Generating Fractal Tiles using Voronoi Diagrams

Tomasz Dobrowolski
Gdańsk University of Technology, Poland
Department of Algorithms and System Modelling

E-mail: `tomkh@sphere.pl`

Abstract

This paper introduces a fractal partition class defined over a unit wrapped space. The cells of a fractal partition have self-similarity property and can be used as a set of tiles that can tile R^n space periodically or quasi-periodically with non-uniform tiling density. An algorithm for generating set of fractal cells using Voronoi diagram computation is proposed with several applications in computer graphics and computational chemistry.

1. Introduction

A fractal partition is constructed using a sequence of Voronoi partitions over a unit wrapped space (i.e. wrapped on a torus surface). A natural number called color is assigned to every Voronoi cell. Fractal iteration process generates more and more dense Voronoi partitions. A fractal cell is defined as a union of Voronoi cells with the same color. A detailed fractal partition definition is proposed in the following sections.

2 Basic definitions

Symbol N denotes the set of natural number: $N = \{1, 2, 3, ...\}$, also called a set of colors. Z denotes the set of integers. R denotes the set of real numbers. $N \subset Z \subset R$. Other sets are constructed using common set-builder notation. Vector of p elements is denoted as $p = (p_1, p_2, ..., p_n)$, where $n = |p|$ is the length of p. A unit segment is denoted as $I = \{x : x \in R \wedge 0 \leq x < 1\}$. A metric space (D^n, δ) is a linear space over D^n set with a distance function $\delta : D^n \times D^n \rightarrow R$, that satisfy the following conditions: $\delta(p, q) \geq 0 \wedge \delta(p, p) = 0 \wedge \delta(p, q) = \delta(q, p) \wedge \delta(p, q) + \delta(q, r) \geq \delta(p, r)$ for any $p, q, r \in D^n$. An Euclidean space (R^n, δ) is a metric space with a distance function $\delta : R^n \times R^n \rightarrow R$, $\delta(p, q) = \delta(q, p) = \sqrt{\sum_{1 \leq i \leq n} (p_i - q_i)^2}$. A unit wrapped (I^n, δ) space is a metric space with a distance function $\delta : I^n \times I^n \rightarrow R$, $\delta(p, q) = \delta(q, p) = \sqrt{\sum_{1 \leq i \leq n} \min{(1 - |p_i - q_i|, |p_i - q_i|)})^2}$, where $|x| = \max{(-x, x)}$ is an absolute value of a real number $x \in R$.

A *color Voronoi partition* $V(P, \kappa)$ of (D^n, δ) space is generated by a countable set of points $P \subseteq \{p(i) | i \in N \wedge p(i) \in D^n\}$, called *generator points* or *generators*, and a *coloring function* $\kappa : P \rightarrow N$, where $\kappa(p)$ is a color of a point $p \in P$. A set of cells $V = \{V_k | k \in N \wedge V_k \subset D^n\}$ is constructed as follows:

- A *Voronoi cell* generated by $p \in P$ is a set of points in D^n closest to p; $Cell(p) = \{s | s \in D^n \wedge \forall_{q \in P} \delta(s, p) \leq \delta(s, q)\}$.
- A wall of a Voronoi cell $Cell(p)$ is $\bigcup_{q \in P \wedge q \neq p} Cell(p) \cap Cell(q)$.
- A subset $P_k \subset P$ of generator points with color k is $P_k = \{p | p \in P \wedge \kappa(p) = k\}$.
- A cell with color k is $V_k = \bigcup_{p \in P_k} Cell(p)$.
- A wall of a cell V_k is $\bigcup_{m \in N \wedge m \neq k} V_k \cap V_m$.

Note that if no points in P have color k, V_k is an empty set. A neighborhood graph of cells in V is $Graph(V) = (C, E)$, where C is a set of vertices that represent possible cell colors $C = \{c | c \in N \wedge \exists_{p \in P} \kappa(p) = c\}$ and E is a set of edges: $\{j, k\} \in E \iff V_j \cap V_k \neq \emptyset$.

3 Construction of a fractal partition

Let us start with a finite set of m different points $P^0 = \{p(1), p(2), ..., p(m)\} \subset I^n$ and a color Voronoi partition $V^0(P^0, \kappa^0)$ over a unit wrapped (I^n, δ) space, where $\forall_{p(i) \in P^0} \kappa^0(p(i)) = i$, so every point in P^0 has unique color.

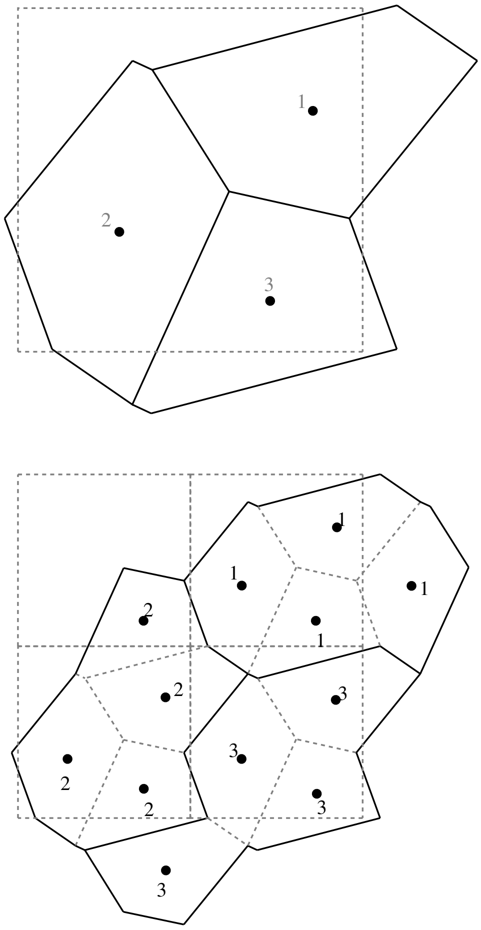

Figure 1. An exemplary set of random points P^0 in I^2, $m = 3$, with denoted cells of color Voronoi partition V^0 and V^1 for $d = 2$. Additional walls of Voronoi cells in V^1 are denoted as dotted lines and colors of generators P^0 and P^1.

We construct a sequence of Voronoi partitions $V^h(P^h, \kappa^h)$, where $P^h = \{(p_1 + \frac{a_1}{d}, p_2 + \frac{a_2}{d}, ..., p_n + \frac{a_n}{d}) | p \in P^{h-1} \wedge a \in \{0, 1, ..., d-1\}^n\}$, and $\kappa^h(p \in P^h) = \min\{\kappa^{h-1}(q) : q \in P^{h-1} \wedge \forall_{s \in P^{h-1}} \delta(q,p) \leq \delta(q,s)\}$, a color of every $p \in P^h$ is the color of the closest point to p in P^{h-1} with a lowest color (if more than one point in P^{h-1} has equal closest distance to p).
Note that if P^0 has m points, P^1 has $d^n m$ points, $|P^2| = d^n d^n m = (d^n)^2 m$ and $|P^h| = (d^n)^h m$ points.

This sequence determines a limit partition $T(n, d, m, P^0) = V^\infty$, a set of m fractal cells $T(n, d, m, P^0) = \{V_k^\infty | 1 \leq k \leq m\}$, where $V_k^\infty = \{p | \exists_H \forall_{h > H} p \in V_k^h\}$ is a limit set of cells with color k in successive V^h iterations. T(n, d, m, P^0) is called a *fractal partition* with lacunarity d generated by a set of m points P^0 over a unit wrapped (I^n, δ) space.

It is interesting that for a given $n, d, m \in N$ there are finitely many $T(n, d, m, P^0)$ sets possible.
Lets call $\Psi(n, d, m)$ a number of all possible different $T(n, d, m, P^0)$ sets that can be constructed for a given $n, d, m \in N$ from any $P^0 \subset I^n, |P^0| = m$. The question what is the value of $\Psi(n, d, m)$ in the general case seems open. Obviously $\Psi(n, d, 1) = 1$ for all n, d, because if $m = 1$, there is only one cell in every iteration and a limit set. Also it is simple to check that $\Psi(1, 2, 2) = 1$.

If we put some constraints over the P^0 set, we can generate fractal partitions whose cells are connected set of points on (I^n, δ) space wrapped over a torus surface. The simple constraint is $Graph(V^0) = Graph(V^1)$, that guarantees that all cells will be connected sets. It is more difficult to find constraints that guarantee that at least one cell will be a disconnected set.

4 Generation algorithm

In order to generate and visualize an approximation of the fractal partition $T(n, d, m, P^0)$, we can precalculate a point substitution rule that maps every point in P^0 to a set of unwrapped points from P^1, as shown in Figure 2. The same substitution rules can be applied to generate P^k from P^{k-1}. Fractal cells in $V^k(P^k, \kappa^k)$ for sufficiently big k gives a good approximation of the fractal partition V^∞ (Fig. 2).

Algorithm for constructing all substitution rules $S(p(i)) \subset R^n \times N$ for every point $p(i) \in P^0$ is as follows:

1. Start with $S(p(i))$ empty for every $p(i) \in P^0$.
2. For every $a \in \{0, 1, ..., d-1\}^n$ and every $p(i) \in P^0$:
   (a) let $q = (p(i)_1 + \frac{a_1}{d}, p(i)_2 + \frac{a_2}{d}, ..., p(i)_n + \frac{a_n}{d}) \in P^1$,
   (b) find the closest point $r \in P^0$ to q using wrapped distance δ,
   that satisfies $\forall_{s \in P^0} \delta(q,r) < \delta(q,s) \vee \delta(q,r) = \delta(q,s) \wedge \kappa(r) \leq \kappa(s)$,
   (c) calculate unwrapped difference between q and r as $u = (u_1, u_2, ..., u_n)$,
   where $u_j = q_j - r_j + \lfloor r_j - q_j + \frac{1}{2} \rfloor$ for $1 \leq j \leq n$,
   (d) add (u, i) to $S(r)$.

Algorithm for generating unwrapped $V^k(P^k, \kappa^k)$ for a given lacunarity d, where the rule index c is the color of the point $p(c) \in P^0$:

1. Start with P^k empty.

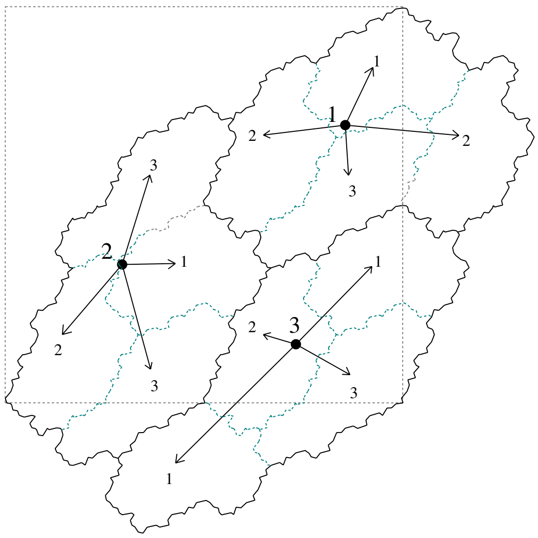

Figure 2. A set of 3 **fractal cells** $T(2, 2, 3, P^0 = \{(.86, .3), (.3, .65), (.73, .85)\})$**. More iterations gives better self-similarity (i.e. a cell** V_1^∞ **is an union of four appropriately translated and scaled cells:** $V_1^\infty, V_2^\infty, V_2^\infty, V_3^\infty$**). For every point in** P^0 **a substitution rule is denoted by arrows.**

2. For each $p(c) \in P^0$:
   call *Sub*(0, k, c, c, $p(c)$) (declared below).

3. Calculate geometry of Voronoi diagram for P^k in R^n (Mark de Berg and others [3]). Extract surface of fractal cells of $V^k(P^k, \kappa^k)$ made of faces that lies between Voronoi cells with different color.

Procedure *Sub*(i = iteration level, k = max. iterations, s = starting color, c = rule index, p = point):
  if ($i < k$)
    for each $(u, c') \in S(p(c))$:
      call *Sub*($i + 1$, k, s, c', $p + \frac{u}{d^i}$)
  else
    add p to P^k and assign color s to p, $\kappa^k(p) = s$.

4.1 Algorithm complexity analysis

We know that $|P^k| = (d^n)^k m$, where d^n is constant (d is lacunarity and n dimension of R^n space) and m is number of points in P^0. Every recursion level i require $|P^i|$ steps, thus the algorithm of generating every point in P^k require $|P^0| + |P^1| + ... + |P^k|$ steps. We can derive easily that $(d^n)m + (d^n)^2 m + ... + (d^n)^k m = \frac{(d^n)^{k+1} - 1}{d^n - 1} m = ((d^n)^k * (d^n) - 1)\frac{m}{d^n - 1} = (d^n)^k * C - D$, therefore the algorithm require $\Theta((d^n)^k)$ steps to generate P^k set. P^k set is used as an input to Voronoi diagram computation algorithm, thus for $n = 2$ we can use algorithm with linear complexity (Fortune [6]) to extract fractal cell surface in $O(|P^k|) = O((d^2)^k)$ steps.

The presented algorithm is far from optimal for extracting only a surface of V^k fractal cells. Note that Voronoi diagram of P^k contains only m different cell shapes, because P^k set is periodic tiling of P^0. Therefore we can calculate Voronoi diagram V^0 and calculate a periodic tiling of it after every recursive substitution. A good optimization trick is to ignore all Voronoi cells and their generator points that are surrounded by Voronoi cells with the same color during recursive substitutions as well as when calculating final geometry of a fractal cell surface. An optimal algorithm would require number of steps proportional to number of faces in the output fractal cell surface (or number of Voronoi cells for which one of the neighboring cell have different color).

5 Final remarks and applications

We proposed a class of fractal partitions generalized to R^n space that has some interesting properties for potential applications. Fractal cells can be used as tiles to tile an R^n space periodically. The important property of such tiles is self-similarity of tile groups. Every tile is an union of a subset of tiles from the same set, translated and scaled down by a lacunarity d factor, so every tile can be replaced with a set of smaller tiles and space will be always fully partitioned by them. That gives possibility to quasi-periodical tilings with non-uniform density (multi-resolution).

The presented fractal partition generation algorithm can be used to model very rough surfaces in computer graphics applications. A single fractal cell can represent an irregular noisy rock. For greater lacunarity d fractal cell surface has lower frequency. Every fractal cell matches to each other tightly, thus single rocks can be arranged into a more complex surface with an arbitrary topology (i.e. to model an underwater cave, Dobrowolski [4]).

Another interesting potential application might be generation of artificial molecules that can dock in many different places. Figure 3 shows a set of spheres built over a 3rd iteration of substitution algorithm (center of spheres are points from P^3 set, starting from P^0 with $m = 4$ points, lacunarity was set to $d = 3$). Generated molecules might be used for testing a rigid protein docking algorithms (Cherfils, Janin [2]) or as a functional blocks in artificial chemistry simulations. Orientation of the molecules inscribed into each fractal cell could be randomized and given as an input to a docking algorithm. Then a proper docking algorithm would find the best matching configuration.

For artificial chemistry applications a chemical element type could be assigned to each generator point in P^0 (i.e.

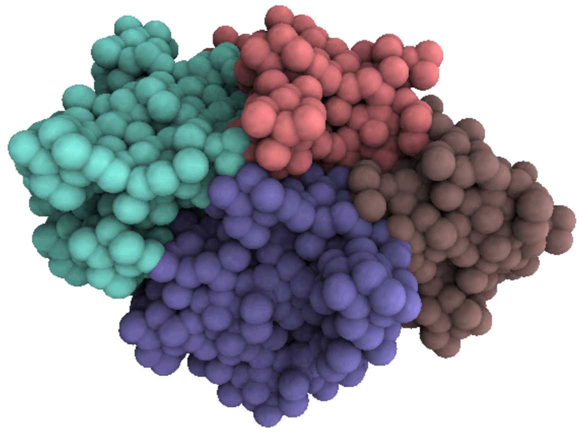

Figure 3. A set of artificial molecules that can dock in many different places, generated using a few iterations of fractal partition in R^3. Molecule centers are partition generator points.

Oxygen, Carbon, Silicon). A generated P^k set can represent a crystal structure periodically repeated on a cubic lattice.

References

[1] G. Cantor. On the power of perfect sets of points (de la puissance des ensembles parfait de points). *Acta Mathematica*, 4:381–392, 1884.

[2] J. Cherfils and J. Janin. Protein docking algorithms: simulating molecular recognition. *Current Opinion in Structural Biology*, 3:265–269, 1993.

[3] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications, 2nd edition*. Springer-Verlag, 2000.

[4] T. Dobrowolski. Suboceanic 64k. *http://ged.ax.pl/∼tomkh*, 2005.

[5] R. Engelking. *General Topology*. Taylor & Francis, 1977.

[6] S. Fortune. Voronoi diagrams and delaunay triangulations. *Handbook of Discrete and Computational Geometry*, pages 377–388, 2004.

[7] F. Preparata and M. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[8] N. M. Priebe. Towards a characterization of self-similar tilings via derived voronoi tessellations. *Geometriae Dedicata*, 79:239–265, 2000.

[9] B. Solomyak. Dynamics of self-similar tilings. *Ergodic Th. Dynam. Sys.*, 17:695–738, 1997.