

Hyperspectral Rendering and Compression

Congreso Español de
Informática Gráfica
2022

Alfonso López
Juan Roberto Jiménez
Juan Manuel Jurado
Francisco Ramón Feito

The logo for CEIG 2022 features the acronym "CEIG" in large, bold, yellow 3D letters. The letter "G" is partially obscured by a red, translucent shape resembling a heart or a stylized letter "G". Below the letters is a dark, wavy surface with light reflections, suggesting water or liquid.

Introducción a imágenes hiperespectrales

2

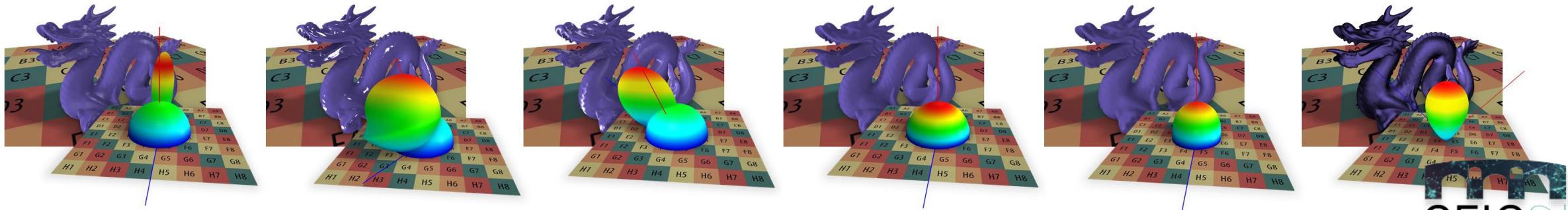
- Relevancia del **modelado de materiales** en el ámbito de Informática Gráfica.
 - *Bidirectional Texture Function* (BTF), *Bidirectional Reflectance Distribution Function* (BRDF), etc.
- **Bases de datos de materiales caracterizados en un amplio intervalo de longitudes de onda.**
- Sensores hiperspectrales para explorar el comportamiento del material más allá del espectro visible.
 - Permiten adquirir datos más allá de unas pocas bandas del espectro visible e infrarrojo cercano.
 - Pueden integrarse con plataformas tan versátiles como drones.
 - Reducción de coste → Mayor acceso a datos hiperespectrales.



Introducción a imágenes hiperespectrales

3

- Relevancia del **modelado de materiales** en el ámbito de Informática Gráfica.
 - *Bidirectional Texture Function* (BTF), *Bidirectional Reflectance Distribution Function* (BRDF), etc.
- Sensores hiperspectrales para explorar el comportamiento del material más allá del espectro visible.
 - **Goniofotómetro.**
 - Mayor coste.
 - Mayor número de restricciones en la configuración del sistema.
 - Variantes para medir materiales anisotrópicos, capturar refracción, normales, o cualquier otra característica que permita modelar sintéticamente la superficie objetivo.



Introducción a imágenes hiperespectrales

4

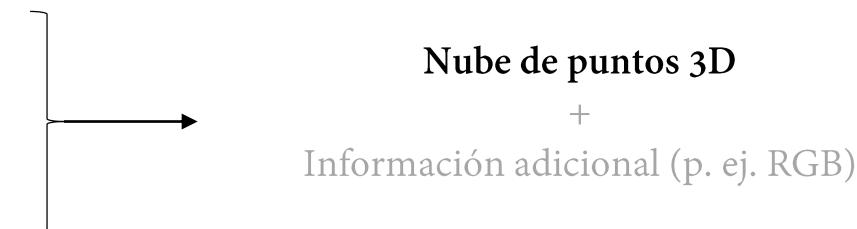
- Relevancia del **modelado de materiales** en el ámbito de Informática Gráfica.
 - *Bidirectional Texture Function* (BTF), *Bidirectional Reflectance Distribution Function* (BRDF), etc.
- Sensores hiperspectrales para explorar el comportamiento del material más allá del espectro visible.
 - Más recientemente, empleados a bordo de **drones**.
 - Mejora de resolución espacial ypectral de sensores a bordo.
 - Utilización de dispositivos basados en imagen:
 - Más versátiles.
 - Caracterización de escenarios reales.
 - **Mayor distorsión geométrica.**
 - En combinación con otros sensores, como RGB.



Generación de escenarios hiperspectrales 3D

5

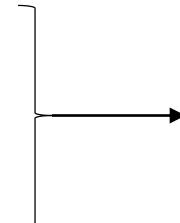
- Nuevas soluciones capaces de integrar sensores LiDAR e hiperspectrales.
- Escaso trabajo previo relacionado con la generación automática de modelos hiperespectrales 3D.
- Aplicaciones del modelado 3D:
 - Asociación de características geométricas/espaciales e información hiperespectral.
 - Modelado de interacciones complejas entre superficies: especularidad, interreflejos, transparencias, *subsurface scattering*, etc.
- Facilitado por el uso de drones.
 - Adquisición de información desde diferentes puntos del espacio.
- Soluciones clásicas:
 - *Radio Detection and Ranging* (RaDAR).
 - *Light Detection and Ranging* (LiDAR).
 - *Structure-from-Motion* (SfM): sensores basados en imagen.



Generación de escenarios hiperspectrales 3D

6

- Nuevas soluciones capaces de integrar sensores LiDAR e hiperspectrales.
- Escaso trabajo previo relacionado con la generación automática de modelos hiperespectrales 3D.
- Aplicaciones del modelado 3D:
 - Asociación de características geométricas/espaciales e información hiperespectral.
 - Modelado de interacciones complejas entre superficies: especularidad, interreflejos, transparencias, *subsurface scattering*, etc.
- Facilitado por el uso de drones.
 - Adquisición de información desde diferentes puntos del espacio.
- Soluciones clásicas:
 - *Radio Detection and Ranging* (RaDAR).
 - *Light Detection and Ranging* (LiDAR).
 - *Structure-from-Motion* (SfM): sensores basados en imagen.

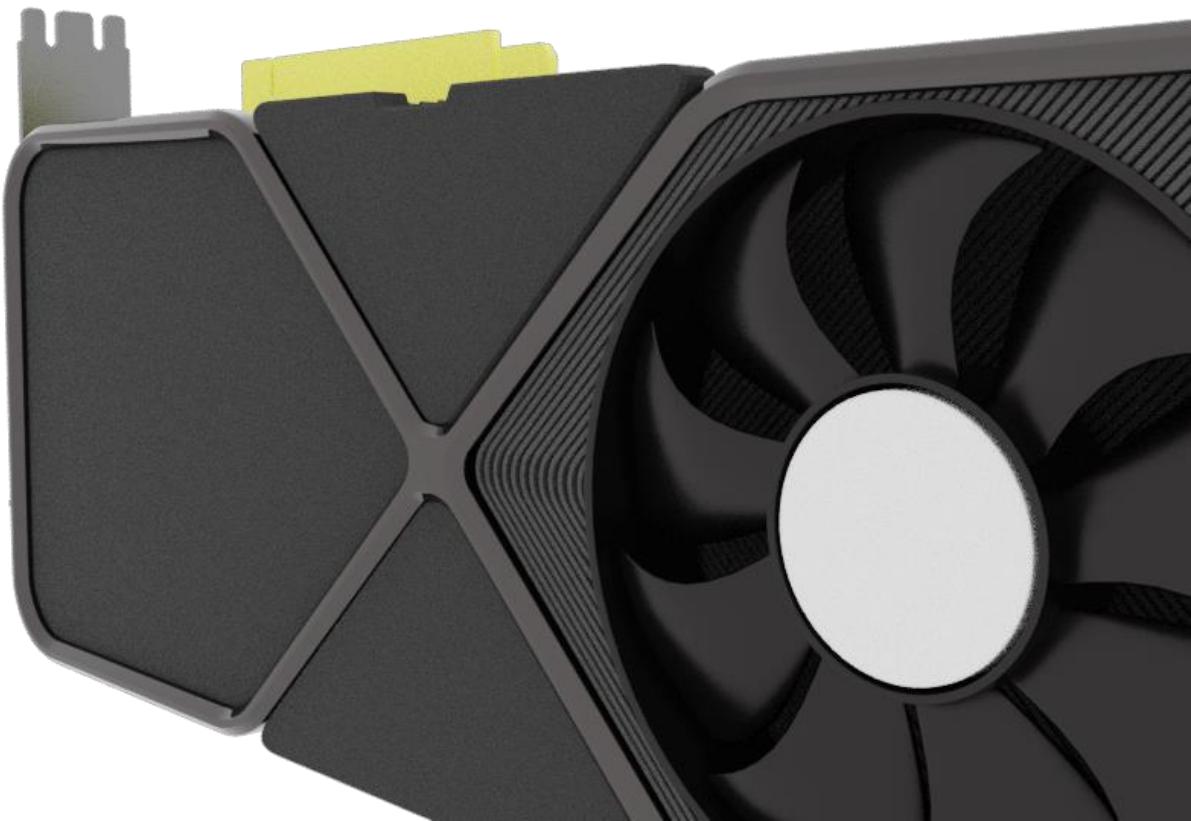


Nube de puntos 3D
Mallas de triángulos

Fusión de datos hiperespectrales

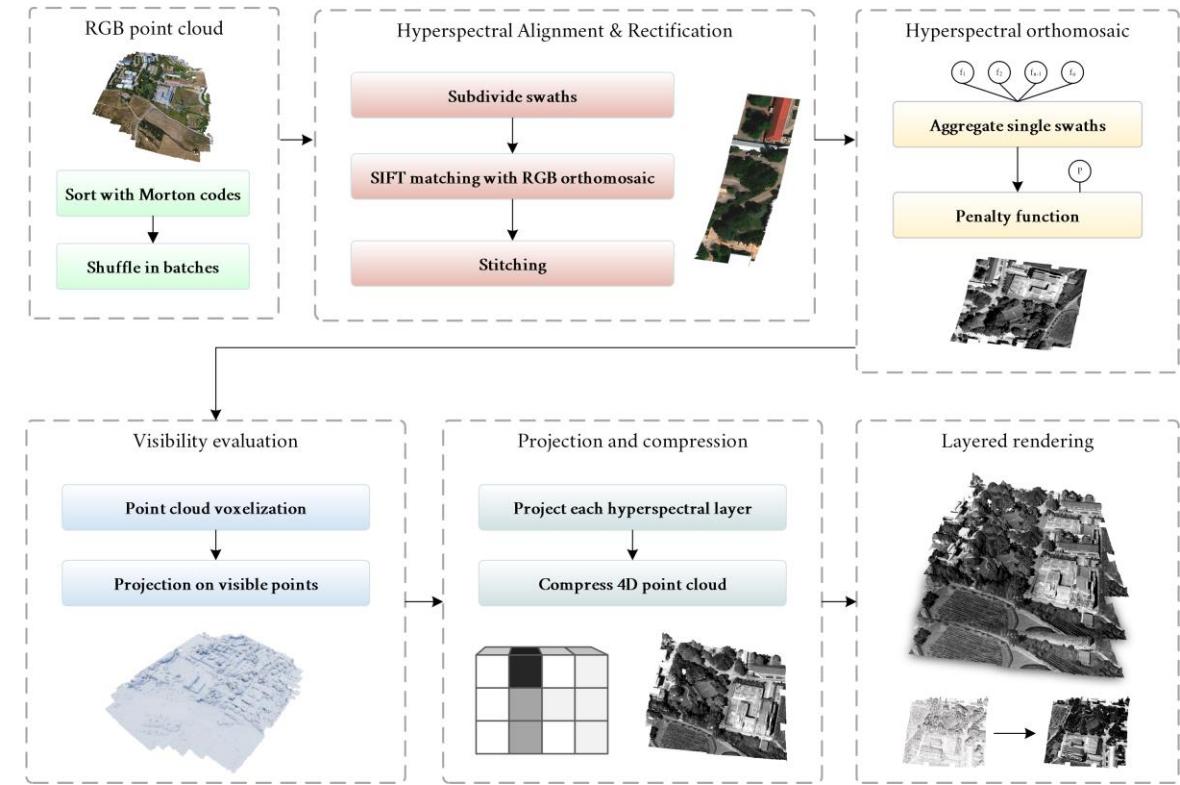
7

- Sólo hiperespectral (single capture).
 - Imágenes independientes para diferentes longitudes de onda.
 - Hiperespectral + RaDAR/LiDAR.
 - Hiperespectral + sensores RGB (SfM).
-
- Además de la fusión de datos, las nubes de puntos suelen presentar un número elevado de puntos.
 - Procesamiento en GPU.
 - *Cloud computing*.



Objetivos de este trabajo

- Generación de una nube 3D hiperespectral.
- Aceleración de *pipeline* de generación de nube.
 - Procesamiento en GPU.
- Mínimo almacenamiento en memoria.
 - Compresión de hipercubo.
 - También debido al procesamiento en GPU.
- Visualización de capas hiperespectrales en tiempo real.
 - *Frame rate* interactivo.
 - Mejora de visualización para observar información hiperespectral.



1

Rectificación y registro de bandas hiperespectrales

- Datos de entrada:
 - Bandas hiperespectrales capturadas por un sensor tipo *push-broom*.
 - Headwall's Nano-Hyperspec.
 - Dron DJI Matrice 600 Pro (M600).
 - Imágenes RGB capturadas utilizando un dron Phantom 4.
 - Adquisición de ortomosaico RGB del área de estudio.
 - Procesamiento mediante Pix4DMapper, utilizando SfM.
 - Marcado de puntos de control (*Ground Control Points*) para georeferenciar con exactitud.

1

Rectificación y registro de bandas hiperespectrales

- Objetivo: obtener un ortomosaico hiperespectral.
- **Deformación geométrica de imágenes hiperespectrales.**
 - Rectificación manual mediante software (p. ej. *SpectralView*).
 - Distintos parámetros de rectificación para cada banda.
 - Procesamiento automático propuesto en un trabajo previo:
 1. **Subdivisión de imágenes hiperespectrales.**
 - Garantiza deformaciones más uniformes en una subdivisión.

1

Rectificación y registro de bandas hiperespectrales

- Objetivo: obtener ortomosaico hiperespectral.
- **Deformación geométrica de imágenes hiperespectrales.**
 - Procesamiento automático propuesto en un trabajo previo:
 1. Subdivisión de imágenes hiperespectrales.
 2. Extracción de características en común entre el ortomosaico RGB y las subdivisiones del hipercubo.
 - *Oriented FAST and Rotated BRIEF* (ORB).
 - Selección de mejores emparejamientos empleando la distancia de Hamming.

1

Rectificación y registro de bandas hiperespectrales

- Objetivo: obtener ortomosaico hiperespectral.
- **Deformación geométrica de imágenes hiperespectrales.**
 - Procesamiento automático propuesto en un trabajo previo:
 1. Subdivisión de imágenes hiperespectrales.
 2. Extracción de características en común entre el ortomosaico RGB y las subdivisiones del hipercubo.
 3. Cálculo de matriz de homografía.

1

Rectificación y registro de bandas hiperespectrales

- Objetivo: obtener ortomosaico hiperespectral.
- **Deformación geométrica de imágenes hiperespectrales.**
 - Procesamiento automático propuesto en un trabajo previo:
 1. Subdivisión de imágenes hiperespectrales.
 2. Extracción de características en común entre el ortomosaico RGB y las subdivisiones del hipercubo.
 3. Cálculo de matriz de homografía.
 4. Transformación de subdivisión hiperespectral.

1

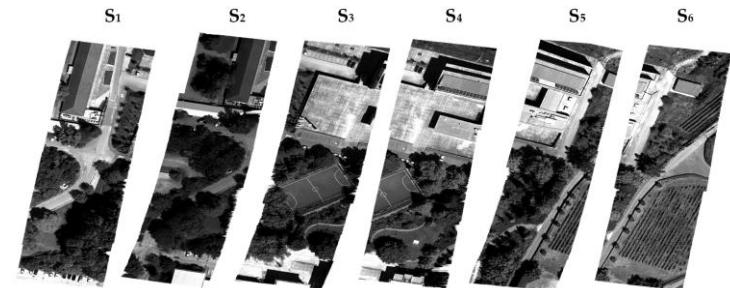
Rectificación y registro de bandas hiperespectrales

- Objetivo: obtener ortomosaico hiperespectral.
- **Deformación geométrica de imágenes hiperespectrales.**
 - Procesamiento automático propuesto en un trabajo previo:
 1. Subdivisión de imágenes hiperespectrales.
 2. Extracción de características en común entre el ortomosaico RGB y las subdivisiones del hipercubo.
 3. Cálculo de matriz de homografía.
 4. Transformación de subdivisión hiperespectral.
 5. Validación.
 1. Distancia entre puntos de control en imágenes hiperespectrales y ortomosaico RGB.
 2. Distancia mayor que umbral → Modifica tamaño de fragmento y repite proceso.

2

Generación de mosaico hiperespectral

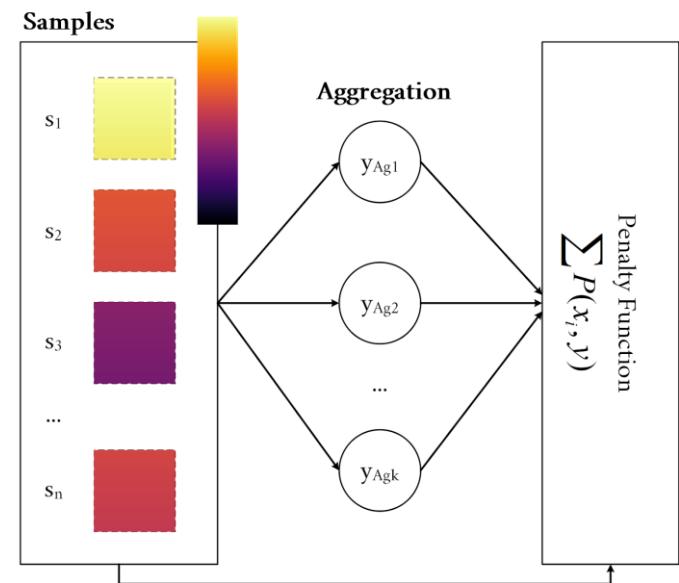
- Datos de entrada.
 - Bandas hiperespectrales con una menor deformación geométrica. Geolocalizadas en sistema UTM tras registrarlas con ortomosaico RGB.
 - Ortomosaico RGB.
- Inicialización de ortomosaico con *Axis-Aligned Bounding Box* (AABB) global de todas las bandas.
 - Muestreo de bandas hiperespectrales para cada píxel del ortomosaico.
- Solape de bandas:
 - Selección de valores hiperespectrales mediante:
 1. Funciones de agregación.
 2. Funciones penalty (Distancia).



2

Generación de mosaico hiperespectral

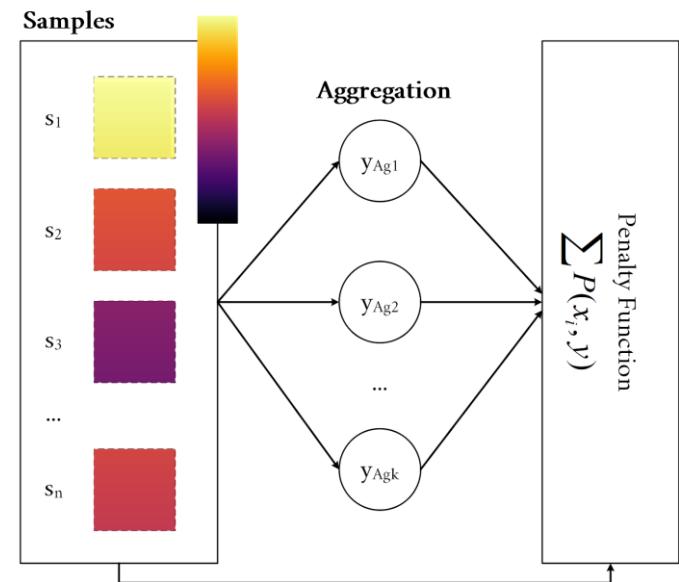
- Solape de bandas:
 - Selección de valores hiperespectrales mediante:
 1. Funciones de agregación.
 - Media aritmética, media harmónica, media geométrica, máximo, mínimo, etc.
 - Subrutinas en shader.
 - $Y: \mathbb{R}^n \rightarrow \mathbb{R}$



2

Generación de mosaico hiperespectral

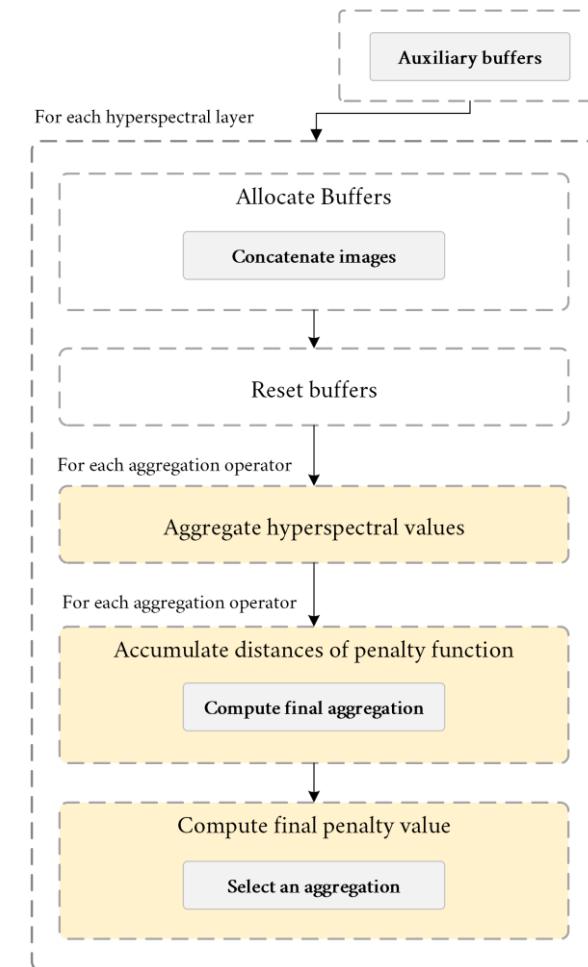
- Solape de bandas:
 - Selección de valores hiperespectrales mediante:
 1. Funciones de agregación.
 - Media aritmética, media harmónica, media geométrica, máximo, mínimo, etc.
 - Subrutinas en shader.
 - $Y: \mathbb{R}^n \rightarrow \mathbb{R}$
 2. Funciones penalty.
 - Distancia de agregación, y , a los valores agregados, x_i .
 - $P: \mathbb{R}^2 \rightarrow \mathbb{R}$
 - $P(x_i, y) = (x_i - y)^2$
 - $D(X, y) = \sum_{x_i \in X} P(x_i, y)$



2

Generación de mosaico hiperespectral

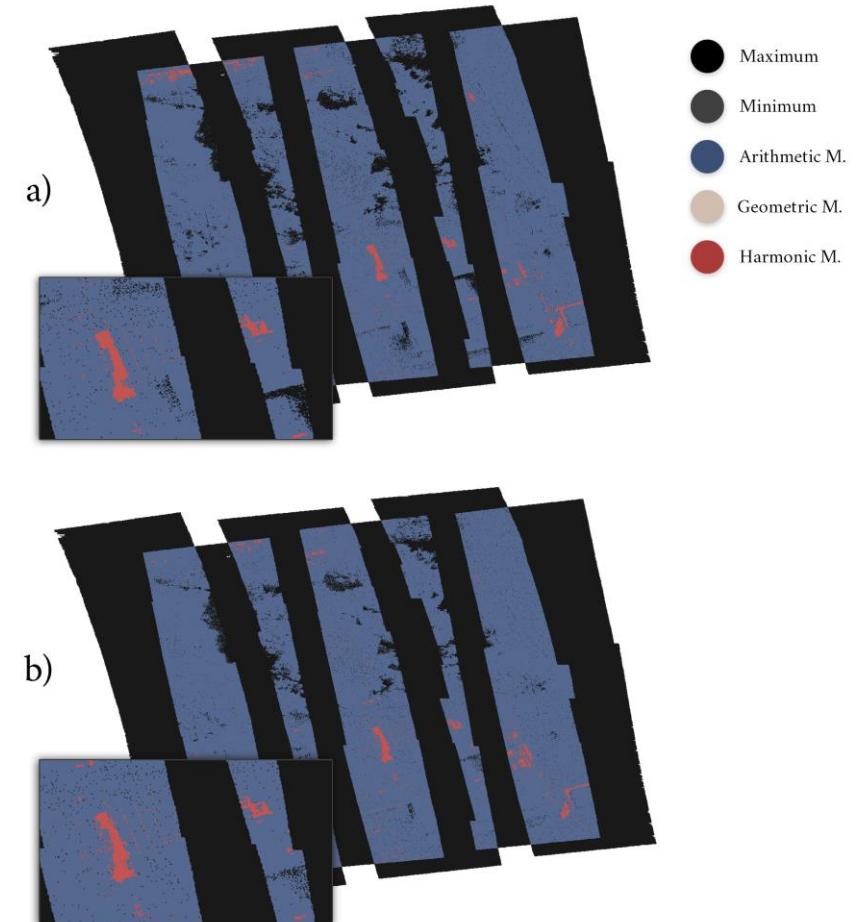
- Combinación de bandas hiperespectrales en GPU, mediante compute shaders:
 1. Transferencia de datos a GPU.
 2. Cálculo de agregaciones.
 3. Finalización de agregaciones (basados en media) + distancia de agregación a valores.
 4. Selección de agregación.



2

Generación de mosaico hiperespectral

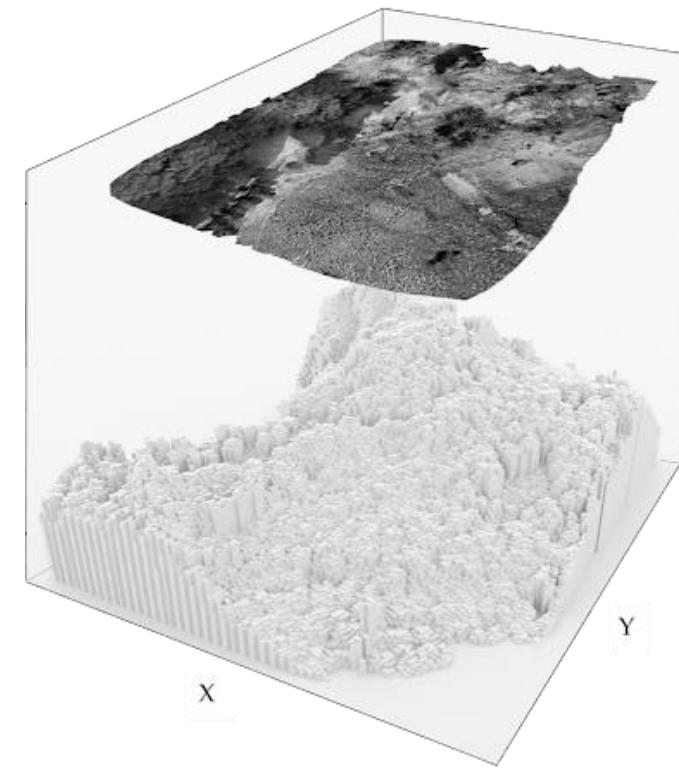
- Combinación de bandas hiperespectrales en GPU, mediante compute shaders:
 1. Transferencia de datos a GPU.
 2. Cálculo de agregaciones.
 3. Finalización de agregaciones (basados en media) + distancia de agregación a valores.
 4. Selección de agregación.
- Procesamiento por banda hiperespectral.
 - Funciones de agregación seleccionadas pueden variar por capa.



3

Mapping de ortomosaico hiperespectral

- Vuelo hiperespectral desarrollado en dirección de visión *nadir*.
 - Proyección empleando la representación 2.5D de la nube RGB.
 - Selección en GPU de punto visible en cada píxel del ortomosaico hiperespectral.
 - Punto de mayor altura.
- Nube de puntos RGB downsampled debido a la resolución hiperespectral.
 - 330M puntos → 17.5M puntos visibles en vuelo hiperespectral.
- Transferencia de imágenes a GPU.
 - Sampling de valores hiperespectrales:
 - Punto en nube.
 - Posición de este respecto a la caja envolvente de ortomosaico hiperespectral.

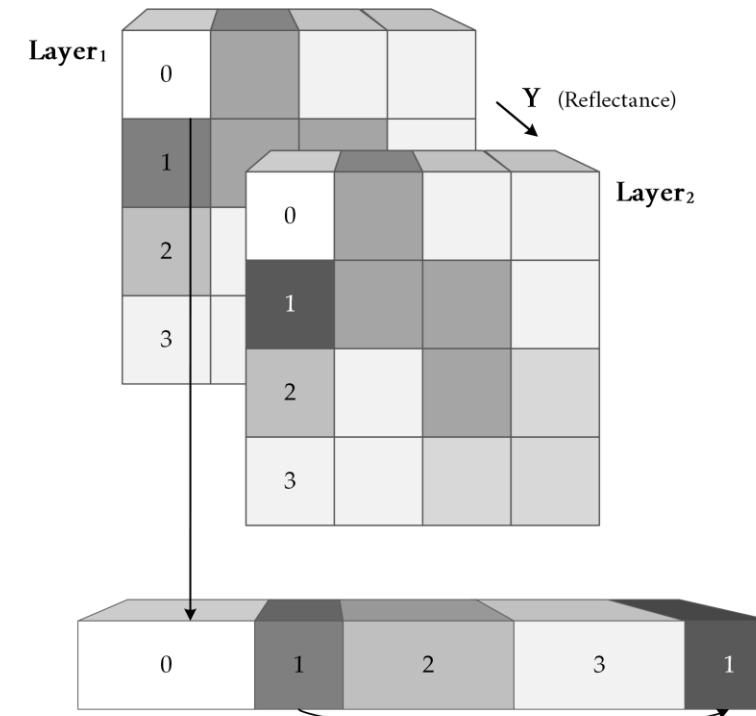


→ Coordenada de textura 2D

4

Compresión de hipercubo

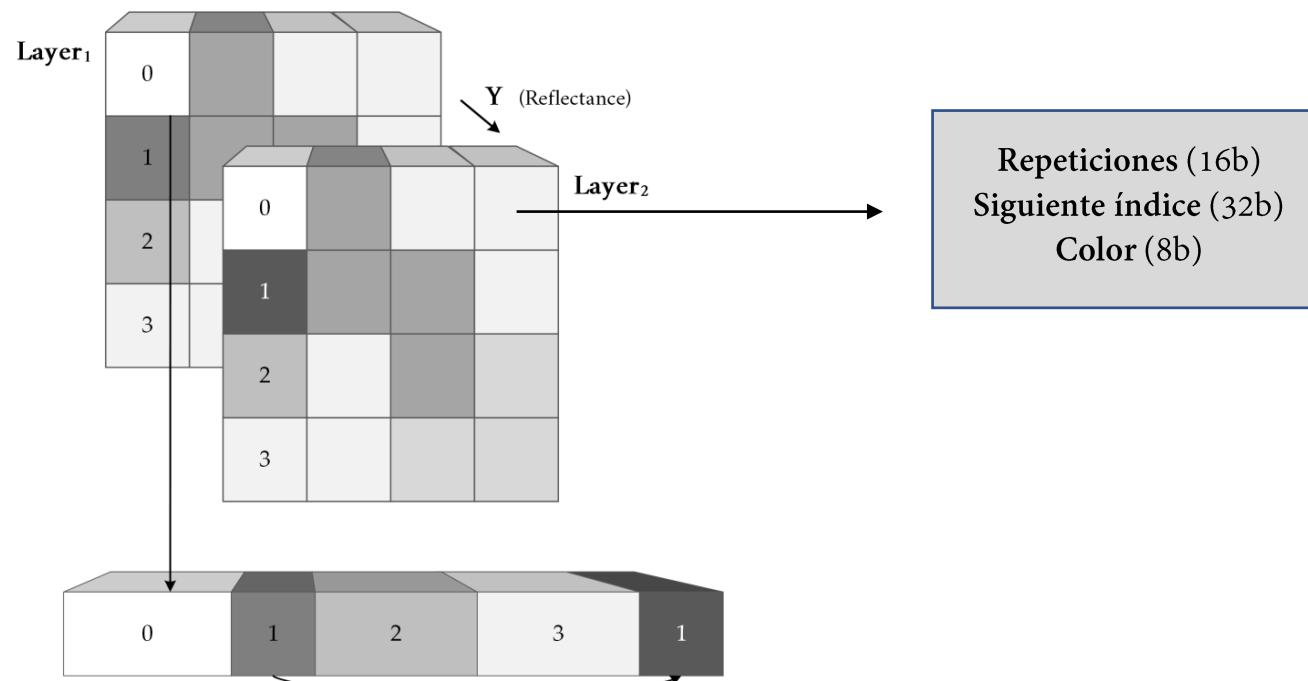
- Nube de puntos RGB downsampled debido a la resolución hiperespectral.
 - 330M puntos → 17.5M puntos visibles en vuelo hiperespectral.
 - Incluso con esta reducción de tamaño...
 - $17.5M * 8 \text{ bits } (2^8) * 270 \text{ capas} = 4.4 \text{ Gigabytes.}$
- Existen algoritmos de compresión sin pérdida.
 - Satélites y drones.
 - Enfocado a compresión *offline*.
 - Descompresión *real-time* no trivial y lenta.
- Reducir tamaño de almacenamiento.
- Deep Learning / Machine Learning. Reducción de número de features.



4

Compresión de hipercubo

- Elevada compresión de hipercubo reduciendo espacio hiperespectral a 2^8 valores diferentes.

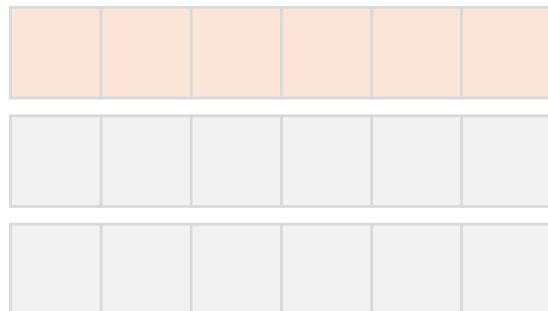


Pavia Centre (92.13%)
Pavia University (92.19%)
Kennedy Space (95.43%)
Salinas Valley (96.39%)
Cuprite (96.41%)

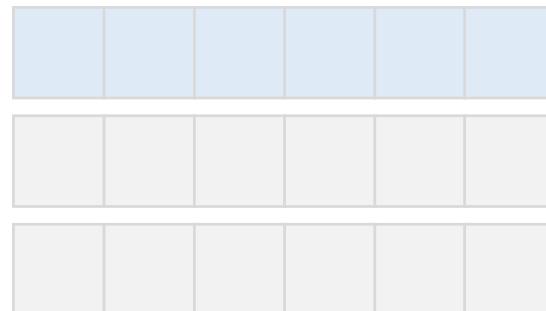
4

Compresión de hipercubo

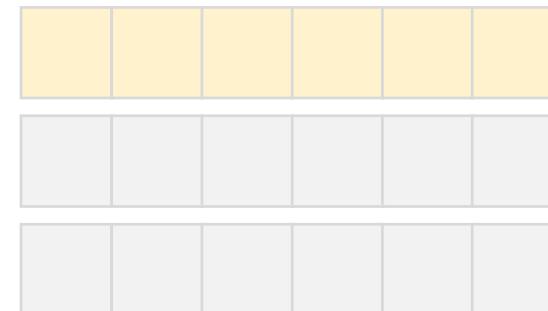
- Nubes de millones de puntos.
 - *Shader Storage Buffer Objects* (SSBOs) con grandes limitaciones de memoria.
 - Partición de nube en *buffers*.
 - Tamaño acorde a un *buffer* con compresión cero.
 - Integrando 270 capas.



P_1 , 30M



P_2 , 30M



P_3 , 30M

5

Rendering

- Utilización de *compute shaders* para renderizado de nubes de puntos.
 - Evita etapas del *pipeline de rendering*.
 - Utilización de *depth buffer* para determinar puntos visibles en *viewport*.
 - 64 bits: distancia + índice.
 - Mínimo atómico acarrea índice.
 - Mejora de visualización.
 - Agregación de color de múltiples puntos para un mismo píxel.
 - Apariencia uniforme para nubes de puntos más dispersas.

GL_POINTS



shuffleNV
ballotThreadN
shuffleXorNV

5

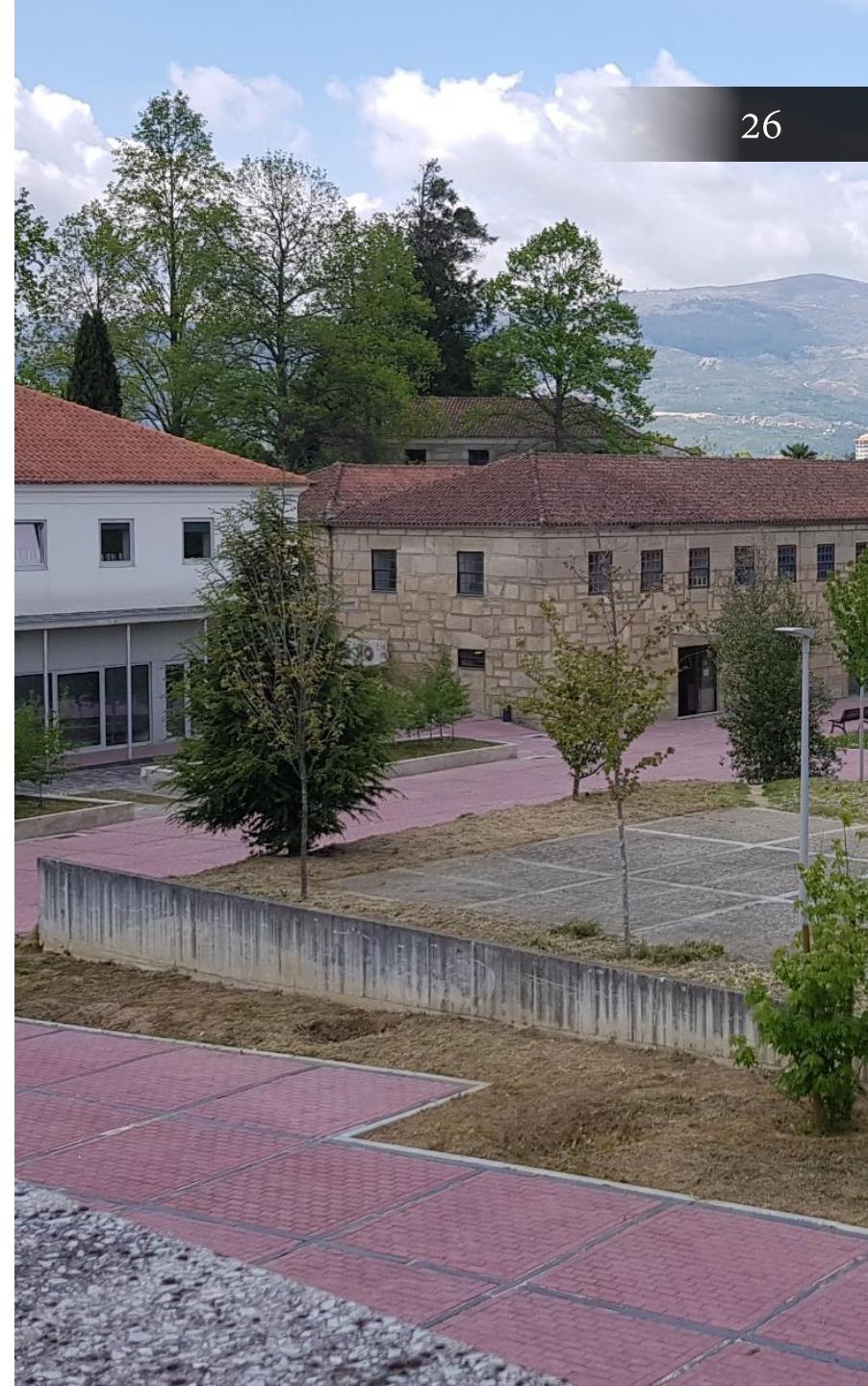
Rendering

- Principales problemas en rendering de hipercubo:
 - Iteración por capas.
 - *Frame rate* de últimas capas mucho menor que para las primeras.
- Rendering parcial de nube de puntos.
 - *Buffer* de índices aleatorios.
 - Desplazamiento de índices en cada frame.
 - Composición de nube en completa en función de frame rate y número de índices activados por *frame*.



Experimentación

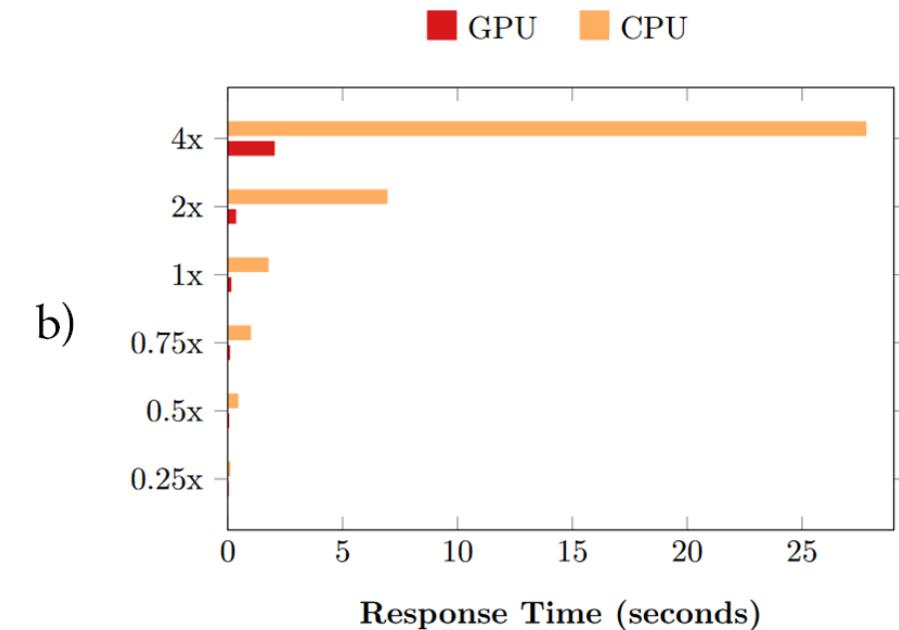
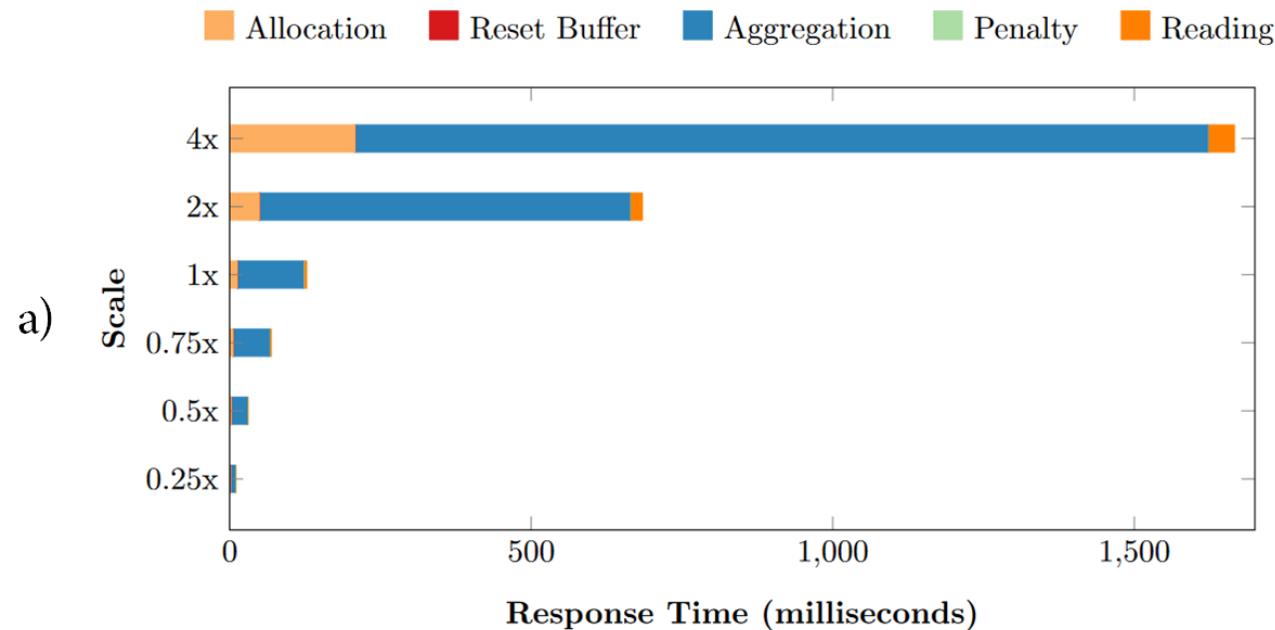
- Nube de puntos RGB:
 - 150M y 330M de puntos.
- Información RGB e hiperespectral sobre 4 hectáreas.
 - University of Trás-os Montes e Alto Douro.
- Tarjeta gráfica empleada:
 - NDIVIA GTX 1070 (8GB VRAM).
- OpenGL's *compute shaders* y OpenMP para paralelismo CPU.



Experimentación

27

- Generación de ortomosaico hiperespectral:



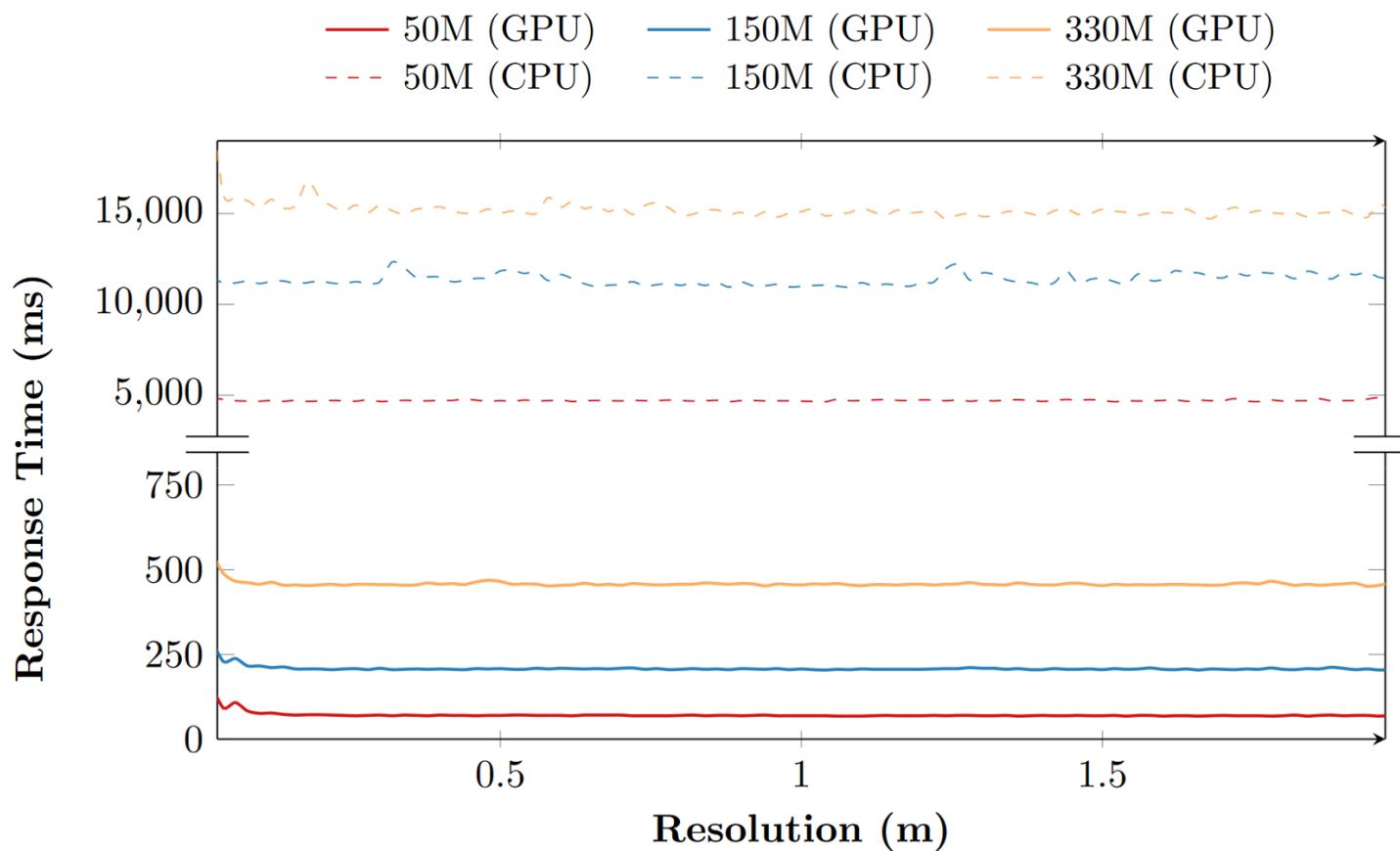
Speedups:

- 0.5x: 74.69%
- 1x: 92.54%

Experimentación

28

- Voxelización (2.5D) y recuperación de puntos visibles:



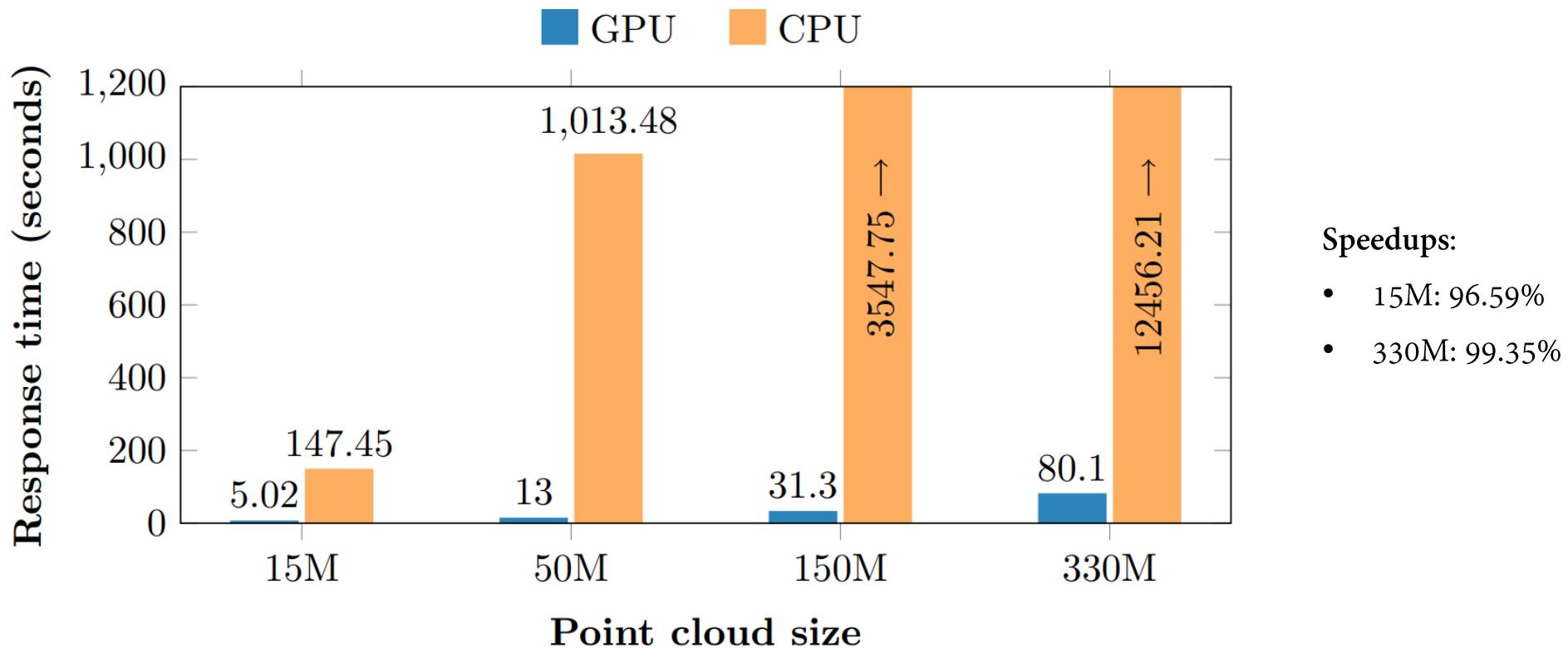
Speedups:

- 330M: 96.66%

Experimentación

29

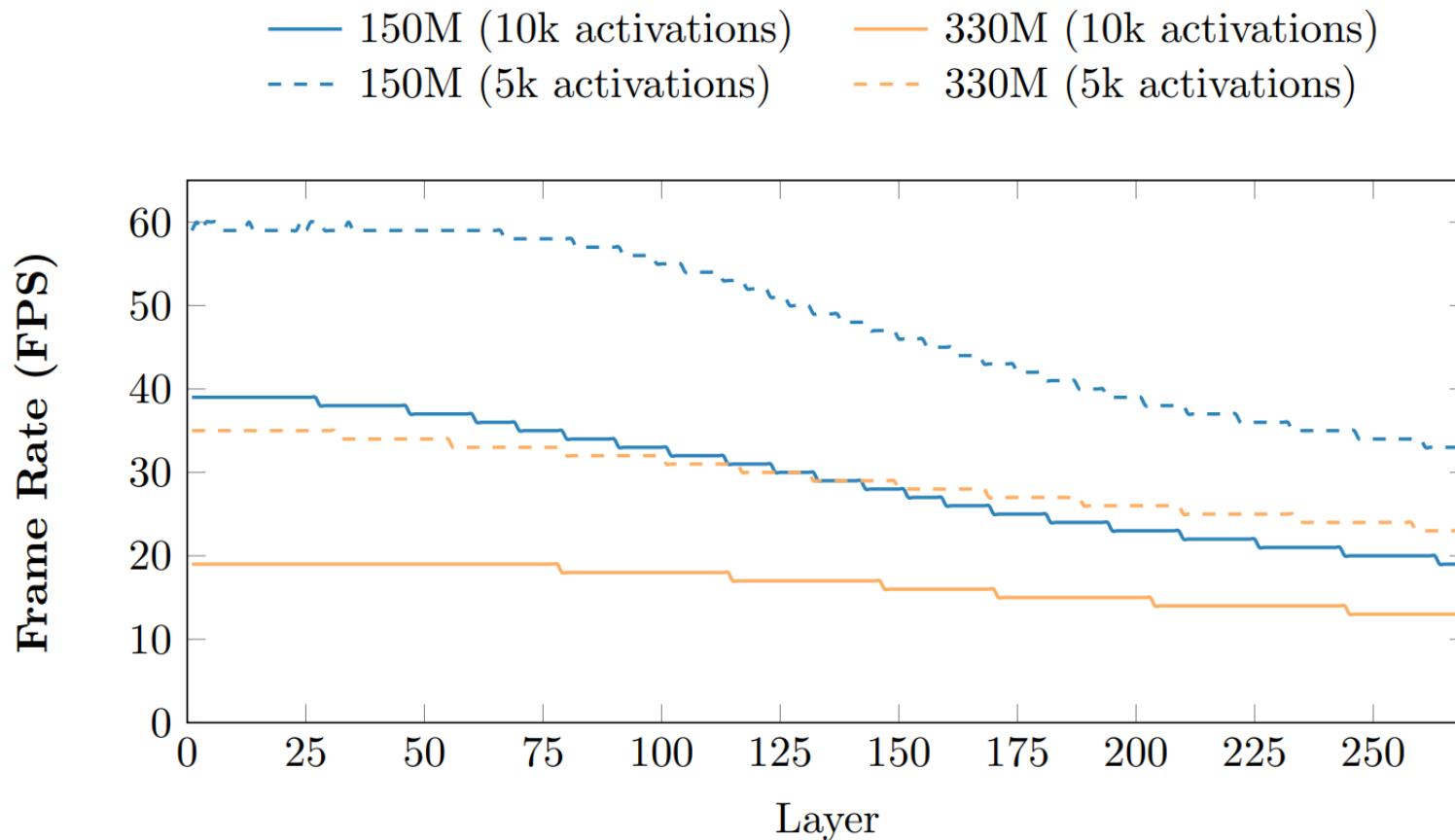
- Construcción de nube hiperespectral, incluyendo compresión:



Experimentación

30

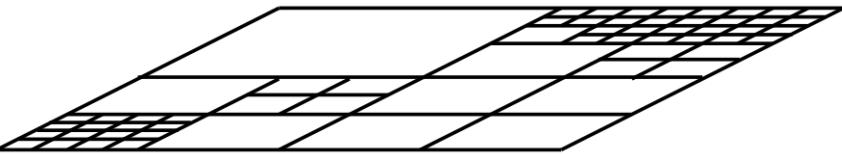
- Rendering en tiempo real:

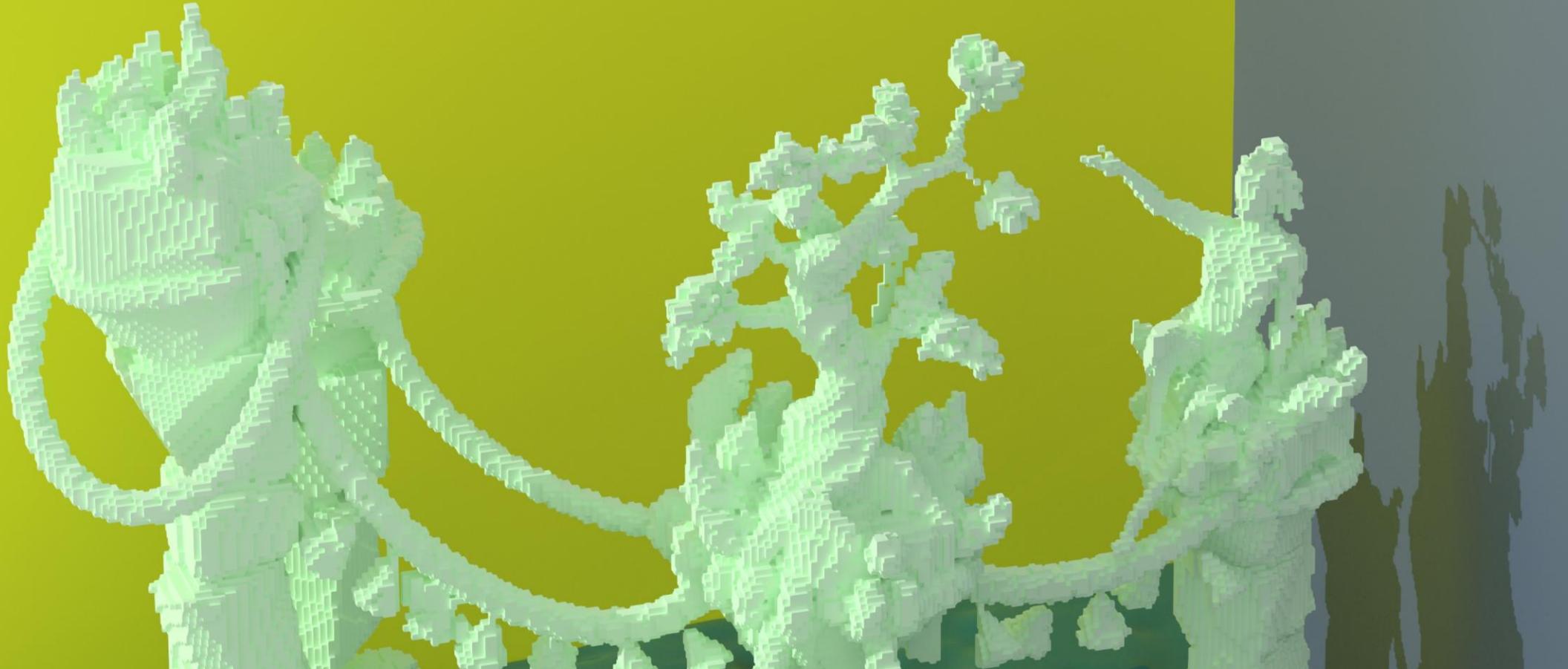


Conclusiones

31

- Generación automática de nubes de puntos hiperespectrales.
- Pipeline en GPU.
 - Prácticamente todas las etapas mejoran su rendimiento por encima del 90%.
- Mejora de visualización de nubes de puntos poco densas.
- Compresión de hipercubo para tareas de visualización.
- **Trabajo futuro:**
 - Compresión de datos en X, Z.
 - Aumentar rendimiento de rendering.
 - Reducir almacenamiento en memoria.





¡Gracias! ¿Preguntas?



Grupo de Gráficos y
Geomática de Jaén



Universidad
de Jaén