



Máster de Tecnologías de Telecomunicación

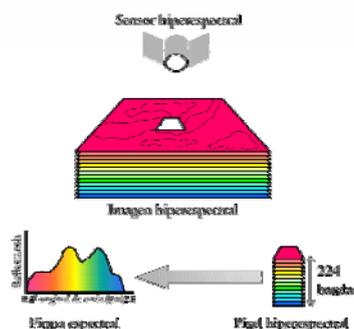
Trabajo Fin de Máster

Metodología de alto nivel para implementar el algoritmo *Vertex Component Analysis* en una GPU



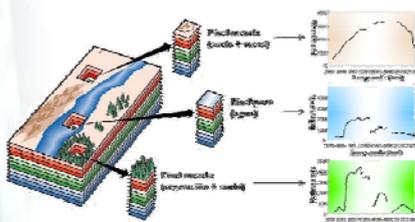
Ricardo Topham González
 Dr. D. Gustavo Marrero Callicó
 Dr. D. Sebastián López Suárez

Julio 2011



Resumen:

- El trabajo que se presenta desarrolla una metodología de alto nivel para implementar el algoritmo *Vertex Component Analysis* en una GPU. La metodología desarrollada permitirá programar y ejecutar el algoritmo en una GPU aprovechando el potencial que ofrece su arquitectura para la computación en paralelo.
- La metodología hará uso de las funciones disponibles en el *Parallel Computing Toolbox* de MATLAB creadas expresamente para la programación en una GPU y también de la interacción que ofrece con NVIDIA CUDA. Por si fuera poco, la misma podrá ser extrapolada a otros algoritmos de análisis hiperespectral.
- Hace unos años no era posible programar en GPUs desde MATLAB; ahora gracias a las nuevas funciones disponibles en el *Parallel Computing Toolbox* sí se puede, ofreciendo una serie de funcionalidades bastante interesantes. De esta forma se realizará un enfoque novedoso que permitirá sacar el máximo partido posible a estos dispositivos desde el entorno MATLAB.
- Palabras clave: *endmember*, extracción, hiperespectral, GPU, VCA



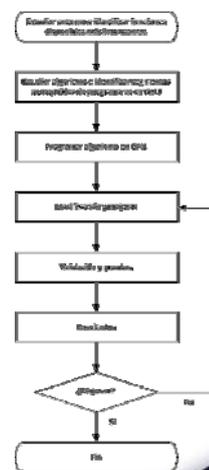
La mayor parte de los píxeles de una imagen hiperespectral corresponden a una mezcla de componentes puros a nivel macroscópico. Estos píxeles mezcla pueden expresarse como una combinación de elementos espectralmente puros, denominados *endmembers*. el modelo lineal de mezcla expresa los píxeles mezcla como una combinación lineal de firmas asociadas a componentes espectralmente puros en la imagen. Debido a ello, los algoritmos de extracción de *endmembers* son de vital importancia para el análisis de imágenes hiperespectrales.

La imagen de la izquierda muestra de forma gráfica el concepto anterior.

El algoritmo *Vertex Component Analysis* encuentra los *endmembers* de la imagen a partir de un procedimiento que consiste en proyectar la imagen un total de "p" veces (número de *endmembers* que se quieren calcular) sobre un vector perpendicular al subespacio formado por los *endmembers* ya determinados, requiriéndose un total de "p" iteraciones.

La metodología desarrollada permite ejecutar el algoritmo en una GPU haciendo uso del *Parallel Computing Toolbox* de MATLAB. De esta forma se facilita la programación de la GPU, evitando pasando por un entorno CUDA, mucho menos amigable que el anterior.

El flujo de diseño se muestra a la derecha de estas líneas.



Conclusiones:

- ✓ Con este trabajo se ha conseguido desarrollar una metodología que ha permitido implementar el algoritmo *Vertex Component Analysis* en una GPU NVIDIA GeForce GTX 480.
- ✓ El *Parallel Computing Toolbox* de MATLAB dispone de varias funciones interesantes para programar en GPUs, pero aún hay muchas que no admite y que impiden sacarle el máximo partido a la arquitectura paralela de una GPU.

