

Multi-Class Inverted Stippling

Supplemental Material

Christoph Schulz, Kin Chung Kwan, Michael Becher, Daniel Baumgartner,
Guido Reina, Oliver Deussen, Daniel Weiskopf

The followings are included in this material:

- Black-and-white stippling results
- Evaluations
- Color stippling results

Black-&-White

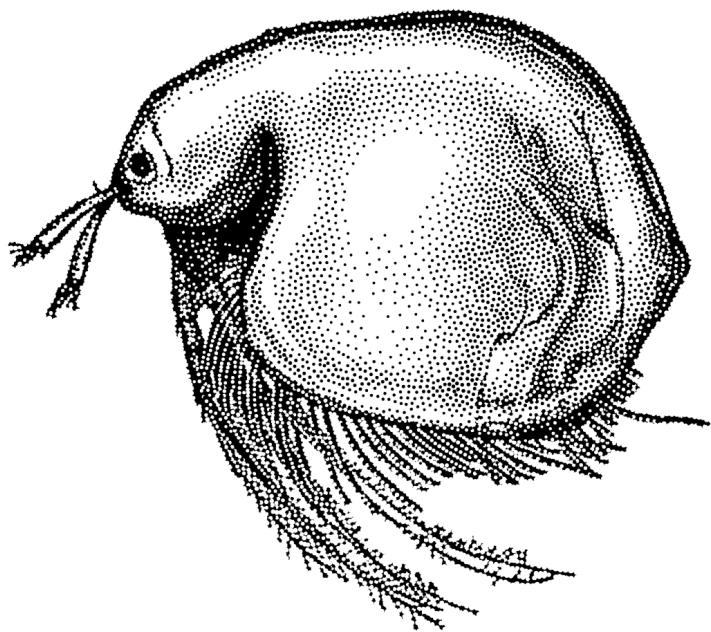
Stippling Results



Source

Ours

Figure 1: “Pirate.” Example results generated by our method.

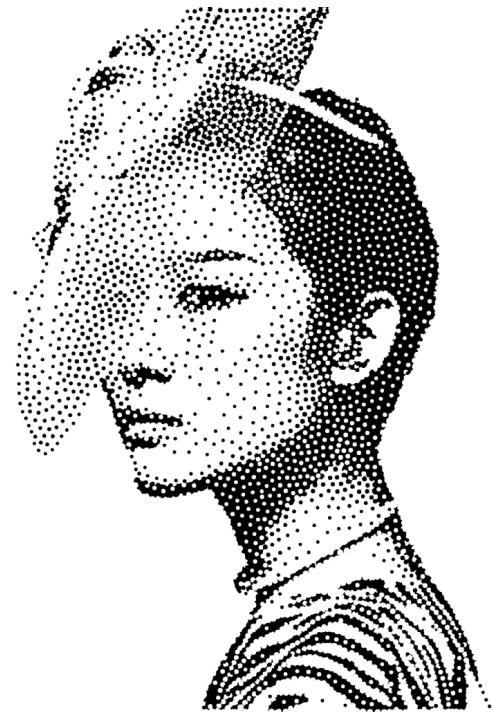


Ours

Figure 2: “Bug.” Resolution: 1024×1024 , dot size: 2–4, #dot: { 6586, 20456, 8903} , CPU Stippling time: 10.3s, Cleaning time: 1.9s. Source: <https://news.wisc.edu/newsphotos/waterflea1.html>



Source

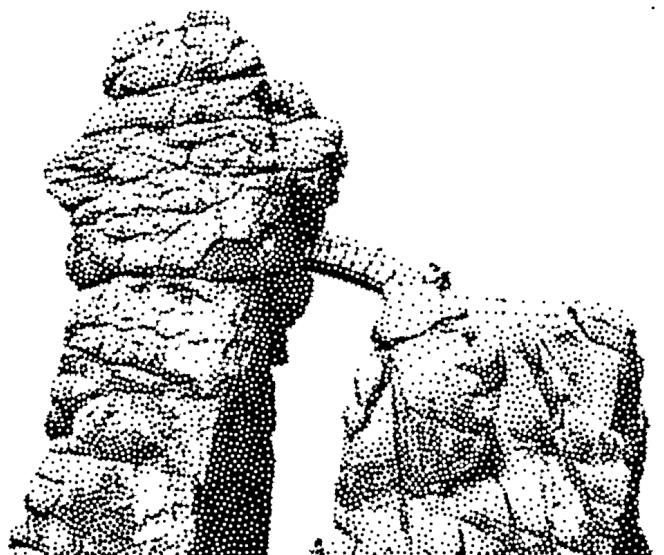


Ours

Figure 3: “Lady.” Resolution: 800×800 , dot size: 2–4, #dot: { 3441, 15664, 4943} , CPU Stippling time: 5.7s, Cleaning time: 3.4s. Source image credit: Flickr user “Búi Linh Ngân / CC by 2.0



Source



Ours

Figure 4: “Rocks.” Resolution: 1000×1000 , dot size: 2–4, #dot: { 15790, 4361, 33716}, CPU Stippling time: 6.3s, Cleaning time: 9.2s.

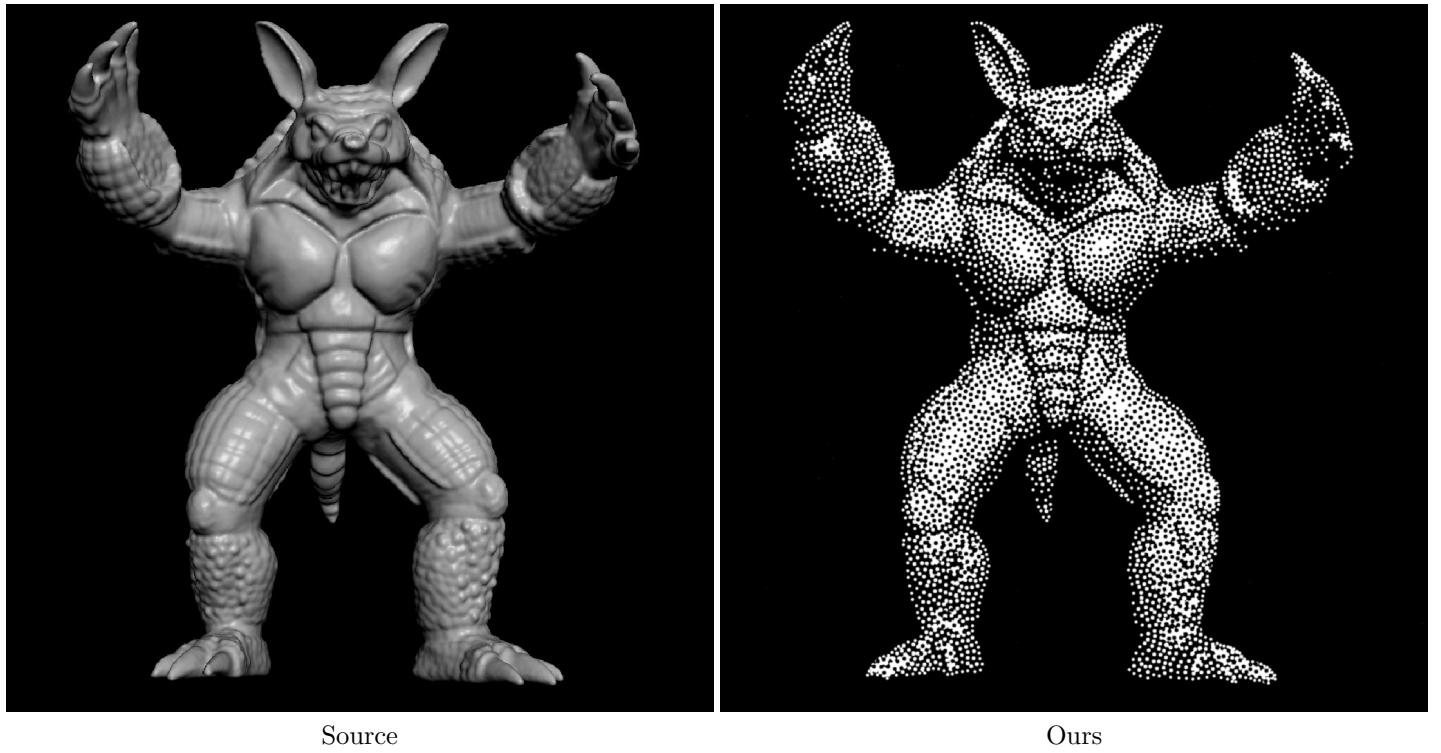


Figure 5: “Armadillo.” Resolution: 898×898 , dot size: 2–4, #dot: { 15790, 4361, 33716}, CPU Stippling time: 6.0s, Cleaning time: 6.0s. Source model credit: Stanford University Computer Graphics Laboratory.

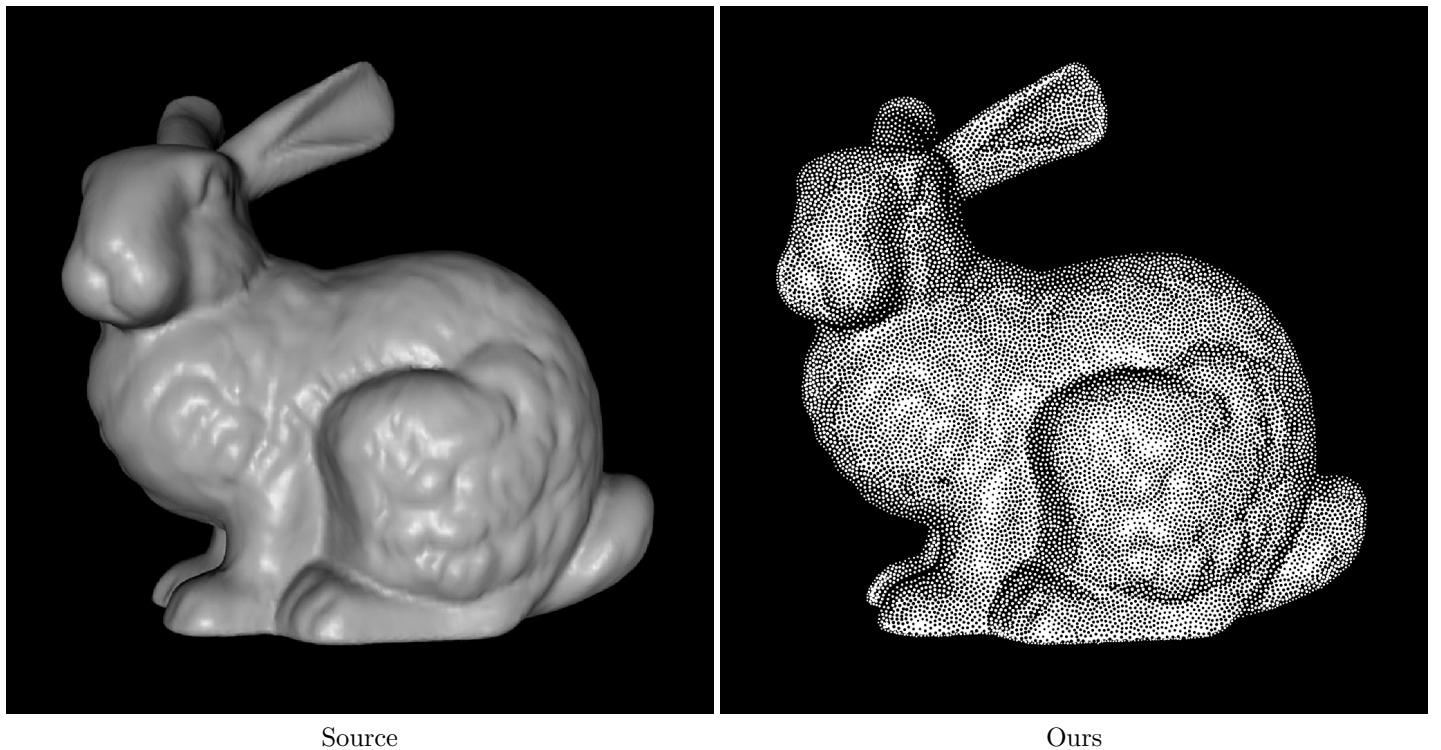


Figure 6: “Bunny.” Resolution: 800×800 , dot size: 1–3, #dot: { 24210, 11595, 34007}, CPU Stippling time: 12.3s, Cleaning time: 6.4s. Source model credit: Stanford University Computer Graphics Laboratory.

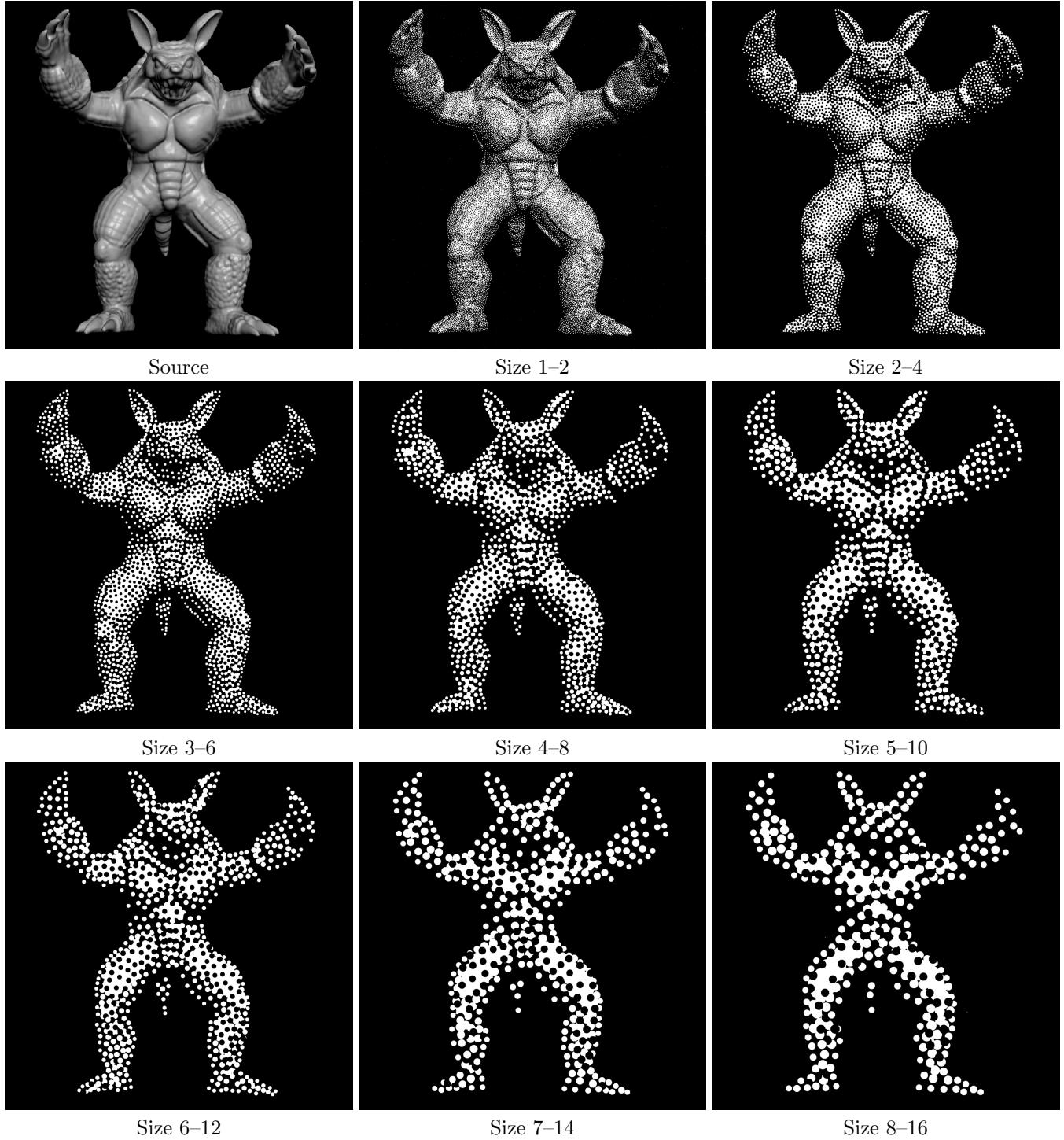


Figure 7: Our results with different dot size. The visual quality are largely impacted by the dot sizes. Source model credit: Stanford University Computer Graphics Laboratory.

Evaluations

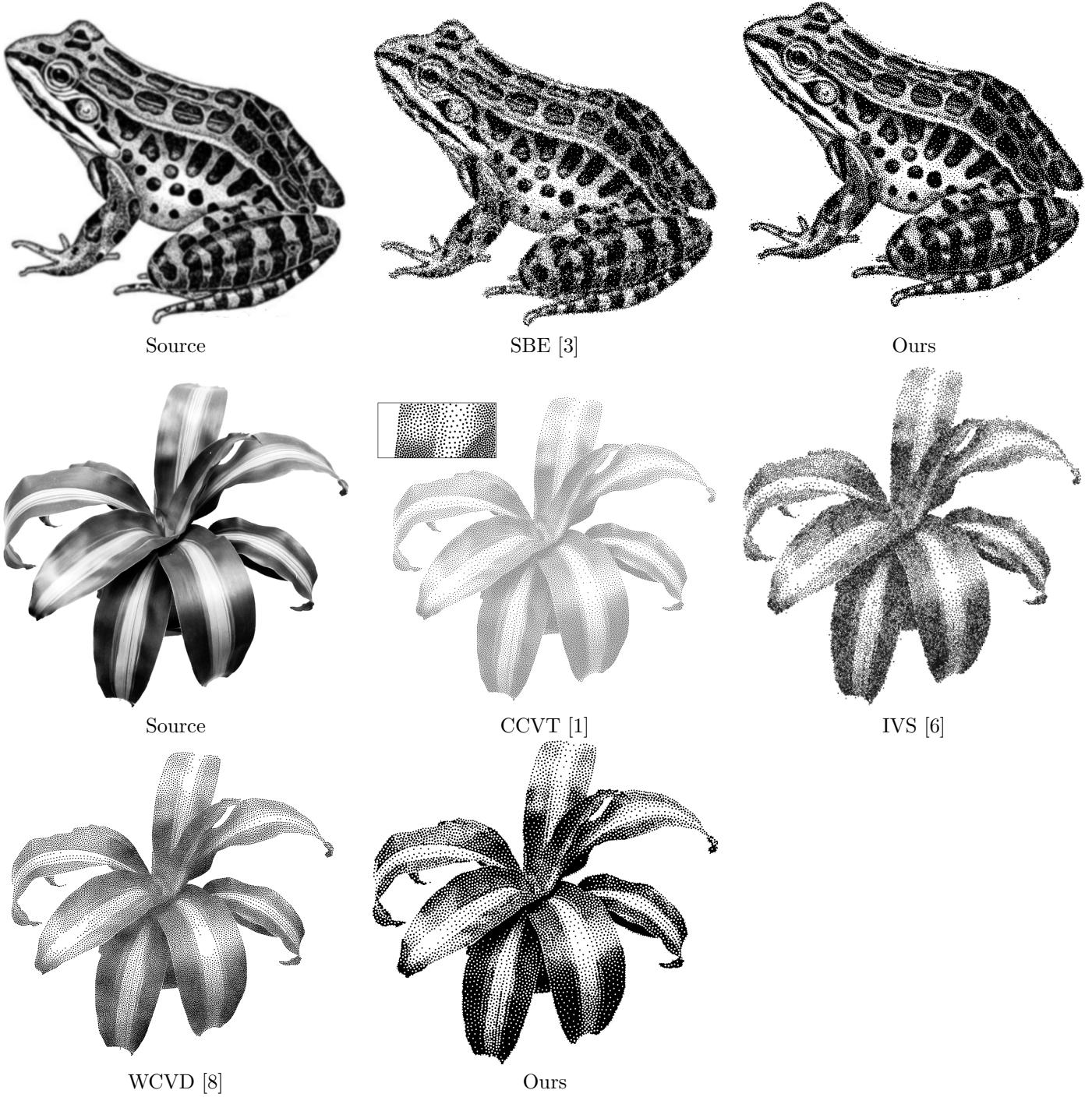
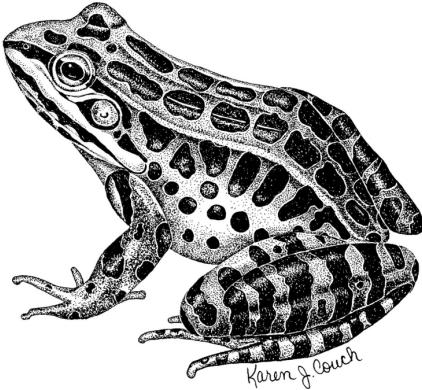


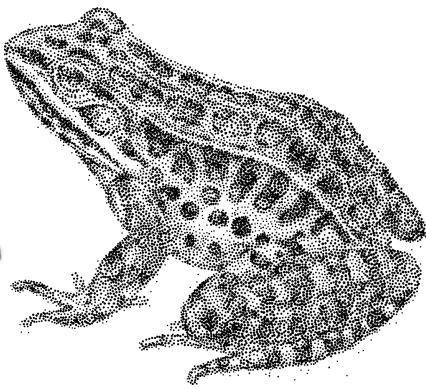
Figure 8: Comparisons of the images from the original paper. Used with permissions.



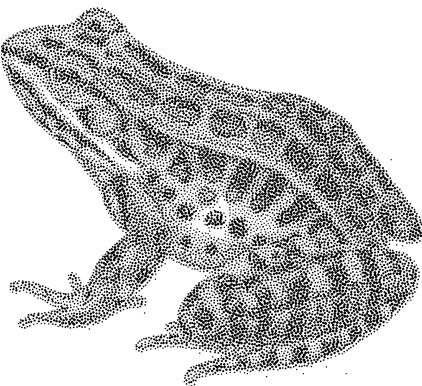
(a) Original artwork



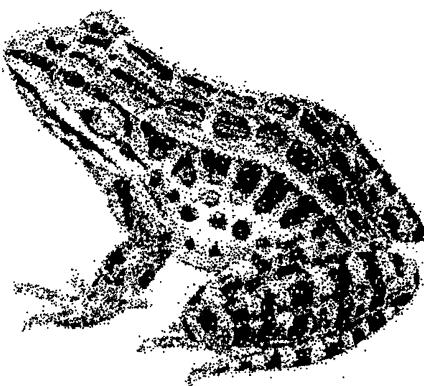
(b) Commonly used source



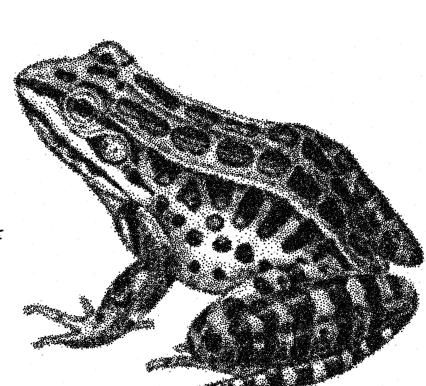
(c) WCVD [8]
Num: 9.9k, Size: 2 - 4



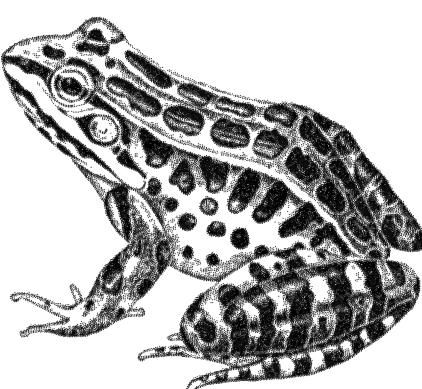
(d) CCVT [1]
Num: 9.7k, Size: 2 - 4



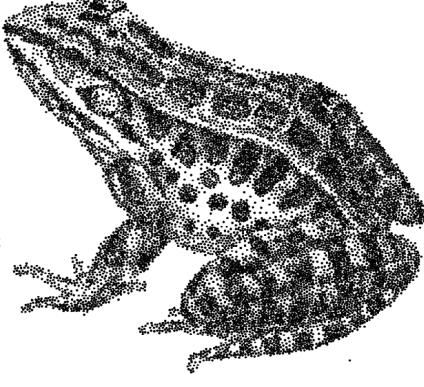
(e) SBE [3]
Num: 1.2M



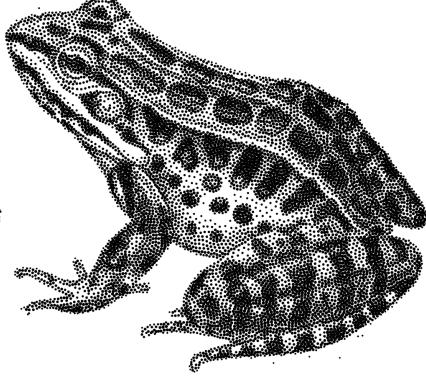
(f) RWT [4]
Num: 80k



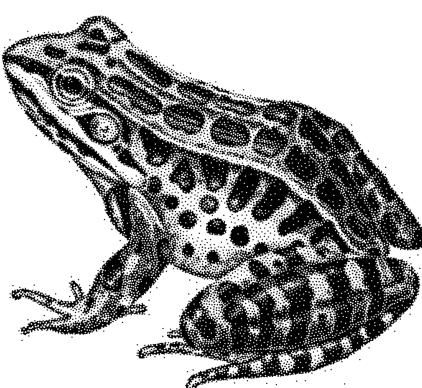
(g) CAS [5]
Num: 71.7k, Size: 2 - 4



(h) IVS [6]
Num: 1.3k, Size: 3



(i) LBG [2]
Num: 8.1k, Size: 2 - 4



(j) Ours
Num: 9k+18k+18k, Size: 2 - 4

Figure 9: Comparison of “frog” results generated by different methods. We try to keep the stipple size and numbers identical for fair comparisons. (c-e & g) are from StippleShop [7]. The stipple number of (e-i) and ours are automatically assigned by the method. (f & h & i) are from their provided demo program.

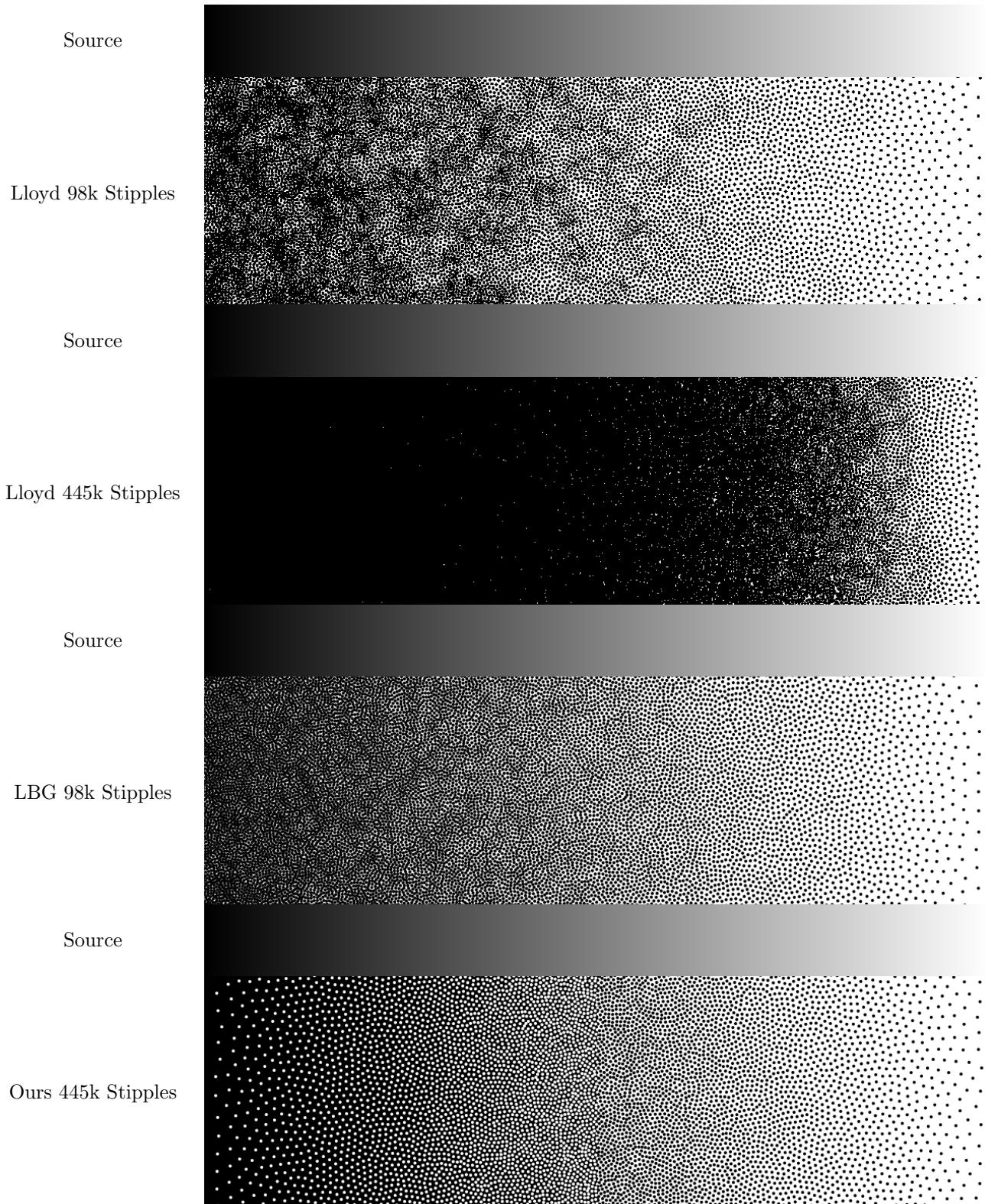


Figure 10: A comparison of the intensity of different methods. Stipples size: 2. The stipple number of LBG are automatically defined by their algorithm.

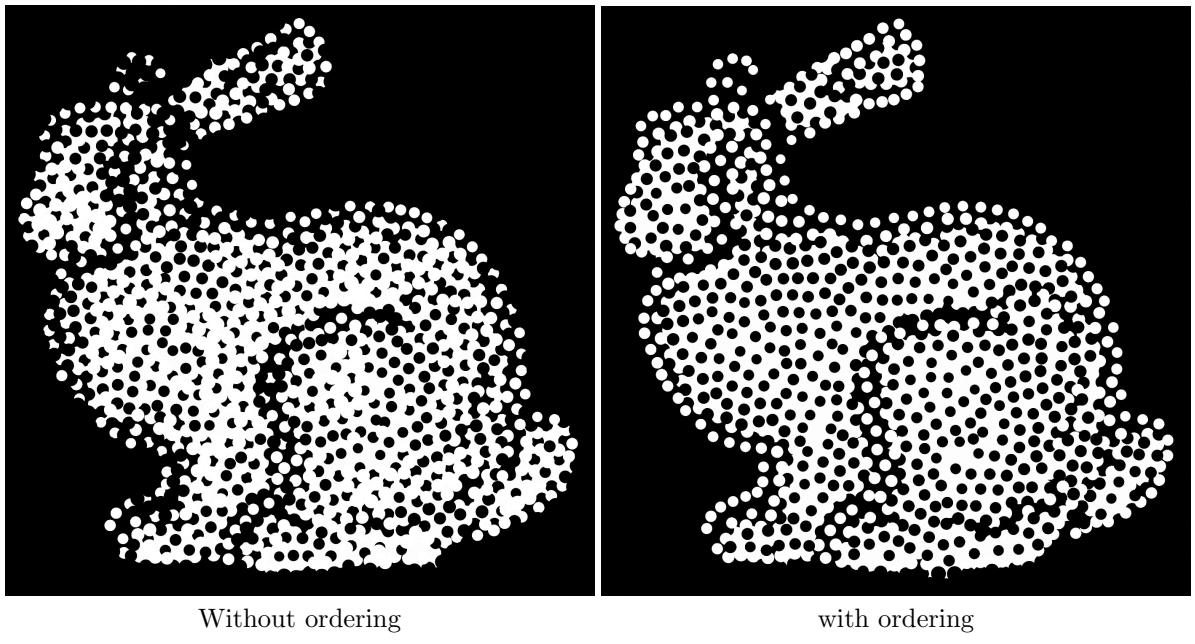


Figure 11: Another example of comparison with and without our ordering in inverted stippling

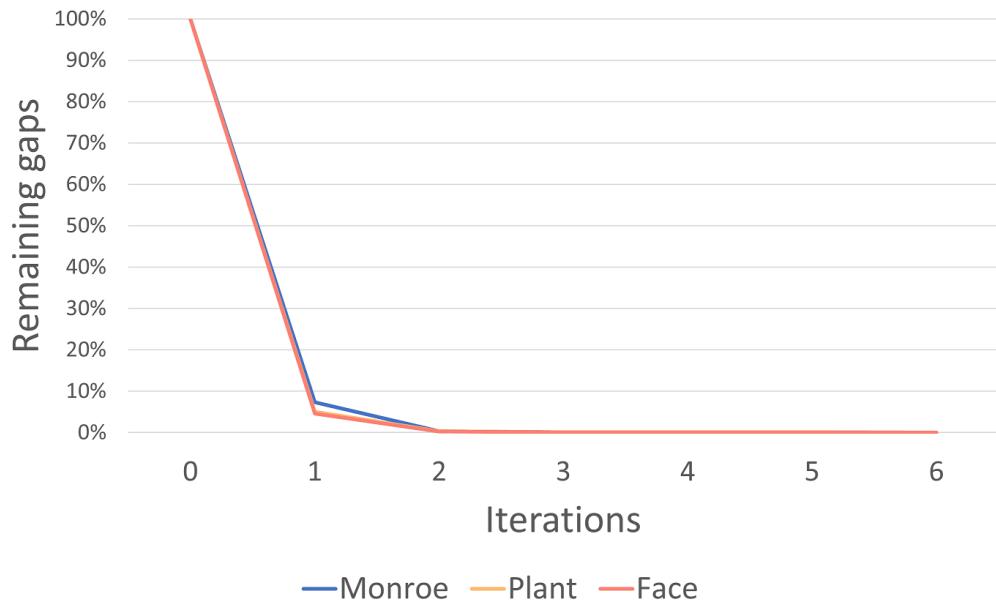


Figure 12: The plot of remaining gap for three different data during the first five iterations in our gap-filling approach. Monroe is a data with black background. Plant is a balanced black and white date. Face is a data mainly with bright color.

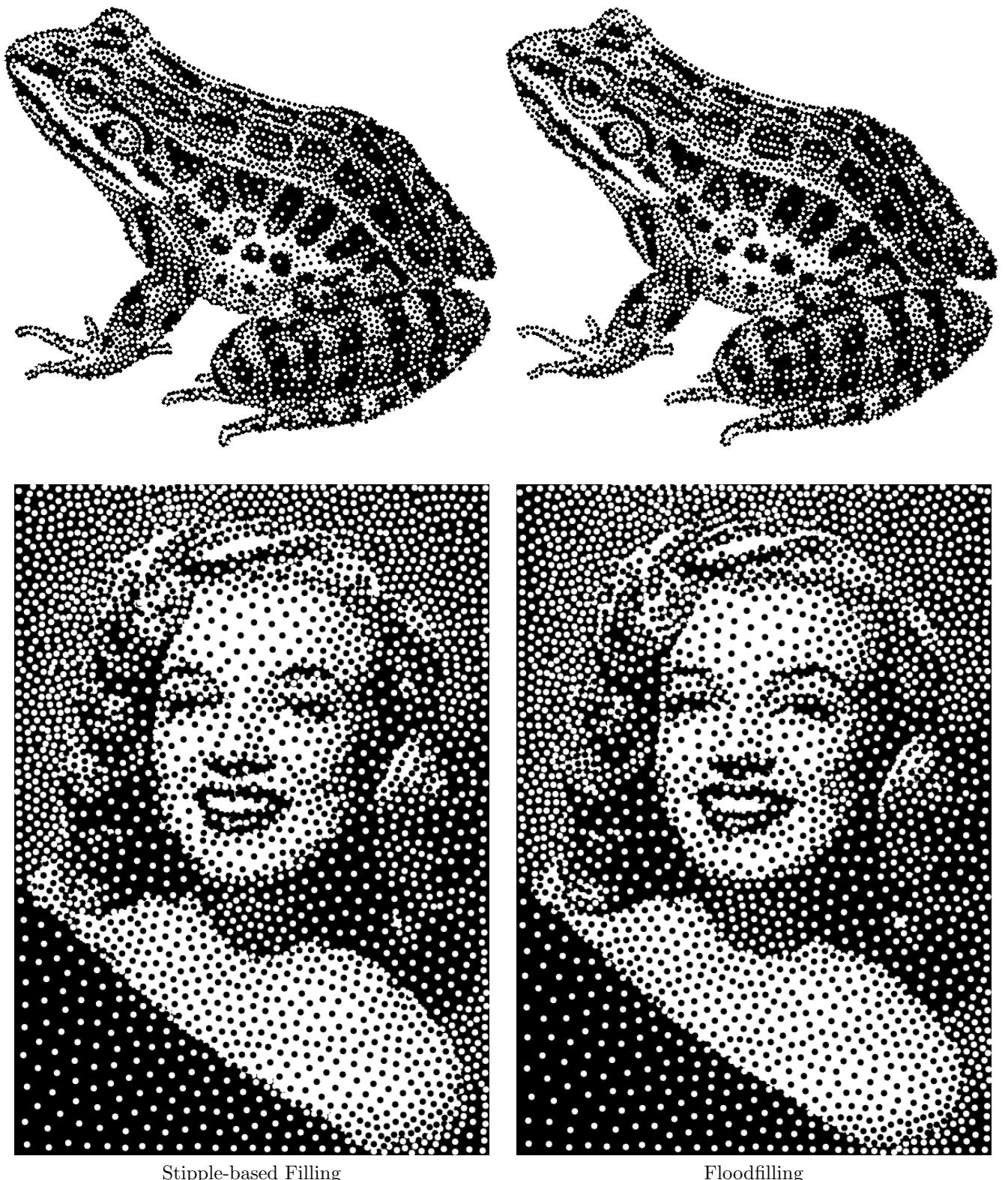


Figure 13: Comparison of using our stipple-based stippling and acceleration with flood filling. There is no obvious visual difference.

Color Stippling Results



Figure 14: “Street,” Resolution: 1000×667 , dot size: 1–2, #dot: { 26820, 35689, 12545, 9544 } , time: 4.4s. Source image used with Pixabay License.

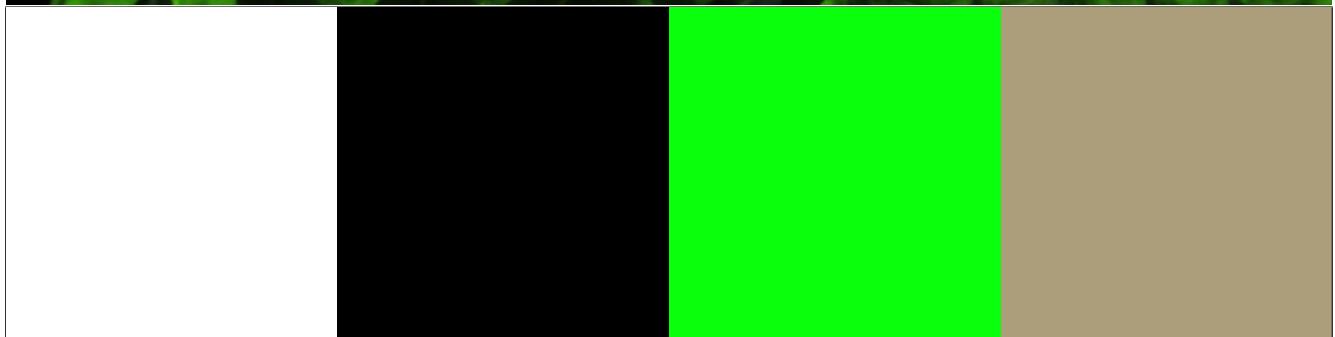


Figure 15: “Leave,” Resolution: 1920×1080 , dot size: 2–3, #dot: { 9029,57844,33000,30818 } , time: 13.4s. Source image used with Pixabay License.

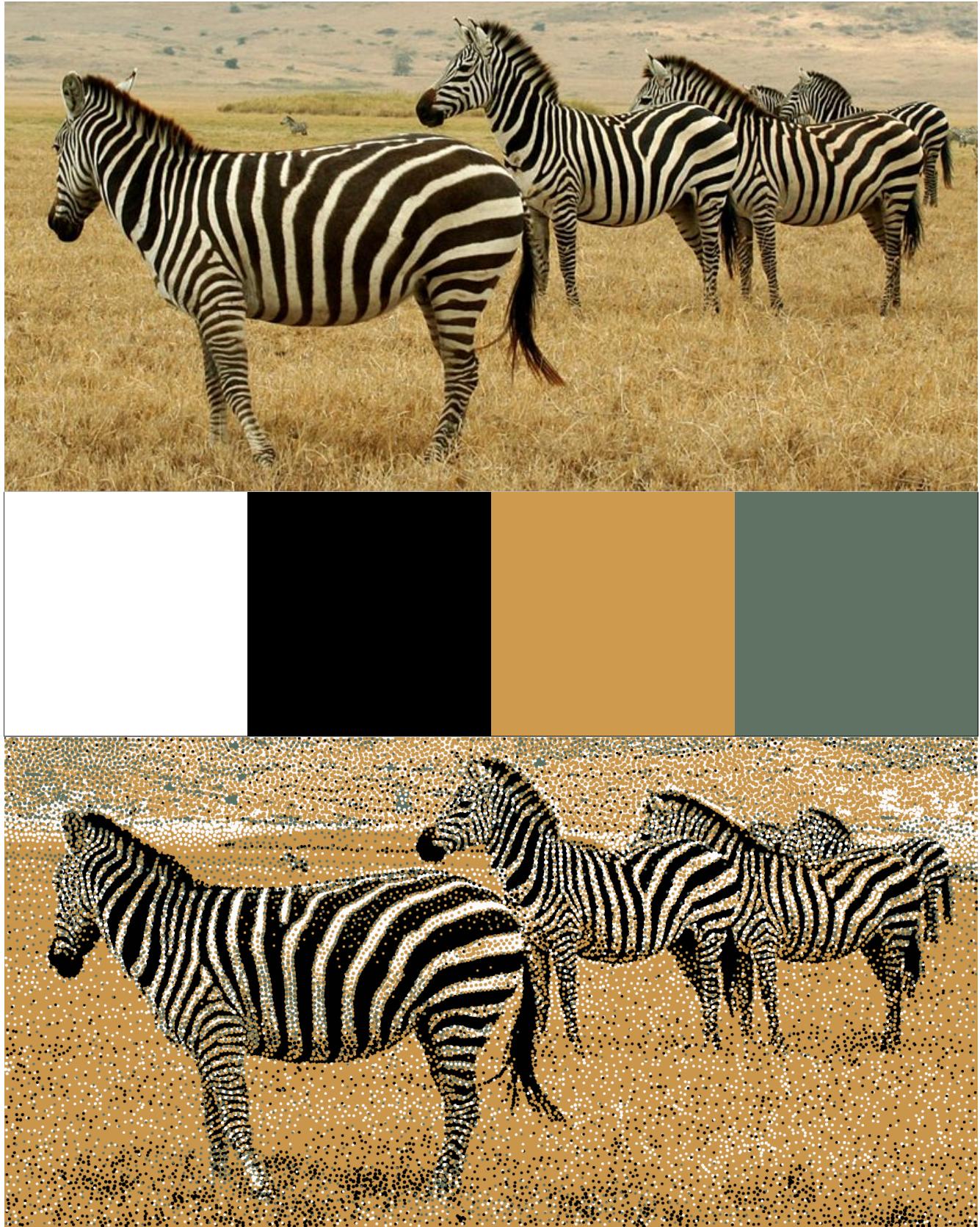


Figure 16: “Zebras,” Resolution: 800×402 , dot size: 1–2, #dot: { 10802,10702,22621,5647 } , time: 1.8s. Source image credit: Flicker user “Gaurav P.” / CC by 2.0

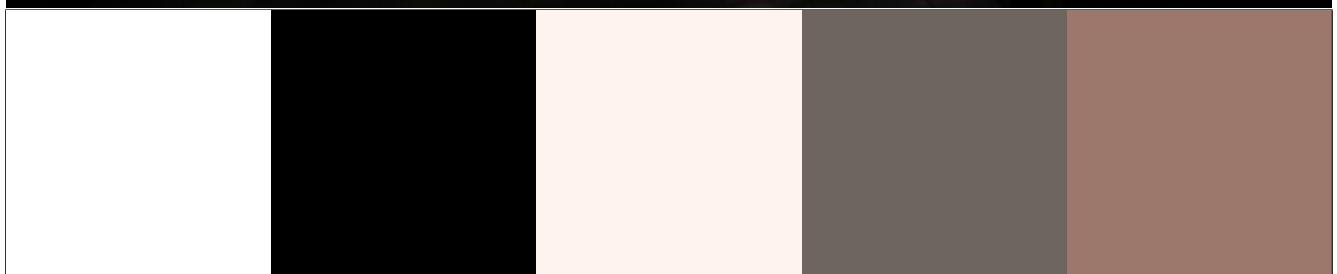


Figure 17: “Wedding,” Resolution: 1024×683 , dot size: 2–4, #dot: { 1747, 13229, 682, 837, 879 }, time: 4.7s. The source image is in public domain.

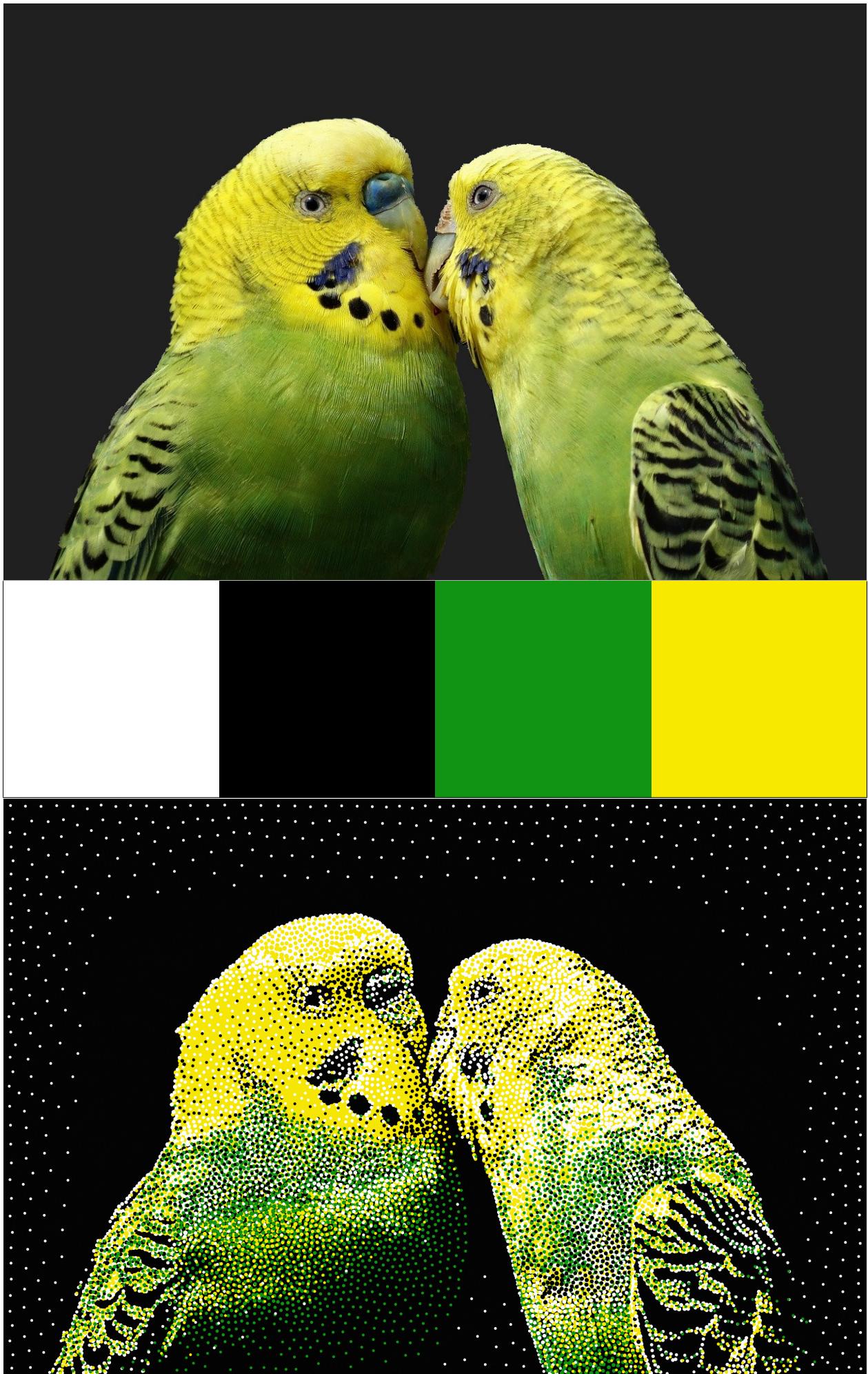


Figure 18: “Two birds,” Resolution: 1280×853 , dot size: 2–4, #dot: { 6596, 19060, 3998, 7428 } , time: 4.3s. Source image used with Pixabay License.

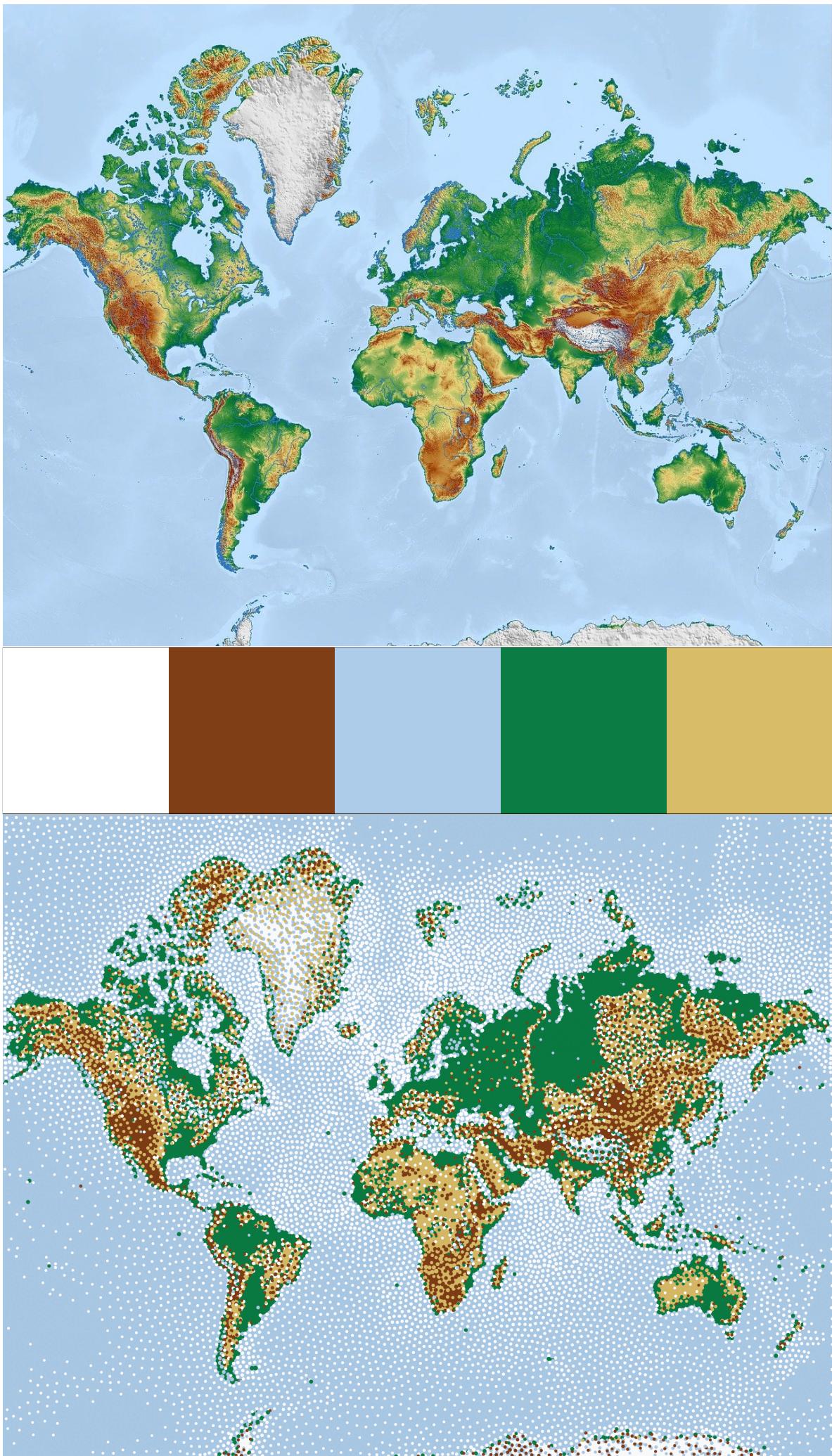


Figure 19: “Map,” Resolution: 1280×991 , dot size: 2–4, #dot: { 11841, 3021, 20388, 5321, 4156 } , time: 10.1s. Source image used with Pixabay License.

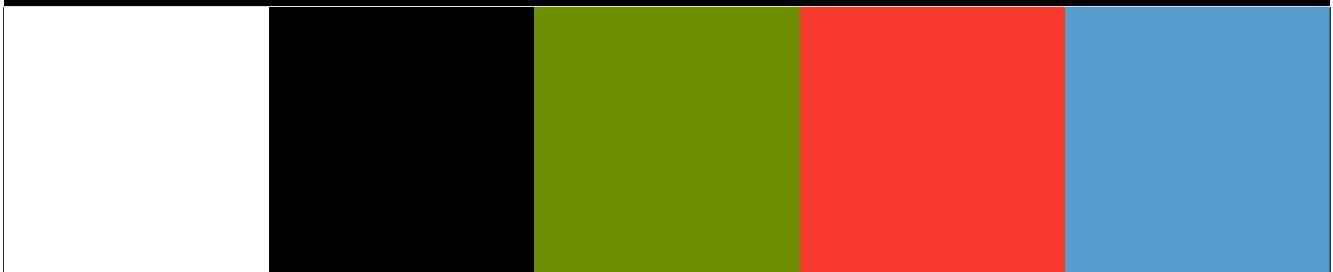


Figure 20: "Frog2," Resolution: 580×379 , dot size: 2-3, #dot: { 2560, 27263, 3361, 3906, 3032 } , time: 1.5s. Source Image credit: Sonja Pauen / CC By 2.0

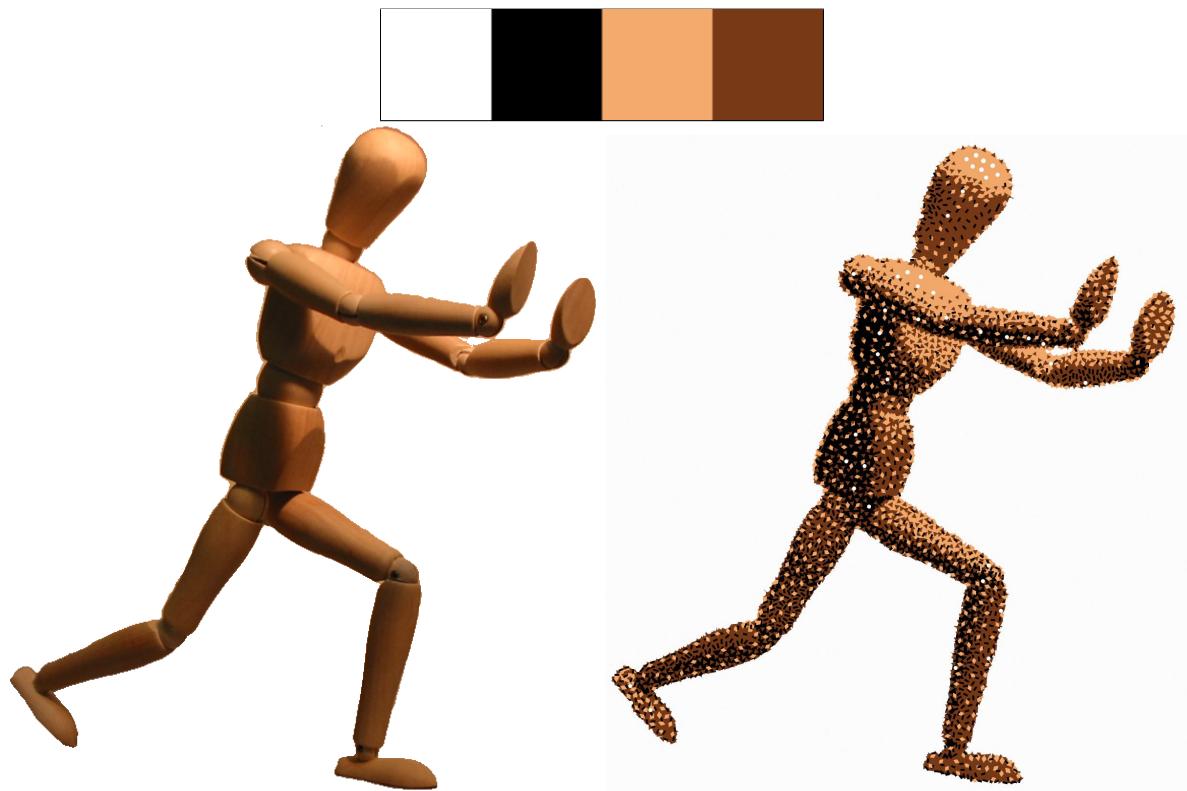


Figure 21: “Figure5.” Source image credit: Ardian Secord. Used with permission.



Figure 22: “Rose.” The source image is in public domain.

References

- [1] BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. Capacity-constrained point distributions: A variant of lloyd's method. *ACM Trans. Graph.* 28, 3 (2009), 1–8.
- [2] DEUSSEN, O., SPICKER, M., AND ZHENG, Q. Weighted linde-buzo-gray stippling. *ACM Trans. Graph.* 36, 6 (2017), 1–12.
- [3] KIM, S. Y., MACIEJEWSKI, R., ISENBERG, T., ANDREWS, W. M., CHEN, W., SOUSA, M. C., AND EBERT, D. S. Stippling by example. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (2009), NPAR '09, p. 41–50.
- [4] KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. Recursive wang tiles for real-time blue noise. *ACM Trans. Graph.* 25, 3 (2006), 509–518.
- [5] LI, H., AND MOULD, D. Structure-preserving stippling by priority-based error diffusion. In *Proc. Graph. Interf.* (2011), GI 2011, Canadian Human-Computer Communications Society, pp. 127–134.
- [6] MA, L., CHEN, Y., QIAN, Y., AND SUN, H. Incremental voronoi sets for instant stippling. *Vis. Comput.* 34, 6–8 (2018), 863–873.
- [7] MARTÍN, D., ARROYO, G., RODRÍGUEZ, A., AND ISENBERG, T. A survey of digital stippling. *Comput. Graph.* 67 (2017), 24–44.
- [8] SECORD, A. Weighted voronoi stippling. In *Proc. Sym. on Non-Photorealistic Animation and Rendering* (2002), NPAR '02, pp. 37–43.