



**UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES**

**CENTRO DE CIENCIAS BÁSICAS**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

# **TESIS**

**ANÁLISIS DE IMÁGENES HIPERESPECTRALES USANDO  
CÓDIGOS DE CADENA**

**PRESENTA**

Israel Chávez Delgado

**PARA OBTENER EL GRADO EN**

**Maestro en Ciencias de la Computación**

**TUTOR**

Dr. Hermilo Sánchez Cruz

**INTEGRANTES DEL COMITÉ TUTORAL**

Dr. Juan Humberto Sossa Azuela

Dr. Martín Gerardo Rodríguez

Aguascalientes, Ags, 2 de mayo del 2018



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
FORMATO DE CARTA DE VOTO APROBATORIO

**JOSÉ DE JESÚS RUIZ GALLEGOS**  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

P R E S E N T E

Por medio del presente como Asesor designado del estudiante **ISRAEL CHÁVEZ DELGADO** con ID 141928 quien realizó la tesis titulada: **ANÁLISIS DE IMÁGENES HIPERESPECTRALES USANDO CÓDIGOS DE CADENA**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E  
"Se Lumen Proferre"  
Aguascalientes, Ags., a 29 de Mayo de 2018.



Hermilo Sánchez Cruz  
Director de tesis

c.c.p.- Interesado  
c.c.p.- Secretaría de Investigación y Posgrado  
c.c.p.- Jefatura del Depto. de Ciencias de la Computación  
c.c.p.- Consejero Académico  
c.c.p.- Minuta Secretario Técnico



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
FORMATO DE CARTA DE VOTO APROBATORIO

**JOSÉ DE JESÚS RUIZ GALLEGOS**  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

P R E S E N T E

Por medio del presente como Asesor designado del estudiante **ISRAEL CHÁVEZ DELGADO** con ID 141928 quien realizó la tesis titulada: **ANÁLISIS DE IMÁGENES HIPERESPECTRALES USANDO CÓDIGOS DE CADENA**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATE NTAMENTE  
"Se Lumen Proferre"  
Aguascalientes, Ags., a 29 de Mayo de 2018.

Juan Humberto Sossa Azuela  
Asesor de tesis

c.c.p.- Interesado  
c.c.p.- Secretaría de Investigación y Posgrado  
c.c.p.- Jefatura del Depto. de Ciencias de la Computación  
c.c.p.- Consejero Académico  
c.c.p.- Minuta Secretario Técnico



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
FORMATO DE CARTA DE VOTO APROBATORIO

**JOSÉ DE JESÚS RUIZ GALLEGOS**  
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

P R E S E N T E

Por medio del presente como Asesor designado del estudiante **ISRAEL CHÁVEZ DELGADO** con ID 141928 quien realizó la tesis titulada: **ANÁLISIS DE IMÁGNEES HIPERESPECTRALES USANDO CÓDIGOS DE CADENA**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E  
"Se Lumen Proferre"  
Aguascalientes, Ags., a 25 de MAYO de 2018.

Martín Gerardo Rodríguez  
Asesor de tesis

c.c.p.- Interesado  
c.c.p.- Secretaría de Investigación y Posgrado  
c.c.p.- Jefatura del Depto. de \_\_\_\_\_  
c.c.p.- Consejero Académico  
c.c.p.- Minuta Secretario Técnico





UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES

**ISRAEL CHÁVEZ DELGADO**  
**MAESTRÍA EN CIENCIAS CON OPCIÓN A LA COMPUTACIÓN Y**  
**MATEMÁTICAS APLICADAS**  
**PRESENTE.**

Estimado alumno:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **“ANÁLISIS DE IMÁGENES HIPERESPECTRALES USANDO CÓDIGOS DE CADENA”**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

**ATENTAMENTE**

Aguascalientes, Ags., a 29 de mayo de 2018

*“Se lumen proferre”*

**EL DECANO**

A handwritten signature in black ink, appearing to read 'José de Jesús Ruíz Gallegos'.

**M. en C. JOSÉ DE JESÚS RUÍZ GALLEGOS**

c.c.p.- Archivo.

## **Agradecimientos**

A mi institución, la Universidad Autónoma de Aguascalientes, por haberme permitido formarme en ella y haberme dado los grandes valores con los que cuenta. Al Consejo Nacional de Ciencia y Tecnología (Conacyt), por brindarme el apoyo económico para dedicarme por completo a mi maestría.

A mis asesores, el Dr. Martín Gerardo Rodríguez y el Dr. Juan Humberto Sossa Azuela, por su valiosa asesoría y enseñanza sobre los diferentes temas utilizados para la realización de esta tesis. A mi Director de Tesis, el Dr. Hermilo Sánchez Cruz, por su paciencia, cuando no se lograban los objetivos; por su dedicación, al siempre haber tenido tiempo ayudar y para resolver dudas; y principalmente, por su motivación, guía y aliento, con las cuales se obtuvo una participación en un congreso celebrado en Suiza.

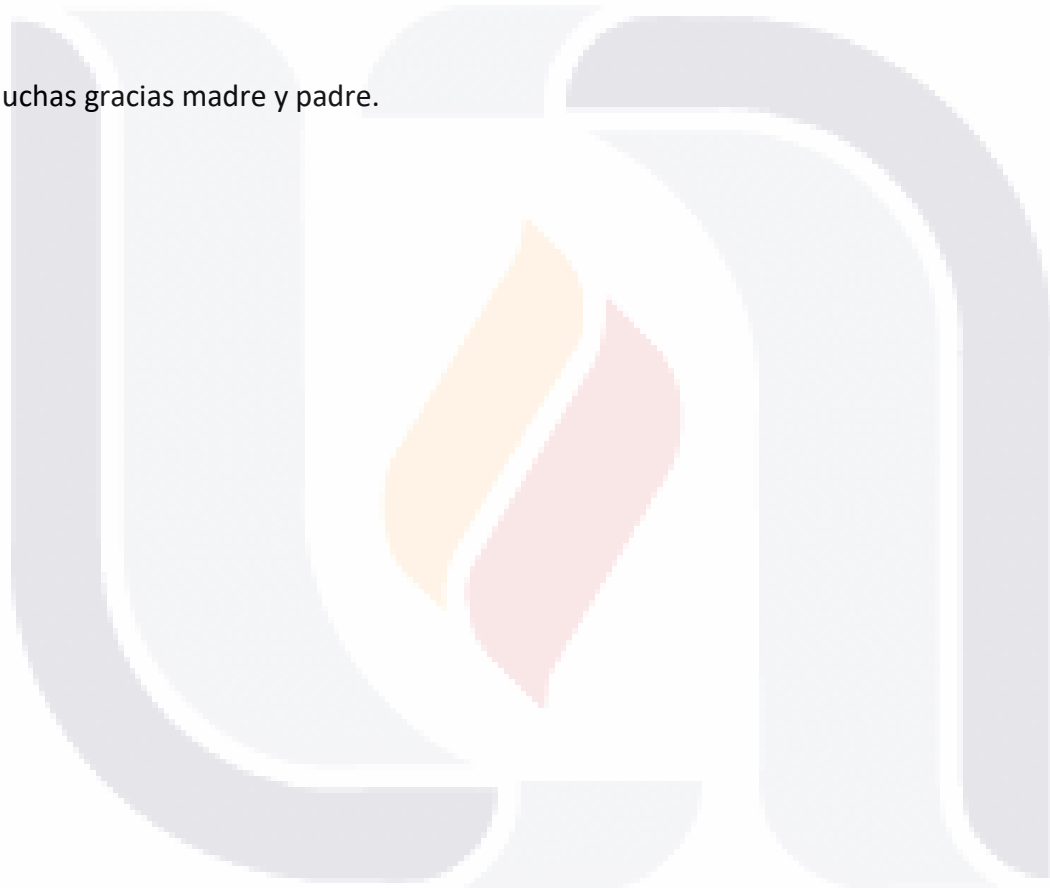
A mi familia, por el apoyo incondicional en cada decisión que he tomado, por los consejos, comprensión y ayuda en aquellos momentos donde las cosas se han puesto difíciles y por siempre haber confiado en mí y haberme enseñado a confiar en mí mismo para nunca rendirme y poder lograr mis objetivos.

A Dios por la vida, por las alegrías, tristezas, bendiciones y todas las grandes experiencias que me ha dejado vivir. Y principalmente por permitirme tener y disfrutar a mi familia. Ya que sin todo esto no hubiera podido concluir con satisfacción un ciclo más de estudios.

## Dedicatoria

Dedico esta tesis a mis padres, quienes con sus sacrificios, esfuerzos y enseñanzas me hicieron la persona que soy en la actualidad. Todos mis logros son gracias a ustedes, ya que siempre me han motivado a dar lo mejor de mí, enseñándome a no confiarme y seguir luchando por nuevas metas.

Muchas gracias madre y padre.



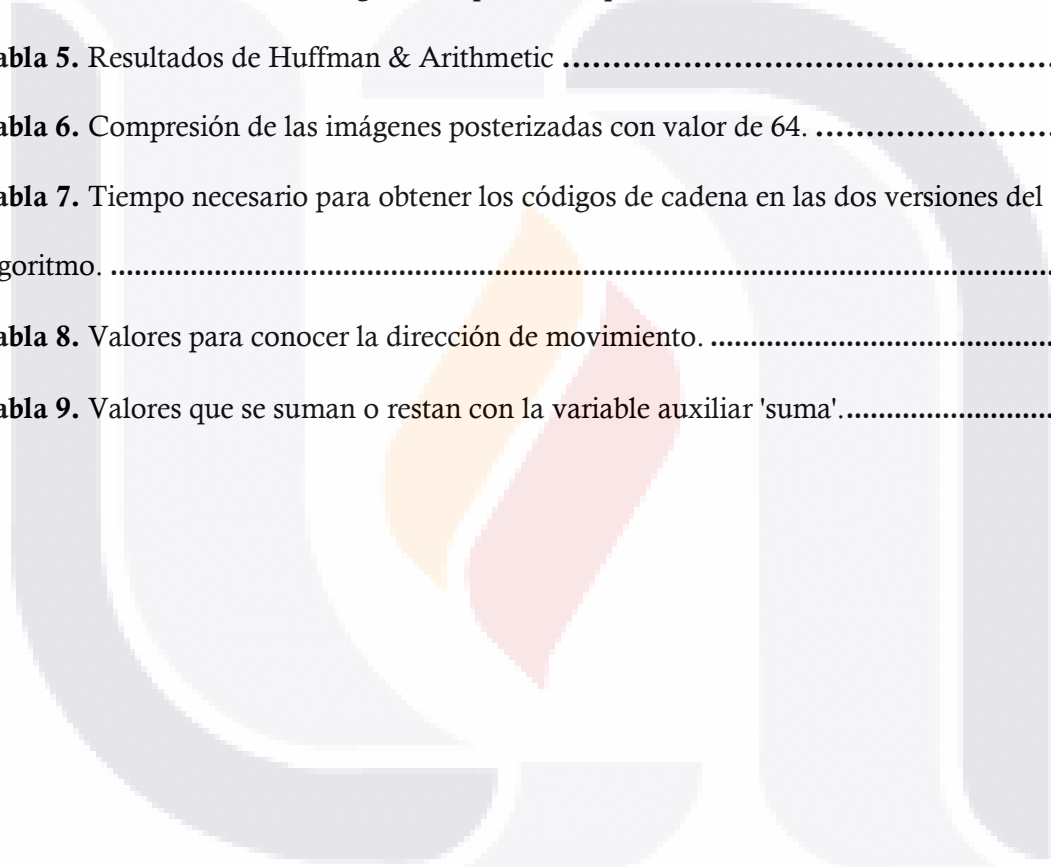
# Índice de contenidos

<b>I. Motivaciones y objetivos</b> .....	<b>11</b>
1.1 Motivaciones.....	11
1.2 Objetivos.....	11
<b>II. Introducción</b> .....	<b>12</b>
<b>III. Planteamiento del problema</b> .....	<b>16</b>
3.1 Antecedentes.....	16
3.2 Diagnóstico.....	19
3.2.1 Definición de los problemas.....	19
3.2.2 Descripción de los problemas.....	20
3.2.3 Evaluación de los problemas.....	21
3.2.4 Conclusiones y recomendaciones.....	22
3.3 Justificación.....	22
<b>IV. Fundamentación teórica</b> .....	<b>23</b>
<b>V. Metodología</b> .....	<b>31</b>
5.1 Diseño de la intervención o aplicación.....	31
5.2 Reducción de ruido.....	36
5.3 Compresión de códigos de cadena.....	43
5.4 Búsqueda de zonas de interés y clasificación por morfología.....	45
5.5 Software utilizado y explicación de los algoritmos desarrollados.....	50
<b>VI. Resultados</b> .....	<b>52</b>
<b>VII. Conclusiones</b> .....	<b>60</b>
<b>VIII. Futuros trabajos</b> .....	<b>62</b>
<b>IX. Referencias</b> .....	<b>63</b>



# Índice de tablas

<b>Tabla 1.</b> Dirección de movimiento de los 11 símbolos utilizados del código F26. ....	<b>28</b>
<b>Tabla 2.</b> Matriz de equivalencia de F8 a AF8. ....	<b>34</b>
<b>Tabla 3.</b> Resultados de las imágenes después de aplicar los filtros. ....	<b>37</b>
<b>Tabla 4.</b> Resultados de las imágenes después de la posterización. ....	<b>41</b>
<b>Tabla 5.</b> Resultados de Huffman & Arithmetic .....	<b>44</b>
<b>Tabla 6.</b> Compresión de las imágenes posterizadas con valor de 64. ....	<b>44</b>
<b>Tabla 7.</b> Tiempo necesario para obtener los códigos de cadena en las dos versiones del algoritmo. ....	<b>47</b>
<b>Tabla 8.</b> Valores para conocer la dirección de movimiento. ....	<b>49</b>
<b>Tabla 9.</b> Valores que se suman o restan con la variable auxiliar 'suma'.....	<b>50</b>



# Índice de ilustraciones

<b>Ilustración 1.</b> Las ocho direcciones del código de Freeman .....	27
<b>Ilustración 2.</b> Las 26 direcciones del código de Freeman.....	27
<b>Ilustración 3.</b> Las ocho direcciones del código AF8.....	29
<b>Ilustración 4.</b> Obtención de códigos de cadena en orden zig-zag horizontal. ....	32
<b>Ilustración 5.</b> Ejemplos de obtención de códigos mediante las intensidades de un renglón par e impar. ....	33
<b>Ilustración 6.</b> Forma completa creada en AutoCAD mediante polilíneas obtenidas del código F26.....	35
<b>Ilustración 7.</b> Figura vista desde arriba. ....	35
<b>Ilustración 8.</b> Apreciación del movimiento de los símbolos de los códigos de cadena. ....	36
<b>Ilustración 9.</b> Polilíneas obtenidas antes de aplicar posterización.....	39
<b>Ilustración 10.</b> Polilíneas obtenidas después de posterizar .....	39
<b>Ilustración 11.</b> Polilíneas obtenidas después de posterizar en 64. ....	40
<b>Ilustración 12.</b> Porcentaje de error al utilizar Posterización .....	43
<b>Ilustración 13.</b> Valores posibles en el primer símbolo del renglón. ....	48
<b>Ilustración 14.</b> Valores posibles en los siguientes símbolos. ....	48
<b>Ilustración 15.</b> Ángulos de movimiento correspondientes a los símbolos de AF8.....	49
<b>Ilustración 16.</b> Frecuencia de símbolos F8 antes y después de posterizar.....	53
<b>Ilustración 17.</b> Frecuencia de símbolos F26 antes y después de posterizar. ....	54
<b>Ilustración 18.</b> Frecuencia de símbolos AF8 antes y después de posterizar. ....	55
<b>Ilustración 19.</b> Resultados obtenidos en la imagen espectral de luz azul. ....	57

**Ilustración 20.** Resultados obtenidos en la imagen espectral de luz verde. .... 57

**Ilustración 21.** Resultados obtenidos en la imagen espectral de luz ultravioleta. .... 58

**Ilustración 22.** Resultados obtenidos en la imagen espectral de luz violeta. .... 59



## Resumen

Con el análisis de imágenes hiperespectrales se puede obtener la información necesaria para poder hacer una clasificación de los objetos encontrados en el cubo hiperespectral, en el que los ejes 'X' y 'Y' están ocupados por la componente posicional, mientras que el eje 'Z' está ocupado por la longitud de onda. Usando la longitud de onda de los pixeles se puede hacer una clasificación exacta, ya que una de las propiedades de las imágenes hiperespectrales es la cantidad de imágenes espectrales que contienen, las cuales van desde la luz ultravioleta hasta el infrarrojo, de esta forma los pixeles del cubo hiperespectral tendrán una firma espectral única, esta firma está compuesta por los valores de longitud de onda de cada imagen espectral, por lo que aquellos pixeles con una firma espectral igual nos dice que estos pixeles pertenecen a un mismo objeto.

Aunque hay que comprender que este análisis no dirá que objetos hay en la imagen, para esto es necesario la ayuda de un experto el cual dirá que zonas pertenecen a que objeto. El principal problema al trabajar con imágenes hiperespectrales es el precio de las herramientas para crear el cubo hiperespectral y hacer su respectivo análisis. Para esto se propone trabajar de una manera diferente para encontrar las zonas de interés y clasificarlas, esto es mediante los códigos de cadena. Aprovechando los estudios previos sobre reconocimiento de patrones en imágenes binarias en 2D, se puede trabajar de la misma manera en imágenes en tres dimensiones, estas imágenes serán en 3D ya que se analizaran los patrones en los movimientos de los códigos de cadena al recorrer las intensidades de los pixeles. Al analizar los patrones es posible reconocer que zonas son iguales a otras, por lo que de esta forma se estará haciendo una aproximación a la clasificación real, ya que se debe de tener en cuenta que no es posible hacer una clasificación exactamente igual a la que se obtendría con la longitud de onda.

## **Abstract**

Hyperspectral Remote Sensing (HRS) it was initially used mainly to analyze Earth's surface and for military actions. Over time, many techniques has been created by public and private companies, because of the facilities HRS gives to analyze not just cell grounds images taken by satellites or aircrafts. But there is a problem when we want to use this technique, and is the high costs of the hyperspectral sensors required to do the analysis.

So, to solve this we need to use other techniques that does not require the use of these sensors. For this, we used chain codes to use their information to search and analyze patterns obtained of the grayscale image intensities. Doing this, we have the possibility to count and classify zones easily.

The chain codes of the intensities give us the visual information of the image with no loss and in a 2D shape. If we mix the vertical and horizontal 2D shapes we obtain the morphology of the pixels reflectance, to summarize we obtain a 3D shape. So, first we must find the zones depending on the intensity values (2D shapes) and then we classify that zones depending on their morphology, using the 3D shapes.

Like every method, this one has its advantages and disadvantages. With this we can only know the intensity value of the reflectance of each pixel, so we cannot know properties that we can obtain with wavelength. But, if we only want to classify zones by their reflectance, we can do it easily using chain codes.

It is important to mention that this method (or a similar) that was not use before. The results obtained in this work may not be the best to work in the practice, but in the next years and doing more experiments will improve a lot.



1. **Motivaciones y objetivos.** Respuesta a las preguntas ¿Por qué se empezó a trabajar con esta investigación? Y ¿Qué es lo que se quiere lograr al finalizar el trabajo?
2. **Introducción.** Breve explicación de lo que se planea hacer junto con la información de las herramientas y técnicas con las que se trabajará a lo largo del documento.
3. **Planteamiento de la problemática a atender a través del trabajo práctico.** Principales problemas encontrados al trabajar en el análisis de imágenes hiperespectrales así como las posibles y mejores soluciones. Dentro de este capítulo se encuentran los siguientes bloques.
  - 3.1. **Antecedentes.** Se describe el estado del arte de las técnicas de análisis de imágenes hiperespectrales y de códigos de cadena, explicación de las fases principales y esenciales en la investigación.
  - 3.2. **Diagnóstico.** Se presentan los principales problemas encontrados al empezar a analizar los diferentes aspectos del sistema, junto con una explicación de sus posibles soluciones.
  - 3.3. **Justificación.** La clasificación y análisis de contaminantes en muestras microscópicas es de gran ayuda para el experto, ya que le da información necesaria para conocer el entorno en el que estas muestras han estado, y a que químicos reaccionan cada uno de los contaminantes, por lo que debe existir un método que sea flexible tanto en facilidad de uso como en costo.
4. **Fundamentación teórica.**
5. **Metodología.** Explicación del paso a paso del desarrollo del sistema, tanto de investigaciones, problemas encontrados, diferentes versiones de los algoritmos, entre otros.
6. **Resultados.** Se muestran los resultados obtenidos al aplicar los métodos de cada uno de los códigos de cadena utilizados, señalando las ventajas y desventajas de estos códigos así como cuál otorga mejores resultados.
7. **Conclusiones.** Resumen acerca de los resultados obtenidos, explicando que variables son mejores para los experimentos.
8. **Futuros trabajos.** Se mencionan los diferentes proyectos con los que se mejorará el sistema y se obtendrán mejores resultados en futuros trabajos.

# TESIS TESIS TESIS TESIS TESIS

## Capítulo 1. Motivaciones y objetivos

### 1.1 Motivaciones

El análisis de propiedades en imágenes hiperespectrales de muestras microscópicas es de gran ayuda, ya que se puede obtener información de gran interés como los es: los diferentes materiales dentro de la muestra junto con su exacta localización, la clasificación de estos materiales y su cuantificación.

En las técnicas clásicas de análisis de estas imágenes es necesario contar con sensores especiales para obtener la información con la que estas trabajan. Estos sensores son de un alto costo por lo que no están al alcance de cualquier individuo.

Por esta razón, es necesaria alguna forma de aproximar los resultados obtenidos con las técnicas clásicas. En este trabajo, se hace una explicación detallada de cómo hacer una clasificación de contaminantes en muestras microscópicas.

Conviene destacar que la forma en cómo se está trabajando es nueva y no se ha utilizado anteriormente, por lo que no hay datos en la literatura con lo que se pueda comparar. Por consiguiente, el tiempo necesario para desarrollar la metodología empleada en este trabajo fue mayor de como se esperaba.

### 1.2 Objetivos

Con este trabajo se pretende trabajar con la morfología de las intensidades de las imágenes hiperespectrales. Creando nuevas técnicas para el análisis y clasificación de sus propiedades, sin la necesidad de tener que adquirir equipo costoso. Teniendo esto en cuenta se plantea el siguiente objetivo global: **diseñar y desarrollar un nuevo método para el análisis y clasificación de propiedades en imágenes hiperespectrales, el cual deberá de trabajar solo con las intensidades de las imágenes.**

Este objetivo general se puede lograr mediante una serie de objetivos específicos, los cuales los mencionamos a continuación:

- ✓ Eliminar el ruido de las imágenes espectrales de la muestra, y comprimir la cantidad de información de las mismas, todo esto sin perder información visual de las imágenes.
- ✓ Desarrollar un método para identificar zonas de interés (contaminantes) en las distintas imágenes espectrales usando códigos de cadenas.
- ✓ Desarrollar un método para clasificar los contaminantes de acuerdo a su morfología.

## **Capítulo 2. Introducción**

Las finalidades del análisis de imágenes son la cuantificación de propiedades, reconocimiento de patrones, entre otras. Todo esto inició gracias a las facilidades que tiene el cerebro, ya que el ser humano está rodeado de imágenes captadas por el ojo, estas son a colores y en blanco y negro. Y gracias al procesamiento con el que cuenta el cerebro podemos ser capaces de reconocer objetos con solo verlos un momento, sin importar a posición en la que se encuentren, el color que tengan o incluso si el objeto está incompleto, nuestro cerebro es capaz de asociar la forma del objeto e instantáneamente decir que objeto es. Esta facilidad con la que trabaja el cerebro llevó a pensar a los desarrolladores e investigadores de inteligencia artificial, que hacer una computadora o sistema capaz de analizar e interpretar las imágenes iba a ser un trabajo relativamente fácil, sin embargo, no fue así, y aún tras años de investigación sigue siendo un problema muy complejo.

Para entender los procesos antes mencionados, se hace uso de la visión computacional, la cual nos ayuda a construir máquinas con capacidades similares a la visión con la que contamos la mayor parte de los seres vivos [1].

A lo largo del tiempo han existido varias definiciones de visión, entre éstas podemos mencionar las siguientes.

- “Visión es saber que hay y dónde mediante la vista” (Aristóteles, s.f.).
- “Visión es recuperar de la información de los sentidos (vista) propiedades válidas del mundo exterior”, Gibson [2].
- “Visión es un proceso que produce a partir de las imágenes del mundo exterior una descripción que es útil para el observador y que no tiene información irrelevante”, Marr [3].

Aunque las tres son válidas, la más cercana a la definición que se maneja actualmente sobre la visión computacional, es la definición de Marr, la cual cuenta con tres aspectos importantes que debemos tener presentes [1].

1. Visión es un proceso computacional
2. La descripción a obtener depende del observador.
3. Es necesario eliminar la información que no sea útil

El procesamiento de imágenes es un área ligada a la visión computacional, aunque estas dos suelen ser muy parecidas, sus objetivos finales son diferentes. Mientras que en la visión su objetivo principal es el reconocimiento y localización de objetos, el objetivo del procesamiento de imágenes viene siendo el de mejorar la calidad de las imágenes para ser utilizadas posteriormente [1].

Los problemas que se presentan cuando desea realizar un análisis de imágenes hiperespectrales son los siguientes:

- Alto precio de los sensores y cámaras requeridas.
- Gran número de imágenes para analizar y diferencia de valores de longitud de onda.

Para analizar las imágenes hiperespectrales es necesario contar con cámaras y sensores especiales para crear primeramente el cubo hiperespectral, estos sensores adquieren imágenes en una gran cantidad de bandas espectrales contiguas para habilitar la extracción de la reflectancia espectral de una escala de pixel que puede ser

directamente comparada con una medida espectral similar. Se le han dado diferentes nombres a este campo de teledetección, entre los cuales se incluyen “Espectrometría de imágenes (*Imaging spectrometry, imaging spectroscopy*)” y “Teledetección hiperespectral (*hyperspectral remote sensing*)”. Aunque tienen diferentes significados en una traducción directa del termino (i.e. *spectrometry* = “medición”, *spectroscopy* = “visión”, *hyperspectral* = “demasiadas bandas”), el significado y percepción de la comunidad de teledetección es la misma: “la obtención de imágenes en cientos y contiguas bandas espectrales, tal que por cada elemento de la imagen es posible derivar un espectro de reflectancia completo”.

Una vez obtenido este cubo, es fácil clasificar las diferentes zonas de la imagen, ya que cada pixel cuenta con una firma única creada por diferentes valores de longitud de onda. Esta firma sirve para obtener los diferentes clústeres que se encuentran en la imagen [4].

Por lo que si se cuenta con estas cámaras y sensores, simplemente no se podrá generar el cubo hiperespectral y no se contará con la firma única de cada pixel para poder hacer la clasificación deseada.

Es posible generar este cubo hiperespectral, sin contar con las cámaras y sensores necesarios. Con la ayuda de un espectrómetro se puede obtener la longitud de onda de cada pixel de la imagen, aunque al hacer esto tendrá que enfrentar con dos problemas graves, los cuales son:

1. El espectrómetro nos dará diferentes datos dependiendo de la pantalla donde se desea usarlo, ya que depende el brillo y el contraste de la pantalla que se esté utilizando, depende también de programa donde se abran las imágenes, aunque uno no se dé cuenta, puede variar la información del pixel en ciertos programas y por último y más importante la calidad y formato de la imagen.
2. Generar la base de datos de la relación RGB – Longitud de Onda. El principal problema de esto es que los monitores solo cuentan con pixeles con colores RGB (Red, Green, Blue) por lo que no se puede conocer los valores de los espectros no visibles como ultravioleta y/o hiperespectrales.



Como se puede observar no es recomendable trabajar de esta forma para el análisis de las imágenes espectrales, ya que se necesitaría conocer previamente las características del monitor que se va a utilizar, y en el caso de que se cambie el monitor, se necesitaría cambiar los valores de relación RGB – Longitud de onda, aunque se debe tomar en cuenta que las aproximaciones dadas por este proceso son buenas contando el gasto realizado.

Analizando junto con el especialista los problemas encontrados, se pudo observar que el objetivo principal que se buscaba no era hacer la clasificación completa como lo hacen los sistemas de análisis comerciales. Haciendo estudios sobre los contaminantes se observó que estos se diferencian de los demás por su valor de reflectancia de la luz aplicada sobre las muestras, entonces, esto nos dice que la zona donde se encuentra un mismo contaminante tendría valores de reflectancia casi iguales, por lo que los grupos de intensidad de reflectancia en las diferentes zonas de la imagen nos estarían diciendo cuales zonas comparten propiedades similares y cuales son completamente diferentes. Por lo que los valores RGB así como la longitud de onda de los pixeles de las imágenes no son de gran importancia ya que lo que se quiere buscar y clasificar son las zonas con mayor reflectancia al promedio.

Para hacer esto se utilizaron códigos de cadena, ya que se quería conocer limitantes o más utilidades de estos mismos. Como no hay trabajos previos que analicen la reflectancia de este tipo de imágenes utilizando los códigos de cadena, no se conocen que códigos dan mejores resultados que otros, tanto en tiempo como en facilidad de procesamiento por lo que se utilizaron los códigos de cadena básicos para observar que resultados se obtienen con estos.

Con los códigos de cadena se puede obtener la información requerida sin ocupar mucho espacio en memoria. Usualmente los códigos de cadena son utilizados para representar el contorno de las imágenes binarias. Estas imágenes son en 2D, mientras que en el proceso que se desea realizar es en forma tridimensional, ya que se tendrá que mover directamente en las intensidades de los pixeles. Como las zonas de interés tienen

una intensidad mayor, estas contarían con patrones específicos los cuales se necesitarían estudiar para clasificar todos los patrones posibles en las imágenes.

En la actualidad existen diferentes códigos de cadena, e incluso existen algunos llamados Crack Codes compuestos por F4 [5], 3OT [6] y VCC [7], pero como antes se mencionó, para una primera aproximación se hará uso de los códigos de cadena F8 [5], F26 [8] y AF8 [9] porque son los adecuados para representar imágenes en escala de grises y son más versátiles a mantener la información de las formas dadas por las intensidades cuando se visitan los píxeles tanto en posición como en intensidad. Estos códigos de cadena son más ricos en simbología e información, dando la facilidad a encontrar características y patrones, a diferencia de los Crack Codes que son ortogonales.

## **Capítulo 3. Planteamiento del problema**

### **3.1 Antecedentes**

Para hacer una explicación del estado del arte en cuanto a análisis de imágenes hiperespectrales usando códigos de cadena, separaremos esto en dos partes, las cuales serían: análisis de imágenes hiperespectrales y códigos de cadena.

Los espectros de reflectancia han sido utilizados por muchos años para obtener información composicional de la superficie de la tierra. La reflectancia espectral en un visible y cercano infrarrojo ofrece una técnica rápida y económica para determinar la mineralogía de las muestras y obtener información acerca de la composición química. La transición electrónica y el proceso de transferencia de carga asociado con la transición de los iones de metal determinan en gran parte la posición de las características de absorción para diagnóstico en una longitud de onda de un visible y cercano infrarrojo de una región de los espectros de los minerales [11]. La posición, forma, profundidad, anchura y asimetría de estas características de absorción están

controladas por una particular estructura de cristal en donde las especies absorbidas están contenidas por una estructura química del mineral. Por lo tanto, las variables que caracterizan la absorción pueden estar directamente relacionadas a la mineralogía de la muestra [12].

Los espectrómetros han sido por un largo tiempo para aplicaciones militares tales como la detección de camuflaje en vegetación real. Debido a la naturaleza de clasificación de la información y los sensores, no se puede mencionar mucho acerca del origen y las aplicaciones en las que se han utilizado.

El primer escáner/espectrómetro de imágenes fue *Scanning Imaging Spectroradiometer (SIS)* construido a principios de los setentas por el centro espacial Johnson de la NASA. Después de eso, los datos del espectrómetro aéreo fueron obtenidos en 1981 usando un espectrómetro de perfil unidimensional desarrollado por *Geophysical Environmental Research Company* quienes obtuvieron la información en 576 canales cubriendo el rango de longitudes de onda de 0.4-2.5  $\mu\text{m}$  seguido por *Shuttle Multispectral Infrared Radiometer (SMIRR)* en 1981 [13]. El *Airborne Imaging Spectrometer (AIS)*, fue desarrollado por *NASA Jet Propulsion Laboratory* el cual empezó a operar en 1983, obteniendo 128 bandas espectrales en el rango de 1.2-2.4  $\mu\text{m}$ , y contaba con un rango de vista de 3.7 grados, dando como resultado imágenes de 32 bits. Una siguiente versión *AIS-2*, cubría una región de 0.8-2.4  $\mu\text{m}$  y además obtenía imágenes de 64 bits. Desde 1987, la Nasa opera el sucesor de los sistemas *AIS*, el cuál es llamado *Airbone Visible/Infrared Imaging Spectrometer (AVIRIS)* [14]. Desde ese tiempo muchas compañías privadas también empezaron a tomar parte en el rápido desarrollo de la espectrometría.

Los datos por espectrografía se pueden obtener usando las técnicas de *whiskbroom*, *pushbroom*. Los lectores de imágenes que trabajan con la técnica *whiskbroom* son escáneres electromecánicos. Los lentes ópticos barren de un borde de la región de interés hasta el otro. El campo de visión del escáner puede ser detectado por un solo detector o un detector de una sola línea. Simultáneamente el movimiento del satélite o el avión garantiza el escaneo de barrido sobre la tierra. Esto quiere decir que el tiempo por cada celda de tierra debe ser muy corto para un campo de visión instantáneo (IFOV

por su siglas en inglés) porque cada línea escaneada consiste en múltiples celdas de tierra que serán detectadas. En *whiskbroom* cada pixel es visto por separado lo que permite un amplio campo de visión y solo un detector en cada banda espectral tiene que ser calibrado. La desventaja es el corto tiempo, lo que limita la resolución espacial y espectral como también el ruido de las imágenes [4].

Los escáneres que usan la técnica *pushbroom* son escáneres electrónicos que usan una línea de detectores para escanear sobre una escena bidimensional. El número de píxeles es igual al número de celdas de tierra para una franja determinada. El movimiento del avión o el satélite provee el escaneo a lo largo de la dirección, por lo tanto, el inverso de la línea de frecuencia es igual al tiempo de permanencia del píxel. Usando un detector bidimensional, una dimensión puede representar el ancho de franja (dimensión espacial, 'Y') y la otra el rango espectral [4].

Por cada canal espectral que el sensor obtiene, la radiancia es variable. Mejor dicho un canal detecta solo fotones de una longitud de onda particular el canal mide la radiancia en un rango de longitud de onda que se estrecha unos pocos nanómetros más bajos y unos pocos nanómetros más altos que la longitud de onda del canal central. La curva que describe el declive de los niveles de radiancia alrededor de la longitud de onda del canal central por cada canal se conoce como función de respuesta espectral (SRF por sus siglas en inglés) el cual se puede deducir con mediciones monocromáticas.

En resumen a lo largo de la historia, las técnicas del análisis de datos de imágenes espectrales han seguido varias etapas que dependen de la evolución en los instrumentos de observación remota [15]. En un principio, estos medios de observación remota se caracterizaban por estar montados sobre plataformas exclusivamente espaciales y por esta razón las técnicas de análisis derivadas fueron basadas en enfoques fundamentalmente espaciales. En estos tiempos, existen instrumentos que dan la capacidad de medir singularidades en el espectro de la luz reflejada por los diferentes materiales presentes en el mundo real, y como consecuencia se iniciaron las técnicas basadas en espectroscopia.

Hoy en día, hay una gran cantidad de instrumentos o sensores capaces de medir singularidades espectrales utilizando diferentes longitudes de onda [16]. Esta disponibilidad de estos nuevos instrumentos dio la facilidad de una redefinición del concepto de imagen digital a través de la extensión de la idea de pixel. Como se sabe, un pixel está constituido por un único valor discreto, pero en imágenes hiperespectrales, un pixel consta de un conjunto de valores, los cuales pueden ser entendidos como vectores N-dimensionales [17], donde N representa el número de bandas espectrales en las que el sensor mide información.

### **3.2 Diagnóstico**

En esta etapa del desarrollo se desea evaluar la situación actual en que se encuentra la investigación, los resultados obtenidos ayudarán a evaluar implementaciones necesarias para mejorar o automatizar el proceso.

Algunas de las consultas que se desean implementar, por medio de preguntas al experto son:

1. ¿Se cuenta con sistemas o aplicaciones para obtener información de las imágenes hiperespectrales?
2. ¿Es necesario algún pre-procesamiento de las imágenes obtenidas?
3. ¿Qué información se desea obtener del análisis de imágenes?

#### **3.2.1 Definición de los problemas**

Los problemas que se encontraron al tratar de trabajar con imágenes hiperespectrales sin sensores hiperespectrales son:

1. Calidad y ruido en las imágenes obtenidas por microscopio.
2. Gran cantidad de imágenes espectrales para analizar.



3. Poca o nula información de métodos alternativos de análisis de imágenes hiperespectrales.

### 3.2.2 Descripción de los problemas

#### Problema 1.- Calidad y ruido en las imágenes obtenidas por microscopio.

##### Causa

- Herramientas utilizadas.

##### Problema

- Ruido generado por lentes o portaobjetos contaminados o por algún otro factor que no se puede controlar.

##### Efecto

- Necesidad de eliminar el, ruido generando tiempo extra en el sistema.

#### Problema 2.- Gran cantidad de imágenes espectrales para analizar

##### Causa

- Alto costo de los sensores hiperespectrales.
- Falta de recursos.

##### Problema

- No se puede crear un cubo hiperespectral que contenga longitudes de onda.

##### Efecto

- Alta cantidad de imágenes que se tienen que obtener.
- Demasiado tiempo para obtener las imágenes espectrales.

**Problema 3.- Poca o nula información de métodos alternativos de análisis de imágenes hiperespectrales**

**Causa**

- Los métodos que utilizan sensores hiperespectrales generan resultados exactos, ya que trabajan con longitudes de onda.

**Problema**

- Al querer utilizar otra característica aparte de la longitud de onda, no es posible con los métodos comerciales.

**Efecto**

- Investigar y experimentar con varios métodos que puedan funcionar, lo cual genera tiempo extra en cada método experimentado.
- No hay información con la cual se pueda comparar los resultados obtenidos.

**3.2.3 Evaluación de los problemas**

Al no contar con las herramientas necesarias para realizar el análisis de este tipo de imágenes, el primer problema que se genera es la cantidad de imágenes espectrales que vamos a obtener. Se debe comprender que este problema no es posible eliminarlo, ya que no hay otra forma de crear el cubo hiperespectral, por lo que se tiene que encontrar una forma de minimizar lo más posible este problema.

En un principio se pueden determinar las siguientes soluciones:

- Limitar el número de imágenes espectrales obtenidas.
- Optimizar al máximo el algoritmo de análisis.

Al platicar con el experto sobre este problema, se llegó a la conclusión de eliminar la primera solución, ya que para hacer un óptimo análisis es necesario contar con el mayor número de imágenes espectrales. Por lo que nos optó por continuar con la segunda

TESIS TESIS TESIS TESIS TESIS

solución, para esto es necesario analizar el algoritmo, para observar cuales aspectos se pueden mejorar para obtener los resultados en el menor tiempo posible, esto se estudiara más adelante cuando se llegue al proceso de desarrollo, sin embargo algunas posibles soluciones es eliminar la información innecesaria, y emplear lo más posible procesos multitareas.

De los dos problemas restantes, solo a uno de ellos se le puede dar solución. La calidad y ruido de las imágenes depende tanto del microscopio como de la cámara utilizada, pero aun así es posible eliminar el ruido generado en estas. Existen muchos métodos para eliminar ruido en las imágenes, por lo que es cuestión de probar con todos o varios de estos métodos, para observar cuál de ellos otorgan mejores resultados.

#### **3.2.4 Conclusiones y recomendaciones**

En base a las observaciones del diagnóstico y el levantamiento de información se concluye lo siguiente:

- No se puede disminuir el tiempo de la obtención de las imágenes espectrales.
- Se recomienda guardar las imágenes en la calidad más alta posible, para de esta forma poder reconocer que pertenece y que no pertenece a la muestra.
- Es importante eliminar la información innecesaria en cada una de las imágenes, para disminuir el tiempo de procesamiento en cada una de ellas.
- Emplear procesos multitarea para reducir el tiempo de procesamiento.

### **3.3 Justificación**

El análisis de imágenes hiperespectrales es un área de estudio muy amplia, la cual se puede utilizar desde microscopia hasta imágenes geo-satelitales. Existen varias complicaciones cuando se desea realizar este tipo de análisis en muestras de microscopio, pero la principal es el alto costo de las herramientas necesarias para

realizarlo, como cámaras y sensores hiperspectrales. Ya que por lo regular sus costos son algo altos para el usuario promedio, lo cual es un gran impedimento para investigadores que no cuenten con los recursos necesarios para esto.

Por esta razón debe existir un método el cual permita al investigador realizar una clasificación y análisis de los contaminantes existentes en muestras microscópicas, de esta forma se puede conocer mediante una serie de experimentos a que químicos reaccionan los contaminantes, lo que daría información acerca de las propiedades de estos y posiblemente su nombre.

#### **Capítulo 4. Fundamentación teórica**

Para obtener las imágenes espectrales donde se observen las características de reflectancia y absorbancia de la luz en las muestras es necesario aplicar técnicas de microscopia. Existen varias técnicas de microscopia de luz en las que se incluyen *bright-field*, *dark-field*, *fluorescence*, y *phase contrast microscopy*. Aunque cada método tiene aplicaciones específicas y ventajas, *bright-field microscopy* es el más comúnmente usado en clases introductorias y laboratorios clínicos. Muchas aplicaciones de investigación usan *electron microscopy* por su habilidad para producir imágenes de más alta calidad [18].

*Bright-field microscopy* produce una imagen hecha de luz que es transmitida por medio de una muestra. La muestra restringe la transmisión de la luz y aparece sombreada contra un fondo brillante. El material se observa sin coloración. La luz pasa directamente y se aprecian detalles que estén naturalmente coloreados [19].

*Dark-field microscopy* utiliza una luz muy intensa en forma de un cono hueco concentrado sobre el espécimen. El campo de visión del objetivo se encuentra en la zona hueca del cono de luz y solo recoge la luz que se refleja en el objeto. Por ello las porciones claras de la muestra aparecen como un fondo oscuro y los objetos minúsculos que se están analizando aparecen como una luz brillante sobre el fondo [18].

*Fluorescence microscopy* utiliza una sustancia natural en las células o un colorante fluorescente aplicado al corte es estimulado por un haz de luz, emitiendo parte de la energía absorbida como rayas luminosas, esto se conoce como fluorescencia. La luz fluorescente de mayor longitud de onda se observa como si viniera directamente del colorante [20].

*Phase contrast microscopy* usa componentes ópticos especiales para explotar las diferencias sutiles en los índices de refracción del agua y componentes citoplásmicos para producir contraste. Las ondas de luz que están en fase (que significa, sus picos y valles coinciden exactamente) se refuerzan unas a otras y su intensidad total incrementa. Las ondas de luz que están fuera de fase por exactamente media longitud de onda se cancelan unas a otras y resulta en no intensidad, o en otras palabras, obscuridad. Las longitudes de onda que están fuera de fase por cualquier cantidad producirán algún grado de cancelación y resultará un brillo que es menor que el máximo pero mayor al oscuro. Por esto, la muestra y sus partes aparecerán con varios niveles de luminosidad y obscuridad [20].

Una vez obtenidas las imágenes se proceden a analizarlas mediante técnicas espectrales las cuales se basan en el hecho de que todos los materiales reflejan, absorben y emiten energía electromagnética de forma distinta en diferentes longitudes de onda [21].

Como se mencionó anteriormente el análisis hiperespectral utiliza sensores hiperespectrales, para adquirir una gran cantidad de imágenes digitales utilizando canales espectrales muy cercanos entre sí. Al hacer esto, se obtiene una firma espectral característica para cada pixel [16]. Este proceso facilita la identificación y cuantificación de los materiales presentes en la escena [15]. El resultado que obtenido de la toma de datos del sensor hiperespectral puede ser representado en forma de cubo de datos, donde, dos dimensiones representan la ubicación espacial de un pixel, y una tercera dimensión que representa la singularidad espectral de cada pixel en diferentes longitudes de onda.

En la práctica, el uso de sensores hiperespectrales permite una mejor determinación de la composición interna de cada pixel, los cuales en las imágenes



geográficas raramente estarán compuestos por un único material. Existen diferentes técnicas de clasificación de patrones los cuales se basan en la asignación de etiquetas o clasificación individual a cada uno de los píxeles.

**Clasificación temática.** Tienen como objetivo la obtención de un mapa temático, donde cada píxel de las imágenes espectrales está debidamente etiquetado como perteneciente a una clase concreta [22]. En esta clasificación pueden existir clases adicionales denominadas fondo o resto que representan a los píxeles que no han sido clasificados en ninguna de las clases anteriores.

**Detección de targets.** Su objetivo principal es la identificación de un material denominado target entre todos los píxeles en la imagen. Estas técnicas han sido utilizadas de forma muy extensa en aplicaciones de detección de objetivos en imágenes hiperespectrales [23].

Los algoritmos de clasificación antes mencionadas son llamados algoritmos de clasificación binaria, aunque existen más tipos de clasificación de imágenes hiperespectrales, los cuales pueden dividirse en dos categorías [24]: algoritmos supervisados y algoritmos no supervisados.

**Algoritmos supervisados.** Estos algoritmos parten de un cierto conocimiento sobre las clases existentes, a partir de esto pueden derivarse los criterios de clasificación. Estas aproximaciones pueden darse por un previo estudio de las firmas espectrales características de las clases existentes [24].

**Algoritmos no supervisados.** A diferencia de los algoritmos supervisados, estos se basan en la presunción de que no existe ningún tipo de conocimiento a priori sobre las clases existentes. Su objetivo es identificar de manera automática, clases o agrupaciones de píxeles, utilizando métricas de similaridad [24].

Dentro de las técnicas de clasificación supervisadas, podemos destacar los filtros de similaridad (matched filters) y el método SAM (Spectral Angle Mapper), ambos basados en estadísticas de primer orden.

**Filtros de Similitud (MF).** Estos filtros han sido utilizados en numerosos problemas de análisis hiperespectral, ya que estos proporcionan mecanismos eficientes para detectar objetos utilizando un conocimiento previo acerca de los mismos [25]. Estos algoritmos han demostrado su eficacia como método de identificación de objetos [26].

**Método SAM.** Consiste en calcular el coseno del ángulo espectral entre una firma espectral característica de un objeto que se desea identificar y las firmas espectrales de todos los píxeles de la imagen. Para los píxeles donde su valor obtenido sea cercano a cero tendrán una alta probabilidad de estar constituidos por el material buscado [27].

Como se puede observar los algoritmos utilizados para el análisis de imágenes hiperespectrales se basan en las firmas espectrales de los píxeles, por lo que para usarlos es necesario contar con sensores hiperespectrales, de lo contrario será necesario contar con características completamente visibles o desarrollar otro tipo de algoritmo de clasificación.

Para desarrollar un algoritmo propio de clasificación primeramente es necesario saber si se utilizará un algoritmo supervisado o no supervisado. En este caso al conocer la principal característica de los contaminantes, la cual es su valor de reflectancia, se puede optar por utilizar un algoritmo supervisado. Para lo cual se hará uso de los códigos de cadena obtenidos de las intensidades de las imágenes espectrales en escala de grises. Los códigos de cadena que se utilizaron son los siguientes:

El código de Freeman de ocho direcciones F8, fue introducido en 1961, el cual cuenta con un conjunto de símbolos  $F8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ , los cuales representan el movimiento en el sentido de las manecillas del reloj con un ángulo de  $45^\circ$  entre cada símbolo. En la Ilustración 1 se pueden apreciar estos movimientos.

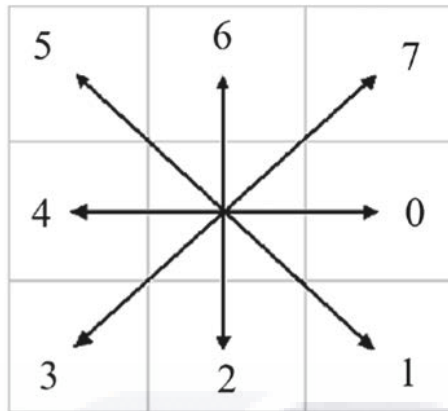


Ilustración 1. Las ocho direcciones del código de Freeman

El código de cadena F26 trabaja usando vecindad 26, lo que significa conectividad en cara, arista y vértice. Y al igual que F8, este código no es invariante ante rotación. Su conjunto de símbolos es  $F26 = \{A-Z\}$ . Estos símbolos representan uno de los veintiséis movimientos posibles que puede tener un voxel, los cuales se pueden ver en la Ilustración 2.

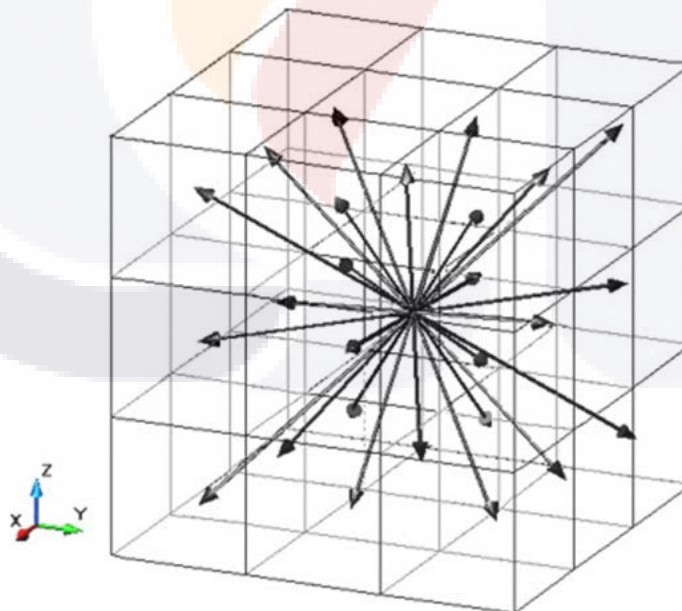


Ilustración 2. Las 26 direcciones del código de Freeman.

Como se mencionó, el código de cadena F26 consta de veintiséis símbolos, pero de la forma en como estamos trabajando con las imágenes para obtener estos símbolos, solamente utilizamos 11 de los 26 posibles, ya que estamos usando los ocho símbolos correspondientes a los movimientos de F8 y tres símbolos extras que representan la transición de un renglón a otro, ya que esta transición solo puede ser para abajo en vertical, como se puede visualizar en la ilustración 2. En la Tabla 1 se representa el movimiento en el espacio 3D (x, y, z) de los únicos símbolos que utilizamos.

*Tabla 1. Dirección de movimiento de los 11 símbolos utilizados del código F26.*

Símbolo	Movimiento (x, y, z)	Símbolo	Movimiento (x, y, z)
<b>A</b>	(0, 1, 0)	<b>Z</b>	(0, 0, -1)
<b>I</b>	(0, 1, 1)	<b>Q</b>	(0, 1, -1)
<b>Y</b>	(0, 0, 1)	<b>O</b>	(1, 0, 0)
<b>M</b>	(0, -1, 1)	<b>G</b>	(1, 0, 1)
<b>E</b>	(0, -1, 0)	<b>W</b>	(1, 0, -1)
<b>U</b>	(0, -1, -1)		

AF8 fue propuesto por Kui y Zalik en 2005, está basado en los cambios obtenidos con cada par de símbolos del código F8 cuanto siguen el contorno. Este código de cadena a diferencia de los F8 y F26 es invariante ante rotación. Cuenta con el mismo conjunto de símbolos que F8, el cual es  $AF8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ , pero no representa los mismos movimientos ya que estos están compuestos por pares de símbolos del código F8.

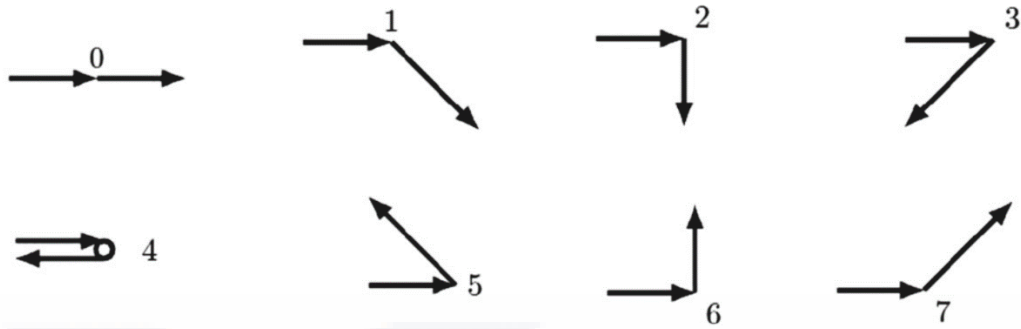


Ilustración 3. Las ocho direcciones del código AF8.

Para utilizar estos códigos debemos elegir la conectividad con la que trabajaremos. La conectividad entre pixeles se basa en establecer los límites en objetos y regiones en una imagen. Para esto es necesario determinar si dos pixeles son adyacentes entre sí, utilizando vecindades según la dimensión de la imagen.

Para imágenes en 2D, existen dos vecindades.

- a) **Vecindad 4.** Se consideran pixeles conectados en direcciones perpendiculares, arriba, abajo, derecha e izquierda.
- b) **Vecindad 8.** Se consideran los mismos pixeles conectados en vecindad 4 más los pixeles vecinos diagonales.

Para imágenes en 3D, existen también dos vecindades.

- a) **Vecindad 6.** Al igual que en la vecindad 4 se consideran voxeles conectados en direcciones perpendiculares, arriba, abajo, derecha e izquierda, más los voxeles superior e inferior.
- b) **Vecindad 26.** Se consideran todos los voxeles alrededor, los seis voxeles vecinos por cara, lo doce por arista y los ocho por vértice.

Una vez obtenidos los códigos de cadena de las imágenes espectrales, es necesario eliminar lo más posible el ruido que existe en ellos, este ruido usualmente está representado por “picos”. Para esto, es posible usar diferentes tipos de filtros, para ver

con cuál se obtienen mejores resultados sin perder información visual de las imágenes. A continuación se dará una breve explicación de los filtros utilizados:

***Gaussian blur.*** Este método actúa sobre cada píxel de la imagen, estableciendo su valor como el promedio de todos los valores de los píxeles incluidos en un radio definido, entre más alto sea el valor mayor efecto de desenfoque tendrá. Matemáticamente, aplicar este filtro a una imagen es lo mismo que *convulsionar* la imagen con una función Gaussiana. La aplicación de este desenfoque tiene el efecto de reducir los componentes de alta frecuencia de la imagen [28].

***Median.*** Es una técnica no lineal de filtrado digital, comúnmente usada para remover el ruido de una imagen o señal. Esta reducción de ruido es un paso típico de pre-procesamiento para mejorar los resultados de un procesamiento posterior. Este filtro es muy usado en procesamiento de imágenes digitales porque bajo ciertas condiciones, preserva los bordes mientras remueve el ruido [29].

***Posterization.*** La posterización de una imagen implica la conversión de una graduación continua de tono a varias regiones de menos tonos, con cambios abruptos de un tono a otro. En otras palabras, es la reducción de un rango de tonos de una imagen a algunos colores similares [30].

***Reduce noise.*** Las imágenes tomadas tanto con cámaras digitales como cámaras convencionales obtendrán ruido de una variedad de fuentes. Para un posterior uso de estas imágenes usualmente se requiere eliminar completamente o parcialmente ese ruido. El filtro *reduce noise* reduce el ruido luminoso y el ruido de color, así como el ruido introducido a la imagen con luz insuficiente [31].

Para el problema de los métodos de compresión podemos apoyarnos de los métodos ya existentes, para observar cuál de ellos funciona mejor con nuestros códigos de cadena. Explicaremos brevemente los algoritmos de compresión utilizados en este trabajo:

***LOCO-I/JPEG-LS.*** M. J. Weinberger, presentó su algoritmo de compresión de imágenes sin pérdidas, este tiene buenos porcentajes de compresión, igualando o incluso superando esquemas basados en codificación aritmética [32].



*Context-based, Adaptive, Lossless Image Coding (CALIC).* We y Memon introdujeron este algoritmo, el cual es superior a su predecesor inmediato Universal Context Modeling (UCM) por 2% [33].

*Huffman algorithm.* David A. Huffman desarrolló este algoritmo en 1952. Su algoritmo toma un alfabeto de  $n$  símbolos junto con sus frecuencias asociadas, y posteriormente produce un código Huffman para ese alfabeto y esas frecuencias [34].

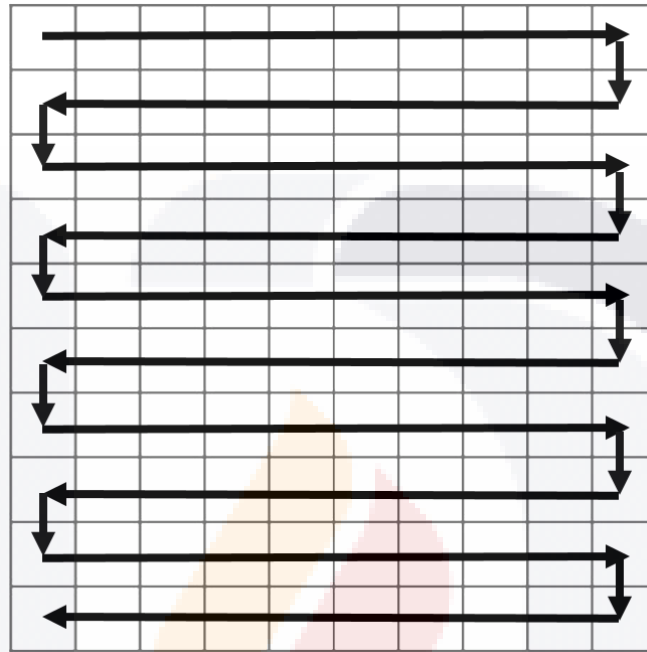
*Arithmetic coding.* Es un método de codificación de entropía usado en compresión de datos sin pérdida. Cuando una cadena de texto es convertida a una codificación aritmética, los caracteres más frecuentemente usados se guardarán con menos bits y los no tan frecuentes se guardarán con más bits, dando como resultado menos bits usados en total [35].

## **Capítulo 5. Metodología**

### **5.1 Diseño de la intervención o aplicación**

Para obtener los códigos, primeramente se tienen que convertir las imágenes a escala de grises, esto es porque se van a analizar las intensidades, así que con los valores de 0 a 255 de esta escala es más que suficiente, una vez convertidas se procede a obtener la matriz de intensidades de los píxeles de cada imagen. Para esto se va recorriendo cada píxel de la imagen obteniendo el valor de este, de esta forma se creará la matriz de intensidades de cada imagen, una vez obtenida esta matriz, se procede a recorrerla en forma de zig-zag horizontal, *i.e.* el primer renglón es visitado de izquierda a derecha, el segundo de derecha a izquierda y así sucesivamente como podemos en la Ilustración 4. Al hacer este recorrido se tiene que hacer otro recorrido interno por píxel, con el cual se puede ver claramente cómo se comportan las intensidades de cada píxel, donde se

buscaran los patrones antes mencionados, este recorrido interno se explicará a continuación.



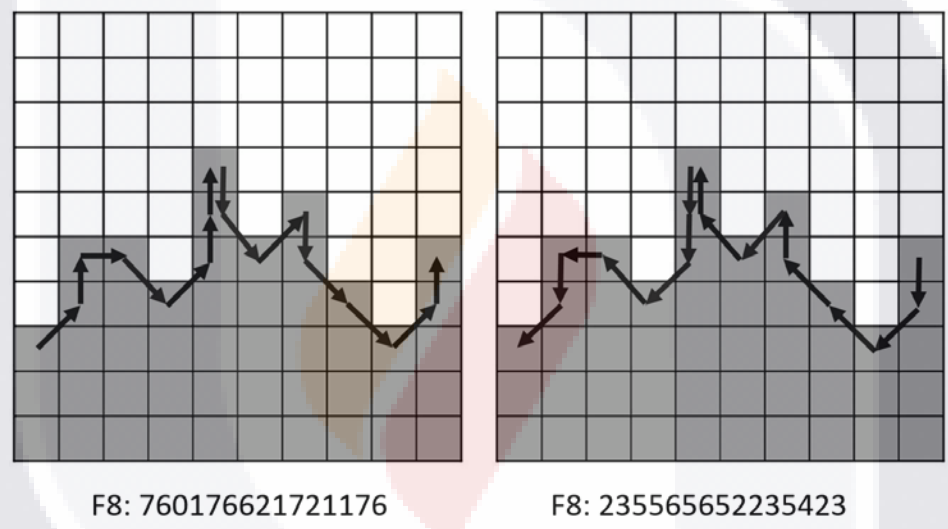
*Ilustración 4. Obtención de códigos de cadena en orden zig-zag horizontal.*

Con la ayuda de una variable auxiliar la cual llamaremos AuxVar con valor inicial de 0, la cual se comparará con la intensidad  $I(i, j)$ , haciendo uso de un convenio,  $i$  y  $j$  tendrán valor inicial de 0.

- I. Si  $I(i, j)$  es mayor que AuxVar, se incrementa AuxVar en uno y se añade el símbolo '6' para F8 y 'Y' para F26.
- II. Si  $I(i, j)$  es igual a AuxVar, se compara AuxVar con  $I(i, j+1)$ 
  - a. Si AuxVar es mayor por dos o más, se disminuye AuxVar en uno y se añade el símbolo '2' para F8 y 'Z' para F26.
  - b. Si AuxVar es mayor por uno, se disminuye AuxVar en uno, se añade el símbolo '1' para F8 y 'Q' para F26, y se avanza a  $I(i, j+1)$ .
  - c. Si AuxVar es menor, se incrementa AuxVar en uno, se añade el símbolo '7' para F8 e 'T' para F26 y se avanza a  $I(i, j+1)$ .

- d. Si AuxVar es igual, se añade el símbolo '0' para F8 y 'A' para F26 y se avanza a  $I(i, j+1)$ .

**Nota:** Se debe de tomar en cuenta que para los renglones pares se obtienen los códigos de cadena de izquierda a derecha, y para renglones impares es de derecha a izquierda, el método anterior está explicado para los renglones pares, si se quiere usar para renglones pares solo se tiene que cambiar los símbolos hacia la derecha por los símbolos hacia la izquierda respectivamente para cada código de cadena, i.e. en F8 el símbolo '7' representa un movimiento hacia arriba-derecha así que se cambia por un '5', el cual representa un movimiento hacia arriba-izquierda.



*Ilustración 5. Ejemplos de obtención de códigos mediante las intensidades de un renglón par e impar.*

En la figura anterior, se observan dos ejemplos, el de la izquierda es la codificación de un renglón par y el de la derecha representa la codificación de un renglón impar. En cada uno de ellos se empieza en el valor de la intensidad del pixel y de ahí nos desplazaremos según el algoritmo antes mencionado.

Para que el algoritmo funcione con F26 se necesitan agregar algunos pasos extra, ya que F26 da la posibilidad de obtener los códigos de cadena de la transición de un renglón a otro. Así que solo se usaran estos pasos extra si se llega al valor de la intensidad del último pixel de cada renglón, y se utilizará la misma variable auxiliar AuxVar para compararla con la intensidad  $I(i + 1, j)$ .

- I. Si AuxVar es mayor por uno o dos, se disminuye el valor de AuxVar por uno y se agrega el símbolo 'Z'.
- II. SI AuxVar es mayor por uno, se disminuye AuxVar por uno, se agrega el símbolo 'W' y se avanza a la posición  $I(i+1, j)$ .
- III. Si AuxVar es menor, se aumenta AuxVar por uno, se agrega el símbolo 'G' y se avanza a la posición  $I(i+1, j)$ .
- IV. Si AuxVar es igual, se agrega el símbolo 'O' y se avanza a la posición  $I(i+1, j)$ .

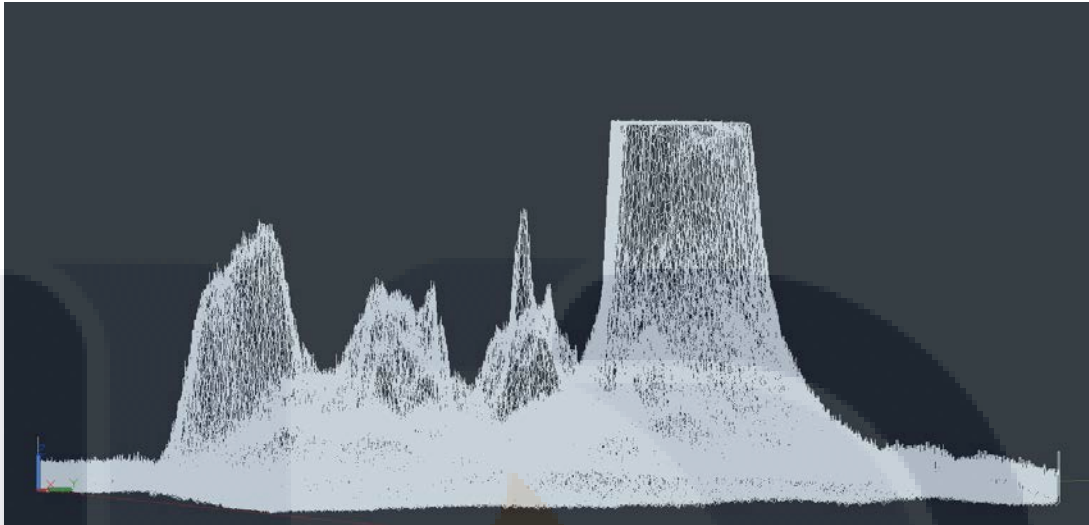
Al leer el algoritmo se puede observar que no se agregaron los códigos para AF8 en ningún momento, esto es porque este algoritmo solo sirve para F8 y F26, así que para el caso de AF8 se obtienen sus símbolos mediante la siguiente tabla de equivalencia (Tabla 2), esta tabla de equivalencia es del código F8 a AF8, ya que cada símbolo este último código representa la combinación de dos símbolos continuos de F8.

*Tabla 2. Matriz de equivalencia de F8 a AF8.*

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	7	0	1	2	3	4	5	6
2	6	7	0	1	2	3	4	5
3	5	6	7	0	1	2	3	4
4	4	5	6	7	0	1	2	3
5	3	4	5	6	7	0	1	2
6	2	3	4	5	6	7	0	1
7	1	2	3	4	5	6	7	0

Una vez obtenidos los códigos de cadena, se procede a obtener los comandos de polilíneas de AutoCAD (3DPoly). Las polilíneas son de gran ayuda para visualizar si se están obteniendo de manera correcta los códigos de cadena y principalmente para ver cómo se comportan estos y poder estudiar sus comportamientos y visualizar sus patrones de forma más clara y visible. En la Ilustraciones 6-8 se muestra un ejemplo del

comportamiento del código de cadena F26 en la muestra, vista desde distintos puntos donde se pueden apreciar los comportamientos de los símbolos.



*Ilustración 6. Forma completa creada en AutoCAD mediante polilíneas obtenidas del código F26.*

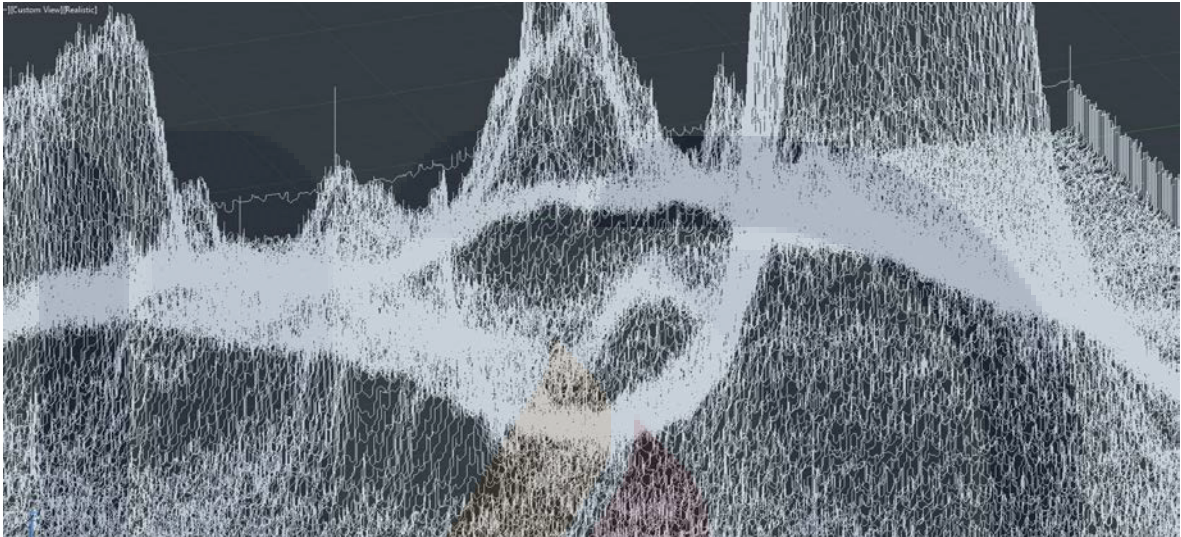
En la Ilustración 6 podemos ver la figura creada utilizando las polilíneas, donde se puede observar que hay zonas que resaltan de las demás, algunas de estas zonas tienen una morfología completamente diferente, mientras que hay otras que comparten una morfología parecida.



*Ilustración 7. Figura vista desde arriba.*



En la Ilustración 7, se observa la misma figura de polilíneas, solo que vista casi totalmente desde arriba. Aquí es más fácil ver que existen zonas que resaltan más en la imagen, las cuales son el objetivo a buscar y clasificar.



*Ilustración 8. Apreciación del movimiento de los símbolos de los códigos de cadena.*





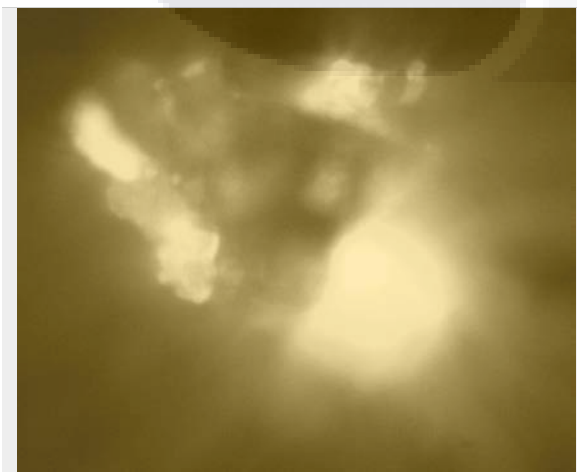

Al empezar a estudiar los patrones de los tres códigos de cadena en sus respectivas polilíneas se observó que hay demasiados picos, lo que dificultará de gran manera el tratar de encontrar patrones generales, ya que habrá una gran variedad de ellos y tratar de encontrar todos nos llevaría una gran cantidad de tiempo, y además podría haber la posibilidad de que estos patrones encontrados no nos funcionen por esta misma variedad de ellos ya que al empezar a buscar concordancias entre ellos dándonos diferencias donde nos debería mostrar que son parecidos.

## **5.2 Reducción de ruido**

Para evitar los picos en los códigos de cadena es necesario reducir el ruido que ocasiona estos en las imágenes, por lo que se utilizaron diferentes filtros para lograr esto. En la Tabla 3 se pueden observar los resultados obtenidos al aplicar algunos filtros.

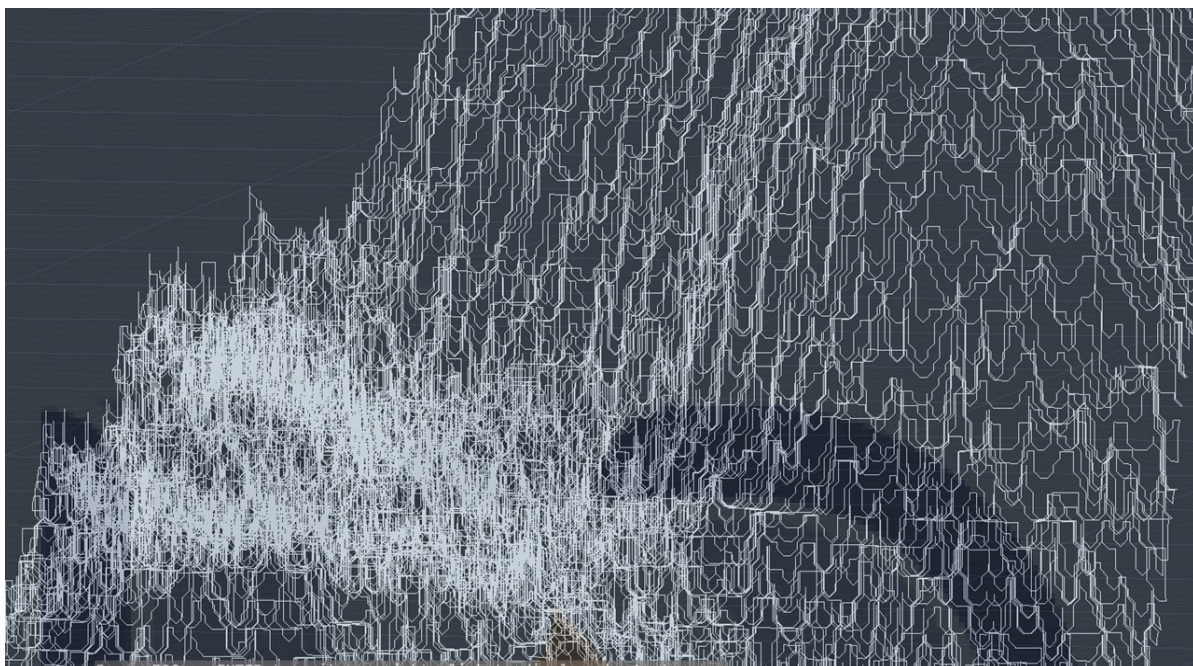


Tabla 3. Resultados de las imágenes después de aplicar los filtros.

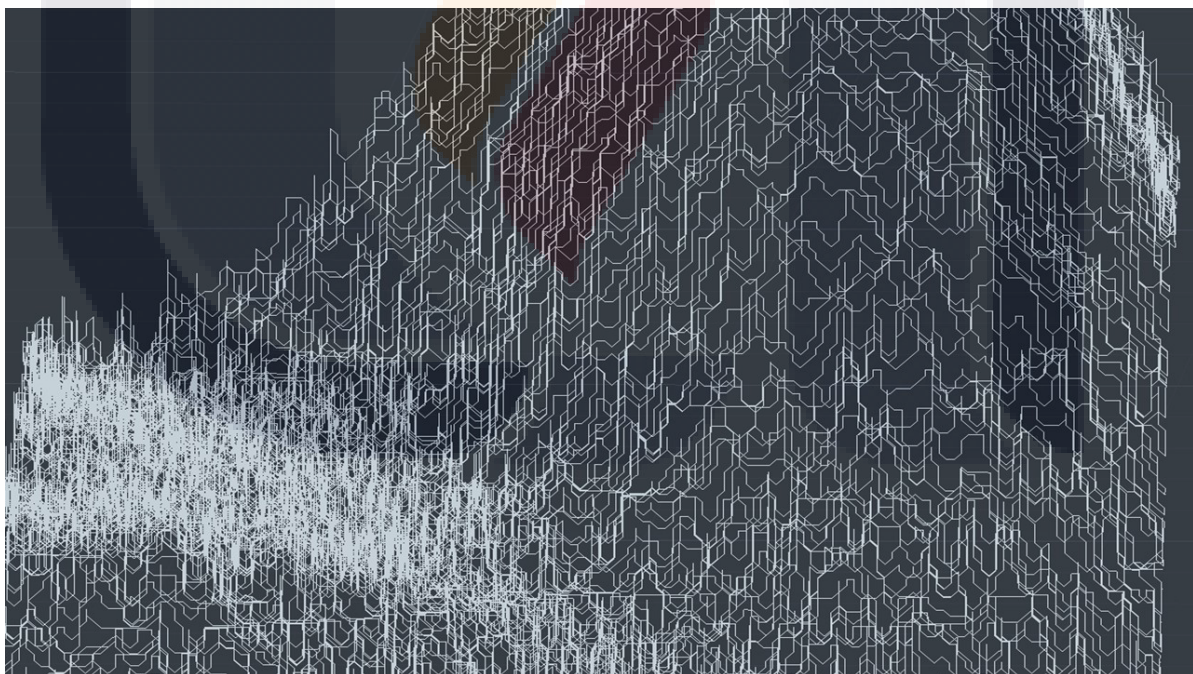
Imagen Original	Gaussian Blur
	
Median	Posterization
	
Average	Reduce Noise
	

En los resultados obtenidos de *Gaussian blur*, *Average Blur* y de *Reduce Noise* se puede ver que estos filtros hicieron las imágenes un poco estilo caricatura, haciendo que la información visual de la imagen original cambiara, lo que daría resultados no deseados. Quedando únicamente las imágenes resultantes de dos filtros, *Median* y *Posterization*, estas imágenes mantuvieron la información visual pero el filtro *Median* cambió la gama de colores de muchos píxeles, por lo que igual que las imágenes anteriores, estaría cambiando una gran cantidad de datos, y por esta razón también fue rechazado. Quedándonos únicamente con *Posterization*, el cual mantuvo la misma información visual que la original.

Lo que se buscaba inicialmente era que se redujera el ruido de la imagen, para que de igual manera se redujeran la cantidad de picos manteniendo movimientos uniformes. Por lo que después de obtener la imagen con *Posterization*, se procedió a obtener sus códigos de cadena junto con sus polilíneas para visualizar los resultados. En las ilustraciones 9-11 se pueden apreciar los resultados.

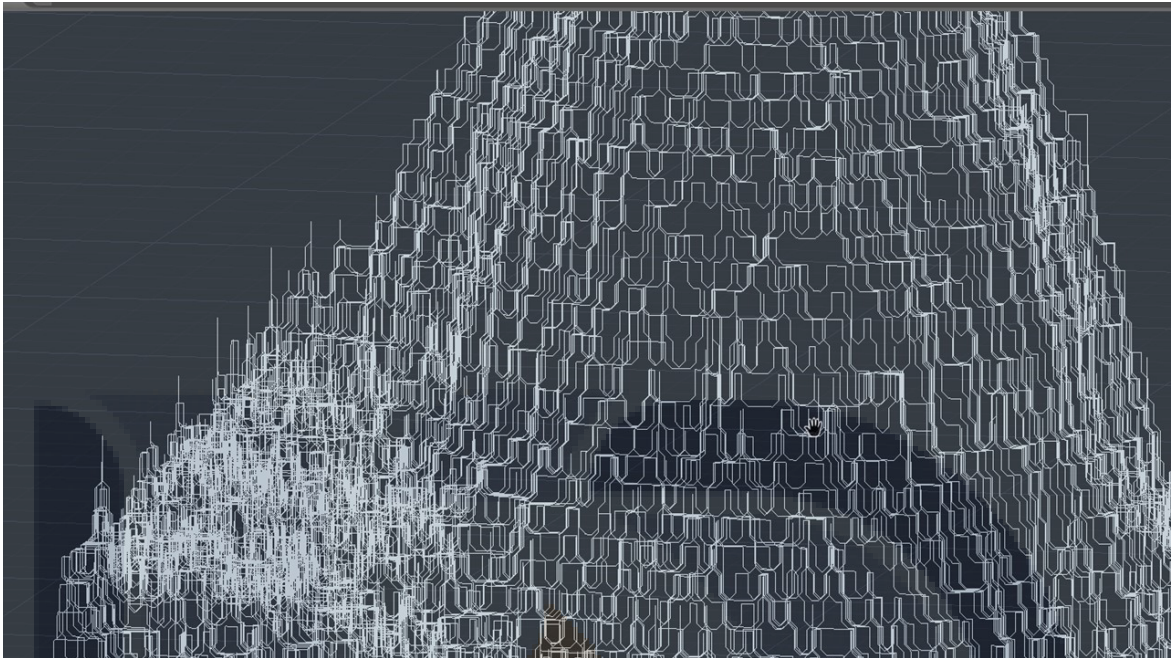


*Ilustración 9. Polilíneas obtenidas antes de aplicar posterización.*



*Ilustración 10. Polilíneas obtenidas después de posterizar*





*Ilustración 11. Polilíneas obtenidas después de posterizar en 64.*

La razón por la que solo se muestra una parte de todas las polilíneas, es porque el proceso de graficar estas mismas es muy tardado, ya que hay que recordar que los valores de intensidad van desde 0 hasta 255, lo cual en muchos pixeles hace un recorrido muy grande, y aparte las imágenes son grandes, midiendo 1252 x 1040, haciendo que el proceso sea todavía más tardado. Por esto fue que se decidió graficar solo unas partes para observar los resultados.

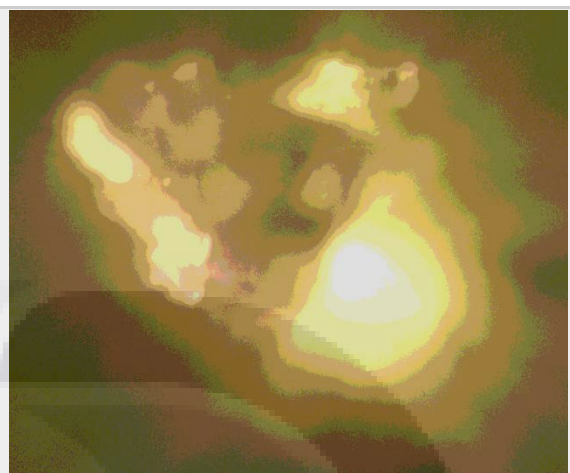
Se puede ver claramente que el ruido en las imágenes se redujo de manera considerable, reduciendo la cantidad de picos y haciendo los movimientos de los códigos de cadena más uniformes.

Después de esto, se experimentaron con diferentes valores de posterización como; 8, 16, 32, 64 y 128. Al saber que las posterización da resultados favorables, solo queda ver cual valor de posterización nos da los mejores resultados posibles. Con ayuda de Paint.net se obtuvieron las imágenes posterizadas con los valores antes mencionados, en la Tabla 4 podemos ver estas imágenes.

Tabla 4. Resultados de las imágenes después de la posterización.

Imagen Original

Posterizada con valor de 8



Posterizada con valor de 16

Posterizada con valor de 32



Posterizada con valor de 64

Posterizada con valor de 128



Claramente en las imágenes posterizadas en 8 y 16 se muestra un espectro que no pertenece a las imágenes, lo cual es una alteración a los datos de esta, y como se mencionó anteriormente un cambio grande en la información original puede dar resultados no deseados, por lo que se descartaron las imágenes posterizadas con estos valores.

La imagen con valor de posterización de 32 no tiene el espectro que muestran las imágenes anteriores, por lo que se procedió a comparar esta imagen con la imagen posterizada con valor de 64. Como se puede ver, estas imágenes en si son idénticas, la diferencia recae en sus píxeles. Ya que haciendo zoom en las dos imágenes se pueden apreciar la diferencia que hay entre estos. Esto es por el proceso de la posterización el cual es la conversión de una graduación continua de tonos a varias regiones de un menor número de tonos [28].

Al encontrar esto, se decidió descartar el valor de posterización de 32, quedando solo los valores 64 y 128. Después de hacer varias comparaciones acerca de su información visual se llegó a la conclusión de que cualquiera de las dos mantiene la misma información. Por lo que se procedió a comparar el comportamiento de sus códigos de cadena. Aquí fue donde se pudo apreciar una diferencia entre ellas para poder seleccionar solo uno de los dos valores de posterización restantes. Esta diferencia se basa principalmente en la reducción de ruido de la imagen.

Como podemos ver en la Ilustración 10 correspondiente a la obtención de polilíneas después de aplicar posterización con valor de 128, la reducción del ruido fue menor que en la Ilustración 11 correspondiente a la posterización con valor de 64. En la Ilustración 10 se observa que incluso varios de los picos se mantuvieron y el comportamiento de sus símbolos no es tan uniforme como en la de posterizado con valor de 64. Por esta razón se descartó el valor de 128, por lo que pudimos concluir que el mejor valor de posterizado para disminuir el ruido es 64. Esto se puede apreciar claramente en la Ilustración 10, donde observamos como el error va disminuyendo mientras más grande sea el valor de posterización. El error usando los valores de 64 y



128 es casi el mismo así que simplemente es cuestión de comparar sus resultados, los cuales explicamos anteriormente.

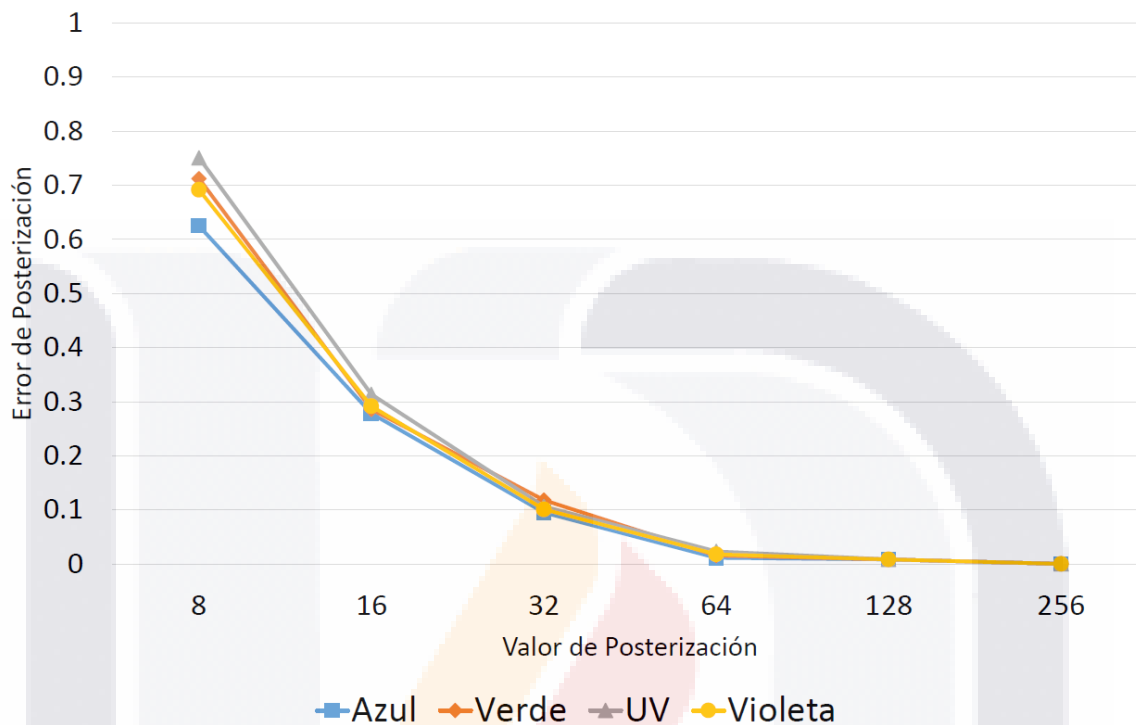


Ilustración 12. Porcentaje de error al utilizar Posterización

### 5.3 Compresión de códigos de cadena

Después de limpiar el ruido en los códigos de cadena procedemos a comprimir estos con los compresores más comunes (*Huffman*, *Aithmetic*, *JPEG-LS*, *CALIC* y *COMI*). En un inicio se empezó utilizando los más antiguos, los cuales son: *Huffman* y *Arithmetic*. En la Tabla 5 se muestran los resultados promedios en bytes obtenidos de estos compresores.

Tabla 5. Resultados de Huffman & Arithmetic

Archivo	Original	Huffman	Arithmetic	Promedio
F8	3,667,930	1,024,671	956,047	<b>1,882,883</b>
AF8	3,666,890	800,422	773,743	<b>1,747,018</b>
F26	3,670,538	1,174,138	1,126,666	<b>1,990,447</b>
<b>Promedio</b>	<b>3,668,453</b>	<b>999,744</b>	<b>952,152</b>	

Huffman da archivos de mayor tamaño que los de Arithmetic, aunque no es mucha la diferencia. Si comparamos los resultados de los tres códigos de cadena podemos ver que el que mejor trabaja con estos compresores es AF8 y el mejor compresor es Arithmetic. Como estos dos compresores son más antiguos que los últimos tres, se obtuvieron archivos más chicos al utilizar JPEG-LS, CALIC y COMI. Aunque JPEG-LS da mejores resultados en el espectro verde, se debe recordar que el experto tiene que hacer el análisis posterior en todas las imágenes, ya que algunas propiedades pueden aparecer en algunas imágenes y en otras no. En la Tabla 6 se muestran los resultados obtenidos con estos últimos compresores. Se usó COMI para los tres códigos de cadena que estamos utilizando: F8, F26 y AF8. Aunque dan resultados similares en todas las imágenes espectrales, el que mejor resultados dio fue F8.

Tabla 6. Compresión de las imágenes posterizadas con valor de 64.

	Azul	Verde	UV	Violeta	Promedio
<b>BMP</b>	1,303,158	1,303,158	1,303,158	1,303,158	1,303,158
<b>CALIC</b>	470,925	296,256	486,540	477,250	432,743
<b>JLS</b>	437,249	129,340	313,178	307,615	296,846
<b>COMI - F8</b>	276,006	239,837	308,901	294,177	279,730
<b>COMI - AF8</b>	281,514	244,422	314,398	299,536	284,968
<b>COMI - F26</b>	276,911	240,730	310,015	295,222	280,720

Como se puede observar, usando imágenes posterizadas con valor de 64 junto con el compresor COMI se superaron los archivos resultantes de JLS y CALIC. Algo que llamó la atención fue que con Huffman y Arithmetic, AF8 comprimía mejor que F8, pero usando COMI, F8 es el que comprime mejor, se debe analizar porque pasa esto, pero como una primera aproximación se puede decir que es porque Huffman y Arithmetic no usan dependencias en símbolos anteriores para comprimir, mientras que COMI usa combinación de dependencias.

#### **5.4 Búsqueda de zonas de interés y clasificación por morfología**

Una vez conocido esto, se obtuvieron nuevamente sus códigos de cadena junto con los comandos para graficar las polilíneas. Posteriormente se estudiaron los códigos de cadena para ver cuál de ellos era más conveniente usar para encontrar patrones, al estudiar los tres códigos de cadena nos dimos cuenta que F26 en nuestro caso es lo mismo que usar F8, ya que solo da la información de la transición entre renglones, pero esta información no es de utilidad, por lo que lo descartamos.

Al comparar AF8 con F8 se pueden observar dos diferencias. La primera es la cantidad de símbolos utilizados en ambos, como un símbolo de AF8 corresponde a dos de F8, la cantidad será menor por el número de renglones en la matriz. En la teoría AF8 debe tener el mismo número de símbolos que F8, porque al final el símbolo faltante se obtiene al combinar el símbolo final con el inicial de F8. Lo que no pasa en este caso, ya que el recorrido no regresa a su punto inicial en cada renglón, obteniendo un símbolo menos por renglón para AF8. La segunda diferencia recae en las propiedades de cada código, ya que AF8 es invariante ante rotación y F8 no lo es.

Al estudiar estas diferencias se optó por utilizar AF8, primeramente por la cantidad de símbolos, ya que al momento de analizar estos códigos en busca de patrones, el número de símbolos para analizar es menor reduciendo el tiempo de análisis. Se podrá pensar que esto no reducirá mucho el tiempo, pero se debe de tomar en cuenta que para las imágenes hiperespectrales se utilizan una gran cantidad de imágenes, y como se

desconoce el número de imágenes que desee utilizar el usuario, se debe tener en cuenta el peor de los casos, por lo que se tiene que reducir el tiempo lo mejor que se pueda en cualquiera de los procesos con los que cuenta el sistema. Y la segunda razón por la que se seleccionó el código AF8 es por su propiedad de invariancia ante rotación, lo cual es de gran ayuda en caso de que se utilicen las mismas imágenes pero tomadas en diferente posición, lo cual tendría que dar los mismos resultados ya que es la misma imagen, sin ninguna alteración en su información.

Al hacer esto se observó que el proceso con el que se obtienen los códigos de cadena se podía mejorar en tiempo considerablemente. En un inicio, este proceso se hacía de manera lineal, recorriendo desde la posición (0, 0) de la matriz de intensidades hasta la posición de la esquina inferior derecha de la matriz haciendo el recorrido en zig-zag horizontal antes explicado (Ilustración 4), haciéndolo de esta forma dependíamos completamente de la cantidad de renglones, columnas y los valores de intensidad de cada pixel. Ya que si los renglones y columnas eran muchas, y aparte los valores de cada pixel fueran muy diferentes entre sí, ocasiona un aumento en el tiempo de procesamiento.

Así que se optó por mejorar el algoritmo que obtenía los códigos de cadena. Lo único que se hizo fue cambiar el proceso lineal por uno asíncrono, el cual por medio de threads (hilos) iba a ir obteniendo los códigos para cada renglón de manera simultánea, sin tener que esperar que termine el renglón anterior para continuar. De esta forma se logró reducir el tiempo, lo cual es de gran ayuda, porque como ya se mencionó anteriormente, se debe de tomar en cuenta que no sabemos el número de imágenes con las que trabajará el usuario, por lo que se tiene que estar preparado en caso de que este número sea elevado. En la Tabla 7 se puede apreciar el tiempo de proceso antes y después de hacer el cambio.

Tabla 7. Tiempo necesario para obtener los códigos de cadena en las dos versiones del algoritmo.

<b>Código</b>	<b>Versión 1</b>	<b>Versión 2</b>
<b>F8</b>	1.05 min	15 seg
<b>AF8</b>	1.05 min + 5 seg	15 seg + 5 seg
<b>F26</b>	1.12 min	20 seg

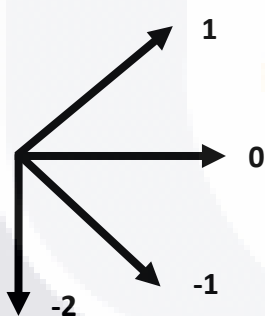
Una vez obtenidos todos estos resultados, se procedió a buscar los patrones en los códigos, pero se volvió a encontrar un problema importante, el cual era la cantidad de información innecesaria. Esta información que no es útil es todos aquellos códigos obtenidos en zonas donde no se encontraba la muestra, esto quiere decir, que es basura en el portaobjetos, en la cámara o en el lente del microscopio, así que la información de esas zonas no interesa obtener. Para que no ocurra esto, se diseñó un algoritmo para eliminar la información no deseada.

Básicamente, el algoritmo analiza la imagen tomada con luz blanca. En las zonas que encuentre oscuras buscara hoyos, los cuales los rellenara, esto para separar las zonas donde se encuentre la muestra de las que no. Una vez hecho esto todo lo demás que sea de un color más claro, usualmente entre gris y blanco, lo pondrá totalmente blanco y las zonas más oscuras las pondrá totalmente en negro. Esto servirá como banderas para el algoritmo de análisis para saber que pixeles si pertenecen a la muestra y cuáles no, por lo que al inicio del algoritmo de análisis pondrá en blanco las zonas donde no hay muestra en las imágenes espectrales, analizando únicamente los pixeles que nos interesan.

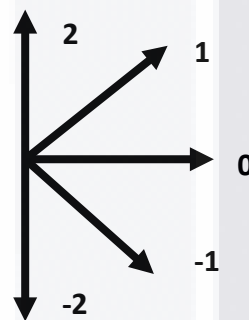
Ya una vez borrados todos los pixeles que no son de interés y optimizados los algoritmos de obtención de códigos, se puede empezar a analizar estos mismos. Como se mencionó anteriormente el código que se seleccionó para trabajar es AF8, ya que este reduce la cantidad de símbolos a analizar y por ende el tiempo de análisis. Pero existe un problema al querer trabajar con este código, ya que en este a diferencia de los otros dos códigos no es posible conocer la dirección del movimiento de cada símbolo, como en F8 que cada símbolo tiene una dirección de movimiento única. Esto genera un

conflicto al algoritmo de análisis de códigos, porque este tiene dos funciones principales, las cuales son, encontrar las zonas de interés y clasificar estas zonas. Para encontrar las zonas se empieza a seguir los símbolos del código, cuando ve que paso el valor promedio de los pixeles guarda la posición en la que se encuentra, para saber que desde ese punto empieza una zona de interés. Sigue moviéndose dependiendo del símbolo y cuando vuelve a bajar del valor promedio guarda también esa posición para saber que ha salido de la zona de interés. Hace esto mismo durante el recorrido de símbolos de los códigos de cadena, encontrando las posiciones del perímetro de todas las zonas de interés que haya en las imágenes.

Al no poder conocer la dirección de movimiento del símbolo no se pueden encontrar las zonas de interés de esta forma, por lo que se utilizó un código auxiliar para poder conocer la dirección en los códigos AF8. Este código lo llamaremos código 012, el cual consta de cinco símbolos, los cuales podremos ver en la siguiente imagen.



*Ilustración 13. Valores posibles en el primer símbolo del renglón.*



*Ilustración 14. Valores posibles en los siguientes símbolos.*

**Nota:** No se va a obtener un nuevo código, simplemente mediante una tabla de equivalencia de estos dos códigos se obtendrá la dirección de movimiento en la que nos encontramos.

Para poder utilizar el apoyo del código 012 se necesita acordar el ángulo de movimiento de los símbolos del código AF8, en la siguiente imagen se puede observar



los ángulos correspondientes a cada símbolo, en este caso utilizamos ángulos negativos ya que el movimiento que hizo fue en el sentido de las manecillas del reloj.

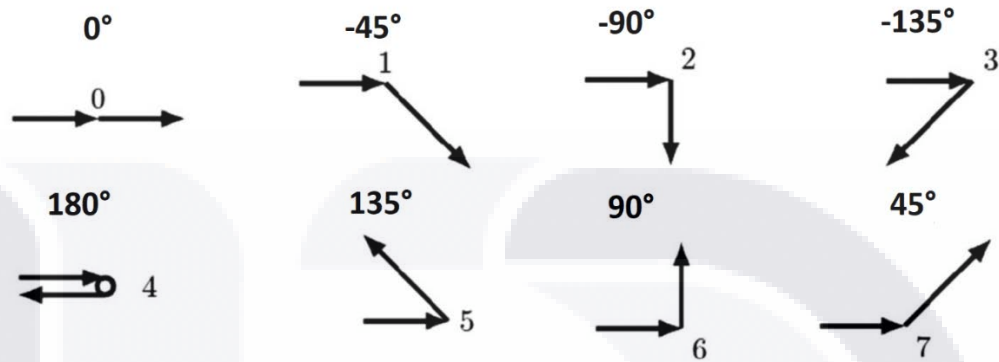


Ilustración 15. Ángulos de movimiento correspondientes a los símbolos de AF8.

Combinando los símbolos del código 012 con los ángulos de correspondientes a los símbolos de AF8 es más fácil ver hacia donde se está moviendo. En nuestro caso no todas las combinaciones son posibles, en la Tabla 8 se puede ver más a detalle los resultados de todas las combinaciones. Los valores resultantes son los símbolos de 012.

Tabla 8. Valores para conocer la dirección de movimiento.

Símbolo	0°	-45°	-90°	-135°	180°	135°	90°	45°
2	2	1	0	-1	-2	X	X	X
1	1	0	-1	-2	X	X	X	2
0	0	-1	-2	X	X	X	2	1
-1	-1	-2	X	X	X	2	1	0
-2	-2	X	X	X	X	1	0	-1

Como se mencionó anteriormente no solo es conocer la dirección de movimiento, también se debe conocer si pasamos el valor promedio de intensidad. Para esto es necesario llevar un contador que nos vaya diciendo si pasamos el límite o no, con la misma tabla de combinación es posible obtener esto, aunque los valores resultantes

serán diferentes ya que en la tabla anterior obtenemos la dirección. En la Tabla 9 se puede observar la tabla de combinaciones para saber si aumentar o disminuir nuestra variable auxiliar. Esta variable auxiliar en cada inicio del renglón se le asigna el valor de intensidad del pixel de esa posición.

*Tabla 9. Valores que se suman o restan con la variable auxiliar 'suma'.*

Símbolo	0°	-45°	-90°	-135°	180°	135°	90°	45°
<b>2</b>	+1	+1	+0	-1	-1	X	X	X
<b>1</b>	+1	+0	-1	-1	X	X	X	+1
<b>0</b>	+0	-1	-1	X	X	X	+1	+1
<b>-1</b>	-1	-1	X	X	X	+1	+1	+0
<b>-2</b>	-1	X	X	X	X	+1	+0	-1

Al usar este método de forma zig-zag horizontal se es capaz de encontrar las zonas de interés en las imágenes. Para hacer la clasificación de estas zonas se utiliza el mismo método pero ahora en zig-zag vertical, pero no en toda la imagen sino, simplemente aplicándolo a las zonas de interés, al combinar estos nuevos datos con los anteriores se puede obtener la morfología de las zonas encontradas anteriormente.

La morfología completa nos ayuda a hacer una clasificación más exacta, ya que lo que dice es el comportamiento de cada zona, por lo que los contaminantes que tengan las mismas propiedades tendrán un comportamiento igual o similar. Para desarrollar este método de clasificación, primeramente se obtienen los códigos de cadena en forma vertical de cada zona, después se obtienen los puntos dominantes, guardando su ubicación espacial y el movimiento que realizaron, para posteriormente compáralos con los de otras zonas, dejando como posible error un 10%.

### 5.5 Software utilizado y explicación de los algoritmos desarrollados

En total se utilizaron tres softwares externos auxiliares, los cuales fueron los siguientes:

- **GIMP**

Se utilizó para aplicar los diferentes filtros utilizados a las imágenes espectrales.

- **Matlab**

Con ayuda de sus funciones para procesamiento de imágenes, se crearon las funciones de borrado de fondo y llenado de huecos. Y también se utilizó para obtener la matriz de intensidades de nuestras imágenes.

- **Netbeans**

Al tener experiencia utilizando este IDE, y al ser uno de los que nos ha dado mejores resultados se decidió utilizarlo para desarrollar nuestro sistema usando el lenguaje JSP.

Como tenemos la capacidad de utilizar JAVA en JSP, se aprovechó esto para trabajar de forma orientada a objetos, por lo que la mayor parte de nuestros algoritmos se pensaron para que trabajaran rápidamente y para distintos casos. A continuación mencionaremos y explicaremos nuestros algoritmos generados.

- **Obtener códigos de cadena (AF8, F8 y F26)**

El funcionamiento de este algoritmo es simple, primeramente lee la matriz de intensidades de cada imagen espectral visitando celda por celda cada renglón o fila de la matriz, para posteriormente ir obteniendo cada uno de los códigos de cadena utilizados junto con sus respectivos comandos de polilíneas de manera simultánea. Esta desarrollado de tal manera que funcione utilizando zig-zag horizontal como vertical y sin necesidad de tener que visitar toda la matriz, ya que se pueden mandar arreglos de coordenadas para trabajar solamente en zonas específicas.

En un principio este algoritmo recorría toda la matriz siguiendo un camino, es decir, linealmente. Esto se cambió para que trabajará de manera asíncrona, y obtuviera los códigos de cadena de cierta cantidad de renglones o filas de manera simultánea, reduciendo el tiempo de trabajo.

- **Encontrar zonas de interés**

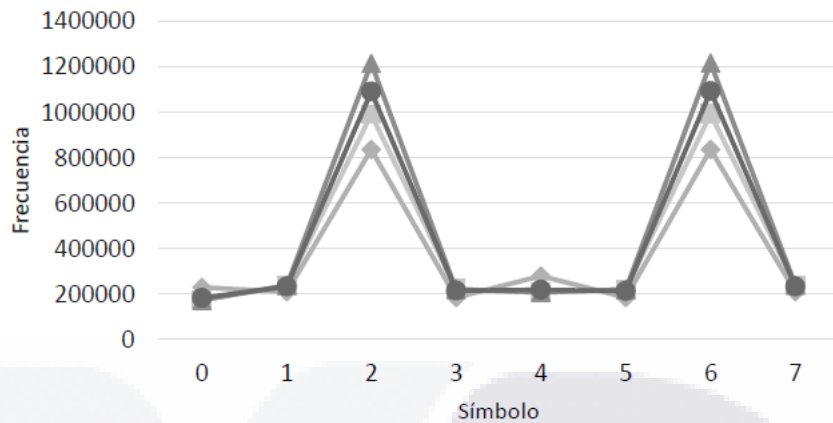
Al obtener los códigos de cadena, procedemos a obtener el código 012 al mismo tiempo, comparando el valor auxiliar de subida o bajada para saber si entramos o salimos a una zona de interés, en el caso de que sea verdadero cualquiera de los dos casos se guardan sus respectivas coordenadas.

- **Clasificar zonas de interés**

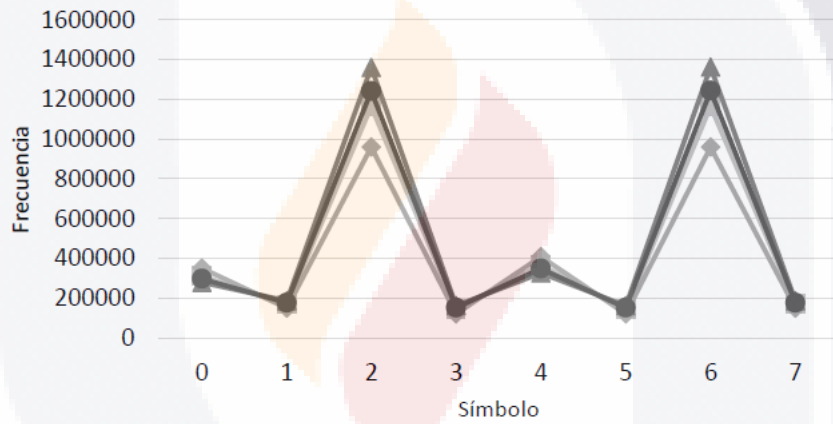
Se recibe el conjunto de coordenadas de cada zona, y se procede a llamar al algoritmo de Obtención de Códigos de Cadena, para obtener sus códigos en vertical. Al finalizar este proceso se obtuvieron varios puntos dominantes en toda la zona, los cuáles se guardan en un arreglo, para una vez finalizado el proceso de obtención de códigos para cada zona se proceda a comparar estos arreglos de puntos dominantes y hacer su respectiva clasificación.

## **Capítulo 6. Resultados**

Al reducir el ruido en las imágenes, lo más importante es ver si la información visual se alteró o se mantuvo. Como se mencionó en el capítulo 5, el filtro de posterizado es el que hace un mejor trabajo para esto, ya que mantiene esta información y reduce en gran medida el ruido que genera los picos en los códigos de cadena. Esto se puede ver en las ilustraciones 16-18.



a) Original



b) Posterizado a 64

Ilustración 16. Frecuencia de símbolos F8 antes y después de posterizar.

En la ilustración 16 se puede observar que hay dos símbolos que dominan más que los otros en ambas imágenes, tanto en la imagen original como en la posterizada, los cuáles son 2 y 6. Si observamos en la gráfica que representa las frecuencias de los símbolos después de posterizar se puede ver que la frecuencia de los símbolos 0 y 4 es mayor que en la original, esto es porque el ruido en la imagen original tiene demasiados picos, y después de posterizar, esos picos se aplanan haciendo más movimientos rectos.

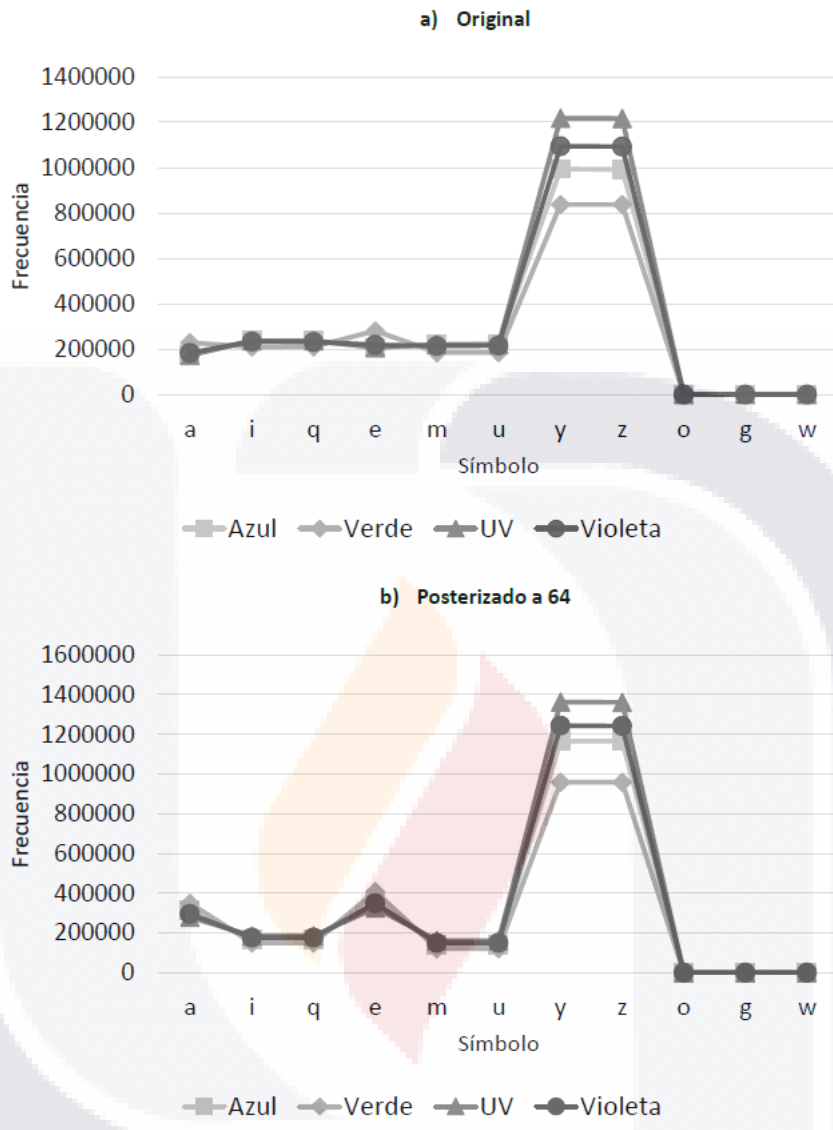


Ilustración 17. Frecuencia de símbolos F26 antes y después de posterizar.

En la Ilustración 17 al igual que en la Ilustración 16 las formas se mantienen en ambas imágenes, tanto en la original como en la posterizada. Aunque no se ve a simple vista, la frecuencia de los símbolos es igual que en la de F8, sin contar los últimos tres símbolos, esto es porque en este método, de los 26 símbolos posibles de F26 solo se usan 11, que vendrían siendo los ocho símbolos equivalentes a F8 y tres símbolos más que son usados para la transición de un renglón a otro.



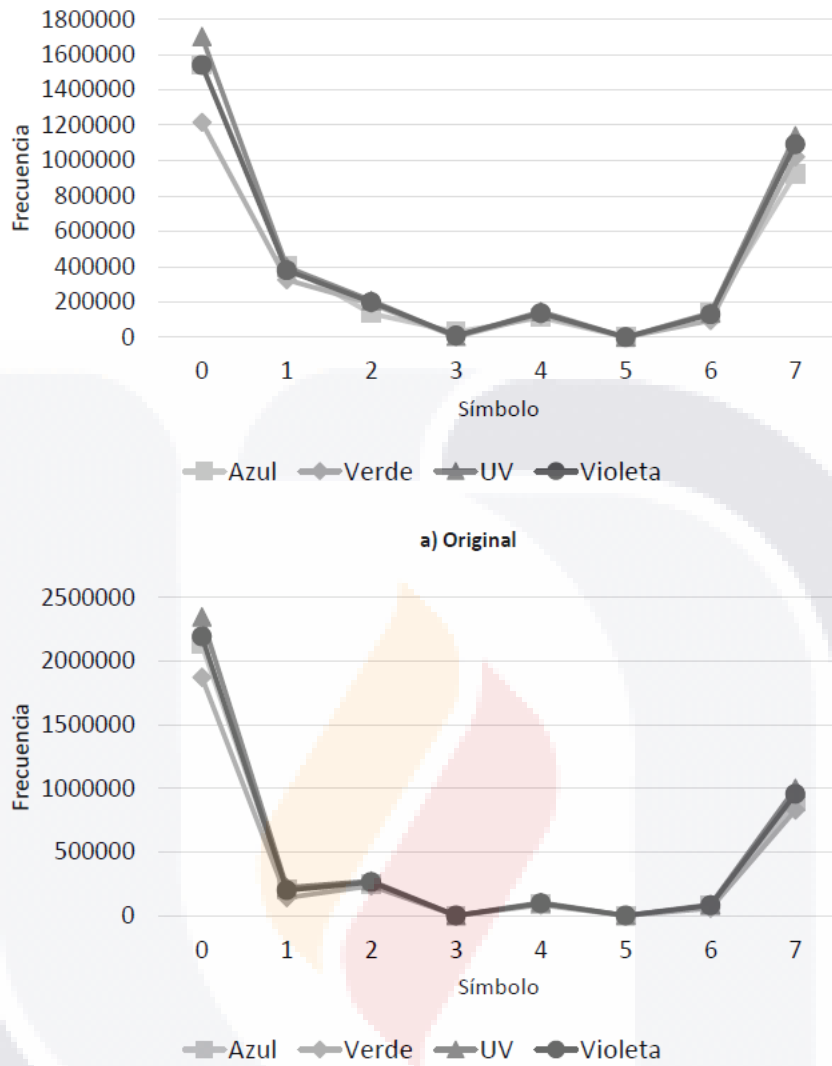


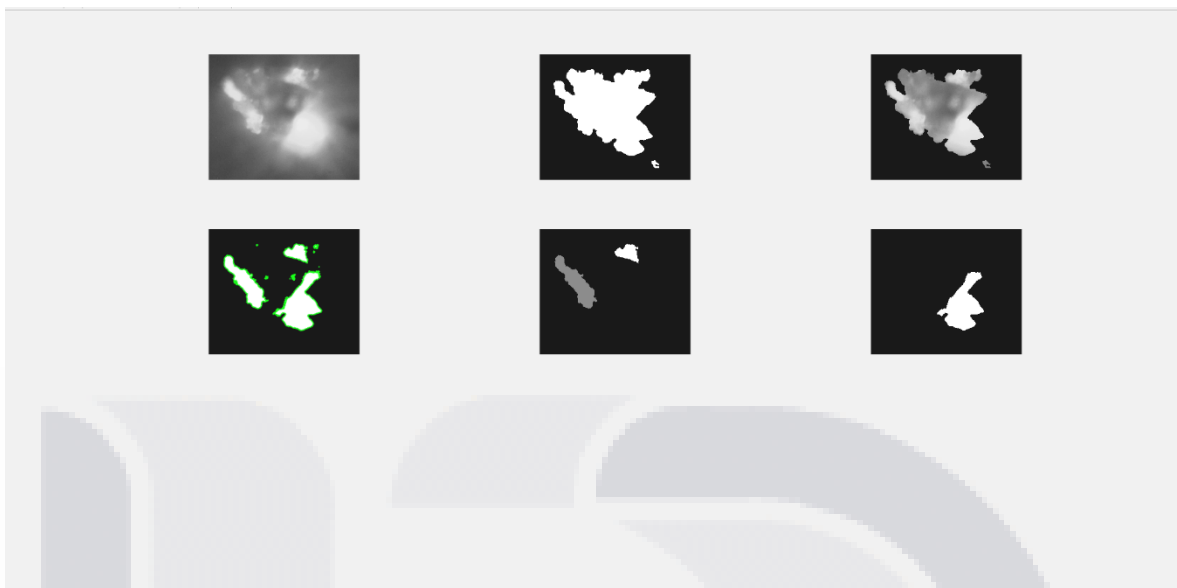
Ilustración 18. Frecuencia de símbolos AF8 antes y después de posterizar.

En la Ilustración 18 se pueden ver las frecuencias de los símbolos del último código de cadena usado AF8. Y aunque también se mantienen las formas en ambas imágenes, la principal diferencia entre esta y las ilustraciones anteriores, es que se puede observar que la posterización funciona mejor para este código de cadena, ya que la reducción de los símbolos fue mayor haciendo que casi la mayor parte de los símbolos se acercaran a cero, e incluso uno de ellos (3) se logró reducir hasta cero. El único símbolo que se incrementó su frecuencia fue el 0, y esto es porque este símbolo

representa los movimientos rectos no importa si son para arriba, abajo, izquierda, derecha o incluso en diagonal.

Una vez reducido el ruido en las imágenes y haber comprimido sus códigos de cadena, se facilitó la búsqueda de patrones en los códigos de cadena de cada una de ellas. Incluso, se logró reducir en gran medida el tiempo de análisis de los datos, esto no solo se logró al aplicar el filtro de posterizado, también fue de gran ayuda eliminar el fondo de las muestras, lo que venía siendo la superficie del portaobjetos, el cual puede contener basura la cual el sistema la contaría como parte de la muestra, por esta razón se optó por codificar solo la información que pertenece a la muestra y trabajar únicamente con ella.

Como se mencionó al final de nuestra metodología, una vez obtenidos los códigos de la muestra se procede a buscar zonas de interés (contaminantes) dentro de ella, al encontrarlas se procede a calcular y analizar su morfología para posteriormente cuantificar y clasificar estas zonas. Este proceso lo podemos observar más a detalles en las ilustraciones 19-22. El cual abarca desde el borrado de fondo, seguido por la búsqueda de zonas de interés hasta la clasificación de estas zonas. Las primeras cuatro imágenes son el proceso antes de clasificar y las siguientes, son los resultados de la clasificación.



*Ilustración 19. Resultados obtenidos en la imagen espectral de luz azul.*

En la Ilustración 19, se muestran los resultados de la imagen espectral azul. Las últimas dos imágenes representan la clasificación resultante de las zonas de interés encontradas, como podemos ver dos de ellas tienen propiedades similares por lo que se clasificaron en el mismo grupo, quedando en otro grupo la zona de mayor reflectancia.



*Ilustración 20. Resultados obtenidos en la imagen espectral de luz verde.*

En la Ilustración 20, se encuentran los resultados de la imagen espectral de luz verde. El proceso de clasificación nos dio resultados similares a los obtenidos de la imagen espectral de luz azul, separando los elementos en dos grupos. La única diferencia fue que en este espectro separo uno de los dos elementos del primer grupo, ya que estos no emitían tanta reflectancia en este espectro, y por esa razón el algoritmo de búsqueda de zonas de interés supo diferenciar estas dos zonas.



*Ilustración 21. Resultados obtenidos en la imagen espectral de luz ultravioleta.*

La Ilustración 21 nos muestra los resultados obtenidos de la imagen espectral de luz ultravioleta. Este espectro dio los mismos resultados que el espectro azul, solo existe una diferencia, la cual es el tamaño de los elementos encontrados, aunque no es tan grande la diferencia.



*Ilustración 22. Resultados obtenidos en la imagen espectral de luz violeta.*

En la Ilustración 22 se observan los resultados obtenidos del espectro violeta. Este espectro fue el único que nos dio una clasificación diferente al de los otros espectros. Aunque encontró las mismas zonas que los demás, su clasificación consistió en tres grupos.

En estos resultados se puede observar el distinto comportamiento que tienen los contaminantes al aplicarles diferentes espectros de luz. El comportamiento que más nos interesa es la intensidad de reflectancia para cada espectro, porque como se puede ver en las imágenes, se encontraron las mismas zonas, pero en varias de ellas hubo una variación en su clasificación debido a su intensidad de reflectancia.

## Capítulo 7. Conclusiones

Después de seguir el método propuesto, es posible darse cuenta de varios puntos importantes, con los que se puede hacer una búsqueda y clasificación de contaminantes en imágenes hiperespectrales con mayor eficacia, tanto en rapidez como en exactitud, sin la necesidad de gastar una alta cantidad de dinero. A continuación se mencionaran estos puntos, junto con su respectiva descripción.

- La **posterización con valor de 64** es de gran ayuda al momento de reducir el ruido en las imágenes espectrales sin perder información visual, y a su vez al hacer esto, la simbología obtenida con los códigos de cadena es más uniforme lo que facilita el análisis de sus símbolos.
- La **compresión de los códigos de cadena utilizando COMI** reduce en gran medida los tamaños de los archivos, siendo capaz de darnos mejores resultados que los obtenidos en la literatura. Siendo esta una de las razones principales para trabajar con códigos de cadena en imágenes espectrales.
- Es necesario realizar más experimentos y usar más códigos de cadena, pero por el momento el **código que dio mejores resultados es AF8**. Por mejores resultados se refiere a la facilidad y velocidad que nos da para el análisis de sus datos.
- Una de las ventajas del código AF8 es que es invariante ante rotación, aunque también puede ser una desventaja. En este caso es ambas; es ventaja porque se usan menos símbolos (simbología, no cantidad de símbolos), y desventaja porque sus símbolos no dicen mucho por si solos, por esta razón **se usa un código auxiliar para conocer la dirección de movimiento** en la que se encuentra en cada pixel. Esto es necesario cuando buscamos las zonas de interés, antes explicado en el capítulo 5
- La manera en cómo se codificaron las imágenes es haciendo un recorrido zig-zag en horizontal, por lo que el resultado final puede verse como si fuera la forma 3D del objeto, pero no es así, simplemente es un conjunto de líneas 2D horizontales, por lo que una vez encontradas las zonas de interés **es necesario obtener líneas**



**2D verticales, ya que de esta forma se puede analizar la morfología** de cada zona, dándonos la facilidad de clasificarlas.

- Se realizó un artículo con el tema “Compresión de Imágenes Hiperespectrales usando Códigos de Cadena” el cual fue presentado en Zurich, Suiza en la 5ta Conferencia de Avances en Computación, Comunicación y Tecnologías de la Información (CCIT 2017) [36].



## Capítulo 8. Futuros trabajos

Con los resultados obtenidos a lo largo de la investigación, se pudieron identificar puntos de mejora tanto para los algoritmos como para el método. A continuación, se plantean algunas sugerencias de las posibles mejoras.

- Se analizarán los resultados al usar otros códigos de cadena, ya que en este trabajo usamos los básicos (F8, AF8, F26). Una vez obtenidos estos resultados se compararán el tiempo de procesamiento, cantidad y uniformidad de los símbolos, entre otros, para poder obtener una conclusión final de cuál de todos los códigos de cadena existentes es mejor para trabajar con este método.
- Mejorar el algoritmo de búsqueda de zonas de interés, para mejorar el tiempo de procesamiento. Esto se puede lograr evitando el doble trabajo, el cual consiste en codificar la imagen y después analizar los códigos. Ya que el código 012 nos permite trabajar al mismo tiempo que se codifica, los algoritmos de codificación y búsqueda de zonas de interés se pueden juntar en uno mismo, reduciendo significativamente el tiempo.

## Capítulo 9. Referencias

- [1] L. E. Sucar y G. Gómez, *Visión Computacional*, <http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>, 2012.
- [2] J. J. Gibson, *The Ecological Approach to Visual Perception*, Boston: Houghton Mifflin, 1979.
- [3] D. Marr, *Vision*, San Francisco: Freeman, 1982.
- [4] F. D. van der Meer y S. M. de Jong, *Imaging Spectrometry*, New York: Kluwer Academic Publishers, 2002.
- [5] H. Freeman, «On The Encoding of Arbitrary Geometric Configurations,» *IRE Transactions on Electronic Computers EC-10*, pp. 260-268, 1961.
- [6] H. Sánchez-Cruz y R. M. Rodríguez-Dagnino, «Compressing bilevel images by means of a three-bit chain code,» *Optical Engineering SPIE*, vol. 44, pp. 1-8, 2005.
- [7] E. Bribiesca, «A new chain code,» *Pattern Recognition*, vol. 32, pp. 235-251, 1999.
- [8] H. Freeman, «Computer processing of line drawings,» *ACM Computing Surveys*, vol. 6, pp. 57-97, 1974.
- [9] Y. Kui-Lui y B. Zalik, «An efficient chain code with Huffman coding,» *Pattern Recognition*, vol. 38, pp. 553-557, 2005.
- [10] R. G. Burns, «Mineralogical Application to Crystal Field Theory,» *Cambridge University Press*, p. 224, 1970.
- [11] G. R. Hunt y J. W. Salisbury, «Visible and near-infrared spectra of minerals and rocks, Sedimentary rocks,» *Modern Geology*, vol. 5, pp. 211-217, 1976.
- [12] H. Y. Chiu y W. Collins, «A spectroradiometer for airborne remote sensing,» *Photogrammetric Engineering and Remote Sensing*, vol. 44, pp. 507-517, 1978.
- [13] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen y W. M. Porter, «The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS),» *Remote Sensing Environment*, vol. 44, pp. 121-144, 1993.

- TESIS TESIS TESIS TESIS TESIS
- [14] G. Shaw y D. Manolakis, «Signal processing for hyperspectral image exploitation,» *IEEE Signal Processing Magazine*, vol. 19, pp. 12-16, 2002.
- [15] D. Landgrebe, «Hyperspectral Image Data Analysis,» *IEEE Signal Processing Magazine*, vol. 19, pp. 17-28, 2002.
- [16] J. W. Boardman, F. A. Kruse y R. O. Green, Mapping target signatures via partial unmixing of AVIRIS data, California: Summaries of the VI JPL Airbone Earth Science Workshop, 1995.
- [17] M. J. Leboffe y B. E. Pierce, A Photographic Atlas for the Microbiology Laboratory, Colorado: Morton Publishing Company, 2011.
- [18] S. Bradbury y B. Bracegirdle, Introduction to Light Microscopy, New York: Springer-Verlag New York Inc., 1998.
- [19] L. Shore, Diagnostic Medical Parasitology, Washington, D.C: ASM Press, 2006.
- [20] R. N. Clark, «Spectroscopy of Rocks and Minerals, and Principles of Spectroscopy,» de *Manual of Remote Sensing*, New York, John Wiley and Sons, 1999.
- [21] S. V. Stehman, «Selecting and Interpreting Measures of Thematic Classification Accuracy,» *Remote Sensing of Environment*, vol. 62, pp. 77-89, 1997.
- [22] S. S. Chiang, C. I. Chang y I. W. Ginsberg, «Unsupervised target detection in hyperspectral images using projection pursuit,» *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 1380-1391, 2001.
- [23] N. Keshava y J. F. Mustard, «Spectral unmixing,» *IEEE Signal Processing Magazine*, vol. 19, pp. 44-57, 2002.
- [24] C. C. Funk, J. Theiler, D. A. Roberts y C. C. Borel, «Clustering to improve matched filter detection of weak gas plumes in hyperspectral thermal imagery,» *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 1410-1420, 2001.
- [25] A. Ifarraguerri y C. I. Chang, «Multispectral and hyperspectral image analysis with projection pursuit,» *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, pp. 2529-2538, 2000.
- [26] L. M. Bruce, C. Morgan y S. Larsen, «Automated detection of subpixel targets with continuous and discrete wavelet transforms,» *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 2217-2226, 2001.

- [27] L. G. Shapiro y G. C. Stockman, *Computer Vision*, Prentice Hall, 2001.
- [28] T. Huang., G. Yang y G. Tang, «A fast two-dimensional median filtering algorithm,» *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 27, nº 1, pp. 13-18, 1979.
- [29] M. Langford, *The Darkroom Handbook*, New York: Dorling Kindersley Limited, 1981.
- [30] R. Gonzalez, R. Woods y S. Eddins, *Digital Image Processing using Matlab*, Prentice Hall, 2003.
- [31] M. J. Weinberger, G. Seroussi y G. Sapiro, «The LOCO-I lossless image compression algorithm: principles and standarization into JPEG-LS,» *IEEE Trans. Image Process*, vol. 9, pp. 1309-1324, 2000.
- [32] X. Wu y N. D. Memon, «Context-based, adaptative, lossles image coding,» *IEEE Trans. Commun*, vol. 45, pp. 437-444, 1997.
- [33] D. A. Huffman, «A method for the construction of minimum-redundancy codes,» *Proc. IRE*, vol. 40, pp. 1098-1101, 1952.
- [34] D. J. MacKay, «Information Theory, Inference, and Learning Algorithms,» *Cambridge University Press*, vol. 7, pp. 110-130, 2003.
- [35] N. Board, *The complete Book on Printing Technology*, Asia Pacific Business Press Inc., 2003.
- [36] I. Chávez-Delgado, H. Sánchez-Cruz y J. H. Sossa-Azuela, «Hyperspectral image compression using chain codes,» de *Advances in Computing, Communication and Information Technology*, Zurich, 2017.