

Proposal for CSE528 Computer Graphics Course Project

Implementation of “*Painting with Polygons: A Procedural Watercolor Engine*”

Khan Mostafa
109365509

This project is focusing on implementing a procedural watercolor engine, presented by *S. DiVerdi, A. Krishnaswamy, R. Měch, and D. Ito* on their paper, “*Painting with Polygons: A Procedural Watercolor Engine*”. The goal is to create a range of watercolor like dynamic behavior executable on lower end hardware in real-time.

The algorithm adopts a sparse representation to model watercolor strokes and uses random walk to update position in each time step. Paint pigments are represented as “splat” particles – each being a complex polygon of n vertices. The algorithm has four major operations.

At *first*, stroke input is recorded by placing stamps in uniform length increment along the stroke path. Stamps are set of splats which store age, flow and several other attributes. At each stamp, water are also placed on canvas and stored separately as rasterized 2D grid of cells called water-map.

Secondly, splats are advected at each time step embodying biased random walk behavior. Vertex’s own velocity, splat’s bias velocity, water velocity, paint viscosity, roughness of canvas media and gravity are modeled at this point. These helps to achieve branching and roughening aspects of watercolor.

Over time, different advection directions may create unrealistically straight hard edges due to local undersampling. This artifact is dealt by the *third* operation, which is to periodically resample boundary of each splat.

Fourth operations is lifetime management. Here, a splat has three stages of life: flowing, fixed and dried. Initially a splat is flowing; after certain steps (age) they are fixed – but can potentially be rewetted to resume advection. After a certain period the splat is dried and then rasterized into dry pigment buffer and removed from the simulation. Rewetting helps to achieve effects like back runs and feathered edges. Lifetime management also mimics water flow, granulation and other watercolor affects.

The paper represents a way to implement brush types to reproduce a variety of watercolor characteristics. It demonstrates five brush types.

To summarize, all four operations of the algorithm and brush types achieves common watercolor effects, such as, color blending, feathering, edge darkening, non-uniform pigment density, granulation, back runs, rewetting, glazing etc.

The project aims at implementing an overall 70% of what this paper represents. To do so, following ten weeks are scheduled as such,

Week	Dates	Task
1	09/23~09/29	Thorough reading and understanding of the paper
2,3	09/30~10/13	Implementation of <i>Paint Initialization</i> and basic user interface
4,5	10/14~10/27	Implementation of <i>Pigment Advection</i>
6	10/28~11/03	Preparation for mid-term demo, partial implementation of <i>Sampling Management</i>
7	11/04	mid-term system demo
	11/5~11/10	Implementation of remaining part of <i>Sampling Management</i>
8,9	11/11~11/24	Implementation of <i>Lifetime Management</i>
10	11/25~12/01	User interface enhancement
11	12/02~12/05	Submission preparation: report + software + presentation slide
	12/06	Submission due

I will start working for one brush type (either simple or wet-on-dry) and will try to add more brush types if possible.