

Article

# 3D Radiometric Mapping by Means of LiDAR SLAM and Thermal Camera Data Fusion

Davide De Pazzi <sup>1</sup>, Marco Pertile <sup>2,\*</sup> and Sebastiano Chiodini <sup>2</sup><sup>1</sup> School of Engineering, University of Padova, 35131 Padova, Italy<sup>2</sup> Department of Industrial Engineering, University of Padova, 35131 Padova, Italy

\* Correspondence: marco.pertile@unipd.it

**Abstract:** The ability to produce 3D maps with infrared radiometric information is of great interest for many applications, such as rover navigation, industrial plant monitoring, and rescue robotics. In this paper, we present a system for large-scale thermal mapping based on IR thermal images and 3D LiDAR point cloud data fusion. The alignment between the point clouds and the thermal images is carried out using the extrinsic camera-to-LiDAR parameters, obtained by means of a dedicated calibration process. Rover's trajectory, which is necessary for point cloud registration, is obtained by means of a LiDAR Simultaneous Localization and Mapping (SLAM) algorithm. Finally, the registered and merged thermal point clouds are represented through an OcTree data structure, where each voxel is associated with the average temperature of the 3D points contained within. Furthermore, the paper presents in detail the method for determining extrinsic parameters, which is based on the identification of a hot cardboard box. Both methods were validated in a laboratory environment and outdoors. It is shown that the developed system is capable of locating a thermal object with an accuracy of up to 9 cm in a 45 m map size with a voxelization of 14 cm.

**Keywords:** thermal camera; LiDAR; SLAM; data fusion; OcTree



**Citation:** De Pazzi, D.; Pertile, M.; Chiodini, S. 3D Radiometric Mapping by Means of LiDAR SLAM and Thermal Camera Data Fusion. *Sensors* **2022**, *22*, 8512. <https://doi.org/10.3390/s22218512>

Academic Editor: Jason Rambach

Received: 15 October 2022

Accepted: 31 October 2022

Published: 4 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

3D radiometric mapping is attracting growing attention in robotics, as it allows knowledge of the surrounding thermal scenario, making it possible to integrate mapping and path planning.

In this work, a 3D long-range radiometric mapping methodology aimed at indoor and outdoor environments has been developed. It was chosen to represent the surface temperature distribution map using an OcTree data structure. The advantages of this approach are numerous: (1) it allows the filtering and discretization of the thermal point cloud, (2) it allows us to volumetrically represent the thermal environment and for the robot to interact directly with it, and (3) compared to a point cloud, an OcTree data structure allows for a more computationally efficient representation of space. Compared to [1] we used an affordable solid-state LiDAR (LIVOX Horizon) characterized by a high detection range (up to 250 m). Furthermore, our method provides a direct thermal camera to LiDAR extrinsic calibration without going through the calibration with respect to an RGB camera. Furthermore, in this work, a metrological characterization of the mapping method was also carried out. The method uses the pose information obtained by reconstructing the rover's trajectory with a LiDAR SLAM [2] algorithm as a basis for 3D point cloud registration. The association of the point-cloud elements with the radiometric information is obtained by performing the back projection of each LiDAR scan on the corresponding thermal image. The space is represented by an OcTree-based voxel grid, and for each occupied cell, the temperature obtained from the average of the temperatures of the points contained within is stored as the representative value.

The main contributions of this paper are as follows:

- Application of an extrinsic calibration method, originally developed to calibrate one or more optical cameras with respect to a 3D LiDAR, to the calibration of an IR camera system.
- Development of an accurate large-scale long-range thermal mapping method, capable of working even in the presence of poor lighting conditions in both indoor and outdoor scenarios.
- Development of an occupancy grid mapping based on thermal voxels stored in an OcTree object. The method offers the possibility of producing several maps at different resolutions (intended for different applications, e.g., navigation if low resolution and manipulation if high resolution) from the same data structure.
- Error analysis and study of the influence of the voxel size on the accuracy with which a thermal cluster is recognized.

The structure of the paper is divided as follows: in Section 2 is described the adopted extrinsic calibration technique, in Section 3 the three-dimensional thermal mapping method is presented, in Section 4 the experimental setup and preliminary results are shown, and in Section 5 the concluding remarks are reported.

## 2. Related Work

Thermography has a broad spectrum of applications ranging from rescue robotics to precision agriculture, industrial plant monitoring, and rover navigation. In the literature, many use cases can be found: Ulloa et al. [3] present a method for autonomous victim identification in a post-disaster environment based on thermal image processing and a computational neural network (CNN); Prashar et al. [4] highlight the benefit of infrared thermography as a tool for field phenotyping; Liu et al. [5] present a remote optical thermography detection method for silicone polymer insulating materials used in the power industry; Al-Habaibeh et al. [6] developed a method to evaluate thermal insulation performance of buildings using thermal imaging, while Cunningham et al. [7] present a slip prediction method for planetary rovers based on soil thermal inertial measurements.

Thermal mapping methods are mainly divided into four categories, as shown in Table 1. The point cloud that describes the geometry of the environment, which is going to be registered with thermal images, can be generated by means of: (1) Structure from Motion (SfM), (2) RGB-D sensor, (3) 3D laser and (4) 3D LiDAR SLAM. In the first category (thermal images registration over a point cloud generated by means of RGB images SfM) we can find the work of Ham and Golparvar-Fard [8]. The major drawbacks are the fact that this method leads to the generation of a sparse point cloud and fails under poor illumination conditions. Belonging to the second category (generation of 3D thermograms obtained by combining thermal images and RGB-D sensor) we can find the works of Muller et al. [9] and Vidas et al. [10,11]. Although they produce very detailed dense maps, these methods are limited to being used in indoor environments and to having a small operating range, due to the construction technology of RGB-D cameras. In the third category (registration method between infrared images and 3D laser scanner models), we can find the work of González-Aguilera et al. [12]. The system is capable of creating a very detailed model with long-range mapping capability for indoor and outdoor environments; however, the scan is limited to one position. Belonging to the fourth category are the methods based on LiDAR SLAM, the SLAM allows the registration of subsequent LiDAR scans, extending the operating range of this type of system. LiDAR SLAM is widely used for autonomous driving, precision agriculture, and industrial automation. A procedure for their characterization can be found in [13]. Oreifej et al. [1] presents a sensor suite comprising a set of 2D LiDARs, infrared cameras, and optical cameras, targeted at portable backpacks and capable of building 3D thermal maps. Borrmann et al. [14] proposes a system consisting of a mobile platform equipped with a 3D laser scanner, an RGB camera, and a thermal camera. Data from all sensors collected from various positions are combined

into a common reference frame using calibration and scan matching using all detected data from all sensors together in the same reference frame.

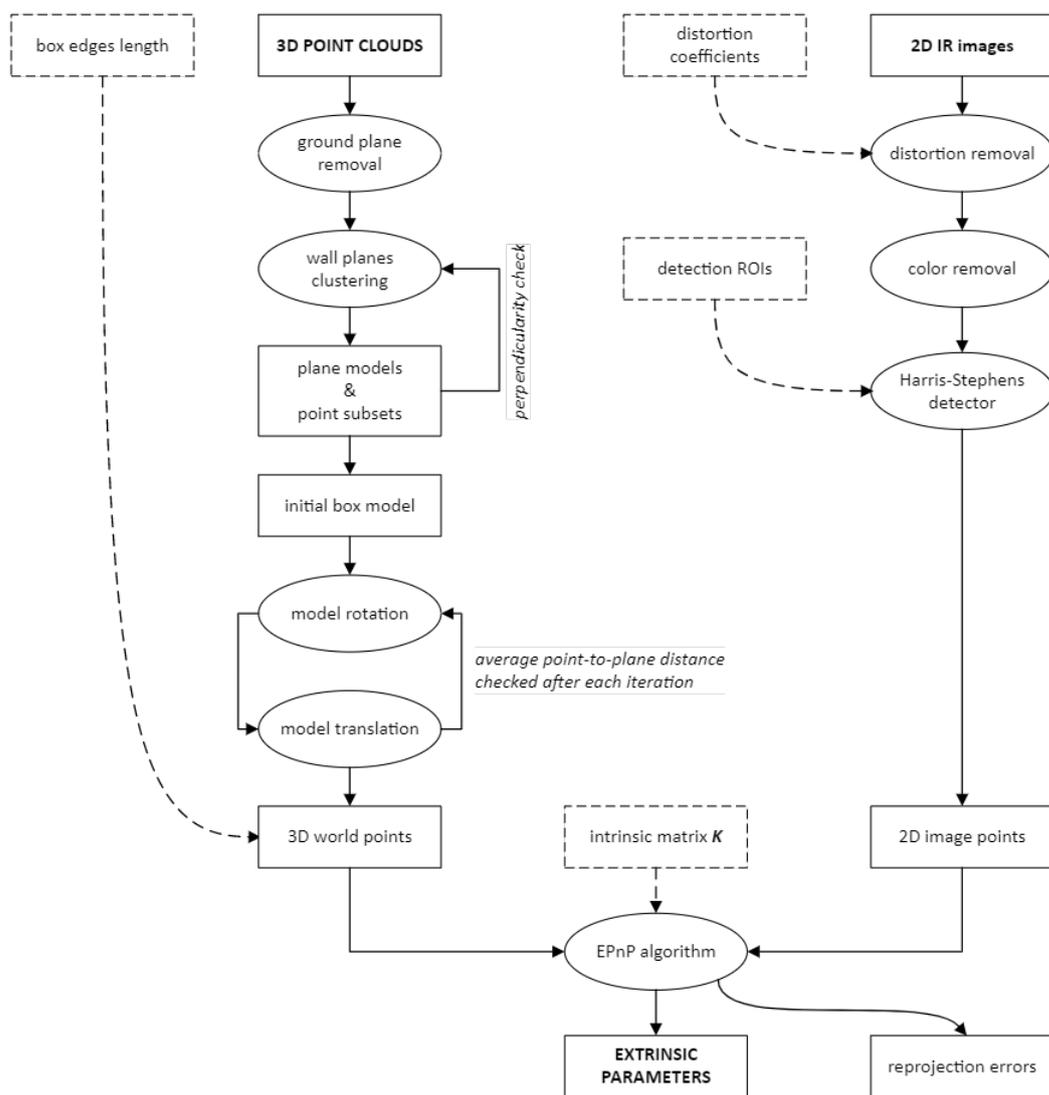
**Table 1.** Existing 3D radiometric mapping methods.

Approach	Advantages	Disadvantages
Structure from Motion (SfM) [8,15,16]	<ul style="list-style-type: none"> <li>• Long operating range</li> <li>• Outdoor and indoor</li> <li>• Lightweight and affordable</li> </ul>	<ul style="list-style-type: none"> <li>• Generates sparse map</li> <li>• Fails with poor illumination</li> <li>• No absolute scale information</li> </ul>
RGB-D [9–11,17]	<ul style="list-style-type: none"> <li>• Generate dense map</li> <li>• Lightweight and affordable</li> </ul>	<ul style="list-style-type: none"> <li>• Small operating range</li> <li>• Limited to indoor environments</li> </ul>
3D-laser [12,18,19]	<ul style="list-style-type: none"> <li>• Long-range mapping capability</li> <li>• Outdoor and indoor</li> <li>• Accurate</li> </ul>	<ul style="list-style-type: none"> <li>• Bulky and expensive</li> <li>• Scan limited to one location</li> </ul>
3D-LiDAR SLAM ([1,14,20] and proposed work)	<ul style="list-style-type: none"> <li>• Long-range mapping capability</li> <li>• Long operating range</li> <li>• Accurate</li> <li>• Lightweight and affordable</li> <li>• Generate dense map</li> </ul>	<ul style="list-style-type: none"> <li>• Large amount of data</li> </ul>

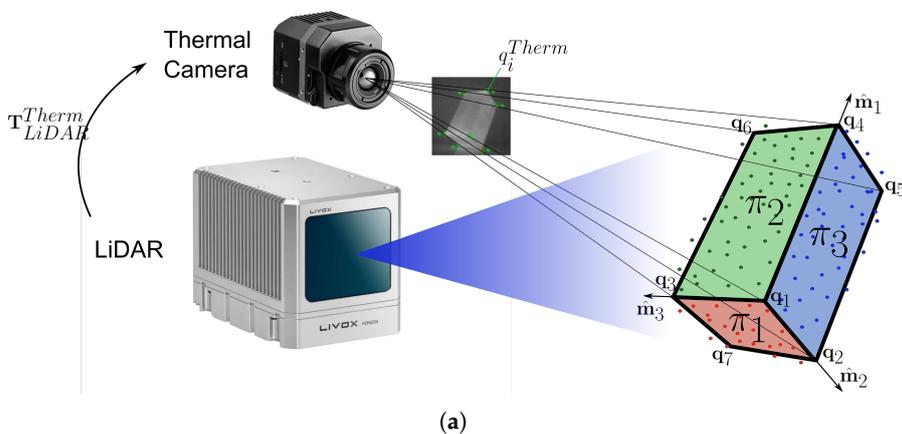
### 3. Calibration Method

This section describes how to calibrate the thermal camera with respect to the LiDAR, to compute the extrinsic parameters  $\mathbf{T}_{Therm}^{LiDAR} = [\mathbf{R}|\mathbf{t}]$  required to perform a data fusion. The method used was originally developed and tested in [21] for the calibration of several RGB cameras with respect to a 3D LiDAR, and has been adapted to thermal cameras in this work. The adoption of this method was motivated by the need to repeat the calibration several times outside a laboratory environment, which would make it impractical to use a complex and cumbersome target. In addition, the need to use a target recognizable, both in the thermal images and in the point clouds, led to the discarding of checkerboard-like patterns (characterized by LiDAR scattering and plane fitting issues) in favor of a simpler three-dimensional geometry.

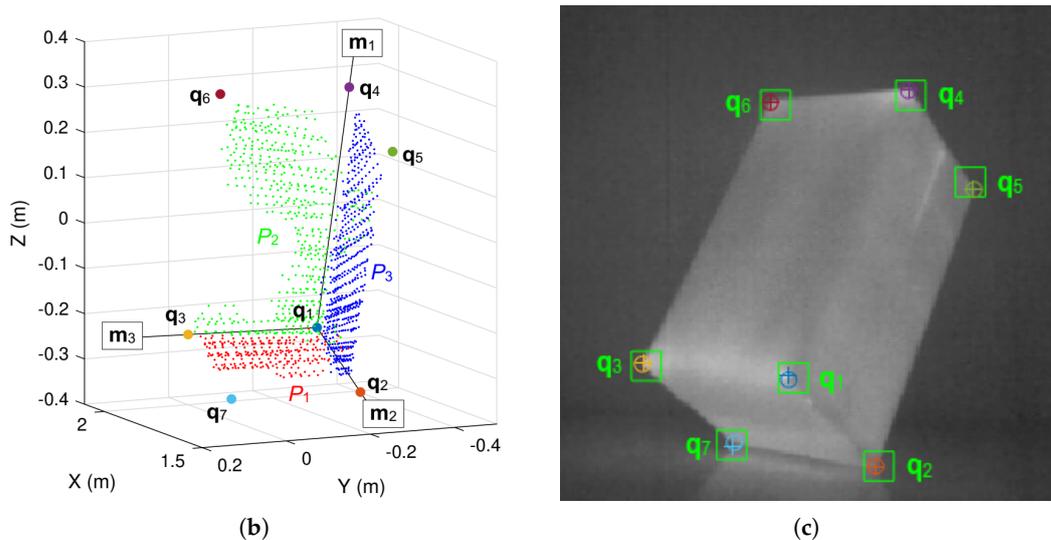
The calibration algorithm analyzes a 3D point cloud and a 2D thermal image of the same object to determine the pose of the thermal camera in the reference frame defined by LiDAR. Here, the calibration target is assumed to be an ordinary cardboard box of known size, which must be oriented so that it has three walls in full view of the instruments at the same time. The camera pose in the LiDAR reference frame can be found by solving a *Perspective-n-Point* (PnP) problem [22] that involves the seven visible corners of the box. The block diagram of the camera-LiDAR calibration process is shown in Figure 1, and a figure that gives an overview of the calibration process is shown in Figure 2a.



**Figure 1.** Block diagram of the camera-LiDAR calibration process. The algorithm takes a point cloud and a thermal image of the calibration target as input, returning the extrinsic parameters and the average reprojection error as output. The point cloud and the thermal image must have been acquired simultaneously, with the instruments in a fixed configuration. Each calibration dataset is processed separately, in order to select the one producing the lowest average reprojection error.



**Figure 2.** Cont.



**Figure 2.** (a) Overview of the calibration process. (b) Refined target model produced by the calibration algorithm, showing the orientation of the edge vectors and the position of the corners. The line of sight of the instruments is parallel to the positive X-axis. Here, the box is seen from its right side. The front, side, and top walls (sorted by increasing elevation of the centroid) are colored red, blue, and green, respectively. If the box had been viewed from the left side, the reference frame would have been rotated (with the side wall in green and the top wall in blue) without changing the ordering of the corners. (c) Grayscale thermal image of the target, highlighting the regions in which the Harris-Stephens algorithm searches for each corner. Colored crosses mark the position of the image points and colored circles mark the position of the reprojected world points. Each point is represented with the same color as the corresponding corner in the 3D box model.

### 3.1. Point Cloud Processing

The first step of the proposed calibration process is to detect and cluster the walls of the box in the 3D point cloud. Each wall is represented by a subset  $\pi_i$  of the original point cloud and by a parametric model  $P_i = (A_i, B_i, C_i, D_i)$  where

$$A_i X + B_i Y + C_i Z + D_i = 0 \tag{1}$$

is the equation of a plane expressed in world coordinates (which coincide with the LiDAR’s reference frame) and

$$\mathbf{n}_i = \{A_i, B_i, C_i\}^T \tag{2}$$

is the (non-unit) normal vector of that plane.

The ground plane can be detected and removed because, unlike any of the box walls, it can be assumed to be perpendicular to the vertical axis of the world frame. The algorithm then searches the groundless point cloud for the plane fit with the highest number of inliers, sets those points apart, and repeats the process to find two other planes. At this point, the remaining outliers are simply discarded.

The plane fitting function used here implemented a *M-estimator Sample Consensus* (MSAC) algorithm [23] to identify the point cloud subset associated with each plane model. The planes found, however, will never be exactly perpendicular to each other (as the walls of the box actually are). Moreover, the inherent randomness of a RANSAC-based algorithm means that if no valid solutions are found, the process can be repeated to obtain a different result (as the calibration point clouds are expected to be of limited size, this step would not significantly increase the computational cost of the process). For this reason,

the previous step must be repeated until the (unit) normals  $\hat{\mathbf{n}}_i$  of the three planes satisfy the following condition:

$$|\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2| + |\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_3| + |\hat{\mathbf{n}}_2 \cdot \hat{\mathbf{n}}_3| \leq E_{max} \quad (3)$$

where  $E_{max}$  is a user-defined threshold. Once a set of planes is accepted, the associated point cloud subsets  $\pi_i$  are ordered according to the elevation of their centroids. The orientation of the target can then be inferred from the position of the second plane (the side face of the box) with respect to the other two (the front and top faces of the box).

### 3.2. Target Model Refinement

The second step of the calibration process is to create and refine a suitable model of the target, from which to obtain the three-dimensional coordinates of its visible corners.

The intersection of the three plane models  $P_i$  returns three ordered edge vectors:

$$\begin{aligned} \mathbf{e}_1 &= \hat{\mathbf{n}}_2 \wedge \hat{\mathbf{n}}_3 \\ \mathbf{e}_2 &= \hat{\mathbf{n}}_1 \wedge \hat{\mathbf{n}}_3 \\ \mathbf{e}_3 &= \hat{\mathbf{n}}_1 \wedge \hat{\mathbf{n}}_2 \end{aligned} \quad (4)$$

and the single 3D point:

$$\mathbf{q}_1 = - \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix}^{-1} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \end{Bmatrix} \quad (5)$$

that is common to all of them. However, each edge vector can be thought of as the normal of one of the box walls, while  $\mathbf{q}_1$  corresponds to the corner closest to the instruments. Since the box walls and their normals are assumed to be perpendicular to each other, the box model (shown in Figure 2b) uses

$$\begin{aligned} \mathbf{m}_1 &= \mathbf{e}_1 \\ \mathbf{m}_2 &= -\mathbf{e}_1 \wedge \mathbf{e}_2 \\ \mathbf{m}_3 &= \mathbf{m}_1 \wedge \mathbf{m}_2 \end{aligned} \quad (6)$$

in place of the original edge vectors. Each new vector points in the direction of the edge of the box it represents (that is, toward the “interior” of the box, as shown in Figure 2b) because its orientation is automatically adjusted by the algorithm.

This initial box model must be refined iteratively to better fit the original point cloud. Each iteration applies three rotations and three translations, in order to minimize the sum of squared distances of all points in each subset  $\pi_i$  with respect to the corresponding plane model (identified by the point  $\mathbf{q}_1$  and the unit vector  $\hat{\mathbf{m}}_i$  parallel to  $\mathbf{m}_i$ ). When the whole model is rotated by an angle  $\theta_k$  around each edge  $\hat{\mathbf{m}}_k$  ( $k = 1, 2, 3$ ) the cost function to be minimized is

$$f_k(\theta) = \sum_{\substack{i=1 \\ i \neq k}}^3 \sum_{j=1}^{N_i} \left\| \left( \mathbf{p}_j^i - \mathbf{q}_1 \right)^T \mathbf{R}_k^\theta \hat{\mathbf{m}}_i \right\|^2 \quad (7)$$

where  $N_i$  is the number of points  $\mathbf{p}_j^i$  in the subset  $\pi_i$  and  $\mathbf{R}_k^\theta$  is the axis-angle rotation matrix [24] for  $\hat{\mathbf{m}}_k$  and  $\theta_k$ . When the whole model has instead translated a distance  $\delta_k$  along each edge  $\hat{\mathbf{m}}_k$  ( $k = 1, 2, 3$ ) the cost function to be minimized is

$$g_k(\delta) = \sum_{j=1}^{N_k} \left\| \left( \mathbf{p}_j^k - \mathbf{q}_1 - \delta \hat{\mathbf{m}}_k \right)^T \hat{\mathbf{m}}_k \right\|^2. \quad (8)$$

Here, the Levenberg-Marquardt algorithm [24] is used to solve the minimization problems. This algorithm (of standard use in non-linear optimization problems) was

chosen because it is more robust than the Gauss-Newton algorithm: its slightly lower speed, on the other hand, is not an issue because the number of iterations required to optimize the box model is expected to be small. To update the model, each rotation is applied to the edge vectors and each translation is applied to the reference point. Iterations stop as soon as the average point-to-plane distance, computed over the whole model, falls below a given threshold.

Since the true length of the box edges can be measured, the three-dimensional coordinates of the unknown corners can be derived from the refined model as follows:

$$\begin{aligned} \mathbf{q}_2 &= \mathbf{q}_1 + l_2 \hat{\mathbf{m}}_2 & \mathbf{q}_5 &= \mathbf{q}_2 + l_1 \hat{\mathbf{m}}_1 \\ \mathbf{q}_3 &= \mathbf{q}_1 + l_3 \hat{\mathbf{m}}_3 & \mathbf{q}_6 &= \mathbf{q}_3 + l_2 \hat{\mathbf{m}}_2 \\ \mathbf{q}_4 &= \mathbf{q}_1 + l_1 \hat{\mathbf{m}}_1 & \mathbf{q}_7 &= \mathbf{q}_4 + l_3 \hat{\mathbf{m}}_3 \end{aligned} \quad (9)$$

where  $l_k$  refers to the edge identified by the  $k$ -th vector.

### 3.3. Thermal Image Processing

The intrinsic matrix of the thermal camera, together with the coefficients of radial and tangential distortion, was obtained with the Zhang method [25]. In this case, a flat checkerboard pattern of wood and aluminized tape, suitable for infrared detection, was used as the calibration target.

To obtain the two-dimensional image points corresponding to the three-dimensional world points identified in the previous step, an undistorted thermal image of the target is converted to grayscale and then processed with the Harris-Stephens algorithm [26] to extract the relevant features.

Each corner of the box is searched for inside a square area with a side of 15 pixels, whose position within the image (as shown in Figure 2c) must be defined manually before starting the calibration process. This is necessary to make sure that the image points are processed in the same order as the world points but also to prevent any unrelated feature of the target's surface (or of the background) from being detected.

### 3.4. Extrinsic Parameters Determination

Knowing both the world coordinates and the corresponding image coordinates of all the visible corners of the box allows us to set up a PnP problem with  $n = 7$  points. The problem is then solved by an *Efficient PnP* (EPnP) algorithm [27]. This method addresses the general problem for  $n \geq 4$  by expressing each of the  $n$  reference points as the weighted sum of four virtual non-coplanar control points, whose coordinates in the camera frame are treated as the unknown terms of the problem. Some implementations of the EPnP algorithm include a subsequent phase of iterative Gauss-Newton optimization to further refine the results. In any case, the solution can be found if at least four corners of the target have been successfully identified in both the point cloud and the thermal image.

The extrinsic parameters of the thermal camera with respect to the LiDAR consist of a rotation matrix  $\mathbf{R}$  and a translation vector  $\mathbf{t}$  such that

$$\alpha \begin{Bmatrix} \mathbf{u} \\ 1 \end{Bmatrix} = \mathbf{K}[\mathbf{R} | \mathbf{t}] \begin{Bmatrix} \mathbf{w} \\ 1 \end{Bmatrix} = \mathbf{P}_{Therm}^{LiDAR} \begin{Bmatrix} \mathbf{w} \\ 1 \end{Bmatrix} \quad (10)$$

where  $\mathbf{w} = \{X, Y, Z\}^T$  are the coordinates of a point in the world and  $\mathbf{u} = \{u, v\}^T$  are the pixel coordinates of its reprojection on the image plane (for a given scale factor  $\alpha$ ). The intrinsic matrix of the thermal camera is

$$\mathbf{K} = \begin{bmatrix} f_u & s & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

where  $u_c$  and  $v_c$  identify the optical center,  $f_u$  and  $f_v$  are the focal lengths in pixels and  $s$  is a skew coefficient. Here  $\mathbf{P}_{Therm}^{LiDAR}$  is a  $3 \times 4$  world-to-image transformation matrix, so if the

world point  $\mathbf{q}_i$  can be matched with a given image point  $\mathbf{u}_i$  then the resulting reprojection error can be computed as:

$$\varepsilon_i = \left\| \begin{Bmatrix} \mathbf{u}_i \\ 1 \end{Bmatrix} - \frac{1}{\alpha} \mathbf{P}_{Therm}^{LiDAR} \begin{Bmatrix} \mathbf{w}_i \\ 1 \end{Bmatrix} \right\|. \quad (12)$$

If multiple point clouds and thermal images are available, the calibration algorithm automatically selects the set of extrinsic parameters which leads to the lowest average reprojection error over all the  $n$  valid points.

#### 4. Thermal Mapping Method

This section describes how a set of 3D LiDAR point clouds can be combined with a set of radiometric IR thermal images to create a three-dimensional representation of the temperature distribution in the environment the rover is traversing.

This process occurs in three steps: first, the color map and the temperature map extracted from each thermal image must be projected onto the corresponding point cloud, producing a sequence of thermal point clouds; the thermal point clouds are then registered and merged together into a single scene, using the pose information obtained by reconstructing the trajectory of the rover with a suitable LiDAR SLAM algorithm; finally, the registered and merged thermal point clouds are processed by a spatial decomposition algorithm to produce a voxel map, in which each cell of the 3D grid can be associated with a specific temperature value.

The map building is supposed to be performed offline, therefore all the point clouds produced by the LiDAR and all the radiometric images produced by the thermal camera are assumed to be available at the start of processing. The two datasets, however, must have been acquired at the same time, and with the instruments, in the same configuration they had when the extrinsic calibration was performed. Here the rover's trajectory estimation is performed using the LiDAR SLAM algorithm *livox\_horizon\_loam* ([https://github.com/Livox-SDK/livox\\_horizon\\_loam](https://github.com/Livox-SDK/livox_horizon_loam), accessed on 10 October 2022) (an improved version of LOAM *LiDAR Odometry And Mapping* algorithm [2]). This solution was chosen because it is a robust, relatively simple, and reasonably reliable method, with high drift compensation capabilities. The block diagram of the mapping process is shown in Figure 3.

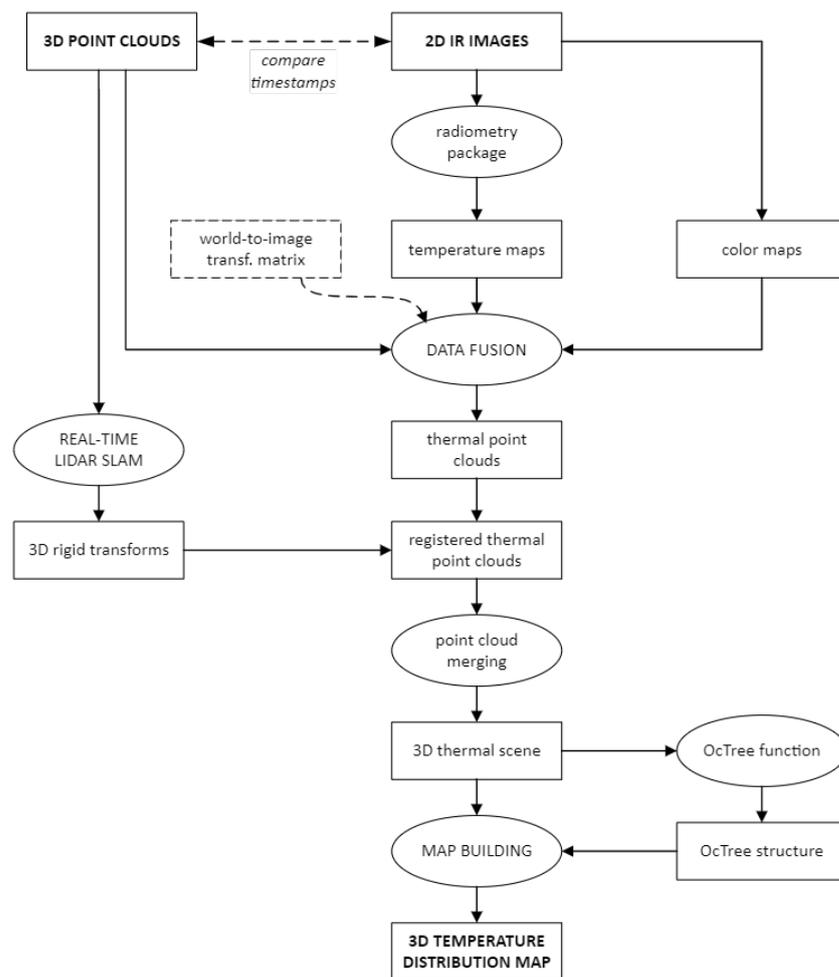
##### 4.1. Data Fusion

The data fusion between the LiDAR and the thermal camera consists of associating with each element of a point cloud the information from a specific pixel of the corresponding thermal image (that is, the values from a specific location in the color and temperature maps extracted from that image). An overview of the data fusion procedure is shown in Figure 4.

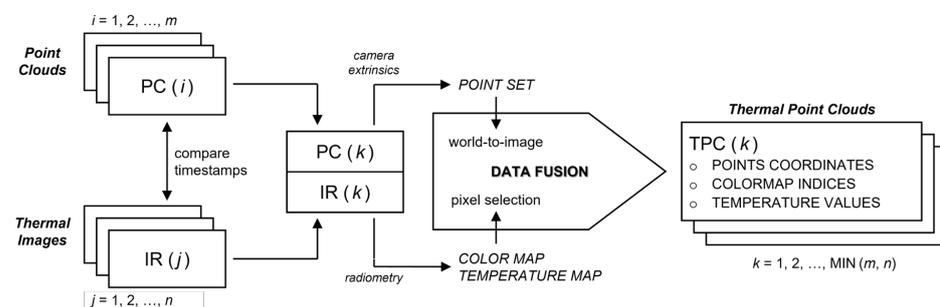
Equation (10) can be used to project the LiDAR points from the world coordinates to the image plane, as the camera has been calibrated with respect to the LiDAR and the projection matrix  $\mathbf{P}_{Therm}^{LiDAR}$  is known. Since the extrinsic parameters provided by the calibration algorithm minimize the reprojection errors, but doesn't eliminate them, the data fusion function selects the pixel

$$\mathbf{x}_i = (r_i, c_i) = ([v_i], [u_i]) \quad (13)$$

closest to the image point  $\mathbf{u}_i = \{u_i, v_i\}^T$  returned by (10) for a given world point. Both the color maps and the temperature maps extracted from the thermal images are matrices, so the vertical coordinate  $v_i$  of the point determines the row index  $r_i$  of the pixel and the horizontal coordinate  $u_i$  of the point determines the column index  $c_i$  of the pixel. As shown in (13),  $u_i$  and  $v_i$  are rounded to the nearest integer to identify the coordinates  $c_i$  and  $r_i$  of the pixel in question.



**Figure 3.** Block diagram of the mapping process. The algorithm takes a sequence of time-paired 3D point clouds and thermal images as input, returning a three-dimensional temperature distribution map of the scene as output. The 3D poses returned by the LiDAR SLAM (*livox\_horizon\_loam*) algorithm could also be used to create an occupancy map with the associated point clouds. Here the configuration of the camera-LiDAR system is expressed by the world-to-image transformation matrix  $\mathbf{P}_{Therm}^{LiDAR} = \mathbf{K}[\mathbf{R} | \mathbf{t}]$  that was obtained through the extrinsic calibration process.



**Figure 4.** Overview of the data fusion procedure. The algorithm takes a sequence of 3D point clouds and thermal images as input, returning a set of 3D thermal point clouds as output. First, the timestamp of each point cloud  $PC(i)$  is compared with the timestamp of each thermal image  $IR(j)$  to extract a sequence  $\{PC(k), IR(k)\}$  of time-paired couples. The data fusion itself consists in converting the world coordinates of each point in  $PC(i)$  to image coordinates, and using those coordinates to select a single pixel from the color map and the temperature map extracted from  $IR(k)$ . This produces a sequence of thermal point clouds  $TPC(k)$  in which each element is characterized not only by its world coordinates but also by a color and a temperature value.

Temperature maps can be extracted from each image file using dedicated software provided by the manufacturer of the thermal camera. Each radiometric thermal image includes among its metadata the coefficients required to convert the pixel intensity values into the apparent temperature values. The coefficients are automatically computed by the internal radiometric software of the thermal camera, taking into account a number of parameters set by the user: air temperature, relative humidity, target emissivity, and approximate observation distance must be specified manually before any data can be collected. At present, the user-defined parameters can be set only once per image and are applied to all pixels, so they must necessarily represent average values compatible with most of the observed scene.

Before the thermal mapping process can begin, it is necessary to make sure that the point clouds and the thermal images acquired at the same time are correctly matched to each other. This means either associating each thermal image with a point cloud (if the dataset provided by the LiDAR has more elements than the one provided by the thermal camera) or associating each point cloud with a thermal image (if the dataset provided by the thermal camera has more elements than the one provided by the LiDAR). In both cases, the synchronization between the two datasets is performed by minimizing the absolute difference between their respective sets of timestamps.

After data fusion, all elements of each thermal point cloud carry two additional pieces of information: the color of a thermal image pixel (stored as an RGB triplet, to be used for the visual representation of the results) and the corresponding value from the associated temperature map (treated as a “point intensity” property in a standard point cloud model, for ease of processing). Note that 3D points outside the area included in the thermal camera’s FOV correspond to 2D points outside the image, and therefore their coordinates in the image plane will be either negative or greater than the size of the image itself. Such points are represented with the same out-of-palette color and given an invalid (i.e., NaN) temperature value, to be easily identifiable by both the user and the algorithm.

#### 4.2. Point Cloud Registration and Merging

The coordinates of the three-dimensional points acquired by the LiDAR are always expressed in the reference frame of the instrument, whose origin and orientation change continuously with respect to the environment as the rover moves (see Figure 5a): the thermal point clouds, therefore, must be registered before they can be merged into a single scene.

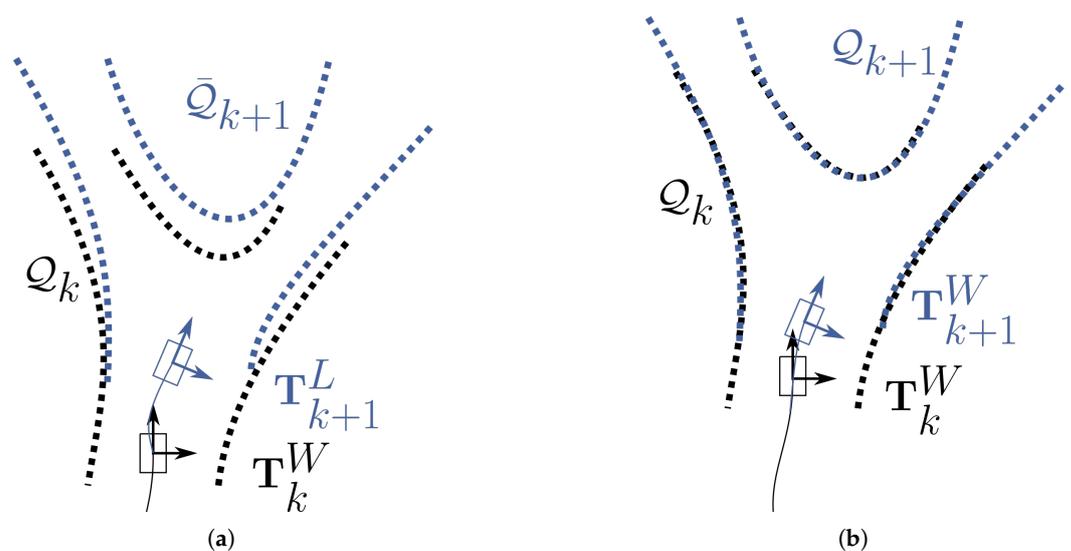
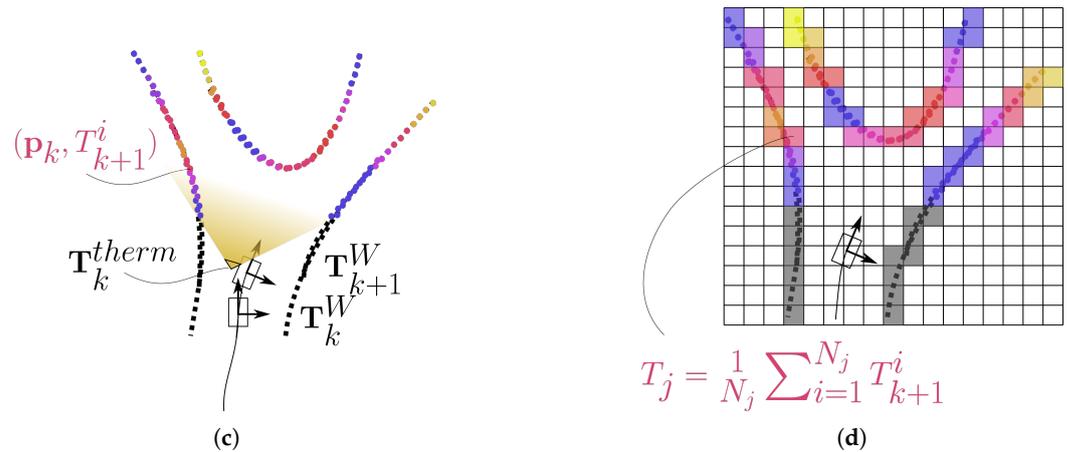


Figure 5. Cont.



**Figure 5.** (a) Acquisition of the point clouds by the LiDAR while the rover is moving. Each set of points is initially expressed in its own reference frame. (b) Reconstruction of the trajectory by the LiDAR SLAM (LOAM) algorithm. The motion estimation  $T_k^I$  provided by the odometry function is combined with the rover positioning  $T_k^W$  determined by the mapping function to register the new point cloud  $Q_{k+1}$  (blue coloured) on the current map, returning the 3D transformation  $T_{k+1}^W$  which adds a new segment to the reconstructed trajectory. (c) After the data fusion has been performed, the sequence of transformations  $T_{1...N}^W$  is used to align the thermal point clouds (rainbow coloured), before they are merged together. (d) The registered and merged thermal point clouds are then discretized with a 3D grid (here shown in section) by an *OcTree*-based spatial decomposition function. In the end, each cell is associated with the average temperature of the points it contains, producing a temperature distribution map.

Registering a point cloud with respect to another, assuming that they overlap at least partially, means finding and applying the rigid transformation that minimizes an appropriate function of the distance between the two sets of points. If each point cloud can be aligned to the previous one, then, by composing all the related transformations, the entire sequence can be expressed in the reference frame of the first point cloud produced. Since the speed at which the LiDAR acquires the points is higher than the movement speed of the rover, motion-induced distortion in the point clouds is expected to be negligible. Moreover, it can be assumed that all transformations include only rotations and translations, without any scaling. In this case, the sequence of rigid transformations that allows one to register the thermal point clouds (see Figure 5c) is provided by the LiDAR SLAM algorithm (see Figure 5b) as the one which describes the rover's trajectory.

The registered thermal point clouds can be merged together, preserving the shape and continuity of the structures (such as plane surfaces, edges, and corners) they contain. However, the merging cannot simply consist of grouping all the points into the same set: doing so would produce excessive concentrations of elements in the overlapping areas between two consecutive point clouds. Therefore, the algorithm processes the registered and merged thermal point clouds with a three-dimensional box filter. After the volume is divided into cubes of constant size, all elements within the same cube are replaced by a new point identified by averaging their coordinates. The purpose here is only to equalize the density of the resulting thermal scene, so the box filter can have a resolution (that is, a number of cells per unit length) much higher than the grid which will later be used for the spatial decomposition algorithm. Notice that the box filter affects all the properties of each point, including its temperature (which was stored as a "point intensity" value in the adopted point cloud model): this means that the new point which replaces all the point cloud elements within a given cubic volume will have a temperature value equal to the average of the temperatures of the point cloud elements it replaces.

Before proceeding with the spatial decomposition, the thermal scene must undergo some post-processing operations. First, to improve the efficiency of the spatial decompo-

sition algorithm, the thermal scene is further downsampled by randomly selecting only a given fraction of its elements. Then the thermal scene could also be leveled by fitting a plane to the available points and determining the 3D rigid transformation that would make that plane normal to a certain reference vector, but only if the surface on which the rover is moving has a constant and known inclination. In all other cases, an “offset” described by the rotation matrix  $\mathbf{R}_0$  (computed for a roll angle  $\phi$  and a pitch angle  $\theta$ ) could instead be applied. The two angles can be derived [28] using the classical relations

$$\begin{aligned}\phi &= \text{atan2}(a_Y, a_Z) \\ \theta &= \text{atan2}\left(a_X, \sqrt{a_Y^2 + a_Z^2}\right)\end{aligned}\quad (14)$$

where  $a_X$ ,  $a_Y$ , and  $a_Z$  are the linear accelerations provided by the LiDAR’s built-in IMU (or by other dedicated onboard sensors installed on the rover). Neither the downsampling nor the leveling affects the temperature values of the elements of the thermal scene.

#### 4.3. Spatial Decomposition

In the previous steps of the thermal mapping process, a set of point clouds and a set of radiometric thermal images were combined into a thermal scene  $S$

$$S = (\mathbf{p}_k, RGB_k, T_k \mid k = 1, \dots, N_p) \quad (15)$$

in which a 3D point  $\mathbf{p}_k = \{X_k, Y_k, Z_k\}^T$  is also associated with a color  $RGB_k$  and an apparent temperature  $T_k$  (as shown in Figure 5c). Notice that the number of points  $N_p$  can be very large, even for a relatively small scene.

The final step of the process aims to discretize the scene  $S$  with a three-dimensional grid, grouping its points into objects that represent finite portions of space. Each of these regions will be associated with the average temperature of the points it contains, building a 3D temperature distribution map with the same resolution as the grid used.

The algorithm used to decompose the scene is based on the *OcTree* model, a hierarchical data structure that recursively subdivides a volume into eight identical elements, also known as *voxels* [29]. This kind of decomposition can be represented by a “tree” whose branches could be pruned or merged to keep the structure memory efficient and quick to access. This form of decomposition was preferred over other possible solutions, such as those based on *kd-tree* data structures, because it guarantees a regular subdivision of space (that is, all the volume elements at a given level of an *OcTree* structure have the same shape and the same size).

To process the scene, the algorithm attaches to each voxel a *bin* of 3D points, represented by their coordinates and also by the index that identifies each of them in the original point cloud. The first bin encompasses the whole scene, and so contains all the points, but can be decomposed into eight children bins, which in turn could be decomposed further. Subdivision stops if an adequate condition is met, indicating that the required map resolution has been reached (or exceeded). Here, the stopping condition is expressed as a lower limit to the acceptable voxel edge length: the actual size of the voxels created, however, is determined only by the size of the volume to be decomposed and by the subdivisions required to reach that threshold.

In the end, to create a temperature distribution map, the bins (and the points they contain) must be processed back into *thermal* voxels (see Figure 5d): regions of space of known size characterized by a temperature value. Here, the algorithm selects only all the bins at a given level that contain at least 8 points, then recovers the temperature values of all the elements inside through their indices and averages them. The resulting voxel temperature

$$T_j = \frac{1}{N_j} \sum_{i=1}^{N_j} T^i \quad (16)$$

can also be expressed as

$$\tau_j = \left\lfloor 1 + 255 \frac{T_j - T_{min}}{T_{max} - T_{min}} \right\rfloor \quad (17)$$

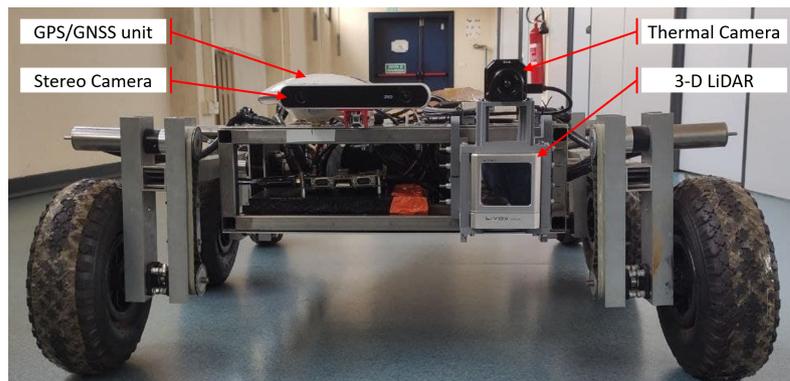
where  $T_{max}$  and  $T_{min}$  represent the maximum and minimum expected temperatures for the scene, respectively, while  $\tau_j$  is used to select one of the 256 RGB triplets of a colormap. This allows us to represent the set of thermal voxels  $V$ :

$$V = (\mathbf{w}_j, T_j, \tau_j \mid j = 1, \dots, N_v) \quad (18)$$

as a set of colored cuboids, identified through the location  $\mathbf{w}_j$  of one of their corners (specifically, the closest to the origin of the world reference frame). Note that the number  $N_v$  of voxels in  $V$  can be expected to be much lower than the number  $N_c$  of points in  $S$ , as the resolution of the map should always be lower than the density of the thermal scene for the mapping process to be effective.

## 5. Experiments

The experimental setup for testing the proposed calibration and thermal mapping algorithms consists of a FLIR Vue Pro R 336 radiometric thermal camera and a Livox Horizon 3D LiDAR. One of the notable features of the LiDAR used is the adoption of a non-repeating scanning pattern, which gradually increases FOV coverage over time (as shown in Table 2). The two sensors can be seen in Figure 6.



**Figure 6.** Instruments installed on board the MORPHEUS rover [30] when the proposed thermal mapping algorithm was tested outdoors. The 3D printed mounting structure, supporting the thermal camera and the LiDAR, was secured to the front of the rover’s frame (on the right), together with a Swift Navigation GPS/GNSS unit and a StereoLabs ZED binocular RGB stereo camera (on the left).

The thermal camera and the LiDAR were placed on a 3D-printed mounting structure and then calibrated. The calibration algorithm described in Section 3 analyzes only specific portions of the input thermal image, and the input point cloud can be cropped so as to include only the target box and its immediate surroundings. This allowed us to acquire the calibration datasets in an environment that was not specifically modified for such a purpose: the only precaution adopted was to briefly heat the target with a heat gun, to produce a suitable thermal contrast with respect to the (much colder) background.

**Table 2.** Instrument specifications.

Thermal Camera	
Detector Technology	Uncooled VOx Microbolometers
Spectral Region	7.5–13.5 $\mu\text{m}$ (LWIR)
Detector Resolution	336 $\times$ 256 pixels
FOV	35° $\times$ 27° (9 mm optics)

Table 2. Cont.

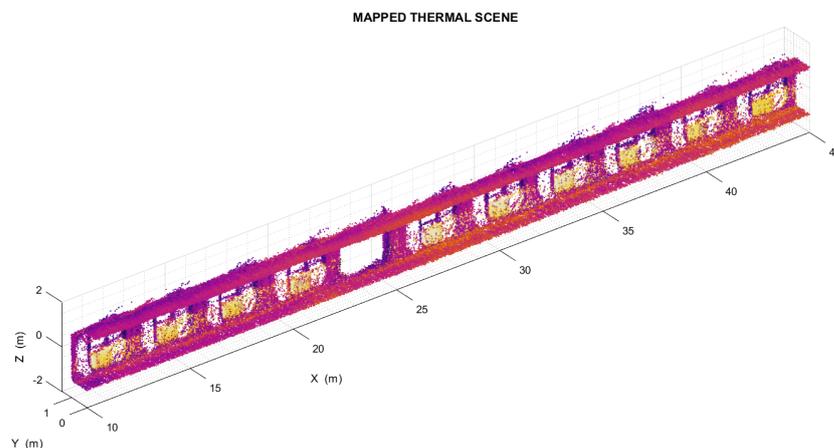
3D LiDAR	
Maximum Detection Range	90 m @ reflectivity 10%
	130 m @ reflectivity 20%
	260 m @ reflectivity 80%
Minimum Detection Range	0.5 m
FOV	81.7° × 25.1°
FOV Coverage	60% @ 0.1 s
	98% @ 0.5 s
Point Acquisition Rate	>240,000 points/s
Random Error ( $1 - \sigma$ @ 20 m)	0.02 m (distances)
	0.05° (angles)

### 5.1. Indoor Test

To test the proposed thermal mapping method in an indoor environment, the instrument mounting (together with a laptop computer and the external battery used to power the LiDAR) was temporarily placed on a small cart and pushed down a flat straight corridor (shown in Figure 7). This area contains a number of thermally distinctive elements, which are regularly arranged throughout its length. These elements can be clearly recognized in the thermal scene (Figure 8) and therefore should appear on the corresponding 3D temperature distribution map as easily identifiable voxel arrangements.



**Figure 7.** Visual (left) and thermal (right) images of the indoor environment used to test the proposed algorithm. Notice how the thermally distinctive elements on both sides of the corridor are still recognizable in the thermal scene (Figure 8) as well as in the corresponding temperature maps (Figure 9a,b).



**Figure 8.** Left side of the thermal scene produced during the test in the indoor environment.

The analysis was performed on two temperature distribution maps of the same environment, each of which was divided into two halves to simplify the representation of the results:

- A *low resolution* map, with a voxel edge size of 0.27 m resulting in  $\sim 3.7$  voxels/m (shown in Figure 9a).
- A *high resolution* map, with a voxel edge size of 0.14 m resulting in  $\sim 7.1$  voxels/m (shown in Figure 9b).

The two maps were created from the same OcTree data structure, so they describe the exact same set of points. Specifically, the high resolution map represents the subdivision from the ninth level of the source structure, while the low resolution map represents the subdivision from the eighth level of the same structure.

To determine whether this map accurately describes the test environment, the spatial distribution of its thermal voxels was analyzed automatically: all voxels with a temperature above a suitable threshold (in this case,  $T_{th} = 37.5$  °C) were extracted and grouped into separate *clusters*, each made up of at least 3 voxels in close proximity to each other (as shown in Figure 10a,b). The distance between these clusters, whose location could be found by averaging the coordinates of the centroids of all their constituent voxels, was compared with the distance between the environmental features that they represent. The spacing between the thermally distinctive elements was previously measured on-site, using a measuring tape whose accuracy of 1 mm is approximately an order of magnitude greater than that achieved by the proposed method.

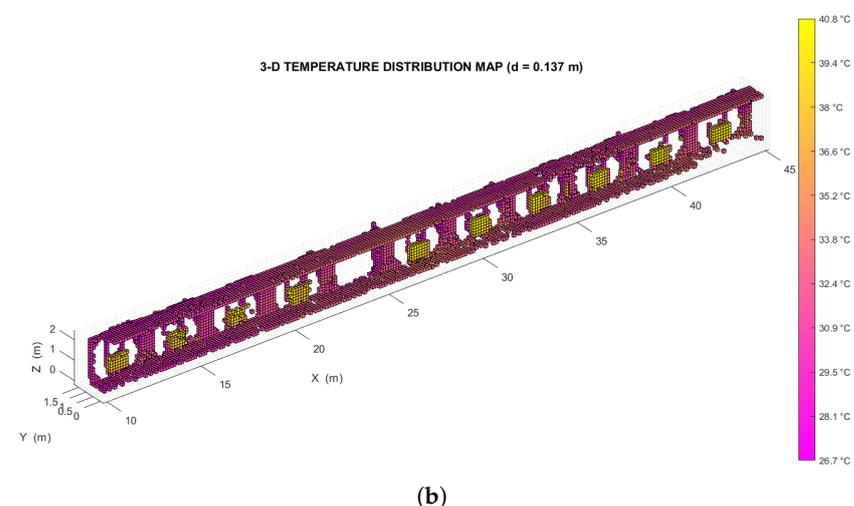
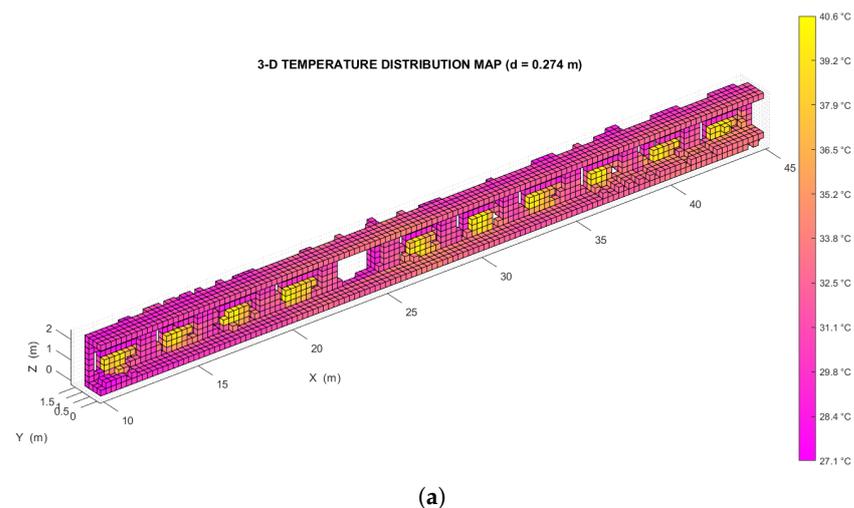
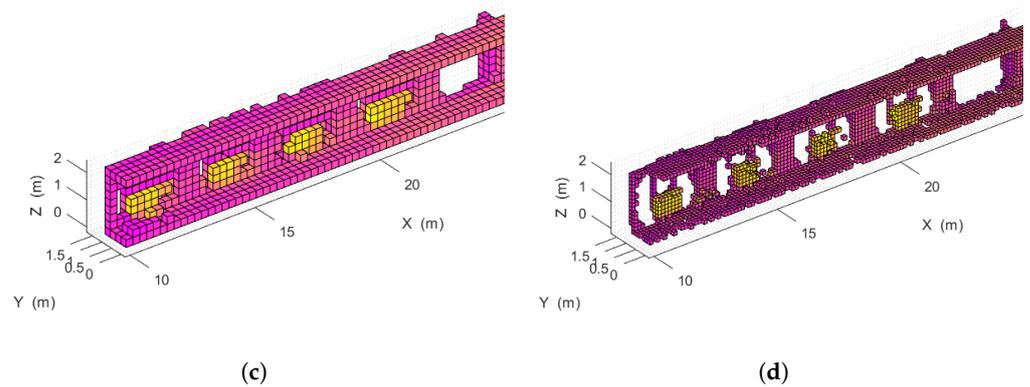
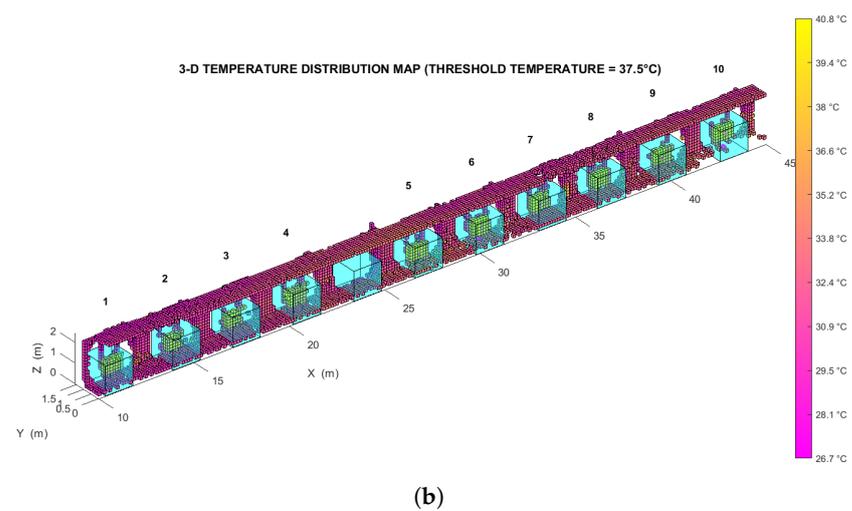
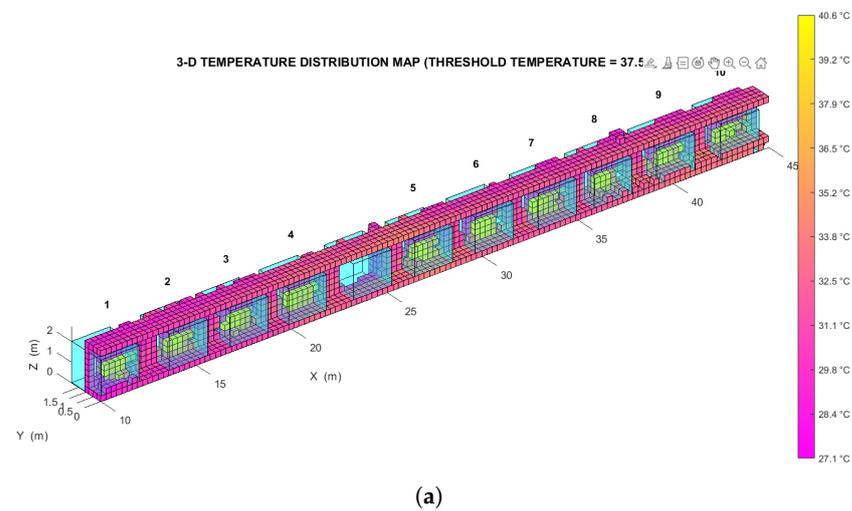


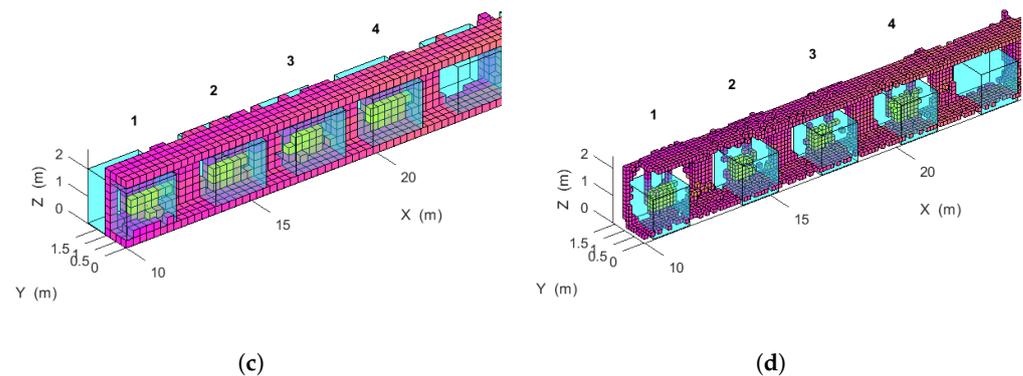
Figure 9. Cont.



**Figure 9.** (a) Left side of the low resolution temperature distribution map relative to the indoor test environment. (b) Left side of the high resolution temperature distribution map relative to the same environment. (c) Detail of the low resolution indoor temperature distribution map. (d) Detail of the high resolution indoor temperature distribution map.

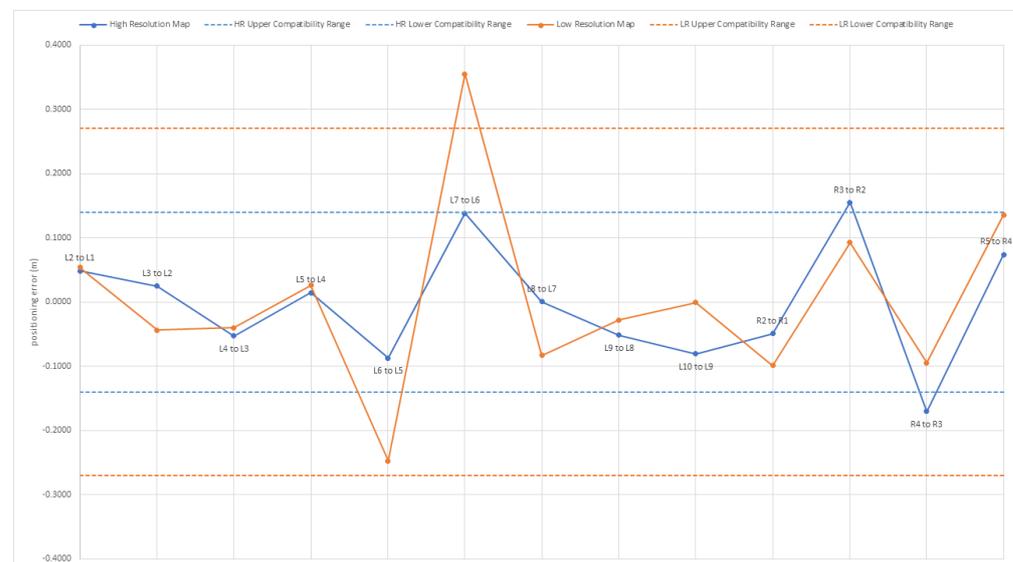


**Figure 10.** *Cont.*



**Figure 10.** (a) Voxel clusters detected, with a threshold temperature of  $37.5\text{ }^{\circ}\text{C}$ , on the left side of the low resolution temperature distribution map relative to the indoor test environment. (b) Voxel clusters detected, again with a threshold temperature of  $37.5\text{ }^{\circ}\text{C}$ , on the left side of the high resolution temperature distribution map relative to the indoor test environment. (c) Detail of the clusters in the low resolution indoor temperature distribution map. (d) Detail of the clusters in the high resolution indoor temperature distribution map.

The plot in Figure 11 illustrates the results of this analysis in the two cases considered, showing the distance error (that is, the difference between the estimated and the expected separation along the X-axis) for each pair of consecutive clusters, on the left and right side of the corridor. Note that to account for the discretization of the thermal scene, a maximum difference equal to the size of a voxel is still considered acceptable. The standard deviation of the error for the low resolution map is  $0.145\text{ m}$ , and the standard deviation of the error for the high resolution map is  $0.093\text{ m}$ .



**Figure 11.** Distance errors relative to the temperature distribution maps produced in the indoor test environment. Dashed lines delimit the compatibility range in the two cases, which is equal to plus or minus the size of one voxel. The error standard deviation has been computed for both the low resolution map and the high resolution map.

These metrological performances are slightly better than those obtained in the previous research work of [14] where the positioning error obtained is equal to  $4\text{ cm}$ . Considering that its test was carried out in a room with a maximum size of  $5\text{ m}$ , it leads to a relative error of  $1\%$ . We obtain a positioning error of  $9\text{ cm}$  (by making a  $14\text{ cm}$  voxelization) in an environment with a maximum length of  $45\text{ m}$ , which corresponds to a relative error of  $0.2\%$ .

## 5.2. Outdoor Test

To test the proposed thermal mapping method in an outdoor environment (shown in Figure 12), the mounting with the thermal camera and the 3D LiDAR was installed on the MORPHEUS rover [30] (shown in Figure 6), designed and built by the students of the University of Padova [30]. When the rover is moving outdoors, the output of the SLAM algorithm can be compared with the positioning data provided by an onboard GPS/GNSS receiver. In this case, Horn's quaternion-based method [31] was used to align the two paths (as shown in Figure 13) and evaluate their compatibility.

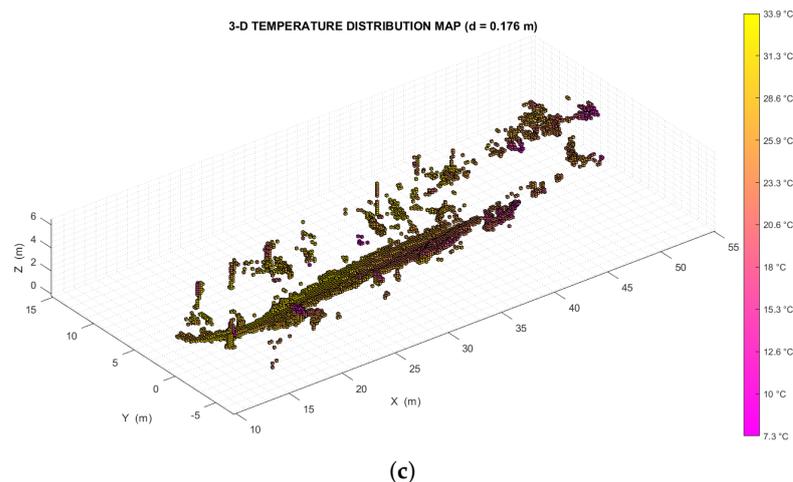
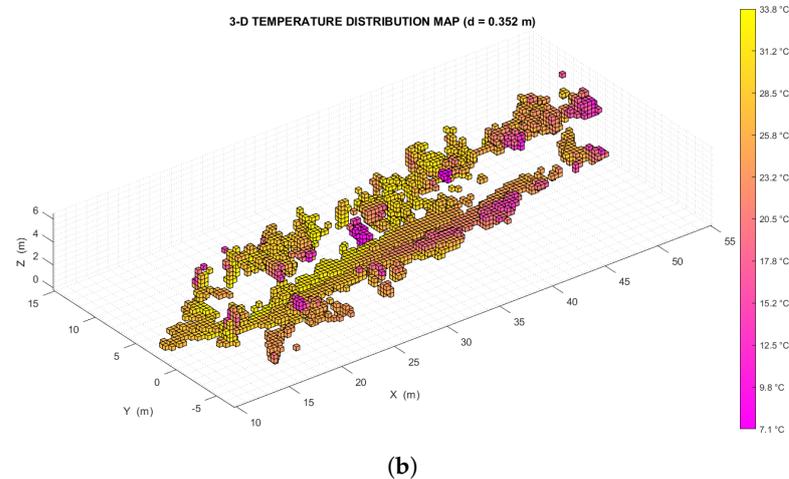
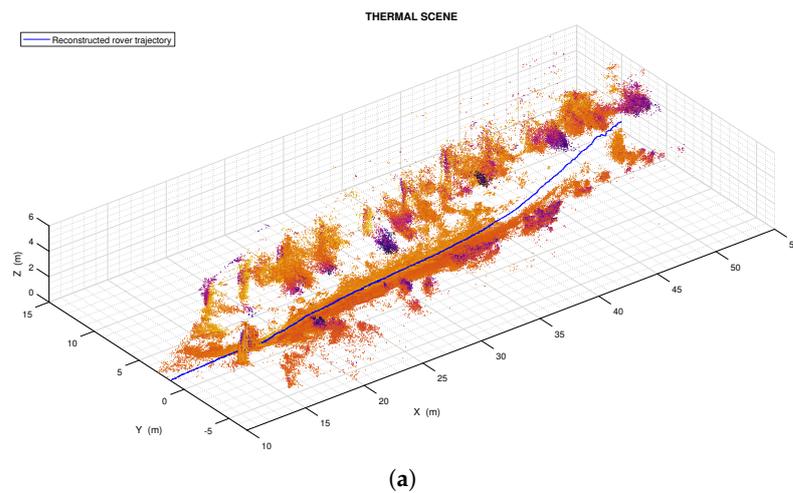
This second test was intended primarily to demonstrate that the *livox\_horizon\_loam* algorithm is capable of reconstructing the trajectory of the rover (as shown in Figures 13 and 14a), thus providing a sufficiently valid basis for the registration of the thermal point clouds, and that the thermal mapping algorithm is capable of producing temperature distribution maps of large size (handling a large number of points and voxels).



**Figure 12.** Visual (left) and thermal (right) images of the outdoor environment used to test the proposed algorithm.



**Figure 13.** Comparison between the trajectory reconstructed in real-time using the *livox\_horizon\_loam* algorithm (blue) and the path traced by the rover's GPS/GNSS receiver (purple). Here the Horn's quaternion-based algorithm was used to align the two paths and to represent the reconstructed trajectory in the same latitude/longitude/altitude coordinate system used for the GPS/GNSS points.



**Figure 14.** (a) Portion of the thermal scene produced during the test in the outdoor environment. Here the blue line represents the estimated trajectory of the rover, which was reconstructed using the *livox\_horizon\_loam* algorithm. (b) Low resolution (voxel size equal to 0.35 m) temperature distribution map relative to the outdoor test environment. (c) High resolution (voxel size equal to 0.18 m) temperature distribution map relative to the outdoor test environment. Again, the high resolution map was obtained from the same OcTree data structure as the low resolution temperature distribution map (with the difference being that the first map corresponds to the eighth level of the structure, and the second to the ninth level).

## 6. Conclusions

This article describes a thermal mapping method based on data fusion between a 3D LiDAR and a thermal imaging camera. The target application sees thermal environment monitoring via UGVs (*Unmanned Ground Vehicles*). The calibration method used to align the reference systems of the LiDAR and the thermal imaging camera is also presented. LiDAR point cloud registration with the UGV relative pose reconstruction was based on a LiDAR SLAM algorithm (*livox\_horizon\_loam*). For efficient representation of the 3D temperature distribution map, a discretization with a voxel grid by means of an OcTree-based spatial decomposition function was performed. The method was characterized in an indoor environment by tracing the presence of hot elements, such as radiators. The standard deviation of the localization error of a thermally significant element has been shown to be equal to 0.145 m for a low resolution map (voxel size equal to 0.27 m) and a standard deviation of 0.093 m for a high resolution map (voxel size equal to 0.14 m). The two maps were created from different levels of the same OcTree structure, that is from the same set of thermally-characterized 3D points. The method was also tested outdoors, by integrating the sensors onboard a UGV. The tests performed showed how adopting too low a map resolution reduces the complexity of the representation, but also its level of detail (which could prevent the detection of some of the features within the map itself). On the other hand, adopting too high a map resolution leads to the number of voxels being comparable to the number of points, preventing the temperature values from being averaged significantly.

**Author Contributions:** Conceptualization, S.C, M.P.; Data curation, D.D.P.; Investigation, S.C.; Methodology, D.D.P. and S.C.; Resources, M.P.; Software, D.D.P.; Supervision, S.C, M.P.; Writing—original draft, D.D.P.; Writing—review & editing, D.D.P., S.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CNN	Computational Neural Network
EPnP	Efficient Perspective- <i>n</i> -Point
FOV	Field Of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IR	InfraRed
LiDAR	Light Detection And Ranging
LOAM	LiDAR Odometry And Mapping
LWIR	Long-Wave InfraRed
MSAC	M-estimator Sample Consensus
NaN	Not a Number
PC	Point Cloud
PnP	Perspective- <i>n</i> -point
RANSAC	RANdom SAMple Consensus
SfM	Structure from Motion
SLAM	Simultaneous Localization And Mapping
TPC	Thermal Point Cloud
UGV	Unmanned Ground Vehicle
VOx	Vanadium Oxide

## References

1. Oreifej, O.; Cramer, J.; Zakhor, A. Automatic generation of 3d thermal maps of building interiors. *ASHRAE Trans.* **2014**, *120*, C1.
2. Zhang, J.; Singh, S. LOAM: LiDAR Odometry and Mapping in Real-Time. *Robot. Sci. Syst.* **2014**, *2*, 1–9.
3. Cruz Ulloa, C.; Prieto Sánchez, G.; Barrientos, A.; Del Cerro, J. Autonomous thermal vision robotic system for victims recognition in search and rescue missions. *Sensors* **2021**, *21*, 7346. [[CrossRef](#)] [[PubMed](#)]
4. Prashar, A.; Jones, H.G. Infra-red thermography as a high-throughput tool for field phenotyping. *Agronomy* **2014**, *4*, 397–417. [[CrossRef](#)]
5. Liu, L.; Mei, H.; Guo, C.; Tu, Y.; Wang, L.; Liu, J. Remote optical thermography detection method and system for silicone polymer insulating materials used in power industry. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 5782–5790. [[CrossRef](#)]
6. Al-Habaibeh, A.; Sen, A.; Chilton, J. Evaluation tool for the thermal performance of retrofitted buildings using an integrated approach of deep learning artificial neural networks and infrared thermography. *Energy Built Environ.* **2021**, *2*, 345–365. [[CrossRef](#)]
7. Cunningham, C.; Nesnas, I.A.; Whittaker, W.L. Improving slip prediction on mars using thermal inertia measurements. *Auton. Robot.* **2019**, *43*, 503–521. [[CrossRef](#)]
8. Ham, Y.; Golparvar-Fard, M. An automated vision-based method for rapid 3D energy performance modeling of existing buildings using thermal and digital imagery. *Adv. Eng. Inform.* **2013**, *27*, 395–409. [[CrossRef](#)]
9. Müller, A.O.; Kroll, A. Generating high fidelity 3-D thermograms with a handheld real-time thermal imaging system. *IEEE Sensors J.* **2016**, *17*, 774–783. [[CrossRef](#)]
10. Vidas, S.; Moghadam, P.; Bosse, M. 3D thermal mapping of building interiors using an RGB-D and thermal camera. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 2311–2318.
11. Vidas, S.; Moghadam, P. HeatWave: A handheld 3D thermography system for energy auditing. *Energy Build.* **2013**, *66*, 445–460. [[CrossRef](#)]
12. González-Aguilera, D.; Rodríguez-Gonzálvez, P.; Armesto, J.; Lagüela, S. Novel approach to 3D thermography and energy efficiency evaluation. *Energy Build.* **2012**, *54*, 436–443. [[CrossRef](#)]
13. Cattini, S.; Cassanelli, D.; Di Cecilia, L.; Ferrari, L.; Rovati, L. A Procedure for the Characterization and Comparison of 3-D LiDAR Systems. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–10. [[CrossRef](#)]
14. Borrmann, D.; Nüchter, A.; Đakulović, M.; Maurović, I.; Petrović, I.; Osmanković, D.; Velagić, J. A mobile robot based system for fully automated thermal 3D mapping. *Adv. Eng. Inform.* **2014**, *28*, 425–440. [[CrossRef](#)]
15. Dino, I.G.; Sari, A.E.; Iseri, O.K.; Akin, S.; Kalfaoglu, E.; Erdogan, B.; Kalkan, S.; Alatan, A.A. Image-based construction of building energy models using computer vision. *Autom. Constr.* **2020**, *116*, 103231. [[CrossRef](#)]
16. Daffara, C.; Muradore, R.; Piccinelli, N.; Gaburro, N.; de Rubeis, T.; Ambrosini, D. A cost-effective system for aerial 3D thermography of buildings. *J. Imaging* **2020**, *6*, 76. [[CrossRef](#)]
17. Rangel, J.; Soldan, S.; Kroll, A. 3D thermal imaging: Fusion of thermography and depth cameras. In Proceedings of the International Conference on Quantitative InfraRed Thermography, Bordeaux, France, 7–11 July 2014; Volume 3.
18. Adamopoulos, E.; Patrucco, G.; Volinia, M.; Giroto, M.; Rinaudo, F.; Tonolo, F.G.; Spanò, A. 3D Thermal Mapping of Architectural Heritage. In Proceedings of the Euro-Mediterranean Conference, Virtual Event, 2–5 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 26–37.
19. Campione, I.; Lucchi, F.; Santopuoli, N.; Seccia, L. 3D thermal imaging system with decoupled acquisition for industrial and cultural heritage applications. *Appl. Sci.* **2020**, *10*, 828. [[CrossRef](#)]
20. Shin, Y.S.; Kim, A. Sparse depth enhanced direct thermal-infrared SLAM beyond the visible spectrum. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2918–2925. [[CrossRef](#)]
21. Pusztai, Z.; Hajder, L. Accurate Calibration of LiDAR-Camera Systems using Ordinary Boxes. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 394–402.
22. Quan, L.; Lan, Z. Linear n-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 774–780. [[CrossRef](#)]
23. Forsyth, D.A.; Ponce, J. *Computer Vision: A Modern Approach*; Prentice Hall: Hoboken, NJ, USA, 2011.
24. Zisserman, A.; Hartley, R. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
25. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
26. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.
27. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [[CrossRef](#)]
28. Merhav, S. *Aerospace Sensor Systems and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1998.
29. Meagher, D. Geometric modeling using OcTree encoding. *Comput. Graph. Image Process.* **1982**, *19*, 129–147. [[CrossRef](#)]
30. Chiodini, S.; Giubilato, R.; Pertile, M.; Tedesco, A.; Accardo, D.; Debei, S. Design of a user-friendly control system for planetary rovers with CPS feature. In Proceedings of the 2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Naples, Italy, 23–25 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 317–321.
31. Horn, B.K. Closed-form solution of absolute orientation using unit quaternions. *Josa A* **1987**, *4*, 629–642. [[CrossRef](#)]