MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Artistic Pattern Generation by a Model of Territory Competing

Kokichi Sugihara

METR 2007–27 April 2007

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Artistic Pattern Generation by a Model of Territory Competing

Kokichi Sugihara
Department of Mathematical Informatics,
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
sugihara@mist.i.u-tokyo.ac.jp

Abstract

This paper presents a method for generating artistic patterns using the Voronoi diagram. In this method, we start with a few of seed points, generate new sites at random, and assign them to one of the seed points based on the Voronoi diagram of the sites. The resulting territory patterns are colored by choosing paths in a color space. The patterns generated by this method have been adopted for calendars and magazine covers.

1. Introduction

Computer graphics has achieved great success in generating tremendous amount of beautiful and artistic pictures, which could have never been generated without computers [1]. They include photo-realistic pictures [7] such as water [5] and rain [11], and artistic pictures [13, 8] to mention a few. Tools for artists such as a virtual brush [12] have also been developed.

Among them, there is an interesting class of pictures generated by purely mathematical mechanism. A typical mechanism is “fractal”. The fractal is a concept representing self-similarity in a geometric and/or statistical sense. This concept was first coined by Mandelbrot [6] in 1982, and has been observed behind a wide range of natural phenomena such as shapes of lands, shapes of rivers, shapes of trees, etc [2, 10]. Because of this reason, the fractal has also been used in computer graphics widely in order to generate nature-like shapes by simple mechanisms [8].

Another mathematical mechanism that can generate beautiful pictures is “chaos”. From a computer graphics point of view, we are interested in complicated chaotic phenomena generated by simple procedures such as nonlinear dynamic systems [3]. Using such systems, we can generate complicated and beautiful pictures; typical examples are the pictures of Judia sets [3], and those of Mandelbrot sets [3, 4].

In this paper we will show that the concept of the Voronoi diagram [9] can also be used to generate artistic pictures; some examples of such pictures are shown in Fig. 1.

The outline of the procedure to generate those pictures is the following. First, we place a few of seed points in the plane, and define their territories by the Voronoi regions of the seed points. Next, we simulate territory competing phenomena by adding new points to the territories, and recomputing the shapes of the Voronoi regions. Finally, we assign gradually changing colors to the resulting Voronoi regions, and thus obtain the pictures.

The resulting pictures have been used in several contexts. Fig. 2 shows a calendar issued in the School of Information Science and Technology of the University of Tokyo, while Fig. 3 shows covers of mathematical magazine “Suugaku Seminar” published by Japanese publisher, Nippon-Hyoron-Sha, whose main readers include university students and high school teachers of mathematics.

The remaining part of the paper is organized as follows. The Voronoi diagram is defined in Section 2, and the territory competing procedure is constructed in Sections 3 and 4. Then a color assigning rule is presented in Section 5, and concluding remarks are given in Section 6.

2. Voronoi Diagrams

We start with the definition of the Voronoi diagram. For two points P and Q, we denote by $d(\mathrm{P},\mathrm{Q})$ the Euclidean distance between P and Q. Let $S = \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_n\}$ be the set of n points in the plane $\mathbf{R}^2$. For point P_i, we define region $R(S;\mathrm{P}_i)$ by

$$R(S;\mathrm{P}_i) = \bigcap_{j \neq i} \{\mathrm{P} \in \mathbf{R}^2 \mid d(\mathrm{P},\mathrm{P}_i) < d(\mathrm{P},\mathrm{P}_j)\}. \quad (1)$$

$R(S;\mathrm{P}_i)$ represents the set of all points that are nearer to P_i than to any other points of S. We call $R(S;\mathrm{P}_i)$ the *Voronoi region* of P_i.

The plane is partitioned into the Voronoi regions $R(S;\mathrm{P}_1), R(S;\mathrm{P}_2), \ldots, R(S;\mathrm{P}_n)$ and their boundaries.

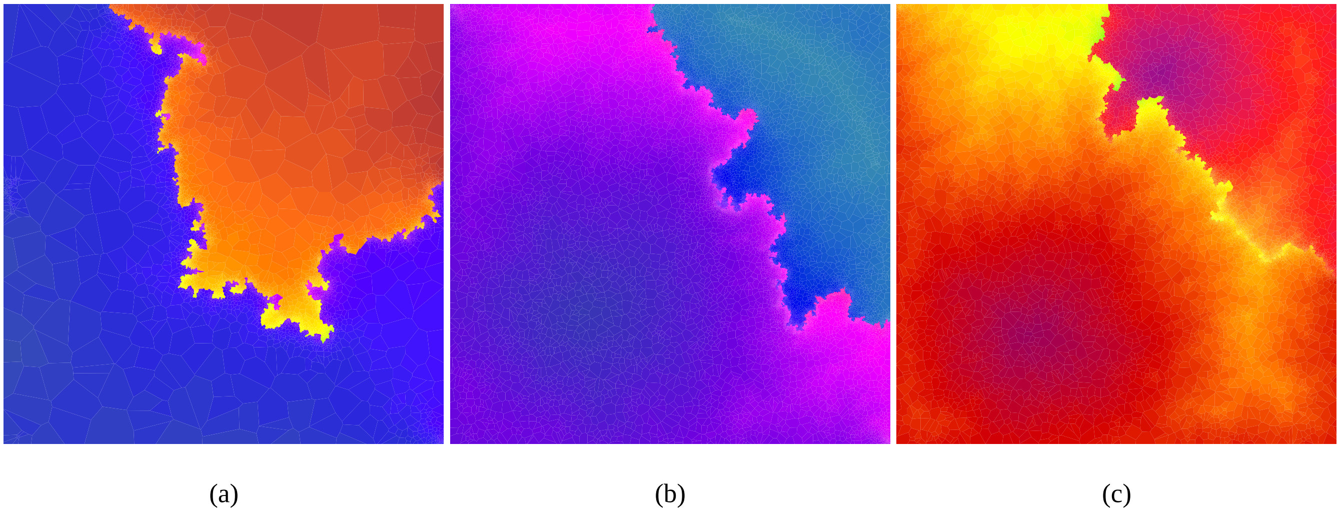

(a) (b) (c)

Figure 1. Examples of pictures generated by the territory-competing procedure.

Figure 2. Territory-competing picture used in a calendar.

We call this partition the *Voronoi diagram* for S. The elements of S are called *generators* of the Voronoi diagram.

Fig. 4(a) shows an example of the Voronoi diagram, where small black circles and small white circles represent the generators. In this figure two of the generators are represented by double circles; the reason of that will be explained in the next section.

The boundary shared by two Voronoi regions is part of the perpendicular bisector of the line segment connecting the associated two generators. Therefore, the Voronoi regions are convex polygons.

Since the Voronoi region $R(S; \mathrm{P}_i)$ represents the collection of points which are nearest to P_i among all the points in S. this region may be interpreted as the territory of P_i. Hence, the Voronoi diagram can be used as a model of a territory pattern.

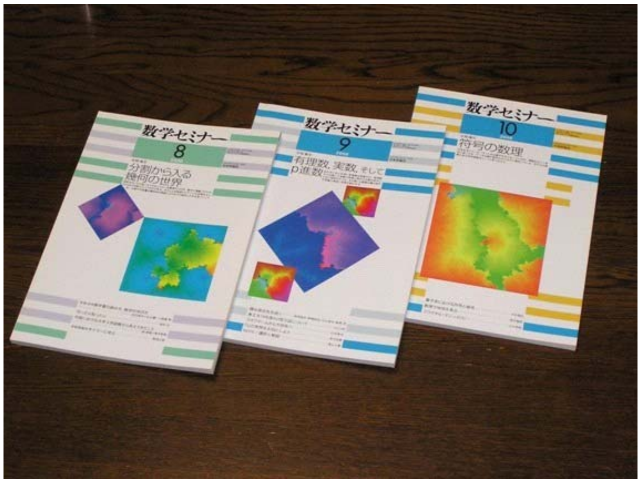


Figure 3. Territory-competing pictures used on magazine covers.

3. Territory Competing Procedure

Using the Voronoi diagram, we construct a procedure that simulates a territory-competing phenomenon. Here we describe the procedure for the case where there are exactly two territories.

First we place two points, say P_1 and P_2, in the plane and call them the *seed points*. In the example of Fig. 4(a), there are two points surrounded by circles, one is the black point surrounded by a circle (let this point be P_1), and the other is represented by the double white circles (let this point be P_2).

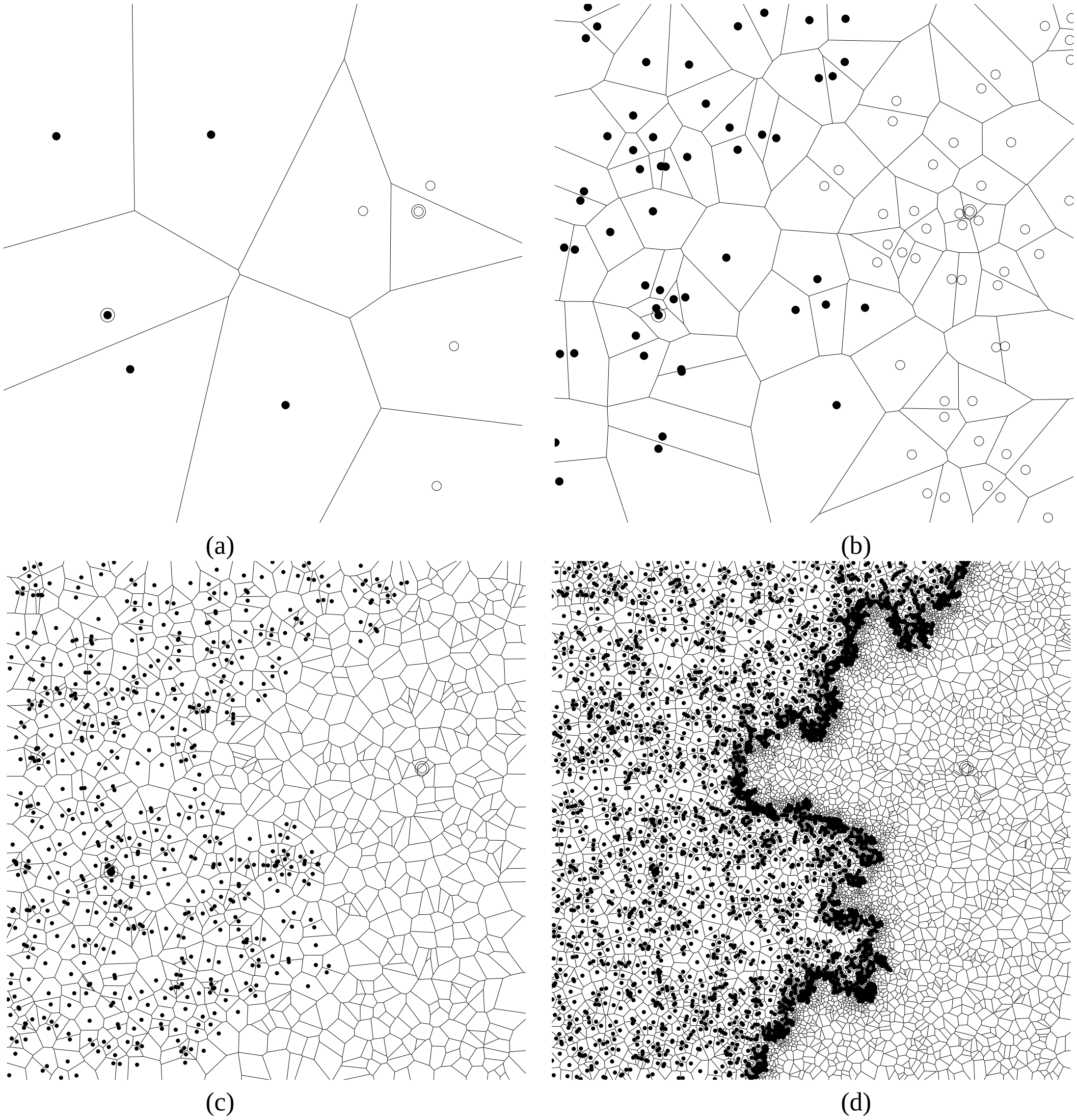


Figure 4. Change of territories based on the territory competing model: (a) represents a territory pattern for 10 points, and (b), (c), (d) represent territory patterns for 100, 1000, 6996 points respectively.

Let $S_2 = \{\mathrm{P}_1, \mathrm{P}_2\}$ be the set of these two seed points. Let $g(\mathrm{P}_i)$ denote the territory number, and we define

$$g(\mathrm{P}_1) = 1, \quad g(\mathrm{P}_2) = 2. \tag{2}$$

The Voronoi diagram for S_2 consists of two half planes bounded by the perpendicular bisector between P_1 and P_2. We consider the Voronoi region of P_1 as the territory 1 and the Voronoi region of P_2 as the territory 2.

Next, we generate new points $\mathrm{P}_3, \mathrm{P}_4, \ldots, \mathrm{P}_i, \ldots$ one by one at random, and for each point P_i we execute the following.

Let V_{i-1} be the Voronoi diagram for $S_{i-1} = \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_{i-1}\}$. Find the Voronoi region of V_{i-1} that contains P_i, that is, find point $\mathrm{P}_j \in S_{i-1}$ such that

$$\mathrm{P}_i \in R(S_{i-1}; \mathrm{P}_j). \tag{3}$$

Next, we assign the territory number $g(\mathrm{P}_j)$ to the new point P_i, that is, we define

$$g(\mathrm{P}_i) = g(\mathrm{P}_j). \tag{4}$$

Let us consider the above procedure in more detail. For a fixed i, the points in S_{i-1} are classified into two groups, points belonging to the territory 1 and these belonging to the territory 2. For set X of points in the plane, let $\mathrm{cl}(X)$ be the closure of X. We define

$$R_1^{i-1} = \bigcap_{g(\mathrm{P}_j)=1} \mathrm{cl}(R(S_{i-1};\mathrm{P}_j)), \tag{5}$$

$$R_2^{i-1} = \bigcap_{g(\mathrm{P}_j)=2} \mathrm{cl}(R(S_{i-1};\mathrm{P}_j)). \tag{6}$$

The region R_1^{i-1} is the collection of the Voronoi regions of the points in territory 1, and the region R_2^{i-1} is the collection of the Voronoi regions of the points in territory 2. The plane is divided into these two regions, and $R_1^{i-1} \cap R_2^{i-1}$ constitutes the boundary between them.

Therefore, for a new point P_i, the above procedure is first to find the territory that contains P_i and next to add P_i to this territory and thus augmenting this territory.

Without loss of generality, let us assume that $\mathrm{P}_i \in R_1^{i-1}$ and consequently P_i is classified as a point in the territory 1. Then, in general the territory 1 glows and the territory 2 shrinks. Hence we get

$$R_1^i \supseteq R_1^{i-1}, \quad R_2^i \subseteq R_2^{i-1}. \tag{7}$$

Note that the territory 1 is not necessarily greater than before. If the Voronoi region $R(S_i;\mathrm{P}_i)$ is surrounded by only Voronoi regions of points in the territory 1, P_i gets its region from the points in the same territory and hence the territory itself does not change. The territories change only when the new point P_i is generated near the boundary of the territory.

In Fig. 4, (b) represents the Voronoi diagram for 100 points, where the black dots represent points belonging to the territory 1 and the white circles represent points belonging to the territory 2. On the other hand, Fig. 4(c) and (d) represent the Voronoi diagram for 1000 points, and that for 6996 points, respectively, where the points belonging to the territory 2 are not drawn in order to avoid unnecessary complication.

In general, we start with k seed points, where k is a small positive integer, typically 2 or 3, and carry out the next procedure until the number of points reaches prespecified integer n.

Procedure 1 (Territory competing).

```
1. For a given set $S_k = \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_k\}$ of seed points,
   assign the territory numbers $g(\mathrm{P}_1) = 1, g(\mathrm{P}_2) = 2, \ldots, g(\mathrm{P}_k) = k$,
   and construct the Voronoi diagram for $S_k$.
2. Let $i \leftarrow k + 1$.
3. Generate new point $\mathrm{P}_i$.
4. Find the Voronoi region $R(S_{i-1};\mathrm{P}_j)$ that contains $\mathrm{P}_i$,
   and give to $\mathrm{P}_i$ the territory number by $g(\mathrm{P}_i) \leftarrow g(\mathrm{P}_j)$.
5. Let $S_i \leftarrow S_{i-1} \cup \{\mathrm{P}_i\}$, and construct the Voronoi diagram for $S_i$.
6. If $i = n$ stop. Otherwise $i \leftarrow i + 1$ and goto Step 3. ∎
```

4. Modification of the territory Competing Procedure

In Section 3, we present the basic structure of the Territory competing procedure. In actual generation of the pictures, we add many technical details to this basic procedure. Here, we present two major modifications.

4.1. Connectivity of the Territories

Simple execution of Procedure 1 can generate a territory consisting of two or more connected components. Suppose that the new point P_i is assigned the territory number j. Then, the territory j glows, and the resulting region of the territory j is still connected. However, the other territories lose part of their regions, and consequently may become disconnected.

An example of this situation is shown in Fig. 5. Suppose that, after some repetition of Steps 3, 4, 5 and 6 of Procedure 1, the territory pattern shown in Fig. 5(a) is generated, where the black circles in this figure belong to territory j, and the white circles belong to territory l. Suppose, furthermore, that a new point P_i is generated at the point represented by the double circles. Since this point belongs to the territory j, the territory number j is assigned to P_i in Step 4 of Procedure 1, and the new territory pattern is generated in Step 5. Then, the territory pattern changes to that in Fig. 5(b), where the territory j grows while the territory l shrinks, which tears the territory l into two connected components as shown in this figure.

We want to avoid such disconnected territories and hence modify Procedure 1 so that all the territories are guaranteed to be connected. For this purpose, if territory l becomes disconnected because of the growth of the territory j, we consider the unique connected component of l that contains its seed point as the territory l, and rename the other components as part of the territory j as shown in Fig. 5(c). By this modification, we can guarantee all the territories to be connected.

4.2. Control of the Density of Points

We generate new points at random. However, it is not necessarily uniform. Actually if we generate new points according to the uniform distribution, most of the points do not contribute to the change of the territories because they are entirely contained in a single territory. That is inefficient from a computational complexity point of view.

In order to generate a complicated pattern more efficiently, we control the density of new points in such a way

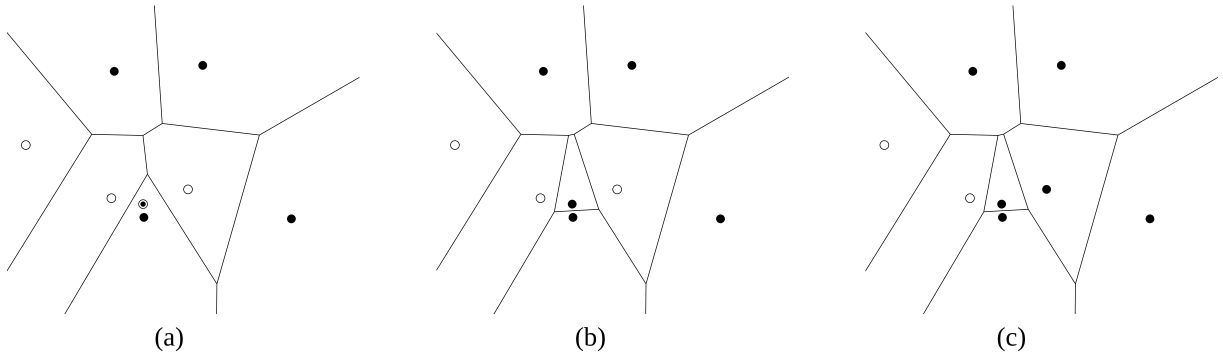


Figure 5. Disconnected territory generated by a simple application of Procedure 1.

that more points are generated around its boundaries of the territories.

In Fig. 4 (d), for example, we can see that the points are denser near the boundary of the two territories.

In order to control the density of points efficiently, we use the bucketing technique. That is, we divide the square region of the picture plane into four subsquares recursively, and generate more points in the subsquares that contain the territory boundaries than the other subsquares.

In the present implementation of the software system, the depth of the recursion of square subdivision increases as the number i of the repetition in Procedure 1 increases. More specifically, for some fixed positive integer n_0 and positive real s, we partition the whole square region into

$$\lfloor \log_2 si \rfloor \times \lfloor \log_2 si \rfloor \tag{8}$$

subsquares, when $i \geq n_0$, where $\lfloor x \rfloor$ represents the maximum integer not greater than x. For all the examples in this paper, we adopted $n_0 = 1000$ and $s = 0.1$. In other words, we did not partition the original square for $i < 1000$ (i.e., for the first 1000 repetitions of Steps 3, 4, 5 and 6 of Procedure 1), partitioned into 49 ($= 7 \times 7$) squares for $1000 \leq i \leq 1279$ (note that $\lfloor \log_2 si \rfloor = 7$ for $1000 \leq i \leq 1279$), partition into 64 ($= 8 \times 8$) squares for $1280 \leq i \leq 2559$ (note that $\lfloor \log_2 si \rfloor = 8$ for $1280 \leq i \leq 2559$), etc.

Moreover, we judge that a bucket (i.e., a subsquare) contains the territory boundaries if the bucket contains points belonging to mutually different territories.

5. Color Assignment

Suppose that we are given a Voronoi diagram, and a grouping of the generating points into territories. Our next task is to paint this pattern with colors. For this purpose we assign colors to the generating points; in other words we paint each Voronoi region in a uniform color.

First, we define the distance from the seed point to each Voronoi region belonging to the same territory as the seed points. Specifically, for seed point k, we do the following.

Procedure 2 (Distance from a seed point).
1. Define the distance of the seed point P_k as 0.
2. $j \leftarrow 1$.
3. Assign distance j to all the points (i) that belong to the territory k, (ii) that has not yet got the distance value, and (iii) that are adjacent to Voronoi regions with distance $j - 1$.
4. If all the points belonging to the territory k get their distance values, stop. Otherwise $j \leftarrow j + 1$, and go to Step 3. ▮

Let us denote by $D(\mathrm{P}_i)$ the distance of point P_i from the associated seed point. Hence, $D(\mathrm{P}_i)$ is a non-negative integer.

Each color can be represented by a point in a three-dimensional color space. Among may color spaces, we use the HSB space. A point x in this space is represented by a three-dimensional vector $\boldsymbol{x} = (x_1, x_2, x_3)$, where

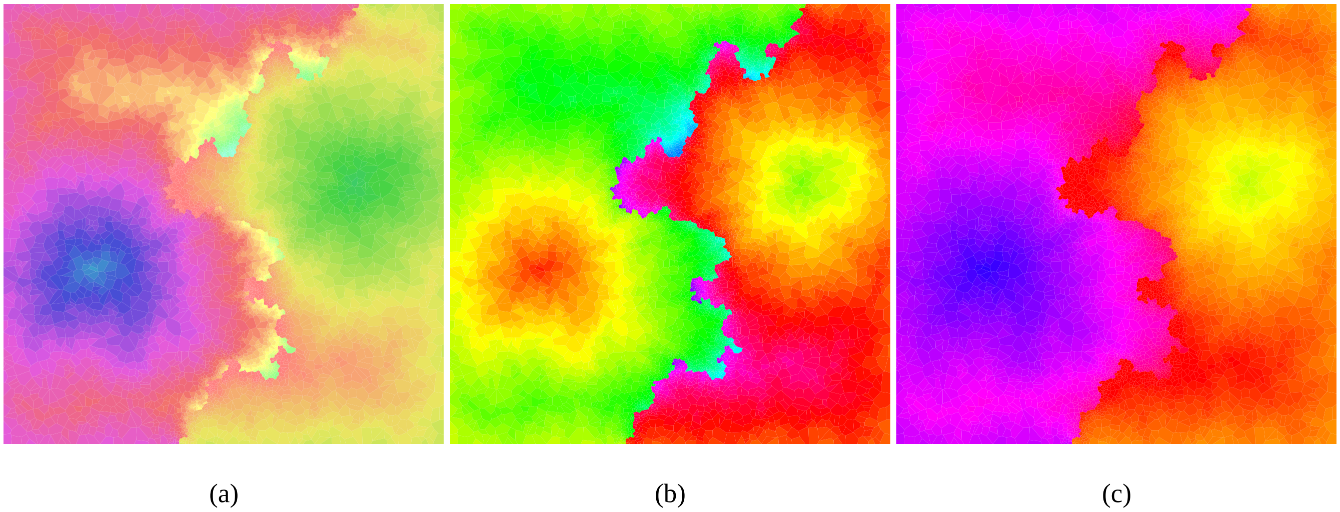
$$0 \leq x_1, x_2, x_3 \leq 1.$$

The first coordinate x_1 represents hue in a cyclic manner. $x_1 = 0$ corresponds to red, and as x_1 increases, the associated hue changes gradually to yellow, green, blue, purple and goes back to red when x_1 reaches $x_1 = 1$. The second coordinate x_2 represents the saturation; $x_2 = 0$ corresponds to the least clear color, while $x_2 = 1$ corresponds to the most clear color. The third coordinate x_3 represents the brightness. $x_3 = 0$ corresponds to the darkest color, while $x_3 = 1$ corresponds to the brightest color.

We assign colors to the Voronoi regions in the following way. For each territory k, we choose color

$$c^k = ({c_1}^k, {c_2}^k, {c_3}^k) \tag{9}$$

at random and assign it to the associated seed point. Next, we choose a vector $d^k = ({d_1}^k, {d_2}^k, {d_3}^k)$ and in order to



(a) (b) (c)

Figure 6. Examples of pictures generated by color assignments to the territory pattern in Fig. 4(d).

choice the color for P_i we shift the color c^k in the direction d^k by the distance $D(\mathrm{P}_i)$, and if necessary, we modify the resulting point so that it is inside the color space. More specifically, we define

$$\begin{pmatrix} x_1{}' \\ x_2{}' \\ x_3{}' \end{pmatrix} = D(\mathrm{P}_i) \begin{pmatrix} d_1{}^k \\ d_2{}^k \\ d_3{}^k \end{pmatrix} + \begin{pmatrix} c_1{}^k \\ c_2{}^k \\ c_3{}^k \end{pmatrix}, \qquad (10)$$

and then compute

$$x_1 = \begin{cases} \text{fractional part of } 1 - |x_1{}'| & (\text{if } x_1{}' < 0), \\ x_1{}' & (\text{if } 0 \le x_1{}' \le 1), \\ \text{fractional part of } x_1{}' & (\text{if } 1 < x_1{}'), \end{cases} \qquad (11)$$

$$x_2 = \begin{cases} 0 & (\text{if } x_2{}' < 0), \\ x_2{}' & (\text{if } 0 \le x_2{}' \le 1), \\ 1 & (\text{if } 1 < x_2{}'), \end{cases} \qquad (12)$$

$$x_3 = \begin{cases} 0 & (\text{if } x_3{}' < 0), \\ x_3{}' & (\text{if } 0 \le x_3{}' \le 1), \\ 1 & (\text{if } 1 < x_3{}'). \end{cases} \qquad (13)$$

We assign the resulting color (x_1, x_2, x_3) to the Voronoi region of P_i.

The intuitive meaning of the computation of the color value (x_1, x_2, x_3) is as follows. In the HSB system, each color is represented by a point in the unit cube in the (x, y, z) coordinate system. For the Voronoi region corresponding to the seed point of territory k, we assign the color $(c_1{}^k, c_2{}^k, c_3{}^k)$, and for the Voronoi regions of P_i belonging to the same territory we basically assign colors $(x_1{}', x_2{}', x_3{}')$ as defined by eq. (10), where the color is changed from $(c_1{}^k, c_2{}^k, c_3{}^k)$ and the amount of change is proportional to the distance $D(\mathrm{P}_i)$. However, the point computed by eq. (10) may not be included in the unit square. In that case, the color is not defined. Hence, we modify the point $(x_1{}', x_2{}', x_3{}')$ to (x_1, x_2, x_3) in such a way that the resulting point belongs to the unit square. Since the x_1 axis represents a color category in a cyclic manner, we choose the fractional part by eq. (11). Since the x_2 represents the degree of clearance between the least clear color $x_2 = 0$ and the most clear color $x_2 = 1$, and hence the outside value is changed to the nearest point in $0 \le x_2 \le 1$ by eq. (12). Similarly the x_3 axis represents the degree of brightness between the darkest color $x_3 = 0$ and the brightest color $x_3 = 1$, and hence the outside value is changed to the nearest point in $0 \le x_3 \le 1$ by eq. (13).

In summary, for the territory k we choose six parameters $c_1{}^k, c_2{}^k, c_3{}^k, d_1{}^k, d_2{}^k, d_3{}^k$ and assign colors according to the above equations.

Fig. 6 shows examples of the colored pictures obtained from the territory pattern in Fig. 4(d). We can see that pictures with a variety of different tastes can be generated from one and the same territory pattern.

6. Concluding Remarks

In this paper we presented a method for generating artistic pictures based on the territory competing model. In this model, the territories are represented by collections of Voronoi regions for points, and the territories change according to the assignment of the territory numbers to new points generated at random.

The method proposed in this paper offers a new simple mechanism for creating artistic pictures. There is a menu of artistic pattern generators, which includes fractal and chaos as typical examples. This paper augments the menu by adding the Voronoi diagram as a new member.

The directions for future researches include (i) to extend and generalize the mechanism in order to add a wide rage of variants of pattern generators, (ii) to offer a tool kit for painters, (iii) to combine this pattern generator with activities of human artists and (iv) extension to the three-dimensional space.

Acknowledgements

The author would like to express his thanks to many people who gave suggestions and encouragements, particularly to Mr. Yoshiaki Kawakatsu who encouraged me to this direction of work based on his methodology called "dream engineering".

References

[1] E. Angel: *Interactive Computer Graphics*. Second Edition, Addison-Wesley, Reading, 2000.

[2] M. F. Barnsley: *Fractals Everywhere, Second Edition.* Academic Press, New York, 1993.

[3] R. L. Devaney: *An Introduction to Chaotic Dynamical Systems, Second Edition*. Perseus Books Publishing, New York, 1989.

[4] J. Gleick: *CHAOS — Making a New Science.* William Morris, New York, 1987. pp. 167–180, 1978.

[5] A. Iglesias: Computer graphics for water modeling and rendering — A survey. *Future Generation Computer Systems* wol. 20 (2004), pp. 1355–1374.

[6] B. B. Mandelbrot: *The Fractal Geometry of Nature*. W. H. Freeman, New York, 1982.

[7] E./ Nakamae and K. Tadamura: Photorealism in computer graphics — Past and present. *Computers and Graphics*, wol. 19 (1995), pp. 119–130.

[8] S. Nikiel and A. Goinski: Generation of volumetric escape time fractals. *Computers and Graphics*, wol. 27 (2003), pp. 977–982.

[9] A. Okabe, B. Boots, K. Sugihara and S.-N. Chiu: *Space Tessellations—Concepts and Applications of Voronoi Diagrams, Second Edition.* John Wiley and Sons, Chichester, 2000.

[10] C. Pickover (ed.): *The Pattern Book — Fractals, Art, and Nature*. World Scientific, Singapore, 1995.

[11] P. Rousseau, V. Jolivet and D. Ghazanfarpour: Realistic real-time rain rendering. *Computers and Graphics*, wol. 30 (2006), pp. 507–518.

[12] S. Xu, M. Tang, F. C. M. Lau and Y. Pan: Virtual hairy brush for painterly rendering. *Graphical Models*, vol. 66 (2004), pp. 263–302.

[13] Y. Zou, W. Li, J. Lu and R. Ye: Orbit trap rendering method for generating artistic images with cyclic or dihedral symmetry. *Computers and Graphics*, wol. 30 (2006), pp. 470–473.