BULLDOZER, A FREE OPEN SOURCE SCALABLE SOFTWARE FOR DTM EXTRACTION

D. Lallement[1]*, P. Lassalle[1], Y. Ott[2]

[1] Centre National d'Etudes Spatiales (CNES), 31400, Toulouse, France - (dimitri.lallement, pierre.lassalle)@cnes.fr
[2] Thales Services SAS, 31670 Labège, 31400, France - yannick.ott@thalesgroup.com

KEY WORDS: Digital Terrain Model, Digital Surface Model, cloth simulation, open source, scalability

ABSTRACT:

The paper introduces a software called Bulldozer, designed to extract a Digital Terrain Model (DTM) from a Digital Surface Model (DSM) obtained from various sensors. The software is based on a modified version of the multi-scale Drap Cloth principle to process noisy DSMs of any size, employing a tiling strategy and a stability margin to ensure consistent results. A parameter called max object size is introduced to differentiate objects from the ground during the drap cloth process. Gravity steps and ground distance sampling resolution are adjusted based on the input DSM. No-data and noisy values present in the DSM are detected and converted into no-data values to improve the quality of the Cloth simulation. The paper describes a memory-aware parallel execution strategy using both the multiprocessing and the shared memory Python modules. A benchmark dataset has been created to analyze the results and compare them with alternative approaches and reference datasets. Bulldozer offers an extensive Python API. It is open-source and available on PyPi and GitHub. Additionally, a QGIS plugin has been developed.

1. INTRODUCTION

A Digital Surface Model is a 3D representation of the landscape that can be derived from stereo satellite stereo matching or LiDAR acquisitions. Our tool Bulldozer Lallement et al. (2022) allows the extraction of a Digital Terrain Model (DTM), which models the ground elevation, from the DSM. It extracts the topographic landscape by removing the "above ground" objects such as buildings and high vegetations from the DSM. DTMs are used for many applications such as topographic maps or hydrological models for risk studies, i.e flooding and submersions. The Digital Height Model (DHM) , which is the difference between the DSM and the DTM, is also used for many applications such as urban digital twins reconstructions. In this context, the DHM is used to estimate the height of buildings when modeling cities (Level Of Detail 1 or LOD1) or as an input for semantic classification models (land use maps) in order to distinguish high and low vegetation. As a reminder, the differences between a DSM, a DTM and a DHM are illustrated in Figure 1. This paper presents a scalable software, called Bulldozer, to extract a Digital Terrain Model (DTM) from a Digital Surface Model (DSM) provided by any type of sensor (LiDAR, Photogrammetric with stereo correlation). The proposed algorithm is a modified version of the multi-scale Drap Cloth principle Leotta et al. (2019) adapted to process noisy DSM of arbitrary size thanks to a tiling strategy and a given stability margin that ensures identical results to those obtained without tiling. We introduce an intuitive input parameter, called max object size, that defines what are the objects that have to be distinguished from the ground in order to prevent the drap cloth from falling into them. The values of the gravity steps are also changed and correlated to the ground distance sampling resolution of the input DSM. No-data and noisy values are often present in the input DSM and this is particularly true for stereo ones. These outliers penalize significantly the quality of the Cloth simulation because they can be considered as aberrant ground points with a severe impact on the elevation of the neighborhood. To minimize the impact of such values to the quality of the resulting DTM, we propose an algorithm to detect noisy areas and turn them into no-data values. Then several strategies depending on the size of the no-data areas are applied to fill them with interpolated height information. On the computational aspect, we describe how the whole pipeline is executed in parallel thanks to the multiprocessing Python module and how the shared memory concept is used in order to reduce the memory footprint. To analyze and compare our results with DTM ground truth references, we have created a benchmark dataset containing both stereo DSMs computed with softwares such as CARSMichel et al. (2020) or MicMacRupnik et al. (2017) from Spot6, Pléiades and Pléiades Néo perfect sensor images and LiDAR-HD DSMs provided by IGN. Bulldozer is an open source tool that can be installed via pip from PyPi and the code can be retrieved on github. The code is written in Python and Cython and exposes an exhaustive API to access to all the methods of the pipeline. Finally, a QGIS plugin has been developed and made available.

Figure 1. Differences between a DSM, a DTM and a DHM.

1.1 Structure

This paper is organized as follows. Section 2 presents an overview of the related methods for DTM extraction by distinguishing methods that relies on exogenous data such as land cover map and methods that do not require any information. Section 3 lists the main contribution of this paper which consists of the methods of pre-treatment applied to the DSM before the DTM extraction, the modification of the Cloth simulation algorithms and its adaptation for large scale processing.

* Corresponding author

- In Section 3.1, we introduce different strategies to fill no-data regions in the input DSM depending on their areas and their localisations within the image. We show that those strategies are critical to ensure a high quality of the ground height elevation estimation during the multi-scale Cloth Simulation.
- In Section 3.2, we introduce a new input parameter, denoted max_object_size, which determines the height of the decimated DSM pyramid. This new parameter defines which objects will be considered as above-ground or ground and therefore will pilot the fall of the cloth under them.
- In Section 3.3, we propose a tiling strategy for the multi scale Cloth simulation for a memory-aware parallel execution. Particularly, we define a stable margin in pixels to consider around each DSM tile at each level to ensure identical results to those obtained if the DSM was entirely processed in memory. We introduce the notion of internal scalability which represents the capacity of Bulldozer to process input DSMs of arbitrary size.
- In Section 3.4, we present metrics and height profiles of DTMs computed with Bulldozer over different geographical landscapes with various ground elevation variations (hills, mountains, lowlands). Thoses results demonstrate the ability of Bulldozer to provide DTMs of high quality for various landscapes.
- In Section 4, we discuss about computational and architectural aspects of Bulldozer. Bulldozer is open source and accessible on github and Pypi. It provides richfull interfaces via API and CLI. Bulldozer is also memory efficient and proposes a multi-processing execution of the DTM extraction.

Finally, Section 5 describes the current studies on Bulldozer with the motivation to allows the software to be incremental (what we call external scalability), i.e, capable of extracting DTMs on parts of landscape without knowing the neighborhood and hydro compatible.

2. RELATED METHODS

2.1 DTM extraction using exogenous data

In their publication Champion and Boldo (2006), the authors presented an algorithm that can generate digital terrain models (DTMs) using photogrammetric digital surface models (DSMs) and above-ground masks. Their algorithm employs an Elastic Grid method to interpolate height values of the estimated ground and includes a strategy to reject outliers in the input masks to enhance its robustness. The authors demonstrated the viability of their approach and yielded promising results. However, the algorithm requires an above-ground mask for the area of interest as input, which entails training an AI model capable of predicting above-ground elements. Obtaining such a generic model for various types of landscapes could be a tedious task.

Beumier and Idrissa (2015) proposed a 3-step algorithm for computing the digital terrain model (DTM) from the digital surface model (DSM). Firstly, the DSM is segmented into regions using a gradient threshold. Then, in the second step, regions presenting slow height differences at their borders are retained after filtering. Finally, a hierarchical interpolation is performed in the third step to fill in areas corresponding to high gradients or discarded regions. This approach appears to yield promising results when applied to LIDAR DSMs or correlation DSMs from very high-resolution aerial images. However, the quality of the resulting DTM is heavily dependent on the quality of the DSM. Consequently, this method may have limitations when applied to very noisy correlation DSMs computed from satellite images such as Pléiades.

2.2 DTM extraction without exogenous data

Krauß (2018)proposed a simplified approach for generating digital terrain models (DTMs) from digital surface models (DSMs) over urban areas, without requiring an above-ground mask as input. In a pre-processing step, the algorithm detects probable above-ground areas by identifying steep edges between high objects and the ground. An advantage of this approach is that it can handle incomplete DSMs containing no data values due to occlusions. The algorithm scans the entire DSM in four or eight directions to identify high pixels using two thresholds, UpStep and DownStep. High pixels are removed at the end of the exploration, and the final DTM is obtained using any common interpolation method. The resulting DTMs appear to be promising with this simplified approach. However, the algorithm makes a strong assumption when occlusions are encountered. The last valid height seen is considered as the ground, and the next valid height is on the roof and, therefore, considered high, which may not always be the case. Furthermore, the authors concluded that a method for extracting DTMs from DSMs cannot cover all cases and advised using a different method for processing rural landscapes.

A technique for deriving a digital terrain model (DTM) from a digital surface model (DSM) involves the cloth simulation principle introduced in Zhang et al. (2016). The basic concept involves inverting the DSM and dropping a cloth on it under the influence of gravity. The cloth is represented by a grid, with tension applied between each adjacent node, and the initial position of the cloth is above the inverted DSM. The tension between adjacent nodes is set by the user, and gravity is discretized into steps. At each step, the positions of all nodes in the grid are lowered by a constant offset along the z-axis, and new node positions are compared to the corresponding DSM node positions to check if any node has moved below its corresponding DSM node. Any identified nodes are then assigned to the corresponding DSM node position. The simulation stops when the cloth no longer moves. Figure 2 illustrates this principle.

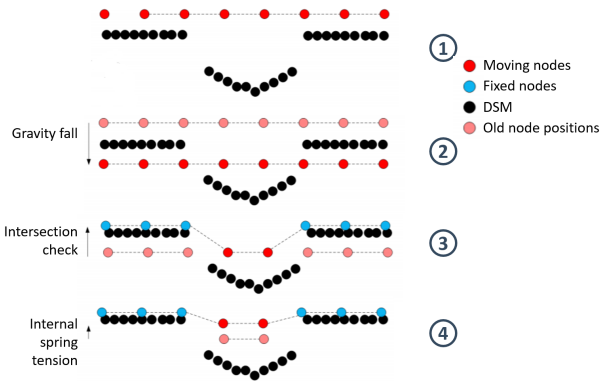

Figure 2. Cloth simulation.

Leotta et al. (2019) presented an optimized version of the cloth simulation method that significantly reduces the processing time. In their approach, the cloth simulation is performed on a decimated digital surface model (DSM) pyramid from top to bottom. At each level, a fixed number of outer iterations (nouter) are executed. Within each outer iteration, the cloth is dropped by one gravity step and tensed ninner times. Finally, an intersection check is performed with the current decimated DSM to reassign the cloth pixels to the corresponding DSM height if they cross the DSM. The resulting cloth is up-sampled to the next level and this process is repeated until the final spatial resolution is reached.

3. METHODOLOGY

3.1 pre-processing DSM strategies to fill holes

Stereo DSMs, generated from multi-view optical satellite acquisition, can contain numerous occlusions. Occluded parts could be considered as objects of minimum height which are hidden by higher neighboring objects. Following this assumption, we propose a recursive pre-processing method to fill those holes in the DSM before extracting the DTM in order to avoid potential structuring nodata areas in the resulting DTM. For each occluded part, valid contour pixels are identified and a height histogram is computed. The first mode of this histogram corresponds to a cluster of pixels representing the valid elements of the landscape at the minimum height. This group of pixels is selected for the interpolation of the pixels in the occluded area. We use the inverse distance weighting strategy Shepard (1968) to perform the interpolation which is applied recursively for each circle of occluded pixels from the outermost circle to the center as illustrated in Figure 3. At each recursive step, the current occluded pixels are interpolated with valid and previous interpolated values. For small nodata areas (surface of a few pixels), which are occurring due to ambiguities in the correlation, a different strategy is applied to fill them. These isolated small nodata regions do not contain useful topographic information for the neighborhood and can be ignored for the Cloth Simulation. So instead of interpolating them before performing the Cloth Simulation, they are assigned to an arbitrarily high value in order to not disturb the DTM extraction. The Drape will stick to the neighboring points and those small no-data regions will be automatically interpolated.

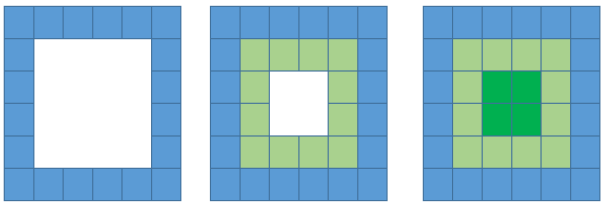

Figure 3. Recursive IDW for filling no-data regions.

Another strategy to fill occluded parts has also been implemented especially for building shadows in urban landscapes. It consists of recovering valid height pixels around the occluded part. Then an elevation histogram is built and we identify with a signal peak detection algorithm the first mod. Physically, this first mod corresponds to the height of the lowest object surrounding our occluded part, which can potentially be the ground. All neighboring pixels which belong to this mod are then selected to compute a 3D plane equation. The pixels of the occluded part are then filled by computing their height thanks to this plane equation.

3.2 Modified DTM extraction step

We propose a modified version of the multi-scale Cloth simulation approachLeotta et al. (2019). In the original algorithm, the height of the decimated DSM pyramid was controlled by a hard-coded threshold, set to 100 pixels, on one of the dimensions (X or Y) of the most decimated DSM image. When this minimum dimension is reached, the construction of the pyramid is stopped. This stopping criterion is applied for input DSM of arbitrary size with any ground distance sampling. The consequence of such criterion is the difficulty for the cloth to stick to the ground when the elevation of the landscape varies a lot. This difficulty is explained by the loss of a lot of information details at the most decimated DSM when the input DSM is large. For large input DSMs, the number of levels is high and quickly the number of outer iterations and the gravity step become small, preventing the cloth from reaching the ground under the hills or the mountains. To solve this issue, we introduce a new parameter, called max_object_size, that controls what are the objects identified as above-ground or ground. Intuitively, this parameter can be seen as the inverse of a spatial cut frequency. Therefore, this parameter determines the number of levels of the decimated DSM pyramid in order to not minimize the previously described errors of the initial algorithm. The closest power of 2 to $\frac{max\_object\_size}{2}$ is determined and this power corresponds to the number of levels minus 1. Thus, the number of levels of the pyramid does not depend on the size and the spatial resolution of the input DSM but on a cut spatial frequency that determines which signal can be considered as ground or not. An illustration of the meaning of max_object_size is shown in Figure 4.

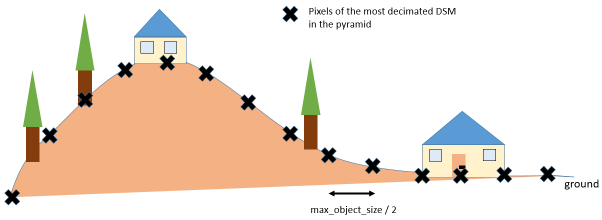

Figure 4. Decimated DSM pyramid controlled by max_object_size.

3.3 Internal scalability

3.3.1 Definition 2 types of scalability are distinguished for an algorithm called internal and external scalabilities. The internal scalabiltiy is the capability of an algorithm to process input data of arbitrary size with a running time that evolves linearily with respect to input size while ensuring stable results. The external scalability is the capability of an algorithm to process incremental input data within a linear time while ensuring stable results.

3.3.2 Scalable Cloth simulation: tiling strategy and stable margin In this study, only the internal scalability of Bulldozer will be described. Its external scalability is currently studied. To satisfy the property of internal scalability, the multi-scale Cloth Simulation has been adapted for using a tiling strategy. The goal is to ensure identical results to those obtained if the whole DSM was processed at once into memory. As a reminderLallement et al. (2022), for each level of the DSM pyramid, a first loop with $nouter$ iterations is executed where at each

iteration a gravity step is applied. A nested second loop with $ninner$ iterations is executed where at each iteration a uniform filter of size 3 x 3 pixels is applied. We propose to process each level in parallel with a synchronization point between each level. For a given level, the DSM is divided into tiles padded with a stability margin to ensure identical results to those obtained without tiling. The expression of this stability margin in pixels is straightforward:

$$margin = nouter * ninner * filter\_half\_size$$

For each level, the tiling strategy is activated when the size of the tile with its additional stable margin is lower than the global size of the decimated DSM. Once the tile is processed, the stable result is extracted by removing the stable margin and all the stable results are then aggregated to provide the resulting DTM for this level. A simple illustration of a tile with a stable margin is shown in Figure 5

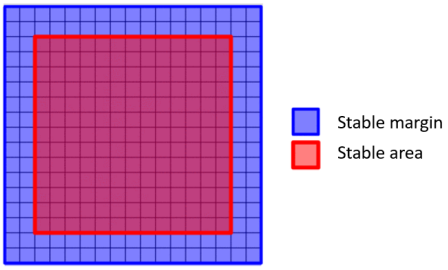

Figure 5. A DSM tile with its additionnal stable margin before the execution of the Cloth simulation at a given level.

3.4 Results

In this section, we analyse the quality of the DTM extracted by Bulldozer and compare it with a reference DTM, called RGE alti, provided by IGN. The IGN RGE alti is a very high resolution DTM provided over French territory. With Bulldozer, DTMs were extracted from DSMs generated from LiDAR or optical sensors, such as Pléaides (with a GSD of 50 cm), Pléiades Néo (30cm) and Lidar HD (15cm). The goal is to evaluate the performance of Bulldozer with different kinds of sensors over regions of interest with various ground elevations (urban, coastal, mountainous, etc.). Since the quality of the input DSM provided as input of Bulldozer has a major impact on the output DTM, we also chose to study two open source DSM extraction softwares: CARS Michel et al. (2020) from CNES and MicMac Rupnik et al. (2017) from IGN. The DSM was computed from stereo satellite images. For the reproducibility of the results, we cannot provide the input DSMs generated from Pleiades and Pleiades Neo sensors due to license issue. However, the Lidar DSM used is provided by LidarHD mission which is in open data[1]. The following metrics have been used for the evaluation: mean, standard deviation (STD), Root Mean Square Error (RMSE), median and Median Absolute Deviation (MAD). The statistics are available in Table 1 and a profile showing the difference between the input DSM, the Bulldozer DTM and the RGE alti DTM for each kind of evaluated sensor is shown in Figure 6.

[1] https://geoservices.ign.fr/lidarhd

Location *Sensor* *DSMtool* *Topograhy*	Mean	STD	RMSE	median	MAD
Montpellier *Lidar* *CARS* *urban*	0.75	44.70	44.72	0.02	0.48
Nice *Pléiades* *CARS* *rural* *mountains*	-1.81	2.35	2.97	-1.85	0.77
Montpellier *Pléiades* *MicMac* *urban*	1.45	1.93	2.41	1.03	0.80
Paris *PléiadesNéo* *CARS* *urban*	-0.47	1.99	2.05	-0.48	0.96

Table 1. Comparison between the DTM produced with Bullodzer and the IGN RGE alti ground truth.

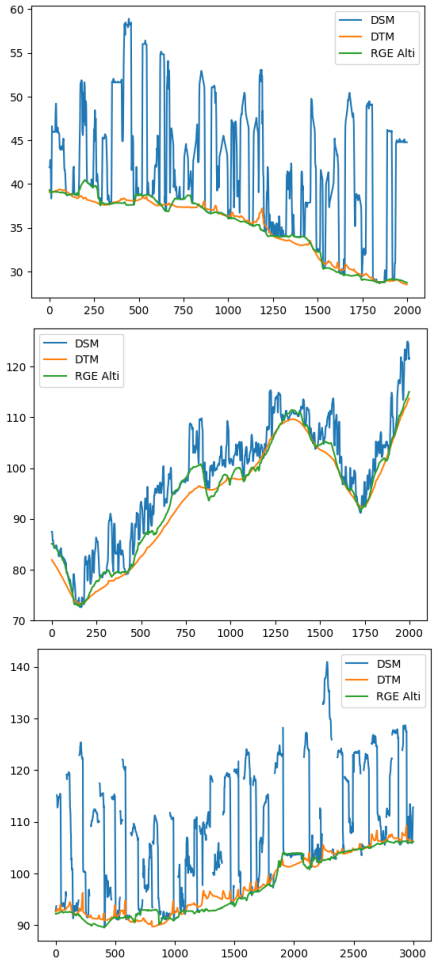

Figure 6. 1D profile for each kind of evaluated sensor. Top: Lidar, Middle: Pléiades, Bottom: Pléaides Néo. Blue: DSM, Green: Ground truth (RGE Alti) and Orange: Bulldozer DTM.

4. BULLDOZER SOFTWARE

4.1 Accessibility of the software

Since our previous version Lallement et al. (2022), Bulldozer has been made accessible for the geoscience and remote sens-

ing community. Bulldozer can handle any input DSM as long as it is in a raster format. It can handle both rasterized airborne DSMs and low resolution satellite DSMs. It's also robust to noisy and incomplete DSMs. In addition to the possibility of taking into account very heterogeneous DSM inputs in terms of resolution and quality, a huge effort has been made on the software to facilitate its use. Bulldozer is distributed under the non contaminating open source license Apache V2.0 in order to allow anyone to use it and integrate into his own pipeline. The complete source code of Bulldozer is published on Github[2]. A Python package of Bulldozer is available on Pypi[3] so that anyone can install it quickly. An exhaustive API has been developed, covering all the main steps of Bulldozer which can be called in standalone (e.g. pits filling) without having to execute the full pipeline.

4.2 Interfaces

In order to reach a maximum number of users we have developed several interfaces for executing Bulldozer. Firstable, for the end users, a QGIS plugin has been published to allow users who do not have a computer science background to use the tool through QGIS GUI. Then a Command Line Interface (CLI) has been designed to run Bulldozer in a terminal. It works with a configuration file or by directly overriding the input parameters as CLI arguments. Finally, a Python API has been set up to allow Bulldozer to be used within an existing tool or through a script. Like the CLI, it can be launched with a configuration file or by overriding the Bulldozer parameters as arguments. The interfaces have been thought to be retrocompatible and adapted for future changes, such as generating level maps (contour each 10m for example). For this feature, two new parameters will be added into the configuration file: one to activate the option and the second one to set the step parameter. But the previous configuration files will still work thanks to a simple automatic mechanism that sets default values for a parameter which is not present as argument of the function. It guarantees that Bulldozer version upgrades will not require updates of existing configuration files. If in the future some parameters become useless, it will not affect Bulldozer's operability.

4.3 Programming Language

The Bulldozer pipeline was developed in Python even if the performances of this language are quite low compared to the majority of languages. This choice facilitates the arrival of new contributors and new users in the geoscience and remote sensing field. However, the core functions which perform the computational parts are developed in Cython in order to profit from the C++ performance in a Python context. It results in a high performance code wrapped into a popular high level language that is simple to integrate or address.

4.4 Multiprocessing and Shared Memory implementation

For the whole pipeline, each step is implemented using a tiling strategy with the right stable margins. Each padded tile is processed independently using the multiprocessing feature ProcessPoolExecutor available in Python concurrent futures module. This feature could cause problems concerning the memory consumption when being applied to big numpy arrays. Indeed, the forking process strategy of Linux operating systems can lead to a duplication of allocated resources from the parent process to the forked child. To avoid these issue and also to reduce I/O operations, a new feature called Shared Memory available since Python 3.8 is used in Bulldozer to share DSM, DTM and mask big arrays for all the forked processes. This allows us to only use the required allocated ressources but it has the drawback to manually handle the ressource memory allocation. To simplify the use of this concept, a Python class, called Shared, has been developped to encapsulate all the memory allocation logic. It is interesting to note that this class can be easily used outside of Bulldozer for your own project dealing with large images.

[2] https://github.com/CNES/bulldozer
[3] https://pypi.org/project/bulldozer-dtm/

5. PERSPECTIVES

5.1 External stability for 3D large-scale Earth Observation missions

Current and incoming LiDAR and spatial Earth Observation missions will provide a massive quantity of 3D data. The spatial mission CO3D Lebègue et al. (2020) will deliver very high resolution DSMs at large scale over emerging landscapes. The IGN LiDAR HD (ref LIDAR HD)mission is currently delivering high density point clouds of French national territory. Such missions are challenging because of the massive amount of data to process. In order to address this task, the French spatial agency (CNES) has developed CARS Michel et al. (2020), a tool for generating large-scale DSMs, and Bulldozer to produce DTMs from the CARS DSMs. In addition to the constraints of image size and computation time, these missions bring another challenge: updating part of existing DEMs (DSM or DTM). Indeed, these large-scale missions do not directly produce the global DSMs but tiles that, when combined, cover large areas. The challenge consists in updating parts of regions that have been already generated and in particular preserving consistency and continuity on the borders and overlap areas of the new tiles. Bulldozer must have internal stability: regardless of the configuration of the system (CPU, RAM, etc.) on which the tool is launched the result must be the same. Moreover, it requires external stability: for a given area, for all ROI, the result must be the same within a stable area padded by a stability margin which will be determined in further studies. We have launched a series of research and development projects to address this issue and we're currently working actively on this subject. [illustration?]We also want to work on removing some remaining magic numbers to replace them with values related to input physical information (image dimension and resolution, max object size, etc.). However, this is a sensitive subject because we want to ensure both convergence and stability. A solution would be to keep maximum convergence values (hardcoded values) while limiting the number of iterations depending on the convergence of the Cloth Simulation. For example, keep the n_outer at a maximum value of 100 but with the possibility to stop before when the convergence of the drape is reached.

5.2 Coastal Digital Twins and hydro compatibility

We also want to improve the quality of DTM extracted from landscape covering coastal areas. We noticed a quality drop in the shoreline DTM areas resulting from the noisy areas in water (due to the impossibility to correlate in those regions) and we would like to address this issue. Contrary to a river where the sheet can stick to the other side and then apply a tension which will restore the edges, on sea surface, the drape does not have other points to stick to and it ends up in precision loss on these

areas. Further studies will also be explored to make Bulldozer DTMs hydro compatible. Different projects are focusing on the simulation of flooding Kettig et al. (2021), submersion or fluid flow in general and need realistic DTMs as input to their model.

References

Beumier, C., Idrissa, M., 2015. Deriving a DTM from a DSM by uniform regions and context. *EARSeL eProceedings*, 14(1), 16.

Champion, N., Boldo, D., 2006. A robust algorithm for estimating digital terrain model from digital surface models in dense urban areas. *In ISPRS Commission 3 Symposium on Photogrammetric Computer Vision*, Citeseer.

Kettig, P., Baillarin, S., Blanchet, G., Taillan, C., Ricci, S., Nguyen, T.-H., Huang, T., Altinok, A., Chung, N. T., Valladeau, G., Goeury, R., Roumagnac, A., 2021. The scoflooddam project: New observing strategies for flood detection, alert and rapid mapping. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 1464–1467.

Krauß, T., 2018. A New Simplified DSM-to-DTM Algorithm – dsm-to-dtm-step. *Preprints.org 2018*. https://doi.org/10.20944/preprints201807.0017.v1.

Lallement, D., Lassalle, P., Ott, Y., Demortier, R., Delvit, J.-M., 2022. BULLDOZER: An automatic self-driven large scale Digital Terrain Model extraction method from Digital Surface Model. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-409-2022.

Lebègue, L., Cazala-Hourcade, E., Languille, F., Artigues, S., Melet, O., 2020. CO3D, a worldwide one-meter accuracy DEM for 2025. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* http://dx.doi.org/10.5194/isprs-archives-XLIII-B1-2020-299-2020.

Leotta, M. J., Long, C., Jacquet, B., Zins, M., Lipsa, D., Shan, J., Xu, B., Li, Z., Zhang, X., Chang, S.-F., Purri, M., Xue, J., Dana, K., 2019. Urban semantic 3d reconstruction from multiview satellite imagery. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1451–1460.

Michel, J., Sarrazin, E., Youssefi, D., Cournet, M., Buffe, F., Delvit, J.-M., Emilien, A., Bosman, J., Melet, O., L'Helguen, C., 2020. A new satellite imagery stereo pipeline designed for scalability, robustness and performance. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* https://doi.org/10.5194/isprs-annals-V-2-2020-171-2020.

Rupnik, E., Daakir, M., Pierrot Deseilligny, M., 2017. MicMac – a free, open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*. https://doi.org/10.1186/s40965-017-0027-2.

Shepard, D., 1968. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. *Proceedings of the 1968 ACM National Conference*. http://dx.doi.org/10.1145/800186.810616.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*, 8(6). https://www.mdpi.com/2072-4292/8/6/501.