

BULLDOZER: AN AUTOMATIC SELF-DRIVEN LARGE SCALE DTM EXTRACTION METHOD FROM DIGITAL SURFACE MODEL

D. Lallement¹, P. Lassalle¹, Y. Ott², R. Demortier² and J. Delvit¹

¹ Campus de la Donnée, Centre National d'Etudes Spatiales (CNES), 31400, Toulouse, France

² Thales Services SAS, 31670 Labège, 31400, France

Commission II, WG II/4

KEY WORDS: Digital Surface Model, Digital Terrain Model, cloth simulation, scalability, self-driven

ABSTRACT:

This study presents a Digital Terrain Model extraction method called Bulldozer. The only required input of Bulldozer is a Digital Surface Model generated from any sensors (usually optical or LIDAR) with any kind of software. After reviewing both the initial DrapCloth algorithm (Zhang et al., 2016) and its multi scale implementation (Leotta et al., 2019), some issues have been highlighted when extracting DTM from stereo satellite images such as the lost of ground adhesion under rising terrain areas, the appearance of sinks due to correlation issues when computing the DSM and finally the lack of scalability when processing large input data. Bulldozer has been developed to tackle all these issues and proposes a full automatic scalable pipeline composed of a pre-processing step to clean noisy DSMs by detecting and smoothing disturbed areas, a DTM extraction step based on a modified DrapCloth algorithm to stick to the ground under rising terrain and a post-processing step to smooth sharp sinks. The scalability has been solved using a tiling strategy and the definition of a stability margin that ensures identical results to those obtained if the whole DSM would have been processed at once in memory. As a result, Bulldozer outperforms its concurrent with respect to runtime execution while providing high quality DTMs over various types of landscapes.

1. INTRODUCTION

1.1 Definition

As shown in Figure 1, from at least one stereo pair of satellite or aerial images, it's possible to reconstruct a 3D scene. Tools such as CARS (J. Michel and L'Helguen, 2020), MicMac (Ewelina Rupnik, 2017), or ASP (Shean, 2017) can be used to generate a Digital Surface Model (DSM) modeling the surface of the landscape as seen by the satellite from images acquired with different viewing angles. The Digital Terrain Model represents an estimation of the ground as illustrated in Figure 2. It is often computed from DSM with algorithms described in Section 2. Two main types of algorithms are used to extract a DTM from a DSM: the first one gathers methods that rely on slopes to classify above ground and ground areas. The second type relies on an above-ground mask input to discriminate the ground and the objects. For both types, DTMs are then generally computed by interpolation between detected ground areas.

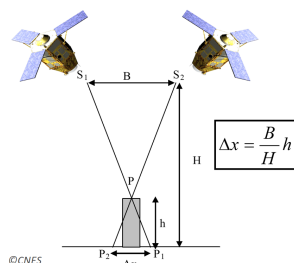


Figure 1. Stereo DSM computation.

1.2 Application

DTMs are used in many application domains: topography, hydrography, bathymetry, land cover map, 3D urban reconstruc-



Figure 2. Difference between DSM, DTM and DHM.

tion (LOD), military needs, etc. It can also be used to improve orthorectification of very high resolution satellite images (e.g. Pleiades Neo). DTMs are also often used as auxiliary data for specific remote sensing applications.

1.3 Needs

A massive influx of 3D data seems to be emerging with new Earth Observation spatial missions such as CO3D (production at large scale of very high resolution DSMs over emerged landscapes (L. Lebègue, 2020)), but also with the increase of airborne LIDAR DSMs¹. This trend motivates the French spatial agency (CNES) to focus on the development of tools to process 3D data at large scale. In this context, a DTM extraction software called Bulldozer has been designed to process DSM rasters at large scale without any exogenous data while being robust to noisy and no-data values. Another motivation was to handle any kind of sensor and all kind of topography (coastal, mountain, etc.).

1.4 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

¹ For example, French territory covering through the IGN LIDAR HD mission.

map and methods that do not require any information. Section 3 lists the main contribution of this paper which consists of the adaptation of the multi-scale Cloth Simulation algorithm to process noisy photogrammetric DSMs at large scale in order to extract DTMs.

- In Section 3.2, we introduce pre-processing steps to handle no-data points, noise and correlation errors in the input DSM that have a critical impact on the quality of the ground height estimation. Particularly, we describe a new processing step to detect disturbed areas that will be removed for the following algorithm steps.
- In Section 3.3.1, we define a new input parameter, called *max_object_size*, to control the ground adhesion of the cloth in order to adapt the algorithm to landscapes with various topographies.
- In Section 3.3.2, we add a new step before the DTM extraction to ensure the cloth to stick to the ground.
- In Section 3.3.3, we propose a scalable multi-scale cloth simulation by defining a stability margin that allows the use of a tiling strategy to process DSMs of arbitrary size. This new scalable framework aims at ensuring an identical DTM to the one obtained if the whole DSM had been processed without tiling.
- In Section 3.4, we add a post-processing step to detect and fill residual sinks that could remain in the DTM due to missing detections of disturbed areas.
- In Section 3.4.2, we propose the user to control the target resolution and projection of the DTM.
- In Section 4.1, we demonstrate the capacity of Bulldozer to manage heterogeneous sensors DSM.

Section 3.4.3 describes the time and memory complexity of our proposed algorithm. Finally, Section 4 presents some results that exhibit the correctness of our method, its multi-sensor ability and its runtime efficiency with respect to the size of the input DSM.

2. RELATED METHODS

2.1 DTM extraction using exogenous data

In (Champion and Boldo, 2006), the authors proposed an algorithm to derive DTMs from photogrammetric DSMs and above-ground masks. The principle of the algorithm is to use an Elastic Grid method to interpolate height values of the estimated ground. To be robust to outliers contained in input masks, the authors added a strategy to reject those them. They demonstrated the feasibility of their approach and showed promising results. However, their method always needs an input above-ground mask on the area of interest. Therefore, this requires also to train an AI model for predicting above-ground elements. This AI model must also be generic for various type of landscapes, which can be fastidious to obtain.

In (Beumier and Idrissa, 2015), the author proposed a 3-step algorithm to compute the DTM from the DSM. The first step consists of a connected component segmentation of the DSM into regions based on a gradient threshold. In the second step, resulting regions are filtered by keeping only the ones that present

slow height differences at their borders. Finally, in the third step a hierarchical interpolation is done to fill the areas corresponding to high gradients or discarded regions. This method seems to show promising results when applied to LIDAR DSMs or correlation DSMs from very high resolution aerial images. However, the quality of the DTM is strongly dependent on the quality of the DSM. This method can show some limits for very noisy correlation DSMs computed from satellite images such as Pléiades.

2.2 DTM extraction without exogenous data

In (Krauß, 2018), the authors proposed a simplified approach for computing a DTM from a DSM over urban areas without using an above-ground mask as input. In a pre-processing step, they detect probable above ground areas by identifying steep edges between high objects and the ground. One nice advantage of their approach is that it can handle incomplete DSMs that contain no data values coming from occlusions. The algorithm consists of scanning the entire DSM with respect along 4 or 8 directions to identify high pixels by using two thresholds *UpStep* and *DownStep*. At the end of the exploration, all pixels tagged as high are removed and the final DTM is obtained by using any common interpolation-method. With this simplified approach, the resulting DTMs seem to be promising. However a strong assumption is made when occlusions are encountered. The last valid height seen is considered as a ground and the next valid height is on the roof and therefore considered as high, which is not always the case. In addition, the authors concluded that a method for extracting the DTMs from DSMs can not cover all the cases and they advise to use a different method for processing rural landscapes.

One possibility to derive a DTM from a DSM relies on the cloth simulation principle firstly proposed in (Zhang et al., 2016). The idea is to turn upside down a DSM and drop a cloth on it by gravity. The cloth is represented by a grid where a tension is applied between each adjacent node. The initial position of the cloth is above the reversed DSM and the tension between each adjacent node is defined by the user. The gravity is discretized into steps. At each step, the positions of all nodes of the grid are decreased by a constant offset along the z-axis. All new grid node positions are compared to the corresponding DSM node positions to check if one of them is below it. These identified nodes are then reassigned to the corresponding DSM node position. The algorithm stops when the cloth does not move anymore. Figure 3 is an overview of the cloth simulation algorithm.

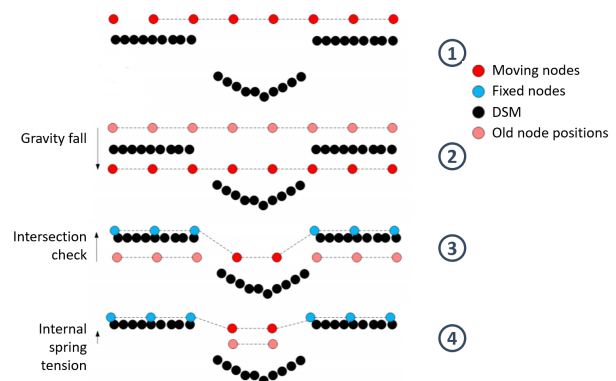


Figure 3. Cloth simulation

In (Leotta et al., 2019), the authors proposed an optimized implementation of the cloth simulation by reducing significantly the running time. The cloth simulation is executed on a decimated DSM pyramid from top to bottom. At each level, n_{outer} iterations are applied. For each iteration, the cloth is dropped by a gravity step and tensed n_{inner} times. A final intersection check is done with the current decimated DSM to reassign pixels of the cloth to the DSM height if they cross the DSM. The resulting cloth is up-sampled to next level and this until the final spatial resolution is reached. After each level, n_{outer} and the gravity $step$ are reduced as follows:

$$n_{outer} = \frac{n_{outer}}{2 * 2^{maxlevel-level}} \quad (1)$$

$$step = \frac{step}{2 * 2^{maxlevel-level}} \quad (2)$$

3. METHODOLOGY

3.1 Workflow overview of the proposed method

Bulldozer is a software that has been built around a pipeline of interchangeable modules. Each module is composed of functions with a simple API and very few dependencies. This architecture allows to call these functions in standalone mode outside the DTM extraction process (denoising, etc.) but also to replace them by other functions doing the same task but with a different strategy. As shown in Figure 4, the pipeline consists of three main modules: *DSM_preprocess*, *DTM_extraction* and *DTM_postprocess*. These modules are composed of sub-functions generating the intermediate products that are used to produce a DTM in the end.

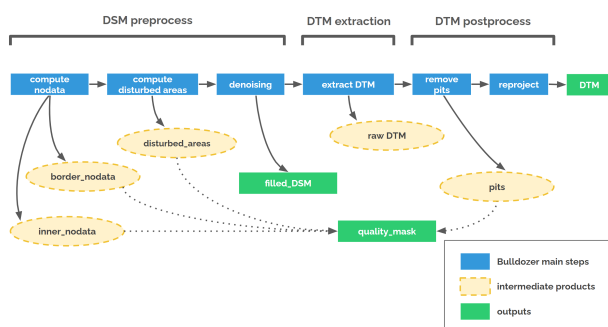


Figure 4. Bulldozer simplified pipeline.

3.2 Pre-processing step

The preprocess pipeline of Bulldozer is described in Figure 5. It aims at preparing the input DSM in order to facilitate the extraction of DTM. The first step of the pre-processing dissociates the landscape boundary of the DSM from no-data values. The second step consists of detecting and removing noisy areas that impact the ground estimation during the cloth simulation. There is also an additional optional step that fills the noisy and no-data areas of the input DSM in order to provide a filled DSM to the user.

3.2.1 No-data processing Since Bulldozer does not take exogenous data as input, no-data values must be classified into several categories: border-no-data, inner no-data (occlusions) and disturbed no-data. The *compute_nodata* module is used

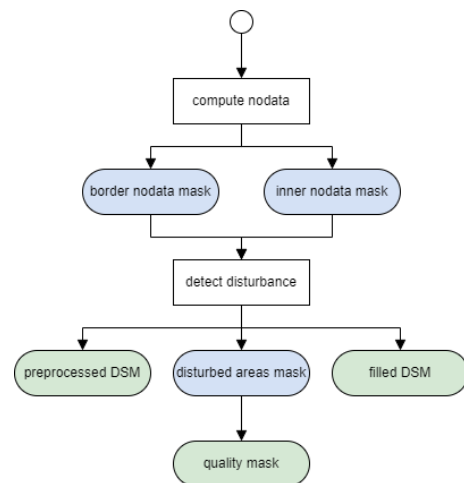


Figure 5. Bulldozer pre-processing simplified pipeline chart (Blue: intermediate products, Green: outputs).

to differentiate the no-data areas that correspond to the edge of the image (*border_nodata*) from those that are the result of uncertainties in the correlator during the DSM computation (*inner_nodata*). To detect the *border_nodata* pixels, we cannot simply detect the no-data pixels that are connected to the edge of the image. Indeed, as illustrated in Figure 6 if we apply this assumption, then the river would be considered as *border_nodata* while it's no-data pixels that come from correlation errors due to water. Therefore, the *border_nodata* detection method will try to delimit the border between *inner_nodata* and *border_nodata* as shown in figure 6. The function will first detect *border_nodata* by doing an analysis along the two axis of the image (vertical and horizontal) and once these pixels have been classified, all remaining no-data pixels are considered as *inner_nodata*. The *border_nodata* are then assigned to an extreme value (9000m) in order to not disturb the the cloth simulation. The *inner_nodata* remain assigned to no-data value.

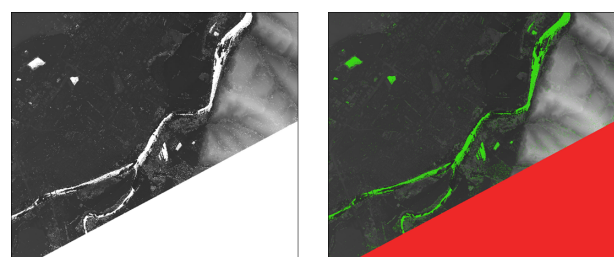


Figure 6. Left: Input DSM. Right: detected no-data areas (Red: border no-data, Green: inner no-data).

3.2.2 Detection of disturbed areas In photogrammetric satellite DSMs, huge anormal variation of elevation can be observed in some areas, denotaed as disturbed areas in this paper. Figure 8 illustrates this phenomenon. This issue is mainly due to three factors that occur during the computation of the DSM. The first is occlusion. As shown in Figure 7, depending on the angle at which the scene is viewed during a stereo acquisition, part of the scene may not be seen. As a result, during the correlation step in the DSM calculation, these areas generate altimetric aberrations. These disturbed areas are mainly found in narrow spaces: streets, etc. A second source of disturbed areas are the uniform and poorly textured areas in the images used to compute a DSM: shadow area, snow, etc. Once again,

during the correlation step when generating the DSM, the correlator will not necessarily succeed in finding the homologous points between the two images because the pixels are too similar (lack of texture), resulting in irregular disparity maps. This again results in altimetric aberrations. The last source of disturbed areas in the DSMs comes from highly mobile areas in the scene. If between the two acquisitions used to make the DSM, some elements have moved, the correlator will not be able to match them. That why, water areas are very problematic because even if they can be textured (foam), the difference in appearance due to the time that elapses between the two acquisitions prevents correlation and leads to errors in altimetry estimations. All these situations lead to a degradation of the DSM quality and generate noise-like areas. In our method, these noisy areas have to be handled before the DTM extraction in order to prevent the sheet from getting stuck on them and leading to dropouts in the DSM as shown in Figure 9. In the pre-processing step there is a routine to detect and remove them by replacing those areas with nodata. Those no-data are not interpolated because there is a risk of filling the street at the height of the top of the buildings. The detection is done by a thresholding method: for each pixel of the raster, a check of the altimetric difference is made with respect to its neighbors as shown in Figure 9. If this difference is higher than a threshold, the pixel is considered as a disturbed area and is replaced by nodata.

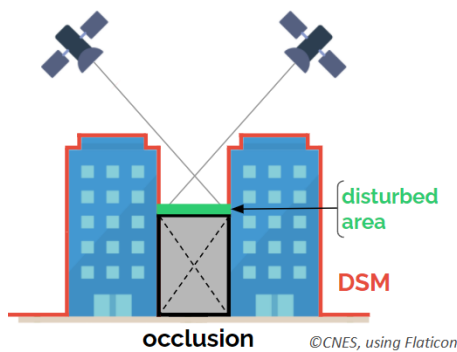


Figure 7. Occlusion illustration.

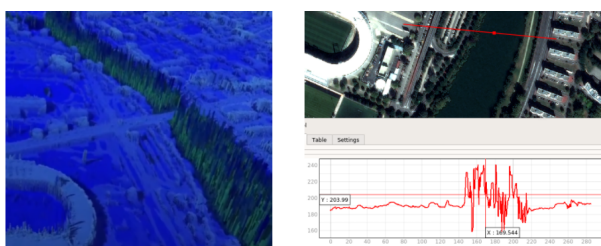


Figure 8. Left: 3D disturbed areas. Right: disturbed area profile.

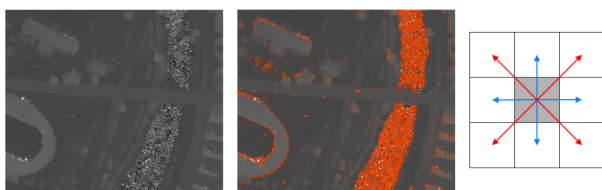


Figure 9. Left: Input noisy DSM. Center: detected disturbed areas. Right: Detection pattern.

3.2.3 Outputs The pre-processing step produces a DSM ready to use for the DTM extraction step (filled on the edges

and de-noised DSM). A quality mask containing the different types of no-data (*inner_nodata* and *border_nodata*) and the disturbance areas is also generated. This mask informs about low confidence areas in the final DTM, the ones that have been interpolated during the extraction. The user can also generate a filled and denoised DSM: the *inner_nodata* and disturbed areas are interpolated but the *border_nodata* remain assigned to no-data value.

3.3 DTM extraction step

3.3.1 Control of the size of the decimated DSM pyramid

To enforce the DTM to stick on areas with relief, a new input parameter is proposed: *max_object_size*. With this parameter, all elements in the landscape covered by the DSM that have a spatial frequency lower than $\frac{1}{max_object_size}$ will be considered as the ground and the DTM will stick to these elements. In the new algorithm, this parameter is used to compute the height of the multi-scale pyramid of DSMs. Instead of considering a minimal size of the dezoomed DSM in the initial algorithm (Leotta et al., 2019), the maximum dezoom factor is determined by computing the nearest power of two of $\frac{max_object_size}{2}$ by analogy with the Shannon sampling theorem. With this modification, the cloth simulation output does not depend on the size of the input DSM but rather on the user ground spatial frequency (show some figures with different value of *max_object_size*).

3.3.2 Prevent the lost of ground adhesion

In (Leotta et al., 2019), the most decimated DTM is initialized at the minimum valid height of the input DSM. This strategy leads to a difficulty of the DTM to stick to the ground where the landscape presents various elevation profile such as hills or mountains. The gravity step is too small to reach the ground in this case. To solve this issue, we add a new step before the cloth simulation. The DTM is initialized to the most decimated DSM in the highest level of the pyramid. Then, a tension with a force *spring_force* is applied on this DTM n_{lost} times. The resulting DTM is then used as input of the cloth simulation step. Thanks to *max_object_size* and this strategy, we ensure that the DTM will never unhook from hills or mountains. Figure 10 shows 1D profiles on the area of Pech David in Toulouse and illustrates the influence of the parameter *max_object_size* on the cloth to stick to the ground. A 1D profile on the same area computed with the original cloth simulation (Leotta et al., 2019) is shown to be compared to our approach. For all the graphs in Figure 10, the red line represents the DSM 1D profile. At the top of the figure, the brown line represents the 1D profile obtained with the cloth simulation algorithm described in (Leotta et al., 2019). In the second graph from the top, the blue line represents the 1D profile obtained with our approach with *max_object_size* = 64 meters, the cyan line in the third graph is the 1D profile with *max_object_size* = 32 meters and *max_object_size* = 16 meters for the last 1D profile at the bottom.

3.3.3 Stability margin and scalability

In order to produce DTMs at large scale and keep a linear evolution of the running time with respect to the size of the input DSM, a parallel tiling strategy execution is proposed. However, in order to keep identical results to those obtained without using a tiling strategy, a stability margin representing a crown of pixels around each tile is proposed at each level l of the pyramid:

$$margin(l) = n_{outer}(l) * n_{inner} * spring_force \quad (3)$$

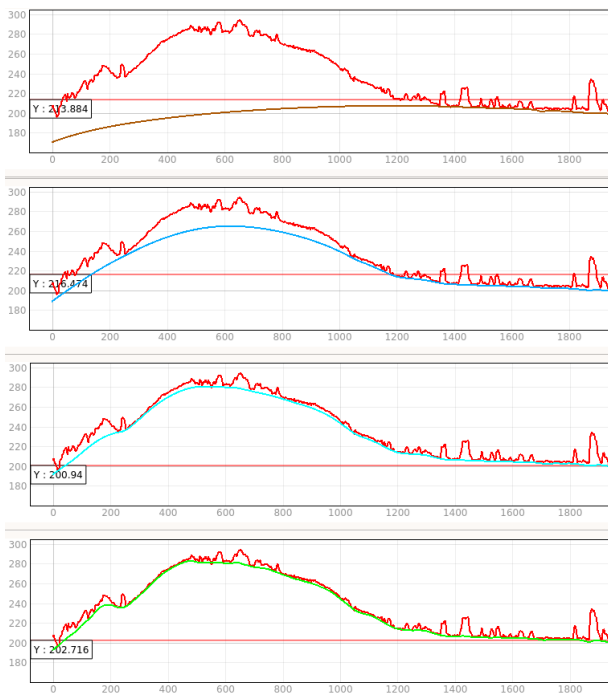


Figure 10. From Top to Bottom: Initial Cloth Simulation - Proposed Cloth Simulation with $max_object_size = 64$ - Proposed Cloth Simulation with $max_object_size = 32$ - Proposed Cloth Simulation with $max_object_size = 16$

Therefore at a given level l , the tiling strategy is activated if $2 * margin(l) < size(tile)$. Indeed, it is not necessary to activate a tiling strategy for the first levels where the size of dezoomed DTMs are relatively small. The modified cloth simulation is described in algorithm 1.

3.4 Post-processing

Figure 11 shows the post-processing pipeline proposed in Bulldozer. It consists of a mandatory step to improve the quality of the DTM by detecting and removing residual pits. Two additional steps are added: the first one allows to generate a Digital Height Model (DHM) that represent the *above ground* as presented in the Figure 2. The second one is designed to reproject the outputs into a new Coordinates Reference System (CRS).

3.4.1 Pits processing Some input DSM outliers may not be detected during the pre-processing step. During the cloth simulation, the cloth get stucked on those points which lead to visible pits. The simulated cloth gets hooked on these points during the DTM extraction and this issue produces pits. In order to detect them, a low frequency version of the DTM is computed by applying a uniform filter on the raw DTM from the DTM extraction step. Pits are the areas for which the low frequency DTM and the raw DTM are not equal. To fill these pits, the value of the low frequency DTM is used. All the interpolated pits areas are stored in the quality mask.

3.4.2 Outputs Upon exiting the post-processing step, Bulldozer produces at least a cleaned DTM and a quality mask updated with the pit areas that have been filled in. The user can also generate a DHM (Figure 2) in order to extract the *above ground* (buildings, vegetation, etc.). All Bulldozer results can be resampled to a target resolution and reprojected into a new Coordinate Reference System (CRS).

Algorithm 1 Proposed cloth simulation algorithm

```

Input  $DSM \leftarrow$  Input Data
Input  $max\_object\_size \leftarrow$  DTM spatial frequency control
Input  $n_{lost} \leftarrow$  Number of iterations to prevent a lost of ground adhesion
Input  $n_{outer} \leftarrow$  Initial number of outer iterations
Input  $n_{inner} \leftarrow$  Number of inner iterations
Input  $spring\_force \leftarrow$  Size of tension applied to the cloth
Output DTM

1: procedure CLOTHSIMULATION()
2:    $max\_object\_size\_pix \leftarrow \frac{max\_object\_size}{dsm\_resolution}$ 
3:    $nb\_levels \leftarrow get\_nb\_levels(\frac{max\_object\_size\_pix}{2})$ 
4:    $dsms \leftarrow build\_pyramid(nb\_levels)$ 
5:    $dtm \leftarrow dsm\_pyramid[nb\_levels - 1]$ 
6:    $dtm \leftarrow prevent\_lost\_adhesion(dtm, n_{lost})$ 
7:    $step = \frac{maxZ - minZ}{n_{outer}}$ 
8:    $curr\_n_{outer} = n_{outer}$ 
9:    $level = nb\_levels - 1$ 
10:  while  $level > -1$  do
11:    if  $< nb\_levels - 1$  then
12:       $dtm \leftarrow upsample(dtm, dsms[level])$ 
13:    end if
14:     $margin = curr\_n_{outer} * n_{inner} * spring\_force$ 
15:    if useTile then
16:       $dtm \leftarrow TiledClothSimu(margin)$ 
17:    else
18:       $dtm \leftarrow ClothSimu()$ 
19:    end if
20:     $level - = 1$ 
21:     $step = \frac{step}{2^{nb\_levels - level}}$ 
22:     $curr\_n_{outer} = \max(1, \frac{n_{outer}}{2^{nb\_levels - level}})$ 
23:  end while
24:  return  $dtm$ 
25: end procedure

```

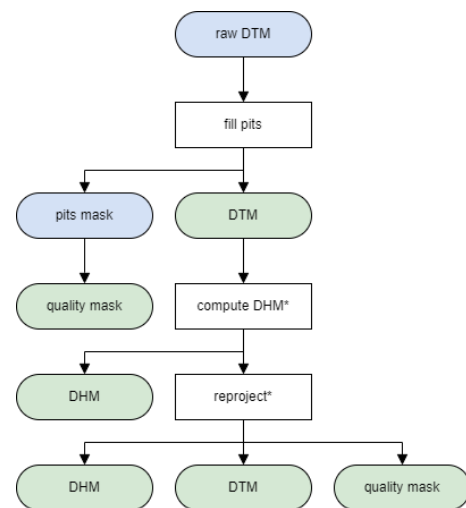


Figure 11. Bulldozer post-processing simplified pipeline chart (Blue: intermediate products, Green: outputs, *: optional step).

3.4.3 Computation aspect The whole Bulldozer pipeline was designed to be fully scalable. To do this, all the processing steps are done by tiles. This approach allows the software to be executed in a distributed and parallel environment. Bulldozer relies on the use of multi-processing in Python and the implementation of core functions in Cython in order to minimize the runtime and memory usage. Table 1 illustrates that Bulldozer can handle DSMs of arbitrary sizes. The memory usage is linear with respect to the tile size ($O(n)$) where n is the number of pixels in the DSM.

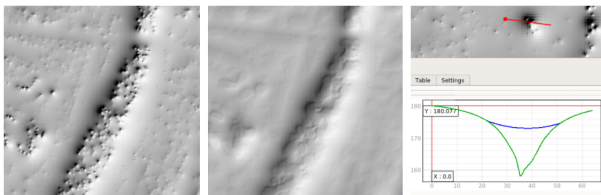


Figure 12. Left: DTM with pits from the DTM extraction step. Center: DTM after the post-processing step. Right: pit profile (Green: DTM from DTM extraction step, Blue: DTM after post-processing step).

Input MNS shape (px)	Runtime	Memory used
1000*1000	11s	259Mb
5000*5000	55s	1,8Gb
10000*10000	02min47	3,1Gb
40000*40000	27min49	38Gb

Table 1. Bulldozer runtime statistics according to the input DSM size. Execution environment: 16 CPU, 60GB RAM with Float32 input DSM. Bulldozer configuration: *max_object_size* = 16, *n_outer* = 100, *n_inner* = 10, *filled_DSM* = *True*

4. RESULTS

4.1 Multi-sensor compatibility

Bulldozer has been designed to take as input any DSM as long as it is provided in raster format. This tool can be used on rasterized airborne Lidar DSMs as much as on very low resolution satellite DSMs. One of Bulldozer’s strengths is its ability to handle input DSMs of very heterogeneous quality. Unlike the methods presented in section 2, Bulldozer is more robust to handle poor quality input DSMs (noisy or incomplete).

4.2 Benchmark

In order to overview the performances of Bulldozer, a benchmark has been set up. This benchmark is based on a comparison between Bulldozer and a highly resolved ground truth provided by IGN² over French territory. In order to evaluate the ability of the method to process heterogeneous landscapes, the areas were selected for their topographic diversity: urban area (high frequency reliefs), rural area (few reliefs) and mountainous area (high altimetric variation). In this benchmark, the DSMs were generated with the CARS software from Pleiades stereo and tri-stereo images (50 cm resolution). The results of the comparison between the DTM computed with Bulldozer results and ground truth are available in Table 2. The selected metrics are: mean, standard deviation (STD), Root Mean Square Error (RMSE), median and Median Absolute Deviation (MAD). Median and MAD are relevant to be selected since they are robust to outliers. These results prove the versatility of Bulldozer since it is able to compute DTMs for a wide variety of landscapes while maintaining a low elevation error compared to the ground truth. The results are similar to the ones obtained with the methods using exogenous data described in section 2.1. Figure 13 shows for each studied area: the input DSM, the output DTM computed by Bulldozer and the ground truth. Figure 14 illustrates the Cumulative Distribution Function (CDF) of the height difference between the computed DTM and the RGE alti for each study area. Figure 15 allows to visualize the performance of Bulldozer when compare to the RGE ground truth. In this figure, correlation issues are also observed in the DSM which res-

² RGE ALTI®. Source: <https://geoservices.ign.fr/rgealti>

Location <i>Topography</i>	Mean	STD	RMSE	median	MAD
Nice <i>rural mountains</i>	-2.22	3.93	4.52	-0.82	1.49
Toulouse <i>periurban hills</i>	2.28	1.82	2.92	1.98	0.97
Paris <i>urban flat</i>	-1.77	2.36	2.95	-1.87	0.76

Table 2. Comparison between the DTM produced with Bulldozer and the IGN RGE alti ground truth.

ults in altimetric aberrations. However thanks to the robustness of Bulldozer, it does not affect the generated DTM.

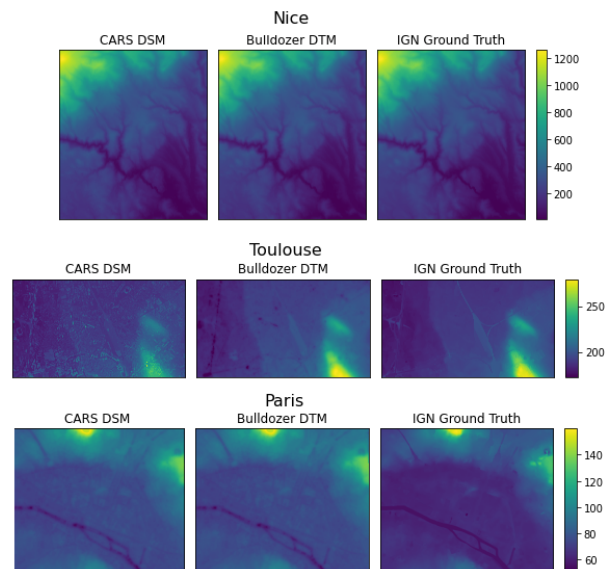


Figure 13. DSM, computed DTM and ground truth for each area used in the benchmark. Top: Nice, Middle: Toulouse, Bottom: Paris.

5. CONCLUSIONS & PERSPECTIVES

In this paper, we describe a modified cloth simulation algorithm to compute a DTM from a DSM without using exogenous data that means having the information of ground pixels as input. Our method can be applied to DSMs of any type over landscapes such as urban and rural areas with various terrain elevation. Our method is notably robust to incomplete information such as occlusions and correlation noises in correlation DSMs. Finally, our method is fully scalable and can be applied efficiently on input DSMs of arbitrary size. A first research axis is to improve the management of the occlusion areas and the detected disturbed areas by filling properly those regions in the input DSM. We are currently working on a filling strategy based on the estimation of a plan computed by a least square with the smallest heights of the border pixels of these regions. Then, a DTM can be initialized using this filled DTM by a local inverse distance weighting interpolation using local height minima. The first results are really promising. A second axis of research would be the analyse the benefits of using an above-ground mask as input of the cloth simulation algorithm by straightly assigning some pixels of the cloth to the DSM pixels marked as ground. Finally, a third axis of research would

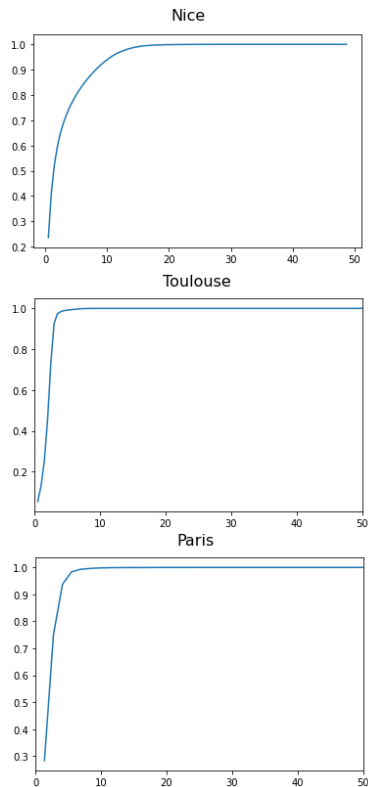


Figure 14. Top: Nice CDF in meter, Middle: Toulouse CDF in meter, Bottom: Paris CDF in meter.

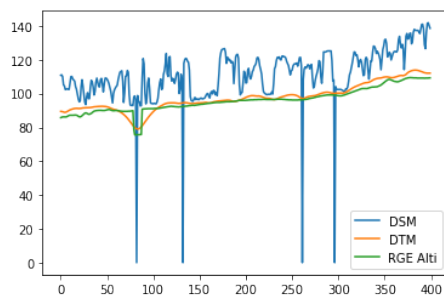


Figure 15. 1D profile of an area in Paris. Blue: DSM, Green: Ground truth (RGE Alti) and Orange: Bulldozer DTM.

be to take into account the confidence map provided by the recent CARS software via Pandora to improve the detection of the disturbed areas.

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.
- Ewelina Rupnik, Mehdi Daakir, M. P. D., 2017. MicMac – a free, open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*. <https://doi.org/10.1186/s40965-017-0027-2>.

J. Michel, E. Sarrazin, D. Y. M. C. F. B. J. M. D. A. E. J. B. O. M., 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>.

Krauß, T., 2018. A New Simplified DSM-to-DTM Algorithm—dsm-to-dtm-step.

L. Lebègue, E. Cazala-Hourcade, F. L. S. A. O. M., 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.

Shean, D. E., O. A. Z. M. B. E. S. I. R. J. C. C. P. M. P. J., 2017. An automated, open-source pipeline for mass production of digital elevation models (DEMs) from very-high-resolution commercial stereo satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116. <https://doi.org/10.1016/j.isprsjprs.2016.03.012>.

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>.