 1\}$ tal que

$$T^m(x) = \begin{cases} 1 & \text{si } x \geq m \\ 0 & \text{si } x < m \end{cases}, \quad (1)$$

con $M, m \in \mathbb{Z}$, donde m es el valor del umbral y $\{0, 1, \dots, M\}$ es el conjunto de niveles de gris.

De acuerdo a esta definición, el valor de la imagen original $x \in \{0, 1, \dots, M\}$ puede ser reconstruida realizando la sumatoria de imágenes binarias obtenidas luego de aplicar el operador T^m , para umbrales m variando entre 1 y M . La Figura 1 muestra un ejemplo de aplicar el operador de umbralización a una señal unidimensional discreta, con valores de umbral $m = 1, 2, 3, 4, 5$.

$$\begin{aligned} X &= [2, 1, 4, 5, 3, 2, 4, 3] \\ X^1 &= [1, 1, 1, 1, 1, 1, 1, 1] \\ X^2 &= [1, 0, 1, 1, 1, 1, 1, 1] \\ X^3 &= [0, 0, 1, 1, 1, 0, 1, 1] \\ X^4 &= [0, 0, 1, 1, 0, 0, 1, 0] \\ X^5 &= [0, 0, 0, 1, 0, 0, 0, 0] \end{aligned}$$

Figura 1: Descomposición por umbrales aplicado a una señal unidimensional.

La función f utilizada para construir el filtro S_f posee la propiedad de *stacking* si se cumple que

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

donde n es la longitud de los vectores de entrada, entonces

$$\forall X, Y \in \{0, 1\}^n, X \leq Y \Rightarrow f(X) \leq f(Y). \quad (2)$$

La Figura 2 muestra un esquema de aplicación de un filtro *stack* a una señal unidimensional. S_f es el filtro que se le aplica a cada una de las señales binarias obtenidas luego de la umbralización.

La función f se genera por medio de un algoritmo adaptativo y alcanza su estado final cuando todas sus componentes están ordenadas, cumpliendo la propiedad de *stacking*. En este trabajo se utiliza el algoritmo adaptativo expuesto en [6].

3. Detección de bordes utilizando representación B-spline para curvas

En esta sección presentamos el algoritmo de detección de bordes utilizado en este trabajo para detectar el contorno del objeto de interés. Con este propósito se utiliza la representación B-spline para curvas, que tiene la

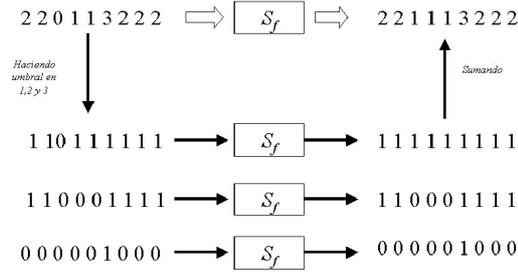


Figura 2: Esquema de aplicación de un filtro *stack* a una señal unidimensional.

ventaja de que la ecuación de la curva queda representada por sus puntos de control, lo cual reduce la cantidad de parámetros y por lo tanto el costo computacional. Además puede elegirse el orden de la representación, de acuerdo a la suavidad necesaria para la curva. Para más detalles sobre este tema consultar [1] y [3].

Las curvas utilizadas en este trabajo para extracción de contornos son cerradas, de orden 3 ó 4, especificadas por funciones B-spline periódicas.

Sea E una escena compuesta por el fondo B y una región R con su borde ∂R . Se desea encontrar la curva C_B que ajusta al borde de la región ∂R en la imagen. Definimos una región inicial de búsqueda, especificada por un polígono, cuyos vértices son los puntos de control que generan una curva B-spline. Si un punto pertenece al borde

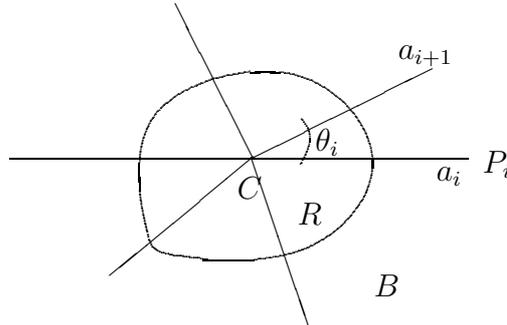


Figura 3: Segmentos Radiales utilizados para la extracción de los puntos de borde.

de la imagen, entonces una muestra tomada sobre un entorno del punto exhibe un cambio en la intensidad de los niveles de gris y es considerado un punto de transición. Entonces se consideran N segmentos $a^{(i)}$, $i \in \{1, \dots, N\}$ con la forma $a^{(i)} = \overline{CP_i}$. Aquí C es el centroide de la región inicial, el extremo P_i es un punto fuera de la región y $\theta = \text{ang}(a^{(i)}, a^{(i+1)}) \forall i$ es el ángulo entre dos segmentos consecutivos como muestra la Figura 3.

Después de aplicar el filtro explicado en la sección 2, el punto de borde sobre cada segmento se encuentra convolucionando los datos de la imagen sobre el segmento con una máscara dada por $[-2, -1, 0, 1, 2]$.

4. Resultados

En esta sección se presentan los resultados obtenidos al aplicar el algoritmo a una imagen sintética con ruido *speckle*. Para generar la función f se entrena el filtro con una imagen con ruido gaussiano, de varianza $\sigma = 20$. Este filtro encontrado se aplica a la imagen con ruido *speckle* de, al menos, 4 looks. Fue generada una imagen

SAR sintética, utilizando la distribución \mathcal{G}_A^0 , desarrollada por Frery *et al.* en [2], con dos áreas definidas, una correspondiente al objeto y otra correspondiente al fondo. Los parámetros utilizados fueron $\alpha_0 = -3$, $\alpha_1 = -10$ para el objeto y fondo respectivamente. La Figura 4(a) muestra la imagen binaria original del objeto que se desea encontrar. La Figura 4 (b) muestra la imagen contaminada con ruido gaussiano, utilizada para entrenar el filtro. La Figura 4 (c) muestra la imagen contaminada con ruido *speckle*. La Figura 5(a) muestra el resultado de aplicar el filtro generado y por último la Figura 5(b) muestra el resultado de aplicar el algoritmo de detección de contornos con B-spline a la imagen de la Figura 5 (a). Se observa que la curva obtenida ajusta muy bien al contorno del objeto.

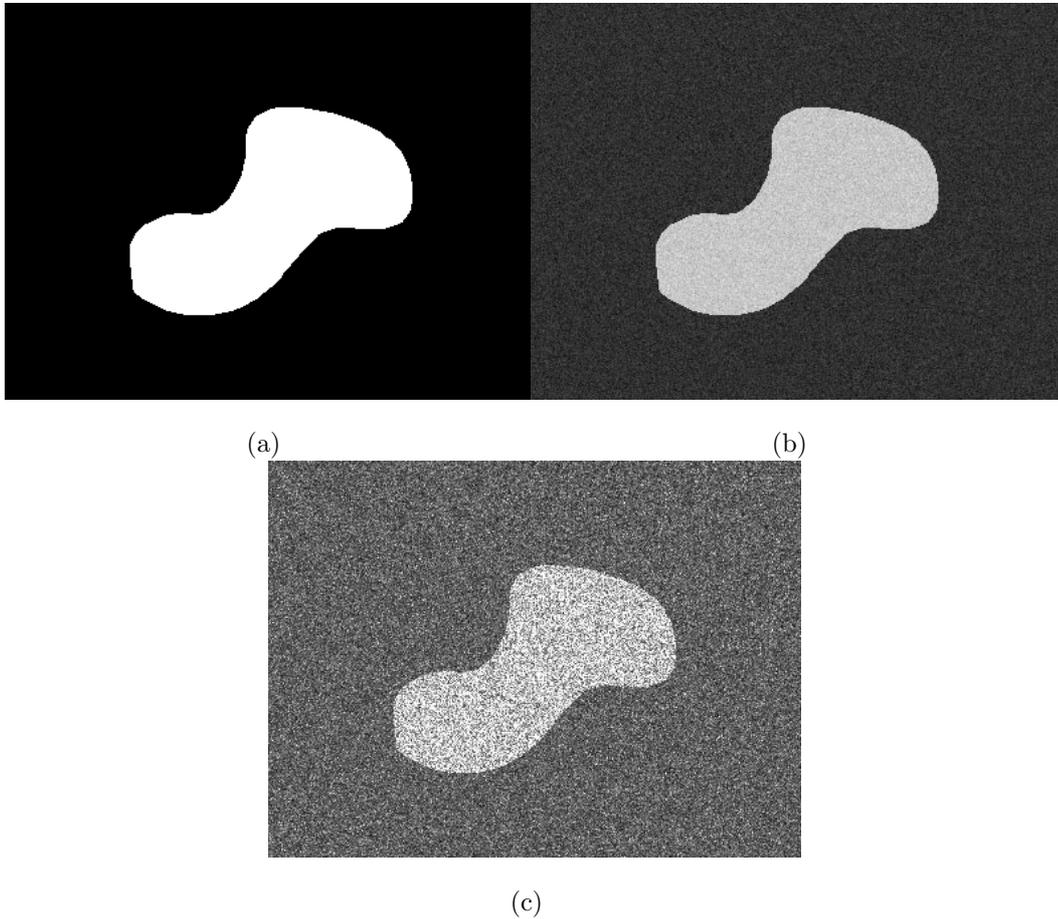
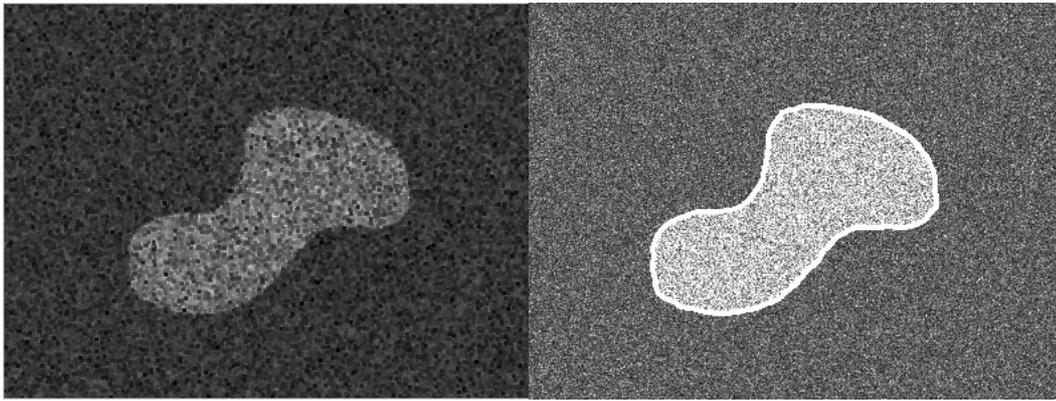


Figura 4: (a)Imagen binaria original. (b)Imagen contaminada con ruido gaussiano utilizada para entrenar el filtro, $\sigma = 20$. (c) Imagen contaminada con ruido *speckle*

5. Conclusiones

Los métodos clásicos de filtrado de imágenes resultan poco satisfactorios en imágenes con ruido *speckle*, dado que este ruido proviene del sistema de captura y se modela como el producto de dos variables aleatorias independientes, en lugar de ser aditivo. En este trabajo se presenta un método para encontrar el contorno de regiones en imágenes con ruido *speckle*. El proceso comienza generando un filtro no lineal adaptativo, entrenado con una imagen binaria original y otra con ruido gaussiano. Luego el filtro construido se aplica a la imagen con ruido *speckle*, utilizando ventanas deslizantes de 3×3 . Para terminar se aplica un algoritmo de contornos B-spline deformables, que tiene la ventaja de obtener una curva determinada por pocos parámetros (puntos de control) y que provee de una fórmula matemática para el contorno del objeto deseado.



(a)

(b)

Figura 5: (a) Imagen speckle de 4 looks filtrada luego de entrenar con ruido gaussiano. (b) Borde del objeto encontrado

Referencias

- [1] A. Blake and M. Isard. *Active Contours*. Springer Verlag, 1998.
- [2] A. C. Frery, H.-J. Müller, C. C. F. Yanasse, and S. J. S. Sant'Anna. A model for extremely heterogeneous clutter. *IEEE Transactions on Geoscience and Remote Sensing*, 35(3):648–659, May 1997.
- [3] M. J. Gambini, M. Mejail, J. C. Jacobo-Berles, H. Muller, and A. C. Frery. Automatic contour detection in sar images. In *5th European Conference on Synthetic Aperture Radar ISBN = 3-8007-2828-1 198*, pages 993–996, 2004.
- [4] O. Germain and P. Réfrégier. Edge location in sar images: Performance of the likelihood ratio filter and accuracy improvement with an active contour approach. *IEEE Transaction on Image Processing*, 10(1), 2001.
- [5] M. E. Mejail, A. C. Frery, J. Jacobo-Berles, and O. H. Bustos. Approximation of distributions for SAR images: Proposal, evaluation and practical consequences. *Latin American Applied Research*, 31:83–92, 2001.
- [6] Jisang Yoo, Kelvin L. Fong, Jr-Jen Huang, Edward J. Coyle, and George B. Adams III. A fast algorithm for designing stack filters. *IEEE Trans.on image processing*, 8(8):772–781, 8 1999.