

Research Article

Image-Guided Voronoi Aesthetic Patterns with an Uncertainty Algorithm Based on Cloud Model

Tao Wu^{1,2} and Limin Zhang²

¹*School of Information Science and Technology, Lingnan Normal University, Zhanjiang 524048, China*

²*Guangdong Engineering and Technological Development Center for E-Learning, Zhanjiang 524048, China*

Correspondence should be addressed to Tao Wu; wutao@whu.edu.cn

Received 3 March 2016; Revised 17 June 2016; Accepted 26 June 2016

Academic Editor: Marco Perez-Cisneros

Copyright © 2016 T. Wu and L. Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tessellation-based art is an important technique for computer aided aesthetic patterns generation, and Voronoi diagram plays a key role in the preprocessing, whose uncertainty mechanism is still a challenge. However, the existing techniques handle the uncertainty incompletely and unevenly, and the corresponding algorithms are not of high efficiency; thus it is impossible for users to obtain the results in real time. For a reference image, a Voronoi aesthetic pattern generation algorithm with uncertainty based on cloud model is proposed, including uncertain line representation using an extended cloud model and Voronoi polygon approximation filling with uncertainty. In view of the different parameters, seven groups of experiments and various experimental analyses are conducted. Compared with the related algorithms, the proposed technique performs better on running time, and its time complexity is approximatively linear related to the size of the input image. The experimental results show that it can produce visually similar effect with the frayed or cracked soil and has three advantages, that is, uncertainty, simplicity, and efficiency. The proposal can be a powerful alternative to the traditional methods and has a prospect of applications in the digital entertainment, home decoration, clothing design, and various fields.

1. Introduction

Computational aesthetics is a major unsolved problem in computer science and engineering [1, 2]. Over the last decade, there has been increasing interest in using computational intelligence approaches to solve this problem [3–8]. Of those, tessellation-based Voronoi art is an important technique and can be widely applied in various fields, that is, architecture, jewellery design, fashion design, and so forth, and Voronoi diagram plays a key role in its preprocessing [9]. In fact, tessellation-based or area-based visual representations are common to many artistic works or computer-based visualization systems. Also, it would be beneficial to researchers to develop and test the computational intelligence algorithms for creating interesting and aesthetic images and videos based on Voronoi diagram.

In recent years, several approaches to create nicely looking patterns have been described in literature. For example, Kaplan first proposed Voronoi-based art and applied it to

decorative design [10]; the similar style includes portrait stylization [11], image filter [12], and inscribed curve [13]. These above methods fall into Voronoi generation with an accurate perspective. More recently, some uncertain methods with computational intelligence algorithms have been surfaced and received some attentions, such as the fuzzy border of Voronoi polygon [14] and the probabilistic Voronoi model [15, 16]. In addition, the uncertain Voronoi structure also exists in other techniques for computational aesthetics, and Isenberg [17] and Kim et al. [7] proposed the abstraction method, respectively. Based on this, Michael et al. provided a random method for Voronoi art [8].

Nonetheless, each method has its own advantages, disadvantages, and applied situations. The existing methods with uncertainty are based on fuzzy set or probability statistics, and there are some drawbacks of the processing, and their results are unsatisfactory or even questionable in some cases. Firstly, the existing methods cannot completely capture the uncertainty in the Voronoi generation. The fuzzy Voronoi

diagram theoretically extends the border as a transition region, which is still a continuous plane, not applicable to the image stylization, while the random Voronoi diagram only adds a shadow to the real boundary. In fact, the uncertainty includes fuzziness, randomness, and the connection of them. Thus, any partial solution to only one of these aspects that does not take the other into account is incomplete. Secondly, the existing methods with uncertainty are not very efficient. The rendering is point-based operation, which is simple but time-consuming and difficult to apply in real-time production. We believe that the mathematical representation of a concept with uncertainty is one of the foundations of computational intelligence, and uncertainty is an inherent part of Voronoi art in real world applications; then the aesthetic generation with uncertainty is still a challenge. Therefore, we should further extend and discuss the traditional approaches from a developmental point of view. Cloud model, different from statistical methods and fuzzy set methods, can handle such uncertainty in a better way, since it provides us with more design degrees of freedom, at least the second-order uncertainty.

In this context, we proposed an image-guided Voronoi aesthetic patterns with an uncertainty algorithm based on cloud model (iVPC for short). It is noted that what it meant is not the Virtual Private Cloud, and the cloud model is completely different from cloud computing. Our intentions are threefold: (1) how can the image-guided Voronoi art be produced using a cloud model-based algorithm? (2) What is a suitable configuration of the cloud model-based algorithm for image-guided Voronoi art? (3) How different types of Voronoi decomposition can affect the aesthetic qualities of the rendered images?

Cloud model, compared to similar techniques, fuzzy set and rough set, is a cognitive model between a qualitative concept and its quantitative instantiations and successfully used in various applications [18–22]. The cloud model-based rendering consists of four main steps, including random reference points-based or image reference-based Voronoi decomposition, uncertain line representation using the extended cloud model, uncertain Voronoi polygon approximation filling, and then generating an aesthetic image with the Voronoi art style.

Our method satisfies the USE property, and it stands for uncertainty, simplicity, and efficiency: (1) the proposed method is involved as a novel technique with uncertainty using cloud model, although there have been some efforts in using computational intelligence algorithms. (2) The proposed method is very simple, and only the uncertain representation step on Voronoi polygon and its border are introduced into the classical Voronoi-based method. (3) From the perspective of running time, the proposed method is efficient, and its time complexity is approximately linear with the size of the original image for image-guided decomposition.

The remainder of this paper is organized as follows. We describe our iVPC algorithm in detail: first uncertain line using cloud model in Section 2, next uncertain Voronoi polygon in Section 3, and then cloud model-based algorithm and its time complexity in Section 4. In Section 5, we

investigate the parameter configuration and show several examples of the resulting images and then conduct four groups of experiments, including visual comparisons, time performance analysis, quantitative comparisons, and the user study. Finally, we discuss the results and give some ideas for future improvements in Section 6.

2. Uncertain Line Using Cloud Model

2.1. Cloud Model. Cloud model, proposed by Li et al. [18, 19], is the innovation and development of membership function in fuzzy theory and uses probability and mathematical statistics to analyse the uncertainty [19]. In theory, there are several forms of cloud model, successfully used in various applications, including image processing, data mining, geological analysis, and knowledge engineering [18–22]. However, the normal cloud model is commonly used in practice, and its theoretical foundation is the universality of normal distribution and bell membership function [18, 21].

Let U be a universe set described by precise numbers, and let C be a qualitative concept related to U . Given a number $u \in U$, which randomly realizes the concept C , u satisfies $u \sim N(\text{Ex}, \text{En}n^2)$, where $\text{En}n \sim N(\text{En}, \text{He}^2)$, and the certainty degree of u on U is as below:

$$\mu(u) = \exp\left(-\frac{(u - \text{Ex})^2}{2\text{En}n^2}\right), \quad (1)$$

and then the distribution of u on U is defined as a normal cloud, and u is as a cloud drop.

The overall property of a concept $C(\text{Ex}, \text{En}, \text{He})$ can be represented by three numerical characters of normal cloud model, expected value Ex , entropy En , and hyper-entropy He . Ex is the mathematical expectation of the cloud drop distributed in universal set. En is the uncertainty measurement of the qualitative concept, and it is determined by both randomness and fuzziness of the concept. He is the uncertain measurement of entropy, which is determined by randomness and fuzziness of entropy En [18]. It is worth noting that hyper-entropy He of a cloud model is a deviation measure from a normal distribution; hence, the distribution of cloud drops can be regarded as a generalized normal distribution.

The kernel of normal cloud model is the transform between qualitative concept and quantitative data [21], and it is realized by normal cloud generators. On one hand, forward normal cloud generator is the mapping from qualitative concept to quantitative values, and it produces the cloud drops to describe a concept when three numerical characters and the number of cloud drops are input. On the other hand, backward normal cloud generator provides the transformation from quantitative numerical values to quality concept, and a normal cloud model with three numerical characters is defined by computing mean, absolute central moment with the first order, and variance of sample data. Essentially, the normal cloud generators are two algorithms based on probability measure space [18]; these processes are uncertain and cannot be expressed by a precise function. More information about normal cloud model can be obtained from [18].

2.2. *Uncertain Line.* According to the above idea, a line segment can also be represented by an extended cloud model, if considering it as a qualitative concept with the numerical characters. Arbitrary line is a part of a continuous domain in 2D plane, and an uncertain transformation from the quantitative value and the qualitative concept can be easily achieved using a cloud model. Based on the traditional Gaussian cloud model, we define the uncertain line and set of sampling points.

Given two endpoints of a line segment, that is, $\mathbf{P}_1 = (px_1, py_1)$ and $\mathbf{P}_2 = (px_2, py_2)$, where px_1, py_1, px_2, py_2 are the coordinates, we can take the slope k and the vertical intercept b as a key concept to be expressed by cloud model, which is determined by (Ex_k, En_k, He_k) and (Ex_b, En_b, He_b) . Consequently, a line segment would be defined by the parameter set $\{k, b\}$ as below:

$$\left(\begin{bmatrix} Ex_k \\ Ex_b \end{bmatrix}, \begin{bmatrix} En_k \\ En_b \end{bmatrix}, \begin{bmatrix} He_k \\ He_b \end{bmatrix} \right), \quad (2)$$

where Ex_k, Ex_b are the real values of slope and vertical intercept can be calculated by the following:

$$\begin{aligned} Ex_k &= \frac{py_2 - py_1}{px_2 - px_1}, \\ Ex_b &= py_1 - px_1 \frac{py_2 - py_1}{px_2 - px_1}. \end{aligned} \quad (3)$$

Once a vertical line is involved into the above equation, that is, $px_2 = px_1$, the slope does not exist, the abscissa should be swapped with the ordinate, and the only thing is an *if* statement using matrix transposition in program implementation. Even so, the abscissa values of two endpoints in Voronoi polygons are almost impossibly the same.

Let D , consisting of quantitative values, be a continuous domain in the plane, and let a line L be the qualitative concept related to D . For a line $l \in L$, it randomly realizes the concept L , which is determined by the slope k and the vertical intercept b . In addition, k, b satisfy $k \sim N(Ex_k, Enn_k^2)$ and $b \sim N(Ex_b, Enn_b^2)$, where $Enn_k \sim N(En_k, He_k^2)$ and $Enn_b \sim N(En_b, He_b^2)$, and the certainty degree of l on L is as below:

$$\mu = \mu_k \mu_b, \quad \mu_k = e^{-\frac{(k-Ex_k)^2}{2Enn_k^2}}, \quad \mu_b = e^{-\frac{(b-Ex_b)^2}{2Enn_b^2}}. \quad (4)$$

Each random realization l of a concept L is expressed by a set of points; then the distribution of l on D is defined as an extended Gaussian cloud model, and μ is as the certainty degree. For simplicity, we only take one point instead of a set of points. (lx_i, ly_i) ($i = 1, 2, \dots, n$) denotes the selected point on the i th random realization, corresponding to the line l_i , and n is the number of random process. The point set $\{(lx_1, ly_1), (lx_2, ly_2), \dots, (lx_n, ly_n)\}$, corresponding to all the cloud drops, constitutes a cloud concept on the line, and the approximate entity can be called uncertain line.

Different cloud drops, with various parameters and from a Gaussian cloud model, correspond to different slope and intercept, and all of these cloud drops are one of forms on

the line segment in 2D space. A given line segment can be approximated by n discrete cloud drops using a generation algorithm, and then the quantitative value and the qualitative concept can be transformed to each other uncertainly. The detailed algorithm is as follows.

Step 1. Given \mathbf{P}_1 and \mathbf{P}_2 , swap the abscissa with the ordinate if $px_2 = px_1$.

Step 2. According to (3), calculate the expected values $[Ex_k, Ex_b]^T$, and set the entropy $[En_k, En_b]^T$ and hyper-entropy $[He_k, He_b]^T$.

Step 3. Generate a normal random number Enn_k using the expectation En_k and the variance He_k^2 and another random number k_i using the expectation Ex_k and the variance Enn_k^2 .

Step 4. Generate a normal random number Enn_b using the expectation En_b and the variance He_b^2 and another random number b_i using the expectation Ex_b and the variance Enn_b^2 .

Step 5. For the selected point on the i th random realization l_i , obtain the abscissa value $lx_i = \text{rand}(1)(px_2 - px_1) + px_1$, where $\text{rand}(1)$ denotes a random number in $[0, 1]$, and then calculate the ordinate value $ly_i = k_i lx_i + b_i$.

Step 6. Calculate the certainty degree μ_i using (4) and then take this line l_i with certainty degree μ_i as a cloud drop on the concept L , and (lx_i, ly_i) is one of the discrete random samples. Of course, the number of cloud drops increases by 1.

Step 7. Repeat Steps 3 to 6 until the number of cloud drops reaches the predefined value n .

Given the endpoints $\mathbf{P}_1 = (0, 1)$ and $\mathbf{P}_2 = (4, 9)$, the expected values are calculated as $Ex_k = 2$ and $Ex_b = 1$, and the other numerical characters are fixed as $En_k = 0.1 Ex_k = 10He_k$ and $En_b = 0.1 Ex_b = 10He_b$. We can generate 50 cloud drops, and the location relations between the sample points and the line are drawn on the coordinate system. As shown in Figure 1(a), these sample points are around the line segment with a nonuniform distribution. Intuitively, most of offset values between the sample points and the line segment are acceptable, and then a concept "near or around the line segment" is achieved. In this point of view, an uncertain representation of a line segment is feasible and reasonable using the extended cloud model.

Next, we generate 500 cloud drops, and the relations among μ_k, μ_b , and μ are listed in Figure 1(b). With the increase of μ_k and μ_b , μ is also increased. The distribution of cloud drops near the peak is more intensive, and these cloud drops contribute more to representing the concept. It is noted that the certainty degree is a fuzzy measure on the concept, whose calculation itself is a fuzzy process (as Step 6), and this fuzzy measure characterizes the membership degree of sampling and reflects the value ranges of the accepted sample points on the domain space. But other than the pure fuzzy problems, randomness is also involved, since the sample points depend on the abscissa values of random sampling, as Step 5,

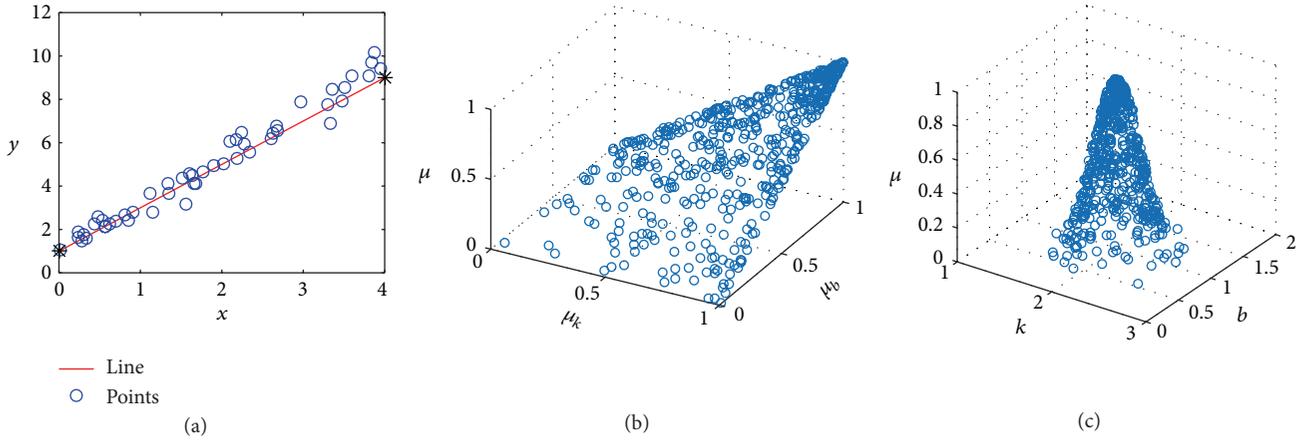


FIGURE 1: Uncertain line and its cloud drops: (a) a line with the sample points, (b) the joint distribution of certainty degree for μ_k, μ_b , and (c) the joint distribution of certainty degree k, b .

the random measure indicates the dispersion of cloud drops representing a line concept.

Furthermore, the relations among k , b , and μ are listed in Figure 1(c). Different groups of cloud drops play the different roles to the same concept, and there is a thick and dense distribution around the peak. With the slope k and the intercept b closing to the expected values Ex_k, Ex_b , the certainty degree of cloud drop is also larger, and it is coincident with the result in Figure 1(b). In fact, $\mu_k \rightarrow 1, \mu_b \rightarrow 1$ with $k \rightarrow Ex_k, b \rightarrow Ex_b$, and there is a close correlation between these two according to (4). The determination on the slope and the intercept is a random process, characterizing the dispersion degree of the line sampling, but it is not just a pure random problem; the associated certainty degree reflects the fuzziness.

In general, cloud model studies the randomness of membership grade, which is considered as fuzziness, randomness, and the connection of them. The proposed processing of randomness and fuzziness is not independent, and then it would take into account the correlation of them. The probabilistic Voronoi methods used expectation, variance, and other statistical characteristics, which reflected the randomness, but did not touch the fuzziness. While the fuzzy Voronoi theory calculated the membership degree using an accurate method, it failed to capture the randomness. We represent an uncertain line using cloud model and take into account the fuzziness, the randomness, and the relationship in between. Cloud model-based method can describe the concept “near a line segment,” which is more in accord with human cognition.

There is not a simple line processing in the Voronoi boundaries, and then an intuitive idea is to process and draw each boundary sequentially. Given a Voronoi polygon with four vertices, $\mathbf{P}_1 = (0, 1)$, $\mathbf{P}_2 = (4, 9)$, $\mathbf{P}_3 = (2, -15)$, and $\mathbf{P}_4 = (-1, -5)$, the uncertain expression is shown in Figure 2, and the sample points approximate the four lines.

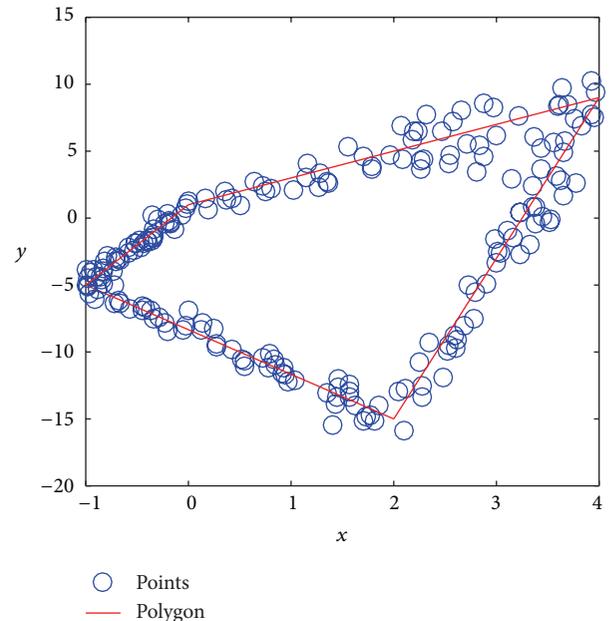


FIGURE 2: Uncertain polygon and its points.

3. Uncertain Voronoi Polygon Using Cloud Model

3.1. Image-Guided Voronoi Decomposition. For image-guided Voronoi art, the generation of Voronoi diagram is completely dependent on the image content. There exist various techniques, including image segmentation [23] and half-tone processing [24]. In the following, we use the Floyd-Steinberg dithering algorithm [25] since it is a simple, popular, and efficient method based on error diffusion. If involving the user interaction, Photoshop and other popular software can also be used to obtain a half-tone image.

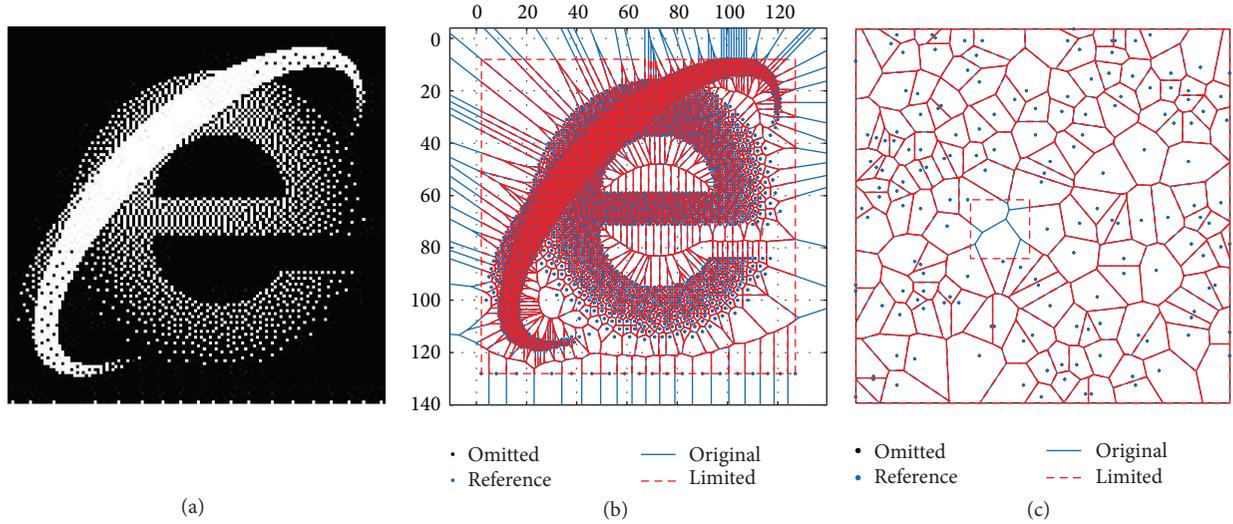


FIGURE 3: Voronoi decomposition: (a) the dithered IE logo image, (b) Voronoi decomposition for (a), and (c) an unsupervised decomposition.

As shown in Figure 3(a), the IE logo is processed by Floyd-Steinberg dithering algorithm. The pixels with nonzero grayscale values are selected and their coordinate values are as the reference points to generate Voronoi polygons. The number of the reference points is denoted by N , which is automatically determined by the Floyd-Steinberg algorithm. Then the Voronoi decomposition is generated with an outer-boundary constraint, and we use the VoronoiLimit method proposed by Jakob [26]. The decomposition result is shown in Figure 3(b), where the dot represents a reference point, and the dotted line represents a new constrained boundary.

We call the above decomposition supervised decomposition, which is image-guided. Moreover, the reference point can be also produced by a random method, that is, unsupervised decomposition.

Given the width w , the height h , and the density factor ρ , the reference points are randomly generated, and the Voronoi decomposition result is varied for each different running. The number of the reference points is $N = \text{round}(hw\rho)$.

Figure 3(c) shows an example of unsupervised decomposition; there is a rectangular area containing an internal hole. Also for simplicity, we investigate the parameters with the unsupervised decomposition in Section 5.

3.2. Uncertain Voronoi Aesthetic Patterns. After the Voronoi polygons are determined by Voronoi decomposition, the rendering can be done in a natural way. A line from each reference point to each sample point is drawn and filled one by one, and all lines in each polygon are drawn with the same color, while different polygons are rendered with various colors. Then the aesthetic patterns with different styles can be produced.

Taking the quadrilateral in Figure 2 as an example, the reference point is the centroid of the polygon $\mathbf{P}_r = (5/4, -5/2)$; the filling result is shown in Figure 4(a). The lines mainly cover the triangular region determined by the endpoints of the line and the reference point. But the distribution of lines

is not uniform, and some are sparse and some are intensive. In other words, the number of cloud drops may affect the aesthetic patterns.

In the case of enough number of cloud drops, we add an extra parameter related with the certainty degree, that is, intensity factor τ , and the aesthetic pattern can be controlled by removing the cloud drops and their sample points, whose certainty degree is less than τ .

Similarly, all uncertain lines are processed one by one, and the polygons are rendered with various styles. Although the method is very simple, there are two problems not solved: (1) near the vertices of polygons, the sample points from the adjacent borders are different; then the filling would cause the cross or overlap of lines. (2) The iterative rendering is of low efficiency and is time-consuming. To avoid the weaknesses and improve the algorithm performance, we provide the following method to process the uncertain polygon, which is an improved version of the uncertain line.

Given the t -sided polygon of a Voronoi decomposition, the vertex set $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_t\}$ ($t \geq 3$), and its centroid $\mathbf{P}_r = (px_r, py_r)$ (also as the reference point), the sides are determined by $\mathbf{P}_1\mathbf{P}_2, \mathbf{P}_2\mathbf{P}_3, \dots, \mathbf{P}_{t-1}\mathbf{P}_t, \mathbf{P}_t\mathbf{P}_1$, and each side is represented by n cloud drops according to the previous section; then the polygon corresponds to nt sample points. Consequently, all the cloud drops are determined by a point set $\{(lx_1, ly_1), (lx_2, ly_2), \dots, (lx_{nt}, ly_{nt})\}$, and their polar angles can be calculated as below:

$$\theta_j = \arg \tan (py_r - ly_j, px_r - lx_j), \quad (5)$$

where $j = 1, 2, \dots, nt$.

Once the polar angles are obtained, all sampling points are ordered and reorganized clockwise, taking the reference point as the center, and the duplicate samples are removed; then a polygon is simultaneously filled by the fill() function in Matlab.

Following this step, the aforementioned quadrilateral is filled, and 200 cloud drops are generated. The result is shown

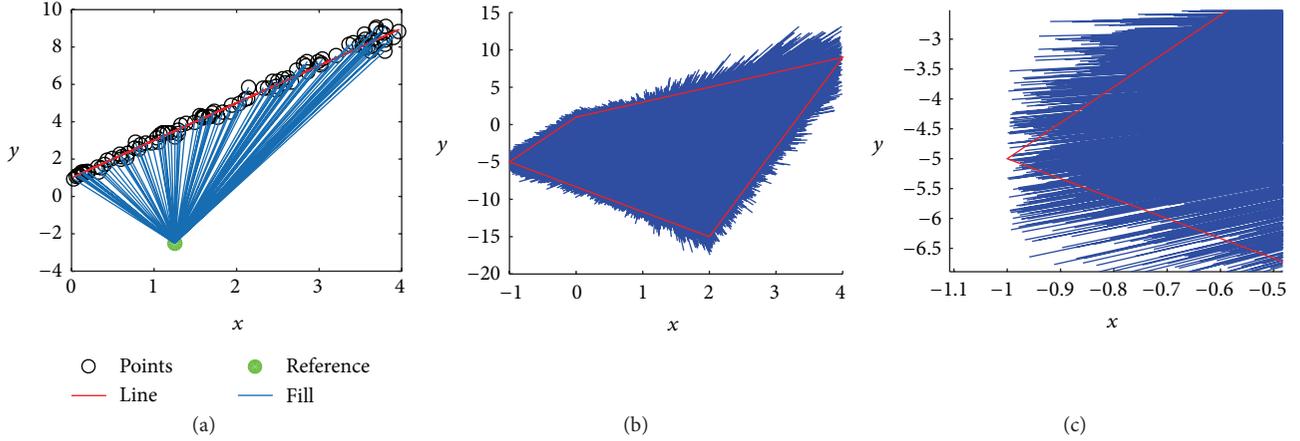


FIGURE 4: Filling uncertain polygon: (a) lines from the reference points and the sample points, (b) the filled polygon, and (c) the detail with an enlarged scale for (b).

in Figure 4(b); there are rough boundaries instead of the original accurate ones. As shown in Figure 4(c), there is the detail with an enlarged scale near P_4 . The number of sampling points is enough, but the fill result is delicate, without halting and with no overlapping.

4. The Proposed Algorithm

4.1. The Overview of iVPC Algorithm. To sum up, the proposed iVPC algorithm includes five main steps, and it is described as the following.

Step 1 (initialization). The parameters include the type of Voronoi decomposition, the numerical characteristics of cloud model $[En_k, En_b]^T$ and $[He_k, He_b]^T$, the number of cloud drops n , the number of the reference points which is denoted by N , the intensity factor τ , the colormap, and the selected background color. It should be noted that N is determined by the width w , the height h , and the density factor ρ for unsupervised decomposition, while, for supervised decomposition, N is autoset by the Floyd-Steinberg algorithm.

Step 2 (Voronoi decomposition). For image-guided decomposition, the reference points are produced by Floyd-Steinberg dithering algorithm. For unsupervised decomposition, the reference points are produced randomly. Then N Voronoi polygons are obtained.

Step 3 (uncertain line). For the c th Voronoi polygon, the cloud drops and sample points for t sides are generated by the method in Section 2.

Step 4 (uncertain Voronoi polygon). Voronoi polygon is approximated and filled using the method in Section 3.

Step 5 (loop). The counter c is incremented by $c = c + 1$. Repeat Steps 3 and 4 until all the Voronoi polygons are processed; that is, $c = N$.

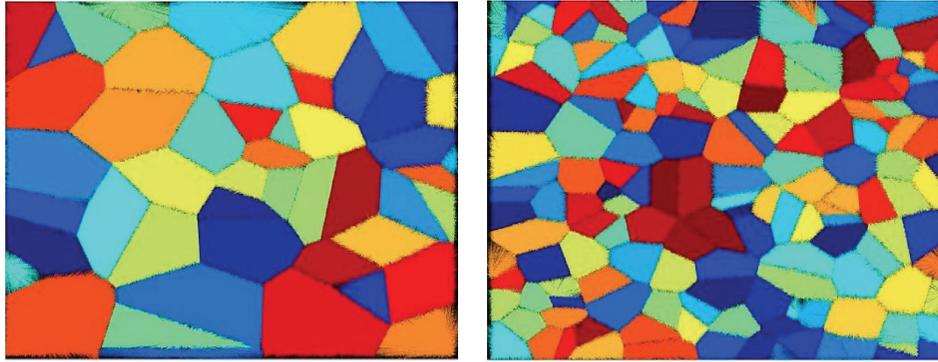
4.2. Time Complexity. Step 1 is a simple initialization; the time complexity is $O(1)$, and it is not the key step when analysing the time performance. Step 2 is the necessary time cost for all of algorithms on Voronoi art generation, which depends on the decomposition technique. Theoretically, this step can be as a preprocessing. In Step 3, n cloud drops consume the time of order $O(n)$, and for a t -sides polygon, the iteration takes time $O(nt)$. In Step 4, the time cost in a single loop is mainly the sort of polar angle; the average time is $O((nt) \log(nt))$, and the other operations take time $O(nt)$ in the worst case, and $O(1)$ in the best case. In addition, Steps 3 and 4 are repeated N times. Thus, the time complexity of these two steps would approximately be $O(Nnt)$. Although t is different for all the Voronoi polygons, it is not very large, generally less than 6. That is to say, $t \ll Nn$.

In general, our algorithm costs time $O(Nn)$. More accurately, it is $O(hw\rho n)$ in the case of unsupervised decomposition, and $\rho n \ll hw$ is necessary for aesthetic reason; we will further investigate this point. A similar condition is also necessary for image-guided decomposition; that is, N cannot be too large. Therefore, the time complexity of the proposed algorithm is approximately linear with the size of the original image for image-guided decomposition, and the time complexity analysis indicates that our method is of high efficiency.

5. Experimental Results

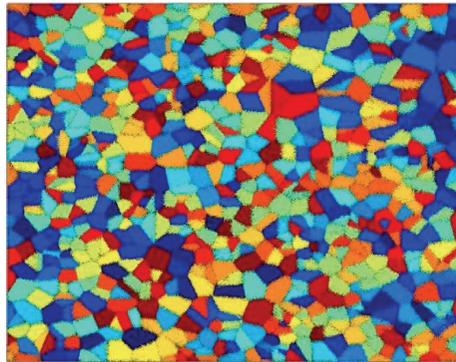
5.1. Parameter Configuration. In order to investigate the parameter configuration of the proposed algorithm, both image-guided decomposition and unsupervised decomposition are considered, and we conduct seven groups of experiments.

5.1.1. Group 1. In this group, we test the density factor ρ for unsupervised decomposition. Given $h = w = 128$, three aesthetic patterns are generated using $\rho = 0.001, 0.01, 0.05$, respectively. As shown in Figure 5(a), it is indicated that the lower the density factor, the larger the area of the Voronoi

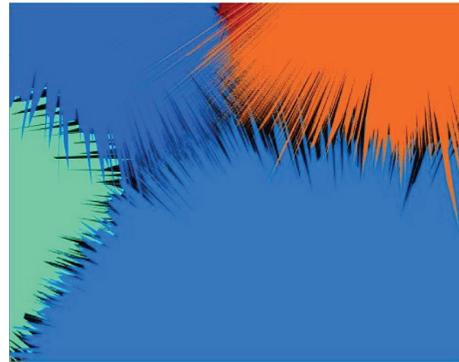


(a)

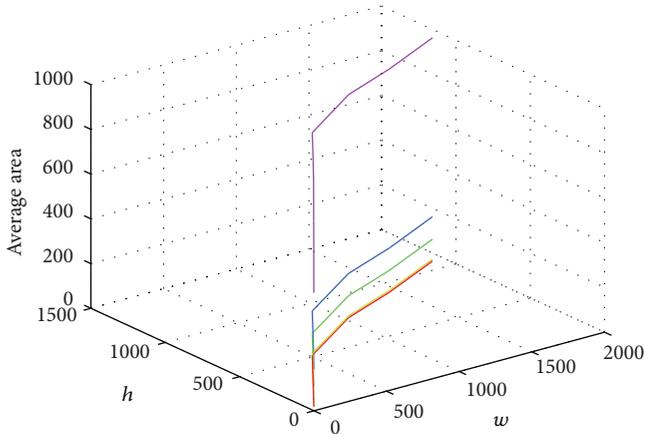
(b)



(c)

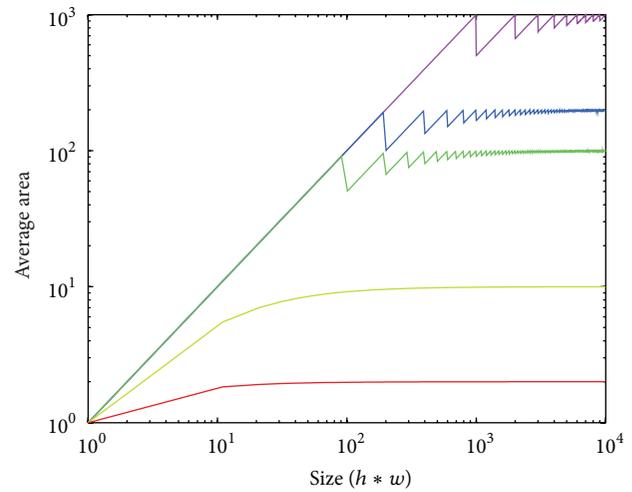


(d)



$\rho = 0.001$ $\rho = 0.1$
 $\rho = 0.005$ $\rho = 0.5$
 $\rho = 0.01$

(e)



$\rho = 0.001$ $\rho = 0.1$
 $\rho = 0.005$ $\rho = 0.5$
 $\rho = 0.01$

(f)

FIGURE 5: Voronoi decomposition with various densities: (a) $\rho = 0.001$, (b) $\rho = 0.01$, (c) $\rho = 0.05$, (d) the detail with an enlarged scale for (b), (e) average area of each Voronoi polygon varied with the height and the width, and (f) average area of each Voronoi polygon varied with the image size.

polygon and the less aesthetics the result pattern. While when the density factor is too high, the area of Voronoi polygon should be too small and an inordinate number of the reference points and extremely time-consuming performance would be inevitable, an example is shown in Figure 5(c).

Therefore, a moderate density factor is beneficial to the aesthetic effect.

The whole image is as Figure 5(b), which is with a proper density of reference points, a reasonable number of polygons, and better aesthetics. For detailed analysis, a subimage from

TABLE 1: Statistical results on cloud drops and their contribution with various cloud models.

Cloud models			Mean and standard deviation							
Ex	En	He	[Ex - 0.67En, Ex + 0.67En]		[Ex - En, Ex + En]		[Ex - 2En, Ex + 2En]		[Ex - 3En, Ex + 3En]	
0	0.1	0.01	49.957	1.629	68.479	1.554	95.248	0.661	99.646	0.183
2	0.1	0.01	50.035	1.573	68.475	1.484	95.267	0.676	99.630	0.185
4	0.5	0.1	51.095	1.568	69.127	1.457	94.661	0.683	99.392	0.252
8	10	1	50.203	1.550	68.560	1.509	95.261	0.669	99.638	0.189
0	0.1	1	16.261	1.174	21.764	1.309	34.810	1.538	44.679	1.584
2	2	5	39.922	1.543	50.523	1.578	70.709	1.428	81.879	1.197
4	0.5	2	30.340	1.457	39.234	1.577	58.077	1.549	70.096	1.452
8	0.1	0.5	26.271	1.373	34.324	1.489	51.940	1.541	63.768	1.502

Figure 5(b) is listed in Figure 5(c), which is with an enlarged scale. As can be seen from Figure 5(c), the edge or border is *soft* or uncertain as well as frayed (referred to in [8]). In the proposed method, the default value of density factor is fixed as $\rho = 0.01$.

Furthermore, we analyse the reason to fix the parameter in detail. Given the desired height h and width w , the number of reference points can be determined according to the density factor ρ , and then the average area of each Voronoi polygon can be calculated. We set the related parameters as $h \in \{32, 64, 128, 256, 512, 1024, 1440, 1920\}$, $w \in \{32, 64, 128, 256, 512, 768, 900, 1080\}$, and $\rho \in \{0.001, 0.005, 0.01, 0.1, 0.5\}$. As can be seen from Figure 5(e), for each ρ , the average area of Voronoi polygon is increased with a larger h or w , and it shows a tendency toward stabilization. Also, we investigate the combined influence of image size (hw as one index) in log-log scale coordinates system; the similar result can be obtained from Figure 5(f). With image size varying from 1 to 10^4 , the average area of Voronoi polygon is increased with a lower ρ . However, too large polygon would result in less aesthetic effect. According to the following groups, we generate a result image with default size of 128×128 , then the area of Voronoi polygon is about 100 dots with the total area 16384 dots, and the proportion of each polygon shares about 6%. Therefore, we fix the density factor $\rho = 0.01$ by keeping a good balance between aesthetic effect and time complexity. Certainly, the density factor ρ is still an open parameter. For a given image size, one can estimate and obtain the optimal ρ according to Figures 5(e) and 5(f).

5.1.2. Group 2. In this group, we investigate the intensity factor τ , which determines the number of the removed cloud drops. Two aesthetic patterns with unsupervised decomposition are generated using $\tau = 0.0, 0.9406$, respectively. The results and the corresponding details are shown in Figure 6. With $\tau = 0.0$, none of cloud drops are removed from the set of sample points; the result image in Figure 6(a) is very delicate and with very few black backgrounds, and the Voronoi borders in Figure 6(b) show a similar effect of frosted glass. With $\tau = 0.9406$, majority of cloud drops are removed;

the result in Figure 6(c) is relatively rough, and there are several cavities near the boundaries, as shown in Figure 6(d).

In the normal cloud model, each cloud drop contributes the concept differently. In Figure 6(e), we present a visual result from a statistical view. Given $Ex = 0, En = 0.1, He = 0.01$, we generate 1000 cloud drops, and the procedure runs 1000 times. We count the number of cloud drops falling into a given interval, such as $[Ex - 0.67En, Ex + 0.67En]$, $[Ex - En, Ex + En]$, $[Ex - 2En, Ex + 2En]$, and $[Ex - 3En, Ex + 3En]$. The overall result is listed in Figure 6(e). Although there is a little difference on the results of 1000 runs between each other, the statistical feature of each run is absolutely similar. One can observe that about 50% cloud drops lie in the interval $[Ex - 0.67En, Ex + 0.67En]$, and most of cloud drops (with the proportion of about 70%) are in the interval $[Ex - En, Ex + En]$.

In addition, we repeat this procedure with three other cloud models, $Ex = 2, En = 0.1, He = 0.01$, $Ex = 4, En = 0.5, He = 0.1$, and $Ex = 8, En = 10, He = 1$. The statistical results are listed in Table 1. With a proper He/En , the distributing proportion of cloud drops in each interval is generally the same.

In fact, the Gaussian function satisfies *three-sigma rule*, and, as a consequence, normal cloud model has *3En rule*. The majority of cloud drops lie within the interval $[Ex - 3En, Ex + 3En]$, and, specifically, cloud drops within the interval $[Ex - 0.67En, Ex + 0.67En]$, called the backbone elements, only account for 22.33% of the universe set but contribute 50% of the cloud model. Then, cloud drops within the interval $[Ex - En, Ex + En]$ make up 33.33% of the universe and contribute 68.26% of the cloud concept. For these cloud drops, the threshold of the certainty degree is 0.6065, that is, the intensity factor in the proposed algorithm. Additionally, about 11.11% of cloud drops are within $0.33En$ away from Ex and contribute 25.86% of the cloud concept; then the intensity factor is 0.9406. Similarly, other cases can be also calculated. To obtain the universal aesthetics, the default value of intensity factor is fixed as $\tau = 0.6065$ in the proposed method.

While with an extreme He/En , the distributing proportion of cloud drops is quite different, as shown in the latter four rows of Table 1. In this condition, the corresponding

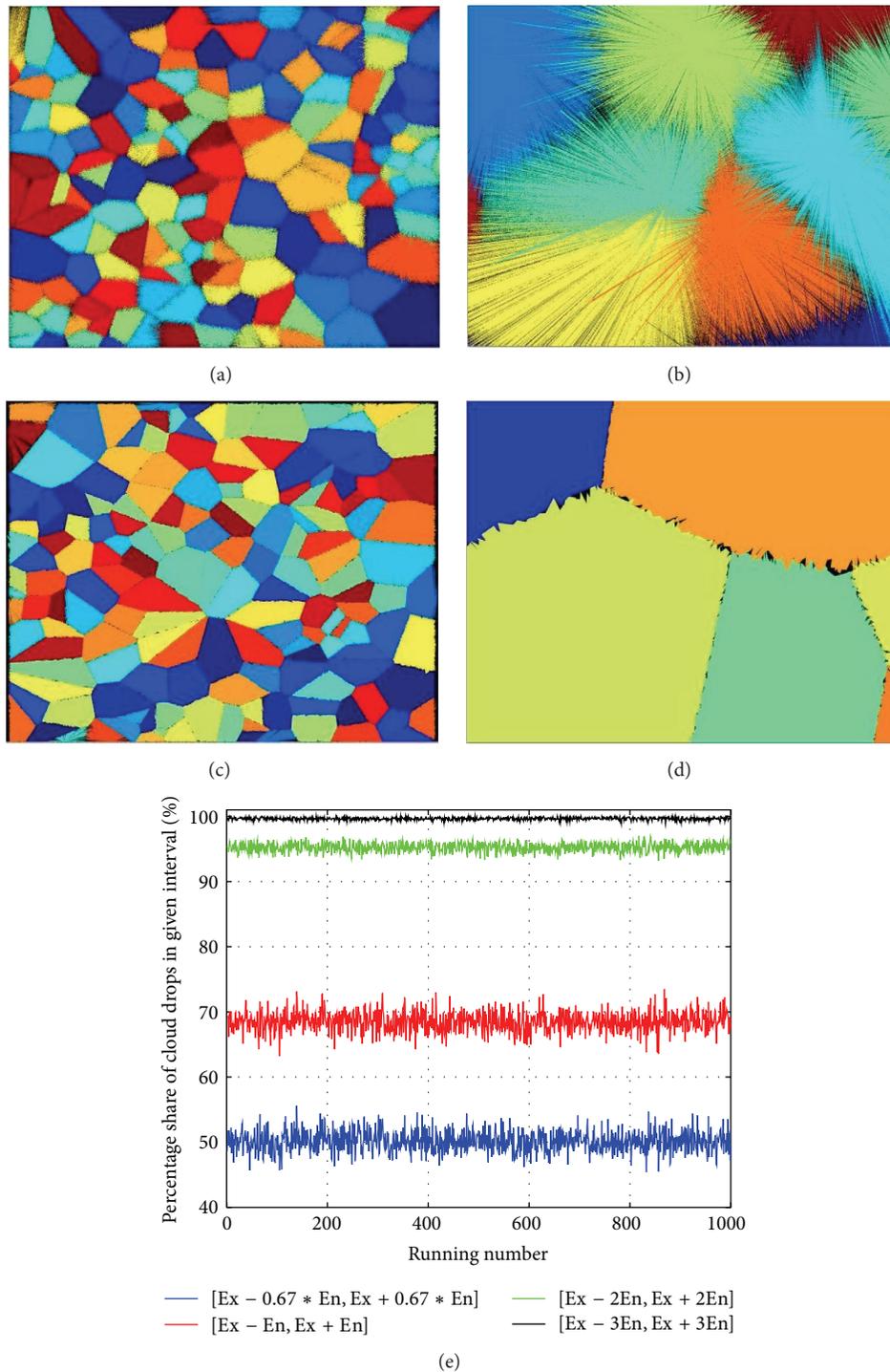


FIGURE 6: Rendering with various degrees: (a) $\tau = 0.0$, (b) the detail with an enlarged scale for (a), (c) $\tau = 0.9406$, (d) the detail with an enlarged scale for (c), and (e) statistical result on cloud drops and their contribution.

cloud model is atomized. In Group 5, we will further discuss this condition.

5.1.3. *Group 3.* In this group, we try to provide a study on the color schemas for the proposed method. All the above results used the color schema of the Jet colormap and the black

background, which is default for the unsupervised decomposition. Besides that, the summer colormap, the gray colormap, and the white background are also tested. Three aesthetic patterns are generated, as shown in Figure 7, and the other three results (EPS results) are attached in the Supplementary Material available online at <http://dx.doi.org/10.1155/2016/9837123>.

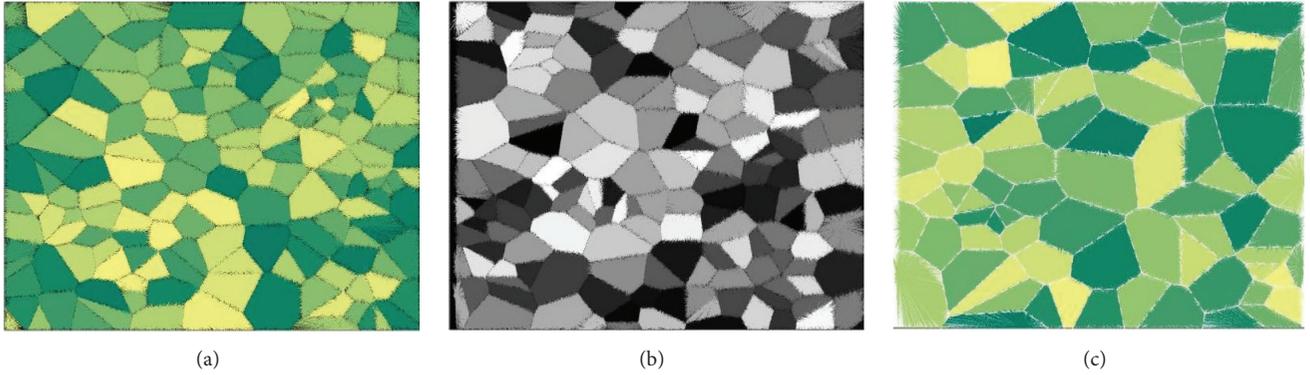


FIGURE 7: Rendering with different color schemes: (a) the summer colormap, (b) the gray colormap, and (c) the summer colormap with white background.

In most cases of image-guided decomposition, the distribution of Voronoi polygons is dense, and a too bright color is not conducive to the rendering of image objects; then the color schema is with the summer colormap and the black background, which is the default for the image-guided decomposition. In general, the experimental results suggest that the color schema is more important to image-guided decomposition than unsupervised decomposition. Certainly, any other color schemas are also optional, and it is an open problem.

5.1.4. Group 4. In this group, we investigate the number of the cloud drops n , which is also very important to the time performance of the proposed method, as mentioned above. Two aesthetic patterns with unsupervised decomposition are generated using $n = 10, 5000$, respectively. The results and the corresponding details are shown in Figure 8. With $n = 10$, sample points cannot well approximate the lines; the filling result in Figure 8(a) is too rough, and there are larger void spaces near the border, as the black patches in Figure 8(b). With $n = 5000$, the number of sample points in Figure 8(c) is excessive, the overlapping between each couple of polygons is very common, and then the uncertainty of the lines and polygons is weakened, as shown in Figure 8(d). The uncertain line and polygon would become the thick and “hard” one in the case of very large n .

Comparing Figure 8(b) with Figure 6(d), they are similar in aesthetics effect, although the former is used with a less number of the cloud drops and the latter is with a higher intensity factor. In fact, most of the cloud drops and the sample points are still removed even given a higher intensity factor and a larger number of the cloud drops, and it is equivalent to the case of a less number of the cloud drops.

Taking the above endpoints of a line segment as an example, that is, $\mathbf{P}_1 = (0, 1)$ and $\mathbf{P}_2 = (4, 9)$, the reference point is $\mathbf{P}_r = (5/4, -5/2)$, and we generate cloud drops with varied cloud model (the entropy $[1/px_r, 1/py_r]^T$ and the hyper-entropy $[0.1/px_r, 0.1/py_r]^T$). The procedure is repeated 1000 times, and the number of cloud drops increases 5 iteration by iteration. In other words, the number of cloud drops is varied from 1 to 5000. For each iteration, we record

the mean squared error (MSE) between the cloud drops and the real values in the line $\mathbf{P}_1\mathbf{P}_2$. The results are shown in Figure 8(e), and for the reference, the average MSE is also drawn. With the number of cloud drops less than 500, the MSE values are fluctuated violently, which leaves the uncertain process in an unstable state. Meanwhile, a larger number of cloud drops would result in to be more time-consuming according to the time complexity. Thus, a modest n about 500 to 1000 is beneficial. In the proposed method, the default value of the number of cloud drops is fixed as $n = 500$.

5.1.5. Group 5. In this group, we investigate the numerical characteristics of the cloud model, and an improvement is added into the proposed algorithm. Two aesthetic patterns with unsupervised decomposition are generated; the results are shown in Figure 9. Given both the entropy and the hyper-entropy $[0, 0]^T$, the cloud model has degenerated into a normal distribution; the uncertainty is completely evaporated and equivalent to the certain and accurate processing, as the traditional method does. From Figure 9(a), the aesthetic effect is identical with the classical Voronoi art.

From another perspective, cloud drops present a generalized Gaussian distribution when $He < En/3$; the corresponding cloud model is generally suitable for the representation of qualitative concept. Conversely, cloud drops appear in atomized state in case of $He > En/3$, the samples seriously deviate from Gaussian distribution, and then a cloud concept is difficult to reach and achieve. As shown in the latter four rows of Table 1, the distributing proportion of cloud drops is quite different from a normal cloud model. Thus, we should avoid this condition using parameter setting.

We present an example with both the entropy and the hyper-entropy $[5, 5]^T$, the cloud model has satisfied the atomized condition, and the result image has none of aesthetics. Sample points cannot effectively approximate the Voronoi polygons, and it fails to represent lines and fill polygons with uncertainty, as shown in Figure 9(b). From this perspective, an additional step should be introduced into Step 4 in Section 4, besides the intensity factor of the certainty degree, that is, filtering cloud drops and sample points outside the polygons. This step ensures that the proposed algorithm

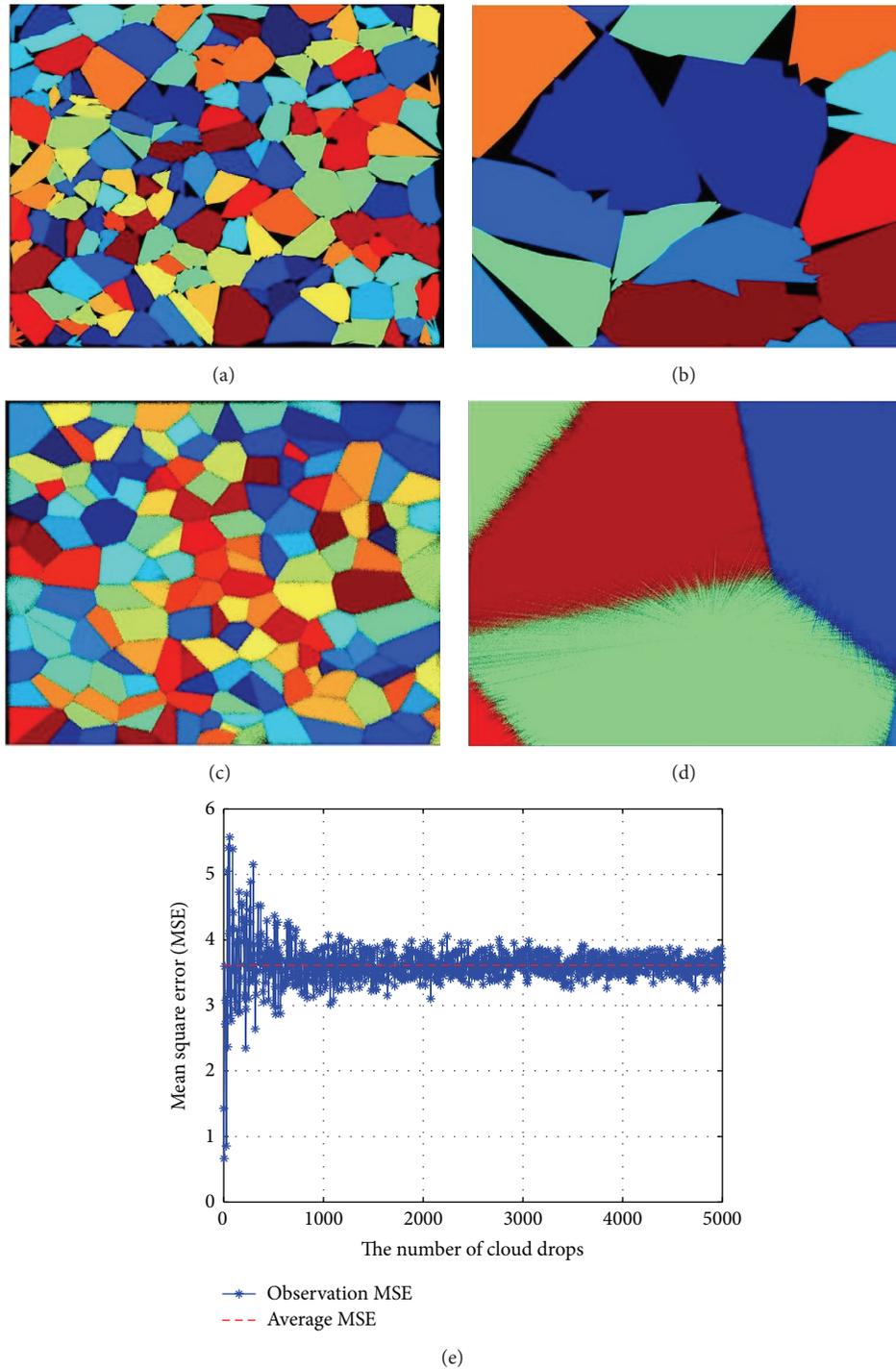


FIGURE 8: Rendering with various number of cloud drops: (a) $n = 10$, (b) the detail with an enlarged scale for (a), (c) $n = 5000$, (d) the detail with an enlarged scale for (c), and (e) mean squared error of cloud drops.

is robust even if involving invalid values of the entropy or the hyper-entropy. As a result, another aesthetic style is generated.

Furthermore, we generate another aesthetic pattern with the entropy $[0.1, 0.1]^T$ and the hyper-entropy $[0.01, 0.01]^T$; the result is shown in Figure 9(c). With the constant and

nonzero values of entropy and hyper-entropy, the rendering result is low-aesthetic, although the condition of the generalized Gaussian distribution is satisfied. Obviously, the result image is asymmetric, and its left is more delicate. This is because k_i and b_i are generated by constant numerical characteristics of cloud model, and these two parameters

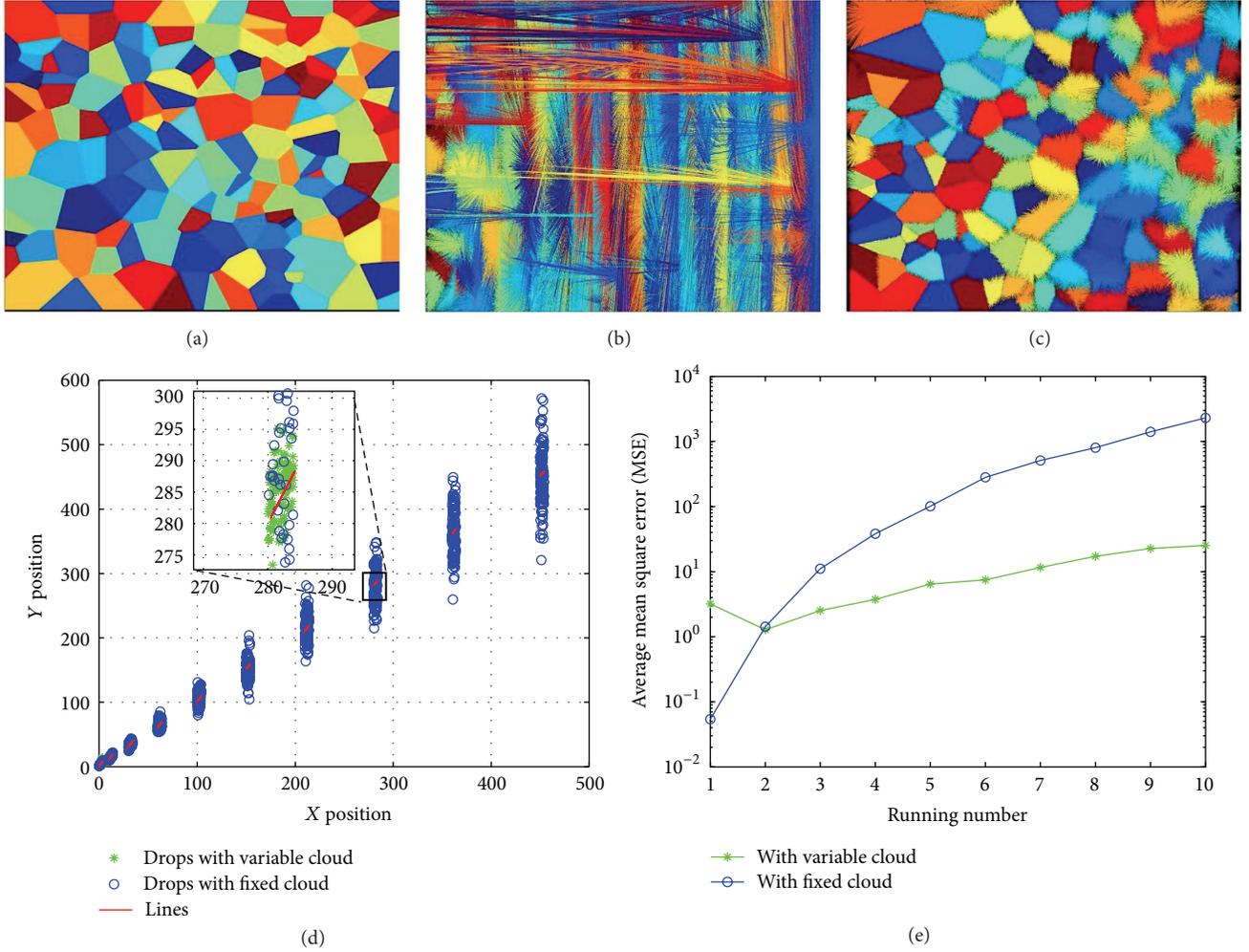


FIGURE 9: Rendering with various parameters of cloud model: (a) the degradation of cloud model, (b) the atomization of cloud model, (c) the constant and nonzero values for En, He, (d) uncertain line with fixed and variable cloud models, and (e) average MSE of fixed and variable models.

approximate the real values at different levels, and then the left sample points with smaller lx_i would be the lower deviation from the real line, according to $ly_i = k_i lx_i + b_i$. Otherwise, the right sample points would be the higher deviation. Thus, the right half plane appears rougher than the left. To void this problem, at least one of entropy and hyper-entropy is fixed as a reduction function related to the positions of reference points.

Considering $\mathbf{P}_1 = (0, 1)$ and $\mathbf{P}_2 = (4, 9)$ with the reference point $\mathbf{P}_r = (5/4, -5/2)$ as the starting, the expected values are calculated by (2), and then we generate 1000 cloud drops; half of them are from fixed cloud models with the entropy $[0.1, 0.1]^T$ and the hyper-entropy $[0.01, 0.01]^T$, and the other half are from variable cloud models with the entropy $[1/px_r, 1/py_r]^T$ and the hyper-entropy $[0.1/px_r, 0.1/py_r]^T$. This procedure is repeated 10 times with an incremental position, and the increment of one step is 10 on both x - and y -axis. Then, from left to right and from bottom to top, there are ten lines by ten runs. The positions of cloud drops

are drawn in Figure 9(d) as well as the lines. Only the lines and the drops from fixed cloud models are visible, while the drops from variable cloud models are generally not visible. For further observation, we also show an enlarged scale for the 8th line, and, in this state, we could vaguely see the drops from variable cloud models. That is to say, the drops from variable cloud models cluster around the represented line, and they are covered by those from fixed cloud models, which seriously spread along the line. With the increment of X and Y position, the distribution on drops from fixed cloud models is more dispersed. Additionally, we list the average MSE values between cloud drops and real values of lines. As can be seen from the Y -log system in Figure 9(e), with a smaller coordinate values, the average MSE value of fixed cloud model seems to be less than that of variable cloud model, while, with the increase of coordinate values, variable cloud models achieve absolutely smaller average MSE values. When the X and Y position is about 500, the fixed cloud models produce 1000 times average MSE value as much as

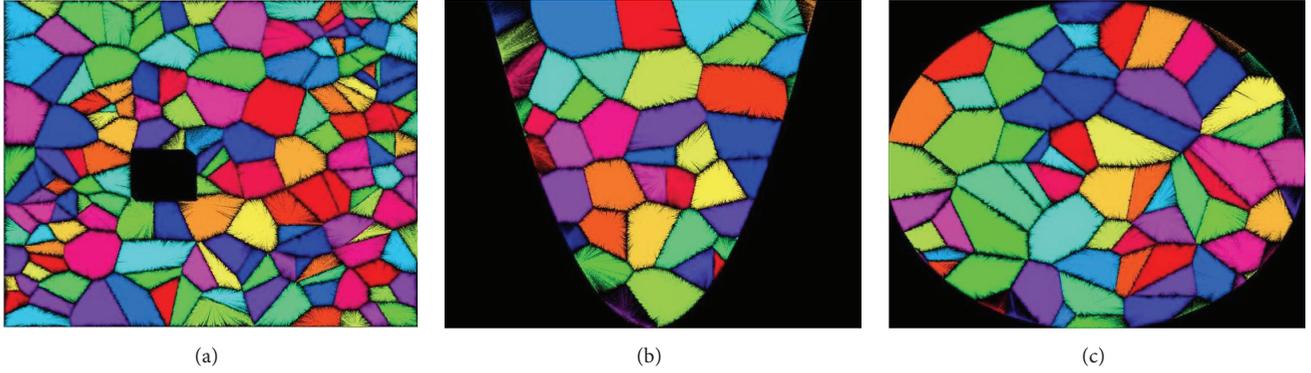


FIGURE 10: Rendering with various shapes: (a) the rectangle with an inside hole, (b) the area with a parabolic outer boundary, and (c) a circular region.

that of variable cloud model. Therefore, the default setting in the proposed method is with the entropy $[1/px_r, 1/py_r]^T$ and the hyper-entropy $[0.1/px_r, 0.1/py_r]^T$.

5.1.6. Group 6. In this group, we render three Voronoi-based images with various shapes. For all of the patterns, the size is as $h = w = 128$. The first one is a rectangle and its central part is blown-up, determined by $(40, 70), (40, 50), (60, 50), (60, 70)$. The decomposition result is listed in Figure 3(c), and the final pattern is shown in Figure 10(a), with a high and aesthetical quality. The second is an irregular area surrounded by the parabola $y = 0.05(x - 64)^2 + 1$ and the line $y = 128$. The output in Figure 10(b) remains generally satisfactory, with only fewer faulty borders because of the shape constraint. The last is a disk determined by $(x - 64)^2 + (y - 64)^2 \leq 64^2$, and the result is also acceptable, as shown in Figure 10(c). In summary, these aesthetic patterns indicate that the proposed algorithm can tolerate various shape constraints and generate a relatively comfortable visual effect. In theory, any area with any shape constraint can obtain similar stylized image through the proposed method, which provides possible applications in various fields, such as tessellation-based image stylization, image mosaic, and Voronoi art.

5.1.7. Group 7. Using the proposed algorithm, four images are processed, and the result images are listed in Figure 11, attached by the original images. As shown in Figures 11(a) and 11(d), the positions of the background pixels are as the coordinates of the reference points, while those of the object pixels are in Figures 11(c) and 11(b). To be clear, there is the result of the IE logo image in Figure 11(a), whose half-tone image is as mentioned above in Figure 3(b), and another result is attached in the Supplementary Material. Summarily, these results are favourable, and all of them demonstrate the high aesthetics effect, which is similar to the stroke based painterly rendering, but different from the existing methods.

5.2. Visual Comparisons. This subsection provides a qualitative comparison of our output against the relative methods. All of methods, including the traditional method, the FCD

TABLE 2: Average running times for various image sizes.

Size		Running times (s)		
h	w	Ours	Classical method	FCD
32	32	0.386	0.245	7.533
64	64	0.765	0.295	34.209
128	128	2.412	0.418	265.404
256	256	8.351	1.006	2478.647
512	512	34.199	4.416	>14400
1024	768	109.572	22.586	>14400
1440	900	199.271	51.778	>14400
1920	1080	352.121	115.754	>14400

method [8], and ours, are implemented in Matlab and performed on a 2.4 GHz Core i7 PC with 8 GB RAM. As shown in Figures 12(b) and 12(e), the Voronoi-based rendering by the classical method is very crisp, but without uncertainty, while the results by our method and the FCD method are soft and rough, as shown in Figures 12(a) and 12(c). Even so, there is the obvious difference between them. Result in Figure 12(f) by the FCD method appears in rotational symmetry, where each reference point is the center of symmetry. Although it is regular and organized, the handling of the uncertainty is too rigid, and there are signs that the result is man-made and artificial. Compared with the FCD method, our method in Figure 12(d) handles the uncertainty using cloud model and makes the result closer and more harmonious with the nature. In fact, the user study in the following is also indicating that our results are more like an effect with the frayed or cracked soil. However, we do not mainly focus on evaluating aesthetics of the generated patterns, and our method still captures the uncertainty better than the existing methods.

5.3. Time Performance Analysis. In this subsection, we investigate the running time of the proposed method. We run 10 times to get mean values for each group of parameters. The running times are listed in Table 2. With the increase of the size of the input image, the generation time of all methods have increased. From the time cost's point of view,

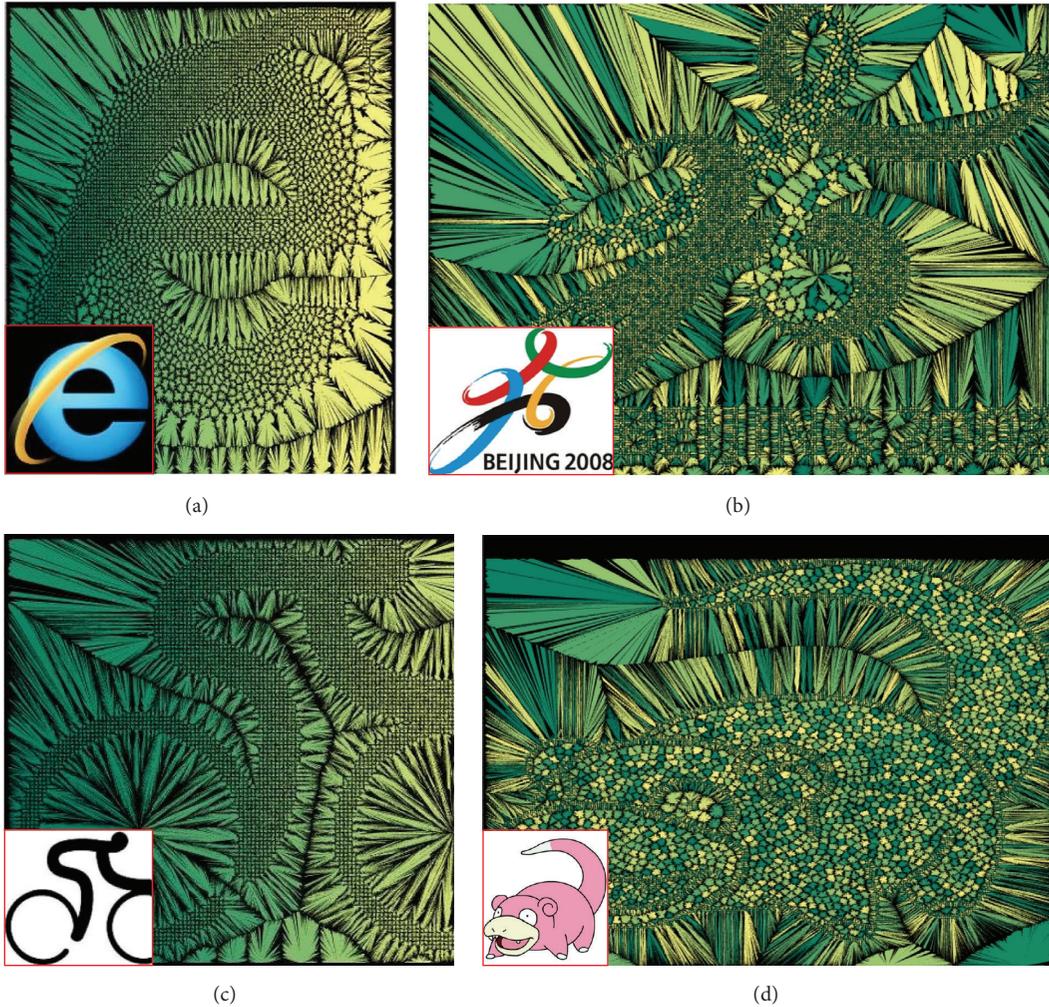


FIGURE 11: Voronoi stylization for images: (a) the IE logo, (b) the Beijing Olympic logo, (c) the Olympic cycling logo, and (d) a cartoon picture.

the classical method is the fastest, but noneffective, and none of the uncertainty is handled, which can be seen from the above sections. For a novel visual effect, our method and the FCD method spend much time to replace the rendering operation in the traditional method, and the running time increases inevitably. The FCD method is the most time-consuming, even its time cost is about 50 times as much as the cost of the proposed method. With a larger size of the input image, our method is more time-saving than the FCD method. Specifically, the FCD method cannot obtain the output spanning 4 hours in the case of 512×512 and we label the time > 14400 seconds as shown in Table 2.

The results of Table 2 are also supported by the theoretical analysis on the time complexity. The FCD method takes the pixel as the basic processing unit, the outer loop iterates hw times, and the innermost loop to find the reference point for each pixel scans each Voronoi polygon, which costs the time $O(N)$ at least. Thus, the time cost of the FCD method is about $O(Nhw)$ ($O(\rho h^2 w^2)$ for the unsupervised decomposition), which is the quadratic complexity of the size of the input image, and more time-consuming than ours $O(hw)$.

To further investigate the time performance, we provide an analysis on the change of time costs varied with image size (i.e., hw). The original data is from Table 2. Using the `polyfit()` function in Matlab, we try to construct a straight line that has the best fit to these data points. The results are shown in Figure 13, and the fitting result of the FCD method has been moved since it is far from a straight line, suffers a startling deviation, and seriously influences the illustration of the results by ours and the classical method. Clearly, our method shows a good performance to line fitting. Theoretically, in some extension, it has a significant linear correlation with the image size.

It would be specially mentioned that hw here corresponds to the size of the input image for image-guided decomposition, but not the image resolution or size of the generated result. Our method uses the technique of vector drawing and can be saved as various vector format, which can be chosen by users; for example, the scanning precision of EPS (Encapsulated PostScript) reaches 600 DPI (Dots Per Inch). In a way, the proposed method is efficient from the view of running time and can be approximately satisfied by

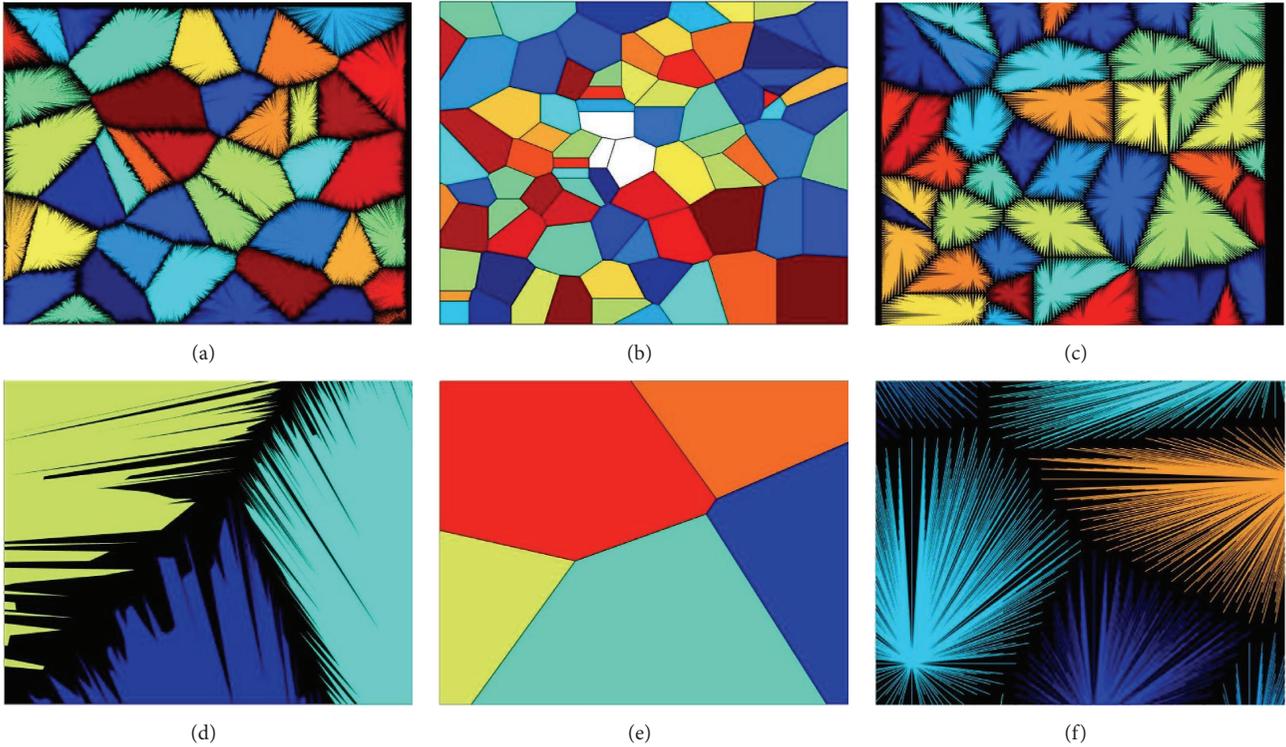


FIGURE 12: Results by various methods: (a) ours, (b) the classical method, (c) the FCD method, (d) the detail with an enlarged scale for (a), (e) the detail with an enlarged scale for (b), and (f) the detail with an enlarged scale for (c).

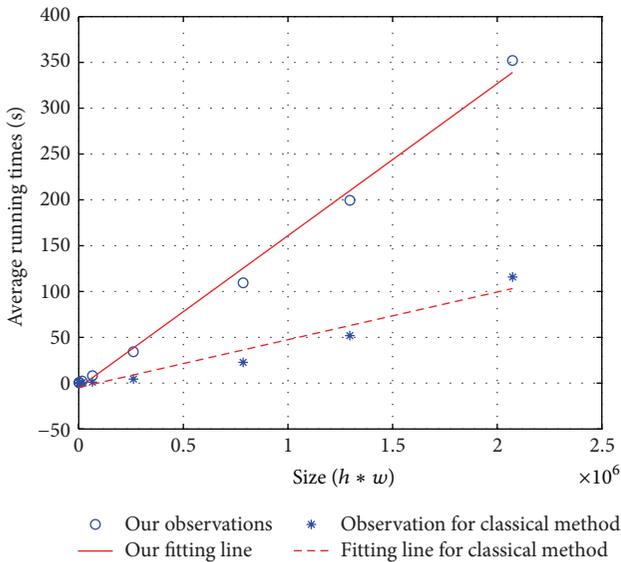


FIGURE 13: Line fitting on time performance of our method and the classical method.

the need of the real-time applications, because our method runs with less time cost and the generated visual results are acceptable.

5.4. Quantitative Comparisons. In this subsection, we use five indexes to provide a quantitative comparison, including

TABLE 3: Quantitative evaluation results.

Index	Ours	Classical method	FCD
Index_B	0.8638	0.6696	0.7265
Index_F	0.7006	0.8673	0.9276
Index_S	0.8775	0.7166	0.7634
Index_G	0.6881	0.3863	0.6383
Index_K	0.7581	0.6693	0.5565

Benford's law index_B [27], fractal dimension index_F [28], Shannon entropy index_S [29], global contrast factor index_G [30], and Kolmogorov complexity index_K [31]. The result images in Section 5.2 are involved, and the total number is 200, in which our method, the classical method, and the FCD method account for 40%, 40%, and 20%, respectively. Because of randomness, we record the score of each image and then average them according to the used method. For comparison purposes, each row of the score of each measure is normalized.

The quantitative evaluation results are listed in Table 3. Benford's law index_B is the measure of the distribution of intensity of pixels. The index_B score gradually reduces in the order of the classical method, the FCD method, and our method, and the corresponding average deviation becomes larger and larger. Images with a higher index_F were considered complex, and images with a lower one were more uncertain. Our results show the uncertainty with the lowest index_F. The Shannon entropy index_S rewards images with

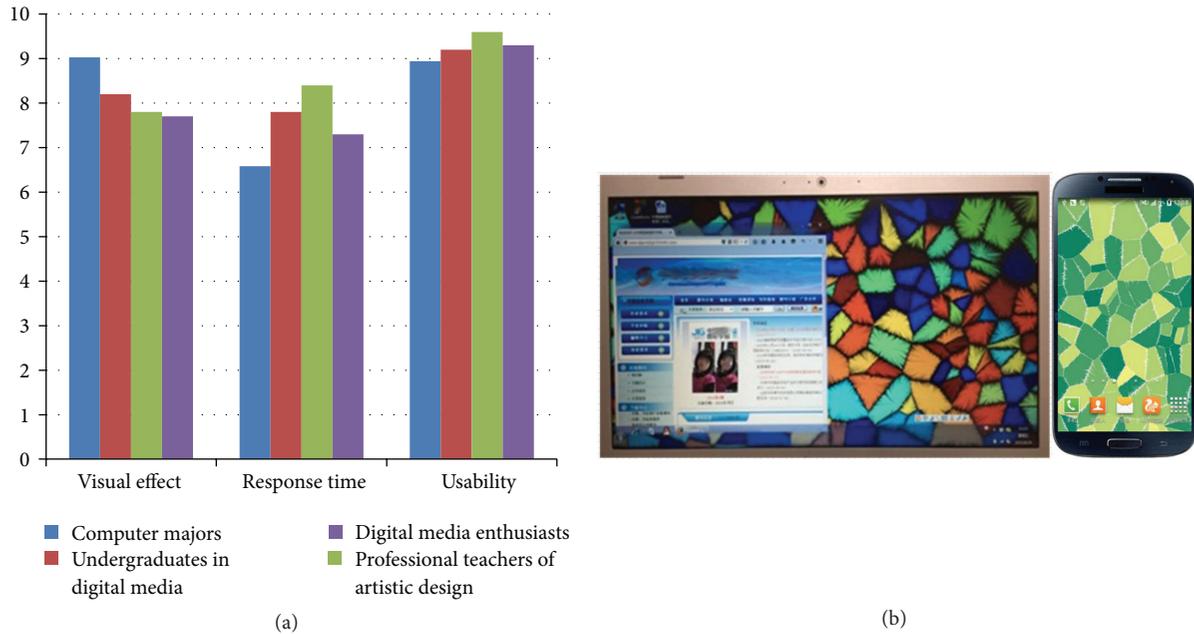


FIGURE 14: User survey results: (a) the scores from user survey and (b) the screenshot using the proposed aesthetic image.

a uniform distribution of brightness values, and our results obtain the highest of all, while the classical results are the lowest. From this perspective, our method generates the patterns with more uncertainty. The global contrast factor calculates and values contrast on various resolutions of an image, and a lower contrast index $_C$ reflects a lower aesthetic effect. The index $_C$ score of the proposed method ranks first, followed by the FCD method and the classical method, and it is indicated that our method generates the results with the highest aesthetics. The Kolmogorov complexity index $_K$ and the Shannon entropy index $_S$ complement each other; their results reflect the similar meaning. Overall, our method is certainly more efficient from the perspective of uncertain or frayed visual effect, since it achieves a higher index $_B$ (the distribution of intensity of pixels is more deviated from Benford's law), a lower index $_F$ (more uncertain), a higher index $_S$ and index $_K$ (nonuniform distribution of brightness values), and a higher index $_C$ (higher aesthetic effect).

However, quantitative measure of aesthetics remains a challenging task, although for specific image applications some recent progress has been made in this direction. Still, the main purpose by the quantitative comparison in Table 3 is to assess the difference between pair of the aesthetic patterns generated by different methods, and it should not be strictly limited to scores of the various styles. In other words, the quantitative measure of aesthetics is a necessary and beneficial complement to the visual comparison and the user study. For a full evaluation of the new artistic preprocessing technique, we design a user study involving both amateur users with some art knowledge and professional art students, as the following subsection.

5.5. The User Study. Humans are an integral part of defining what is aesthetic; thus we employ a user study to assess the

performance in this subsection. Using the provided GUI, the users control the type of Voronoi decomposition and fix the parameters. Each user is required to obtain 50 images independently. Then, the study asked the users to score each image according to a sliding scale of measure value, using a Likert scale in the range [0 10], and 10 means a perfect result. A total of 100 users participated in this investigation; they volunteered from our university, and there are 87 computer majors, 5 undergraduates in digital media, 5 professional teachers of artistic design, and 3 digital media enthusiasts. The evaluation covers threefold: (1) visual effect: we asked the user to score the aesthetic feeling of each result, and 10 means the most aesthetic. (2) Response time: each participant is asked to score the satisfaction of running time, and 10 means the most satisfied. (3) Usability: the users choose a score to measure the possibility of the generated patterns used in real applications, and 10 means the most possible.

Figure 14(a) shows a summary of the survey results, and Figure 14(b) shows two most pleased results, used in computer background and mobile wallpaper. The computer majors score the highest visual effect, and the average score of the visual effect is more than 7.5. For the subjectivity of the aesthetics problem, our work is only based on the opinions of our colleges and students who have seen our patterns. The users are generally satisfied with the performance of the proposed method, and majority of them said that our results were looking nicely and more like an effect with the frayed or cracked soil. The response time cannot perfectly meet the most demanding of the computer majors, that mark the lowest for this measure. The average score of the response time is more than 6.5, and the professional teachers of artistic design show that the response time of the proposed iVPC method is acceptable, since they often encountered the longer wait at work. The average score of the usability is above 8,

and the computer majors are pleased to take these results as wallpapers (see Figure 14(b)), and the others also think the generated patterns are helpful to their daily designs. In summary, the user study shows a good performance of the proposed method.

6. Summary and Conclusion

In this paper, an uncertainty algorithm based on cloud model for image-guided Voronoi aesthetic patterns has been proposed. As a computational intelligence tool, cloud model handles the uncertainty more completely and more freely, and it cannot be considered as randomness compensated by fuzziness, fuzziness compensated by randomness, second-order fuzziness, or second-order randomness. To obtain the default parameters, we conduct seven groups of experiments to test the proposed method. Using both visual and quantitative comparisons, we prove the efficacy of the proposed method using two groups of experiments. Compared with the related methods, experimental results show that the Voronoi-based aesthetic patterns with *soft* borders can be generated by using the new technique. Also, the proposed iVPC algorithm has better performance on running time, and its time complexity is approximately linear related to the size of the input image. The real engineering applications using the proposed algorithm, such as home decoration and clothing design, are under the investigation and implementation and would be reported in the future. Nevertheless, we still see a good potential that the soft rendering technique could be applied to diagrams from computer-based visualization and graphics.

There are a couple of issues that should be considered in the proposal: (1) the proposed method used a considerable number of line drawing, and the vector output consumes more file storage space. For the input image with size of 256×256 , the output file needs about 30MB. Thus, how to generate other formats, for example, SVG (Scalable Vector Graphics), and reduce the file space is one of the useful extensions. (2) The proposed method is implemented in Matlab, and other languages for engineering applications should be employed, for example, C++ and Java. Then, how to integrate with the existing software is another feasible direction. The extension of the technique is currently under investigation and will be reported later.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

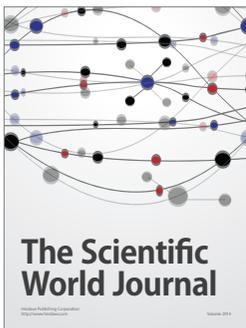
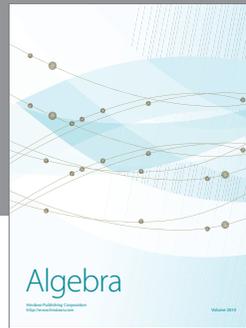
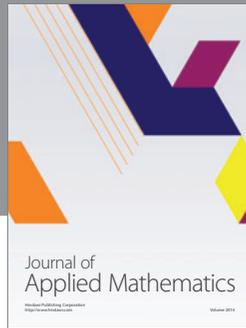
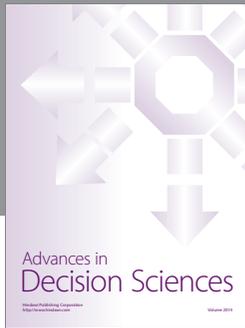
Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (under Grant no. 61402399), by the National Key Basic Research and Development Program (no. 2012CB719903), by the Foundation for Distinguished Young Teachers in Higher Education of Guangdong, China (no. YQ2014117), and by the Foundation of Humanities and Social Sciences Research in Ministry of Education, China (no. 14YJCZH161).

References

- [1] M. David, "Image-guided fracture," in *Proceedings of the Graphics Interface Conference*, pp. 219–226, Canadian Human-Computer Communications Society, Victoria, Canada, May 2005.
- [2] F. Hoenic, "Defining computational aesthetics," in *Proceedings of the 1st Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 13–18, 2005.
- [3] P. Barile, V. Ciesielski, K. Trist, and M. Berry, "Animated drawings rendered by genetic programming," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '09)*, pp. 939–948, New York, NY, USA, 2009.
- [4] D. L. Atkins, R. Klapaukh, W. N. Browne, and M. Zhang, "Evolution of aesthetically pleasing images without human-in-the-loop," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, 2010.
- [5] E. Den Heijer and A. E. Eiben, "Investigating aesthetic measures for unsupervised evolutionary art," *Swarm and Evolutionary Computation*, vol. 16, pp. 52–68, 2014.
- [6] C. M. Fernandes, A. M. Mora, J. J. Merelo, and A. C. Rosa, "Photorealistic rendering with an ant algorithm," in *Computational Intelligence: International Joint Conference, IJCCI 2012 Barcelona, Spain, October 5–7, 2012 Revised Selected Papers*, vol. 577 of *Studies in Computational Intelligence*, pp. 63–77, Springer, New York, NY, USA, 2015.
- [7] S. Kim, R. Maciejewski, A. Malik, Y. Jang, D. S. Ebert, and T. Isenberg, "Bristle maps: a multivariate abstraction technique for geovisualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1438–1454, 2013.
- [8] B. Michael, V. Corinna, and W. Daniel, "Frayed cell diagrams," in *Proceedings of the ACM Workshop on Computational Aesthetics (CAe '14)*, pp. 93–96, British Columbia, Canada, August 2014.
- [9] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.
- [10] C. S. Kaplan, "Voronoi diagrams and ornamental design," in *Proceedings of the 1st Annual Symposium of the International Society for the Arts, Mathematics, and Architecture*, pp. 277–283, 1999.
- [11] L. Golan, "Segmentation and symptom created for british arts quarterly zoo," <http://www.flong.com/projects/zoo>.
- [12] M. David, "A stained glass image filter," in *Proceedings of the 14th Eurographics Workshop on Rendering*, pp. 20–25, Eurographics Association, Leuven, Belgium, 2003.
- [13] W. Brian, "Determining an aesthetic inscribed curve," in *Proceedings of the 8th Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, Eurographics Association*, pp. 63–70, Annecy, France, 2012.
- [14] M. Jooyandeh, A. Mohades, and M. Mirzakhah, "Uncertain voronoi diagram," *Information Processing Letters*, vol. 109, no. 13, pp. 709–712, 2009.
- [15] D. Sun and Z. Hao, "Group nearest neighbor queries based on voronoi diagrams," *Computer Research and Development*, vol. 47, no. 7, pp. 1244–1251, 2010.
- [16] X. Wang, H. Zhang, Q. Fang, Y. Ge, and Z. Wang, "Research on probabilistic Voronoi model and algorithm for coverage in WSN," *Chinese Journal of Sensors and Actuators*, vol. 25, no. 5, pp. 702–706, 2012.
- [17] T. Isenberg, "Visual abstraction and stylisation of maps," *The Cartographic Journal*, vol. 50, no. 1, pp. 8–18, 2013.

- [18] D. Li and Y. Du, *Artificial Intelligence with Uncertainty*, Chapman & Hall, Boca Raton, Fla, USA, 2007.
- [19] D. Li, C. Liu, and W. Gan, "A new cognitive model: cloud model," *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 357–375, 2009.
- [20] K. Qin, K. Xu, F. Liu, and D. Li, "Image segmentation based on histogram analysis utilizing the cloud model," *Computers and Mathematics with Applications*, vol. 62, no. 7, pp. 2824–2833, 2011.
- [21] G. Wang, C. Xu, and D. Li, "Generic normal cloud model," *Information Sciences*, vol. 280, pp. 1–15, 2014.
- [22] T. Wu, J. Xiao, K. Qin, and Y. Chen, "Cloud model-based method for range-constrained thresholding," *Computers & Electrical Engineering*, vol. 42, pp. 33–48, 2015.
- [23] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [24] D. Oliver and I. Tobias, "Halftoning and stippling," in *Image and Video-Based Artistic Stylisation*, pp. 45–61, Springer, Berlin, Germany, 2013.
- [25] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *Proceedings of the Society of Information Display*, pp. 75–77, 1976.
- [26] S. Jakob, Constraining the vertices of a voronoi decomposition, <http://cn.mathworks.com/matlabcentral/fileexchange/34428voronoilimit>.
- [27] J.-M. Jolion, "Images and benford's law," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 1, pp. 73–81, 2001.
- [28] B. Spehar, C. W. G. Clifford, B. R. Newell, and R. P. Taylor, "Universal aesthetic of fractals," *Computers and Graphics*, vol. 27, no. 5, pp. 813–820, 2003.
- [29] J. Rigau, M. Feixas, and M. Sbert, "Informational aesthetics measures," *IEEE Computer Graphics and Applications*, vol. 28, no. 2, pp. 24–34, 2008.
- [30] K. Matkovic, L. Neumann, A. Neumann et al., "Global contrast factor—a new approach to image contrast," in *Proceedings of the 1st Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 159–168, 2005.
- [31] J. Rigau, M. Feixas, and M. Sbert, "Conceptualizing birkhoff's aesthetic measure using shannon entropy and kolmogorov complexity," in *Proceedings of the 3rd Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 105–112, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

