GPU-BASED MAPPING OF THERMAL IMAGERY FOR GENERATING 3D OCCLUSION-AWARE POINT CLOUDS

Alfonso López, Juan M. Jurado, Carlos J. Ogayar, Francisco R. Feito

Department of Computer Science, University of Jaén

IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium | 978-1-6654-2792-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IGARSS46834.2022.9884240

ABSTRACT

This work describes an efficient approach for generating large 3D thermal point clouds considering the occlusion of camera viewpoints. For that purpose, RGB and thermal imagery are first corrected and fused with an intensity correlation-based algorithm. Then, absolute temperature values are obtained from the normalized data. Finally, thermal imagery is mapped on the point cloud using the Graphics Processing Unit (GPU) hardware. The proposed occlusion-aware mapping algorithm is massively parallelized using OpenGL's compute shaders. Our solution allows generating dense thermal point clouds in a lower response time compared with other notable software solutions (e.g., Agisoft Metashape or Pix4Dmapper) that yield results with a significantly lower point density.

***Index Terms*—** Thermography, 3D point cloud, GPGPU, Occlusion, UAV, Thermal Mapping, 3D reconstruction

1. INTRODUCTION

Thermography or Infrared (IR) thermal imaging is one of the main data sources concerning Remote Sensing (RS). Although visual cameras have been a standard imaging tool, thermal devices avoid some of their main drawbacks. These sensors are based on passive sensing technology, and therefore, they do not rely on external energy sources. Hence, thermal cameras allow describing a surface through its temperature. Thermal measurements can be acquired from airborne and satellite platforms, though their spatial and temporal resolution is less adequate for fine-grained monitoring tasks. Thus, Unmanned Aerial Vehicles (UAV) emerge as an appropriate platform due to their lower altitude, together with low-cost and lightweight thermal devices. However, consumer-grade thermal devices are less prohibitive at the expense of lower resolution and more sensor defects, such as some noise sources causing thermal results to appear blurred and smoothed out.

A fundamental problem in RS is the generation of 3D environments to assess image features on 3D geometry, mainly presented as point clouds. Regarding thermography, the described defects harden the reconstruction of 3D models. As a result, this problem has been previously addressed in the literature using multiple approaches. Photogrammetric techniques such as Structure from Motion (SfM) represent a baseline for modeling 3D thermal point clouds, as this procedure is part of some relevant software applications. However, the acquired information depends on the surface emissivity, and thus the identification of keypoints can lead to erroneous reconstructions [1, 2]. Furthermore, photogrammetric approaches are prone to errors when images show repetitive patterns and uniform textures [3].

Instead of solely using thermal imagery, previous studies investigated the generation of 3D point clouds with an alternative data source for projecting thermal imagery, mainly RGB imagery due to its higher resolution. This approach allows generating much more dense thermal point clouds. However, these methods need either calibrating both sensors or identifying features to fuse both data sources. To accurately capture common keypoints, multiple features descriptors have been proposed, such as Sobel, Canny and Hough transformations, as well as improvements of Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) operators [4]. However, some feature descriptors highlight edges and corners barely recognizable in natural environments. Unlike previous descriptors, the Enhanced Correlation Coefficient (ECC) consists of an optimization problem whose objective function is the image correlation [1]. Finally, 2.5D point clouds have also been addressed by combining thermal orthomosaics and 3D RGB point clouds.

Beyond the challenges of the procedure itself, building a 3D model is a time-consuming task, even for popular software taking advantage of CUDA-compatible GPUs. Recently, large point clouds with millions of points have been processed in modern GPUs for real-time rendering [5], whereas occlusion has barely been addressed in point clouds [6]. It can be approached by estimating a triangle mesh from k-nearest neighbors (KNN), although it is not suitable for complex scenarios, whereas the semantic segmentation of the point cloud also helps detecting occlusion. In this work, z-buffers are proposed to handle occlusion for the mapping process, frequently implemented as a sequential algorithm. Hence, the main contribution of this work is the estimation of large, occlusion-aware, 3D thermographic point clouds using a massively parallel approach to accelerate the mapping process. The core is an accurate registration of RGB and thermal images described in recent work [1].

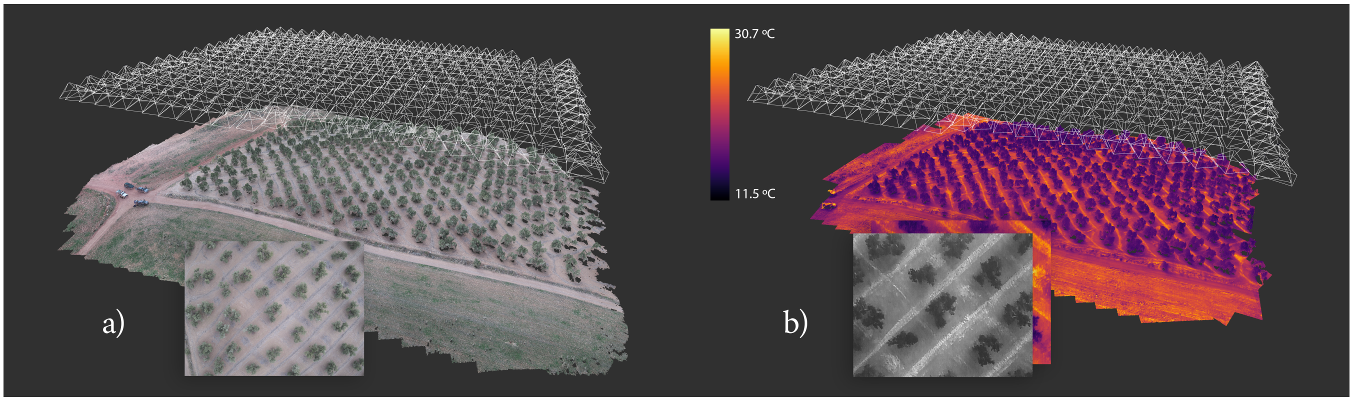

Fig. 1. a) RGB point cloud and b) its corresponding thermal reconstruction. Viewpoints of RGB images are also depicted above both point clouds. The thermal point cloud is colored according to a color-ramp applied to the temperature.

2. METHODOLOGY

This section briefly describes the mapping methodology, based on previous work [1], and the proposed parallel occlusion detection on the GPU.

2.1. Fusion of thermal and RGB imagery

Registering visible and thermal data poses multiple challenges regarding imaging defects and differences for each camera viewpoint [1]. Firstly, RGB images show a barrel effect due to their wide-angle lens. As opposed to visible imagery, thermal data presents a pincushion distortion because of its small field of view. However, distortion can be removed through the camera matrix, K, radial and tangential distortion coefficients $(k_1, k_2, p_1, p_2, k_3)$. Such correction is achieved by performing an inverse transformation, i.e., pixels from the undistorted image compute the corresponding position from the original image, thus avoiding unfilled values in the result.

Once corrected, both images are fused using the ECC algorithm. For that purpose, images are processed by pairs and linked to a homography matrix returned by ECC. This transformation matrix yields a local maximum correlation for the intensity in both images. Therefore, we use a method robust to non-linear intensity variations between thermal and visible imagery, instead of correlating the extracted image features. To apply the ECC algorithm, RGB images are cropped and downscaled to thermal resolution, and thus the resulting composite matrix C_i allows to project RGB pixels into the thermal image plane. However, note that some points may be out of boundaries.

2.2. Computation of temperature

Although we aim to query absolute temperature values, grayscale thermal data represents a normalized value from the temperature acquired on the surveyed scene. The projection of normalized data to absolute values depends on several environmental parameters and factors defined by the own manufacturer, most of them defined in the embedded metadata. In this work, the temperature is computed using the formulae suitable for most FLIR devices [7].

2.3. Thermal image mapping

Besides generating 3D thermal point clouds, this work aims to provide dense point clouds with visible and thermal data. Thus, this section parts from an accurate RGB point cloud previously generated through SfM with high density. From this point cloud and the calibration of RGB cameras, a projection matrix P_i can be estimated as $K \cdot \left[R| - Rt_{local}\right]$ for each camera, where K represents 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, and t_{local} is the camera position in the local coordinate system of the point cloud. Consequently, P_i allows projecting 3D points into RGB imagery, whereas C_i outputs the corresponding thermal coordinates.

2.4. Occlusion

The projection, as described, omits the occlusion problem. Hence, thermal samples from foreground surfaces could be attached to background surfaces (see Figure 2). Several approaches have been proposed to deal with occlusion, mainly split into volumetric and 2D. The main drawbacks of volumetric methods are given by time-consuming spatial searches and the selection of volumetric shapes to capture spatial occlusion. On the other hand, 2D-based approaches use the well-known z-buffer to map each pixel with one 3D point at most. This process is even integrated with GPU-based rendering pipelines to draw a scene. However, general-purpose shaders can solve the occlusion problem with lower response time [6].

2.4.1. *z-buffer modeling*

Compute shaders are general-purpose shaders for GPGPU (general-purpose computing on graphics processing units) programming that can be used for tasks not related to rendering. However, it is also convenient to speed up simple rendering tasks, as it omits unnecessary pipeline stages [5]. As opposed to the rendering pipeline, z-buffers are not self-contained in this shader stage and, therefore, must be implemented through a Shader Storage Buffer Object (SSBO). These buffers present a limited capacity, whereas their data transfer is also time-consuming, either $CPU \rightarrow GPU$ or vice-versa. Another main challenge of building a z-buffer is given by the concurrent access to buffer indices, which can be avoided using atomic blocks. Consequently, values can be modeled as integers of 64 bits (`uint64_t`), storing the minimum distance observed and the corresponding 3D point's index, where the most significant bits correspond to the distance. Thus, a minimum atomic operator (`atomicMin`) selects the nearest distance while carrying the point index. Through this encoding, we can handle large point clouds of up to 2^{32} points. Regarding distance encoding, GLSL allows transforming floating-point values into unsigned integers (`floatBitsToUint`) that can be shifted 32 bits to occupy the most significant bits.

The z-buffer is initially filled with ∞; thus, a null point index is given by $2^{32}-1$. Once computed, it can be downscaled to half its size by extracting indices and discarding distance information, thereby reducing the size and response time of data transfers from GPU to CPU.

2.4.2. *Point cloud ordering*

The objective of altering the point order in memory is to evaluate its impact on the performance of the proposed occlusion method. Initially, points present an unknown order after being processed by an external software tool. Thus, two different layouts are proposed here. First, points can be randomly shuffled, thereby eliminating data locality. Then, we can group points with similar mapping results in close buffer indices by computing the Z-order curve (also known as Morton curve). By ordering the 3D point cloud, points discarded or visible from a viewpoint are clustered throughout the sorted buffer, while also grouping the updates of the GPU buffer. To sort the point cloud, we implement the Radix Sort algorithm on the GPU.

2.4.3. *Depth buffer representation*

As a result of the limited resolution of thermal imagery, their pixels represent a wide area. Therefore, the depth buffer should adopt higher dimensions for generating large point clouds. Otherwise, a significant amount of 3D points would be discarded by projecting them to the same pixel from a camera viewpoint. Accordingly, thermal dimensions can be divided by a factor, d, thereby generating cells of length $\frac{1}{d}$. An optimal value of d must be set so that the occlusion is correctly detected and the algorithm does not significantly increase its response time.

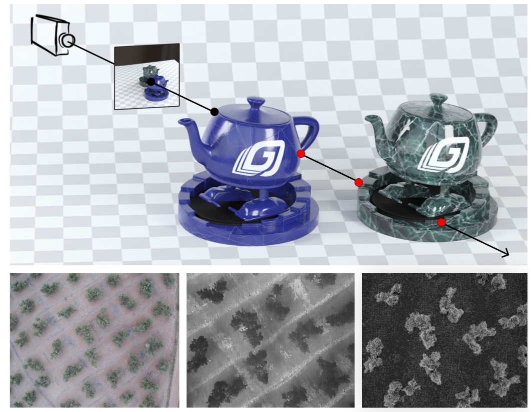

Fig. 2. Occlusion overview, where the left model is occluding the right mesh in a pixel. a) RGB image, b) thermal image and c) z-buffer of the corresponding thermal image.

3. EXPERIMENTAL RESULTS

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 thermal reconstructions from these solutions also present some geometrical errors. The evaluation was performed on a PC with Intel Core i7-7700 3.6 GHz, 16 GB RAM, GTX 1070 GPU (Pascal architecture) and Windows 10 OS. The proposed methodology is implemented in C++ along with OpenGL (Open Graphics Library) for rendering and massively parallel computing tasks. Besides GPU computing, some methods on the CPU side are also parallelized using the multiprocessing OpenMP framework.

As a case study, we evaluated the performance in a dataset with RGB and thermal images acquired by the dual-device DJI Zenmuse XT2, consisting of 410 image pairs from an olive orchard. The reconstructed RGB point cloud, used as a reference for the proposed method, consists of 98M points. Different depth buffer resolutions ($d = 1$ and $d = 10$) are also assessed. Regarding point cloud ordering, points are initially shuffled to show the benefits of spatial sorting. The results shown here are averaged over four executions. The preprocessing stage includes the registration of thermal and RGB images for the proposed method.

As depicted in Figure 3, the best configuration of our method outperforms other commercial solutions in terms of both response time and point cloud size. Agisoft Metashape

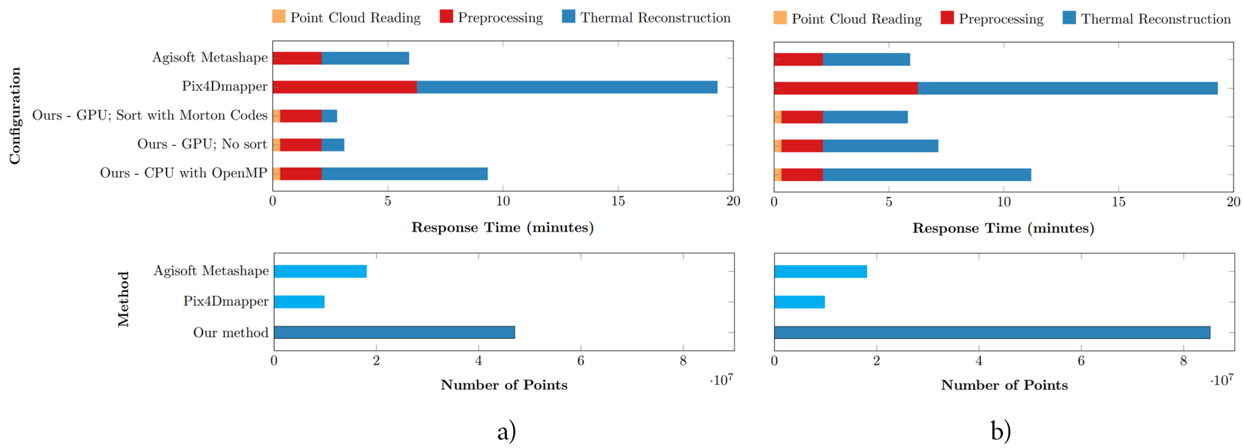

Fig. 3. Evaluation of the proposed method. a) Reconstruction using a depth buffer with a dimension resize factor of $d = 1$, whereas b) uses $d = 10$. Top row shows the response time per stage, while bottom charts report the resulting point cloud size.

also presents a competitive performance, although the point cloud with maximum quality solely consists of 18M points. On the other hand, the baseline of our method is an RGB point cloud with 98M points. With $d = 10$, we achieve a thermal point cloud with 84M points, whereas the response time is similar to the CUDA-based approach of Agisoft Metashape.

4. CONCLUSIONS

Current state-of-art solutions to build thermal point clouds are limited by thermal resolution or present intricate approaches for fusing visible and thermal imagery. In addition to mapping, occlusion also poses a challenge regarding reconstruction efficiency. Using the ECC algorithm, images are rapidly registered regardless of the observed environment. To avoid a high response time, GPU hardware can be used to speed up the processing pipeline. Furthermore, the OpenGL API and its new extensions reduce the program complexity. As observed, time-consuming 3D reconstructions can be completed in a few minutes at most for complementary data sources, such as thermal or multispectral imagery, while outperforming commercial software in terms of point cloud size.

5. ACKNOWLEDGMENTS

This work 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 TIN2017-84968-R.

6. REFERENCES

[1] Alfonso López, Juan M. Jurado, Carlos J. Ogayar, and Francisco R. Feito, "An optimized approach for generating dense thermal point clouds from UAV-imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 182, pp. 78–95, Dec. 2021.

[2] Yingying Kong, Henry Leung, Bowen Zhang, Shiyu Xing, and Yingying Chen, "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)*, Dec. 2018, pp. 307–310.

[3] Małgorzata Jarzabek-Rychard, Dong Lin, and Hans-Gerd Maas, "Supervised Detection of Façade Openings in 3D Point Clouds with Thermal Attributes," *Remote Sensing*, vol. 12, no. 3, pp. 543, Jan. 2020, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[4] L. Hoegner, S. Tuttas, Y. Xu, K. Eder, and U. Stilla, "Evaluation of Methods for Coregistration and Fusion of Rpas-Based 3d Point Clouds and Thermal Infrared Images," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 49B3, pp. 241–246, June 2016.

[5] Markus Schütz, Bernhard Kerbl, and Michael Wimmer, "Rendering point clouds with compute shaders and vertex order optimization," *Computer Graphics Forum*, vol. 40, no. 4, pp. 115–126, 2021.

[6] Alfonso López, Juan Manuel Jurado, Emilio José Padrón, Carlos Javier Ogayar, and Francisco Ramón Feito, *Comparison of GPU-based Methods for Handling Point Cloud Occlusion*, The Eurographics Association, 2021.

[7] Giordano Teza and Arianna Pesci, "Evaluation of the temperature pattern of a complex body from thermal imaging and 3d information: A method and its matlab implementation," *Infrared Physics and Technology*, vol. 96, pp. 228–237, 2019.