

DIPORSI: Un Entorno Software Distribuido para el Tratamiento de Imágenes de Teledetección Aplicado a Sistemas de Información Geográfica

José A. Gallud*, Jesús García-Consuegra* y José M. García**

*Departamento de Informática
Universidad de Castilla-La Mancha
Campus Universitario s/n, 02071 Albacete (España)

**Departamento de Ingeniería y Tecnología de Computadores
Facultad de Informática de la Universidad de Murcia
Campus de Espinardo, 30080 Murcia (España)

Resumen. Este artículo presenta un entorno software distribuido llamado DIPORSI (Distributed Processing of Remotely Sensed Imagery) orientado a la aceleración del procesamiento de imágenes de teledetección aplicados a Sistemas de Información Geográfica (SIG). En particular, se ha estudiado a fondo el algoritmo de rectificación geométrica sobre imágenes captadas por el sensor TM a bordo del satélite Landsat.

1. Introducción

Un Sistema de Información Geográfica (SIG) almacena información sobre el mundo como una colección de niveles temáticos que pueden relacionarse espacialmente, de acuerdo a un sistema de coordenadas. Esta característica ha demostrado ser de gran utilidad para resolver problemas de toda índole, desde la optimización de las rutas de los transportes públicos, o la realización de planes de ordenación urbanos, hasta los análisis de impacto medioambiental [6].

Las imágenes procedentes de sensores remotos (como satélites) se utilizan en los SIG para llevar a cabo toda clase de estudios medioambientales, urbanísticos, etc. La realización de dichos estudios se apoya en un conjunto de operadores que permiten analizar las imágenes extrayendo información de forma automática. Ejemplos de tales operadores son los diferentes filtros, realzado de bordes, transformaciones geométricas, etc. Debido al tamaño de las imágenes, al elevado número de escenas (datos multitemporales y multiespectrales) de una misma área y a la cantidad de algoritmos de procesamiento de imágenes que se utilizan, el coste computacional que requieren este tipo de sistemas es significativo [7][1].

Las operaciones que más tiempo de cómputo necesitan son la rectificación y la clasificación. Mediante el proceso de rectificación se intentan eliminar los errores radiométricos y geométricos que acompañan a una imagen captada mediante sensores remotos. Los algoritmos de clasificación permiten extraer información temática a partir de las imágenes (como los mapas de usos del suelo) [5][2].

El sistema que estamos interesados en diseñar debe contemplar toda clase de operaciones sobre imágenes que el usuario introduce utilizando un lenguaje interpretado tipo script o archivo de comandos. Desde el punto de vista de la plataforma hardware y software el sistema deberá contemplar un entorno heterogéneo en el que conviven estaciones de trabajo, grandes computadores y ordenadores personales.

Antes de abordar el problema del diseño del sistema distribuido basado en un lenguaje de archivo de comandos para procesar imágenes de teledetección, nos hemos querido centrar en el estudio del coste de paralelización de una de las operaciones como es la rectificación geométrica. Con este estudio queremos enfrentarnos al diseño del sistema global con la experiencia adquirida en la implementación de uno de los casos particulares.

Un modo de acometer las exigencias en cuanto a coste computacional consiste en recurrir a *cluster* de estaciones de trabajo, aprovechando la ventaja adicional de no tener que realizar una inversión económica elevada en equipos. El vertiginoso aumento de las prestaciones de las estaciones de trabajo, la implantación masiva de redes de computadores, junto con la progresiva reducción del coste económico de

adquisición de este equipamiento, está posibilitando el crecimiento en dotación de un gran número de laboratorios, tradicionalmente con pocos recursos, que ofrecen una potencia de cálculo antes inalcanzable en muchos sentidos [3] [4].

El resto del artículo se organiza en los siguientes apartados. En la sección 2 se comentan brevemente los trabajos de investigación más destacados de entre los relacionados con el tema objeto de estudio. En la sección 3 se realiza una descripción de todo el proceso de rectificación geométrica aplicado a imágenes de teledetección. La sección 4 presenta el estudio del algoritmo secuencial. En la sección 5 podemos encontrar una introducción a DIPORSI. La sección 6 presenta los escenarios de prueba, así como los resultados obtenidos al ejecutar diferentes implementaciones sobre una serie de plataformas. Por último, en la sección 7 se presentan las conclusiones y se plantean las posibles líneas de trabajo futuro.

2. El paralelismo en teledetección y SIG

La rectificación es una de las operaciones que tiene como objetivo corregir los errores, radiométricos o geométricos, que se producen en el momento de captar la imagen (debidos a la curvatura de la tierra, a la órbita del satélite, y a la influencia de la atmósfera en la imagen resultante), y que permiten que todos los píxeles de la imagen tengan una coordenada espacial y un valor numérico que representa la respuesta en una frecuencia determinada.

El proceso de rectificación depende de las características del sensor. En este artículo nos hemos centrado en el sensor Thematic Mapper a bordo del satélite Landsat.

El trabajo más directamente relacionado con el que proponemos aparece en [5]. En el contexto más general de la clasificación de imágenes Landsat, aparece la referencia obligada al proceso de corrección geométrica de las imágenes previa a los algoritmos de clasificación. Lo más destacable del proyecto que se resume en el citado artículo es la manera de tratar la rectificación de las imágenes Landsat, lo que constituye nuestro objetivo central, diciendo que tiene un bajo coste computacional. Por lo demás el programa se ofrece en dos plataformas: Unix o Windows.

En [7] se expone un trabajo de paralelizar un algoritmo de segmentación de imágenes Landsat mediante un *cluster* de estaciones de trabajo y un Cray T3E, para comparar resultados. Otro trabajo interesante relacionado con segmentación de imágenes Landsat (crecimiento de regiones) y que también lleva a cabo una comparación de resultados de ejecución en distintas máquinas es el presentado en [1].

Una de las descripciones más acertadas de lo que se busca realizar con este artículo lo podemos encontrar en [8]. En él se justifica la conveniencia de utilizar computación paralela para el procesamiento de imágenes Landsat en un contexto de monitorizar ecosistemas.

3. La rectificación de imágenes Landsat

El proceso de corrección de imágenes de teledetección trata de eliminar cualquier anomalía detectada, bien sea en su localización, bien sea en los niveles digitales de los píxeles que la componen. El objetivo de la corrección es disponer los datos de la forma más cercana posible a una adquisición idónea, tanto en la posición como en la radiancia.

Las correcciones radiométricas son un conjunto de técnicas que tratan de modificar los niveles digitales de la imagen para acercarlos a los que tendrían de haber realizado una recepción ideal.

En particular, algunas de las tareas para realizar la corrección radiométrica de la imagen incluyen:

- Restauración de líneas o píxeles perdidos: aparecen líneas blancas o negras o una serie de píxeles aislados de aspecto muy contrastado (efecto sal y pimienta). El único modo de restaurar esos píxeles consiste en utilizar la información de los píxeles vecinos, debido al fenómeno de la autocorrelación espacial.

- Corrección del bandeo de la imagen: este problema aparece cuando algunos detectores codifican la información de modo diferente al resto, apareciendo un bandeo de la imagen.
- Corrección del efecto atmosférico y de las variaciones de iluminación: se trata de obtener unos nuevos valores de los niveles digitales de la imagen que tenga en cuenta las condiciones atmosféricas, con la idea de aumentar la precisión de los estudios geográficos a partir de la imagen.

Las correcciones que nos interesan en este artículo son las denominadas correcciones geométricas, que incluyen cualquier cambio en la posición que ocupan los píxeles que forman una imagen.

El hecho de trabajar con datos digitales hace especialmente flexible la modificación de la geometría de la imagen. La expresión general de este tipo de funciones es:

$$\begin{aligned} f(c') &= f_1(c, l) \\ f(l') &= f_2(c, l) \end{aligned} \tag{1}$$

Las coordenadas columna y línea (c' y l') de la imagen corregida son función de las coordenadas columna y línea de la imagen de entrada o del mapa al cual se pretende superponer la imagen. Por tanto, esta transformación puede emplearse tanto para corregir cartográficamente una imagen, como para superponer dos o más imágenes entre sí.

Con la transformación geométrica se pretende encontrar una relación que transfiera los niveles digitales de la imagen a su posición cartográfica, de acuerdo a una proyección escogida, normalmente UTM (Universal Transversal de Mercator). De este modo, la imagen resultante puede combinarse con otra clase de información con referencia espacial o emplearse en Sistemas de Información Geográfica.

La transformación geométrica que utiliza una imagen de referencia se conoce con el nombre de reSIGtrado de imágenes y tiene como objeto realizar estudios multitemporales (como puede ser el estudio de la evolución de la superficie regada en una zona determinada).

La corrección geométrica puede abordarse siguiendo alguno de estos dos métodos:

1. Corrección orbital: este método trata de aplicar las transformaciones inversas a las que realiza el sensor en el momento de captar la imagen. Para ello se necesita disponer de las características orbitales de la plataforma y de las especificaciones del sensor. De este modo se pueden corregir errores sistemáticos, como los derivados de la rotación o curvatura terrestre y de la inclinación de la órbita.
2. Puntos de control: En este caso se trata de modelar el error geométrico de la imagen a partir de una serie de puntos con coordenadas conocidas denominados puntos de control. El error se modela inductivamente, ya que en las funciones de transformación se incluyen todas las fuentes de error, suponiendo que los puntos son suficientemente representativos de la deformación geométrica de la imagen.

El método basado en modelos orbitales suele utilizarse cuando las imágenes no presentan referencias visuales que imposibilitan la obtención de puntos de control, como las zonas marinas, grandes coberturas de nubes o en aquellas imágenes captadas por sensores con una resolución espacial baja (como AVHRR o Meteosat). La principal ventaja de utilizar estos métodos consiste en que el proceso puede realizarse de forma automática.

El método basado en puntos de control es mucho más tedioso, requiere gran intervención humana para localizar adecuadamente los puntos de control, pero ofrece una alta precisión cuando se trabaja sobre imágenes donde es fácil identificar elementos comunes a la cartografía básica.

Inicialmente, en este artículo nos hemos centrado en la corrección geométrica de imágenes Landsat TM en el contexto de SIG medioambientales, por lo que no profundizaremos más en el método de corrección basado en modelos orbitales, para dedicarnos en exclusiva a los métodos de corrección geométrica basados en puntos de control.

La corrección basada en puntos de control asume que no se conoce la fuente de los errores pero que éstos pueden modelarse a partir de ajustar unas ecuaciones a un conjunto de puntos, cuya localización es

conocida tanto en la imagen o mapa de referencia como en la imagen a corregir. Normalmente la corrección a partir de puntos de control se realiza en tres fases:

1. Localización de los puntos de control: puntos comunes a la imagen y al mapa.
2. Obtención de las funciones de transformación entre las coordenadas de la imagen y las del mapa.
3. Transferencia de los niveles digitales desde la imagen original sin corregir a la imagen rectificadora.

4. Estudio del algoritmo secuencial de rectificación geométrica

Con la descripción del proceso de rectificación basado en puntos de control se cierra la fase más teórica en lo referente a los procesos de teledetección. A continuación vamos a centrarnos en describir algunos detalles de la implementación del algoritmo secuencial.

El algoritmo secuencial que hemos implementado recorre las dimensiones de una imagen vacía, calculando para cada píxel la coordenada UTM que le correspondería en la imagen original, gracias a las ecuaciones de transformación, y obteniendo el nivel digital a partir de la imagen original con ayuda de los métodos de interpolación.

Se ha optado por implementar esta solución por ser mucho más sencilla que la descrita inicialmente en la sección anterior, que consistía en recorrer la imagen original calculando para cada píxel la coordenada UTM en la imagen corregida, asignando después el nivel digital mediante uno de los métodos conocidos.

Volvamos a la cuestión del análisis de la eficiencia del algoritmo. Tenemos que fijarnos tanto en la complejidad espacial como temporal. En cuanto al coste espacial tenemos que una escena completa Landsat-TM (*Thematic Mapper*) requiere aproximadamente 37MB. Habitualmente, en las aplicaciones similares, se permite trabajar con varias bandas simultáneamente. El sensor TM a bordo del satélite Landsat ofrece hasta 7 bandas. Esto significa que el coste espacial de la rectificación requeriría 7x37MB sólo para abrir las imágenes originales. Sin embargo, la rectificación genera una imagen totalmente nueva y normalmente de dimensiones mayores. Así pues, para el cálculo del coste espacial hay que considerar hasta un máximo de 7x50MB, lo que supone un total de 609MB en el peor de los casos.

En cuanto al coste temporal, es bastante inmediato deducir que el algoritmo arriba expuesto tiene una complejidad temporal que es del $O(n^2)$, siendo n la dimensión de la imagen. Hay que tener en cuenta que en el peor de los casos, el método de interpolación utilizado exigiría acceder a los 16 píxeles más cercanos para obtener el nivel digital de cada uno de los píxeles de la imagen corregida.

5. Introducción a DIPORSI (Distributed Processing of Remotely Sensed Image)

La versión distribuida es bastante similar a la versión secuencial ya que vamos a realizar un paralelismo de datos. Las imágenes corregidas se van a obtener como consecuencia de la ejecución del mismo programa en todos los nodos, trabajando sobre zonas diferentes de la imagen final.

Lo único que necesita conocer cada nodo es el número de filas de la imagen resultante que debe procesar. Recordemos que tal y como está definida la implementación del proceso de rectificación, cada nodo recorre su parte de la imagen resultante, cuyas dimensiones se obtienen mediante las ecuaciones de transformación, y se encarga de asignar el nivel digital adecuado a cada píxel en función del método de interpolación que se utilice.

Al utilizar un paralelismo de datos bastante inmediato simplificamos al máximo el diseño de las versiones distribuidas del algoritmo de rectificación. El escenario distribuido se compone de un proceso maestro o raíz que lanza tantos procesos distribuidos como se desee y que distribuye la carga uniformemente entre ellos.

Esta simplicidad teórica del sistema distribuido nos llevó a desarrollar un entorno de ejecución de toda clase de algoritmos de procesamiento de imágenes teledetectadas, para no limitarnos solamente a la rectificación. Tal entorno de ejecución se denominó DIPORSI que consiste funcionalmente en una capa implementada a partir de funciones MPI (*Message Passing Interface*) que define un conjunto de funciones

distribuidas (operaciones sobre imágenes) y un conjunto de funciones de control de la operación. DIPORSI es controlado por un proceso principal mientras que el resto de procesos distribuidos permanecen a la espera de recibir una orden de procesamiento sobre unos datos determinados.

DIPORSI ha demostrado ser bastante eficaz funcionalmente y ha sido desarrollado inicialmente para plataformas distribuidas basadas en win32 y portado posteriormente a Unix/Linux. Como podremos observar en los apartados siguientes, DIPORSI ha conseguido buenos resultados en la reducción del tiempo de cómputo.

Una primera observación debe realizarse sobre la cuestión del acceso a las imágenes. El formato que hemos decidido utilizar tiene la ventaja de ser sencillo de utilizar y el inconveniente de que no emplea ningún factor de compresión. Como ya hemos explicado en los apartados precedentes, se escogió el formato de datos Ilwis por ser uno de los más extendidos, de modo que cualquier otro formato de imagen puede convertirse a formatos de Ilwis.

Como ocurre en otras aplicaciones que trabajan con imágenes o con secuencias de video, una de las primeras cuestiones a resolver consiste en determinar el modo de acceder a las imágenes. En el algoritmo distribuido tenemos dos posibles alternativas:

1. Repartir todos los datos de entrada entre todos los nodos.
2. Acceder a los datos mediante NFS o como recurso compartido.

Las primeras versiones de DIPORSI contaban con el reparto de los datos entre todos los nodos del sistema distribuido, aprovechando las características de las funciones de difusión de MPI. Sin embargo, el tiempo de respuesta de la aplicación no podía competir con los tiempos de las versiones secuenciales debido al coste que conlleva la distribución de los datos. No hay que olvidar que, en el escenario que hemos contemplado, estamos trabajando con tres ficheros de 40MB con las imágenes sin corregir y otros tres ficheros de 50MB que contienen las imágenes corregidas.

Otro inconveniente del reparto de los datos es la reconstrucción de la imagen resultante (o imagen corregida) al finalizar el proceso. Al distribuir los datos entre todos los nodos, cada proceso trabajando en un nodo diferente genera una parte de la imagen final. El proceso principal debe encargarse de reconstruir la imagen completa a partir de los resultados que cada nodo va enviando.

Ambos argumentos nos hicieron decantarnos por implementar el sistema distribuido contando con las capacidades y funciones distribuidas de los sistemas operativos que hemos utilizado. Nos estamos refiriendo al servicio NFS de sistemas Unix/Linux y a las funciones de acceso remoto de las plataformas win32. De este modo, cada proceso trabajará en paralelo con los demás y accederá a los datos como si fueran locales, dejando la tarea de adquirir las imágenes, a medida que se vayan necesitando, al sistema operativo.

Un leve inconveniente de utilizar el tiempo de respuesta es que sufrimos pequeñas variaciones al realizar diferentes ejecuciones, variaciones que consideramos despreciables en relación al resultado total. Como es natural esta variabilidad del tiempo de respuesta, consecuencia de ejecutar los procesos en sistemas multitarea, no afecta al tiempo real consumido por el proceso que es siempre el mismo. Esto último se aplica indistintamente a ambas plataformas software, Linux y Win32.

6. Escenario de pruebas y resultados

Una vez realizadas todas las observaciones, vamos a presentar los resultados organizados en plataformas diferentes. Para lo cual comenzaremos describiendo las características de cada *cluster*, aunque en este trabajo se recogen los resultados obtenidos en la última de las plataformas:

1. *Cluster* de 8 Pentium 166, 32 MB con ethernet, Windows NT 4.0 y WinMPI
2. *Cluster* de 12 Pentium 166, 32MB con ethernet, Linux con MPICH
3. *Cluster* de 8 Pentium II, 64 MB con fast ethernet, Linux con MPICH
4. *Cluster* de 4 Pentium II Dual, 64 MB con fast ethernet, Linux con MPICH

Anteriormente hemos mencionado que la versión de Linux y la Windows NT difieren no sólo en la plataforma hardware y software utilizada, radicalmente diferente, sino que el esquema funcional de DIPORSI ha cambiado de forma considerable. Por eso no es justo establecer una comparación entre los resultados obtenidos en el *cluster* de biprocesadores con Linux y los obtenidos en el *cluster* de Pentium 166 con Windows NT 4.0. Por lo demás, la mejora introducida ya fue descrita anteriormente y consiste en no distribuir los datos dentro de la propia aplicación sino que esa operación es realizada por el sistema operativo (mediante NFS o NTFS).

Una de las comparativas que es obligado mostrar en una misma gráfica es la que representa los tiempos de respuesta del algoritmo secuencial en relación con los diferentes tiempos de respuesta de la versión distribuida. En la Figura 1 podemos observar estos resultados.

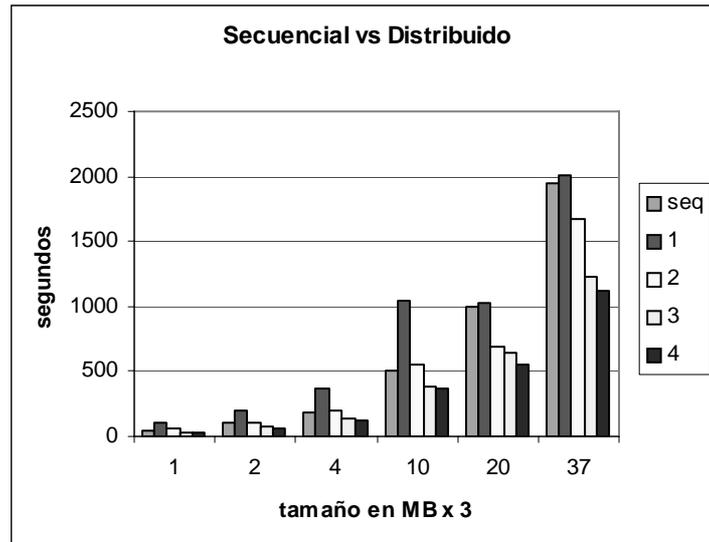


Fig. 1. Comparativa entre las versiones secuencial y distribuida (para 1, 2, 3 y 4 nodos con Pentium II Dual y NFS sobre fast ethernet con Linux RedHat) al ejecutar la rectificación geométrica utilizando convolución cúbica.

Antes se ha comentado la reducción temporal que se logra al ejecutar la versión distribuida, tal y como se puede observar en la Figura 1. Sin embargo, si representamos el *speedup* (Figura 2) observaremos que, aun habiendo obtenido un recorte del tiempo de respuesta de casi 15 minutos con la versión distribuida de 4 nodos, el recorte teórico debería haber sido mucho mayor y aproximadamente igual a 25 minutos.

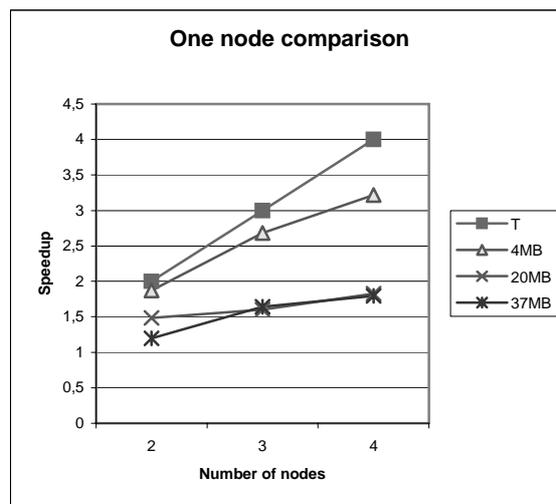


Fig. 2. *Speedup* para los datos representados en la Figura 1.

7. Conclusiones y trabajo futuro

Este artículo presenta el estudio de la aplicación del paradigma de la computación paralela y distribuida en la aceleración de los procesos que intervienen en los sistemas de teledetección. En particular, se ha estudiado a fondo el algoritmo de rectificación geométrica que se aplica a imágenes captadas por el sensor TM a bordo del satélite Landsat.

Posteriormente hemos presentado el diseño del sistema distribuido, con las diferentes variantes en función de la plataforma (DIPORSI para win32 y Linux). Finalmente, los resultados han puesto de manifiesto las ventajas de la computación distribuida frente al coste de implantación de sistemas distribuidos, en relación a la mejora de las prestaciones de la rectificación geométrica.

La principal conclusión de este artículo es que nos parece, a pesar de la reducción del tiempo de respuesta obtenido, desproporcionada la inversión en sistemas distribuidos para la resolución de un problema relativamente sencillo como el de la rectificación geométrica. Esto es mucho más evidente si se toma el método del vecino más próximo como el método más adecuado para realizar la georreferenciación.

Sin embargo, el futuro de las aplicaciones pasa por la interoperabilidad y los sistemas distribuidos. En este sentido, este artículo ofrece una interesante aportación al diseño de sistemas distribuidos, al utilizar plataformas heterogéneas y tecnologías de sistemas de archivos distribuidos actualmente en uso. El futuro de los sistemas de teledetección es la proliferación de sistemas distribuidos de tiempo real para la prevención y actuación ante catástrofes naturales, por un lado, y la elaboración de sistemas que ofrezcan servicios de procesamiento de imágenes de teledetección aprovechando las posibilidades de Internet.

Referencias

- [1] D. A. Bader. Parallel Algorithms for Image Enhancement and Segmentation by Region Growing with an Experimental Study. In *The Journal of Supercomputing*. Vol 10, No 2, pages 141-168, 1996.
- [2] H. Fallah-Adl et al. Fast Algorithms for Removing Atmospheric Effects from Satellite Images. In *IEEE Computational Science and Engineering*, Vol. 3, No. 2, Summer 1996.
- [3] J. A. Gallud, J. D. García-Consuegra, G. Sebastián. Distributed Georeferencing of Remotely Sensed LandSat-TM Imagery Using MPI. In *Para'98*, LNCS nº 1541, Springer-Verlag (1998), pages 161-167.
- [4] J. A. Gallud, J. M. García y J. García-Consuegra. Cluster Computing Using MPI and Windows NT to Solve the Processing of Remotely Sensed Imagery. Recent Advances in Parallel Virtual Machine and Message Passing Interface. Lecture Notes in Computer Science (1697), Springer-Verlag, pp. 442-449, 1999.
- [5] J. Janoth et al. MMIPPS – A Software Package for Multitemporal and Multispectral Image Processing on Parallel Systems. In *ACPC '99*. LNCS 1557, pp. 398-407, 1999.
- [6] B. Schmidt-Belz et al. GeoMed for Urban Planning – First User Experiences. In *ACM Workshop on Advances in SIG '98*, pages 82-87, Washington, November 1998.
- [7] J.C. Tilton. A Recursive PVM Implementation of an Image Segmentation Algorithm with Performance Results Comparing the HIVE and the Cray T3E. In *Proceedings of the IEEE Symposium of Massively Parallel Processing*, 1998.
- [8] C. J. Turner and G. J. Turner. Adaptive Data Parallel Methods for Ecosystem Monitoring. In *Proceedings of the ACM Conference on Supercomputing '94*, pages 281-291, 1994.