



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA  
INGENIERÍA ELÉCTRICA – PROCESAMIENTO DIGITAL DE SEÑALES**

**CARACTERIZACIÓN DE IMÁGENES CON HIELO PANCAKE  
MEDIANTE TRANSFORMADA HERMITE Y CUANTIZACIÓN  
VECTORIAL**

**TESIS  
QUE PARA OPTAR POR EL GRADO DE  
MAESTRA EN INGENIERÍA ELÉCTRICA**

**PRESENTA:  
ING. CECILIA DELGADO SOLORZANO**

**TUTOR PRINCIPAL:  
DR. BORIS ESCALANTE RAMÍREZ  
FACULTAD DE INGENIERÍA**

**CIUDAD UNIVERSITARIA, CD. MX. DICIEMBRE 2018**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



## **JURADO ASIGNADO**

Presidente: Dr. Savage Carmona Jesús

Secretario: Dr. Padilla Castañeda Miguel A.

Vocal: Dr. Escalante Ramírez Boris

1 er. Suplente: Dr. Rascón Estebané Caleb

2 d o. Suplente: Dra. Olveres Montiel Jimena

Lugar o lugares donde se realizó la tesis: Ciudad Universitaria

**TUTOR DE TESIS:**

Dr. Boris Escalante Ramírez

---

FIRMA

**Acerca del trabajo:**

”No es lo que sabes... es la capacidad que tienes de aprender cosas nuevas”

Dr. Buzo

”Los que sobreviven son aquellos que saben adaptarse a los cambios”

Dr. Savage

**Acerca del infinito:**

”No confío en el infinito de Cero a Uno”

Ing. Cecilia Delgado

”Que se quede el infinito sin estrellas...”

Bobby Capó

# Agradecimientos

Agradezco el apoyo de Conacyt- Sener por la beca otorgada que fue parte importante para el desarrollo de este trabajo y al proyecto PAPIIT IN116917.

Mi más profundo agradecimiento a mi hermano Jonathan que siempre ha confiado en mí, sin su apoyo no lo hubiera logrado. También mi agradecimiento eterno a mi familia en especial a mi tía Rosa que siempre ha visto por mis estudios, he logrado llegar hasta este nivel gracias a ella. Por supuesto que mis padres también son parte primordial de este logro.

Agradezco a mis profesores de la maestría por todo el conocimiento aportado en este tiempo. Al Dr. Boris y al equipo del laboratorio Lapi, por su paciencia y amabilidad conmigo.

Al Dr. Savage por toda su aportación sobre el algoritmo de Cuantización Vectorial por su apoyo a desarrollar esta técnica en imágenes. Al Mtro Abel Pacheco por su contribución en la máscara de la zona de interés de la imagen E9 y los debates que contribuyeron a ideas en el tema.

A mis compañeros de maestría tanto del Posgrado de Ingeniería como del PCIC, por todo lo que estudiamos y vivimos juntos en este proceso.

# Resumen

El presente trabajo explora la versatilidad del algoritmo de Cuantización Vectorial [13] en imágenes junto con la Transformada Hermite para generar codebooks.

Un primer paso en el procesamiento de imágenes es la segmentación, algunos métodos muy usados son superpíxeles y umbrales. La segmentación puede hacerse de diversas formas dependiendo de la característica que se escoja, esta puede ser el color, escala de grises, vecindad o combinación entre ellas. Para utilizar Cuantización Vectorial específicamente el algoritmo LBG, se escogió la respuesta al sensor como principal característica, de esta forma se generan regiones de respuesta homogénea al sensor en distintos centroides.

Posteriormente a partir de ellas y junto con estadísticos de primer orden se genera un descriptor de textura.

Cuantización Vectorial como su nombre lo indica es principalmente para vectores, por ello se buscó combinar con la técnica de Transformada Hermite, que entrega coeficientes que describen la textura de una imagen. Generalmente son representados como imágenes submuestreadas, también son un conjunto de vectores que describen a cada punto dentro de la imagen; por ello pueden utilizarse junto a Cuantización Vectorial para generar un codebook que describa la textura.

Si se aplica Cuantización Vectorial a cada coeficiente de la Transformada Hermite, pueden observarse los patrones dentro de cada coeficiente, de tal forma que se podría determinar si algún patrón en algún coeficiente no corresponde al patrón principal del coeficiente cero, entonces el conjunto de este centroide podría eliminarse. Al hacer la Transformada inversa de Hermite se obtendría la imagen filtrada. Actualmente este proceso se realiza con umbrales.

# Contenido

<b>Agradecimientos</b>	<b>v</b>
<b>Resumen</b>	<b>vi</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del Problema . . . . .	2
1.1.1. Objetivos . . . . .	3
1.1.2. Organización de la tesis . . . . .	3
<b>2. Estado del Arte</b>	<b>4</b>
2.1. Antecedentes . . . . .	4
2.1.1. Misión SAR, Satélite Sentinel, Resolución . . . . .	4
2.1.2. Hielo Pancake, zona de hielo Pancake . . . . .	5
2.1.3. Ruido Speckle . . . . .	6
2.2. Vecindad . . . . .	8
2.3. Segmentación y Agrupamiento . . . . .	8
2.3.1. Agrupamiento. . . . .	9
2.3.2. K-means vs. Cuantización Vectorial . . . . .	10
2.3.3. Umbrales. . . . .	11
2.3.4. Superpíxeles . . . . .	12
2.4. Textura . . . . .	13
<b>3. Propuesta del Sistema</b>	<b>16</b>
3.0.1. K-means . . . . .	16
3.0.2. Cuantización Vectorial . . . . .	17
3.1. Uso del Algoritmo Cuantización Vectorial . . . . .	19
3.2. Metodología . . . . .	20
3.2.1. Uso de CV1D como segmentador. . . . .	20
3.2.2. Regiones de Textura Homogénea . . . . .	20
3.2.3. Transformada Hermite con Cuantización Vectorial Forma Vectorial . . . . .	21
3.2.4. Metodología para imágenes SAR . . . . .	26
<b>4. Pruebas y Resultados</b>	<b>29</b>
4.1. Pruebas propuestas . . . . .	29
4.1.1. Para su uso como segmentador . . . . .	29
4.2. Cuantización Vectorial para segmentar.(Superpíxeles vs Cuantización Vectorial)	30
4.2.1. Imagen Kobi . . . . .	30
4.2.2. Fotografía hielo Pancake . . . . .	37



---

4.3. Umbrales vs Cuantización Vectorial . . . . .	40
4.4. Imágenes Satelitales . . . . .	42
4.5. Textura por Regiones Estadísticas . . . . .	44
4.5.1. Pruebas con Cuantización Vectorial y Transformada Hermite . . . . .	52
4.5.2. CV1D en Coeficiente cero de TH . . . . .	52
4.5.3. Uso de Cuantización Vectorial junto con vector de coeficientes de Transformada Hermite, para generar un codebook. . . . .	55
<b>5. Conclusiones y Trabajo Futuro</b>	<b>57</b>
5.1. Conclusiones . . . . .	57
5.2. Trabajo Futuro . . . . .	58
<b>A. Anexo: Ventanas de análisis imágenes SAR</b>	<b>63</b>
<b>B. Anexo: Tabla de centroides vectoriales de TH con CV</b>	<b>67</b>
<b>C. Anexo: Código empleado para Cuantización Vectorial</b>	<b>70</b>
<b>Bibliografía</b>	<b>71</b>

# Lista de Figuras

1.1. Imagen SAR D419 . . . . .	1
1.2. Imagen SAR E9 con segmentación del experto . . . . .	2
2.1. Fotografía de hielo Pancake . . . . .	6
2.2. Principio de Huygens . . . . .	7
2.3. Vecindad 4 de un píxel . . . . .	8
2.4. Vecindad 8 de un píxel . . . . .	8
2.5. Histograma de una imagen . . . . .	11
2.6. Ejemplos de Textura . . . . .	14
3.1. Imagen de la diosa Gaia . . . . .	23
3.2. Transformada Hermite de la imagen de la Diosa Gaia . . . . .	24
3.3. Representación de los vectores de coeficientes de Transformada Hermite. . . . .	25
3.4. Diagrama de procesos con Cuantización Vectorial . . . . .	26
3.5. Imagen E9 con segmentación del experto y ventanas de análisis . . . . .	27
3.6. Diagrama Clasificación de la Imagen . . . . .	27
3.7. Diagrama de flujo de selección de ventanas para clasificar por TRE. . . . .	28
4.1. Imágenes de perro Kobi . . . . .	31
4.2. CV1D imagen Kobi sencilla . . . . .	32
4.3. Contornos juntos Kobi sencilla . . . . .	33
4.4. Comparación CV1D vs superpíxeles . . . . .	34
4.5. Resultado CV1D en Kobi con margen blanco . . . . .	35
4.6. Resultado Superpíxeles en Kobi con margen . . . . .	36
4.7. Fotografía de hielo Pancake. . . . .	37
4.8. Resultado Superpíxeles en imagen hielos Pancakes . . . . .	38
4.9. Resultado CV1D en imagen hielos Pancake . . . . .	39
4.10.CV1D vs Umbrales . . . . .	40
4.11.Histogramas CV1D vs Umbrales . . . . .	41
4.12.Pseudocolor CV1D vs Umbrales . . . . .	41
4.13.Imagen D419 con Segmentación y ventana de análisis . . . . .	42
4.14.Ventana de análisis de D419 . . . . .	43
4.15.Máscaras obtenidas a partir de CV1D para 4 centroides . . . . .	43
4.16.Resultado de CV1D en ventana de imagen D419 . . . . .	43
4.17.Ventana de análisis zona de hielo Pancake 4. . . . .	44
4.18.Ejemplo de ventana de análisis de imagen E9 . . . . .	44
4.19.Máscaras para ventana de análisis de E9 . . . . .	45
4.20.Ventanas cuya clasificación está en 2 categorías diferentes. . . . .	46

4.21. Ventana hielo1 cuya clasificación está en 2 categorías diferentes. . . . .	47
4.22. Máscara de Tierra para la imagen E9. . . . .	47
4.23. Resultado de clasificación con TRE . . . . .	48
4.24. Posibles falsos positivos o positivos falsos imagen E9 . . . . .	49
4.25. Zona de interés y Máscara de tierra para la imagen D419. . . . .	50
4.26. Clasificación de imagen D419 con codebooks de imagen E9 . . . . .	51
4.27. CV1D aplicado en el coeficiente cero de la Transformada Hermite de la imagen E9. . . . .	53
4.28. CV1D aplicado en el coeficiente cero de la Transformada Hermite de la imagen E9, con máscara de tierra. . . . .	54
4.29. Resultado coeficiente cero TH con CV1D con máscara de tierra . . . . .	54
4.30. Máscaras de extracción de vectores de la imagen E9 . . . . .	56
5.1. TH de ventana 6 de hielo Pancake . . . . .	60
5.2. CV1D aplicada a cada coeficiente de TH . . . . .	61
5.3. Diagrama Propuesta para generar filtro con TH y CV1D . . . . .	62
A.1. Subimágenes de la zona de hielo . . . . .	63
A.2. Falso color zona de hielo . . . . .	64
A.3. Subimágenes de la zona de hielo Pancake . . . . .	64
A.4. Falso color para zona de hielo pancake . . . . .	65
A.5. Contornos zona de hielo pancake . . . . .	65
A.6. Subimágenes de la zona de mar . . . . .	66
A.7. Falso color zona de Mar . . . . .	66

# Lista de Tablas

<b>2.1.</b> Características de las imágenes SAR . . . . .	5
<b>2.2.</b> K-means vs Cuantización Vectorial (LBG) . . . . .	10
<b>4.1.</b> Tabla vectores de Textura por Regiones Estadísticas . . . . .	45
<b>4.2.</b> Matriz de confusión para el codebook conformado por una ventana por cada región a separar con un total de 12 vectores. . . . .	46
<b>4.3.</b> Matriz de confusión para el codebook conformado por 7 ventanas por cada región a separar con un total de 84 vectores. . . . .	46
<b>B.1.</b> Centroides TH con CV zona hielo Pancake . . . . .	67
<b>B.2.</b> Centroides TH con CV zona hielo compacto . . . . .	68
<b>B.3.</b> Centroides TH con CV zona de Mar . . . . .	68
<b>B.4.</b> Centroides TH con CV zona de Tierra . . . . .	69

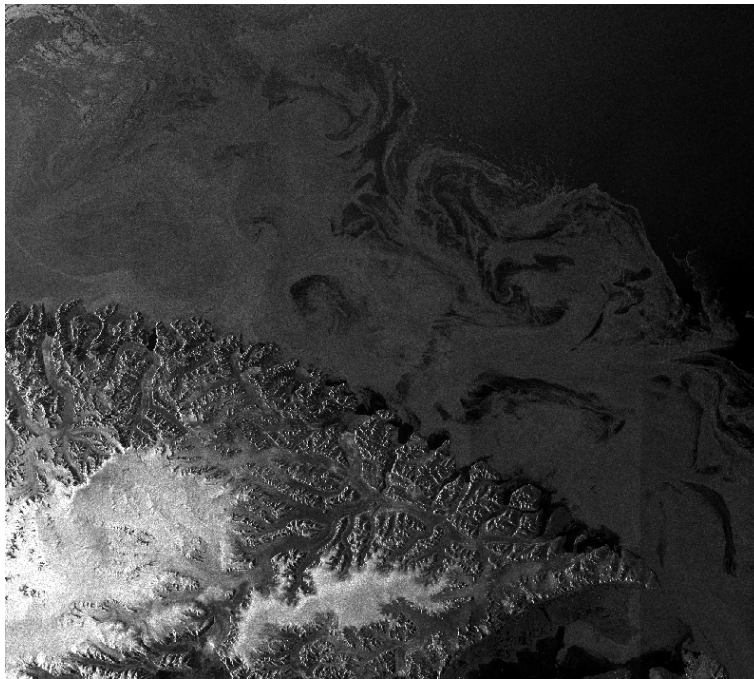
## Abreviaturas

<b>Abreviatura</b>	<b>Término</b>
<i>SAR</i>	Satelites de Apertura Sintética
<i>TH</i>	Transformada Hermite
<i>CV</i>	Cuantización Vectorial
<i>CV1D</i>	Cuantización Vectorial Unidimensional
<i>LBG</i>	Algoritmo Linde, Buzo, Gray
<i>TRE</i>	Texturas por Regiones Estadísticas

# 1. Introducción

En el año 2014 fue lanzada al espacio la misión Sentinel para toma de imágenes satelitales de apertura sintética de radar (SAR). Uno de los objetivos de esta misión es la búsqueda de zonas con hielo Pancake. El estudio de la salinidad, el proceso de convección del océano y la variación del tamaño de estas zonas está relacionado directamente con el cambio climático.

El tiempo de análisis de estas imágenes puede ser mayor al tiempo de vida del hielo, esto es debido a las dimensiones de la imagen que pueden ser de 10453 x 9996 píxeles y el proceso manual de segmentación puede ser muy tardado o tedioso y cansado.

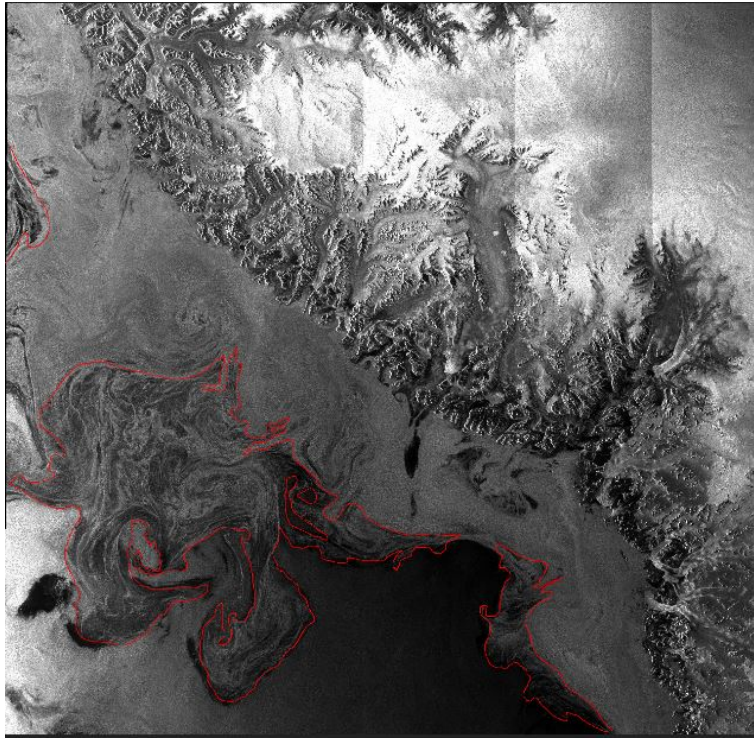


**Figura 1.1.** Imagen SAR con hielo Pancake

Existen varios métodos de segmentación de imágenes, entre ellos están superpíxeles, crecimiento de regiones, aquellos basados en bordes; los basados en histogramas, en los cuales se escogen valores llamados umbrales para separar los objetos, otros usan transformadas como la de Hermite.

## 1.1. Planteamiento del Problema

Las imágenes satelitales SAR tiene una dimensión de 10453 x 9996 pixeles, pero puede variar.



**Figura 1.2.** Imagen SAR. La segmentación a mano de la zona de hielo Pancake está en color rojo

Esto constituye un reto en el tiempo de procesamiento debido al tiempo de vida tan corto del hielo Pancake.

Conservar la cantidad de pixeles es útil para medir la extensión en metros cuadrados del área de estudio.

Se desea encontrar la zona del hielo pancake, discriminándola de la zona de mar, la zona de hielo y la zona terrestre.

La zona de estudio está conformada por hielo Pancake y Frazil. El contraste entre la tonalidad de ambos proporciona la apariencia grasosa peculiar de la zona. Se obtendrán características mediante su textura, utilizando Cuantización Vectorial y la Transformada Hermite, generando codebooks.

### 1.1.1. Objetivos

1. Caracterizar el hielo Pancake en imágenes SAR, a través de Transformada Hermite y Cuantización Vectorial.
2. Probar el algoritmo de Cuantización Vectorial para caracterizar la textura del hielo Pancake.
3. Utilizar el algoritmo de Cuantización Vectorial junto con Transformada Hermite, para generar un codebook de textura.

### 1.1.2. Organización de la tesis

La presente Tesis muestra en su capítulo 2 el estado del Arte, una idea general de la obtención de las imágenes satelitales de la misión SAR con los satélites Sentinel. La resolución que estos satélites presentan, así como las complicaciones concernientes a estas imágenes.

También se muestran las bases del procesamiento digital de imágenes, una vista a conceptos como: Vecindad, Textura, Agrupamiento, Segmentación. Y métodos de segmentación como: K- means, Cuantización Vectorial LBG, Umbrales, Superpíxeles, ya que estos son relevantes para el desarrollo del presente trabajo.

Se plantea la propuesta del sistema en el capítulo 3 proponiendo los procedimientos y las técnicas empleadas para probar la versatilidad del método de Cuantización Vectorial LBG (algoritmo Linde–Buzo–Gray) en imágenes y su combinación con la Transformada Hermite.

El capítulo 4 trata del análisis realizado a las imágenes en cada uno de los pasos descritos en la propuesta del sistema que nos permite caracterizar la zona de hielo Pancake. Encontrando límites entre zonas que podrían constituir diferente densidad en el agua de mar de esta zona.

Las conclusiones y trabajo futuro aparecen en el capítulo 5. El uso del algoritmo de Cuantización Vectorial LBG en imágenes se presenta en esta tesis. Algunas ideas en post-procesamiento en combinación con Transformada Hermite son planteadas para su desarrollo en un futuro.

Los apéndices contienen imágenes adicionales y código desarrollado para este trabajo.



## 2. Estado del Arte

### 2.1. Antecedentes

El Instituto de Ciencias de la Atmósfera y del Océano de Bolonia Italia, a través del investigador Flavio Parmaggiani, tiene colaboración con la Facultad de Ingeniería en su división de estudios de Posgrado en Ingeniería Eléctrica para la detección y segmentación de zonas con hielo Frazil-Pancake en imágenes SAR. Se busca un método de automatización de este proceso que es realizado manualmente y la cantidad de imágenes que deben ser analizadas sobre pasa el esfuerzo humano.

Se están realizando 2 trabajos al respecto: uno de maestría, cuya técnica se basa en la transformada Hermitiana y generación de Superpíxeles [26]. El otro trabajo es de doctorado cuya técnica se llama Edgeflow. Ésta permite detectar cambios en la textura y color en la imagen, permitiendo determinar donde están los bordes, la virtud de esta técnica es que solo debe definir un parámetro: la desviación estándar[17].

En el presente capítulo se presentan los conceptos de ruido Speckle, vecindad, segmentación, agrupamiento y textura. Se describen algoritmos como: Superpíxeles, Cuantización Vectorial LBG, K-means.

#### 2.1.1. Misión SAR, Satélite Sentinel, Resolución

La información de esta subsección fue consultada de:  
<https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/sentinel-1> (abril 2018)  
El satélite Sentinel-1 junto con el satélite sentinel-2 forman una constelación cuyo objetivo principal es el monitorear el océano y extensión terrestre. El objetivo principal es la adquisición de datos SAR de banda C.

***Entre sus objetivos están:***

- Observar el hielo marino y monitorear el movimiento de los icebergs.
- Producir cartas de hielo de alta resolución.
- Pronosticar condiciones de hielo en el mar.

***Resolución y ancho de pasada (cuatro modos):***

- Modo de mapa de banda: Swath de 80 km, resolución espacial de 5 x 5 m.
- Franja ancha interferométrica: franja de 250 km, resolución espacial de 5 x 20 m.
- Modo de franja extra ancha: Swath de 400 km, resolución espacial de 25 x 100 m.
- Modo de onda: 20 km x 20 km, resolución espacial de 5 x 20 m.

Las dimensiones de la nave son: 3.9 x 2.6 x 2.5 m con un peso (en el momento del lanzamiento) de 2300 kg.

Características de las imágenes SAR utilizadas son:

**Tabla 2.1.** Imágenes de alta resolución nivel 1 GRD [2]

Modo	Resolución rg x az	Espacio Píxel rg x az	Número de looks	ENL	Tamaño
EW	93x87 m	40x40 m	6x2	10.7	127Mb

***Imágenes empleadas.***

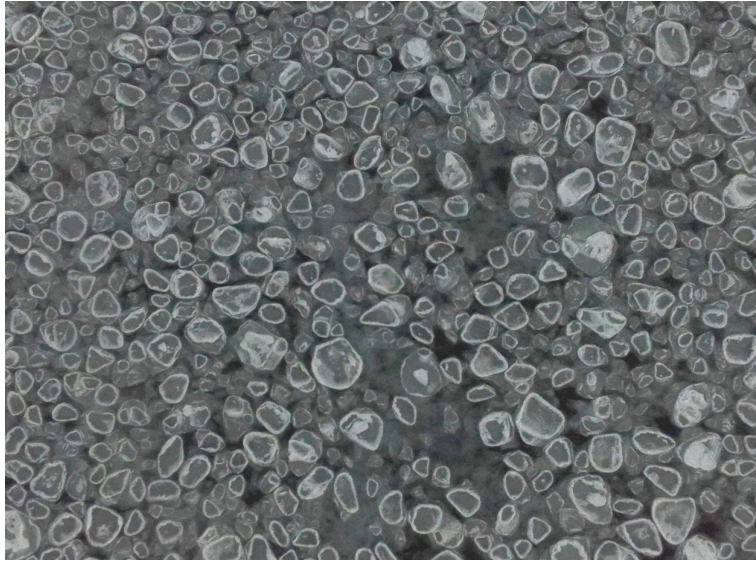
Se utilizaron 2 imágenes SAR con las siguientes dimensiones:

1. *Subset\_9E49* 10453x9996 pixeles nombrada **E9**.
2. *Subset\_D419* 5131x6556 pixeles nombrada **D419**.

**2.1.2. Hielo Pancake, zona de hielo Pancake**

Una de las razones por la cual se estudia el hielo Pancake, es por su estrecha relación con la estimulación del proceso de corrientes de convección del océano y la salinidad; lo cual está muy ligado al estudio de cambio climático [27]. En el polo Norte el hielo aumenta su tamaño de septiembre a marzo teniendo un mínimo aproximado de 8 kilómetros cuadrados a 15 millones de kilómetros cuadrados. Estos valores varían año con año. [11].

El proceso de formación del Pancake comienza con un diámetro de milímetros que se va agregando con otros pequeños cristales de hielo que se encuentran por la acción del viento y el movimiento ondular del océano. De esta manera comienza a crecer en diámetro y en grosor. Eventualmente pueden unirse varios pedazos de Pancake logrando una larga extensión de hielo flotante. El hielo Pancake tiene su forma predominantemente circular de 30 centímetros a 3 metros de diámetro y 10 centímetros de espesor [11].



**Figura 2.1.** Fotografía de hielo Pancake

La combinación de hielo Frazil-Pancake y la turbulencia del agua, genera una apariencia grasosa o lechosa al agua. La apariencia del hielo Frazil en las imágenes SAR es en tonos oscuros, mientras que para el hielo Pancake es en tonalidades claras.

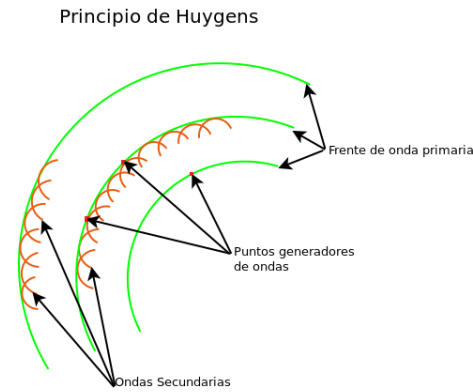
Estas son características de partida para analizar las imágenes de Satélite SAR de la misión Sentinel, en búsqueda de zonas con hielo Pancake [2].

### 2.1.3. Ruido Speckle

El ruido "Speckle" ocurre en todos los tipos de imágenes coherentes como son: las imágenes de apertura sintética de radar (SAR), imágenes acústicas, imágenes de iluminación por láser [12, 8].

Para comprender este fenómeno se toman los siguientes conceptos del libro "Introducción a la Geofísica" del Ingeniero Enrique del Valle Toledo [6].

- *Principio de Huygens.* Todo punto de frente de onda se comporta como un nuevo centro generador de ondas.
- *Principio de Fermat.* El movimiento ondulatorio entre dos puntos sigue la trayectoria de tiempo mínimo.
- *Frente de onda.* Es un lugar geométrico de todos los puntos que tienen el mismo estado de vibración o igual tiempo de viaje.



**Figura 2.2.** Se representa el principio de Huygens.

- En un medio homogéneo e isotrópico los desplazamientos de una onda serán proporcionales al incremento de tiempo y a la velocidad del medio, por lo tanto los nuevos frentes de onda serán paralelos.
- En medios no homogéneos o anisotrópicos los desplazamientos dependerán de la distribución de las velocidades de tal forma que los desplazamientos serán diferentes para iguales intervalos de tiempo, cambiando la forma de los frentes de onda subsecuentes y de las características de la trayectoria.
- En un punto pueden generarse varias trayectorias para el mismo movimiento, pero el primer indicio que se perciba en un punto de observación será el correspondiente al de trayectoria de tiempo mínimo.
- Las ondas elásticas en un punto pueden interpretarse como varios movimientos, pero que solo corresponden a diferentes trayectorias originadas en la misma fuente.

Esta explicación del comportamiento de las ondas en diversos medios nos ayuda a comprender la generación del ruido Speckle en imágenes SAR, tomando en cuenta que la atmósfera es un medio no homogéneo anisotrópico y estratificado.

La señal emitida por el satélite es dispersada en las diversas capas de la atmósfera, generando múltiples frentes de onda con trayectorias y velocidades diferentes que al ser detectadas por el sensor producen puntos blancos en zonas donde la interferencia ha sido altamente constructiva y puntos negros donde la interferencia ha sido altamente destructiva[8].

Algunas características de este ruido son:

- \* Es dependiente de la señal.
- \* Está espacialmente correlacionado.

La mejor forma de describirlo es a través de propiedades estadísticas, puesto que tiene un comportamiento Rayleigh[12].

El caso de un sistema de imagen ruidoso corresponde a un medio transmisor aleatoriamente variable.

## 2.2. Vecindad

Los pixeles que rodean directamente un píxel  $p$  de coordenadas  $(x,y)$  son definidos como vecinos de  $p$ . El conjunto de 4 pixeles que colindan horizontal y verticalmente con  $p$  se denominan vecindad de 4 ( $N_4[p]$ ) y están localizados en las coordenadas  $(x + 1, y)$ ,  $(x - 1, y)$ ,  $(x, y + 1)$ ,  $(x, y - 1)$  [19]



**Figura 2.3.** El diagrama muestra la vecindad 4 del píxel central

Cuando se toman adicionalmente los pixeles localizados diagonalmente a  $p$ , se habla de vecindad 8 ( $N_8[p]$ ). Las coordenadas de estos pixeles son:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y - 1), (x - 1, y + 1) [19]$$



**Figura 2.4.** El diagrama muestra la vecindad 8 del píxel central

## 2.3. Segmentación y Agrupamiento

Segmentar una imagen es dividirla o separarla en regiones con un atributo semejante. El atributo básico para una imagen monocromática es la amplitud de la luminosidad, es decir, la escala en grises. Mientras que en una imagen a color el atributo principal es el color. Otros atributos útiles son la textura y los bordes. Segmentar no implica clasificar, solamente divide

la imagen en segmentos[22].

Para Haralick [9] una buena segmentación debe ser uniforme y homogénea con respecto a la característica que se está empleando, como el tono de gris o la textura. Además, las regiones interiores deben ser simples y con pocos agujeros. Las regiones adyacentes deben contener valores diferentes con respecto al valor al cual son homogéneas. También recalca que no hay teoría en segmentación de imágenes, por lo cual las técnicas deben enfatizar una de las propiedades deseadas en lugar de otra.

### 2.3.1. Agrupamiento.

Agrupamiento es buscar un patrón que revele la organización de los datos, en grupos en los cuales se puedan encontrar características de similitud o de diferencias entre los patrones, con el objetivo de obtener conclusiones sobre estos grupos.[25]

Los patrones están representados en términos de características en forma de vectores de características unidimensionales.

Los pasos básicos que se deben seguir para desarrollar un agrupamiento son:

*Selección de la característica:*

- Medida de proximidad.
- Criterio de agrupamiento.
- Algoritmo de agrupamiento.
- Validación de resultados.
- Interpretación de los resultados.

*Categorías de algoritmos de agrupamiento:*

Un procedimiento de agrupamiento trata de identificar una característica específica por la cual puede juntar datos en el mismo grupo. Puede dividirse en las siguientes categorías:

- + Algoritmos secuenciales.
- + Algoritmos de agrupamiento jerárquico.
- + Algoritmos aglomerativos: por teoría de matrices o por teoría de grafos.
- + Algoritmos de división
- + Algoritmos basados en función de costos.
- + Algoritmos duros de agrupamiento.

Esta última categoría se describirá en la metodología debido a su importancia en este trabajo. A continuación, se menciona algunos puntos importantes.

El algoritmo de agrupamiento duro se basa en lo siguiente:

- Cada vector a clasificar pertenece a un único grupo en específico.
- La asignación se basa en un criterio óptimo especificado.

El algoritmo más famoso en esta categoría es el Isodata o K-means o algoritmo Lloyd[15] el cual es ampliamente utilizado en segmentación. El algoritmo LBG [13] se basa en el algoritmo anterior pero con modificaciones. El algoritmo LBG fue desarrollado para procesamiento de audio y de voz. En imágenes se ha empleado principalmente para codificación y compresión.

En la técnica de codificación el primer paso es descomponer la imagen en conjuntos de vectores. Después se forma un alfabeto que representa dichos vectores usando un algoritmo iterativo como el LBG (algoritmo Linde–Buzo–Gray). La cuantización es utilizada para encontrar el vector representativo más cercano del alfabeto al dato y etiquetarlo. La compresión de la imagen se encarga de enviar la correspondiente etiqueta. [28]

### 2.3.2. K-means vs. Cuantización Vectorial

Se realizó la siguiente tabla comparativa entre los dos métodos de agrupamiento duro.

<b>K-means</b>	<b>Cuantización Vectorial</b>
Inicia con valores aleatorios definidos.	Inicia con un centroide. (valor medio de los datos, que es perturbado.)
Utiliza distancia euclidiana para determinar los valores mas cercanos a cada dato representativo.	Utiliza una distancia para determinar los valores mas cercanos a cada centroide. En voz se utiliza la distancia de Itakura - Saito, para imágenes se utilizará la euclidiana.
Una vez determinados los grupos genera los nuevos centroides a partir de la media de cada grupo.	Una vez determinados los grupos genera los nuevos centroides a partir de la media de cada grupo.
Para determinar los nuevos grupos se utilizan solo los datos mas cercanos	Para determinar los nuevos grupos se utilizan todos los datos nuevamente.

**Tabla 2.2.** K-means vs Cuantización Vectorial (LBG)

### 2.3.3. Umbrales.

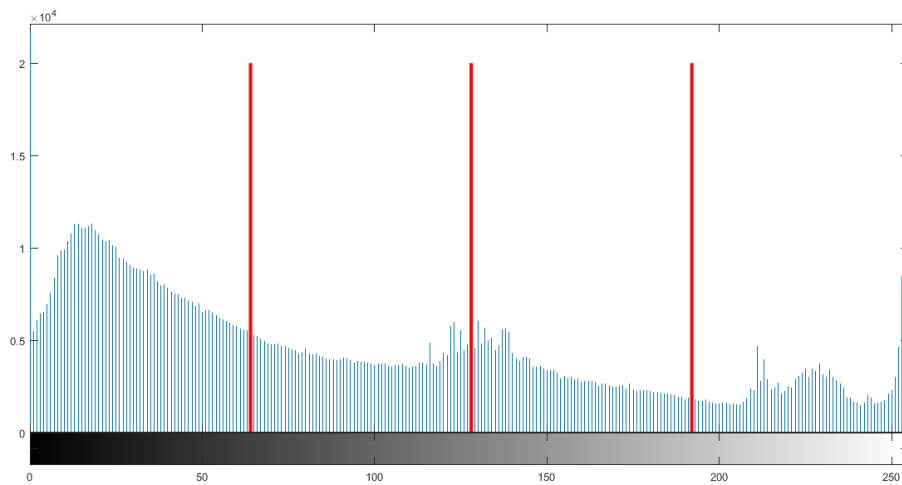
Los umbrales son usados en la detección de bordes. En este caso el umbral se escoge como la amplitud del nivel de contraste mínimo detectado como borde y aquellos que no son bordes [22].

Algunas imágenes pueden ser caracterizadas como portadoras de un objeto con cierto grado de brillo sobre un fondo con un grado de brillo completamente diferente [22]. Es decir, son aquellas imágenes en las que el objeto es blanco y el fondo es negro, en este tipo de imágenes los umbrales son de gran utilidad pues de una forma muy fácil se puede segmentar el objeto. Sin embargo, puede existir dificultad si la imagen contiene ruido o el fondo no es uniforme.

Existen varios métodos para establecer los umbrales, uno de ellos es el siguiente descrito por Petrou [20]:

Si la imagen presenta una escala de grises  $[Gmin, Gmax]$  podemos escoger  $M$  valores  $t_1, t_2, ..t_M$  de tal forma que  $Gmin < t_1 < ..t_M < Gmax$  y analizar la imagen en cada  $M + 1$  partes, agrupando en la primer parte todos los pixeles que pertenezcan al rango  $[Gmin, T_1)$ . En la segunda todos los pixeles cuyos valores de grises estén en el rango  $[t_1, t_2)$  y así sucesivamente, teniendo el número de partes que se elija y como máximo el número de niveles de gris presente en la imagen. Cuando los umbrales se escogen de forma aleatoria, puede ocurrir que alguna parte contenga muy pocos pixeles.

Una forma sencilla para obtener umbrales es:  $(Valormáximo - Valormínimo)/N$  donde  $N$  representa el número de partes que se desee. Por ejemplo, el siguiente histograma tiene un máximo de 255 y un mínimo de 0, al calcular 4 umbrales estarán asignados en: 64, 128, 192, 255. Las líneas rojas del siguiente histograma ilustran estas separaciones.



**Figura 2.5.** Histograma de una imagen. Las líneas rojas son umbrales



### 2.3.4. Superpíxeles

Radhakrishna et al [1] propusieron el algoritmo de Superpíxeles, el cual, agrupa los píxeles en regiones perceptualmente significativas, recomendado para reemplazar la estructura de rejilla de los píxeles. Sirven para capturar la redundancia de la imagen y facilitar el cálculo de características de la imagen.

Utiliza una adaptación del algoritmo de generación de superpíxeles "*k-means*" con 2 diferencias importantes:

1. Reduce el número de cálculos limitando la búsqueda de espacio a una región proporcional al tamaño del superpíxel
2. Se mide la distancia ponderada combinando la proximidad espacial y de color.

El algoritmo trabaja en imágenes a color con el espacio de color *CIELAB*. Como primer paso debe decidirse el número de regiones buscadas  $k$ . Su centro se colocará con los parámetros  $C_i = [l_i \ a_i \ b_i \ x_i \ y_i]^T$  distribuido en una rejilla espaciada cada  $S$  píxeles de separación. Se consigue generar superpíxeles más equitativos en tamaño con un intervalo de  $S = \sqrt{\frac{N}{k}}$  para la rejilla.

Posteriormente se mueve el centro a la ubicación de menor gradiente dentro de una vecindad de 3x3 píxeles. Se mejora la velocidad de cálculo con respecto al *k-means* convencional, haciendo que cada semilla busque los píxeles más cercanos a su vecindad que tiene una región de 2Sx2S alrededor del centro.

Se utiliza una distancia euclidiana con 5 dimensiones acorde al espacio *labxy* pero ocasiona lo siguiente:

Para superpíxeles grandes tiene mayor importancia la proximidad espacial que la de color provocando que no se acerque a los contornos.

Para superpíxeles pequeños sí se logra la convergencia.

Se normaliza tanto la distancia espacial como la de color, para poderla representar en una sola distancia.

Las fórmulas planteadas son:

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \quad (2.1)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (2.2)$$

$$D = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \quad (2.3)$$

Donde:

La ecuación 2.1 corresponde al espacio de color, la ecuación 2.2 a la distribución espacial y la 2.3 es la distancia que engloba ambas. Para normalizar tienen por un lado la distancia máxima espacial dentro de un grupo, corresponde al intervalo de muestreo de tal forma que  $N_s = S = \sqrt{\frac{N}{K}}$  en cambio para el color fijan  $N_c$  a una constante  $m$  entonces la ecuación 2.3 queda:

$$D_r = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \quad (2.4)$$

En práctica utilizan:

$$D_r = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (2.5)$$

Cuando  $m$  es muy grande la contribución espacial es más importante y genera superpíxeles más compactos, cuando  $m$  es pequeña los superpíxeles tienden a acercarse a los bordes, pero no son regulares en tamaño ni forma. Para el espacio de color de CIELAB recomiendan un rango de [1,40] para  $m$ .

Indican que la ecuación 2.5 puede adaptarse para escalas de grises haciendo  $d_c = \sqrt{(l_j - l_i)^2}$

## 2.4. Textura

Entre los investigadores de textura existe la dificultad para llegar a un consenso de definición de la textura, debido a que toman en cuenta los factores para caracterizarla o la forma de describirla. Esto es a través de patrones repetitivos, a través de estadísticos, tipo de textura, tipo de imagen [4, 3, 21].

Pratt[22] nos presenta un resumen de 3 autores que proporcionaron su definición:

1. Pickett [21]: La textura es usada para describir la variación de arreglos de dos dimensiones. Los elementos y reglas de espaciamiento o distribución pueden ser manipuladas arbitrariamente, proporcionando una permanente característica de repetitividad.
2. Hawkins [10]: La noción de textura parece depender de 3 ingredientes: 1) un orden local es repetido sobre una región, la cual es extensa en comparación al tamaño local, 2) el orden consiste en un arreglo no aleatorio de partes elementales, 3) las partes son entidades aproximadamente uniformes que tienen dimensiones semejantes en toda la región texturizada.

- Roy [22]: Presenta desde un punto de vista de visión que el enfoque principal es la variación de intensidad de una superficie en particular o una región en la imagen. Aún con esto surge la duda sobre si se está describiendo el objeto físico observado o la imagen derivada de ello. Es decir, puede ser la aspereza de la superficie o la estructura o composición del material lo que da lugar a sus propiedades visuales. La textura la definen como las características de variación de intensidades de una región en una imagen, con las cuales debería ser posible reconocerlo, describirlo y esbozar sus límites.

Petrou 2006, nos da una definición muy concisa de la textura:

*'Es la variación de los datos a una escala menor que la escala de interés'*

La textura puede ser el resultado de la variación de la forma de la superficie de la imagen, de la interacción de las sombras y las partes mejor iluminadas de un objeto. [20]

Aunque es muy concisa logra dar la esencia de la textura: dentro de la imagen existe un sector en el cual los datos tienen un cambio que no necesariamente cubre la totalidad de la imagen. Además, es el punto de coincidencia de los demás autores: "variación de los datos"; algunos lo presentan como estadístico, otros de intensidad, o en la variación de los patrones.

Existen 2 amplias categorías de textura: Regular e irregular. Las texturas realizadas por objetos hechos por el hombre son regulares mientras que las naturales son irregulares [20]. Cada una tiene sus técnicas para describirlas. Las texturas regulares se describen a través de patrones y las irregulares se utilizan descriptores estadísticos [20].



(a) Textura Regular

(b) Textura Irregular

**Figura 2.6.** Ejemplos de Textura

Dentro de las técnicas estadísticas se encuentran la matriz de coocurrencia y la Transformada Hermite que se abordará más adelante.

Petrou [20] nos define 2 tipos de imágenes en relación a la textura contenida:

1. *Imágenes de textura estacionaria en grises.* Es una imagen que contiene un único tipo de textura, la cual llena la imagen entera y sus propiedades estadísticas son las mismas en todos lados dentro de ella. Este tipo de imágenes son útiles para identificar texturas mediante sus características texturales, de tal forma que se puede generar un catálogo. Al llegar una nueva puede compararse y determinar a cuál tipo de textura pertenece.
2. *Imágenes de textura no estacionaria en grises.* Son imágenes que contienen más de un tipo de textura en ella. El problema de interés de este tipo de imágenes es la segmentación de texturas[20].

“En una región de una imagen la textura es determinada por la forma de distribución de los niveles de grises de cada píxel en dicha región” [25].

Para poder cuantificar esta distribución existen varios métodos[20] [5]:

- *Estadísticos:*
  - Características estadísticas de primer orden.
  - Características estadísticas de segundo orden.
  - Niveles de gris Run Lengths”.
- *Geométricos:*
  - Momentos Geométricos
  - Fractales
  - Código Cadena
- *Con transformación:*
  - Transformada Hermite
  - Transformada Gabor
  - Transformada Fourier.

Para caracterizar la textura con una transformada se utilizan las características estadísticas de primer orden, para generar vectores que representan dicha textura.

## 3. Propuesta del Sistema

Las imágenes pueden ser consideradas como cúmulos de datos distribuidos en 3 dimensiones, donde 2 dimensiones corresponden a la distribución espacial y la tercera corresponde al valor del píxel.

En general se analiza un píxel y su vecindad del vecino más cercano como en el método de superpíxeles el cual utiliza en su interior el algoritmo de K-means que se describe a continuación:

### 3.0.1. K-means

Pertenece a los algoritmos de agrupamiento duros, utiliza la distancia euclidiana como medida de distorsión entre los vectores  $x_i$  y las representaciones de agrupamiento  $\theta_j$  [14, 25, 15] .

$$J(\theta, U) = \sum_{i=1}^N \sum_{j=1}^m u_{ij} \|x_i - \theta_j\|^2 \quad (3.1)$$

donde  $u_{ij}$  tiene valor de 1 cuando la función de distorsión es el mínimo y cero para cualquier otro caso

El vector  $\theta_j$  es el valor medio para todos los valores contenidos en el agrupamiento. Utilizando esta distancia el algoritmo converge al mínimo de la función de costo. Esto provoca que los agrupamientos sean compactos.

El algoritmo consiste en lo siguiente:

1. Elegir de forma arbitraria los valores  $\theta_j$  para los  $\theta_j$  desde  $j = 1, \dots, m$ .
2. Repetir lo siguiente:
  - Hacer un ciclo desde  $i = 1$  hasta  $N$ .
  - Determinar los grupos representativos es decir  $\theta_j$  para  $x_i$ .
  - Hacer  $b(i) = j$ .
  - Fin del ciclo.
  - Hacer un ciclo desde  $j = 1$  hasta  $m$ .
  - Actualización del parámetro: determinar  $\theta_j$  es el valor medio de los vectores  $x_i \in X$  con  $b(i) = j$ .
  - Fin del ciclo.

Lo anterior se repite hasta que no haya variación de  $\theta_j$  en dos iteraciones consecutivas.

#### Ventajas:

- Simplicidad computacional: el tiempo es  $O(Nmq)$  donde  $q$  es el número de iteraciones requeridas para converger.

#### Desventajas:

- No puede garantizar la convergencia al mínimo global.
- Diferentes inicializaciones no llevan al mismo resultado.
- Es sensible al ruido y a un dato mal clasificado.

### 3.0.2. Cuantización Vectorial

En la sección 2.3.1 se menciona el parecido del algoritmo K-means con LGB. A continuación, se describe este algoritmo.

Una definición de cuantización vectorial es [18]: en un espacio euclidiano de  $k$  dimensiones  $R^l$ , se realiza un mapeo a través de la función de transferencia  $Q$  a un subgrupo finito  $Y$  del espacio  $R^l$ . Esto es:

$$Q : R^l \rightarrow Y \quad (3.2)$$

donde  $Y = (\mathbf{x}_i; i = 1, 2 \dots, N)$  es el grupo de vectores de reproducción y  $N$  es el número de vectores en  $Y$ . También puede verse como la combinación de 2 funciones, una que codifica los vectores de entrada  $\mathbf{x}$  y genera un código específico para volver a producir el vector con la función de transferencia  $Q(\mathbf{x})$ . Y un decodificador, que utiliza el código generado para reproducir el vector  $\hat{\mathbf{x}}$ . Utilizan una función de distorsión  $d(\mathbf{x}, \hat{\mathbf{x}})$  que es una penalidad para reproducir el vector  $\mathbf{x}$  en  $\hat{\mathbf{x}}$ , de tal forma que el mejor mapeo de  $Q$  es aquel que minimiza la función de distorsión  $d(\mathbf{x}, \hat{\mathbf{x}})$ . Una medida de distorsión es el error cuadrático:

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \sum_{i=0}^{K-1} (x_i - \hat{x}_i)^2 \quad (3.3)$$

Al subgrupo generado se le llama codebook. Contiene la información suficiente para representar la señal de donde se obtuvo. Existen varias formas de obtenerlo, a continuación, se desarrolla una de ellas que se adaptado en este trabajo para obtener codebook que otorguen características de la imagen estudiada.

El equipo de Linde, Buzo y Gray[13] en el estudio de compresión de señales de voz, utilizaron el algoritmo de Lloyd [14], mejor conocido como K-means, haciendo las siguientes modificaciones:

- El valor inicial es el valor promedio de todos los datos.

- Para hacer los subgrupos se utilizan todos los datos de entrenamiento y no solo los más cercanos.

Este algoritmo de cuantización vectorial es conocido como LBG, a continuación se describen todos los pasos:

1. Buscar el primer centroide obteniendo la media aritmética de todos los datos.

$$\mathbf{C}_0 = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (3.4)$$

Donde  $\mathbf{x}_i$  son los vectores de entrenamiento:

$$\begin{aligned} \mathbf{x}_1 &= [x_{11}, x_{12}, \dots, x_{1K}] \\ \mathbf{x}_2 &= [x_{21}, x_{22}, \dots, x_{2K}] \\ &\vdots \\ \mathbf{x}_N &= [x_{N1}, x_{N2}, \dots, x_{NK}] \end{aligned}$$

2. Perturbar este centroide 3.4 mediante la suma o multiplicación con un factor muy pequeño  $\varepsilon$ :

$$C_1 = C_0 + \varepsilon \quad (3.5)$$

$$C_2 = C_0 - \varepsilon \quad (3.6)$$

3. Dividimos el espacio en 2 regiones con los puntos más cercanos a cada uno de los dos centroides, es decir, agrupamos los vectores de entrenamiento en 2 grupos, nombrados con la letra G, de acuerdo con el centroide más cercano:

$$\begin{aligned} d_1 &= \text{distancia}(\mathbf{x}_i, \mathbf{C}_{j+1}) \\ d_2 &= \text{distancia}(\mathbf{x}_i, \mathbf{C}_{j+2}) \\ & i = 1, \dots, N \end{aligned}$$

$j=0$ , hasta número de centroides definido por el usuario en potencia de 2.

- Si la distancia  $d_1$  es menor que la distancia  $d_2$ :

$$d_1 < d_2 \quad \mathbf{x}_i \longrightarrow G_{j+1}$$

Se asigna al grupo  $G_{j+1}$

- En caso contrario:

$$\mathbf{x}_i \longrightarrow G_{j+2}$$

Se asigna al grupo  $G_{j+2}$

4. Se vuelve a obtener un centroide para cada una de las regiones con los datos pertenecientes a cada región, esto les provoca un cambio de ubicación.

Con estos nuevos centroides, se vuelve a generar las dos regiones con los datos más cercanos. Este paso se hace repetidas veces hasta que el centroide no cambie de valor, o la variación se encuentre en el rango aceptable que fue definido por el usuario.

Matemáticamente sería:

- Si la distancia global acumulada  $D_g^z$  menos la distancia global de la nueva ubicación  $D_g^{z+1}$  es mayor al rango aceptable:

$$| D_g^z - D_g^{z+1} | \geq \varepsilon \quad (3.7)$$

recalcular los centroides:

$$C_{0+1} = \frac{1}{NG_{j+1}} \sum_{i=1}^{NG_{j+1}} \mathbf{x}_i^{j+1}$$

$$C_{0+2} = \frac{1}{NG_{j+2}} \sum_{i=1}^{NG_{j+2}} \mathbf{x}_i^{j+2}$$

donde  $NG_{j+1}$  representa el número total de vectores del grupo  $G_{j+1}$ . De forma semejante para  $G_{j+2}$

- ir al punto 2.
  - En caso de no cumplir la condición:
    - ir al punto 3.
5. Una vez que los centroides no cambian de lugar, se vuelve a perturbar cada uno con el factor  $\varepsilon$ .

Este proceso se repite hasta obtener el número de centroides deseado.

### 3.1. Uso del Algoritmo Cuantización Vectorial

Si se consideran los datos sin su distribución espacial, podrían analizarse por la respuesta que tienen al sensor de la cámara, con la idea principal de que aquellos con características semejantes tendrán respuesta parecida al sensor, formando de manera natural su vecindad y límites. Pueden generarse varias vecindades distribuidas aleatoriamente dentro del espacio de la imagen, que tengan la misma respuesta al sensor.

El número de centroides que se propone debe corresponder al número de patrones que se considera existen en los datos [25].



En imágenes ha sido utilizado principalmente para compresión y en forma vectorial [18]. Se hace énfasis en la forma en que se utiliza puesto que en el presente trabajo se busca la versatilidad de este algoritmo utilizándolo con vectores 1 dimensional (escalar) o vectores con mayor dimensión.

El algoritmo trabaja en potencias de 2, por lo cual el número de centroides debe ajustarse a una potencia de este número, es decir, 2, 4, 8, 16, 32... Para el análisis de una imagen se propone el uso de 4 u 8 centroides dependiendo de la complejidad de la imagen que se esté analizando.

En el presente trabajo se propone utilizar el algoritmo LBG pero en formas diferentes:

1. Uso del algoritmo LBG en forma unidimensional (CV1D) junto con los valores de píxel para segmentar.
2. Con la segmentación CV1D generar regiones de textura homogénea.
3. Uso de Cuantización Vectorial junto con vector de coeficientes de Transformada Hermite, para generar un codebook.

Estos puntos se describen en la siguiente sección.

## 3.2. Metodología

### 3.2.1. Uso de CV1D como segmentador.

Se ocupa el algoritmo LBG modificando únicamente el dato de inicio, en lugar de ser un vector, será el valor del píxel, para ello la distancia utilizada entre el centroide y el valor de entrenamiento es el valor absoluto  $|x - x'|$ . Una vez determinado el grupo al que pertenece, se coloca en la posición  $(x, y)$  que le corresponde dentro de la imagen. Este valor en Superpíxeles puede lograrse si en la ecuación 2.5 el peso  $m$  tiende a cero. Se compara el desempeño con algoritmos como Superpíxeles sin modificarlo y con Umbrales.

### 3.2.2. Regiones de Textura Homogénea

En la sección Textura 2.4 se presentaron varias formas para analizar la textura estacionaria o semi-estacionaria, para ello se debe analizar la variación en los detalles. Varios métodos utilizan una segmentación y los estadísticos de primer y/o de segundo orden. Con base en esta idea se propone utilizar en una ventana de análisis, en la cual se realizará la segmentación con CV1D para generar regiones de respuesta semejante al sensor. Esto proporciona un valor mínimo, máximo y el centroide de la región. A estos 3 parámetros se añaden los estadísticos de primer orden de cada región para formar un conjunto de vectores que describan la textura presente en la ventana de análisis. El número de vectores dependerá del número

de centroides que se decida utilizar en la imagen.

A continuación, se presenta las características estadísticas de primer orden.

### Características estadísticas de primer orden

La variable aleatoria  $I$  representa los niveles de gris de una región de interés [20]. El histograma de primer orden  $P(I)$  está definido como:

$$P(I) = \frac{\text{número de píxeles en el nivel}}{\text{total de número de píxeles en la región.}} \quad (3.8)$$

$P(I)$  representa la fracción de píxeles con el nivel de gris  $I$ .

Definamos los siguientes conceptos:

Momentos:

$$m_i = E[I^i] = \sum_{I=0}^{N_g-1} I^i P(I) \quad (3.9)$$

$i=1, 2, 3 \dots$

donde  $m_0 = 1$  y  $m_1 = E[I]$  es el valor medio de  $I$ .

Momentos Centrales:

$$\mu_i = E[(I - E[I])^i] = \sum_{I=0}^{N_g-1} (I - m_1)^i P(I) \quad (3.10)$$

Los momentos centrales más usados son:  $\mu_2, \mu_3, \mu_4$ . donde  $\mu_2 = \sigma_2$  es la varianza;  $\mu_3$  es skewness y  $\mu_4$  es la curtosis del histograma. La varianza es la medida del ancho del histograma, es decir, la medida en cómo los datos varían desde la media. Skewness es la medida del grado de asimetría respecto a la media; y la curtosis es la medida de la forma del histograma, la cual puede ser: platykurtica, leptokurtica y mesokurtica.

### 3.2.3. Transformada Hermite con Cuantización Vectorial Forma Vectorial

En esta sección se propone el uso de Cuantización Vectorial junto con la Transformada Hermite en forma vectorial. Comencemos con la técnica de la transformada Hermite.

#### Transformada Hermite.

La Transformada Hermite fue desarrollada por Martens [16], contribuciones en las aplicaciones de esta técnica han sido plasmadas en trabajos de Silvan [24] y Escalante [7]. A continuación, se presenta la teoría de esta técnica.

Un modelo de descripción de la imagen es generado por la Transformada Hermite que es un caso especial de la transformada polinomial[7].

El proceso de esta transformada es el siguiente:

Primero se separa en ventanas con una función local  $\omega(x, y)$  que toma lugar en varias posiciones sobre la imagen de entrada.

Después la información local de cada ventana de análisis es expandida en términos de una familia de polinomios ortogonales  $G_{m,n-m}(x, y)$  y satisfacen la condición ortogonal.

$$\int_{-\infty}^{\infty} \omega^2(x, y) G_{m,n-m}(x, y) G_{l,k-l}(x, y) dx dy = \delta_{nk} \delta_{nl} \quad (3.11)$$

para  $n, k=0, \dots, \infty$ ;  $m=0, \dots, n$ ;  $l=0, \dots, k$   
donde  $\delta_{nk}$  denota la función Kronecker.

La función de ventana gaussiana se asemeja al campo receptivo de la visión humana, la ecuación que la representa es la siguiente:

$$\omega(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (3.12)$$

y los polinomios ortogonales asociados son los polinomios de Hermite:

$$G_{n-m}(x, y) = \frac{1}{\sqrt{2^n (n-m)! m!}} H_{n-m}\left(\frac{x}{\sigma}\right) H_m\left(\frac{y}{\sigma}\right) \quad (3.13)$$

donde  $H_n(x)$  denota el  $n$ -ésimo polinomio de Hermite.

A partir del producto de la señal original  $L(x, y)$  con función de ventana  $\omega(x-p, y-q)$ , se genera una función periódica ponderada definida como:

$$W(x, y) = \sum_{(p,q) \in S} \omega(x-p, y-q) \quad (3.14)$$

Esta función debe ser diferente de cero para todos los pares de coordenadas  $L(x, y)$  entonces queda:

$$L(x, y) = \frac{1}{W(i, j)} \sum_{(p, q) \in S} L(x, y) \omega(x - p, y - q) \quad (3.15)$$

El contenido de la señal en cada función de ventaneo es descrito como la suma ponderada de los polinomios  $G_{m, n-m}(x, y)$  de grado  $m$  en  $x$  y de grado  $n-m$  en  $y$ . En una implementación discreta la función de ventana gaussiana puede aproximarse a través de funciones de ventaneo binomial y en ese caso estas funciones ortogonales se conocen como polinomios de Krawtchouck.

En ambos casos los coeficientes de los polinomios  $L_{m, n-m}(p, q)$  son calculados por la convolución de la imagen original  $L(x, y)$  con la función de filtro  $D_{m, n-m}(x, y) = G_{n-m}(-x, -y) \omega^2(-x, -y)$  seguido por un submuestreo en las posiciones  $(p, q)$  del muestreo lattice  $S$ .

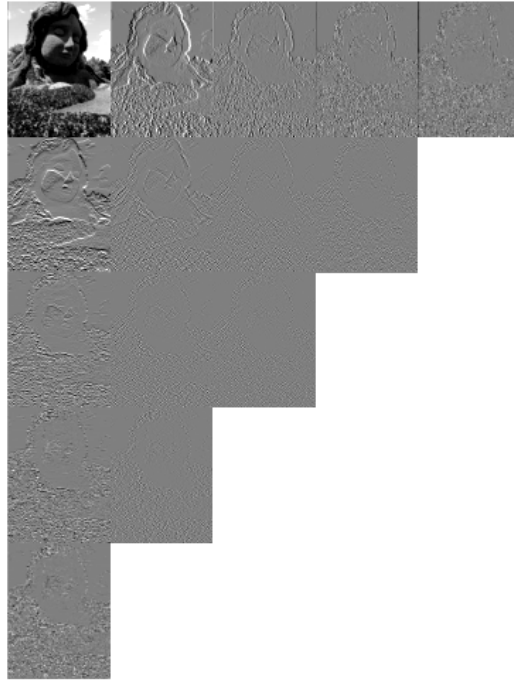
$$L_{m, n-m}(p, q) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} L(x, y) D_{m, n-m}(x, y) dx dy \quad (3.16)$$

A continuación, se muestra una imagen de la diosa Gaia de Mosai Culture del Parc Jacques-Cartier, Gatineau, Ontario, Canadá.



**Figura 3.1.** Imagen de la diosa Gaia

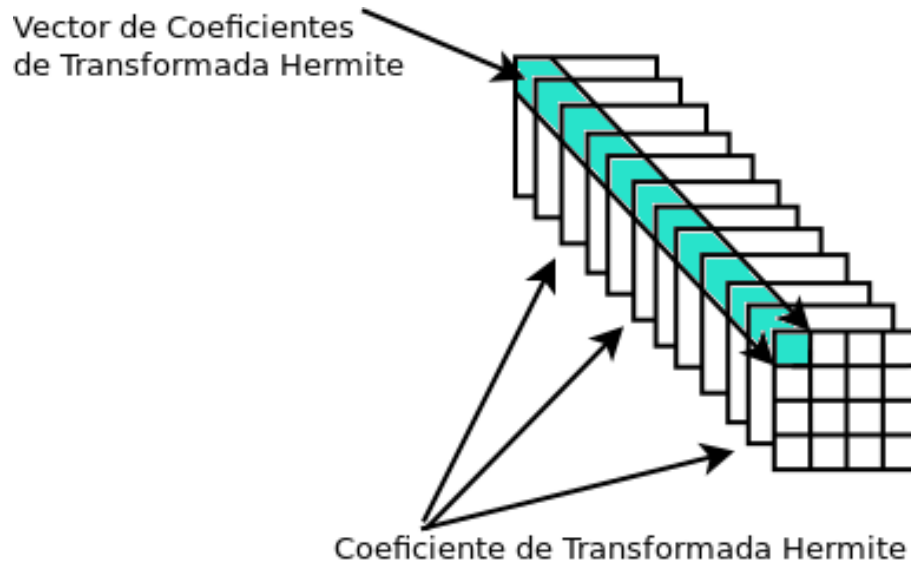
Y la Transformada Hermite de la imagen anterior:



**Figura 3.2.** Transformada Hermite de la imagen de la Diosa Gaia

En la figura **3.2** se observa la Transformada Hermite de la imagen **3.1** se utilizaron los parámetros  $N=40$ ,  $D=4$ ,  $T=2$ ; el resultado da 15 coeficientes, los de mayor detalle son aquellos que están en la primer fila y los de la primer columna.

La Transformada Hermite es un conjunto de coeficientes que puede ser considerado como vectores característicos de la textura. El siguiente diagrama representa la forma en que se forman dichos vectores.



**Figura 3.3.** Representación de los vectores de coeficientes de Transformada Hermite.

Tomando de esta forma a la Transformada Hermite se le puede aplicar el algoritmo LGB en forma vectorial y generar un codebook para representar las diferentes texturas que tenga una imagen.

En este caso vectorial la medida de distorsión toma la siguiente forma:  
Para el centroide 1:

$$d_1(\mathbf{x}_j, \hat{\mathbf{x}}_1) = \sqrt{[(x_1 - \hat{x}_1)^2 + (x_2 - \hat{x}_1)^2 + \cdots + (x_j - \hat{x}_1)^2]} \quad (3.17)$$

Para el centroide 2:

$$d_2(\mathbf{x}_j, \hat{\mathbf{x}}_2) = \sqrt{[(x_1 - \hat{x}_2)^2 + (x_2 - \hat{x}_2)^2 + \cdots + (x_j - \hat{x}_2)^2]} \quad (3.18)$$

Y así sucesivamente hasta el centroide n:

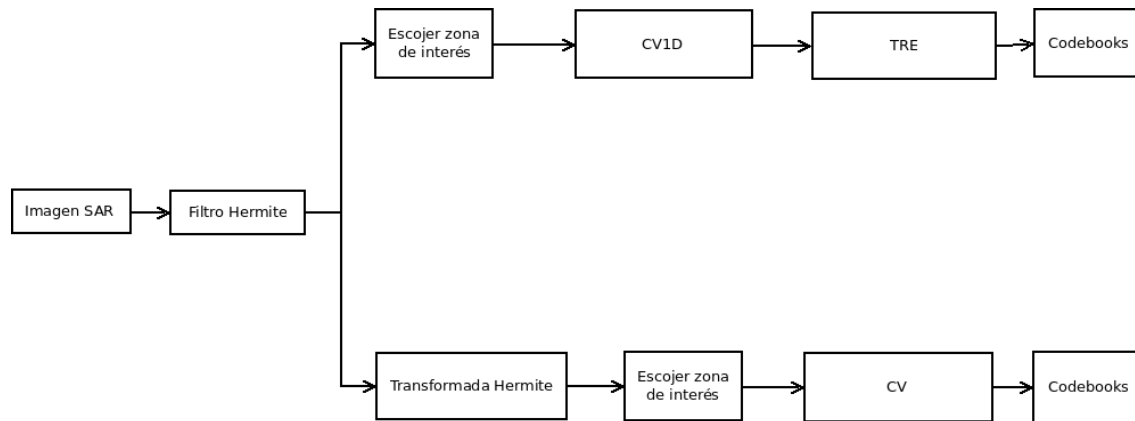
$$d_n(\mathbf{x}_j, \hat{\mathbf{x}}_n) = \sqrt{[(x_1 - \hat{x}_n)^2 + (x_2 - \hat{x}_n)^2 + \cdots + (x_j - \hat{x}_n)^2]} \quad (3.19)$$

Donde  $\hat{x}$  denota centroide.

El vector será asignado al grupo cuya distancia haya sido la menor.

### 3.2.4. Metodología para imágenes SAR

Hasta ahora se han descrito metodologías en forma aislada para poder desarrollar su contexto matemático. En el siguiente Diagrama se presentan 2 procedimientos que pueden realizarse:



**Figura 3.4.** Diagrama de procesos con Cuantización Vectorial

En la figura 3.4 puede observarse que la imagen SAR tiene un preprocesamiento con un filtro Hermite, que está basado en la Transformada Hermite que es un caso especial de transformada polinomial, este filtro es desarrollado en [7] y usado en [26], su objetivo es la disminución del ruido speckle. Posteriormente puede hacerse 2 procedimientos diferentes:

1. La parte superior:

- \* Se escoge la zona de interés y se aplica CV1D para dividir la imagen en regiones de respuesta semejante al sensor.
- \* Y posteriormente se obtienen sus estadísticos de primer orden para tener la textura por regiones estadísticas y formar el codebook.

2. La parte inferior:

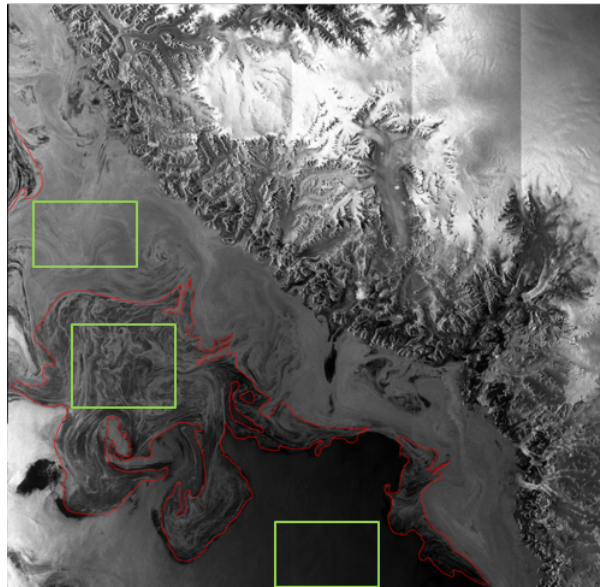
- \* Se utiliza la Transformada Hermite en toda la imagen.
- \* Se escoge la zona de interés y se utiliza CV con ellos para generar un codebook.

*Nota:* es necesario generar un codebook para cada zona a separar.

El tamaño de la imagen satelital SAR E9 permite generar varias subimágenes de las diferentes zonas a analizar: la zona de hielo, zona de hielo Pancake, la zona de mar.

En cada una de estas zonas se obtuvieron 7 subimágenes de 512 x 512 píxeles, que su contenido se aproxime a una imagen de textura estacionaria o semi-estacionaria, de tal forma que la textura de una zona puede ser analizada en forma individual.

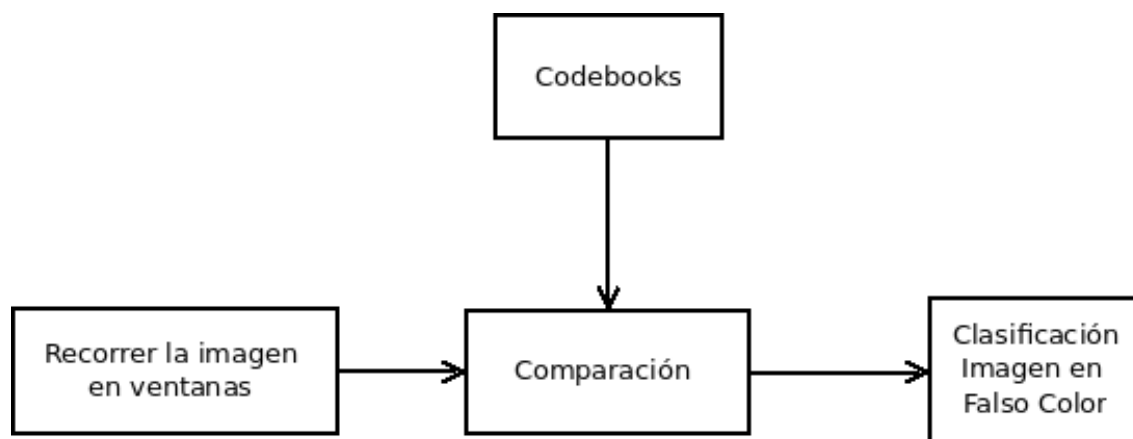
La siguiente imagen muestra en contornos verdes las zonas de donde se extrajeron las subimágenes:



**Figura 3.5.** Se muestra la imagen principal de análisis. La línea roja contiene la segmentación a mano del experto en pancake. Los recuadros verdes son las zonas de subimágenes

El procesamiento se puede consultar en la sección 4.5 y en el anexo A.

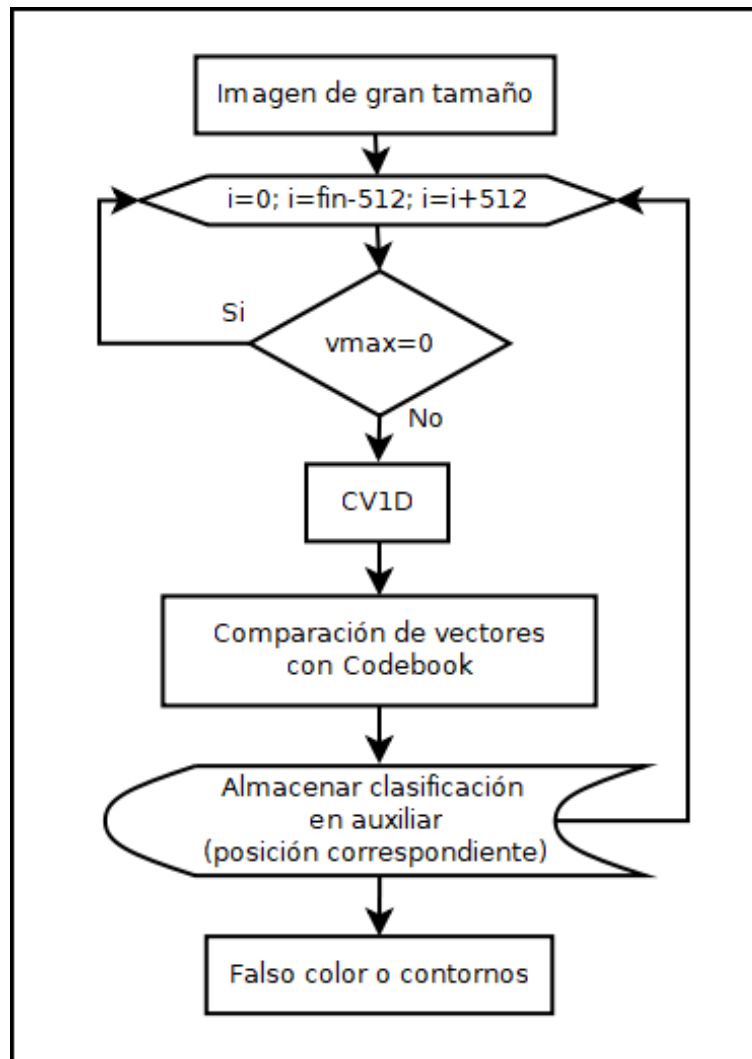
Con una imagen diferente a aquella donde se generaron los codebooks se recorrerá en ventanas de 512 x 512 píxeles, se obtendrá su vector de características de textura con regiones de textura homogéneas o con la transformada Hermite y se comparará con los codebooks de la otra imagen para determinar a qué tipo de zona pertenece. Esto se explica en el siguiente esquema en forma general:



**Figura 3.6.** Diagrama Clasificación de la Imagen



En forma específica para texturas por regiones homogéneas se generó un algoritmo que recorre la imagen en ventanas de 512 x 512 píxeles de la siguiente manera: toma la primera ventana se aplica CV1D para 4 centroides generando las regiones homogéneas de las que se obtienen los vectores estadísticos de textura. Estos se comparan con cada uno de los vectores estadísticas de textura del codebook y se determina a cual región pertenece por medio de la distancia euclidiana. Al terminar este proceso se avanza a la siguiente ventana de 512 x 512 píxeles, y así sucesivamente hasta terminar la imagen. Si el algoritmo encuentra una ventana que contenga puros ceros (máscara de tierra) le colocará valores de 1 (para que salga en blanco esa sección) y continúa a otra ventana. La siguiente figura 3.7 muestra el algoritmo descrito anteriormente.



**Figura 3.7.** Diagrama de flujo de selección de ventanas para clasificar por TRE.

# 4. Pruebas y Resultados

Las pruebas realizadas en esta sección fueron realizadas en una computadora Dell con las siguientes características:

Procesador

Intel(R) Xeon(R) CPU E%-2620 v4 2.10GHz 2.10GHz

Memoria RAM 16 Gb

Tipo de Sistema: sistema operativo de 64 bits, procesador x64

Software

Windows 10 Pro version 1709

## 4.1. Pruebas propuestas

### 4.1.1. Para su uso como segmentador

Se proponen las siguientes pruebas:

1. Con la finalidad de probar el algoritmo de Cuantización Vectorial unidimensional (CV1D) para segmentar y compararlo con la segmentación de superpíxeles que en su interior maneja el algoritmo de k-means que es muy semejante, se utilizará la imagen del perro Kobi que utiliza el algoritmo de superpíxeles de Matlab. A lo largo de este capítulo para el algoritmo de superpíxeles se utilizó el comando proporcionado en el software Matlab, se sugiere al lector tener en cuenta que al hablar de los superpíxeles fueron generados por dicho software y puede consultarse mayor detalle en la página web de Matlab.
2. Se utiliza una fotografía de hielo Pancake aplicando la comparación del punto anterior.
3. El método CV1D de segmentación se compara con el método de umbrales. Esto se hace utilizando una ventana seleccionada que contiene una zona de hielo Pancake.

## 4.2. Cuantización Vectorial para segmentar.(Superpíxeles vs Cuantización Vectorial)

En esta sección se desarrolla el uso del algoritmo de Cuantización Vectorial con vectores de un elemento, es decir, unidimensional, como segmentador en una imagen. Para ello se utilizan la imagen del perro Kobi en dos versiones y una imagen de hielos Pancakes. Se realiza la comparación con el algoritmo de superpíxeles que ha sido empleado para segmentar hielo Pancake[26].

### 4.2.1. Imagen Kobi

En el capítulo 2 se desarrolla la comparación entre el algoritmo k-means y Cuantización Vectorial. En esta sección se desarrolla una prueba entre el algoritmo de superpíxeles que utiliza el algoritmo de k-means en su interior y el algoritmo de Cuantización Vectorial utilizado en forma unidimensional (CV1D) para segmentar, por lo cual se le pide al lector que en esta sección tome en cuenta que al hablar de Cuantización Vectorial se hace referencia al algoritmo LBG pero utilizado en forma unidimensional. En otra prueba se utilizará en forma vectorial.

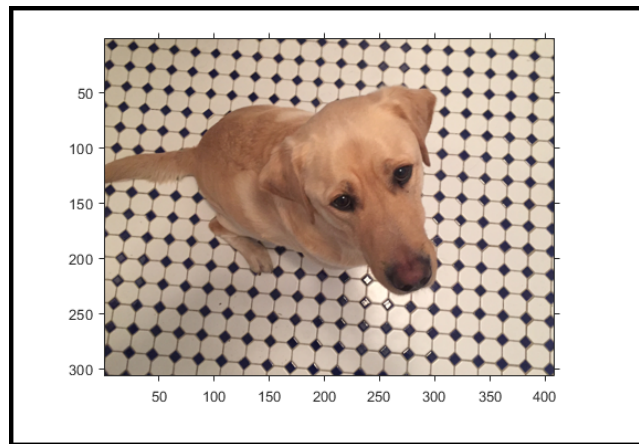
Es necesario mencionar que el algoritmo de superpíxeles trabaja mejor en imágenes a color, pero también puede ser empleado en imágenes en gris. Por su parte Cuantización Vectorial trabaja en escala de grises.

La imagen de Kobi es utilizada en el ejemplo de superpíxeles a color de Matlab. En estas pruebas la imagen de Kobi tiene un contorno blanco y ejes vertical y horizontal 4.1(b), en otra modalidad se utiliza la imagen sin el contorno blanco ni ejes y en escala de grises 4.1(a).

Esta imagen tiene pocos objetos en ella: un perro, mosaicos blancos y mosaicos negros, la respuesta hacia el sensor no tiene mucha complejidad, es decir se pueden identificar 3 o 4 respuestas diferentes. Tomando en cuenta lo anterior se utiliza CV1D en la imagen 4.1(a).



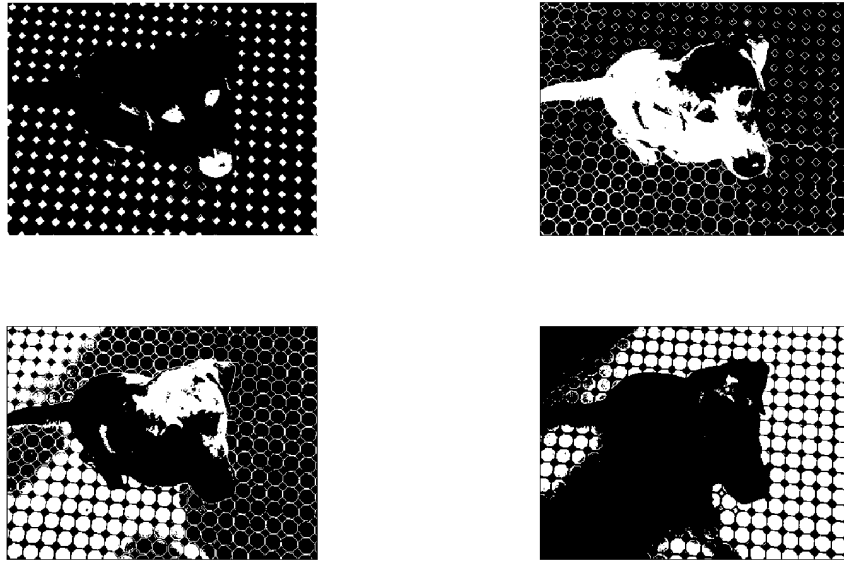
(a) Imagen de Kobi en escala de grises



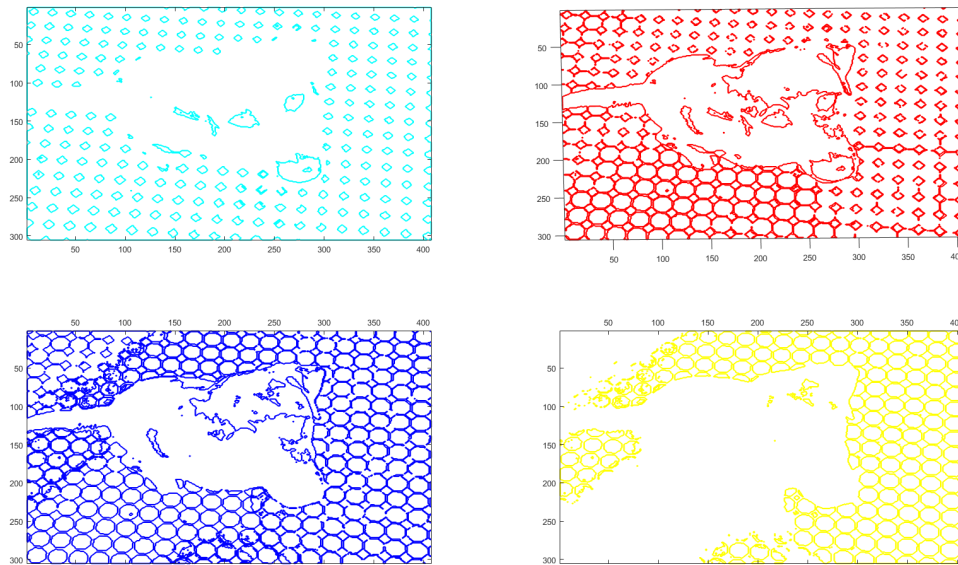
(b) Imagen de Kobi a color tiene un margen blanco al rededor de la imagen

**Figura 4.1.** Se muestran las imágenes utilizadas para este experimento a) Kobi en grises y b) Kobi a color

Para mostrar la separación de las regiones, se genera una imagen binaria, donde los números unos corresponden a los píxeles que pertenecen al centroide y los números ceros son píxeles que pertenecen a otro centroide. Estas imágenes binarias son en blanco (los unos) y negro (los ceros); comúnmente son nombradas máscaras. A continuación, se muestran las máscaras generadas de la separación con CV1D.



(a) Se muestra las máscaras generadas por Cuantización Vectorial



(b) Se muestra los contornos generados

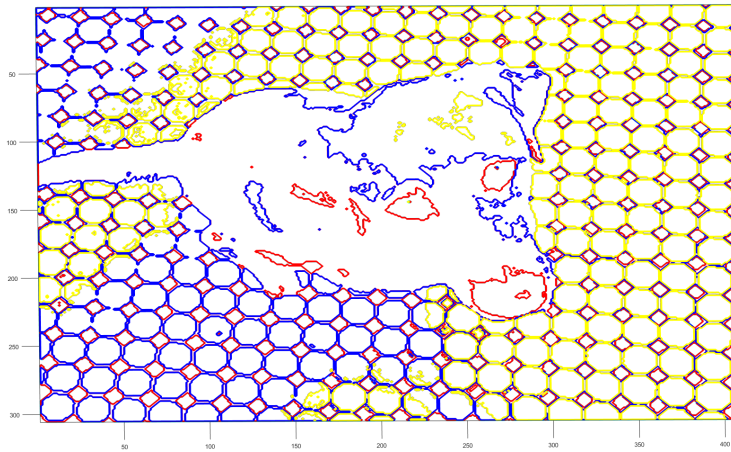
**Figura 4.2.** Máscaras a) Imágenes binarias de las máscaras de los 4 centroides generados y b) Contornos generados a partir de las máscaras.

En la figura 4.2(a) se presentan imágenes binarias de cada centroide, como puede observarse en la máscara superior izquierda se obtuvieron los mosaicos negros, los ojos y nariz del perro Kobi. En la máscara superior derecha se obtiene la mayor parte del cuerpo de Kobi, y las líneas de división del mosaico. En la máscara ubicada en la esquina inferior izquierda se captura la parte del mosaico que tiene sombras, una parte de la cabeza y cuerpo de Kobi que tiene diferente respuesta al sensor que el resto del cuerpo. Finalmente, en la esquina inferior derecha se obtuvo la mayor parte de los mosaicos blancos.

Estas máscaras pueden ser utilizadas para obtener contornos dentro de la imagen. En la figura 4.2(b) se muestran los contornos obtenidos con las máscaras anteriormente descritas.

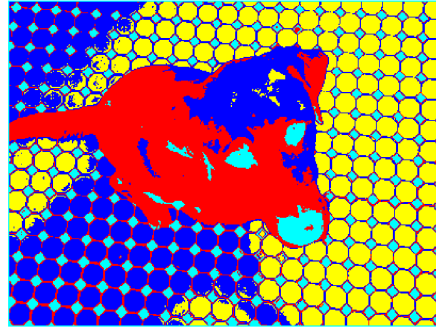
Esto es de utilidad ya que se podría obtener la segmentación de un cuerpo presente en la imagen, con un poco de morfología matemática, con las máscaras de la esquina superior derecha y la esquina inferior izquierda podría obtenerse el contorno de Kobi. O en la máscara de la esquina superior izquierda podría obtenerse fácilmente el contorno de los ojos y nariz; o en el caso en que los mosaicos negros fuesen el objeto de interés su obtención sería muy sencilla.

En la siguiente figura 4.3 se muestran todos los contornos juntos.

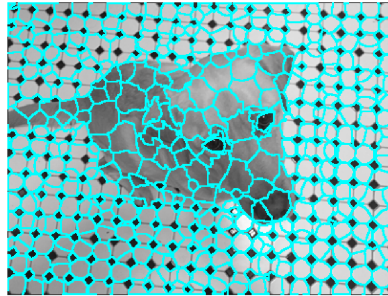


**Figura 4.3.** Se muestran los contornos generados por las máscaras

A continuación se muestra una imagen en falso color generada a partir de las máscaras obtenidas con CV1D: 4.4(a), y los superpíxeles obtenidos. 4.4(b). Con ambos algoritmos es necesario hacer un procesamiento posterior para aislar a Kobi, sin embargo, el algoritmo de CV1D puede apreciarse más fácilmente el patrón de Kobi.



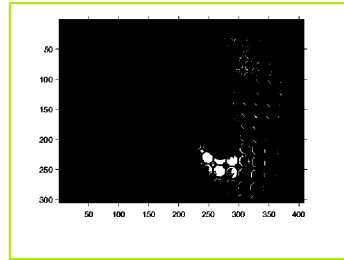
(a) Se muestra las regiones generadas por Cuantización Vectorial



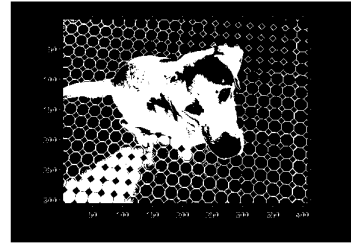
(b) Los superpíxeles generados con el comando de Matlab

**Figura 4.4.** Comparación CV1D vs superpíxeles a) Regiones generadas por Cuantización Vectorial unidimensional b) superpíxeles generados.

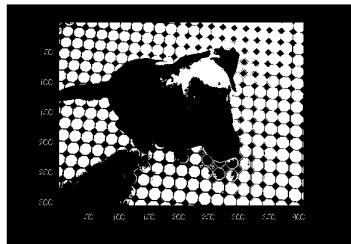
Para aumentar el desafío a ambos algoritmos se usa la imagen de Kobi con una modalidad: se agrega un contorno blanco alrededor de la imagen, cuenta con ejes coordenados y números. Veamos las máscaras generadas:



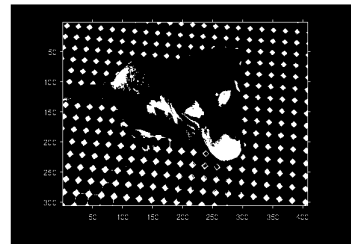
(a) Máscara centroide 1



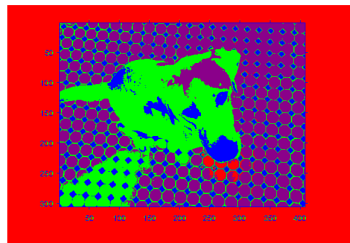
(b) Máscara centroide 2



(c) Máscara centroide 3



(d) Máscara centroide 4



(e) Se muestra las regiones generadas por Cuantización Vectorial.

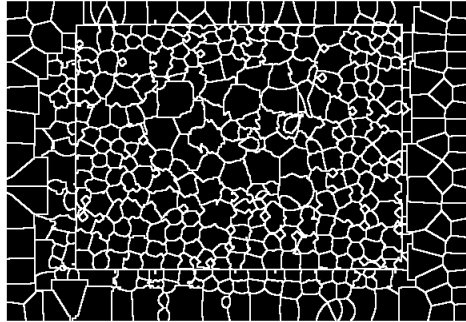
**Figura 4.5.** Se muestran las máscaras generadas por CV1D y una imagen en falso color.

La máscara correspondiente al centroide 1 es la figura 4.5(a) se ha agregado un contorno verde exclusivamente para apreciar los límites de la imagen, debido a que contiene el contorno blanco, estos píxeles están en valor 1 y con el fondo de la imagen en blanco se pierde. En la figura 4.5(e) se colocó en color rojo. La máscara del centroide 2 en figura 4.5(b) se aprecia la mayor parte del cuerpo del perro Kobi, también algunos mosaicos que están bajo la sombra del perro, en verde en figura 4.5(e). En el centroide 3, figura 4.5(c), han quedado los mosaicos blancos y una parte de la cabeza de Kobi: la más brillante, color morado en figura

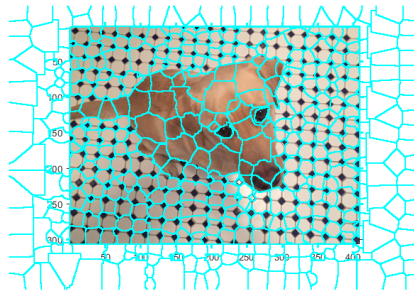


4.5(e). Finalmente en la máscara del centroide 4, figura 4.5(d) están los mosaicos negros, la nariz y ojos del perro, esto en azul en figura 4.5(e).

Por su parte los superpíxeles generaron los siguientes resultados:



(a) Mascara Superpíxeles



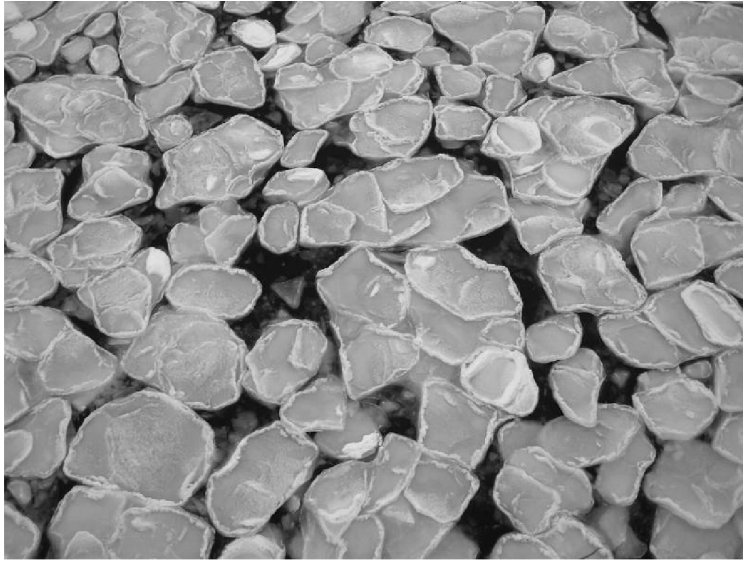
(b) Se muestran los superpíxeles generados.

**Figura 4.6.** Al agregar un contorno blanco a la imagen Cuantización Vectorial tiene mejor resultado que superpíxeles.

La máscara de superpíxeles figura 4.6(a) presenta la línea de contorno de cada uno de los 500 superpíxeles solicitados al algoritmo, exactamente igual a como se presenta en la página de Matlab para el ejemplo del algoritmo de superpíxeles. Se logra apreciar un cuadrado, los demás superpíxeles son irregulares y solo cuando al poner de fondo la imagen original, figura 4.6(b), es cuando se pudo identificar que el marco blanco queda contenido en numerosos superpíxeles hacia el exterior del cuadrado, hacia el interior del cuadrado los superpíxeles son más pequeños fuera del perro y más grandes en el cuerpo, se aprecia que cada ojo está contenido en un superpíxel, hay superpíxeles que contienen tanto mosaicos negros y blancos al mismo tiempo, algunos mosaicos blancos tiene su propio superpíxel pero no está bien definido, no es capaz de separar los números que están junto a los ejes.

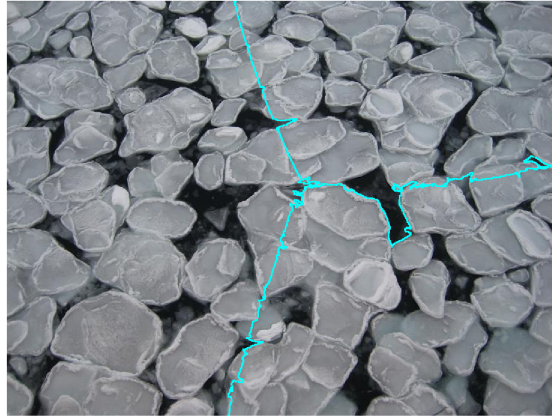
### 4.2.2. Fotografía hielo Pancake

Las pruebas de la sección anterior fueron realizadas en la siguiente fotografía de hielos Pancake figura 4.7. Es una imagen en escala de grises. Analizando esta fotografía puede observarse que los tonos más oscuros corresponden al agua, pueden apreciarse dos tonos de grises diferentes que son parte del hielo y en blanco la parte más densa del hielo.

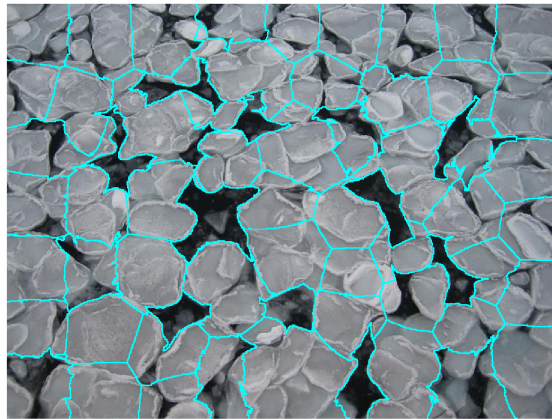


**Figura 4.7.** Fotografía de hielo Pancake.

En este caso se realizó la prueba con superpíxeles solicitando 4 superpíxeles. El algoritmo dividió la fotografía en 3 regiones espaciales figura 4.8(a), pero no se acerca a ningún Pancake o contenido de la fotografía. Se probó el algoritmo para 67 superpíxeles figura 4.8(b). Incrementando el número se puede apreciar que los superpíxeles contienen píxeles de respuesta parecida, sin embargo hacia las orillas de la imagen presenta dificultad y tienden a ser muy cuadrados y hay partes en las que los píxeles contenidos son muy diferentes, es decir contienen parte del agua y parte de hielo.



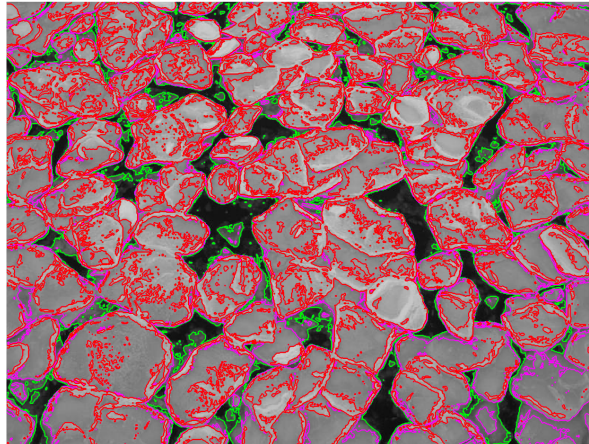
(a) Se muestra el resultado de pedir 4 superpíxeles al algoritmo



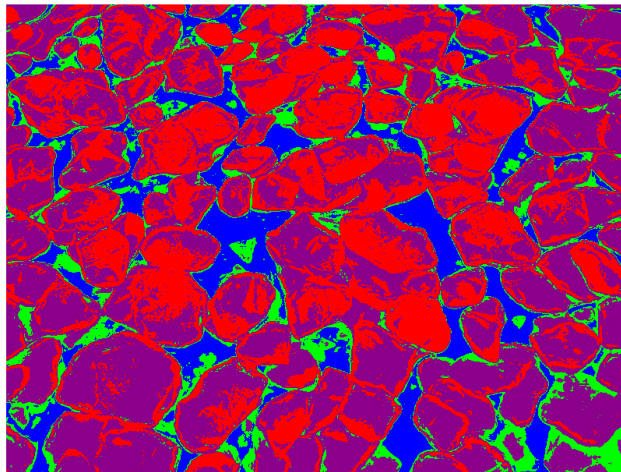
(b) Se muestra los 67 superpíxeles generados.

**Figura 4.8.** Algoritmo superpíxeles solicitando a) 4 superpíxeles y b) 67 superpíxeles.

La prueba con CV1D, se realiza para 4 centroides, cada uno logra separar las zonas de respuesta homogénea al sensor en una máscara de las cuales se obtienen los contornos y la imagen en falso color que se muestran a continuación en la figura 4.9:



(a) Contornos obtenidos de las máscaras.



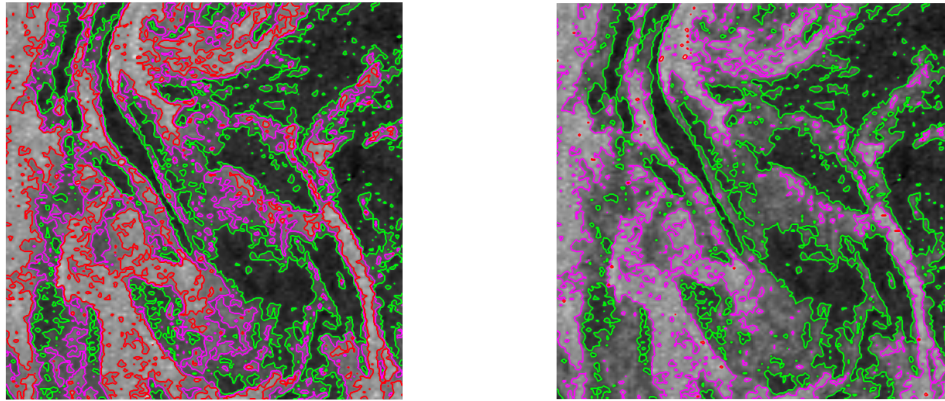
(b) Se muestra los superpíxeles generados.

**Figura 4.9.** Segmentación del hielo Pancake con CV1D con 4 regiones.

En los contornos obtenidos, figura 4.9(b), el de color verde contienen un tipo de hielo que está sumergido en el agua. Los contornos rojo y magenta son partes diferentes del hielo. La roja es la zona más densa del hielo, no se presentó el contorno azul en esta imagen, pero en la imagen de falso color, figura 4.9(b), se observa muy bien que el centroide en azul separa la parte del agua. Los colores corresponden a los contornos, solo se modificó el magenta a morado para lograr un mayor contraste en las 2 zonas diferentes del hielo.

### 4.3. Umbrales vs Cuantización Vectorial

La prueba de esta sección, se realiza con un recorte de 512 x 512 píxeles de la zona de hielo Pancake de la imagen satelital E9. Esta prueba consiste en generar 4 regiones con el algoritmo CV1D y con Umbrales descrito en capítulo 2. Los resultados se aprecian en las siguientes imágenes:



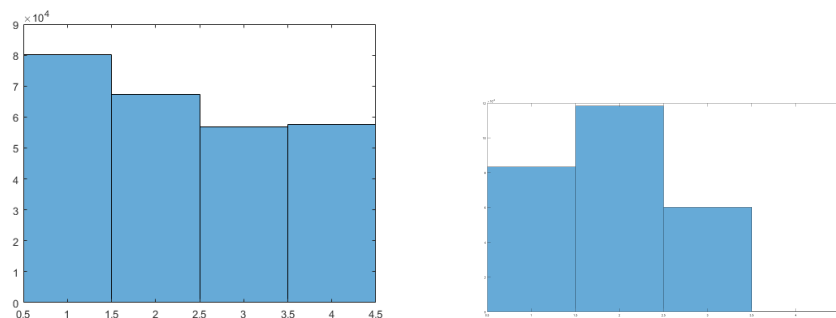
(a) Se muestra los contornos generados por CV1D.

(b) Se muestra los contornos generados por Umbrales.

**Figura 4.10.** Comparación entre regiones generadas por a)CV1D y b) Umbrales

Como puede observarse la diferencia principal está en los contornos rojo y magenta, en CV1D el área correspondiente al contorno rojo es más amplia, mientras que en los Umbrales apenas se encuentran algunos puntos. Puede notarse que los contornos magentas de los Umbrales asemejan mucho a los contornos rojos de CV1D. Sin embargo, en este último el área de contorno magenta proporciona un área de detalle más fino que podría pertenecer a un tipo de hielo de diferente grosor.

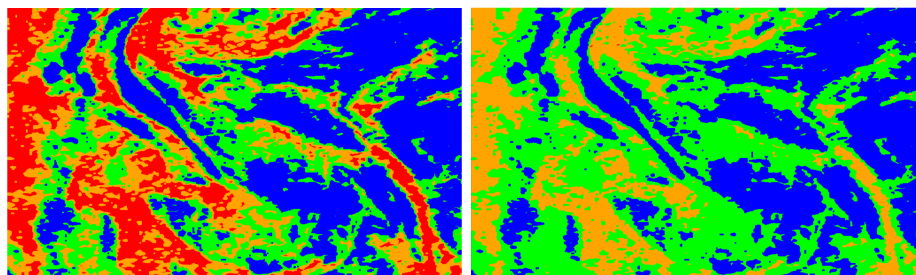
Si observamos los histogramas:



(a) Se muestra el histograma generado por Cuantización Vectorial. (b) Se muestra el histograma para los umbrales.

**Figura 4.11.** Comparación entre histogramas de las regiones generadas por a) Cuantización Vectorial y b) Umbrales.

Puede observarse que la Cuantización Vectorial genera secciones más homogéneas en cuanto a cantidad de datos incluida en ellas. Mientras que en umbrales puede generar sectores con muy pocos datos esto es un problema que mencionan algunos autores[23]. Se generaron unas imágenes en pseudocolor en las cuales se observa más claramente las diferencias en las regiones:



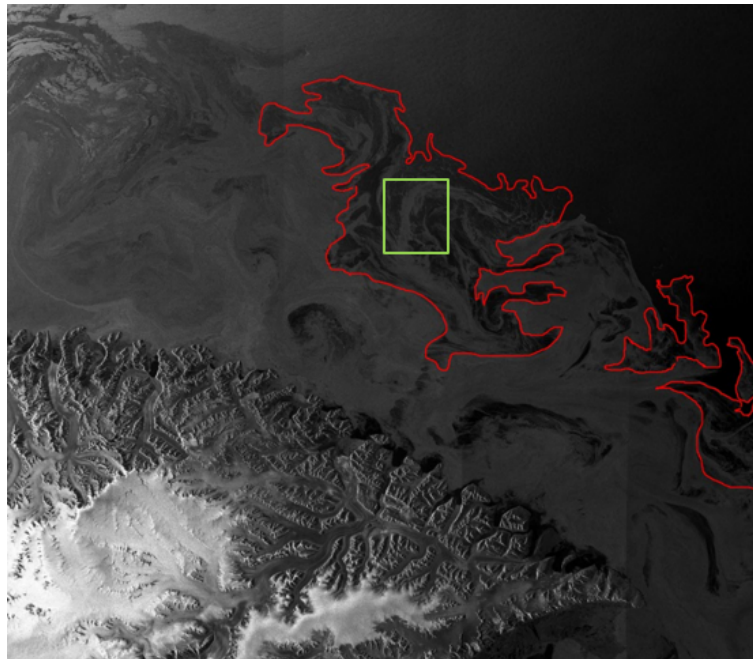
(a) Se muestran las regiones en pseudocolor generados por Cuantización Vectorial (b) Se muestran las regiones en pseudocolor generados por umbrales

**Figura 4.12.** Comparación en pseudocolor de las regiones generadas por a) CV1D y b) Umbrales

Observe que la imagen de Umbrales pareciera que solo tiene 3 diferentes zonas, en realidad tiene 4 solo que la cuarta sección tiene apenas unos cuantos datos en color rojo y se pierden en el resto de la imagen.

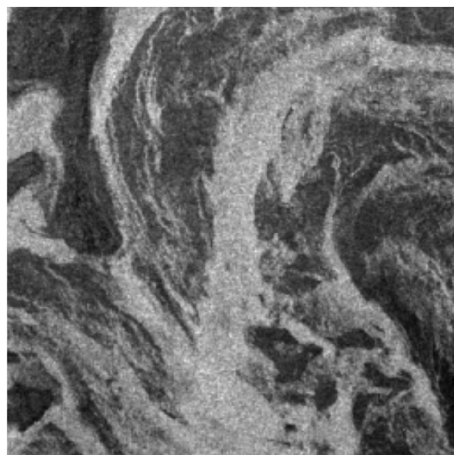
## 4.4. Imágenes Satelitales

Con las pruebas anteriores del perro Kobi y la fotografía de los hielos Pancake, en las cuales los objetos tienen formas bien definidas y es fácil distinguir los límites entre ellos, se pudo observar que la Cuantización Vectorial empleado en forma unidimensional CV1D es útil para separar regiones con respuesta semejante al sensor. Esto permite poder emplearlo en una imagen mucho más compleja como las imágenes satelitales SAR, en las cuales no se tiene una forma conocida fácil de identificar, pero se tiene la certeza que los centroides agruparán todos los elementos semejantes siguiendo un patrón contenido en la imagen. En la presente sección se utiliza la imagen satelital SAR D419, sus dimensiones son 8557 x 6763 pixeles, tiene la segmentación a mano del experto en hielo Pancake en contorno rojo.



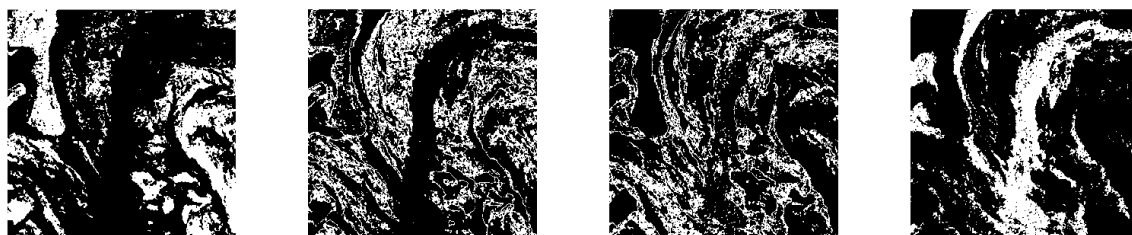
**Figura 4.13.** Se muestra la imagen de pruebas con su segmentación en rojo y en verde una ventana de análisis

Dentro de la zona clasificada como hielo Pancake se seleccionó una ventana para pruebas (recuadro verde) para probar el algoritmo de CV1D a continuación se muestra la ventana:



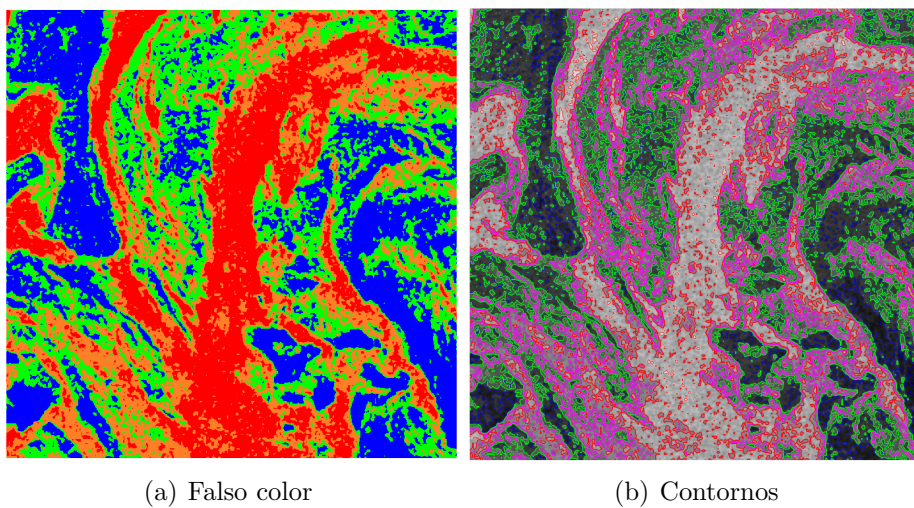
**Figura 4.14.** Ventana de análisis de D419

Después de aplicar CV1D se obtuvieron las siguientes máscaras:



**Figura 4.15.** Máscaras obtenidas a partir de CV1D para 4 centroides

A partir de las máscaras anteriores se generó la imagen en falso color y los contornos que se muestran a continuación:



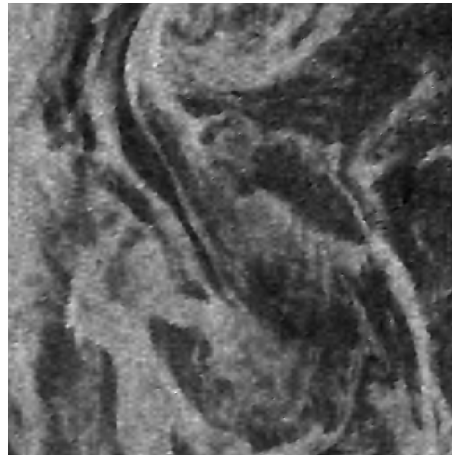
**Figura 4.16.** Resultado de CV1D en ventana de imagen D419



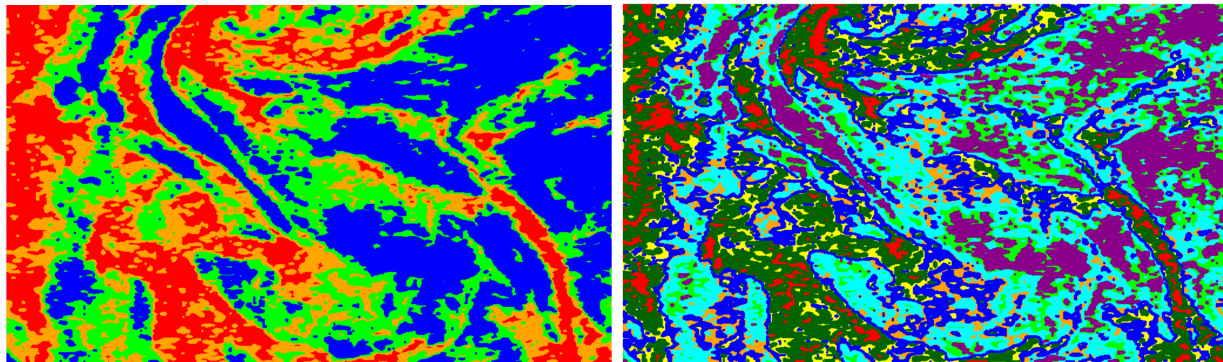
## 4.5. Textura por Regiones Estadísticas

A las 21 ventanas obtenidas de los recuadros verdes de la imagen **3.5** se les aplicó el algoritmo CV1D como en las secciones anteriores. Su análisis se presenta en el anexo A y las máscaras obtenidas en cada ventana son utilizadas en esta prueba denominada: Texturas por Regiones Estadísticas (TRE).

Se seleccionó el recorte número 4 de la zona con hielo Pancake, para mostrarse como ejemplo. Se separó para 4 y para 8 centroides. Se muestra a continuación estas separaciones:



**Figura 4.17.** Ventana de análisis zona de hielo Pancake 4.

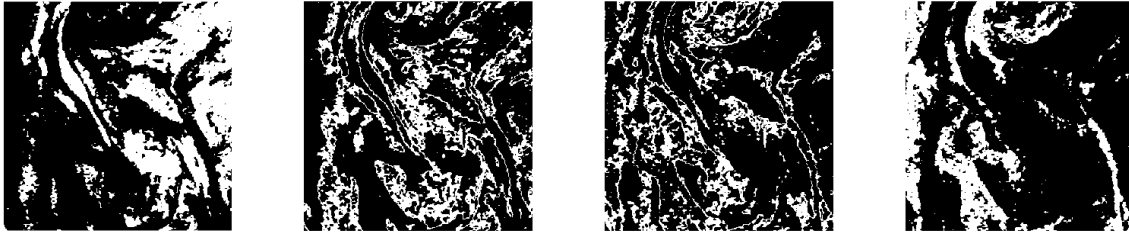


(a) Con 4 centroides.

(b) Con 8 centroides.

**Figura 4.18.** Ejemplo de ventana de análisis de imagen E9

Al analizarlos se nota que para 8 centroides existe cierta tendencia entre los bordes continuos. Esto puede deberse a que existe una continuidad en el material y no necesariamente sean 2 zonas diferentes, por lo cual, se determinó que con 4 centroides para esta imagen eran suficientes. Se presentan las máscaras para los 4 centroides.



**Figura 4.19.** Máscaras obtenidas a partir de 4 centroides.

Cada centroide contiene un cúmulo de píxeles que responde de manera semejante al sensor. El algoritmo CV1D entrega un máximo, un mínimo y el promedio o centroide. Para generar una descripción de textura a partir de estos centroides se calcula en cada máscara sus estadísticos de primer orden: desviación estándar, curtosis, skewness, también el porcentaje que representa del total de la ventana. Así la descripción de la textura en esa ventana se conforma de 4 vectores (uno por cada centroide) de 7 elementos cada uno (estadísticos de primer orden y porcentaje). En la siguiente tabla se presenta los 4 vectores que describen la textura de esta ventana:

Mínimo	Máximo	Centroide	STD	Curtosis	Skewness	Porcentaje
188.179	372.572	326.710	28.124	2.690	-0.444	0.306
372.576	468.068	418.438	27.599	1.807	0.075	0.257
468.070	568.921	517.699	29.646	1.763	0.045	0.216
568.923	940.530	620.1464	36.407	4.248	0.983	0.219

**Tabla 4.1.** Tabla vectores de Textura por Regiones Estadísticas

A partir del análisis anterior se generaron 2 codebooks de la siguiente forma:

El primero con una ventana de cada región a separar, es decir, una ventana de zona de hielo proporcionando 4 vectores para esta zona, una de zona de hielo Pancake (4 vectores), una para la zona de mar(4 vectores). En total tiene 12 vectores para este codebook.

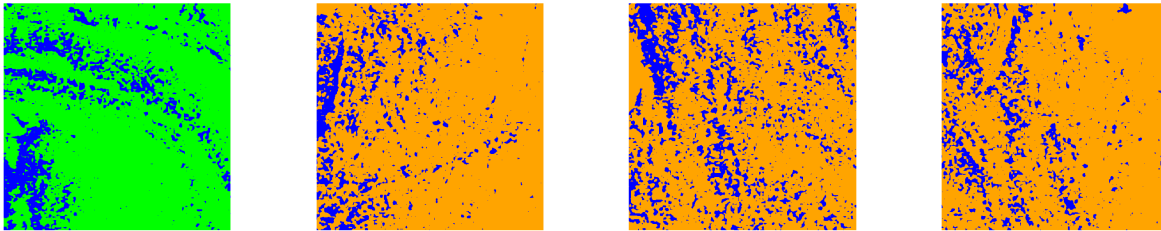
Como se mencionó anteriormente se tienen 21 ventanas de la imagen, 7 para cada zona a separar. Cada una de ellas fue clasificada con el codebook anterior para probar su desempeño. Se obtuvo el siguiente resultado:

Cuando la ventana era clasificada en su totalidad en una sola clasificación se obtiene una ventana de un solo color: verde para zona de hielo, azul para zona de hielo Pancake y amarilla para zona de mar. El resultado puede observarse en la siguiente Matriz de confusión.

	Hielo	Hilo Pancake	Mar
Hielo	5.38	1.63	0
Hilo Pancake	0	7	0
Mar	0	0.448	6.552

**Tabla 4.2.** Matriz de confusión para el codebook conformado por una ventana por cada región a separar con un total de 12 vectores.

Del total de 21 imágenes clasificadas se obtuvo el 90.1523 % clasificadas en su categoría, 9.8476 % clasificadas en otra categoría. A continuación, se muestran las ventanas que tuvieron 2 clasificaciones.



(a) Ventana de hielo 2.    (b) Ventana de mar 3.    (c) Ventana de mar 5.    (d) Ventana de mar 6.

**Figura 4.20.** Ventanas cuya clasificación está en 2 categorías diferentes.

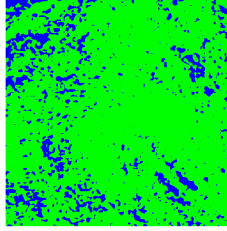
En el segundo codebook se utilizó con las 21 ventanas a cada una se aplicó CV1D para obtener los 4 vectores de descripción de textura por ventana. En total se tienen 28 vectores de descripción de textura para cada región.

Con estos vectores se genera un codebook que contiene las 3 regiones a separar en un total de 84 vectores. Se probó con estas mismas imágenes su desempeño obteniendo la siguiente Matriz de confusión:

	Hielo	Hilo Pancake	Mar
Hielo	6.86	0.14	0
Hilo Pancake	0	7	0
Mar	0	0	7

**Tabla 4.3.** Matriz de confusión para el codebook conformado por 7 ventanas por cada región a separar con un total de 84 vectores.

Del total de 21 imágenes clasificadas se obtuvo el 99.33 % clasificadas en su categoría, 0.76 % clasificadas en otra categoría.



**Figura 4.21.** Ventana hielo1 cuya clasificación está en 2 categorías diferentes.

Con estos 2 codebooks se probó el desempeño del algoritmo de Texturas por Regiones Homogéneas.

El codebook de 12 vectores se utilizó para clasificar la imagen completa **3.5** la cual contienen zonas que no están en el codebook. Esto es aprovechado para probar cómo se clasifica el resto de la imagen a excepción de la tierra, puesto que esta parte queda fuera del estudio. Se quitó a través de una máscara de tierra figura **4.22**.

El tamaño de la imagen permite dividirla en 15 x 18 ventanas de 512 x 512 píxeles, es decir, 270 subimágenes a procesar. La máscara de tierra permite reducir a 172 subimágenes a procesar.

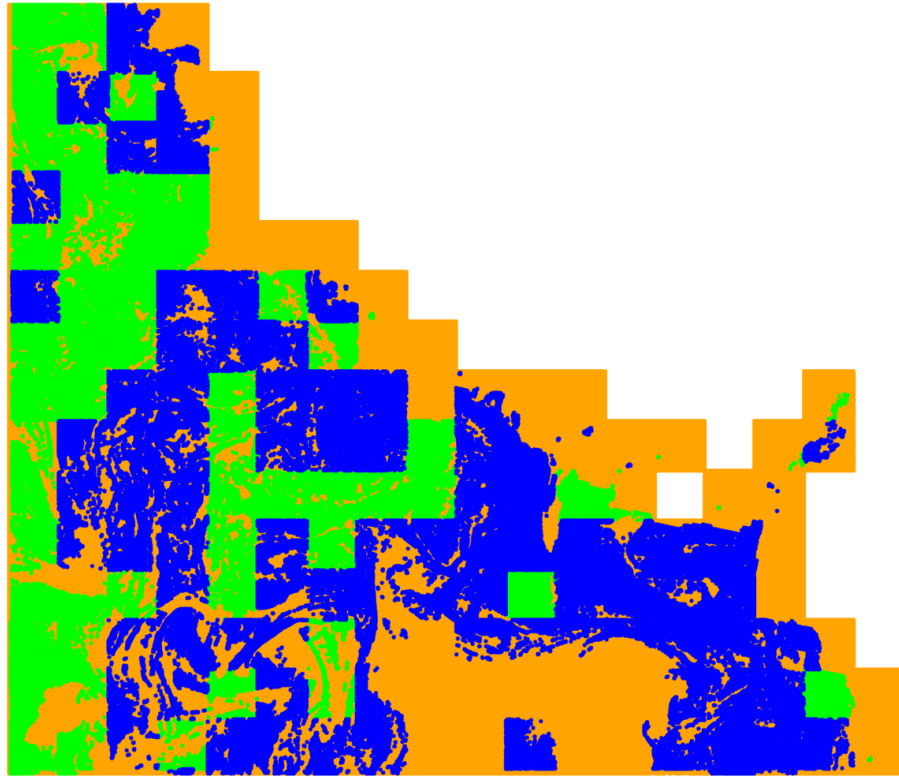


**Figura 4.22.** Máscara de Tierra para la imagen E9.

El algoritmo para recorrer la imagen y clasificarla es el descrito en la figura **3.7**.

El tiempo de proceso del algoritmo CV1D para generar la división en 4 centroides, en una imagen de 512 x 512 es de 4 minutos con 56 segundos. Para una imagen de 1024 x 1024 es de 18 minutos con 21 segundos.

A continuación se presenta el resultado de esta clasificación en la figura 4.23.

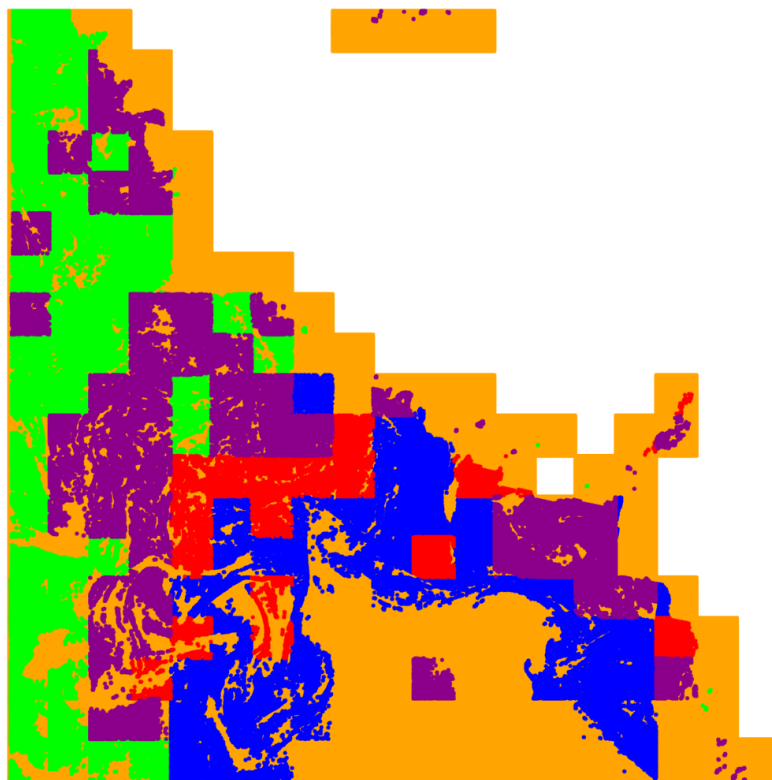


**Figura 4.23.** Se muestra el resultado obtenido de la clasificación con Textura por regiones estadísticas.

Se puede observar gran parte en blanco que corresponde a la máscara de tierra. El color verde fue designado para la zona de hielo, el color azul se usó para la zona de hielo Pancake y el color amarillo para la zona de mar. En la imagen se ven cuadrados, estos corresponden a cada ventana que se procesó de 512 x 512 píxeles.

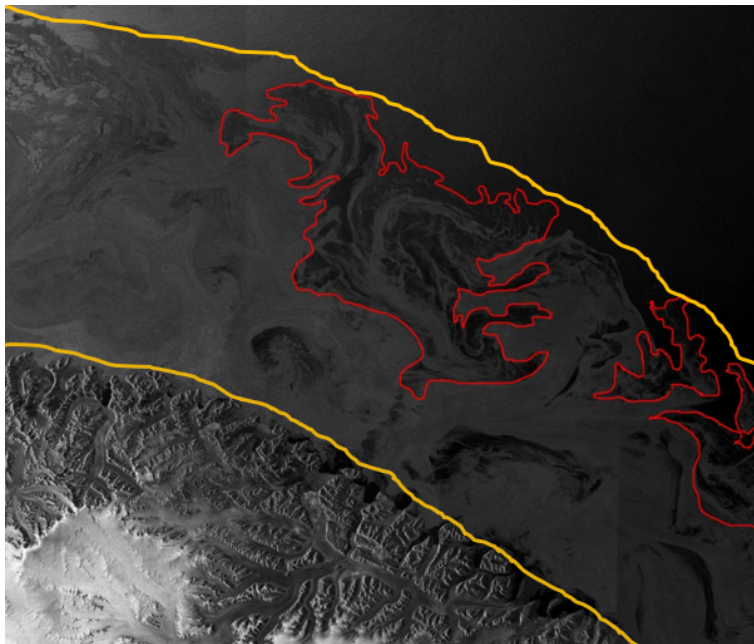
El algoritmo tardó 28 horas con 48 minutos en procesar las 172 subimágenes.

La región de hielo Pancake es complicada puesto que está conformada por hielo en diversas densidades y la mezcla entre ellas forma la zona de interés. Esto puede provocar que existan zonas de hielo contenidas en la zona de interés, o zonas de mar en la zona de interés. Esta complejidad puede provocar falsos positivos o positivos falsos, que deberán ser analizados por el experto para decidir a qué parte pertenece. En la figura 4.24 se muestra una imagen con posibles falsos positivos o positivos falsos marcados en morado y rojo.



**Figura 4.24.** Se presentan los posibles falsos positivos o positivos falsos.

El codebook generado con la imagen E9, se probó con la imagen D419, figura 4.25(a), a la cual se colocó un par de líneas amarillas que enmarcan la zona donde puede existir hielo y hielo Pancake. Además, se agregó una máscara de tierra, figura 4.25(b) , ubicada en la esquina inferior izquierda. En este caso se dejó en negro en los resultados. Los colores corresponden a azul para zona de hielo Pancake, verde zona de hielo, amarillo zona de mar. A continuación se muestran las imágenes generadas:

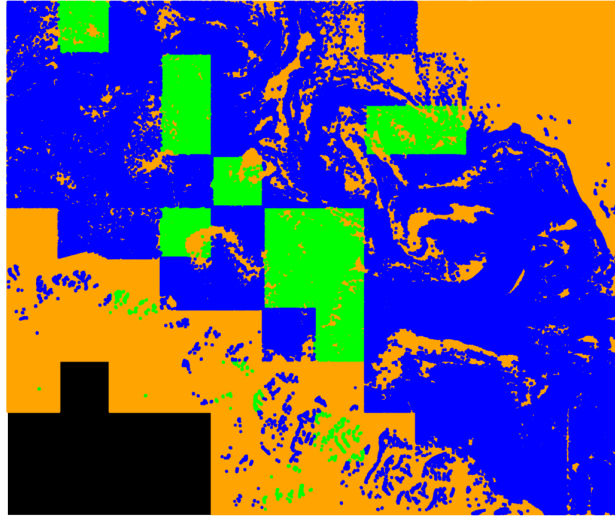


(a) Las líneas amarillas delimitan la zona de estudio de la imagen.

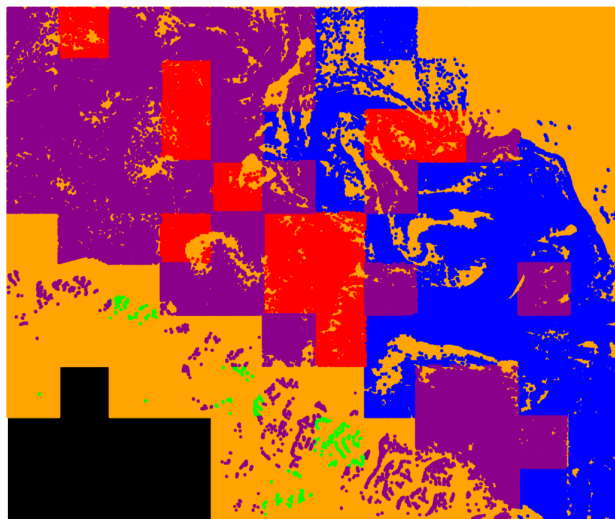


(b) Máscara de tierra.

**Figura 4.25.** Zona de interés y Máscara de tierra para la imagen D419.



(a) Resultado directo.



(b) Posibles falsos negativos.

**Figura 4.26.** Clasificación de imagen D419 con codebooks de imagen E9 a) Resultado directo y b) Posibles falsos positivos.

La figura 4.26(a) muestra la clasificación realizada. El color amarillo corresponde a los centroides de zona de mar, el azul a los centroides de zona de hielo Pancake y el verde a los centroides de hielo.

Analizando los resultados puede notarse la esquina superior derecha en amarillo clasificada como zona de mar, lo cual es correcto, en la esquina inferior izquierda se tiene una parte negra que pertenece a la zona de la máscara de tierra, recordando que el algoritmo al



encontrar el valor máximo de la ventana igual a cero no procesa esa ventana y la deja en cero.

En seguida, en dirección hacia el centro, se tiene una parte amarilla, en esta parte también es máscara de tierra, pero el algoritmo la clasificó, puesto que se encontraron datos diferentes de cero en la venta y realizó el proceso. El centroide más cercano a la máscara de tierra (ceros) es el de la parte de mar (valores más bajos de la imagen) por lo cual, aunque esté erróneo, lo clasificó en el centroide correcto.

En la franja en azul y verde (centro de la imagen en diagonal) las regiones están clasificadas en los centroides correctos puesto que esa zona corresponde a hielo y a hielo Pancake. Como se explicó anteriormente la zona de estudio es complicada, el material es el mismo: agua congelada, pero en diferentes densidades y tamaños. La diferencia incluso en textura puede ser muy sutil. Esto genera que haya partes que puedan ser falsos positivos como se observa en la figura 4.26(b) en color morado y rojo.

Al compararse estas zonas con la segmentación del experto, se observa que en este caso, todos podrían ser falsos positivos de hielo Pancake (morado) y pertenecer a la zona de hielo y la zona roja pertenecer a la zona de hielo y por tanto estar bien clasificados.

#### 4.5.1. Pruebas con Cuantización Vectorial y Transformada Hermite

Las siguientes secciones proponen explorar la versatilidad del algoritmo de Cuantización Vectorial junto con la Transformada Hermite (TH).

Las pruebas de esta sección se realizan con la imagen E9 para obtención de codebook. Se generaron las siguientes pruebas:

1. En el coeficiente cero de la transformada Hermite se aplica CV1D para 4 centroides y se presenta en falso color.
2. Se utilizan todos los coeficientes de la transformada Hermite como vector de características de la textura. Se utiliza Cuantización Vectorial con estos vectores para generar un codebook de centroides vectoriales de la zona de hielo Pancake y otro codebook con el resto de la imagen.

#### 4.5.2. CV1D en Coeficiente cero de TH

Siguiendo las pruebas que se han desarrollado en este capítulo, se prueba CV1D con el coeficiente cero de la Transformada Hermite.

En la sección 3.2.3 se explica la metodología de la Transformada Hermite. A la imagen E9 se le aplicó TH con los siguientes parámetros:

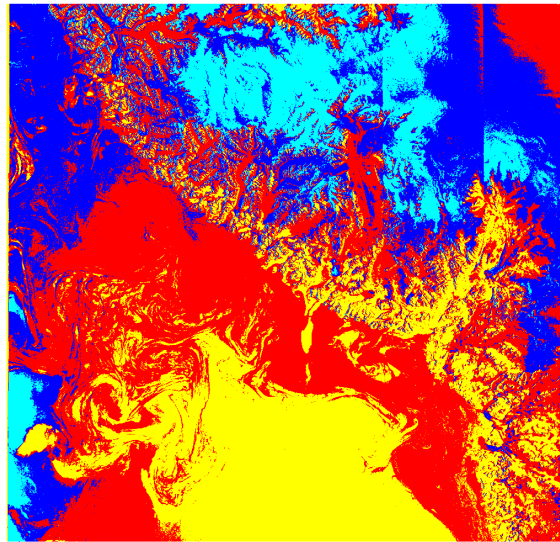
- $N=40$ , número de píxeles que contribuyen a la ventana de análisis.
- $D= 4$ , número de polinomios.

- $T=2$ , número de paso de píxeles.

Estos parámetros son los propuestos en el ejemplo de la Transformada Hermite. En la siguiente sección se utilizan otros parámetros con base en el análisis previo en la sección 4.5.

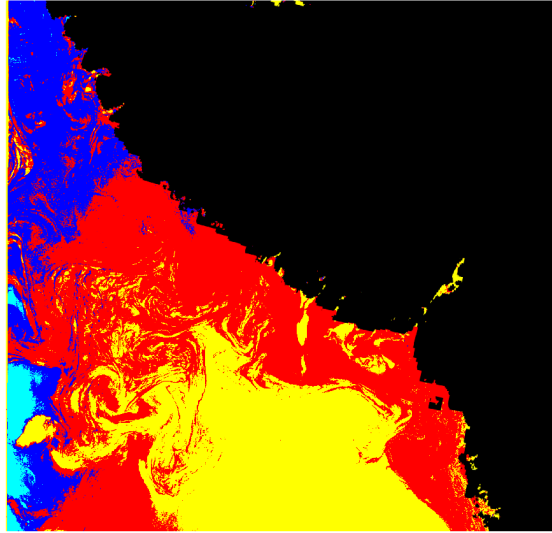
El resultado es la reducción del coeficiente cero a la mitad del tamaño original de la imagen, además cada valor es la contribución del tamaño de la ventana de análisis es decir 40 píxeles. En la siguiente sección se utilizan otros parámetros.

Puede realizarse de dos formas la prueba CV1D, una de ellas es procesar toda la imagen y posteriormente ponerle la máscara de tierra. Esto se puede observar en las siguientes imágenes:



(a) CV1D en toda la imagen.

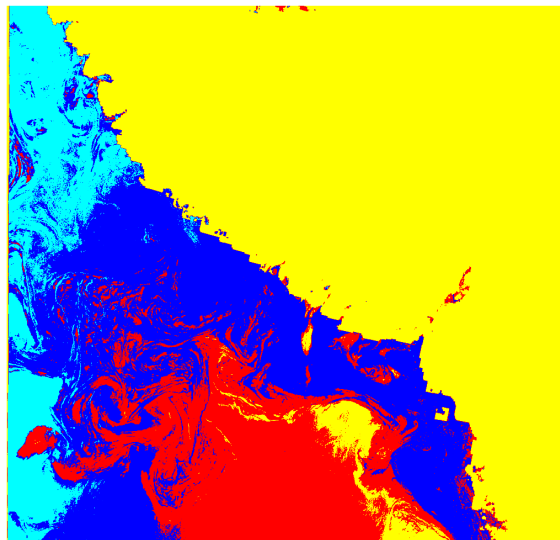
**Figura 4.27.** CV1D aplicado en el coeficiente cero de la Transformada Hermite de la imagen E9.



(a) Máscara colocada después del procesamiento.

**Figura 4.28.** CV1D aplicado en el coeficiente cero de la Transformada Hermite de la imagen E9, con máscara de tierra.

Al poner la máscara de tierra antes de aplicar CV1D pueden cambiar los resultados como se observa a continuación:



**Figura 4.29.** Se agregó la máscara de tierra al experimento.

### 4.5.3. Uso de Cuantización Vectorial junto con vector de coeficientes de Transformada Hermite, para generar un codebook.

Hasta el momento las pruebas realizadas han sido unidimensionales. Se ha podido observar que el uso del algoritmo LGB en forma unidimensional (CV1D) es útil para dividir una imagen en 4 centroides que siguen los patrones principales de respuesta al sensor.

Probar esto es de gran importancia, puesto que la comprensión de patrones es comprensible a 3 dimensiones, pero a un número mayor de dimensiones es difícil imaginar cual es el patrón que se obtiene.

La siguiente prueba utiliza el algoritmo LGB en forma vectorial a través de la TH para generar un codebook que caracteriza la textura, en vectores, en este trabajo se usan 15 dimensiones como se explicará a continuación.

En la sección 3.2.3 se menciona la forma en que puede usarse la TH en forma vectorial. Esto se ilustra en la figura 3.3.

El análisis con ventanas de las pruebas de las secciones anteriores ayudó a determinar que en una imagen SAR una ventana de 512 x 512 permite observar las características de una zona de hielo Pancake, todavía una ventana de 128 x 128 permite diferenciar la textura de las 3 zonas dentro de la imagen. Este parámetro se tomó en cuenta para la siguiente prueba, en el tamaño de ventana de contribución  $N$ .

Entre los parámetros que se probaron para la TH en la imagen E9 están: ventana de contribución  $N = 512$ , paso  $T = 128$  número de polinomios  $D = 4$ . El resultado fue la reducción de la imagen a 79 x 82 píxeles. Esto en vectores es de 6478 de 15 dimensiones. Otros parámetros fueron:  $N = 128$ ,  $T = 32$  y  $D = 4$ . La imagen se redujo a 313 x 327 píxeles, es decir 102351 vectores de 15 dimensiones.

Finalmente, se realizó la Transformada Hermite a con los siguientes parámetros:

Tamaño de ventana de contribución  $N = 128$   
Tamaño de paso entre ventanas  $T = 16$   
Número de polinomios de Hermite  $D = 4$

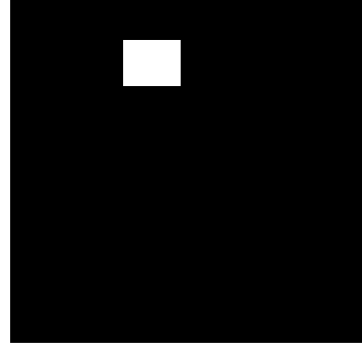
Los vectores que proporciona la TH contienen información de la contribución de su vecindad en este caso son  $N = 128$  píxeles. El paso entre ventanas ( $T = 16$ ) provoca la reducción de la imagen a 625 x 654 píxeles que forman un total de 408,750 de vectores característicos de textura. Finalmente, el número de polinomios  $D = 4$  genera 15 dimensiones en los vectores.

La información proporcionada en esta última prueba es adecuada para generar el codebook. Entre las cosas que se cuidaron fue no reducir demasiado el tamaño de la imagen y tener información suficiente para generar el codebook. Otro detalle que se debe cuidar para generar el codebook es seleccionar las zonas a separar.

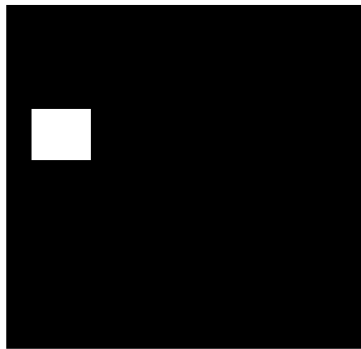
El profesor Abel Pacheco contribuyó generando la máscara de zona de hielo Pancake 4.30(a) a partir de la imagen segmentada por el experto y por medio de las librerías de OpenCV y se muestra a continuación.



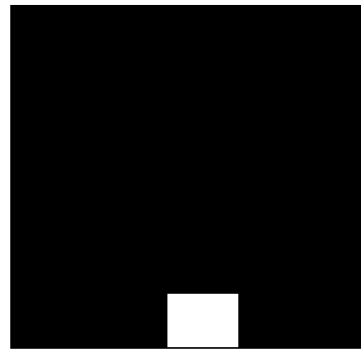
(a) Máscara zona hielo pancake



(b) Máscara zona de tierra



(c) Máscara zona de hielo



(d) Máscara zona de mar

**Figura 4.30.** Máscaras de extracción de vectores de la imagen E9

A esta máscara se aplicó la Transformada Hermite con los mismos parámetros para conseguir el mismo tamaño. Se utilizó para extraer los vectores correspondientes a la zona de interés. Se obtuvieron 67,323 vectores.

El resto de la imagen contiene 341,427 vectores de la zona a separar. Se extrae una muestra de las otras zonas a separar: la zona de hielo 10,044 vectores, la zona de mar 12,513 vectores y la zona de tierra 42,542 vectores.

A cada una se aplica Cuantización Vectorial para obtener 8 centroides de cada zona. El conjunto de estos centroides constituye el codebook el cual se muestra en el Anexo B en una tabla para cada zona y pueden probarse con la imagen D419, en un trabajo futuro.

# 5. Conclusiones y Trabajo Futuro

## 5.1. Conclusiones

La Cuantización Vectorial agrupa los datos siguiendo patrones contenidos en la señal. Se probó la versatilidad del algoritmo de Cuantización Vectorial en imágenes usándose en forma unidimensional (CV1D) con el valor de píxel como característica principal. Esto funciona para segmentar, separando las regiones de respuesta homogénea al sensor correspondientes al número de centroides.

Al usar la respuesta al sensor como característica principal, cada centroide forma vecindades que se apegan al contorno de los objetos. El resultado de esta separación es una máscara por cada centroide, cada una puede ser usada para separar objetos o segmentos de dicho objeto que en conjunto con otros segmentos que estén en otra máscara generen el objeto de interés. Por ejemplo, en el experimento de Kobi, los ojos y nariz estaban en una máscara mientras el cuerpo en otra, pero en ambos casos eran los conjuntos más grandes que contenía la máscara.

Otra forma de usar las máscaras es para generar los contornos que limitan cada zona de la imagen o también son usadas para generar imágenes en falso color y visualizar la separación.

La Cuantización Vectorial genera centroides en números de potencia de 2. Sí se requiere un número impar de centroides puede agregarse a la imagen un marco en color blanco o negro según convenga, en inglés se le conoce como "padding", este marco será contenido en un centroide, como pudo apreciarse en el experimento del perro Kobi con marco blanco. Un efecto similar sucede si se agrega máscara de tierra en las imágenes satelitales.

En comparación con otros algoritmos se encontró que CV1D es mejor que Umbrales, debido a que CV1D, al estar basado en el algoritmo LBG genera patrones contenidos en la señal. Mientras que en Umbrales solo se escoge un valor y se elimina la señal, pero no necesariamente sigue un patrón.

Con otros algoritmos como superpíxeles que toman la ubicación espacial la distribución no necesariamente se pega al contorno de los objetos. Para ambos casos se necesita un pos-procesamiento, sin embargo, CV1D como segmentador permite la identificación de los objetos sin necesidad de poner la imagen como fondo, esto en superpíxeles es necesario, puesto que sin la imagen de fondo no podría decirse a qué corresponde cada superpíxel.

El algoritmo de Cuantización Vectorial empleado en forma unidimensional es útil para

cambiar la resolución de la imagen sin alterar su tamaño, como ocurre con otros algoritmos.

Se puede analizar la textura en imágenes de la naturaleza obteniendo ventanas que solo contenga la textura de interés. Esta ventana es segmentada en 4 regiones con respuesta semejante al sensor por medio de Cuantización Vectorial. De estas regiones se obtuvieron estadísticos de primer orden para generar 4 vectores que en conjunto describen la textura contenida en la ventana analizada.

A partir de varias ventanas de análisis de las zonas a separar de la imagen E9 se generó un codebook que fue utilizado tanto para clasificar el resto de la imagen E9, así como la imagen D419 en ambos casos la zona de mar fue clasificada correctamente, la máscara de tierra se clasificó en el centroide de mar, que es el centroide más cercano y por tanto la clasificación es correcta. La zona de hielo y Pancake en la imagen D419 fue localizada correctamente, aunque la separación entre estas 2 zonas no está bien determinada, pero es debido a la complejidad de la imagen. Es decir, ambas zonas contienen hielo de alta densidad y los centroides quedan traslapados.

La transformada Hermite, proporciona vectores que caracterizan la textura, fueron empleados junto con el algoritmo de Cuantización Vectorial para generar un codebook, que está listo para clasificar otra imagen.

Se probó la versatilidad del algoritmo de Cuantización Vectorial en imágenes: como algoritmo de segmentación, como herramienta para caracterizar texturas: con Estadísticos de primer orden (Texturas por Regiones Estadísticas) y en forma vectorial junto con la Transformada Hermite. Aún tiene formas por desarrollarse como se propone a continuación en Trabajo Futuro.

## 5.2. Trabajo Futuro

### *Segmentación con CV1D*

A lo largo de este trabajo se mostró que en Cuantización Vectorial se puede separar una imagen en regiones de respuesta homogénea al sensor. En ellas las vecindades se generan en forma natural distribuidas en todo el espacio de la imagen. En el caso de imágenes con objetos bien definidos cada máscara puede contener partes de un objeto, por ejemplo, los ojos y nariz del perro Kobi en una centroide mientras que el cuerpo quedó en otro centroide.

En otro caso una máscara puede contener partes de objetos diferentes por ejemplo la máscara que contiene los ojos del perro, también contiene los mosaicos negros. Un post-procesamiento es separar estos objetos de cada centroide.

Posteriormente se pueden unir las vecindades más cercanas para conformar un objeto, logrando su segmentación. Este post-procesamiento puede hacerse de la siguiente manera: Tomar una máscara que contenga las vecindades a separar, con ella obtener los valores ori-

ginales de la imagen y posteriormente aplicar nuevamente CV1D a esa máscara, con lo cual se separarían las vecindades.

Por otra parte las librerías de OpenCV tienen algoritmos para seleccionar contornos por su importancia en dimensión, esto ayudaría a separar las vecindades más grandes en cada centroide.

### *Cuantización Vectorial con Transformada Hermite como vector*

En la sección 3.2.3 se generó un codebook para cada zona de interés de la imagen E9, este codebook debe probarse en la misma imagen y en la imagen D419 de forma similar a como se hizo en la sección 4.5 Textura por Regiones Homogéneas.

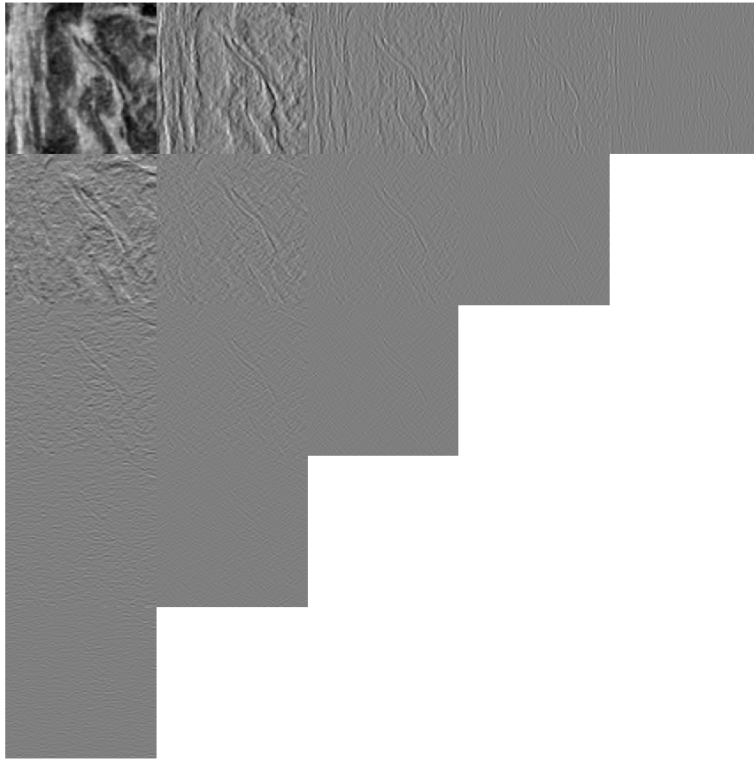
### *Transformada Hermite con Cuantización Vectorial 4 centroides*

Puede probarse la Cuantización Vectorial sobre cada uno de los coeficientes de la Transformada Hermite. Esto permite ver en falso color los patrones contenidos en cada coeficiente, de tal forma que podría identificarse algún patrón que no corresponda a aquellos detalles principales del coeficiente cero. Al estar separados los patrones podría eliminarse alguno ruidoso, de modo que al hacer la Transformada Hermite inversa la imagen quedase filtrada. Este procedimiento se hace de forma semejante pero con eliminación de Umbrales [7].

La ventaja con Cuantización Vectorial es que directamente se elimina un patrón y no solo datos seleccionados.

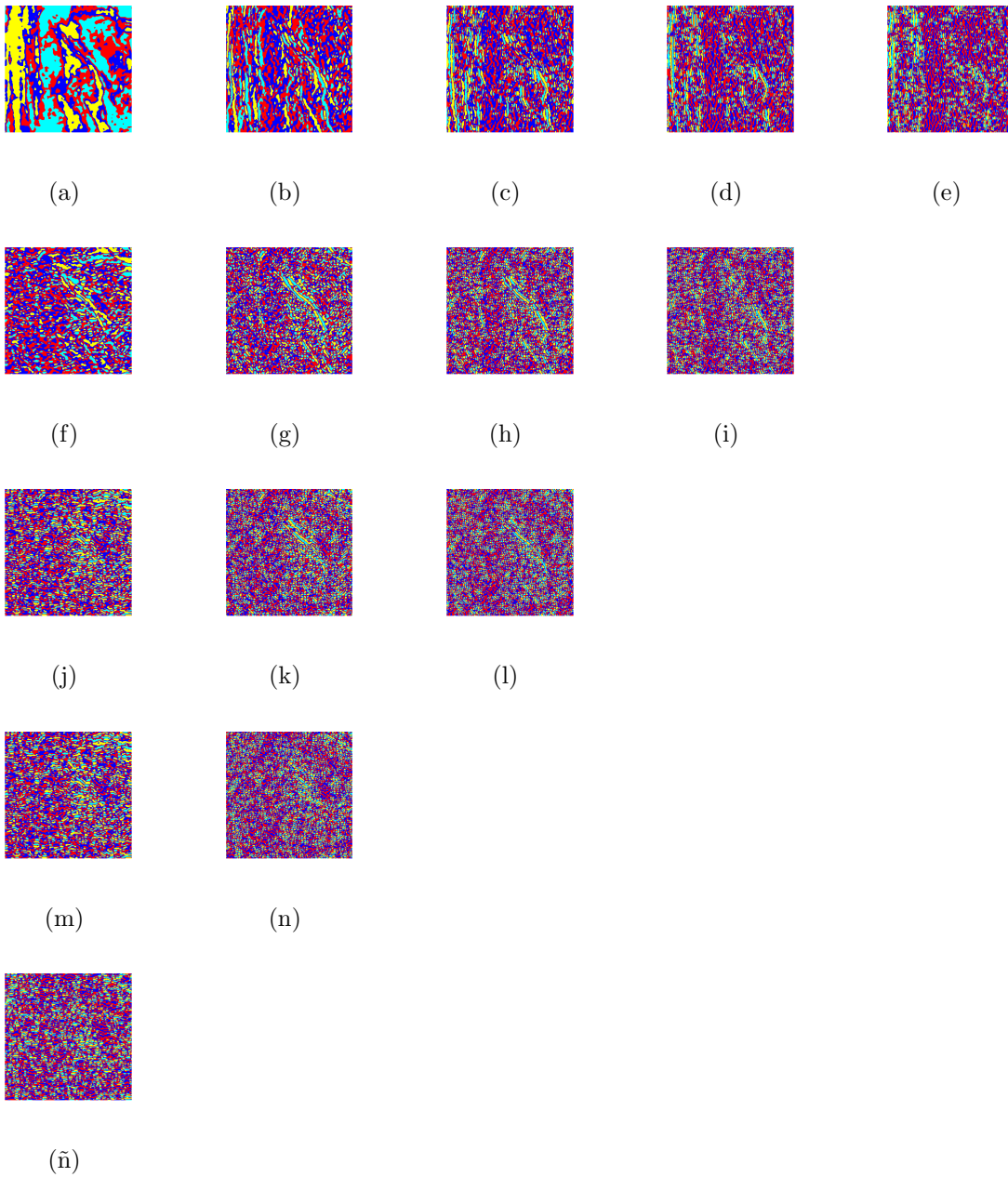


La siguiente imagen es la Transformada Hermite de la ventana 6 de zona de hielo Pancake.

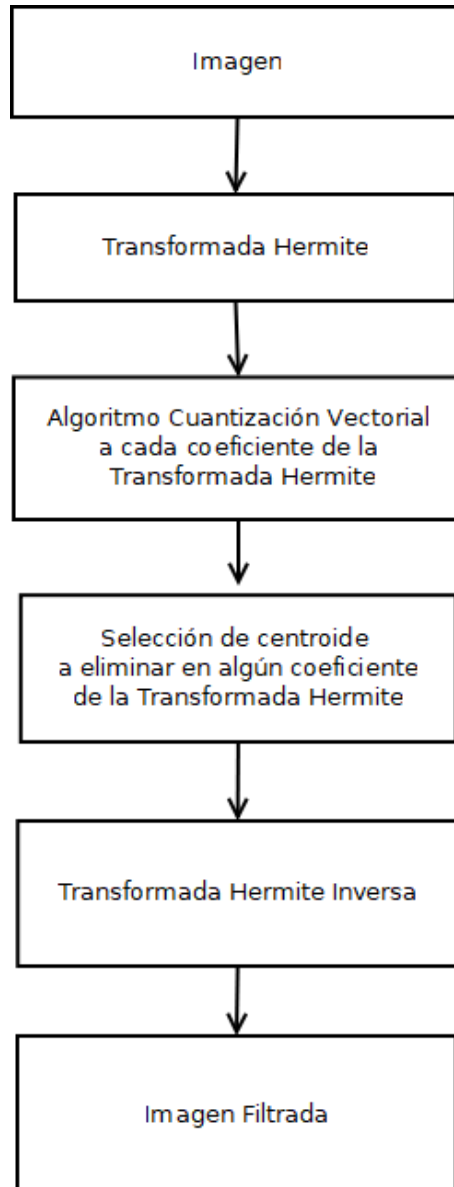


**Figura 5.1.** Se muestra la ventana 6 de la zona con hielo Pancake de la imagen E9

A continuación, se presentan los coeficientes de la transformada Hermite segmentados individualmente con CV1D.



**Figura 5.2.** CV1D aplicada a cada coeficiente de TH. Cada coeficiente está cuantizado con 4 regiones.



**Figura 5.3.** Diagrama Propuesta para generar filtro con TH y CV1D

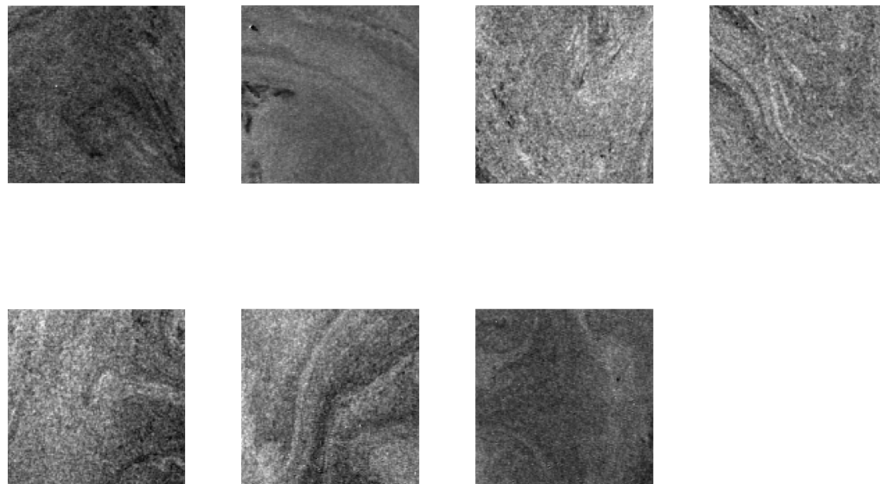
En la figura 5.3 se muestra el proceso propuesto para generar un filtro a partir de la Transformada Hermite y CV1D. Primero se realizaría la Transformada Hermite a la imagen, posteriormente se aplicaría CV1D a cada uno de los coeficientes de la Transformada Hermite. Se realizaría un análisis para buscar un centroide que no corresponda a los patrones del coeficiente cero, se eliminaría para finalmente hacer la Transformada Hermite inversa y obtener la imagen filtrada.

# A. Anexo: Ventanas de análisis imágenes SAR

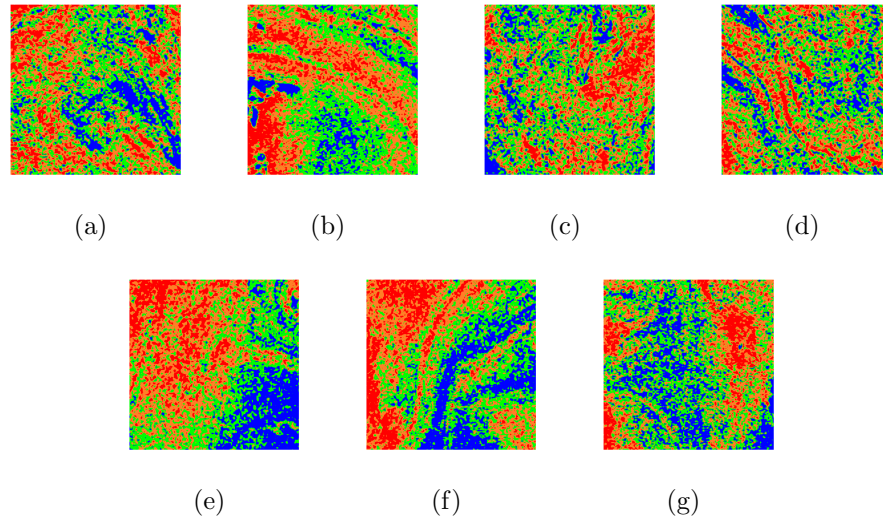
En referencia a a sección 4.5 del capítulo 4 se presentan en la imagen **3.5**, 3 cuadros verdes que son las zonas a separar, el primer cuadro de arriba hacia abajo es la zona de hielo, el segundo es la zona de hielo Pancake, y el cuadro inferior es la zona de mar. Se presentan los resultados en ese orden.

En cada una de las zonas de interés se obtuvieron subimágenes de 512 x 512 píxeles.

Para la zona de hielo se obtuvieron 7 subimágenes figura **A.1** nombradas: subimagen hielo1, hielo2, continuando la numeración de izquierda a derecha y de arriba hacia abajo.

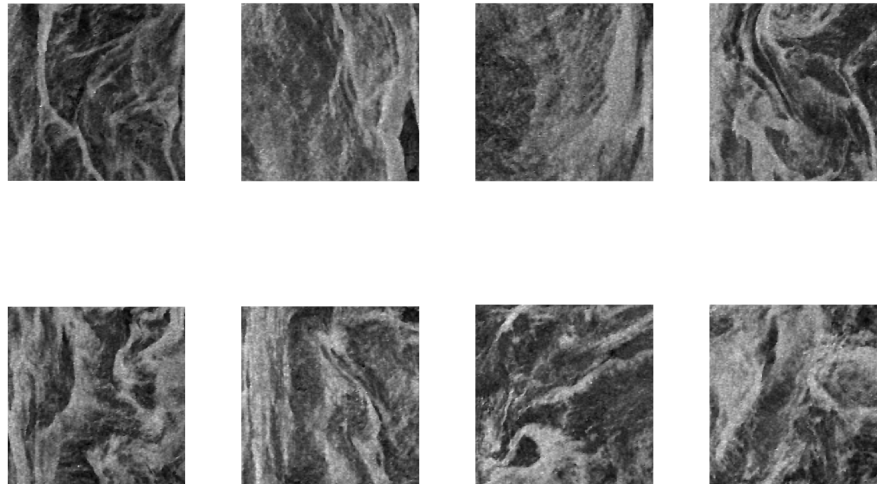


**Figura A.1.** Subimágenes de la zona de hielo



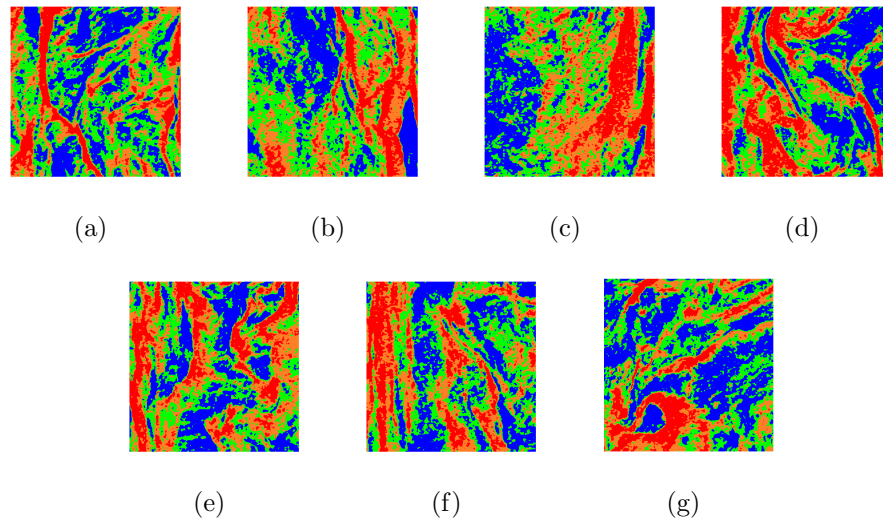
**Figura A.2.** Se muestran en falso color la segmentación de la ventanas de la zona de hielo

Para la zona de hielo Pancake se muestran en la figura **A.3**; se nombran: subimagen pancake1, pancake2 y así sucesivamente de izquierda a derecha y de arriba hacia abajo.



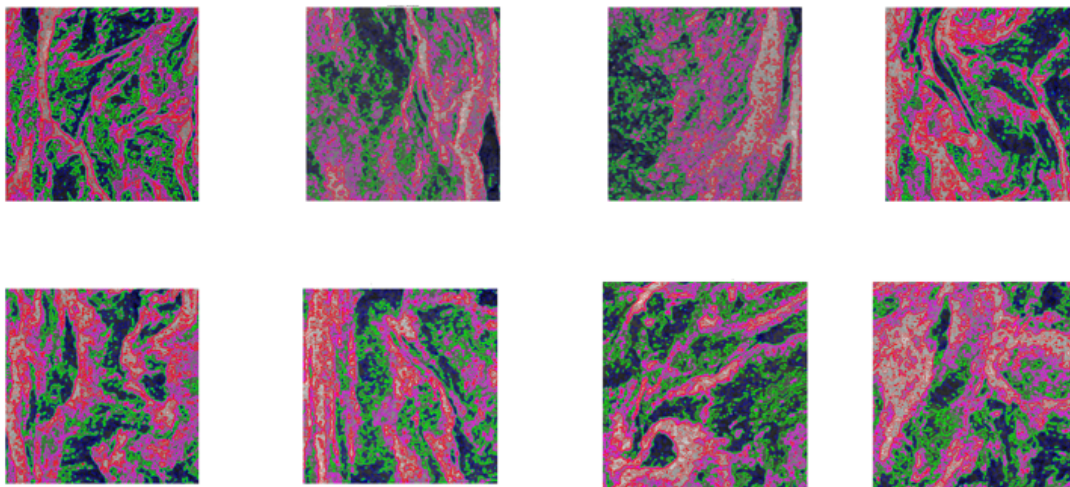
**Figura A.3.** Subimágenes de la zona de hielo Pancake

Cada una de estas subimágenes se procesó con CV1D y con las máscaras se obtuvieron las imágenes en falso color siguientes:



**Figura A.4.** Se muestran en falso color la segmentación de la ventanas de la zona de hielo Pancake

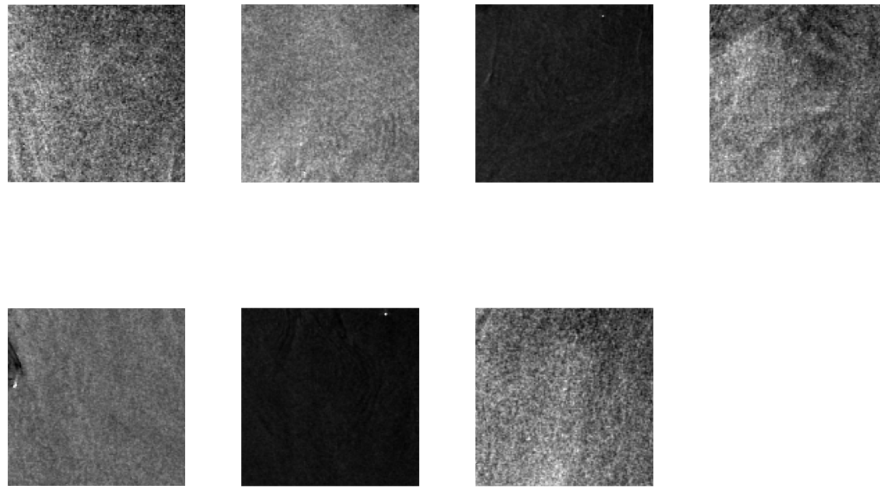
De forma análoga con las máscaras se obtuvieron los contornos para las regiones:



**Figura A.5.** Se muestran los contornos generados a partir de los centroides obtenidos para la zona de hielo Pancake

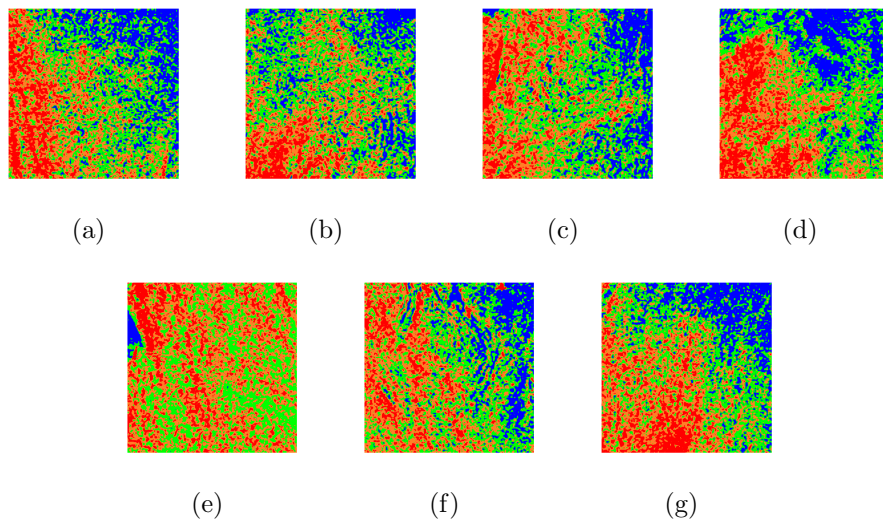
En cada una de las regiones se calcularon los vectores estadísticos para generar el codeblock de clasificación descrito en la sección 4.5

En la figura **A.6** se aprecian los recortes para la zona de mar, en total son 7 nombrados de forma similar a las zonas anteriores, submar1, submar2...



**Figura A.6.** Subimágenes de la zona de mar

Las imágenes en falso color obtenidas son las siguientes:



**Figura A.7.** Se muestran en falso color la segmentación de las ventanas de la zona de Mar

## B. Anexo: Tabla de centroides vectoriales de TH con CV

En este Anexo se muestra el resultado de los centroides vectoriales obtenidos con la Transformada Hermite y la Cuantización Vectorial. El encabezado es Coeficiente se ha abreviado como C y el número de coeficiente correspondiente.

<i>C0</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	<i>C9</i>	<i>C10</i>	<i>C11</i>	<i>C12</i>	<i>C13</i>	<i>C14</i>
798.48	-1.05	0.72	-8.99	1.38	-4.18	0.43	0.13	0.11	0.51	1.45	-0.49	0.80	-0.33	0.64
358.85	0.48	-0.52	1.22	0.01	0.93	-0.08	-0.01	-0.03	0.02	-0.21	0.05	-0.21	-0.03	-0.17
516.22	0.22	-0.12	-2.68	-0.59	-2.61	-0.20	0.00	-0.03	0.10	0.40	0.10	0.35	0.11	0.43
201.96	0.84	-0.66	1.95	0.34	1.82	-0.13	0.08	0.00	0.00	-0.39	-0.05	-0.26	-0.03	-0.35
623.89	0.71	0.10	-6.83	-0.64	-3.73	-0.68	-0.02	-0.09	-0.02	1.34	0.18	0.64	0.14	0.74
283.21	0.55	-0.89	1.28	0.13	0.73	-0.03	-0.03	-0.01	0.04	-0.17	0.00	-0.08	-0.03	-0.03
433.76	-0.01	-0.07	-0.15	0.16	-0.92	-0.10	0.00	0.08	-0.12	-0.04	-0.08	0.08	0.00	0.23
107.37	0.79	-0.66	1.94	-0.25	2.16	0.47	-0.05	0.04	-0.07	0.08	0.00	-0.26	0.04	-0.36

**Tabla B.1.** Tabla de 8 centroides generados con los coeficientes (C0, C1, ... C14) de la transformada Hermite y Cuantización Vectorial, de la zona de hielo Pancake. Cada renglón es un centroide.



<i>C0</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	<i>C9</i>	<i>C10</i>	<i>C11</i>	<i>C12</i>	<i>C13</i>	<i>C14</i>
724.572	0.019	-0.490	-5.218	-0.391	-3.387	-0.291	0.104	0.107	0.002	1.541	-0.109	0.960	0.0390	1.265
590.732	-0.557	-0.758	0.380	-0.418	0.3243	-0.113	-0.141	-0.043	-0.006	0.019	0.179	-0.006	0.090	0.164
653.578	0.049	0.086	0.104	-0.022	0.0122	-0.222	-0.102	-0.110	-0.088	-0.047	0.065	-0.038	0.131	-0.012
519.095	-0.157	-0.191	0.650	-0.020	0.228	0.005	-0.054	-0.019	-0.137	-0.268	-0.063	-0.069	0.047	0.090
682.701	-0.199	-0.155	-2.458	0.471	-2.167	0.022	-0.015	0.105	-0.233	0.516	-0.124	0.494	-0.230	0.698
554.071	-0.859	-0.735	0.081	0.248	0.538	0.137	0.030	0.034	0.054	-0.040	-0.119	-0.053	-0.141	-0.431
625.633	-0.770	-0.664	2.228	-0.142	0.974	0.196	0.095	0.074	0.077	-0.851	0.022	-0.507	-0.030	-0.093
478.535	-0.429	-0.359	1.643	0.056	1.490	0.037	-0.001	-0.018	-0.095	-0.346	0.088	-0.373	0.068	-0.336

**Tabla B.2.** Tabla de 8 centroides generados con los coeficientes (*C0*, *C1*, ... *C14*) de la transformada Hermite y Cuantización Vectorial, de la zona de hielo compacto. Cada renglón representa un centroide.

<i>C0</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	<i>C9</i>	<i>C10</i>	<i>C11</i>	<i>C12</i>	<i>C13</i>	<i>C14</i>
200.9	0.098	0.034	-1.11	0.060	-1.037	-0.145	-0.025	-0.0009	0.149	0.387	-0.014	0.290	-0.030	0.387
148.6	-0.239	0.015	0.035	-0.024	0.034	0.058	-0.005	0.023	0.046	0.014	0.011	-0.019	0.019	0.011
175.0	-0.109	-0.04	0.231	-0.028	0.234	-0.049	0.022	-0.007	0.094	-0.066	0.007	-0.078	-0.0211	-0.055
121.9	-0.115	0.139	0.333	0.001	0.267	-0.002	-0.006	0.0003	0.018	-0.132	-0.004	-0.083	-0.003	-0.050
187.5	-0.218	0.115	-0.09	-0.017	-0.099	0.062	0.003	-0.010	0.019	0.033	0.005	-0.010	0.028	0.049
135.5	-0.203	0.116	0.043	0.0187	0.0318	0.026	0.014	-0.0008	0.0141	0.003	-0.0006	0.002	-0.015	-0.017
161.3	-0.312	0.113	-0.03	0.050	-0.093	0.066	-0.007	0.023	0.003	0.037	-0.037	0.021	-0.018	0.093
77.0	-0.248	0.131	1.076	-0.137	0.344	0.209	-0.041	0.0251	-0.061	-0.177	-0.035	-0.051	0.032	0.057

**Tabla B.3.** Tabla de 8 centroides generados con los coeficientes (*C0*, *C1*, ... *C14*) de la transformada Hermite y Cuantización Vectorial, de la zona de Mar. Cada renglón representa un centroide.

<i>C0</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	<i>C9</i>	<i>C10</i>	<i>C11</i>	<i>C12</i>	<i>C13</i>	<i>C14</i>
1553.14	-1.60	-5.4	-40.4	2.5	-22.96	-1.07	0.82	-1.28	0.51	10.83	-0.71	6.94	-0.015	5.83
916.15	15.21	-2.3	5.89	-1.1	3.12	-6.05	1.11	-2.28	0.21	-1.28	0.40	-0.49	0.031	-0.78
1212.9	6.25	-1.7	-2.16	0.84	-1.48	-1.11	0.006	-1.14	-0.66	0.304	0.04	0.24	-0.16	0.29
685.82	20.86	-11.3	19.65	0.9	12.99	-5.88	1.493	-1.48	1.34	-5.14	0.22	-4.02	-0.46	-2.30
1358.1	2.13	1.84	-6.41	1.4	-4.81	-0.11	-0.79	0.37	-1.01	0.97	-0.31	0.18	-0.24	0.857
982.9	-266.3	-24.1	-32.3	-6.6	-19.98	114.8	5.75	50.40	7.82	6.99	1.38	9.76	0.768	10.62
1072.8	5.40	-1.26	-0.46	0.74	-0.11	-1.74	-0.68	-0.64	-0.08	0.93	-0.86	0.78	-0.33	-0.06
424.03	-2.25	-0.64	29.8	-7.51	18.14	-3.66	-1.16	-0.82	-2.24	-4.50	2.36	-5.34	2.194	-3.91

**Tabla B.4.** Tabla de 8 centroides generados con los coeficientes (*C0*, *C1*, ... *C14*) de la transformada Hermite y Cuantización Vectorial, de la zona de Tierra. Cada renglón representa un centroide.

## C. Anexo: Código empleado para Cuantización Vectorial

```
1 function [r copia copial pp j u]=lbgimagencc(imma)
2 %% proyecto tratando la imagen con LBG punto a punto
3 aim= double(imma);
4 figure ,imshow(aim , [])
5 u=size(aim);
6
7 r=reshape(aim , [] ,1);
8 rr=mean(r);
9
10 %% para 'a' centroides
11 a=3;
12 tic;
13 for f=1:a
14
15 [copia]=perturbaimag(rr);
16
17 %% %Agrupacion por centroide perturbado
18 [copial err ]=agrupaimag(r , copia);
19
20 copia=copial;
21
22 %% reacomodo de los centroides
23 %% generando nuevos centroides de regiones sin perturbar
24 ees1=max(err);
25 j=1;
26 while ees1>=1 || j~=350 %% agrupa con mas de 2 centroides
27     [copial err pp]=agrupaimag(r , copial);
28     j=j+1;
29     ees1=max(err);
30
31 end
32 rr=copial;
33 end
34 toc
35
36 end
```

# Bibliografía

- [1] ACHANTA, R. ; SHAJI, A. ; SMITH, K. ; LUCCHI, A. ; FUA, P. ; SÜSSTRUNK, S.: "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012), Nov, Nr. 11, p. 2274–2282. – ISSN 0162–8828
- [2] AGENCY, European S.: "*Sentinel Online*". 2000-2018. – "[Web;accedido el 1 diciembre 2016]"
- [3] AHUJA, N. ; ROSENFELD, A.: "Mosaic Models for Textures". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-3 (1981), Jan, Nr. 1, p. 1–11. – ISSN 0162–8828
- [4] CROSS, G. R. ; JAIN, A. K.: "Markov Random Field Texture Models". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5 (1983), Jan, Nr. 1, p. 25–39. – ISSN 0162–8828
- [5] DAVIES, E.R.: "*Machine Vision: Theory, Algorithms and Practicalities*". "University of London", 2004. – 800 p.
- [6] DELVALLE, Toledo E.: "*Apuntes de introducción a la geofísica*". 1. UNAM, Facultad de Ingeniería, 2007
- [7] ESCALANTE, B. ; LOPEZ, A.: "*The Hermite Transform: An Efficient Tool for Noise Reduction and Image Fusion in Remote- Sensing*". CH Chen. Taylor and Francis Group, 2006. – 537–556 p.
- [8] GOODMAN, J. W.: "Some fundamental properties of speckle". En: *J. Opt. Soc. Am.* 66 (1976), Nov, Nr. 11, p. 1145–1150
- [9] HARALICK, R.M. ; SHAPIRO, L.G.: "Image Segmentation Techniques ". En: *Computer Vision Graphics and Image Processing* 29 (1985), Nr. 1, p. 100–132
- [10] HAWKINS, J.K.: "Textural Properties for Pattern Recognition". En: *Picture Processing and Psychopictorics* (1970), p. 281–296
- [11] JEFFRIES, Martin O. "*Sea Ice. Ice information*". 2006
- [12] KUAN, D. ; SAWCHUK, A. ; STRAND, T. ; CHAVEL, P.: "Adaptive restoration of images with speckle". En: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35 (1987), Mar, Nr. 3, p. 373–383. – ISSN 0096–3518

- 
- [13] LINDE, Y. ; BUZO ; GRAY: "An algorithm for vector quantizer design ". En: *IEEE Transactions on Communications* COM-28 (1980), p. 84–95
- [14] LLOYD, S.P.: "Least Squares Quantization in PCMs". En: *Bell Telephone Laboratories Paper, Murray Hill* (1957)
- [15] MACQUEEN: "Some Methods for Classifications and Analysis of Multivariate Observations". En: *Proc of Fifth Berkeley Symposium on Math, Stat and Prob* 1 (1967), p. 281–296
- [16] MARTENS, Jean B.: "The Hermite Transform - Applications". En: *IEEE Transactions on Acoustic, Speech and Signal Processing* 38 (1990), Nr. 9, p. 1607
- [17] MENDOZA, E. U.: "*Sea Ice Detection from Synthetic Aperture Radar Images*". Doctorado en Ingeniería. Protocolo de Tesis, UNAM, 2016
- [18] NASRABADI, M. N. ; KING, Robert A.: "Image Coding Using Vector Quantization: A review". En: *IEEE Transactions on Communications* 36 (1988), Nr. 8, p. 957–971
- [19] PASSARIELLO, G.: "*Imágenes Medicas: Adquisición, Análisis, Procesamiento e Interpretación*". "Universidad Simon Bolívar", 1995. – 45 p.
- [20] PETROU, M. ; GARCIA, S.P.: *Dealing with Texture*. 1. Wiley, 2006. – ISBN 978–0–470–02628–1
- [21] PICKETT, R.M.: "Visual Analysis of Texture in the Detection and Recognition of Objects". En: *Picture Processing and Psychopictorics* (1970), p. 289–308
- [22] PRATT, K. W.: "*Digital Image Processing*". 4. Wiley-Interscience, 2007. – 782 p.. – ISBN 978–0–471–76777–0
- [23] RUSS, C. J. ; NEAL, F. B.: "The Image Processing Handbook". En: *Taylor & Francis Group* (2016), p. 1013
- [24] SILVAN, Cardenas J. ; ESCALANTE, Ramirez B.: "The multiscale Hermite Transform for Local Orientation Analysis". En: *IEEE Transactions on Image Processing* 15 (2006), Nr. 5, p. 1236–1253
- [25] THEODORIDIS, S. ; KOUTROUSMAS, K.: "*Pattern Recognition*". 4. Elsevier, 2009. – ISBN 978–1–59749–272–0
- [26] TRIANA, V.P. G.: "*Detección y segmentación de hielo Frazil/Pancake basada en superpíxeles y la transformada de Hermite usando imágenes SAR*". Maestría en ingeniería Eléctrica. UNAM, 2017. – 55 p.
- [27] WADHAMS, et a.: "The use of SAR to Measure ocean waves dispersión in Frazil-Pancake icefields". En: *Journals of Physical oceanography* 32 (2002), p. 1721–1746

- 
- [28] WANG, L. ; GOLDBERG, M.: Progressive image transmission using vector quantization on images in pyramid form. En: *IEEE Transactions on Communications* 37 (1989), Dec, Nr. 12, p. 1339–1349. – ISSN 0090–6778