Computers and Electronics in Agriculture 207 (2023) 107712

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Efficient generation of occlusion-aware multispectral and thermographic point clouds

Alfonso López [a,*], Carlos J. Ogayar [a], Juan M. Jurado [b], Francisco R. Feito [a]

[a] Department of Computer Science, University of Jaén, Spain
[b] Department of Software Engineering, University of Granada, Spain

ARTICLE INFO

Keywords:
Multispectral
Thermography
3D point cloud
GPGPU
UAV

ABSTRACT

The reconstruction of 3D point clouds from image datasets is a time-consuming task that has been frequently solved by performing photogrammetric techniques on every data source. This work presents an approach to efficiently build large and dense point clouds from co-acquired images. In our case study, the sensors co-acquire visible as well as thermal and multispectral imagery. Hence, RGB point clouds are reconstructed with traditional methods, whereas the rest of the data sources with lower resolution and less identifiable features are projected into the first one, i.e., the most complete and dense. To this end, the mapping process is accelerated using the Graphics Processing Unit (GPU) and multi-threading in the CPU (Central Processing Unit). The accurate colour aggregation in 3D points is guaranteed by taking into account the occlusion of foreground surfaces. Accordingly, our solution is shown to reconstruct much more dense point clouds than notable commercial software (286% on average), e.g., Pix4Dmapper and Agisoft Metashape, in much less time (−70% on average with respect to the best alternative).

1. Introduction

Remote Sensing (RS) research concerns multiple acquisition devices and multispectral data obtained in different wavelength ranges. These are frequently visible, thermographic and multispectral imagery, either collapsed into a few or hundreds of bands (hyperspectral). Thus, the main challenge of multiple sensing systems is to fuse all this information into a sole data model with multiple layers. Fusion methodologies have been long investigated to build multi-layer models that allow subsequent analyses to focus only on the data. From these models, Machine Learning (Pádua et al., 2022) and Deep Learning (Jia et al., 2021; Hu et al., 2022) represent a widespread field to extract further information.

Besides image fusion, which has been greatly investigated, 3D representations of surveyed environments have been gaining interest (Jurado et al., 2022a). Firstly, interaction with 3D environments speeds up tasks for human operators, such as detecting building failures (Lin et al., 2019). Also, 3D reconstructions are much more valuable and informative for the client counterpart. Finally, it reduces the analysis of multiple datasets into a single one where comparisons are effectively performed. Among 3D representations, point clouds have been significantly preferred over mesh models as they do not involve the reconstruction of surveyed surfaces (Park and Lee, 2019) into polygons. Instead, a discretized version is provided, though its rendering has been proved to achieve nearly mesh-like results with modern hardware and large point clouds (Schütz et al., 2021).

Among other factors, the raising interest in 3D models is possible due to the use of Unmanned Aerial Vehicles (UAV) in RS. In contrast to satellite imagery, UAVs provide a cost-efficient solution with higher spatial and temporal resolution (Singh et al., 2022). Higher spatial resolution can be either achieved with lower flight altitude or devices with higher resolution. On the other hand, temporal resolution refers to the monitoring frequency of an area of interest, in the order of days and weeks for publicly available satellites (Alvarez-Vanhard et al., 2021). Nevertheless, cost-efficient vehicles and devices also lead to images with more defects and errors, either motivated by UAV instability (Akhoundi Khezrabad et al., 2022) or the optical mechanism (Mohamad et al., 2021; López et al., 2021a). Besides this, the reduced flight altitude requires wider-angle lenses, thus generating more geometric distortions. Though they can be reverted, the number of steps for data fusion increases and so does the overall complexity.

Finally, highly detailed 3D point clouds with hundreds of millions of points are complex to operate on commodity hardware. Despite this, they are necessary to provide in-depth analyses and valuable renderings (Schütz et al., 2021). As a result, the reconstruction and processing of large point clouds are computationally demanding tasks

* Corresponding author.
E-mail addresses: allopezr@ujaen.es (A. López), cogayar@ujaen.es (C.J. Ogayar), jjurado@ujaen.es (J.M. Jurado), ffeito@ujaen.es (F.R. Feito).

https://doi.org/10.1016/j.compag.2023.107712
Received 7 November 2022; Received in revised form 8 January 2023; Accepted 13 February 2023
Available online 2 March 2023

that require massively parallel algorithms and modern Graphics Processing Units (GPU) with increased capacity to operate many parallel threads. Therefore, solutions for fusing RS data are not solely aimed at providing accurate results, but also at efficiently generating them. The baseline for the reconstruction of 3D point clouds can be established using commercial software, with most of them taking advantage of GPUs to speed up the reconstruction (Jiang et al., 2020). Hence, their processing time is related both to the degree of parallelism as well as the size of input data. Accordingly, large point clouds are expected to worsen the performance, though they provide valuable data for mesh-like renderings and neighbourhood-derived processing (e.g., normal estimation).

In this work, an efficient solution is proposed for the fusion of (1) visible, (2) thermographic and (3) multispectral information, acquired by UAVs. Accordingly, we make effective the methodology proposed for image fusion in a previous work (López et al., 2021a), while also being extended to 3D. Furthermore, the proposed work is developed using massively parallel algorithms, supported by GPUs and multi-core CPUs. Commercial software is used as a baseline for evaluating the efficiency of our framework. Besides parallel processing, our approach is also focused on maximizing the size of point clouds. To this end, images with lower resolution are not directly processed to reconstruct 3D point clouds. Instead, the fusion methodology allows projecting additional sources of information over models derived from RGB estimations, which are known to generate results of higher dimensionality.

Hence, the main contributions of this work are as follows: (1) fusion of multiple imagery datasets from UAVs and (2) generation of larger point clouds than current commercial software with (3) lower response time.

2. Related work

In this section, previous work related to the fusion of UAV imagery as well as the generation and processing of large point clouds is reviewed. Then, an overview of existing commercial software is presented. Despite LiDAR being a frequently investigated 3D data source, it is omitted from this work as we solely include images. Studies that are following cited work with thermal or multispectral information, although there may be slight processing differences among them. For example, thermal imagery is known to be much noisier.

Fusion of information. The registration of multiple data sources is typically reviewed for pairs of datasets, with visible imagery being the reference. This choice is due to its higher resolution and better visibility of key points, e.g., ground control points (GCPs) distributed uniformly to map the result into a standard coordinate system. Fusion techniques are mainly classified into five categories (Jurado et al., 2022a): (1) Photogrammetric techniques plus additional alignment steps, (2) Fusion of orthomosaics and point clouds, (3) Registration of 2D and 3D features, (4) Fusion of imagery and (5) Calibration of acquisition devices.

Methods based on photogrammetric techniques are those based on the generation of 3D point clouds using Structure from Motion (SfM). To this end, GCPs can be used to ensure that point clouds are located in the same coordinate system (Zheng et al., 2020; González et al., 2019; Dahaghin et al., 2021). These can be followed by additional alignment steps, such as Iterative Closest Point (ICP) whether the resulting 3D point clouds do not correlate well (Hoegner et al., 2018; Webster et al., 2018). More advanced methods use ICP as a refinement registration, preceded by noise filtering and coarse registering of 3D point clouds. Noisy datasets can also be approached using Fast Global Registration (FGR) (Lin et al., 2019), which is proven to perform better. Instead of aligning multiple point clouds, these can be mapped into orthomosaics. However, the expected result is a 2.5D point cloud (Juszczyk et al., 2021; Adán et al., 2020), i.e., voxel-based representations.

Regarding 3D point clouds, it is also feasible to find 3D features to be recognized on imagery from another data source, thus registering both datasets. Most studies using this workaround rely on Harris 2D–3D as well as SIFT3D (Speeded-Up Robust Feature) (Zhu et al., 2021), though the first one is preferred as it is known to provide a higher number of keypoints. Due to challenges concerning the repeatability of features (for instance, building corners), the search space is limited (Lin et al., 2019) or pairing mismatches are discarded by human operators (Zhu et al., 2021). Hence, other techniques seem more robust for forestry and agriculture datasets. The alignment in the image space is performed using a wide range of techniques, with SIFT (Scale-Invariant Feature Transform), Canny and Sobel (Hoegner et al., 2016b,a) being the most popular. However, the main concern for multi-source fusion is the correlation of images from different wavelength intervals and thus depicted with different intensities. Furthermore, these differences may not be explained using linear models. Hence, methods working in the frequency domain or based on correlation are also frequent in the literature (Javadnejad et al., 2020), such as Enhanced Correlation Coefficient (ECC) (López et al., 2021a,b) and Radiance Invariant Feature Transform (RIFT) (Lin et al., 2019). With the rising impact of Deep Learning, Convolutional Neural Networks (CNN) have also been used to perform the fusion of visible and other data sources (Piao et al., 2019). Another workaround is to use the acquisition variables to project additional imagery into the reference, thus relying on the precision of parameters measured per image (Dlesk et al., 2022). Once correlated, additional sources of information can be projected into a reference point cloud (e.g., RGB) (Javadnejad et al., 2020; López et al., 2021b). However, this approach requires considering the scene occlusion to avoid assigning information from background to foreground surfaces, otherwise leading to inaccurate colour aggregations (López et al., 2021; Jurado et al., 2022b; Kong et al., 2018).

Finally, previous work has also assessed the calibration of systems composed of several sensors, thus extracting the boresight and lever-arm transformation matrices (Javadnejad et al., 2020; Hoegner et al., 2018). Hence, differences among pairs of images in datasets are explained through these matrices. However, the computation of these matrices can be preceded by manual or semi-automatic feature recognition to calibrate multi-sensor architectures. Also, the calibration of sensor acquisition has been proved to perform worse than fusing information in the image space (Javadnejad et al., 2020).

Optimized processing of 3D point clouds. The processing of large point clouds is time-consuming, and therefore, multi-core CPU and GPU hardware enable speeding up these tasks. Hence, the procedure from reading point clouds to obtaining a processed result is optimized to enhance performance and reduce memory usage, which is frequently bounded by a few gigabytes in GPU. Compression of point clouds has been previously achieved by taking advantage of stack-based and quad-tree data structures and limited colour space resolution, instead of using HDR imaging. Then, point clouds are processed using either popular multi-core CPU frameworks, such as OpenMP, or GPU hardware together with frameworks such as OpenGL, CUDA or OpenCL. Frequent processing of 3D point clouds consists of normal estimation (Sanchez et al., 2020), projection of additional sources of information (López et al., 2021, 2022; Javadnejad et al., 2020), semantic segmentation (Poux et al., 2022) and polygonal mesh estimations (Pan et al., 2021; Wiemann et al., 2018), where most of them can benefit from collaborative work among threads.

Particularly, occlusion detection is treated in this work to avoid projecting data into background surfaces. Jurado et al. (2022b) investigated the use of depth buffers to detect occluded surfaces for projecting multispectral imagery. In this regard, López et al. (2021) compared the performance of CUDA and OpenGL, with the second one offering a better performance despite requiring less configuration. Once processed, the rendering is also known to be time-consuming for large point clouds. To this end, alternative pipelines and spatial orderings of point clouds have been proposed to enhance the number of frames per second displayed to users. Schütz et al. (2021), Kerbl et al. (2017) described the use of modern OpenGL extensions and sorting based on

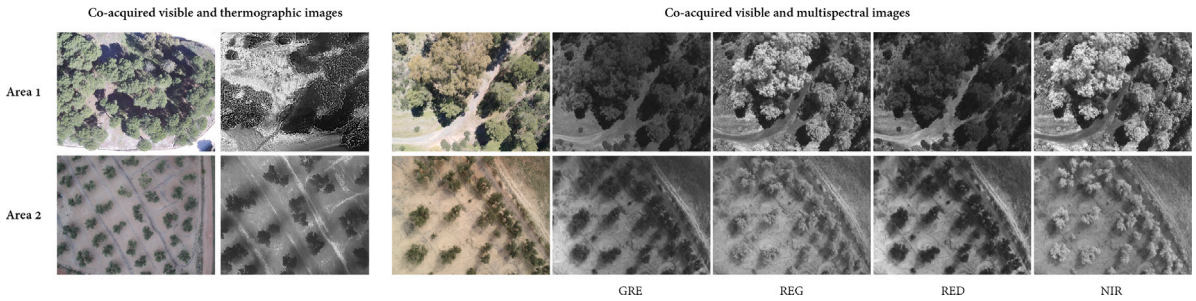


Fig. 1. Samples of thermal and multispectral imagery acquired in two different areas.

Morton codes to accelerate the visualization. Lately, point clouds have been split into batches, thus enabling batch-level optimizations (Schütz et al., 2022).

Photogrammetric commercial software. Available solutions including SfM are composed of both open-source packages and commercial solutions. The first group includes ColMap, AliceVision, Zephyr and VisualSfM, whereas notable commercial software includes Pix4DMapper (Zheng et al., 2020), Agisoft Metashape (Grechi et al., 2021), RealityCapture and Autodesk Recap 360 (Lafi et al., 2017). The first two commercial solutions are sped-up using multi-core CPU and GPU acceleration for image matching, whereas Bundle Adjustment (BA) is performed in the CPU. Jiang et al. (2020) revised the efficiency of SfM implementations by applying it over four datasets. According to their evaluation, performed using commodity hardware, Metashape showed the worst performance both for BA and feature matching, whereas RealityCapture showed similar performance during feature matching and significantly better during BA. However, Agisoft Metashape generated the densest point clouds for every dataset. On the other hand, recent work (López et al., 2021b) reports a significant improvement of Agisoft Metashape over Pix4DMapper with better hardware by taking advantage of CUDA-enabled GPUs.

3. Materials and methods

This section describes the methodology followed to fuse information as described in previous work and delves into the mapping of imagery into 3D point clouds. More specifically, details concerning GPU acceleration and occlusion detection are the focus of this work. The overall procedure is depicted in Fig. 2.

3.1. Multispectral imagery

Four multispectral images were acquired for each viewpoint using a Parrot Sequoia device: Green (GRE), Near Infrared (NIR), Red Edge (REG) and Red (RED). It co-acquires RGB images along with multispectral imagery. Hence, the differences between these lenses can be described through homography transformations covering variations in the distortion coefficients and orientation. Multispectral lenses from Parrot Sequoia have a focal length of 4mm, whereas the RGB lens has 15.9mm, thus following fisheye and perspective models. Regarding the dimensionality of outputs, the resolution is 1280 × 960px and 4608 × 3456px for multispectral and visible imagery, respectively. Fig. 1 shows the four multispectral bands on the right side, as captured by the drone. In this work, 1720 (1st dataset) and 1536 (2nd dataset) multispectral images are used in the following sections.

3.2. Thermographic imagery

Similarly to multispectral data, thermal images are co-acquired with RGB information by a DJI Zenmuse XT2 device. In contrast to multispectral, the thermal lens is described through a perspective projection (19mm) and its outcome has a low resolution (640 × 512px). RGB lens has a focal length of 8mm, whereas its images present a higher resolution (4000 × 3000px). Regarding file formats, thermal data is stored as grayscale RJPEG (Radiometric JPEG) images, a proprietary format that allows reconstructing the scene temperature. However, the temperature is of no interest in this work as fusion will rely on the grayscale representation. Some samples of thermal images are depicted in Fig. 1. In this work, 575 (1st dataset) and 410 (2nd dataset) thermographic images are used in the following sections.

3.3. Drone

Previous sensors were mounted in a quadcopter drone (DJI Matrice 210). Although their respective sensors synchronously acquire visible and multispectral/thermal imagery, these are not synchronized between them. Missions were planned so that the view direction is *nadir*, and frontal and side overlap are 90% and 85%, respectively. Flight altitude was set to 45m for the first area and 50m for the second one.

3.4. Rectification of image distortion

Images acquired either by thermographic or multispectral devices follow distinct distortion models. RGB and multispectral imagery present fisheye distortion due to their lower focal length, though it is especially visible in multispectral images. Thermographic lenses present a much larger focal length, thus following a pincushion distortion. However, distortion correction can be parameterized by five coefficients (radial and tangential distortion coefficients, $(k_1, k_2, p_1, p_2, k_3)$) in both cases, along with the camera matrix, K. Hence, the correction of every dataset is performed regardless of the camera source. An exception to this is given by multispectral imagery, which defines its own distortion model with four coefficients arranged in a 4th-degree equation (Ruiz et al., 2019).

Note that distortion coefficients are not frequently read from image metadata. Instead, the bundle adjustment phase of SfM is used to estimate these coefficients. Hence, the main discussion of this work is established in the 3D reconstruction rather than in the calibration. In this study, the output of Pix4Dmapper's BA is used as the result of the calibration stage.

3.5. Mapping and fusion of multi-source imagery

According to the surveying devices, two different pipelines are here established. The first pairs visible images with thermographic data by determining the image's id and the corresponding co-acquired thermal image. Similarly to the described pipeline, multispectral images are linked to the source RGB image. However, there exists a hierarchy among multispectral images where the Green (GRE) layer is the root. Hence, properties of other layers are calculated according to this, including the matrices for fusing images.

Based on previous work, the ECC registration method has been shown effective enough for the fusion of images in different wavelength intervals, such as visible and infrared. Control parameters of ECC are the number of maximum iterations upon convergence (n) as well as the aimed precision (ϵ). Another intrinsic parameter is the dimensionality

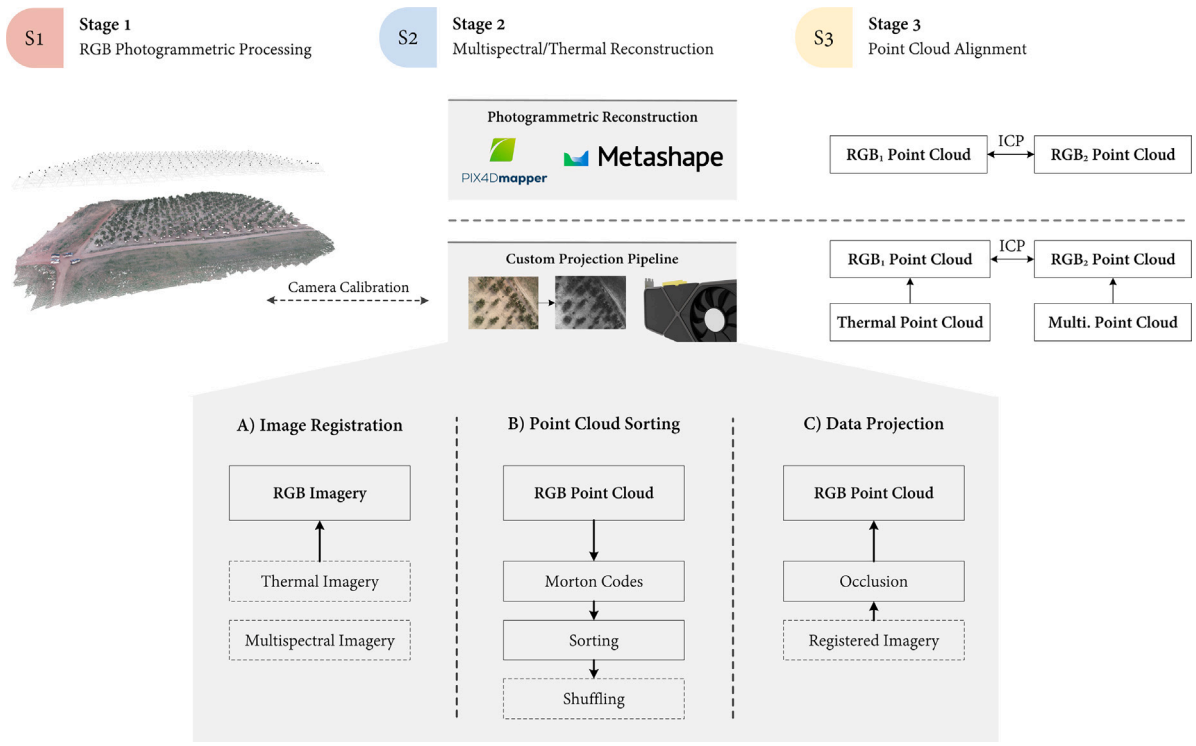


Fig. 2. Overall procedure of the proposed method. First, the visible point cloud is reconstructed. Then, alternative data sources are projected into the previous point cloud. Finally, point clouds are aligned through the estimation of rigid transformations. The reconstruction of alternative point clouds is compared against popular commercial solutions.

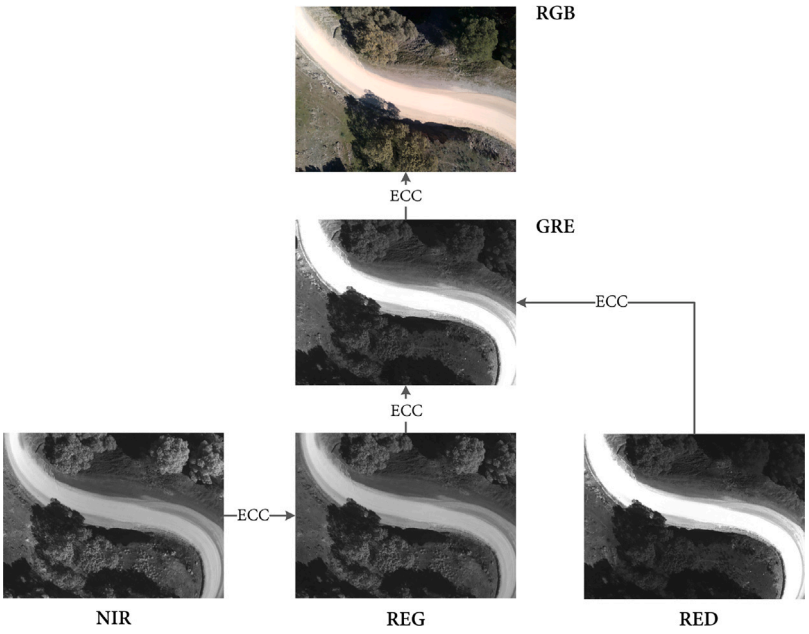


Fig. 3. Hierarchical registration of multispectral imagery.

of input images, as it considerably reduces latency. Given two images of the same dimensionality, ECC seeks the matrix that maximizes the intensity correlation of both images. For a context with images from any source, differences among them are considered unknown and thus explained using homography matrices. Otherwise, affine and euclidean matrices of lower size could be applied to reduce latency. A brief discussion is established in Experimental results and discussion to determine optimal parameters regarding the fusion of visible, thermal and multispectral information.

According to the peculiarities of multispectral images, these are fused hierarchically rather than sequentially. With this approach, images with more similar intensity are first registered. More specifically, the next pipeline is followed: *GRE* → *RGB*, *RED* → *GRE* → *RGB*, *REG* → *GRE* → *RGB* and *NIR* → *REG* → *GRE* → *RGB* (see Fig. 3). Still, each image is only registered once, as the whole chain concatenation is computed using matrix composition. Note that previously cited chains project multispectral imagery to the visible plane, whereas the aimed projection goes from the visible plane to an alternative data source. Hence, the inverse matrix is calculated, including the scale matrix that

allows comparing images of larger dimensionality with low-resolution images. Besides scaling, visible lenses may acquire a much larger area according to their focal length. Thus, RGB images are also cropped according to this difference, plus an extra size required by rotations. However, convergence is not always achieved and this scenario is identified by calculating the angles of the resulting quadrilateral shape, which are discarded according to a threshold α for highly distorted forms.

The result of this section is a composite matrix for every RGB image, either relating it to a thermal or multispectral image. These are not related to more than one data source as they cannot be considered to be triggered synchronously. At this point, point clouds can be fed with additional information since their projection matrix to the RGB image plane is known.

3.6. Projection in 3D point clouds

The following step is to project 3D points into their original data source. As occurs with photogrammetric pipelines, the RGB point cloud may be first generated since it represents the most accurate and dense 3D result. From that representation, it can be enhanced by including additional spectral information at every point. Otherwise, a low-density version of the RGB point cloud would be generated from another data source. Hence, this work reuses the starting RGB point cloud, whose viewpoints are calibrated during SfM. For each one, the projection $P_c \cdot X$ is computed as $K \cdot [R|-R \cdot t_{local}] \cdot X$, with K being the RGB camera matrix, calculated with the focal length and principal point coordinates, R is the rotation matrix derived from yaw, roll and pitch angles, t_{local} is the camera position in the local coordinate system of the point cloud and X is a 4D point ($[x\ y\ z\ 1]^T$).

The aforementioned P_c matrix allows projecting 3D points into individual RGB image planes. From these, the previously computed inverse homography matrices allow translating RGB coordinates into a different Cartesian coordinate system. Hence, resulting projections corresponding to 3D points may fall out of image boundaries. As a result, augmented point clouds do not present the same dimensions as the first one, mainly in the boundaries. This is due to the smaller focal length of devices coupled to RGB, though the resolution is preserved (points per m^2).

3.7. Occlusion

The main drawback of the previously described projection is that spectral data from images could be projected into 3D points that were not visible from a flight viewpoint. Hence, this could lead to point clouds with incorrect colouring for additional layers. In this work, occlusion is handled using images that depict the scene depth as grayscale values, also known as z-buffers. Due to the aim of this work, z-buffers also store the index of the nearest point projected at each pixel. The points whose projection depth is higher than the one stored at the pixel are discarded. Otherwise, points could be modelled as geometric shapes to be intersected by ray-casting, thus allowing us to determine the nearest point. However, this approach requires large data structures to organize dense point clouds. Besides memory consumption, the choice of the geometric shape is not trivial either.

From a ∞-filled image, points close to the viewpoint are projected to determine the minimum depth at each pixel, generating a z-buffer similar to the output of depth sensors. However, z-buffers with low resolution also lead to point clouds of lower density, similar to those generated by commercial software. To avoid this, the dimensionality of z-buffers is upscaled by a factor s that helps on balancing memory consumption and point density.

Memory usage is especially relevant for hardware such as the GPU. In this work, accelerated algorithms are implemented in OpenGL's shading language (GLSL) and general-purpose compute shaders. Hence, this framework bounds the maximum size of a buffer according to GPU capabilities, though it is frequently much lower. z-buffers are typically part of the rendering pipeline, but they are not integrated into compute shaders. They must be implemented as a low-level mechanism supported by buffers known as Shader Storage Buffer Objects (SSBO). Despite the lack of integrated z-buffers, compute shaders provide a more efficient and flexible solution to read, write and perform trivial operations in buffers. Furthermore, it avoids some unnecessary stages from the rendering pipeline, including polygon rasterization or interpolations.

Regarding memory allocation, z-buffers are here modelled as integers of 64 bits (`uint64_t`) using OpenGL's modern extensions, to be equally split between distance and index. Distance is not a numeric integer, $d \in \mathbb{R}$, though its low-level representation can be interpreted as an integer using `floatBitsToUint`. For a viewpoint, xyz_c, and a 3D point (xyz_p) with index i_p, the Euclidean distance is computed, transformed into an integer and concatenated to i_p (Eq. (1)). With this representation, fewer and greater comparisons work as usual. Thus, these operators can be applied to build a z-buffer by selecting the minimum distance while also carrying a point index. To this end, the atomic *min.* operator (`atomicMin`) is used.

$$z_{k,l} \leftarrow d(xyz_p, xyz_c) \frown i_p \tag{1}$$

The procedure to project the point cloud into z-buffers is following described in Algorithm 1 if it fits in the GPU's VRAM and the maximum size of OpenGL's buffers. Otherwise, Algorithm 2 is followed. Note that the second scenario omits the smaller procedures shown in Algorithm 1. Because data transfers of z-buffers in CPU ↔ GPU are notably smaller, point batches are iterated in the outer *for* and only transferred once. Instead, each viewpoint's z-buffer is transferred once per point batch. Once all the points have been projected into every z-buffer, the indices indicating visible points are obtained, as proposed in Algorithm 1.

Algorithm 1 Simplest case for projecting points in z-buffers, where all the points fit in GPU's VRAM.

Points are notated as P_{GPU} and V_{CPU} are camera viewpoints placed at v_{p_i} whose images have dimensionality $x \cdot y$. The term id refers to $thread_{id}$ in the GPU.
Build $z\text{-buffer}_{GPU} \leftarrow \text{uint64}(x \cdot y)$ and $\text{indices}_{GPU} \leftarrow \text{uint}(x \cdot y)$
if Sort P_{GPU} **then**
    $P_{GPU} \leftarrow \text{sortPoints}(P_{GPU})$ ▷ (Algorithm 3)
end if
procedure BUILD z-BUFFERS
    for v in V_{CPU} **do**
        $M \leftarrow v_{projection}$
        procedure RESET z-BUFFER IN GPU
            $z\text{-buffer}_{GPU\,i} \leftarrow \text{UINT64}_{MAX}\ \forall i \in [0, x \cdot y[$
        end procedure
        procedure FILL z-BUFFER IN GPU(M, v_p)
            $p_{3D} \leftarrow P_{GPU}[\text{id}]$
            $p_{3D} \leftarrow \left[M \cdot [p_{3D_x}, p_{3D_y}, p_{3D_z}, 1]^T\right]^T_{xyz}$
            $p_{2D} \leftarrow [p_{3D_x}/p_{3D_z}, p_{3D_y}/p_{3D_z}]^T$
            if p_{2D} inside image **then**
                $\text{dist} \leftarrow \text{floatBitsToUint}(\text{length}(p_{3D} - v_p))$
                $\text{tag} \leftarrow p_{index} | (\text{dist} << 32)$
                $\text{atomicMin}(z\text{-buffer}_{GPU}[\text{id}], \text{tag})$
            end if
        end procedure
        procedure OBTAIN INDICES FROM z-BUFFER
            $\text{index}[\text{id}] \leftarrow z\text{-buffer}_{GPU}[\text{id}]\ \&\ \text{UINT}_{MAX}$
        end procedure
        visible $\leftarrow \text{indices}_{GPU}$
        Process visible to assign point cloud colours
    end for
end procedure

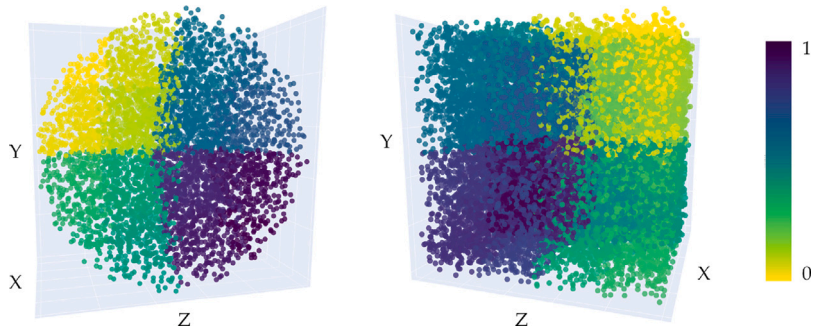


Fig. 4. Colouring of two randomized point clouds in [0, 1] according to the Morton encoding with 30 bits.

Algorithm 2 Split of the point vector if it does not fit in GPU's VRAM during their projection in z-buffers.

P_{GPU} refers to points, V_{CPU} are camera viewpoints placed at v_{p_i} whose images have dimensionality $x \cdot y$ and n_{max} is the maximum number of points that can be allocated in the GPU.
Build $z\text{-buffer}_{CPU} \leftarrow \text{uint64}(x \cdot y \cdot |V_{CPU}|)$
$z\text{-buffer}_{CPU_i} \leftarrow \text{UINT64}_{MAX}\ \forall i \in [0, x \cdot y \cdot |V_{CPU}|[$
Build $z\text{-buffer}_{GPU} \leftarrow \text{uint64}(x \cdot y)$ and $\text{indices}_{GPU} \leftarrow \text{uint}(x \cdot y)$
$n_{left} \leftarrow |P_{GPU}|$
if Sort P_{GPU} **then**
    $P_{GPU} \leftarrow \text{sortPoints}(P_{GPU})$ ▷ (Algorithm 3)
end if
while $n_{left} > 0$ **do**
    $n_{current} \leftarrow \min(n_{left}, n_{max})$
    Transfer point batch to GPU
    for v in V_{CPU} **do**
        Transfer to GPU part of $z\text{-buffer}_{CPU}$ from v
        Complete $z\text{-buffer}_{GPU}$ with current point batch
        Read $z\text{-buffer}_{GPU}$ into the pointer of $z\text{-buffer}_{CPU}$
    end for
    Update n_{left}
end while
for v in V_{CPU} **do**
    Transfer to GPU part of $z\text{-buffer}_{CPU}$ from v
    Split indices from $z\text{-buffer}_{GPU}$into indices_{GPU}
    visible $\leftarrow \text{indices}_{GPU}$
    Process visible to assign point cloud colours
end for

3.8. Point cloud sorting

Point clouds generated by software present an unknown order regardless of their coordinates, *xyz*. However, the outcome of the projection depends on the sampled 3D volume. As a result, points within an area could be visible from a camera viewpoint, whereas other parts of the point cloud may be discarded. Consequently, PCs of unknown order also require unordered updates in memory buffers. However, organized updates are known to be more efficient due to faster memory writing. Although there is no such ordering in 3D space, there exist methods for the hierarchical clustering of geometry, such as the Z-order curve (also known as Morton curve, depicted in Fig. 4). The outcome is a 1D buffer where close indices in memory are linked to spatially close 3D points.

For large PCs, changing their order presents a significant delay even for the most efficient sorting algorithms. Therefore, GPGPU programming allows sorting the PC in parallel to reduce the response time during the projection procedure. In this work, Radix sort is used and adapted to GPU programming by splitting it into two different steps: (1) up-sweep (reduce) and (2) down-sweep (Nguyen, 2007). Following this approach, the buffer is sorted with a complexity of $\mathcal{O}(m \log n)$, with m being the number of bits and n being the number of points. Prior to sorting, *xyz* coordinates are converted to values of 30 bits (m) that encode individual coordinates (10 bits for each one). These are frequently known as Morton codes, though they can have a variable bit length.

Despite the enhancement of PC sorting, GPU-based algorithms are handled on the GPU through work groups. These are small groups of threads that are allowed to share data. Thus, the main drawback of sorting the PC buffer as a whole is that workload is not uniformly shared among work groups. Instead, the ordered PC can be shuffled according to the size of the workgroups. Further insight into these optimizations is provided in Experimental results and discussion.

The algorithmic flow for the described sort configuration is shown in Algorithm 3. First, points are sorted with Radix Sort according to Morton encoding. Otherwise, a random order is used. Then, the point buffer can be reordered according to groups of size $n_{shuffle}$. The second stage is implemented as multi-core CPU rather than GPU to avoid duplicating large point buffers in a limited VRAM.

3.9. Alignment of heterogeneous data sources

Image matching was performed in visible and alternative imagery from the same sensor. This way, images are known to be acquired with a similar timestamp, and thus ECC converges in a reasonable time. However, RGB point clouds can be reused for different data sources whether they are represented in the same local coordinate system. Despite georeferencing providing accurate positioning, point clouds from distinct data sources may not exactly overlap. Hence, point clouds are further processed by finding the rigid transformation matrix that is able to align both of them with a precision below a threshold (Fig. 5). In this work it is performed with the Iterative Closest Point (ICP) algorithm.

4. Experimental results and discussion

In this section, we aim to compare the proposed methodology with other notable software solutions for building thermal point clouds, such as Pix4Dmapper or Agisoft Metashape, using their highest-quality reconstruction. We focus our tests on the response time and the point cloud size, although some reconstructions from these solutions also present geometrical errors. The evaluation was performed on a PC with AMD Ryzen Threadripper 3970X 3.6 GHz, 256 GB RAM, two Nvidia RTX A6000 GPU and Windows 10 OS. The proposed methodology is implemented in C++ using the Open Graphics Library (OpenGL) both for rendering and computing. CPU-based methods are accelerated using the OpenMP multithreading framework.

Algorithm 3 Point cloud sorting.

P_{GPU} refers to the point cloud and n_{max} is the maximum number of points that can be allocated in the GPU.
$n_{left} \leftarrow |P_{GPU}|$
indices $\leftarrow$ uint $\lceil |P_{GPU}| \rceil$
while $n_{left} > 0$ **do**
  $n_{current} \leftarrow \min(n_{left}, n_{max})$
  Transfer the following $n_{current}$ batch to GPU
  procedure SORT IN GPU
    $codes_{GPU} \leftarrow mortonGPU(P_{GPU})$
    indices $\leftarrow$ RadixSort($P_{GPU}, codes_{GPU}$)
    procedure RADIXSORT($P_{GPU}, CODES_{GPU}$)
      for bit $\leftarrow 0 \rightarrow 30$ **do**
        Apply bit mask 1 << bit toP_{GPU}
        Up-sweep phase of prefix scan
        Reset last position to zero
        Down-sweep phase of prefix scan
        Reorder $P_{GPU}$with new positions
      end for
    end procedure
  end procedure
  procedure SORT IN MULTI-CORE CPU
    if Shuffle **then**
      $n_{groups} \leftarrow \lceil n_{current} / n_{shuffle} \rceil$
      $rdn_{group} \leftarrow$ RandomVector($n_{groups}, 0, 1$)
      $iota_{group} \leftarrow [0, 1, ..., n_{groups} - 1]$
      $id_{old}, id_{new} \leftarrow$ Sort($rdn_{group}, iota_{group}$)
      for group in (id_{old}, id_{new}) **do**
        Move point batch from id_{old} to id_{new}
      end for
    else
      Reorder points with indices
    end if
  end procedure
  Update P_{GPU} and n_{left} with $n_{current}$
end while

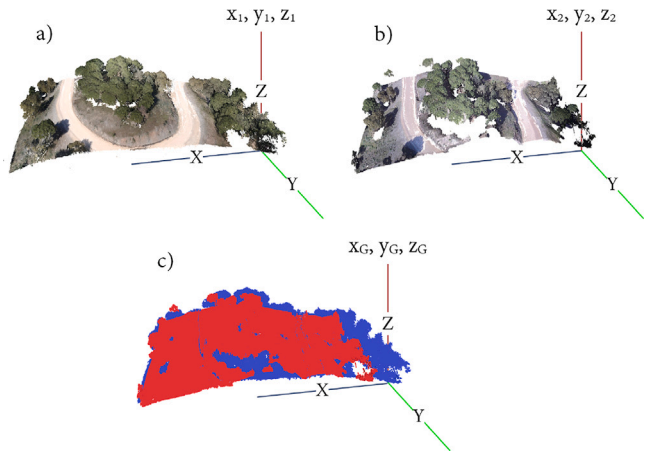


Fig. 5. ICP registration process: (a) RGB point cloud from the first dual device, (b) RGB point cloud from the second dual device, and (c) both point clouds aligned by minimizing their distance, with *G* referring to a Global system.

The devices and datasets used in this section have been previously presented in Section Materials and methods, whereas the study area is following described. The baseline RGB point clouds consist of 115M (multispectral) and 97M (thermal) points for the first dataset, whereas the second dataset has point clouds with a dimensionality of 126M and 96M points. To further stress this evaluation and achieve higher point density, we have used a single depth buffer resolution, $d \leftarrow 10$. Regarding point cloud order, points are initially shuffled to show the benefits of spatial sorting. Apart from this, two different setups are here shown: (1) sorted with Morton codes and (2) sorted and shuffled in small groups. The experimental results are obtained by repeating every test four times and averaging them.

Regarding the configuration of commercial solutions, the steps of image alignment and reconstruction of a dense point cloud (i.e., densification) were performed with the highest quality. Since both of them are known to be able to use CUDA-enabled GPUs, they were enabled to accelerate their pipeline on the GPU. Furthermore, the latter is known to be accelerated with the multi-GPU framework CUDA and therefore, was expected to perform better due to the availability of two GPUs.

4.1. Study area

The proposed method has been evaluated over two different environments located in the region of Jaén, Spain. To show its effectiveness in multiple fields, data regarding a (1) forestry area and (2) olive grove was processed. Hence, the fusion of information is proven to work well in scenarios with more significant features (2) and not recognizable features (1). Fig. 6 shows the location of both areas, surveyed with the equipment described in Materials and methods. The flight altitude was 454m and 595m above sea level, respectively. The forestry scene is composed of repetitive patterns from vegetation and human-made structures. The first area covers nearly 1.97ha, whereas the second comprises 2.44ha.

4.2. Image processing

The image alignment stage is configured according to the ECC parameters: (1) the number of iterations to converge and (2) the aimed precision. The first is set to 400 iterations to cover the number of iterations to converge with a wide margin. Besides these parameters, the dimensionality of the compared images is also relevant since a lower number of details also implies a faster convergence. However, images were not downscaled in this study to guarantee the quality of the alignment. Finally, the hierarchical alignment of multispectral images is also studied by means of confusion matrices that show the correlation of pairs of bands. Accordingly, Fig. 7 justifies the hierarchy shown in Fig. 3. Note that NIR and REG layers are reciprocally the most similar. However, REG images are more similar to the root, GRE, and thus NIR layers were aligned to REG images.

Regarding the sought precision, images were aligned using $\epsilon \leftarrow 10^{-5}$ for all the data sources, according to the similarity observed in Fig. 8. The similarity was here normalized to improve the readability of the chart, though the average intensity differences through the experimental results were observed to be in the order of 1^{-3}. Lower values (higher exponent) harden the convergence of the ECC alignment and increase the latency in the order of ms. Hence, an intermediate value that balances both intensity and response time was used. On the other hand, the number of iterations is set to guarantee the procedure converges before reaching such a number. Therefore, a large value such as $it \leftarrow 400$ was enough for our case studies.

4.3. Response time

First, we compared the response time of commercial solutions against three configurations of our method. The response time is decomposed into three stages: (1) reading the visible point cloud, only for our method, (2) preprocessing stage, including image registration, and (3) reconstruction of the dense point cloud. The second step is also related to the procedure of image alignment in commercial software. The results are shown in Fig. 11, and further insight is provided

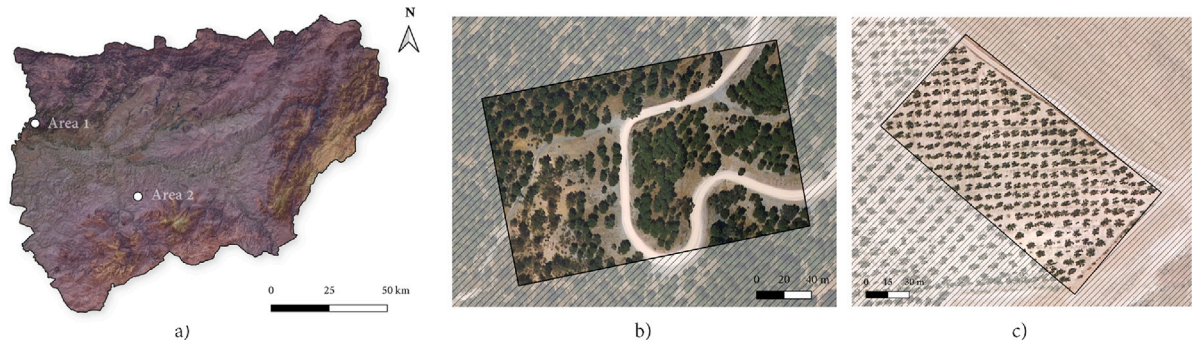


Fig. 6. Overview of surveyed region and areas. (a) Region of Jaén, (b) forest and (c) olive grove.

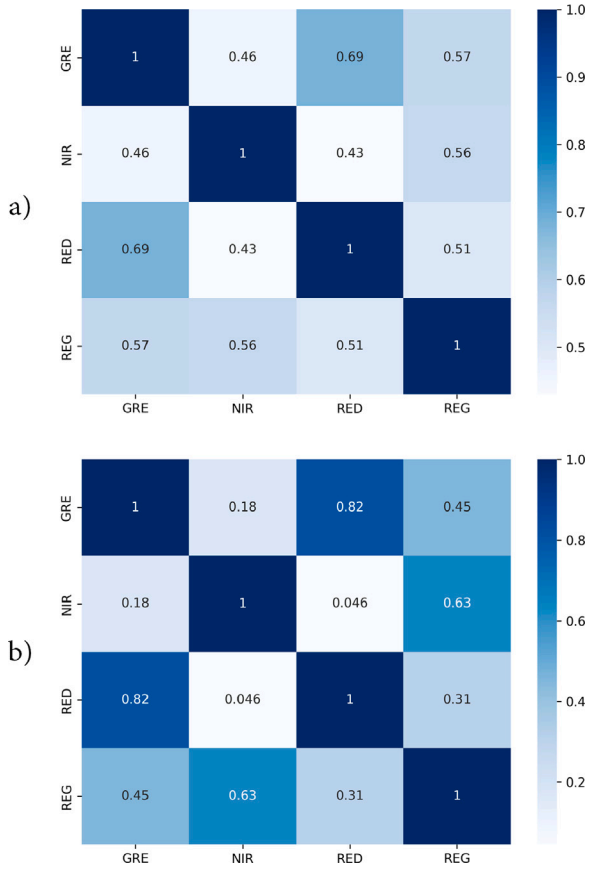


Fig. 7. Confusion matrix to depict the similarity of multispectral images in the first (a) and second dataset (b).

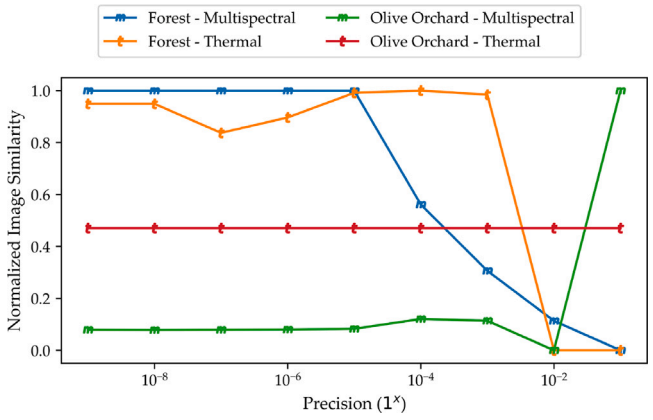


Fig. 8. Image similarity measured after performing ECC alignment with a precision equal to 1^x. Lower values seek the most accurate alignment. The similarity is normalized in [0, 1] for improving the visualization.

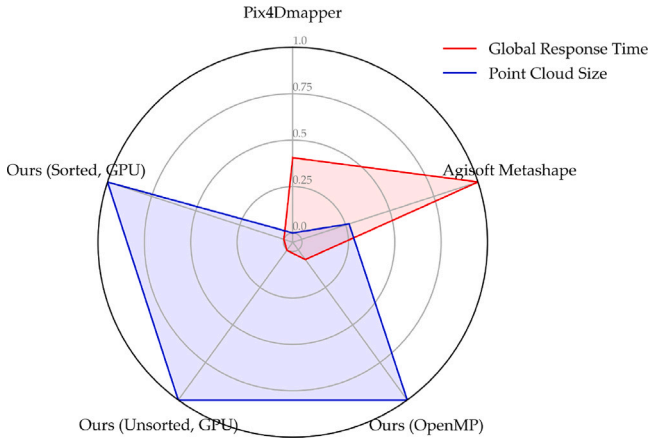


Fig. 9. Average response time and point cloud size obtained by commercial software and three versions of the proposed method.

in Tables 2 and 3. Despite our method reconstructing larger point clouds, the latency is significantly lower than commercial solutions. This difference is further exploited for GPU-based solutions and larger datasets since part of the measured latency comes from the allocation of GPU buffers that can be reused for multispectral layers. Thus, the baseline latency from allocation does not increase with a larger number of images. Accordingly, our sorted GPU solution improves Agisoft Metashape by 76.64% on average to build thermographic point clouds. The improvement of multispectral reconstruction is 62.5% with respect to Pix4Dmapper.

The evaluations are performed with a z-buffer with $d \leftarrow 10$, which was shown to balance latency and point cloud dimensionality. Larger values of d involve higher data transfers, while also risking to include background points with foreground data. Otherwise, lower values of d reduce the dimensionality of point clouds at expense of minimizing the latency from data transfers. Based on this, $d \leftarrow 1$ would provide the baseline improvement of ECC method to build a sparse point cloud. Regarding image preprocessing, both procedures iterate over the whole dataset. However, our method only evaluates pairs of images instead of surrounding images. This advantage is considerably

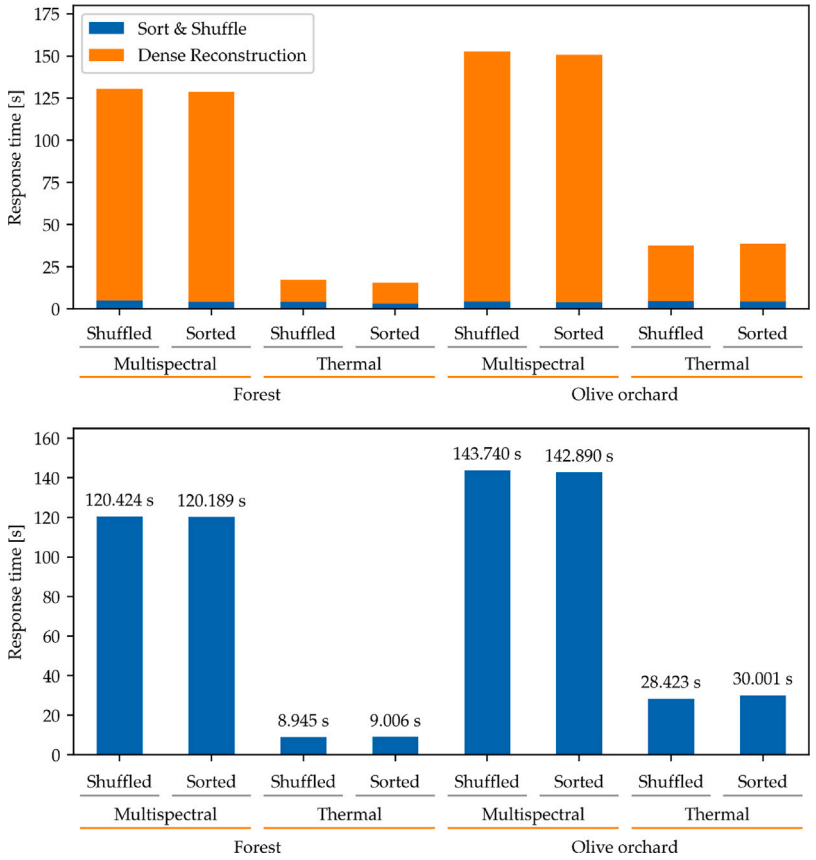


Fig. 10. Comparison of performance of the dense reconstruction stage using sorted & shuffled data and globally sorted data. The first image compares the overall latency of the dense reconstruction, whereas the second image only compares the projection procedure.

diminished for multispectral images since ECC is launched five times for each viewpoint.

4.4. Point cloud size and normalized response time

The proposed method is included as part of a pipeline where dense point clouds of RGB and alternative data sources are jointly reconstructed. In this scenario, the first point cloud is considerably larger than those obtained with imagery of lower resolution and less recognizable features. From this baseline, the number of points of the RGB point cloud is reduced according to their visibility from the whole set of camera viewpoints and the depth buffer's resolution. Still, most of them are preserved and linked to data interpolated from thermal and multispectral imagery.

Point clouds reconstructed from multispectral images are dense due to the provided variety of reflectance and the higher number of recognized features. However, the higher sparsity of thermal point clouds is observed in Fig. 12. For the thermographic datasets, our method increases the point cloud size by 366.3% (forest) and 475.52% (olive orchard) with respect to Agisoft Metashape. In this regard, this commercial solution reconstructs the densest point clouds from the compared software, both for multispectral and thermal imagery. Despite this, it is significantly slower for larger datasets such as the multispectral. For the latter data source, the size increases by 149.89% (forest) and 141.93% (olive orchard) also in comparison with Agisoft Metashape.

Previous results based on global latency present a wider gap whether they are normalized according to the point cloud's dimensionality. The normalized latency of our best method improves the results of the best commercial solution by 95.51% and 96.9% on average for thermal and multispectral images, respectively.

4.5. Overall comparison

The results of the five compared methods over four datasets are summarized in Fig. 9. The global latency and point cloud size are normalized in $[0, 1]$. The first metric must be minimized, whereas the second one ought to be maximized to provide better visualization and details of the target dataset. According to this figure, the three proposed methods present an outstanding performance in terms of response time and point cloud dimensionality. These are followed by Agisoft Metashape, which is considerably slower, albeit generating larger results than the last alternative, Pix4Dmapper.

4.6. Data shuffling in GPU

As stated by previous work (Schütz et al., 2021; Kerbl et al., 2017), specific GPU patterns are used by manufacturers to distribute the workload among thread groups. This knowledge can help to further reduce the latency of GPU-based solutions, in comparison with globally sorted point clouds. Thus, the objective is to shuffle points in batches to balance the workload of every workgroup. As the outcome of a batch is expected to depend on the camera viewpoint, the data shuffling ought to help to reduce latency by distributing visible and non-visible points. To this end, we conducted a comparison of data shuffled or not after being sorted, with groups of size $g \leftarrow 32$. The results are depicted in Fig. 10 by splitting the latency into the Sort & Shuffle and Reconstruction stages. Numeric results are also annotated in Table 5.

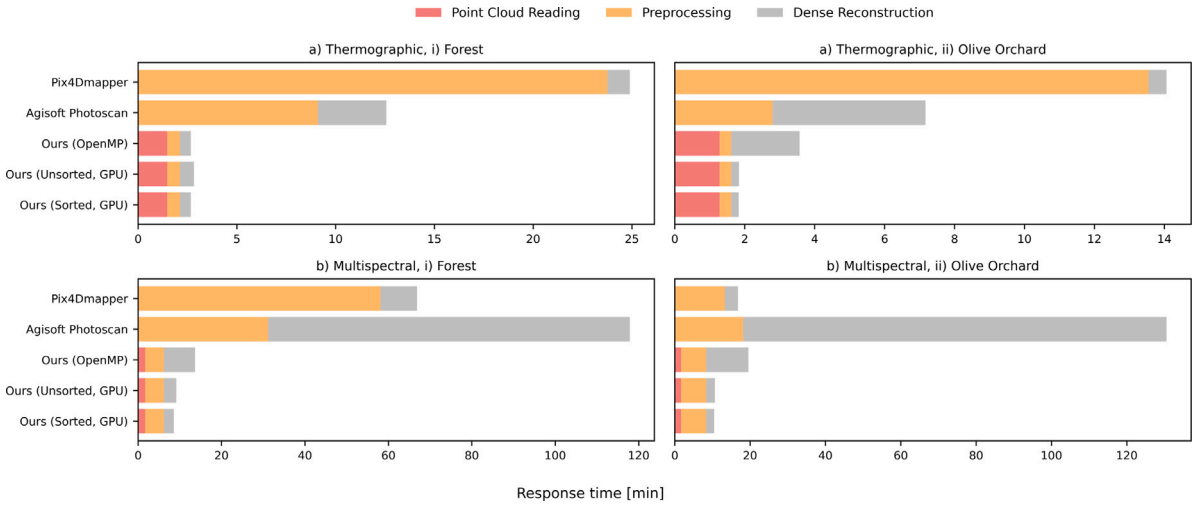


Fig. 11. Accumulated response time for every data source (a. thermographic and b. multispectral), study area (1. forestry and 2. olive orchard) and solution, including commercial software and different configurations of our method.

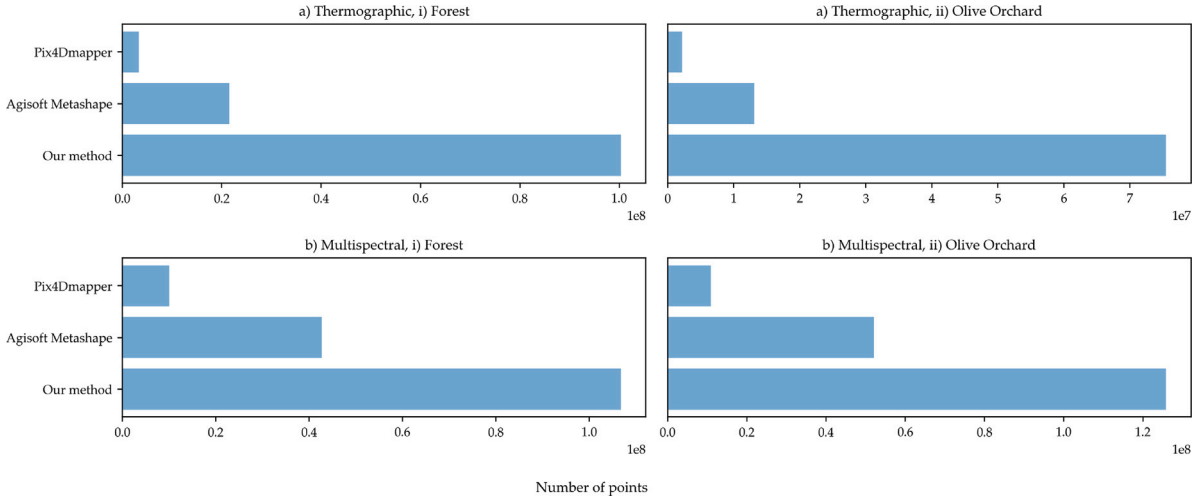


Fig. 12. Continuation of Fig. 11 on the dimensionality of the reconstructed point clouds.

To provide a better insight into the data shuffling, the second image shows only the response time from the dense reconstruction. The shuffled ordering also involves global sorting prior to organizing the point cloud into batches. Thus, the Sort & Shuffle procedure is more time-consuming than solely sorting. Accordingly, the summed response time of the Shuffled version is higher for every configuration whether we include both stages. Otherwise, the results offer slight changes whether only the reconstruction stage is considered. It is observed that data shuffling effectively helps to reduce latency in smaller point clouds (thermographic), whereas larger point clouds worsen the latency obtained by the global sort. Unlike previous studies, the proposed case study is very unbalanced for individual viewpoints. From hundreds of millions of points, only a few million are visible from a camera viewpoint. Thus, the workload balancing was not effective. Even if so, the Shuffle stage increases the latency for a one-time process, thus making it not worth it. Instead, real-time processes such as rendering are more likely to take advantage of this technique.

4.7. Visual results

Table 1 shows the rendering of multispectral and thermographic point clouds obtained by the three compared solutions. In this regard, both commercial solutions are observed to reconstruct incomplete and erroneous geometry for thermographic data. Instead, our method projects these data over an RGB point cloud that is accurately reconstructed. Multispectral imagery presents higher dimensionality and more recognized features, thus leading to better reconstructions even for commercial software, regardless of the reduced point density. Hence, the reconstructed multispectral point clouds are similar among all the compared methods. The main drawback of photogrammetric multispectral point clouds is the existence of gaps, which are much less frequent in RGB point clouds despite vegetation being hard to reconstruct.

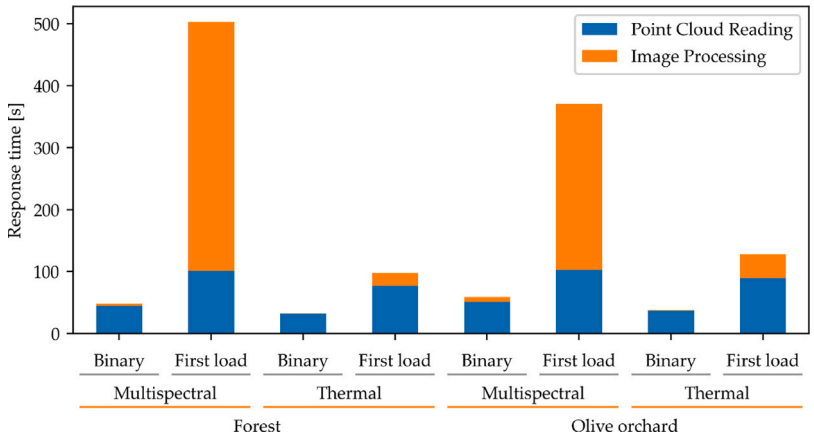


Fig. 13. Comparison of latency for the first two stages of our method in the first and subsequent application launches (after storing calculated data in binary files).

4.8. Binary data

Previous results are measured by (1) loading a PLY point cloud, (2) reading TIFF and JPEG images and (3) reconstructing the dense point cloud from previous data. However, these are known to be time-consuming stages that can be treated differently after the first launch. First, point clouds are stored in binary file format. Similarly, the image alignment results can be stored in binary files. Once aligned, RGB imagery is not loaded again. Finally, the reconstructed point cloud is stored as an additional colour to be attached to previously stored binary point clouds. Therefore, Fig. 13 compares the latency of binary and non-binary reading of the first two stages. The latency of reading the dense reconstruction is equal to the first stage since they load point clouds of the same dimensionality. The numeric data is shown in Table 4. Thus, the proposed procedure can be sped up for later executions in an end-user application.

5. Conclusions and future work

The reconstruction of multiple 3D point clouds is a common task in surveying work. It may solely include visible imagery, though it is frequent to include other sensors that shed light on some of the features of the studied field. For instance, archaeological work in Remote Sensing can be supported by thermographic imaging. Hence, the reconstruction of 3D point clouds of further data sources, such as thermal and multispectral, can be performed by taking advantage of previous RGB reconstructions. Although all of them can be performed with commercial software, we identified several drawbacks, including (1) high response time, (2) low point density and (3) incorrect estimations without further knowledge, e.g., GCPs. The latter drawback occurs due to the lower dimensionality and higher number of defects of thermal and multispectral imagery, thus hardening the recognition of image features.

We proposed to integrate the reconstruction of thermal and multispectral 3D point clouds in an accelerated projection-based methodology. To this end, the reconstruction of RGB point clouds was performed with traditional software. Then, the subsequent reconstructions were done with much less latency (−69.58% on average in comparison with the best alternative) and yielded point clouds whose size increased over 140%. Accordingly, improvements on the latency are above >95% if the response time is normalized according to the point cloud's size. Despite both commercial software and our method using GPU-accelerated procedures, ours was shown to outperform two widespread solutions in terms of response time and size.

Still, GPU parallelism was implemented with GLSL, which provides little flexibility and control of the execution pipeline (e.g., overlapping data transfers and thread logic). Furthermore, it runs over a single GPU as it is part of a rendering framework. As a future work, the GPGPU solution could transit to a flexible and multi-GPU framework such as CUDA. Regarding the explored data, further data sources could be integrated into the multi-layer point cloud, such as hyperspectral reflectance. Finally, instead of performing the projection over RGB point clouds, LiDAR outcome could be used as a dense point cloud where other data sources are projected.

CRediT authorship contribution statement

Alfonso López: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Carlos J. Ogayar:** Conception and design of study, Writing – original draft, Writing – review & editing. **Juan M. Jurado:** Conception and design of study, Acquisition of data, Writing – original draft, Writing – review & editing. **Francisco R. Feito:** Conception and design of study, Acquisition of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This result has been partially supported by the Spanish Ministry of Science, Innovation and Universities via a doctoral grant to the first author (FPU19/00100) and through the research project TED2021-132120B-I00 funded by MCIN/AEI/10.13039/501100011033/ and ERDF funds "A way of doing Europe". All authors approved the version of the manuscript to be published.

Appendix. Numeric and graphic results

See Tables 1–5.

Table 1
Graphic results of the dense reconstruction of (1) Agisoft Metashape, (2) Pix4Dmapper and (3) our proposed method. Multispectral point clouds are rendered with the GRE band.

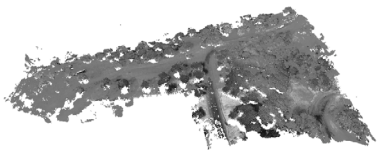
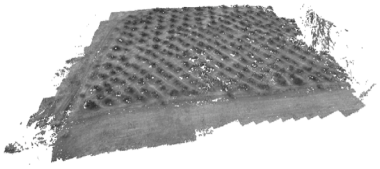
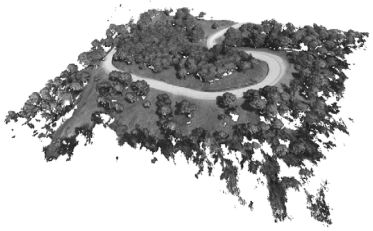
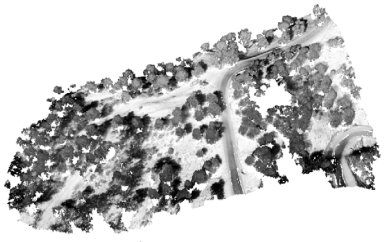
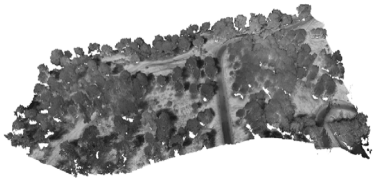
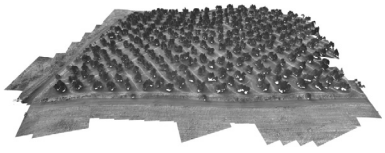
Configuration	Data source	Dataset 1	Dataset 2
A. Metashape	Thermal		
	Multispectral		
Pix4Dmapper	Thermal		
	Multispectral		
Our method	Thermal		
	Multispectral		

Table 2
Numeric results of both commercial solutions and proposed configurations of our method in two different thermal datasets. The global response time is split into reading point cloud (Stage 1), pre-processing (Stage 2) and densification (Stage 3), i.e., generation of a dense point cloud.

Configuration	Stage 1	Stage 2	Stage 3	Global latency	Normalized latency	Point cloud size	Aligned images
Thermal Dataset 1 (Forestry)							
Pix4Dmapper	–	23.76 min	1.13 min	24.89 min	449.08 µs/point	3,325,454 points	98%
Agisoft Metashape	–	**9.10** min	3.47 min	12.57 min	35.05 µs/point	21,514,286 points	62.95%
Ours - OpenMP, No Sort	1.48 min	**0.64** min	9.77 min	11.89 min	6.90 µs/point	**100,322,449 points**	**99.287%**
Ours - GPU, No Sort			0.71 min	2.83 min	1.69 µs/point		
Ours - GPU, Global Sort			**0.546** min	**2.666** min	**1.594 µs/point**		

(*continued on next page*)

Table 2 (*continued*).

Configuration	Stage 1	Stage 2	Stage 3	Global latency	Normalized latency	Point cloud size	Aligned images
Thermal Dataset 2 (Olive Orchard)							
Pix4Dmapper	–	13.55 min	0.51 min	14.06 min	388.92 µs/point	2,169,058 points	90%
Agisoft Metashape	–	**2.80** min	4.37 min	7.17 min	32.79 µs/point	13,117,583 points	**99.39%**
Ours - OpenMP, No Sort	1.28 min	**0.34** min	1.95 min	3.57 min	2.83 µs/point	**75,494,967 points**	89.75%
Ours - GPU, No Sort			0.217 min	1.837 min	1.459 µs/point		
Ours - GPU, Global Sort			**0.208** min	**1.828** min	**1.452 µs/point**		

Table 3
Continuation of Table 2 for multispectral imagery.

Configuration	Stage 1	Stage 2	Stage 3	Global latency	Normalized latency	Point cloud size	Aligned images
Multispectral Dataset 1 (Forestry)							
Pix4Dmapper	–	58.11 min	**8.76** min	66.87 min	398.35 µs/point	10,071,939 points	97%
Agisoft Metashape	–	**31.22** min	86.64 min	117.86 min	165.49 µs/point	42,731,004 points	67.74%
Ours - OpenMP, No Sort	1.71 min	**4.45** min	7.50 min	13.66 min	7.67 µs/point	**106,780,612 points**	**100%**
Ours - GPU, No Sort			2.99 min	9.15 min	5.14 µs/point		
Ours - GPU, Global Sort			**2.42** min	**8.58** min	**4.82 µs/point**		
Multispectral Dataset 2 (Olive Orchard)							
Pix4Dmapper	–	**13.3** min	**3.5** min	16.8 min	92.56 µs/point	10,889,523 points	**100%**
Agisoft Metashape	–	18.15 min	112.36 min	130.51 min	150.52 µs/point	52,021,396 points	**100%**
Ours - OpenMP, No Sort	1.67 min	**6.69** min	11.19 min	19.55 min	9.32 µs/point	**125,857,793 points**	99.934%
Ours - GPU, No Sort			2.31 min	10.67 min	5.08 µs/point		
Ours - GPU, Global Sort			**2.08** min	**10.44** min	**4.97 µs/point**		

Table 4
Response time of the first two stages if (1) it is the first load or (2) binary data has already been stored as a result of a previous load. Stage 1 refers to point cloud reading, whereas Stage 2 involves the image processing prior to the dense reconstruction.

Dataset	Binary		First load	
	Stage 1	Stage 2	Stage 1	Stage 2
(a) Thermal, (i) Forest	**0.61** min	**0.01** min	1.48 min	0.64 min
(a) Thermal, (ii) Olive orchard	**0.52** min	**0.01** min	1.28 min	0.34 min
(b) Multispectral, (i) Forest	**0.85** min	**0.12** min	1.71 min	4.45 min
(b) Multispectral, (ii) Olive orchard	**0.73** min	**0.06** min	1.67 min	6.69 min

Table 5
Latency of point cloud sorting and reconstruction if the ordering is performed globally or globally plus shuffled into small groups.

Dataset	Global sort		Shuffled sort ($g \leftarrow 32$)	
	Point cloud sort	Dense reconstruction	Point cloud sort	Dense reconstruction
(a) Thermal, (i) Forest	**0.071** min	0.500 min	0.076 min	**0.473** min
(a) Thermal, (ii) Olive orchard	**0.052** min	0.150 min	0.052 min	**0.149** min
(b) Multispectral, (i) Forest	**0.065** min	**2.381** min	0.074 min	2.395 min
(b) Multispectral, (ii) Olive orchard	**0.071** min	**2.003** min	0.082 min	2.007 min

References

Adán, A., Quintana, B., García Aguilar, J., Pérez, V., Castilla, F.J., 2020. Towards the use of 3D thermal models in constructions. Sustainability 12 (20), 8521. http://dx.doi.org/10.3390/su12208521, URL: https://www.mdpi.com/2071-1050/12/20/8521. Number: 20 Publisher: Multidisciplinary Digital Publishing Institute.

Akhoundi Khezrabad, M., Valadan Zoej, M.J., Safdarinezhad, A., 2022. A new approach for geometric correction of UAV-based pushbroom images through the processing of simultaneously acquired frame images. Measurement 199, 111431. http://dx.doi.org/10.1016/j.measurement.2022.111431, URL: https://www.sciencedirect.com/science/article/pii/S0263224122006613.

Alvarez-Vanhard, E., Corpetti, T., Houet, T., 2021. UAV & satellite synergies for optical remote sensing applications: A literature review. Sci. Remote Sens. 3, 100019. http://dx.doi.org/10.1016/j.srs.2021.100019, URL: https://www.sciencedirect.com/science/article/pii/S2666017221000067.

Dahaghin, M., Samadzadegan, F., Dadrass Javan, F., 2021. Precise 3D extraction of building roofs by fusion of UAV-based thermal and visible images. Int. J. Remote Sens. 42 (18), 7002–7030. http://dx.doi.org/10.1080/01431161.2021.1951875, Publisher: Taylor & Francis.

Dlesk, A., Vach, K., Pavelka, K., 2022. Photogrammetric co-processing of thermal infrared images and RGB images. Sensors 22 (4), 1655. http://dx.doi.org/10.3390/s22041655, URL: https://www.mdpi.com/1424-8220/22/4/1655. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

González, O., Lizarraga, M.I., Karaman, S., Salas, J., 2019. Thermal radiation dynamics of soil surfaces with unmanned aerial systems. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Olvera-López, J.A., Salas, J. (Eds.), Pattern Recognition. In: Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 183–192. http://dx.doi.org/10.1007/978-3-030-21077-9_17.

Grechi, G., Fiorucci, M., Marmoni, G.M., Martino, S., 2021. 3D thermal monitoring of jointed rock masses through infrared thermography and photogrammetry. Remote Sens. 13 (5), 957. http://dx.doi.org/10.3390/rs13050957, URL: https://www.mdpi.com/2072-4292/13/5/957. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Hoegner, L., Abmayr, T., Tosic, D., Turzer, S., Stilla, U., 2018. Fusion of 3D point clouds with TIR images for indoor scene reconstruction. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLII-1. Copernicus GmbH, (ISSN: 1682-1750) pp. 189–194. http://dx.doi.org/10.5194/isprs-archives-XLII-1-189-2018, URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-1/189/2018/.

Hoegner, L., Tuttas, S., Stilla, U., 2016a. 3D building reconstruction and construction site monitoring from RGB and TIR image sets. In: 2016 12th IEEE International Symposium on Electronics and Telecommunications. ISETC, pp. 305–308. http://dx.doi.org/10.1109/ISETC.2016.7781118.

Hoegner, L., Tuttas, S., Xu, Y., Eder, K., Stilla, U., 2016b. Evaluation of methods for coregistration and fusion of RPAS-based 3D point clouds and thermal infrared images. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLI-B3. Copernicus GmbH, (ISSN: 1682-1750) pp.

241–246. http://dx.doi.org/10.5194/isprs-archives-XLI-B3-241-2016, URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B3/241/2016/.

Hu, X., Xie, C., Fan, Z., Duan, Q., Zhang, D., Jiang, L., Wei, X., Hong, D., Li, G., Zeng, X., Chen, W., Wu, D., Chanussot, J., 2022. Hyperspectral anomaly detection using deep learning: A review. Remote Sens. 14 (9), 1973. http://dx.doi.org/10.3390/rs14091973, URL: https://www.mdpi.com/2072-4292/14/9/1973. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

Javadnejad, F., Gillins, D.T., Parrish, C.E., Slocum, R.K., 2020. A photogrammetric approach to fusing natural colour and thermal infrared UAS imagery in 3D point cloud generation. Int. J. Remote Sens. 41 (1), 211–237. http://dx.doi.org/10.1080/01431161.2019.1641241, Publisher: Taylor & Francis.

Jia, S., Jiang, S., Lin, Z., Li, N., Xu, M., Yu, S., 2021. A survey: Deep learning for hyperspectral image classification with few labeled samples. Neurocomputing 448, 179–204. http://dx.doi.org/10.1016/j.neucom.2021.03.035, URL: https://www.sciencedirect.com/science/article/pii/S0925231221004033.

Jiang, S., Jiang, C., Jiang, W., 2020. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. ISPRS J. Photogramm. Remote Sens. 167, 230–251. http://dx.doi.org/10.1016/j.isprsjprs.2020.04.016, URL: https://www.sciencedirect.com/science/article/pii/S0924271620301131.

Jurado, J.M., López, A., Pádua, L., Sousa, J.J., 2022a. Remote sensing image fusion on 3D scenarios: A review of applications for agriculture and forestry. Int. J. Appl. Earth Obs. Geoinf. 112, 102856. http://dx.doi.org/10.1016/j.jag.2022.102856, URL: https://www.sciencedirect.com/science/article/pii/S1569843222000589.

Jurado, J.M., Padrón, E.J., Jiménez, J.R., Ortega, L., 2022b. An out-of-core method for GPU image mapping on large 3D scenarios of the real world. Future Gener. Comput. Syst. 134, 66–77. http://dx.doi.org/10.1016/j.future.2022.03.022, URL: https://www.sciencedirect.com/science/article/pii/S0167739X22000978.

Juszczyk, J.M., Wijata, A., Czajkowska, J., Krecichwost, M., Rudzki, M., Biesok, M., Pyciński, B., Majewski, J., Kostecki, J., Pietka, E., 2021. Wound 3D geometrical feature estimation using poisson reconstruction. IEEE Access 9, 7894–7907. http://dx.doi.org/10.1109/ACCESS.2020.3035125, Conference Name: IEEE Access.

Kerbl, B., Kenzel, M., Schmalstieg, D., Steinberger, M., 2017. Effective static bin patterns for sort-middle rendering. In: Proceedings of High Performance Graphics. HPG '17, Association for Computing Machinery, New York, NY, USA, pp. 1–10. http://dx.doi.org/10.1145/3105762.3105777.

Kong, Y., Leung, H., Zhang, B., Xing, S., Chen, Y., 2018. 3-D point cloud reconstruction of infrared images based on improved structure from motion. In: 2018 2nd European Conference on Electrical Engineering and Computer Science. EECS, pp. 307–310. http://dx.doi.org/10.1109/EECS.2018.00063.

Lafi, G.A., Zhu, Z., Dawood, T., Zayed, T., 2017. 3D thermal and spatial modeling of a subway tunnel: A case study. In: Computing in Civil Engineering 2017. American Society of Civil Engineers, pp. 386–394. http://dx.doi.org/10.1061/9780784480823.046, URL: https://ascelibrary.org/doi/10.1061/9780784480823.046.

Lin, D., Jarzabek-Rychard, M., Tong, X., Maas, H.-G., 2019. Fusion of thermal imagery with point clouds for building façade thermal attribute mapping. ISPRS J. Photogramm. Remote Sens. 151, 162–175. http://dx.doi.org/10.1016/j.isprsjprs.2019.03.010, URL: https://www.sciencedirect.com/science/article/pii/S0924271619300796.

López, A., Jurado, J.M., Jiménez-Pérez, J.R., Feito, F.R., 2022. Generation of hyperspectral point clouds: Mapping, compression and rendering. Comput. Graph. 106, 267–276. http://dx.doi.org/10.1016/j.cag.2022.06.011, URL: https://www.sciencedirect.com/science/article/pii/S0097849322001145.

López, A., Jurado, J.M., Ogayar, C.J., Feito, F.R., 2021a. A framework for registering UAV-based imagery for crop-tracking in precision agriculture. Int. J. Appl. Earth Obs. Geoinf. 97, 102274. http://dx.doi.org/10.1016/j.jag.2020.102274, URL: https://www.sciencedirect.com/science/article/pii/S030324342030917X.

López, A., Jurado, J.M., Ogayar, C.J., Feito, F.R., 2021b. An optimized approach for generating dense thermal point clouds from UAV-imagery. ISPRS J. Photogramm. Remote Sens. 182, 78–95. http://dx.doi.org/10.1016/j.isprsjprs.2021.09.022, URL: https://www.sciencedirect.com/science/article/pii/S0924271621002604.

López, A., Jurado, J.M., Padrón, E.J., Ogayar, C.J., Feito, F.R., 2021. Comparison of GPU-based Methods for Handling Point Cloud Occlusion. In: Ortega, L.M., Chica, A. (Eds.), Spanish Computer Graphics Conference (CEIG). The Eurographics Association, http://dx.doi.org/10.2312/ceig.20211364.

Mohamad, M.N., Reba, M.N.M., Hossain, M.S., 2021. A screening approach for the correction of distortion in UAV data for coral community mapping. Geocarto Int. 1–33. http://dx.doi.org/10.1080/10106049.2021.1958066, Publisher: Taylor & Francis.

Nguyen, H., 2007. Gpu Gems 3, first ed. Addison-Wesley Professional.

Pádua, L., Matese, A., Di Gennaro, S.F., Morais, R., Peres, E., Sousa, J.J., 2022. Vineyard classification using OBIA on UAV-based RGB and multispectral data: A case study in different wine regions. Comput. Electron. Agric. 196, 106905. http://dx.doi.org/10.1016/j.compag.2022.106905, URL: https://www.sciencedirect.com/science/article/pii/S0168169922002228.

Pan, H., Guan, T., Luo, K., Luo, Y., Yu, J., 2021. A visibility-based surface reconstruction method on the GPU. Comput. Aided Geom. Design 84, 101956. http://dx.doi.org/10.1016/j.cagd.2021.101956, URL: https://www.sciencedirect.com/science/article/pii/S0167839621000029.

Park, H., Lee, D., 2019. Comparison between point cloud and mesh models using images from an unmanned aerial vehicle. Measurement 138, 461–466. http://dx.doi.org/10.1016/j.measurement.2019.02.023, URL: https://www.sciencedirect.com/science/article/pii/S0263224119301411.

Piao, J., Chen, Y., Shin, H., 2019. A new deep learning based multi-spectral image fusion method. Entropy 21 (6), 570. http://dx.doi.org/10.3390/e21060570, URL: https://www.mdpi.com/1099-4300/21/6/570. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

Poux, F., Mattes, C., Selman, Z., Kobbelt, L., 2022. Automatic region-growing system for the segmentation of large point clouds. Autom. Constr. 138, 104250. http://dx.doi.org/10.1016/j.autcon.2022.104250, URL: https://www.sciencedirect.com/science/article/pii/S0926580522001236.

Ruiz, A.L., Rodríguez, J.M.J., Anguita, C.J.O., Higueruela, F.R.F., 2019. Multispectral Registration, Undistortion and Tree Detection for Precision Agriculture. In: Casas, D., Jarabo, A. (Eds.), Spanish Computer Graphics Conference (CEIG). The Eurographics Association, http://dx.doi.org/10.2312/ceig.20191209.

Sanchez, J., Denis, F., Coeurjolly, D., Dupont, F., Trassoudaine, L., Checchin, P., 2020. Robust normal vector estimation in 3D point clouds through iterative principal component analysis. ISPRS J. Photogramm. Remote Sens. 163, 18–35. http://dx.doi.org/10.1016/j.isprsjprs.2020.02.018, URL: https://www.sciencedirect.com/science/article/pii/S0924271620300575.

Schütz, M., Kerbl, B., Wimmer, M., 2021. Rendering point clouds with compute shaders and vertex order optimization. Comput. Graph. Forum 40 (4), 115–126. http://dx.doi.org/10.1111/cgf.14345, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14345.

Schütz, M., Kerbl, B., Wimmer, M., 2022. Software rasterization of 2 billion points in real time. Proc. ACM Comput. Graph. Interact. Tech. 5 (3), 24:1–24:17. http://dx.doi.org/10.1145/3543863.

Singh, A.P., Yerudkar, A., Mariani, V., Iannelli, L., Glielmo, L., 2022. A bibliometric review of the use of unmanned aerial vehicles in precision agriculture and precision viticulture for sensing applications. Remote Sens. 14 (7), 1604. http://dx.doi.org/10.3390/rs14071604, URL: https://www.mdpi.com/2072-4292/14/7/1604. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Webster, C., Westoby, M., Rutter, N., Jonas, T., 2018. Three-dimensional thermal characterization of forest canopies using UAV photogrammetry. Remote Sens. Environ. 209, 835–847. http://dx.doi.org/10.1016/j.rse.2017.09.033, URL: https://www.sciencedirect.com/science/article/pii/S0034425717304455.

Wiemann, T., Mitschke, I., Mock, A., Hertzberg, J., 2018. Surface reconstruction from arbitrarily large point clouds. In: 2018 Second IEEE International Conference on Robotic Computing. IRC, pp. 278–281. http://dx.doi.org/10.1109/IRC.2018.00059.

Zheng, H., Zhong, X., Yan, J., Zhao, L., Wang, X., 2020. A thermal performance detection method for building envelope based on 3D model generated by UAV thermal imagery. Energies 13 (24), 6677. http://dx.doi.org/10.3390/en13246677, URL: https://www.mdpi.com/1996-1073/13/24/6677. Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

Zhu, J., Xu, Y., Ye, Z., Hoegner, L., Stilla, U., 2021. Fusion of urban 3D point clouds with thermal attributes using MLS data and TIR image sequences. Infrared Phys. Technol. 113, 103622. http://dx.doi.org/10.1016/j.infrared.2020.103622, URL: https://www.sciencedirect.com/science/article/pii/S1350449520306708.