

**Desarrollo de un Sistema Middleware para Sensores Remotos con Fuente de Datos
Distribuida**

Por

Alcides Alvear Suárez

Tesis sometida en cumplimiento parcial
De los requisitos para el grado de

**MAESTRÍA EN CIENCIAS EN
INGENIERÍA DE COMPUTADORAS**

**UNIVERSIDAD DE PUERTO RICO
MAYAGUEZ CAMPUS, 2003**

Aprobada por:


Manuel Rodríguez, Ph.D.
Presidente, Comité Graduado

12/5/2003
Fecha


Isidoro Couvertier, Ph.D.
Miembro, Comité Graduado

8 dic 2003
Fecha


Wilson Rivera, Ph.D.
Miembro, Comité Graduado

12/5/2003
Fecha


Pedro I. Rivera, Ph.D.
Miembro, Comité Graduado

12/8/2003
Fecha


José Luis Cruz Rivera, Ph.D.
Chairperson of the Department

12/5/03
Fecha


Dorothy Boffman, Ph.D.
Representante Estudios Graduados

12/5/03
Fecha

ABSTRACT

This thesis describes the tool ImageCutter, which does part of the megaproject TerraScope. The effected design uses a system middleware for, across the Internet, to connect a set of computers creating a virtual computer of great coverage, capable to handle information related to hyperspectral images. The provided information, as well as the resources of calculation, they can be located physically in diverse sites. These resources include components both of hardware and of software (ej. Matlab, Java and final user of applications). The system ImageCutter allows the processing image of distributed form and realizes the division of images of agreement to the specifications of the user. Tried the image, this ImageCutter in capacity of: to recover the information, to gather the new byline and to offer new forms of interpretation to the client. The architecture of the component ImageCutter and the algorithms used for the design of the tool will be discussed in the theoretical and methodological foundation of the thesis.

RESUMEN

Esta tesis describe la herramienta ImageCutter, que hace parte del megaproyecto TerraScope. El diseño efectuado es un sistema middleware que utiliza el Internet, para conectar un conjunto de computadoras creando una computadora virtual de gran cobertura, capaz manejar información relacionada con imágenes hiperspectrales. La información provista, así como los recursos de cómputo, pueden estar localizados físicamente en diversos sitios. Estos recursos incluyen componentes tanto de hardware como de software (ej. Matlab, Java y aplicaciones de usuario final). El sistema ImageCutter permite el procesamiento de imágenes de forma distribuida y realiza la división de imágenes de acuerdo a las especificaciones del usuario. Procesada la imagen, ImageCutter esta en capacidad de: recuperar la información, recolectar la nueva data y ofrecer al cliente nuevas formas de interpretación. La arquitectura del componente ImageCutter y los algoritmos usados para el diseño de la herramienta se discutirán en la fundamentación teórica y metodológica de la tesis.

CONSENTIMIENTO

Por este medio autorizo a la Biblioteca del Recinto Universitario de Mayagüez a permitir copiar en parte o en su totalidad el documento de trabajo realizado para la obtención de mi grado. La copia parcial o total del mismo será únicamente para propósitos de investigación.

CONSENT

I do hereby authorize the Library of University of Puerto Rico at Mayagüez to allow partial or complete copying of this document for research purposes.

© Copyright by

Alcides Alvear-Suárez

2003

DEDICATORIA

A mis queridos hijos Felipe, Leonardo y Alejandro, para recordarles que la perseverancia es el condimento que ajusta el espíritu y aviva los sueños.

A mi gran amor Elizabeth por su apoyo y su constante voz de aliento.

A mi padre Víctor, a mi madre Rita y a mis hermanos por sus oraciones que siempre me fortificaron espiritualmente.

A mis amigos por las ideas que de algún modo están reflejadas en la realización de este trabajo.

RECONOCIMIENTOS

Al Dr. Manuel Rodríguez-Martínez por sus invaluable consejos durante la preparación y ejecución del presente trabajo.

A los miembros del comité graduado Doctores: Pedro I. Rivera-Vega, Wilson Rivera e Isidoro Couvertier por sus sugerencias y correcciones pertinentes para la preparación del presente manuscrito.

A los estudiantes Angel Villalaín y Jaime Ferrá por su ayuda en la elaboración del código implementado para la creación de la herramienta ImageCutter.

A los Doctores Luis Jiménez y Miguel Vélez de CenSSIS y LARSIP, por el aporte económico que permitió la realización de mis estudios.

Al personal de LARSIP, especialmente a Emmanuel Arzuaga y Marie Ayala por el soporte computacional ofrecido durante todo el desarrollo de este trabajo.

A los recursos ofrecidos por *Engineering Research Centers Program of the National Science Foundation* bajo la concesión EEC-9986821.

CONTENIDO

LISTA DE TABLAS	X
LISTA DE FIGURAS	XI
CAPITULO 1 INTRODUCTION.....	1
CAPITULO 2 PREVIOUS WORK.....	4
2.1 SERVICE PROCESSING SYSTEM USING DISTRIBUTED DATABASE.....	5
2.2 WORLD METEOROLOGICAL ORGANIZATION DISTRIBUTED DATABASES (WMO- DDBS)	7
2.3 SPATIAL INDICES	7
2.4 AN INTERNET PROTOCOL FOR VIRTUAL ENVIRONMENTS - DWTP	8
2.5 TERRASERVER	8
2.6 MIDDLEWARE BASED ON A CODE SHIPPING ARCHITECTURE - MOCHA	9
2.7 HERRAMIENTA - TOOLBOX TOOL DESARRALLADA EN MATLAB.....	10
2.7.1 Sensores Remotos	11
2.7.2 Reconocimiento de Patrones	11
2.7.3 Entrenamiento (Training).....	11
2.7.4 Firmas (Signatures).....	12
2.7.5 Imágenes Hiperespectrales.....	12
2.7.6 Clasificación.....	13
2.7.7 Reglas de clasificación Paramétricas	13
2.7.8 Ecuaciones Usadas	15

2.8 VENTANA PRINCIPAL DEL TOOLBOX	16
CAPITULO 3 ARQUITECTURA DE IMAGECUTTER	20
3.1 ARQUITECTURA DE IMAGECUTTER	20
3.2 MODELO GENERAL DEL SISTEMA	21
3.3 ALCANCE DE IMAGECUTTER.....	24
3.4 PROCESOS QUE EJECUTA IMAGECUTTER	24
3.5 SOFTWARE QUE COMPONE IMAGECUTTER	25
3.6 IMAGECUTTER SCREEN	27
3.6.1 Ventana Principal.....	27
3.6.2 Pantalla para cargar la imagen.....	29
3.6.3 Procesos Iniciales	29
3.6.4 Pantalla de ImageCutter con imagen cargada	31
3.6.5 Pantalla para ejecutar proceso seleccionado	32
3.6.6 Pantalla con resultado final.....	33
CAPITULO 4 MATERIALES Y METODOS.....	34
4.1 DISEÑO EXPERIMENTAL.....	34
4.2 METODOLOGÍA.....	36
CAPITULO 5 DISCUSIÓN Y RESULTADOS	39
5.1 RESULTADOS.....	39
5.1.1 RESULTADOS CON LA IMÁGEN DE INDIANA	40
5.1.2 RESULTADOS CON LA IMÁGEN ANASCO	43
5.1.3 RESULTADOS CON LA IMÁGEN KENNEDY.....	46

CAPITULO 6 CONCLUSIONES Y TRABAJO FUTURO.....	48
6.1 CONCLUSIONES	48
6.2 TRABAJO FUTURO	50
REFERENCIAS	52
APÉNDICE A. IMÁGENES USADAS EN LAS PRUEBAS	54
APÉNDICE B. TIEMPOS DE PROCESOS DE IMÁGENES USADAS EN LAS PRUEBAS	57

LISTA DE TABLAS

Tabla 1: Características de las imágenes usadas en las pruebas	39
Tabla 2. Tiempo de procesamiento de Indiana con 1, 2, 4, 6 y 8 nodos	41
Tabla 3. Tiempo de procesamiento de Anasco con 1, 2, 4, 6 y 8 nodos	43
Tabla 4. Tiempos de proceso de Imágen Kennedy con 1, 2, 4, 6 y 8 nodos.....	46

LISTA DE FIGURAS

Figura 1: Arquitectura de Mocha	10
Figura 2: Ventana Principal	17
Figura 3: Mapa de Clasificación.....	18
Figura 4: Flujo de Clasificación.....	19
Figura 5: Esquema general del Sistema	20
Figura 6: Modelo general del Sistema	21
Figura 7: Procesamiento de una imagen con ImageCutter	23
Figura 8: Estructura del Software de ImageCutter.....	25
Figura 9: Ventana Principal de ImageCutter.....	27
Figura 10: Pantalla para comenzar con ImageCutter.....	29
Figura 11: Comando para abrir un nodo	30
Figura 12: Vista de una imagen cargada con ImageCutter	31
Figura 13: Ejecuta proceso de clasificación seleccionado	32
Figura 14: Mapa de Clasificación, resultado final.....	33
Figura 15: Esquema del diseño experimental	35
Figura 16. Comportamiento del proceso de Indiana con 1, 2, 4, 6 y 8 nodos.....	42
Figura 17: Detalle del comportamiento de Indiana con 2, 4, 6 y 8 nodos	43
Figura 18: Comportamiento del proceso de Anasco con 1, 2, 4, 6 y 8 nodos	44
Figura 19: Detalle del comportamiento de Anasco con 2, 4, 6 y 8 nodos	45
Figura 20: Comportamiento del proceso de Kennedy con 1, 2, 4, 6 y 8 nodos	47

Figura 21. Detalle del comportamiento de Kennedy con 2, 4, 6 y 8 nodos 48

CAPITULO 1 INTRODUCCIÓN

Uno de los mayores obstáculos en el procesamiento de imágenes hiperespectrales obtenidas con la utilización de sensores remotos, es la carencia de herramientas eficientes y fáciles de utilizar para procesar grandes cantidades de data. Procesar información de manera local requiere de un sistema de comunicaciones rápido y en ocasiones a pesar de disponer de estas herramientas no siempre es factible realizar el procedimiento, pues manipular gran cantidad de información implica aumentar el tiempo de ejecución y por tanto elevar el costo computacional. Afortunadamente, la tecnología de hoy ofrece la posibilidad de crear sistemas capacitados para hacer frente a estos inconvenientes, aprovechando los recursos de cómputo que pueden existir en universidades y centros de investigación. Por otra parte, la existencia bases de datos con imágenes sin procesar y dispersas alrededor del mundo ha sido objeto de preocupación por parte de los investigadores que trabajan en esta área.

Para tratar algunos de estos inconvenientes, el *Advance Data Management Group* (ADMG) en la Universidad de Puerto Rico en Mayagüez está desarrollando el sistema TerraScope que utiliza la infraestructura del Internet para crear un sistema distribuido capaz de procesar imágenes de manera rentable. El sistema de TerraScope se basa en componentes middleware que acerca en forma virtual gran cantidad de computadoras localizadas física y geográficamente en distintas partes del mundo para crear una computadora virtual de gran alcance; capacitada para localizar datos, procesarlos y

ofrecerle a los usuarios una amplia gama de posibilidades que le facilitan realizar clasificaciones supervisadas a partir de una data preexistente.

El middleware es una infraestructura capacitada para correlacionar en forma eficiente herramientas software y hardware, mostrando a los usuarios un sistema homogéneo. Además proporciona un método de comunicación cliente - servidor, o entre servidores distantes. También puede ofrecer servicios de administración, seguridad y manejo de fallas de memoria. Dependiendo del producto, puede proporcionar servicios de aplicaciones específicas tales como acceso a bases de datos, tratamiento transaccional o réplica de los datos [1]. El acceso y la integración de los datos es una de las categorías más importantes y representa 60 a 70 % del mercado middleware. Esta herramienta fue desarrollada para proporcionar acceso universal a las bases de datos de imágenes, de datos relacionales, jerárquicos y archivos planos. Este tipo de bases de datos middleware se limita a pasar información entre una base de datos y un usuario.

Uno de los componentes principales de TerraScope es el ImageCutter, que tiene entre sus funciones realizar la partición de imágenes de la manera requerida por el usuario, distribuye la data en los diversos nodos disponibles, procesa la información, recolecta la nueva data, envía los resultados parciales producto de las operaciones distribuidas y ofrece al usuario en pocos minutos un nuevo producto. ImageCutter fue desarrollado en Java y utiliza MATLAB para realizar diferentes procesos de clasificación supervisada. ImageCutter también permite a los usuarios crear y someter código a través del Web, invocando bases de datos con imágenes disponibles. Una vez efectuado el

procedimiento el cliente recibirá un URL que le notificará el lugar donde se localiza el resultado. En la práctica, la herramienta ImageCutter estará disponible para todos los usuarios que requieran del procesamiento y del análisis de imágenes multiespectrales. También proporcionará una manera conveniente de compartir datos y resultados a nivel interdisciplinario y en futuro se proyecta como una herramienta útil en medicina, geología, biología y química. Sistemas similares han sido desarrollados para realizar una búsqueda exhaustiva de imágenes y mostrar vistas particulares de las mismas al usuario, tal es el caso del proyecto Mocha de la Universidad de Maryland, del Toolbox de Universidad de Puerto Rico y el TerraServer creado por Microsoft.

La tesis se organiza de la siguiente manera: en el capítulo 2 se citan los trabajos previos relacionados con el tema y se hace una breve descripción del Toolbox que fue la herramienta usada para comparar los tiempos de procesamiento a nivel local y los tiempos usados por ImageCutter. El capítulo 3 describe la arquitectura de los componentes de ImageCutter y describe su modo de operar. El capítulo 4 establece el diseño experimental y explica la metodología usada para el desarrollo del trabajo. El capítulo 5 presenta los resultados y la interpretación de los resultados obtenidos y finalmente en el capítulo 6 presentan las conclusiones del trabajo.

CAPITULO 2 TRABAJOS PREVIOS

La percepción remota es la adquisición y medición de información del mundo físico mediante la detección de radiación, de manera que se pueda identificar y catalogar la materia observada. Una de las técnicas utilizadas para la detección de materiales es la percepción hiperspectral. Esta técnica utiliza información de distintas bandas de luz para identificar las propiedades del mundo físico. Dicha información es almacenada en la memoria de la computadora para ser procesada e interpretada. Implícitamente estas imágenes al tener gran envergadura ofrecen dificultades para su almacenamiento y procesamiento.

Las personas nos comunicamos con el mundo exterior a través de los cinco sentidos. Cada cosa que se percibe se ve limitada a los parámetros de percepción. A medida que avanza la tecnología, se han desarrollado nuevos métodos para ampliar las opciones de percepción e interpretar de mejor manera el mundo físico. Con el paso del tiempo, se han creado sensores más potentes que tienen la capacidad de captar información de más de doscientos largos de ondas distintos, permitiendo así hacer una gráfica continua del espectro de onda para cada píxel de la imagen.

Las imágenes que surgen de esta técnica se conocen como imágenes hiperspectrales, y la grafica del espectro se denomina curva espectral. Esta técnica provee un registro completo de las respuestas de materiales en los distintos largos de

ondas en consideración. Una de las utilidades de la técnica de percepción hiperespectral es el estudio de la superficie terrestre. Se pueden obtener imágenes hiperespectrales por medio aéreo, y comparar las curvas espectrales de la superficie con las curvas de materiales puros estudiados en laboratorios, y de esta manera identificar materiales en la superficie terrestre. Pero un problema que surge es que al tratar de aumentar la resolución en las imágenes se aumenta la información a ser almacenada y por tanto se ocupa mas espacio en la memoria de la computadora; ya que por lo general los datos son extremadamente grandes. Al respecto, se han identificado dos campos principales de investigación. Uno de ellos es la elaboración de mejores algoritmos y programas para automatizar la detección de materia en las imágenes. El otro consiste en buscar la manera para disminuir el espacio de almacenamiento de los datos hiperespectrales, sin sacrificar información indispensable para la detección de materiales, dentro de este contexto se encuentra IMAGECUTTER.

2.1 Análisis de imágenes usando sistemas de bases de datos distribuidas

Por definición un Sistema de Bases de Datos Distribuidas, son aquellas cuyos datos se encuentran físicamente dispersos en varias máquinas, unidas entre sí por medios de comunicación que las integra en forma lógica. En esta área se han desarrollado muchos trabajos de investigación, uno de ellos es el desarrollado en la Universidad de los Andes, en la ciudad de Mérida. El sistema de telemedicina requiere del uso de tecnología en comunicaciones, informática y experiencia médica [2]. Con este

proyecto es posible realizar el diagnóstico de una enfermedad a distancia, mediante la comunicación remota entre médicos y especialistas por medio de la videoconferencia [3]. Para la implementación de sistemas de telemedicina fue necesario desarrollar aplicaciones acordes a las necesidades de los usuarios (médicos y especialistas). En este sentido se desarrolló un sistema de bases de datos con imágenes que tenían la particularidad de accederse a ellas en forma remota.

Actualmente estas aplicaciones están basadas en el concepto de sistemas distribuidos, dicho sistemas permiten distribuir las actividades en diferentes computadoras y ejecutarse concurrentemente para finalmente alcanzar un objetivo definido [4]. El concepto de sistemas distribuidos se extiende a las bases de datos para conformar los sistemas manejadores de bases de datos distribuidas, las cuales son una colección de datos que pertenecen lógicamente al mismo sistema pero están diseminadas en diferentes sitios de una red de computadoras.[5]. Es posible actualmente desarrollar todo este tipo de aplicaciones con lenguajes de alto nivel, orientados al desarrollo de sistemas abiertos (sistemas con alto nivel de portabilidad), además de poder interactuar con aplicaciones desarrolladas en otros lenguajes, tal es el caso del lenguaje Java [6], que permite a través de la definición de applets (programas en java ejecutables desde un navegador de internet, por medio de páginas html), la ejecución de una amplia variedad de aplicaciones en forma remota.

2.2 Organización Meteorológica Mundial (OMM)

Las bases de datos distribuidas de la organización meteorológica del mundo también hacen su contribución en esta área. El proyecto de OMM, coordinado por el Departamento del Reloj del Tiempo del Mundo, se ha desarrollado para proporcionar acceso a los datos meteorológicos, la sede principal de este proyecto se encuentra en Ginebra, Suiza.

Los propósitos de la OMM son facilitar la cooperación internacional en el establecimiento de redes de observaciones meteorológicas e hidrológicas y su fin es promover el intercambio rápido de información. Además de estandarizar las observaciones de este tipo, se encarga de publicar un uniforme con las observaciones y los resultados procesados estadísticamente. También fomenta el uso de la meteorología en la aviación, la agricultura y a otras actividades humanas; promueve la hidrología operacional y anima la investigación y el entrenamiento en meteorología.

2.3 Índices Espaciales

Los índices espaciales para objetos permiten buscar datos rápidamente según sus localizaciones en el espacio. Las imágenes se sitúan dentro de una base de datos facilitando al usuario la búsqueda de la imagen de interés. Se sabe que los métodos tradicionales para la indexación de direcciones (ej. *hashing* y los *B-Trees*) no son los

mejores para objetos situados en espacios multidimensionales. Una estructura dinámica de los índices llamada R-Tree organiza los datos, basados en la localización espacial y proporciona los algoritmos de búsqueda necesaria para encontrar datos y para mantener la estructura actualizada [7].

2.4 Un protocolo de Internet para Ambientes Virtuales (DWTP)

Esta herramienta es un protocolo de capa que se usa en ambientes virtuales compartidos en el Internet. Proporciona una arquitectura de red escalable para los mundos virtuales distribuidos. Hasta hace muy poco tiempo, los mundos virtuales compartidos estaban limitados a ciertas áreas con aplicaciones particulares y a redes de alta velocidad. Actualmente, están disponibles y llegan a una gran cantidad de usuarios mediante la introducción de la realidad virtual que modela el habla (VRML) y que se ha establecido como un estándar para el intercambio de los mundos 3D en el internet.

2.5 TerraServer

Microsoft TerraServer es otro de los productos desarrollados en esta área. Específicamente, es un depósito de imágenes aéreas, satelitales y topográficas de la tierra, usando una base de datos SQL (almacén espacial de los datos) disponible vía Internet. Es uno de los ocho más grandes atlas en línea con más de un Terabyte de imágenes examinados por el instituto de geología de Estados Unidos (USGS). Utiliza los browsers

del Internet SPIN-C2 para proporcionar interfaces del espacio y el texto intuitivos de los datos de la imagen. Los usuarios no necesitan hardware, ni software especial ni el conocimiento para "hojear" imágenes. TerraServer describe como el sistema puede descomponer las imágenes grandes del geo-espacio (que son difíciles de manipular) en subimágenes que son más fáciles de utilizar. La metadata sobre las imágenes también se almacena dentro de la base de datos SQL y está disponible vía el Internet. TerraServer demuestra que la base de datos relacional de fines generales puede manejar depósitos de imágenes de gran escala, y que un Web-browser puede ser un buen sistema para la presentación de las imágenes del geo-espacio.

2.6 Un Sistema Middleware de Bases de Datos (MOCHA)

MOCHA [8, 9, 10] es un sistema middleware para integrar bases de datos heterogéneas. Se diseñó para interconectar fuentes de datos distribuidos sobre una red de computadoras. Estos sistemas siguen típicamente una arquitectura centrada alrededor de un servidor que integra los datos. Este servidor provee aplicaciones de cliente con una visión uniforme y un mecanismo para tener acceso a los datos disponibles en cada fuente. Una vista uniforme de los datos es hecha por la imposición de un modelo global de la información local usado por cada fuente. MOCHA está diseñada para trabajar en ambientes de gran área y se basa en la idea de que las funciones definidas por el usuario se deben mover a sitios remotos por el sistema middleware. Esto se observa en el código compilado de Java que envía los tipos y las funciones de datos avanzados a los instrumentos que se utilizarán en fuentes alejadas de los datos [11, 12].

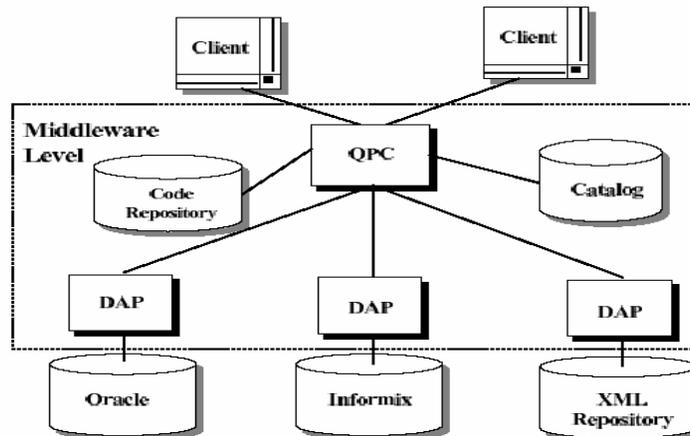


Figura 1: Arquitectura de Mocha

2.7 Herramienta - Toolbox Desarrollada en Matlab

Como parte de este proyecto se utilizó un Toolbox desarrollado en Matlab en la Universidad de Puerto Rico para hacer clasificación supervisada, clasificación no supervisada y reducción de la dimensionalidad de imágenes multispectrales adquiridas de diversos sensores disponibles en el momento. Este Toolbox se puede ejecutar desde cualquier computadora con UNIX, Windows y Macintosh quienes deben tener un ambiente de Matlab instalado [13]. Antes de hacer la descripción del Toolbox usado en percepción remota, se dan algunos conceptos básicos importantes para una mayor comprensión del mismo.

2.7.1 Sensores Remotos

Percepción remota es la adquisición y medición de información acerca del mundo físico mediante la detección de radiación y partículas, de manera que el usuario pueda identificar y catalogar los objetos observados. Dicha información se analiza a través de imágenes.

Una información espectral de una escena puede ser representada por una imagen espectral, donde cada "píxel" tiene las coordenadas espaciales x , y y además de la coordenada espectral L , que representa la radianza de un albedo en el intervalo de longitud de onda de una banda espectral.

2.7.2 Reconocimiento de Patrones

Es la ciencia -y arte- de encontrar patrones en los datos, que pueden ser extraídos a través de clasificación.

2.7.3 Entrenamiento (Training)

Es el proceso de definir los criterios por los cuales esos patrones son reconocidos. En un training supervisado el proceso es controlado por el usuario, que selecciona grupos de píxeles que representan patrones o figuras de distintos tipos de coberturas, que reconoce o que puede identificar con la ayuda de otros elementos tales como fotos aéreas

o datos de campo. El conocimiento de los datos, y de las clases deseadas, es requerido en este caso antes de la clasificación. Por la identificación de patrones, el usuario puede entrenar (train) a la computadora para identificar posteriormente grupos de píxeles con las mismas características que las definidas.

2.7.4 Firmas (Signatures)

El resultado del training es un set de firmas (*Signatures*) que define una muestra de entrenamiento o cluster. Cada firma corresponde a una clase, y es usada junto con una regla de decisión para asignar los píxeles presentes en el archivo imagen a una clase. Las firmas pueden ser paramétricas o no paramétricas.

Una firma paramétrica esta basada en los parámetros estadísticos (ej: media, matriz de covarianza) de los píxeles del cluster. Una firma no paramétrica, en cambio, se basa en objetos discretos (polígonos o rectángulos) que se definen en una figura del espacio de la imagen.

2.7.5 Imágenes Hiperespectrales

Con el paso del tiempo, se han creado sensores más elaborados que tienen la capacidad de captar información de más de doscientos largos de ondas distintos, permitiendo así hacer una gráfica continua del espectro de onda para cada píxel de la imagen. Las imágenes que surgen de esta técnica se les conoce como imágenes

hiperespectrales, y la grafica del espectro se le conoce como curva espectral. Esta técnica provee un registro completo de las respuestas de materiales en los distintos largos de ondas en consideración.

2.7.6 Clasificación

- Clasificadores Paramétricos: Maximum Likelihood, Fuzzy Maximum Likelihood, Euclidean distance, Fuzzy Euclidean distance, Fisher's to linear discriminant, Mahalanobis distance and angle detection.
- Clasificadores No Paramétricos: Clasificador KNN

Una clasificación es el proceso de extracción de la información en las imágenes para reconocer patrones y objetos homogéneos. Los métodos de clasificación se utilizan para examinar las áreas de la superficie terrestre que exhiben un significado igual en imágenes digitales.

2.7.7 Reglas de clasificación Paramétricas

Cada píxel es clasificado en función de representaciones estadísticas.

a)- Distancia mínima: También llamada distancia espectral, calcula la distancia espectral de cada píxel con la media de cada firma. Cada píxel es asignado a la clase cuya

media espectral es más cercana al valor del píxel. Las ventajas son que no quedan píxeles sin clasificar y que es un método rápido. Las desventajas, que píxeles espectralmente lejos de cualquier firma son asignados de todas formas a alguna clase. Tampoco considera la variabilidad en las clases, con lo cual se pueden clasificar incorrectamente ya sea sobreclasificando (en agua por ejemplo en donde hay poca variación) o subclasificando (en zonas urbanas, en donde hay alta variabilidad).

b)-Distancia de Mahalanobis: Es similar al anterior, solo que también utiliza la matriz de covarianza en la ecuación. La ventaja es que toma en cuenta la variabilidad de las clases. Es más útil que la distancia mínima en casos en donde los criterios estadísticos deben tenerse en cuenta pero en donde los factores de peso (disponible con *Maximum Likelihood*) no son necesarios. La desventaja es que tiende a sobreclasificar las firmas con valores grandes en la matriz de covarianza (si hay una gran dispersión de los píxeles en un cluster entonces la matriz de covarianza de esa firma contendrá valores grandes. Como es un método paramétrico, depende duramente de una distribución normal de los datos en cada banda.

c)-Maximum Likelihood: Esta basado en la probabilidad de que un píxel pertenezca a una dada clase. La ecuación básica asume que esta probabilidad es igual para todas las clases y que las bandas de entrada tienen distribuciones normales. Si este no es el caso, tal vez se pueden obtener mejores resultados con el método de paralelepípedo o de mínima distancia, o corriendo una primera pasada de clasificación con el paralelepípedo como criterio de clasificación, previo al uso de Maximum

Likelihood. Como ventaja tiene que es el más preciso de los clasificadores (si las muestras tienen una distribución normal) ya que considera la mayoría de las variables. La desventaja es que es más lento de los métodos y que al igual que el anterior depende duramente de una distribución normal de los datos. También tiende a sobreclasificar firmar con valores grandes en la matriz de covarianza.

2.7.8 Ecuaciones Usadas

Las ecuaciones usadas para el desarrollo de cada uno de los algoritmos de los siete Clasificadores son:

2.7.8.1 Distancia Euclidiana (Euclidean Distance)

En este clasificador, toda la información está en valores esperados y no tiene matriz de covarianza. El límite de decisión es siempre lineal.

$$g_i(X) = -(X - M_i)^T (X - M_i)$$

2.7.8.2 Discriminante Lineal de Fisher's y Mahalanobis

$$g_i(X) = -(X - M_i)^T \Sigma^{-1} (X - M_i)$$

2.7.8.3 Maximum Likelihood

Máxima probabilidad (Maximum Likelihood) es el método de clasificación supervisada más común y usada con imágenes multiespectrales en sensores remotos.

$$g_i(X) = -\frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i)$$

2.8 Ventana Principal del Toolbox

La figura 2 muestra la ventana principal de este Toolbox. Esta ventana posee en su parte superior, una barra de herramientas que permite la interacción de los usuarios con la herramienta, permite entre otras cosas: cargar la imagen en diferentes formatos y seleccionar el tipo de proceso a realizar. Este Toolbox está desarrollado en su totalidad en Matlab y permite hacer procesos de clasificación supervisada, no supervisada y reducción de la dimensionalidad; técnica que es usada para reducir el número de bandas de una imagen, mediante un análisis de tipo estadístico de tal manera que el número de bandas en las que queda reducida la imagen pueden reflejar las características principales de la imagen [14, 15 y 16]. Cuando las imágenes son demasiado grandes, éstas no pueden ser cargadas por Matlab en los equipos que se describen en la metodología de este trabajo. Una imagen en estos casos puede considerarse grande cuando su tamaño físico supera los 60 MB.

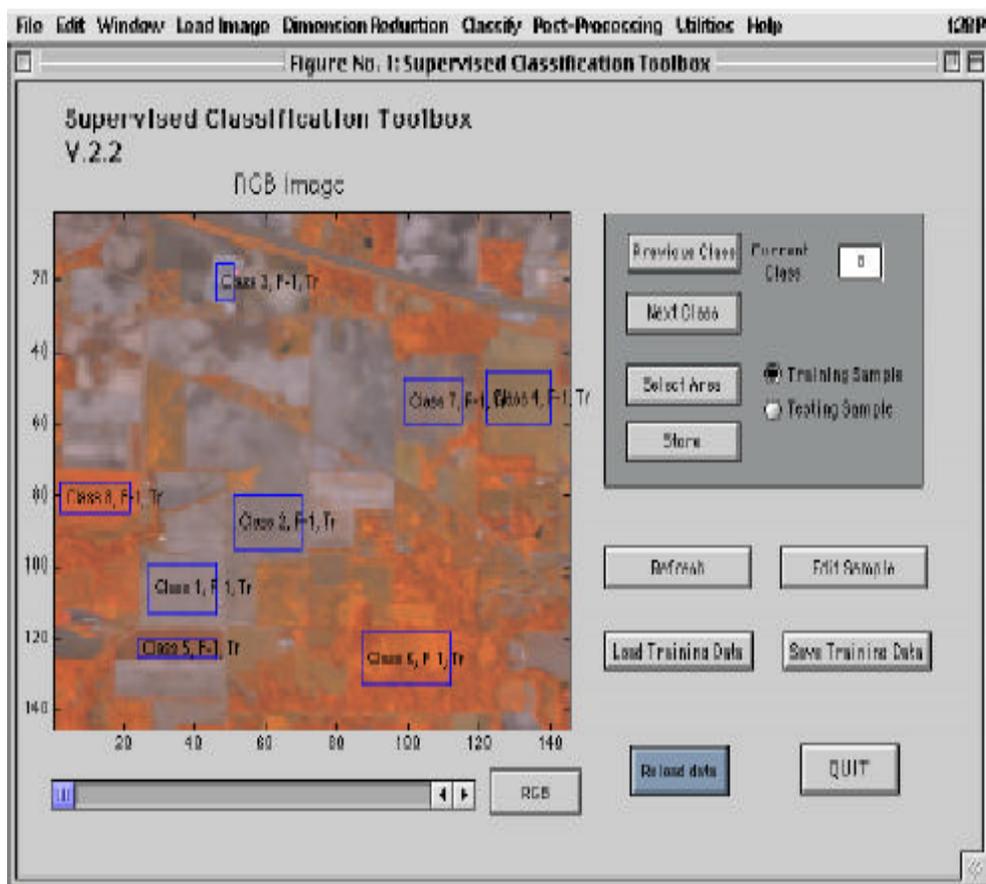


Figura 2: Ventana Principal

Para realizar un proceso de clasificación supervisada con el Toolbox, se carga primero la imagen en uno de los formatos que provee la herramienta, después se selecciona el tipo de proceso a realizar y finalmente se ejecuta el proceso.

La figura 3 es el resultado de un proceso de clasificación supervisada con esta herramienta, utilizando para esto ocho clases y una imagen en 20 bandas, 145 filas, 145 columnas; esta imagen fue tomada con el sensor Aviris.

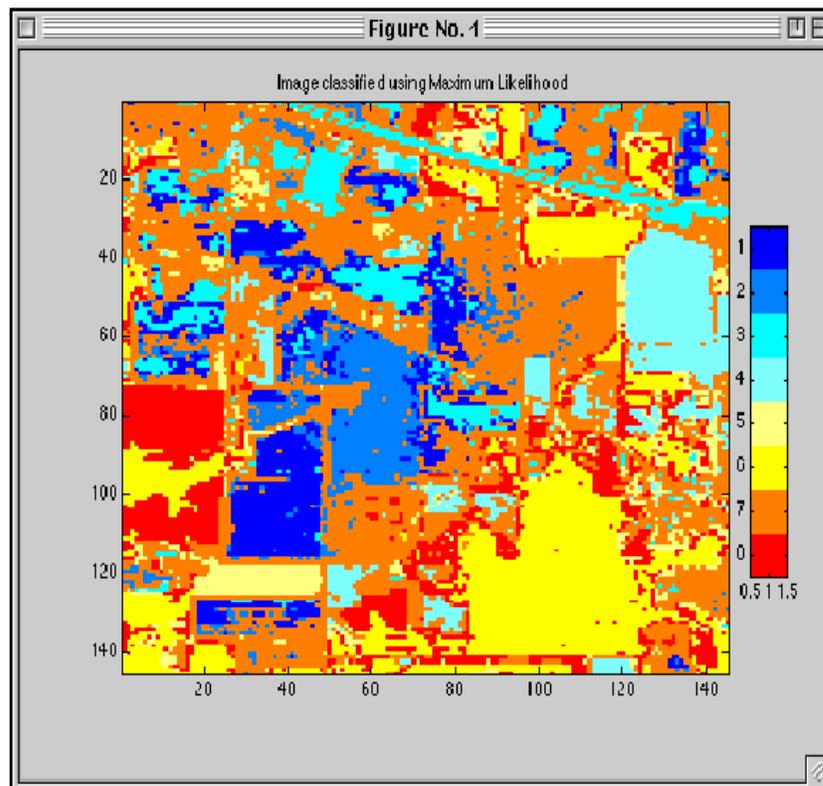


Figura 3: Mapa de Clasificación

La figura 4 ilustra la secuencia de los pasos que el cliente debe seguir para ejecutar operaciones de clasificación supervisada. Cada una de estas acciones se incluye en el menú principal del Toolbox.

Un proceso similar se sigue para ejecutar clasificación no supervisada, excepto el entrenamiento del algoritmo.

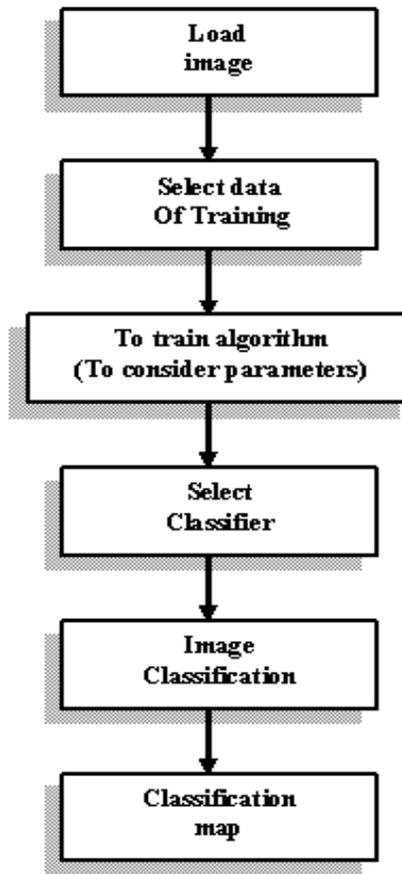


Figura 4: Flujo de Clasificación

CAPITULO 3 ARQUITECTURA DE IMAGECUTTER

Esta sección presenta la arquitectura general de ImageCutter y la discusión del modo operacional.

3.1 Arquitectura de ImageCutter

La figura 5 muestra un esquema general del modo operacional del sistema. Ilustra los posibles caminos que seguirán los requerimientos iniciados del lado del cliente, con respecto a las imágenes a procesar, al procesamiento y a los resultados finales.

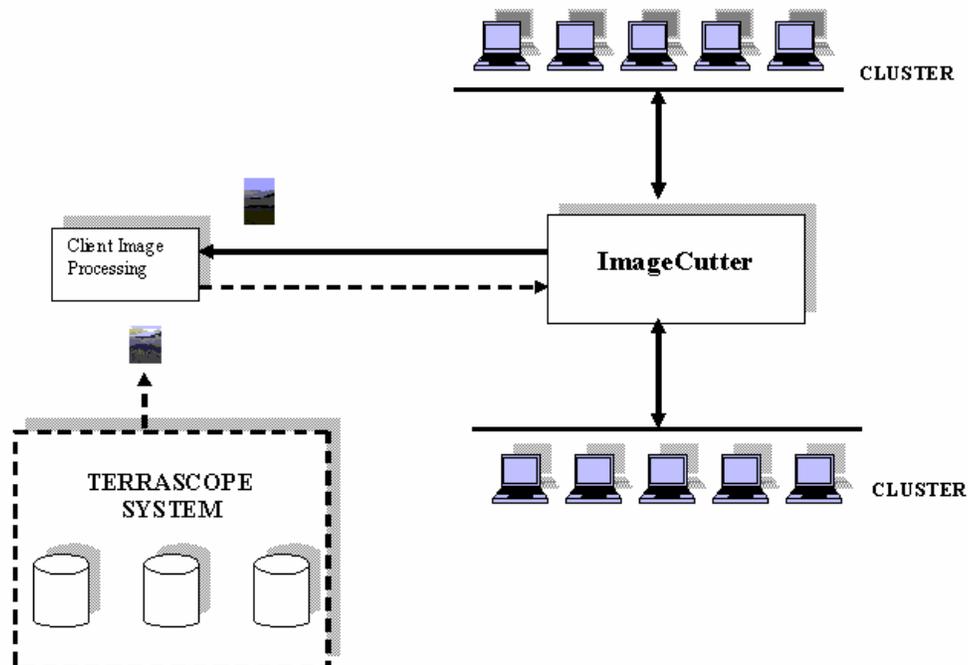


Figura 5: Esquema general del Sistema

Dado un cliente (ver sección de Toolbox [13]), este se comunica con el componente de ImageCutter. Este mecanismo de comunicación se basa en los servlets de Java [17], y permite el envío de la información, tales como requerimientos y parámetros de las imágenes. El servlet correspondiente en el lado de ImageCutter analiza estos requerimientos y envía una orden a los diversos nodos de modo que en este se ejecute la acción requerida por el usuario. Una vez que el proceso finaliza, los resultados son devueltos al usuario usando el mismo servlet.

3.2 Modelo general del Sistema

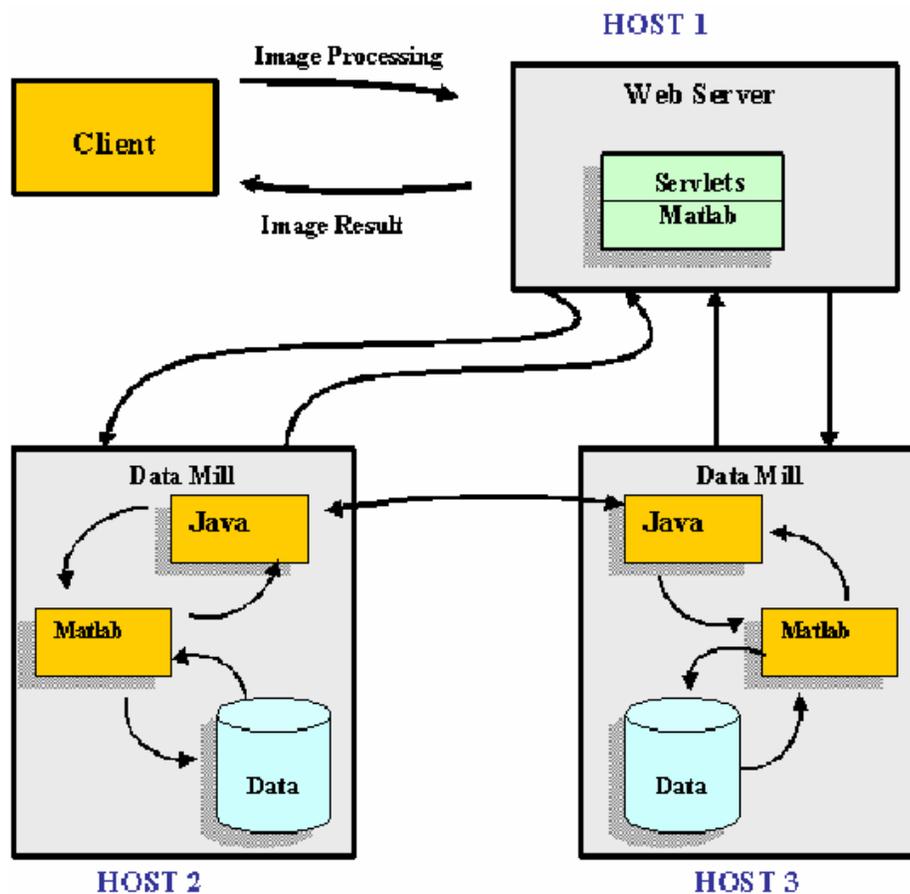


Figura 6: Modelo general del Sistema

La figura 6 muestra una vista más detallada de la interacción entre un cliente y el componente de ImageCutter. También ilustra la interacción del ImageCutter con diversos sitios en el sistema de TerraScope.

La imagen es recibida por el ImageCutter, que analiza y determina la operación para ejecutarse. El ImageCutter entonces identificará los sitios que pueden ayudar en el proceso de cada imagen. Cada uno de estos sitios se llama un DataMill. Se asume que Matlab está funcionando en cada DataMill. Este Matlab es puesto en ejecución a través de una llamada desde Java (que funciona en el DataMill) que acepta los comandos del ImageCutter para que procese eficientemente una imagen dada; el ImageCutter romperá la imagen en pedazos más pequeños, y envía cada pedazo a un DataMill diferente. Además, les enviará cualquier código requerido (Matlab o Java) que sea necesario para procesar cada imagen.

La comunicación entre el ImageCutter y un DataMill se pone en ejecución a través de *Sockets* [18]. Una vez que cada DataMill termine su operación, enviará los fragmentos procesados de la imagen de nuevo al Web-Server; en este sitio es donde se reconstruirá la nueva imagen.

Finalmente, desde el Web-Server y a través de una conexión HTTP, se envía al cliente la imagen resultado o una URL que indica la ubicación de esta.

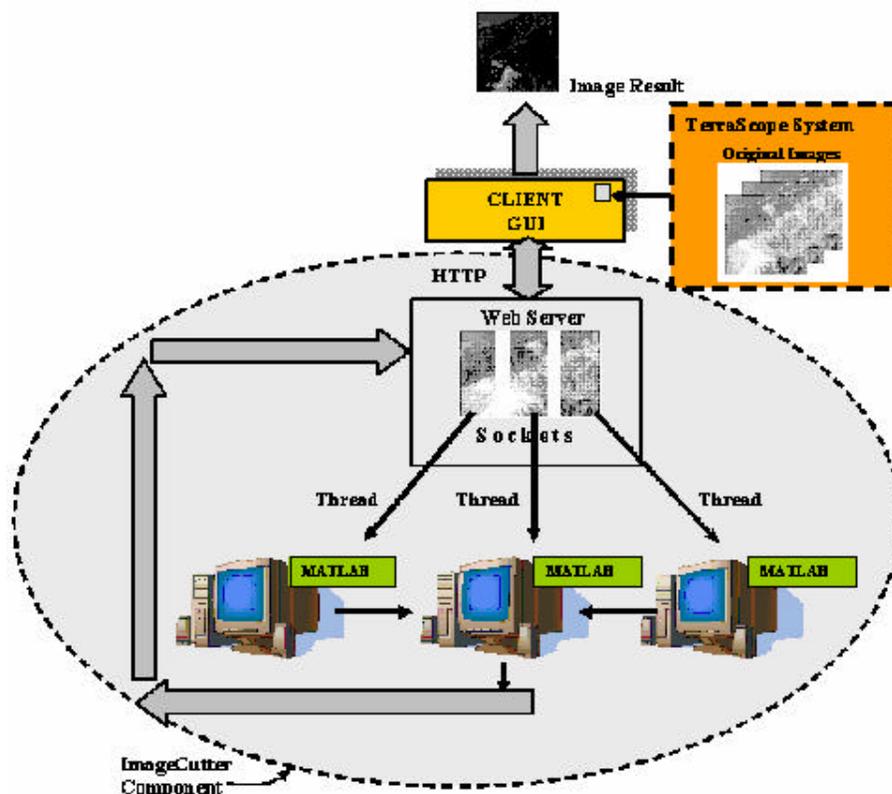


Figura 7: Procesamiento de una imagen con ImageCutter

Como El ImageCutter utiliza *Sockets* para comunicar los diferentes programas (en este caso Java y Matlab), crea un *Thread* por cada pedazo de imagen; por este mecanismo envía a cada nodo el correspondiente pedazo de imagen. En cada uno de estos nodos (*Data Mill*) un servlet invoca Matlab que es el encargado de ejecutar el código que hace el proceso que el usuario elige. Una vez cada *Data Mill* termina su proceso devuelve el pedazo de imagen procesada al Web Server. Cuando el Web Server termina de recibir todos los pedazos de imagen procesados, los reconstruye (en Matlab) y procede a cambiar el formato de la imagen de Matlab a JPEG. Una vez ImageCutter cambia el formato de la imagen, procede a mostrar el resultado final al cliente.

3.3 Alcance de ImageCutter

Son muchos los procesos que se pueden hacer con imágenes multiespectrales: Filtros, reducción de la dimensionalidad, clasificación no paramétrica, clasificación paramétrica, etc. Existen varias reglas de la clasificación de imágenes: No paramétrica, paramétrica, píxeles no clasificados y traslapeo de clases.

ImageCutter está desarrollado para trabajar con imágenes multiespectrales que aplican clasificación paramétrica.

3.4 Procesos que ejecuta ImageCutter

ImageCutter fue diseñado para ejecutar siete procesos de clasificación supervisada, estos son:

Maximum Likelihood

Fuzzy Maximum Likelihood (Maximum Likelihood)

Euclidean distance (Minimum Distance)

Fuzzy Euclidean distance (Minimum Distance)

Fisher's to linear discriminant (Mahalanobis Distance)

Distancia Mahalanobis

Angle detection (Minimum Distance)

3.5 Software que compone ImageCutter

La estructura de ImageCutter desde el punto de vista del Software que lo compone, se muestra en la figura 8.

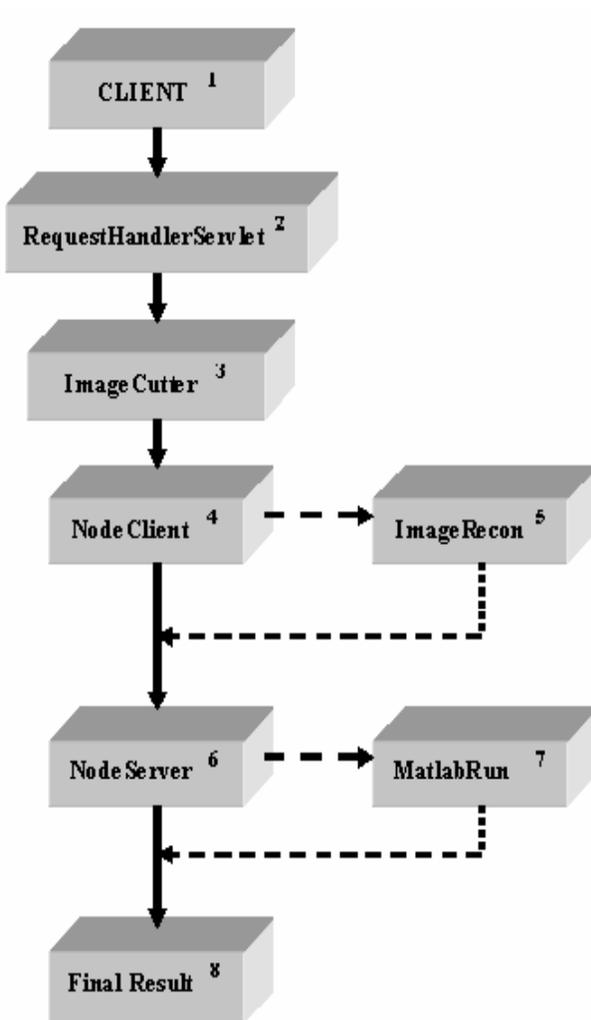


Figura 8: Estructura del Software de ImageCutter

A continuación se hace una breve Descripción de cada una de las clases que se muestran en los rectángulos de la figura anterior enumerados en la parte superior derecha:

CLIENT: Es el usuario propiamente dicho y es quien abre el sistema.

RequestHandlerServlet: Este Servlet toma el Requerimiento del Cliente y lo procesa.

ImageCutter: Busca la imagen en las bases de datos, teniendo en cuentas los requerimientos del Cliente, una vez encuentra la data, corta la imagen en tantos pedazos como nodos activos se encuentren en el sistema. Los formatos de las imágenes deben ser: JPG, GIF, TIF y MAT.

NodeClient: Este Servlet envía a los nodos un pedazo de la imagen que fue cortada con el Servlet anterior y llama el Servlet *ImageRecon*.

ImageRecon: Este Servlet se encarga de reconstruir la imagen.

NodeServer: Es el nodo. Este Servlet toma el pedazo de imagen que le fue entregado, llama el Servlet *MatlabRun*.

MatlabRun: Ejecuta el respectivo código con la imagen recibida que le fue entregada.

FinalResult: Es el resultado final que se le entrega al Cliente.

3.6 Pantallas de ImageCutter

ImageCutter es una herramienta creada usando todas las técnicas de usabilidad y la interacción Humano-Computadoras que facilita el manejo de esta a los usuarios. Las pantallas aparecen más adelante, describen el uso y manejo de la herramienta por parte de los usuarios.

3.6.1 Ventana Principal



Figura 9: Ventana Principal de ImageCutter

Para acceder ImageCutter y ver la pantalla de la figura 9, los usuarios deben ejecutar el servlet *mainServlet* desde SUN-ONE Studio o desde el Web Browser: <http://localhost:8081/Terrascope/Index.jsp>.

En el menú de barras en la parte superior de esta pantalla se encuentran dos opciones:

- 1) **File**: Permite cargar imágenes en varios formatos diferentes (JPEG, TIF y MAT)

- 2) **Classification**: esta opción permite hacer clasificación supervisada de imágenes, para lo cual la herramienta le permite a los usuarios escoger el tipo de clasificador que quiere aplicar a su imagen; esta información se le muestra a los usuarios en la parte derecha de esta pantalla.

Adicionalmente la herramienta permite a los usuarios crear sus propias muestras de entrenamiento, esto se hace con la ayuda de un applet que facilita el movimiento a través de la imagen usando el “mouse” y los “scroll bar”. Para crear las muestras de entrenamiento, los usuarios solo deben marcar con la ayuda del “mouse” la cantidad de muestras por clase, éstas se deben guardar antes de marcar la siguiente muestra, y así sucesivamente para cada una de las clases que determinarán la clasificación.

3.6.2 Pantalla para cargar la imagen



Figura 10: Pantalla para comenzar con ImageCutter

Esta ventana permite a los usuarios cargar las imágenes, brindando la posibilidad de escoger el tipo de formato de la imagen. Este formato es aquel en el que la imagen fue almacenada (MAT, JPEG, TIF). Estas imágenes son cargadas desde un Web-Browser y permite también almacenar los resultados con el fin de aumentar el repositorio de la base de datos.

3.6.3 Procesos Iniciales

Antes de comenzar cualquier proceso con una imagen, se deben activar los nodos en el sistema, esta cantidad depende del número de máquinas que conforman el Cluster,

por lo tanto en ningún caso este número podrá ser mayor que la cantidad de computadoras (procesadores) en el Cluster. ImageCutter se puede ejecutar de dos formas: Local y Distribuido.

Proceso Local: Para este proceso se abre un Web-Browser igual al descrito en la sección 3.6.1 (<http://localhost:8081/Terrascope/Index.jsp>) y desde esta única máquina, que sería el localhost, se hace todo el proceso con la imagen.

Proceso Distribuido: Para ejecutar ImageCutter de forma distribuida se debe hacer lo siguiente: Activar los nodos en el sistema, para esto se crean los *Thread* ejecutando el comando que se muestra en la figura 11, donde *nodeServer* es el *Servlet* que crea el nodo y 8081 es el número del puerto en el cual queda activo el *Thread*. El número de nodos activos en el sistema debe ser menor que la cantidad de máquinas que conforman el Cluster. Uno de estos nodos se comportaría como servidor.

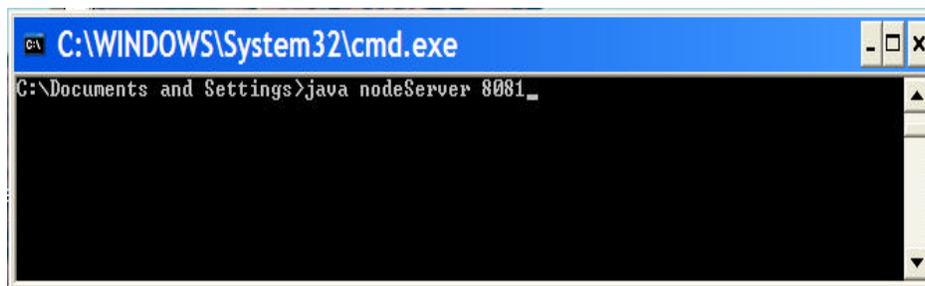
A screenshot of a Windows command prompt window. The title bar is blue and contains the text 'C:\WINDOWS\System32\cmd.exe'. The main area of the window is black with white text. The text shows the current directory as 'C:\Documents and Settings>' followed by the command 'java nodeServer 8081_'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Figura 11: Comando para abrir un nodo

Es importante recordar que Matlab debe estar instalado tanto en el nodo servidor como en los nodos de proceso.

3.6.4 Pantalla de ImageCutter con imagen cargada



Figura 12: Vista de una imagen cargada con ImageCutter

La pantalla anterior muestra una imagen cargada por ImageCutter. En este caso se abre una sesión Matlab que ejecuta el proceso de cargar la imagen seleccionada (si la imagen está en formato Matlab). Posteriormente se carga un archivo con las muestras de entrenamiento correspondientes a dicha imagen; este archivo también debe estar en formato Matlab; una vez terminado este proceso, ImageCutter verifica la cantidad de nodos activos y divide la imagen en esta misma cantidad.

3.6.5 Pantalla para ejecutar proceso seleccionado



Figura 13: Ejecuta proceso de clasificación seleccionado

Esta pantalla permite a los usuarios, escoger uno de los siete procesos de clasificación supervisada que provee ImageCutter. En este caso, la herramienta envía cada pedazo de imagen a un nodo diferente y en estos sitios ejecuta un Servlet llamado *MatlabRun*, que es el encargado de realizar el proceso de clasificación seleccionado.

3.6.6 Pantalla con resultado final

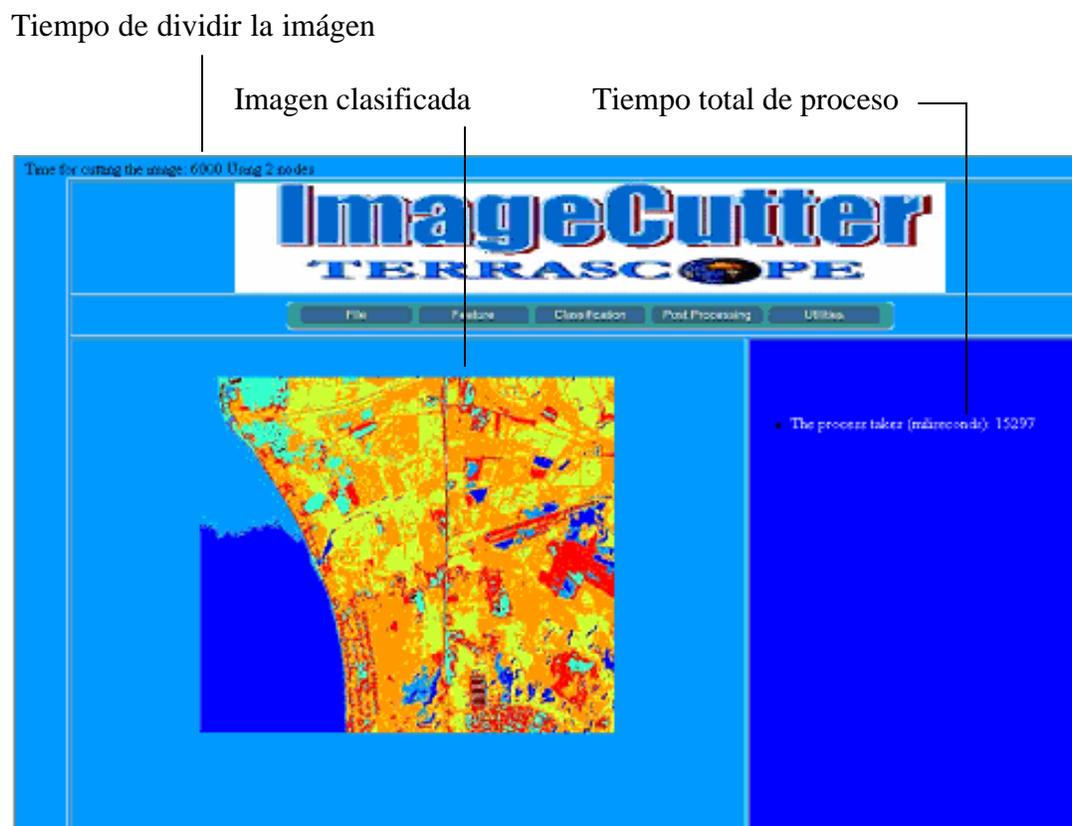


Figura 14: Mapa de Clasificación, resultado final

Esta pantalla muestra el resultado final obtenido aplicando el clasificador escogido por el usuario. Los tiempos que se indican en la figura 14, están dados en mili segundos y corresponden en la parte superior, al tiempo de dividir toda la imagen en tantas partes como nodos activos hay en el sistema y en la parte media-derecha de esta pantalla el tiempo total empleado por ImageCutter en todo el proceso.

CAPITULO 4 MATERIALES Y METODOS

Este proyecto fue realizado con la colaboración del grupo ADM en la Universidad de Puerto Rico, recinto Universitario de Mayagüez y CenSSIS (*Center Subsurface Sensing and Imaging Systems*), como un mecanismo para compartir recursos y conocimientos entre las instituciones académicas que conforman el proyecto CenSSIS: Universidades de Puerto Rico, Northeastern, Boston y el Instituto Politécnico Rensselaer. El objetivo principal del proyecto CenSSIS es procesar y analizar imágenes de la superficie de la tierra, información oceanográfica, imágenes biomédicas y otro tipo de imágenes que cada uno de estos institutos educativos ha obtenido de satélites como: Modis, RadarSat, Aviris y cámaras multiespectrales. Mucha de esta información se encuentra en cintas magnéticas y de forma aislada. Debido a que actualmente existe muy poca oportunidad de intercambio de este tipo de data de una forma automática entre las instituciones que conforman CenSSIS; es a partir de estas necesidades, que se plantea, la necesidad de crear el proyecto TerraScope, que será una de las soluciones que permitan el intercambio y el proceso de imágenes entre la comunidad de CenSSIS y de todas aquellas instituciones que quieran compartir esta y su propia información.

4.1 Diseño Experimental

La figura 15 muestra el diseño experimental seguido durante el desarrollo de este trabajo; este diseño consiste principalmente de la elaboración de un código fuente en Java

y Matlab con el fin de realizar clasificación supervisada de imágenes multiespectrales; partiendo de esta idea como meta principal, se creó ImageCutter, que divide las imágenes de acuerdo al número de nodos activos en el sistema, y envía estos pedazos de imagen y código Matlab a los nodos de procesos con el propósito de realizar clasificación supervisada de forma distribuida, para mejorar el Performance en el sistema.

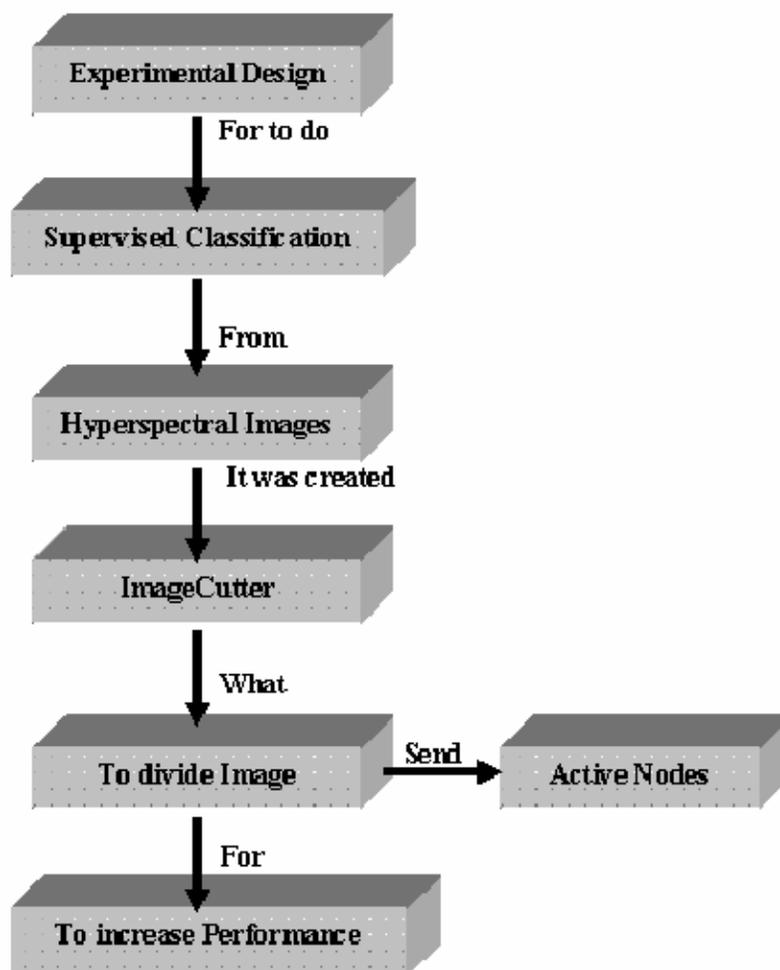


Figura 15: Esquema del diseño experimental

4.2 Metodología

La metodología empleada durante el desarrollo de la tesis tiene dos partes: La primera parte para el desarrollo del software de ImageCutter que consistió básicamente de los siguientes pasos:

a). Determinar y adecuar las máquinas a utilizar para el desarrollo de las pruebas;

la descripción del equipo utilizado es el siguiente:

- Ocho PC-DELL, Ram 1.0GB, Disco Duro de 60GB, Procesador Pentium 4 de 2.4GHz, OS Windows XP.
- Una HP, Ram 512MB, Disco Duro de 60 GB, Procesador Pentium 4 de 2.4 GHz, OS Windows 2000.
- Una red de comunicación TCP/IP de 100Mb/s.

b). Instalar el software necesario para el desarrollo del código de ImageCutter en

dos de las ocho PC-DELL:

- J-Builder 9
- MySQL 1.4
- Sun ONE Studio 4 EC
- Java:

El ImageCutter esta desarrollado en *Java Versión 1.4.1_01* usando *Servlets*, que se ejecutan en el servidor y no presentan ningún tipo de interfaz gráfica puesto que están totalmente controlados por un servicio de red que puede ser un servidor

Web. La decisión de hacer ImageCutter con Servlets está basada en los siguientes criterios:

1. La plataforma de desarrollo es independiente de la plataforma en la que se este ejecutando. Otras soluciones como ISAPI o NSAPI son dependientes de la plataforma y de los servidores donde se ejecuta haciendo muy costoso una migración en la plataforma de ejecución.
 2. Ejecución multithread. Cada una de las peticiones sobre el servlet creará una instancia que se ejecutará de manera independiente. A no ser de que le indiquemos lo contrario. El servlet permanece cargado en memoria por lo que atiende rápidamente las peticiones.
 3. Un servlet puede llamar a otro servlet, incluso a métodos de otros servlets. Esto nos permite que un servlet realice balanceado de carga entre diferentes servlets. Además, desde un servlet, podemos redirigir una petición sobre otro servlet (en la misma máquina o en una máquina remota).
 4. Conexión a Bases de Datos. A través de los servlets podemos establecer conexiones a diferentes tipos de bases de datos. Esta característica acopla perfecta a los servlets dentro de una arquitectura cliente/servidor en 3 capas (cliente - servidor - datos).
 5. Recursos compartidos entre usuarios. Los servlets pueden definir estructuras o información que va a ser compartida por diferentes usuarios que utilicen el servlet.
- Matlab Release 13 con los siguientes Toolboxes:

▪ Simulink	Versión 5.0	(R13)
▪ DSP Blockset	Versión 5.0	(R13)
▪ Data Acquisition Toolbox	Version 2.2	(R13)
▪ Image Processing Toolbox	Version 3.2	(R13)
▪ Instrument Control Toolbox	Version 1.2	(R13)
▪ MATLAB Compiler	Versión 3.0	(R13)
▪ Optimization Toolbox	Versión 2.2	(R13)
▪ SB2SL (converts System Build to Simu..	Version 2.5	(R13)
▪ Signal Processing Toolbox	Versión 6.0	(R13)
▪ Statistics Toolbox	Versión 4.0	(R13)

La segunda parte de esta metodología tiene que ver con las pruebas realizadas a la herramienta ImageCutter. Es importante aclarar que se hacen dos tipos de pruebas, unas realizadas con el Toolbox (este Toolbox, es el que se describe en el capítulo 2 y no se desarrolla o no es parte de esta tesis, se toma solo como punto de referencia y de comparación) y las pruebas que se realizan con ImageCutter. El Toolbox funciona únicamente de forma local. El procedimiento seguido para las pruebas es el siguiente:

- a). Se clasifica cada una de las imágenes con el Toolbox y se toman los tiempos de procesamiento, estos datos se guardan en una tabla
- b). Se clasifican las imágenes utilizando ImageCutter con dos, cuatro, seis y ocho nodos, los datos se guardan en una tabla.
- c). Comparar los resultados obtenidos con el Toolbox y con ImageCutter.

CAPITULO 5 DISCUSIÓN Y RESULTADOS

Como se discutió previamente, ImageCutter procesa imágenes en forma distribuida, mediante la división de una tarea en varias subtareas que son ejecutadas en diferentes nodos de proceso. Al trabajar una imagen en forma distribuida se pretende disminuir el tiempo de ejecución y se espera que éste sea proporcional a la cantidad de recursos computacionales empleados para su procesamiento. Para comparar la eficiencia de ImageCutter respecto al Toolbox se midieron los tiempos de procesamiento usando los siete clasificadores paramétricos y adicionalmente se realizó el proceso con 2, 4, 6 y 8 nodos, respectivamente.

5.1 Resultados

Los Resultados obtenidos en este trabajo están basados en varias imágenes multiespectrales en formato Matlab. En la tabla 1 se da una lista de los atributos de las imágenes usadas en las pruebas.

Tabla 1: Características de las imágenes usadas en las pruebas

Imágen	Filas	Columnas	Bandas	MATLAB		DISK
				Tamaño (BxRxC)	Bytes (Double)	Tamaño (bytes)
Indiana	145	145	20	20x21025	3,364,000	3,366,912
Anasco2	409	565	7	7x231085	12,940,760	1,622,016
Kennedy	397	268	60	60x106396	51,070,080	51,200,000

Es importante tener en cuenta que el tamaño físico de las imágenes en el disco, son diferentes a los tamaños empleados por Matlab en el momento de cargarlas; esto se debe principalmente a que Matlab por defecto carga todas las imágenes en doble precisión (64 bits). Adicionalmente se debe considerar la resolución espacial, ya que está directamente relacionada con los tiempos de procesamiento; por ejemplo: la imagen de Indiana ocupa un espacio de 3,366,912 bytes en disco y la imagen de Anasco ocupa un espacio de 1,622,016 bytes, sin embargo esta misma imagen ocupa 12,940,760 bytes cuando es cargada por Matlab, debido a la resolución espacial (número de filas y columnas) de Anasco es mucho mayor que en la imagen de Indiana.

En las tablas 2, 3 y 4, la columna identificada como LOCAL, corresponde al tiempo de procesamiento con el Toolbox (un nodo); las columnas marcadas como **Ef** son la eficiencia de ImageCutter con dos, cuatro, seis y ocho nodos, comparados con los resultados locales obtenidos del Toolbox; y la información central corresponde a los tiempos totales que tarda ImageCutter para: 1) dividir la imagen; 2) enviar cada pedazo de imagen a los nodos de proceso; 3) invocar Matlab y clasificar cada pedazo de imagen en los nodos de proceso; 4) devolver los pedazos de imagen clasificados al nodo servidor y 5) reconstruir la imagen y mostrar resultados.

5.1.1 Resultados con la imagen de Indiana

La tabla 2 muestra los tiempos de proceso local, con dos, cuatro, seis y ocho nodos, estos tiempos están dados en segundos y corresponde al tiempo total de proceso.

Tabla 2. Tiempo de procesamiento de Indiana con 1, 2, 4, 6 y 8 nodos

Indiana	Local	2 Nodos	4 Nodos	6 Nodos	8 Nodos	Ef2	Ef4	Ef6	Ef8
Clasif.	(seg)	(seg)	(seg)	(seg)	(seg)	(%)	(%)	(%)	(%)
ML	62.19	13.01	13.51	13.74	13.73	79.08	78.28	77.91	77.92
FML	66.99	12.75	13.15	13.23	13.97	80.97	80.37	80.25	79.15
ED	29.21	12.66	12.78	13.30	14.84	56.66	56.25	54.47	49.20
FED	30.56	12.63	12.76	13.11	14.60	58.67	58.25	57.10	52.23
FLD	51.91	12.89	12.95	13.02	13.50	75.17	75.05	74.92	73.99
MD	53.64	12.74	13.07	13.49	13.55	76.25	75.63	74.85	74.74
AD	36.80	12.53	12.55	13.32	13.90	65.95	65.90	63.80	62.23

Los tiempos parciales de envío, recibo, partición y clasificación se encuentran en el apéndice B. El tiempo de clasificación se refiere al tiempo que tarda el algoritmo para clasificar un pedazo de la imagen en uno de los nodos.

Como se puede observar, los porcentajes de eficiencia para 2, 4, 6 y 8 nodos son similares entre si para cada uno de los clasificadores, sin embargo cuando el número de nodos aumenta, la eficiencia tiende a disminuir lentamente, pero para efectos del proceso esta diferencia no es significativa. También se observa que en todos los casos, la eficiencia obtenida con ImageCutter esta por encima del 56%, con respecto a los resultados obtenidos con el Toolbox. Los tiempos parciales de enviar, recibir, partir, reconstruir y clasificar se encuentran en las tablas B-1, B-2 y B-3 del apéndice.

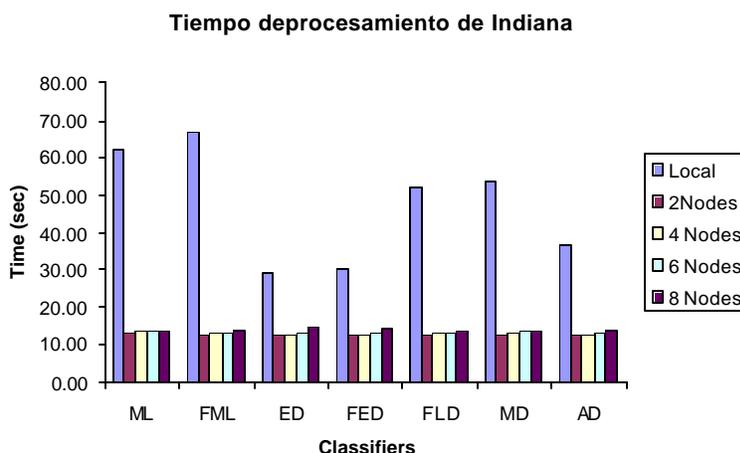


Figura 16. Comportamiento del proceso de Indiana con 1, 2, 4, 6 y 8 nodos

La figura 16 muestra el comportamiento general del Toolbox e ImageCutter; se observa una gran diferencia entre los tiempos obtenidos con las dos herramientas; en todos los casos, éstos se encuentran por debajo de los 15 segundos utilizando ImageCutter, mientras que para el Toolbox, los tiempos oscilan entre los 30 y 70 segundos.

En la figura 17 se muestra el comportamiento del tiempo de procesamiento empleado por ImageCutter con 2, 4, 6 y 8 nodos, se puede observar que para todos los clasificadores el tiempo de proceso total se encuentra por debajo de los 15 segundos y el mejor comportamiento es el que se obtuvo con dos procesadores; esto se debe principalmente a que la imagen utilizada en esta prueba es muy pequeña y cuando se utilizan ocho nodos se incurre en un mayor *overhead* generado en la división, envío y proceso de clasificación de la imagen.

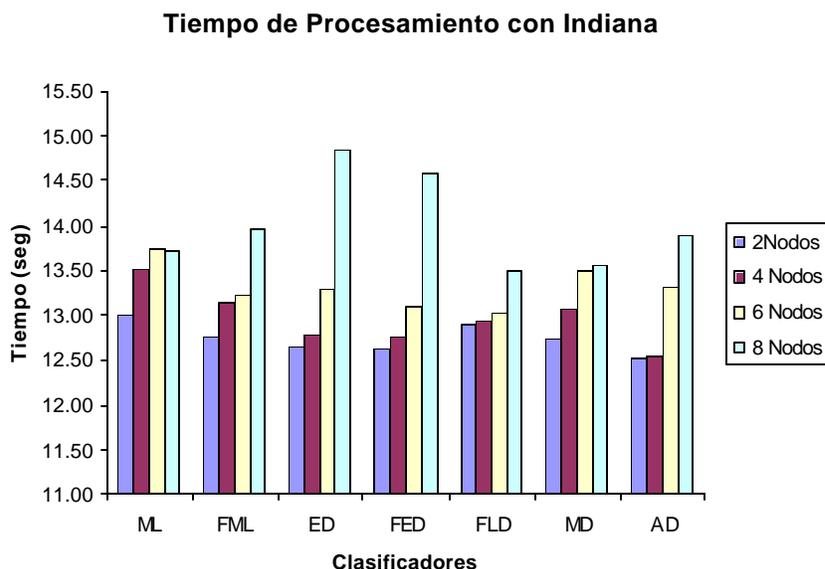


Figura 17: Detalle del comportamiento de Indiana con 2, 4, 6 y 8 nodos

5.1.2 Resultados con la Imagen Anasco

Tabla 3. Tiempo de procesamiento de Anasco con 1, 2, 4, 6 y 8 nodos

Anasco	Local	2 Nodos	4 Nodos	6 Nodos	8 Nodos	Ef2	Ef4	Ef6	Ef8
Clasif.	(seg)	(seg)	(seg)	(seg)	(seg)	(%)	(%)	(%)	(%)
ML	435.67	19.36	15.83	22.65	16.74	95.56	96.37	94.80	96.16
FML	467.67	25.00	16.74	15.46	17.23	94.65	96.42	96.69	96.32
ED	309.33	15.79	15.16	14.62	16.03	94.90	95.10	95.27	94.82
FED	312.00	15.81	15.21	14.49	16.67	94.93	95.13	95.36	94.66
FLD	403.33	18.53	16.70	14.70	16.48	95.41	95.86	96.36	95.91
MD	402.33	17.18	16.22	15.10	16.43	95.73	95.97	96.25	95.92
AD	394.67	17.49	16.44	21.22	16.25	95.57	95.83	94.62	95.88

Esta imagen es físicamente más pequeña que la imagen de Indiana, pero el proceso de clasificación requiere un mayor costo computacional, debido a su resolución espacial. La Tabla 3 muestra los resultados obtenidos con el Toolbox (un nodo o proceso local) y con ImageCutter, con 2, 4, 6 y 8 nodos.

Los resultados de los tiempos obtenidos para cada uno de estos procesos se encuentran en las tablas B-4, B-5 y B-6 del apéndice.

La figura 18 muestra el comportamiento de ImageCutter y el Toolbox, se observa que para todos los clasificadores los tiempos que se obtuvieron con ImageCutter están por debajo de los 25 segundos, con el Toolbox, se encuentran por encima de los 300 segundos..

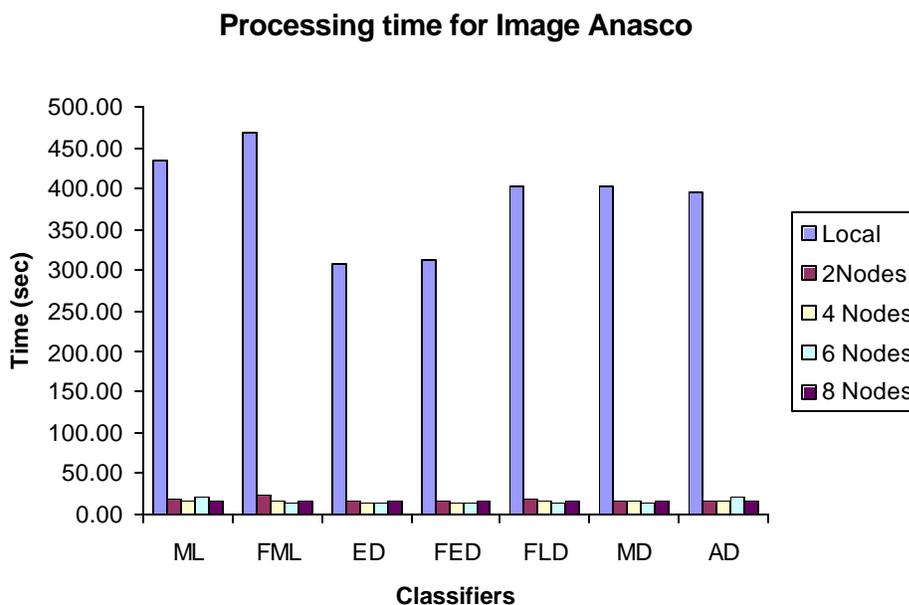


Figura 18: Comportamiento del proceso de Anasco con 1, 2, 4, 6 y 8 nodos

Un aspecto importante observado con esta imagen es la eficiencia obtenida con ImageCutter, esta se encuentra por encima del 90% para todos los Clasificadores; sin embargo no se observó un patrón de comportamiento en los diferentes clasificadores (ver figura 19); se observa por ejemplo que para los clasificadores de Euclidean Distance (ED) y Fuzzy Euclidean Distance (FED), se obtuvieron los menores tiempos, esto se debe fundamentalmente a que en las ecuaciones de estos clasificadores no se emplean las matrices de covarianza. También se observó que para todos los clasificadores el mayor tiempo obtenido, fue el realizado con dos procesadores y se podría pensar que el mejor de los tiempos es el que se obtuvo con 4 y 6 procesadores, sin embargo y a pesar de los mismos resultados, no se puede afirmar de forma clara una cantidad de procesadores con respecto al mejor rendimiento de la herramienta.

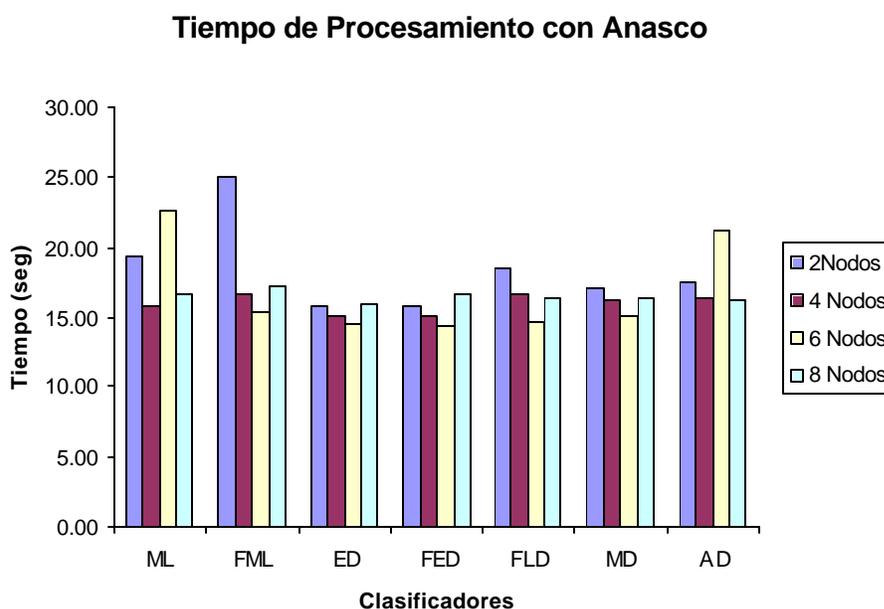


Figura 19: Detalle del comportamiento de Anasco con 2, 4, 6 y 8 nodos

5.1.3 Resultados con la Imágen Kennedy

Esta es la imagen de mayor Tamaño usada en las pruebas, sus dimensiones originales son 397 filas, 268 columnas y 204 bandas; es importante recalcar que esta imagen con toda su dimensionalidad no puede ser cargada por el Toolbox en una máquina como la que se describe en la Metodología de este trabajo, al intentar hacer este proceso, Matlab genera el siguiente error: *Out of Memory*; ImageCutter está en capacidad de procesar este tipo de imágenes, sin embargo para poder realizar un proceso de clasificación supervisada con una imagen de gran dimensionalidad (muchas bandas), se requieren muchas muestras de entrenamiento. Para estas pruebas se hizo de forma independiente (del Toolbox y de ImageCutter) una reducción de la dimensionalidad (se redujo de 204 a 60 bandas), con el fin de poder comparar las dos herramientas.

Tabla 4. Tiempos de proceso de Imágen Kennedy con 1, 2, 4, 6 y 8 nodos

Kennedy	Local	2	4	6	8	Ef2	Ef4	Ef6	Ef8
Clasif.	(seg)	(seg)	(seg)	(seg)	(seg)	(%)	(%)	(%)	(%)
ML	1204.67	106.02	75.84	119.38	114.20	91.20	93.70	90.09	90.52
FML	1202.00	76.86	72.24	77.67	107.61	93.61	93.99	93.54	91.05
ED	154.00	67.96	70.04	64.66	106.08	55.87	54.52	58.01	31.12
FED	162.67	54.47	69.92	70.72	104.02	66.52	57.02	56.53	36.05
FLD	1097.33	59.45	72.37	69.07	109.49	94.58	93.40	93.71	90.02
MD	1090.33	57.07	73.25	70.49	105.23	94.77	93.28	93.53	90.35
AD	215.00	55.16	73.53	63.04	104.66	74.34	65.80	70.68	51.32

En la tabla 4 se encuentran los tiempos obtenidos con ImageCutter y el Toolbox, así como la eficiencia de ImageCutter, esta se encuentra entre un 30% y 50% para los clasificadores que no emplean la matrix de covarianza en sus ecuaciones (ED, FED, AD) para el resto de los casos (ver figura 20) los tiempos de procesamiento se encuentran por encima de los mil segundos.

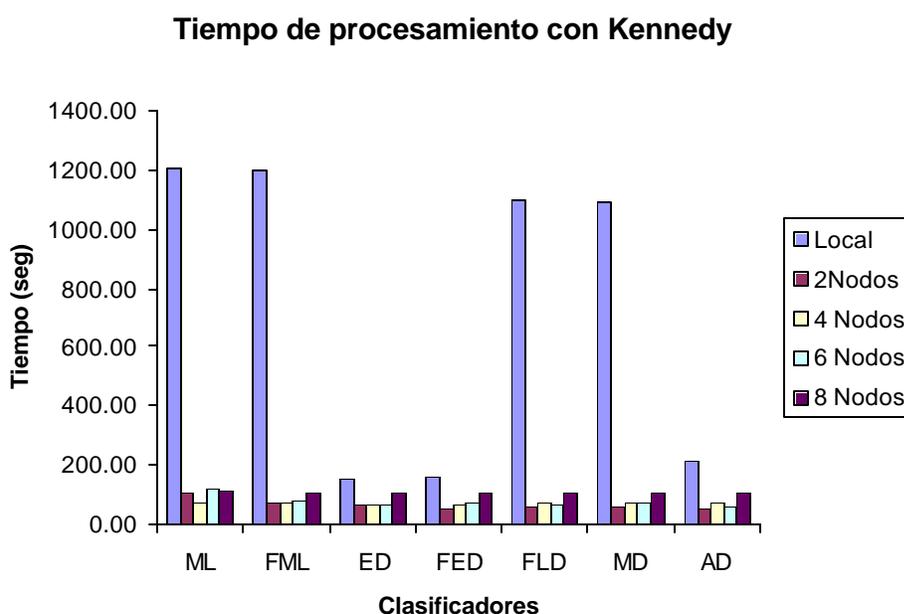


Figura 20: Comportamiento del proceso de Kennedy con 1, 2, 4, 6 y 8 nodos

La figura 21 muestra el detalle del procesamiento distribuido; como se puede observar, con esta imagen tampoco se puede establecer un patrón de comportamiento entre los Clasificadores y el número de procesadores activos en el sistema.

Este tipo de comportamiento (ver figura 21) puede ser debido a contenciones en la red, a que el laboratorio donde se realizaron las pruebas no estaba totalmente dedicado a

este proceso. Los tiempos de enviar, partir, clasificar y reconstruir la imagen se encuentran en las tablas B-7, B-8 y B-9 del apéndice.

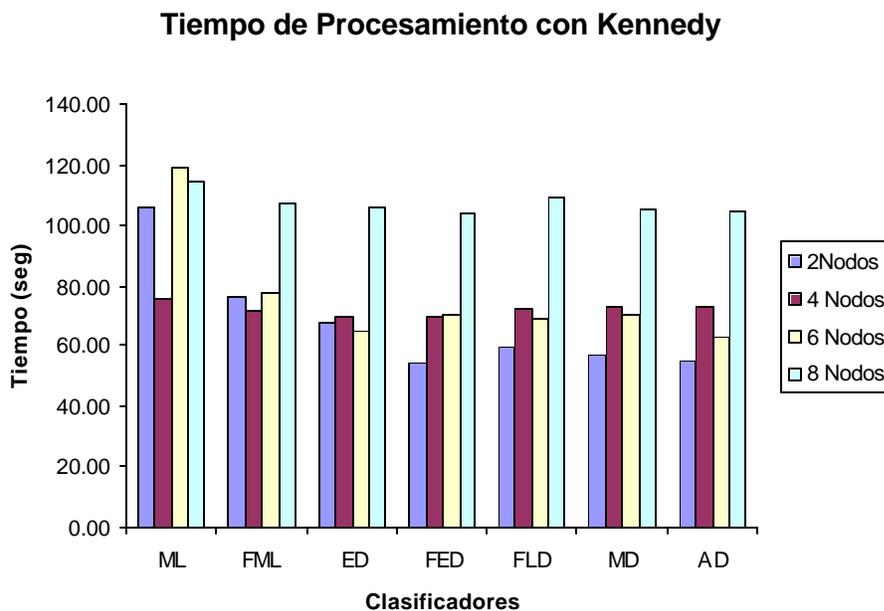


Figura 21. Detalle del comportamiento de Kennedy con 2, 4, 6 y 8 nodos

CAPITULO 6 CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

- ImageCutter es una herramienta que hace posible la clasificación de imágenes multispectrales de una forma distribuida, si el tamaño de estas son suficientemente grandes.

- ImageCutter permite a los usuarios enviar imágenes adquiridas a las computadoras que conforman un cluster para ejecutar el análisis requerido.
- ImageCutter puede ayudar a disminuir el costo computacional incurrido en el procesamiento de imágenes.
- Los procesos que mayor tiempo emplean en la ejecución que realiza ImageCutter son: la división de imágenes y el proceso de un pedazo de imagen en un nodo específico.
- La cantidad de nodos activos en el sistema, no implican necesariamente la cantidad óptima para dividir una imagen y obtener el mejor desempeño del sistema.
- ImageCutter puede combinar las cualidades de áreas como las bases de datos, procesamiento distribuido, soporte y análisis de imágenes espectrales, las bondades del internet, para aprovecharlas en los campos de los sensores remotos, la medicina, Geología, biología y la química entre otros.
- ImageCutter es una herramienta que puede abrir un camino en el área de la investigación que pueda conducir a la creación de nuevos proyectos y/o a la reestructuración de este trabajo, que pueda ampliar el campo de acción para el cual fue diseñado.

6.2 Trabajo Futuro

Unificar el código: Una de las características de ImageCutter tiene que ver con la división de imágenes, esta tarea se realiza usando Matlab; si esta división se logra hacer con Java e inclusive con C/C++, el desempeño de la herramienta en tiempos de procesamiento puede mejorar sustancialmente; adicionalmente esto permitiría una nueva forma de implementar un verdadero procesamiento paralelo o de computación de alto desempeño utilizando herramientas como MPI.

Ampliar el campo de operación: Otras de las tareas por realizar es implementar nuevos procesos: clasificación no supervisada y reducción de la dimensionalidad como componentes principales. Cuando la imagen es dividida, las fronteras de cada división deben pasar información a la división contigua; esto implica condiciones especiales que se deben considerar en el manejo de la información durante los procesos de paralelización.

División de Imágenes: Actualmente ImageCutter divide las imágenes con el uso de Matlab, si esta división se realizara con Java, es muy probable que el desempeño de la herramienta aumente.

Creación de un modelo: Otras de las tareas es implementar un modelo matemático, que les permita a los usuarios sugerirle, en cuántas partes debe dividir una imagen dado su Tamaño y el número de nodos activos en el sistema.

REFERENCIAS

- [1] Rod A. Fatoohi, "Middleware Study Sets Foundation for Building a Distributed Applications Infrastructure," NAS News, March-April 1998, Vol. 3, No. 2, <http://www.sci.nas.nasa.gov/Pubs/NASnews/98/03/middleware.html>.

- [2] Marisela Archila, Guillermo Montilla, Vytautas Subacius," Telemedicina. Realidad Virtual. Robótica. Redes. Supercomputadoras. ¿ Hacia dónde va la Bioingeniería ? ", Proyecto :Telemedicina, Seminario de Bioingeniería en : " Técnica Ingenieril para la práctica médica ", Universidad Central de Venezuela, Centro de Bioingeniería, Febrero 28-29, Caracas, pp 21-26, Caracas, 1996.

- [3] Kim Y., Cabral J., Parsons M., Lipski G., Kirchoerler M., Sado A., Bender N., iSEAHAWK: A telemedicine project in the Pacific Northwest, Department of Electrical Engineering, University of Washington, Seattle, 1995.

<http://icslee.washington.edu/projects/gsp9/spie95/seahawk/>

- [4] Chacón R. Edgar A., Briceño José, Angulo Nestor. " Integration of Widely Distributed Complex Automation Systems", Memorias del I Congreso International de Automation, CONAI, Sao Paulo, Brasil, Septiembre, pp 113-120, 1992.

- [5] Ceri S. and Pelagatti G., " Distributed Database Principles and Systems", Mac Graw-Hill International Editions, New Jersey, 1984.

- [6] Jamie Jaworski, " Java Developer's Guide ", Sams Net, Indiana, 1996.

- [7] Guttman Antonin, "R-Trees: A Dynamic Index Structure for Spatial Searching", National Science Foundation and Air Force Office of Scientific Research, 1984.

- [8] M. Rodriguez and N. Roussopoulos, "MOCHA: A Self-Extensible Database Middleware System for Distributed Data Sources" Institute for Advanced Computer Studies & Department of Computer Science University of Maryland, College Park, MD 20742, 2000.

- [9] M. Rodríguez Martínez and N. Roussopoulos. An Architecture for a Mobile and Dynamically Extensible Distributed DBMS. Technical Report ISR-TR 98-10, CSHCN-TR 98-2, University of Maryland, January 1998.

- [10] M. Rodriguez and N. Roussopoulos. MOCHA: A Self-Extensible Middleware Substrate for Distributed Data Sources. Technical Report MIACS-TR 98-67, CS-TR 3955, University of Maryland, October 1998.
- [11] Atsushi Ito, Teru Yagi, Nobuo Furuya, Masanori Fujioka, Yoshikazu Ikeda. "Service Processing System using *Distributed Data base*". 1988.
- [12] Oracle Corporation. Oracle Transparent Gateways, 1999. URL:<http://www.oracle.com/gateways/html/transparent.html>.
- [13] Arzuaga Emmanuel, et. al, "Supervised Classification Toolbox V 2.2", Larsip 1999.
- [14] Jiménez and Landgrebe, "Supervised classification in high dimensional space: Geometrical, Statistical, and Asymptotical properties of multivariate data", IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, vol 28, no. 1, pp 39-53, 1998.
- [15] J.A. Richards, X. Jia, "Remote Sensing Digital Image Analysis", pp. 283-288, 318-319, Springer, 1999.
- [16] Ifarraguerri et. al, "Unsupervised Hyperspectral Image Analysis with Projection Pursuit", IEEE Transactions on Geoscience and Remote Sensing, vol 38, no. 6, pp 2529-2538, 2000.
- [17] García de Jalón Javier, Rodríguez José Ignacio, Mingo Iñigo, Imaz Aitor, Brazales Alfonso, Larzabal Alberto, Calleja Jesús, García Jon, "Aprenda Java", Escuela Superior de Ingenieros Industriales, Universidad de Navarra. 1999.
- [18] Tanenbaum Andrew, "Computers Networks", Third edition Pearson. 1997.

Apéndice A. Imágenes usadas en las pruebas

Imágen 1. Indiana.mat

Esta es una imagen en formato Matlab, la imagen original fue tomada con el satellite Aviris que posee un sensor de 220 bandas; la imagen usada en estas pruebas fue reducida a 20 bandas y su tamaño es de 20x21025 (bandas por píxeles).

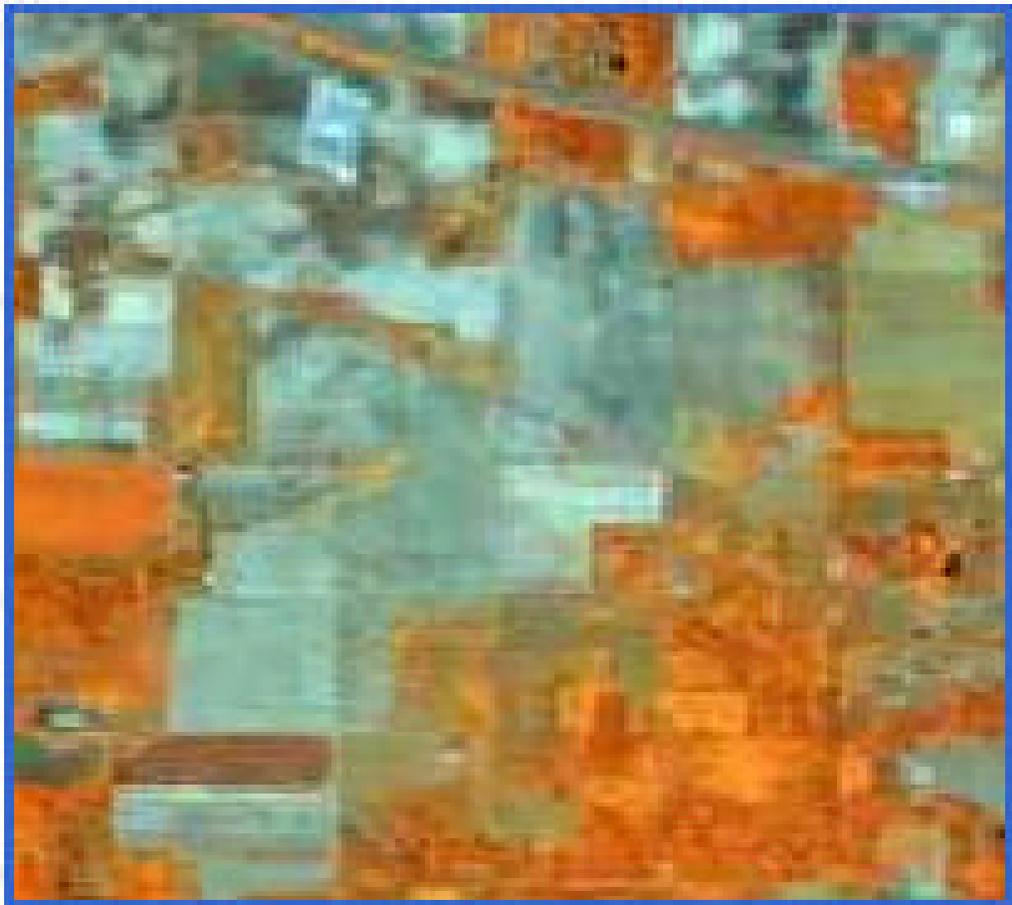


Figura 4: Imagen de Indiana en formato Matlab

Imágen 2. Anasco.mat

Imágen en formato Matlab, fue tomada por el satélite Lansat que posee un sensor de 7 bandas. El Tamaño de esta imágen es de 7x231085 (bandas por píxeles).

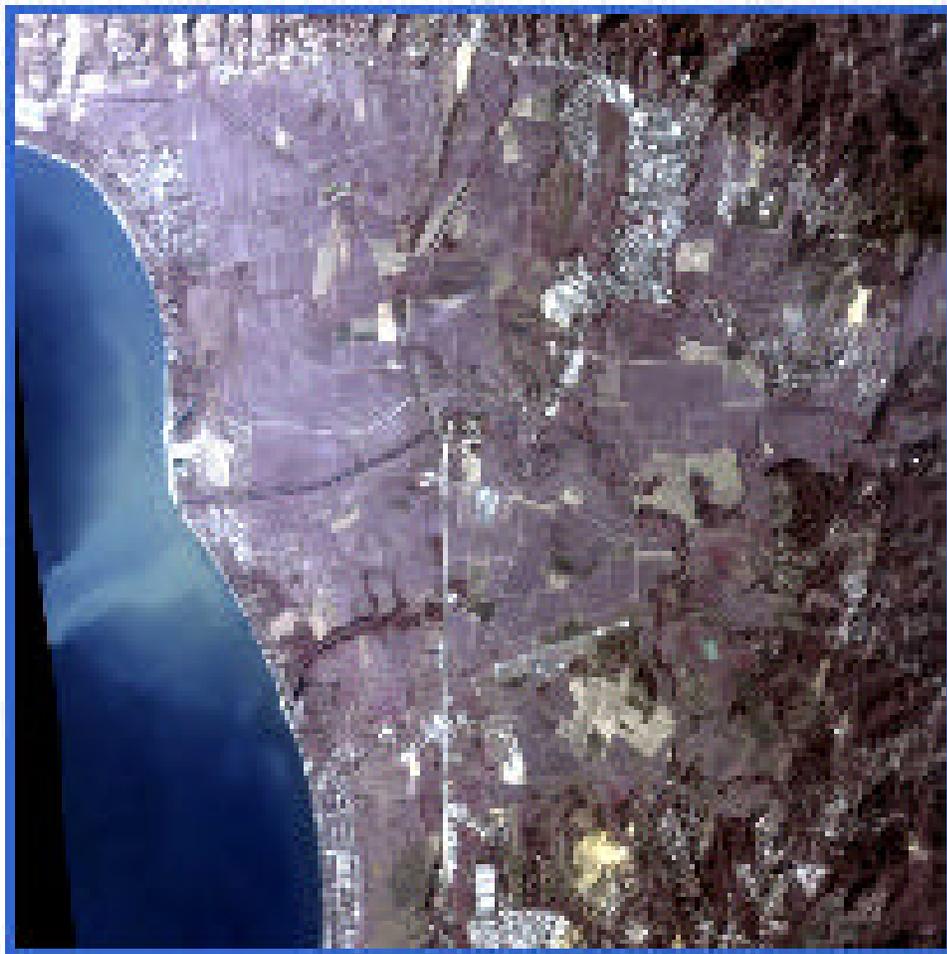


Figura 5: Imágen de Anasco-Puerto Rico en formato Matlab

Imágen 6. Kennedy.mat

Imágen en formato Matlab, fue tomada por el satélite AVIRIS con un sensor de 204 bandas. Su Tamaño es de 204x106396 (bandas by píxeles).



Figura 6: Imágen de Kennedy en formato Matlab

Apéndice B. Tiempos de procesos de imágenes usadas en las pruebas

Tabla B-1. Tiempo de procesamiento de Indiana con 2 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	13.01	0.47	0.21	5.67	7.10	0.22	13.67
FML	12.75	0.44	0.17	5.09	6.88	0.19	12.77
ED	12.66	0.44	0.21	5.18	6.67	0.18	12.68
FED	12.63	0.45	0.19	5.12	6.88	0.19	12.83
FLD	12.89	0.45	0.13	5.18	6.98	0.20	12.94
MD	12.74	0.46	0.20	9.97	6.96	0.17	17.76
AD	12.53	0.46	0.20	5.10	6.66	0.18	12.60

Tabla B-2. Tiempo de procesamiento de Indiana con 4 nodos

Clasif.	4 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	13.51	0.79	0.22	5.15	7.15	0.24	13.55
FML	13.15	0.78	0.24	5.13	6.85	0.21	13.21
ED	12.78	0.80	0.21	5.10	6.50	0.19	12.80
FED	12.76	0.80	0.18	5.07	6.61	0.20	12.86
FLD	12.95	0.87	0.21	5.17	6.49	0.21	12.95
MD	13.07	0.80	0.21	5.12	6.80	0.22	13.15
AD	12.55	0.79	0.19	5.02	6.55	0.19	12.74

Tabla B-3. Tiempo de procesamiento de Indiana con 8 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	13.73	1.38	0.18	5.17	6.85	0.23	13.81
FML	13.97	1.35	0.32	4.99	6.98	0.28	13.92
ED	14.84	3.28	0.20	4.83	6.89	0.26	15.46
FED	14.60	1.34	0.18	5.99	6.93	0.29	14.73
FLD	13.50	1.33	0.22	4.84	6.97	0.28	13.64
MD	13.55	1.33	0.20	4.82	6.93	0.26	13.54
AD	13.90	1.38	0.32	4.89	7.04	0.25	13.88

Tabla B-4. Tiempo de procesamiento de Anasco con 2 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	19.36	0.89	0.14	6.89	9.25	2.13	19.30
FML	25.00	0.86	0.19	12.77	10.73	0.46	25.01
ED	15.79	0.87	0.13	6.46	7.83	0.46	15.75
FED	15.81	0.85	0.24	5.98	8.32	0.42	15.81
FLD	18.53	1.08	0.21	7.95	8.96	0.39	18.59
MD	17.18	0.85	0.18	6.54	9.20	0.39	17.16
AD	17.49	0.85	0.17	6.64	9.42	0.39	17.47

Tabla B-5. Tiempo de procesamiento de Anasco con 4 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	15.83	1.47	0.23	6.19	7.76	0.35	16.00
FML	16.74	1.60	0.22	6.20	8.92	0.37	17.31
ED	15.16	1.51	0.22	6.03	7.11	0.35	15.22
FED	15.21	1.49	0.21	5.99	7.34	0.39	15.42
FLD	16.70	1.48	0.18	6.95	7.76	0.37	16.74
MD	16.22	1.60	0.23	6.55	7.61	0.38	16.37
AD	16.44	1.51	0.21	6.07	7.94	0.38	16.11

Tabla B-6. Tiempo de procesamiento de Anasco con 8 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	16.74	2.54	0.23	5.94	7.46	0.63	16.80
FML	17.23	2.54	0.20	6.22	7.80	0.45	17.21
ED	16.03	2.56	0.21	5.74	7.17	0.45	16.13
FED	16.67	2.58	0.23	6.30	7.16	0.46	16.73
FLD	16.48	2.61	0.21	5.86	7.43	0.43	16.54
MD	16.43	2.59	0.22	5.98	7.31	0.40	16.50
AD	16.25	2.58	0.21	5.59	7.42	0.41	16.21

Tabla B-7. Tiempo de procesamiento de Kennedy con 2 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	106.02	28.48	0.22	56.96	15.25	4.91	105.82
FML	76.86	26.69	0.26	28.39	15.28	0.29	70.91
ED	67.96	26.44	0.20	30.11	16.80	0.48	74.03
FED	54.47	26.52	0.19	16.00	11.94	0.31	54.96
FLD	59.45	26.65	0.23	16.98	15.27	0.29	59.42
MD	57.07	26.37	0.24	14.94	15.22	0.35	57.12
AD	55.16	26.47	0.19	15.40	12.80	0.30	55.16

Tabla B-8. Tiempo de procesamiento de Kennedy con 4 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	75.84	48.09	0.20	15.70	11.66	0.34	75.99
FML	72.24	48.15	0.26	12.64	11.57	0.34	72.96
ED	70.04	48.09	0.20	12.78	9.11	0.31	70.49
FED	69.92	48.26	0.17	11.33	9.97	0.32	70.05
FLD	72.37	48.05	0.19	12.28	11.54	0.41	72.47
MD	73.25	48.08	0.23	11.59	13.31	0.33	73.54
AD	73.53	48.26	0.23	14.77	9.85	0.49	73.60

Tabla B-9. Tiempo de procesamiento de Kennedy con 8 nodos

Clasif.	2 Nodos	Enviar	Devolver	Partir	Procesar	Unir	Total
ML	114.20	81.53	0.22	18.01	22.16	0.47	122.39
FML	107.61	80.27	0.23	17.48	9.25	0.46	107.69
ED	106.08	80.47	0.23	17.05	7.92	0.35	106.02
FED	104.02	80.16	0.24	15.39	7.93	0.35	104.07
FLD	109.49	80.24	0.21	19.41	9.17	0.37	109.40
MD	105.23	80.47	0.23	15.42	8.91	0.37	105.40
AD	104.66	80.74	0.21	15.00	8.41	0.33	104.69