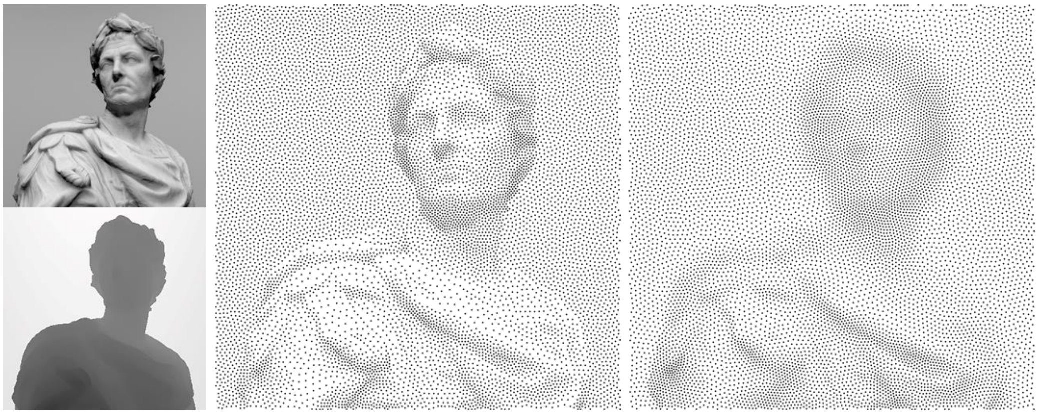
2.5D Computational Image Stippling

Kin-Ming Wong
artixels
mwkm@artixels.com

Tien-Tsin Wong
The Chinese University of Hong Kong
ttwong@cse.cuhk.edu.hk



(a) *Input pair (image + depth).* (b) *Regular stippling.* (c) *Stippling with depth of field using our method.*

Fig. 1. *2.5D computational image stippling examples (10,240 points)*

Abstract

We present a novel 2.5D[1] image stippling process that renders the photographic depth-of-field effect direct as an integral feature without any need of image filtering computation. Our approach relies on an additional depth image to produce the effect. The proposed method is based on a recent physically based blue noise sampling technique, which allows sampling naturally from spatial data, such as a 3D point cloud. The separation of the image data and its spatial information under our proposed 2.5D setting enables additional creative possibilities of image stippling art. Our approach can also produce an animated sequence that mimics the rack focus effect with good temporal coherence.

1. Introduction

Image stippling has a long history, dating back to the 16th century as a printmaking technique introduced by Giulio Campagnola [1] for reproducing smooth tones, shading and image details. This image-making technique uses only strong tone dots as the sole pictorial elements, and it demands an extremely skilful spatial arrangement. After centuries, stippling is still ubiquitous because of its unique aesthetics, the transparency of the process, and its simplicity as an art form.

Computational image stippling connects tightly to blue noise adaptive sampling techniques. Deussen and Isenberg [2] offer an excellent comprehensive review of its development. The term blue noise was formally defined and characterized by Ulichney [3] in his dithering research work. Figure 1b shows an example of how the structureless blue noise points reproduce pleasantly the underlying image tone with subtly varying yet uniform distribution.

Early research work in computer graphics related to blue noise and image stippling was driven by the need for tone reproduction improvement for early digital printing and display devices. Floyd and Steinberg [4] proposed the error diffusion technique, which stands as one of the best examples of how dithering improves tone reproduction. In the rendering research community, Dippé and Wold [5] proposed the use of Poisson disk sampling in rendering with reference to work on the study of spatial pattern of photo-receptors by Yellott [6].

[1] 2.5D image processing refers to techniques which take advantage of the per-pixel distance from camera information, i.e. depth information

Cook [7] further popularized the effectiveness of Poisson disk sampling, which is effectively a quality blue noise sampling point set.

Stippling-focused research work proposed by Deussen [8] relies on the relaxation technique proposed by Lloyd [9] to produce quality stipple drawings. To enable a more interactive experience, Secord [10] introduced a precomputed stipple tile-based approach, along with the weighted Voronoi method. Ostromoukhov et al. [11] and Kopf et al. [12] proposed improved tile-based acceleration techniques for better interactive image stippling.

More modern blue noise research work by Balzer et al. [13], namely the Capacity Constrained Voronoi Tessellation (CCVT) technique, is considered the state-of-the-art blue noise sampling method. CCVT serves as an important model, which inspired additional work. One such work was proposed by De Goes et al. [14], which formulated the capacity constrained model into an optimal transport problem, now commonly known as the BNOT method. The kernel density model proposed by Fattal [15] also set a new standard for blue noise sampling quality.

There are computational image stippling methods that are designed to improve the quality or variety of image stippling art from different perspectives. Pang et al. [16] proposed an approach that emphasizes reproduction of the structural details. Kim et al. [17] proposed an example-based stippling method that enables the use of sampled stippling patterns. Wei [18] introduced multi-class sampling, which enables more sophisticated stippling possibilities, and Li et al. [19] proposed an anisotropic technique, which substitutes dots with adaptive thin directional pictorial elements. Li and Mould [20] proposed a structure aware stippling method, which allows user-defined priority of stipple emphasis.

For the depth-of-field effect, there is no shortage of bitmap image filtering-based techniques [21, 22, 23], which render the photographic effect using an additional depth image. To the best of our knowledge, there has been no attempt to introduce photographic effects to the image stippling process as an integral feature without any pre-processing of the input image.

Our proposed 2.5D image stippling method renders the depth-of-field effect as a computation-free feature. We rely on the physically based blue noise sampling technique proposed by Wong and Wong [24] as the core of our approach. This sampling technique models the sample points as electrically charged particles, which self-organize by movement to reach an equilibrium. We apply an intuitive extension to this blue noise sampling method so that 2.5D image data can be adaptively sampled. This dynamics-based approach also allows us to produce an animated rack focus effect by changing the focus distance during simulation; the animated result shows stable temporal coherence.

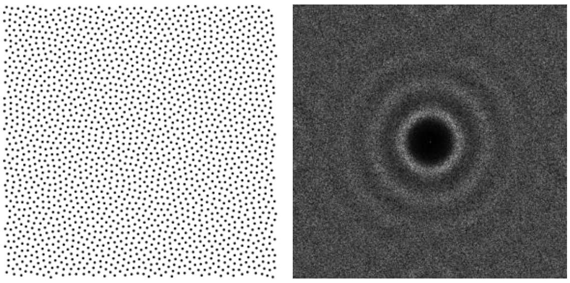
In section 2, we give a brief overview of the blue noise sampling technique used in our method and how it inspired our work. Section 3 describes the details of our extension for 2.5D image data sampling. In section 4, we demonstrate and evaluate the depth of field enabled stippling results from an artistic point of view. And in section 5, we discuss a few creative stippling applications based on our method.

2. Physically based Blue Noise Sampling

In this section, we review the blue noise sampling technique proposed by Wong and Wong [24], which serves as the foundation of our 2.5 image stippling method. This sampling method proposed a very intuitive approach, which models the sampling points as a system of electrically charged particles, with each carrying an identical charge. These like-charged particles repel each other, and the system undergoes self-organization by movement until it reaches an equilibrium state by maintaining a uniform equidistant neighbourhood around each particle. The particles' positions are then computed by integrating the equations of motion using a customized Velocity Verlet numerical integrator [25, 24], described in the original article. The whole idea is not totally innovative. It was first suggested by Hanson [26] and later by Schmaltz [27], but using a pure 2D electric field.

2.1 Uniform Sampling

Given a system of N particles constrained on an imaginary 2D plane, the total electrostatic force exerted on a particle p_i based on Coulomb's inverse-square law is governed by the following equation (eq. 1):



(a) *Uniform point set with* $\mathrm{q_s} = 0.25$*.* (b) *Power spectrum.*

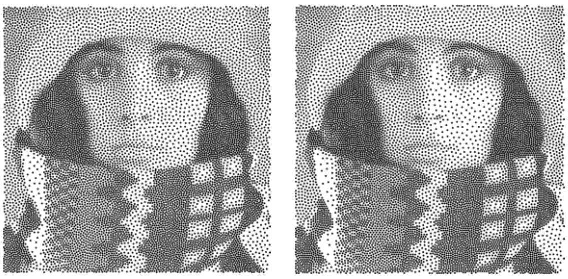
Fig. 2. *Uniform sampling using the physically based blue noise sampling method.* [24]

$$F_i = q_s^2 \sum_{j \neq i}^{N} \frac{1}{\left\|r_i - r_j\right\|^2} \hat{e}_{j,i}$$

where q_s is the amount of charge carried by each particle, r_i and r_j are the positions of particles p_i and p_j, respectively, and $\hat{e}_{j,i}$ is a unit vector pointing from r_j to r_i, which represents the direction of force. The process is simulated in a periodic domain, and the particles self-organize to reach an equilibrium state. Figure 2 shows a uniform point set generated using this physically based technique. This point set exhibits high-quality blue noise characteristics and is reflected by its power spectrum, as shown in Figure 2b.

2.2 Adaptive Sampling

What inspired our 2.5D image stippling approach is the adaptive sampling model proposed by this sampling method. To adaptively sample a varying density function, such as a bitmap image, the sampling method creates an additional imaginary 2D plane, named the density plane. On this new density plane, a regular grid of M non-moving attractively charged particles is created; each particle's charge is determined by the corresponding pixel that it represents. The amount of charge q_k carried by a given particle p_k on the density plane is defined as follows (eq. 2):



(a) $\mathrm{q_s} = 0.05$*.* (b) $\mathrm{q_s} = 0.35$*.*

Fig. 3. *Impact of sampling particle's charge* q_s *on adaptive sampling.*

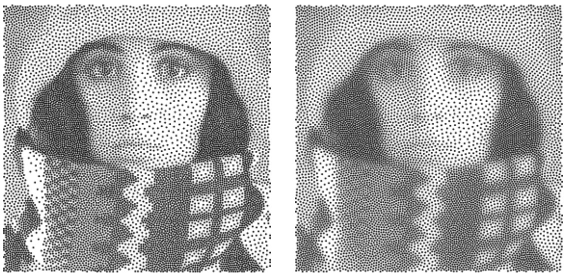
$$q_k = -A(1.0 - I_k)$$

where I_k is the pixel's intensity value that the particle p_k represents, and A is a positive valued coefficient determined by the total charge of the particles on the sampling plane. This relationship guarantees a total balance of potential. The force exerted on a particle p_i on the sampling plane by the charges on the density plane is governed by the following equation (eq. 3):

$$G_i = q_s^2 \sum_{k=1}^{M} \frac{q_k}{\|r_i - r_k\|^2} \hat{e}_{k,i}$$

The total force experienced by a particle p_i can be expressed as the sum of equations (1) and (3).

We carefully examined the stipple images produced by this blue noise sampling method, and we noticed that the amount of charge q_s carried by the sampling particles has an important impact on the overall image quality. Figure 3 shows a pair of stipple images produced using different values of q_s. A higher value of q_s produces an impression of better contrast. We believe it is a logical consequence that the larger force between sampling particles produces more space in the areas of low density (or brighter area), so it boosts the overall contrast. It is not



(a) *Small inter-plane distance.* (b) *Large inter-plane distance.*

Fig. 4. *Adaptive Sampling with different amount of inter-plane distance.*

hard to see that Figure 3b offers better contrast than Figure 3a. For a lower a value of q_s, we note that the points are obviously less structured, and they seem to be more sensitive to subtle local image structures too. In our experience, a higher value of q_s accelerates the convergence if it is a necessary factor to consider.

The density plane is by design placed tightly and parallel to the sampling plane to control the local density of the sampling particles. Wong and Wong [24] briefly demonstrated the impact of this inter-plane distance to the adaptive sampling results, and they named it a parameter for sharpness control. Figure 4 shows the effects of this parameter. It has an intuitive physical meaning here because according to Coulomb's inverse-square law, attractive force should be weakened and less localized when the distance between the sampling and the density planes increases, resulting in a stipple image that gives a blurred impression, as shown in Figure 4b. Although the force applied by the density plane, as expressed in Equation (3), assumes a planar arrangement of the particles, the model itself does permit a 3D configuration, as mentioned in Wong and Wong [24]. Our method exploits this 3D configuration possibility as the foundation of our depth-of-field effect integrated stippling technique.

3. 2.5D Image Stippling

By extending the idea of using a 2D density plane for adaptive sampling, we propose substituting the planar setup of density particles with a height-field alike configuration. In our new model, each density particle has its own depth from the sampling plane defined by an additional depth image. We also introduce a new parameter d_f, which defines the focus distance, so the density particles at a distance d_f from the sampling plane give an in-focus impression in the stipple result.

To achieve this visual effect, we displace the whole density field towards the sampling plane by d_f, so the in-focus density particles exert a strong attraction to the sampling particles. Based on this new proposal, we adapt Equation (3) to accommodate the changes. The force exerted by this new configuration is now governed by the following equation (eq. 4):

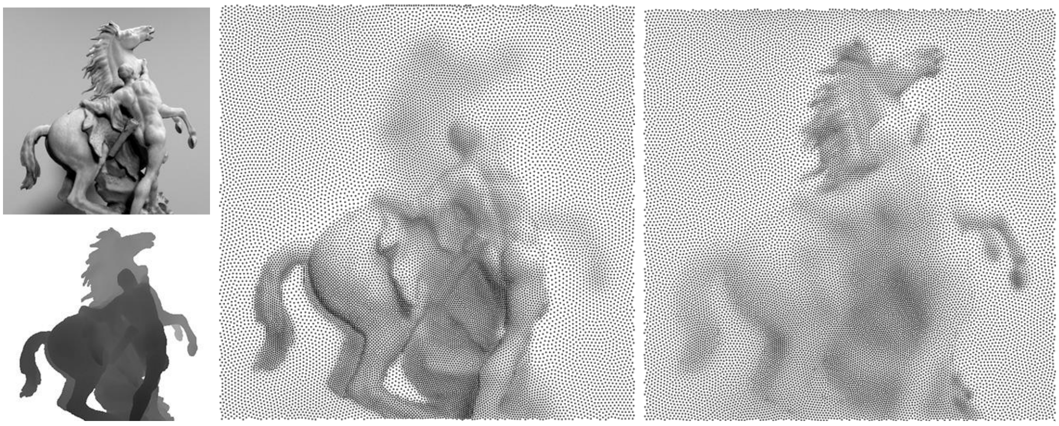
$$G'_i = q_s^2 \sum_{k=1}^{M} \frac{q_k}{\|r_i - r'_k\|^2 + \varepsilon} \hat{e}_{k,i}$$

where $r'_k = r_k - (0,0,d_f)$ is the new position of density particle p_k, $\hat{e}_{k,i}$ is a unit vector pointing from r'_k to r_i, and ε maintains a minimum distance between particles to avoid instability. To control the amount of depth of field, the depth component of all density particles can be globally scaled to achieve the desired degree of field depth.

We use the same numerical integrator described in Wong and Wong [24]; the algorithm is outlined in Algorithm 1. Using OpenGL compute shaders, we implemented a simple GPU application based on our method. Figure 5 shows an example of how our method is used to create stipple images from the same input with different focus distances. The average computation time of this example is 326ms per iteration, using an nVIDIA Geforce GT 650M mobile GPU.

Algorithm 1 Numerical Integrator

```
1. Position Update:
     $\vec{x}(t+\delta t) = \vec{x}(t) + min(\mathbf{D}, \vec{v}(t)\delta t + \frac{1}{2}\vec{a}(t)\delta t^2)$
2. Acceleration Update:
     Compute $\vec{a}(t+\delta t)$ using $\vec{x}(t+\delta t)$
3. Velocity Update:
     $\vec{v}(t+\delta t) = min(\frac{\mathbf{D}}{\delta t}, \mathbf{S}\vec{v}(t) + \frac{1}{2}(\vec{a}(t) + \vec{a}(t+\delta t))\delta t)$
4. Repeat
```



(a) *Input pair (image + depth).* (b) *Focus on the front, depth = 0.25.* (c) *Focus on the back, depth = 0.55.*

Fig. 5. *Image stippling examples with depth-of-field effect using our method; both used q_s = 0.3 and 150 iterations to converge.*

where **S** is a user-defined damping factor of a range of [0,1), which improves convergence. We find that a value of 0.95 works best in most scenarios. **D** defines the maximum per time-step displacement of each particle, which we keep constantly at 0.002, using a normalized coordinate system in our periodic simulation setting.

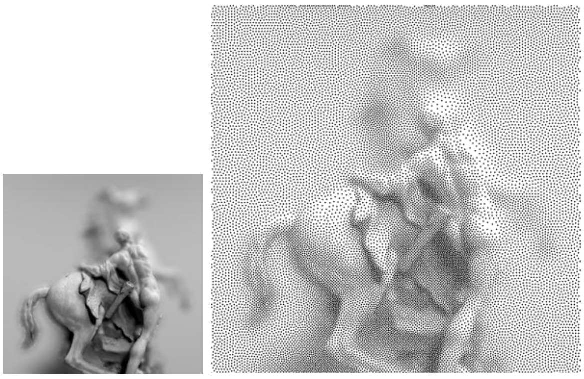
4. Evaluation

In this section, we evaluate the visual quality and image characteristics of our rendered output. In the PDF version of this paper, all stipple images are embedded in vector form for better visual examination.

4.1 Pre-filtered Depth of Field

The depth-of-field effect is traditionally achieved by applying adaptive filtering to a bitmap image, based on a depth map. We evaluate the qualitative difference between our results using the traditional approach from an artistic point of view instead of a technical one because our method is not designed to parallel or match the filtering result of the bitmap image-based technique.

We used commercial software [28] to obtain a pre-filtered bitmap, which is made to match the degree of depth of field in Figure 5b. Figure 6b shows a regular stippling result of the pre-filtered depth-of-field input using our method; it is not hard to observe that the stipple image using pre-filtered input maintains better contrast and a stronger photographic impression. Our depth-of-field result in Figure 5b, however, has a stronger illustrational and handcrafted quality. As our approach does not intend to accurately simulate the bitmap image filtering process, we believe that our result has a unique look with its own aesthetic qualities.

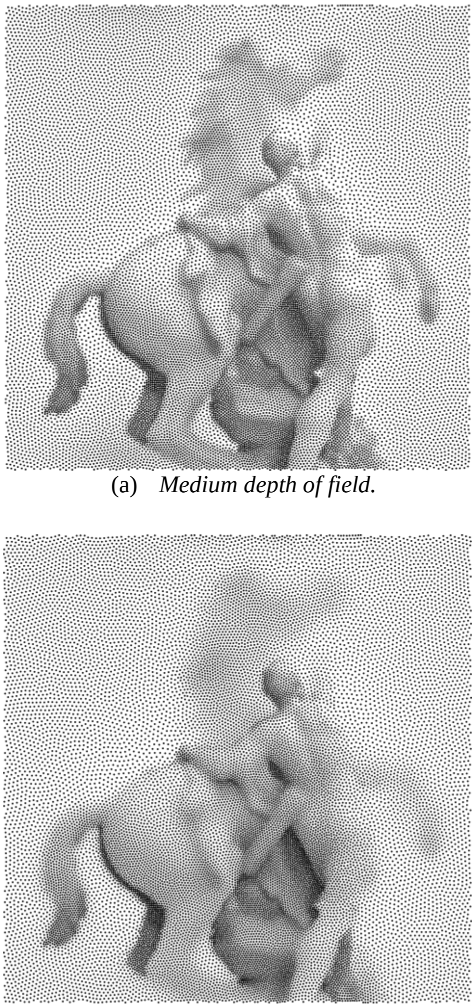


(a) *Pre-filtered input.* (b) q_s *= 0.3.*

Fig. 6. *Stipple image of the pre-filtered depth of field image.*

4.2 Degree of Depth of Field

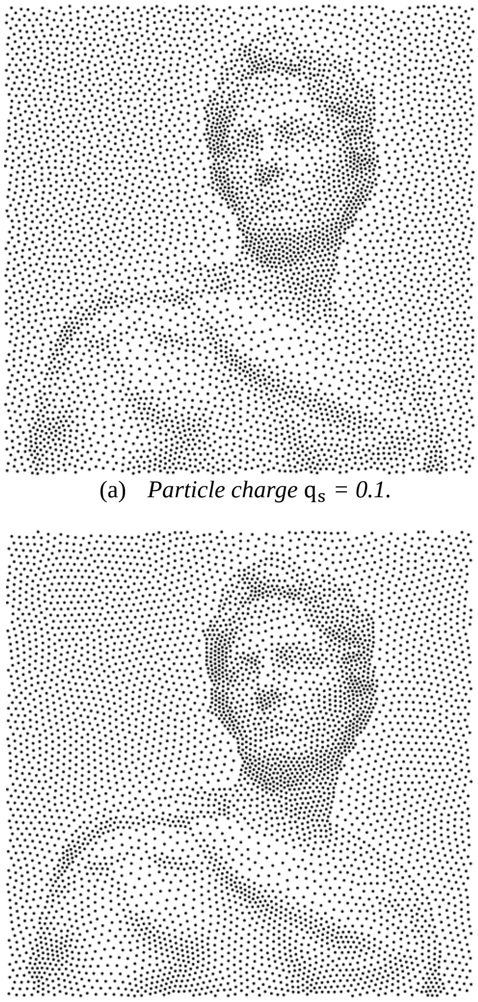
Our model allows different degrees of depth of field by globally scaling the depth component of the input depth map. Figure 8 shows two stippling results rendered with different depth scaling factors, while all other settings remain identical. The one with shallow depth of field,



(a) *Medium depth of field.*

(b) *Shallow depth of field.*

Fig. 7. *Different degrees of depth of field image.*



(a) *Particle charge* q_s *= 0.1.*

(b) *Particle charge* q_s *= 0.5.*

Fig. 8. *Effects of particle charge.*

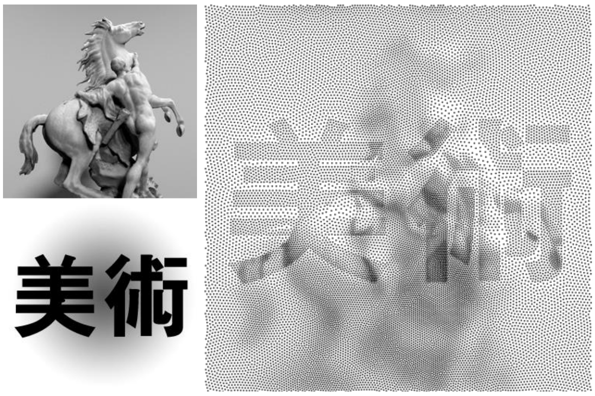
shown in Figure 7b, demonstrates stronger tone and local contrast on the dark in-focus areas. We believe this is a consequence of the relatively stronger attraction force and denser in-focus neighbourhood.

4.3 Tone and Feature Reproduction Characteristics

As mentioned above, the sampling particle's charge has an impact on the overall image contrast. This is an inherent property of the sampling method [24], but we take a deeper look at how this parameter q_s affects the overall image quality. We use a pair of stipple images with the same depth of field settings using a lower number of sample points (5,120 points) to illustrate our observations more clearly.

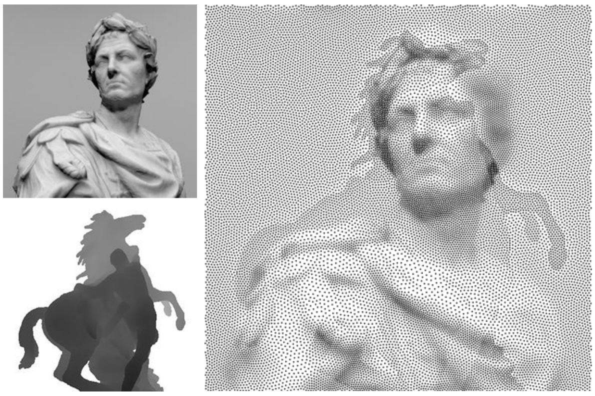
Figure 8a is produced using a smaller particle charge. It is not hard to observe that the stipple points on this image are far less structured than the ones in Figure 8b. The stipple points rely on various subtle and continuously varying density distributions to reveal the underlying image. This characteristic helps to maintain the subtle local tonal changes, and the whole image possesses a more organic quality from an artistic point of view.

In contrast, the stipple points in Figure 8b are more structurally organized; this is especially clear on the silhouettes and other sharp features. The overall image has more technical clarity, and better overall image contrast. We believe this setting is good for instructional or graphical illustration purposes.



(a) *Mixed inputs.* (b) *Stipple output.*

Fig. 9. *Mixed input for stylized stippling.*



(a) *Mixed inputs.* (b) *Stipple output.*

Fig. 10. *Mixed input for graphic design.*

5. Creative Possibilities

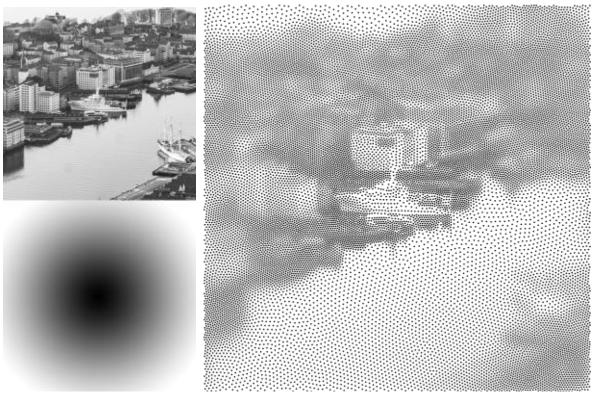
In this section, we explore various creative possibilities with our proposed method, ranging from general manipulation to photographic processing and animated sequence output.

5.1 Mixed Input as Masked Processing

As our method relies on a separate given depth image, users can always use a depth map that is not necessarily related to the image as a means to achieve other creative effects. Figures 9 and 10 show two creative uses of mixing an unrelated depth map to an image map to create a masked stippling.

5.2 Image Processing

To render the depth-of-field effect for bitmap images, image features more distant from the focus require more processing because of a larger filter kernel to process, but this does not apply to our stippling method. For general bitmap image processing based on convolution, we may loosely relate the filter kernel radius in bitmap image processing with the depth component of a density particle in our method. As an example of this connection, we follow how bitmap image processing creates a tilt-shift effect to a given image; this is usually achieved by applying a blurring process with a global radially increasing filter kernel radius. We reproduce it with a depth map which mimics the approach. Figure 11 shows the input pair and the result.



(a) *Input pair.* (b) *Stipple output.*

Fig. 11. *Tilt-shift alike image filtering.*

We believe this analogy between the kernel radius and the density particle's depth would serve as a good research direction for exploring systematic processing techniques for stipple images, or more precisely, point-based images.

5.3 Temporal Coherence of Stipple Image Sequence

We include with this paper a short video as supplemental material to demonstrate how our dynamics-based stippling method can be used to generate an animated sequence of stipple images that mimics the rack-focus effect. It can be used direct as the initialization point set for the next stipple computation. As long as the focus distance shifts slowly, the convergence of the new stipple image can happen in one or just a few time-steps in our experience.

More importantly, the two consecutive stipple images often demonstrate good temporal coherence. This is the advantage of the global, dynamics-based blue noise method proposed by Wong and Wong. [24] This temporal coherence is often hard to achieve with the sequential method or algorithms which rely on randomization.

Theoretically, this temporal coherence characteristic should also apply to animated video clip input, provided there is no vigorous change in image content, but this potential was not explored in the original paper.

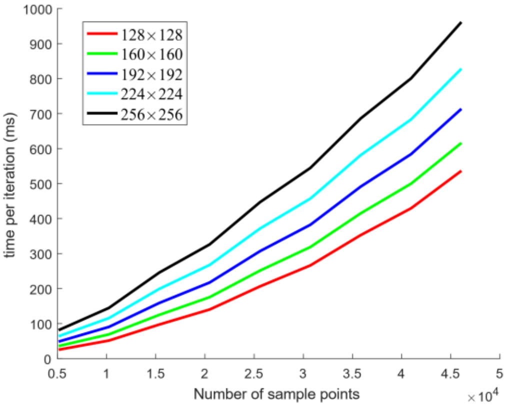
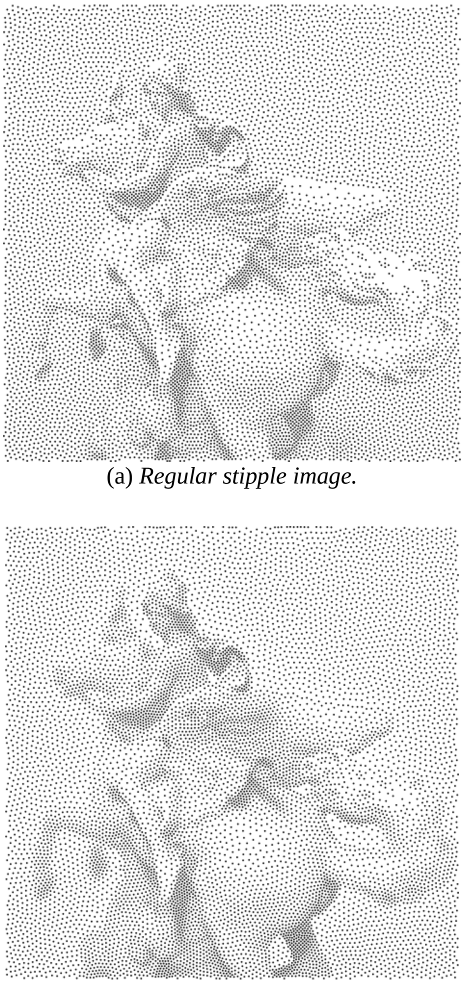


Fig. 12. *Per-iteration time performance on GT650M.*



(a) *Regular stipple image.*

(b) *Our stipple result with depth of field.*

Fig. 13. *Inconsistency of perceived brightness.*

6. Performance

We implemented a simple graphics processing unit (GPU) application using OpenGL compute shaders without any specialized algorithmic acceleration. Stippling computation time depends only on the number of sample points and the input image size; the degree of depth of field has no impact on our performance. For a stippling of 10,240 points and an input bitmap of size 256 × 256, each iteration takes less than 150ms on a modest Geforce GT650M notebook GPU.

Our compute shader parallelizes in a per sample point fashion, and the OpenGL compute shader allows us to maximize the use of local memory to minimize the GPU global memory bottleneck. A summary of timing information is provided in Figure 12, showing how computation time increases with the number of sample points under different input bitmap sizes. Although we believe our method should run impressively on more modern GPUs, to compute stippling with several hundred thousand sample points at an interactive rate, an algorithmic level acceleration is definitely necessary. The physically based blue noise sampling method [24] we use is practically an N-Body simulation, so any algorithmic acceleration for an N-Body simulation should work for our method too. The multi-level summation method proposed by Hardy et al. [29] and the non-equidistant fast Fourier transform-based acceleration method by Gwosdek et al. [30] are both applicable to our method.

In addition, the electric field of the density particles can be theoretically precomputed as a high resolution look-up table for runtime interpolation.

7. Discussion

We have presented a novel 2.5D image stippling method which is able to render certain photographic effects for free. Based on a global blue noise sampling technique, our method generates an animated sequence with effects with good temporal coherence.

However, we are aware that our method cannot maintain the consistency of the overall image brightness across stipples. Figure 13 shows a pair of images; Figure 13a is a regular stipple image, and in Figure 13b the depth-of-field effect was applied. There is an obvious tone difference between them, which can be explained by the concentration of attraction force. To provide overall brightness consistency, we believe that an algorithm to adjust the number of sample points has to be in place. This could be considered for future research.

References

1. G. Flocco, *La giovinezza di Giulio Campagnola'in L'Arte*, vol. xvii (1915).
2. Oliver Deussen and Tobias Isenberg, "Halftoning and stippling," in *Image and Video-Based Artistic Stylisation* (Springer), 45–61.
3. Robert A. Ulichney, "Dithering with blue noise," *Proc. IEEE* 76, no. 1 (1988): 56–79.
4. Robert W. Floyd, "An adaptive algorithm for spatial gray-scale," *Proc. Soc. Inf. Disp.,* 17 (1976): 75–77.
5. Mark A.Z. Dippé and Erling Henry Wold, "Antialiasing through stochastic sampling," *ACM Siggraph Computer Graphics* 19, no. 3 (1985): 69–78.
6. John I. Yellott, "Spectral consequences of photoreceptor sampling in the rhesus retina," *Science* 221, 4608 (1983).
7. Robert L. Cook, "Stochastic sampling in computer graphics," *ACM Transactions on Graphics (TOG)* 5, no. 1 (1986): 51–72.
8. Raanan Fattal, "Blue-noise point sampling using kernel density model," *ACM Transactions on Graphics (TOG)* 30 (2011): 48.
9. Stuart Lloyd, "Least squares quantization in PCM," *IEEE transactions on information theory* 28, no. 2 (1982): 129–137.
10. Adrian Secord, "Weighted voronoi stippling," *ACM Proceedings of the 2nd international symposium on non-photorealistic animation and rendering,* (2002): 37–43.
11. "The Foundry." *Nuke* 10.0, Vol. 3 (2016).
12. Johannes Kopf, Daniel Cohen-Or, Oliver Deussen, and Dani Lischinski, "Recursive Wang tiles for real-time blue noise," *ACM* 25 (2006).
13. Michael Balzer, Thomas Schlömer, and Oliver Deussen, "Capacity-constrained point distributions: a variant of Lloyd's method," *ACM* Vol. 28 (2009).
14. Fernando De Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun, "Blue noise through optimal transport," *ACM Transactions on Graphics (TOG)* 31, no. 6 (2012): 171.
15. Oliver Deussen, Stefan Hiller, Cornelius Van Overveld, and Thomas Strothotte, "Floating points: A method for computing stipple drawings," *Computer Graphics Forum* 19 (2000): 41–50.
16. Wai-Man Pang, Yingge Qu, Tien-Tsin Wong, Daniel Cohen-Or, and Pheng-Ann Heng, "Structure-aware halftoning," *ACM Transactions on Graphics* (TOG) 27 (2008): 89.
17. Sung Ye Kim, Ross Maciejewski, Tobias Isenberg, William M. Andrews, Wei Chen, Mario Costa Sousa, and David S. Ebert, "Stippling by example," *ACM Proceedings of the 7th International Symposium on Non-photorealistic Animation and Rendering* (2009): 41–50.
18. Li-Yi Wei, "Multi-class blue noise sampling" *ACM Transactions on Graphics (TOG)* 29, 4 (2010): 79.
19. Hua Li and David Mould, "Structure-preserving stippling by priority-based error diffusion," *Canadian Human-Computer Communications Society Proceedings of Graphics Interface* (2011): 127–134.
20. Hongwei Li, Li-Yi Wei, Pedro V Sander, and Chi-Wing Fu, "Anisotropic blue noise sampling," *ACM Transactions on Graphics (TOG)* 29 (2010): 167.
21. Joe Demers, "Depth of field: A survey of techniques," *GPU Gems* 1, 375 (2004), U390.

22. Jhonny Göransson and Andreas Karlsson, "Practical post-process depth of field," *GPU Gems* 3 (2007), 583–606.

23. Martin Kraus and Magnus Strengert, "Depth-of-Field Rendering by Pyramidal Image Processing," *Computer Graphics Forum*, 26 (2007): 645–654.

24. Kin-Ming Wong and Tien-Tsin Wong, "Blue Noise Sampling using an N-Body based Simulation Method," *The Visual Computer, Proceedings of Computer Graphics International* 33, 6-8 (2017): 823-832.

25. William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson, "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters," *The Journal of Chemical Physics* 76, 1 (1982): 637–649.

26. Kenneth M. Hanson, "Halftoning and Quasi-Monte Carlo," *Los Alamos National Library* (2005), 430–442.

27. Christian Schmaltz, Pascal Gwosdek, Andrés Bruhn, and Joachim Weickert, "Electrostatic halftoning," *Computer Graphics Forum* 29 (2010): 2313–2327.

28. Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin, "Fast hierarchical importance sampling with blue noise properties," *ACM Transactions on Graphics (TOG)* 23 (2004): 488–495.

29. David J. Hardy, John E. Stone, and Klaus Schulten, "Multilevel summation of electrostatic potentials using graphics processing units," *Parallel computing* 35, 3 (2009):164–177.

30. Pascal Gwosdek, Christian Schmaltz, Joachim Weickert, and Tanja Teuber, "Fast electrostatic halftoning," *Journal of real-time image processing* 9, 2 (2014): 379–392.