CEIG – Spanish Computer Graphics Conference (2019)
A. Jarabo and D. Casas (Editors)

Procedural location of roads using desire paths

Pablo Real[1], Francisco Martínez-Gil[1], Rafael J. Martínez-Durá[2] & I. García-Fernández[1]

[1]CoMMLab, ETSE-UV, Valencia, Spain
[2]LSyM, IRTIC-UV, Valencia, Spain

Abstract

Procedural modelling of realistic environments that include elements derived from human activity can largely reduce production cost in animation, video-games and feature films. We address the problem of placing roads and other human-made elements, such as buildings, in a way that is consistent with the scene relief. Our approach is based on the calculation of so called desire paths, by means of the generation of many optimal paths according to different cost or distance functions. Using this idea, we rely on the information of the terrain properties to emulate the exploration of the scenario by a large number of pedestrians. From the routes generated by the pedestrians, we define walkability and habitability maps on the scene that can be later used to locate roads or buildings.

CCS Concepts

•***Computing methodologies*** → ***Shape modelling; Procedural animation; Mesh geometry models;***

1. Introduction

Generation of complex scenarios can be a really time consuming task for artists. As a consequence, procedural modelling of scenes has been a relevant topic in animation and computer games and thus has been growing steadily and broadening its range from the mere generation of the terrain relief to the distribution of vegetation and road networks or the modelling of buildings and cities [MKM89, Mus93, EM03, DL06, KM06, SDKT*09, Pat12, PL12, AAC*17, FBP17, TPC*19, MHS*19].

In this context, some authors have used the scene relief to decide how to distribute vegetation across the generated landscape [DHL*98, Ham01]. However, to the best of our knowledge the proposals in the bibliography to relate the relief with the location of human-made structures are scarce and based mainly on user-defined landmarks [KM07].

In this paper we address the automatic placement of roads and buildings based on the geometric properties of the base height map in a scene. Our main contribution is a method to generate walkability and habitability maps from the relief. In order to do this, we emulate the appearance of *desire paths* on the scene. Desire paths are defined as paths that appear as a result of the erosion generated by humans or animals, and tend to reflect the shortest paths between points in some sense. We use path planning algorithms to generate a series of minimum cost paths that link two points using different families of heuristic functions. From the set of generated routes, the cells in the map are labelled using the number of nearby passing paths.

1.1. Overview of the method

Given a particular scene, our goal is to build a map that describes how likely is that the different locations host roads and other human like structures, such as buildings. We start from a height field for the relief of the scene on a regular rectangular grid.

We first build a series of paths across the landscape that are plausible human paths according to different criteria. This step is done by means of the A^* algorithm and a set of heuristics that try to reflect different pedestrians profiles, as described in Section 3.1.

Using the information of the generated paths and the local geometric properties of the terrain surface we mark each cell with a value that expresses how convenient we consider that roads or buildings are placed in that location. This second step is done separately for roads, generating a walkability map, defined in Section 4.1, and buildings, generating a habitability map as described in Section 4.2. The results of the proposed method are presented and discussed in Section 5.

2. Related work

Generation of complex 3D scenes, such as the ones used in videogames, computer animation or VFX, is often done by means of procedural techniques, since it is much faster than manual modelling [EM03]. Many techniques exist that range from mere terrain generation [Mus93, SDKT*09] to more complex scenes including vegetation [DHL*98, Ham01, AAC*17, TPC*19, MHS*19] and roads or cities [KM06]. In our work, we will focus on the location of human-made elements, assuming a pre-built elevation map.

DOI: 10.2312/ceig.20191199

Extensive work has been done to generate urban maps, that are later populated with buildings. Sun et al. [SYBG02] reconstruct the road network of a specific city by identifying underlying patterns of the network. Other methods rely on multiple agents that build road networks controlled by specific constraints [LWW*07]. In this work, they consider information about land use at different locations to determine city growth. However, these methods only use minimal prevention of steep, non-practicable roads, and barely use the underlying height-map to influence the road network generated. Parish & Müller [PM01] build a city street map by defining a set of rules that take into account elevation to decide street size and intersection angle. As in these works, we intent to generate road networks, but, in our case, on an interurban landscape.

At a different scale, different authors have also used pedestrian dynamics to determine walking paths in different situations. Agethen et al. [AGF*18] use probabilistic methods to reproduce walk paths on an automotive assembly line. In [KKS18] isochrone maps are created to evaluate the walking distance between zones with Open Street Map. Our proposal has in common with these methods that they are based on the compilation of the information from many pedestrians. However, these methods only take into account the information about obstacles and not height variations.

In our method we will use algorithms for path search in graphs that will take into account the local properties of the landscape to prevent the usage of unrealistic locations. Like in [AGF*18] we will consider a set of walk paths to identify most walkable and habitable regions.

3. Generation of paths

Throughout all this work we will denote by v all the vertices that define the height-map, with $v = (x, y, z)$, being x the width component, y the height component and z the depth component. Furthermore, when we talk about distances, we will denote by v_s the starting vertex and by v_e the ending vertex, and α_{se} the elevation angle between those two vertices. Note that in this case we have:

$$\alpha_{se} = \arctan\left[\frac{(y_e - y_s)}{\sqrt{(x_e - x_s)^2 + (z_e - z_s)^2}}\right]$$

We will also consider $D(v)$ the *difficulty* of the vertex, which we will define as a measure how difficult it is to walk through a given vertex. We take $D(v)$ to be the variance in height of the neighbouring vertices, so if we consider $N(v)$ to be the set of neighbours of v, we define $D(v)$ as:

$$D(v) = \frac{1}{|N(v)|} \sum_{n \in N(v)} (y_n - y_v)^2$$

Note that this definition of difficulty will give values near 0 when the vertex and its neighbours form a nearly horizontal plane. Lying on a plane that is not horizontal, like on a hillside, the steeper the slope, the less practicable the place will be considered. This measure will also detect regions with a rough profile. This measure is not perfect, as zones like valleys can have a large difficulty value, but we can compensate this with a finer mesh.

3.1. Distances

We will obtain the paths for the experiments by using $A*$ with different distances between points to obtain the path of minimal distance with each of those distances. We will then compare each of them to determine if any of the distances provides a reasonable path.

Each distance will try to replicate a different trait of human travel, so as to replicate the real paths that pedestrians would use.

3.1.1. Weighted height distance

Pedestrians will often prefer routes with smaller changes in elevation even in the case that the routes are longer, so we define the *weighted height distance* as:

$$d(v_s, v_e) = \sqrt{(x_e - x_s)^2 + k^2(y_e - y_s)^2 + (z_e - z_s)^2}$$

Note that this distance is simply the norm of the vector $v_e - v_s$ but with the height component weighted by a factor $k \geq 1$. Also note that $k = 1$ represents the usual euclidean distance.

3.1.2. Slope distance

Pedestrians will often prefer smoother slopes, so we define the *slope distance* as:

$$d(v_s, v_e) = (1 + |\alpha_{se}|)\sqrt{(x_e - x_s)^2 + (z_e - z_s)^2}$$

In this case we only take the horizontal distance, and we add an additional cost due to the effort of the slope.

3.1.3. Terrain difficulty distance

Pedestrians will often prefer routes that avoid walking along a steep slope and rather walk along flatter paths, that are perpendicular to the uphill direction. Then we define the *difficulty distance* as:

$$d(v_s, v_e) = (1 + p)\sqrt{(x_e - x_s)^2 + (y_e - y_s)^2 + (z_e - z_s)^2}$$

The factor p is the average difficulty of the zone between the vertices v_s and v_e, defined by:

$$p = \frac{D(v_s) + D(v_e)}{2}$$

Note that this *difficulty distance* is equivalent to the euclidean distance but considering an added cost whenever the zone that we are traversing is steep.

3.1.4. Geographic model distance

In [Tob93], geographic dispersion models are analysed and discussed. Among them, a model of pedestrians velocity when walking on footpaths in hilly terrain is presented. According to this model, a usual estimation of the velocity of a pedestrian when walking on a slope is given by:

$$v = 6e^{-3.5|\tan\alpha + 0.05|}.$$

If we consider the distance to be equal to the time that it takes to traverse from one vertex to the other, then we can define the *realistic distance* as:

$$d(v_s, v_e) = \frac{\sqrt{(x_e - x_s)^2 + (y_e - y_s)^2 + (z_e - z_s)^2}}{6e^{-3.5|\tan\alpha + 0.05|}}$$

4. Generation of walkability and habitability maps

Next we define the concepts of *walkability* and *habitability* for a vertex. With these definitions, a map can be built to determine regions in which the location of roads or buildings is desirable.

4.1. Walkability maps

Our objective is to study and identify the zones with more transit, so we will generate multiple paths between random points over the whole height-map. For each vertex, we will measure how many of those paths traverse through the given vertex. We will use the square of the normalised value for each cell, so that vertices with a higher number of passes have more presence and those with fewer passes do not stand out as much. We can express this as:

$$W(v) = \left(\frac{\text{\# of paths traversing this vertex}}{\text{max \# of paths traversing a vertex}} \right)^2$$

However, this measure alone won't properly determine areas of interest, as it is to be expected that zones with a lot of vertices each with high walkability would be considered also walkable, even if the vertices itself have fewer paths traversing over them, so we will consider $S(v)$ to be the *smoothed walkability*, obtained by means of a weighted average over its neighbours, with the weights being the inverse of the distance. Also note that due to how A* works, even small changes in distance will be discarded by A*, so this smoothing and a large number of paths will also be useful to reproduce that variability.

4.2. Habitability maps

Our next objective is to study and determine zones with greater habitability, that is, zones with more potential to start a settlement. Obviously this depends on a lot of factors such as natural resources, but we will only consider in this work the factors that we have already established, that is, the walkability and the difficulty of the terrain. We propose the following measure to quantify habitability:

$$H(v) = \left(1 - \frac{D(v)}{\max_i D(v_i)}\right)(1 + S(v)).$$

With this measure, zones with greater difficulty $D(v)$ (that is, steeper or rougher zones) will get low habitability, and zones with greater walkability will get their habitability increased. The idea of this measure is to mark as more habitable the cells that often belong to a path and have low difficulty value.

5. Results

For the following experiments we have considered a height-map on a 257×257 regular grid, generated using the diamond-square algorithm [FFC82]. The total size of the landscape represents a $2570m \times 2570m$ region, so each square of the grid has a side of $10m$. The examples are shown for height-map with a big mountain in the middle, in order to better reflect our results. The mountain height is approximately $1100m$. A greyscale representation of the height-map, together with a view of the resulting terrain, can be found on Figure 1. We used as neighbours of each vertex the vertices that were no more than 50 meters apart from it, with the euclidean distance.

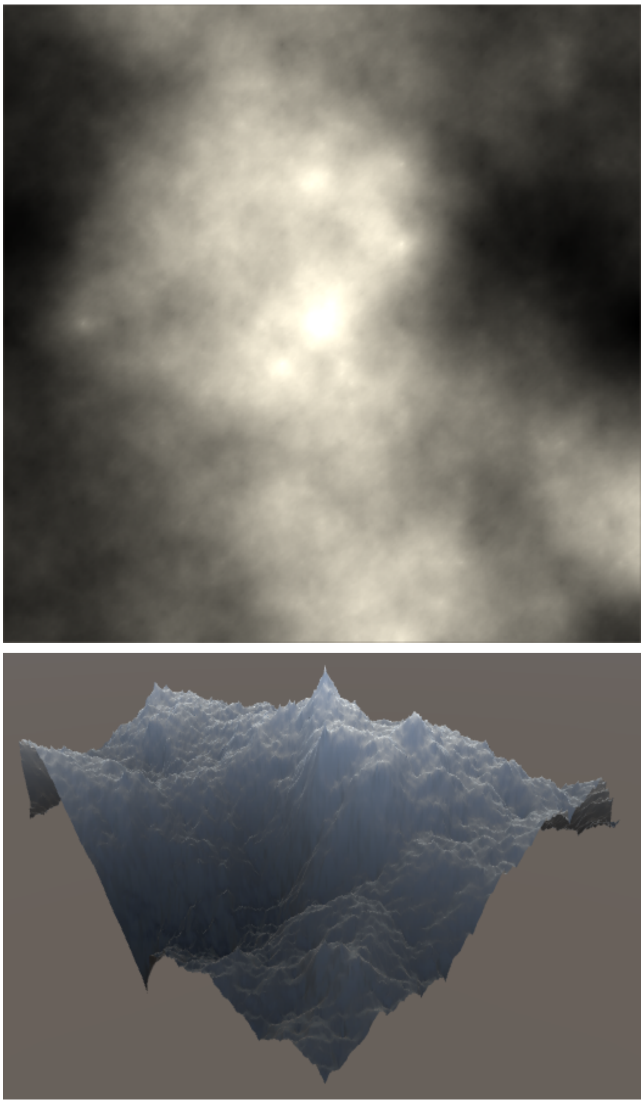

Figure 1: *Height-map of the terrain in greyscale (top) and the corresponding terrain (bottom). Lower areas are darker, while higher areas are brighter.*

5.1. Effect of distance functions

The results are shown for a particular pair of start and end vertices, generated randomly at the upper-right (North-East) and lower-left (South-West) sides of the map. The path of minimal distance is computed with every distance function defined in 3.1 and with different values for their parameters. We can compare the result on Figure 2.

If we observe the weighted height distance, we can observe a path with a lot of small twists, like on a mountain pass, preventing going near the peak of the mountain. This is somewhat similar to what we observe with the slope distance, where the twists have been reduced but at the same time, a seemingly longer path is taken to prevent great changes in angle. A similar effect can be observed

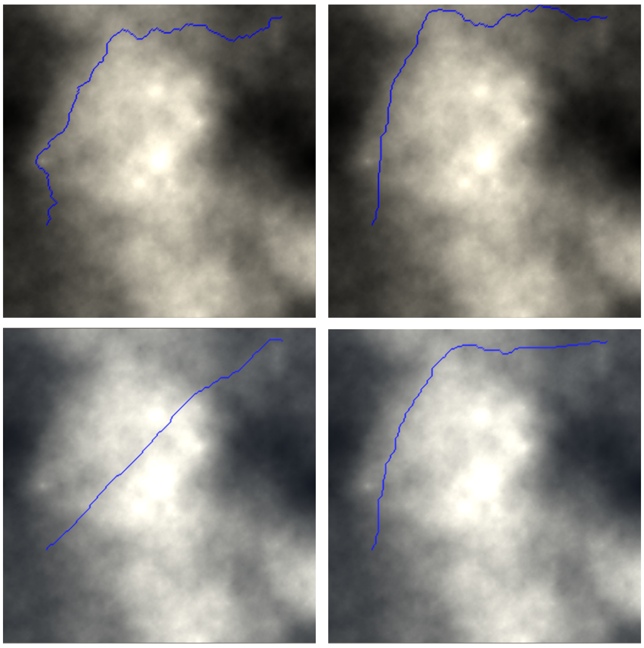

Figure 2: *In blue, the minimal distance path for each of the distances, displayed over the height-map in greyscale. a) Weighted height distance with $k = 10$. b) Slope distance. c) Difficulty distance. d) Realistic distance.*

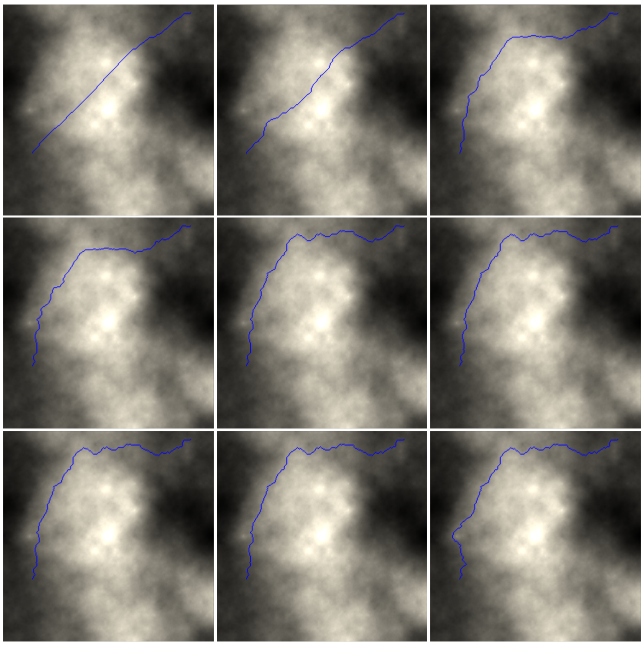

Figure 3: *In blue, the minimal distance path for the weighted height distance with different values of k. From top to bottom, left to right, k goes from 1 to 9.*

when using the realistic distance, which almost replicates the path observed with the weighted height distance.

However, the difficulty distance does not provide the expected results, since the paths traverse the map near the peak of the mountain: it seems that the additional cost of travelling by zones with less difficulty outweigh the additional height, so it seems that this distance does not reflect the feasibility of paths properly.

We will have compared the weighted height distance with more values of k. The obtained paths can be compared on Figure 3.

We can see clearly how a bigger weight on the height component (that is, a greater k) introduces the small twists on the path, which we think that is desirable in order to obtain paths that seem human-made.

5.2. Walkability maps

In order to build the walkability map we have generated 1000 paths with random start and end vertices for the weighted height distance, which has provided the best results in the previous comparison, taking again $k = 10$. We can observe the walkability map in Figure 4, and its smoothed version on Figure 5.

Figure 4: *Walkability ($W(v)$) map for the weighted height distance with $k = 10$, obtained from 1000 random paths. Vertices with more walkability in white.*

We can observe how the paths seemingly circle around the mountain as we could expect, preventing paths near the peak of the mountain. We can see zones in the higher and the lower part where a lot of paths intersect, which are clearly determined as with high walkability.

We can also observe how zones that don't appear clearly on the walkability map are more visible on the smoothed walkability map, meaning that vertices that have almost no paths passing through them have a lot of traffic near them, so we find that this smoothing really helps to reflect the real traffic on the zone.

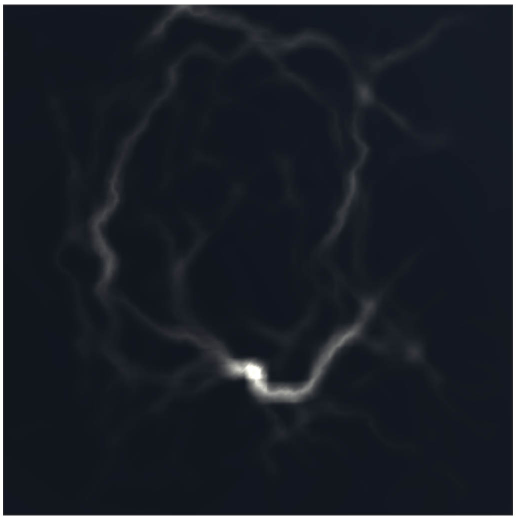

Figure 5: *Smoothed walkability ($S(v)$) map for the weighted height distance with $k = 10$, obtained from 1000 random paths. Zones with more walkability in white.*

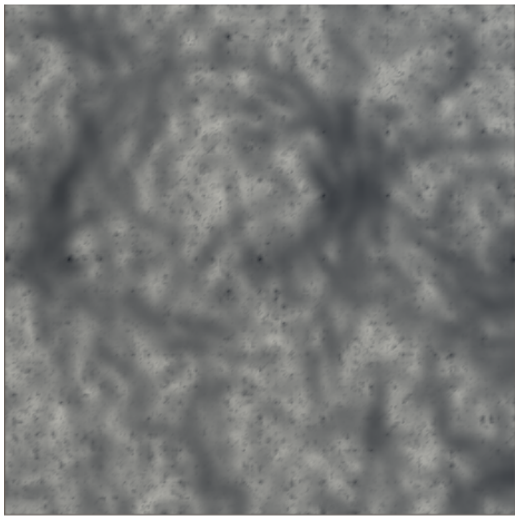

Figure 6: *Habitability map with no walkability ($W(v) = S(v) = 0$). Black means lower habitability, while white means more habitability.*

5.3. Habitability maps

First we have obtained the habitability map with no paths, that is, considering that $W(v) = S(v) = 0$, which can be seen in Figure 6. In this case, habitability results from the roughness and slope of the underlying terrain. Note that there are some small zones that appear completely black: those zones correspond to high frequency noise generated by the diamond-square algorithm, and could be prevented by using erosion techniques.

In addition, we have also obtained the habitability map considering the generated paths and the walkability map computed in Section 5.2. The resulting habitability map can be seen in Figure 7. We observe how two brighter (more habitable) zones appear right above (North) and below (South) the centre of the map, which correspond to the mountain peak. We can also observe how the lower left zone and the zone slightly to the right of the peak also reflect the effect of walkability on the habitability. In this map we also can see how the less habitable zones, corresponding to black spots, are inherited directly from the map in Figure 6 due to very bad geometric conditions for settlements.

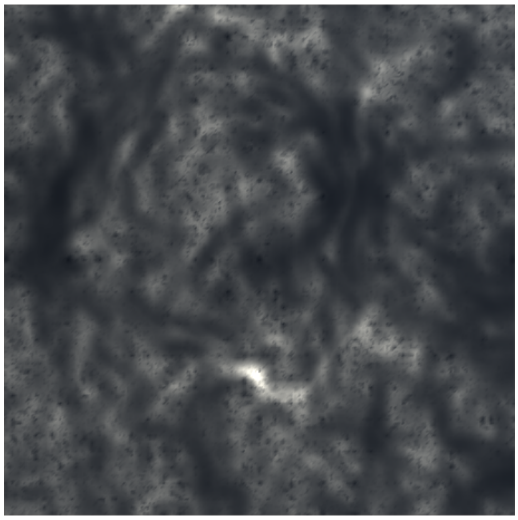

Figure 7: *Habitability map with 1000 paths generated. Black means lower habitability, while white means more habitability.*

6. Road and settlement generation

We have not generated sample roads or buildings so far. However, the results obtained in form of walkability maps could be clearly used as a base for laying roads and establishing a road network. Moreover, these walkability maps could be used as a basis to produce anthropogenic erosion. Walkability maps could also be used when designing parks or similar projects, in order to trace the paved paths in the regions where desire oaths would later be formed.

As we said when defining our habitability measure, it should reflect zones with more potential to start a successful settlement, so the next step would be to identify those zones with greater habitability and generate settlements on those zones. There are various ways of generating this settlements. For example, we could mark those zones and feed them to other algorithms, which use interest points to generate roadways and cities. Another way would be to expand this system with another improvement, added to the one previously mentioned on walkability. The idea would be using interest points to condition in some way the paths that we have obtained, favouring paths that go near interest points as settlements or geographically interesting points as vantage points or mountain peaks.

7. Conclusion and future work

We have proposed a method to generate maps that determine zones appropriate to set roads and human settlements, based on the underlying relief map. However, the results have been presented for a single landscape. The system has to be tested on different terrains, with variable mountain ranges, valleys or islands. We could also use real terrain (e.g. extracted from on-line databases) to compare if the results obtained by our method are compatible with real human constructions.

We have observed that some of the distances lack the traits expressed on others (for example, difficulty distance lacks the aversion for height displacements). We need to explore the combination of those traits on a new distance function that is more versatile.

The proposed approach opens a series of research lines to improve the results and to enhance its usability. The geometry of the resulting paths is not evaluated in any manner. A possible improvement could be to use the geometry of the path when obtaining the paths of minimal distance. For example, we could add restrictions as smooth turns to prevent twisty roads that would not be transitable by vehicles in real life. Both the erosion step and the generation of points of interest can be introduced in a feed-forward loop that would lead to new paths (probably reinforcing existing ones) that redefine walkability and habitability. This approach would enable not only the generation of a landscape, but also the animation of its evolution.

References

[AAC*17] Argudo O., Andujar C., Chica A., Guérin E., Digne J., Peytavie A., Galin E.: Coherent multi-layer landscape synthesis. *The Visual Computer 33*, 6-8 (2017), 1005–1015. 1

[AGF*18] Agethen P., Gaisbauer F., Froehlich P., Manns M., Rukzio E.: Towards realistic walk path simulation in automotive assembly lines: A probabilistic approach. *Procedia CIRP 67* (01 2018), 464–469. doi:10.1016/j.procir.2017.12.243. 2

[DHL*98] Deussen O., Hanrahan P., Lintermann B., Měch R., Pharr M., Prusinkiewicz P.: Realistic modeling and rendering of plant ecosystems. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 275–286. URL: http://doi.acm.org/10.1145/280814.280898, doi:10.1145/280814.280898. 1

[DL06] Deussen O., Lintermann B.: *Digital design of nature: computer generated plants and organics*. Springer Science & Business Media, 2006. 1

[EM03] Ebert D. S., Musgrave F. K.: *Texturing & modeling: a procedural approach*. Morgan Kaufmann, 2003. 1

[FBP17] Fita J. L., Besuievsky G., Patow G.: A perspective on procedural modeling based on structural analysis. *Virtual Archaeology Review, 2017, vol. 8, núm. 16, p. 44-50* (2017). 1

[FFC82] Fournier A., Fussell D., Carpenter L.: Computer rendering of stochastic models. *Commun. ACM 25*, 6 (June 1982), 371–384. URL: http://doi.acm.org/10.1145/358523.358553, doi:10.1145/358523.358553. 3

[Ham01] Hammes J.: Modeling of ecosystems as a data source for real-time terrain rendering. In *Proceedings of the First International Symposium on Digital Earth Moving* (London, UK, UK, 2001), DEM '01, Springer-Verlag, pp. 98–111. URL: http://dl.acm.org/citation.cfm?id=647811.738117. 1

[KKS18] Khodnenko I., Kudinov S., Smirnov E.: Walking distance estimation using multi-agent simulation of pedestrian flows. *Procedia Computer Science 136* (2018), 489 – 498. 7th International Young Scientists Conference on Computational Science, YSC2018, 02-06 July2018, Heraklion, Greece. URL: http://www.sciencedirect.com/science/article/pii/S1877050918315618, doi:https://doi.org/10.1016/j.procs.2018.08.256. 2

[KM06] Kelly G., McCabe H.: A survey of procedural techniques for city generation. *Institute of Technology Blanchardstown Journal 14* (01 2006). 1

[KM07] Kelly G., McCabe H.: Citygen: An interactive system for procedural city generation. In *Fifth International Conference on Game Design and Technology* (2007), pp. 8–16. 1

[LWW*07] Lechner T., Watson B., Wilensky U., Tisue S., Felsen M., Moddrell A., Ren P., Brozefsky C.: *Procedural modeling of urban land use*. Tech. rep., North Carolina State University. Dept. of Computer Science, 2007. 2

[MHS*19] Makowski M., Hädrich T., Scheffczyk J., Michels D. L., Pirk S., Pałubicki W.: Synthetic silviculture: Multi-scale modeling of plant ecosystems. In *Proc. of SIGGRAPH2019* (2019). 1

[MKM89] Musgrave F. K., Kolb C. E., Mace R. S.: The synthesis and rendering of eroded fractal terrains. *SIGGRAPH Comput. Graph. 23*, 3 (July 1989), 41–50. URL: http://doi.acm.org/10.1145/74334.74337, doi:10.1145/74334.74337. 1

[Mus93] Musgrave F. K.: *Methods for realistic landscape imaging*. PhD thesis, Yale University, New Haven, CT, 1993. 1

[Pat12] Patow G.: User-friendly graph editing for procedural modeling of buildings. *IEEE Computer Graphics and Applications 32*, 2 (March 2012), 66–75. doi:10.1109/MCG.2010.104. 1

[PL12] Prusinkiewicz P., Lindenmayer A.: *The algorithmic beauty of plants*. Springer Science & Business Media, 2012. 1

[PM01] Parish Y. I., Müller P.: Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 301–308. 2

[SDKT*09] Smelik R. M., De Kraker K. J., Tutenel T., Bidarra R., Groenewegen S. A.: A survey of procedural methods for terrain modelling. In *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)* (2009), pp. 25–34. 1

[SYBG02] Sun J., Yu X., Baciu G., Green M.: Template-based generation of road networks for virtual city modeling. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2002), VRST '02, ACM, pp. 33–40. URL: http://doi.acm.org/10.1145/585740.585747, doi:10.1145/585740.585747. 2

[Tob93] Tobler W.: Three presentations on geographical analysis and modeling: Non- isotropic geographic modeling; speculations on the geometry of geography; and global spatial analysis (93-1). *UC Santa Barbara: National Center for Geographic Information and Analysis* (1993). URL: https://escholarship.org/uc/item/05r820mz. 2

[TPC*19] Tang L., Peng X., Chen C., Huang H., Lin D.: Three-dimensional forest growth simulation in virtual geographic environments. *Earth Science Informatics 12*, 1 (2019), 31–41. doi:10.1007/s12145-018-0356-4. 1