A Survey of Digital Stippling

Domingo Martín, Germán Arroyo, Alejandro Rodríguez, Tobias Isenberg

HAL Id: hal-01528484
https://inria.hal.science/hal-01528484

Submitted on 29 May 2017

A Survey of Digital Stippling

Domingo Martín[1]   Germán Arroyo[1]   Alejandro Rodríguez[1]   Tobias Isenberg[2]

[1] *Department of Software Engeneering, University of Granada, Spain*   [2] *Inria and Université Paris-Saclay, France*

Abstract

In this article we survey techniques for the digital simulation of hand-made stippling—one of the core techniques developed within non-photorealistic/expressive rendering. Over the years, a plethora of automatic or semi-automatic stippling algorithms have been proposed. As part of this expanding field of research, techniques have been developed that not only push the boundaries of traditional stippling but that also relate to other processes or techniques. Our general goal in this survey is thus to increase our understanding of both hand-made and computer-assisted stippling. For this purpose we not only provide an overview of the work on digital stippling but also examine its relationship to traditional stippling and to related fields such as halftoning. Finally, we propose several directions of future work in the field.

Keywords: Stippling, NPR stippling, hand-made stippling, halftoning

1. Introduction

Stippling is the art of creating visual representations (primarily) with dots. As it is the case with other artistic techniques, there has been a great interest in trying to reproduce this form of artistic expression using computers. This interest has lead to the development of a number of techniques in the past years [1, 2]. Through the process of developing computer-aided stippling techniques we also have gained a better understanding of the characteristics of traditional stippling. In some cases, the development in this area has even created new ways of expression that have previously not been possible (e. g., animated stippling of 3D shapes [3], zoomable stipple images [3, 4], stippling with dedicated/arbitrary primitives [5, 6], and stippling of volumetric models [7, 8]). In other cases, however, the borders between stippling and other techniques have also become somewhat fuzzy, leading to potential confusion. The difference between stippling on the one side and halftoning or computer-generated noise on the other side, for instance, can sometimes be difficult to see. For a healthy evolution of the field we believe that it is necessary, therefore, to establish a clear taxonomy of the different stippling approaches, methods, and techniques. This taxonomy will allow researchers to guide their work, will facilitate the comparison of their work with that of others, and will allow us to ponder future lines of research. This taxonomy will also allow practitioners to select the technique that best fits the requirements of their specific application case.

In comparison to other taxonomies, we want to study the different methods from a qualitative point of view, not focusing in how the methods work [2] but instead what can be achieved with them. This approach will allow us to specify a set of important criteria that can be used to classify the different methods. For example, we are less interested in explaining how Lloyd's method [9, 10] works than we are keen to infer that it produces evenly distributions in which the stipple points generally do not overlap[1]—yet that it frequently produces visible regularity artifacts. Also, we want to be able to study when and how stippling can be compared to halftoning [1].

The main contribution of this paper—in addition to providing the most comprehensive survey of stippling techniques to date—is thus the identification of qualitative criteria by which we classify the different approaches. While we integrate Martín et al.'s [14] recent classification of stipple dot characteristics in our work, we formulate a much more comprehensive stippling taxonomy by distinguishing many aspects of four high-level criteria: the point distribution, the specific stipple dot characteristics, the reproduction of the stippling, and questions of the goals and application domain. This taxonomy allows practitioners to select a specific approach depending on their constraints. Finally, we also examine and evaluate the approaches that have been used to measure the quality of digital stippling.

It is important to note that the concept of stippling can be interpreted in many ways, from a rather purist notion related to traditional hand-made artworks to a view that stippling is, broadly, painting with dots. In an inclusive effort, we include all the possible notions in this survey, allowing the readers to form their own opinion.

The remainder of our survey is structured as follows. We first briefly describe the art and craft of traditional stippling in Section 2 as it is the foundation of any digital
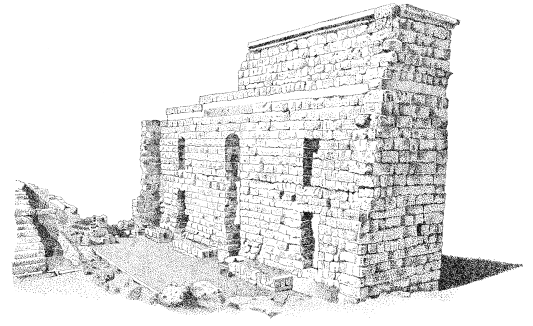
---

[1]The non-overlapping of stipple points was a goal in early stippling approaches that used Lloyd's method (e. g., [5, 11]), but depending on their chosen size dots can end up overlapping each other (e. g., [12, 13]).
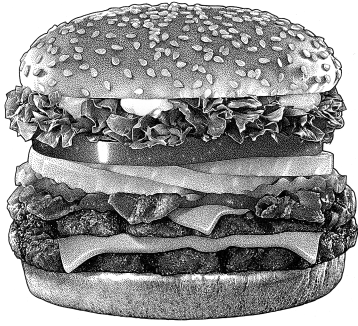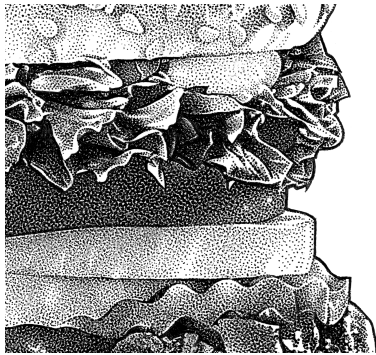
(a) Hand-made example. © Randy Glass. (b) Hand-made example. © Miguel Endara. (c) Hand-made example. © Elena Piñar.

(d) Detail. © Randy Glass. (e) Detail. © Miguel Endara. (f) Detail. © Elena Piñar.

Figure 1: Examples of hand-made stippling (some of the images have been down-sampled). All images used with permission.

reproduction. Then, we discuss the aspects and design choices that govern digital stippling in Section 3, leading to a list of criteria for our taxonomy of digital stippling. Next, we survey the individual techniques in Section 4 and provide a table that classifies each technique according to the taxonomy. After a brief analysis of our findings in Section 5 we discuss the resulting implications for the relationship between stippling and halftoning in Section 6. Section 7 then describes our final contribution—the software suite StippleShop which implements several of the techniques—, before we conclude the paper in Section 8.

2. A general discussion of traditional stippling

To be able to successfully reproduce a technique such as traditional stippling (e. g., Figure 1) it is essential to first get a clear understanding of its characteristics. We thus begin by emphasizing that an important aspect of traditional hand-made stippling is that it is an art form. Stipple artists—including those who specialize in technical illustration—not only aim to faithfully represent the depicted objects but they also strive to transmit information through the "filter" of an artist to make their images understandable. This artistic aspect distinguishes stippling from other reprographic techniques that also aim at reproducing tone on bi-level devices, but which do so in an algorithmic way.

On a low/instrumental level [14], the goal in stippling is to represent the tone, the shape, the texture, etc. of the subject matter using dots. Stipple artists thus place dots

made with a pen, typically in black ink, over a clear media, usually white paper. This is a time-consuming process [15] which requires a great number of hours to arrive at the final result.[2] The implication of the tediousness of the process is also that the created artworks are often of rather small size—not comparable to the sizes that, for example, oil paintings can have. Related to the size of the final image is also that it is generally assumed that stipple images are being looked at from reading distance, i. e. from a distance of approx. 35 cm–65 cm. The individual dots are still recognizable at this distance in their shape and color [14]—in fact, stipple artists are instructed in their education that "every dot should have a purpose" [15]. This reliance of stippling on the individual dot can be contrasted to halftoning in which a lot more dots/pixels are placed. Here, each dot/pixel should ideally disappear such that only its contribution to the reproduction of tone is important [16, 17].[3] Based on this observation it becomes clear that it is important for the reproduction (and re-creation) of stippling to take the original sizes and resolutions of the medium and the stipple dots into account [14, 18, 19].

Furthermore, it is important to understand that stipple images are traditionally created as still images which cannot be changed after they have been created. This means

---

[2]See, for example, the creation of an artwork by Miguel Endara at http://vimeo.com/33091687.

[3]This rule may be changed when the halftoning is used as an inherent aspect of artistic expression, for example in some works of pop art such as by Andy Warhol or Roy Lichtenstein.

that one of the most important aspects of stippling is the pattern of dot placement—it conveys the local tone in the image. Stipple artists typically aim to avoid any visible pattern of dots such as chains [15], but some features can be suggested by intentionally placing points in forms of lines [15, 20]. Variations such as hedcut stippling exist in which the dots are placed almost exclusively in the form of dedicated features [21, 22, 23] that have a relationship to hatching styles [24]. In addition, hand-placed stipple dots can also (intentionally) merge into larger clusters in which the individual dots overlap [15, 25]. Stipple artists also frequently combine stippling with other techniques such as lines, for example using stippling to represent the tone and texture and lines to emphasize edges and outlines.

Based on these observations we can now go on to discuss the characteristics of digital stippling as it is represented in the literature. From this discussion we then derive a classification scheme according to which we, later-on, survey the individual stippling techniques.

3. Digital stippling

Digital stippling, like hand-made stippling, also uses dot-like primitives to represent an image. Yet, while it is inspired by hand-made stippling, digital stippling has more flexibility because several processes can be extended and/or improved: no human has to manually place the dots, we are not restricted to pens making a certain type of dot, and we can take advantage of the speed of computers. This means, for example, that many more stipples can be placed within the same amount of time, the produced images no longer need to be static, we can experiment with distribution schemes and dot shapes, and we can combine stippling easily with other rendering types such as drawing important lines.[4] At the same time, other challenges arise, for example in the form of the constraints of output media (displays, printers, plotters) and the need to encode the long experience of stipple artists of where and how to place which kinds of dots into an algorithm.

Ultimately, the question of which goal one wants to achieve with digital stippling and for which application domain the stippling is to be used [30] is thus guiding the selection of a specific algorithm. To better be able to classify the different algorithms we thus distinguish the following four high-level criteria for our taxonomy of digital stippling: (1) the placement strategies of dots primitives, (2) the specific character of the placed primitives, (3) the techniques used for the reproduction of the generated image(s), and (4) the intended application domains. We first discuss these different criteria in general in this section, before surveying the complete literature in Section 4.

---

[4]For more information on the latter see, for example, the surveys on sparse line drawings [26, 27], silhouettes/occluding contours [28], and feature lines [29].

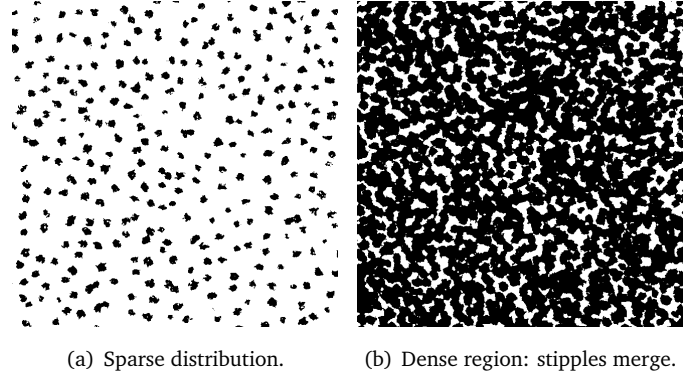(a) Sparse distribution. (b) Dense region: stipples merge.

Figure 2: Details from a manually created stipple illustration (from Figure 1(c)), showing the merging of stipples for denser regions. The artwork is © 2009 Elena Piñar, used with permission.

3.1. Dot placement

The most obvious quality criteria for digital stippling is the issue of where stipple dots should be placed. This question drove the creation of computer-based stippling techniques from the very beginning because it is also one of the main issues in traditional stippling. Researchers were inspired by the manual technique; so they began by trying to algorithmically reproduce the hand-made example. So they derived their **basic dot placement characteristics** from the originals to be emulated [11, 31]:

- the dots should be placed evenly for a given tone,

- yet the dots should not be placed too regularly, and

- the density of the dots represents the shade/intensity of a given area in the image to be reproduced.

While these characteristics have remained valid for virtually all digital stippling since its beginnings, additional aspects have also started to play a role. For example, in the first approaches researchers implicitly assumed that the placed stipple dots should not overlap. Later it was shown to the NPAR community [25], however, that certain traditional stipple techniques rely on the **overlapping and merging of stipples** for certain tonal ranges (Figure 2). This insight led to some techniques being devised that support overlapping and merging stipple dots [18, 19, 32, 33].

Another characteristic of early stippling was the creation of **unwanted regularity artifacts** [34] of the placed stipple points (Figure 3), due to the inherent characteristics of Lloyd's method [9, 10] and the stippling that used its unweighted [11, 31] and weighted [12] centroidal Voronoi diagrams for dot distribution. These artifacts are undesirable because they suggest structures where there are none—stipple artist train themselves to avoid them as outlined in Section 2. Several techniques have been created since then to avoid the unwanted artifacts such as the use of randomness/noise [18, 19], probability density functions [32, 33, 35], Poisson disc distributions [4, 7, 36, 37, 38], non-repetitive tiling [4], capacity-constrained relaxation [34], or example-based dot placement [25].
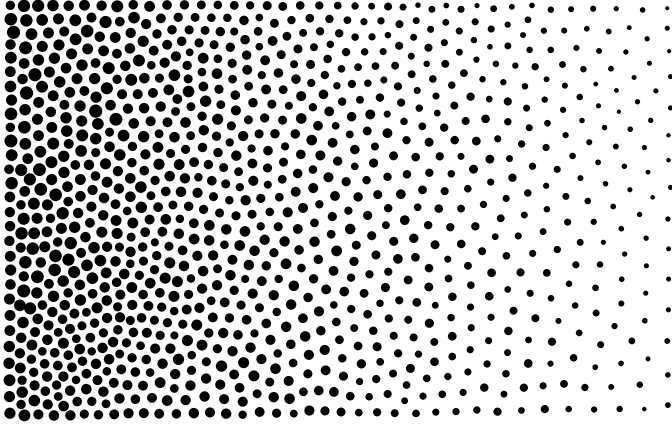
3

Figure 3: Unwanted regularity artifacts (visible point chains) due to Lloyd's method [9, 10] which relies on centroidal Voronoi diagrams.

In particular, the use of **example-based stipple dot placement** has advantages for the goal of faithfully replicating manual stippling. Because this approach attempts to copy distribution characteristics from real examples [25] it can not only capture the style of individual stipple artists but it also avoids the impression of being 'too perfect' that purely **algorithmic stipple placement** may convey—even if the algorithmically derived distribution of the dots satisfies established (mathematical) criteria. Poisson disc sampling, for example, produces distributions with an even yet random distribution of dots, while ensuring that a minimum distance is always maintained [10, 37]. This distribution has blue noise spectral characteristics, which is considered to produce visually pleasing patterns. In dark regions, however, less flexibility for random placement exists due to the dots' density—resulting in the impression of an automatic placement. In manual stippling, in contrast, artists occasionally make 'mistakes' even in such dense regions and dots may overlap—the resulting distributions thus appear less perfect and, consequently, more 'natural.'[5]

Example-based stipple dot placement can be one way to more closely match a hand-drawn target style. To be able to truly match it, however, it is also essential to take the original sizes of the examples as well as the size and the resolution of the digital output medium into account [18, 19]. Such **resolution-dependent stippling**, therefore, no longer provides the freedom of choosing an arbitrary number of stipple points but derives this number from the spatial size (e. g., A5) of the intended output image, based on the stipple density of the hand-made example. The intended output pixel resolution is then also used to obtain the matching stipple dot templates in an example-based and resolution-dependent way (see details in Section 3.2). Moreover, only resolution-dependent stippling techniques that are guided by hand-made examples can produce distributions that an artist would consider to be 'ideal' for reproducing a given tone, where the correct density depends on the 'copied' artists, the tone, and even the specific pens and paper of the exemplar [14].

---

[5]Natural appeal relates to the concept of imperfect symmetry [39].
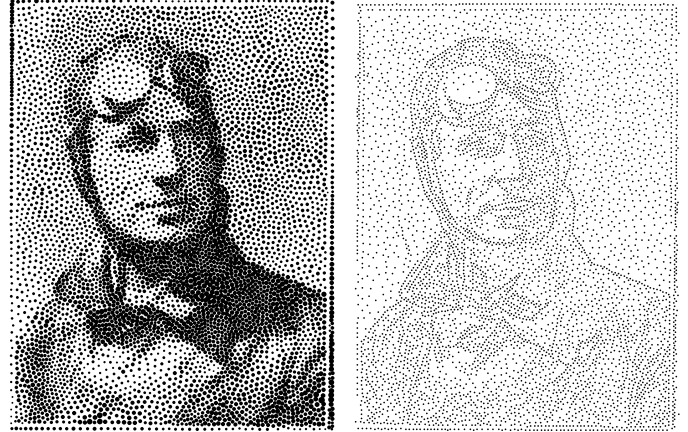
(a) Centroidal Voronoi stippling [12].   (b) Structure-aware stippling [20].
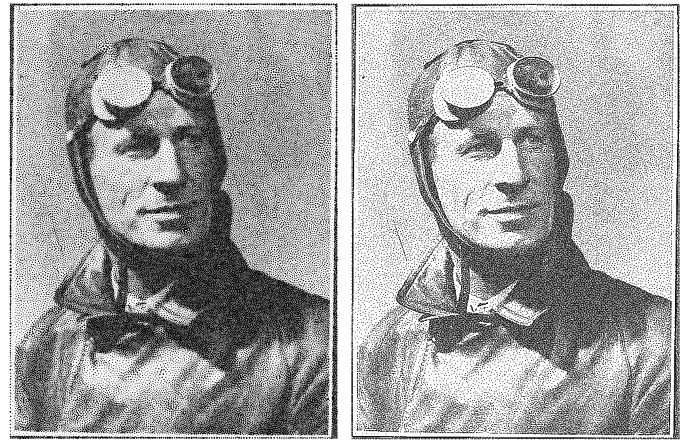
(c) Centroidal Voronoi stippling [12].   (d) Structure-aware stippling [40].

Figure 4: The varying support of intentional artifacts in stippling, for low (top, 5,869 stipples) and high (bottom, 46,014 stipples) stipple counts. Weighted centroidal Voronoi stippling (a),(c) can arrange the stipples somewhat along linear structures if the source image has a high contrast, but dedicated algorithms [20, 40] can specifically arrange stipples to better emphasize such structures in images (b),(d). Images (b),(d) courtesy of and © 2016 David Mould, used with permission.

In contrast to the basic stippling principle of evenly placing dots while avoiding unwanted artifacts, sometimes it may be important to introduce **intentional artifacts** into a stipple image. For example, the chain artifacts that can be observed in Figure 3 can also be intentionally used for emphasizing linear features such as edges in an image or lines. Most digital stippling techniques automatically arrange the dots more or less along features when some precautions are taken. For example, an early form was already described by Deussen et al. [11, 31] who used tone thresholds to achieve such effects. In weighted Voronoi stippling [12], edges can be achieved by increasing the contrast of the source images. More recently, however, several approaches [20, 40] facilitated the emphasis of dedicated linear structures in digital stippling by intentionally placing stipple dots along such lines as part of the algorithm (Figure 4). A special form of stippling that almost exclusively relies on the stipple dots being arranged along linear structures is **hedcut stippling**. Several authors have created techniques

4

Figure 5: Frames from a continuous and seamless zoom into a stipple image using the Wang tiling approach by Kopf et al. [4].
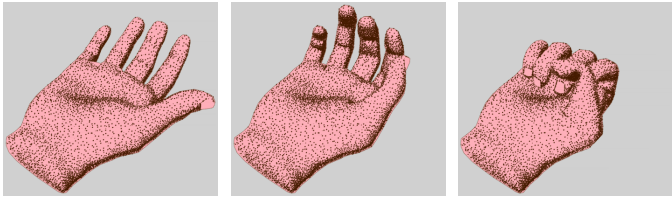

Figure 6: Three frames from an animation of a stippled 3D model [3]. Images © 2003 and courtesy of Oscar Meruvia Pastor, used with permission.

[21, 22, 23] to replicate this traditional illustration style that focuses on the depiction of portraits.

The goal of the techniques and placement aspects discussed so far has been an increasingly faithful replication of the original hand-made stippling. There may, however, also be other goals for digital stippling. For example, a stippled illustration with an unrealistically high number of stipple dots can convey the impression of precision for the depicted objects [41]. When we drop the constraint of producing a still image as output, moreover, we open up a plethora of possibilities for digital-only stippling. For example, one may want to be able to interactively *zoom* into stippled images to see more or less detail. This goal requires a means to add and remove stipples at appropriate points without introducing visible spatial or temporal artifacts as demonstrated by Kopf et al. [4] (Figure 5).

Alternatively, one may drop the constraint of the flat, two-dimensional stippling domain and allow ***stippling directly on 3D objects*** (e. g., [3, 8, 36, 42, 43]). 3D shapes, however, are typically explored in an interactive way. Beyond simple animations of 3D shapes it may also be desirable to ***animate the 3D shapes themselves*** [3, 36] (e. g., see Figure 6) or, in the two-dimensional case, to ***animate the stippled 2D image*** (i. e., perform video stippling) [36].[6] In all these cases it is essential that the 2D or 3D

stippling approaches—similar to the previously mentioned zooming into a stippled 2D image—also need to support the smooth increase or decrease of stipple point densities to maintain the perceived tone levels despite the animation, provided that our goal is to keep the stipple dot size constant. Moreover, for all these animations it is essential to maintain frame coherence[7] as well as to avoid the "shower door effect"[8] for stippled 3D scenes or videos.

3.2. Dot/primitive character

In Martín et al.'s [14] recent discussion of stipple dot characteristics the authors examine the dot shapes, color, texture as well as the resolution dependence of the dots used in digital stippling. While we keep this classification, we treat dot shape as a single criterion and its color and/or texture as a second, independent aspect. The reason is that color and/or texture of the dots have only recently been introduced into the field—initially most digital stippling approaches simply assumed all dots to be black marks on white paper. Moreover, many if not most early approaches simply placed ideal ***circles as stipple dots*** (Figure 7(b)).

A first move away from this circle-based stippling was the use of dedicated, ***arbitrary stipple dot shapes*** (Figure 7(c)). While circle-based approaches could certainly exchange the circles for other shapes (e. g., [11, 35]), some approaches took the shape of the dots into account when deriving their placement. For example, Hiller et al. [5] extended Lloyd's method to take the shapes of the stipple marks into account, while Dalal et al. [6] used a frequency-based packing approach for the placement of their arbitrary dot primitives—both with the result of extending stippling beyond the traditional manual technique. Others explored the creation of general distributions of patterns in the plane (e. g., [46, 47]—techniques which can also use example-based primitive placement but which are not typically being considered as digital stippling) or placed short marks that resemble those of a certain type of hatching illustrations [35, 48, 49, 50] for which overlapping was desired and, therefore, did not require a shape-aware processing. Similarly, illustrative visualization techniques that made use of stippling to show directional information, in particular for neuro-scientific visualizations [38, 51, 52], made use of 'stipple' marks that resembled short hatching strokes (see the examples in Figure 8).

As an alternative to circles or arbitrary shapes, some researchers interpreted the stipple dot simply as a fundamental dot primitive and thus used the most simple digital mark possible—the ***pixel*** (Figure 7(a); e. g., [7, 8, 25, 42, 49, 53, 54, 55]). This approach has advantages, for example, in the illustration of volumetric data (e. g., [7, 8, 54]) because no elaborate computation is necessary to render pixel dots. Also, it provides a possibility to reduce stippling

---

[6]In fact, 3D animations could also be seen as pre-captured videos in which the stipples are then processed in a purely 2D image-space process, but this leads to the "shower door effect" [36].

[7]Frame ***incoherence*** can also intentionally be used as an artistic form of expression [36], such as in certain cartoons (see, e. g., http://cartoons.redbull.com/).

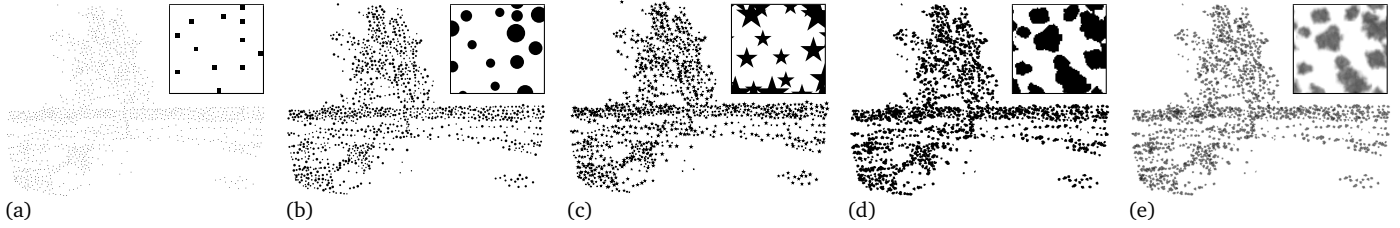[8]For an explanation of the shower door effect see [44, 45].

Figure 7: Possible stipple dot characteristics: (a) stipples are pixels, (b) stipples are circles (or blobby shapes), (c) stipples are arbitrary shapes, (d) stipples are example-based binary black-and-white scans of real stipples, or (e) stipples are example-based grayscale/color scans of real stipples.
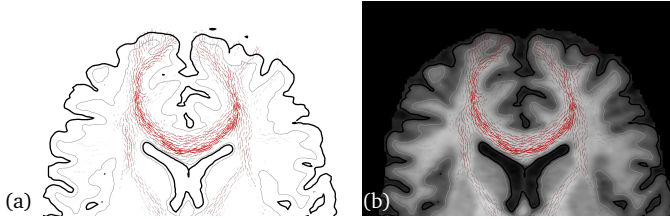

Figure 8: Goldau et al.'s [51, 52] fiber stippling technique for the visualization of probabilistic tractography data, (a) with context in form of silhouettes computed from T1-weighted MRI data and (b) as a multimodal visualization with context shown in form of T1-weighted MRI data texture-mapped onto a cutting plane. Images © 2014 and courtesy of by Mathias Goldau, used with permission.

to only its distribution and, at the same time, facilitates comparably high stipple point densities [25].

The opposite research direction, in contrast, is to not only take the stipple dot placement into account, but to try to base also the stipple dot shapes on examples from hand-made stippling. The first approach in this direction of *example-based stipple dots* simply used a library of scanned stipple dots from an exemplar [25]—in a binary black-and-white form (Figure 7(d); also see the scan from a real artwork in Figure 2 that uses the same approach). More recent work, however, argues that not only should also the *grayscale or color textures* of the example dots be taken into account [32, 33] (Figure 7(e)),[9] but also the *resolution-dependence of the dots* [18, 19]—just like for the dot distributions. Martín et al. [14] even show that these dot exemplars depend on the paper type and the pen that the artist has used.

3.3. Reproduction/representation of the result

The ultimate goal of any digital image generation process is, of course, to look at and appreciate the images in the end. Therefore, we need to take the possible reproduction mechanisms into account. Two general ways for image reproduction are currently available to us: computer displays and printing.[10] It is essential to realize, however, that these two means of reproduction have vastly different properties, with important implications for their potential use to show the results of digital stippling:

pixel/dot resolution: Current computer displays reach up to 806 ppi for small devices (Sony Z5 Premium; as of 2016), but modern desktop monitors have resolutions in the order of 150 ppi (e. g., a 30" 4K display has 147 ppi).[11] Printing, in contrast, can typically achieve 1200 dpi for laser printers, professional inkjet printers can reach 4800 dpi × 2400 dpi and more, and professional imagesetter hardware can reach resolutions of 2400 dpi and above. A more in-depth discussion can be found in Martín et al.'s [14] work.

color resolution: Computer displays typically have 8 bit resolution for each of the three color channels, with modern HDR displays increasing this resolution to 10 bit or more. Printers, in contrast, only have a single bit per color: they either place an ink dot or not.[12] Any color gradient thus needs to be produced by halftoning mechanisms [16, 17], thus decreasing the available resolution and potentially leading to visible artifacts when reproducing stipple dots when printing at a resolution that is too low [14].

display speed: Computer displays are capable of changing the shown image quickly, with refresh rates of 60 Hz or above being possible. Such setups thus allow dynamic content and interactive exploration. Printed images, on the other hand, do not have this capability,

---

made by a plotter would be different from the previous, so they would only be able to reproduce stipple dot distributions reliably. Related to plotters are 'drawing robots' (e. g., [59, 60, 61, 62]), but these have similar dot reproduction issues as plotters and are even less readily available. Finally, some experiments with stippling via aerial drones have been carried out [63], with the result that reproduction of the dot distribution also became an increasingly random process.

[11]Desktop displays with up to 280 ppi have recently been announced such as Dell's 32" 8K UltraSharp UP3218K.

[12]The less readily available class of dye-sublimation printers can produce continuous tone in a similar way as a monitor, based the CMY color model. They are normally used to produce true color photos. Their resolutions reach 1200 ppi (e. g., Virtuoso SG400) and 1440 × 720 ppi (e. g., Roland Texart RT-640). Another rare class are RGB digital printers which are used as a replacement of the classical photo development. Here, photo paper is exposed to RGB lasers and is then chemically developed. RGB digital printers produce true color images but usually at low pixel density at approx. 300 ppi and up to 600 ppi (e. g., Fujifilm Frontier LP5500).

---

[9]Some of the early stippling approaches also used grayscale (e. g., [54, 56]) or colored stipples (e. g., [57]). Other approaches rendered black stipples into pixel images with a white background [48, 49, 57, 58], which led to gray tones due to the use of anti-aliasing.

[10]Plotters are yet another means with even some interesting properties for stippling due to the use of a pen-like tool to make marks. However, plotters are only used in special application scenarios today and are no longer generally available. Also, like in hand-made stippling, each dot

their image cannot be changed after it is finished.[13] Such images can thus only be explored by (manually) looking closer at parts of the printed, static image.

Based on these considerations, early stippling thus focused virtually exclusively (with only few exceptions [8, 56]) on black stipple marks on a white background. This had the advantage of an easy reproduction also on bi-tone printers as no halftoning was necessary which would lead to visible artifacts on 1200 dpi printers. Only later were grayscale or color-based stippling techniques developed as outlined in Section 3.2, with the implication that their results can only truthfully be reproduced on ≥2400 dpi printers or (with limitations depending on the pixel resolution) on computer displays [14]—if the goal is a faithful emulation of traditional stippling.

If this goal/constraint is partially dropped, however, the reproduction is more flexible. In particular when restricting oneself to 1 bit black-and-white images/reproduction, potentially with example-based stipple point shapes/outlines, reproduction is possible on both regular printers and computer displays. Normal 1200 dpi printers can deal with such images because no halftoning is necessary, while computer displays can interpolate (sub-sample) the dot shapes with the background by taking advantage of their increased color resolution. Moreover, computer displays can also enable the advanced stippling techniques that provide zooming or 3D rendering mentioned in Section 3.1.

A restriction to binary 'tone' also has an advantage for the representation of the resulting images: they can both be efficiently encoded in suitable **pixel-based or vector-based file formats**. Vector formats can represent the shapes/outline of the stipple dots in a scale-independent way, in particular, for simple stipple dot shapes and can have advantages for the size of the produced file [64]. Pixel-based images, in contrast, make it easier to represent example-based stipple shapes and can lead to similar on-screen and print reproduction qualities if high pixel resolutions are combined with 1 bit color scales to avoid interpolation and halftoning.

3.4. Goals and applications

As with any other non-photorealistic, expressive, or illustrative rendering technique [30], with stippling one can also have a **variety of goals** that highly depend on the intended application domain. We have mentioned some of these goals already. For example, one goal could be the **emulation of the traditional art of hand-made stippling** or of the special case of hedcut stippling, with a focus on two-dimensional still images. Such a technique could

have a number of application domains. One of these is the creation of tools for **professional artists and illustrators** who would be freed from the tedious task of placing the individual dots. In this case, however, the tool should not only produce high-quality output but also allow both higher-level forms of interactive control [18, 19] as well as an interaction continuum that also allows the artists to adjust individual dots [30]. After all, many decisions by illustrators and artists as they produce their artwork currently cannot be automated.

Another application domain for the same goal is the use of stippling as a medium for people who are not able or willing to hire a professional illustrator. For example, for personal purposes one may enjoy a stipple effect in an image processing tool like there are many other similar NPR tools integrated into today's image processing suites. Alternatively, stippling could potentially be used in form of an automatic illustration filter, for instance when (mass-) producing medical illustrations for patients.

A different goal is the use of the technique in computer-based **environments that can be interactively explored** in which a stippled image is not the final product but where stippling is, instead, a style element. For such application domains we have different constraints for a stippling algorithm. For example, a truthful replication of the traditional art form and thus, for example, the specific form of the dots may be less important and the characteristics of the point distributions may be more fundamental. Such application domains will typically also not need a resolution-dependent distribution since the final visuals will not likely be guided by the size of a traditional artwork.

Depending on the specific goal, the actual stippling algorithm has to support different **working procedures** as well as **forms of interactivity**. Tools for artists, for example, should not only be fully automatic but support a semi-automatic guidance of the stippling process. At the same time, the processing has to be fast to support real-time responses, even if only a still image is produced. In other application domains the users may have less artistic skill, thus an automatic procedure would be sufficient. For animated results the processing has to take frame coherence and 2D/3D aspects such as the 'shower door effect' into account. Yet other applications even aim toward interactive exploration of the stippled objects so that real-time processing is essential.

3.5. A taxonomy of stippling approaches

Based on the considerations in Sections 3.1–3.4 we can now formulate our taxonomy of digital stippling approaches. As shown in Table 1, we distinguish, on a high level, between the topics of the four subsections. For each of them we add a number of specific criteria that were discussed in the different subsections. With this taxonomy in place we now continue to discuss important aspects of the individual stippling techniques in the next section.

---

[13]e-Ink displays that are becoming increasingly available are somewhat in-between printed images and computer displays, with lower refresh rates than regular displays, true grayscale reproduction without halftoning, but only with a low (currently 4 bit) 'color' channel resolution. Their pixel resolution, however, only reaches up to 300 ppi today—too low for a proper reproduction of most forms of digital stippling if the goal is a faithful emulation of the hand-made examples [14].

Table 1: Comparison of stippling techniques according to the four main groups of characteristics outlined in Section 3. ***Dot distribution:*** **process to place dots** (Wang Tiling (**WT**)/Dart Throwing (**DT**)/Lloyd's process(**Ll**)/Example Based (**EB**)/Error Diffussion (**ED**)/Other (**O**)), **distribution type** (Example-Based/Algorithm-Based), **resolution dependence** (Y/N), the **support of feature-oriented stippling** (Y/N), the **support for zooming** (Y/N), **dimension of the space in which the stipples are distributed**, connected to the dimension of the input data (2 for a 2D image and 3 for a 3D model (Volume=V, Surface=S, Point-sampled=P, Implicit =I), and the **support for animation** (i. e., tone/shape changes) of the stippled images/objects (Y/N). ***Dot characteristics:*** the **stipple dot shapes** that are supported (Pixel=P/ideal Circle=C/Rounded blob=R/Example-Based dots=EB/Arbitrary primitives=A), the **stipple color** (Black & White=BW, Grayscale=G, Color=C), and the **resolution dependence of the dots** (Y/N). ***Image reproduction:*** **type of the produced image** (Pixel=P/Vector=V; note that vector images can, of course, also be rasterized into pixel images). ***Application domain:*** the **goal of the approach** (Traditional Stippling=TS, Hedcut Stippling=HS, Illustration=I, interactive Exploration=IE), the **procedure** used in creating stipple images (Automatic=A, Semi-Automatic=SA, Manual=M), and the **result characteristics** (Static image=S, Animation=A, Real-Time interaction=RT). Notes: ?—uncertain; ⊛—limited by the resolution of the model; ◎—2D geometry-based space is obtained from a 3D model; and ⊕—the synthesized data is adjusted to the output size.

article/paper	dot distribution							dot characteristics			reprod.	application domain		
	dot placement process	distribution type	resolution dependency	feature-oriented stippling	support for zooming	dimensionality of stipple space	support for animation	supported stipple dot shapes	stipple color	resolution dependence	produced image	goal of the approach	procedure	result characteristics
Deussen et al. [11, 31]	Ll	AB	N	N	N	2	N	C/A	BW	N	V	TS	SA	S
Secord [12]	Ll	AB	N	N	N	2	N	C	BW	N	V	TS	SA	S
Secord et al. [35]	O	AB	N	N	N	2&3/S	Y	C/A/P	BW	N	P	I	A	A
Lu et al. [7]	O	AB	N	N	Y	3/V	Y	P	BW	N	P	I	A	A
Lu et al. [42]	O	AB	N	N	Y	3/S	Y	P	BW	N	P	I	A	A
Meruvia Pastor & Strothotte [57]	O	AB	N	N	Y	3/S	Y	C	BW	N	P	TS	A	A
Lu et al. [8]	O	AB	N	N	Y	3/S+V	Y	P	C	N	P	I/E	A	A/RT
Meruvia Pastor et al. [3]	O	AB	N	N	Y	3/S	Y	C	BW	N	P	TS/E	A	A
Hiller et al. [5]	Ll	AB	N	N	Y	2	Y	A	BW	N	V	TS/I	A	S
Meruvia Pastor & Strothotte [65]	O	AB	N	N	Y	3/S	Y	C	BW	N	P	TS/I	A	A
Xu and Chen [56]	O	AB	N	N	Y⊛	3/P	Y	A?	G	N	P	I	A	A
Zakaria and Seidel [53]	O	AB	N	N	?	3/P	?	P	BW	N	P	I	A	A
Schlechtweg et al. [58, 66]	O	AB	N	Y	N	2	N	C	BW	N	V	TS/I	A	S
Foster et al. [48]	O	AB	N	N	N	3/I	N	C/A	BW	N	P	TS/I	A	S
Hausner [67]	Ll	AB	N	Y	N	2	N	A	C	N	P	TS/I	A	S
Yuan et al. [49]	O	AB	N	N	Y	2◎&3/S	Y	P/A	BW	N	P	TS/I	A	S/A
Smith et al. [68]	Ll	AB	N	N	N	2	Y	A	C	N	P	?	A	A
Kopf et al. [4]	WT	AB	N	N	Y	2	Y	C	BW	N	V	TS/I/E	A	RT?
Dalal et al. [6]	Ll+O	AB	N	N	N	2	Y	A	BW?	N	P	TS/I	A	S/A
Barla et al. [50]	-	EB	N	N	N	2	N	A	BW	N	V	I	A	S
Barla et al. [46]	-	-	-	-	-	-	-	-	C	-	-	I	A	S
Mould [20]	O	AB	N	Y	N	2	N	C	BW	N	V	TS/I	A	S
Krüger and Westermann [54]	O	AB	N	N	N	3/S+V	N	P/C	BW/G	N	P	E	A	RT
Schmidt et al. [55]	O	AB	N	N	N	3/I	N	P	BW	N	P	TS/I/E	A	RT
Vanderhaeghe et al. [36]	DT	AB	N	N	Y	2&3/S	Y	C	BW	N	P	TS/I	A	A
Kim et al. [21]	-	AB	N	Y	N	2	N	C	BW	N	P	HS	A	S
Kim et al. [25]	EB	EB	Y	N	N	2	N	P/EB	BW	Y⊕	P	TS	A	S
Kim et al. [22]	-	AB	N	Y	N	2	N	C	BW	N	V	HS	A	S
Martín et al. [18, 19]	ED	EB	Y	N	N	2	N	EB	G	Y	P	TS	SA	S
Arroyo et al. [32, 33]	O	AL	N	N	N	2	N	R	G/C	N	P	TS/IE	SA	S
Ascencio-Lopez et al. [37]	O	AB	N	N	N	2	N	C	BW	N	P	TS/I	A	S
Li and Mould [40]	ED	AB	N	Y	N	2	N	C	BW/G/C	N	V	TS/I	A	S
Son et al. [23]	-	AB	N	Y	N	2	N	R	G	N	P	HS	A	S
Martín et al. [14]	-	-	-	-	-	-	-	EB	G	Y	P	-	-	S
distributions with BN properties	-	AB	N	N	Y/N	2	Y/N	-	-	-	-	-	-	-
traditional hand-made stippling	-	-	Y	N	N	2	N	-	G	-	P	TS	M	S
hand-made hedcut stippling	-	-	Y	Y	N	2	N	-	G	-	P	HS	M	S
clustering halftoning (AM)	O	-	-	N	N	2	N	A	BW	-	P	-	A	S
green noise halftoning (GN)	ED+O	-	-	N	N	2	N	A+P	BW	-	P	-	A	S
error diffusion halftoning (FM)	ED	-	-	N	N	2	N	P	BW	-	P	-	A	S

8
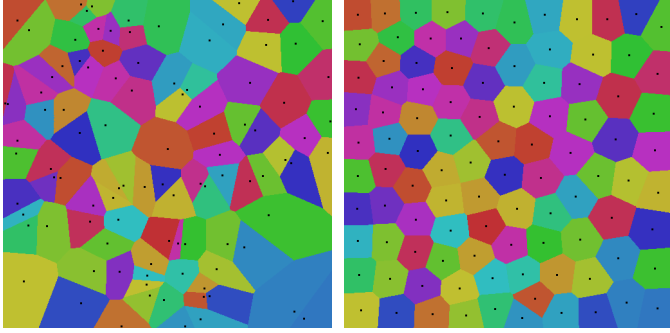
(a) An initial distribution. (b) Converged distribution.

Figure 9: Principle behind Lloyd's method [9, 10]: the iterative computation of centroidal Voronoi diagrams.
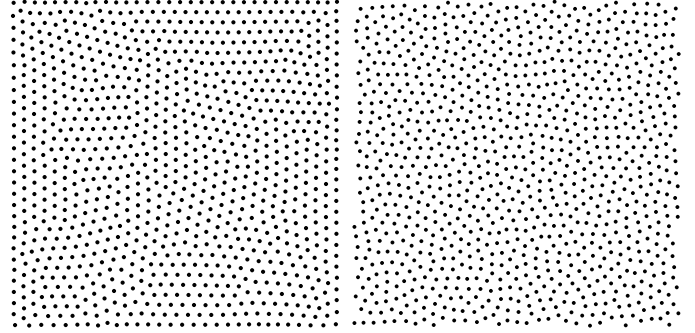

(a) Result from Lloyd's method. (b) Capacity-constrained distribution.

Figure 10: Comparison of the result of Lloyd's pure method with a result using Balzer et al.'s [34] capacity-constrained variant of it.

4. Review of the individual stippling techniques

We group the discussion of the different approaches roughly based on which type of dot distribution they use and/or the goal that they are trying to support: First, Section 4.1 covers approaches based on Lloyd's method [9, 10] in the 2D plane. Second, Section 4.2 describes techniques that are based on other distribution functions in the 2D plane. Then, Section 4.4 discusses methods that rely on 3D models. Next, we review example-based approaches in Section 4.5, structure-guided stippling in Section 4.6, and the special form of hedcut stippling in Section 4.7. Finally, we mention a number of other techniques in Section 4.8 that do not fit into the previous classification, before Section 4.9 surveys those papers that not so much introduce a new stippling technique but which analyze, measure, or evaluate digital (and hand-made) stippling.

4.1. Stippling in 2D based on centroidal Voronoi diagrams

The main goal of the approaches in this first category is to evenly distribute stipple dots, while avoiding them to come too close to each other. Lloyd's method [9, 10] provides a mechanism to achieve this goal. It is based on computing the Voronoi diagram of an initial point distribution, and then to move the dots to the centroids of the computed Voronoi diagram. This *relaxation* process is iteratively repeated, leading eventually to an even dot distribution (e. g., Figure 9) in form of a centroidal Voronoi diagram (CVD)—albeit with some chain artifacts as discussed in Section 3.1.

Deussen et al. [11, 31, 69] were the first to apply this approach to stippling. They start with an initial point distribution derived from a halftoning technique to have dense regions for dark parts of the image, and less dense regions for lighter sections. They then apply CVD-based relaxation locally using manually controlled brushes, for two main reasons: (1) the computation of the CVD was quite slow on the hardware of 2000, and (2) an unconstrained iteration would lead to a completely even distribution as shown in Figure 9(b)—removing all resemblance to the image to be represented. This property of Lloyd's method, however, can also be used to produce even distributions for the emulation of pointillist painting as demonstrated

by Hausner [67]. Also related to the use of CVD but using them to constrain the placement of mosaic tiles is the work of Hausner [70].

Secord [12, 13] then suggested an essential improvement of Deussen et al.'s approach: to weigh the computation of the Voronoi regions based on the tone of the image that is to be reproduced. The main advantage of the use of such weighted centroidal Voronoi diagrams (WCVDs) is that now the color/gray value of the image is automatically taken into account when iteratively placing the stipple dots—an interactive use of brushes is no longer necessary. Moreover, to address the expensive computation of WCVDs, Secord [12] also described a fast stippling process that is based on pre-computed stipple dot distributions.

While both Deussen et al. [11, 31] and Secord [12] have modulated the stipple dot size based on the gray level to be reproduced, they all still used circular stipple dots. Hiller et al. [5, 71] thus extended the WCVD approach to lines and polygons, using a pixel-based approach to compute the WCVDs. Smith et al. [68] use the same general idea and apply it to animated mosaics, but they can also produce animated stipple renderings with arbitrary shapes this way. Dalal et al. [6], finally, improve on Hiller et al.'s and Smith et al.'s results by using an approach similar to WCVDs, but they determine the distance of the pixels to the perimeter of each tile and minimize it—simplified by employing the Fast Fourier Transform (FFT). The resulting spectrally packed dot distributions place the different shapes such that the perceived tone reproduction is more even than with the 'pure' WCVD approach.

Balzer et al. [34] later addressed one of the major issues of WCVDs—their tendency to lead to regularity artifacts such as visible point chains. They demonstrate that traditional WCVD stippling only leads to these regularity artifacts if the iteration is not stopped in time, yet it seems that there are no ideal points in time to stop the iteration—it highly depends on the number of points used. By introducing a capacity constraint for each dot's Voronoi region during the relaxation in their new technique that converges to an artifact-free distribution, Balzer et al. ensure that each dot has an equal importance in the resulting stipple distribution—similar to the goals of traditional illustrators

(refer back to Section 2). Without regularity artifacts as in 'normal' WCVD results (see a comparison to a regular WCVD distribution in Figure 10), the resulting distributions are thus well suited for algorithmic stippling. Chen et al. [72] later developed a variation of Balzer et al.'s [34] capacity-constrained method using a new energy function to control the process, while de Goes et al. [73] created a variation that reformulated the process as a constrained optimal transport problem that can be optimized.

In a study not primarily aimed at stippling, Deussen [74] explored ways to change Lloyd's method to produce interesting and aesthetically pleasing dot distributions in which the points are not necessarily evenly distributed. Based on an energy value that is controlled by the user, his method is also able to avoid the regularity artifacts, while also allowing clustered point sets to be produced. Because Deussen only aims for producing dot distributions that do not represent an image, however, his approach is thus not really suitable for stippling.

4.2. Stippling in 2D based on distribution functions

One of the major disadvantages of (W)CVD-based stippling is that it relies on a repeated (iterative) relaxation step to determine the final dot distribution, which is computationally expensive—even on fast hardware. Moreover, most (W)CVD-based stippling methods only result in distributions with "suboptimal blue noise characteristics" [34]—while blue noise characteristics are useful for stippling because they indicate a minimum distance between the points and the absence of any dominant directions in the frequency domain [11], i.e., lacking any spikes of energy [75]. Consequently, authors have tried to find approaches to achieve appropriate dot distributions in one go.

Secord et al. [13, 35] base their approach on a probability density function (PDF) derived from the input image that is used to re-distribute a set of initially uniformly distributed random points. This approach not only eliminates the iterative processing but also allows the creation of frame-coherent animations because the PDF varies continuously for animated image sequences. Secord et al. apply their approach not only to stippling but also to the distribution of other primitives including styles such as hatching.

Kopf et al. [4, 76] also present a fast algorithm for producing point distributions with blue noise properties by pre-computing suitable distributions and then describing a way to tile several pre-computed sets in a non-periodic fashion. They base their approach on McCool and Fiume's [10] progressive variant of Cook's [77] dart throwing algorithm, assign the (pre-computed) point sequence that results from it to a number of Wang tiles, and then adjust the distributions such that two adjacent Wang tiles can be placed in a seamless fashion. The resulting tilings can produce infinitely large point distributions without visible repetition artifacts which have blue noise properties. Moreover, Kopf et al. also ensure that the tiles can be used recursively, ultimately allowing users to zoom into the point sets as discussed in Section 3.1 and shown in Figure 5. This aspect
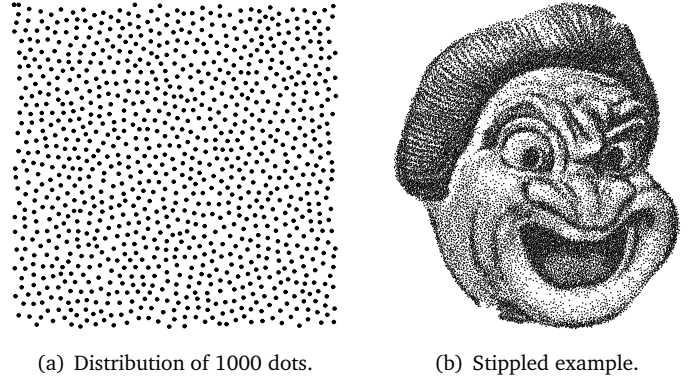

(a) Distribution of 1000 dots.   (b) Stippled example.

Figure 11: Results from using Wachtel et al.'s [79] Polyhexes-based point distribution. Images © 2016 and courtesy of Helene Perrier, David Coeurjolly, and Victor Ostromoukhov, used with permission.

can also be beneficial, for example, when Kopf et al.'s [4] approach is used on surfaces in 3D visualizations [78].

Several other tiling-based approaches (using pre-computed datasets) to achieve point distributions with blue noise characteristics exist which can be applied to stippling. For example, Hiller et al. [80, 81] used Lloyd's method to pre-process a set of tiles with several dots each, ensuring that the edge conditions with possible neighbors are fulfilled. Ostromoukhov et al. [82] used a recursive Penrose tiling with one dot per tile, improve the blue noise characteristics using Lloyd's method, and then render based on an importance map and a dot importance that is derived from the Fibonacci number system. Lagae and Dutré [83] used Wang corner tiling with several dots per tile whose distributions are generated by dart throwing [10, 77], also followed by Lloyd's method. While all these three methods only support the interactive zoom up to a maximum magnification, Ostromoukhov [84] used a polyomino-based tiling with one dot per tile with the ability—like Kopf et al. [4]—to use recursion for higher resolutions. More recently, Wachtel et al. [79] provided a fast tile-based algorithm to compute point distributions that have a specified Fourier spectrum (example in Figure 11). This property is achieved using a tile distribution with blue noise properties for which points are sampled based on precomputed offsets such that the offsets follow user-specified spectral properties.

Other techniques compute point distributions with blue noise characteristics on the fly, to be used for multiple applications in computer graphics including stippling. For example, Jones [85] introduced an $O(N \log N)$ algorithm based on a Voronoi diagram, but without relaxation. In each step, the algorithm searches for the region with the most free space left to place a new point. Dunbar and Humphreys [86, 87] described a similar incremental algorithm to compute Poisson-disk distributions that is based on a dedicated data structure as well as a modification for Poisson-disk-like distributions that only has linear runtime, an approach that was later generalized by Bridson [88]. White et al. [89] used a hierarchical procedure to produce Poisson-disk distributions with dart throwing, also with linear runtime, while Wei [90] used a grid-based parallel

10

implementation on the GPU. Fattal [91] used a statistical interacting particle model and derive appropriate point distributions for stippling or otherwise by sampling this model at multiple scales.

A GPU-based on-the-fly algorithm that also takes the intended tonal dot density for the stippling into account was described by Ascencio-Lopez et al. [37]. In contrast to the constant disk size used by the previous techniques, Ascencio-Lopez et al. changed the disk radius for the sampling locally to improve the quality of the distributions for the application to stippling. In a related approach, Arroyo et al. [32, 33] used the input image as well as artistic input to specify and control a probabilistic density function for future stipple placement, and sampled it randomly to find the position for the next stipple dot.

In addition to the tile-based methods and on-the-fly computation, also some iterative methods have been used. For example, Xu et al. [92] developed a method that is based on a capacity-constrained version of the Delaunay triangulation. In a way this forms the "dual" of the capacity-constrained Voronoi tessellation introduced by Balzer et al. [34], but has faster computation times. Zhou et al. [93] developed an iterative method that produces point distributions that complies with a user-defined spectra. They achieve this goal by converting the Fourier spectrum function into a differential distribution function, which is used to compute the final set. The method can be used to produce stippled images similar to other approaches using a blue noise spectrum, while other spectra lead to other interesting distributions and patterns. Ahmed et al. [94] later demonstrated the generation of point distributions with controlled spectra based on AA patterns (an ornamental point set), while Jiang et al. [95] used an approach based on the Smoothed Particle Hydrodynamics method (otherwise used for fluid dynamics) to compute point distribution with high-quality blue-noise characteristics. This latter method can adjust the generated distributions between WCVD-based ones [12] and those based on capacity-constrained Voronoi tessellations [34].

4.3. Generic distribution functions applicable to stippling

The problem of distributing points in the plane, however, is one that has applications much beyond stippling. For example, the usual goal of point distributions is to comply with some mathematical properties such as blue noise. Then, they can be used in several applications as texturing, point-based rendering, Monte Carlo methods, etc.; stippling being only one of the possibilities. So, many more techniques have been conceived whose discussion would be beyond the scope of our survey. We thus just mention a few more here which did not make a direct connection to stippling, but which are relevant nonetheless. This collection starts with early contributions such as the quasi-random computation of points in a k-dimensional unit hypercube by Halton [96], point distributions by Sobol [97] that facilitate the computation of integrals of a unit n-dimensional hypercube, and the use of Poisson and jittered sampling

to avoid aliasing by Dippé and Wold [98]. Later, Ebeida et al. [99] showed a method to compute uniform Poisson-disk distributions that are both maximal and bias-free and which can be used over non-convex domains. Schlömer et al. [100] presented a method that applies the strategy of iteratively moving points from a sample to the farthest point or all points in the sample to increase the overall distance between each pair of points, resulting in almost optimal blue-noise characteristics. Öztireli and Gross [101] proposed a method for obtaining point distributions that are based on the pair correlation function, allowing the authors to capture an example input distribution and to synthesize a new one based on it. Heck et al. [102] addressed the problem of aliasing in point distributions and define two new types of blue noise sampling patterns depending on the oscillation level in the high frequencies: step blue noise and single-peak blue noise. Their resulting patterns prevent structured aliasing. Reinert et al. [103] developed a point distribution scheme for 3D space that has the property that blue-noise characteristics are maintained even when they are projected to lower dimensional spaces. Yuksel [104], in contrast, presented an approach based on random distributions but which can be modified with an elimination process to achieve Poisson disk distributions. Ahmed et al. [105], finally, placed emphasis on distributions that simultaneously have blue noise and low-discrepancy properties.

In addition to those approaches that produce point distributions in the 2D plane, some researchers also explored point distributions on surfaces in 3D space. For example, Bowers et al. [106] described an approach to generate Poisson disk distributions for surfaces of 3D models that could be used for stippling. However, several dedicated techniques exist for 3D stippling as we describe next.

4.4. Stippling on 3D models

While stippling is traditionally a purely two-dimensional technique, it can be of advantage to place the dots directly on or in a 3D model. This approach facilitates, in particular, interactions with stippled models while maintaining full frame-coherence (i. e., without the shower door effect) as well as to use models whose shape is animated or changing otherwise. Generally, authors have presented 3D stippling approaches for most model types including polygonal models, point clouds, volumetric data, and implicit surfaces.

In the first group of **polygonal models**, most approaches simply place stipple particles onto the 3D surface, project these to the image space, select the stipples to be shown based on the resulting density and illumination, and then render stipple dots at the respective 2D locations.[14] A first approach by Lu et al. [8, 42, 108] follows exactly this process: They first randomly seed a dense set of points

---

[14]This approach also has been demonstrated, e. g., by Meier [44], to be able to prevent the shower door effect for other stroke-based [107] non-photorealistic rendering techniques.
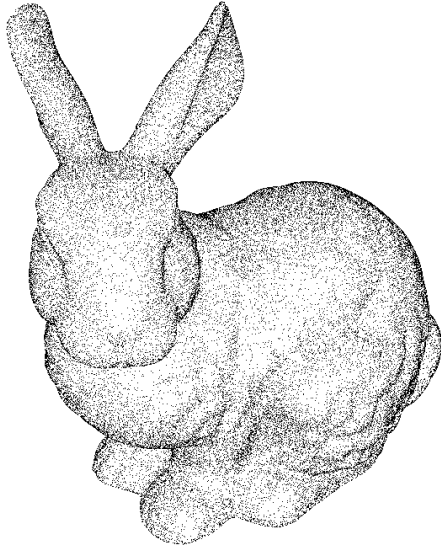
Figure 12: Example for Lu et al.'s [8, 42] stippling of polygonal surfaces. © 2002 and courtesy of Aidong Lu, used with permission.

Figure 13: Example for Schmidt et al.'s [55] stippling of implicit surfaces.

on the polygonal surface, then redistribute them using CVDs [11], and finally render a subset based on the illumination and the projected size of the polygon (e. g., Figure 12). Meruvia Pastor et al. [3, 57, 65, 109] use a similar approach but organize the points hierarchically to achieve frame coherence in an animation (e. g., Figure 6), either based on randomized placement [3, 57] or on a graph-based point relaxation process [65]. This last work by Meruvia Pastor and Strothotte also proposes two new applications of stippling using 3D models: illustration of transparent surfaces and the reproduction of virtual objects for archaeology. Baer et al. [110, 111, 112] later described a frame-coherent GPU implementation of a hierarchical texture-based stippling and demonstrate their technique, in particular for the illustrative visualization of medical surface models.

Three special forms of stippling for 3D surface models were introduced by Yuan et al. [49], Vanderhaeghe et al. [36], and Costa Sousa et al. [43]. Yuan et al. [49] compute a conformal parameterization of the 3D model and derive a corresponding geometry image, and then use this setup to make all computation in this hybrid geometry-image domain—using Ostromoukhov et al.'s [82] tiling approach as the basis. The benefit of this process is that the computations can be implemented using well-known and efficient 2D techniques, while still being able to make use of 3D information. Both animation and still image generation are possible, with the latter using additional randomness to improve stipple point placement. Vanderhaeghe et al. [36], instead, focus entirely on animated 3D stippling. Their approach is to place the stipple dots in 2D space first, then project them to the 3D scene and track their positions during the animation, project them back to 2D, and then to re-evaluate each dot for potentially deleting some and introducing other new ones. This method yields good 2D distributions, illustrates 3D motion well, and can even deal with deformable models and video material. Costa Sousa
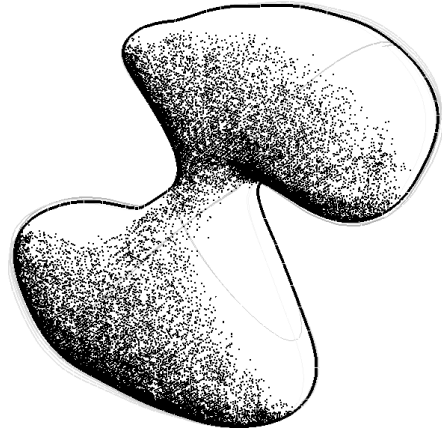
et al. [43], finally, introduced a stippling-inspired method they call "precise ink drawing" which attaches small ink marks to each edge of the polygonal model, resulting in a stippling-like appearance for densely tessellated models.

Two approaches exist for the generation of stippling images for **point clouds**. Zakaria and Seidel [53] focus on rendering the silhouettes/occluding contours for point data that originates from 3D scans, and describe a method to add stippling to those regions that are close to the silhouette/occluding contour. They thus construct a point group hierarchy from the scanned points, each group with one point dedicated to be a potential stipple point. At run-time, they then select the hierarchy level based on the desired stipple density, i. e., lower levels (fewer points per group) for denser regions. Xu and Chen [56] also work with scanned 3D data, but classify each input point to be either a feature or non-feature point. For the latter they distinguish between dithering points and others, and use the dithering points to create a style that resembles stippling.

Volumetric models are commonly used in medical visualization and physical simulations, and several authors have introduced methods to generate stipple illustrations from them. The approaches generally differ from most others as here the stipple points illustrate volume units, not object surfaces. Lu et al. [7, 8, 108] first distribute gradient-dependent number points per voxel in a pre-processing stage such that they approximate a Poisson disc distribution when seen along the gradient direction. At runtime, they select a subset of these points based on the resolution of the rendering, based on a randomly pre-assigned rendering priority of each point. Krüger and Westermann [54], instead, based their approach on a 3D noise texture, using shaders to select the (stipple) fragments to be ultimately shown depending on illumination and transfer functions.[15] Finally, some of the previously mentioned methods for 2D point distributions can also be applied to 3D volumetric models such as, e. g., Jiang et al.'s [95] approach.

---

[15]Krüger and Westermann's [54] 3D noise texture technique can also be applied to (polygonal) surface models.
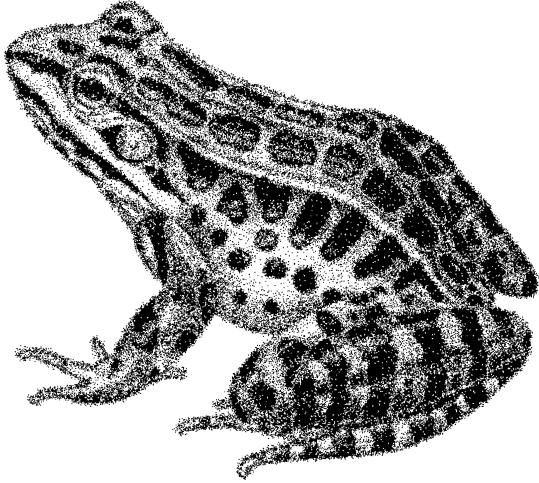
Figure 14: Example for Kim et al.'s [25] example-based stippling. Image from [25], © ACM, used with permission.

While **implicit models** are related to volumetric models in that they encode both surface and volumetric aspects, stippling techniques have so far focused on depicting the iso-surfaces of implicit shapes. Foster et al. [48, 113], for instance, seed points based on ray-surface intersection computations and then compute attraction/repulsion forces to move them further toward the surface. Based on illumination and orientation w.r.t. the view direction, particles are then selected to be rendered either as stipples or in a different style. Schmidt et al. [55] extract a coarse surface base mesh, assign a number of overlapping surfels [114] to each face (similar to Meier [44]), and then distribute stipple points randomly on each surfel. The rendering process uses this hierarchy, culling stages as early as possible to ensure real-time performance (see the example in Figure 13). Vital Brazil et al. [115], finally, based their stippling technique on Hermite RBF implicits [116] that, in turn, are based on a set of sampling points (with normals) that define the object's surface. Using this set or a newly generated set of seed points, the authors recursively multiply the number of seeds, back-projecting them onto the iso-surface each time. At runtime, these potential stipple points are filtered according to illumination and image-space point density.

4.5. Example-based stippling

All digital stippling techniques discussed so far synthesize stippling solely based on the generic properties of stippling that researchers observed in hand-made examples or from studying the literature (see Section 2). We now discuss approaches that, in contrast, aim to extract specific properties directly from specific examples and use those to generate new images in an **example-based** approach, either by using examples for distributing stipple points (i. e., **example-based stippling**) or by using examples to create stipple dots (i. e., **example-based stipples**).

Barla et al. [46, 50, 117] used such an example-based approach to capture the style of a drawn input pattern

(such as a section of hand-drawn stippling) and then to synthesize larger patterns in the same style. They capture the organizational characteristics of the input pattern by assessing element clusters and their distribution. Based on this analysis they synthesize new patterns using a neighborhood matching algorithm to generate both new perceptually similar stroke clusters as well as several groups of them. The synthesis relies on Lloyd's method to generate seed positions but then uses the learned characteristics for actual element and stroke placement.

While this approach produces convincing results, the problem with Barla et al.'s technique is that it only generates the patterns themselves—without considering the target tone. Kim et al. [25] thus considered a whole set of stippling patters in an example drawing, extracting different patches for different tonal shades. They then extract the centers of the stipples, compute a gray-level co-occurrence matrix (GLCM) of the resulting distribution texture, and use it to record several texture statistics. They can then synthesize new distribution textures for each of the different tonal levels using a rejection sampling that minimizes the errors between the statistical measurements between sample and synthesized patch. They either use the synthesis results directly or place stipple dot textures at the new locations based on the scans from the exemplar (Figure 14). This overall approach works well for light and dark tones but has problems with middle tones where stipples can merge in the hand-drawn examples (see Figure 15(c)).

Martín et al. [18, 19] addressed this latter problem by treating dots no longer as completely black marks—they capture and use the stipple dot examples as grayscale/color textures. In addition to allowing artists to affect a number of higher-level properties, Martín et al. based their whole process on an analysis of the real spatial sizes (of example images, stipples, and output media) and use the resulting resolutions to be able to synthesize results based on the intended output size and reproduction resolution (see Figure 16). Their dot distribution is based on traditional halftoning together with random perturbations, adjusted to the derived target image size. The use of grayscale stipples then allows them to achieve results with merging stipples (e. g., see Figure 16(b) and 16(c)), whether reproduced on a grayscale medium or discretized for binary output.

4.6. Structure-aware stippling

Generally, most techniques discussed so far focus on the stippling of image areas. Nonetheless, some of them either unintentionally (i. e., techniques based on Lloyd's method) or intentionally (e. g., Martín et al.'s [19] extension of their original [18] approach) could arrange some stipple dots such that they followed linear structures, a feature sometimes used by stipple artists to emphasize such dedicated lines in an image. A number of **structure-aware stippling** techniques, however, focus specifically on replicating this aspect of stipple renderings as discussed next.

To specifically emphasize image edges (i. e., lines of high local gradient or of any other importance measure),
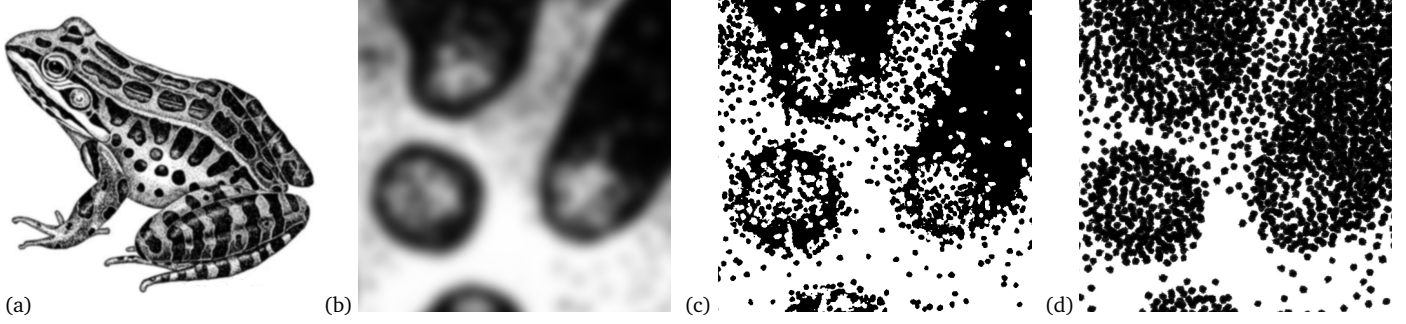
Figure 15: Example-based methods: (a) original, (b) detail section, (c) detail produced with Kim et al.'s [25] example-based method, and (d) detail produced with Martín et al.'s [18, 19] grayscale method. Images (a) and (b) are in the public domain (also see Appendix A.3).
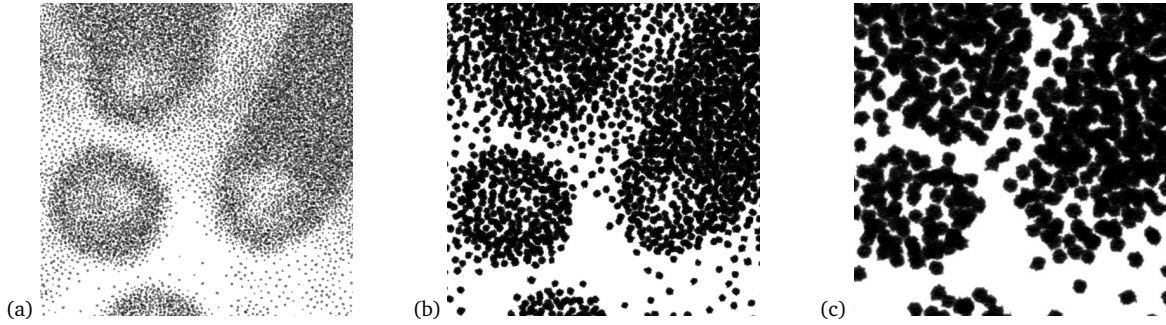

Figure 16: Effect of using different target pixel resolutions for the same pixel output size (4096^2 pixels) in Martín et al.'s [18, 19] grayscale method, using the same detail section as in Figure 15(b): (a) 300 ppi target resolution (i. e., to be shown at 34.7 cm × 34.7 cm), (b) 600 ppi target resolution (i. e., 17.3 cm × 17.3 cm), and (c) 1200 ppi target resolution (i. e., 8.7 cm × 8.7 cm). The example shows that, the lower the pixel density of the reproduction is, the higher is the number and the smaller is the size (in pixels) of the dots that are to be used for the same amount of space in the image.

Mould [20] first converts the input image into a weighted graph, each pixel being connected to its eight direct neighbors. Then node weights are determined based on image darkness and local gradient, and edge weights are computed as averages of the node weights. Then, Mould [20] starts the region growing process at a given site (e. g., an image corner), always extending the frontier of the current region(s) by following the shortest path. As soon as this shortest path exceeds a threshold, a new site is placed on the location of the frontier with the highest gradient magnitude, resulting in high gradients being emphasized. The region growing then continues around the newly placed site, thus also ensuring a minimum distance between any two stipple dots, resulting in blue noise qualities. Example results are shown in Figure 4(b) and Figure 17(a).

Using a method for contrast-improving halftoning [118] as the basis, Li and Mould [40, 119] later introduced a method that, while also emphasizing linear features, produces results that better preserve the tone and contrast of the original images. Their overall approach is to treat the pixels of an input image in an order of priority derived from their importance for contrast and structure—extreme (black or white) pixels are treated first. Then, if a stipple dot is placed, the introduced error is distributed among the neighboring pixels, leading to an update of their priorities. Li and Mould [40] also used stipples of varying sizes and can distribute the error in an irregular fashion to avoid the forming of distribution patterns in dense regions (e. g., see Figure 4(d) and Figure 17(b)).
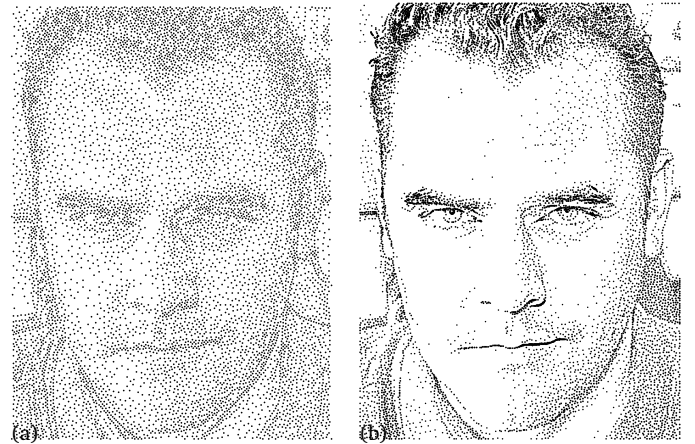

Figure 17: Comparing (a) Mould's [20] (8,467 stipples) with (b) Li and Mould's [40] (8,431 stipples) structure-aware stippling techniques. Both images are © 2016 and courtesy of Davild Mould, used with permission.

To illustrate the two approaches for structure-aware stippling, Figure 17 compares them for the same source image and using roughly the same number of stipples. The figure demonstrates the ability of both techniques to preserve important structural elements, yet that Li and Mould's [40] technique better preserves the contrast and thus also reduces noise in areas with fewer stipples. However, Li and Mould's technique optimizes the output with lower stipple counts using the structural information , so Figure 4 shows a comparison of Mould's [20] technique for low stipple counts with Li and Mould's [40] technique at high counts,

Figure 18: Example of Kim et al.'s [22] hedcut stippling. The image is © 2013 and courtesy of Ross Maciejewski, used with permission.

Figure 19: Examples of Son et al.'s [23] hedcut stippling. The images are © 2016 and courtesy of Minjung Son, Yunjin Lee, Henry Kang, and Seungyong Lee, used with permission.

and both to the WCVD technique.

4.7. Hedcut stippling

A form of stippling that relies exclusively on arranging the stipple dots along linear structures is hedcut stippling.[16] Here, linear paths are derived from the image features and then arranged in a parallel fashion, along which stipple dots are placed. Sometimes even two or more layers of crossing lines are combined, such that the stipple dots are lined up along multiple directions.

A first attempt[17] to simulate this style in digital form was created by Kim et al. [21]. They emulate the manual process by extracting feature lines in the image, use these to produce a distance field, and use these in turn to generate parallel guiding lines along which the stipple points are arranged using a constrained Lloyd's algorithm. For the final image, the authors controlled the tone by modulating the size of the stipples based on the tone image.[18] Kim et al. [22] later improved on Kim et al.'s [21] method by not only using image features to extract the guiding lines but by also considering isophote lines. These lines of isoluminance add shading features to the resulting image as shown in Figure 18.

Son et al. [23] further improved on these results by extracting not only a set of parallel lines from the feature map but first a feature vector field and, from that, a structure grid—much like Ostromoukhov's [24] parametric grids that he used for hatching. This structure grid follows directions perpendicular and parallel to features extracted from the image. Son et al. then arrange stipple dots at the locations where the grid lines intersect, and parameterize the dots based on the structure grid and the tone map. Two example results are shown in Figure 19.

---

[16]Excellent hand-made examples of this traditional technique can be found, for instance, on the sites by artist Randy Glass at `http://randyglassstudio.com/wsj.html` and artist Kevin Sprouls at `http://www.sprouls.com/wall_street_portraits.php`.

[17]One could argue that the very first attempt was published almost 10 years earlier by Ostromoukhov [24] who, while concentrating on hatching-like styles, also showed examples that resemble hedcut stippling.

[18]Wang [120] seems to discuss a "replication" of Kim et al.'s [21] approach with an additional threshold matrix approach for speedup, but we cannot verify this fact due to our lack of Chinese language skills.

4.8. Other creation methods and applications

A number of other techniques should be mentioned that we could not easily classify into any of the previously used techniques. In particular, there are techniques that apply stippling for a specific (illustrative) purpose. Among them is the proposal by Meruvia Pastor and Strothotte [65] to create illustrations of animated transparent surfaces, allowing them to show the inside of 3D objects as well as the reproduction and animation of archeological objects from 3D scanned data. Another example, Goldau et al.'s [51, 52] "fiber stippling," is shown in Figure 8. Here, stippling is used to visualize the results of probabilistic tractography—a way to study connectivity between different regions of the brain for which a range of more or less illustrative visualization techniques exists [121]. The "fiber stipple" primitives in this case range from circular dots to short dashes, depending on the orientation of the fibers with respect to the depicted slice of the brain data. The stipples are randomly distributed and can be displayed in a semi-transparent fashion—both depending on the local probability of a tract being present. This distribution inherently indicates the probabilistic character of the data. Hlawitschka et al. [38] later proposed an improvement for the fiber stipples' distribution by using a Poisson-disk distribution to avoid the overlapping of fiber stipples.

Another application of stippling to illustrative visualization was described by Berkiten et al. [122] who described a method to represent epigraphy—archaeological black and white drawings of human inscriptions. One of the styles they described represents the obtained shapes with a texture that consists of grayscale dots which they call "stippling." While this approach allows them to depict the shape of the highlighted features well, it is more related to noise textures (e. g., [54]) than to traditional hand-drawn stippling as we describe it in Section 2.

As a way to combine stippling with other NPR techniques, Schlechtweg et al. [58, 66, 123] provided a unified framework for stroke-based rendering that also includes hatching, sparse lines, paint strokes, and mosaics. Their approach is based on simple autonomous agents, the RenderBots, which use local information from the source image plus from a stack of G-buffers [124] to control their be-

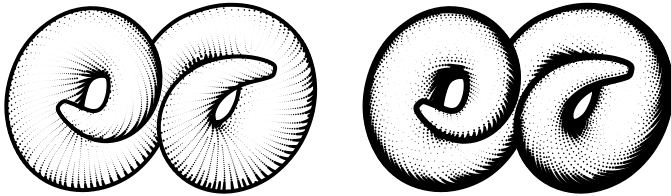Figure 20: Stippling generated with Schlechtweg et al.'s [58] RenderBots.

Figure 21: Examples of Zander et al.'s [125] line stippling.

havior. For stippling (e.g., Figure 20), each StippleBot represents a single stipple dot and uses the intensity information to compute the force that moves it, depending on a search radius and other StippleBots around it.

Zander et al. [125, 126], finally, used stippling not for filling image areas but, instead, to represent gray lines in a binary (stippled) form—in a way a 1D halftoning. Their line stippling thus only arranges the stipple dots in one dimension (e.g., Figure 21), adjusting the spacing to the gray level of the line to be represented. They apply the technique for computer-generated hatching illustrations.

4.9. Measuring and benchmarks

In addition to creating new techniques to produce computer-based stippling it is also important to reflect on the results of such new methods. For most techniques in NPR and illustrative rendering, however, this reflexion is not easy because there are many subjective and qualitative assessments to be made [127]. For the specific sub-domain of digital stippling we thus have to look at approaches that relate both to expressive rendering techniques in general and to stippling and characteristics in particular.

In the former group of image aesthetics in general, for example, Halper et al. [128] investigated the relationship between psychology and NPR, leading to suggestions on how different rendering styles can change a viewer's perception of and interaction with the respective images. Others have investigated the relationship between structure and aesthetic [119, 129] or composition principles to get certain aesthetic results [130]. However, researchers such as Hall and Lehmann [131] also expressed skepticism about the usefulness of trying to measure the "artistic" qualities of NPR results and question whether this is possible.

For studying digital stippling specifically, researchers used either perceptual studies (i.e., using human participants) or physical analyses of the results (i.e., no humans involved; e.g., spectral analysis). As an example of the first of these two types, Isenberg et al. [41] conducted an experiment that compared hand-drawn illustrations with computer-generated ones, including stippling as an illustration style. They found that people could easily distinguish both types specifically for stippling, in particular due to the WCVD-driven dot placement artifacts and the tendency to use a high number of stipple dots in the computer-generated images—because it was possible. Yet, this high number also let participants to state that they appreciated the computer-generated images for their precision, while they enjoyed the hand-drawn images for their "character."

In a similar vein, Maciejewski et al. [132] discussed the differences between an aesthetic of hand-drawn stippling and one that relates to NPR stippling. For the former they cite the need for the artist to pay attention to the subject matter, to the marks that are generated, to the density and randomness of dots, and to avoiding or emphasizing image features using the dot placement. For the latter, they name the dot count to be placed, the choice of shading technique, and the choice of specific shape for the marks.

Martín et al. [14] then investigated this last aspect—albeit for hand-made images—by studying the perception of dots based on samples on different paper generated by different pens. Their analysis showed that the pen's nib size and the paper type can have an influence on whether the dots are perceived by participants as gray or black, as irregular or rounded, and with a varying or constant size.

In contrast to such studies with human participants, researchers also analyzed objectively measurable qualities for stippled images and stipple/point distributions. One of these techniques is the use of spectral analysis to assess the quality of a point distribution, with "blue noise" qualities [133] commonly being cited as a goal to be achieved for point distributions. Such types of analysis have been carried out for many of the approaches discussed in Sections 4.1–4.3, for example by Deussen et al. [11], Hiller et al. [80], Jones [85], Kopf et al. [4], Balzer et al. [34], Dunbar and Humphreys [86, 87], Ostromoukhov [84], and Ascencio-Lopez et al. [37]. In addition, Wei and Wang [134] compared standard spectral analysis techniques such as Fourier spectra with newly developed methods, also applying them to stippled images.

As an alternative to spectral analysis, Maciejewski et al. [135] used a metric based on the gray level co-ocurrence matrix (GLCM). It allowed the authors to study three different aspects of the stipple dot distribution (contrast, energy, and correlation) and, based on this analysis, to distinguish different hand-drawn and digital stippling styles. This approach was later also used as a basis for an example-based stippling technique [25].

A final analysis technique was used by Li and Mould [40] to assess the quality of their structure-aware stippling technique. This approach relied on the structural similarity (SSIM) index [136] as well as the measurement of tone using peak signal-to-noise ratio (PSNR). Both approaches allow Li and Mould to demonstrate that their approach preserves a higher structural similarity to the original images than other stippling techniques.

5. Analyzing the results

As already noted, Table 1 summarizes the different stippling approaches that we surveyed in the previous section. As we can see in the table and as we showed in the discussion, work in this area has yielded a rich selection of techniques with different properties and application domains. In the table we focus on the techniques whose primary goal was digital stippling, we do not include most of the many techniques for point distributions which are summarized in the table as a group—a proper comparison of these techniques is beyond the scope of this survey.

With the help of the table and the discussion in the previous section, future approaches can now be classified and compared to existing work. Moreover, practitioners can use our comparison to select an appropriate technique for their purpose, be it the production of still images from 2D input, the use of stippling in a 3D context, the animation of stippling in a 3D or 2D domain, or the use of example-based techniques such as to aid illustrators in their work. These examples show that, while digital stippling is inspired by the traditional process, the computer-based techniques deviate from it in many ways. By no longer being bound by the constraints of the manual process they thus open up new ways of expression with digital media.

In fact, most methods we discussed primarily dealt with the problem of point distribution for filling areas. Today, many approaches exist that solve the problem of arranging dots with specific spectral properties as outlined in Sections 4.1 and 4.2. Fewer approaches deal with the properties of the dots or with specific dot arrangements to capture specific features. Only fairly recently have example-based techniques for the dot distribution and the dot properties been explored (Section 4.5), but the latter would open up possibilities to truly support traditional stipple artists in their illustration work. Such a use in a creative environment would also require a sufficient support of interactive control of the digital stippling process, an aspect that is only supported by few of the techniques.

Two tricky aspects of the different approaches are whether they allow/facilitate the overlapping of dots and whether visible artifacts are being produced in the dot distributions. In the early approaches, the fact that two dots were overlapping was named as not desirable—yet, in more recent approaches such overlapping was intentionally used to be able to reproduce certain tonal ranges more faithfully compared to the hand-made examples. The second aspect, the visibly regular arrangement of dots in certain approaches (such as those based on Lloyd's basic method), can be solved with either example-based arrangement or with many of the recent approaches that focus primarily on sampling positions in the plane (see Section 4.2). In fact, it would be quite interesting to repeat some of the studies mentioned in Section 4.9 to see if, with today's methods, we would be able to pass the NPR Turing test [137]—even if some researchers [131] may object to such experiments. Also some approaches that arrange dots along features intentionally have also been presented (Sections 4.6 and 4.7) that are also inspired by traditional stippling techniques.

6. Relationship of (digital) stippling with halftoning

One of our initial motivations for this survey was to better understand the relationship between stippling and halftoning. As observed by several authors previously (e.g., [20, 11, 40]), both stippling and halftoning share the goal of representing a continuous-tone image for a binary medium. We started out in Section 2, however, by observing that hand-made stippling is an art form, in contrast to other techniques for bi-tonal reproduction such as halftoning which are purely algorithmic ways to convert images. For the latter, a continuum of techniques exists between amplitude modulation halftoning (also called clustered-dot dithering; see Figure 22(a)) [138] and error-diffusion halftoning (also called frequency modulation halftoning; see Figure 22(c)) [139]. The former leads to relatively regular representations, while the latter produces more stochastic results. In-between these two extremes, green-noise halftoning [139, 140, 141] (e.g., Figure 22(b)) combines properties of both approaches.

Nevertheless, all forms of halftoning use the pixel or printer dot as their primary primitive, with the goal of placing the dots such that their pattern leads to a perception of the correct tone, ideally without the individual points being visible at all.[19] The goal in traditional stippling is quite the opposite as we pointed out in Section 2 and as commented elsewhere as well (e.g., [14, 19, 11]): each dot is larger and has a purpose. In addition to the technical advantages of stippling over halftoning in the early days of printing mentioned by Secord [12], this aspect gives stipple artists a high degree of control over the final image (e.g., [1, 12, 40]): In addition to simply reproducing tone, dots are often placed guided by features and details in the image, sometimes combined and interacting with other primitives such as lines [11]. In digital stippling to date, however, such control is lacking—only partial (algorithmic) solutions exist (e.g., Sections 4.6 and 4.7). Digital stippling today, therefore, largely lacks the ***aesthetic*** control found in the traditional art, just like halftoning,[20] while it is able to reproduce the aspect of large dots generally placed in the fashion of the artistic example. Algorithms that focus on only deriving suitable dot distributions can also apply these distributions to pixel or printer dot primitives, connecting aspects of digital stippling to those of halftoning.

7. StippleShop

The primary goal of this survey is certainly the comparison of the existing approaches and the discussion of aspects that

---

[19]Exceptions for this goal exist as well: for example, certain screening masks can lead to dot-like patterns that are larger than pixels or printer dots (e.g., [142]). Also, the electrostatic halftoning approach by Schmaltz et al. [143] includes a control of the granularity of the output, leading to results that resemble stippling.

[20]Some other ways to control the shape of the halftoning primitive for artistic purposes—yet still in an algorithmic fashion—exist as well such as Ostromoukhov's [144] artistic halftoning.
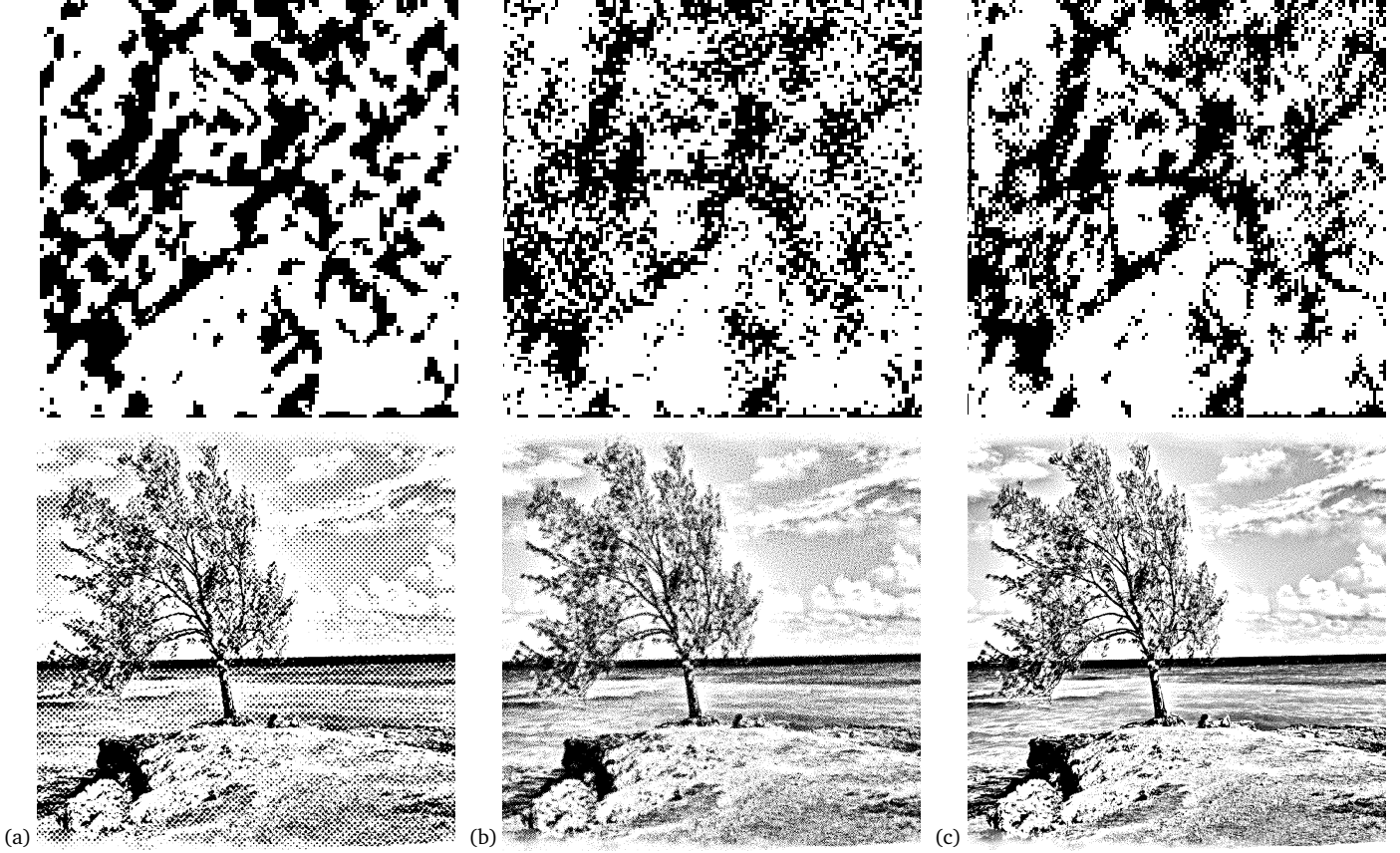
Figure 22: Halftoning methods (bottom) and details of the same zone (top): (a) amplitude modulation, (b) green-noise, and (c) error diffussion.

may affect one technique to be selected over another one in a specific application. Beyond this contribution, however, our goal was also to make available some of the existing techniques for researchers to evaluate and for practitioners to use. For this purpose we have not only collected a set of existing tools and research prototypes (Appendix A.1 and Appendix A.2) as well as benchmark images (Appendix A.3) and study/example datasets (Appendix A.4), but we also provide a new tool that incorporates several of the discussed methods: StippleShop (also see Appendix B).

StippleShop is a filter- and effect-based tool that implements the 2D stippling/dot placement approaches by Secord [12] (based on WCVDs), by Kopf et al. [4] (Wang tiling), by Balzer et al. [34] (dot distributions based on capacity-constrained Voronoi tessellations), by Kim et al. [25] (stippling by example), by Martín et al. [18, 19] (resolution-dependent grayscale stippling), and by Li and Mould [40] (structure-aware stippling). In addition, related methods and additional tools are included such as some halftoning techniques, different dot export schemes, a number of other filters and combination schemes, and two tools to assess the results. Details are provided in Appendix B and executables for Windows x64 and Linux are included as part of the submission.

8. Conclusion

With this survey we have provided an in-depth overview of digital stippling approaches, connecting it to its artis-

tic example and showing how the digital approaches can go beyond the limitations of the hand-made technique. In particular, we believe that our classification based on qualitative criteria allows future work to discuss different stippling approaches in a more focused way. Moreover, we hope that our discussion facilitates, in particular,

- a better understanding of the field as a whole, its goals, and its relationship to its artistic counterpart,

- the selection of an appropriate approach for a given application scenario by practitioners, and

- further research on digital stippling based on the properties and limitations of the existing techniques.

There are several lines of research that are left open, while others have only been partially addressed by the research community so far. We conclude our survey by mentioning some of them below.

One important goal of further work is the continued evaluation of hand-made and digital stippling to better understand its qualitative and quantitative properties and to find the best approach for a given task. In the past, researchers have studied low-level stippling characteristics such as the aesthetics of the dots and their placement, for example to drive example-based synthesis (Section 4.5). It is also necessary, however, to increase our knowledge related to its high-level aspects. For example, most ap-

proaches traditionally used a more or less direct mapping from tone in a source image (or shading on a 3D model) to stipple density. Yet, researchers have already discussed that this approach does not match the approach of stipple artists [18, 19], so an example-based high-level tone mapping would be interesting to investigate. Similarly, future research could try to understand how different stippling approaches and parameterizations are perceived by people, to guide practitioners in their design choices. In addition, low-level studies should similarly be conducted in the future to allow us to better understand the relationship of point distributions in hand-made stipple dot placement with those in digital stippling approaches and generic point distributions with mathematical qualities (Sections 4.2 and 4.3). Finally, the last work that compared digital stippling with hand-made approaches was published in 2006 [41]—an updated study that investigates the techniques published in the meantime would be quite informative.

Second, as we pointed out in Sections 3.4 and 6, a stippling approach that would not only support a purely algorithmic image filtering but which provides interactive control across a wide range of input granularity is still an open research problem, one whose solution would finally allow us to provide stippling tools to stipple artists and illustrators. Such work would also have an important impact beyond stippling on non-photorealistic rendering and illustrative visualization in general.

Third, so far digital stippling has been realized based primarily on a direct algorithmic encoding of the dot placement process (Sections 4.1–4.4 and 4.6–4.7) or on capturing image statistics from hand-made exemplars (Section 4.5). Recent work in deep learning and neural style transfer (NST; e. g., by Gatys et al. [145, 146] and others) has shown promising results in capturing and synthesizing other stylistic forms of depiction. The algorithms that have already been described can already be tuned in a way that can lead to new possibilities such as applying different learned styles in separated areas of the pictures [147]. Researchers that investigate neural style transfer are currently only beginning to explore different stylization techniques such as automatic pencil art [148], portrait stylization [149], and video stylization [150], so promising results could also be expected for digital stippling. Deep learning and neural style transfer, however, is not exempt of issues. The amount of data required to train complex neural networks is huge. For stippling, however, it may be difficult to get access to a large quantity of high-quality examples created by real artists, making it difficult to establish the basis for the learning process. Another problem is that deep learning to date cannot maintain details in the resulting pictures. The quality of the primitives (the stipple dots), however, is key for any stippling process—it would be disastrous to lose the dots and their character and to end up with simple noise. Nonetheless, we believe that deep learning can lead to exciting opportunities in the near future provided that the mentioned issues are resolved— maybe by exploring combinations between NST techniques and established stippling approaches.

Acknowledgments

References

[1] Deussen, O., Isenberg, T.. Halftoning and stippling. In: Rosin, P., Collomosse, J., editors. Image and Video based Artistic Stylisation; vol. 42 of *Computational Imaging and Vision*; chap. 3. London/ Heidelberg: Springer; 2013, p. 45–61. doi: 10.1007/978-1-4471 -4519-6_3

[2] Kyprianidis, J.E., Collomosse, J., Wang, T., Isenberg, T.. State of the "art": A taxonomy of artistic stylization techniques for images and video. IEEE Transactions on Visualization and Computer Graphics 2013;19(5):866–885. doi: 10.1109/TVCG.2012.160

[3] Meruvia Pastor, O.E., Freudenberg, B., Strothotte, T.. Real-time animated stippling. IEEE Computer Graphics and Applications 2003;23(4):62–68. doi: 10.1109/MCG.2003.1210866

[4] Kopf, J., Cohen-Or, D., Deussen, O., Lischinski, D.. Recursive Wang tiles for real-time blue noise. ACM Transactions on Graphics 2006;25(3):509–518. doi: 10.1145/1141911.1141916

[5] Hiller, S., Hellwig, H., Deussen, O.. Beyond stippling – Methods for distributing objects on the plane. Computer Graphics Forum 2003;22(3):515–522. doi: 10.1111/1467-8659.00699

[6] Dalal, K., Klein, A.W., Liu, Y., Smith, K.. A spectral approach to NPR packing. In: Proc. NPAR. New York: ACM; 2006, p. 71–78. doi: 10.1145/1124728.1124741

[7] Lu, A., Morris, C.J., Ebert, D.S., Rheingans, P., Hansen, C.. Non-photorealistic volume rendering using stippling techniques. In: Proc. VIS. Los Alamitos: IEEE Computer Society; 2002, p. 211–218. doi: 10.1109/VISUAL.2002.1183777

[8] Lu, A., Morris, C.J., Taylor, J., Ebert, D.S., Hansen, C., Rheingans, P., et al. Illustrative interactive stipple rendering. IEEE Transactions on Visualization and Computer Graphics 2003;9(2):127– 138. doi: 10.1109/TVCG.2003.1196001

[9] Lloyd, S.P.. Least squares quantization in PCM. IEEE Transactions on Information Theory 1982;28(2):129–137. doi: 10.1109/TIT. 1982.1056489

[10] McCool, M., Fiume, E.. Hierarchical Poisson disk sampling distributions. In: Proc. Graphics Interface. San Francisco: Morgan Kaufmann Publishers Inc.; 1992, p. 94–105. doi: 10.20380/GI1992.12

[11] Deussen, O., Hiller, S., van Overveld, C., Strothotte, T.. Floating points: A method for computing stipple drawings. Computer Graphics Forum 2000;19(3):41–50. doi: 10.1111/1467-8659. 00396

[12] Secord, A.. Weighted Voronoi stippling. In: Proc. NPAR. New York: ACM; 2002, p. 37–43. doi: 10.1145/508530.508537

[13] Secord, A.. Random marks on paper. Non-photorealistic rendering with small primitives. Master's thesis; Department of Computer Science, The University of British Columbia, Canada; 2002.

[14] Martín, D., del Sol, V., Romo, C., Isenberg, T.. Drawing characteristics for reproducing traditional hand-made stippling. In: Proc. NPAR. Goslar, Germany: Eurographics Assoc.; 2015, p. 103–115. doi: 10.2312/exp.20151183

[15] Hodges, E.R.S., editor. The Guild Handbook of Scientific Illustration. Hoboken, NJ, USA: John Wiley & Sons; 2nd ed.; 2003.

19

[16] Blatner, D., Roth, S.. Real World Scanning and Halftones. Peachpit Press; 1993.

[17] Green, P.. Understanding Digital Color. GATF Press; 2nd ed.; 1999.

[18] Martín, D., Arroyo, G., Luzón, M.V., Isenberg, T.. Example-based stippling using a scale-dependent grayscale process. In: Proc. NPAR. New York: ACM; 2010, p. 51–61. doi: 10.1145/1809939.1809946

[19] Martín, D., Arroyo, G., Luzón, M.V., Isenberg, T.. Scale-dependent and example-based stippling. Computers & Graphics 2011;35(1):160–174. doi: 10.1145/1809939.1809946

[20] Mould, D.. Stipple placement using distance in a weighted graph. In: Proc. CAe. Goslar, Germany: Eurographics Assoc.; 2007, p. 45–52. doi: 10.2312/COMPAESTH/COMPAESTH07/045-052

[21] Kim, D., Son, M., Lee, Y., Kang, H., Lee, S.. Feature-guided image stippling. Computer Graphics Forum 2008;27(4):1209–1216. doi: 10.1111/j.1467-8659.2008.01259.x

[22] Kim, S., Woo, I., Maciejewski, R., Ebert, D.. Automated hedcut illustration using isophotes. In: Proc. Smart Graphics. Berlin/Heidelberg: Springer; 2010, p. 172–183. doi: 10.1007/978-3-642-13544-6_17

[23] Son, M., Lee, Y., Kang, H., Lee, S.. Structure grid for directional stippling. Graphical Models 2011;73(3):74–87. doi: 10.1016/j.gmod.2010.12.001

[24] Ostromoukhov, V.. Digital facial engraving. In: Proc. SIGGRAPH. New York: ACM; 1999, p. 417–424. doi: 10.1145/311535.311604

[25] Kim, S., Maciejewski, R., Isenberg, T., Andrews, W.M., Chen, W., Sousa, M.C., et al. Stippling by example. In: Proc. NPAR. New York: ACM; 2009, p. 41–50. doi: 10.1145/1572614.1572622

[26] Rusinkiewicz, S., Cole, F., DeCarlo, D., Finkelstein, A.. Line drawings from 3D models. In: ACM SIGGRAPH Classes. New York: ACM; 2008, p. 39:1–39:356. doi: 10.1145/1401132.1401188

[27] Li, L., Zhou, Y., Liu, C., Xu, Y., Fu, J.. State-of-the-art line drawing techniques. In: Proc. CICA 2011. Netherlands: Springer; 2012, p. 1249–1257. doi: 10.1007/978-94-007-1839-5_135

[28] Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte, T.. A developer's guide to silhouette algorithms for polygonal models. IEEE Computer Graphics and Applications 2003;23(4):28–37. doi: 10.1109/MCG.2003.1210862

[29] Lawonn, K., Preim, B.. Feature lines for illustrating medical surface models: Mathematical background and survey. In: Linsen, L., Hamann, B., Hege, H.C., editors. Visualization in Medicine in Life Sciences III. Berlin/Heidelberg: Springer; 2016, p. 93–132. doi: 10.1007/978-3-319-24523-2_5

[30] Isenberg, T.. Interactive NPAR: What type of tools should we create? In: Proc. NPAR (Expressive). Goslar, Germany: Eurographics Assoc.; 2016, p. 89–96. doi: 10.2312/exp.20161067

[31] Deussen, O., Hiller, S., van Overveld, C.W.A.M., Strothotte, T.. Computer-generated stipple drawings. In: Proc. VMV. Sankt Augustin, Germany: infix; 1999, p. 329–338.

[32] Arroyo, G., Martín, D., Luzón, M.V.. Stochastic generation of dots for computer aided stippling. Computer-Aided Design and Applications 2010;7(4):447–463. doi: 10.3722/cadaps.2010.447-463

[33] Arroyo, G., Martín, D., Luzón, M.V.. A stochastic approach to simulate artists behaviour for atomatic felt-tipped stippling. In: Proc. Congress on Evolutionary Computation. Los Alamitos: IEEE; 2010, p. 1–8. doi: 10.111109/CEC-2010.5586091

[34] Balzer, M., Schlömer, T., Deussen, O.. Capacity-constrained point distributions: A variant of Lloyd's method. ACM Transactions on Graphics 2009;28(3):86:1–86:8. doi: 10.1145/1531326.1531392

[35] Secord, A., Heidrich, W., Streit, L.. Fast primitive distribution for illustration. In: Proc. EGWR. Goslar, Germany: Eurographics Assoc.; 2002, p. 215–226. doi: 10.2312/EGWR/EGWR02/215-226

[36] Vanderhaeghe, D., Barla, P., Thollot, J., Sillion, F.X.. Dynamic point distribution for stroke-based rendering. In: Rendering Techniques. Goslar, Germany: Eurographics Assoc.; 2007, p. 139–146. doi: 10.2312/EGWR/EGSR07/139-146

[37] Ascencio-Lopez, I., Meruvia-Pastor, O., Hidalgo-Silva, H.. Adaptive incremental stippling using the Poisson-disk distribution. Journal of Graphics, GPU, and Game Tools 2010;15(1):29–47. doi: 10.1080/2151237X.2010.10390650

[38] Hlawitschka, M., Goldau, M., Wiebel, A., Heine, C., Scheuermann, G.. Hierarchical Poisson-disk sampling for fiber stipples. In: Proceedings of the Workshop on Visualization in Medicine and Life Sciences. Goslar, Germany: Eurographics Assoc.; 2013, p. 19–23. doi: 10.2312/PE.VMLS.VMLS2013.019-023

[39] Vrkonyi, P., Domokos, G.. Imperfect symmetry: A new approach to structural optima via group representation theory. International Journal of Solids and Structures 2007;44(14–15):4723–4741. doi: 10.1016/j.ijsolstr.2006.11.047

[40] Li, H., Mould, D.. Structure-preserving stippling by priority-based error diffusion. In: Proc. Graphics Interface. Waterloo, ON, Canada: CHCCS; 2011, p. 127–134. doi: 10.20380/GI2011.17

[41] Isenberg, T., Neumann, P., Carpendale, S., Costa Sousa, M., Jorge, J.A.. Non-photorealistic rendering in context: An observational study. In: Proc. NPAR. New York: ACM; 2006, p. 115–126. doi: 10.1145/1124728.1124747

[42] Lu, A., Taylor, J., Hartner, M., Ebert, D.S., Hansen, C.D.. Hardware-accelerated interactive illustrative stipple drawing of polygonal objects. In: Proc. VMV. Aka GmbH; 2002, p. 61–68.

[43] Costa Sousa, M., Foster, K., Wyvill, B., Samavati, F.. Precise ink drawing of 3D models. Computer Graphics Forum 2003;22(3):369–379. doi: 10.1111/1467-8659.00684

[44] Meier, B.J.. Painterly rendering for animation. In: Proc. SIGGRAPH. New York: ACM; 1996, p. 477–484. doi: 10.1145/237170.237288

[45] Bénard, P., Bousseau, A., Thollot, J.. State-of-the-art report on temporal coherence for stylized animations. Computer Graphics Forum 2011;30(8):2367–2386. doi: 10.1111/j.1467-8659.2011.02075.x

[46] Barla, P., Breslav, S., Thollot, J., Sillion, F.X., Markosian, L.. Stroke pattern analysis and synthesis. Computer Graphics Forum 2006;25(3):663–671. doi: 10.1111/j.1467-8659.2006.00986.x

[47] Hurtut, T., Landes, P.E., Thollot, J., Gousseau, Y., Drouilhet, R., Coeurjolly, J.F.. Appearance-guided synthesis of element arrangements by example. In: Proc. NPAR. New York: ACM; 2009, p. 51–60. doi: 10.1145/1572614.1572623

[48] Foster, K., Jepp, P., Wyvill, B., Costa Sousa, M., Galbraith, C., Jorge, J.A.. Pen-and-ink for BlobTree implicit models. Computer Graphics Forum 2005;24(3):267–276. doi: 10.1111/j.1467-8659.2005.00851.x

[49] Yuan, X., Nguyen, M.X., Zhang, N., Chen, B.. Stippling and silhouettes rendering in geometry-image space. In: Proc. EGSR. Goslar, Germany: Eurographics Assoc.; 2005, p. 193–200. doi: 10.2312/EGWR/EGSR05/193-200

[50] Barla, P., Breslav, S., Markosian, L., Thollot, J.. Interactive hatching and stippling by example. Tech. Rep.; INRIA; 2006.

[51] Goldau, M., Wiebel, A., Gorbach, N.S., Melzer, C., Hlawitschka, M., Scheuermann, G., et al. Fiber stippling: An illustrative rendering for probabilistic diffusion tractography. In: Proc. BioVis. Los Alamitos: IEEE Computer Society; 2011, p. 23–30. doi: 10.1109/BioVis.2011.6094044

[52] Goldau, M., Hlawitschka, M.. Multi-modal visualization of probabilistic tractography. In: Linsen, L., Hamann, B., Hege, H.C., editors. Visualization in Medicine and Life Sciences III. Berlin: Springer; 2016, p. 195–218. doi: 10.1007/978-3-319-24523-2_9

[53] Zakaria, N., Seidel, H.P.. Interactive stylized silhouette for point-sampled geometry. In: Proc. GRAPHITE. New York: ACM; 2004, p. 242–249. doi: 10.1145/988834.988876

[54] Krüger, J., Westermann, R.. Efficient stipple rendering. In: Proc. IADIS Computer Graphics and Visualization. Lisbon: ADIS; 2007, p. 19–26.

[55] Schmidt, R., Isenberg, T., Jepp, P., Singh, K., Wyvill, B.. Sketching, scaffolding, and inking: A visual history for interactive 3D modeling. In: Proc. NPAR. New York: ACM; 2007, p. 23–32. doi: 10.1145/1274871.1274875

[56] Xu, H., Chen, B.. Stylized rendering of 3D scanned real world environments. In: Proc. NPAR. New York: ACM; 2004, p. 25–34. doi: 10.1145/987657.987662

[57] Meruvia Pastor, O.E., Strothotte, T.. Frame-coherent stippling. In: Eurographics Short Presentations. Goslar, Germany: Eurographics

Assoc.; 2002, p. 145–152. doi: 10.2312/egs.20021002

[58] Schlechtweg, S., Germer, T., Strothotte, T.. Renderbots—Multi agent systems for direct image generation. Computer Graphics Forum 2005;24(2):137–148. doi: 10.1111/j.1467-8659.2005.00838.x

[59] Yao, F., Shao, G.. Painting brush control techniques in Chinese painting robot. In: Proc. ROMAN. Los Alamitos: IEEE Computer Society; 2005, p. 462–467. doi: 10.1109/ROMAN.2005.1513822

[60] Lindemeier, T., Pirk, S., Deussen, O.. Image stylization with a painting machine using semantic hints. Computers & Graphics 2013;37(5):293–301. doi: 10.1016/j.cag.2013.01.005

[61] Tresset, P., Leymarie, F.F.. Portrait drawing by Paul the robot. Computers & Graphics 2013;37(5):348–363. doi: 10.1016/j.cag.2013.01.012

[62] Lindemeier, T., Metzner, J., Pollak, L., Deussen, O.. Hardware-based non-photorealistic rendering using a painting robot. Computer Graphics Forum 2015;34(2). doi: 10.1111/cgf.12562

[63] Galea, B., Kia, E., Aird, N., Kry, P.G.. Stippling with aerial robots. In: Proc. CAe. Goslar, Germany: The Eurographics Association; 2016, p. 125–134. doi: 10.2312/exp.20161071

[64] Isenberg, T., Carpendale, M.S.T., Sousa, M.C.. Breaking the pixel barrier. In: Proc. CAe. Goslar, Germany: Eurographics Assoc.; 2005, p. 41–48. doi: 10.2312/COMPAESTH/COMPAESTH05/041-048

[65] Meruvia Pastor, O.E., Strothotte, T.. Graph-based point relaxation for 3D stippling. In: Proc. Mexican International Conference on Computer Science. Los Alamitos: IEEE Computer Society; 2004, p. 145–152. doi: 10.1109/ENC.2004.1342599

[66] Germer, T., Schlechtweg, S.. RenderBots—Multi-agent systems for NPR. NORSIGD Info, medlemsblad for NORSIGD 2005;2005(1):4–5.

[67] Hausner, A.. Pointillist halftoning. In: Proc. CGIM. Calgary, AB, Canada: ACTA Press; 2005, p. 134–139.

[68] Smith, K., Liu, Y., Klein, A.. Animosaics. In: Proc. SCA. New York: ACM; 2005, p. 201–208. doi: 10.1145/1073368.1073397

[69] Hiller, S.. Relaxierte Punktverteilungen in der Computergraphik. Diplom thesis; University of Magdeburg; Germany; 1999. In German.

[70] Hausner, A.. Simulating decorative mosaics. In: Proc. SIGGRAPH. New York: ACM. ISBN 1-58113-374-X; 2001, p. 573–580. doi: 10.1145/383259.383327

[71] Hiller, S.. Interaktive und automatische Verfahren zur Objektanordnung in der Computergrafik. Ph.D. thesis; University of Konstanz; Germany; 2006. In German.

[72] Chen, Z., Yuan, Z., Choi, Y.K., Liu, L., Wang, W.. Variational blue noise sampling. IEEE Transactions on Visualization and Computer Graphics 2012;18:1784–1796. doi: 10.1109/TVCG.2012.94

[73] de Goes, F., Breeden, K., Ostromoukhov, V., Desbrun, M.. Blue noise through optimal transport. ACM Transactions on Graphics 2012;31(6):171:1–171:11. doi: 10.1145/2366145.2366190

[74] Deussen, O.. Aesthetic placement of points using generalized Lloyd relaxation. In: Proc. CAe. Goslar, Germany: Eurographics Assoc.; 2009, p. 123–128. doi: 10.2312/COMPAESTH/COMPAESTH09/123-128

[75] Mitchell, D.P.. Generating antialiased images at low sampling densities. ACM SIGGRAPH Computer Graphics 1987;21(4):65–72. doi: 10.1145/37402.37410

[76] Kopf, J.P.. Efficient Methods for Tile-Based Synthesis and Computational Photography. Ph.D. thesis; University of Konstanz; Germany; 2008.

[77] Cook, R.L.. Stochastic sampling in computer graphics. ACM Transactions on Graphics 1986;5(1):51–72. doi: 10.1145/7529.8927

[78] Svetachov, P., Everts, M.H., Isenberg, T.. DTI in context: Illustrating brain fiber tracts in situ. Computer Graphics Forum 2010;29(3):1024–1032. doi: 10.1111/j.1467-8659.2009.01692.x

[79] Wachtel, F., Pilleboue, A., Coeurjolly, D., Breeden, K., Singh, G., Cathelin, G., et al. Fast tile-based adaptive sampling with user-specified fourier spectra. ACM Transactions on Graphics 2014;33(4):56:1–56:11. doi: 10.1145/2601097.2601107

[80] Hiller, S., Deussen, O., Keller, A.. Tiled blue noise samples. In:

Proc. VMV. Berlin: Akademische Verlagsgesellschaft Aka GmbH; 2001, p. 265–272.

[81] Cohen, M.F., Shade, J., Hiller, S., Deussen, O.. Wang tiles for image and texture generation. ACM Transactions on Graphics 2003;22(3):287–294. doi: 10.1145/882262.882265

[82] Ostromoukhov, V., Donohue, C., Jodoin, P.M.. Fast hierarchical importance sampling with blue noise properties. ACM Transactions on Graphics 2004;23(3):488–495. doi: 10.1145/1015706.1015750

[83] Lagae, A., Dutré, P.. An alternative for Wang tiles: Colored edges versus colored corners. ACM Transactions on Graphics 2006;25(4):1442–1459. doi: 10.1145/1183287.1183296

[84] Ostromoukhov, V.. Sampling with polyominoes. ACM Transactions on Graphics 2007;26(3):78:1–78:6. doi: 10.1145/1276377.1276475

[85] Jones, T.R.. Efficient generation of Poisson-disk sampling patterns. Journal of Graphics, GPU, and Game Tools 2006;11(2):27–36. doi: 10.1080/2151237X.2006.10129217

[86] Dunbar, D., Humphreys, G.. A spatial data structure for fast Poisson-disk sample generation. ACM Transactions on Graphics 2006;25(3):503–508. doi: 10.1145/1141911.1141915

[87] Dunbar, D., Humphreys, G.. Using scalloped sectors to generate Poisson-disk sampling patterns. Tech. Rep. CS-2006-08; University of Virginia; USA; 2006.

[88] Bridson, R.. Fast Poisson disk sampling in arbitrary dimensions. In: ACM SIGGRAPH Sketches. New York: ACM; 2007, p. 22:1. doi: 10.1145/1278780.1278807

[89] White, K.B., Cline, D., Egbert, P.K.. Poisson disk point sets by hierarchical dart throwing. In: Proc. RT. Los Alamitos: IEEE Computer Society; 2007, p. 129–132. doi: 10.1109/RT.2007.4342600

[90] Wei, L.Y.. Parallel Poisson disk sampling. ACM Transactions on Graphics 2008;27(3):20:1–20:9. doi: 10.1145/1360612.1360619

[91] Fattal, R.. Blue-noise point sampling using kernel density model. ACM Transactions on Graphics 2011;30(4):48:1–48:12. doi: 10.1145/2010324.1964943

[92] Xu, Y., Liu, L., Gotsman, C., Gortler, S.J.. Capacity-constrained Delaunay triangulation for point distributions. Computers & Graphics 2011;35(3):510–516. doi: 10.1016/j.cag.2011.03.031

[93] Zhou, Y., Huang, H., Wei, L.Y., Wang, R.. Point sampling with general noise spectrum. ACM Transactions on Graphics 2012;31(4):76:1–76:11. doi: 10.1145/2185520.2185572

[94] Ahmed, A.G.M., Huang, H., Deussen, O.. AA patterns for point sets with controlled spectral properties. ACM Transactions on Graphics 2015;34(6):212:1–212:8. doi: 10.1145/2816795.2818139

[95] Jiang, M., Zhou, Y., Wang, R., Southern, R., Zhang, J.J.. Blue noise sampling using an SPH-based method. ACM Transactions on Graphics 2015;34(6):211:1–211:11. doi: 10.1145/2816795.2818102

[96] Halton, J.H.. Algorithm 247: Radical-inverse quasi-random point sequence. Commun ACM 1964;7(12):701–702. doi: 10.1145/355588.365104

[97] Sobol, I.M.. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics 1966;7(4):86–112. doi: 10.1016/0041-5553(67)90144-9

[98] Dippé, M.A.Z., Wold, E.H.. Antialiasing through stochastic sampling. ACM SIGGRAPH Computer Graphics 1985;19(3):69–78. doi: 10.1145/325165.325182

[99] Ebeida, M.S., Davidson, A.A., Patney, A., Knupp, P.M., Mitchell, S.A., Owens, J.D.. Efficient maximal Poisson-disk sampling. ACM Transactions on Graphics 2011;30(4):49:1–49:12. doi: 10.1145/2010324.1964944

[100] Schlömer, T., Heck, D., Deussen, O.. Farthest-point optimized point sets with maximized minimum distance. In: Proc. HPG. New York: ACM; 2011, p. 135–142. doi: 10.1145/2018323.2018345

[101] Öztireli, A.C., Gross, M.. Analysis and synthesis of point distributions based on pair correlation. ACM Transactions on Graphics 2012;31(6):170:1–170:10. doi: 10.1145/2366145.2366189

[102] Heck, D., Schlömer, T., Deussen, O.. Blue noise sam-

pling with controlled aliasing. ACM Transactions on Graphics 2013;32(3):25:1–25:12. doi: 10.1145/2487228.2487233

[103] Reinert, B., Ritschel, T., Seidel, H.P., Georgiev, I.. Projective blue-noise sampling. Computer Graphics Forum 2015;35(1):285–295. doi: 10.1111/cgf.12725

[104] Yuksel, C.. Sample elimination for generating Poisson disk sample sets. Computer Graphics Forum 2015;34(2):25–32. doi: 10.1111/cgf.12538

[105] Ahmed, A.G.M., Perrier, H., Coeurjolly, D., Ostromoukhov, V., Guo, J., Yan, D.M., et al. Low-discrepancy blue noise sampling. ACM Transactions on Graphics 2016;35(6):247:1–247:13. doi: 10.1145/2980179.2980218

[106] Bowers, J., Wang, R., Wei, L.Y., Maletz, D.. Parallel Poisson disk sampling with spectrum analysis on surfaces. ACM Transactions on Graphics 2010;29(6):166:1–166:10. doi: 10.1145/1882261.1866188

[107] Hertzmann, A.. A survey of stroke-based rendering. IEEE Computer Graphics and Applications 2003;23(4):70–81. doi: 10.1109/MCG.2003.1210867

[108] Lu, A.. Volume illustration. Ph.D. thesis; Purdue University; USA; 2005.

[109] Meruvia Pastor, O.E.. Stippling for non-photorealistic animation and rendering. Ph.D. thesis; Otto-von-Guericke University of Magdeburg; Germany; 2003.

[110] Baer, A., Tietjen, C., Spindler, M., Preim, B.. Hardwaregestütztes Stippling von medizinischen Oberflächenmodellen. In: Proc. BVM. Berlin: Springer; 2006, p. 266–270. In German. doi: 10.1007/3-540-32137-3_54

[111] Baer, A., Tietjen, C., Bade, R., Preim, B.. Hardware-accelerated Stippling of Surfaces derived from Medical Volume Data. In: Proc. EuroVis. Goslar, Germany: Eurographics Assoc.; 2007, p. 235–242. doi: 10.2312/VisSym/EuroVis07/235-242

[112] Baer, A.. Hardwaregestütztes Stippling von medizinischen Oberflächenmodellen. Diplom thesis; University of Magdeburg; Germany; 2005. In German.

[113] Foster, K.. Precise ink illustrations of polygonal models. Master's thesis; Department of Computer Science, University of Calgary; Canada; 2005.

[114] Pfister, H., Zwicker, M., van Baar, J., Gross, M.. Surfels: Surface elements as rendering primitives. In: Proc. SIGGRAPH. New York: ACM; 2000, p. 335–342. doi: 10.1145/344779.344936

[115] Vital Brazil, E., Macêdo, I., Costa Sousa, M., Velho, L., de Figueiredo, L.H.. A few good samples: Shape & tone depiction for hermite RBF implicits. In: Proc. NPAR. New York: ACM; 2010, p. 7–15. doi: 10.1145/1809939.1809941

[116] Macedo, I., Gois, J.P., Velho, L.. Hermite radial basis functions implicits. Computer Graphics Forum 2011;30(1):27–42. doi: 10.1111/j.1467-8659.2010.01785.x

[117] Barla, P.. Representation and acquisition models for expressive rendering. Ph.D. thesis; INP Grenoble; France; 2006.

[118] Li, H., Mould, D.. Contrast-aware halftoning. Computer Graphics Forum 2010;29(2):273–280. doi: 10.1111/j.1467-8659.2009.01596.x

[119] Li, H.. Perception-motivated high quality stylization. Ph.D. thesis; Carleton University; Canada; 2014.

[120] Wang, H.Q.. Fast feature-guided stippling method based on threshold matrix. Journal of Computer Applications 2010;30(8):2105–2107. In Chinese. doi: 10.3724/SP.J.1087.2010.02105

[121] Isenberg, T.. A survey of illustrative visualization techniques for diffusion-weighted MRI tractography. In: Hotz, I., Schulz, T., editors. Visualization and Processing of Higher Order Descriptors for Multi-Valued Data; chap. 12. Berlin/Heidelberg: Springer; 2015, p. 235–256. doi: 10.1007/978-3-319-15090-1_12

[122] Berkiten, S., Fan, X., Rusinkiewicz, S.. Semi-automatic digital epigraphy from images with normals. In: Proc. NPAR. Goslar, Germany: Eurographics Assoc.; 2015, p. 91–102. doi: 10.2312/exp.20151182

[123] Germer, T.. RenderBots: Multiagentensysteme für NPR-Graphiken. Diplom thesis; University of Magdeburg; Germany; 2004. In German.

[124] Saito, T., Takahashi, T.. Comprehensible rendering of 3-D shapes.

ACM SIGGRAPH Computer Graphics 1990;24(3):197–206. doi: 10.1145/97880.97901

[125] Zander, J., Isenberg, T., Schlechtweg, S., Strothotte, T.. High quality hatching. Computer Graphics Forum 2004;23(3):421–430. doi: 10.1111/j.1467-8659.2004.00773.x

[126] Zander, J., Isenberg, T., Schlechtweg, S., Strothotte, T.. Creating high quality hatching illustrations. Tech. Rep. 12/2004; Dept. of Computer Science, University of Magdeburg; Germany; 2004.

[127] Isenberg, T.. Evaluating and validating non-photorealistic and illustrative rendering. In: Rosin, P., Collomosse, J., editors. Image and Video based Artistic Stylisation; vol. 42 of *Computational Imaging and Vision*; chap. 15. London, Heidelberg: Springer; 2013, p. 311–331. doi: 10.1007/978-1-4471-4519-6_15

[128] Halper, N., Mellin, M., Herrmann, C.S., Strothotte, T., Linneweber, V.. Psychology and non-photorealistic rendering: The beginning of a beautiful relationship. In: Mensch & Computer 2003: Interaktion in Bewegung; vol. 57. Vieweg + Teubner Verlag; 2003, p. 277–286. doi: 10.1007/978-3-322-80058-9_28

[129] Li, H., Mould, D., Davies, J.. Structure and aesthetics in non-photorealistic images. In: Proceedings of Graphics Interface. Toronto: CIPS; 2013, p. 181–188. doi: 10.20380/GI2013.24

[130] Rivotti, V., Proença, J., Jorge, J., Sousa, M.C.. Composition principles for quality depiction and aesthetics. In: Proc. CAe. Goslar, Germany: Eurographics Assoc.; 2007, p. 37–44. doi: 10.2312/COMPAESTH/COMPAESTH07/037-044

[131] Hall, P.M., Lehmann, A.S.. Don't measure—appreciate! NPR seen through the prism of art history. In: Rosin, P., Collomosse, J., editors. Image and Video based Artistic Stylisation; vol. 42 of *Computational Imaging and Vision*; chap. 16. London/Heidelberg: Springer; 2013, p. 333–351. doi: 10.1007/978-1-4471-4519-6_16

[132] Maciejewski, R., Isenberg, T., Andrews, W.M., Ebert, D.S., Sousa, M.C.. Aesthetics of hand-drawn vs. computer-generated stippling. In: Proc. CAe. Goslar, Germany: Eurographics Assoc.; 2007, p. 53–56. doi: 10.2312/COMPAESTH/COMPAESTH07/053-056

[133] Ulichney, R.A.. Dithering with blue noise. In: Proceedings of the IEEE. IEEE; 1988, p. 56–79.

[134] Wei, L.Y., Wang, R.. Differential domain analysis for non-uniform sampling. ACM Transactions on Graphics 2011;30(4):50:1–50:10. doi: 10.1145/2010324.1964945

[135] Maciejewski, R., Isenberg, T., Andrews, W.M., Ebert, D.S., Costa Sousa, M., Chen, W.. Measuring stipple aesthetics in hand-drawn and computer-generated images. IEEE Computer Graphics and Applications 2008;28(2):62–74. doi: 10.1109/MCG.2008.35

[136] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 2004;13(4):600–612. doi: 10.1109/TIP.2003.819861

[137] Salesin, D.H.. Non-photorealistic animation & rendering: 7 grand challenges. Keynote talk at NPAR; 2002.

[138] An Introduction to Digital Scanning. Agfa-Gevaert; 1994.

[139] Lau, D.L., Arce, G.R., Gallagher, N.C.. Digital halftoning by means of green-noise masks. Journal of the Optical Society of America 1999;16(7):1575–1586. doi: 10.1364/JOSAA.16.001575

[140] Lau, D.L., Arce, G.R., Gallagher, N.C.. Green-noise digital halftoning. Proceedings of the IEEE 1998;86(12):2424–2444. doi: 10.1109/5.735449

[141] Lau, D.L., Arce, G.R.. Blue and green noise halftoning models. Signal Processing Magazine 2003;20(4):28–38. doi: 10.1109/msp.2003.1215229

[142] Eissele, M., Weiskopf, D., Ertl, T.. Frame-to-frame coherent halftoning in image space. In: Proc. TPCG. Los Alamitos: IEEE Computer Society; 2004, p. 188–195. doi: 10.1109/TPCG.2004.1314470

[143] Schmaltz, C., Gwosdek, P., Bruhn, A., Weickert, J.. Electrostatic halftoning. Computer Graphics Forum 2010;29(8):2313–2327. doi: 10.1111/j.1467-8659.2010.01716.x

[144] Ostromoukhov, V.. Artistic halftoning: Between technology and art. In: Proc. Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V; vol. 3963 of *SPIE Proceedings Series*. Bellingham, Washington: SPIE; 1999, p. 489–509. doi: 10.1117/12.373433

22

[145] Gatys, L.A., Ecker, A.S., Bethge, M.. A neural algorithm of artistic style. arXiv.org report 1508.06576; 2015.

[146] Gatys, L.A., Ecker, A.S., Bethge, M.. Image style transfer using convolutional neural networks. In: Proc. CVPR. Los Alamitos: IEEE Computer Society; 2016, p. 2414–2423. doi: 10.1109/CVPR.2016.265

[147] Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.. Controlling perceptual factors in neural style transfer. arXiv.org report 1611.07865; 2016.

[148] Li, C., Wand, M.. Combining markov random fields and convolutional neural networks for image synthesis. In: Proc. CVPR. Los Alamitos: IEEE Computer Society; 2016, p. 2479–2486. doi: 10.1109/CVPR.2016.272

[149] Selim, A., Elgharib, M., Doyle, L.. Painting style transfer for head portraits using convolutional neural networks. ACM Transactions on Graphics 2016;35(4):129:1–129:18. doi: 10.1145/2897824.2925968

[150] Ruder, M., Dosovitskiy, A., Brox, T.. Artistic style transfer for videos. In: Proc. GCPR. Cham, Switzerland: Springer International; 2016, p. 26–36. doi: 10.1007/978-3-319-45886-1_3

[151] Mould, D., Rosin, P.L.. A benchmark image set for evaluating stylization. In: Proc. NPAR (Expressive). Goslar, Germany: Eurographics Assoc.; 2016, p. 11–20. doi: 10.2312/exp.20161059

[152] Ostromoukhov, V.. A simple and efficient error-diffusion algorithm. In: Proc. SIGGRAPH. New York: ACM; 2001, p. 567–572. doi: 10.1145/383259.383326

[153] Velho, L., Gomes, J.d.M.. Digital halftoning with space filling curves. ACM SIGGRAPH Computer Graphics 1991;25(4):81–90. doi: 10.1145/127719.122727

[154] Wong, T.T., Hsu, S.c.. Halftoning with selective precipitation and adaptive clustering. In: Paeth, A.W., editor. Graphics Gems V. Academic Press; 1995, p. 302–313.

[155] Land, E.H.. The retinex theory of color vision. Scientific American 1977;237(6):108–128.

[156] Canny, J.. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 1986;8(6):679–698. doi: 10.1109/TPAMI.1986.4767851

[157] Marr, D.. Vision. The MIT Press; 1982.

[158] Kang, H., Lee, S., Chui, C.K.. Flow-based image abstraction. IEEE Transactions on Visualization and Computer Graphics 2009;15(1):62–76. doi: 10.1109/TVCG.2008.81

Appendix A. Resources

As part of this survey we searched for existing online resources to generate example images as well as to verify the characteristics of the different approaches. We list the results of our data collection below for future use.

Appendix A.1. Demos and source code

Below we list some links to demos (binaries and source code) of the described system that are currently available for download.

- approach by Secord [12] (weighted Voronoi stippling):
  - original tool: `http://www.mrl.nyu.edu/~ajsecord/downloads.html` (via `https://archive.org/`; has some problems on modern versions of Windows; source code is available at `http://www.mrl.nyu.edu/~ajsecord/NNR/src/AJS_stipple.tar.gz` resp. the probably more recent version via archive.org using the link in the page above)
  - alternative 1: (not by the original paper author) `http://www.saliences.com/projects/npr/stippling/` (source code is available on GitHub: `https://github.com/thegrandpoobah/voronoi`)
  - alternative 2: (not by the original paper author) `http://www.evilmadscientist.com/2012/stipplegen-weighted-voronoi-stippling-and-tsp-paths-in-processing/`
- approach by Ostromoukhov et al. [82] (fast hierarchical importance sampling): `http://www-labs.iro.umontreal.ca/~ostrom/ImportanceSampling/`
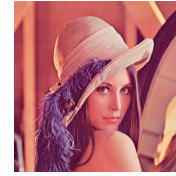

Figure A.23: Lena (Söderberg)—frequently used benchmark image from image processing, from the USC-SIPI Image Database. Used under the fair-use clause.

Figure A.24: Benchmark images used by Secord [12]. Images © 2007 Adrian Secord, used with permission.

- approach by Zander et al. [125, 126] (line stippling): `https://tobias.isenberg.cc/VideosAndDemos/Zander2004HQH`
- approach by Schlechtweg et al. [58, 66] (RenderBots): `https://code.google.com/archive/p/renderbots/` (source only, not by the original paper authors, status unclear)
- approach by Kopf et al. [4] (Wang tiling, continuous stipple zoom): `https://johanneskopf.de/publications/blue_noise/` (source code and set of Wang tiles are also available)
- approach by Ostromoukhov [84] (Polyominoes tiling): `http://www-labs.iro.umontreal.ca/~ostrom/SamplingWithPolyominoes/`
- approach by Schmidt et al. [55] (stippling for implicit 3D models): `https://tobias.isenberg.cc/VideosAndDemos/Schmidt2007SSI`
- approach by Balzer et al. [34] (capacity-constrained point distributions): `https://code.google.com/p/ccvt/` (source code only)
- approach by Kim et al. [25] (example-based stippling): `https://tobias.isenberg.cc/VideosAndDemos/Kim2009SBE`
- approach by Martín et al. [18, 19] (grayscale stippling): `https://tobias.isenberg.cc/VideosAndDemos/Martin2010EBS`
- approach by Kim et al. [22] (hedcut images): `https://www.facebook.com/Hedcutr-Community-301163773275653/` (status unclear)
- approach by de Goes et al. [73] (blue noise through optimal transport): `http://www.geometry.caltech.edu/BlueNoise/stippling.html`
- approach by Wachtel et al. [79] (polyhexes-based sampling with specified Fourier spectra): `https://liris.cnrs.fr/polyhex/`
- approach by Ahmed et al. [105] (low-discrepancy blue noise sampling): `https://liris.cnrs.fr/ldbn/`
- StippleShop as presented in this survey: `https://tobias.isenberg.cc/VideosAndDemos/Martin2017SDS`; see also Section 7 and Appendix B

Appendix A.2. Commercial/proprietary tools

- Artistic Halftone Photoshop plug-ins: `http://www.pixeology.com/ArtisticHalftone/features.html`
- Zian Studio: `http://www.zianstudio.com/`
- SnapDot app by dotwerx: `http://www.snapdotapp.com/`
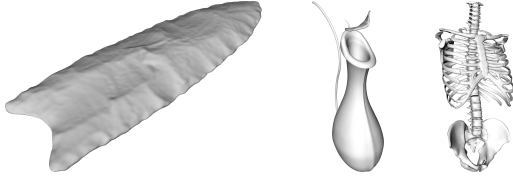
23

Figure A.25: Benchmark images used by Isenberg et al. [41]. Images © 2006 ACM, used with permission.
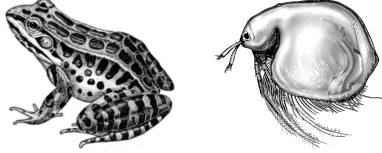

Figure A.26: Benchmark images used by Kim et al. [25]. Images © 2009 SungYe Kim (resized and blurred resp. contrast-enhanced based on public-domain images), used with permission.

Figure A.27: Benchmark images used by Martín et al. [18, 19]. Images © 2010 Domingo Martín Perandrés, used with permission.

Figure A.28: Benchmark images used by Li and Mould [40]. Images © 2011 Hua Li and David Mould, respectively, used with permission.

Appendix A.3. Benchmark images

For being able to compare different stippling approaches it is necessary to use the same or similar kinds of images. While some efforts are underway to define such a canonical image set for NPR in general [151], in stippling a series of images have been used that are, partially, still available:

- the image of Lena Söderberg that is frequently used for benchmarking image processing techniques: `http://sipi.usc.edu/database/database.php?volume=misc&image=12` (see Figure A.23)
- from the seminal paper on weighted Voronoi stippling by Secord [12]: `https://www.mrl.nyu.edu/~ajsecord/npar2002/StipplingOriginals.zip` (see Figure A.24)
- from the paper on zoomable stippling based on Wang tiles by Kopf et al. [4]: `https://johanneskopf.de/publications/blue_noise/stippling_demo/stippling_demo.zip` (part of the demo)
- from the paper on their NPR study by Isenberg et al. [41]: `https://tobias.isenberg.cc/uploads/VideosAndDemos/Study-Templates.zip` (see Figure A.25)
- from the paper on feature-guided image stippling by Kim et al. [21]: `http://cg.postech.ac.kr/research/stippling/results.php`
- from the paper on example-based stippling by Kim et al. [25]: `https://tobias.isenberg.cc/personal/demos/StipplingByExample.zip` (part of the demo; see Figure A.26)
- from the paper on example-based grayscale stippling by Martín et al. [18, 19]: `https://tobias.isenberg.cc/uploads/VideosAndDemos/GrayscaleStipplingBenchmarkImages.zip` (see Figure A.27)

- from the paper on a structure grid for directional stippling by Son et al. [23]: `http://cg.postech.ac.kr/research/structure_grid/results.php`
- from the paper on structure-preserving stippling by Li and Mould [40] (included in the paper PDF; see Figure A.28)

Appendix A.4. Study/example datasets

Below we list links to datasets (i. e., images) from studies on stippling.

- NPR study incl. stippling by Isenberg et al. [41]: `https://tobias.isenberg.cc/VideosAndDemos/Isenberg2006NPR`
- stipple dot analysis by Martín et al. [14]: `https://tobias.isenberg.cc/VideosAndDemos/Martin2015DCR`

Appendix B. StippleShop

StippleShop is programmed using Qt 5 and OpenCV 3.1. Its architecture is as follows: each operator/filter element receives up to two input images and produces a single output image. Each filter receives the input images as pointers to cv::Mat. The main function of each filter is the *update()* function which does the corresponding computation, generating the output image and other possible results. Each filter encapsulates its own graphics interface. So far we have implemented the following filters:

- placement and stippling
  - stippling based on weighted centroidal Voronoi diagrams by Secord [12]
  - recursive Wang tiling by Kopf et al. [4]
  - dot distributions based on capacity-constrained Voronoi tessellations by Balzer et al. [34]
  - stippling by example by Kim et al. [25]
  - resolution-dependent stippling by Martín et al. [18, 19]
  - structure-aware stippling by Li and Mould [40]
- dot output control
  - example-based grayscale dots by Martín et al. [18, 19]
  - SVG output
- halftoning
  - error diffusion by Ostromoukhov [152]
  - contrast-aware halftoning (without priority list) by Li and Mould [118]
  - space filling curves based on the work by Velho and Gomes [153] and Wong and Hsu [154]
  - adaptive clustering and selective precipitation by Wong and Hsu [154]
- contrast enhancement
  - contrast and brightness
  - Retinex filter by Land [155] (based on GIMP code)
- edge detection/enhancement
  - Canny's edge detection [156]
  - DoG: difference of Gaussians [157]
  - borders of the difference of Gaussians: edges are obtained using a scanline algorithm; this is useful because the edges are one pixel thick
  - Kang et al.'s edge detection [158]
- blur
  - Gaussian blur
- other filters
  - color inversion
  - threshold
  - dilation
  - distance field
- combination tools
  - combination of two inputs, pixel to pixel, with an associated operation (AND, OR, SUM, SUBSTRACTION, MULTIPLICATION, DIFFERENCE)
- measuring tools
  - SSIM [136] and PSNR