Generation of Folded Terrains from Simple Vector Maps

Elie Michel, Arnaud Emilien, Marie-Paule Cani

▶ To cite this version:

HAL Id: hal-01147920
https://hal.inria.fr/hal-01147920v2

Submitted on 30 May 2016

Generation of Folded Terrains from Simple Vector Maps

Elie Michel[1], Arnaud Emilien[1,2], Marie-Paule Cani[1]

[1] univ. Grenoble-Alpes, CNRS (Laboratoire Jean Kuntzmann) and Inria
[2] LIGUM, Dept. I.R.O., Montreal University

Abstract

While several terrain generation methods focused on plausible watersheds, the fact that most mountains should not be isolated but rather be part of wider scale mountain ranges was seldom considered. In this work, we present the first procedural method that generates folded terrains from simple user input, in the form of some sparse peak distribution on a vector map. The key idea is to infer possible continental plates from this distribution and to use simplified plate tectonics to generate relevant terrain folds. The resulting terrain with large-scale folds, computed in real-time, can be further refined using standard erosion simulation. This leads to detailed terrains with plausible mountain ranges that match the peak distributions and main rivers specified on simple vector maps.

1. Introduction

Terrains are among the most difficult models to create, due to their specific multi-resolution folded structure. Although real terrain data is available, many applications – from films through video games and to serious games – require terrain models that match specific design goals. Achieving this, was it through the manual edition of some existing terrain data or

Figure 1: *Left: Corsica elevation map, showing typical multi-scale terrain folds (©IGN). Middle: Input vector map. Right : The elevation map we output. Note that although not identical to the real data on the left, our method achieves some plausible interpretation of the vector map, with qualitatively similar multi-resolution folds.*

using the indirect control offered by a procedural generation method, takes time and efforts.

This work addresses the case when user design goals can be expressed through a simple vector map, similar, for instance, to the famous Middle-Earth map in Lords of the Rings. This raises an interesting question: Is it possible to generate plausible terrains from the simple vector data on such maps, including coast lines, main rivers, and some sparse set of mountain peaks? This is the question we are addressing here. Since terrain generation from hydrological maps was already studied, we more specifically focus on generating mountain ranges that match the sparse distribution of peaks on a vector map. To our best knowledge, this specific problem was never tackled before.

Plates tectonic plays a major role in shaping landscapes: interaction between plates causes the terrain to fold, resulting into a series of mountain ranges (Figure 1, left). Our approach to infer plausible folds that are consistent with a user-designed vector map (Figure 1, middle) is to group vector peak elements into possible continental plates, and to use simplified plate tectonic to compute consistent large scale terrain folds. The later are then refined using smaller scale hydraulic erosion simulation while accounting for the rivers on the map. This results into visually appealing terrains that do match the user intent (Figure 1, right).

After reviewing related work (Section 2), we present our processing pipeline (Section 3). We then detail the two main contributions: a method for creating continental plates from a set of peak positions (Section 4) and a simplified tectonic
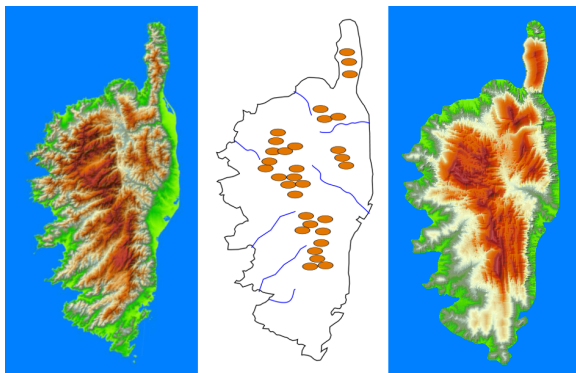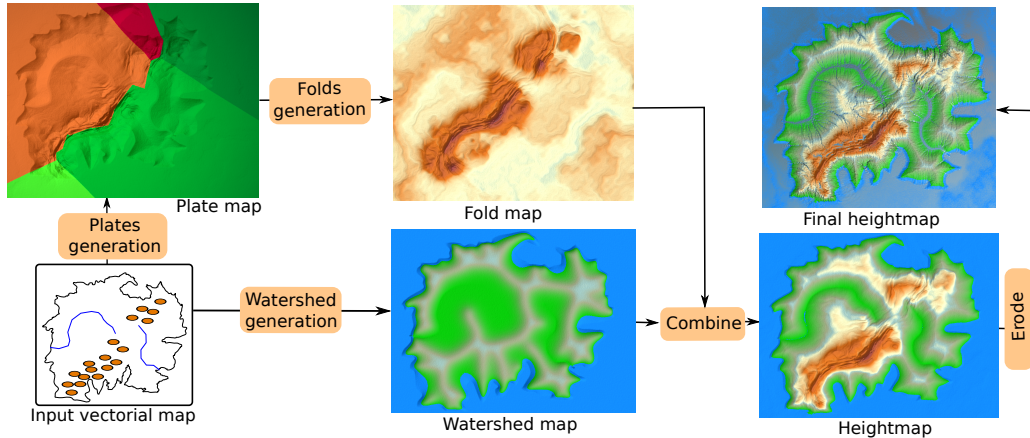
Figure 2: *Overview of our algorithm.*

algorithm for generating terrain folds from these plates (Section 5). The construction of the watershed map used for erosion simulation is then presented (Section 6). We discuss results and limitations before concluding.

2. Related work

The first methods for generating terrains directly tried to capture the fractal nature of mountains, i.e. their self-similar multi-scale structure. This was either achieved using fractal geometry or multi-resolution noise models [EMP*02]. A larger segment of previous work focused on generating plausible watersheds, which eventually created more plausible fractal-like valley structures. These methods either simulated water flows and weathering effects [BF02, MDH07, vBBK08, KBKŠ09], or procedurally built consistent valley systems [KMN88, PH93, BA05, GGG*13]. Most of these terrain generation methods were driven by indirect parameters leading to tedious control through series of trials and errors. In contrast, Genevaux et al. [GGG*13] generated consistent terrains from user-specified rivers-beds depicted on a vector map. In this work, we reuse the idea of such simple user input but tackle the complementary problem of generating plausible mountain ranges from a simple map. We make use of the hydraulic erosion method from [vBBK08] for adding smaller scale features to the folded terrains we generate.

Hydrology is not the only factor shaping landscapes. Indeed, while it organizes valleys at multiple scales, hydrology alone fails structuring mountains into consistent ranges. These ranges result from larger scale geological phenomena, namely folding effects due to plate tectonics. The first attempt, to our knowledge, to take plates tectonics into account for terrain generation [Ale06] generated simple high fields from the simulation of collisions between plates. Contrary to our goals, the result was not used to organize smaller scale mountains into consistent ranges. A more recent method [Vii12] makes use of randomly defined plates to compute a set of locations for mountains. Yet, the method does not provide user control on mountain location and does not include the generation of plausible 3D geometry. In contrast, our goal is to use plate boundaries not as a single fault line, but as some global feature driving smaller scale details.

Lastly, a few methods went further than plate tectonics to model subsurface geology. This falls outside the scope of this work. We refer the reader to [NLVP13].

3. Overview of the method

We use the following pipeline to generate plausible heightmaps from simple vector maps (Figure 2):

1. The user interactively creates a simple vector map, with coast contours, main rivers, and distributions of peaks.
2. We generate possible continental plates and their speed from the peaks, to infer where plate collisions may occur.
3. Plate contours and speeds are used as input of a warping process used to generate noisy terrain folds.
4. To take sketched rivers into account, we generate a watershed map to be able to force the water to flow in them.
5. We combine the fold map with a watershed map.
6. Small scale details are added using erosion simulation.

4. Plates generation

In the real world, collisions between continental plates generate mountain ranges due to terrain folding and their separation creates deep canyons. Geologists make use of different types of data for determining plate boundaries, such as boreholes and seismic sources. But we cannot ask the user for such specialized input. Although inspired from natural plate tectonics, our method therefore simplifies it: we consider plate collisions only and use the location of mountain peaks to extract a plausible set of plates with their contours
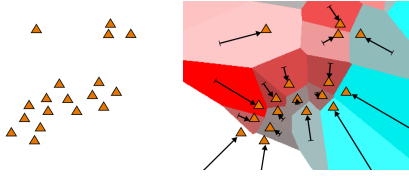
Figure 3: *Input peak positions (left). Sub-plates and associated velocities based on Voronoi cells (right).*
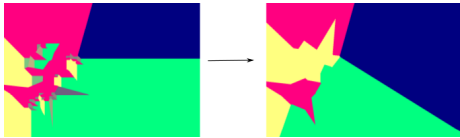

Figure 4: *Sub-plate merging process.*

and velocities. This is done using a bottom-up approach that first generates sub-plates around each peak and then merges them into larger ones. Although extremely coarse from a geological viewpoint, this method matches our goal of generating visually plausible results from simple input.

Sub-plates. We define sub-plates as cells of a Voronoi diagram computed from peak positions on the input map. We estimate velocity u of a sub-plate as the vector from its center of mass c_m to the peak position p. Thus, the nearest a peak is from its cell boundary, the strongest the velocity. Therefore, velocities are directed towards packed regions, as if all sub-plates were converging to central positions and colliding. See Figure 3.

Sub-plate merging process. Tectonic plates are large regions that move together, and thus have a quasi-uniform velocity. Therefore, we use velocity direction to iteratively merge plates. The contour of the resulting continental plates are the locations of terrain folds we have been looking for.

More precisely, we use two simple rules to iteratively merge neighboring plates: Firstly, plates A and B are merged if $u_A \cdot u_B > T_1$, where u_X is the normalized velocity for plate X and $T_1 \in [0,1]$ a user-defined threshold. The smaller T_1, the larger the resulting plates. Secondly, in order to avoid issues of normalizing near-zero vectors, and also to homogenize plate sizes, we merge too little plates with their largest neighbor. A and B are merged if $\frac{\mathcal{A}_A}{\mathcal{A}_B} < T_2$, where $\mathcal{A}_X$ is the area of plate X. We assign the following speed vector to the new plate C, to be ale to iterate (see Figure 4): $u_C = \frac{\mathcal{A}_A u_A + \mathcal{A}_B u_B}{\mathcal{A}_A + \mathcal{A}_B}$.

5. Generation of terrain folds

Terrain folds are consistent sets of features due to collision between plates, and therefore aligned along their boundaries. See the set of horizontal to slightly inclined large
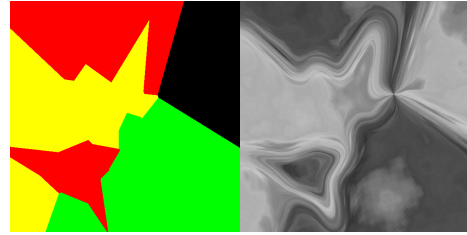

Figure 5: *Initial plate map (left) and resulting fold-based height-map (right).*

folds in Figure 1, left. As simulating the full folding process [Mal10] would lead to heavy computations, we rather use plate shapes and velocities to generate some procedural noise that creates folds using image warping.

Image warping consists in applying a function to an image in order to distort its content. Let I be an image, $I(p)$ the pixel of coordinates p, and $w(p)$ a warping function. The warped image I' is defined by $I'(p) = I(w(p))$. In our method, we use plates velocities to warp a noise function.

Velocity map. We initialize the velocity map V from the associated plate velocity. We then smooth the map using Gaussian blur. The resulting map thus has constant values on each plate, but quickly varying values interpolating two opposite velocity vectors at plate boundaries. This quick, smooth variation will be used to create folds.

Noise warping. The terrain height-map I is initialized using *fractional Brownian motion* noise [MVN68] to get pseudo-random multi-scale details. To get the folding effect, we use the plate velocity map to warp this height-map, using: $w(p) = p + V(p)$. While this only translates height variation on a given plate, this causes the noise to be compressed in a single direction, perpendicular to the plate boundaries, at the border between two plates, due to the quick variation of the translation vector $V(p)$ applied. And as shown in Figure 5 this process generates what we intuitively interpret as folds.

To improve the result, we scale down the resulting height-map I' by a coefficient $k \in]0,1[$ depending on the distance to the peaks drawn on the vector map. This attenuates the folding effects far from peaks, making the result both more natural (the contour of plates should not be visible everywhere) and consistent with user input.

6. Watershed generation and hydraulic erosion

As simple vector maps often depict the main rivers, we allow users to draw them on the input vector map. However, these rivers could be incoherent with the fold-based height-map we just computed, and simply digging rivers into it would not give plausible results. We rather loosely inspire from [GGG*13] to generate a watershed map from the rivers

that is then combined with the fold-based height-map. Finally, we simulate hydraulic erosion as in [vBBK08] to add plausible small scale details.

Watershed map. The watershed map is generated as follows: First, we propagate height along rivers starting from the sea (zero height) using a given small slope factor. Then, we diffuse this height with a larger slope to the other pixels of the map, using a shader. Slope values can be adapted based on distance to the nearest peaks on the vector map to achieve non-linear slope variations. The resulting map defines a terrain where water flows naturally from mountains to rivers and to the ocean.

Combining folds and watershed maps. The watershed elevation map g is combined with the fold-based elevation map f from Section 5 using a mask map m set to some normalized distance to rivers. The resulting map h seamlessly switches from watershed-based information close to rivers to fold-based information: $h(p) = g(p)(1 - m(p)) + f(p)m(p)$.

7. Results

The system is implemented in C++, using OpenGL and GLSL Compute Shaders. Computations are performed on an NVidia 660GTX GPU and an Intel® Xeon® E5-1650 CPU, running at 3.20 GHz with 16 GB of memory.

Figures 1, 2 and the companion video show that our method is able to generate visually realistic terrains. In addition to creating imaginary terrains, we used a map inspired from a real island in Figure 1 to allow qualitative comparison with a real height-map. Performances gathered in Table 1 shows that terrains are generated in a few seconds.

	Plates	Folds	Watershed	Erosion
Fig. 1	2.9 s	2.1 s	6 s	2.4 s
Fig. 2	2.9 s	2.25 s	6.4 s	2.3 s

Table 1: *Computational time, terrain (*1024×1024*).*

Limitations. The use of Voronoi diagrams leads to non-expected results when users input a single line of peaks instead of of thicker mountains clusters: while they expect a single plate boundary along this line, Voronoi cells cut between peaks. Moreover, isolated peaks are not on plate boundaries, and so not on the folded mountain ranges we generate. We could interpret them as volcanoes.

The main issue of our fold generation algorithm is that it generates folds along both convergent and divergent plate boundaries. We could try to mask it to hide folds along divergent boundaries but this would require progressively fading folds where convergent and divergent boundaries meet.

Lastly, the way we merge user-defined rivers with fold-based elevation does not always give good results, since we blend two maps that may not be locally consistent rather than warping them first to make them match.

8. Conlusion

In this paper, we presented a procedural method considering tectonic plate interaction to create a terrain with consistent mountain ranges. Although simple, this method already achieves good results. In future work we would like to extend the method to both oceanic and continental plate types and to consider divergent plates as well as convergent ones.

References

[Ale06] ALEX J.-E.: Creating landscapes with simulated colliding plates. *RIT digital media library, http://hdl.handle.net/1850/1962* (2006). 2

[BA05] BELHADJ F., AUDIBERT P.: Modeling landscapes with ridges and rivers: bottom up approach. In *GRAPHITE* (2005), pp. 447–450. 2

[BF02] BENEŠ B., FORSBACH R.: Visual simulation of hydraulic erosion. In *WSCG* (2002). 2

[EMP*02] EBERT D. S., MUSGRAVE F. K., PEACHEY D., PERLIN K., WORLEY S.: *Texturing and Modeling: A Procedural Approach*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002. 2

[GGG*13] GÉNEVAUX J.-D., GALIN E., GUÉRIN E., PEYTAVIE A., BENEŠ B.: Terrain generation using procedural models based on hydrology. *ACM TOG 32*, 4 (2013), 143. 2, 3

[KBKŠ09] KRIŠTOF P., BENEŠ B., KŘIVÁNEK J., ŠT'AVA O.: Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum 28*, 2 (2009), 219–228. 2

[KMN88] KELLEY A. D., MALIN M. C., NIELSON G. M.: Terrain simulation using a model of stream erosion. *Communications of the ACM 22*, 4 (1988). 2

[Mal10] MALAVIEILLE J.: Impact of erosion, sedimentation, and structural heritage on the structure and kinematics of orogenic wedges: Analog models and case studies. *GSA Today 20*, 1 (2010), 4–10. 3

[MDH07] MEI X., DECAUDIN P., HU B.-G.: Fast hydraulic erosion simulation and visualization on gpu. In *Pacific Graphics* (2007), pp. 47–56. 2

[MVN68] MANDELBROT B. B., VAN NESS J. W.: Fractional brownian motions, fractional noises and applications. *SIAM review 10*, 4 (1968), 422–437. 3

[NLVP13] NATALI M., LIDAL E. M., VIOLA I., PATEL D.: Modeling terrains and subsurface geology. *EG STARS* (2013), 155–173. 2

[PH93] PRUSINKIEWICZ P., HAMMEL M.: A fractal model of mountains and rivers. In *Graphics Interface* (1993), pp. 174–180. 2

[vBBK08] ŠT'AVA O., BENEŠ B., BRISBIN M., KŘIVÁNEK J.: Interactive terrain modeling using hydraulic erosion. In *SCA* (2008), pp. 201–210. 2, 4

[Vii12] VIITANEN L.: *Physically Based Terrain Generation: Procedural Heightmap Generation Using Plate Tectonics*. Metropolia Ammattikorkeakoulu, 2012. 2