Exploring the implementation of gradients in vector graphics images through colour diffusion graphs

University of Victoria

Linnea Goodwin, 2023

This research was supported by the Valerie Kuehne Undergraduate Research Awards, University of Victoria and supervised by Teseo Schneider of the Faculty of Computer Science

References:
Orzan, A., Bousseau, A., Winnemöller, H., Barla, P., Thollot, J., & Salesin, D. (2008). Diffusion Curves: A Vector Representation for Smooth-Shaded Images. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008), 27.

Introduction

The two main methods for digital image generation are bitmap and vector. Bitmap relies on pixel grids, where each pixel contains color data, but loses clarity upon zooming. Vector uses curves and equations to represent images, providing better scalability. However, achieving smooth gradients in vector images requires simulating diffusion, like ink spreading on a wet napkin. I calculated this using the finite element method in my project. Fig. 1 and 2 display a real ink blot on wet paper alongside a simulation I created, generated and simulated for the project. The image generation pipeline shown in Fig. 4 show the in-depth process of how I create images using this technique.

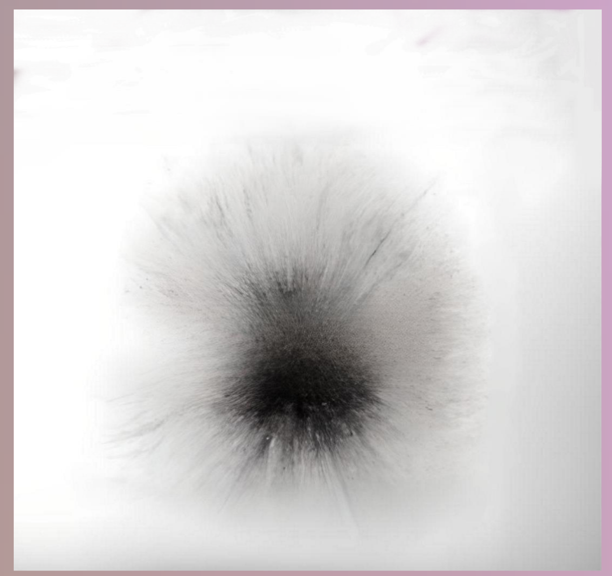


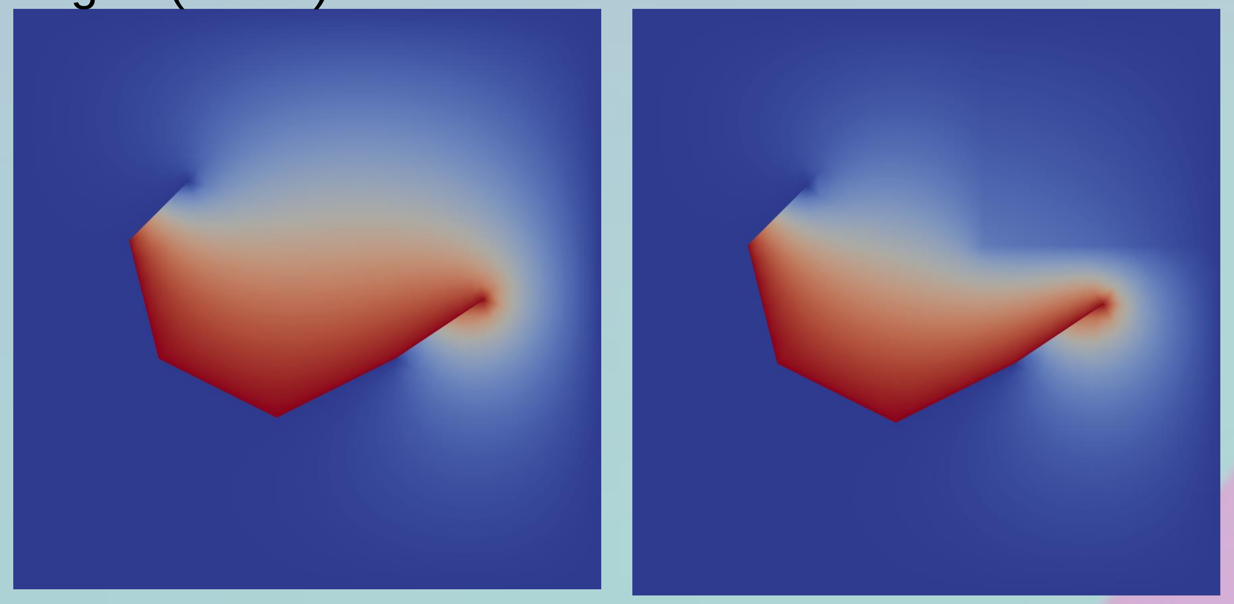
Fig. 1: A real inkblot on a real wet napkin

Fig. 2: A point I generated on a canvas and simulated diffusion on

Diffusion Constant

Everything discussed thus far has already been done in the world of vector graphics. What the project has since started focusing on is the process of slowing down or speeding up the diffusion process in given areas. This is as simple as making a file that maps each triangle in the mesh to a certain number. It was necessary to edit the PolyFEM repository (which handles all the calculus involved in calculating colour values relative to the input for each triangle in the mesh) to allow for such input. Here, you can see that the diffusion constant is normal in all sections of the mesh except for the top-right corner.

Fig. 4 (A & B)



The Future of the Project

This project is nowhere near finished, and I will be continuing my work on it throughout this semester. The diffusion constants has only recently been finished, so we don't have many cool examples. The current plan is to create some nice images and plot out their points–as can be seen here with this ladybug–and then convert those into cool-looking images using all we've created so far. Additionally, we have one more thing we want to implement–time-dependent diffusion. In essence, imagine an ink blot hitting a napkin and changing colours as it spreads. We have not begun work on this, but we will soon! We're hoping to create a research paper out of our findings and publish to a journal that will accept it.

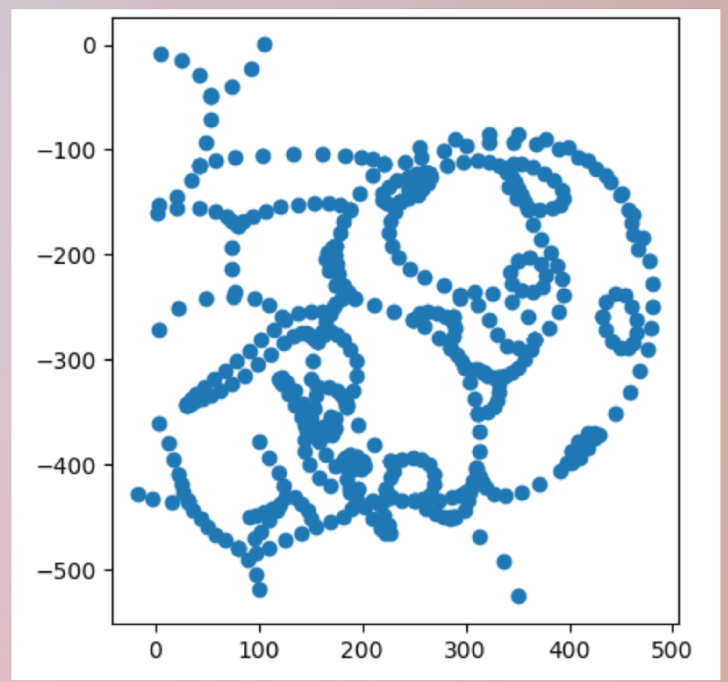
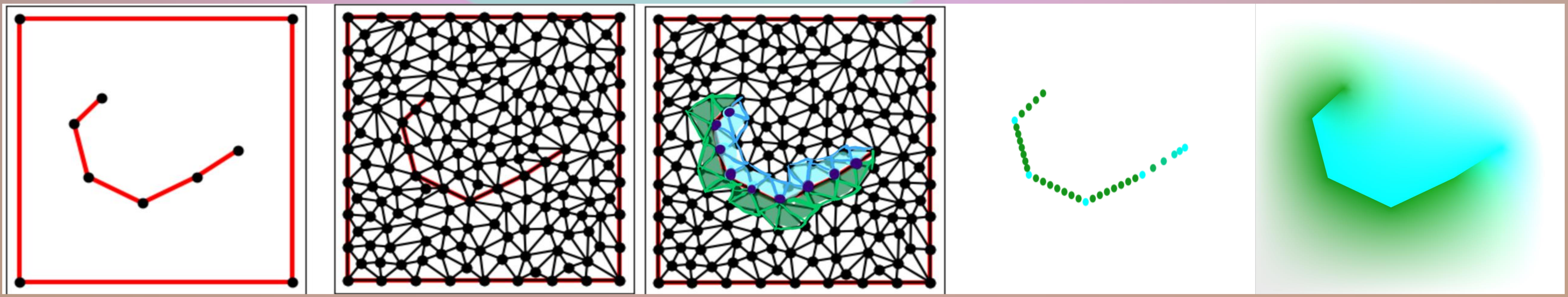


Fig. 5: image data from Diffusion Curves paper

Image Generation Pipeline -- (Every image on this poster was generated using this method)

Fig. 3 (A, B, C, D, E)



Step 1: Plot a series of points to create an image and connect them together in segments (done manually by plotting X, Y points numerically and then connecting them together

Step 2: Use a triangulation algorithm to generate a series of other points with a specified density to create a mesh of triangles

Step 3: If dual sides curves are needed, manually search through every vertex, determine which ones are on the line created in step 1, and remap them all to a new virtual line

Step 4: Run the mesh through a custom-coded algorithm that assigns colour values to every point on any line that I defined in step 1

Step 5: Feed the mesh with the colour values and desired specifications into PolyFEM, a repository created by my instructor (Teseo Schneider) that handles the calculus involved in calculating colour values relative to the input for each triangle in the mesh.