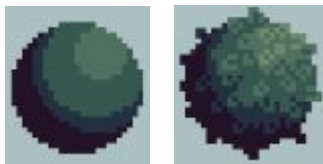


Procedural texturing for pixel art

1. Introduction

Texturing: how pixel art is made to look like an object made from a real material. Schlitter describes a method for texturing that involves arranging repeating elements throughout an image [1].

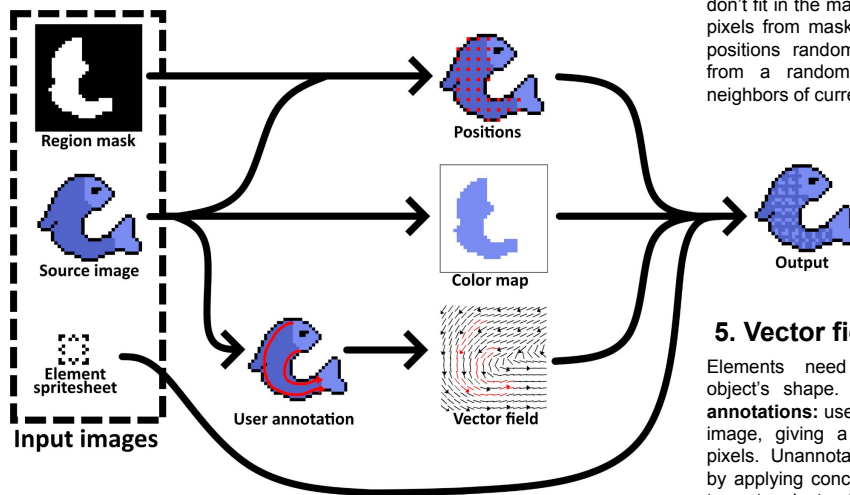


Shaded but untextured sphere next to textured sphere, looking like a bush. By Raymond Schlitter [1].

Repeatedly placing those elements can be tedious, and a procedural way of doing is an important step for procedurally generating pixel art.

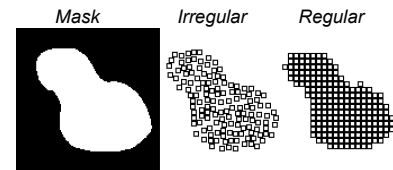
Research question:
How to automate texturing?

2. Algorithm overview



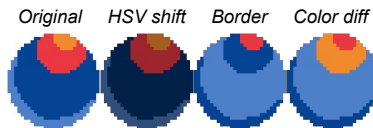
3. Positions

We consider positions, discarding them if they don't fit in the mask. Chosen positions remove pixels from mask. **Irregular patterns** pick all positions randomly. **Regular patterns** start from a random position and repeats for neighbors of current position.



4. Color map

Selects between strategies, according to best fit. Two of these **introduce no new colors** to palette.



5. Vector field

Elements need orientation to show object's shape. Do this through **user annotations**: user draws on top of source image, giving a direction to annotated pixels. Unannotated pixels get direction by applying concept of **diffusion curves** to vectors instead of colors [2].

Solve Poisson eq. to get vectors for all pixels:

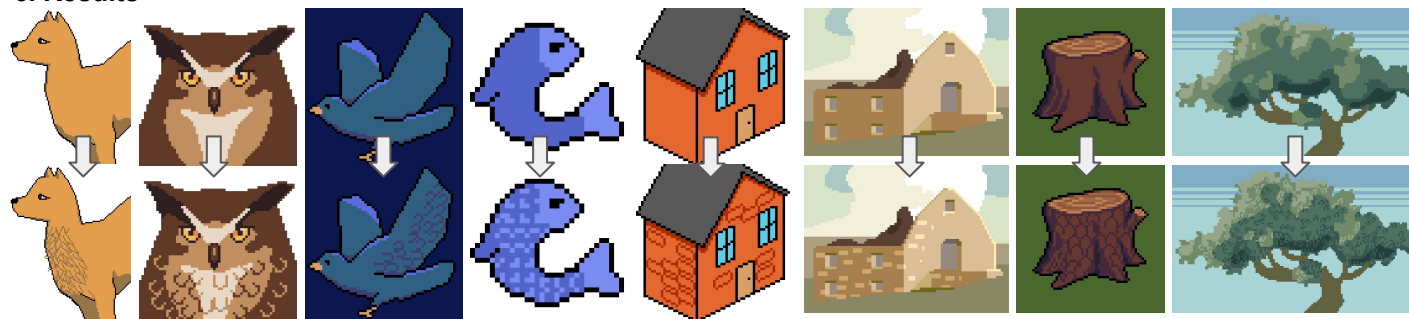
$$\Delta I = 0, \quad \Delta: \text{Laplace operator}$$

$$I: \text{Vectors of image}$$

$$I_k = V_k, \quad \forall k \in \mathcal{A} \quad \mathcal{A}: \text{Annotated pixels}$$

$$V_k: \text{Vector of pixel } k$$

6. Results



7. Conclusion

The algorithm procedurally adds detail to pixel art in order to make represented objects look like they are made from real materials, i.e. partly automates pixel art texturing. Many different kinds of textures can be generated by it, and several parameters allow users to tweak the resulting visuals.

References

- [1] R. Schlitter, "Pixelblog - 2 - Texture," SLNYRD. Accessed: Apr. 26, 2025. [Online]. Available: <https://www.slnyrd.com/blog/2018/2/15/pixelblog-2-texture>
- [2] H. Bezerra, E. Eisemann, D. DeCarlo, and J. Thollot, "Diffusion constraints for vector graphics," in Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering - NPAR 10, Annecy, France: ACM Press, 2010, pp. 35–42. doi: 10.1145/1809939.1809944.