IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 18, 2019, accepted March 31, 2019, date of publication April 3, 2019, date of current version April 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909159

Image Data Field-Based Method for Stippling

TAO WU[1,2], (Member, IEEE), JUNJIE YANG[2], AND XUE QIN[3]

[1]School of Information Engineering, Lingnan Normal University, Zhanjiang 524048, China
[2]Guangdong Engineering and Technological Development Center for E-learning, Zhanjiang 524048, China
[3]College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China

Corresponding author: Tao Wu (wutao@whu.edu.cn)

This work was supported in part by the Natural Science Foundation of Guangdong under Grant 2017A030307030, in part by the National Natural Science Foundation of China under Grant 61402399, and in part by the Provincial Joint Fund of Guizhou under Grant LH20147631.

ABSTRACT Computer-based generative art is in the ascendant and widely applied in media contexts such as web design, game resource, and digital entertainment. The process of creating non-photorealistic rendering images can be enjoyable if some useful and efficient methods are involved. This paper presents an automatic and fast method for digital stippling, which produces stipple renderings from photographs. To achieve this, we employ a novel mechanism based on the data field of an original image. Our inspiration is originated from physical fields. The used data field for an image can keep the balance between spatial and grayscale information in the local neighborhood by potential function, as well as the balance between the local information and the global trend. There involve three major steps in the proposed approach, including the generation of image data field, the reduction of potential centers, and the artistic renderings. The second step contains two types of reduction strategies, potential center elimination for stipple placement, and potential center cutting for mosaic construction. A number of experiments, both visual comparisons and quantitative comparisons, are performed. The results show the feasibility and the efficiency of the proposed method and suggest that the proposed method can generate appealing stipple or mosaic images. This would inspire graphic designers who may be interested in the stippled image automatically that is similar to images created by the artist.

INDEX TERMS Computational aesthetics, data field, digital stippling, halftoning, non-photorealistic rendering.

I. INTRODUCTION

Stippling is an artistic technique that constructs an image from a large number of small dots or marks visually. These dots or marks can vary in size, shape, density, and arrangement to create the illusion of different texture, tone, and shape, thus shades of gray are perceived within the stippled image. As such, stippling is capable of capturing a very wide dynamic range of tones, from white to black, and irregularities in shape can all lead to slight variations, giving rise to various styles of stippling. In general, stippling has been applied to various fields, such as digital entertainment, art emulation and robot art [1]–[4].

In the computer graphics community, many computer-aided algorithms have been developed in an attempt to approximate hand-drawn stippling, most of which employed Lloyd's algorithm [5] or its variants to provide regular dot spacing. Pioneering works by Deussen *et al.* [6] and Secord [7] did the excellent job of tone reproduction with stipples, and many other methods have since taken advantage of it. Deussen *et al.* [6] first used Lloyd's algorithm to create computer generated pen-and-ink illustrations by the simulation of stippling. An initial dot set is generated using a halftoning method and then processed by a relaxation method based on Voronoi diagrams. Later, Secord [7] enhanced this mechanism to generate stipple drawings from grayscale images using weighted centroidal Voronoi diagrams (WVS for short). Hiller *et al.* [8] presented an extension of Lloyd's approach, in which arbitrary shapes can be used in place of dots. Balzer *et al.* [9] also introduced a variant Lloyd's method of capacity-constrained Voronoi tessellation. Deussen *et al.* [10] proposed an adaptive version of Lloyd's algorithm based on Linde-Buzo-Gray algorithm in vector quantization (LBG for short). The above two representative methods based on the Lloyd's algorithm (that is the WVS method and the LBG method) will be

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

involved into a comparative study with our method (see Section IV).

Furthermore, more researchers explored this problem using various other methods. Mould [11] transfromed the input image into a weighted graph and provided a stipple placement approach using distance in the weighted graph, but it is not efficient enough to solve some large-scale problems. Kopf *et al.* [12] introduced a recursive Wang tiles-based technique for rapidly generating large point sets that possess a blue noise Fourier spectrum and high visual quality, and the technique can produce millions of stipple points per second. Example-based approach by Kim *et al.* [13] focused on stippling as an artistic style and discussed the technique for capturing and reproducing stipple features unique to an individual artist. Martín *et al.* [14], [15] also presented an example-based approach to synthesizing stipple illustrations for static 2D images that produces scale-dependent results appropriate for an intended spatial output size and resolution. Li and Mould's structure-dependent approach [16] concentrated on structure preservation by using a priority-based scheme that treats extremal stipples first and preferentially assigns positive error to lighter stipples and negative error to darker stipples, emphasizing contrast. Recently, Ma *et al.* [17] used incremental Voronoi sets and a real-time parallelized rejection strategy to fast produce digital stipple, which makes a fair balance on result quality and computational efficiency.

Comprehensive overviews and comparative studies of digital stippling can be found in the literature [2], [3]. Martín *et al.* [4] provided a recent version, in which Kopf *et al.*'s Recursive Wang Tiles method (RWT for short) [12] and Li and Mould's Structure-Aware Stippling method (SAS for short) [16] are both taken as state-of-art algorithms. These two methods will be also involved into a comparative study with our method (see Section IV).

From the evaluation perspective, Maciejewski *et al.* [18] quantified the difference between a computed-generated stippling and hand-drawn stippling of the same object, and used a metric based on the Gray Level Co-ocurrence Matrix (GLCM for short) to explore the stipple aesthentics. The stipple dot distribution can be studied from three different aspects, that is, contrast, energy, and correlation. Another analysis technique was used by Li and Mould [16] to assess the quality of their SAS method. This approach relied on the Structural SIMilarity index (SSIM for short) [19], as well as the measurement of tone using Peak Signal-to-Noise Ratio (PSNR for short). The above evaluation indexes will be involved into our comparative study as the quantitative analysis methods (see Section IV).

In addition, correlation perception in information visualization was also introduced into the evaluations using Weber's law proposed by Harrison *et al.* [20], and then a different model was provided by Kay and Heer [21]. More recenctly, Spicker *et al.* [22] proposed a measurement related to the number of stipple points to quantify the visual abstraction quality for stipple drawings.

Nevertheless, great efforts should be made in placing the stipple dots in order to ensure quality and rapidity of digital stipple. One of the recent trends in digital stippling is to incorporate structure or orientation information in generating dot distributions. For instance, the structure-aware stippling techniques are particularly useful in preserving visually identifiable structures and textures in the original image. Human artists use stipples not only for tone reproduction, but also to illustrate features, including edge emphasis and texture indication. Also, placing stipples by hand is a time-consuming process, and handmade stipple drawings contain fewer stipples than modern computer generated stipple images.

In this context, we propose an *I*mage *D*ata *F*ield-based method for *S*tippling (IDFS for short), a novel stippling algorithm not based on Lloyd's method. The core idea is to construct a data field for the given image, which we call an image data field. Each pixel in the image is considered as one particle with position, mass and potential. Data field is given to express the power of an item in the image by potential function as the physical field does. The description of potential function is based on the assumption that all observed data in the number universe will radiate their data energies and be influenced by others simultaneously. Thus, the data field of an image can keep the balance between spatial and grayscale information in local neighbourhood by potential function, as well as the balance between the local information and the global trend. Unlike the Lloyd's method, the use of image data field ensures regular spacing of dots along the feature directions as well as their perpendicular directions, and then a good balance on result quality and computational efficiency can be ensured.

In summary, the main contributions of our paper are the following:

- We introduce data field as an effective and unified tool for digital stippling, and then we provide an image data field-based algorithm for the computer-aided stippling. Specifically, we present the basic definition, the generation method, and the rendering algorithm how to produce stipple points from an image data field.
- For the different visual effects, we propose two potential center reduction algorithms based on image data field, that is, potential center elimination for stipple placement and potential center cutting for mosaic construction.
- We perform a comparative study of digital stippling, both visual comparison and quantitative comparison, and show the feasibility and the efficiency of the proposed method.

The remainder of this paper is organized as follows. In Section II, we provide a brief introduction on data field. In Section III, we describe our IDFS algorithm in detail: first image data field generation, next stipple placement by using potential center elimination, and then the stipple rendering algorithm, as well as a variant on mosaic construction. In addition, the parameter configuration is also provided in this section. In Section IV, we evaluate the performance of the proposed method and compare it to state-of-the-art

stippling methods. Finally, we discuss the results and give some ideas for future improvements in Section V.

II. DATA FIELD

Data field is proposed by Li and Du [23] in recent years, and has been of particular interest to researchers. Its main idea is originated from physical field. In nuclear field, nuclear force binds protons and neutrons together to form the nucleus of an atom. Data field provides an analogy with the mechanism of the nuclear field theory. Given a data space, data field describes the complex correlation among data objects, where there are some effects and interactions in an unknown way, and it expresses the power of an item in the universe of discourse by a potential function as the physical field does.

Data field has been introduced into various applications [24]–[26], and also successfully used in image processing [27]–[30]. There are two general categories of data field, static and dynamic ones, and we use the former in the paper. The static data field is corresponding to the stable active field in theoretical physics. Inspired by the preliminary work [31], we introduce the static data field into digital stippling and propose a novel method based on the image data field.

Given a data object p in data space Ω, let $\varphi_p(q)$ be the potential value at any position $q \in \Omega$ produced by p, then $\varphi_p(q)$ can be computed by anyone of the following equations,

$$\varphi_p(q) = M_p^q \times \exp(-(||p-q||/\sigma)^\alpha), \tag{1}$$

$$\varphi_p(q) = G \times M_p^q/(1+(||p-q||/\sigma)^\alpha), \tag{2}$$

$$\varphi_p(q) = M_p^q/(4\pi\varepsilon_0 \times (1+(||p-q||/\sigma)^\alpha)), \tag{3}$$

where $||p-q||$ is the distance between p and q, the strength of interaction $M_p^q \geq 0$ can be regarded as mass or charge of data objects, a natural number α is the distance index, and $\sigma > 0$ is the influential factor that indicates the range of interaction. Additionally, the distance is usually measured by Euclidean, Manhattan or Chebyshev metric.

Eqs.(1)-(3) are three common choices of the above potential function. Eq.(1) imitates nuclear field with Gaussian potential, while Eqs.(2) and (3) imitate gravitational field and electrostatic field respectively, where G and ε_0 are the constants depended on the law of gravitation and the Coulomb law. Mathematically considered, therefore, the latter two functions seem essentially the same. In addition, we need to state that, there are several alternative formulae for $\varphi_p(q)$, such as electromagnetic field, temperature field or nuclear field with exponential potential.

In general, there is more than one object in data space. To obtain the precise potential value of any position under these circumstances, all interactions from data objects should be concerned. Given a data set $D = \{p_1, p_2, \ldots, p_n\}$, because of the overlap, the potential value of any position q in the data space is the sum of all data radiation,

$$\varphi(q) = \sum_{i=1}^{n} \varphi_{p_i}(q), \tag{4}$$

where $\varphi_{p_i}(q)$ is the potential value of q produced by p_i, and calculated by one of Eqs.(1)-(3).

In the following, we use the nuclear field with Gaussian potential as Eq.(1) and then fix $\alpha = 2$. Taking into account the time complex and the efficiency of the proposed data field, we choose the Chebyshev distance for $||p-q||$, since once the influential factor is fixed, at the same time cost, most objects can be considered by using the Chebyshev distance. Furthermore, how to determine the mass M_p^q is still an open problem, and various applications use various definitions of the mass. Three kinds of masses, inertial mass, active gravitational mass, and passive gravitational mass, are defined in theoretical physics. For example, the mass used in [27] and [30] refers to the active gravitational mass of p acting on q, while the mass used in [29] is a passive gravitational mass measure by an object in a known gravitational field. Thus, the active gravitational mass that acts on pixel p by pixel q is significant for image data field. We would define the image data field considering the image intensity as the mass, and then produce a novel method for image stippling.

III. THE PROPOSED ALGORITHM: IDFS

A. IMAGE DATA FIELD GENERATION

Image data field is a natural extension of data field in image processing. Each image pixel is as a data particle with the mass, and has interactions with neighbouring pixels. The potential sum at any pixel is calculated by obeying the law of short-range nuclear force field.

Suppose $\Omega = \{p = (x_p, y_p) | x_p \in [1, w], y_p \in [1, h]\}$ is a finite space consisting of two-dimensional pixels, $g : \Omega \to [0, l-1]$ is a mapping, $g(p)$ denotes the grayscale value of the pixel p, and then an input image is a pair $I_g = (\Omega, g)$, where h, w, and l are the height, width, and intensity of the image respectively.

According to data field, each pixel $p \in \Omega$ is a particle with the mass, and the intensity change interactions (attraction or repulsion) between each other form an image data field on the two-dimensional space Ω. Assuming two pixels $p, q \in \Omega$, let $\varphi_p(q)$ be the potential at any pixel q produced by p, and then it can be computed by,

$$\varphi_p(q) = g(p) \times \exp(-(\max\{|x_p - x_q|, |y_p - y_q|\}/\sigma)^2), \tag{5}$$

where $g(p)$ is the strength of interaction, and can be as the mass of data object in image data field. The parameter σ also denotes the influential factor related with the interaction distance.

Given a two-dimensional space Ω, each pixel acts on each other by some interaction forces, and forms an image data field, the potential of any pixel q can be defined as below,

$$\varphi(q) = \sum_{p \in \Omega} \varphi_p(q), \tag{6}$$

where $p \in \Omega$ denotes the neighbourhood of q in the image.

Generally, there are too many pixels p in a given image space Ω. The above potential simulates the short-range nuclear force's field corresponding to the Gaussian potential

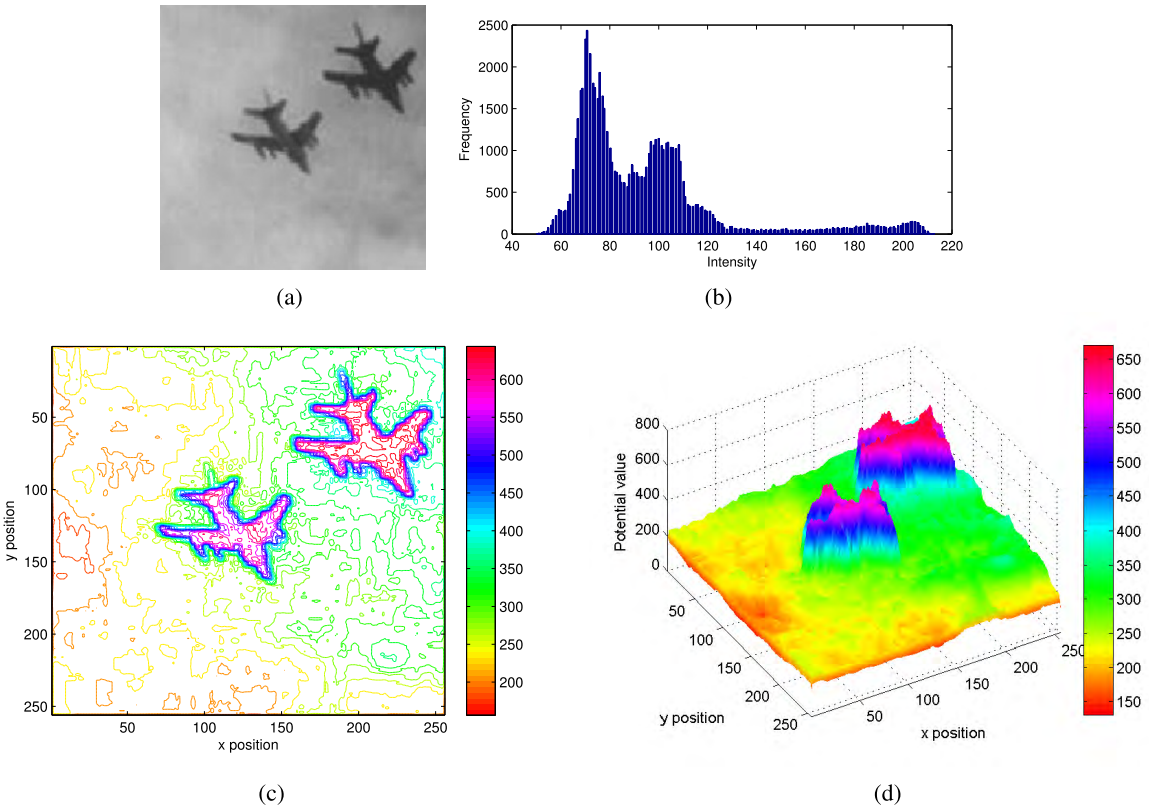


FIGURE 1. The example image and its data field: (a) the original image, (b) the histogram, (c) the equipotential lines, and (d) the equipotential surfaces.

field, which satisfies *three sigma rule*. But, it should be noted that, Eq.(5) is a slightly different from the standard Gaussian function, that is the variance, and then the second multiplier item in Eq.(5) should be converted into $\exp(-\frac{1}{2}(\frac{\max\{|x_p-x_q|,|y_p-y_q|\}}{\sigma/\sqrt{2}})^2)$. Thus, the standard deviation of the Gaussian function is actually $\sigma/\sqrt{2}$. In the image data field, the influential range of a data object is the geometrical neighbourhood with the distance shorter than $3\sigma/\sqrt{2}$, that is to say, data objects beyond a certain distance are almost not influenced by a specified data object, and the potential contributions become zero. Especially for the problems in image processing, too wide range of neighbourhood is time-consuming and unnecessary. To reduce the time cost, the potential value of any pixel q can be simplified as below,

$$\varphi(q) = \sum_{p\in\xi(q)} g(p) \times \exp(-(\max\{|x_p - x_q|, |y_p - y_q|\}/\sigma)^2), \tag{7}$$

where $\xi(q) = \{p|p \in \Omega, max\{|x_p - x_q|, |y_p - y_q|\} \leq 3\sigma/\sqrt{2}\}$ denotes the neighbouring pixels affected by the central pixel q.

In physics, scalar fields often describe the potential energy associated with a particular force. The force is a vector field, which can be obtained as the gradient of the potential energy scalar field. A particle gives out force in all directions, and receives forces from the neighboring particles. We take the original image in Figure 1a as an example, whose histogram is shown in Figure 1b. There are two airplane objects on the uneven background. The histogram of the image is approximately unimodal, and there exists seriously uneven illumination and slight gray level changes with a wide range in the background. Based on the above description, we model a grayscale image as a two-dimensional space, and then generate the image data field using Eq.(7). The equipotential lines produced by this image data field is shown in Figure 1c, where the two dimensions denote the position of each pixel with horizontal and vertical direction, respectively. More visually, Figure 1d demonstrates the equipotential surfaces, in which another dimension is added to denote the magnitude of each potential value. Supposing the image pixel p with the position (x_p, y_p), the value of new dimension is calculated by Eq.(7). Figure 1d indicates that the neighborhood pixels are associated with the central pixel ordered by the interactions, and the potential value of the pixel p moderate, the neighborhood pixels weaker interaction. It is noted that, the image has been pre-processed by step 3 of Algorithm 1 (see Section III-D), thus, two regions of airplanes are obtained higher potential values, even so these pixels are darker than background pixels and their grayscale values are comparably low.

Here, we confine ourselves to the two-class problem of separating objects (foreground) from background. Without losing generality, we assume that the background appears darker than the object, otherwise the original image should be reversed the color of each pixel. In fact, analyzing Eq.(7) intuitively, we know, when the influential factor σ is assigned, the image data field describes the grayscale tendency of pixels in a neighborhood, the higher the potential value of the pixel p, the greater the grayscale level and interaction, and these pixels should be corresponded to the objects in the image. Based on the image data field, the homogeneous regions with stronger interaction and higher grayscale can be obtained from an image by comparing the potential value since the spatial information is considered.

In general, there are several characteristics of the potential value in an image data field:

- Given the influence factor, the potential value of a central pixel is bounded by an interval.
- For a homogeneous neighborhood with the greater grayscale, the potential value of the central pixel is very high. Statistically, this neighborhood owns high mean potential and small standard deviation. Then, these pixels appear more of the objects.
- For a homogeneous neighborhood with the lower grayscale, the potential value of the central pixel is very small, even zero. This neighborhood has small mean potential and standard deviation, and the corresponding pixels are more like the background.
- For a neighborhood consisting of edge or noisy pixels from transition regions, the potential value of the central pixel is generally moderate, somewhere in between those of foreground and background. These pixels usually look not very continuous and uniform, and the potential values are highly unstable.
- The higher the potential value of a central pixel is, the more likely the central pixel is in the interior of a homogeneous region with high grayscale level, that is, from foreground.

Potential value is related to the grayscale tendency in the neighborhood. The flatter the grayscale change is, the more extreme the potential value. When a pixel locates in the interior of a homogeneous region from an image foreground, grayscale change of its neighborhood is slight. Accordingly, the potential value of the pixel is rather high. Similarly, a pixel in the interior of a homogeneous region from an image background, the potential value is fairly low. As for a pixel in the non-homogeneous region (that is transition region), the grayscale change of its neighborhood is sharp, and the corresponding potential is moderate. Thus, pixels in the interior of a homogeneous region from an image foreground will have higher potential than those in the non-homogeneous region and the background. Theoretically, the potential value of a pixel can be as an indicator to detect the candidate central pixel in the possible homogeneous region of objects.

B. STIPPLE PLACEMENT BY POTENTIAL CENTER ELIMINATION

Given a two-dimensional image space Ω, $q^* \in \Omega$ is the central pixel of a neighbourhood, $\varphi(q^*)$ denotes its potential value calculated by Eq.(7), q^* is most likely the local central pixel of the current image data field if $\varphi(q^*)$ is the maximum potential value. We call this pixel as the potential center, and the other neighbor pixels are attracted by q^*, which leads to a homogeneous region of objects.

In a homogeneous region of objects, all the other neighbour pixels are attracted by the central pixel to some extent, but the attracted intensity between any two pixels may be not same, which depends on the respective contribution to the potential of the central pixel in image data field. In particular, the central pixel is attracted by itself with the maximal intensity.

Given a two-dimensional image space Ω, $q^* \in \Omega$ is a potential center, for $\forall q \in \xi(q^*)$, there exists a finite sequence consisting of potential centers $q_1^*, q_2^*, \ldots, q_t^* (t \geq 1)$, $q_t^* = q^*$, and the following conditions are satisfied: (1) $q \in \xi(q_1^*)$; (2) when $t > 1$, for $\forall i \in [1, t-1]$, $q_i^* \in \xi(q_{i+1}^*)$; then q^* and q has a relationship of the potential center attraction, and $S_{q^*} = \{q, q_1^*, q_2^*, \ldots, q_t^*\}$ is the pixel set of potential center attraction.

Pixels are processed in priority order, where higher priority pixels are those closer in higher potential values. Once the current potential center is eliminated, the image data field should be updated according to the potential center attraction. Considering the iterations of image data field, the original image data field $\varphi(q)$ is as $\varphi_0(q)$, and we introduce $k > 0$ as a new indicator of the total number of iterations, and then the detailed definition is formalized as below.

$$\varphi_k(q) = \begin{cases} \varphi(q) & k = 0 \\ \varphi_{k-1}(q) - \sum\limits_{q_k \in S_{q^*}} g(q_k) \times e^{-\frac{||q-q_k||^2}{\sigma^2}} & k > 0 \end{cases} \tag{8}$$

where $\varphi_k(q)$, $\varphi_{k-1}(q)$ denotes the potential value of the pixel q in the k^{th}, $k-1^{th}$ image data field, $||q-q_k||$ is the distance of each pair of pixels, $g(q_k)$ is the grayscale level of the pixel q_k, σ is the influence factor, and S_{q^*} is the pixel set of potential center attraction.

Figure 2a shows the detail of the equipotential lines on a sub-image in an enlarged sense. The equipotential lines are relatively dense around the contour of the airplanes, while sparse on the background. Generally speaking, this equipotential map is simple in structure. Each potential center is shown in Figure 2b after 100 iterations. One can observe that the contour of the upper airplane has been emerged. The equipotential lines of the corresponding 100^{th} image data field are shown in Figure 2c, whose details in an enlarged sense are also listed in Figure 2d. Clearly, this equipotential map is more complex in structure. There are many positions with relatively low potential values, that is, many potential centers have being eliminated. By contrast of the two equipotential lines, more local minimums of the potential values are

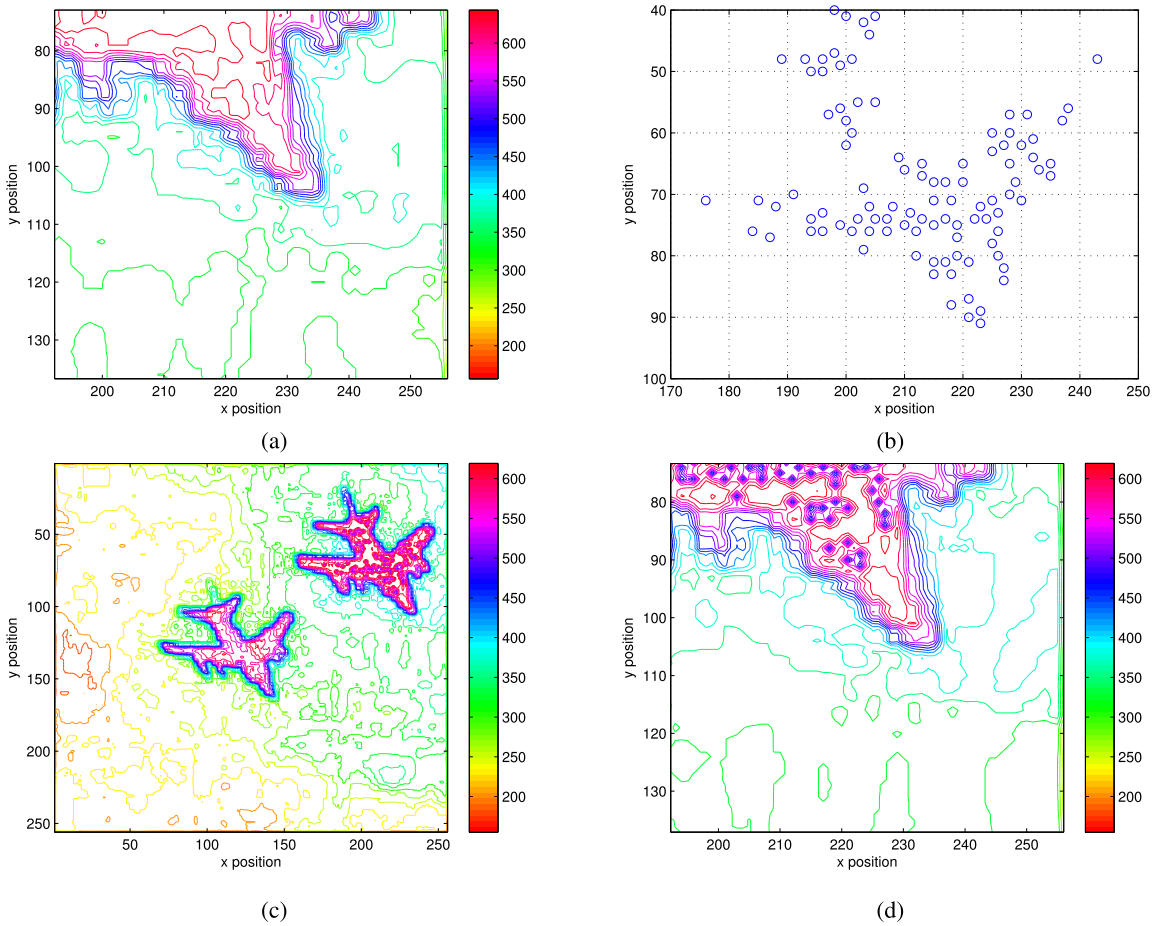


FIGURE 2. Stipple placement by potential center elimination: (a) detail of Figure 1c on a sub-image, (b) the locations of top 100 potential centers, (c) the equipotential lines with 100 iterations, and (d) detail of Figure 2c on a sub-image.

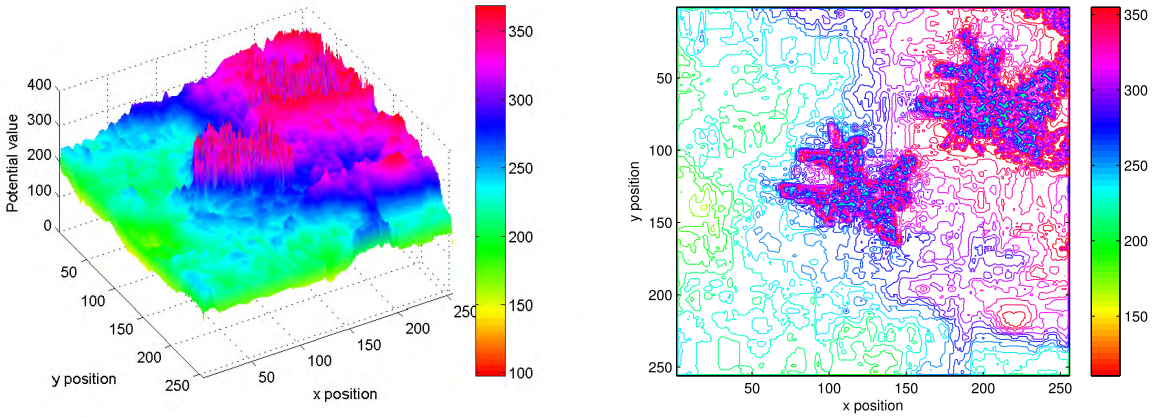
appeared in Figure 2d. Meanwhile, these minimal potential values cover the small regions only surrounding the central pixels, since we do not eliminate the potential contribution of the affected pixels at once in Eq.(8).

Figure 3a shows the equipotential surfaces of the final image data field when the process of potential center elimination is finished. For the reference, the corresponding equipotential lines are shown in Figure 3b. Once enough candidate potential centers are eliminated, the image data field would be very stable and flat. In one sense, the potential centers can be as the reduction of the image data field, as well as the origial image. Clearly, the number of the eliminated potential centers is more, the final image data field should become flatter, and the potential centers can describe the original data more accurately. Theoretically, one can consider all of the pixels as the potential centers with the elaborately fixed parameters, and then eliminate these potential centers, thus, the potential values of all locations in the final image data field would be all zero.

However, the real-time method that produces competitive quality solutions is more important and favorable since it allows users to acquire desired results instantly. Thus, accuracy and performance should be a dilemm. Too precise reduction leads to poor efficiency of the proposed method. On the one hand, once the influential factor is fixed, it will take more time to eliminate more potential centers and obtain a higher accuracy. On the other hand, at a lower time cost, only fewer potential centers can be considered, which leads to a lower accuracy on the original image. While lack of potential centers would also make it difficult to describe the original image information. Considering the time complex and the efficiency, a moderate number is more optimal to fix the number of the eliminated potential centers. However, the problem of determining this parameter without human-in-the-loop is still an open issue, and sustained effort is required for an autonomous processing.

C. STIPPLE RENDERING

The potential centers easy represent the image data distribution or its features, and then these pixels can be as the stipples. Thus, the number of the potential centers would affect the visual result of the image stippling. Our algorithm as presented so far still creates unacceptable results for huge numbers of stipples. It is clear from inspecting Figure 2b that the tone can be improved by making more dramatic adjustments to stipple size. To better reproduce fine details of



(a) (b)

FIGURE 3. Final results by using potential center elimination: (a) the final equipotential surfaces after potential center elimination, and (b) the final equipotential lines.

the original picture, a moderate stipple size should be fixed as much as possible. We propose adjusting the stipple size in the rendering process in such a way as to reduce the effect of stipple counts. Stipple size $r(q)$ varies linearly with the original potential value $\varphi_0(q)$, formalized as below,

$$r(q) = max\{r_{min}, r_{max}\frac{\varphi_0(q) - \varphi_{min}}{\varphi_{max} - \varphi_{min}}\}, \tag{9}$$

where $\varphi_{max}, \varphi_{min}$ respectively denote the maximum and the minimum one of potential values in the image data field φ_0.

According to the above equation, a minimum size stipple is placed at a site of pixels with lower potential values, up to a maximum size stipple when the potential values drops to the highest. Deussen *et al.* [6] argued that artists rarely use stipples which vary in size by more than a factor of two. In Eq.(9), we follow this advice and have our maximum stipple twice the size of the minimum, so $r_{min} = 1$ and $r_{max} = 2$.

Figure 4 shows further results of IDFS on an image, and the number of stipples is 327, 1310, 2621, and 32768 respectively. By using our IDFS method, the result is still acceptable only with 1310 stipples, since the two airplane objects are clearly represented by these stipples. Along with the number of stipples increasing, more and more background pixels are involved into stipples, although the stipple size is comparatively small.

D. OVERALL PROCESS

Algorithm 1 outlines the generation process used, which contains three main components, including image data field generation, potential center elimination and stipple rendering. Given an input image, the proposed method automatically converts it into stipple style by the following steps:

System initialization (step 1) involves setting parameters, and reading in the original images. Some pre-processing (steps 2 and 3) is done, in preparation for the generation of image data field later. This is followed in step 4 by the generation of image data field, to be used as the initial potential value for the iteration process. Steps 5 to 11 form the main iteration loop of the rendering system. Each generation has a maximum potential value in the current image data field, and a potential center is found as one candidate stipple. With the

Algorithm 1 The IDFS Algorithm

Input: The original image I_o and the influence factor σ.

Output: The stipple list L, that is, a list of stipple dots with the size.

1: Read the input image data I_o, and set initialization parameters, for example, $k \leftarrow 0, L \leftarrow \varnothing$.
2: Pre-process the original image and obtain an 8-bit image I_g if involved in a 24-bit image, or set $I_g \leftarrow I_o$ directly.
3: Reverse the color of the 8-bit image if necessary, and obtain the updated image I_u, or set $I_u \leftarrow I_g$ directly.
4: Generate an image data field φ from the image data I_u with the given influence factor pattern σ (see Eq.(7) in Section III-A), and then $\varphi_k \leftarrow \varphi$.
5: **repeat**
6: Find the maximum potential value in the current image data field φ_k, whose corresponding position (X^*, Y^*) is as the current potential center q^*, and then $L \leftarrow L \cup (X^*, Y^*)$.
7: **while** $p \in \xi(q^*)$ **do**
8: Update the $k+1^{th}$ image data field φ_{k+1} using Eq.(8) mentioned in Section III-B.
9: **end while**
10: Draw the k^{th} stipple with the varied size (see Section III-C) and then $k \leftarrow k+1$.
11: **until** The number of stipples is more enough or other stopping criterion is satisfied.
12: Produce the current stipple list L as the final rendering image.

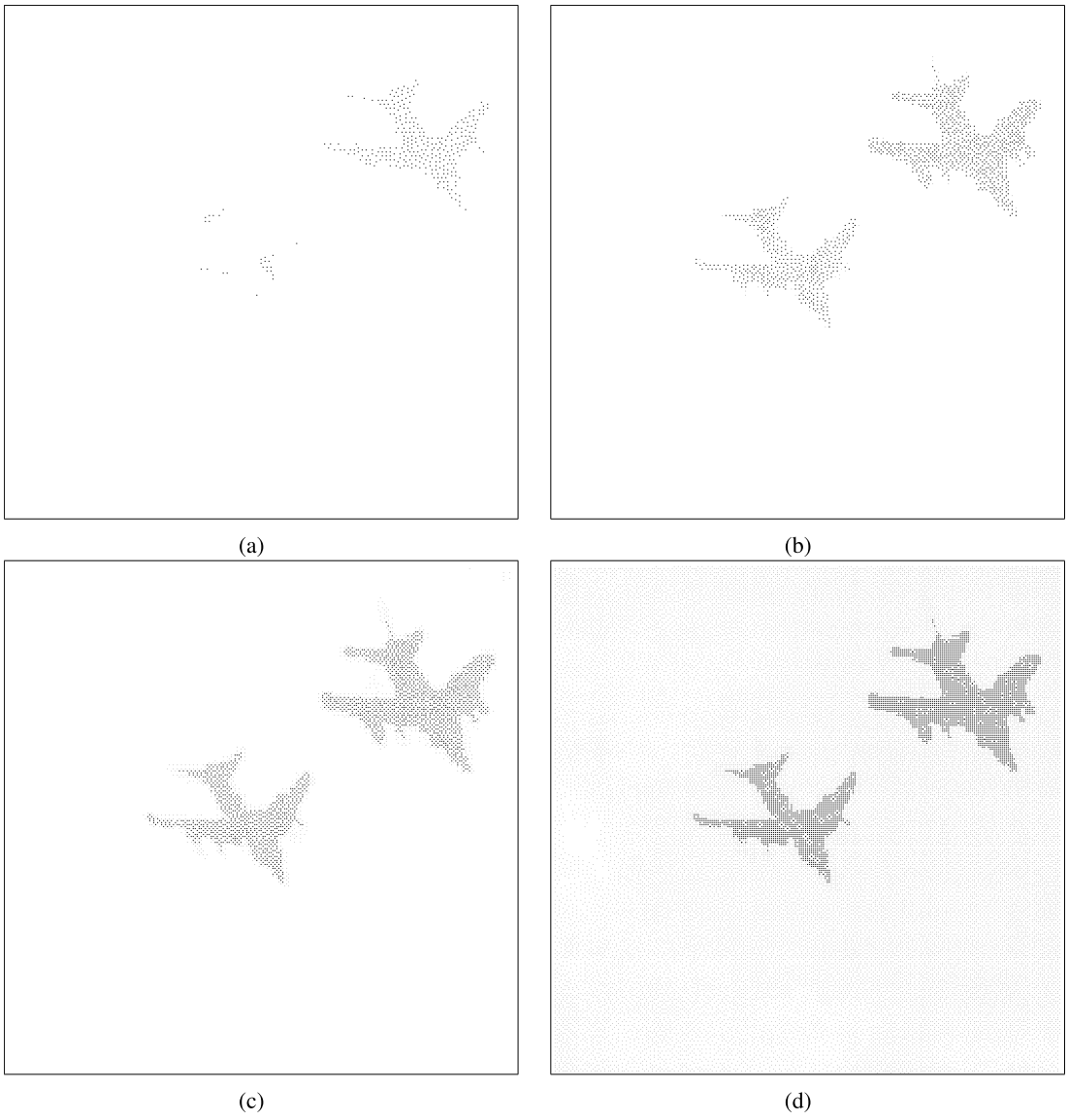


FIGURE 4. Example results from using our IDFS method: (a) 327 stipples, (b) 1310 stipples, (c) 2621 stipples, and (d) 32768 stipples.

incremental generation k, more and more potential centers are eliminated, and then image data field becomes flatter and flatter. Once the enough stipple dots have been prepared, the rendering with the varied size should be applied to them. Finally, the rendering result can be produced as various forms.

Here, we provide a concise description on step 3. We introduced a controlled variable by visual observation and then changed the grayscale value of each pixel p from $g(p)$ in I_g to $255 - g(p)$ in I_u, if involved darker objects in images, where $g(p)$ is the grayscale value of the original image. But the additional processing does not affect any stippling results. We only introduced this process because of the consistency of artistic style, that is, object pixels are always been processed earlier, since bright pixels should be with higher potential values and probably eliminated as a potential center according to the priority.

E. MOSAIC CONSTRUCTION

Mosaics and stipples have a close relationship: the same primitive distribution problems are seen in each other. We can easy modify the step of Algorithm 1 to place mosaic tiles.

The Voronoi diagram is a method of dividing an image into a number of regions. We consider each potential center as the centroid of each Voronoi region. The used method of potential center elimination should differ from Eq.(8). To accelerate the potential center elimination, we remove the effect of all the related pixels, instead of partial influence like Eq.(8). The modified mathematical model is as below,

$$\varphi_k(q) = \begin{cases} \varphi(q) & k = 0 \\ 0 & k > 0 \end{cases} \tag{10}$$

The contrast of Eq.(8) and Eq.(10) shows that the only difference between them is the updated method of the potential

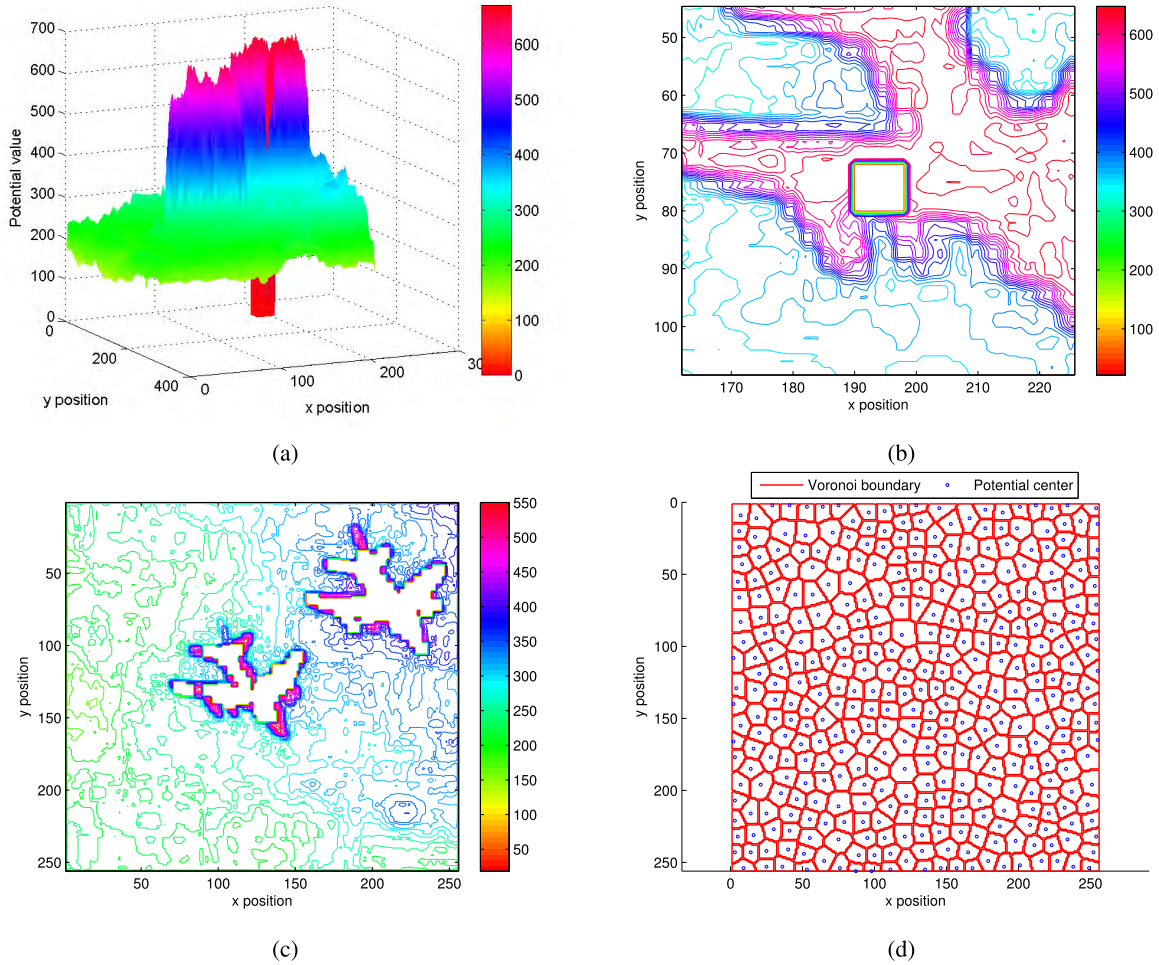


FIGURE 5. Mosaic construction by potential center elimination: (a) the equipotential surfaces with the first potential center elimination, (b) the details equipotential lines from the first iteration, (c) the equipotential lines after 100 iterations, and (d) the Voronoi boundaries related to 382 potential centers.

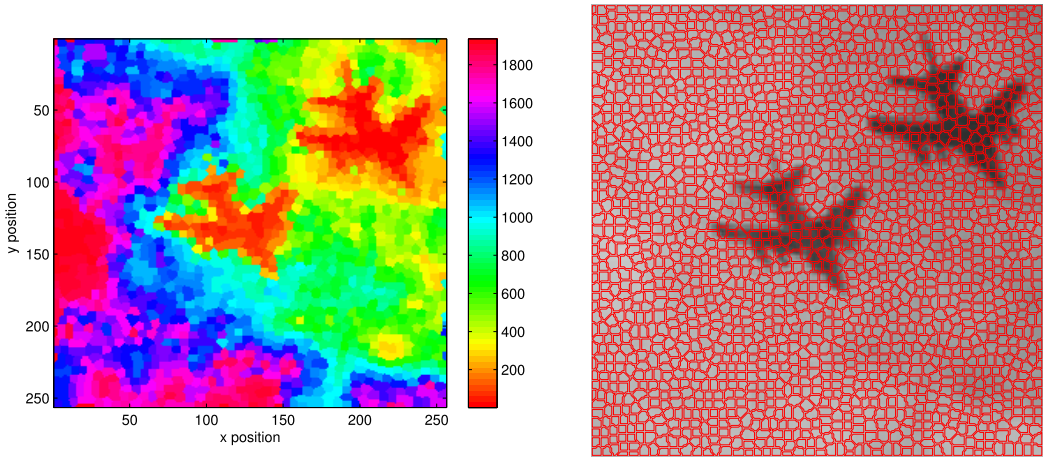
value in the process of potential center elimination. At each iteration, the algorithm moves each potential center, then rearranges the voronoi cells. Figure 5 demonstrates the mosaic construction by potential center elimination. After the first potential center elimination, the equipotential surfaces are shown in Figure 5a. There is a huge empty on the base of the equipotential surface. More clearly, the details equipotential lines of Figure 5a is also listed in Figure 5b, where a square hole is evidently appeared. With 100 times of potential center elimination, the equipotential lines is shown in Figure 5c. The two airplane objects are approximately depicted by the white regions with zero potential values. In addition, the potential values using Eq.(10) are reduced more sharply than those using Eq.(8). In this way, we obtain 382 potential centers, and then generate the Voronoi boundaries related to these potential centers. The result is shown in Figure 5d. Basically, the intuitive analysis by the naked eye indicates that the size of tiles is uniform, and the space arrangement of regions is also appropriate.

Figure 6 shows some examples of mosaics obtained by using our IDFS method. We generate Voronoi diagram with totally 1936 Voronoi cells, whose color is determined by the elimination order of potential centers. As shown in Figure 6a, the first 100 Voronoi cells are rendered using approximate red color. The visual result of 1936 Voronoi regions is overlapped with the original image, as shown in Figure 6b. Overall, the results are acceptable.

F. PARAMETER CONFIGURATION

There are three parameters in the above analysis on image data field-based method for stippling. The most important one is the influential factor σ in definition Eq.(5), and the another two are the minimum and the maximum size of stipples, that is, r_{min} and r_{max} in Eq.(9). The latter two have discussed in Section III-C, which are introduced as optional parameters to control the artistic freedom of the output.

Thus, there is only one pending parameter. Here, we discuss the choice of σ. Although the optimal setting of the parameter σ depends on the image under analysis and on the subjectivity of the user, a good trade-off is the following. The influential factor σ is closely related with the neighborhood size $3\sigma/\sqrt{2}$. A too small neighborhood size will



(a) (b)

FIGURE 6. Examples of mosaics obtained by using our IDFS method: (a) Voronoi diagram, and (b) visual result of 1936 Voronoi regions.

result in imprecise estimate of the original image, while a too large neighborhood size will cause the loss of localization and the high time cost of the implementation. In Eq.(7), the neighbouring pixels $\xi(q)$ related with the central pixel q can be clearly determined by the the neighborhood size $n_s = \lceil 3\sigma/\sqrt{2} \rceil$. Considering the neighborhood size is actually an integer for an image, it is relatively easy to choose, thus we experimentally obtain an optimal n_s value n_s^* from 3 to 11 in this paper, and then the influential factor σ is approximately calculated by $\sqrt{2}n_s^*/3$.

For the termination strategy (step 11) of Algorithm 1, there are two related parameters, that is, G_k and T_p, since the terminating conditions include two components. G_k is the maximum number of stipples, as well as the maximum generation of the iteration, and then the IDFS system is stopped when a fixed number of stipples are acceptable. That is, the iterative process of Algorithm 1 is terminated with $k > G_k$. In addition, the potential difference threshold T_p denotes the difference between the average potential values of two successive generations, which indicates that the result is reaching or has reached a plateau. In other words, if any one of the above conditions is satisfied, the proposed algorithm is terminated.

Certainly, some other suitable termination strategies for Algorithm 1 are also alternative. However, how to choose the termination strategy and set the appropriate parameter value is still an open problem.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. EXPERIMENTAL SETUP

In order to illustrate the performance of the IDFS system, both grayscale and color images, are involved into the experiments. The proposed method is implemented under Matlab 2014 environment.

We conduct two groups of experiments with other existing techniques. In the first group, StippleShop[1] and an interactive reimplementation[2] of the LBG method [10] are used for the related methods. StippleShop is a filter- and effect-based tool that implements the 2D stippling/dot placement approaches, including the WVS method [7], the SAS method [16], the RWT method [12]. In addition, halftoning has strong connections to stippling, and black and white stippling results might be treated as a kind of halftoning. Both stippling and halftoning share the goal of representing a continuous-tone image for a binary medium, and we also focus on investigating suitable dot distributions. Thus, in the second group of experiments, we compare our result images with the halftoning results by OSTromoukhov's error-diffusion algorithm (OST for short) [32], and Li and Mould's Contrast-Aware Halftoning algorithm (CAH for short) [33], these two methods are also implemented by StippleShop.

For a fair comparison, all the methods are auto-parameters or default settings, including the influence factor σ in our method. All the experiments are performed on a PC with a 2.3GHz Core i7 CPU, 16GB RAM and a Nvidia Geforece GTX 1080 Graphics card, running on a Windows 10 operating system. Some original images are shown in Figure 7.

B. RESULTS

1) VISUAL COMPARISON

As a first quality criterion, we investigate the perceptual quality of our IDFS algorithm. Here, we compare and analyze the related methods to help evaluate our approach. For the airplane image, Figure 8 gives a comparison against the results from the SAS method [16] and the RWT method [12], who shares our goal of using stipples for structure, as well as the OST method [32] and the CAH method [33], who shares our goal of using stipples for halftoning. In addition, two Lloyd optimization-based methods, that is the WVS method [7] and the LBG method [10], are also involved into the comparison, since the Lloyd's algorithm is as the classical method for

[1] http://tobias.isenberg.cc//personal/demos/stippleshop.zip

[2] https://github.com/MarcSpicker/LindeBuzoGrayStippling

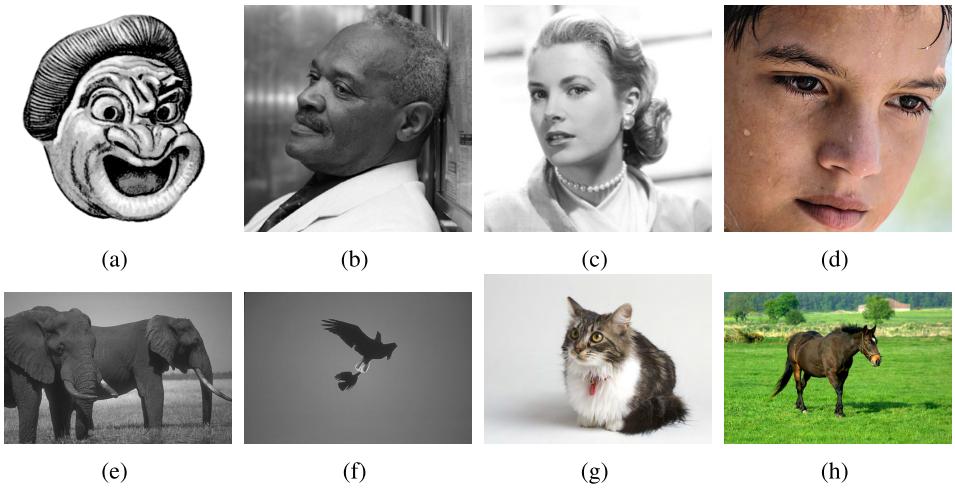


FIGURE 7. Some original images.

digital stippling, and the WVS method is as the historical one, while the LBG method as the recent one.

The two structure-based stippling methods, that is the SAS method and the RWT method, cannot produce the acceptable results, since the two airplane objects are lost and generalized in the background. The two halftoning methods, that is the OST method and the CAH method, yielded the acceptable results, but the process of the uneven background is still far from perfect, and meanwhile too many dots are used. The two Lloyd optimization-based methods showed different performance, the WVS method cannot represent the input well even the iteration time is the longest, and the LBG method created a good approximation, but the dot size should be still further investigated and improved.

Compared with Figure 8, Figure 4 shows that our method preserves structure better than these six previous methods. Visually, the effect of our result images is quite good. Our IDFS method can achieve significantly better tone reproduction and better structure. In other words, our IDFS method can express very complicated and detailed content.

In addition, we further validate the fact that our approach is useful for stippling and it is a better alternative to the existing methods. We only select the best result for each input image by the traditional methods, as well as results by the proposed method, not all the results for eight original images. On the one hand, this simplification makes the paper more compact. On the other hand, we introduced a simple user evaluation on the aesthetic performance of these result images by various methods. We started by inviting 68 users to participate in this investigation, who volunteered from our university, and there are 4 digital stippling novices, 13 digital art enthusiasts, 48 undergraduate students with some art knowledge, and 3 professional teachers of artistic design. All of them declared that they have seen hand-drawn stipple images and are with some related knowledge. Then, the users were asked to evaluate the result images by various methods considering the original images in Figure 7 as the input. Later, they picked out the best results by the compared methods, and completed a task of judging our method's ability to exhibit aesthetic performance equivalent to, or indistinguishable from, those of the traditional methods. The answer is dualistic, just ''yes'' or ''no''.

The most selected result images by the traditional methods are listed in Figure 9. Among the six traditional methods, the OST method is voted three images from eight ones as the best by an overwhelming majority, as shown in Figures 9a, 9b and 9d, and the CAH method also three images, as shown in Figures 9e, 9f and 9h. While the LBG method and the SAS method have produced as the best once, that is for only one image, as shown in Figures 9c and 9g. Furthermore, the RWT method and the WVS method lack the enough support, thus no result iamges by these methods are listed in Figure 9. Specially, for the two Lloyd optimization-based stippling methods, the WVS method has to be given a good initial point distribution, and cannot yield reasonable results for the selected images, while the LBG method generated moderate results, but not perfect, since it should be investigated more advanced cell splitting criteria to achieve adaptive size.

For the reference, all the result images are still attached in the Supplementary Materials available online. It is worth pointing out that, the results for the latter four original images are absent, since the RWT method cannot directly handle the input image with the unequal size.

In general, about 72% of participants (49 users) think the results by our IDFS method are acceptable, that is, the response to the above dual question is ''yes''. This user study shows that our method gained a certain approval, and can be as an alternative to the traditional methods. Some users argued that, with less stipples, our method even produces better results than the traditional methods, compared with Figures 9c to 9f. Some other users felt that the proposed

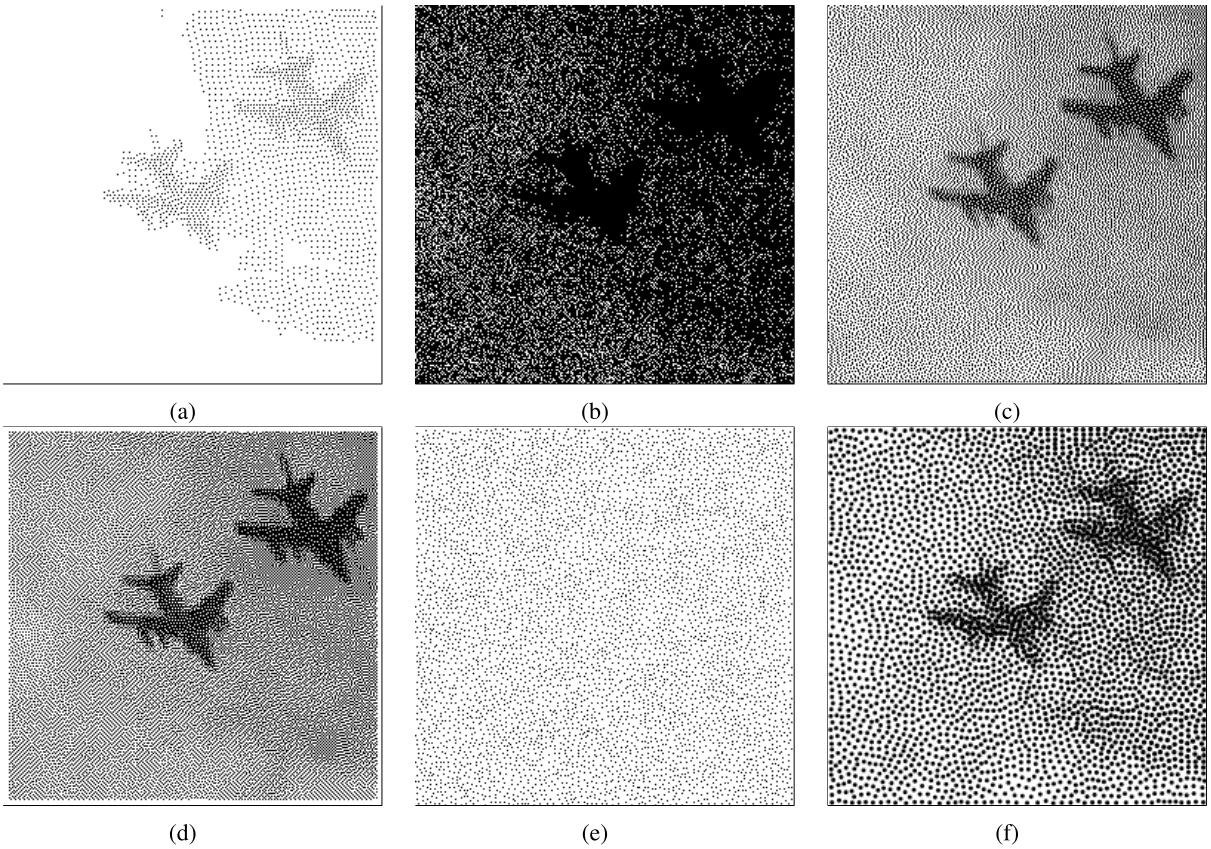


FIGURE 8. Comparisons of the results by various methods: (a) SAS, (b) RWT, (c) OST, (d) CAH, (e) WVS, and (f) LBG.

TABLE 1. Quantitative evaluation results.

Measures	Methods	airplane	No1	No2	No3	No4	No5	No6	No7	No8	Total
PSNR	SAS	8.269	16.352	8.408	16.712	11.781	6.415	6.986	17.641	10.009	11.397±4.428
	RWT	4.581	15.327	**10.971**	15.811	12.015	/	/	/	/	11.741±4.509
	OST	6.619	**17.753**	9.807	17.430	**12.566**	7.798	7.296	16.565	**10.621**	11.828±4.457
	CAH	6.518	17.228	10.774	15.177	12.189	**9.044**	**8.216**	15.886	10.515	11.728±3.686
	WVS	7.592	7.151	4.574	9.305	5.804	4.134	4.952	8.126	5.675	6.368 ± 1.762
	LBG	8.383	12.712	9.578	11.614	8.827	8.691	8.102	10.891	8.619	9.713 ± 1.635
	IDFS	**8.776**	16.934	9.909	**17.558**	11.872	6.926	6.842	**18.517**	10.146	**11.943±4.586**
MSSIM	SAS	0.493	0.675	0.159	0.615	0.310	0.081	0.011	0.726	0.221	0.366±0.269
	RWT	0.004	0.641	0.023	0.393	0.085	/	/	/	/	0.229±0.278
	OST	0.009	0.752	**0.235**	0.648	0.479	0.220	0.085	0.169	0.396	0.333±0.254
	CAH	0.011	**0.826**	0.044	0.459	0.229	0.054	0.012	0.181	**0.401**	0.247±0.273
	WVS	0.011	0.433	0.011	0.299	0.055	0.012	0.010	0.066	0.063	0.106 ± 0.152
	LBG	0.014	0.652	0.044	0.351	0.058	0.029	0.011	0.072	0.121	0.150 ± 0.215
	IDFS	**0.752**	0.734	0.182	**0.665**	**0.546**	**0.221**	**0.679**	**0.756**	0.321	**0.539±0.235**

method itself appears to have the ability to slightly process and understand the meaning of the original images, especially for the salience of images. In summary, the experimental results of these original images indicate that the proposed method is effective in yielding the approximately ideal results of stipple rendering.

2) QUANTITATIVE COMPARISON

For a quantitative comparison, we employ the structural similarity index measure (SSIM) [19] and the peak signal-to-noise ratio (PSNR) to quantify the tone and the structure difference between the stippling result and the original grayscale image.

Table 1 gives the result data for the structure similarity. The proposed IDFS method has obtained higher SSIM values than other previous methods, thus higher structure similarity. Additionally, tone similarity is usually measured by computing the PSNR between two images. Higher PSNR values indicate higher tone similarity. Tone similarity by PSNR is also listed in Table 1. Our method has better tone matching than others, apart from the OST method which was carefully honed to match tone exactly. As a whole, the last column of Table 1 shows the total performance of the various methods, our method produced the highest mean PSNR value, as well as the best mean MSSIM value.

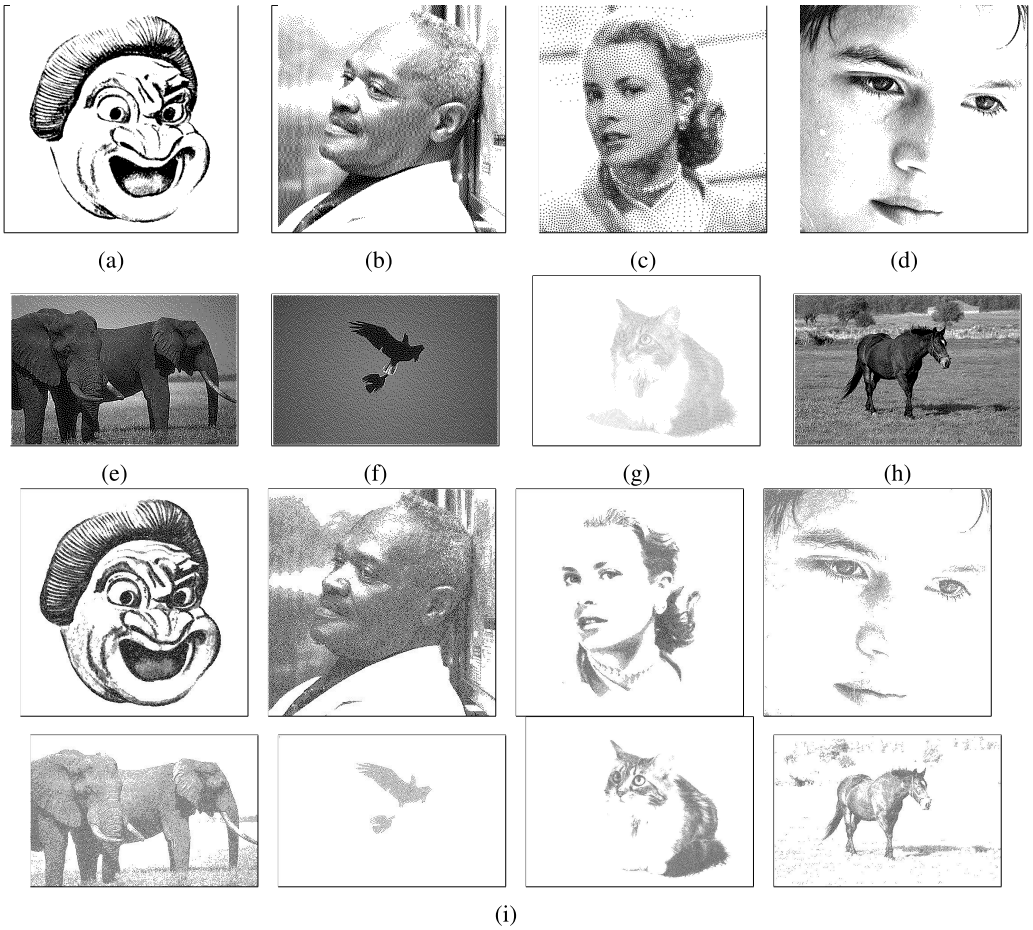


FIGURE 9. Other comparisons of the results by various methods: (a) (b) (d) results by the OST method, (c) results by the LBG method, (e) (f) (h) results by the CAH method, (g) results by the SAS method, and (i) results by our IDFS method.

From the earlier analysis [18] and our priori knowledge, we use GLCMs to explore and evaluate the stipple aesthentics from the perspective of texture analysis. We create GLCMs from these result images, using horizontal offsets of (0, 1), (0, 2), . . . , (0, 40). Figure 10 presents the three texture statistics plots of the result images by various methods. Each plot displays the corresponding texture statistic in relation to the magnitude of the offset direction. Figure 10a shows the texture correlation of the result images by the five methods. All of the correlation plots have smooth curves falling from a high correlation at an offset of one pixel to a low correlation as the offset approaches approximately five pixels. In contrast, the other five texture correlation plots, except results by the WVS method and our approach, resemble a random distribution with the growth of the offset in the second half. But,,according to the above analysis, the visual results by the WVS method are totally unacceptable. Figure 10b shows the texture contrast of the same result images in relation to the offset magnitude. Again, one can see smoother curve in our result by contrast with those by other methods. This curve indicates that the contrast to neighboring pixels is low, but farther from the reference pixel, the contrast becomes high. Figure 10c shows the images' texture energy in relation to the offset magnitude. Here, we see discrepancies among the techniques. Our result image has a smooth curve, with a slightly higher set of energy values corresponding to pixels neighboring the reference. In addition, our IDFS plot is a constant value with a fairly high energy level, indicating a random stipple distribution.

3) DISCUSSION

In this subsection, we provide a brief discussion on the proposal. Our IDFS method used a priority mechanism focusing on extremal potential values, and it introduced some optional parameters to control the number of stipples and the artistic style. The above results for the original images demonstrated overall higher visual quality and relatively less time consumption. Although the proposed approach has been proven to be successful in some cases, none of methods are generally applicable to all images, and different algorithms are usually not equally suitable for a particular application. Our technique based on image data field is superior to other existing methods, especially for those uneven lighting images with salient objects. However, it is currently not good at

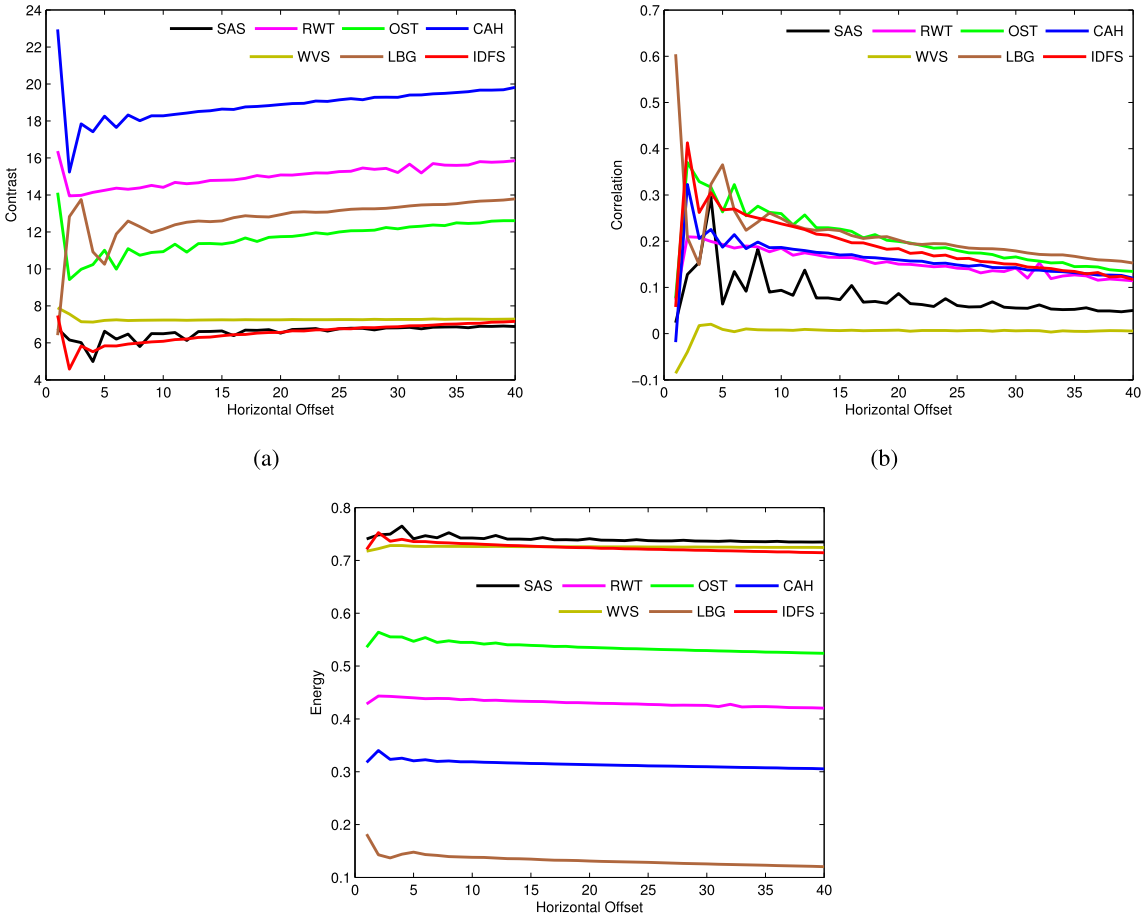


FIGURE 10. Texture statistics with horizontal offset from various methods: (a) contrast, (b) correlation, and (c) energy.

FIGURE 11. Failure results of applying the proposed method: (a) the original image, (b)-(d) the results by our method, the OST method and the LBG method, respectively.

capturing very complex and rich texture. Typically, one of the failure cases is shown in Figure 11, result by our method is simply not acceptable, but among the six traditional methods, the OST method and the LBG method generated proper results. In fact, image data field used in the paper mainly considers the color information of the images, but combines

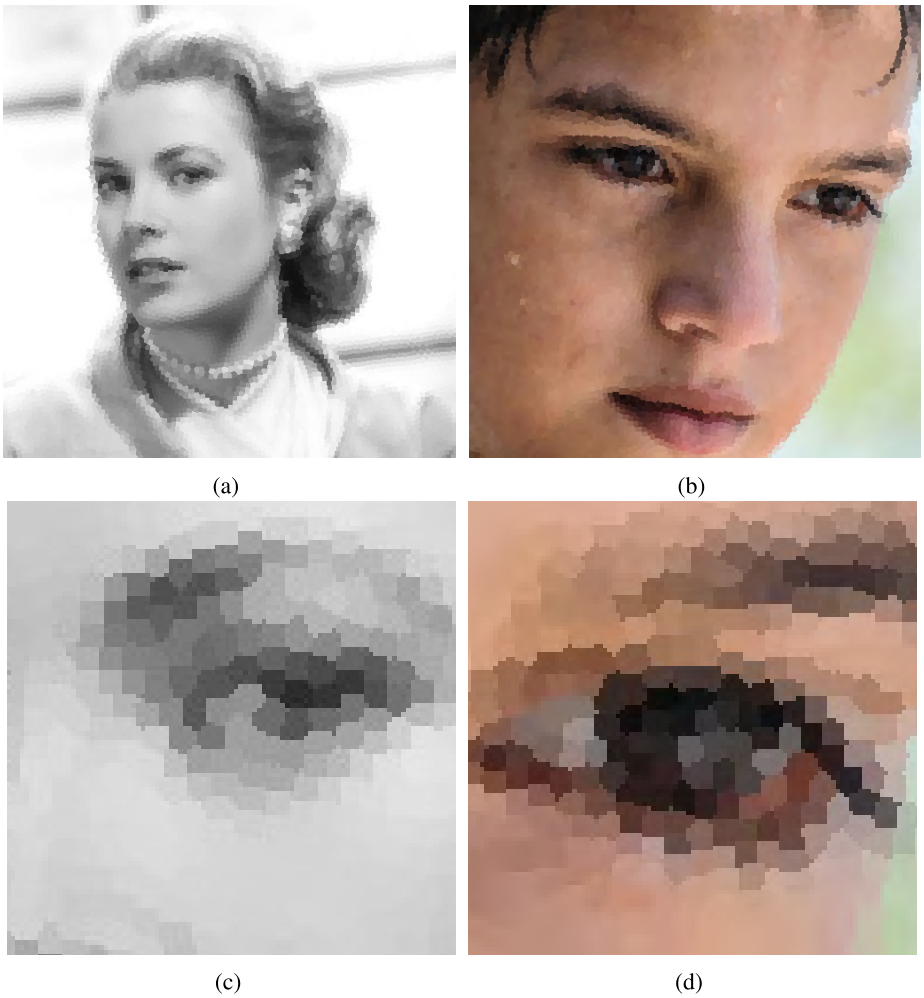


FIGURE 12. Examples of mosaics obtained by using 10000 Voronoi regions: (a) the No 2 image, (b) the No 4 image, (c) detail of (a) on a sub-image, and (d) detail of (b) on a sub-image.

only a limited amount of local structure, which is far from sufficient to express the global texture in images. In other words, our method cannot efficiently represent the texture structure in some more complicated cases, and then produces the unsatisfactory results. Of course, different image data fields with potential functions for different type of input images might be beneficial. But how to choose the suitable potential function is clearly not a simple problem without human intervention.

4) TILE MOSAICS

The proposed image data field can also serve as a general framework for tile mosaics, where the potential centers are arranged to follow the feature flow with some spacing constraint. We use the IDFS method to build tile mosaics. Figure 12 shows two examples of mosaics obtained by using 10000 Voronoi regions. As shown in Figure 12, the result images appear the key details very nicely, and exactly capture the feature of the original images. Also our IDFS method usually use less tiles in the resultant mosaics than those used in previous mosaics of irregular tiles. As a whole, the experimental results of these images suggest that the proposed method is effective in yielding the approximately ideal results of mosaic construction.

C. TIME COMPLEXITY

The main time cost of the proposed IDFS system lies in image data field generation (step 4 in Algorithm 1) and potential centers reduction (steps 6-9 in Algorithm 1) in a window-by-window manner. Considering the original image with the size of hw, the former consumes the time of order $O(4n_s^{*2}hw)$, where the affiliated term is generally much less than hw. The latter scans each pixel once which takes time about $O(hw)$. Additionally, the presented method for stipple rendering (step 10 in Algorithm 1) costs time no more than $O(G_k)$, where the number of stipples G_k is usually satisfied the condition $G_k \ll hw$. Thus, the time complexity of the proposed algorithm is approximately linear in the size of the original image, that is, $O(hw)$. Therefore, the computational complexity analysis indicates that our IDFS system

is enough efficient. In fact, for an image with the size of 512×512, the time consume is usually not more than 1 s in our practice.

In a way, the proposed method is efficient from the view of running time, and it can be approximatively satisfied by the need of the real-time applications, because our method runs with less time cost and the generated visual results are also acceptable.

V. CONCLUSION AND FUTURE WORK

In the paper, we provide an alternative stippling method using image data field. The experiment results, both visual and quantitative comparisons, verify the efficiency and feasibility of the proposal, which can be as an alternative to the existing methods. Also, the proposed IDFS algorithm has a good performance on running time, and its time complexity is approximatively linear related to the size of the input image. To some extent, our method understands and reduces the image information by using the data field evaluating the interaction between pixels, and then achieves stippling rendering with less stipples. The main difference between our method and the previous studies is the proposed method potentially contains an effect of salient region segmentation.

There are a couple of issues that will be considered in the future research: (1) Different potential functions of image data field should contribute to and lead to various visual effects of result images, and more alternative potential functions maybe provide just as or more interesting than the presented result images, thus the extension of the proposed method to introduce more other potential functions is well worth further studying on. (2) Because of our existing knowledge, we would prefer to focus our method more on computer art in the paper, but not computer graphics purely. Nevertheless, the proposed method is only implemented in Matlab, and other languages for engineering applications should be employed, for example, C++ and Java. Then, how to integrate with the existing image processing software is another feasible direction. The extension of the technique is currently under investigation and will be reported later.

APPENDIX

Supplementary data on result images: supplementary data associated with this article can be found online. In the supplementary materials, we provide a ''7-zip'' archiver file, which include various image results, mentioned in Figure 8 and Section IV-B1.

REFERENCES

[1] B. Galea, E. Kia, N. Aird, and P. G. Kry, ''Stippling with aerial robots,'' in *Proc. Joint Symp. Comput. Aesthetics Sketch Based Inter. Modeling Non-Photorealistic Animation Rendering*, Lisbon, Portugal, May 2016, pp. 125–134.

[2] J. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, ''State of the art: A taxonomy of artistic stylization techniques for images and video,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 5, pp. 866–885, May 2013.

[3] O. Deussen and T. Isenberg, ''Halftoning and stippling,'' in *Image Video-Based Artistic Stylisation*, vol. 42, P. Rosin and J. Collomosse, Eds. London, U.K.: Springer, 2013, pp. 45–61.

[4] D. Martín, G. Arroyo, A. Rodríguez, and T. Isenberg, ''A survey of digital stippling,'' *Comput. Graph.*, vol. 67, pp. 24–44, Oct. 2017.

[5] S. Lloyd, ''Least squares quantization in PCM,'' *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[6] O. Deussen, S. Hiller, C. V. Overveld, and T. Strothotte, ''Floating points: A method for computing stipple drawings,'' *Comput. Graph. Forum*, vol. 19, no. 3, pp. 41–50, 2000.

[7] A. Secord, ''Weighted voronoi stippling,'' in *Proc. 2nd Int. Symp. Non-Photorealistic Animation Rendering*, Annecy, France, Jun. 2002, pp. 37–43.

[8] S. Hiller, H. Hellwig, and O. Deussen, ''Beyond stippling—Methods for distributing objects on the plane,'' *Comput. Graph. Forum*, vol. 22, no. 3, pp. 515–522, 2003.

[9] M. Balzer, T. Schlömer, and O. Deussen, ''Capacity-constrained point distributions: A variant of Lloyd's method,'' *ACM Trans. Graph.*, vol. 28, no. 3, pp. 861–868, 2009.

[10] O. Deussen, M. Spicker, and Q. Zheng, ''Weighted linde-buzo-gray stippling,'' *ACM Trans. Graph.*, vol. 36, no. 6, p. 233, 2018.

[11] D. Mould, ''Stipple placement using distance in a weighted graph,'' in *Proc. 2nd Eurograph. Workshop Comput. Aesthetics Graph., Vis. Imag.*, Banff, AB, Canada, Jun. 2007, pp. 45–52.

[12] J. Kopf, D. Cohen-Or, O. Deussen, and D. Lischinski, ''Recursive wang tiles for real-time blue noise,'' *ACM Trans. Graph.*, vol. 25, no. 3, pp. 509–518, 2006.

[13] S.-Y. Kim *et al.*, ''Stippling by example,'' in *Proc. 7th Int. Symp. Non-Photorealistic Animation Rendering*, New Orleans, LA, USA, Aug. 2009, pp. 41–50.

[14] D. Martín, G. Arroyo, M. V. Luzón, and T. Isenberg, ''Example-based stippling using a scale-dependent grayscale process,'' in *Proc. 8th Int. Symp. Non-Photorealistic Animation Rendering*, Annecy, France, Jun. 2010, pp. 51–61.

[15] D. Martín, G. Arroyo, M. V. Luzón, and T. Isenberg, ''Scale-dependent and example-based grayscale stippling,'' *Comput. Graph.*, vol. 35, no. 1, pp. 160–174, 2011.

[16] H. Li and D. Mould, ''Structure-preserving stippling by priority-based error diffusion,'' in *Proc. Graph. Interface*, St. John's, NL, Canada, May 2011, pp. 127–134.

[17] L. Ma, Y. Chen, Y. Qian, and H. Sun, ''Incremental Voronoi sets for instant stippling,'' *Vis. Comput.*, vol. 34, nos. 6–8, pp. 863–873, 2018.

[18] R. Maciejewski, T. Isenberg, W. M. Andrews, D. S. Ebert, M. C. Sousa, and W. Chen, ''Measuring stipple aesthetics in hand-drawn and computer-generated images,'' *IEEE Comput. Graph. Appl.*, vol. 28, no. 2, pp. 62–74, Mar. 2008.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ''Image quality assessment: From error visibility to structural similarity,'' *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[20] L. Harrison, F. Yang, S. Franconeri, and R. Chang, ''Ranking visualizations of correlation using Weber's law,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1943–1952, Dec. 2014.

[21] M. Kay and J. Heer, ''Beyond weber's law: A second look at ranking visualizations of correlation,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 469–478, Jan. 2015.

[22] M. Spicker, F. Hahn, T. Lindemeier, D. Saupe, and O. Deussen, ''Quantifying visual abstraction quality for stipple drawings,'' presented at the 15th Int. Symp. Non-Photorealistic Animation Rendering, Los Angeles, CA, USA, Jul. 2017.

[23] D.-Y. Li and Y. Du, *Artificial Intelligence with Uncertainty*. Boca Raton, FL, USA: Chapman & Hall, 2007.

[24] S.-L. Wang, W.-Y. Gan, D.-Y. Li, and Y. Du, ''Data Field for Hierarchical Clustering,'' *Int. J. Data Warehousing Mining*, vol. 7, no. 4, pp. 43–63, 2011.

[25] S. Wang, J. Fan, M. Fang, and H. Yuan, ''HGCUDF: Hierarchical grid clustering using data field,'' *Chin. J. Electron.*, vol. 23, no. 1, pp. 37–42, 2014.

[26] S. Wang and Y. Chen, ''HASTA: A hierarchical-grid clustering algorithm with data field,'' *Int. J. Data Warehousing Mining*, vol. 10, no. 2, pp. 39–54, 2014.

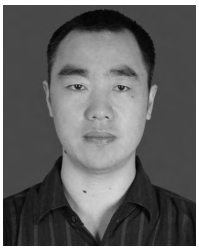
[27] T. Wu and K. Qin, ''Image data field for homogeneous region based segmentation,'' *Comput. Elect. Eng.*, vol. 38, no. 2, pp. 459–470, 2012.

[28] T. Wu and K. Qin, ''Data field-based transition region extraction and thresholding,'' *Opt. Lasers Eng.*, vol. 50, no. 2, pp. 131–139, 2012.

[29] T. Wu and K. Qin, ''Data field-based mechanism for three-dimensional thresholding,'' *Neurocomputing*, vol. 97, pp. 278–296, Nov. 2012.

[30] J. Zhao and M. Jia, "Segmentation algorithm for small targets based on improved data field and fuzzy *c*-means clustering," *Optik*, vol. 126, no. 23, pp. 4330–4336, 2015.
[31] T. Wu, "Image data field-based framework for image thresholding," *Opt. Laser Technol.*, vol. 62, no. 10, pp. 1–11, 2014.
[32] V. Ostromoukhov, "A simple and efficient error-diffusion algorithm," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, Los Angeles, CA, USA, Aug. 2001, pp. 567–572.
[33] H. Li and D. Mould, "Contrast-aware halftoning," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 273–280, 2010.

TAO WU (M'11) was born in 1980. He received the Ph.D. degree in computer science and technology from Wuhan University, in 2012. He is currently a Professor with Lingnan Normal University. He has authored three books and more than 50 articles. His research interests include computational aesthetics, generative art, intelligent digital image processing, and computer vision.



JUNJIE YANG was born in 1969. He received the Ph.D. degree in computer technology and its applications from the Huazhong University of Science and Technology, in 2006. He was a Postdoctoral Researcher, from 2006 to 2009. He is currently a Professor with the Guangdong Engineering and Technological Development Center for E-learning. His research interests include intelligent optimization algorithm and computer vision.

XUE QIN was born in 1980. He received the Ph.D. degree in computer engineering from Wuhan University, in 2013. He is currently an Associate Professor with Guizhou University. His research interests include data mining, semantic web service, and visual arts.