Desarrollo de una Librería en Código Libre Para Clasificar Imágenes Hiperespectrales por Mapeo del Angulo Espectral

Presentado por:

ALVARO ANDRES PARRA BARRETO

Dirigido por:

Ing. CARLOS GERMÁN RAMÍREZ RAMOS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

PROYECTO CURRICULAR DE INGENIERÍA CATASTRAL Y GEODESIA

COLOMBIA

2015

TABLA DE CONTENIDO

CONTENIDO	Pág.
1. INTRODUCCIÓN	01
2. FORMULACION DEL PROBLEMA	03
2.1 PREGUNTAS DE INVESTIGACIÓN	04
3. JUSTIFICACIÓN	05
4. OBJETIVOS	06
3.1 Objetivo general	06
3.2 Objetivos específicos	06
5. MARCO TEÓRICO	07
5.1. TELEDETECCIÓN	07
5.2. HERRAMIENTAS SOFTWARE	07
5.2.1 ERDAS IMAGINE	07
5.2.2 ENVI:	08
5.2.3 GRASS	08
5.3. SATÉLITE EO-1	09
5.3.1 ALI	09
5.3.2 Hyperion	09
5.3.3 LEISA/AC	10
5.4. IMAGEN HIPERESPECTRAL	10
5.5. CLASIFICACIÓN DE IMAGENES HIPERESPECTRALES	12
5.5.1. Análisis de Mezcla Espectral (SMA)	12
5.5.2. Análisis del Pixel	13
5.5.3. Montaje característica espectral (SFF)	13

5.5.4. Análisis del Sub-Pixel 13

5.5.5. Desmezcla Lineal Completa 14

5.5.6. Filtro combinado 14

5.5.7. Mapeo del Angulo Espectral (SAM) 14

5.6. ANTECEDENTES DE INVESTIGACIÓN 16

5.7. SERVICIO GEOLÓGICO DE LOS ESTADOS UNIDOS (USGS) 18

5.8. IMAGEN HIPERESPECTRAL DE CALI 18

5.9. CALI 19

6. METODOLOGÍA 20

6.1. Inicio 20

6.2. Elaboración 20

6.3. Construcción 23

6.4. Transición 23

7. RESULTADOS 31

8. CONCLUSIONES 38

9. BIBLIOGRAFIA 39

10. ANEXOS 43

INDICE DE IMAGENES **Pág.**

1. Imagen 01: Cubo hiperespectral (Hipercubo). 11
2. Imagen 02: Descripción gráfica del método SAM. 15
3. Imagen 03: Salida gráfica imagen hiperespectral Cali. 19
4. Imagen 04: Interfaz Gráfica r.sam. 31
5. Imagen 05: GUI GRAS . 31
6. Imagen 06: Error al cargar una sola banda. 32
7. Imagen 07: Lista desplegable bandas. 32
8. Imagen 08: Error de parámetros. 33
9. Imagen 09: Parámetros X y Y. 33
10. Imagen 10: Parámetro ángulo. 33
11. Imagen 11: Nombre de las salidas graficas. 34
12. Imagen 12: Pestaña opcional. 34
13. Imagen 13: Parámetros de prueba. 35
14. Imagen 14: Porcentaje de avance. 35
15. Imagen 15: Comando de finalización y tiempo. 36
16. Imagen 16: Salida SAM. 36
17. Imagen 17: Salida Mascara. 37

1. INTRODUCCIÓN

Al descubrir la interrelación entre el color, la frecuencia y la longitud de onda por William Herschel, se revelaron los principios usados para caracterizar la incidencia de la luz sobre los objetos [5], principios que hacen parte fundamental de la teledetección.

La teledetección se ha convertido en una herramienta esencial para el estudio, seguimiento y comprensión de fenómenos no solo de la superficie terrestre sino también de los cuerpos celestes.

Los sensores remotos han experimentado grandes avances, desde las primeras cámaras pancromáticas usadas para fotogrametría aérea, pasando por los scanner multiespectrales (transportados por satélites), hasta el sensor CASI (con una resolución espectral de 288 bandas [6]). Ese avance se refleja en su producto "la imagen", la cual incrementó la cantidad de datos almacenados, pasando de multiespectral a hiperespectral; permitiendo de ese modo, mejorar la exactitud en la capacidad de análisis y síntesis, sin embargo, esto demanda mejores capacidades tanto software como hardware para realizar los procesos [37].

Dentro de las diferentes aplicaciones Software especializadas en procesar datos de sensores remotos o raster se conocen Erdas, Envi, Ilwis, Grass, entre otros; los dos primeros son comerciales y el último es Software libre. Los procesos más relevantes que permiten estos software son: la manipulación de información raster y vector, georreferenciación y segmentación de imágenes, análisis espectrales y la clasificación de imágenes [11] [35].

Los datos raster o imágenes se encuentran en diferentes formatos y tipos; los tipos más utilizados en la actualidad son los multiespectrales, pero desde la década pasada el análisis de imágenes hiperespectrales ha tenido gran actividad. La imagen hiperespectral es el producto del desarrollo tecnológico de los sensores espectrales [37].

La clasificación de las imágenes es una tarea que se realiza con el propósito de convertir datos cuantitativos en cualitativos [29], para lograrlo existen diferentes métodos que se han implementado en esta compleja tarea, los más relevante son: máxima probabilidad, Isodata, Fuzzy y, dentro de los algoritmos de inteligencia artificial están la retro-propagación y los árboles de decisiones [25] [8].

El tipo de imagen determina la forma de clasificación, para imágenes hiperespectrales los métodos que predominan son: Mapeo del Angulo Espectral, Análisis de Mezcla Espectral, Análisis de todo el Pixel, Montaje característica espectral, Análisis del Sub-Pixel, entre otros.

El software Grass posee pocas herramientas para el procesamiento de imágenes hiperespectrales, especialmente que permita la clasificación por Mapeo del Ángulo Espectral (por sus siglas en ingles SAM, Spectral Anguler Mapper). Generando la necesidad de desarrollar una librería que permita esa clasificación en el software, migrando la ecuación SAM a un lenguaje de programación soportado por Grass para que se ejecute sobre su kernel; lo que permitirá que los usuarios accedan a una herramienta en software libre para realizar la clasificación SAM.

2. FORMULACIÓN DEL PROBLEMA

El Ingeniero Catastral y Geodesta tiene como objetivo el estudio del recurso tierra utilizando las ciencias básicas, métodos de ingeniería y las ciencias de la tierra, con el apoyo de herramientas tecnológicas [38], en los últimos años la herramienta de mayor apoyo para el desarrollo de sus actividades es el software, los de mayor relevancia son: ArcGis, GvSig, QGis, Geomedia, MicroStatión, Erdas, Envi y Grass; los tres últimos están especializados en el procesamiento de imágenes.

Para cumplir sus objetivos el ingeniero Catastral y Geodesta puede encontrar en el mercado diferentes productos como fotografías aéreas, imágenes de radar, imágenes multiespectrales e hiperespectrales, para este último producto que es relativamente nuevo, los software especializados descritos en el párrafo anterior ya cuentan con módulos que permiten no solamente su despliegue en el visor, sino también su procesamiento.

Al analizar los tres software especializados en datos raster, se identifica que Grass (software libre) cuenta con la menor cantidad de aplicaciones en el módulo para imágenes hiperespectrales; el creciente uso de Grass en los diferentes ámbitos profesionales, genera la necesidad de suplir esa falencia. Esta herramienta tiene la ventaja de estar bajo la licencia GPL (software libre), lo que permite desarrollar librerías para suministrar algoritmos a sus usuarios, sin las restricciones de los software comerciales.

Dentro de la información consultada previamente y descrita en el marco teórico se observa que el método de clasificación en imágenes hiperespectrales con mayor implementación y documentación es el método SAM (por sus siglas en inglés Spectral Angle Mapper). Este método mediante su ecuación convierte los niveles digitales de la imagen en reflectancia aparente, lo que permite evaluar la similitud entre un espectro desconocido y el de referencia en las n-bandas que componen la imagen [3].

Partiendo de lo expuesto anteriormente, se identifica en Grass la inexistencia de una librería en el módulo hiperespectral que permita la aplicación del método de clasificación SAM, para su uso profesional.

2.1. PREGUNTAS DE INVESTIGACIÓN

- ¿Qué sistemas operativos soportan el software Grass?
- ¿Sobre cuál lenguaje de programación se desarrolló Grass?
- ¿El método de clasificación SAM puede ser implementado en el lenguaje de programación en que se desarrolló Grass?
- ¿Qué tan eficiente puede ser la clasificación SAM al procesarla sobre Grass?
- ¿Se cuenta con el acceso a una imagen hiperespectral para realizar las pruebas de la implementación del método SAM?

3. JUSTIFICACIÓN

Las ciencias de la tierra no se escapan a las bondades que ha traído consigo la evolución tecnológica, dentro de los beneficios recibidos se encuentran: el hardware, los sistemas de posicionamiento global, los sensores remotos, los satélites y no menos importante el software. Esos mismos avances han permitido durante los últimos años la automatización de muchos procesos que normalmente eran análogos.

Al analizar Grass se identifica que no tiene implementado dentro de su módulo para imágenes hiperespectral el método de clasificación Mapeo del Ángulo Espectral (SAM). Método que evidentemente en las fuentes consultadas, es usado con mayor frecuencia en las investigaciones con imágenes hiperespectrales. Partiendo de estas premisas, se evidencia una falencia del software; para la comunidad que en la cotidianidad o en sus investigaciones requieren aplicar la metodología SAM sin adquirir software comercial, sería de gran ayuda encontrar en Grass esa solución.

Como consecuencia, surge la necesidad de desarrollar una librería en Grass que permita realizar el método de clasificación Mapeo del Ángulo Espectral (SAM), no solo para aquellos que se encuentren actualmente interesados en el procesamiento de imágenes hiperespectrales, sino también, para aquellos usuarios interesados en las ciencias de la tierra.

4. OBJETIVOS

4.1 Objetivo general

Desarrollo de una librería en código libre (Software Grass) para clasificar imágenes hiperespectrales por el método Mapeo del Ángulo Espectral (SAM por sus siglas en inglés Spectral Anguler Mapper).

.

4.2 Objetivos específicos

- Identificar el lenguaje de programación para el desarrollo de módulos en Grass
- Determinar las librerías básicas desarrolladas en Grass que sirven de soporte para la librería SAM.
- Desarrollo del código en el lenguaje soportado por Grass que permite la lectura de los datos raster.
- Verificar que la lectura de los datos de la imagen sean verídicos.
- Migrar la ecuación de clasificación Mapeo del Ángulo espectral (SAM) al lenguaje soportado por Grass.
- Comprobar el correcto procesamiento de los datos.
- Evidenciar la salida gráfica de los resultados en el visor de Grass.

5. MARCO TEÓRICO

5.1. TELEDETECCIÓN

En teledetección la capacidad de obtener la mayor cantidad de datos espectrales es clave para la toma; debido a que gran cantidad de datos espectrales deben ser sacrificados para definir la firma espectral de interés.

La firma espectral pura de un elemento se llama endmember. Un método para la toma de endmembers es el laboratorio espectrorradiómetro que se centra en una sola superficie o material. Estas firmas son utilizadas en el sensor y en algoritmos de detección, los cuales sirven para definir y perfeccionar la escena tomada y así poder discriminar entre un material o materiales de similares características. Sin embargo, cuando el material de interés no está disponible para las mediciones de laboratorio, debe ser definida dentro de la escena espectral tomada.

5.2. HERRAMIENTAS SOFTWARE

Las herramientas software destacadas para el procesamiento de datos raster (caso de estudio imágenes hiperespectrales) son: Erdas, Envi y Grass. Las cuales son descritas a continuación:

5.2.1 ERDAS IMAGINE: Es una herramienta software de teledetección que permite la edición de imágenes para aplicaciones geoespaciales [24]. La última versión es la del 2015. ERDAS IMAGINE está dirigido principalmente al procesamiento de imágenes y permite al usuario preparar, mostrar y mejorar las imágenes digitales para su uso en sistemas de información geográfica (SIG) o en software de diseño asistido por computador (CAD). Se trata de un paquete de herramientas que le permite al usuario realizar numerosas operaciones en una imagen y generar respuestas de análisis geográficos.

Mediante la manipulación de los valores y las posiciones de datos de imágenes, es posible ver las características que normalmente no serían visibles y para localizar las características que de otro modo serían gráficos. Otros ejemplos de uso incluyen la extracción de características lineales, generación de flujos de trabajo de procesamiento ("modelos espaciales" en ERDAS IMAGINE), importar/exportar datos para una amplia variedad de formatos, orto-rectificación, mosaicos de imágenes,

georreferenciación y extracción de características automáticas de datos en los mapas a partir de imágenes [11].

5.2.1. ENVI: Es una herramienta software que combina procesamiento avanzado de imagen espectral y la tecnología de análisis geoespacial. Soporta imágenes pancromáticas, de LIDAR, SAR, imágenes multiespectrales o hiperespectrales, ENVI tiene herramientas de procesamiento y análisis para ayudar a extraer información. Esta herramienta trabaja por medio de cinco módulos para el análisis avanzado de imágenes. Los módulos de ENVI permiten trabajar con datos NITF, crear DEM y Ortorectificar imágenes entre otras funciones. Actualmente ENVI permite la interoperabilidad con otra herramienta de gran uso para los sistemas de información geográfica (SIG) la cual es ArcGis [12].

5.2.2. GRASS:

El software Grass Gis (por sus siglas en inglés Geographic Resources Analysis Support System), es un Sistema de Información Geográfica que proporciona potentes herramientas raster y vectoriales, así como un motor de procesamiento geoespacial en una única suite software integrada. Grass incluye herramientas para el modelado espacial, la visualización de datos raster y vectoriales, la gestión y el análisis de datos geoespaciales, y el procesamiento de imágenes aéreas y de satélite. Posee también la capacidad de producir gráficos complejos y de generar mapas.

GRASS se ha traducido a veinte idiomas diferentes y soporta una enorme variedad de formatos de datos [40]. Incluye más de 400 módulos de análisis integrados. Adicionalmente, más de 100 módulos y herramientas de libre acceso en el sitio Wiki. Tras 30 años de desarrollo, es el mayor y más antiguo GIS de código abierto disponible. Se puede desarrollar fácilmente con él a través de scripting, siendo suficientemente versátil tanto para ser utilizado en procesos por lotes de datos en paralelo sobre supercomputadoras, como manualmente a través de la interfaz de escritorio, disponible también para PDA's y netbooks [40]. Grass se utiliza actualmente en todo el mundo, tanto en actividades académicas, como comerciales, así como en muchas agencias gubernamentales y consultorías de Medio Ambiente. Corre en variadas plataformas hardware y

es un software libre y de código abierto, publicado bajo los términos de licencia GNU (por sus siglas en ingles General Public License) [40].

5.3. SATÉLITE EO-1

Fue desarrollado con una serie de tecnologías e instrumentos así como su nave espacial, diseñados para permitir la incursión a futuros observatorios de imágenes, lo cual significa aumento y rendimiento simultaneo. Más específicamente, EO-1 ha validado un sensor multiespectral que es mejor significativamente que el sensor Landsat 7 ETM; ha desarrollado un sensor de imágenes hiperespectrales para la tierra y la ciencia; se ha validado la capacidad de la resolución espectral de imágenes de alta / baja espacial que puede corregir los errores sistemáticos en las reflectancias aparentes de la superficie, causados por efectos atmosféricos, principalmente vapor de agua. Las tecnologías de la nave espacial fueron evaluadas y han sido puestos a disposición para futuras misiones. La misión Earth Observing-1 (EO-1) Es el primer satélite del programa de observación terrestre para el nuevo Milenio de la NASA [10] [7]. Fue lanzado el 21 de Noviembre del año 2000. El satélite EO-1 dispone de tres sensores de teledetección:

5.3.1. ALI (Advanced Land Imager) Captura de Imágenes Terrestres

5.3.2. Hyperion (Imaging Spectrometer) Es el primer sensor hiperespectral desde satélite, las imágenes capturan la superficie de la Tierra sin dejar áreas descubiertas, tiene la habilidad de discriminar imágenes.

El instrumento Hyperion ofrece una nueva clase de datos para observar la Tierra con una mejor caracterización de la superficie. Hyperion ofrece un instrumento de grado científico, la calibración se basa en el patrimonio del Instrumento de imágenes hiperespectrales LEWIS. Las capacidades de Hyperion proporcionan una resolución en cientos de bandas espectrales, frente a las diez bandas multiespectrales tradicionales formadas en las misiones Landsat. A través de estas bandas espectrales, los ecosistemas terrestres complejos se pueden visualizar y clasificar con una mejor precisión.

Hyperion proporciona imágenes hiperespectral de alta resolución, capaz de

resolver 220 bandas espectrales a (0,4 a 2,5 micras) con una resolución de 30 metros. El instrumento captura imágenes con una superficie terrestre de 7,5 por 100 kilómetros por cada imagen, y brinda cartografía espectral detallada a través de los 220 canales con alta precisión radiométrica. Los principales componentes del instrumento son los siguientes:

El sistema óptico delantero se diseñó basado en la misión del Satélite Multi- Propósito de Corea (KOMPSAT); cámara óptica (EOC); el telescopio prevé dos espectrómetros rejilla de imágenes separado para mejorar la relación señal-ruido (SNR).

5.3.3. LEISA/AC (Linear Imaging Spectrometer Array/Atmospheric Corrector) es un conjunto de tecnologías que captura imágenes hiperespectrales. Como las imágenes transmitidas por satélites son distorsionadas por los gases en la atmósfera, el AC restaura las imágenes a su forma original.

5.4. IMAGEN HIPERESPECTRAL

En su interés por observar la tierra de forma remota, el ser humano ha desarrollado instrumentos de observación y de medida cuya resolución ha sido cada vez mayor en lo espacial y en lo espectral; el resultado más avanzado es la construcción de imágenes hiperespectrales.

El término "multi" se deriva de la palabra latina para "muchos" e "híper" es la palabra griega que significa "sobre", "arriba" o una "cantidad exagerada"; estos prefijos junto con "espectral", que se refiere a colores, se combinan para formar "multiespectral" e "hiperespectral", que en sentido figurado significa "muchos colores". La ciencia de la teledetección multiespectrales e hiperespectrales se basa en tomar una porción del espectro electromagnético y segmentarlo con el fin de poder hacer cálculos analíticos [5].

Los sensores hiperespectrales desarrollados permiten capturar imágenes digitales en una amplia cantidad de canales espectrales muy cercanos entre sí, generando una firma espectral característica de cada material por cada porción de la escena o píxel. Si se intentara representar de manera gráfica una imagen hiperespectral se obtendría un cubo cuyas dos primeras dimensiones permitirían identificar la posición en el espacio de un pixel determinado de la imagen, mientras que la

tercera dimensión representaría la singularidad espectral de cada píxel de acuerdo con las diferentes longitudes de onda, en donde cada longitud de onda correspondería a un nivel de profundidad espectral. La imagen 01 es una representación gráfica del concepto de imagen hiperespectral, también llamada cubo hiperespectral o hipercubo.

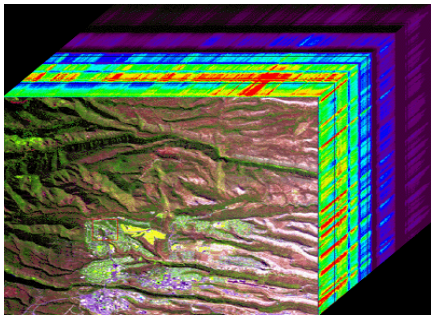

Imagen 01: Cubo hiperespectral (Hipercubo) [28]

Durante el proceso de captura de imágenes de la superficie terrestre, los sensores hiperespectrales emplean diferentes longitudes de onda, generando como resultado lo que se conoce como banda o canal, que representa la tercera dimensión de la imagen hiperespectral. De esta forma, un píxel localizado en las coordenadas (x, y) es un vector de bandas, en el que cada elemento corresponde a una diferente longitudes de onda empleada en el proceso de captura de imágenes.

El cubo hiperespectral o hipercubo representa entonces el resultado de la captura de imágenes, y en él, se encuentran las dimensiones que permiten la ubicación espacial de cada píxel para cada longitud de onda que es una herramienta útil para el análisis global de la imagen.

Muchas implementaciones de análisis en imágenes hiperespectrales sirven para detectar e identificar objetos, requiriendo un alto grado de confianza. Pero el incremento del número de bandas, sin embargo, no conduce necesariamente a una identificación precisa; el uso de muchas bandas también puede tener un efecto de "rendimiento decreciente" o incluso un impacto negativo en la capacidad de explotar la escena espectral. Teniendo en cuenta todos los posibles conjuntos de bandas para una sola evaluación que conlleva mucho tiempo, los métodos analíticos fueron desarrollados para ayudar con la selección de bandas [5].

5.5. CLASIFICACIÓN DE IMAGENES HIPERESPECTRALES

A pesar que los actuales sensores remotos y sistemas de recolección de datos a menudo crean conjuntos de datos extremadamente grandes que son difíciles de trabajar, la información contenida en estos es muy útil. Una opción para el proceso de clasificación de la imagen es la extracción de características. Esta opción reduce las características espectrales o espaciales con transformaciones espectrales o filtros espaciales para los conjuntos de datos que pueden ser fácilmente transformados y explotados. La extracción de las características también puede lograrse mediante la selección de un subconjunto de bandas basadas en las características de ciertos artículos de interés.

La mayoría de los algoritmos sobre imágenes hiperespectrales requieren el acceso a los atributos espectrales. Las bibliotecas espectrales contienen espectros de referencia ya sea medido o simulado desde el campo y las colecciones de laboratorio de los datos de reflectancia o la radiación.

Los métodos de análisis espectral suelen utilizar algoritmos de análisis de imágenes para comparar los espectros de píxeles con un espectro de referencia. Los algoritmos más utilizados para el procesamiento de imágenes hiperespectrales son: análisis de mezcla espectral, análisis de todo el píxel, montaje de la característica espectral, análisis del Sub-píxel, desmezcla espectral lineal completa, Filtro combinado y Cartografía del ángulo espectral (SAM por sus siglas en ingles). Los cuales se describen a continuación:

5.5.1. Análisis de Mezcla Espectral (SMA): Un método para aislar las características espectrales se llama Análisis de Mezcla Espectral (SMA por sus siglas en ingles). SMA es un método estructurado que aborda el problema de píxeles mixto y otros factores que contribuyen a la calidad de la imagen, tales como la calibración y las condiciones de luz. Cada endmember se selecciona en función de su material y su contribución a la escena espectral general. Este método funciona mejor cuando la diversidad espectral y el contenido de la escena no son complejos y las características espectrales de interés son muy pequeñas en la escena [5].

5.5.2. Análisis del Pixel: El método de análisis de todo el píxel se utiliza para determinar si uno o más materiales son abundantes dentro de cada píxel. La similitud espectral del píxel objetivo se compara entonces con los materiales en una biblioteca de referencia. El análisis de todo el píxel se usa a menudo contra el estándar de clasificadores supervisados, el SAM y métodos espectrales de análisis para características de ajuste [5].

5.5.3. Montaje de Característica Espectral (SFF): Otro método para comparar espectros de destino y el pixel es mediante el examen de las características específicas de absorción en los espectros (SFF por sus siglas en ingles). SFF es un algoritmo de detección que utiliza los espectros de la imagen que se hacen coincidir con endmembers de referencia. Con SFF, el usuario especifica un rango de longitudes de onda dentro de la unidad característica de absorción para el elemento elegido. Los espectros de pixel son continuos, en comparación con el espectro de destino mediante dos mediciones. La primera medición es la profundidad de la característica en el pixel, que se compara con la profundidad de la función en el objetivo. La segunda medición es la forma de la característica en el pixel, que se compara con la forma de la función en el destino mediante una técnica de mínimos cuadrados. Un ejemplo avanzado de este método, llamado Tetracorder, ha sido desarrollado por el Servicio Geológico de EE.UU [5].

5.5.4. Análisis del Sub-Pixel: El método de análisis de sub-píxel es un algoritmo de detección muy potente que puede utilizarse para calcular la cantidad de materiales en cada píxel de una imagen. El análisis del sub-píxel puede detectar las cantidades de un objetivo que son mucho más pequeño que el tamaño del pixel en sí. En los casos de buen contraste espectral entre un blanco y su fondo, se ha detectado que los objetivos abarcan el 13% del píxel. Estos algoritmos se encuentran en la discriminación de la firma espectral. Los objetivos que llenan varios píxeles también tienen una mayor significancia estadística para una mejor probabilidad de detección. Sin embargo, cuando el objeto de interés es significativamente pequeño, estos algoritmos pueden conducir a la detección inicial y la posibilidad de identificación adicional. La desventaja del análisis del sub-pixel es el número de detecciones falsas que se producen cuando se precisan mediciones espectrales con una cantidad limitada de información [5].

5.5.5. Desmezcla Lineal Completa: Desmezcla espectral lineal completa es un método para realizar detección de sub-píxeles basado en la teoría de que cada píxel contiene una combinación lineal de los espectros de reflectancia de todos los endmembers dentro del píxel (Adams et al., 1986). Si la relación es lineal, entonces hay más contenido espectral, mayor cantidad que el material contenido dentro del píxel. Esta relación puede ser utilizada para calcular el contenido del material dentro de la escena basada en la firma espectral de la endmember buscado. La desmezcla espectral se logra mediante un conjunto de ecuaciones lineales basadas en el número de bandas en la imagen. La cantidad desconocida se resuelve para determinar la fracción de cada endmember en el píxel [5].

5.5.6. Filtro combinado: Junto con la desmezcla espectral lineal completa, el filtro combinado es otro método para realizar la detección de sub-píxeles. El filtro combinado es una metodología especializada para desmezcla espectral que sólo utiliza endmembers definidos específicamente para los mapas espectrales; también se conoce como desmezcla parcial. El filtro combinado se utiliza para encontrar las firmas espectrales de los materiales que constituyen una parte relativamente menor del contenido espectral de la imagen. Esta técnica requiere de un análisis computacional mucho menor y, por tanto, puede proporcionar un medio rápido para analizar, visualizar y explotar las imágenes. El método se utilizó en las imágenes hiperespectrales tomadas inmediatamente después de la tragedia del transbordador espacial Columbia en Texas el 1 de febrero de 2003. Era fundamental encontrar muchos fragmentos del transbordador espacial como sea posible antes de que el material fuera degradado debido a la exposición a los elementos ambientales [5].

5.5.7. Mapeo del Angulo Espectral (Spectral Anguler Mapper SAM): El método SAM considera todos los píxeles de la imagen y evalúa la similitud de los espectros para reprimir la influencia de la sombra, lo que acentúa las características de reflectancia. El espectro de la imagen se le asigna un factor de correlación entre cero (0) baja correlación y uno (1) alta correlación, se compara con una biblioteca espectral [5]. Con el método SAM, los datos son convertidos a reflectancia aparente, que es la reflectancia cierta con coeficientes de ganancia que se definen por el terreno y las condiciones de iluminación. Un gráfico de dispersión de los

valores de los píxeles de dos bandas de una imagen espectral se puede generar para visualizar los componentes espectrales. La imagen 02 muestra el espectro del píxel y el espectro de destino como puntos.

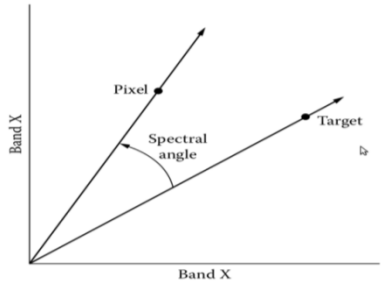

Imagen 02: Descripción grafica del método SAM

El ángulo espectral es el ángulo entre dos vectores procedentes de un origen común. La magnitud del ángulo indica la similitud o disimilitud de los materiales, un ángulo más pequeño se correlaciona a una firma espectral más similar. Este método es relativamente insensible a los cambios en la iluminación en el material objetivo porque los cambios en la luz afectarán la magnitud pero no la dirección del vector. Un objetivo poco iluminado hará que los puntos se vayan a representar cerca del origen [5]

La ecuación que representa el algoritmo Spectral Anguler Mapper es el siguiente:

$$\propto = cos^{-1}\left[\frac{\sum_{i=1}^{n} t_i\, r_i}{\sqrt{\sum_{i=1}^{n} t_i^2 \sum_{i=1}^{n} r_i^2}}\right]$$

Ecuación de Clasificación SAM [31]

En donde t es el espectro de estudio y r el espectro de referencia. El número n representa el número de bandas estudiadas y α es el ángulo entre vectores

5.6. ANTECEDENTES DE INVESTIGACIÓN

En Estados Unidos la primera misión de la Administración Nacional Oceánica y Atmosférica (NOAA por sus siglas en inglés), es la de conocer y predecir los cambios en el ambiente de la tierra la cual requiere una continua capacidad para adquirir, procesar y archivar datos en tiempo real[27]; para tal propósito esta entidad hace uso de las herramientas satelitales, dentro de las que se encuentra el espectrómetro geosincrónico para la representación de la transformada de Fourier (en inglés, Geosynchronous Imaging Fourier Transform Spectrometer GIFTS), el cual provee imágenes del orden de 1.5 terabytes por día [16]. Este volumen de información genera un problema en su procesamiento y archivo en tiempo real debido a las altas tasas de transferencia de datos, para lo que se desarrolló un framework en programación paralela para un clúster que emplea un sistema multi-procesador que combinado con un sistema de archivos paralelos. Este arreglo logra una eficiencia del 85.9% por un número total de 64 procesadores sobre 64 imágenes hiperespectrales de 128 por 128 píxeles [1].

En España el Departamento de Tecnología de Computadores y Comunicaciones de la Universidad de Extremadura, mediante la plataforma de computación paralela Thunderhead localizada en el Centro de Vuelo Espacial Goddard de la NASA en Maryland [17], desarrolló el proyecto “Procesamiento Paralelo Espacio-Espectral de Imágenes Hiperespectrales” .La principal contribución fue el desarrollo y validación de un conjunto de técnicas innovadoras para el análisis espacio-espectral de imágenes hiperespectrales en la reducción dimensional, clasificación no supervisada, análisis de mezcla espectral, compresión de datos y las implementaciones paralelas. Debido a que la mayoría de las técnicas disponibles en el procesamiento avanzado del análisis de este tipo de información no tratan simultáneamente la información espacial y la espectral, esta combinación de técnicas hace uso entero de la información suministrada por el píxel. Se analizó la interpretación de ese conjunto de algoritmos paralelos para el procesamiento espacio-espectral de imágenes hiperespectrales en el contexto de dos diferentes áreas de aplicación. Se proporciona una rápida referencia de las principales observaciones y lecciones aprendidas después de cada experimento. Adicionalmente la Universidad de Extremadura con los mismos equipos mencionados en el párrafo anterior, desarrolló el proyecto “Análisis Comparativo de Diferentes Implementaciones de un Algoritmo Paralelo para la Detección y Clasificación Automática de Objetos en Imágenes Hiperespectrales” sobre el

World Trade Center en New York después de los atentados terroristas del 11 de agosto de 2001. En este trabajo se implementó el algoritmo RX para la detección de anomalías y el algoritmo de clasificación y detección de objetivos automática (por sus siglas en ingles ATDCA), debido a que el proceso de detección de estos objetos en una escena hiperespectral tiene un alto costo computacional.

A nivel latinoamericano, en el año 2011 sobre el territorio de Cuba se investigó sobre el mapeo espectral de calizas en sectores de la Habana usando Imágenes Landsat TM, para lo cual se extrajo la firma espectral desde las imágenes de las materias primas minerales (generalmente calizas) del área de estudio [18]

A nivel nacional y como proyecto de grado en la Universidad Distrital Francisco José de Caldas para el año 2007, se desarrolló una base metodológica para la identificación de cultivos de coca en el departamento de Putumayo, para lo cual se implementaron las herramientas de pre-procesamiento, procesamiento y clasificación espectral del software Envi 4.3 para una imagen hiperespectral capturada por EO1 [3].

En la misma Universidad Distrital en 2008 se trabajó en un proyecto de grado con imágenes hiperespectrales, proponiendo e implementando una metodología para la generación de mosaicos de imágenes para el proyecto SIMCI II de Naciones Unidas; en este trabajo se emplearon técnicas de fusión de datos espaciales y se tuvo como insumos productos ópticos multitemporales, multiespectrales e hiperespectrales [30].

En Colombia igualmente se presentó en el año 2009 en el marco del VIII Congreso de la Asociación Colombiana de Técnicos de la Caña de Azúcar, el “Análisis Preliminar Para la Identificación de Variedades de Caña de Azúcar en Áreas Diferentes al Valle del Río del Cauca, usando imágenes Landsat 7 ETM”; en este trabajo se calculó el grado de separabilidad espectral de las variedades de caña con el algoritmo Spectral Angle Mapper (SAM) [26].

Como se observa son varios los trabajos que han tenido como insumos imágenes hiperespectrales, el común denominador de las investigaciones enunciadas en los párrafos anteriores son: la problemática en la administración de los datos, los extensos tiempos de procesamiento o de respuesta por las herramientas hardware y software al momento de su implementación. Su beneficio es el mejoramiento de la capacidad de síntesis y análisis debido a su mayor exactitud,

convirtiéndose en una herramienta eficaz para identificar objetos y coberturas sobre cualquier superficie.

5.7. SERVICIO GEOLÓGICO DE LOS ESTADOS UNIDOS (USGS)

La imagen hiperespectral que sirvió de base para el presente trabajo, fue obtenida a través de la página del Servicio Geológico de los Estados Unidos o USGS por sus siglas en inglés (United States Geological Survey)

USGS es una agencia científica de los Estados unidos que estudia el territorio, los recursos naturales y los peligros naturales que los amenazan. La agencia tiene cuatro disciplinas científicas: biología, geografía, geología e hidrología, su lema es Ciencia para el mundo cambiante. Es la única agencia científica del Departamento del Interior de los EEUU. Emplea aproximadamente a 10.000 personas y tiene su sede principal en Reston (Virginia); controla el Centro Nacional de Información Sísmica (National Earth quake Information Center) en Golden (Colorado), que se encarga de detectar la localización y magnitud de terremotos alrededor del mundo.

La USGS fue fundada el 3 de marzo de 1879. Le fue asignada la tarea de clasificación de las tierras públicas, el examen de su estructura geológica, recursos minerales y productos de dominio nacional [34].

5.8. IMAGEN HIPERESPECTRAL DE CALI

Las cabeceras municipales que alcanza a capturar la imagen son Jamundí, Cali, Yotoco y Vijes y la Ciudad de Cali con un mayor porcentaje de área respecto a la que tienen las otras cabeceras, representando así un beneficio para el proceso de clasificación que se pretende realizar.

La resolución espacial de la imagen es de 60x60 metros, capturando así un área de 3.600 metros cuadrados por cada pixel; la resolución radiométrica es de int16 o 2 a la 16 = 65.536 niveles digitales; la resolución espectral es de 242 bandas que se encuentran localizadas entre los 0,4 y 2,5 (um) micrómetros, es decir, entre el visible y el infrarrojo; el ancho de cada banda oscila entre los 10 y 12 (nm) nanómetros.

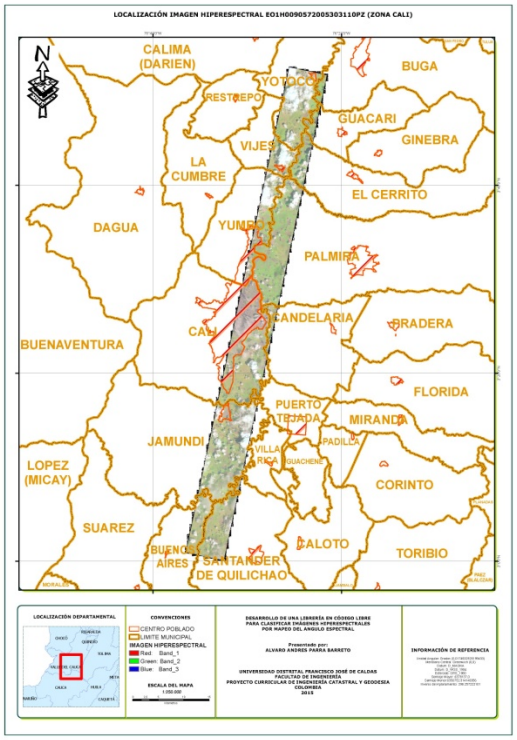

Imagen 03: Salida gráfica imagen hiperespectral Cali

5.9. CALI

Cali es la capital del departamento del Valle del Cauca y la tercera ciudad más poblada de Colombia, con una población de 2.04 millones de habitantes según el censo del DANE 2005. La ciudad forma parte del Área Metropolitana de Santiago de Cali, junto con los municipios aledaños, como Palmira, Yumbo, Jamundí, Florida, Pradera, Candelaria, Dagua y algunos municipios del Norte del Cauca como Puerto Tejada, Santander de Quilichao, Villa Rica, entre otros. Cali fue fundada el 25 de julio de 1536 por Sebastián de Belalcázar. Tiene una extensión de 564 km² y una longitud de 17 km de Sur a Norte y 12 km de Oriente a Occidente.

Cali es uno de los principales centros económicos e industriales de Colombia, además de ser el principal centro urbano, cultural, económico, industrial y agrario del suroccidente del país. Como capital departamental, alberga la Gobernación

del Valle del Cauca, la Asamblea Departamental, el Tribunal Departamental y otros.

El principal río de la ciudad y el departamento es el río Cauca, la parte correspondiente al municipio viene desde la desembocadura del río Jamundí hasta el límite entre Cali y Yumbo. El Cauca es de naturaleza sinuosa en su paso por el departamento del Valle, en 40 km navegados en el municipio se avanzan solamente 26 km en dirección sur-norte tras múltiples meandros.

La precipitación anual en promedio va desde los 900 mm en las zonas más secas hasta los 1.800 mm en las zonas más lluviosas, con 1.000 mm promedio sobre la mayor parte del área Metropolitana de Cali. La temperatura media es de 25 °C (74.4 °F) con un mínimo promedio de 15 °C (66 °F) y un máximo promedio de 32°C (86 °F).

6. METODOLOGÍA

Dada la temática planteada es obligatorio implementar una metodología para el desarrollo de software, se identificaron varios modelos, pero se determina por la estructura del algoritmo que la más adecuada para el presente trabajo es la RUP dado que permite integrar un conjunto de buenas prácticas y tiene un ciclo de vida que se basa en el desarrollo iterativo [22], que permite revisar con facilidad el avance y la planeación de posibles cambios de rumbo.

El ciclo de vida de un proyecto RUP contiene cuatro fases:

a. **Inicio**: Es la fase de la idea, de la visión inicial del producto y de su alcance; incluye el esbozo de una posible arquitectura y las primeras estimaciones. Concluye con el "hito del objetivo".

b. **Elaboración**: Comprende la planificación de las actividades y del equipo necesario, así como la especificación de las necesidades y el diseño de la arquitectura. Termina con el "hito de la arquitectura".

c. **Construcción**: Se refiere al desarrollo del producto hasta que se encuentra disponible para su entrega a los usuarios. Termina con el "hito del inicio de la capacidad operativa".

d. **Transición**: Comprende el traspaso del producto a los usuarios. Termina con el "hito entrega del producto".

6.1 Inicio

De acuerdo a la metodología la fase de inicio contempla el establecimiento de los objetivos y de los alcances del proyecto. Procesos que fueron suplidos con el anteproyecto, por tanto, se pasa a la siguiente fase.

6.2 Elaboración

Para desarrollar la librería que permita implementar el método de clasificación Mapeo del Ángulo Espectral a imágenes hiperespectrales en el software Grass, es necesario inicialmente identificar el tipo de lenguaje en que se desarrolló, cuales lenguajes soporta y el tipo de herramientas suministradas que permitan la elaboración del presente proyecto.

Al consultar la página web oficial de Grass [40], se encuentran varios elementos importantes, en la medida que se convierten en el soporte técnico para el desarrollo de la librería sobre la plataforma Grass, los cuales son:

- Documentación (Manuales, Libros, tutoriales y otros)
- Wiki
- Manual del programador y
- Ejemplo de modulo C

El último elemento incluye desde la versión 6 y está compuesto por dos módulos (r.example y v.example)

A lo largo de las lecturas tanto en su página web como de su documentación, se logra identificar que el lenguaje de programación en el que se desarrolló Grass Gis es C, igualmente se evidencia que existen módulos desarrollados principalmente en C pero también en Python y su interface gráfica se encuentra soportada en WxPython.

En forma general se observa que la mayoría de documentación está enfocada a los desarrollos en C, es decir, el lenguaje C cuenta con mayor soporte en Grass, lo que conlleva adoptar ese lenguaje de programación para el código objeto del presente trabajo.

Una de las herramientas mencionadas con anterioridad es el ejemplo de modulo C, el cual sirve como punto de partida para la elaboración de la librería, después de realizar la instalación de Grass 7 en el sistema operativo Ubuntu 14.04 LTS, alojado en un computador Lenovo Y400 con 6Gb de memoria RAM, un procesador Intel Core i7-3630QM 2.4Ghz y procesador gráfico Nvidia Geforce GT 650M.

Al establecer el lenguaje de programación (C) en el que se debe desarrollar la librería que permita la clasificación de imágenes hiperespectrales en el software libre Grass, el siguiente paso es identificar las bibliotecas que dan soporte en el código fuente, para este último es necesario analizar las propiedades y componentes requeridos en el algoritmo, obteniendo lo siguiente:

- El código debe hacer la lectura de los datos contenidos en el archivo (Imagen)
- Capturar los parámetros definidos por el usuario.
- Hacer el recorrido por todas los datos de la imagen (bandas, filas y columnas)
- Realizar las operaciones matemáticas (sumatoria, multiplicación, división, elevar potencias, calcular raíz cuadrada y usar funciones trigonométricas) sobre los datos contenidos en la imagen. De acuerdo a la ecuación del método SAM.
- Almacenar los resultados.
- Permitir el despliegue del resultado en el visor de Grass.

Después de identificar los componentes y propiedades relacionados anteriormente, se busca las bibliotecas del lenguaje que permitan cumplir con ellos, estás deben ser incorporadas dentro del código fuente, y hacen parte de las bibliotecas standar en el lenguaje C, son:

stdio.h: Esta biblioteca permite trabajar ficheros, igualmente las operaciones de entrada y salida.
stdlib.h: Esta biblioteca permite definir cuatro tipos de variables, varios macros y diversas funciones para procesos generales
string.h: Esta biblioteca permite la manipulación de memoria.
math.h: Esta biblioteca permite operaciones matemáticas y operaciones de compatibilidad

Las siguientes son bibliotecas standar de Grass:
gis.h: Esta biblioteca contiene la definición de variables y tipos de datos para el uso de la mayoría de sus funciones.
raster.h: Esta biblioteca contiene las estructuras de datos, algunos macros, tipos definidos y las enumeraciones que permiten el manejo de datos raster soportados por Grass.
glocale.h: Esta biblioteca incluye la dependencia gráfica para la visualización de los datos raster.

La forma de llamar las bibliotecas descritas anteriormente desde el código fuente,

es antecediendo a la biblioteca el siguiente término:

#include

6.3. Construcción.

Al determinar las bibliotecas y procedimientos que debe realizar el código, se continúa con el desarrollo. De acuerdo a lo establecido el primer paso es la lectura de la imagen, para lo que se tiene:

Es necesario después de la instalación y apertura de Grass, la creación de un directorio donde se almacenara todos los datos del software, posteriormente se crea una carpeta donde almacena el conjunto de mapas (Mapset) para el proyecto, en este caso la imagen hiperespectral.

La imagen se encuentra almacenada en un solo archivo, al cargarlo en Grass, se identificó que cada una de las bandas se almacena en el mapset como un archivo independiente y en un formato de Grass.

Para el siguiente paso que corresponde a la lectura de los datos raster, se identificó en la documentación las siguientes funciones:

In_fd = G_malloc(nfiles * sizeof()) Función que direcciona a los archivos de entrada establecidos en el mapset.

G_malloc (nfiles * sizeof (struct Cell_head)); Corresponde a la función que lee los metadatos de los archivos de entrada

G_malloc (nfiles * sizeof (RASTER_MAP_TYPE)) Función que identifica el tipo de datos en los archivos de entrada.

G_malloc (nfiles * sizeof (void *)) Función que determina el tamaño de la franja de lectura en las imágenes de entrada

Al identificar completamente las funciones que permiten la lectura de la imagen, el siguiente paso es determinar los parámetros necesarios en su procesamiento; los

parámetros que se definen son los siguientes:

**ptr = NULL Puntero Iterador de nombres declarado en Null
*mapset = NULL Puntero de la faja cargada del mapset declarado en Null
*in_fd = NULL Puntero para archivos de entrada declarado en Null
out_mask_fd = 0 Mascara declarada para salida
out_sam_fd = 0 Archivo declarado para salida
Cell_head *cellhd = NULL Puntero para los Metadatos de la imagen de entrada
**in_buffer = NULL Puntero para la memoria de entrada declarado en Null
*out_sam_buffer = NULL Puntero para la memoria del archivo de salida
*out_mask_buffer = NULL Puntero para la memoria de la máscara de salida
nfiles = 0 Se declara el número de bandas (archivos) de la imagen
Int nrows = 0 Se declara el número de filas de la imagen
Int ncols = 0; Se declara el número de columnas de la imagen

Adicionalmente es necesaria la definición de otros parámetros:

Int i = 0, j = 0, k = 0 Corresponden a los iteradores para bandas, filas y columnas
**input_raster_names = NULL
row_pixel = 0; Fila del pixel establecido por el usuario
column_pixel = 0 Columna del pixel establecido por el usuario
Angle = 0.0 Ángulo establecido por el usuario que determinar (s/n)
*cat_label = NULL Variable para
*mask_file = NULL Nombre de la máscara establecido por el usuario
*sam_file = NULL Nombre de la salida, establecido por el usuario
unknown_length Variable para almacenar el vector desconocido
dot_product = NULL Variable para almacenar el producto de niveles digitales

Definidos todos los parámetros, funciones y bibliotecas se determina los procedimientos que llevaran al ángulo espectral de cada uno de los píxeles.

El siguiente paso es el cargue de la imagen hiperespectral, pero, como se especificó con anterioridad, Grass transforma cada una de las bandas en un archivo, por tanto, el proceso se convierte en la lectura de dichos archivos, esto se realiza a través de un “for”, así:

```
For (i=0; i<nfiles; i++)
{
in_fd[i] = Rast_open_old (input_raster_names[i], "")
if ( in_fd[i] < 0 )
G_fatal_error (_ ("Unable to open raster map <%s>"), input_raster_names[i]);
data_type[i] = Rast_get_map_type (in_fd[i]);
Rast_get_cellhd (input_raster_names[i], "", &cellhd[i])
in_buffer[i] = Rast_allocate_buf (data_type[i])
}
```

La explicación del anterior código es:
El “for” recorre todos los archivos (nfiles) de entrada (input_raster_names[i]) y los almacenara en el puntero declarado (in_fn). Se determina un condicional (if) que evitara un error de software (bug) si se corre el proceso y no existe un archivo cargado. Seguidamente identifica los tipos de datos que se encuentran en los archivos, lo que permite declarar la memoria de lectura.

Para la salida del producto (cálculos) es necesaria la creación tanto de la máscara y del archivo con los ángulos espectrales. Los nombres son definidos por el usuario en los parámetros de entrada, se crean de la siguiente manera:

```
If ( mask_file != NULL) {
G_message (_ ("Making external mask image %s..."), mask_file)
out_mask_fd = Rast_open_c_new (mask_file); // Crea la imagen de salida
}
If ( sam_file != NULL) {
G_message(_("Making SAM vector image %s..."),sam_file);
Rast_set_fp_type(DCELL_TYPE);
out_sam_fd = Rast_open_fp_new(sam_file); // Crea la imagen de salida
}
```

Otros bug que se identificaron en el desarrollo del código y es necesario evitarlos, son:

```
if ( out_mask_fd < 0 )
G_fatal_error(_("Unable to create raster map <%s>"),mask_file);
```

```
if ( out_sam_fd < 0 )
G_fatal_error(_("Unable to create raster map <%s>"), sam_file);

if (mask_file != NULL && G_legal_filename(mask_file) < 0)
G_fatal_error(_("<%s> is an illegal mask file name"), mask_file);

if (sam_file != NULL && G_legal_filename(sam_file) < 0)
        G_fatal_error(_("<%s> is an illegal sam file name"), sam_file);
```

Durante el proceso se identifica que para el cargue del modulo en Grass se debe ejecutar la siguiente fracción de código:

```
G_gisinit(argv[0])
module = G_define_module()      Se define el modulo a utilizar
module->description = _("SAM module for raster processing")
```

Lo expuesto hace parte de los requerimientos no solamente del software Grass, sino de los parámetros de la librería para poder ejecutar el código de la ecuación de clasificación SAM, identificada en el capitulo 5.5.g. y resaltada a continuación:

$$\propto = cos^{-1}\left[\frac{\sum_{i=1}^{n} t_i\, r_i}{\sqrt{\sum_{i=1}^{n} t_i^2 \sum_{i=1}^{n} r_i^2}}\right]$$

Ecuación de Clasificación SAM [31]

La ecuación que identifica la clasificación SAM debe ser transformada al lenguaje de programación, con el objeto de ejecutar el código y evidenciar no solamente los resultados numéricos, sino gráficos.

El código inicialmente debe determinar una de las variables del producto cruz que conforma el denominador de la ecuación SAM; la cual consiste en calcular la raíz cuadrada de la sumatoria de los niveles digitales en cada una de las bandas elevadas al cuadrado, del espectro de referencia o conocido, esto se logra atreves de un FOR anidado donde la variable de cambio se dirige a través de todas las bandas de la imagen. Grass clasifica los datos almacenados en la imagen como

enteros, flotantes o doubles, por tanto, adicionalmente al “for” se debe generar un switch, el cual determinara el procedimiento a seguir de acuerdo al tipo de dato, así:

```
for (k = 0; k < nfiles; k++){

Switch (data_type[k]) {

Case CELL_TYPE:
reference_values[k] = G_malloc (sizeof (CELL));
*((CELL *) reference_values[k]) = ((CELL *) in_buffer[k]) [column_pixel];
reference_length += POW (*((CELL *) reference_values[k]), 2);
Break;
Case FCELL_TYPE:
reference_values[k] = G_malloc (sizeof (FCELL));
 *((FCELL *) reference_values[k]) = ((FCELL *) in_buffer[k]) [column_pixel];
 reference_length += POW (*((FCELL *) reference_values[k]), 2);
Break;
Case DCELL_TYPE:
reference_values[k] = G_malloc (sizeof (DCELL));
 *((DCELL *) reference_values[k]) = ((DCELL *) in_buffer[k]) [column_pixel];
 reference_length += POW (*((DCELL *) reference_values[k]), 2);
Break;
}}
```

Como se evidencia los tres casos son idénticos, a diferencia del tipo de dato que está procesando. Adicionalmente antes del FOR se creó la variable (reference_length) y la matriz (reference_values[]) con el objeto de almacenar la sumatoria que se efectúa a medida que se ejecuta el FOR y para almacenar la firma del pixel de referencia para su posterior procesamiento con las demás variables de la ecuación SAM.

Por último se describe la parte restante de la ecuación SAM, transformada al siguiente código:

```
for (i = 0; i < nrows; i++) {
for (j = 0; j < ncols; j++){
```

```
for(k = 0; k < nfiles ; k++){
if(j == 0 ){
Rast_get_row(in_fd[k], in_buffer[k], i, data_type[k]);
switch (data_type[k]) {
case CELL_TYPE:
unknown_length[j] += pow(((CELL *) in_buffer[k])[j],2);
dot_product[j] += *((CELL *)reference_values[k]) * (((CELL *) in_buffer[k])[j]);
break;
case FCELL_TYPE:
unknown_length[j] += pow(((FCELL *) in_buffer[k])[j],2);
dot_product[j] += *((FCELL *)reference_values[k]) * (((FCELL *) in_buffer[k])[j]);
break;
case DCELL_TYPE:
unknown_length[j] += pow(((DCELL *) in_buffer[k])[j],2);
dot_product[j] += *((DCELL *)reference_values[k]) * (((DCELL *) in_buffer[k])[j]);
break;
 }        }         }
for (j = 0; j < ncols; j++){
if( unknown_length[j] == 0 ){
out_sam_buffer = 0;
out_mask_buffer = 0;
}
else{
out_sam_buffer[j] = acos( dot_product[j] / ( reference_length * unknown_length)) *
180 / PI;
if(out_sam_buffer[j] < angle) {out_mask_buffer[j] = 1; }
else{   out_mask_buffer[j] = 0;
```

Al igual que en la parte anterior, se realiza la misma operación, pero esta vez en cada uno de los espectros de estudio de la imagen, esto se logra a través de un triple FOR anidado, que hacen el recorrido por las filas, columnas y bandas; paralelamente, se cálculo las otras variables que corresponden a la sumatoria de los niveles digitales de los espectros de estudio elevados al cuadrado, con lo que se determina el producto cruz entre cada uno de los anteriores y el mismo resultado del espectro de referencia; igualmente el proceso va calculando el producto punto entre el nivel digital en cada una de las bandas para los dos

espectros.

El código se optimizó en la tripleta anidada, realizando la mayor cantidad de cálculos dentro de ella, al igual que en la forma como hace la lectura de los píxeles, lo cual reduce la cantidad de procesos y el tiempo significativamente.

En este segmento también se observa que se crean las variables (unknown_length) y (dot_product) para ser utilizadas en el tercer FOR, sin embargo, debe inicializar en cero (0) para no tener ningún residuo del cálculo anterior.

Como se describía en los párrafos anteriores esta parte del código realiza todas las operaciones y hace almacenamiento de cálculos en las variables establecidas, para poder generar el almacenamiento en las matrices “out_sam_buffer” y ” out_mask_buffer” .

7. RESULTADOS

El primer resultado que se logra con el desarrollo de la librería r.sam es la interfaz gráfica (GUI) en el software Grass, la cual implementa el método de clasificación Mapeo del Ángulo Espectral (SAM) en imágenes hiperespectrales. Como se observa en la imagen 04, la interfaz está estructurada únicamente con los espacios para el ingreso de los parámetros por el usuario, los cuales son de carácter obligatorio para que el código se ejecute, la obligatoriedad se resalta con el asterisco naranja.

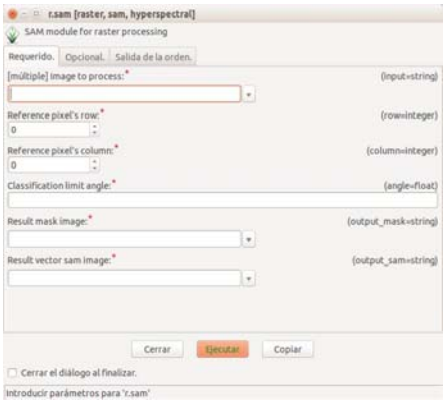

Imagen 04: Interfaz Gráfica r.sam

Los parámetros son: La selección de las bandas a las que se les desea ejecutar la clasificación SAM, estas bandas corresponden a las cargadas previamente en el Mapset al iniciar Grass

Imagen 05: GUI GRAS

La cantidad mínima para poder ejecutar el código corresponde a dos, en caso que se ingrese solamente una, se tendrá una salida con el siguiente error:

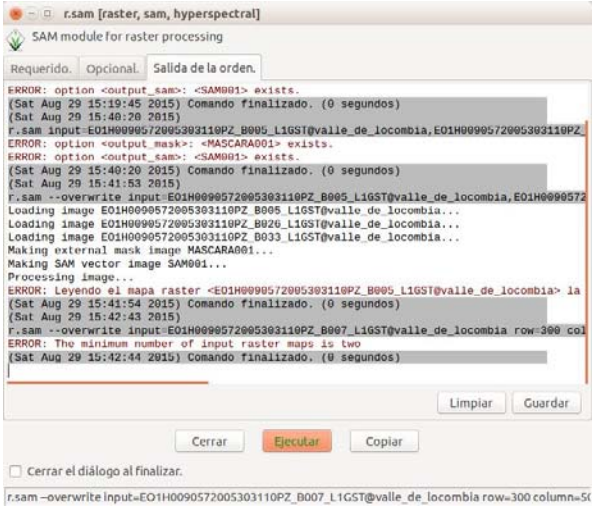

Imagen 06 Error al cargar una sola banda.

La selección de las bandas se podrá realizar a través de la lista desplegable que se muestra en la interfaz, en el momento de oprimir con el curso el botón que tiene una flecha hacia abajo y que se encuentra al lado derecho del cuadro de texto.

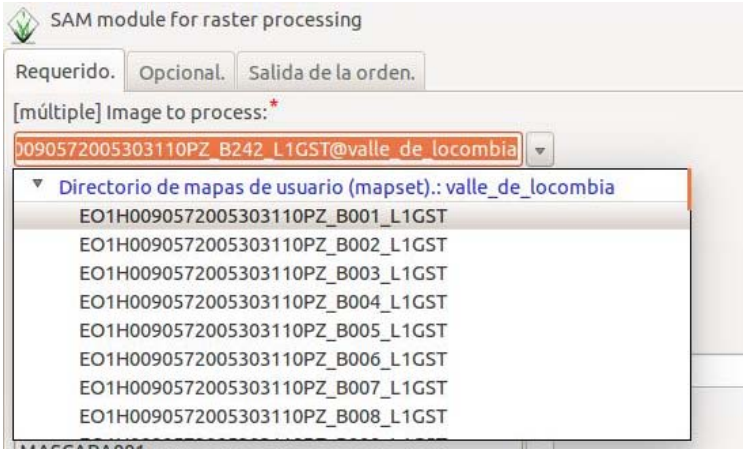

Imagen 07: Lista desplegable bandas

Si por algún motivo se ejecuta r.sam sin parámetros o si faltara uno, el resultado

sería el siguiente cuadro de dialogo:

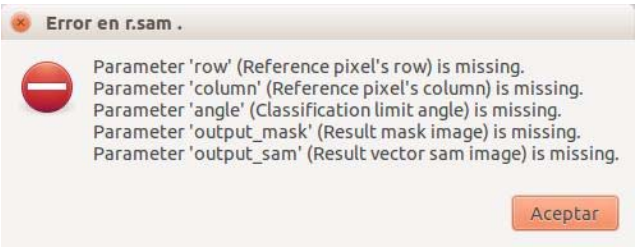

Imagen 08: Error de parámetros.

Los dos siguientes parámetros que deben ser definidos por el usuario, son la localización del píxel de referencia dentro de la imagen, es decir el X y Y, con estos datos se define el espectro de referencia con el que realizara la comparación con el resto de espectros de la imagen.

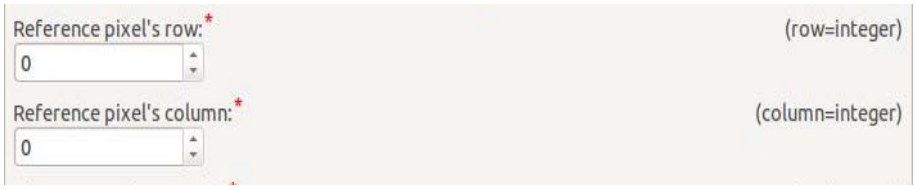

Imagen 09: Parámetros X y Y

El cuarto parámetro solicitado al usuario es muy importante, ya que con este dato se le indica al algoritmo que clasifique los píxeles más parecidos al de referencia y los que no; al procesar los datos que contiene la imagen con la ecuación SAM, se logra calcular los ángulos de las reflectancias aparentes, esta información sumada al conocimiento y experiencia del usuario, logran aproximar la clasificación por este método.

Este dato debe ser ingresado como un número entero, en ningún momento con simbología ya que el campo está definido para almacenar un dato flotante.

Imagen 10: Parámetro Ángulo

Los dos últimos parámetros requeridos por el algoritmo, son los nombres de las capas donde se almacenan los ángulos calculados por el código y la clasificación de los píxeles. En la segunda pestañas de la GUI “opcional” se puede habilitar sobrescribir las capas o archivos existentes, es decir editarlos con los nuevos resultados; esta opción es con el fin de que en el momento que el usuario este haciendo pruebas, no tenga la necesidad de crear una cantidad innecesaria de capas o archivos, sino que tenga la oportunidad de tener eficiencia en el proceso.

Imagen 11: Nombre de las salidas graficas.

Imagen 12: Pestaña opcional

Para probar la librería r.sam se tomo la imagen hiperespectral EO1H0090572005303110PZ.L1R, que fue descargada de la página del Servicio Geológico de los Estados Unidos (www.usgs.gov), con los siguientes parámetros:

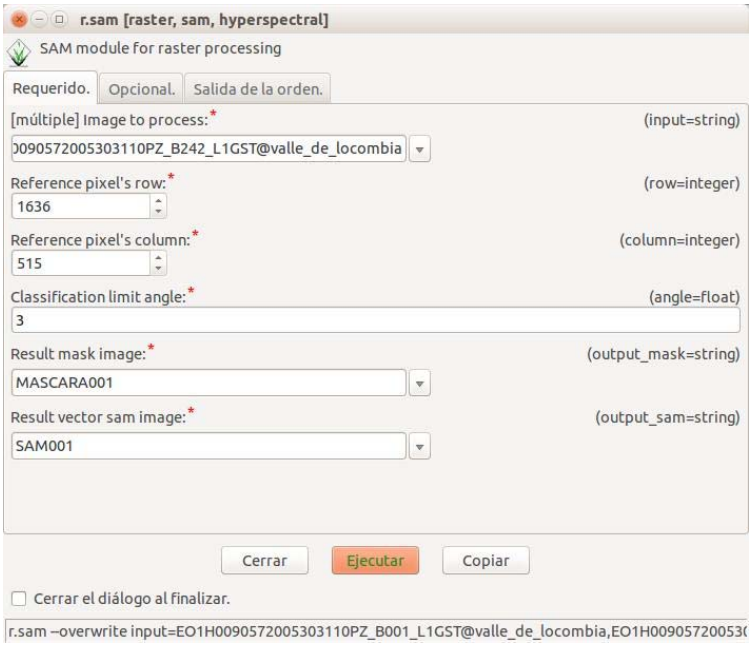

Imagen 13: Parámetros de prueba.

Como se observa en la siguiente imagen mientras que el código se está ejecutando el usuario puede visualizar el porcentaje de avance.

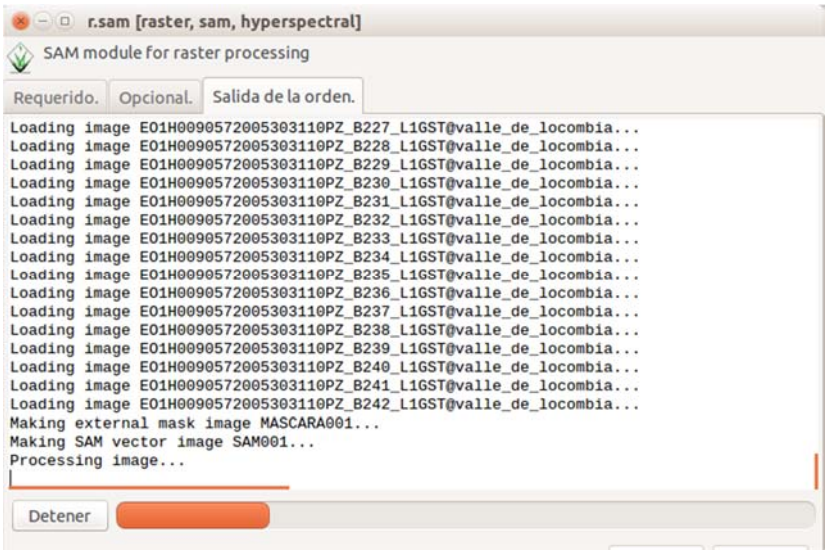

Imagen 14: Porcentaje de avance.

Al finalizar el proceso, la GUI le muestra al usuario el tiempo que se demoro en realizar toda la tarea.

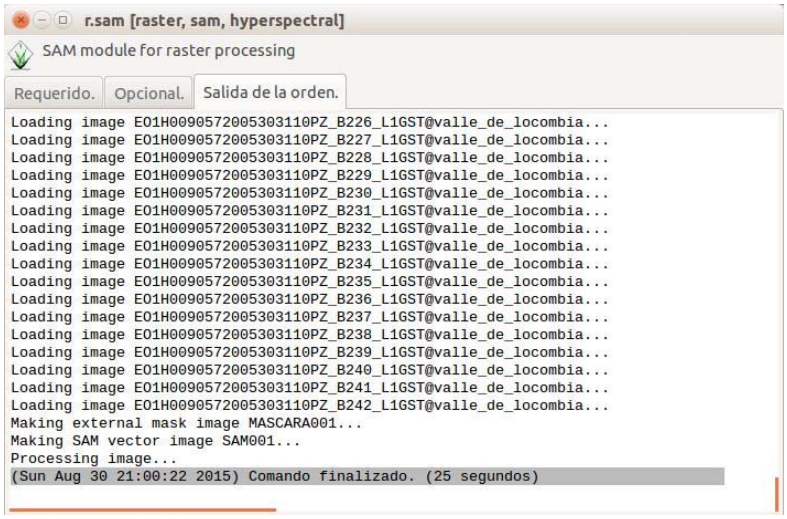

Imagen 15: Comando de finalización y tiempo.

Como resultado final de ejecutar el código elaborado a partir de la ecuación SAM, con unos parámetros establecidos, sobre la imagen hiperespectral, se tiene las dos salidas gráficas: SAM001 y MASCARA001. Las cuales se observan a continuación:

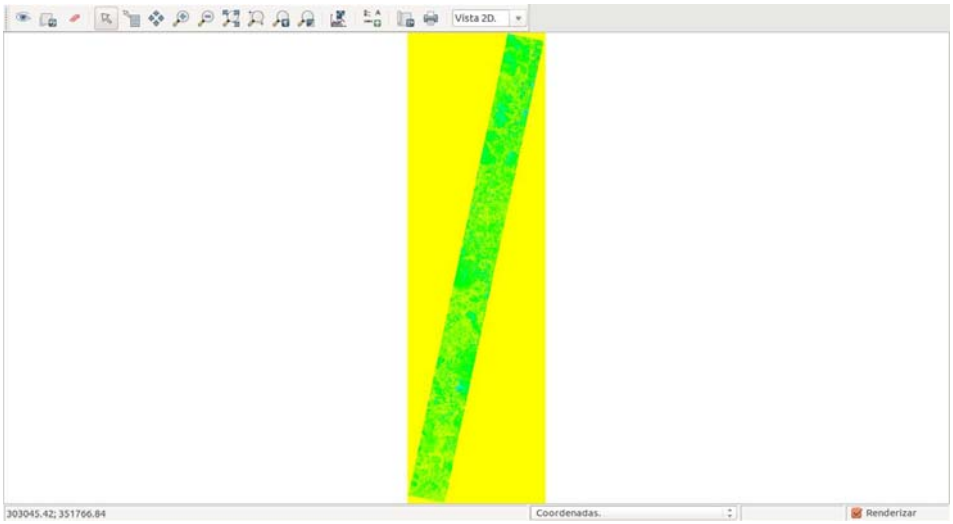

Imagen 16: Salida SAM

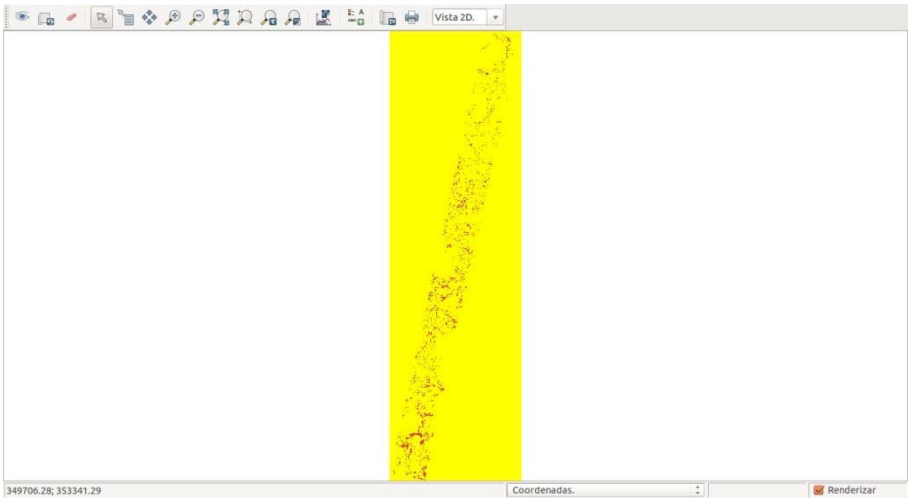

Imagen 17: Salida Mascara.

8. CONCLUSIONES

- Se identificó que la implementación de nuevos módulos y herramientas en el software Grass, deben ser desarrollar en el lenguaje de programación C, ya que cuenta con la mayor documentación, lo que permite un mejor conocimiento de las bondades de Grass.
- Se logró transformar la ecuación de la clasificación Mapeo del Ángulo Espectral (SAM) en un algoritmo mediante el lenguaje C, alojado en la librería r.sam en el software Grass, materializada en la interfaz gráfica de usuario (GUI) r.sam.
- La librería r.sam realiza los cálculos a partir de los parámetros establecidos por el usuario, es decir, que el material de interés o firma de referencia se define de la escena o bandas cargadas por el usuario.
- Se optimizó el tiempo de respuesta de la librería, con la ayuda de los formatos de lectura de imágenes de Grass; inicialmente con GDAL se tenía tiempos de 21 horas aproximadamente, debido a que la lectura de la imagen se realizaba píxel a píxel, en Grass se logro reducir el tiempo a 25 segundos aproximadamente, debido a la lectura de las imágenes fila a fila.

9. BIBLIOGRAFIA

[1] AGARWAL, Abhishek, TEHRANIAN, Shahram y SWARROP, Anand. A robust Framework for distributed processing of GIFTS hyperspectral data. [En línea][Citado el 07 de agosto de 2015]. Disponible en internet: http://ams.confex.com/ams/.

[2] ALONSO, José M. TCP/IP en Unix: programación de aplicaciones distribuidas. Alfaomega, 1999. 194 p.

[3] ANGEL LOPEZ, Yoseline Betzaida. Diseño y evaluación de una metodología para la identificación de cultivos de coca mediante imágenes hiperespectrales. Trabajo de grado para optar el título de Ingeniero Catastral y Geodesia. Bogotá D.C.: Universidad Distrital Francisco José de Caldas. Facultad de Ingeniería. Proyecto curricular de Ingeniería Catastral y Geodesia, 2007. 205 p.

[4] Aviris. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://aviris.jpl.nasa.gov

[5] BORENGASSER, Marcus, HUNGATE, William. Hyperspectral remote sensing Principles and applications. Indiana: Indiana State University. 2008. 130 p.

[6] CASI. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.borstad.com/casi.html

[7] CNICE. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://concurso.cnice.mec.es/cnice2006/material121/unidad3/sat_td.htm

[8] CUITIVA, Ricardo y MESA, Samuel. Clasificación de la cobertura y del uso del suelo urbano usando algoritmos de redes inteligencia artificial. Bogotá D.C.: Universidad Distrital Francisco José de Caldas. Facultad de Ingeniería. Proyecto curricular de Ingeniería Catastral y Geodesia, 2006. 161 p.

[9] DEITEL, Harvey M. y DEITEL Paul J. Como programar en C++ 4ta Edición. México: Prentice Hall, 2003. 1376 p.

[10] Earth Observing-1. [En línea] [Citado el 07 de agosto de 2015]. Disponible en

internet: http://eo1.gsfc.nasa.gov.

[11] ERDAS. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.intergraph.com/

[12] EXELISVIS [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet:http://www.exelisvis.com/

[13] UNIVERSIDAD DE EXTREMADURA. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.unex.es/

[14] GDAL. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.gdal.org/

[15] GENERAL PUBLIC LICENSE (GPL). [En línea][Citado el 07 de agosto de 2015]. Disponible en internet: http://www.gnu.org/copyleft/gpl.html

[16] GIFTS. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://asd- www.larc.nasa.gov/GIFTS/

[17] GODDAN SPACE FLIGHT CENTER. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.nasa.gov/centers/goddard/home/index.html

[18] GONZALEZ A., Virginia y MOREIRA M., Jesús. Mapeo espectral de calizas en sectores de la Habana usando imágenes Landsat TM. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http//www.iga.cu.

[19] GOUGH, Brian y STALLMAN Richard M. An introduction to GCC for the GNU Compiler gcc and g++. Reino Unido: Network Theory Limited, 2004. 120 p.

[20] HESHAM, El-Rewini y MOSTAFA, Abd-El-Barr. Avanced computer architectureand parallel processing. New Jersey: Wiley Series, 2005. 287 p.

[21] IBARRA P., Francisco. Introducción a las arquitecturas paralelas. Alicante: Universidad de Alicante. 1995. 262 p.

[22] JACOBSON, Ivar y BOOCH, Grady. El proceso Unificado del desarrollo de Software. Madrid: Pearson Education, 2000. 438 p.

[23] LEE, R. C., TSENG, S. S., CHANG R. C., y TSAI Y. T. Introduction to the

Design and analysis of algorithms a strategic approach. Londres: McGraw-Hill, 2005. 745 p.

[24] LEICA, Geosystem. Erdas Field Guide. Estados Unidos: Erdas, 2002.644 p.

[25] LIZARAZO Iván. Clasificación de la cobertura y del uso del suelo urbano usando imágenes de satélite y algoritmos supervisados de inteligencia artificial. En: Revista UD y la GEOMATICA. Segundo semestre del 2008, vol. 1, no. II, p. 04-18.

[26] MURILLO S., Paulo José y CARBONELL G., Javier Ali. Análisis preliminar para la identificación de variedades de caña de azúcar en áreas diferentes al valle del río del Cauca, usando imágenes Landsat 7 ETM. En: Congreso de la Asociación Colombiana de técnicos de la Caña de Azúcar. (8: 16-18, agosto: Cali). Memorias. Cali: luéd, 2009 p. 32-38.

[27] NOAA. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.noaa.gov

[28] ORION. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.oriondata-i.com/home/index.php/analisis

[29] RICHARDS, John A., JIA, Xiuping. Remote Sensing Digital Image Analysis. Springer- Verlag, Berlin, Germany. 1999. p. 449.

[30] RUEDA, David Alonso. Generación de mosaicos de imágenes para el proyecto SIMCI II de las Naciones Unidas empleando técnicas de fisión de datos espaciales y utilizando como insumos productos ópticos multiespectrales, multitemporales e hiperespectrales. Trabajo de grado para optar el título de Ingeniero Catastral y Geodesia. Bogotá D.C.: Universidad Distrital Francisco José de Caldas. Facultad de Ingeniería. Proyecto curricular de Ingeniería Catastral y Geodesia, 2008. 175 p.

[31] SCIELO. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.scielo.org.ar/scielo.php?pid=S0004-48222010000300021&script=sci_arttext

[32] SMITH, Randall B. Introduction to Hyperspectral Imaging. [En línea] [citado el 07 de agosto de 2015]. Disponible en internet:

http://www.microimages.com/documentation/Tutorials

[33] TACKETT JR, Jack y GUNTER, David. Edición Especial Linux 3ra Edición. Madrid: Prentice Hall, 1998. 806 p.

[34] USGS. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://www.usgs.gov/

[35] WALL, Kurt. Programación en Linux con ejemplos. Buenos Aires: Prentice Hall, 2000. 568 p.

[37] GONZÁLEZ C., Carlos. Procesamiento a bordo de imágenes hiperespectrales de la superficie terrestre mediante hardware reconfigurable. Memoria para optar al grado de doctor. Madrid, España. Universidad Complutense de Madrid. Facultad de Informática. 2012. 219 p.

[38] Ingeniería Catastral y Geodesia. Universidad Distrital Francisco Jose de Caldas. [En línea] [Citado el 07 de agosto de 2015]. http://ingenieria1.udistrital.edu.co/udin/mod/page/view.php?id=104

[40] GRASS. [En línea] [Citado el 07 de agosto de 2015]. Disponible en internet: http://grass.osgeo.org/

10. ANEXO

BIBLIOTECA GDAL

GDAL es una librería de conversión de formatos para datos raster, tiene licencia de código abierto para la Fundación Geoespacial de Código Abierto (Open Source Geospatial Foundation), como una librería; se presenta como la única aplicación para el llamado de modelos de datos abstractos para todos los formatos soportados. También viene con una variedad de líneas de comandos utilizadas para la conversión y procesamiento de datos. GDAL soporta ciento treinta y tres (133) formatos de archivos, dentro de los que se encuentran BMP, ECW, ENVI, HDR, N1, TIFF, DT1, IMG, MPR, entre otros, permitiendo de este modo acceder a la información que reposa en cada uno de los archivos dependiendo de las necesidades del usuario [14].

GDAL es compatible con el formato de producto Envisat solo para acceso de lectura. Se admiten todos los tipos de muestras. Actualmente, todos Nivel ASAR 1 y los productos anteriores, y algunos productos de MERIS y AATSR son funciones que permiten consultar la información de los archivos copiar y pegar archivos compatibles. GDAL es distribuido juntamente con una serie de funciones para añadir descripciones a los archivos, georeferenciar imágenes, contornos DEM, construir mosaicos, transformar sistemas de coordenadas, interpolar regiones de datos y otras más [14].

Son numerosas las aplicaciones que usan GDAL para el acceso a datos geográficos entre las que se destacan: MapServer, GRASS GIS, Quantum GIS, Google Earth y Kosmos.

Para realizar la apertura, lectura y extracción de la información contenida en la imagen, se describe en el tutorial de GDAL las siguientes recomendaciones e indicaciones para que el proceso tenga un resultado favorable:

Apertura de archivo: Antes de abrir el archivo donde se encuentran los datos raster, por medio de GDAL, es necesario registrar los drivers. Existe un driver para cada formato compatible; normalmente esto se logra con la función GDALAllRegister () que registra todos los controladores conocidos, una vez que los controladores están registrados, la aplicación llama la función GDALOpen ()

para abrir el archivo [14]. El segmento del código donde se realiza la tarea anterior es el siguiente:

```
#include "gdal_priv.h"
#include "cpl_conv.h" // for CPLMalloc()
intmain () {
GDALDataset  *poDataset;
GDALAllRegister();
poDataset = (GDALDataset *) GDALOpen( pszFilename, GA_ReadOnly );
if( poDataset == NULL )    {
...;                       }
```

Hay que tener en cuenta que si la función GDALOpen () retorna NULL significa que la apertura del archivo no fue el indicado, un mensaje de error se emitirá a través de la función CPLError ()[14]. En general todas las librerías de GDAL utilizan CPLError () para la presentación del informe de error. Además, se debe tener en cuenta que pszFilename podría ser una URL, un nombre de archivo con parámetros adicionales añadidos al final el control.

Acceso a los datos del archivo: Como se describe en el modelo de datos de GDAL, un GDALDataset contiene la lista de bandas del archivo [14], todos pertenecientes a la misma zona, y que tiene la misma resolución. También dispone de metadatos, un sistema de coordenadas, una georeferenciación, transformación de tamaño de archivo y otro tipo de información.

```
adfGeoTransform[0] /* top left x */
adfGeoTransform[1] /* w-e pixel resolution */
adfGeoTransform[2] /* 0 */
adfGeoTransform[3] /* top left y */
adfGeoTransform[4] /* 0 */
adfGeoTransform[5] /* n-s pixel resolution (negative value) */
```

En este caso, se tiene una transformada afín. Para la impresión de la información general sobre el conjunto de datos se debe tener en cuenta lo siguiente:

```
doubleadfGeoTransform[6];

printf( "Driver: %s/%s\n",poDataset->GetDriver()->GetDescription(), poDataset-
>GetDriver()->GetMetadataItem( GDAL_DMD_LONGNAME ) );

printf( "Size is %dx%dx%d\n", poDataset->GetRasterXSize(),poDataset-
>GetRasterYSize(), poDataset-   >GetRasterCount() );

if( poDataset->GetProjectionRef()  != NULL )

        printf( "Projection is `%s'\n", poDataset->GetProjectionRef() );

if(poDataset->GetGeoTransform( adfGeoTransform ) == CE_None )

        {

        printf( "Origin = (%.6f,%.6f)\n", adfGeoTransform[0], adfGeoTransform[3] );

        printf( "Pixel Size = (%.6f,%.6f)\n", adfGeoTransform[1],
adfGeoTransform[5] );

        }
```

Del segmento anterior se obtiene la salida de la información básica de la imagen en la terminal o en un archivo si se desea, es decir, el número de bandas, de filas y de columnas que componen, igualmente la proyección.

Lectura de bandas: El acceso a la trama de datos por medio de GDAL, se realiza de una banda a la vez; además, el tamaño del bloque corresponde a la cantidad de columnas de la imagen, los niveles digitales y otro tipo de información disponible también se realiza banda por banda y tramo por tramo [14]. La siguiente función GDALRasterBand da acceso al conjunto de bandas (numeradas desde 1 hasta GetRasterCount (), este ultimo indica el total de bandas presentes en la imagen). El segmento del código es el siguiente:

```
GDALRasterBand  *poBand;
```

```
intnBlockXSize, nBlockYSize;

intbGotMin, bGotMax;

doubleadfMinMax[2];

poBand = poDataset->GetRasterBand( 1 );

poBand->GetBlockSize( &nBlockXSize, &nBlockYSize );

printf( "Block=%dx%d Type=%s, ColorInterp=%s\n", nBlockXSize, nBlockYSize,

        GDALGetDataTypeName(poBand->GetRasterDataType()),
        GDALGetColorInterpretationName(poBand->GetColorInterpretation()) );

adfMinMax[0] = poBand->GetMinimum( &bGotMin );

adfMinMax[1] = poBand->GetMaximum( &bGotMax );

if( ! (bGotMin&&bGotMax) )

        GDALComputeRasterMinMax((GDALRasterBandH)poBand, TRUE,
adfMinMax);

printf( "Min=%.3fd, Max=%.3f\n", adfMinMax[0], adfMinMax[1] );

if(poBand->GetOverviewCount() > 0 )

        printf( "Band has %d overviews.\n", poBand->GetOverviewCount() );

if( poBand->GetColorTable() != NULL )

        printf( "Band has a color table with %d entries.\n", poBand-
>GetColorTable()-   >GetColorEntryCount() );
```

Adicionalmente el fragmento anterior permite tener la información del nivel digital máximo y mínimo, por último el tipo de dato que representa cada nivel digital.

Lectura de la trama de datos: Con GDAL existen varias maneras de leer los datos de los diferentes formatos de imágenes, pero la más común es a través de la función GDALRasterBand::RasterIO () . Este método se encargará automáticamente de la conversión de tipos de datos [14]. El siguiente código leerá la primera línea de los datos en un búfer de tamaño similar, la conversión a punto flotante hace parte de la operación.

```
float *pafScanline;
intnXSize = poBand->GetXSize ();
pafScanline = ( float *) CPLMalloc ( sizeof ( float ) * nXSize);
poBand->RasterIO ( GF_Read , 0, 0, nXSize, 1, pafScanline, nXSize, 1,
GDT_Float32, 0, 0);
```