

# Digital Terrain Model From UAV Photogrammetric Data

Jules Morel<sup>1</sup> , Alexandra Bac<sup>2</sup>  and Takashi Kanai<sup>1</sup> 

<sup>1</sup>Graduate school of Arts and Sciences, The University Of Tokyo, Japan

<sup>2</sup>Laboratoire d'Informatique et des Systèmes, Aix Marseille University, France

---

## Abstract

*This paper presents a method designed to finely approximate ground surfaces from UAV photogrammetric point clouds by relying on statistical filters to separate vegetation from potential ground points, dividing the whole plot in similar complexity sub-plots through an optimized tiling, and filling holes by blending multiple local approximations through the partition of unity principle. Experiments on very different terrain topology show that our approach leads to significant improvement over the state-of-the-art method.*

## CCS Concepts

• **Computing methodologies** → **Point-based models; Mesh models; Modeling methodologies;**

---

## 1. Introduction

Digital Terrain Model (DTM) acquisition and extraction is a key issue in a large field of applications ranging from geosciences to environmental science, water management, archaeology, but also civil engineering and forest management. Manual data collection methods by surveyors is a repetitive and onerous task which, moreover provides only sparse data. Therefore, laser scanners (or LiDAR) have become a major tool in the last twenty years. Thanks to technological advances, airborne LiDAR has been first enriched by terrestrial LiDAR, and then, some ten years ago, by UAV (Unmanned Aerial Vehicle) LiDAR. However, both terrestrial and aerial acquisition campaigns entail costs ranging from 10 to 100k€ while UAV alternatives are sensibly cheaper. Therefore, such solutions remain hardly available in many applications and contexts (such as emerging countries or small scale studies).

In this context, UAV photogrammetric data appear as a promising new low-cost alternative. However, despite their nice aspect, data present serious shortcomings when it comes to capturing a 3D scene. As illustrated in Figure 1, points are mostly “surface points”; hence the cloud contains hardly any information below vegetation. As a consequence, this context is a real challenge in terms of DTM segmentation and reconstruction. In the present work, we propose a method inspired by the work on terrestrial LiDAR data [MBK20]. Specifically, we introduce a novel segmentation algorithm to efficiently separate the point clouds that make up the terrain surface from vegetations, followed by an implicit surface reconstruction optimized for such incomplete data. We also propose an optimal tiling method so as to tightly fit the large region of terrains to implicit surfaces with less artifacts.

The paper is organized as follows: in Section 2, we describe UAV photogrammetric data and the corresponding state-of-art for DTM

extraction and reconstruction. In Section 3, we introduce our approach and Section 4 details its validation. In Section 5, we conclude on both the feasibility of DTM extraction from UAV photogrammetric data and on our method.

## 2. Overview of data and methods

### 2.1. UAV photogrammetric data

Whereas LiDAR data characteristics are now well-known from the computer science community as well as “end users” such as ecologists, geomaticians, archaeologists or geologists, UAV photogrammetric data are, to the best of our knowledge, a new frontier still largely unexplored. However, they offer a very interesting alternative to LiDAR data in terms of cost-benefit ratio. Despite their outreach, LiDAR technologies (airborne, terrestrial and unmanned) remain very costly. Airborne LiDAR acquisition campaigns provide point cloud covering hundreds to thousands of kilometers with a resolution around 20 cm. Terrestrial LiDAR is both more local (usually around 100m) and more precise (resolution is around 1 cm at 10 m from the scanner). Lastly, UAV LiDAR provides an intermediate solution with a resolution ranging from 2 to 5 centimeters and plots covering up to some dozens of kilometers.

In this context, UAV photogrammetric data provide a new low-cost alternative. However, data have very different characteristics. Whereas their resolution remains low (between 5 and 10 centimeters), points are mostly “surface” points, as the passive photographic sensor does not have the penetrating capacity of the LiDAR. Hence, data can hardly be obtained below the vegetation. Figure 1 shows an example of UAV photogrammetric data and two closer views: one on an isolated tree and the other on dense vegetation.



**Figure 1:** UAV photogrammetric data: (up) a general view on mangalam data (South India) - (down) closer views on (left) area (1), (right) area (2).

Therefore, whereas the resolution of data is relatively similar to UAV LiDAR data, ground occlusions are extremely large in the presence of vegetation. The objective of this paper is to prove that in this challenging context, it is still possible to extract a digital terrain model (DTM) despite these occlusions.

Another characteristic of UAV photogrammetric data is the presence of low noise points located under the ground surface. The first row of Figure 3 illustrates this phenomenon which originates in the matching of original 2D images.

To validate our approach, we tested it on 10 very different data sets which are further described in Section 4.1.

## 2.2. Related work

The unstructured, non-planar nature of point clouds makes implicit surfaces a key modeling tool [MBV18]. Moreover, such models structurally smooth the noise by approximating the input points and are tolerant to in-homogeneity and limited occlusion. In the case of DTM extraction from terrestrial LiDAR data, the combination of local implicit patches blended together to give a global surface model delivers sharp elevation models [MBV17, MBK20]. While LiDAR stands as the most accurate solution to retrieve a parcel ground surface, UAV photogrammetric remote sensing is a interesting low-cost alternative. Compared to traditional aircraft usually used during airborne survey, UAV have the ability to acquire accurate data in critical situations where immediate access to 3D geo-information is crucial [RBN\*11, GH15]. Indeed, they can be

deployed promptly on site and they take-off and land vertically, thus no runway area is required. Moreover, the use of simple digital camera instead of costly LiDAR scanner reduces considerably the survey fare. Indeed, being more accessible, this technology can substitute LiDAR sensors on projects that does not require millimeter accuracy [SBP\*18, CLF\*19, TPDF14]. The extraction of DTM on rocky terrain, without vegetation nor building have been explored by Uysal et al. [UTP15]. However, to the best of our knowledge, the extraction of DTM on landscapes presenting trees and buildings has been explored only by Polat et al. [PU17], who apply the cloth-simulation filtering (CSF) [ZQW\*16].

## 3. Approximation of the terrain surface

As illustrated in Figure 2, our DTM extraction is based on three steps. From the raw 3D point cloud, we first select the ground points. Then we divide the ground points extent along a 2D grid to simplify the local approximation and reduce potential side effects. Finally, we estimate the terrain model on the whole extent as an implicit surface and extract its zero level-set. The following sub-sections describe those three steps.

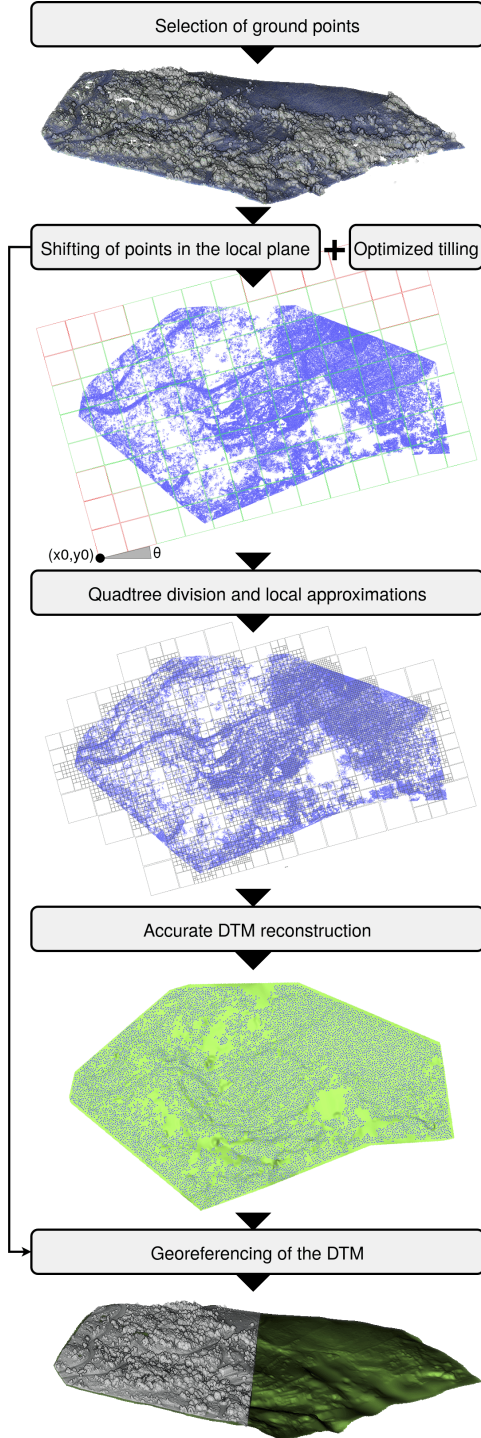
### 3.1. Selection of the ground points

As pointed out in the previous section, the first step of DTM reconstruction consists in segmenting data to extract ground points. In order to do so, we designed a sequence of four filters represented in Figure 3.

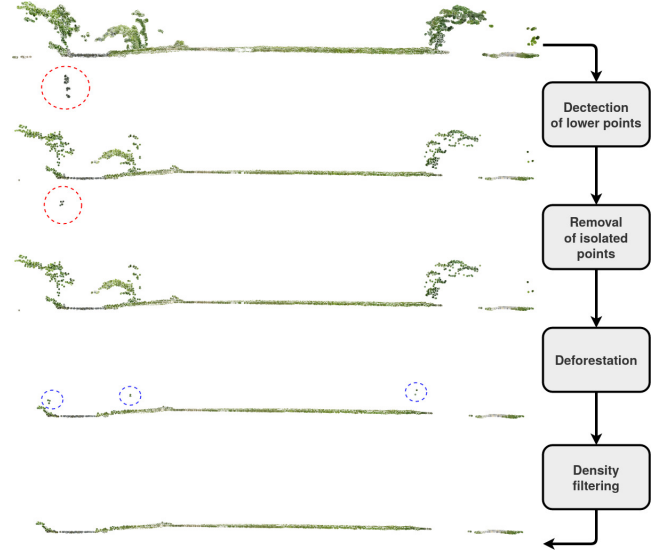
To detect ground points, we rely on the following observation. First, the terrain is clearly described by points of lower elevation. However, as pointed out earlier, point clouds generated from imagery by photogrammetry software happen to produce low noise points. Therefore, we plunge data into a regular 2D grid with cells of size 1 m. But instead of simply selecting points of lower elevation, we apply the following rule: when cells contain more than 20 points, we keep the point whose elevation is closer to the  $n$ th percentile in elevation and the point of lower elevation otherwise. In our experiment  $n = 20$  gives the best result.

Unfortunately, this is not sufficient to select only ground points. First, it is well known that such an approach fails in areas where the terrain is occluded. On Figure 1, it is clear that a cell located inside the occlusion will provide a point in the crown. Moreover, this absence or lack of terrain points also shifts the elevation distribution and hence, low noise points may have been selected (usually as isolated points in the neighborhood of occlusions). Therefore, our second step consists in removing isolated points by a simple analysis of the 27-neighborhood of each cell of the regular grid. We delete points having less than  $n_{\text{neigh}}$  neighbors in this sense (experimentally,  $n_{\text{neigh}} = 6$  gives good results).

Then deforestation consists in removing vegetation points. This problem has been largely studied for LiDAR data. Our data is now sparse and quite similar to airborne LiDAR data. We thus decided to use the approach of [ILS\*06] (inspired from [Axe00]), which triangulates data and then iteratively analyzes steep slope changes. We used the implementation provided by Lastool software [Lse] as the module *lasground*.



**Figure 2:** General overview of our DTM reconstruction method. On the second row, the green cells of the 2D grid are considered for the reconstruction whereas the red ones are discarded. For visualization purpose, the point clouds on the fourth row have been decimated, and only half of the point cloud of the last row is plotted.



**Figure 3:** Filtering of ground points: 1 - Detection of lower points; 2 - Removal of isolated noise points; 3 - Deforestation; 4 - Density filtering. The noise points coming from photogrammetric photo mismatches are circled in red. The vegetation points remaining after deforestation are circled in blue.

However, some vegetation points remain after the previous deforestation step. It is often the case while dealing with forested areas, where the previous step filters out the points corresponding to the highest trees but leaves the ones corresponding to the lowest vegetation. Under the assumption that, in photogrammetric point clouds, ground points are distributed in clusters, we add a filter based on the local density of points. Let  $P = \{\mathbf{p}_i; i = 1 \dots N\}$  denote the point cloud extracted by deforestation. We assign a weight  $d_i$  at each point  $\mathbf{p}_i$  given by :

$$d_i = 1 - \frac{1}{\max_d} \sum_{\mathbf{p}_j \in \mathcal{P}_j^K} \|\mathbf{p}_i - \mathbf{p}_j\| \quad (1)$$

where  $\mathcal{P}_j^K$  is the K-neighborhood of  $\mathbf{p}_i$  (from our experiments  $K = 20$  appears as a good choice), and  $\max_d$  is defined as:

$$\max_d = \max_{\mathbf{p}_i \in P} \left( \sum_{\mathbf{p}_j \in \mathcal{P}_j^K} \|\mathbf{p}_i - \mathbf{p}_j\| \right) \quad (2)$$

Filtering the points of lower weights allows keeping a set of ground points which we rely on.

### 3.2. Optimized tiling

As illustrated in Figure 2, to cope with the average slope present in some drone data-sets, we define, as a preliminary step, a local coordinate system lined up with this slope. In practice, an approximate tangent plane is computed using least square fitting. Let  $(u, v, w)$

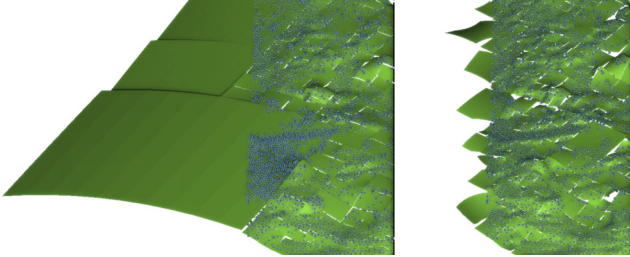


denote a local orthonormal coordinate system such that the direction  $w$  is orthogonal to the fitting plane. The rotation induces by the transformation of the points coordinate in  $(u, v, w)$  and also the translation induced by the map projection are stored in a  $4 \times 4$  matrix  $M$ . Our algorithm performs every operations in the coordinate system  $(u, v, w)$ , then project back the results into the initial map projection by applying the transformation  $M^{-1}$ .

In this local frame, terrain reconstruction is done based on the work [MBK20], developed for terrestrial LiDAR data. This algorithm builds upon a quad-tree division of space to control the reconstruction process according to the local point density. Terrain is then modeled as an implicit function obtained from local quadratic approximations attached to quad-tree leaves, and merged using compactly supported partitions of unity. The support of merging functions is of course proportional to the size of leaves.

Therefore, while quad-tree subdivision guides the multi-scale reconstruction of the surface, it can also lead to surface model artifacts depending on the distribution of points near the boundary of the bounding box. Indeed, large areas without points near the corners tend to block the subdivision of the quad-tree, thus producing large quad-tree cells, as illustrated on Figure 4, and hence large supports for corresponding partitions of unity, which screen the reconstruction of neighboring cells.

To cope with this issue, and also because the photogrammetric point clouds usually cover larger areas than TLS, we first divide the data-sets along with a regular 2D grid, which orientation is optimized according to the points distribution. We then run our DTM algorithm in each cell of this grid.



**Figure 4:** Local quadratic approximations (left) without optimized tilling and (right) with optimized tilling.

In our case, a 2D grid efficiently divides the point cloud if, in each cell of the grid, either points cover the largest part of the cell surface, or the cell is empty. To find such an optimal 2D grid, we define an objective function  $f$  depending on  $x_0$  and  $y_0$  the origin of the grid,  $\theta$  its angle to the  $x$ -axis and its resolution  $l_{cell}$ . Let  $P_i$  denote the subset of points in the  $i$ th cell, and  $S_i$  the surface of the convex hull of  $P_i$ . We define  $g_i$ , the contribution of the  $i$ th cell, as:

$$g_i(x_0, y_0, \theta, l_{cell}) = \begin{cases} \frac{S_i}{l_{cell}^2}, & \text{if } P_i \neq \text{empty} \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

The objective function  $f$  is defined as the average of the contributions of the  $N$  cells of the grid:

$$f(x_0, y_0, \theta, l_{cell}) = \frac{1}{N} \times \sum_{i=1}^N g_i(x_0, y_0, \theta, l_{cell}) \quad (4)$$

The 2D grid presenting the best characteristics for the division corresponds to the set of parameters  $(x_0, y_0, \theta, l_{cell})_{\max} \in \mathbb{R}^4$  which maximizes  $f$ .

As  $f$  is not derivable and not even convex, we approximate optimal parameters using a method inspired by simulated annealing. We regularly distribute a set  $S$  of  $N$  initial grids in the 4D space  $(x_0, y_0, \theta, l_{cell})$  with a resolution  $v = (\delta_x, \delta_y, \delta_\theta, \delta_l)$ . Then, we randomly draw small perturbations  $\delta_v$  for each grid of  $S$  in order to generate a total population of  $N_{\text{test}}$  grids (actually we draw  $N_{\text{test}}/N$  perturbations by grid). We then keep grids of highest  $f$  value (actually the grids of  $p$ -percents highest values where  $p$  is a user-defined parameter) and the process is then repeated over  $k$  iterations while reducing the range of the variations  $\delta_v$ : at the iteration  $k$ , those variations are bounded by  $\frac{\|v\|}{2^k}$ . The process stops when  $\frac{\|v\|}{2^k} < \epsilon$ , where  $\epsilon$  is user-defined. The range of the variations reducing over each iteration, the parameter  $\epsilon$  actually controls the number of iteration, thus the computation time. Experimentally, we noticed that an initial population of  $N = 10$  grids, a test population of  $N_{\text{test}} = 1000$  grids and a filtering ratio  $p = 20\%$  give the best results on all our data sets.

Moreover, several geometric constraints reduce the research domain. The parameter  $l_{cell}$  has its values bounded in the user-defined interval  $[l_{\min}, l_{\max}]$ . The origin of the grid  $(x_0, y_0)$  lays in a tore of thickness  $l_{cell} \times \sqrt{2}$  centered around the data-set. Moreover, to guarantee that a grid of origin  $(x_0, y_0)$  cover all the points of the data-set, the angle swept by each vector originating from  $(x_0, y_0)$  and pointing towards each point of the data-set has to be inferior to  $\frac{\pi}{2}$ . In practice, those limitations of the research domain result in reducing the computation time.

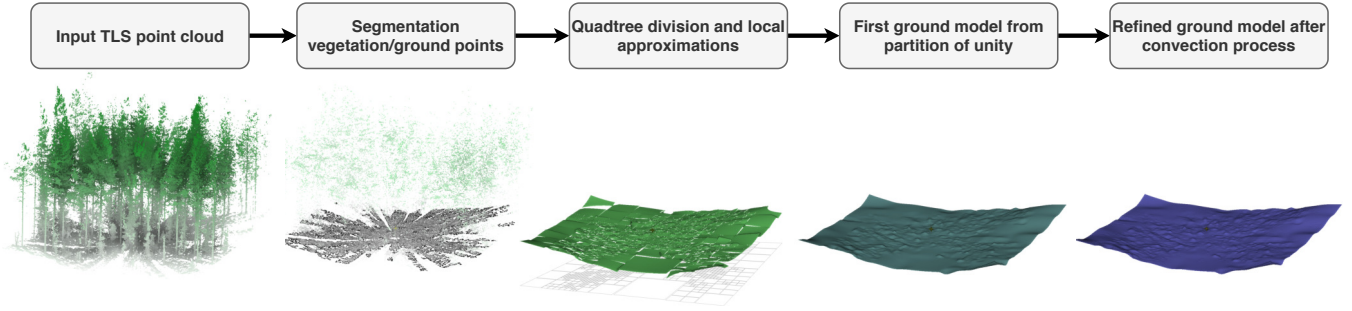
At the end of the maximization process, we keep the grid of best score to guide the 2D division and further adaptive DTM reconstruction in each cell, as presented on the second row of Figure 2.

### 3.3. Estimation of the digital terrain model

In each non empty cell of the optimal grid detailed in the previous section, we run an instance of steps 3 and 4 of the terrain approximation method [MBK20] illustrated in figure 5. In practice, in each cell of the grid, we proceed to a quad-tree division of 3D points. Then, in each quadtree cell  $\mathcal{O}_i$ , we locally approximate the points by a quadric. This local approximation  $g_i$  is done by weighted least square minimization. Finally, we blend the set of all quadric approximations (that, functions attached to each quad-tree cell of each non empty optimal grid cell) to produce a global implicit model  $f$ , computed as:

$$f(\mathbf{x}) = \sum_{\mathbf{c}_i \in C} g_i(\mathbf{x}) \cdot \frac{\Phi_{\sigma'}(\|\mathbf{x} - \mathbf{c}_i\|)}{\sum_{\mathbf{c}_j \in C} \Phi_{\sigma'}(\|\mathbf{x} - \mathbf{c}_j\|)} \quad (5)$$

where  $g_i$  is the local approximation in the cell  $\mathcal{O}_i$  and  $\mathbf{c}_i$  is the center of  $\mathcal{O}_i$ . More precisely we use compactly supported Wendland's RBF functions [Wen05] as a partition of unity to merge the local implicit functions (issued from parametric patches) and compute the global implicit model. The digital terrain model is then extracted by polygonization of the implicit surface using a marching cube like algorithm.



**Figure 5:** Overview of the DTM reconstruction method [MBK20] (figure extracted from the paper). Only steps 3 and 4 are applied here.

#### 4. Experiment and Validation

To estimate the quality of the terrain model produced by our method, we processed several point clouds coming from landscapes presenting a large range of topological characteristics (*i.e.* slope, forest cover, building density). We gathered such data-sets from different sources as described in Section 4.1. Then, Section 4.2 presents our approximation of the terrain models on those plots and compares them to the model obtained with the state-of-the-art method.

##### 4.1. Data

To bring out our method ability to handle diverse topographic features, we considered ten different photogrammetric point-clouds displaying a wide variety of vegetation, hills, cliffs, and buildings. The data-sets "Residence" and "Mangalam" have been collected in south-east India; they present a dense and low tropical forest coverage along with few disseminated buildings. The data-sets "Sance", "Seneca", "Sheffield", "Wietrzna" and "Toledo" come from [ope]; we used the Open Drone Map software to produce 3D geo-referenced point clouds from the batches of open-source aerial photos. "Seneca", "Sheffield" and "Toledo" present a relatively flat terrain with sparse trees and buildings. "Sance" features a castle ruin surrounded by a trench, on a mild terrain slope. "Wietrzna" presents cliffs of a gorge that induce a strong terrain gradient. Finally, the data-sets "Cady", "Courreou" and "Rousson" have been provided by [exa]. They have been collected in Southern France and feature sloppy rocky ground with creek tracks. The plots characteristics are summarized in Table 1.

##### 4.2. Experiments and Discussions

To our knowledge, the approximation of the ground surface from UAV photogrammetric data is still an open issue. [PU17] highlights that the method CSF [ZQW\*16], initially designed airborne LiDAR point clouds, is also efficient to process photogrammetric point-clouds. The validation that we follow compares our results to the one obtained with CSF for each of the ten plots selected for the validation. We used the CSF method implemented in the open-source software CloudCompare [GM16]. Both the minimum quad-tree division of the method [MBK20] and the CSF cloth resolution were set to 1 m. In order to present the results in a clear and readable way, we project the resulting surfaces a geo referenced raster

**Table 1:** Details of the plot features (GPC stands for ground points coverage). • reports the presence of the given feature, •• its significant presence.

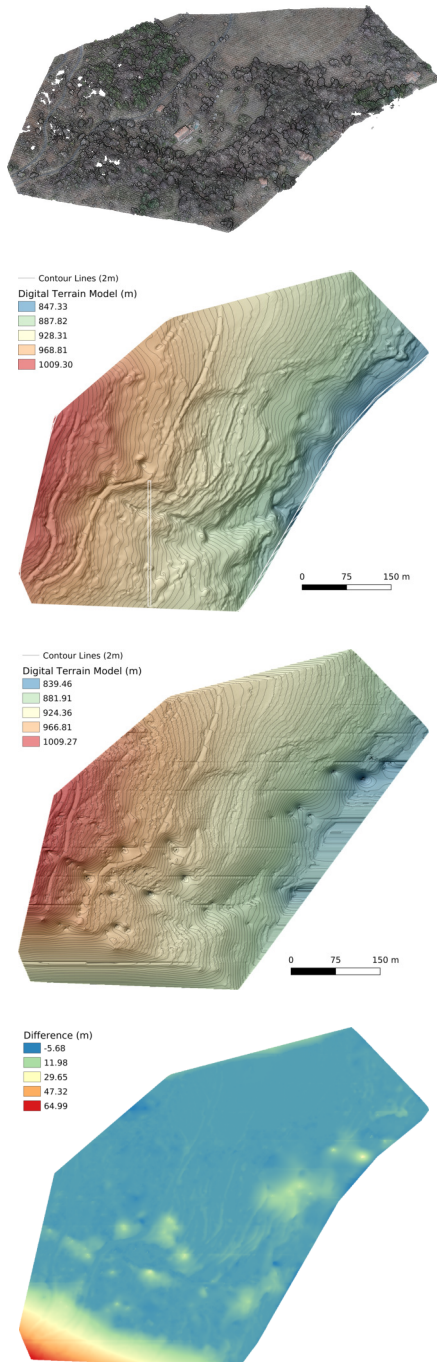
	Slope	Ridge	Tree	Building	Road	Canal	Rock
Sance	•			•		•	
Seneca					•	•	
Cady			•				•
Courreou	••	•	•	•	•		•
Mangalam			••	•			
Rousson	••	••	••				
Residence			••	•	•		
Sheffield			•	••	•		
Wietrzna	•	••					•
Toledo			•	•			

of 50 cm resolution. The rasters are hill shaded, colored according to the altitude and contour lines were added. Tables 2 to 5 present an overview of the point clouds on the first row, our terrain model and the CSF terrain on the second and third row and the difference between both terrain surfaces on the fourth row. For the plots "Courreou", "Sance", "Wietrzna" and "Mangalam", we extracted transects of the initial point clouds, and both reconstructed models produced by our method and CSF. Their comparisons are presented in Figure 6. We also estimated the Hausdorff distance between the models and the ground points of each plot, as seen in Table 6. This last Table also presents the ground points coverage, computed as the ratio of the areas of the convex-hulls defined by the ground points and the initial point clouds.

Here, we discuss the results for each plot individually.

**Courreou** The complexity of this plot lies in its strong slope, its creek bed in the lower half part, and in the road that snakes on the upper side of the hill. Moreover, as it is highlighted in Figure 6, the point cloud presents few photogrammetric errors due to photos mismatches that consist of points lying below the ground surface. In this case, both methods propose a coherent surface fitting the overall slope of the terrain. However, our result describes with more sharpness the complete road layout and both creek mountain-

**Table 2:** Digital Terrain Model for the plot "Courreou". The top row displays a screenshot of the 3D point clouds. The second row presents a shaded elevation map of our digital terrain model. The third row presents a similar shaded elevation model given by the CSF method. And the fourth row display the difference in between both elevation map. The white box represents the transect presented in Figure 6.



sides. It even reveals the terrace cultivation outlines in the center of the plot. The difference between the two elevation model shows that the CSF method fails to filter lower noise points induced by photogrammetric errors; plus it fails to model the extreme lower part of the plot.

**Sance** This point cloud describes a castle ruin surrounded by a water pit. It also presents a patch of missing data in the center corresponding to a few trees. Overall, our method retrieves sharper details of the ruins while the CSF method tends to over smooth the side of the water pit; it appears clearly on the difference image and Figure 6. Both methods propose a different way to fill the missing data patch. Even if it is difficult to draw out the results in such a case, the analysis of the contour lines tends to favor our result which displays smooth and continuous lines in this area.

**Rousson** This challenging data-set displays a hilly terrain covered by trees. Our result presents several bulges corresponding to vegetation that has not been properly filtered out. On the contrary, CSF result is smoother. However, the study of the difference map and the contour lines underscores the tendency of CSF to over-smooth, in the lower part of the plot and on the edges, which tends to increase the distance to the ground points, as seen on the Table 6.

**Wietrznia** This plot characteristic is a deep canyon of a few dozen meters surrounded by cliffs. Our result represents accurately the cliffs wall, and models correctly the abrupt slope discontinuity. On the other hand, the CSF method fails in modeling the cliff properly and over-smooth the topology of the area. This difference appears clearly in Figure 6. It should be noted that on this plot too, some vegetation patches remained after the filtering; they produce few bumps on the upper part of our terrain model.

**Toledo** The point cloud displays a relatively flat landscape with a dense forest cover on the right part. As no ground points could be retrieved under the forest cover, our method ignores that area and proposes a terrain model on the remaining part of the plot. The CSF method over-estimates the elevation of the terrain in the same area.

**Seneca** This plot consists of a relatively flat agricultural landscape cut across by roads with following canals. Our result reproduces faithfully the canals whereas CSF one tends to erode their sides. Moreover, horizontal stripes appear on the CSF result as it fails to reconstruct the border of the plot.

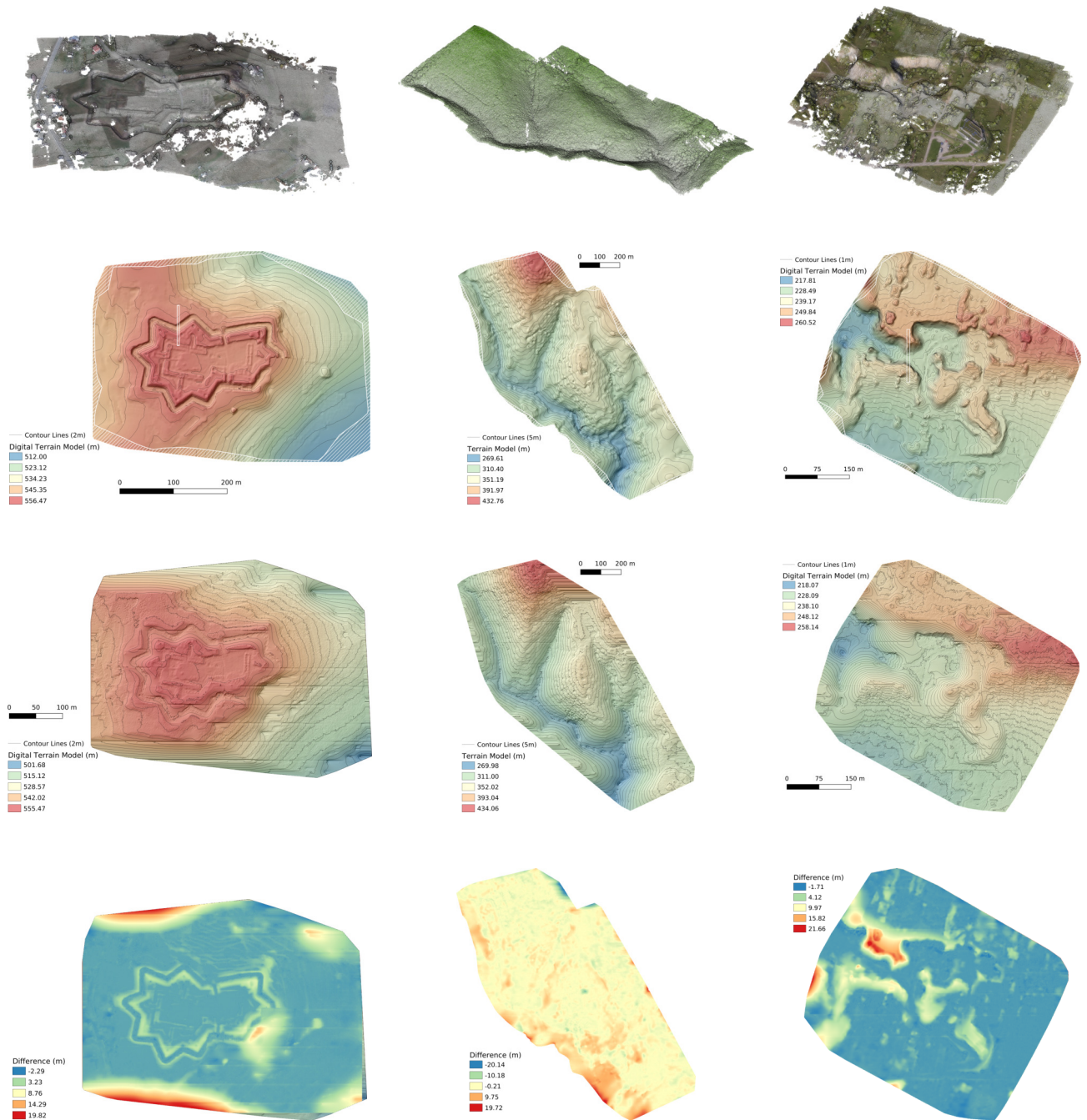
**Residence** Both methods produce a smooth terrain model on this plot covered by forest. In both cases, trees and buildings are correctly filtered out. Nevertheless, CSF does not filter out photogrammetric noise points; as highlighted by the two red patches on the left side of the difference map.

**Mangalam** In the lower right corner, this plot displays a typical south-Indian square water tank surrounded by mounds. It is crossed horizontally by an ancient river bed. Brought out by the analysis of the difference map, our method reproduces sharply those features while CSF over-smooths the river bed and the mounds. Figure 6 highlights the efficient filtering of the vegetation in both case and the over-smooth of the mounds for CSF.

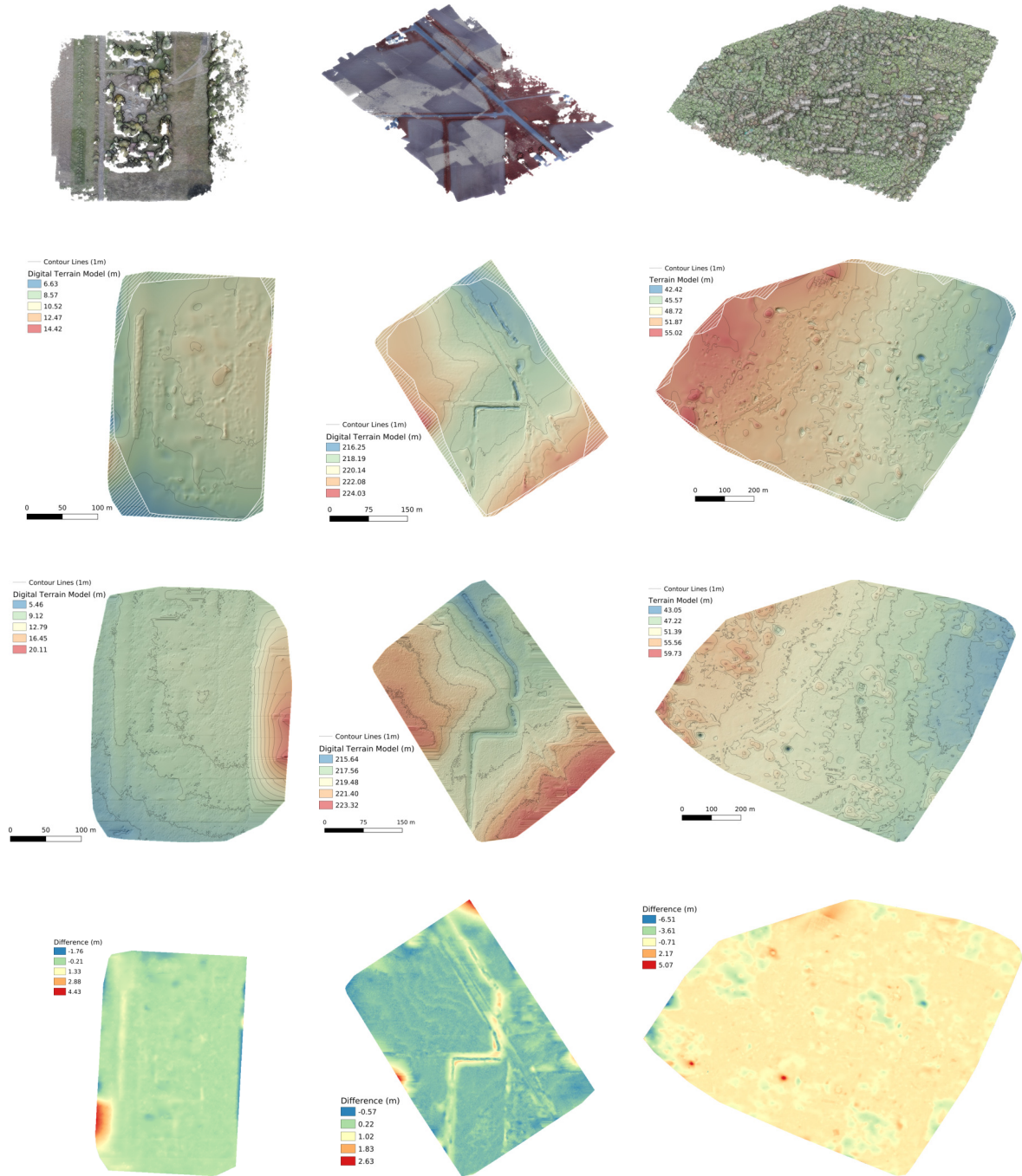
**Cady** The sides of the river bed are over-smoothed by the CSF method, in particular the upper left side of the river. The contour lines of our model highlight the faithful reconstruction of the upper



**Table 3:** Digital Terrain Model for the plots "Toledo" (left column), "Rousson" (middle column) and "Wietrzna" (right column). The top row displays a screenshot of the 3D point clouds. The second row presents a shaded elevation map of our digital terrain model. The third row presents a similar shaded elevation model given by the CSF method. And the fourth row display the difference in between both elevation map. The white boxes represent the transects presented in Figure 6.

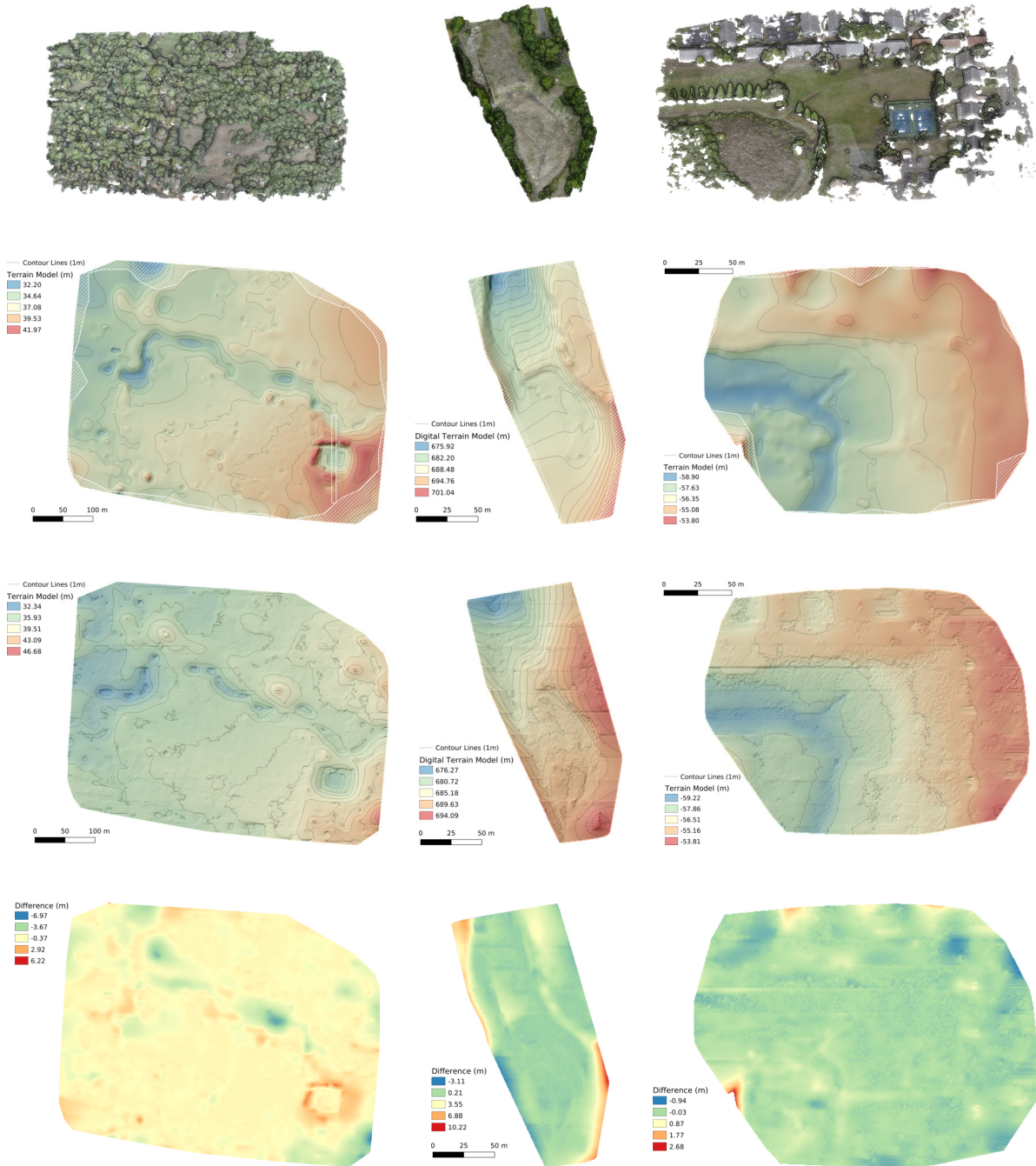


**Table 4:** Digital Terrain Model for the plots "Sance" (left column), "Seneca" (middle column) and "Residence" (right column). The top row displays a screenshot of the 3D point clouds. The second row presents a shaded elevation map of our digital terrain model. The third row presents a similar shaded elevation model given by the CSF method. And the fourth row display the difference in between both elevation map.





**Table 5:** Digital Terrain Model for the plots "Mangalam" (left column), "Cady" (middle column) and "Sheffield" (right column). The top row displays a screenshot of the 3D point clouds. The second row presents a shaded elevation map of our digital terrain model. The third row presents a similar shaded elevation model given by the CSF method. And the fourth row display the difference in between both elevation map. The white box represents the transect presented in Figure 6.

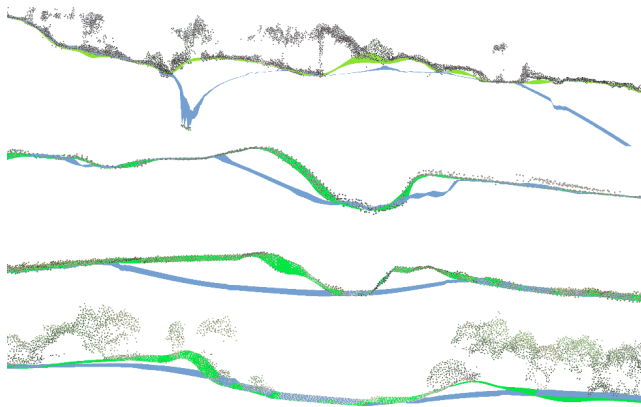


right bank whereas CSF flattens the areas in a continuous slope, which is not coherent with the terrain topology.

**Sheffield** Both methods reconstruct sharply the topology of the simple plot, in particular, the circular pit in the lower-left corner.

**Table 6:** Hausdorff distance from the terrain models to the ground points. We also give the ratio of the overall plot surface covered by the ground points.

	Ground points coverage (%)	Distance (cm) Our method	Distance (cm) CSF
Sance	86.7	$19.85 \pm 17.55$	$88.69 \pm 171.80$
Seneca	82.2	$15.91 \pm 8.01$	$23.37 \pm 20.06$
Cady	74.9	$19.85 \pm 17.07$	$76.50 \pm 105.83$
Courreou	45.5	$24.95 \pm 19.95$	$270.65 \pm 718.56$
Mangalam	45.7	$23.19 \pm 24.65$	$28.41 \pm 46.76$
Rousson	98.8	$64.67 \pm 56.31$	$130.50 \pm 186.41$
Residence	85.4	$20.50 \pm 17.77$	$19.23 \pm 17.70$
Sheffield	41.3	$17.07 \pm 8.40$	$20.12 \pm 11.05$
Wietrzna	77.4	$24.33 \pm 24.77$	$165.05 \pm 307.35$
Toledo	65.1	$16.93 \pm 8.52$	$23.85 \pm 17.43$



**Figure 6:** Transect extraction of the plots "Courreou" on the first row, "Sance" on the second row, "Wietrzna" on the third row and "Mangalam" on the fourth row. Our terrain model appears in green, the CSF model is in blue. The point clouds have been decimated for visualization purpose.

Overall, as illustrated in Figure 6, our method reproduces more precisely the terrain surface. It does not tend to over-smooth the ridges, mounds, and the plot borders, as it is the case with the CSF method. Regardless of the ground points coverage, for every plot except "Rousson", our method produces terrain surface around 20 cm away from the selected ground points, as seen on Table 6. In the case of CSF, the results are satisfactory for relatively flat plots, like "Seneca", "Residence", "Sheffield" or "Toledo", but deteriorates if the terrain presents sudden change of slopes as it is the case for "Sance", "Wietrzna" or "Cady". Brought out by the analysis of the Hausdorff distance, our method also filters out better the noise points under the ground surface. In particular, for the plot "Courreou", the distance of our model to the ground points remains

contained whereas CSF distance presents a mean distance superior to 2 m with a standard deviation over 7 m. The performance gain of our method comes from its ability to filter correctly the noise points under the ground surface and the vegetation points while preserving the topology details. While remaining small on data set relatively flat without noise points like "Residence", "Sheffield" or "Toledo", this gain increases when the points are noisy like in "Courreou" or when they describe a complex topology like in "Wietrzna", "Cady" or "Sance".

Our approach has two main limitations: The first one resides in the filtering of the vegetation; on the plots "Rousson" and "Wietrzna" in particular, several bumps due to unfiltered vegetation points appear on the elevation model. This issue can be coped by using a deep learning approach that has proven to be successful to separate vegetation from ground points [MBK20]. The second limitation is the computation time: while the CSF method computes a terrain model in less than a minute, our approach is 10 times slower, mainly due to the polygonization of the implicit function. Currently, based on a regular marching cube, we plan to improve our polygonizer by following a multi-scale dual contouring approach.

## 5. Conclusion

In this work, we propose an efficient method designed to recover the terrain surface model from 3D photogrammetric point clouds acquired by UAV. Our approach relies on statistical analysis to separate ground points from vegetation points. It handles the irregular point cloud outlines by optimizing a 2D division which later guides the reconstruction by blending implicit quadrics through the partition of unity principle. These contributions enable us to achieve the state of the art performance in terrain reconstruction. In the future, it is worthwhile thinking of how to improve the vegetation filtering by using deep learning, and how to improve the computation time by following a dual contouring approach for the polygonization of the implicit function.

## 6. Acknowledgments

We thank Giulio Enrica and Philippe May at Auroville Centre for Scientific research Trust, India, and Henri Borreill at Exametrics [exa] for providing test data. We also would like to thank the reviewers for their thoughtful comments and efforts towards improving our manuscript and the Japan Society for the Promotion of Science (JSPS) for providing Jules Morel fellowship. The study was supported by Grant JSPS KAKENHI, Grant Number 18F18796, from the Japan Society for the Promotion of Science (JSPS) funds.

## References

- [Axe00] AXELSSON P.: DEM Generation from laser scanner data using adaptive tin models. *International Archives of Photogrammetry and Remote Sensing XXXIII* (2000), 110–117. 2
- [CLF\*19] CAO L., LIU H., FU X., ZHANG Z., SHEN X., RUAN H.: Comparison of uav lidar and digital aerial photogrammetry point clouds for estimating forest structural attributes in subtropical planted forests. *Forests* 10, 2 (2019), 145. 2
- [exa] : Exametrics. [www.exametrics.fr](http://www.exametrics.fr). 5, 10

- [GH15] GONÇALVES J., HENRIQUES R.: Uav photogrammetry for topographic monitoring of coastal areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 104 (2015), 101–111. 2
- [GM16] GIRARDEAU-MONTAUT D.: *CloudCompare*, 2016. 5
- [ILS\*06] ISENBURG M., LIU Y., SHEWCHUK J., SNOEYINK J., THIRION T.: Generating raster dem from mass points via tin streaming. In *GIScience'06 Conference Proceedings* (September 2006), pp. 186–198. 2
- [Ise] ISENBURG M.: <http://lastools.org/>. 2
- [MBK20] MOREL J., BAC A., KANAI T.: High accuracy terrain reconstruction from point clouds using implicit deformable model. In *International Conference on Computational Science* (2020), Springer, pp. 251–265. 1, 2, 4, 5, 10
- [MBV17] MOREL J., BAC A., VÉGA C.: Terrain model reconstruction from terrestrial lidar data using radial basis functions. *IEEE computer graphics and applications* 37, 5 (2017), 72–84. 2
- [MBV18] MOREL J., BAC A., VÉGA C.: Surface reconstruction of incomplete datasets: A novel poisson surface approach based on csrbf. *Computers & Graphics* 74 (2018), 44–55. 2
- [ope] : Open Drone Map. [www.opendronemap.org](http://www.opendronemap.org). 5
- [PU17] POLAT N., UYSAL M.: Dtm generation with uav based photogrammetric point cloud. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* 42 (2017). 2, 5
- [RBN\*11] REMONDINO F., BARAZZETTI L., NEX F., SCAIONI M., SARAZZI D.: Uav photogrammetry for mapping and 3d modeling—current status and future perspectives. *International archives of the photogrammetry, remote sensing and spatial information sciences* 38, 1 (2011), C22. 2
- [SBP\*18] SALACH A., BAKUŁA K., PILARSKA M., OSTROWSKI W., GÓRSKI K., KURCZYŃSKI Z.: Accuracy assessment of point clouds from lidar and dense image matching acquired using the uav platform for dtm creation. *ISPRS International Journal of Geo-Information* 7, 9 (2018), 342. 2
- [TPDF14] TOURNADRE V., PIERROT-DESEILLIGNY M., FAURE P.-H.: Uav photogrammetry to monitor dykes-calibration and comparison to terrestrial lidar. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40, 3 (2014), 143. 2
- [UTP15] UYSAL M., TOPRAK A. S., POLAT N.: Dem generation with uav photogrammetry and accuracy analysis in sahitler hill. *Measurement* 73 (2015), 539–543. 2
- [Wen05] WENDLAND H.: *Scattered Data Approximation*. Cambridge University Press, 2005. 4
- [ZQW\*16] ZHANG W., QI J., WAN P., WANG H., XIE D., WANG X., YAN G.: An easy-to-use airborne lidar data filtering method based on cloth simulation. *Remote Sensing* 8, 6 (2016), 501. 2, 5