Large-scale DTM Generation from Satellite Data

Liuyun Duan
LuxCarta
lyduan@luxcarta.com

Mathieu Desbrun
Caltech
mathieu@caltech.edu

Anne Giraud
LuxCarta
agiraud@luxcarta.com

Frédéric Trastour
LuxCarta
ftrastour@luxcarta.com

Lionel Laurore
LuxCarta
lionel@luxcarta.com

Abstract

*In remote sensing, Digital Terrain Models (DTM) generation is a long-standing problem involving bare-terrain extraction and surface reconstruction to estimate a DTM from a Digital Surface Model (DSM). Most existing methods (including commercial software packages) have difficulty handling large-scale satellite data of inhomogeneous quality and resolution, and often need an expert-driven manual parameter-tuning process for each geographical type of DSM. In this paper we propose an automated and versatile DTM generation method from satellite data that is perfectly suited to large-scale applications. A novel set of feature descriptors based on multiscale morphological analysis are first computed to extract reliable bare-terrain elevations from DSMs. This terrain extraction algorithm is robust to noise and adapts well to local reliefs in both flat and highly mountainous areas. Then, we reconstruct the final DTM mesh using relative coordinates with respect to the sparse elevations previously detected, and induce preservation of geometric details by adapting these coordinates based on local relief attributes. Experiments on worldwide DSMs show the potential of our approach for large-scale DTM generation without parameter tuning. Our system is flexible as well, as it allows for a straightforward integration of multiple external masks (*e.g. *forest, road line, buildings, lake* etc.*) to better handle complex cases, resulting in further improvements of the quality of the output DTM.*

1. Introduction

Achieving automatic 3D modeling from remote sensing imagery in a robust and scalable way remains a challenge. With the development of earth observation techniques, a multitude of data sources are available for 2D semantic understanding and 3D reconstruction. Satellite data, compared to aerial data, offer a complete worldwide coverage with a daily revisit frequency: massive, easily-accessible archives of satellite data provide a scale of coverage that aerial data can not deliver. In order to leverage these existing datasets, automatic 3D modeling from satellite images requires two critical tasks: the generation of Digital Surface Models (DSM) and Digital Terrain Models (DTM). DTMs can seemingly be generated from DSMs by removing the above-ground objects and reconstruct the bare-earth terrain; consequently, height information of 3D models can be obtained by subtracting the DTM from the DSM. However, DSMs computed from various satellites in different conditions have inhomogeneous quality and spatial resolutions, bringing significant challenges for automatic and robust generation of high-quality DTM.

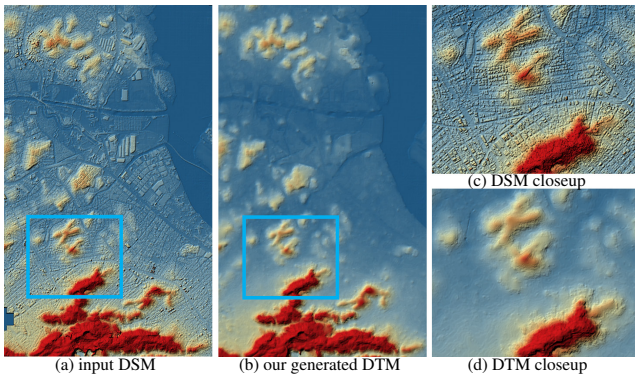


Figure 1. **DTM generation.** Our approach was run on a DSM of Rio de Janeiro (Brazil) with urban areas on reliefs: (a) DSM in 1 meter resolution from satellite imagery, (b) resulting DTM without user interaction, (c) and (d) closeups in a zone with dense building areas on low-amplitude reliefs.

Contributions. In this paper we propose a large-scale DTM generation method capable of handling a variety of spatial resolutions (0.5 to 30 m), quality and noise levels of typical DSMs, either fully automatically or with user guidance. Fig. 1 shows an example of DTM generation demonstrating the potential of our method. Our contributions are threefold:

- A novel multiscale morphological analysis of a DSM is formulated to construct feature descriptors, from which one can extract ground elevations reliably and accurately, even in the presence of noise;
- DTM generation from scattered elevations is achieved via the least squares solution of a sparse linear system based on affine-invariant local coordinates, avoiding the artifacts of interpolation based on differential equations;
- Our scalable method can integrate semantic masks to further improve the quality of the generated DTM, which is desirable for various industrial applications.

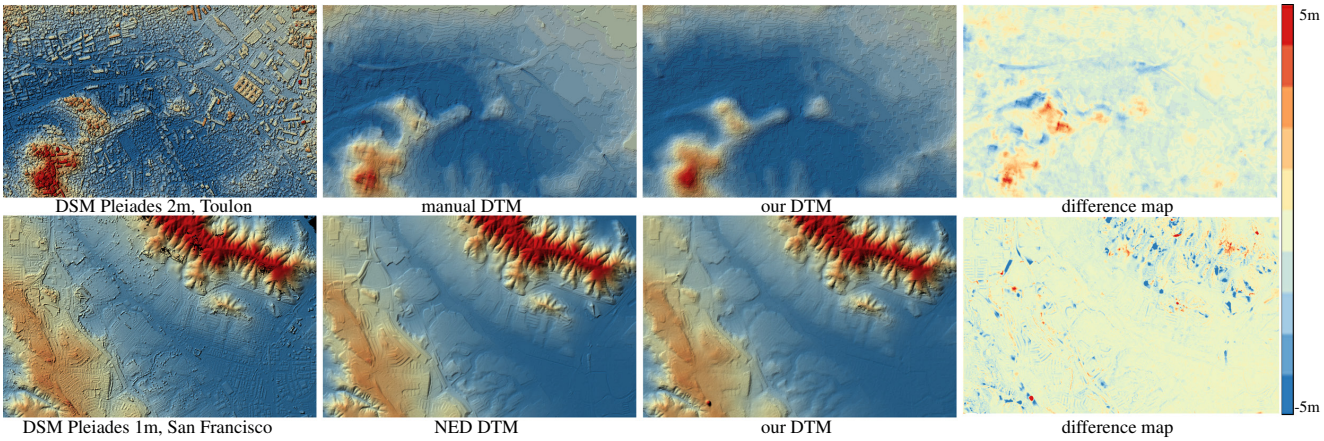


Figure 2. **Comparisons.** DTMs generated from DSMs (left) by our method, compared to a manually produced DTM and the USGS NED DTM (1 meter). Difference maps (right) were computed by subtracting the reference DTM from our DTM, and shown with a color ramp.

2. Related work

DTM generation from satellite data sources consists of two important steps: ground/above-ground classification, followed by surface interpolation. Scattered elevations are extracted from the classification, from which a DTM surface can be reconstructed by interpolating/approximating these known elevations. Related works can be roughly divided into three categories: rule-based terrain extraction, learning-based terrain extraction and surface interpolation.

Rule-based terrain extraction. Geospatial datasets involve a huge variety of types of terrain and objects. Many approaches [2, 18] apply rule-based methods to classify ground/above-ground elevations from satellite based DSMs. Most of these methods make two important assumptions: (a) the terrain is continuous and rarely contains sharp height differences locally; (b) the above-ground objects (buildings, trees *etc*.) are usually not too massive compared to the scale of terrain relief. Based on these assumptions, different rules have been proposed to classify ground/above-ground pixels by analyzing the altitude difference with local neighbors, either through (a) statistics of filtered results (*e.g*., through morphological operations [16]) or (b) terrain propagation constrained by a threshold on gradient values. Since filters are applied to the whole scene, parameter tuning is usually painstakingly difficult when various terrain types are present. Moreover, the fact that the real terrain is not always flat or smooth brings additional difficulties for pure rule-based methods.

Learning-based terrain extraction. With the rapid development of deep learning techniques, learning-based methods have become commonplace for object segmentation due to their impressive performance. For conceptually well-defined tasks such as "cats&dogs" [14] or object instance segmentation [6], deep neural networks are able to learn corresponding feature descriptors automatically from high quality training data. In remote sensing, particularly for DTM generation, deep-learning-based algorithms have been formulated for LiDAR point clouds [7], radiometric images (mostly aerial imagery) plus a DSM raster [5, 20], and even for DSM tiles [13] to extract ground/off-ground points—with decreasing performance since less information is contained in a single satellite DSM tile than in a dense LiDAR data. In sharp contrast to object segmentation, current deep-learning-based terrain extraction methods have *not* yet demonstrated superior terrain extraction accuracy from satellite data [13, 20, 5]. Possible reasons include the lack of sharp contrast between ground and above-ground regions, as well as the huge variety of objects and relief types found in satellite DSMs.

Surface interpolation. Many interpolation methods such as regularized splines [4], triangle-based piecewise cubic polynomials [17], and ANUDEM (Australian National University Digital Elevation Modeling) [8, 9] are commonly used for digital terrain surface reconstruction. In particular, ANUDEM applies an efficient morphological approach to reconstruct a regular grid of elevations from scattered elevations. While a selection of streamlines is usually needed to minimize terrain roughness, the approach is usually quite robust to noise. Gridfit-based methods [12, 10, 3] generate a surface approximation from a set of scattered points based on a hierarchical partitioning of data space. However, relying only on spatial regularization based on Laplacian (or other differential operators) rarely results in a realistic as-flat-as-possible DTM surface: even though a hierarchical space partitioning operation can help reduce unwanted oscillations, the extreme sparsity of bare-terrain points often leads to large artifacts in the reconstruction. In a very different context, a recent approach to non-linear dimensionality reduction proposed a new embedding approach from sparse constraints: the spectral affine kernel embedding (SAKE) method [1] uses a whole set of local relative coordinates of points w.r.t. their closest neighbors to achieve an as-flat-as-possible embedding robust to irregular sampling and noise,

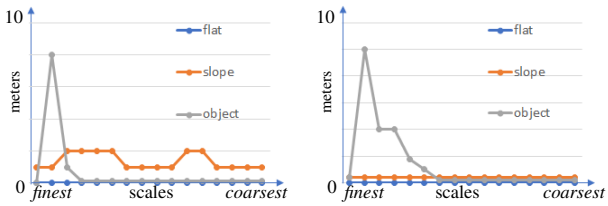


Figure 3. **Multiscale morphological profiles.** Examples of profiles based on $\{D_e^s\}$ (left) and $\{D_{op}^s\}$ (right); horizontal axis represents the 16 scales of analysis, vertical axis in meters.

yet only involving a sparse linear system—thus fitting the requirements of DTM generation quite well.

In this paper, we introduce a new multiscale morphological analysis to extract scattered ground elevations reliably, followed by a modified SAKE interpolation (that we call LAKE interpolation). We will show that this combination of improvements over previous works offers a more flexible and scalable DTM generation method that is adaptive to different terrain types, thus allowing worldwide DTM generation without user interaction.

3. Core foundations

Two core components of our approach – one for terrain extraction, the other for interpolation – are introduced next.

3.1. Morphological operation

Morphology filters [19] are commonly used to remove above-ground objects from DSM data [21]. Two basic operations are involved in morphological filtering: erosion and dilation. For a DSM raster denoted as D, its erosion is denoted as D_e while D_d represents its dilation. Eroded and dilated rasters are easily evaluated as:

$$D_e(i) = \min_{j \in w(i)} D(j), \qquad D_d(i) = \max_{j \in w(i)} D(j), \tag{1}$$

where i denotes a pixel index located at (x_i, y_i) in D, j denotes one of its neighbors with coordinates (x_j, y_j), and $w(i)$ denotes a fixed-size neighboring window around i. The erosion (resp., dilation) extracts the minimum (resp., maxixum) from the set of values within the window around a pixel. From these two simple operations, we can also generate an "opening" morphological operation defined as:

$$D_{op} = D_e \circ D_d. \tag{2}$$

Since above-ground objects in a DSM raster (such as trees and buildings) are usually higher than surrounding elevations, an opening operation removes above-ground objects whose size is smaller than the operation window w. However, applying this morphological operation negatively affects the mountain relief: any non-constant local slope will be altered by such morphological filtering. In this paper, we propose to apply D_e and D_{op} on a sequence of window sizes to classify which DSM pixel is ground-level and which is above-ground, instead of using the eroded/dilated DSM to prevent elevation errors. A multiscale approach removes the need for fine-tuning a single window size of erosion/dilation, and provides rich statistics on the topography even in the presence of noise.

3.2. Spectral affine-kernel embedding

A recent non-linear dimensionality reduction SAKE [1] appraoh aims at embedding irregular and noisy data on smooth low-dimensional manifolds. As it turns out, their approach is particularly appropriate to construct a DTM as a bare-terrain can be thought of as a smooth two-dimensional embedding in $3D$ of a surface given as a few scattered elevations. Originally, SAKE formulates the $2D$ embedding problem as a non-linear deformation that only requires the construction of local affine-invariant coordinates, followed by sparse matrix eigenanalysis. Spatial relationship between neighboring points is built through an affine-precise linear combination of the singular vectors of a relative coordinates matrix. More precisely, for a point p_i with coordinates $X_i = (x_i, y_i)$ and its neighbors in a K-neighborhood with coordinates $X_{j_k} = (x_{j_k}, y_{j_k})$ for $k \in [1, K]$, the relative coordinate matrix E_i is defined as

$$E_i = (X_{j_1} - X_i;\ X_{j_2} - X_i;\ \ldots\ X_{j_K} - X_i). \tag{3}$$

The basis of the kernel of matrix E_i can be constructed through Singular Value Decomposition (SVD): they are the last $K-2$ right singular vectors $\{\mathrm{w}^q \in \mathbb{R}^2\}_{q=1\ldots K-2}$ of unit lengths. Then by definition of the kernel, X_i can be written as a linear combination of its neighbors X_j through

$$\Big(\textstyle\sum_j [\mathrm{w}^q]_j\Big) X_i = \textstyle\sum_j [\mathrm{w}^q]_j X_j \quad \forall q \in [1 \ldots K-2],$$

where $[\mathrm{w}^q]_j$ represents the j^{th} coordinate of the vector w^q. These weights can be understood as the generators of *all possible ways to express a point p_i as a linear combination of its K-neighbors*. The set of $K-2$ linear combinations of neighbors summing up to the original point can thus be written in a matrix form as $L_i X = 0$, where $X = (X_i\, X_{j_1}\, X_{j_2} \ldots X_{j_K})^T$ and the matrix L_i is written as:

$$L_i = \begin{pmatrix} \sum_j [\mathrm{w}^1]_j & \sum_j [\mathrm{w}^2]_j & \ldots & \sum_j [\mathrm{w}^{K-2}]_j \\ -\mathrm{w}^1 & -\mathrm{w}^2 & \ldots & -\mathrm{w}^{K-2} \end{pmatrix}^T. \tag{4}$$

The authors of SAKE argued that using a basis of all such combinations is more geometrically meaningful than using one single local linear relationship per data point (like in the case of Laplacian regularization): in essence, SAKE constructs a "multi-Laplacian" quadratic form assembled based on the neighborhood geometry, which penalizes *any* embedding that is not locally affine. The resulting reconstruction, based on a sparse-matrix eigenanalysis of the sum of all local quadratic forms, offers an as-rigid-as-possible interpolation between scattered anchor points [1] that is particularly appropriate for DTM generation from sparse bare-terrain elevations. In this paper, we further simplify SAKE to transform it into a least-squares problem instead, in order to improve computational efficiency.

4. Multiscale morphological analysis

Morphological filtering as described in Sec. 3.1 can remove small above-ground objects, but may significantly affect the elevations of the DSM in the process. In this section, we propose to use morphological operations at various scales, not as a way to filter out above-ground objects as in [22], but as a classifier of ground/above-ground pixels in the DSM. Our approach is inspired by [16], but adapted to elevation maps by using erosion and opening (instead of opening/closing) operations, and with feature descriptors better adapted to remote sensing.

Multiscale erosion and opening operations. First, we perform erosion and opening operations on S different scales. This simply amounts to evaluating D_e and D_{op} for different window sizes w (see definition in Eq. (1)). Let's denote by s the (integer) scale level, varying from a smallest widow size at scale $s = 1$ to the largest window size at scale $s = S$ where S is the total number of scales we use. Typically, we use $S = 16$ with window sizes of 10m, 20m, 30m, ..., and 160m for high resolution DSMs; low-resolution DSMs (10m accuracy and above) require less scales. For each scale s, we perform on the original DSM D the two morphological operations, which we denote by D_e^s and D_{op}^s respectively. For conciseness, we define $D_e^0 \equiv D_{op}^0 \equiv D$, i.e., the zero scale morphological operations are defined to be the original DSM.

Multiscale spatial profile. From these series of multiscale morphological operations, we deduce two "profile" vectors P_e and P_{op} of size $|P_e| = |P_{op}| = S$ for each DSM pixel, encoding its neighborhood at various scales. For pixel i, the profile vector is simply defined as:

$$P_e(i) = \big(D_e^1(i) - D_e^0(i); D_e^2(i) - D_e^1(i); ...; D_e^S(i) - D_e^{S-1}(i)\big),$$

and similarly for P_{op}; i.e., the profile vectors represent the *change* between two consecutive scales of the results of the morphological operations. Note that as a consequence, these profiles only measure relative spatial properties present at different scales. Fig. 3 shows the intuition behind the profiles derived from erosion and opening.

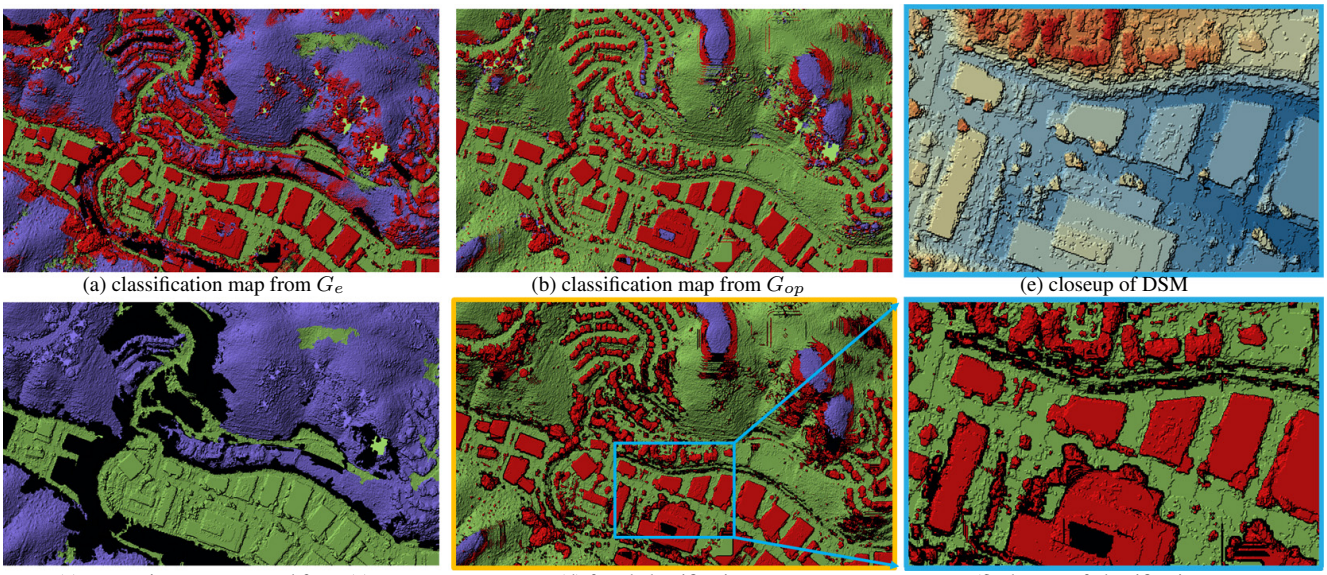
Feature vectors. From the multiscale spatial profiles P_e and P_{op}, we further extract feature descriptors as follows:

- $\bar{F}_e$, $F_e^{\min}$, $F_e^{\max}$, F_e^{var} denote respectively the mean, minimum, maximum and variance of a given profile $P_e(i)$, measuring the distribution of elevation changes in the multiscale erosion operations. Similarly, $\bar{F}_{op}$, $F_{op}^{\min}$, $F_{op}^{\max}$, F_{op}^{var} are computed from $P_{op}(i)$.
- $\overline{\delta F}_e$ and δF_e^{var} are the mean and variance of the vector of differences between adjacent elements of $P_e(i)$, reflecting the distribution of consecutive differences of elevation changes. Same for $\overline{\delta F}_{op}$ and $\delta F_{op}^{\text{var}}$.
- F_e^{sum} denotes the sum of the maximum change $F_e^{\max}$ and of the two values of the profile with indices immediately adjacent (on each side) to the index of the maximum. This feature computes the integral of erosion changes on three consecutive scales around the maximum, helping better distinguish above-ground objects when no sharp elevation step is visible due to noise. Same for F_{op}^{sum}.
- F_e^{flat} and $F_e^{\text{flat}\sigma}$ denotes the mean and standard deviation of the values in $P_e(i)$ after removing the maximum value and the two values with immediately adjacent indices. As the window size increases, an above-ground object will be eventually filtered out, so the profile exhibits a series of major drops surrounded by flat zones since most above-ground objects appear on a fairly flat terrain. This feature thus helps distinguish the presence of above-ground objects. Same for F_{op}^{flat} and $F_{op}^{\text{flat}\sigma}$.
- F_e^{maxZ} denotes the maximum standard score (i.e., z-score) of the elements in the profile $P_e(i)$. This feature helps confirm the presence of an above-ground object by checking if $F_e^{\max}$ is a salient change in the profile. This feature distinguishes above-ground objects from mountainous reliefs as a mountain terrain does not change as sharply as man-made objects or trees.

Classification through erosion and opening. Based on the set of features above, we perform a rule-based classification of all pixels of the input DSM as follows. For each pixel i, we assign a label $\ell_e(i)$ based on the features derived from $P_e(i)$ as follows:

$$\ell_e = \begin{cases} \textit{flat} \text{ if} & (F_e^{\max} < \theta_1)\ \&\ (\overline{F}_e < \theta_2)\ \&\ (|\overline{F}_e| < \theta_3) \\ & \&\ (F_e^{\text{var}} < \theta_4)\ \&\ (\delta F_e^{\max} < \theta_5); \\ \textit{slope} \text{ if} & \big[(F_e^{\min} < \theta_6)\ \&\ (F_e^{\max} < \theta_7)\big] \\ & \mid \big[(\overline{F}_e > \theta_8)\ \&\ (\overline{\delta F}_e > \theta_9)\ \&\ (\delta F_e^{\text{var}} < \theta_5)\big] \\ & \mid \big[(F_e^{\text{maxZ}} > \theta_{10})\ \&\ (F_e^{\text{flat}} > \theta_{11})\big]; \\ \textit{object} \text{ if} & \big[(F_e^{\max} > \theta_7)\ \&\ (F_e^{\text{maxZ}} > \theta_{13})\big] \\ & \mid \big[(F_e^{\text{sum}} > \theta_{14}\ \&\ F_e^{\text{flat}} < \theta_{11} \\ & \quad \&\ F_e^{\text{flat}\sigma} < \theta_{12})\big]; \\ \textit{other} & \text{otherwise.} \end{cases}$$

We define the various thresholds as follows: $\{\theta_i\}_{i=1..14} = \{1, 1, 1.5, 3, 3, -2, 5, 3, 5, 3, 0.5, 1, 2, 1.5\}$. Given that we always considered relative profiles, these parameters are robust to very different types of DSMs with arbitrary quality and do not require tuning. The rules were derived from the following observations: (a) *flat*: the distribution of the profile is almost uniform around 0 at all scales, indicating a rather flat bare-terrain; (b) *slope*: if there is a large difference between max and min and no large jumps, or if the average score is large, or if there are both large jumps and no big flat regions, then the pixel is in a mountainous region;; (c) objects: if the profile exhibits both large max and standard score, or a large jump across multiple scales followed by flat regions, then the pixel is above-ground (presence of building, tree, etc). If none of these cases happen, we classify the pixel as *other* (which typically applies to between



(a) classification map from G_e (b) classification map from G_{op} (e) closeup of DSM

(c) mountain map generated from (a) (d) fused classification map (f) closeup of classification map

Figure 4. **Classification maps.** Results of our erosion-based classification ℓ_e (a) and opening-based classification ℓ_{op} (b), mountain map (c) and the final classification ℓ (d), along with closeups of the DSM (e) and of the final classification map (f). Color convention: green (*flat*), violet (*slope*), red (*object*) and black (*other*).

1% to 3% of the DSM pixels). The associated parameters used in these rules were found after exploring statistics from a number of input DSMs in order to find reliable thresholds.

Finally, the same classification process (with identical thresholds) is achieved based on the multiscale opening-based feature $P_{op}(i)$ of each pixel of the DSM, leading to a second classification $\ell_{op}(i)$ of the initial DSM as well.

5. Classification map

The multiscale morphological analysis we introduced above results in two different classications, ℓ_e and ℓ_{op}, based on a rule-based analysis of the feature descriptors derived from erosion and opening operations. While many pixels in the DSM end up with the same classification in both (Fig. 4(a)&(b)), there are discrepancies that need to be resolved to get a single, high-quality classification map.

5.1. Fused classification

Based on our experiments, multiscale erosion analysis is particularly sensitive to elevation changes, and thus most accurate at detecting above-ground objects; multiscale opening analysis, on the other hand, is better at detecting bare-terrain pixels, be they classified as *flat* or *slope*. Therefore, we perform a fusion of the two classifications into a final classification ℓ as follows:

$$\forall i, \quad \ell(i) = \begin{cases} \textit{object} & \text{if } \ell_e(i) \text{ is } \textit{object}, \\ \ell_{op}(i) & \text{otherwise.} \end{cases}$$

That is, we trust the *object* labels detected by ℓ_e foremost, and otherwise use the resulting classification from ℓ_{op}. This simple fusion allows us to obtain an accurate fused classification ℓ containing sharp transitions between above-ground pixels and bare-terrain pixels (see Figs. 4(d) and closeup (f)) rivaling with (and often outperforming) state-of-the-art results from deep learning DTM generation computed from far less noisy LiDAR data in [5]: we significantly reduce the typical "ambiguity" regions around above-ground objects, allowing for a proper treatment of complex scenes, *e.g.* dense building areas with narrows streets.

Cleanup. Further cleaning up of the final classification map can be performed for added robustness. Through region growing, we identify isolated regions of either *flat* or *slope* pixels of total areas below a certain size; pixels in these regions are simply classified as *other*, as they are likely to be the results of noise or of an ambiguous case.

5.2. External mask integration and user integration

Since false-positive and true-negative errors are unavoidable in large-scale scenarios containing a large variety of different terrains and cities, external masks such as road lines, forests, lakes, or buildings, are always helpful. With the rapid development of semantic classification (in particular using deep learning techniques), an increasing number of masks are becoming available. Our system is flexible enough to integrate such masks into our classification map in a straightforward fashion. For example, road lines help get more bare-terrain information in dense inhabited areas over hilly regions by labeling roads as ground while avoiding *object* areas. Similarly, a forest map can help disambiguate the challenging scenario of a large, dense forest which may be erroneously recognized as flat through automatic analysis. Semantic masks thus always improve the quality of the generated DTMs in large-scale

applications. Moreover, an expert can further modify manually the labeling established by our algorithm.

5.3. Mountain map

Sometimes, establishing specific maps (such as a mountain map) is a desirable task that our multiscale analysis also helps with. We found that a simple post-processing of the erosion-based classification ℓ_e (see Fig. 4(a)) leads to very clean mountain maps: we simply detect regions of *object* pixels surrounded by *flat* pixels and convert them to *flat*; same for regions of *object* pixels surrounded by *slope*, which we convert to *slope*; and finally, regions of *object* pixels surrounded by a mixture of *flat* and *slope* pixels are turned into *other*. The result, shown in Fig. 4(c), nicely delineates flat from mountainous regions, at very little additional computational cost.

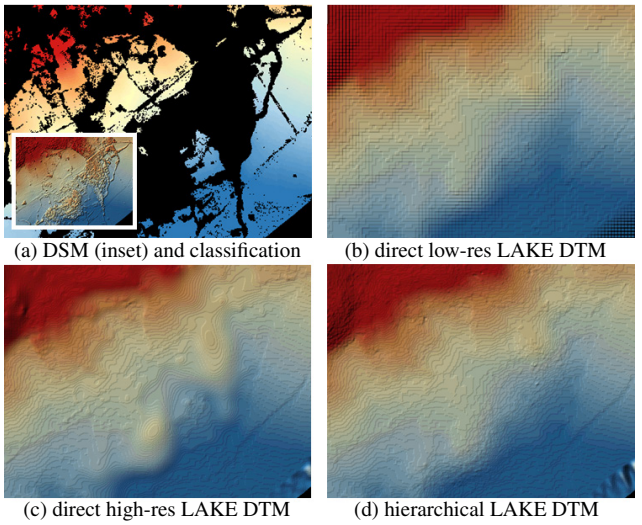
6. Surface interpolation through LAKE

As briefly discussed in Sec. 2, the SAKE approach to surface interpolation removes many of the issues plaguing typical DEI interpolation methods by enforcing a series of local linear combinations between nearby points to penalize all non-linear slopes. In this section, we propose a simpler expression of SAKE, coined LAKE as the Spectral solve involved in the original approach is replaced by a faster Least-squares solve. We then apply this technique to create a DTM map, of the same size as the input DSM, that interpolates the ground elevations (read from the input DSM) of the pixels labeled as either *flat* or *slope*.

6.1. Least-Squares Affine-Kernel Embeddings

To find the elevation $h(i)$ of each pixel i not classified as *flat* or *slope* in the DTM, we proceed in three steps.

Assembling the quadratic form Q. Just like in the original SAKE approach, we first construct a sparse quadratic form Q based on the local weights derived from the right singular vectors of Eq. (3) for each pixel neighborhood as summarized in Sec. 3.2. This quadratic form can be thought as a penalty on elevations to force them to be as locally linear as possible, which will help regularize the interpolation. Since we wish to construct a regular grid of elevations (i.e., the final DTM), the construction of the weights based on the neighborhood of each pixel can be precomputed very efficiently. Compared to SAKE, however, we consider size-varying neighborhoods: for every pixel, we use its one-ring neighborhood (i.e., the 8 pixels touching it) by default, but use a larger two-ring neighborhood (i.e., the 24 closest pixels) if the pixel is labeled as *slope*. This modification of the original approach (which uses a constant neighborhood size) is particularly important in our case: it indirectly adds local rigidity of the interpolation in steep areas, thus preventing the appearance of small spikes which sparse constraints typically generate.



(a) DSM (inset) and classification (b) direct low-res LAKE DTM (c) direct high-res LAKE DTM (d) hierarchical LAKE DTM

Figure 5. **Hierarchical DTM generation.** (a) Initial sparse bare-terrain model from classification; (b)DTM solved in a lower resolution; (c) DTM by LAKE directly from the bare-terrain model; (d) DTM by hierarchical LAKE with elevations from low-res DTM added in large empty zones of the bare-terrain model.

Constraints. We then assemble a sparse diagonal matrix C that encodes the elevation constraints we wish to impose. Matrix C has a non-zero value C_{ii} only if pixel i of the classification map is $\ell(i)=$*flat* or *slope*. We use $C_{ii}=\alpha$ for *flat* elevations and $C_{ii}=2\alpha$ for *slope* elevations to allow for a closer fit of constrained elevations in mountainous regions and a more relaxed fit in nearly flat regions to more efficiently reduce noise in flat parts. We also assemble a vector b containing the elevations of the DSM for pixels labeled *flat* or *slope*, and 0 otherwise. We use $\alpha=(10m/\text{resolution})^2$ in all our results in this paper.

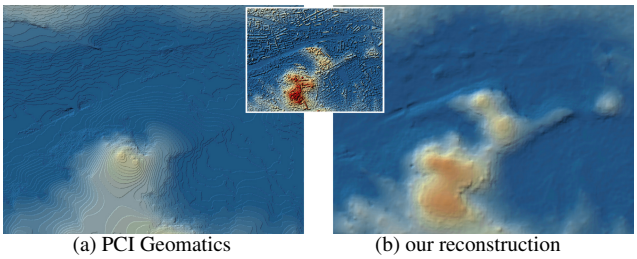
Solve. We now wish to find elevations, stored as a vector H containing the elevation $h(i)$ of every pixel, such that the constraints are satisfied ($CH=b$) while minimizing the quadratic form H^TQH. While the original SAKE proposes an eigensolve, we found much more efficient (and still accurate enough) to simply use a least-squares solution of the set of linear equations: $QH=0$ and $CH=b$. This is achieved by solving the linear system:

$$(Q^TQ+C^TC)H=C^Tb,$$

which can be efficiently evaluated through Conjugate Gradient using the DSM as initialization. The result of this linear solve is our final DTM.

6.2. Hierarchical DTM generation

As a way to both accelerate the linear solve of LAKE and improve robustness to extreme sparsity of constrained elevations (often present for satellite stereo pairs because of occlusions in dense urban areas or in agricultural fields with repetitive texture), we propose a hierarchical DTM generation process that can handle DSMs with very different quality and geographical scenarios. If the constraints given



(a) PCI Geomatics (b) our reconstruction

Figure 6. **Visual Comparison.** For the Côte d'Azur DSM, the PCI result of [15] with careful hand-tuning by an expert (left) removes the relief details over urban regions, while our automatic result captures a proper elevation map (see DSM in inset).

to LAKE contain large empty regions where no elevations are available (for example, no elevation points detected in a zone of 500×500 m^2, that is, 500×500 pixels in a DSM with 1-meter resolution), the flatness prior used in SAKE may create too much of a bowl-shaped depression or bump in this region. So we compute first a low-resolution DTM for which this artifact will be absent or negligible (because the surface tension induced by the quadratic form is stronger and prevents sagging); from this reference DTM, we then insert the computed ground elevations into the large empty zones to evaluate the fine DSM. Much like a multigrid approach, this two-step technique converges also slightly faster as a coarse solution is used to drive the finest solution. Fig. 5 shows the difference between a direct LAKE solve (c) and a hierarchical LAKE solve (d).

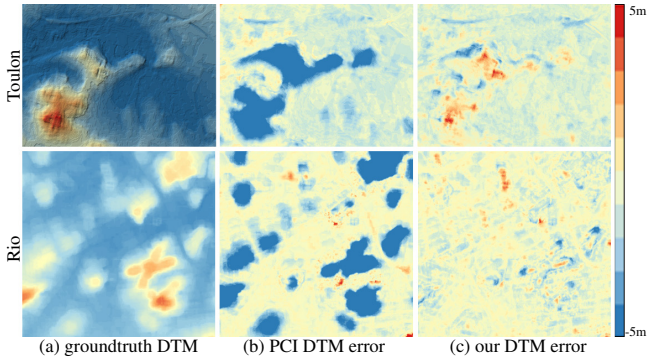
7. Experiments

The DTM generation pipeline we just described was implemented in C++, and all experiments were done on an Intel Core i7 clocked at 4.1GHz with 64GB RAM. Figures shown in this paper were created straight from input DSMs with no parameter tuning and no external masks. Table 1 provides input sizes and running times for all examples shown in this paper.

Times (in min.)	Rio (516M)	Buenos Aires (116M)	Toulon (87M)	Panama (41.6M)	Hatta (35M)
analysis	18	5.1	5.1	4.3	8
classif.	1.9	0.8	0.1	0.1	0.1
DTM gen.	21	10	10	2.5	2
total time	40.9	15.9	15.2	6.9	10.1

Table 1. Performances on DSMs of different types and sizes (Rio from Fig. 1, others from Fig. 2) in terms of running times (for multiscale morphological analysis, classification, DTM generation through LAKE, and total time). Times are expressed in minutes, while sizes are mentioned in number of DSM pixels.

We ran a large number of automatic DTM generations from DSMs of different quality and noise levels, from all over the world. Fig. 8 shows a selection of examples for difficult regions, mixing significant relief and urban structures. Fig. 1 shows Rio de Janeiro as another difficult example. These results offer a visual estimation of the quality of our generated DTMs when no human interaction



(a) groundtruth DTM (b) PCI DTM error (c) our DTM error

Figure 7. **Error plots.** From a manually-produced DTM used as groundtruth (a), we compare the errors of PCI [15] (b) vs. our DTM generation (c), using the same color ramp as in Fig. 2.

is provided. Note that computational times for these DTMs vary from 15 minutes for Toulon to 41 minutes for Rio, averaging 8.93M pixels per min on average. Since every step of our pipeline is easily parallelizable, these timings can be reduced dramatically for machines with more processing power. Our experiments indicate that our morphological analysis takes about 40% of the total time, suggesting that a faster implementation of this part could further reduce the overall computational time. We also measured the quality of our DTMs in Fig. 2 by comparing with two type of ground-truth references: (a) a manual DTM generated by applying progressive filtering of carefully user-segmented regions, and (b) the USGS National Elevation Dataset (NED). After computing the difference between our results and the ground-truth references, we visualized the elevation difference using a corlormap covering from -5 meters to 5 meters. The average absolute error is 1.28 meters over the entire 125km^2 testing areas, and most of the errors are located in mountainous regions and man-altered regions such as bridges. We also compared our results to the current state-of-the-art commercial product from PCI Geomatics [15] in Fig. 6. While this tool generates equally good results in flat urban areas and slightly better on unhabited mountainous regions compared to our method, it fails completely when urban structures or trees are present on rolling hills while our approach continues to perform adequately even in this case.

Note that our approach has been used for the past year by LuxCarta [11] as part of their digital map production pipeline with success, proving its robustness.

8. Conclusions

In this paper, we presented an approach for automatically generating DTMs from arbitrary DSMs. Our method relies on a novel multiscale morphological analysis of the DSM to classify each pixel, and on a subsequent interpolation of reliable elevations through a least-squares smooth embedding approach which does not suffer from the typical oscillations of interpolation methods based on differential

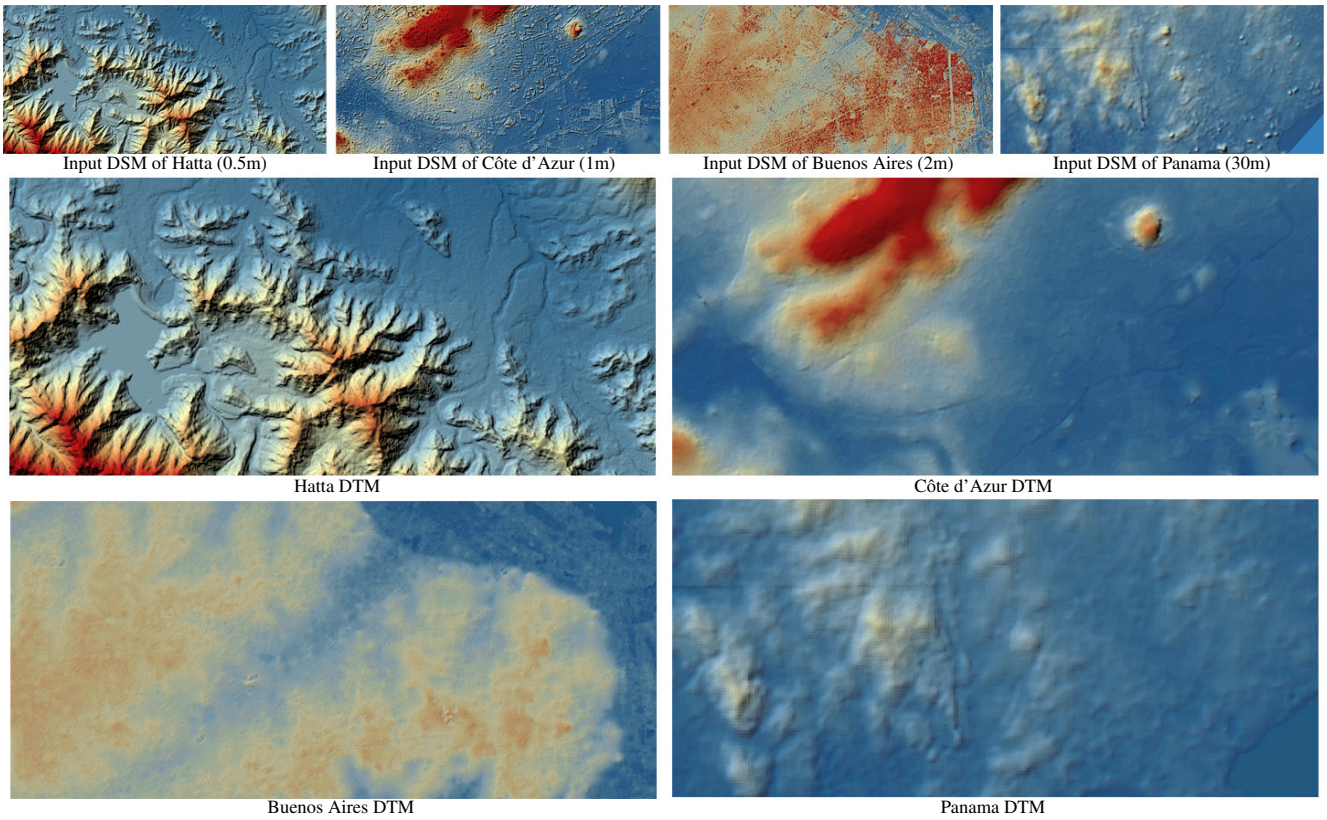


Figure 8. **Generality.** Our method adapts to DSMs of arbitrary terrain types: Hatta, United Arab Emirates, DSM with 0.5 meter spatial resolution from Pleiades; Côte d'Azur, France, DSM with 1 meter from Pleiades; Buenos Aires, Argentine, DSM with 2 meters from SPOT 6; Panama City center, Panama, DSM with 30 meters from ALOS.

equations. We demonstrated that this combination of tools can produce reliable and accurate large-scale DTMs from a large variety of DSMs of various qualities and resolutions, often improving upon state-of-the-art results without any user interaction.

Limitations. While we have been able to successfully construct DTMs of a wide variety of world regions, small relief details on mountains and rocky shores are sometimes mislabeled as *object*s due to their shape similarity with man-made buildings at low resolution. Post-treatment using a mountain map or sea map can help fix these issues, but finding a fully automatic and reliable solution without any additional map would be useful. Moreover, our morphological analysis only focuses on a limited range of scales; we may thus mislabel a wide and dense forest as *flat*. A larger filtering window would remove this issue, but currently, it would come at the cost of slower performances; accelerating the multiscale analysis (*e.g.*, using GPU) could thus be a valuable direction to explore in order to improve both timings and classification accuracy.

Future work. Currently, the labeling of the DSM pixels is done independently for each pixel, allowing embarrassingly-parallel workload. However, the quality of the results (validity of labels and frequency of *other* labels) could be further improved if our rule-based classification considers a small neighborhood, without significantly reducing the efficiency of the classification. Our attempts at implementing this idea never produced much better results, confirming the adequacy of the features derived from our multiscale morphological analysis. Maybe a hybrid method mixing these features with learning-based approaches could be proven useful. Finally, applying our approach to LIDAR is also straightforward, but proper evaluation was not performed due to limited access to large datasets.

Acknowledgments. Our most sincere thanks go first to Max Budninskiy (Caltech) for helping us testing SAKE interpolation early on. Jonathan Lambert provided crucial help with manual ground-truth DTM generation. Sébastien Tripodi also helped with a variety of technical issues throughout the project. Justin Hyland, Véronique Poujade and Yuliya Tarabalka were also very supportive along the way. Finally, MD acknowledges ShanghaiTech for hosting him during the final editing of this paper.

References

[1] M. Budninskiy, B. Liu, Y. Tong, and M. Desbrun. Spectral affine-kernel embeddings. *Computer Graphics Forum*, 36(5):117–129, 2017. 4322, 4323

[2] M. Debella-Gilo. Bare-earth extraction and DTM generation from photogrammetric point clouds including the use of an existing lower-resolution DTM. *International Journal of Remote Sensing*, 37(13):3104–3124, 2016. 4322

[3] J. D'Errico. Surface fitting using gridfit, 2005. MATLAB central file exchange 8998. 4322

[4] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation - Math. Comput.*, 38:181–181, 01 1982. 4322

[5] C. Gevaert, C. Persello, F. Nex, and G. Vosselman. A deep learning approach to DTM extraction from imagery using rule-based training labels. *ISPRS Journal of Photogrammetry and Remote Sensing*, 142:106 – 123, 2018. 4322, 4325

[6] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 4322

[7] X. Hu and Y. Yuan. Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sensing*, 8, 09 2016. 4322

[8] M. Hutchinson. A new procedure for gridding elevation and stream line data with automatic removal of spurious pits. *Journal of Hydrology*, 106(3):211 – 232, 1989. 4322

[9] M. F. Hutchinson, T. Xu, and J. A. Stein. Recent progress in the ANUDEM elevation gridding procedure. In T. Hengl, I. S. Evans, J. P. Wilson, and M. Gould, editors, *Geomorphometry 2011*, pages 19–22, Redlands, CA, 2011. 4322

[10] D. A. Keim and A. Herrmann. The Gridfit algorithm: an efficient and effective approach to visualizing large amounts of spatial data. *Proceedings of IEEE Visualization*, pages 181–188, 1998. 4322

[11] LuxCarta. https://luxcarta.com/. 4327

[12] A. Manconi, P. Allasia, D. Giordan, M. Baldo, G. Lollino, A. Corazza, and V. Albanese. Landslide 3D surface deformation model obtained via RTS measurements, 10 2011. 4322

[13] D. Marmanis, A. Fathalrahman, M. Datcu, T. Esch, and U. Stilla. Deep neural networks for above-ground detection in very high spatial resolution digital elevation models. 01 2015. 4322

[14] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, June 2012. 4322

[15] PCI Geomatics. Geomatica, 2019. http://www.pcigeomatics.com. 4327

[16] M. Pesaresi and J. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Trans. Geoscience and Remote Sensing*, 39:309–320, 2001. 4322, 4324

[17] G. Ramos. Scattered data interpolation using an alternate differential equation interpolant. 03 2019. 4322

[18] G. Sithole and G. Vosselman. Comparison of filtering algorithms. 2003. 4322

[19] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 2004. 4323

[20] G. Tapper. Extraction of DTM from satellite images using neural networks. Master's thesis, Linköping University, Computer Vision. 4322

[21] K. Zhang, S.-C. Chen, D. Whitman, M.-L. Shyu, J. Yan, and C. Zhang. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4):872–882, April 2003. 4323

[22] K. Zhang and Z. Cui. Airborne LIDAR Data Processing and Analysis Tools (ALDPAT), 2007. http://lidar.ihrc.fiu.edu/lidartool.html. 4324