

Interactive Watercolor Painting Simulation

Yiwei Tu

A Capstone Project Proposal

submitted in partial fulfillment of the

requirements of the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

Date

Project Committee:

Kelvin Sung, Committee Chair

Michael Stiber, Committee Member

Yusuf Pisan, Committee Member

Introduction

Watercolor is one of the well-known styles of art painting. It involves mixing a water-based solution before applying the pigments to the canvas. The liquid mixture of the pigmented water-based solution is dispersed and spread on the canvas. Afterwards, the excess water solution is absorbed and evaporated, leaving only traces of pigment on the canvas. It is the combination of these traces of pigment that makes up a watercolor painting. These traces are translucent patterns of pigment, formed by the flow of the pigment-water solution along the grooves of the watercolor canvas. These traces cannot be completely controlled or manipulated, which is the beauty of watercolor. Due to this randomness, it is difficult for artists to reproduce the same watercolor effect, which makes each watercolor painting unique. In addition, the flowing motion of the pigment-watercolor is completely different on dry and wet canvases. Figure 1 shows the watercolor effect of wet in dry (wet brush drawing on dry canvas) and wet in wet (wet brush drawing on wet canvas). Many people, including myself, are fascinated by the characteristics of these watercolors. I am particularly interested in studying and reproducing the mechanisms behind the use of computer graphics techniques to create watercolor paintings.

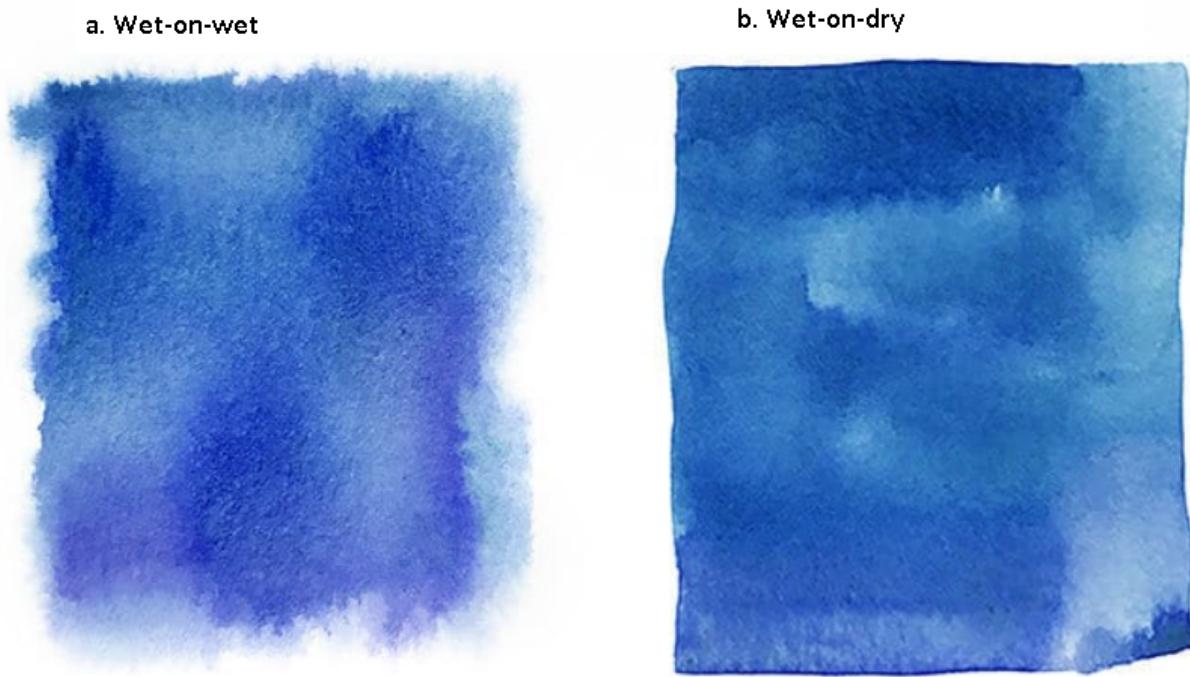


Figure 1. Wet brush with pigment on canvas of different wetness. (a) wet-on-wet (wet brush with pigment drawing on wet canvas); (b) wet-on-dry (wet brush with pigment drawing on dry canvas).

Current research on watercolor simulation is concentrated in two fields: watercolor stylization and digital watercolor painting. Watercolor stylization involves converting the input image to a watercolor style and rendering the anime scene in a watercolor style [1]–[4]. Figure 2 illustrates the watercolor stylization results from Curtis *et al.* [3] and Bousseau *et al.* [4]. While this stylization provides users with the ability to convert from regular images to watercolor, it lacks the creativity of user control. I, however, am interested in the second research area of

watercolor simulation, digital watercolor painting application. It offers artists and watercolor enthusiasts the convenience of being able to create watercolor paintings anytime, anywhere.



Figure 2. Comparison of results from Curtis *et al.* [3] and Bousseau *et al.* [4]. (a) original image, (b) Curtis *et al.* result, (c, d) Bousseau *et al.* result

In the history of the development of painting, from painting on cave walls to colored pottery, on silk, and on paper, the material media of painting has always evolved along with the development of time. With the development of technology and the advent of the digital and graphic age, digital painting has become a new art form derived from the fusion of modern computer technology and the art of painting [5]. In the field of digital painting, paper models are designed and constructed as an alternative to painting bearers, and pressure sensitive board and pressure sensitive pen have replaced painting tools.

Compared to traditional watercolor painting with paints, water solutions and brushes on watercolor paper, digital watercolor, painting on digital devices such as computers and tablets, gives the artist many benefits that traditional painting cannot offer. Based on computer graphics algorithms all required tools are integrated and accessible via a single machine, such as a tablet or smartphone, thus saving the space taken up by painting tools. Because of the benefits of the digital platform, painters are not afraid of damaging or soiling paper or canvas, or of not being able to replicate previous brushstrokes, as they can easily delete or copy them from their operational history [5], [6]. In addition, with the help of the Internet, completed electronic paintings can be easily and quickly disseminated, shared, and brought to public attention, which may help artists attain recognition. Digital watercolors also benefit from these advantages.

Digital watercolor simulation consists of two main tasks: building a physical model of the watercolor painting and rendering the image based on the information from the physical model. Since conventional watercolor painting is done on canvas, where all physical phenomena, including diffusion, absorption, and evaporation of the pigment-water solution, occur, a major approach to simulating watercolor effects is to model the physical properties of watercolor and render the final effect image on canvas. Small *et al.* introduced a basic model to simulate a watercolor canvas using cellular automata [7]. In their work, they divided the simulation of watercolor painting into three parts. First, pigments and water are applied to the paper in different ways. The properties of the paper, as well as environmental factors such as humidity and gravity, are taken into account. Second, the mobility of the pigment and water in discrete time steps to various pressures is calculated. Finally, depending on the simulated state of the discrete time, images can be produced in various ways. In order to reproduce watercolors effectively, not only the physical characteristics of the medium but also the phenomenological characteristics that make watercolors so popular among artists must be studied, approximated, simulated.

Compared to studies on watercolor stylization, there are not many studies on digital watercolor simulation, but there is some basis for inquiry. Small *et al.* proposed to use Cellular Automata to simulate the watercolor painting canvas and its effects [7]. Curtis *et al.* improved Small's model with a more sophisticated three-layer canvas model, more complex shallow water simulation and Kubelka-Munk model to calculate the resulting color for every pixel [3]. Van Laerhoven and Van Reeth made a trade-off between "real-time" and the simulation complexity, and employed a semi-Lagrangian method for faster simulation [6]. Chu and Tai present a physically-based method for simulating ink dispersion in absorbent, using the Lattice Boltzmann equations to simulate ink dispersion [8]. Oh *et al.* proposed a new watercolor painting system that may be used on low-power computing devices like tablets [9]. DiVerdi *et al.* [10] introduced a watercolor simulation system with a particle-based model for pigment flow and vector-based strokes representation for rendering at arbitrary resolution.

The goals of this project are to study various existing solutions for watercolor simulation, and apply appropriate ones to create a real-time digital watercolor painting application. The application would simulate watercolor effects in real time based on user's input from canvas, brush and stroke, and at the same time, mimic real watercolor creation processes, such as a glass of water and a towel to adjust the wetness of brush. In addition to simulating watercolor painting on regular rectangular canvas, as an aspiration goal, this application may simulate watery painting on non-regular three-dimensional objects, such as a sphere, with watercolor canvas as surfaces.

Related Work

Small *et al.* published the first paper related to the simulation of watercolor painting in 1991. They proposed a way to simulate watercolor effects using a system of cellular automata on the Connecting Machine (Small 1991). In this model, each cell needs to know only itself and its immediate environment. Small *et al.* suggest that future work could be to define new kinds of paper that have extremely specific properties not available in nature. However, the results generated from this model still lack realism.

Based on Small's watercolor simulation model, Curtis *et al.* [3] improved the cellular automation to simulate the fluid flow and pigment dispersion of watercolor. This improved method adopts a more sophisticated three-layer paper mode, a more complex shallow water simulation, and a more faithful rendering and optical compositing of pigmented layers based on the Kubelka-Munk model [3]. Figure 3 presents the simulated watercolor effects created by Curtis *et al.*. Because their solution is resolution-dependent, generalizing to a resolution-independent model for watercolor simulation is an important goal for future work. They achieved impressions of real watercolor behavior, but due to the complexity of the simulation model, they were unable to achieve a fully interactive system.

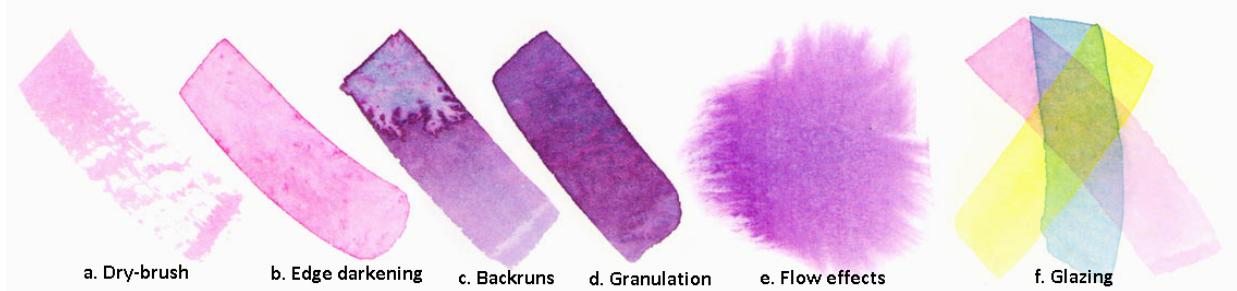


Figure 3. Simulated watercolor effects created by Curtis *et al.* system.

All the above methods to simulate fluid dynamics used the Navier-Stokes equation, which describes the behavior of fluid flow in the continuum approximation. Van Laerhoven and Van Reeth [6], however, utilized the work of Stam to describe a number of fast and stable procedures to simulate fluid flow using implicit solvers [11], [12]. In addition to the different fluid simulation solver, they further improved the three-layer canvas model for their real-time watercolor simulation. Their canvas consists of three active layers, similar to Curtis *et al.* model of canvas, and an unlimited number of passive layers, which contain previously drawn strokes that have dried and no longer participate in the simulation, except when the canvas is rendered. Although their simulation model is able to produce 22 frames per second, it has not been demonstrated that the algorithm can handle high resolutions on commonly available hardware since the system requires high computational loads and resources.

Another way to simulate the pigment-water solution fluidity is the Lattice Boltzmann equation, which is a more detailed description of the behavior of a gas. The following two watercolor simulation works use the Lattice Boltzmann equation to simulate the fluid movement.

Chu and Tai present a physically-based method for simulating ink dispersion in absorbent paper for art creation purposes [8]. In addition to the usage of the Lattice Boltzmann equation for ink flow simulation, they utilized both CPU and GPU for these computational-intensive simulations, with the overall system frame rate of 44 frames per second. Figure 4 depicts a comparison of real ink effects (top) versus digital ink effects created by Chu and Tai's system. Since they limited their dispersion simulation to three pigments at a time with RGBA texture, one of the improvements suggested is to employ the Kubelka-Munk model for simulating optical blending.

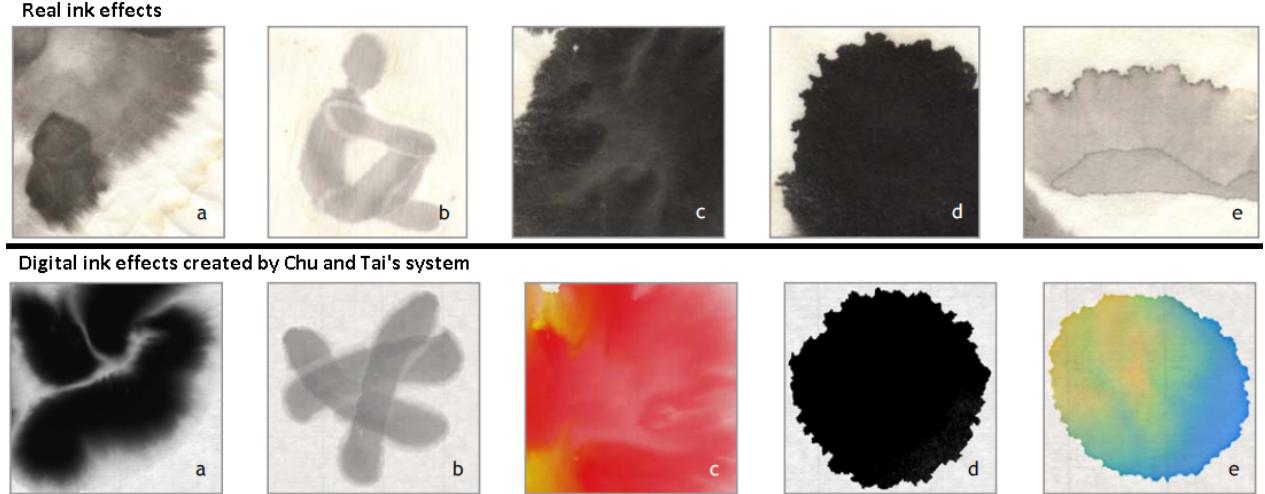


Figure 4. A comparison of real ink effects (top) versus digital ink effects created by Chu and Tai's system. (a) Feathery pattern.
(b) Light fringes. (c) Branching pattern. (d) Boundary roughening. (e) Boundary darkening.

Oh *et al.* presents a novel watercolor painting drawing system which can work even on low powered computing machines, such as tablet PCs [9]. Similar to Chu and Tai's system, their system solves the Navier-Stokes equations for fluid flow in the Lattice Boltzmann method (LBM). In addition, their system deals only with pigmented regions and not the entire drawing canvas, which accelerates the simulation calculation. Figure 5 exhibits two watercolor paintings generated by Oh *et al.* system. However, due to the usage of LBM for fluid flow simulation, detailed fluid behavior is not simulated in the system, which makes the simulation not realistic as real watercolor painting.



Figure 5. Watercolor painting by Oh *et al.* system

DiVerdi *et al.* [10] presents a procedural vector based algorithm for generating watercolor-like dynamic paint behavior in a lightweight manner. Their formulation uses a particle-based model of pigment flow, rather than grid based, which allows for rendering at arbitrary resolutions and is fast to calculate. As the authors mentioned in the paper, the method is not sufficient to describe realistic watercolor painting. Figure 6 displays the comparison of similar strokes made with real watercolor paints (left) versus their algorithm (right). Moreover, it

is hard to guarantee that the method can also show interactive performance on recent slate devices with high resolution.

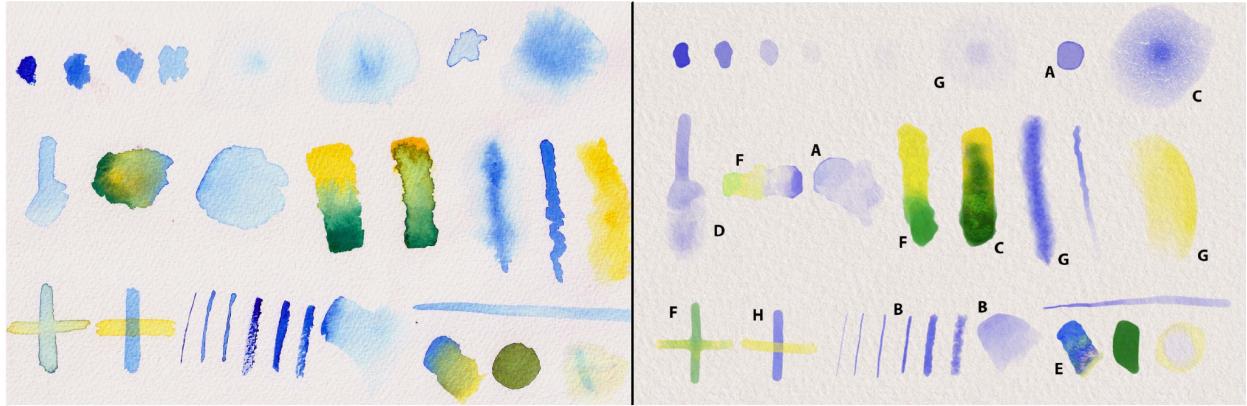


Figure 6. A comparison of similar strokes made with real watercolor paint (left) versus DiVerdi *et al.* algorithm (right). Paper texture is added to our results for comparison purposes. The strokes are chosen to showcase a variety of characteristic watercolor behaviors, including edge darkening (A), nonuniform pigment density (B), granulation (C), rewetting (D), back runs (E), color blending (F), feathering (G), and glazing (H). Strokes exemplifying particular effects have been labeled with the corresponding letter.

Scope

The goals of this project are to further study existing watercolor painting simulation solutions, evaluate and adopt the most appropriate to reproduce a digital watercolor painting application with interactive manipulation, and analyze the results according to defined assessment criteria. The application should, in real-time, render the basic watercolor effects, including edge darkening, wet-on-dry, color glazing, based on users' inputs from stroke and brush. The application will be lightweight, support portable tablet devices, and the simulation results will be presented and tested based on user input. This application may support painting of watercolors on non-flat three-dimensional objects, such as cube, and sphere, that have a watercolor canvas as their surface in addition to painting watercolors on a general canvas.

Measure of Success

The success of this project will be assessed based on the following criterias:

1. Characteristic: The project should simulate the basic watercolor effects, including wet-on-dry, wet-on-wet, edge darkening, and color glazing.
2. Quality of Rendering: simulation of watercolor effects by the project should look real.
3. Performance: simulation of watercolor effects by the project should be in real time.

Schedule

The entire project is divided into the following milestones:

1. **Background:** Study various ways of watercolor simulation in detail.
2. **Tech investigation:** Learn Unity game engine for physical properties simulation, and learn shaders which can handle rendering.
3. **Solution + Implementation:** Adopt appropriate ones to create an application to reproduce watercolor effects based on users input.
4. **Optimization + refinement:** Evaluate the application and complete the report

The timeline for the milestones:

- Milestone 1 & 2: Autumn 2022
- Milestone 3: Winter 2023
- Milestone 4: Spring 2023

Milestones: weeks	Tasks	Deliverables
Background: 2	Detailed studying of [3], [7]	<ul style="list-style-type: none"> - Summary of solution - Structure of implementation
Background: 2	Detailed studying of [6], [8]	<ul style="list-style-type: none"> - Summary of solution - Structure of implementation
Background: 2	Detailed studying of [9], [10]	<ul style="list-style-type: none"> - Summary of solution - Structure of implementation
Background: 1	Comparison and conclusion of background	Implementation plan, detailed technology requirements
Tech: 1	Evaluate existing systems	Recommendations on which tech to choose
Tech: 2	Learn about shaders	
Implementation: 10	Implement a watercolor painting application	Unity executable (application)
Evaluation: 5	Evaluate and refine the application, begin report writing	<p>Evaluation report, including the following sections</p> <ul style="list-style-type: none"> - Measurements with definition - Diagrams and tables - Analysis and explanation
Reporting: 4	Complete the report	Final report for the Capstone project

Reference

- [1] S. E. Montesdeoca *et al.*, “Edge- and substrate-based effects for watercolor stylization,” in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering - NPAR '17*, Los Angeles, California, 2017, pp. 1–10. doi: 10.1145/3092919.3092928.

- [2] S. E. Montesdeoca, H. S. Seah, H.-M. Rall, and D. Benvenuti, “Art-Directed Watercolor Stylization of 3D Animations in Real-Time,” *Comput. Graph.*, vol. 65, pp. 60–72.
- [3] C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin, “Computer-generated watercolor,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, Not Known, 1997, pp. 421–430. doi: 10.1145/258734.258896.
- [4] A. Bousseau, M. Kaplan, J. Thollot, and F. X. Sillion, “Interactive watercolor rendering with temporal coherence and abstraction,” in *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering - NPAR '06*, Annecy, France, 2006, p. 141. doi: 10.1145/1124728.1124751.
- [5] S. Li, “Visual form beyond traditional media—On the expression and construction of watercolor art in digital painting,” *E3S Web Conf.*, vol. 189, p. 03013, 2020, doi: 10.1051/e3sconf/202018903013.
- [6] T. Van Laerhoven and F. Van Reeth, “Real-time simulation of watery paint,” *Comput. Animat. Virtual Worlds*, vol. 16, no. 3–4, pp. 429–439, Jul. 2005, doi: 10.1002/cav.95.
- [7] D. Small, “Simulating watercolor by modeling diffusion, pigment, and paper fibers,” San Jose, CA, Aug. 1991, pp. 140–146. doi: 10.1117/12.44417.
- [8] N. S.-H. Chu and C.-L. Tai, “MoXi: real-time ink dispersion in absorbent paper,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 504–511, Jul. 2005, doi: 10.1145/1073204.1073221.
- [9] Oh, Junkyu and Park, Jinho, “Efficient Watercolor Painting on Mobile Devices,” *Int. J. Contents*, vol. 8, no. 4, pp. 36–41, Dec. 2012, doi: 10.5392/IJOC.2012.8.4.036.
- [10] S. DiVerdi, A. Krishnaswamy, R. Mech, and D. Ito, “Painting with Polygons: A Procedural Watercolor Engine,” *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 5, pp. 723–735, May 2013, doi: 10.1109/TVCG.2012.295.
- [11] J. Stam, “Stable fluids,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, Not Known, 1999, pp. 121–128. doi: 10.1145/311535.311548.
- [12] J. Stam, “Real-time fluid dynamics for games,” Mar. 2003.