R versus Python en la selección de umbrales múltiples para imágenes de radares de apertura sintética (SAR)

Andrea Alejandra Rey, Gisela Verónica Caballero, María de los Ángeles Mabel Ferré

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Centro de Procesamiento de Señales e Imágenes, Medrano 951, (C1179AAQ), Ciudad Autónoma de Buenos Aires, Argentina

arey@frba.utn.edu.ar

Recibido el 13 de marzo de 2020, aprobado el 23 de abril de 2020

Resumen

Las imágenes de radares de apertura sintética (imágenes SAR) han sido consideradas por varios investigadores como la mejor herramienta para monitorear la Tierra. A pesar de las ventajas de este tipo de radares, las imágenes SAR son difíciles de analizar. El objetivo de seleccionar umbrales para una imagen es obtener una nueva imagen simplificada, conservando la información de forma y la estructura geométrica. En este trabajo comparamos el desempeño de los lenguajes R y Python para la selección de umbrales múltiples en imágenes SAR reales, en términos de costo computacional y medidas clásicas para el análisis de calidad de imagen.

PALABRAS CLAVE: IMÁGENES SAR - UMBRALES MÚLTIPLES – SEGMENTACIÓN - R - PYTHON

Abstract

Synthetic aperture radar images (SAR images) have been considered as the best tool for Earth monitoring by many researchers. In spite of the advantages of this kind of radars, SAR images are very difficult to analize. The goal of image thresholding is to find a new simplified image that preserves the same shape information and geometric structure. In this work, we compare the performance of the languages R and Python in real SAR image multiple thresholding, in terms of computational cost and classical image quality assessment measures.

KEYWORDS: NUCLEAR REACTIONS - NUCLEAR REACTORS - TRITONS - REVERSE PROTONS - 48V

Introducción

Las imágenes digitales son transformaciones de datos electrónicos provenientes de medios analógicos, con las características de guardado, organización, recuperación y restauración a través de dispositivos tecnológicos. Las mismas pueden ser pensadas como una matriz bidimensional, que consiste en puntos discretos llamados pixeles. En las imágenes a color, cada píxel puede tomar uno de los tres valores: rojo, verde o azul, con cierto nivel de precisión. Por otro lado, las imágenes en escala de grises se componen de pixeles que toman valores entre 0 y $G - 1$, llamados niveles de gris. En este caso, G indica la cantidad de niveles de gris que posee la imagen, siendo 256 su valor máximo. En numerosos problemas de procesamiento de imágenes, resulta más simple y más eficiente tratar con imágenes en escala de grises que con imágenes a color.

En muchas aplicaciones de procesamiento de imágenes, se necesita abstraer objetos o patrones que serán utilizados luego en tareas de alto nivel. Con ese propósito, los pixeles en la imagen se agrupan en regiones distintivas a través de un proceso que se llama segmentación. Una segmentación exitosa puede simplificar la representación de una imagen en un objeto que sea más significativo y más fácil de analizar.

El objetivo de segmentar una imagen de manera efectiva es separar objetos de su fondo y diferenciar pixeles que tengan valores cercanos para mejorar el contraste. En muchas aplicaciones del proceso de segmentación se busca que las regiones tengan características homogéneas, como por ejemplo su nivel de gris, lo que indica que corresponden al mismo objeto. Hay que tener en cuenta que la exactitud de la etapa de segmentación tiene un gran impacto en la eficacia de los pasos subsiguientes en el procesamiento de imágenes, debido a que la confiabilidad de los resultados de salida depende de la calidad de la imagen de entrada dada por el preprocesamiento.

La ventaja de trabajar con imágenes segmentadas es que éstas requieren de un espacio menor de almacenamiento, tienen una velocidad rápida en su procesamiento y son fáciles de manipular, en comparación con una imagen que posee una escala de grises de 256 niveles.

Una de las técnicas más conocidas en segmentación de imágenes en escala de grises, consiste en la selección de umbrales, a la cual se le ha prestado mucha atención a lo largo de las últimas décadas. Con el fin de encontrar los umbrales que mejor segmentan la imagen, es preferible operar con nociones estadísticas antes que con la misma imagen. En particular, se arma el histograma de la imagen contando la cantidad de pixeles de la misma que tienen el mismo nivel de gris. De este modo, se muestra la frecuencia de cada nivel de gris presente en la imagen. Esta técnica tiene aplicaciones en varias áreas tales como procesamiento de documentos (Kamel y Zhao, 1993; Abak et al. 1997), procesamiento de escenas o mapas (Trier y Jain, 1995; Bhanu, 1986), análisis de imágenes satelitales (Peak y Tag, 1994; Soni *et al.* 2013), inspección automática de materiales en control de calidad (Sezgin y Sankur, 2004; Kapur *et al.* 1985) y en procesamiento de imágenes médicas para un diagnóstico más preciso y un tratamiento adecuado (Mancas *et al.* 2005; Al-Attas y El-Zaart, 2007; Azghani *et al.* 2014).

Las técnicas de selección de umbrales pueden dividirse en dos grupos: biniveladas o multiniveladas. En las biniveladas, la imagen se divide en dos regiones definidas por un único umbral, a partir del cual aquellos pixeles que tengan un nivel de gris superior al mismo se clasifican como el objeto y los demás como el fondo. Existen varios métodos propuestos para la binarización de una imagen. Otsu (1979), basado en análisis discriminante, elige un umbral óptimo maximizando la varianza entre clases y resulta ser una de las mejores técnicas para imágenes del mundo real en general, a pesar del alto costo de procesamiento en función del tiempo consumido. Kapur *et al.* (1985) segmentan la imagen maximizando la entropía del histograma de los niveles de gris. Brink (1992) presenta otra alternativa empleando también el concepto de entropía. Tsai (1985) introduce una técnica fundada en la preservación de momentos. Kittler y Illingworth (1986) se apoyan en la minimización de cierto error. Kurita *et al.* (1992) usan la idea de máxima verosimilitud para hallar el umbral. Wang *et al.* (2002) proponen la elección del umbral basado en la maximización del índice de falta de claridad del histograma de la escala de

grises. Sezgin y Sankur (2004) realizan un sondeo de otros métodos para encontrar un único umbral.

En el caso de las técnicas multiniveladas, el proceso consiste en segmentar la imagen de niveles de gris en varias regiones diferentes, lo que implica la determinación de más de un umbral, dividiendo la imagen en ciertas regiones de brillo que corresponden al fondo y a varios objetos. Reddi *et al.* (1984) proponen un algoritmo iterativo a partir del método de Otsu como generalización multinivelada. Así como el método de Otsu, el método de Kapur también puede extenderse fácilmente para la selección de múltiples umbrales. No obstante, éstos son ineficientes para determinar umbrales óptimos debido al crecimiento exponencial en la complejidad computacional del algoritmo. Además, la precisión del algoritmo decrece cuando el número de umbrales crece.

Las técnicas biniveladas son apropiadas para el procesamiento clásico de imágenes como el análisis automático de imágenes de documentos o partes industriales. Empero, se debe adoptar un método multinivelado en el caso de escenarios más complejos, donde los métodos biniveladas fallan a la hora de producir resultados satisfactorios. Lamentablemente, muchas de las técnicas multiniveladas no son capaces de determinar de manera automática la cantidad requerida de umbrales (Whatmough, 1991).

El tema de segmentación de imágenes ha sido objeto de estudio de muchos investigadores por años. Sin embargo, ciertas características propias de una imagen, como las diferentes formas que toman sus histogramas, hacen que el problema siga abierto, por lo que sería necesario continuar con la investigación. En este trabajo aplicamos técnicas de selección de umbrales múltiples para imágenes de radares de apertura sintética, más conocidas por imágenes SAR (*synthetic aperture radar*), consideradas como una herramienta de monitoreo de la Tierra. Este tipo de imágenes ha cobrado gran importancia puesto que permiten monitorear lugares con difícil acceso y posibilitan detectar la acción del hombre sobre el medioambiente como la deforestación, la evolución de humedales, la presencia de derrames de petróleo, entre otras. En líneas generales, un radar de imágenes es un instrumento que mide la respuesta del terreno ante la radiación electromagnética emitida por su antena en dirección a la superficie terrestre. La radiación emitida por el radar está polarizada linealmente, de forma horizontal (H) o vertical (V). La radiación retrodispersada al chocar con la superficie posee componentes en ambas direcciones. Si se trabaja con una sola polarización de emisión y se detecta una sola componente de radiación retrodispersada, tenemos las siguientes posibilidades HH, HV, VH y VV, donde la primera letra indica la polarización de la radiación emitida y la segunda marca la componente de polarización detectada. Dentro de estos radares, el SAR posee una antena pequeña que emite una serie de pulsos consecutivos y recibe una serie de ecos combinándolos de modo que parezca ser una sola observación (simultánea) de una antena grande. De cada punto del terreno se sabrá cómo distorsiona la amplitud y la fase del pulso. El procesado busca combinar la información obtenida en varios barridos de la antena para recrear un solo "barrido virtual".

Entre las ventajas de este sistema de teledetección podemos resaltar que el radar posee un sistema de iluminación propio permitiéndole adquirir imágenes tanto de día como de noche; además, la radiación electromagnética a la frecuencia de operación de los sistemas SAR atraviesa las nubes sin afectar la calidad de la imagen, por lo que las condiciones climáticas no generan limitación en su adquisición. Sin embargo, estas imágenes también poseen algunas desventajas, entre las que podemos señalar su dificultad en el análisis debido a la presencia del ruido *speckle*. La presencia de este ruido hace que en este tipo de imágenes se pueda observar cierto tipo de granulado o falta de contraste. Este ruido, que es inherente al proceso de captura de las imágenes, es un ruido no gaussiano con lo cual difiere del ruido presente en imágenes ópticas. Una de las técnicas utilizadas para la reducción del ruido *speckle* es la generación de varias vistas *(looks)* a partir de un mismo conjunto de pulsos crudos durante el proceso de generación de la imagen. Esta técnica se conoce como proceso *multilook.* Cada vista es una observación independiente del mismo objetivo. Las mismas se promedian pixel a pixel reduciendo el ruido, pero sacrificando resolución espacial.

Este contexto invita al desarrollo y evaluación de

algoritmos de segmentación de imágenes SAR.

El trabajo continúa de la siguiente manera: presentamos las imágenes de estudio y las herramientas tecnológicas con las que seleccionamos umbrales múltiples. Evaluamos y comparamos estas herramientas en función del tiempo de procesamiento computacional, simpleza del código e índices que miden la calidad de la imagen segmentada. Finalmente, exponemos algunas conclusiones basadas en los resultados obtenidos con el fin de realizar una recomendación.

Parte Experimental

Lenguajes de programación

Nuestro objetivo principal es comparar el rendimiento de dos lenguajes de programación aplicados a la selección de umbrales múltiples en el caso de imágenes SAR.

R (*R Core Team*, 2013) es un lenguaje y entorno para cálculos estadísticos y gráficos. Al ser un proyecto GNU, es un proyecto colaborativo de software con el objetivo de crear un sistema operativo completamente libre. Una de sus mayores ventajas es la facilidad con la que pueden producirse trazados diseñados con calidad de publicación, cuidando los valores predeterminados. Compila y corre en una amplia variedad de plataformas UNIX y sistemas similares (incluyendo FreeBSD y Linux), Windows y MacOS. En particular, trabajamos con la biblioteca imagerExtra (Ochi, 2019) que provee funciones avanzadas para el procesamiento de imágenes basado en la biblioteca imager (Barthelme, 2019).

Python (Van Rossum y Drake Jr, 1995) es un lenguaje de programación interpretado, dinámico y que soporta orientación a objetos. Permite un trabajo rápido y la integración de sistemas de manera más efectiva. Es administrado por la *Python Software Foundation* y posee una licencia de código abierto compatible con la licencia general de GNU a partir de su versión 2.1.1. Las bibliotecas que empleamos son *scikit-image* (Van der Walt *et al.* 2014) que es una colección de algoritmos para el procesamiento de imágenes escrito por una comunidad activa de voluntarios; y numpy (Oliphant, 2006) que es el paquete fundamental para cómputos científicos con Python.

Las versiones que utilizamos son: R 3.6.1 y Python 3.7.4.

Imágenes

Frery *et al.* (1997) introdujeron una familia de distribuciones que modelan datos de intensidad de imágenes SAR. Llamaron a esta distribución G_I^0 , la cual queda definida a partir de tres parámetros. El parámetro α permite caracterizar la textura del objetivo. Para valores cercanos a cero, típicamente en el intervalo $(-3,0)$, la zona de la imagen corresponde a una región muy texturada, o extremadamente heterogénea como es el caso de las zonas urbanas. A medida que el valor disminuye, estamos en presencia de zonas cuya textura o heterogeneidad es menor, como son las regiones de forestación, usualmente en $(-6,-3)$; y de pastura habitualmente en $(-\infty,-6)$. Por otro lado, el llamado parámetro de escala γ , posee una interpretación en términos de brillo. Cuanto menor es su valor, menor es el nivel de intensidad que posee la imagen en ese objetivo. Finalmente, el parámetro L equivale al número de looks.

En Chan *et al.* (2016), se proponen cuatro maneras distintas de generar datos que sigan una distribución G_I^0 con la plataforma R. De estas cuatro maneras elegimos la que se basa en la distribución Pareto, para generar una imagen SAR sintética de tamaño $800\text{x}900$. Esto fue posible porque consideramos el caso de una sola vista; es decir, $L=1$ La imagen consta de seis zonas del mismo tamaño combinando los siguientes valores para los parámetros:

$$\alpha \in \{-1.05,\ -3,\ -10\} \text{ y } \gamma \in \{0.1,\ 100\}.$$

En cuanto a las imágenes SAR reales (que se presentan en la Figura 1), consideramos las siguientes:

▪**British Columbia** (05/06/2010, 06:39:28), nombre del sitio: British Columbia- Campbell River - Clearcut Site. Tamaño de la imagen: $4497\text{x}5239$. En la misma se ve presencia de agua, de zonas urbanas y de baja vegetación.

▪**Greenland** (21/08/2008, 00:33:24), nombre del sitio: KULU. Tamaño de la imagen: 4754x5400. En la misma se ve presencia de agua.

▪**Massachusetts** (16/10/2010, 03:11:55), nombre del sitio: Harvard Forest EMS Tower (HFR1). Tamaño de la imagen: 4740x5385. En la misma se ve presencia de agua y de copas de árboles.

▪**New Mexico** (05/10/2010, 05:33:57), nombre del sitio: Valles Caldera Mixed Conifer. Tamaño de la imagen: 4544x5223. En la misma se ve presencia de copas de árboles y de rocas desnudas.

▪**San Francisco** (1988). Tamaño de la imagen: 900x1024. En la misma se ve presencia de zona urbana, vegetación y agua.

▪**Saskatchewan** (19/07/2010, 05:18:25), nombre del sitio: BOREAS SSA Young Aspen. Tamaño de la imagen: 4604x5226. En la misma se ve presencia de copas de árboles.

La imagen de San Francisco muestra el área de su bahía y fue obtenida por el Airborne Synthetic Aperture Radar (AIRSAR) diseñado y construido por el Jet Propulsion Laboratory (JPL) de la NASA (https://airsar.jpl.nasa.gov/). El resto de las imágenes fueron obtenidas por el Phased Array type L-band Synthetic Aperture Radar (PALSAR) que vuela sobre el Advanced Land Observing Satellite (ALOS) del Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) de la NASA (ORNL DAAC, 2011). De las polarizaciones consideramos la banda HH, que incluye rojo y azul.

Costo computacional

Para comparar el rendimiento de los lenguajes replicamos los procesos correspondientes 50 veces y tomamos el promedio del tiempo en segundos transcurrido (*elapsed time*). Usamos el procesador Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41 GHz, 16.0 GB RAM.

Métricas de calidad de la imagen

Al evaluar la calidad de la imagen se necesitan métricas automáticas y robustas que sean consistentes estadísticamente con lo observado por el ojo humano. Entre las métricas de referencia

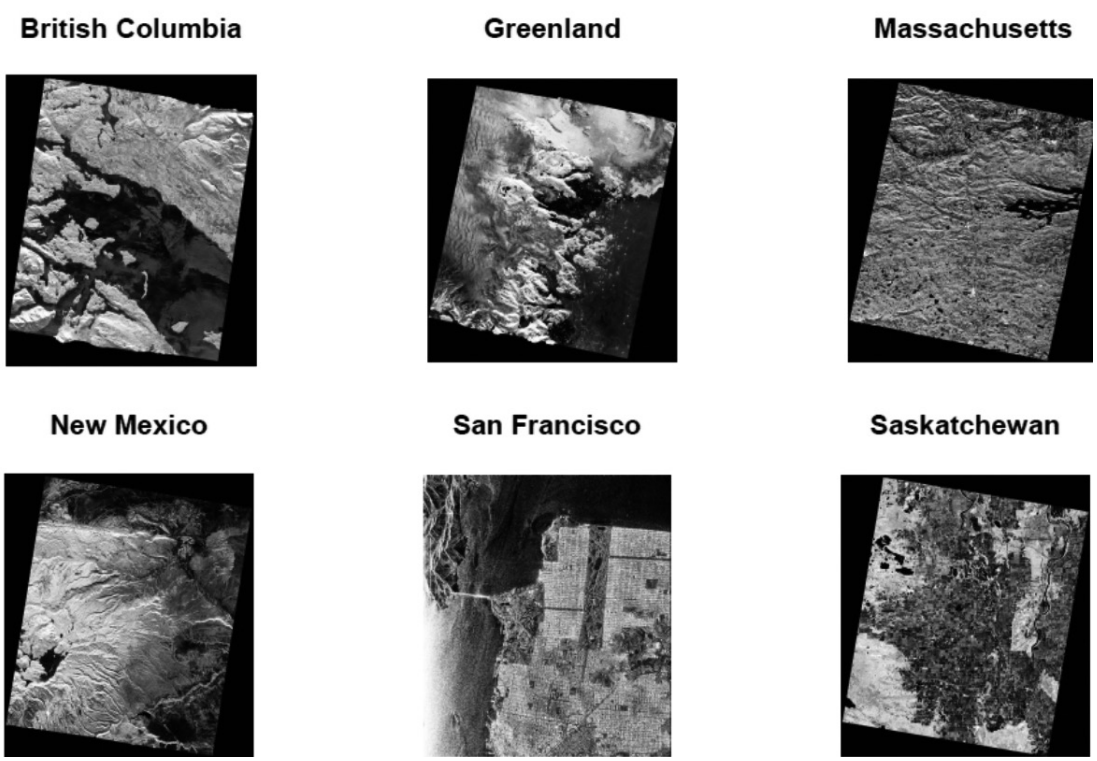

Fig. 1. Imágenes SAR reales sobre las cuales aplicamos segmentación

completa, que utilizamos en nuestro trabajo, se encuentran la proporción pico de señal a ruido, PSNR *(peak signal-to-noise ratio)*, que opera directamente con la intensidad de la imagen no correlacionando bien con las calificaciones subjetivas de fidelidad; y la similitud estructural, SSIM *(structure similarity)*, motivada por la necesidad de captar la pérdida de estructura de la imagen y basada en la hipótesis de que el sistema visual humano está altamente adaptado para extraer la información estructural de la escena visual.

La métrica de referencia completa más simple y más usada para evaluar la calidad de imagen es el error cuadrático medio, MSE *(mean squared error)*, que se calcula como la media de los cuadrados de las diferencias entre las intensidades de la imagen original y de su segmentación. Junto con la misma, se computa el PSNR. Si bien son sencillas, poseen claros significados físicos y son matemáticamente convenientes en el contexto de optimización; las mismas no se adaptan suficientemente bien a la calidad visual percibida (Girod, 1993; Wang *et al.* 2002).

El PSNR es usado en ingeniería para medir la relación entre la máxima potencia posible de una señal y el poder de corrupción que tiene el ruido para afectar la fidelidad de su representación, se mide en decibeles y se define como

$$PSNR = 20\,log\left(\frac{255}{RMSE}\right)$$

donde se considera el logaritmo decimal y, si I e I son respectivamente las imágenes original y segmentada de tamaño *M x N* , la raíz del error cuadrático medio *(root mean squared error)* está dado por

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - \hat{I}(i,j)\right]^2}\,.$$

Cabe destacar que cuanto mayor sea valor del PSNR, mejor será la codificación de la imagen segmentada.

SSIM es introducido por Wang *et al.* (2004) y se calcula en varias ventanas de la imagen del mismo tamaño. Para dos ventanas e , se define

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

donde
μ_x es el promedio de x ,
μ_y es el promedio de y ,
σ_x^2 es la varianza de x ,
σ_y^2 es la varianza de y ,
σ_{xy} es la covarianza de x e y ,
$c_{1_1} = (k_1 D)^2$ y $c_2 = (k_2 D)^2$
son dos variables para estabilizar el cociente con denominador próximo a cero, D es el rango dinámico de los valores de píxeles (en general, es 2 elevado al número de bits por pixel , menos 1) y, por defecto, $k_1 = 0,01$ y $k_2 = 0,03$ Este índice toma valores reales entre 0 y 1. El valor 1 se alcanza solamente cuando la imagen segmentada es idéntica a la imagen original. Mientras que el valor 0 indica que no existe similitud estructural entre ambas.

En estudios recientes (Sheikh *et al.* 2006; Ponomarenko *et al.* 2009) se ha demostrado que SSIM puede ofrecer un mejor comportamiento al predecir la fidelidad de una imagen respecto de otras métricas de evaluación de calidad de imagen.

Resultados y Discusión

La función ThresholdML() de la biblioteca imagerExtra de R segmenta una imagen en escala de grises seleccionando la cantidad (fijada por el usuario) de umbrales, basándose en el algoritmo de una colonia artificial de abejas (Horng, 2011). Esta función ofrece cuatro opciones para parámetros preestablecidos, eligiendo como valor de su argumento thr: fast, precise, manual o dejarlo vacío. Debido a estas alternativas, en nuestra primera etapa de estudio comparamos estas variantes para la imagen sintética segmentando en 6 regiones (5 umbrales), dato que conocemos por la manera en que la generamos. En la Figura 2, exhibimos la imagen sintética original y sus segmentaciones dependiendo de la opción elegida para el argumento thr. Además, en la Tabla 1 mostramos para cada alternativa, el tiempo consumido por el procesamiento, el

Fig. 2. Imagen original y sus correspondientes segmentaciones dependiendo del valor asignado al argumento thr de la función ThresholdML()

Tabla 1. Comparación del rendimiento de la segmentación de la imagen SAR sintética para distintas opciones del argumento thr de la función ThresholdML()

thr =	sin declarar	fast	precise	manual
Tiempo	0,0890	0,0844	0,2412	0,0862
PSNR	93,22075	93,23294	93,42706	93,74376
SSIM	0,999847	0,999999	0,999823	0,999852

PSNR y el SSIM entre la imagen original y la imagen segmentada obtenida en cada caso. En la misma señalamos en gris los mejores resultados obtenidos.

Visualmente, no se observan diferencias significativas entre las distintas segmentaciones. Podemos observar que la alternativa *fast* supera a las demás en costo computacional y en el índice de similitud estructural. Si bien la misma es superada en el valor de la proporción pico entre señal y ruido por la opción manual, el valor alcanzado por *fast* es aceptable. Entonces, en lo que sigue optamos siempre por la opción fast al trabajar con imágenes SAR reales.

Al trabajar en *Python*, la función *threshold_multiotsu*() de la biblioteca *scikit-image* encuentra los umbrales óptimos de una imagen en escala de grises aplicando una generalización del método de Otsu (Otsu, 1979), donde el usuario fija la cantidad de umbrales.

A continuación, mostramos la estructura básica de los códigos de programación implementados para nuestro estudio.

```
# Cargamos las bibliotecas para el procesamiento de imágenes en R
Library (imager)
Library (imagerExtra)

imagen <- load.image (ruta del archivo') # leemos la imagen
k <- asignamos el número de umbrales deseado
segmentada <- ThresholdML(imagen, thr=fast, k) # segmentamos la imagen
save.image(segmentada, nombre del archivo) # guardamos la imagen
```

```
# Cargamos las bibliotecas para el procesamiento de imágenes en Python
from skimage import io
from skimage.filters import threshold_multiotsu
import numpy as np
import matplotlib.pyplot as plt

imagen = io.imread(ruta del archivo) # leemos la imagen
k = asignamos el número de umbrales deseado
umbrales = threshold_multiotsu (imagen, k) # calculamos los umbrales
segmentada = np.digitize(imagen, bins= umbrales) # segmentamos la imagen
plt.imsave (nombre del archivo) # guardamos la imagen
```

Como podemos apreciar, en R se requiere de dos bibliotecas, mientras que en Python se necesitan cuatro. En cuanto a la segmentación, el primero lo hace en una sola línea de código y para el segundo hay que utilizar dos. Si bien, podemos deducir que R es más simple en función de su código, se aprecia en la Figura 3 que Python consume alrededor de la quinta parte del tiempo que consume R para generar la imagen segmentada. Notamos que, para el caso de la imagen de San Francisco, el tiempo de procesamiento en ambos lenguajes es significativamente menor que en el resto de las imágenes, lo que es consecuencia del tamaño de las mismas. Para todas las cantidades de umbrales consideradas, los tiempos de procesamiento son similares excepto para el caso de cuatro umbrales, donde Python consume aproximadamente cuatro veces más, siendo incluso mucho mayor que el que necesita R para el caso de la imagen de San Francisco.

En la Figura 4, observamos el comportamiento esperado para la proporción pico entre señal y ruido que crece en la medida en que el número de umbrales aumenta. Para todas las imágenes de estudio, los resultados son muy buenos en ambos lenguajes. Existe una diferencia entre Python y R a favor del primer lenguaje, y que crece en función del aumento en la cantidad de umbrales. Sin embargo, en el caso de la imagen de San Francisco, el comportamiento de ambos lenguajes es muy parejo. Como hemos mencionado anteriormente, uno de los mayores inconvenientes de las técnicas de selección de umbrales múltiples, es que se desconoce la cantidad de umbrales óptima y se carece de una manera automática de calcularla. Algunos autores, por ejemplo: Arora *et al.*(2008), sugieren elegir como la cantidad recomendada de regiones a aquella para la cual la tasa de crecimiento de PSNR no supera el 10%. En nuestro caso, esto sucede en cuatro clases (equivalentemente, tres umbrales) para las seis imágenes consideradas y aplicando la segmentación en ambos lenguajes.

Finalmente, la última métrica de evaluación que empleamos es la similitud estructural, para la cual mostramos su cómputo en la Figura 5. En la misma podemos observar que los valores obtenidos en todos los casos tratados son extremadamente satisfactorios al estar muy próximos a uno. Nuevamente, Python supera a R, excluyendo a la imagen de San Francisco, donde se emparejan.

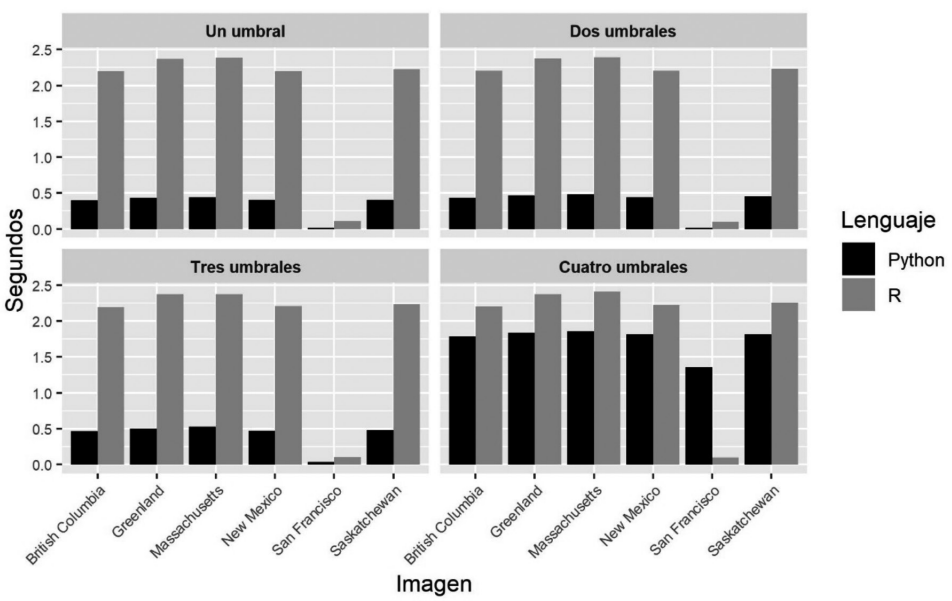

Fig. 3. Tiempo de procesamiento transcurrido (en segundos) en la segmentación de imágenes en función de lenguaje y de la cantidad de umbrales

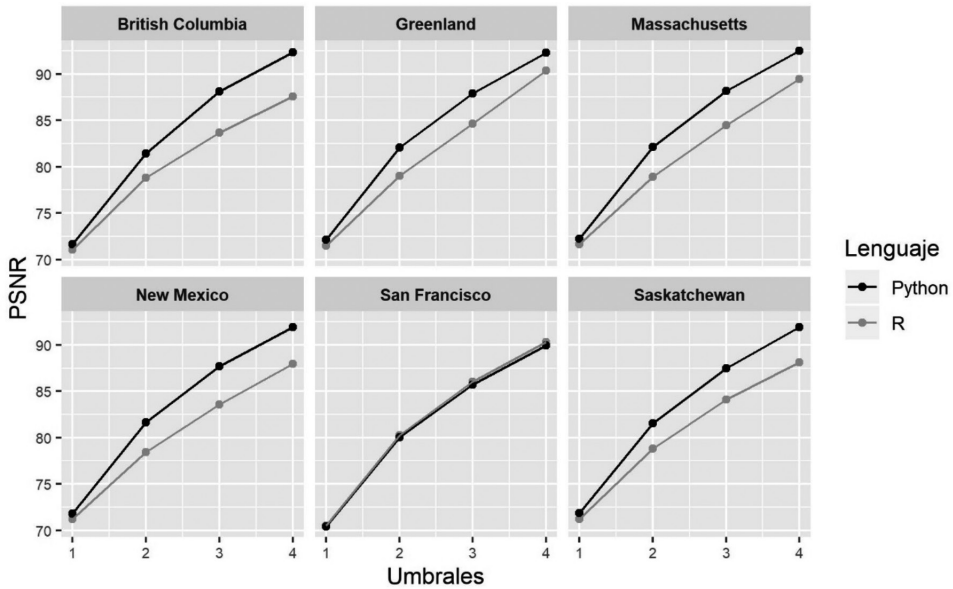

Fig. 4. Cómputo de PSNR en la segmentación de imágenes en función de lenguaje y de la cantidad de umbrales

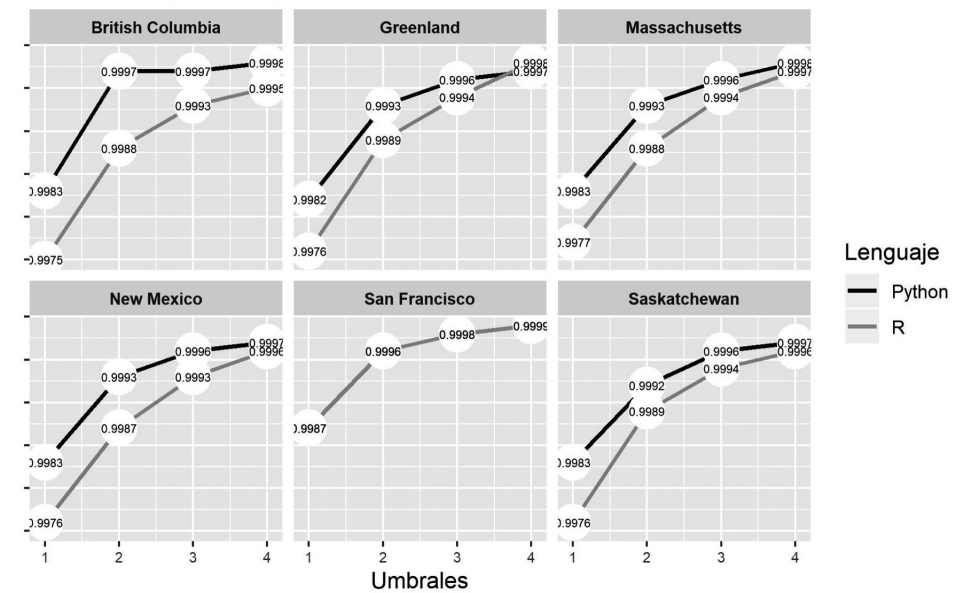

Fig. 5. Cómputo de SSIM en la segmentación de imágenes en función de lenguaje y de la cantidad de umbrales

A modo ilustrativo, en las Figuras 6, 7, 8 y 9, elegimos las imágenes de British Columbia y de San Francisco, para manifestar visualmente los resultados en la segmentación obtenida por cada lenguaje, con variación en la cantidad de umbrales. Como ya mencionamos, se recomienda utilizar tres umbrales para las imágenes analizadas.

En la Tabla 2 exhibimos los valores de los umbrales hallados por cada lenguaje considerando el histograma de los 256 niveles de gris de cada una de las imágenes. Notamos que en todas las imágenes obtenidas por el PALSAR los valores se asemejan entre sí, pero éstos dependen fuertemente del lenguaje con el que se trabaja. Más aún, dentro de estas semejanzas, podemos clasificar a las imágenes en dos grupos: Massachusetts y Greenland por un lado y las otras tres zonas, por el otro. Sin embargo, en el caso de la imagen de San Francisco, obtenida por el AIRSAR, los valores de los umbrales difieren de los obtenidos para el resto de las imágenes, siendo similares entre los dos lenguajes.

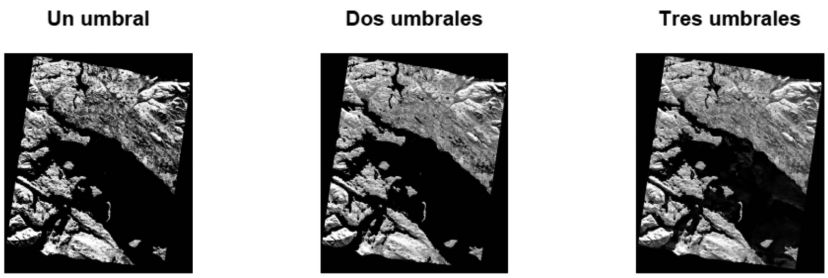

Fig. 6. Segmentación de la imagen de British Columbia en R

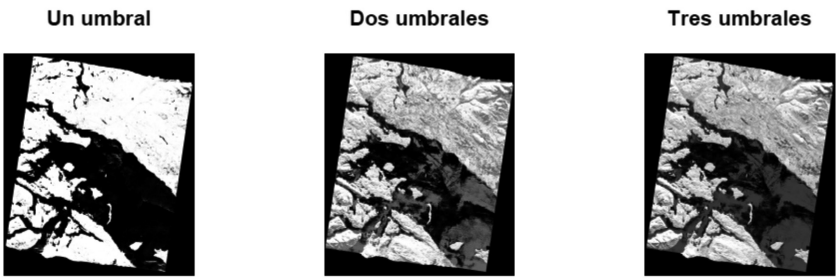

Fig. 7. Segmentación de la imagen de British Columbia en Python

Fig. 8. Segmentación de la imagen de San Francisco en R

Fig. 9. Segmentación de la imagen de San Francisco en Python

Tabla 2. Umbrales óptimos

Imagen	Cantidad de umbrales	Lenguaje	Umbrales
British Columbia	1	Python	105
		R	162
	2	Python	61 - 157
		R	116 - 188
	3	Python	44 - 115 - 185
		R	91 - 148 - 203
	4	Python	33 - 89 - 145 - 200
		R	74 - 121 - 167 - 213
Greenland	1	Python	10
		R	164
	2	Python	60 - 157
		R	120 - 190
	3	Python	41 - 112 - 183
		R	87 - 144 - 200
	4	Python	31 - 87 - 143-199
		R	70 - 116 - 163 - 210
Massachusetts	1	Python	101
		R	163
	2	Python	60 - 157
		R	120 - 189
	3	Python	42 - 113 - 184
		R	92 - 147 - 201
	4	Python	32 - 88 - 144 - 200
		R	70 - 116 - 164 - 208
New Mexico	1	Python	104
		R	161
	2	Python	61 - 158
		R	119 - 190
	3	Python	43 - 114 - 184
		R	92 - 148 - 204
	4	Python	33 - 89 - 144 - 199
		R	75 - 121 - 167 - 213
San Francisco	1	Python	128
		R	131
	2	Python	86 - 171
		R	88 - 175
	3	Python	65 - 129 - 192
		R	66 - 130 - 194
	4	Python	53 - 104 - 155 - 206
		R	53 - 106 - 157 - 207
Saskatchewan	1	Python	104
		R	162
	2	Python	61 - 157
		R	118 - 188
	3	Python	43 - 113 - 183
		R	91 - 147 - 204
	4	Python	33 - 88 - 144 - 199
		R	74 - 121 - 167 - 214

Conclusiones

La segmentación de imágenes es una técnica ampliamente utilizada por las ventajas que la misma ofrece al trabajar con una imagen más sencilla que la original, pero que preserva su misma estructura morfológica. Las imágenes SAR tienen importantes aplicaciones en el estudio de la Tierra. Combinando ambas cuestiones, resulta de interés contar con buenos métodos de segmentación para este tipo de imágenes.

Resaltamos, en primer lugar, las diferencias que hemos observado en todos los indicadores de evaluación elegidos, entre la imagen de la Bahía de San Francisco y el resto de las imágenes. Nos lleva a pensar en dos posibles causas para estas discrepancias: 1.- el radar con el cual se obtienen las imágenes es distinto, 2.- el tamaño de las mismas difiere significativamente, ya que el de la imagen de San Francisco representa aproximadamente el 40% del tamaño de las otras cinco imágenes.

Por otro lado, de acuerdo con nuestros resultados, debemos recomendar Python como el mejor lenguaje para la selección de umbrales múltiples en imágenes SAR. Igualmente, aquel usuario que se sienta más cómodo programando en R, no debe desalentarse ya que los resultados obtenidos por el mismo están dentro de los parámetros esperados para una buena calidad de imagen en la segmentación.

Referencias

ABAK, A. T.; BARIS, U.; SANKUR, B., (1997) The performance evaluation of thresholding algorithms for optical character recognition. In Proceedings of the fourth international conference on document analysis and recognition (Vol. 2, pp. 697-700). IEEE.

AL-ATTAS, R.; EL-ZAART, A., (2007) Thresholding of medical images using minimum cross entropy. In 3rd Kuala Lumpur International Conference on Biomedical Engineering 2006 (pp. 296-299). Springer, Berlin, Heidelberg.

ARORA, S.; ACHARYA, J.; VERMA, A. y PANIGRAHI, P. K., (2008) Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. Pattern Recognition Letters, 29(2), 119-125.

AZGHANI, M.; KOSMAS, P. y MARVASTI, F., (2014) Microwave medical imaging based on sparsity and an iterative method with adaptive thresholding. IEEE transactions on medical imaging, 34(2), 357-365.

BARTHELME, S., (2019) imager: Image Processing Library Based on 'CImg'. R package version 0.41.2.

BHANU, B., (1986) Automatic target recognition: State of the art survey. IEEE transactions on aerospace and electronic systems, (4), 364-379.

BRINK, A. D., (1992) Thresholding of digital images using two-dimensional entropies. Pattern recognition, 25(8), 803-808.

CHAN, D.; REY, A.; CASSETTI, J.; GAMBINI, J. y FRERY, A. C., (2016) Comparison of data generation methods using GI0 distribution (for the single look case), ponencia presentada en el I Latin American Conference on Statistical Computing, Gramado, Brazil.

FRERY, A. C.; MULLER, H. J.; YANASSE, C. D. C. F. y SANT'ANNA, S. J. S., (1997) A model for extremely heterogeneous clutter. IEEE transactions on geoscience and remote sensing, 35(3), 648-659.

GIROD, B., (1993) What's wrong with mean-squared error? Digital images and human vision, 207-220.

HORNG, M. H., (2011) Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Systems with Applications, 38(11), 13785-13791.

KAMEL, M. y ZHAO, A., (1993) Extraction of binary character/graphics images from grayscale document images. CVGIP: Graphical Models and Image Processing, 55(3), 203-217.

KAPUR, J. N.; SAHOO, P. K. y WONG, A. K., (1985) A new method for gray-level picture thres-

holding using the entropy of the histogram. Computer vision, graphics, and image processing, 29(3), 273-285.
KITTLER, J. y ILLINGWORTH, J., (1986) Minimum error thresholding. Pattern recognition, 19(1), 41-47.
KURITA, T.; OTSU, N. y ABDELMALEK, N., (1992) Maximum likelihood thresholding based on population mixture models. Pattern recognition, 25(10), 1231-1240.
MANCAS, M.; GOSSELIN, B. y MACQ, B., (2005) Segmentation using a region-growing thresholding. In Image Processing: Algorithms and Systems IV (Vol. 5672, pp. 388-398). International Society for Optics and Photonics.
OCHI, S., (2019) imagerExtra: Extra Image Processing Library Based on 'imager'. R package version 1.3.2.
OLIPHANT, T. E., (2006) A guide to NumPy (Vol. 1). Trelgol Publishing USA.
OTSU, N. (1979) A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.
ORNL DAAC, (2011) Oak Ridge National Laboratory Distributed Active Archive Center, Alaska Satellite Facility Distributed Active Archive Center, and Japan Aerospace Exploration Agency. SAR Subsets for Selected Field Sites, 2007-2010. Data set. Available on-line [http://daac.ornl.gov] from ORNL DAAC, Oak Ridge, Tennessee, U.S.A. doi:10.3334/ORNLDAAC/993.
PEAK, J. E. y TAG, P. M., (1994) Segmentation of satellite imagery using hierarchical thresholding and neural networks. Journal of Applied Meteorology, 33(5), 605-616.
PONOMARENKO, N.; LUKIN, V.; ZELENSKY, A.; EGIAZARIAN, K.; CARLI, M. y BATTISTI, F., (2009) TID2008-a database for evaluation of full-reference visual quality assessment metrics. Advances of Modern Radioelectronics, 10(4), 30-45.
R CORE TEAM, (2013) R: A language and environment for statistical computing. https://www.R-project.org/.
REDDI, S. S.; RUDIN, S. F. y KESHAVAN, H. R., (1984) An optimal multiple threshold scheme for image segmentation. IEEE Transactions on Systems, Man, and Cybernetics, (4), 661-665.
SEZGIN, M. y SANKUR, B., (2004) Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic imaging, 13(1), 146-166.
SHEIKH, H. R.; SABIR, M. F. y BOVIK, A. C., (2006) A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on image processing, 15(11), 3440-3451.
SONI, V.; BHANDARI, A. K.; KUMAR, A. y SINGH, G. K., (2013) Improved sub-band adaptive thresholding function for denoising of satellite image based on evolutionary algorithms. IET Signal Processing, 7(8), 720-730.
TRIER, O. D. y JAIN, A. K., (1995) Goal-directed evaluation of binarization methods. IEEE transactions on Pattern analysis and Machine Intelligence, 17(12), 1191-1201.
TSAI, W. H., (1985) Moment-preserving thresolding: A new approach. Computer Vision, Graphics, and Image Processing, 29(3), 377-393.
VAN DER WALT, S.; SCHÖNBERGER, J. L.; NUÑEZ-IGLESIAS, J.; BOULONGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E. y YU, T., (2014) scikit-image: Image processing in Python. PeerJ 2:e453.
VAN ROSSUM, G. y DRAKE JR, F., (1995) Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.
WANG, Z.; BOVIK, A. C. y LU, L., (2002) Why is image quality assessment so difficult? In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 4, pp. IV-3313). IEEE.
WANG, Z.; BOVIK, A. C.; SHEIKH, H. R. y SIMONCELLI, E. P., (2004) Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.
WANG, Q.; XUE, J.; ZHAO, R.; CHI, Z. y FENG, D., (2002) On the maximization of the crispness of 2D grayscale histogram for image thresholding. In 6th International Conference on Signal Processing, 2002. (Vol. 2, pp. 981-984). IEEE.