Captura e integración de datos multisensoriales en entornos 3D

Retos y oportunidades para el HPC y enfoques GPGPU

Juan Manuel Jurado

Universidad de Jaén

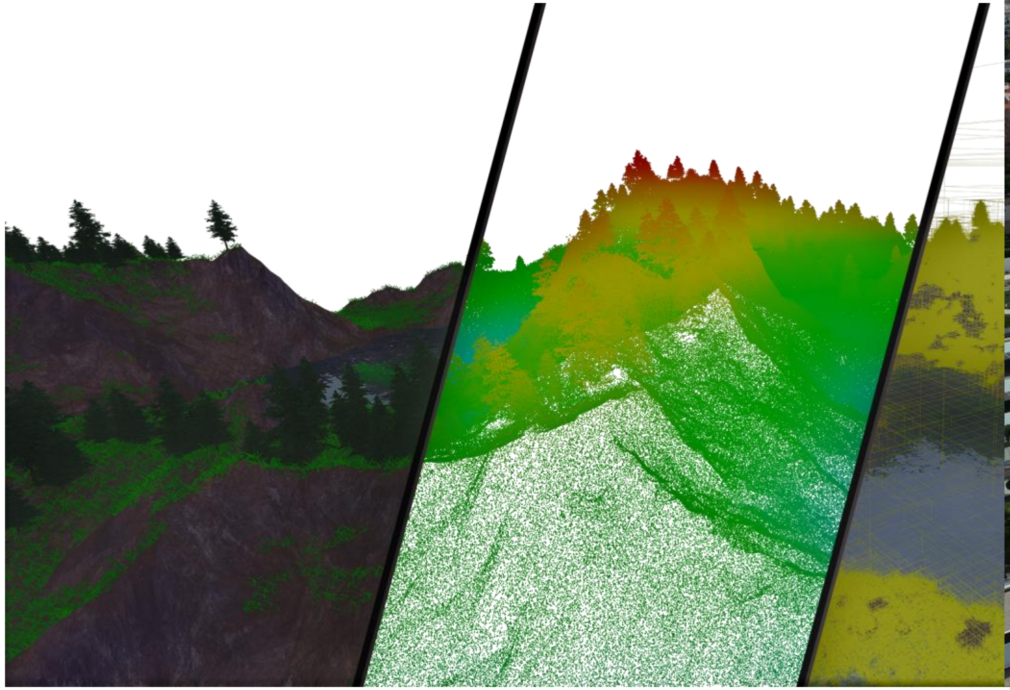
Computer Graphics and Geomatics Lab

https://gggj.ujaen.es/

Universidad de Jaén

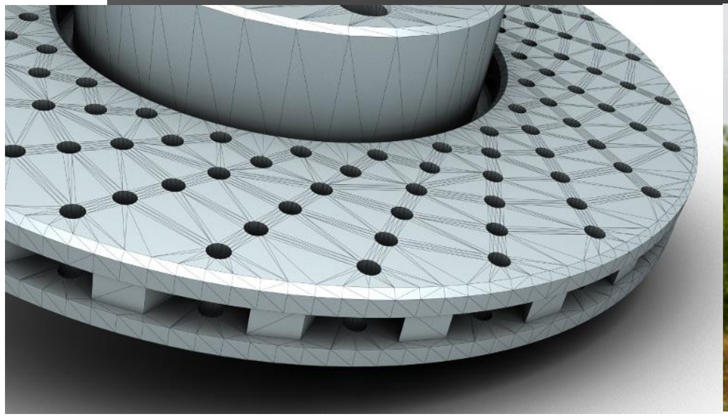
Centro de Estudios Avanzados en Tecnologías de la Información y la Comunicación

https://ceatic.ujaen.es/



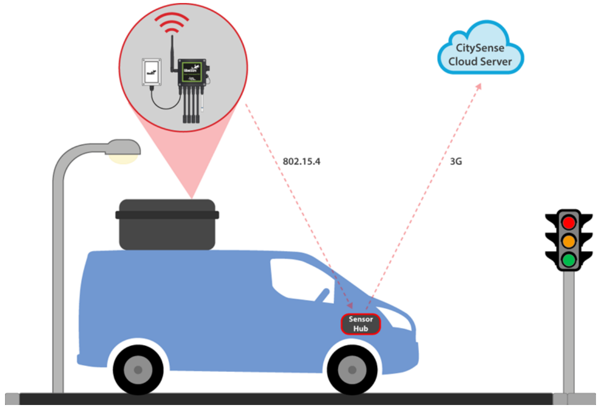
Research lines

- Computational geometry
- Geographic Information Systems (GIS)
- Geometric modeling
- Image processing
- Physical-based simulations
- Characterization of real-world scenes



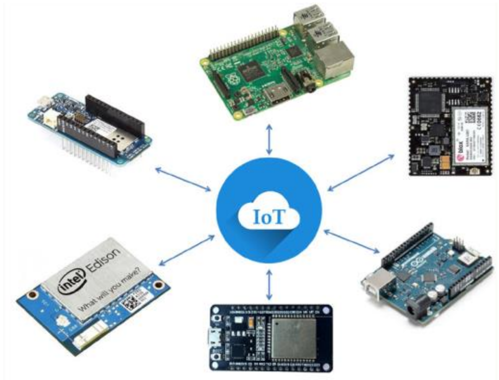
Introduction

World of sensors



Air quality

Agriculture



IoT

UAVs

Generation of huge datasets

Introduction

UAVs

Multispectral camera

High-resolution RGB camera

Thermal camera

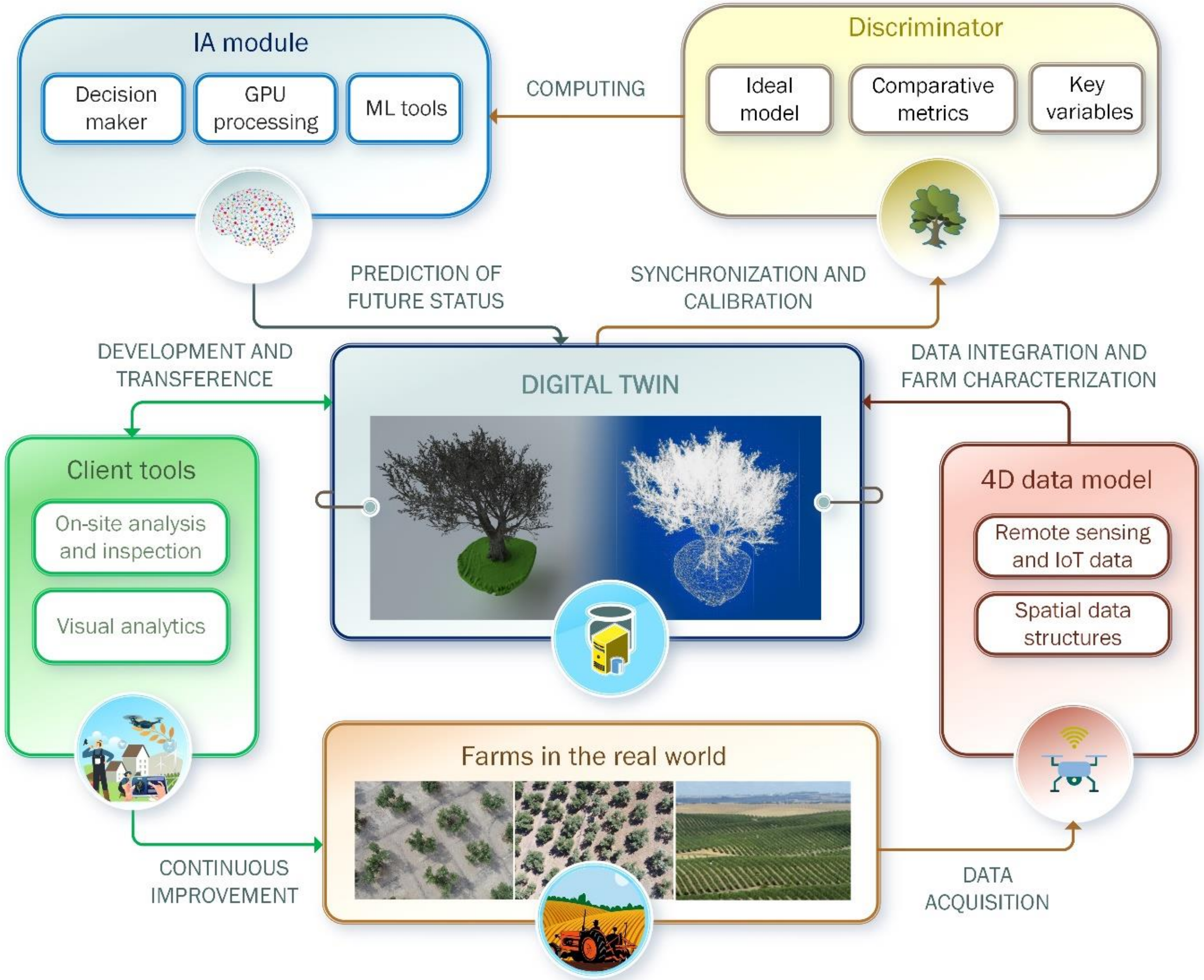
Hyperspectral camera

Introduction

¿What is digital twin?

“Un gemelo digital puede definirse como una representación digital del mundo real que permite comprobar constantemente su comportamiento, analizarlo y actuar en consecuencia, tanto de forma inmediata como prediciendo su comportamiento en el futuro.

En definitiva, un gemelo digital debe gestionar el ciclo de vida del sistema, monitorizándolo, analizando su comportamiento y actuando sobre él de forma inteligente para mantener o conseguir situaciones de comportamiento óptimas.”

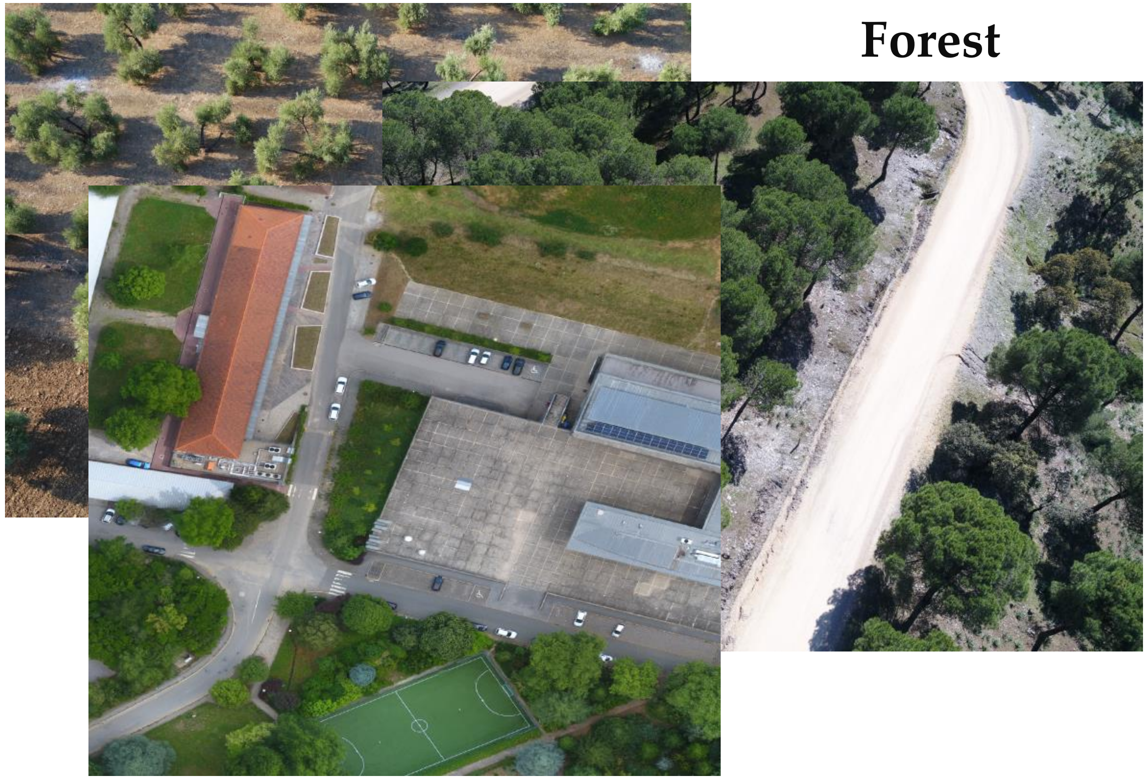


Data acquisition

Study cases

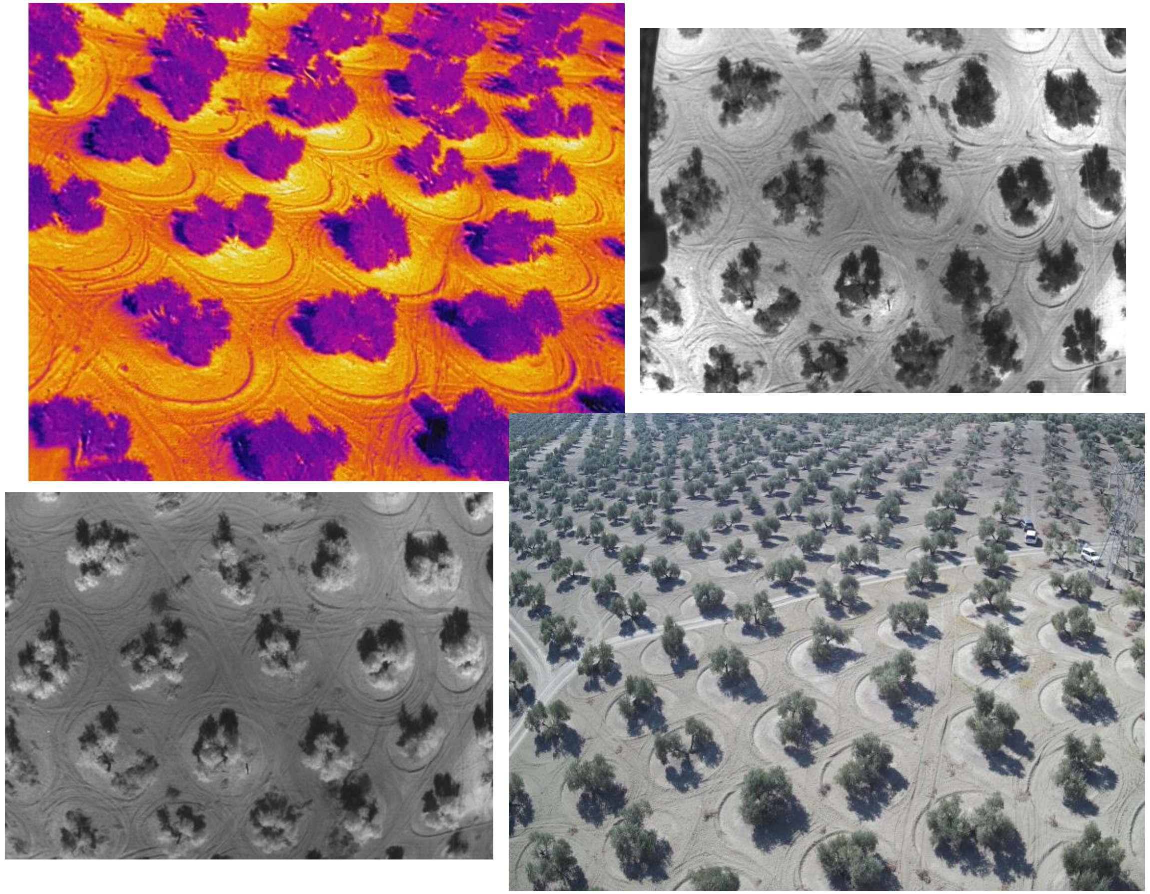
Olive trees

Forest



Urban scenarios

Multi-source images



HPC infrastructure

ADA cluster

SIE

Descripción:

- This cluster provides a compute capacity of 133,632 CUDA cores, provided by Tesla Volta and RTX Turing cards.
- Main features::
  - 1 management node
  - 2 computational nodes based on GPUs (x2 NVIDIA V100)
  - 4 computational nodes based on GPU (x7 NVIDIA GeForce RTK 2080Ti)
- For the next investment:
  - 1 storage node
  - 1 computational node (4 GPUs A100 with NVLINK)

Edge Computing

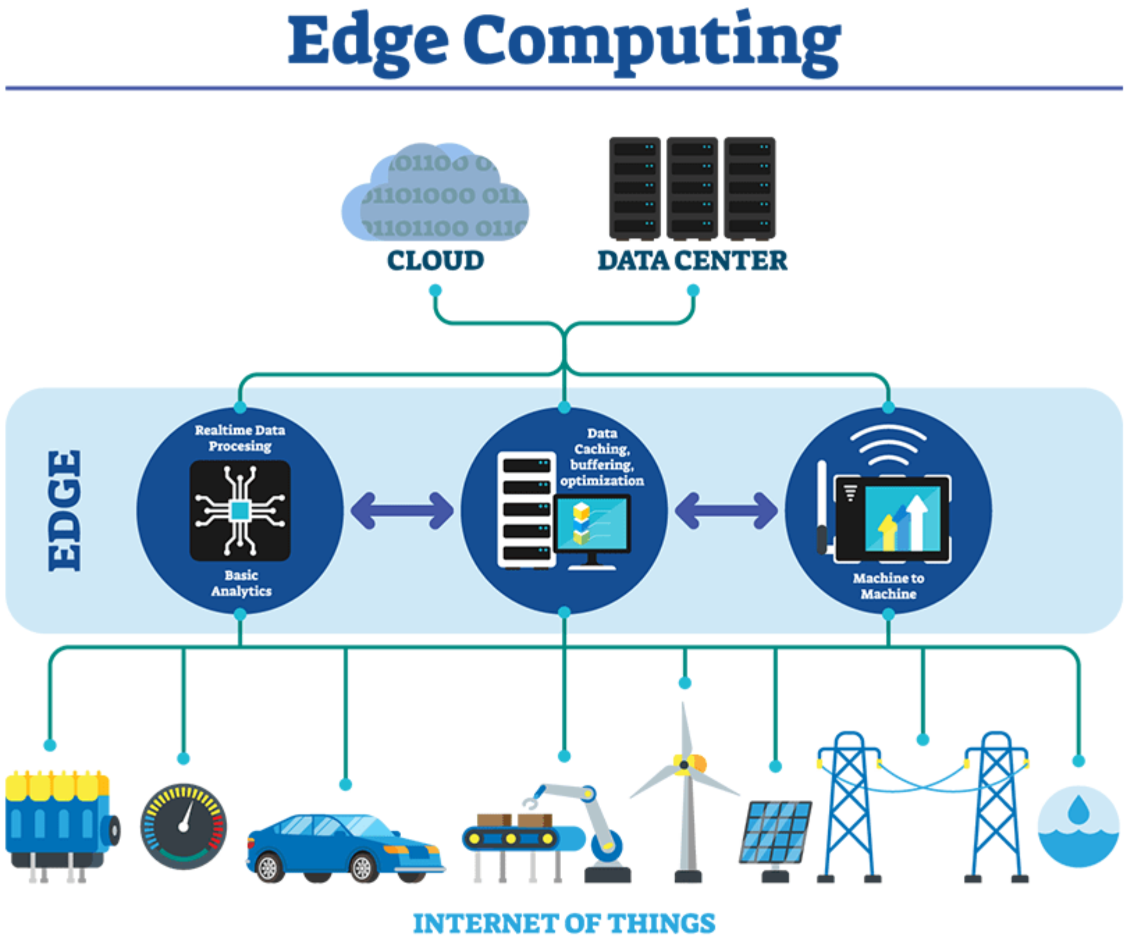
Edge Computing

¿What is edge computing?

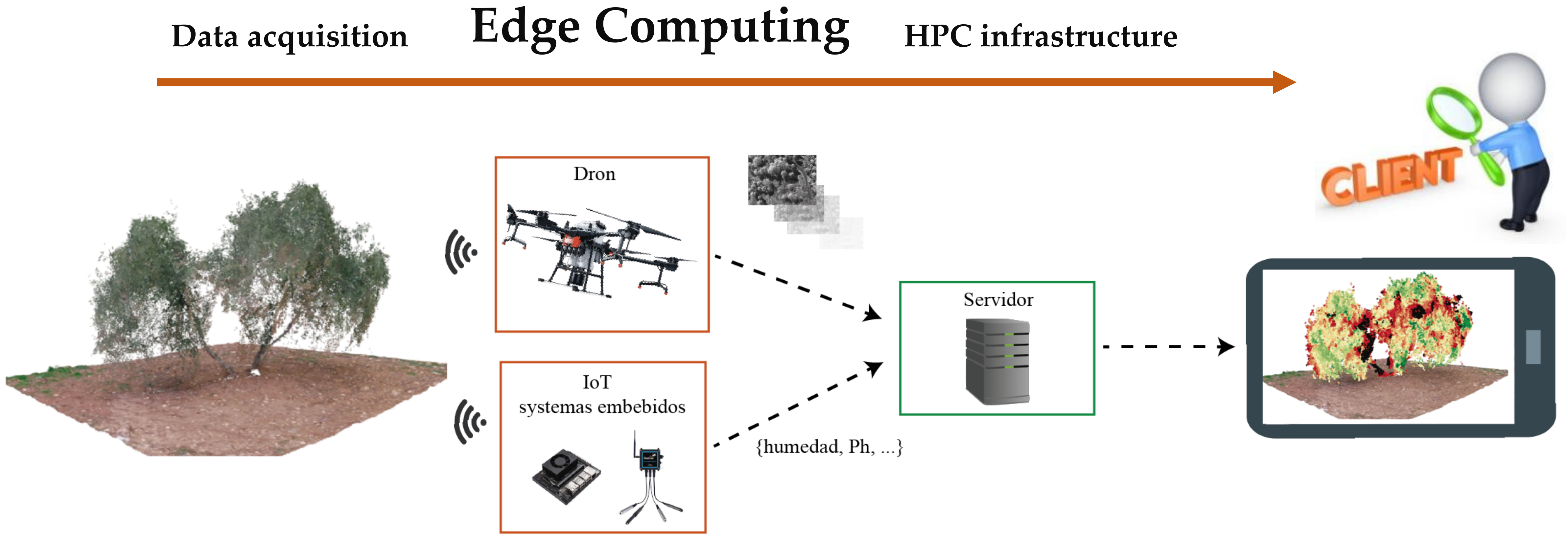
"El Edge Computing es un paradigma complementario al **cómputo en la nube**. En este, los datos son procesados por los dispositivos en vez de depender de servidores centralizados para procesarlos.

Esto tiene ventajas respecto a latencia de respuesta, seguridad y privacidad. Además, permite que dependamos menos de los pocos proveedores de soluciones en la nube, que son Amazon, Google, Microsoft e IBM.

Pero no sólo se pueden resolver problemas de privacidad, también se puede aplicar **aprendizaje máquina** desde el dispositivo, permitiendo que solo se rescate la información más relevante, evitando saturar el ancho de banda de la red."



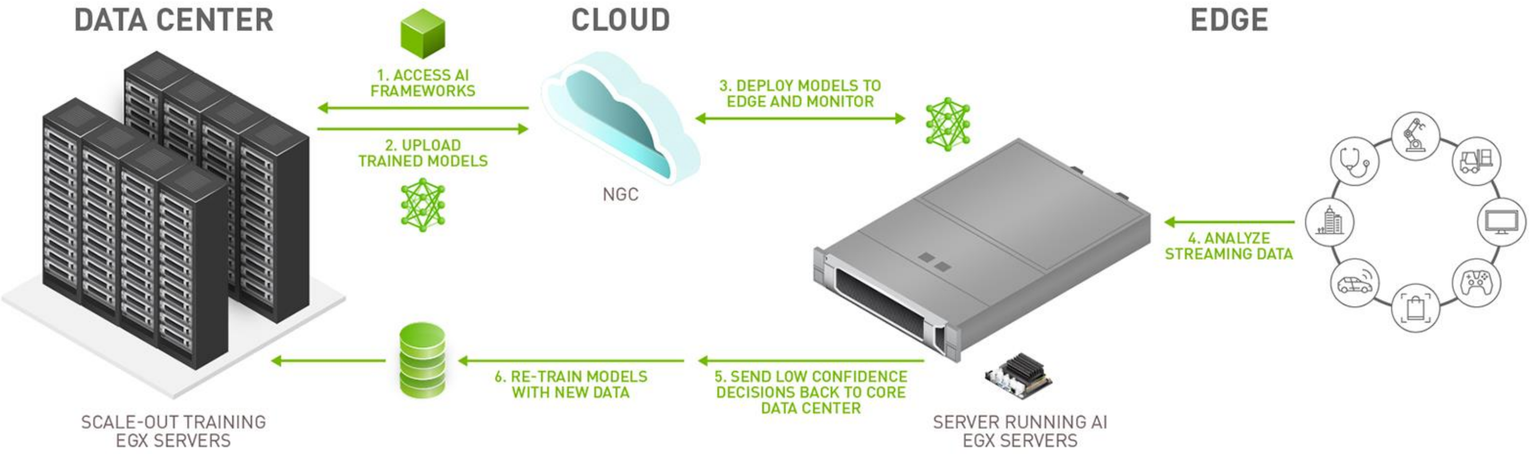
Edge Computing

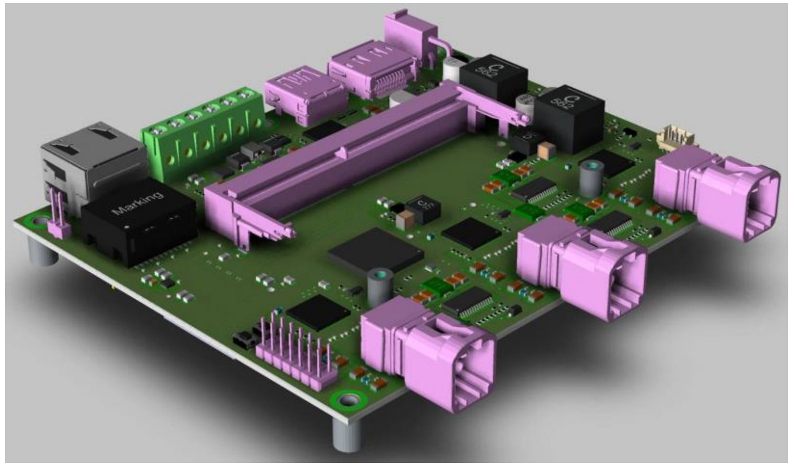


Edge Computing

System on a chip (SoC)

“Un **sistema en chip** (**SoC**, del inglés *system on a chip*) describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador o cualquier otro sistema informático o electrónico en un único circuito integrado o chip.”



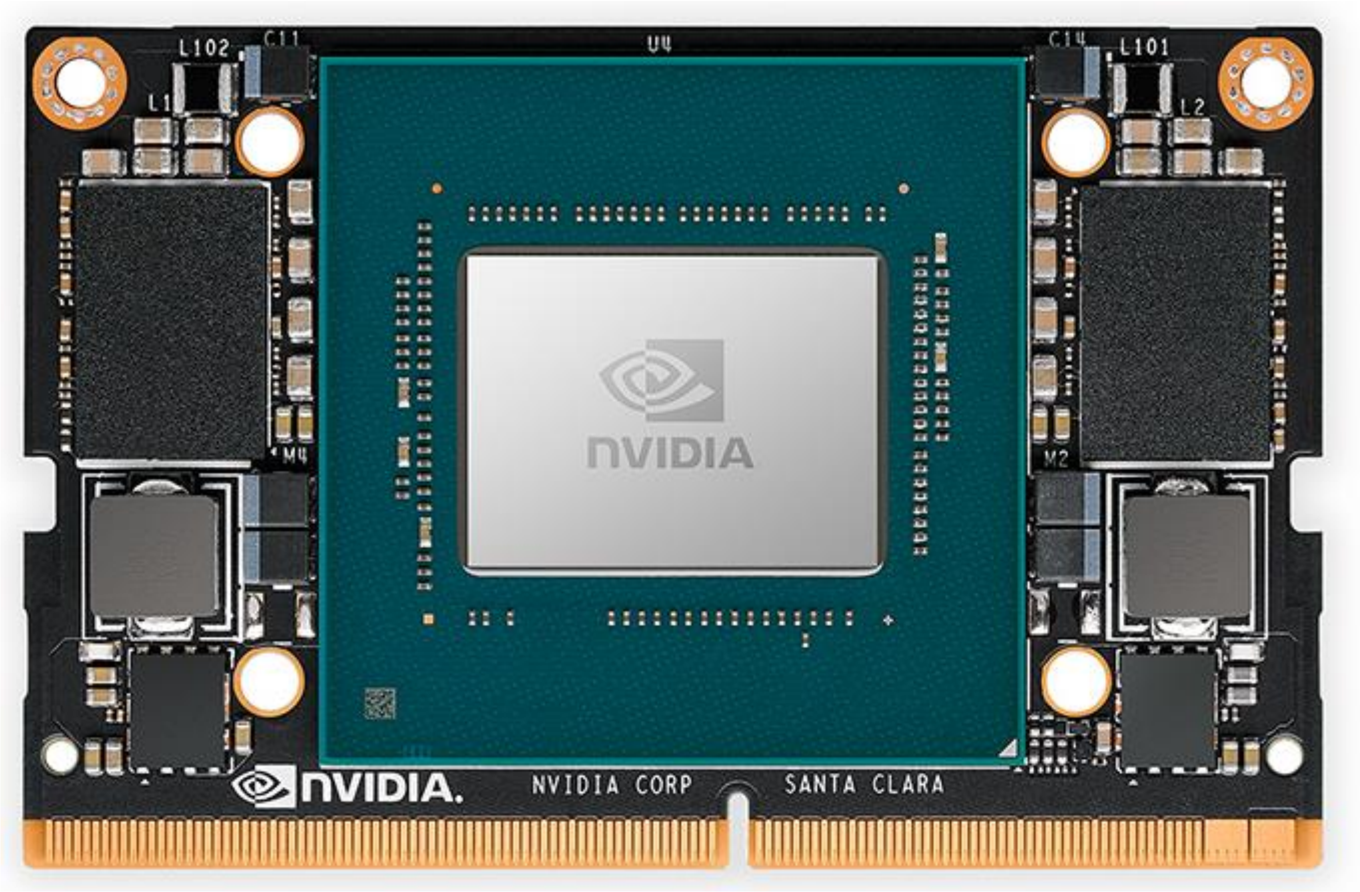


According to market research firm IDC’s “Future of Operations-Edge and IoT webinar,” the edge computing market will be worth $251 billion by 2025, and is expected to continue growing each year with a compounded annual growth rate of 16.4 percent. The evolution of AI, IoT and 5G will continue to catalyze the adoption of edge computing.

Edge Computing

System on a chip (SoC)

NVIDIA® Jetson Xavier™ NX lleva el rendimiento del superordenador a la periferia en un sistema en módulo (SOM) de pequeño formato. Con hasta 21 TOPS de computación acelerada ofrece la potencia para ejecutar redes neuronales modernas en paralelo y procesar los datos de múltiples sensores de alta resolución, un requisito para sistemas completamente de IA.

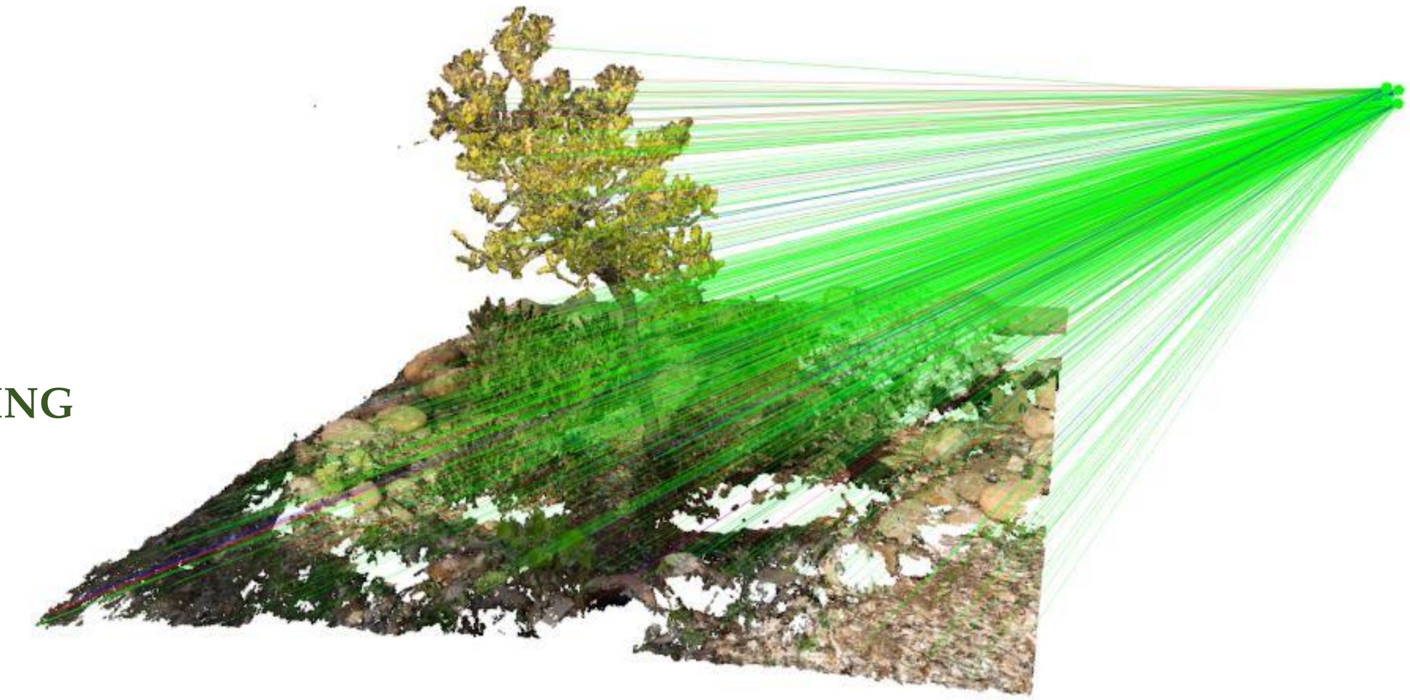


Challenges and motivation

Multi-source image mapping on 3D huge point clouds

Challenges:

- Memory limitation
- Execution time (performance)
- Data paralelization



Solution:

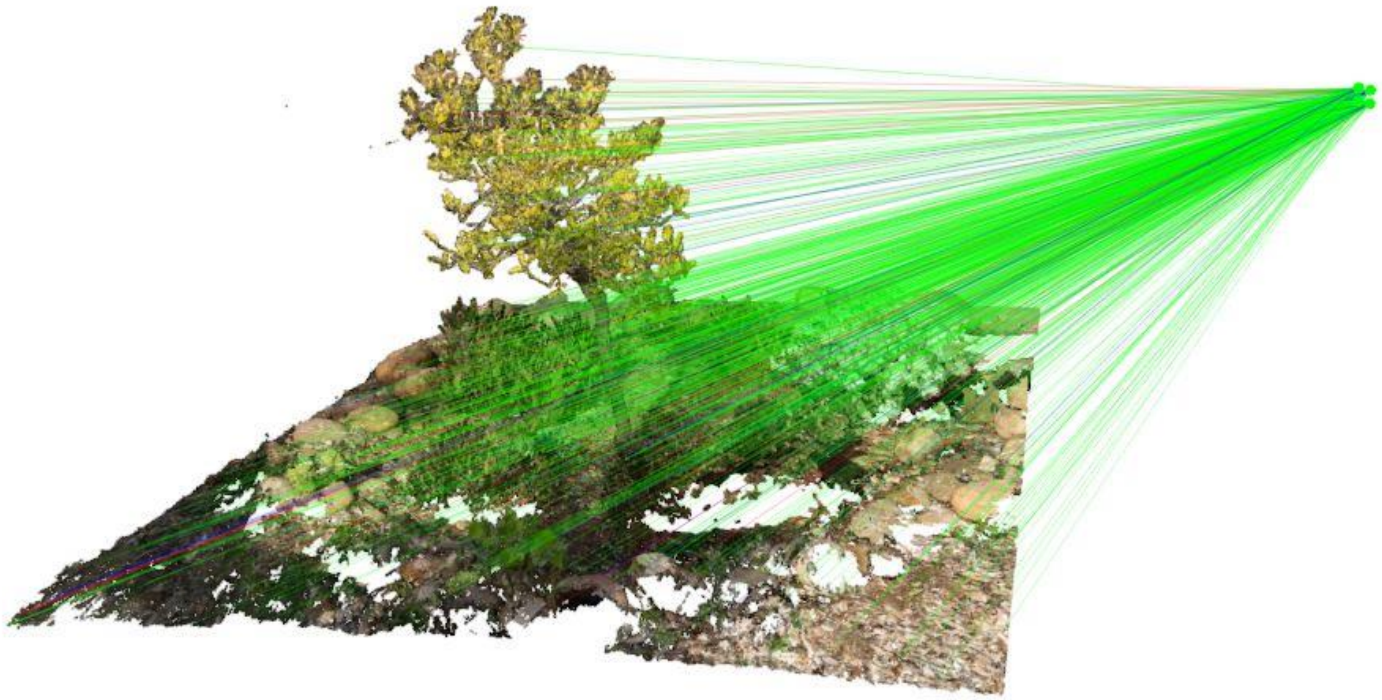
HPC SYSTEM BASED ON GPU-BASED COMPUTING

Challenges and motivation

Multi-source image mapping on 3D huge point clouds

Motivation for GPU development:

- GPUs are designed for computing these operations (projections, plane changes, etc.).
- Data parallelism (no interdependency)

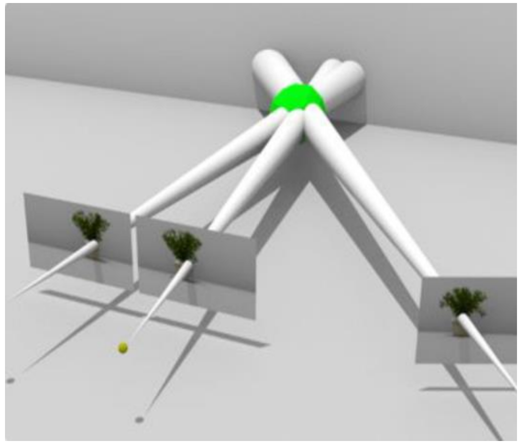


3D Reconstruction

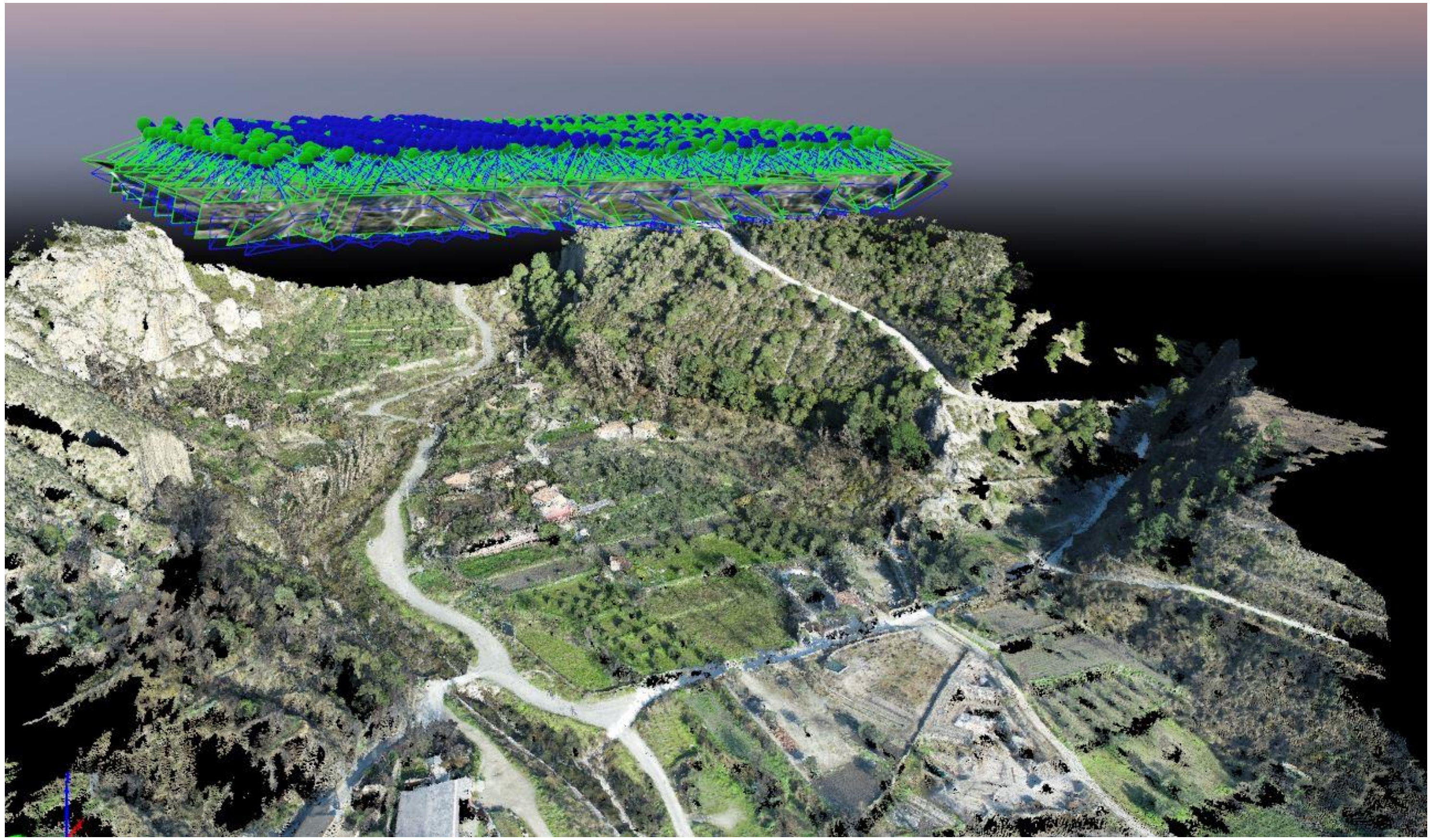
Algorithm: SfM (Structure from Motion)

Description:

Searching for minutiae in images and matching them to each other for the generation of 3D points in a common reference system.



Nube de 40.196.463 puntos



Our proposal

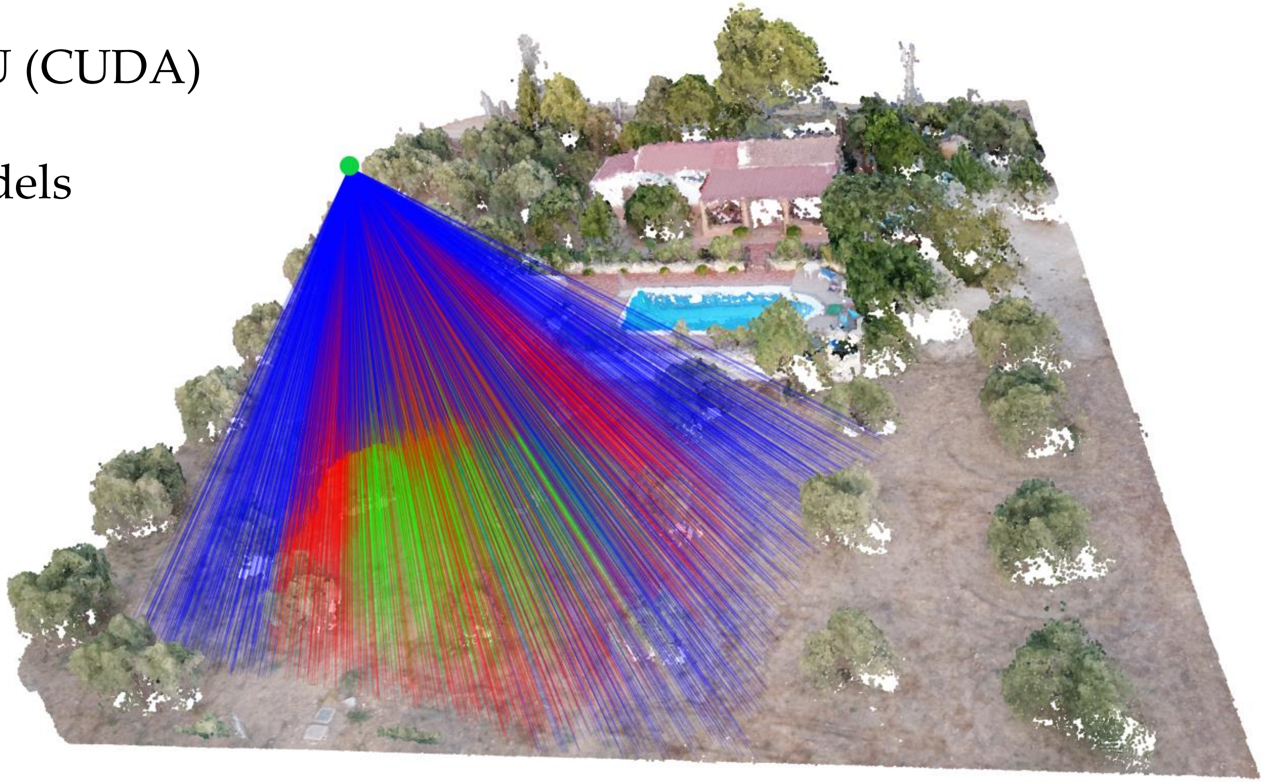
Principales características:

1 – Out-of-the-core method

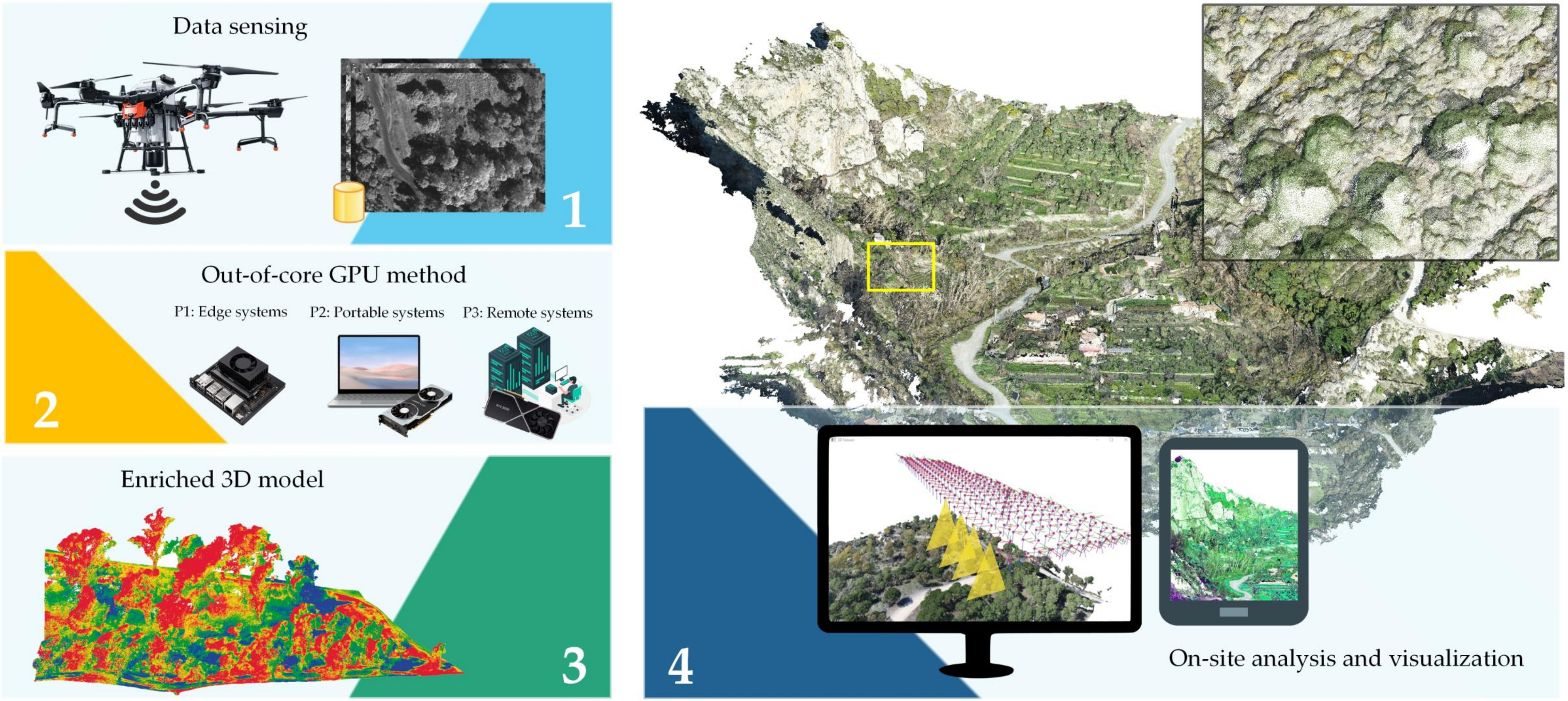
2 – Data parallelization on the GPU (CUDA)

3 – Spatial segmentation of 3D models

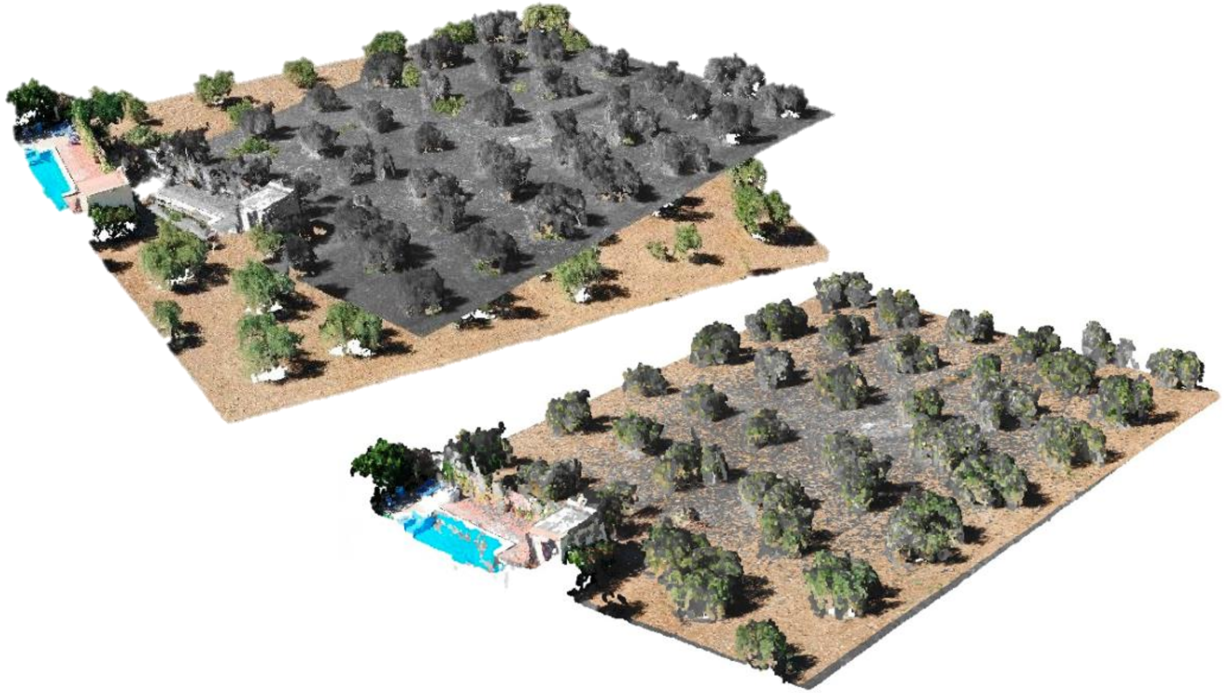
4 – Mapping and occlusion tasks



GPU-based solution



1. Alignment (Iterative closest point)



Hsieh, C. T. (2012, November). An efficient development of 3D surface registration by Point Cloud Library (PCL). In *2012 International Symposium on Intelligent Signal Processing and Communications Systems* (pp. 729-734). IEEE.

$$M = S(sx, sy, sz)\ T(tx, ty, tz)\ R(\beta\ \alpha\ \Omega)$$

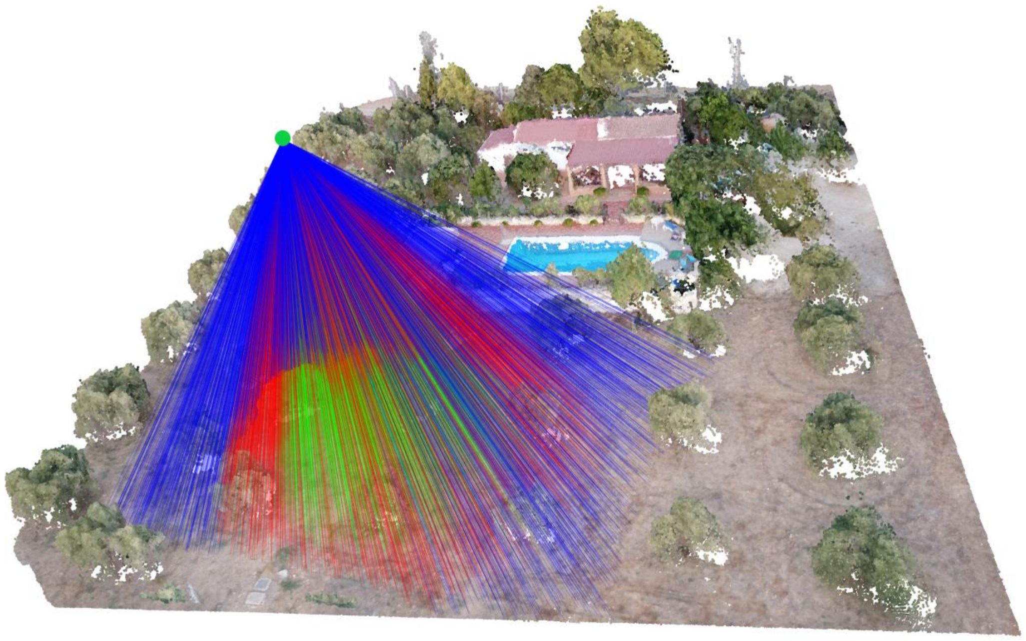
$$S(sx, sy, sz) = \begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

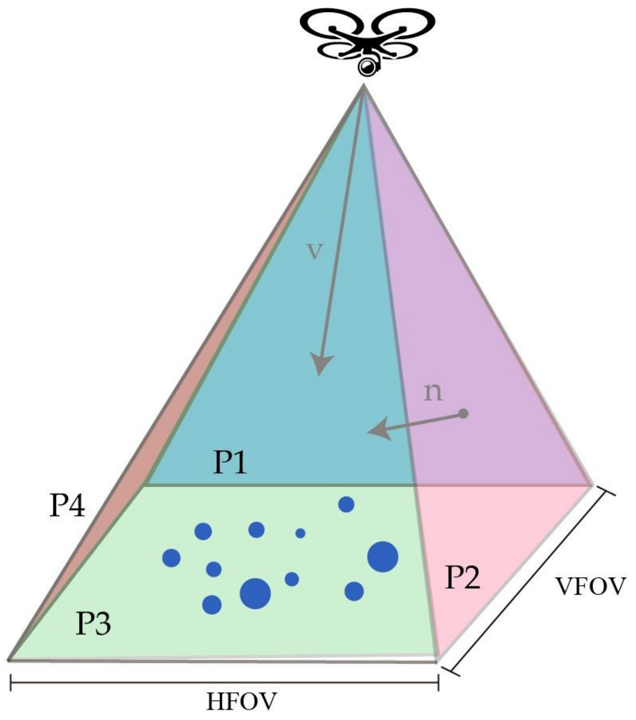
$$T(tx, ty, tz) = \begin{pmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rx(\Omega)\ Ry(\beta)\ Rz(\alpha) = \begin{pmatrix} r11 & r12 & r13 & 0 \\ r21 & r22 & r23 & 0 \\ r31 & r32 & r33 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Multispectral image mapping

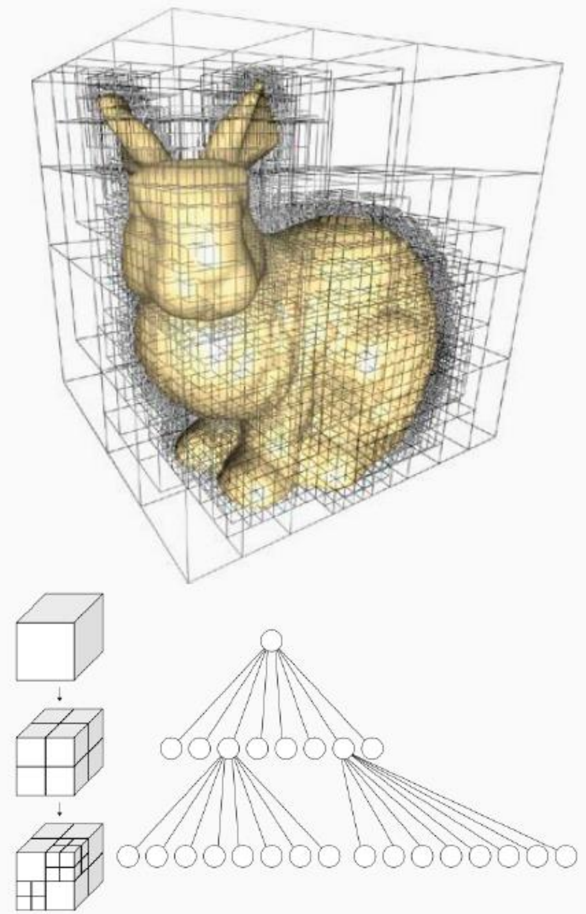
1. Selección de puntos visibles desde la posición de la camara considerando el FoV.





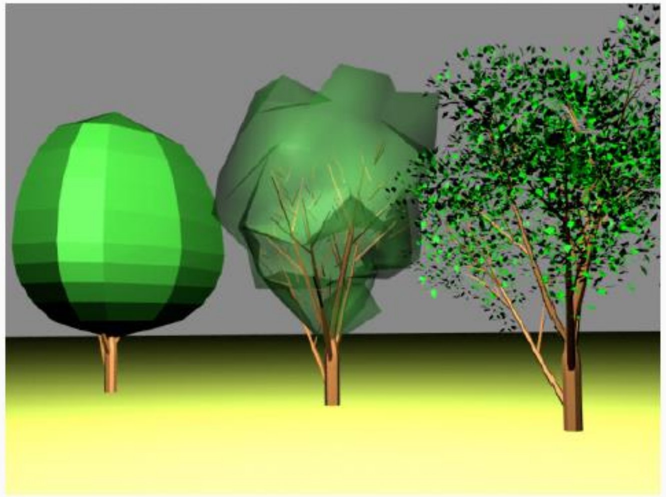
2. Multispectral image mapping

Octree:

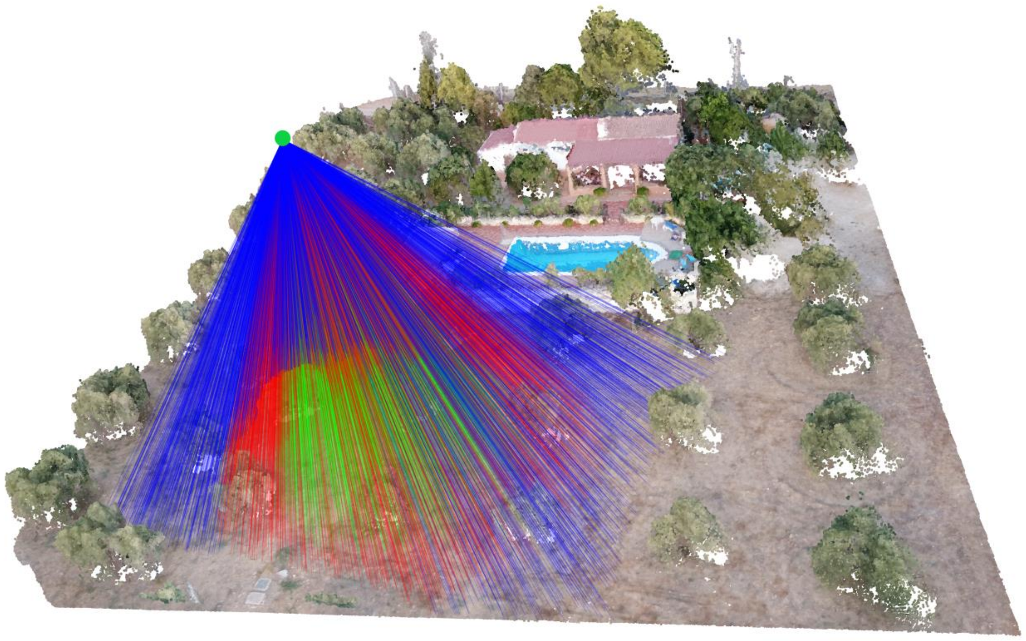


+

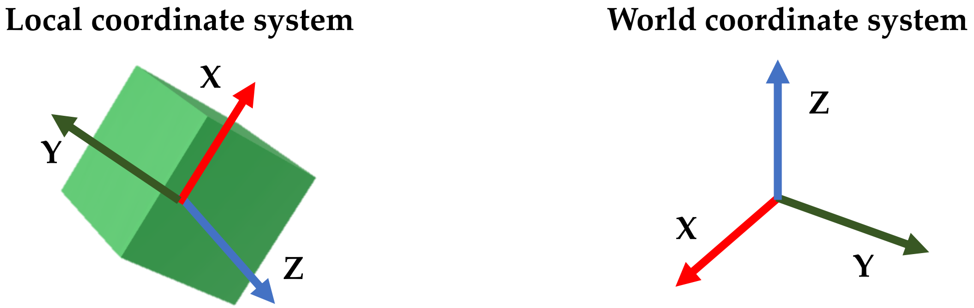
Level of details (LoDs):



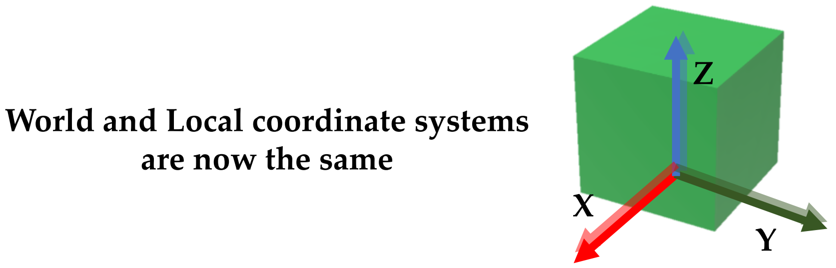
3. Multispectral image mapping



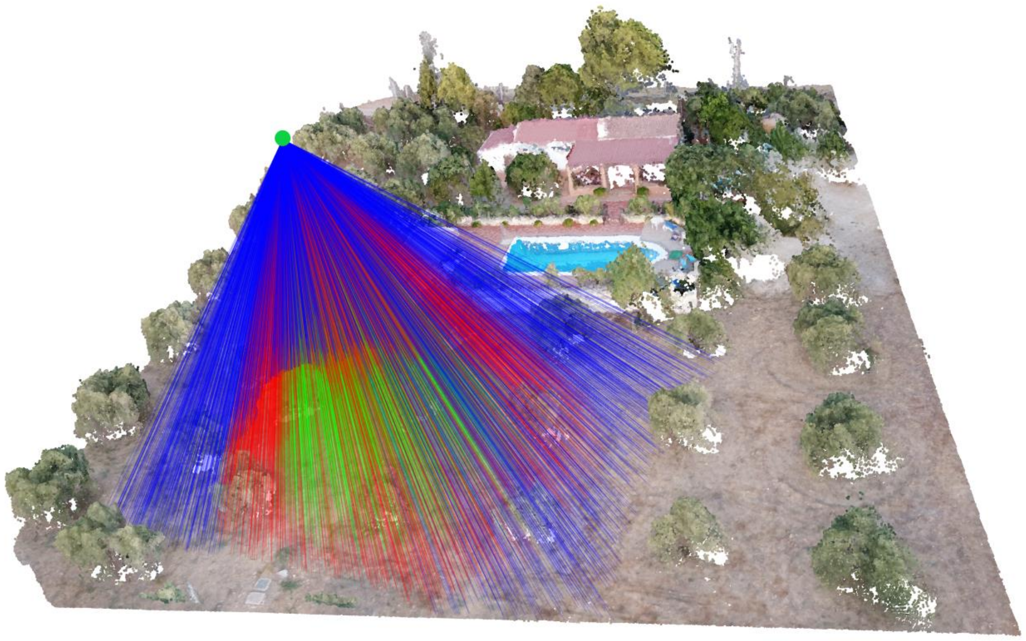
2. Geometric transformation



$$P' = P \cdot (I \cdot -T \cdot R^{-1})$$

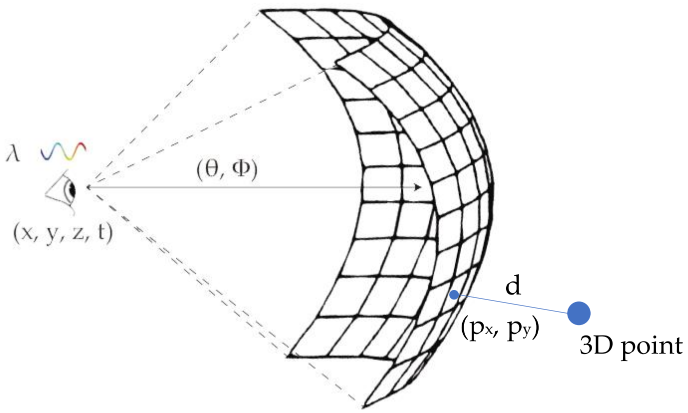


4. Multispectral image mapping



3. Fisheye projection to map each 3D point to its corresponding pixel.

Fish eye model



What is the result?

- 3D point (x, y z)
- Image coordinates
- distance

Polynomial Fisheye Transform

$$\rho = \theta + p_2\theta^2 + p_3\theta^3 + p_4\theta^4$$

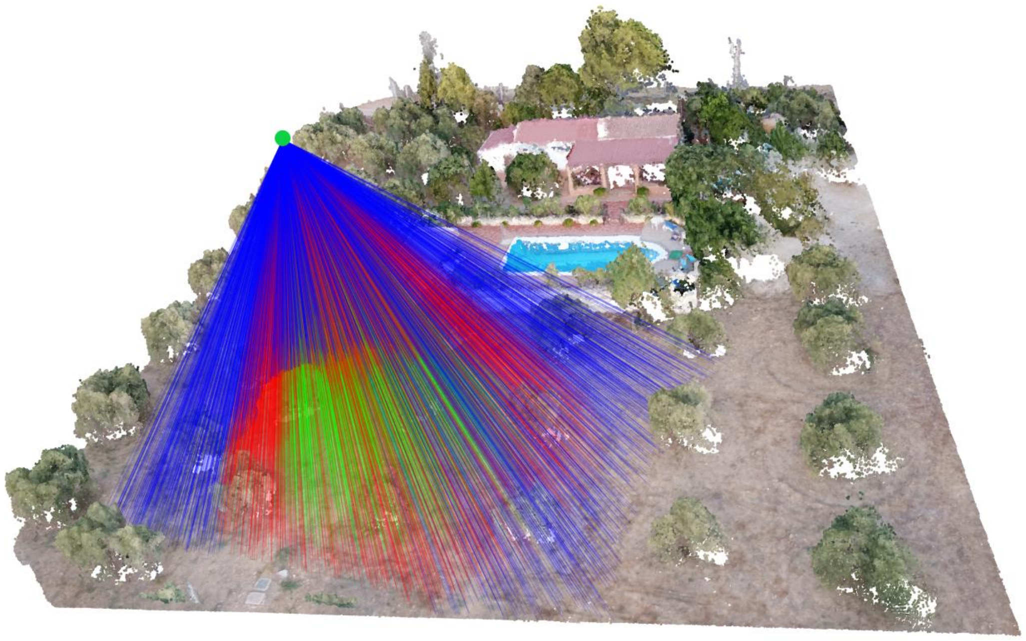
where:

$$\theta = \frac{2}{\pi}\arctan\left(\frac{\sqrt{X^2+Y^2}}{Z}\right); \theta\epsilon[0,1]$$

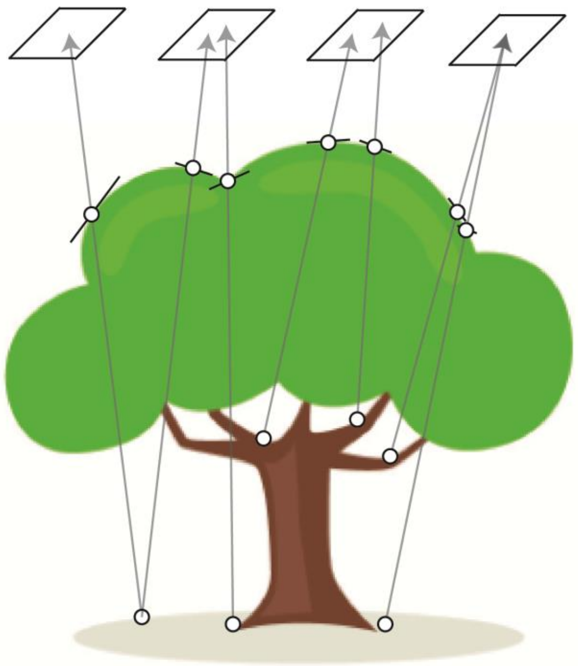
$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} C & D \\ E & F \end{bmatrix} \begin{bmatrix} x_{hbt} \\ y_{hbt} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

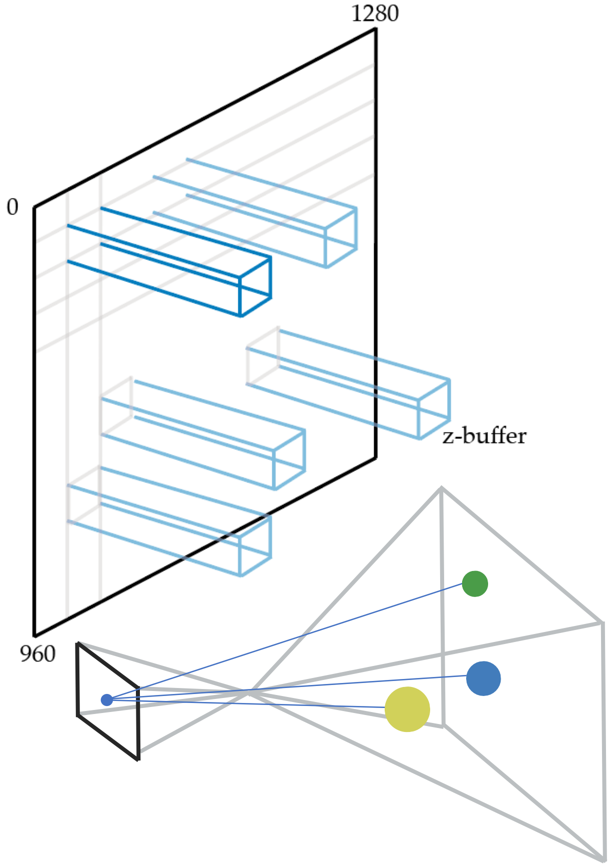
$$\begin{bmatrix} x_{hbt} \\ y_{hbt} \end{bmatrix} = \begin{bmatrix} \frac{\rho X}{\sqrt{X^2+Y^2}} \\ \frac{\rho Y}{\sqrt{X^2+Y^2}} \end{bmatrix}$$

5. Multispectral image mapping



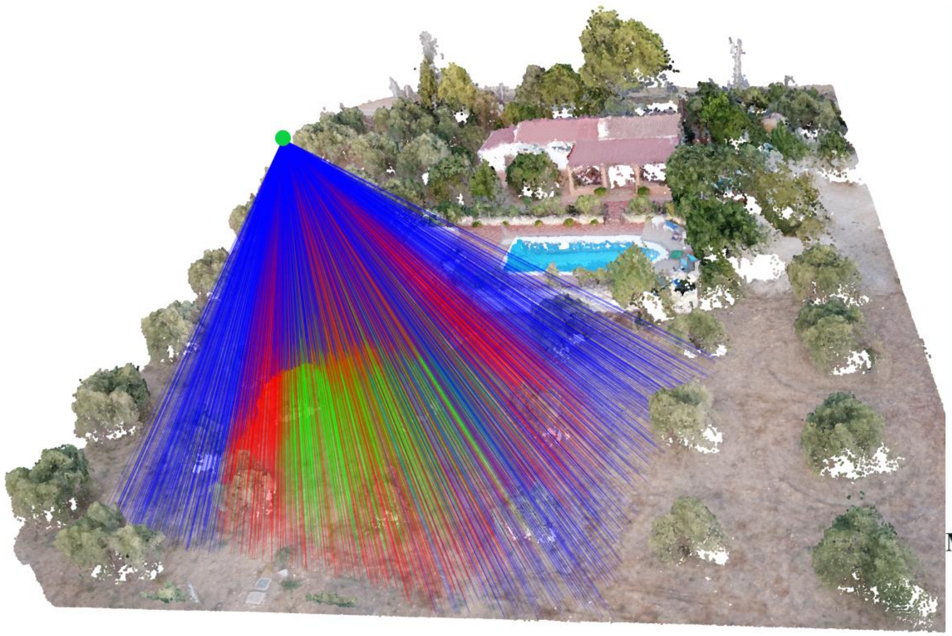
4. Occlusion

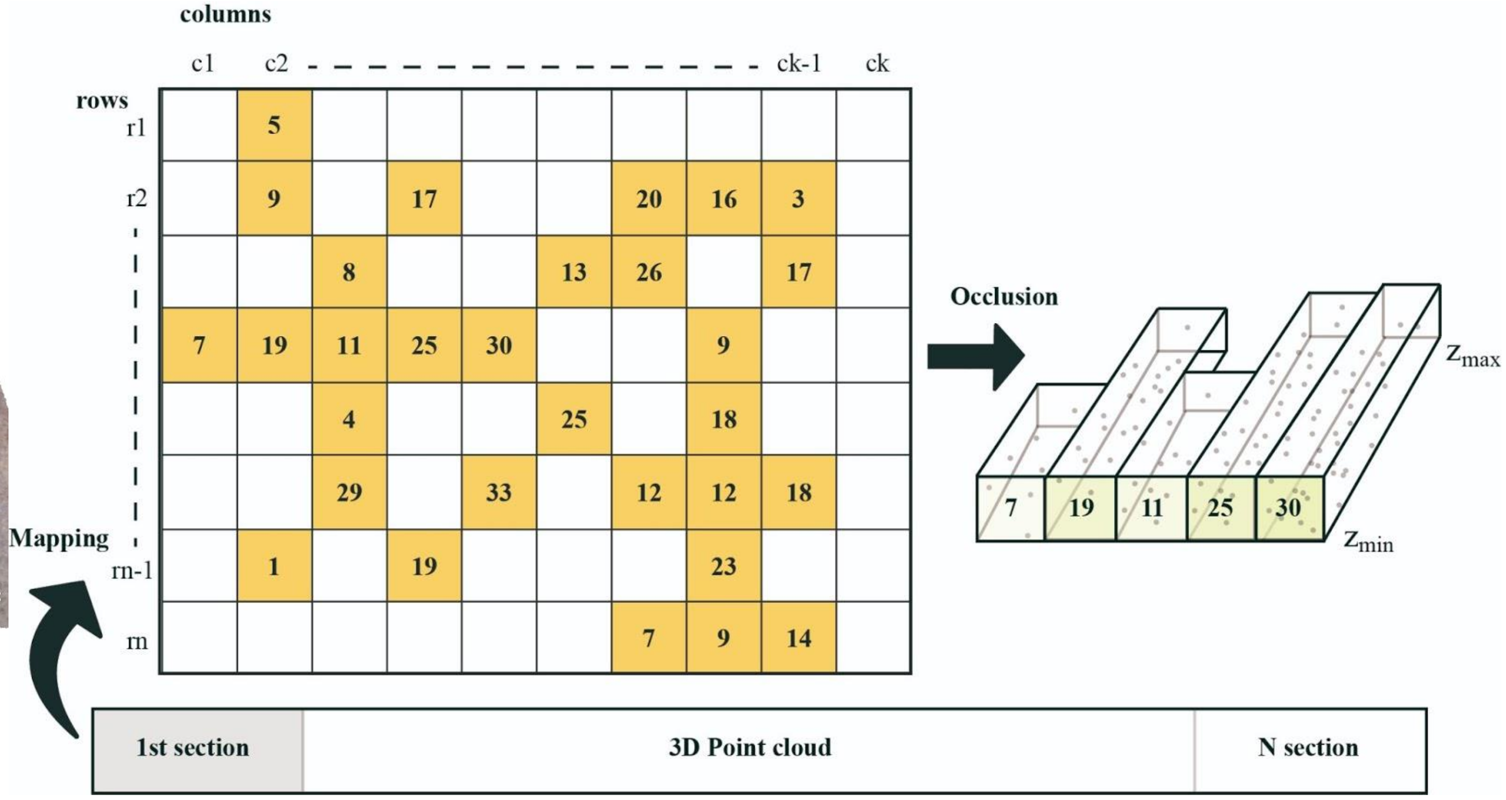




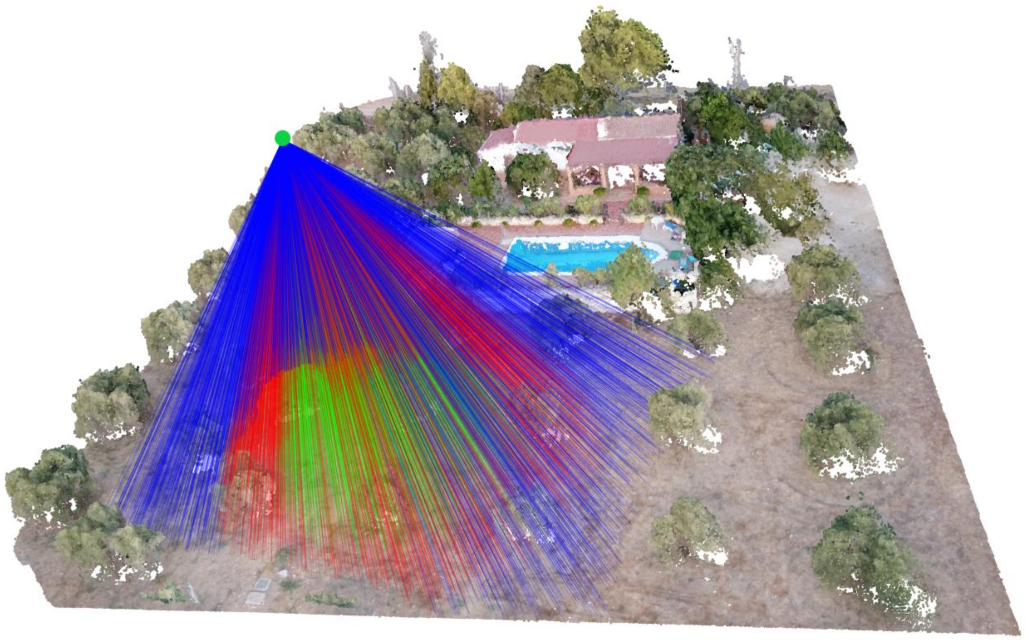
5. Multispectral image mapping

4. Occlusion

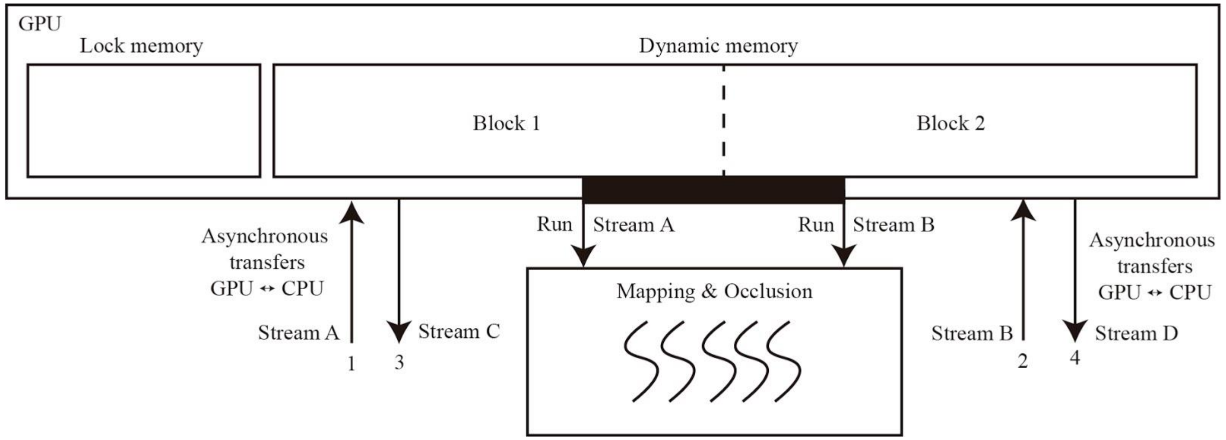


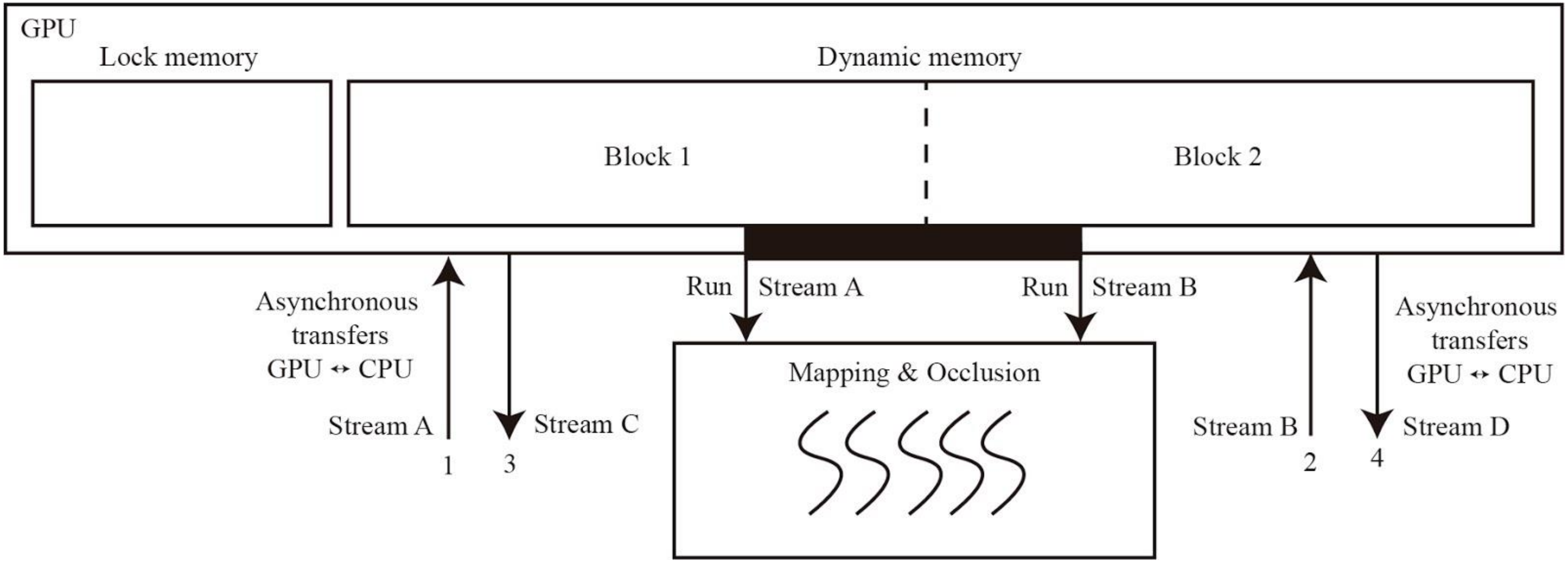


6. Multispectral image mapping



5. GPU-based acceleration using CUDA for mapping and occlusion test.





Workflow of the proposed out-of-core method for both operations: 3D mapping and occlusion test.

Test machine:

- **Procesador**: Intel con 4 núcleos (cores) i7-4790 CPU @ 3.60GHz (hyperthreading activado con 8 cores virtuales).
- **Memoria RAM**: 24 GB, Caché L1: 32 KB para datos y 32 KB para instrucciones, Caché L2: 256 KB, Caché L3: 8 MB.
- **GPU**: Nvidia TITAN V (Nvidia Driver: 450.57) con 5120 núcleos y VRAM de 12 GB.

Test dataset:

- Point cloud: 66 M. of points
- Images: 12

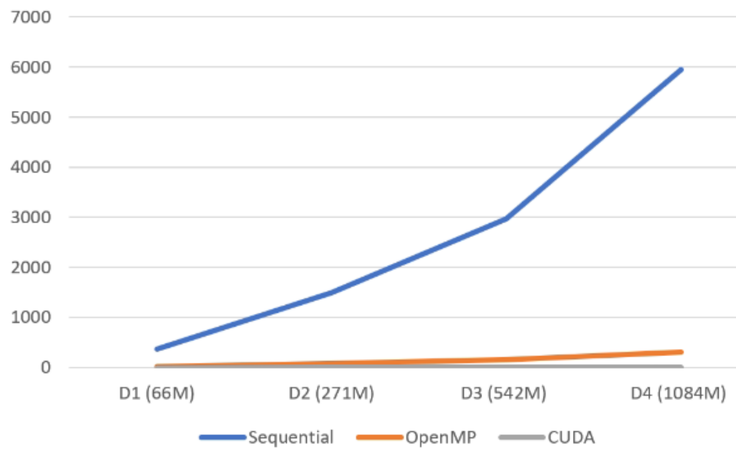
Test:

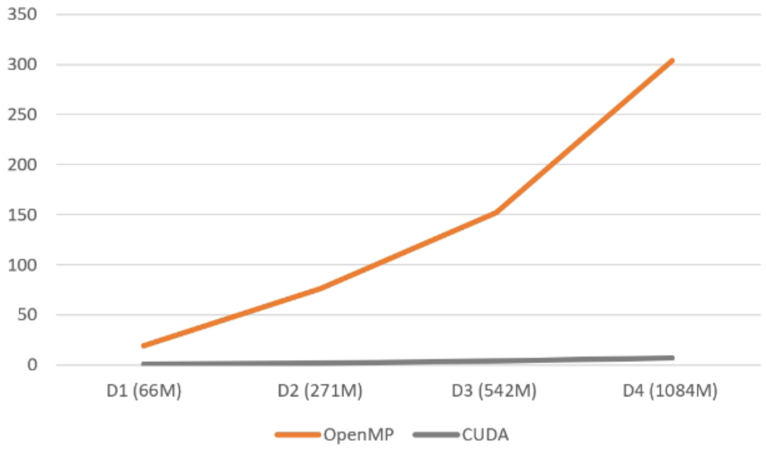
Table 5
CPU baseline in GEU (fastest CPU, all times in seconds) where F1 is the first flight (180 images) and F2 is the second flight (1350 images).

P3-B - AMD Ryzen Threadripper 1950X 16 cores (32 SMT)				
		Sequential	OpenMP 32 thrs	Out-of-core CUDA
F1	D1 (66M)	363.9	19.3	0.64
	D2 (271M)	1492.8	76.4	1.87
	D3 (542M)	2973.6	152.0	3.71
	D4 (1084M)	5950.0	303.6	6.85
F2	D1 (66M)	2667.1	142.3	4.46
	D2 (271M)	10829.7	562.2	12.67
	D3 (542M)	21665.7	1122.9	25.27
	D4 (1084M)	43309.8	2233.6	49.99

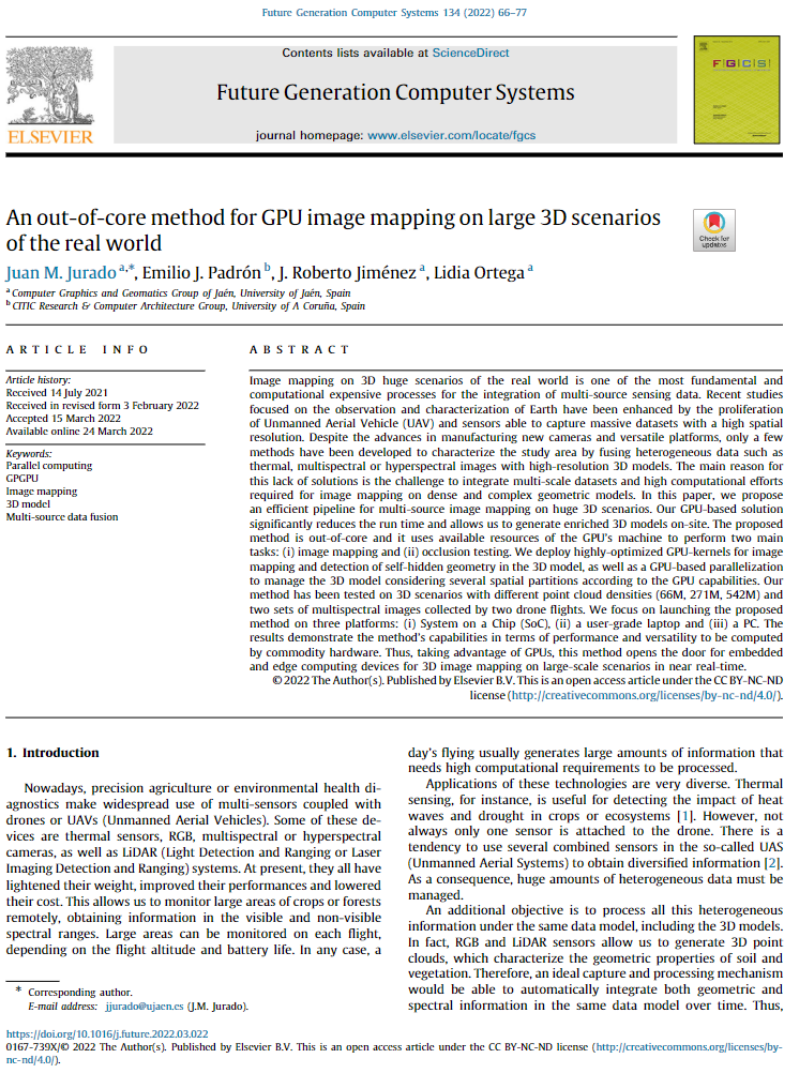
Table 3
Execution time for mapping one image on the 3D model using the Platform P1.

Datasets	Time (ms) per image on average
D1 (66M)	60
D2 (271M)	237
D3 (542M)	474
D4 (1084M, $P_k = 271M$)	237





We observe that the problem is better suited to the GPU architecture and achieves more than 30 times better results than the parallel CPU solution.



Future Generation Computer Systems 134 (2022) 66–77

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

ELSEVIER

An out-of-core method for GPU image mapping on large 3D scenarios of the real world

Juan M. Jurado [a,*], Emilio J. Padrón [b], J. Roberto Jiménez [a], Lidia Ortega [a]

[a] Computer Graphics and Geomatics Group of Jaén, University of Jaén, Spain
[b] CITIC Research & Computer Architecture Group, University of A Coruña, Spain

ARTICLE INFO

Article history:
Received 14 July 2021
Received in revised form 3 February 2022
Accepted 15 March 2022
Available online 24 March 2022

Keywords:
Parallel computing
GPGPU
Image mapping
3D model
Multi-source data fusion

ABSTRACT

Image mapping on 3D huge scenarios of the real world is one of the most fundamental and computational expensive processes for the integration of multi-source sensing data. Recent studies focused on the observation and characterization of Earth have been enhanced by the proliferation of Unmanned Aerial Vehicle (UAV) and sensors able to capture massive datasets with a high spatial resolution. Despite the advances in manufacturing new cameras and versatile platforms, only a few methods have been developed to characterize the study area by fusing heterogeneous data such as thermal, multispectral or hyperspectral images with high-resolution 3D models. The main reason for this lack of solutions is the challenge to integrate multi-scale datasets and high computational efforts required for image mapping on dense and complex geometric models. In this paper, we propose an efficient pipeline for multi-source image mapping on huge 3D scenarios. Our GPU-based solution significantly reduces the run time and allows us to generate enriched 3D models on-site. The proposed method is out-of-core and it uses available resources of the GPU's machine to perform two main tasks: (i) image mapping and (ii) occlusion testing. We deploy highly-optimized GPU-kernels for image mapping and detection of self-hidden geometry in the 3D model, as well as a GPU-based parallelization to manage the 3D model considering several spatial partitions according to the GPU capabilities. Our method has been tested on 3D scenarios with different point cloud densities (66M, 271M, 542M) and two sets of multispectral images collected by two drone flights. We focus on launching the proposed method on three platforms: (i) System on a Chip (SoC), (ii) a user-grade laptop and (iii) a PC. The results demonstrate the method's capabilities in terms of performance and versatility to be computed by commodity hardware. Thus, taking advantage of GPUs, this method opens the door for embedded and edge computing devices for 3D image mapping on large-scale scenarios in near real-time.

1. Introduction

Nowadays, precision agriculture or environmental health diagnostics make widespread use of multi-sensors coupled with drones or UAVs (Unmanned Aerial Vehicles). Some of these devices are thermal sensors, RGB, multispectral or hyperspectral cameras, as well as LiDAR (Light Detection and Ranging or Laser Imaging Detection and Ranging) systems. At present, they all have lightened their weight, improved their performances and lowered their cost. This allows us to monitor large areas of crops or forests remotely, obtaining information in the visible and non-visible spectral ranges. Large areas can be monitored on each flight, depending on the flight altitude and battery life. In any case, a day's flying usually generates large amounts of information that needs high computational requirements to be processed.

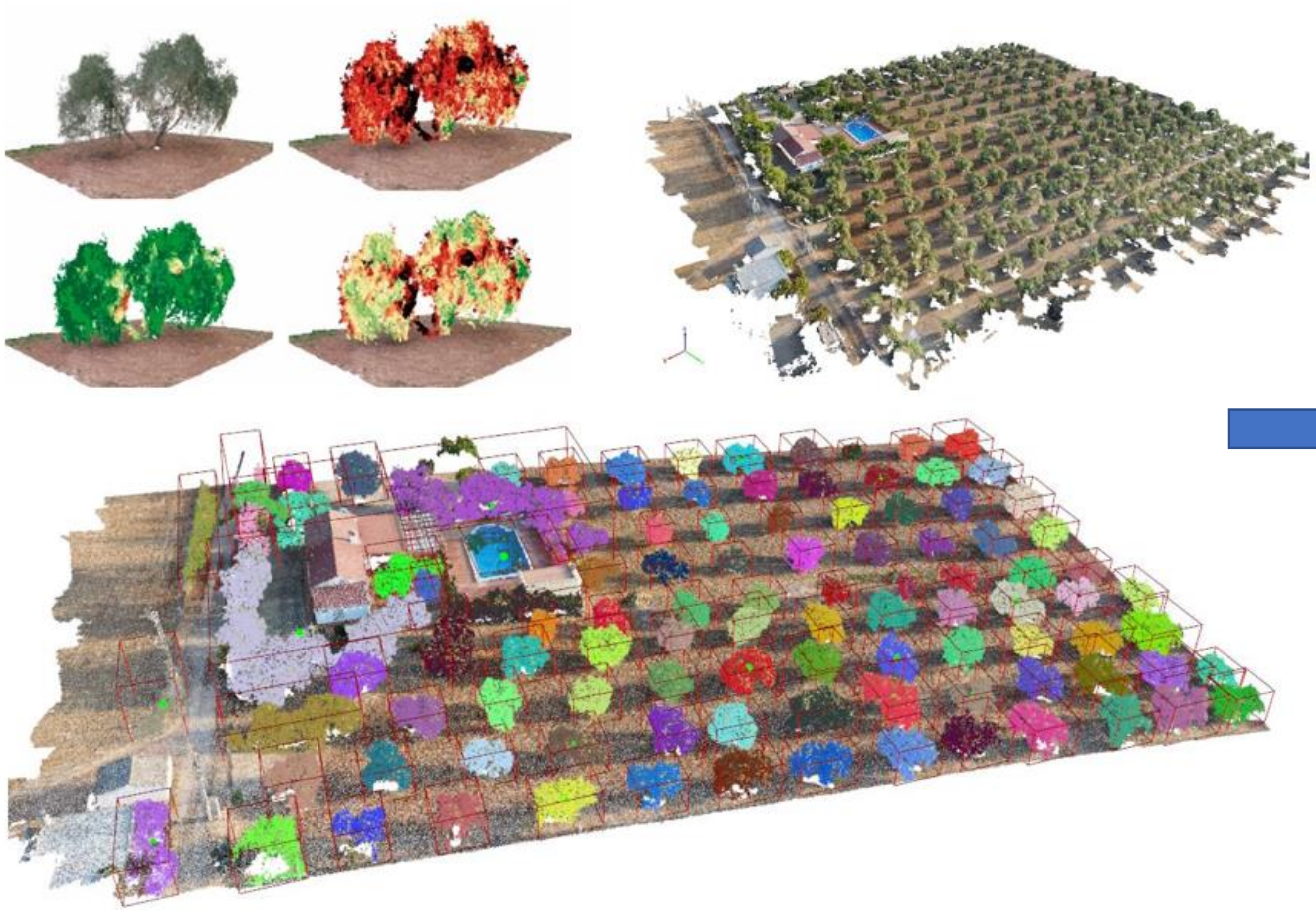
Applications of these technologies are very diverse. Thermal sensing, for instance, is useful for detecting the impact of heat waves and drought in crops or ecosystems [1]. However, not always only one sensor is attached to the drone. There is a tendency to use several combined sensors in the so-called UAS (Unmanned Aerial Systems) to obtain diversified information [2]. As a consequence, huge amounts of heterogeneous data must be managed.

An additional objective is to process all this heterogeneous information under the same data model, including the 3D models. In fact, RGB and LiDAR sensors allow us to generate 3D point clouds, which characterize the geometric properties of soil and vegetation. Therefore, an ideal capture and processing mechanism would be able to automatically integrate both geometric and spectral information in the same data model over time. Thus,

* Corresponding author.
E-mail address: jjurado@ujaen.es (J.M. Jurado).

https://doi.org/10.1016/j.future.2022.03.022

Applications

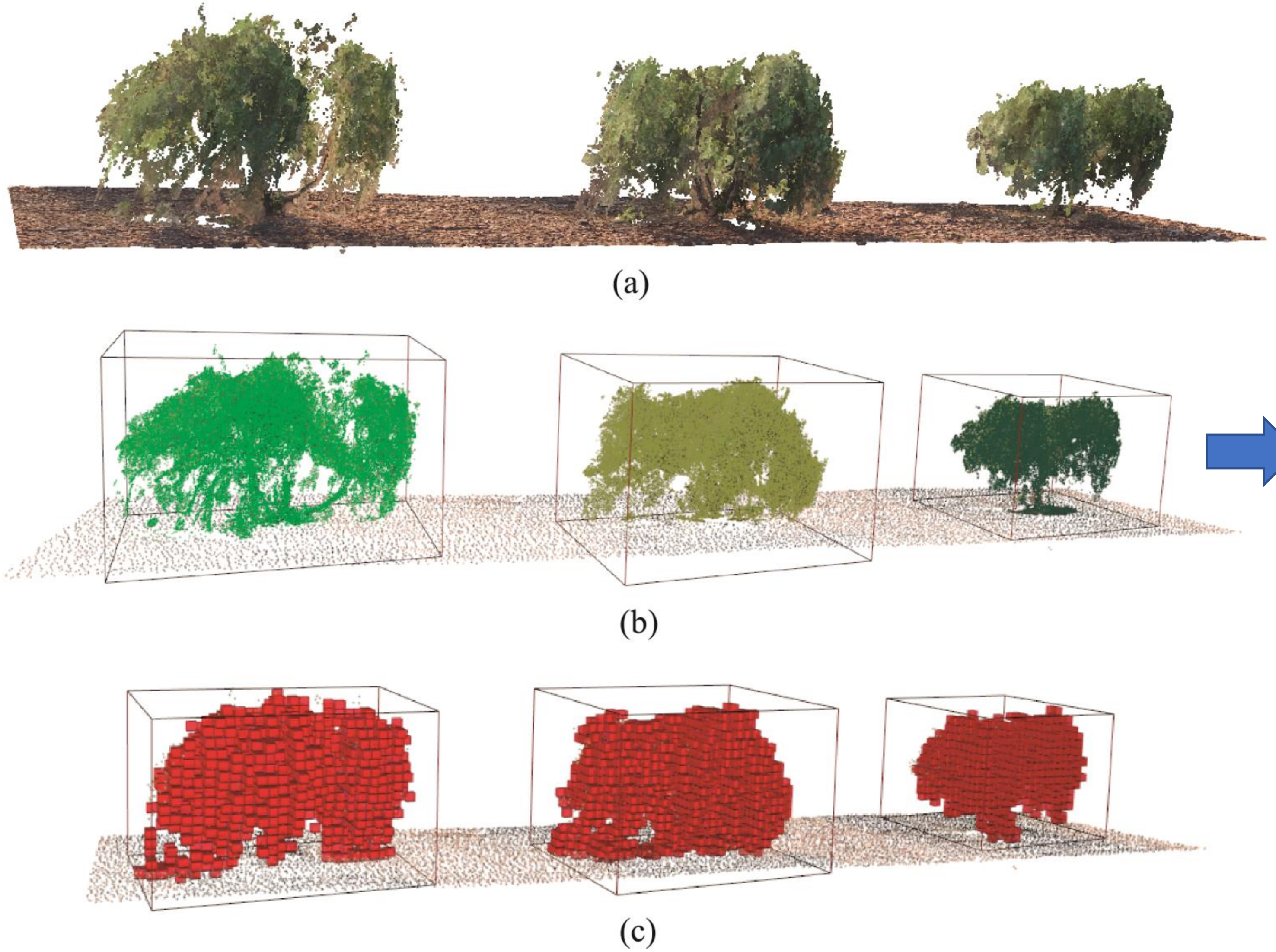


1- Precision agriculture

2- Semantic segmentation

3- Geometric segmentation

Applications

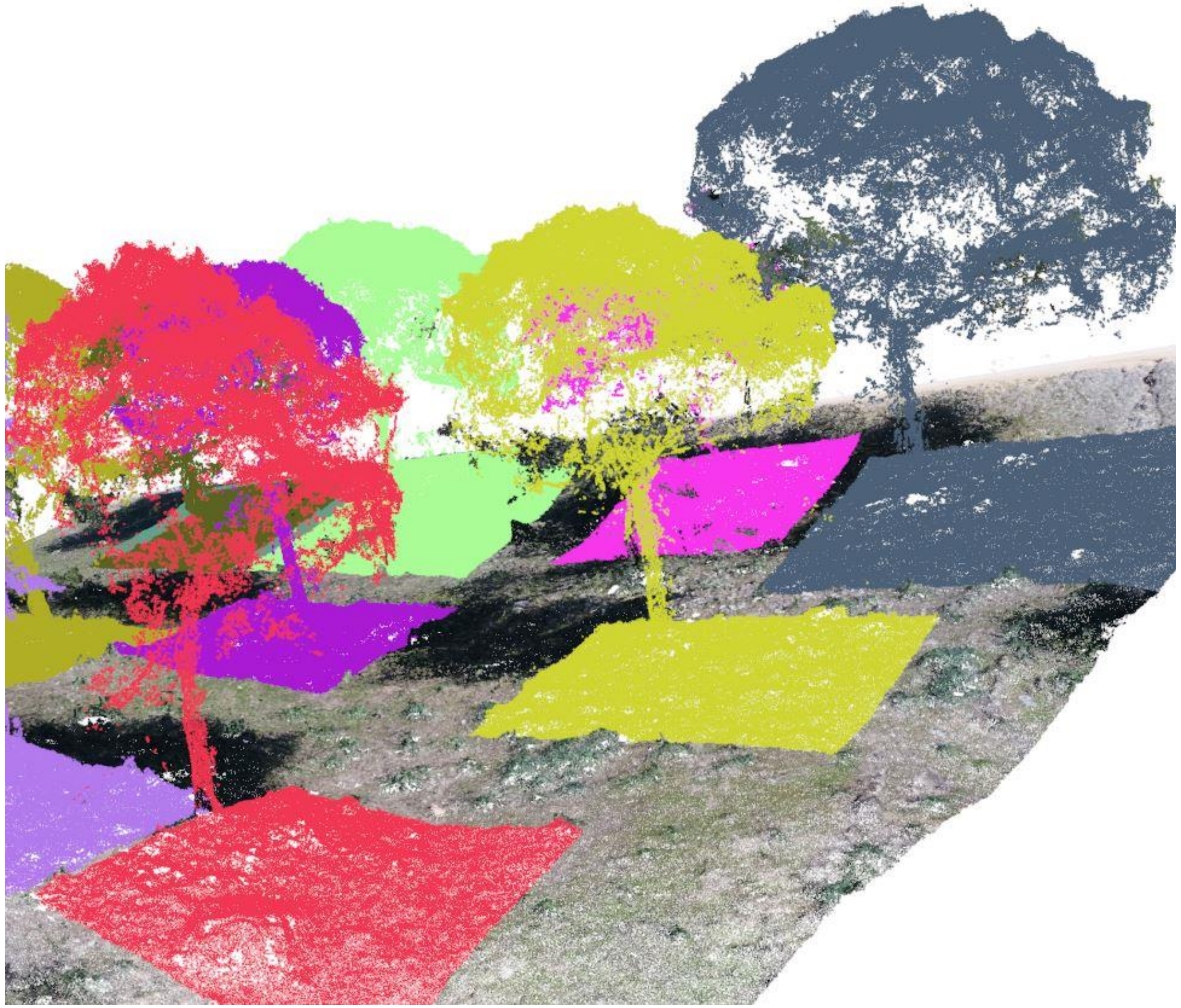


1- Morphological characterization of trees

2- Volume computation

3- Spatial data structures

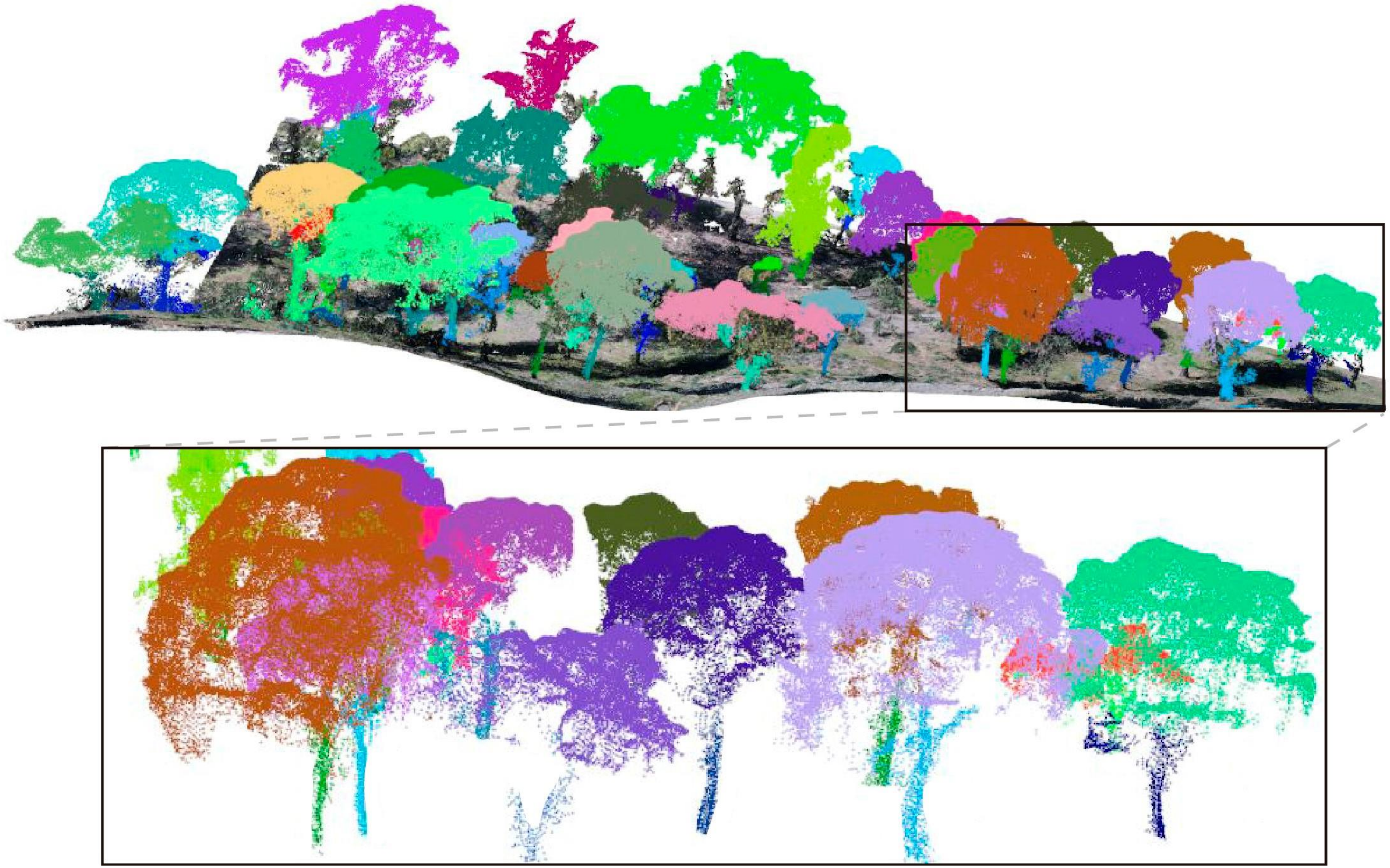
Applications



1- Species recognition

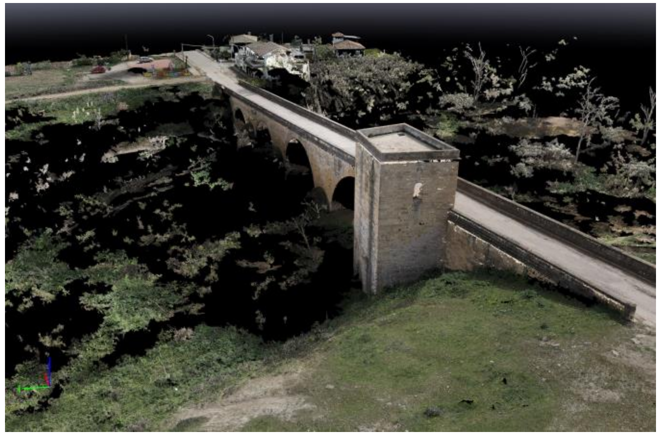
2- Forest inventory

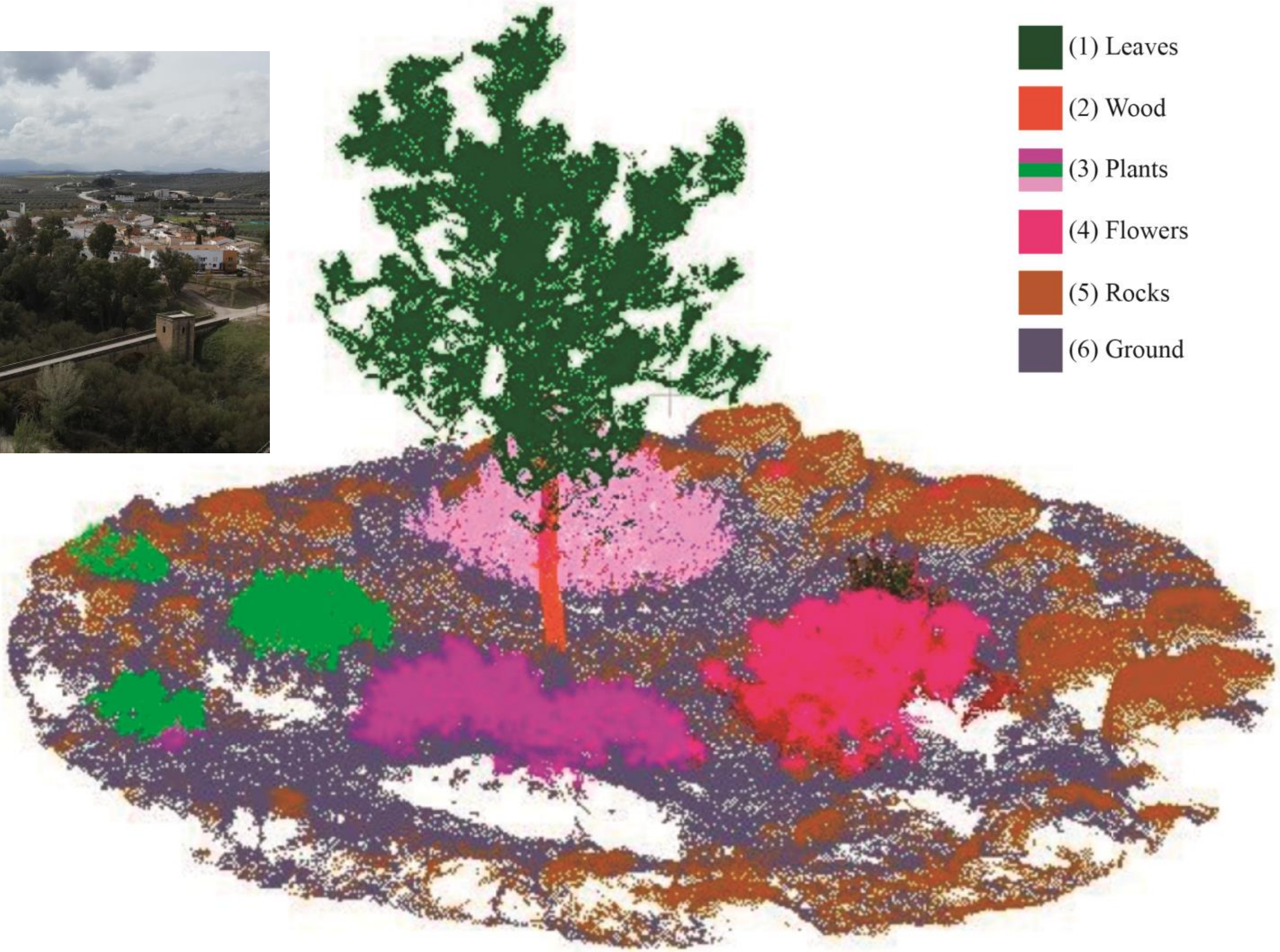
Applications



Applications

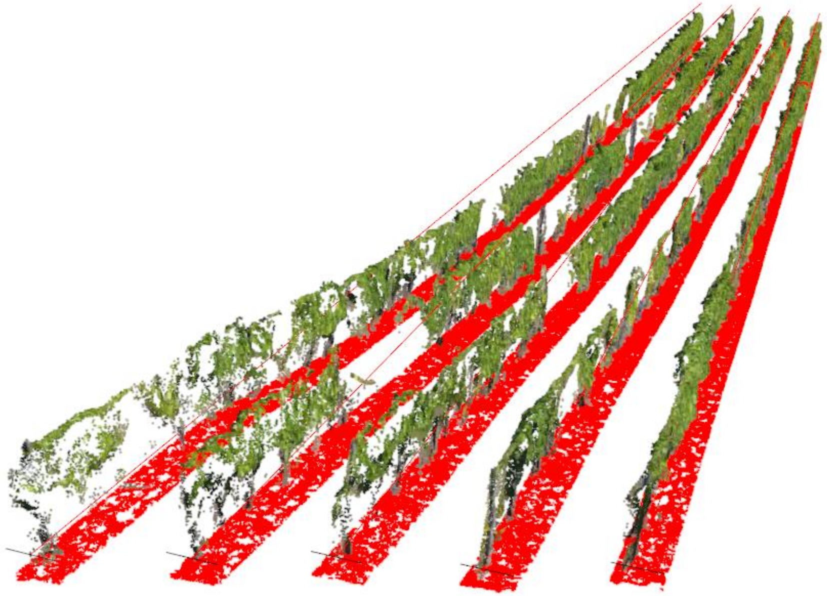
Material segmentation

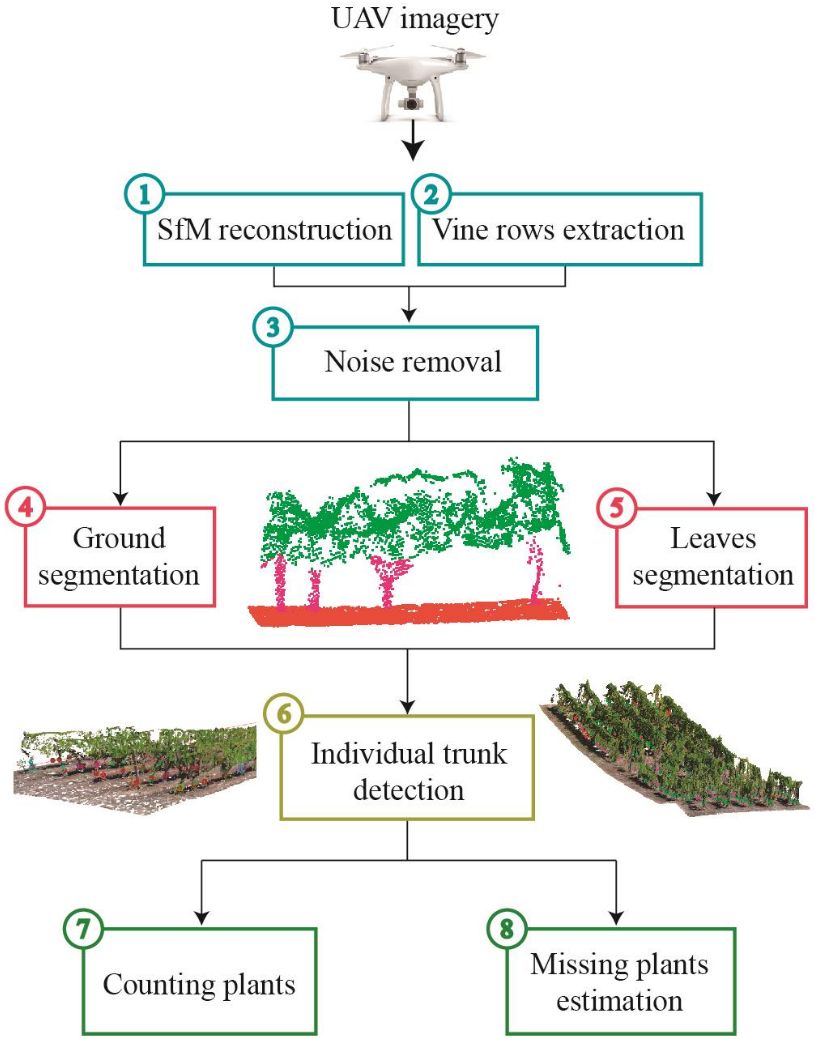


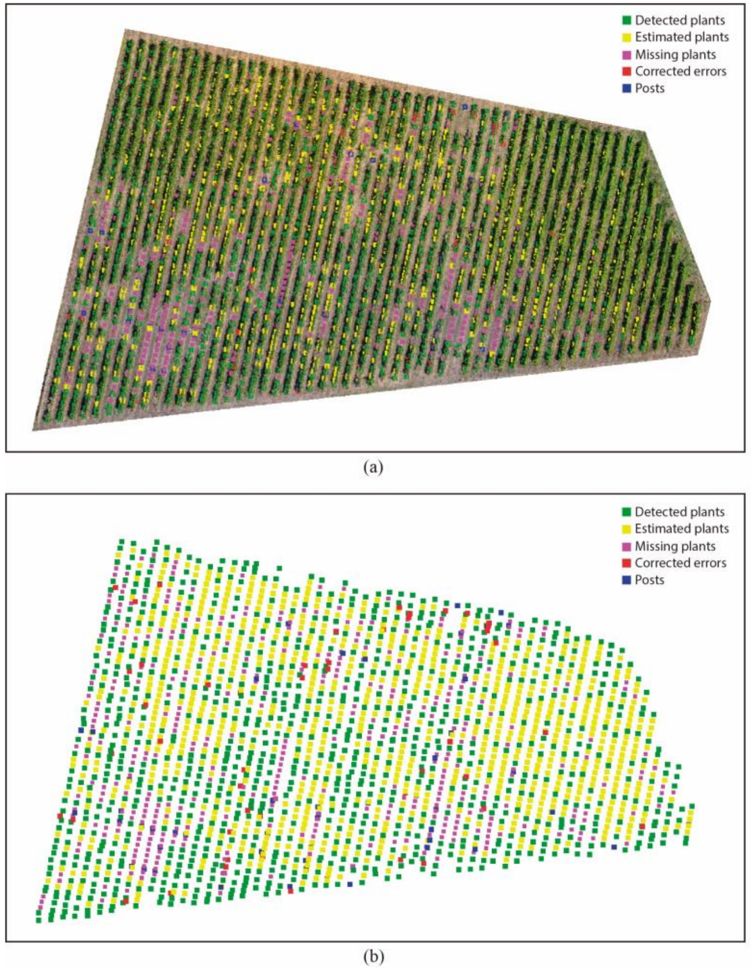


Applications

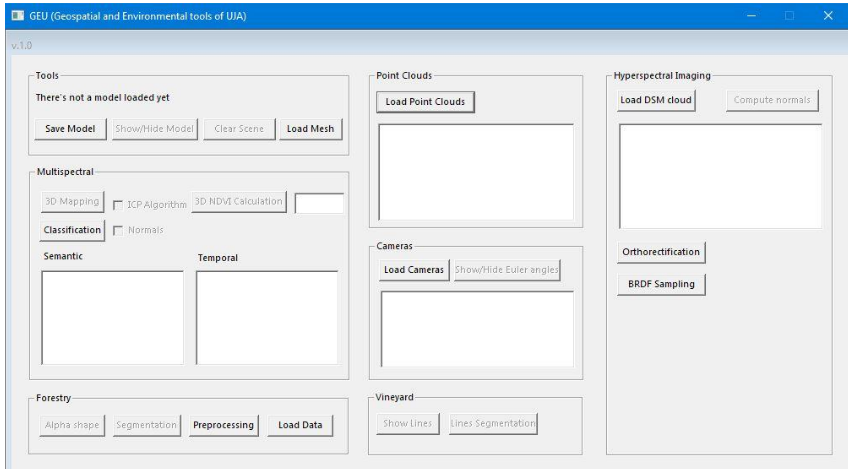
Agriculture

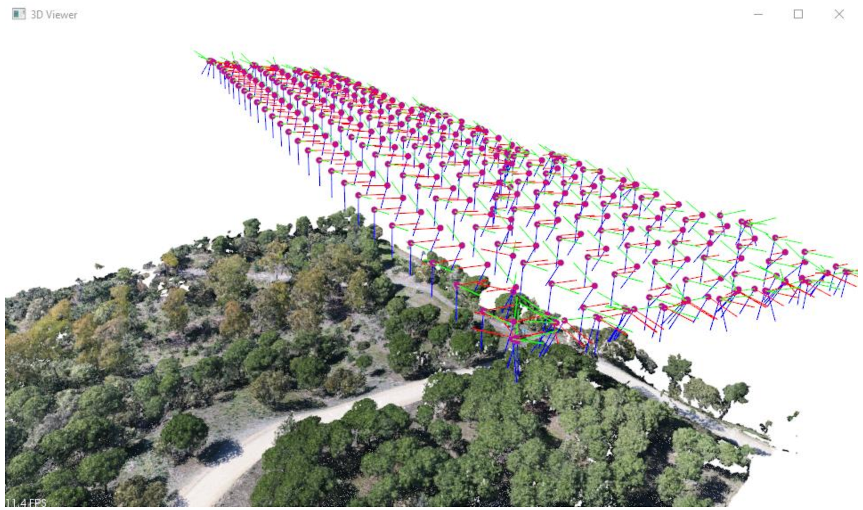


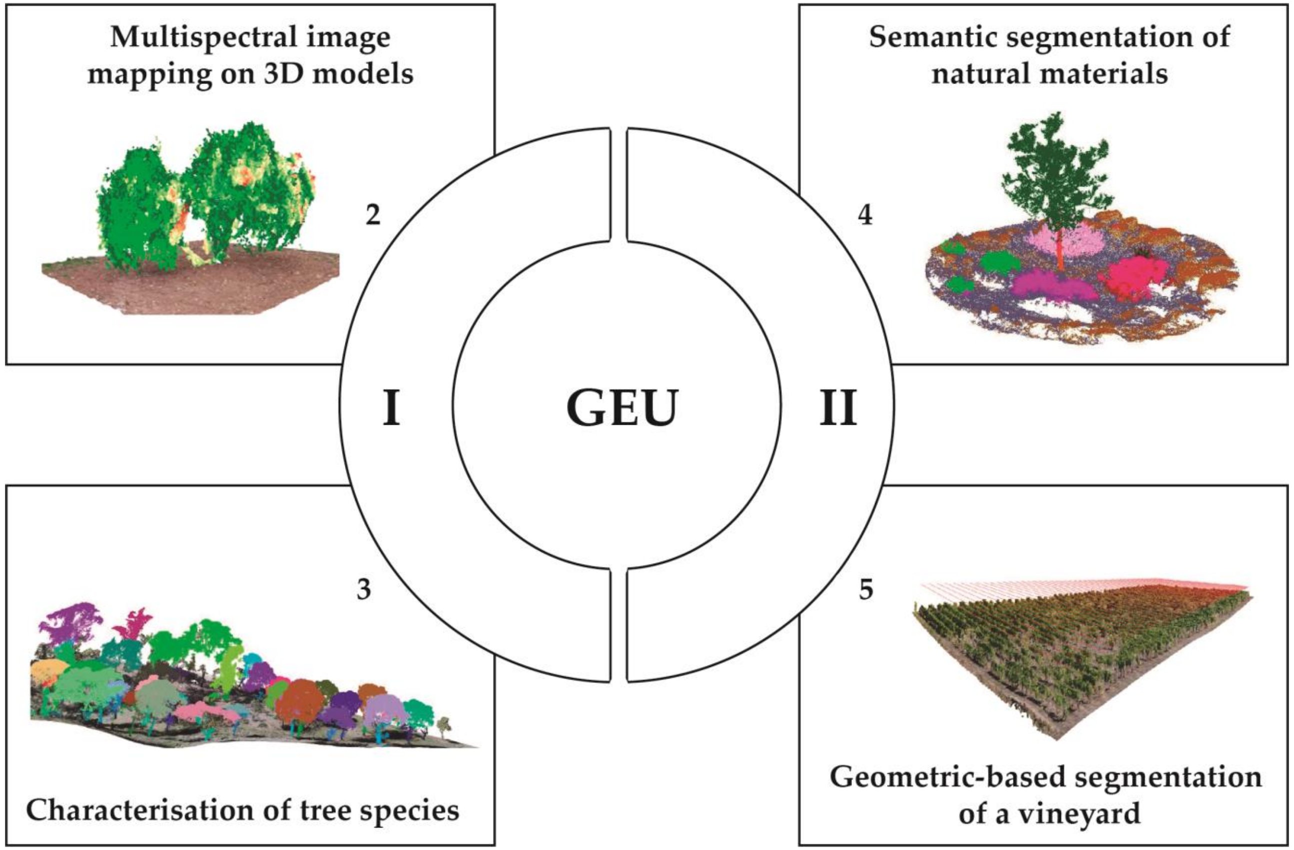




GEU (Geospatial and Environmental tools of UJA)



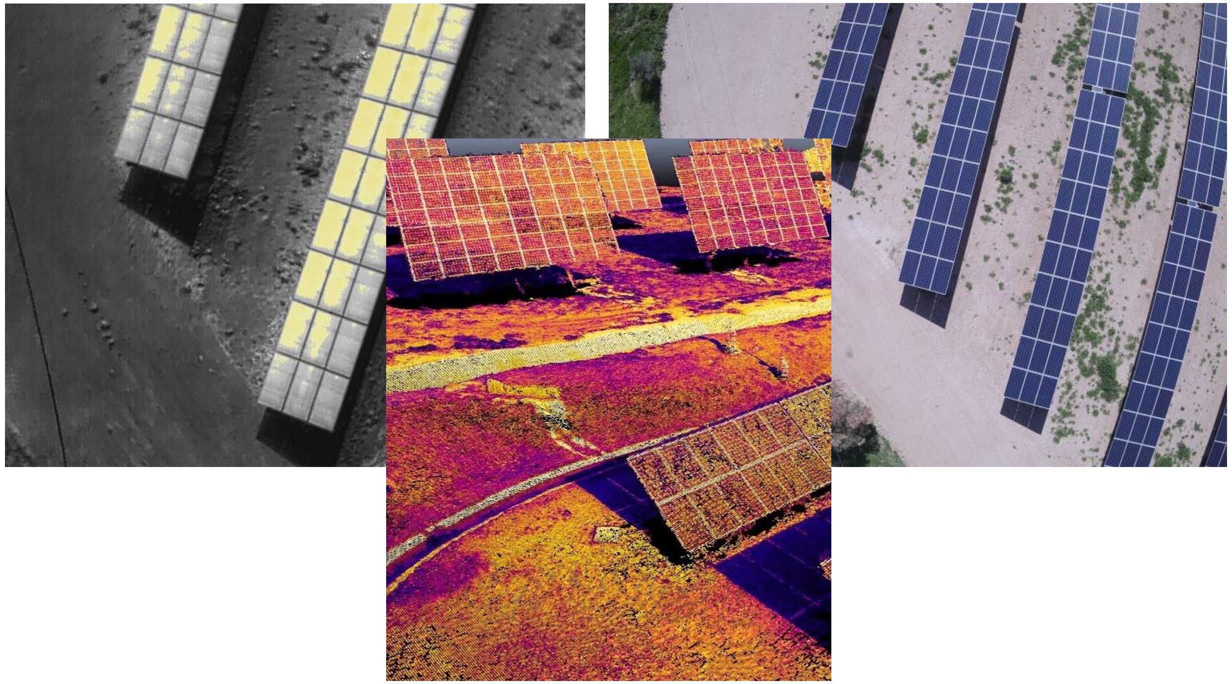




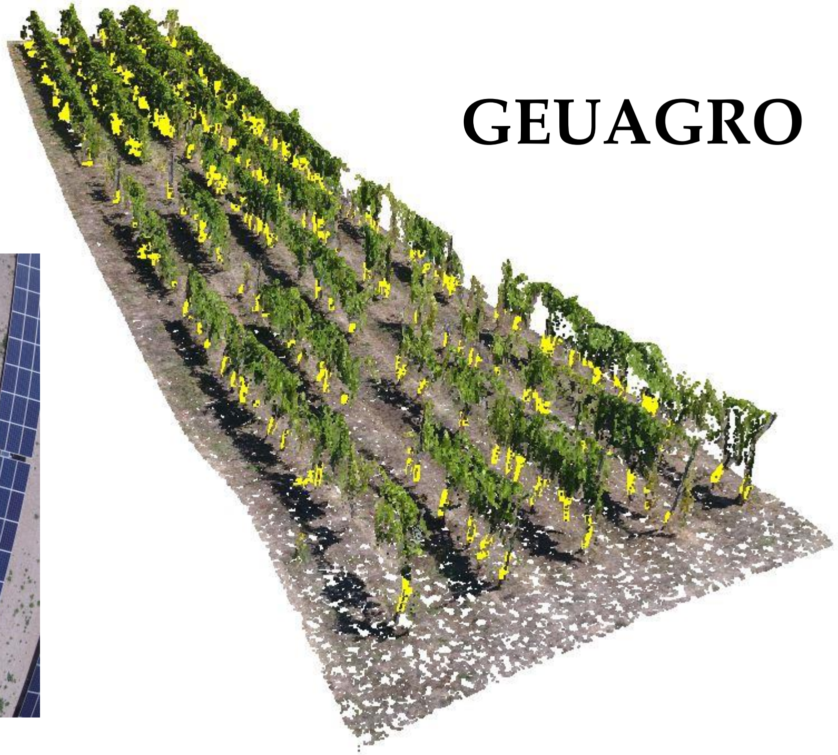
Applications

Our spin-off

GEUSOL



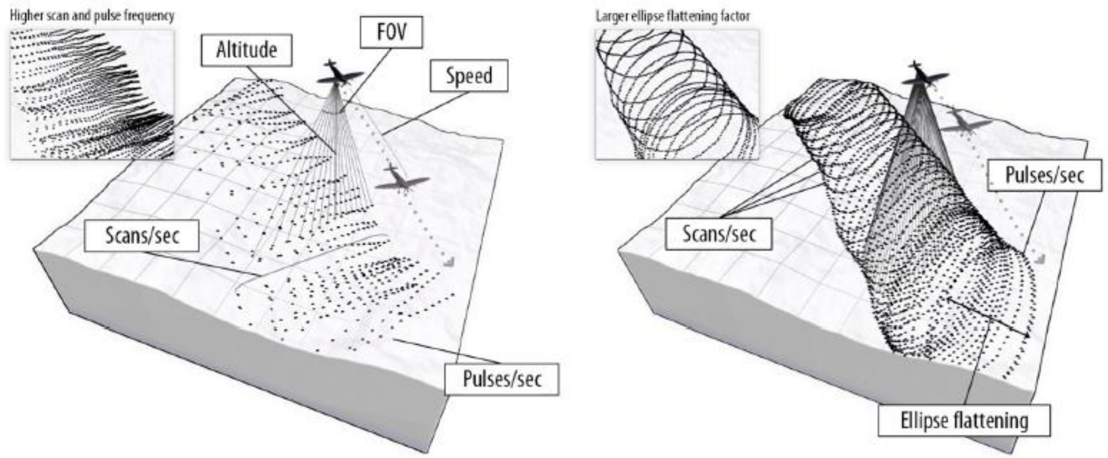
GEUAGRO

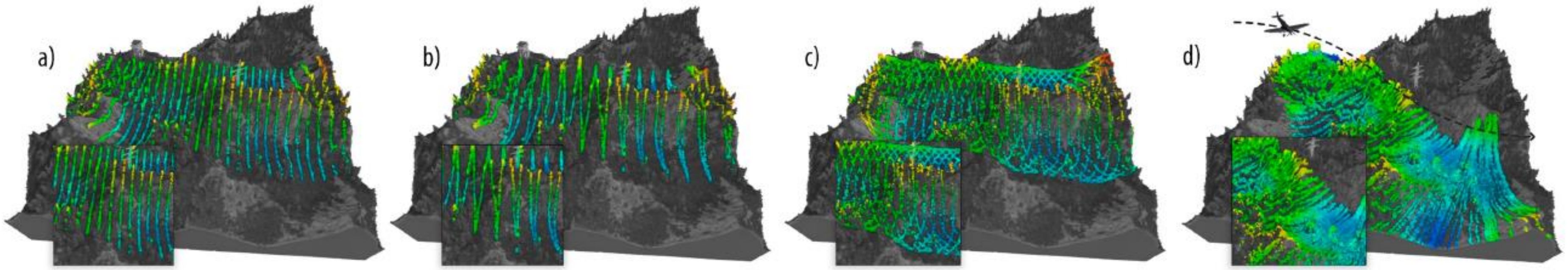


intelig∀

Trabajo futuro

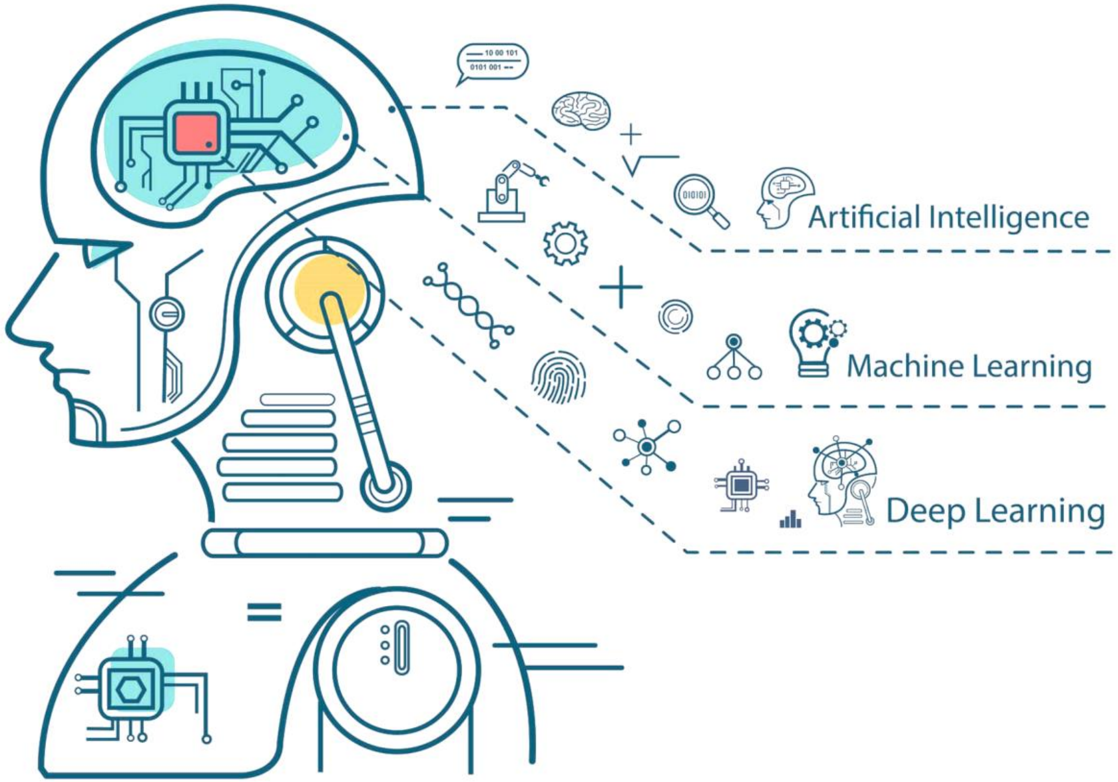
Generación de escenarios sintéticos

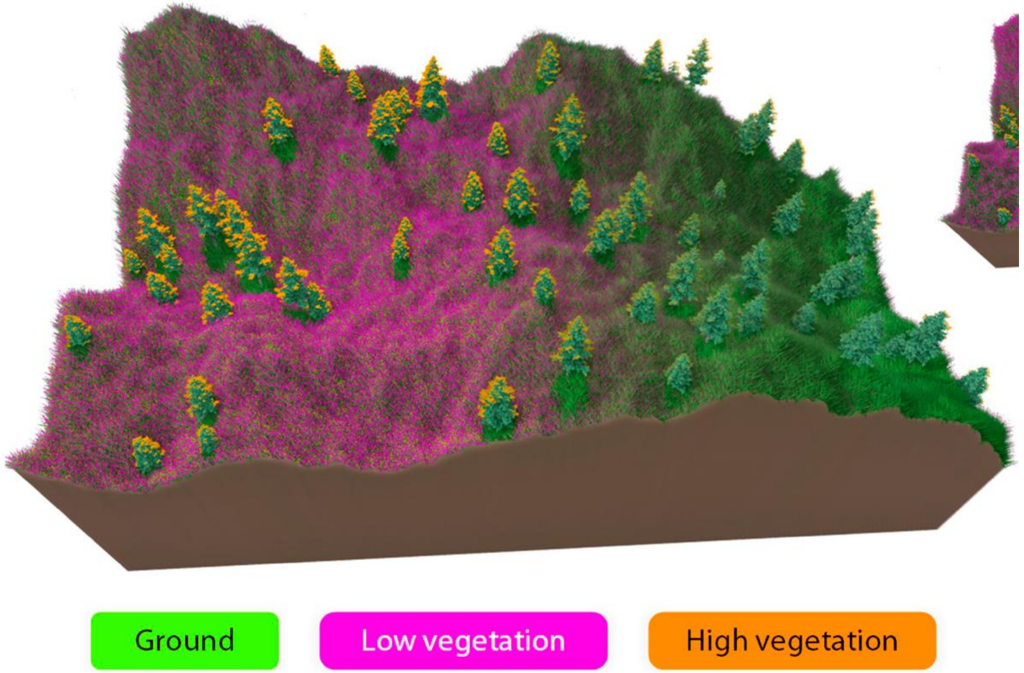




Trabajo futuro

Generación de escenarios sintéticos

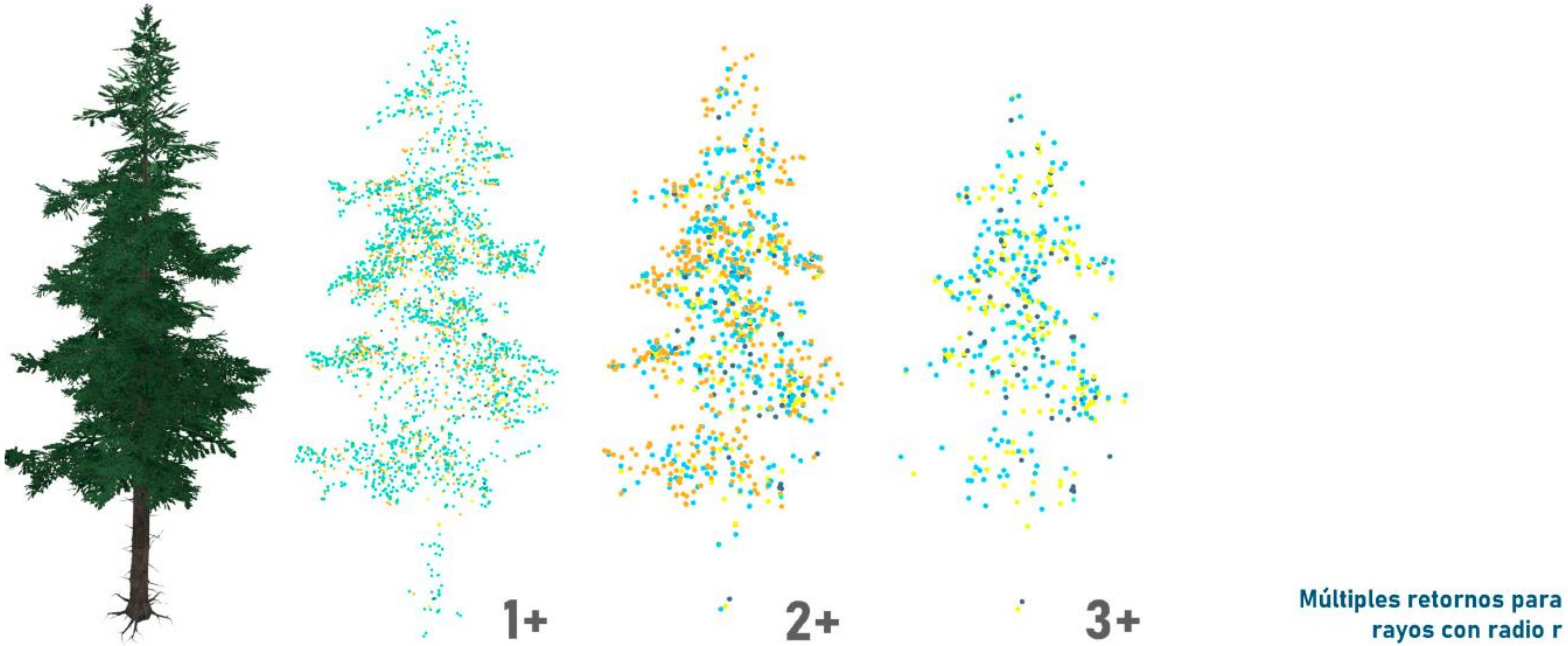




Trabajo futuro

Simulación LiDAR

A. López, C. J. Ogayar, J. M. Jurado and F. R. Feito, "A GPU-Accelerated Framework for Simulating LiDAR Scanning," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-18, 2022, Art no. 3000518, doi: 10.1109/TGRS.2022.3165746.



Trabajo futuro

Eurographics 2022

OVERVIEW

In this study, we propose a method to reconstruct real-world environments based on LiDAR data, thus overcoming density limitations and generating rich environments with ground and high vegetation.

Additionally, our proposal segments the original data to distinguish among different kinds of trees. The results show that the method is capable of generating realistic environments with the chosen density and including specimens of the identified tree types.

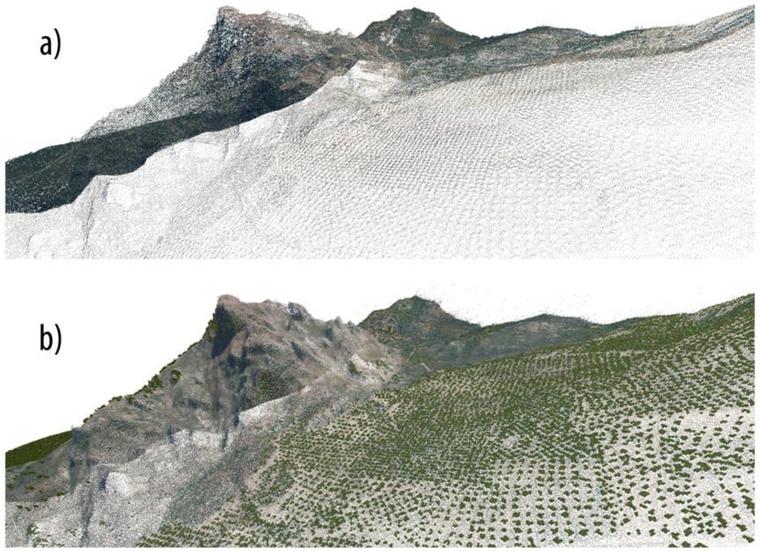
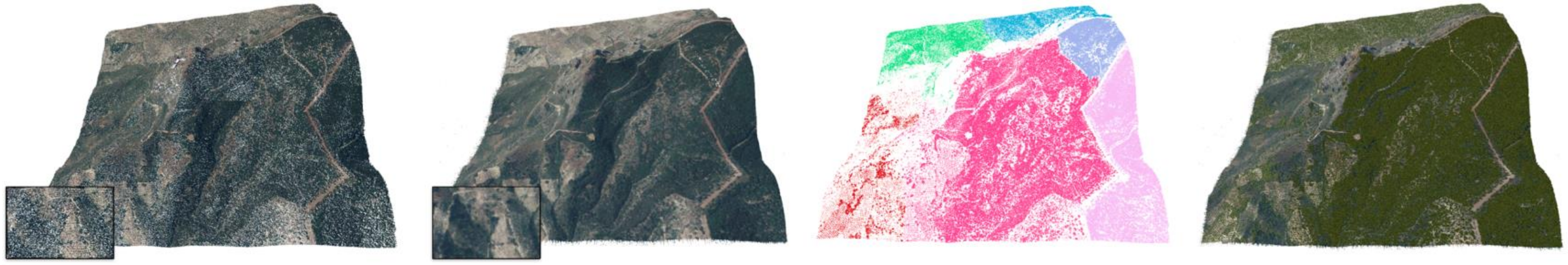
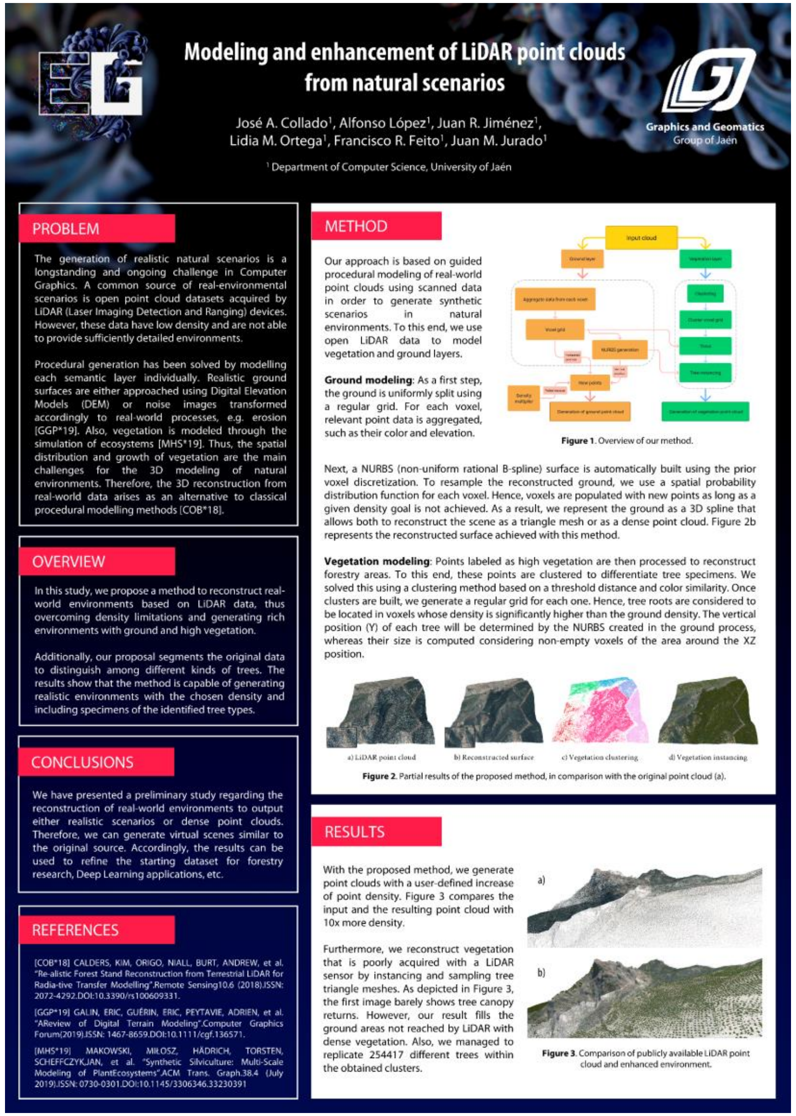


Figure 3. Comparison of publicly available LiDAR point cloud and enhanced environment.



a) LiDAR point cloud   b) Reconstructed surface   c) Vegetation clustering   d) Vegetation instancing



Modeling and enhancement of LiDAR point clouds from natural scenarios

José A. Collado[1], Alfonso López[1], Juan R. Jiménez[1], Lidia M. Ortega[1], Francisco R. Feito[1], Juan M. Jurado[1]

[1] Department of Computer Science, University of Jaén

Graphics and Geomatics Group of Jaén

PROBLEM

The generation of realistic natural scenarios is a longstanding and ongoing challenge in Computer Graphics. A common source of real-environmental scenarios is open point cloud datasets acquired by LiDAR (Laser Imaging Detection and Ranging) devices. However, these data have low density and are not able to provide sufficiently detailed environments.

Procedural generation has been solved by modelling each semantic layer individually. Realistic ground surfaces are either approached using Digital Elevation Models (DEM) or noise images transformed accordingly to real-world processes, e.g. erosion [GGP*19]. Also, vegetation is modeled through the simulation of ecosystems [MHS*19]. Thus, the spatial distribution and growth of vegetation are the main challenges for the 3D modeling of natural environments. Therefore, the 3D reconstruction from real-world data arises as an alternative to classical procedural modelling methods [COB*18].

OVERVIEW

In this study, we propose a method to reconstruct real-world environments based on LiDAR data, thus overcoming density limitations and generating rich environments with ground and high vegetation.

Additionally, our proposal segments the original data to distinguish among different kinds of trees. The results show that the method is capable of generating realistic environments with the chosen density and including specimens of the identified tree types.

CONCLUSIONS

We have presented a preliminary study regarding the reconstruction of real-world environments to output either realistic scenarios or dense point clouds. Therefore, we can generate virtual scenes similar to the original source. Accordingly, the results can be used to refine the starting dataset for forestry research, Deep Learning applications, etc.

REFERENCES

[COB*18] CALDERS, KIM, ORIGO, NIALL, BURT, ANDREW, et al. "Re-alistic Forest Stand Reconstruction from Terrestrial LiDAR for Radia-tive Transfer Modelling".Remote Sensing10.6 (2018).ISSN: 2072-4292.DOI:10.3390/rs100609331.

[GGP*19] GALIN, ERIC, GUÉRIN, ERIC, PEYTAVIE, ADRIEN, et al. "AReview of Digital Terrain Modeling".Computer Graphics Forum(2019).ISSN: 1467-8659.DOI:10.1111/cgf.136571.

[MHS*19] MAKOWSKI, MIŁOSZ, HÄDRICH, TORSTEN, SCHEFFCZYKJAN, et al. "Synthetic Silviculture: Multi-Scale Modeling of PlantEcosystems".ACM Trans. Graph.38.4 (July 2019).ISSN: 0730-0301.DOI:10.1145/3306346.33230391

METHOD

Our approach is based on guided procedural modeling of real-world point clouds using scanned data in order to generate synthetic scenarios in natural environments. To this end, we use open LiDAR data to model vegetation and ground layers.

Ground modeling: As a first step, the ground is uniformly split using a regular grid. For each voxel, relevant point data is aggregated, such as their color and elevation.

Figure 1. Overview of our method.

Next, a NURBS (non-uniform rational B-spline) surface is automatically built using the prior voxel discretization. To resample the reconstructed ground, we use a spatial probability distribution function for each voxel. Hence, voxels are populated with new points as long as a given density goal is not achieved. As a result, we represent the ground as a 3D spline that allows both to reconstruct the scene as a triangle mesh or as a dense point cloud. Figure 2b represents the reconstructed surface achieved with this method.

Vegetation modeling: Points labeled as high vegetation are then processed to reconstruct forestry areas. To this end, these points are clustered to differentiate tree specimens. We solved this using a clustering method based on a threshold distance and color similarity. Once clusters are built, we generate a regular grid for each one. Hence, tree roots are considered to be located in voxels whose density is significantly higher than the ground density. The vertical position (Y) of each tree will be determined by the NURBS created in the ground process, whereas their size is computed considering non-empty voxels of the area around the XZ position.

a) LiDAR point cloud   b) Reconstructed surface   c) Vegetation clustering   d) Vegetation instancing

Figure 2. Partial results of the proposed method, in comparison with the original point cloud (a).

RESULTS

With the proposed method, we generate point clouds with a user-defined increase of point density. Figure 3 compares the input and the resulting point cloud with 10x more density.

Furthermore, we reconstruct vegetation that is poorly acquired with a LiDAR sensor by instancing and sampling tree triangle meshes. As depicted in Figure 3, the first image barely shows tree canopy returns. However, our result fills the ground areas not reached by LiDAR with dense vegetation. Also, we managed to replicate 254417 different trees within the obtained clusters.

a)
b)

Figure 3. Comparison of publicly available LiDAR point cloud and enhanced environment.

Thanks you for your attention!

jjurado@gmail.com

https://www.researchgate.net/profile/Juan-M-Jurado/publications