12-2015

Painting with Turbulence

Chuqiao Wang
Clemson University, chuqiaw@g.clemson.edu

Painting with Turbulence

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Fine Arts
Digital Production Arts

by
Chuqiao Wang
December 2015

Accepted by:
Dr. Robert Geist, Committee Chair
Sam Wang
Dr. Jerry Tessendorf

Abstract

Inspired by a study that identified a strong similarity between Vincent van Gogh's and Jackson Pollock's painting techniques, this thesis explores the interplay between science and art, specifically the unpredictable behaviors in turbulent flows and aesthetic concepts in painting. It utilizes data from a GPU-based air flow simulation, and presents a framework for artists to visualize the chaotic property changes in turbulent flows and create paintings with turbulence data. While the creation of individual brushstrokes is procedural and driven by simulation, artists are able to exercise their aesthetic judgments at various stages during a painting creation. A short animation demonstrates the potential results from this framework.

Acknowledgments

I would like to express my gratitude to my thesis advisor Dr. Robert Geist for his support and inspiration through my thesis research. This thesis would not be possible without his guidance and his knowledge on GPU computing and flow simulation. My sincere thanks go to my committee member Professor Sam Wang, for his artistic insights and inspirations on the project and for overcoming the technical difficulties to serve on my committee. I would also like to thank my committee member Dr. Jerry Tessendorf for his knowledge on computer graphics and help on fixing rendering problems.

I would like to offer special thanks to Dr. Joshua Levine for his expertise on data visualization and suggestions on translating visualization concepts into artistic tools. Also, I would like to thank Dr. Andrea Feeser for her feedback on writing the background section of this thesis.

Finally, I would like to thank my parents and friends for their continuous love and support.

Table of Contents

List of Figures

Chapter 1

Introduction

This thesis presents a method that utilizes simulation data to create art-directable paintings. By employing high-level controls over simulation, artists are able to generate brushstrokes varying in shape, color, and movement. A short animation is included to demonstrate the creation of a collection of brushstrokes that form a painting.

The inspiration for this work was initially drawn from a recent study that identified many similarities between Vicent van Gogh's and Jackson Pollock's painting techniques based on models from computer vision and image processing. While both artists, whether aware of such or not, exhibited a scientific method of painting, this thesis seeks to present a workflow of art creation under influence of scientific simulation, specifically, the process of painting with turbulence. The appearance of turbulence in the works of van Gogh and Pollock was a motivating factor.

The two-part simulation includes an air flow simulation that runs on a Graphics Processing Unit (GPU) and a particle system simulation implemented in commercial software Houdini. The data from the first simulation, which carries information on physical properties of turbulent flow is passed to the second simulation and then incorporated with artistic controls to generate brushstrokes.

Chapter 2

Background

"When I meet God, I am going to ask him two questions. Why relativity? And why turbulence? I really believe he will have an answer for the first."

– Werner Heisenberg

The turbulent motion of fluid flows has always existed as part of the awe-inspiring nature and as a form of art. More than five hundred years ago, Renaissance artist and engineer Leonardo da Vinci identified turbulence as a distinct natural phenomenon and used the term *turbolenza* for the first time [20]. His observation and perception of turbulence was recorded in his sketches, two of which are shown in figure 2.1, along with a description:

> "...the smallest eddies are almost numberless, and large things are rotated only by large eddies and not by small ones, and small things are turned by small eddies and large." [6]

Turbulence in fact encompasses occurrences of movements in a much larger range of scale, "from the interior of biological cells, to circulatory and respiratory systems of living creatures, to countless technological devices and household appliances of modern society, to geophysical and astrophysical phenomena" [8]. The omnipresence of turbulence has made it a subject of study by many of the greatest physicists and engineers. Most of the investigations and developments in turbulence research started in the nineteenth century. In 1822, French physicist and engineer Claude-Louis Navier introduced a set of equations that describe the motion of viscous fluid substances; George Gabriel Stokes later extended the equations to what we currently know as the famous Navier-Stokes equations, which are believed to fully describe turbulent flows [7].

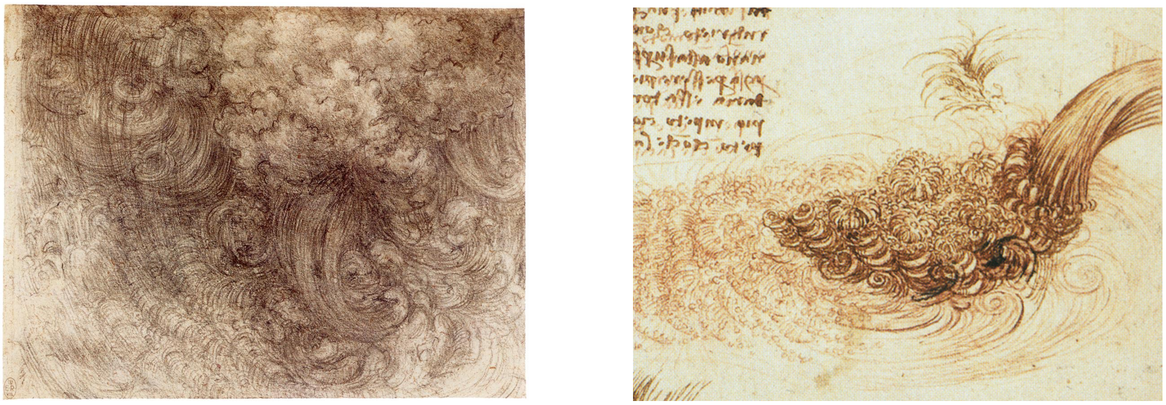


Figure 2.1: Leonardo da Vinci, sketches of turbulence

At the end of the nineteenth century, Osborne Reynolds conducted a series of experiments on pipe flow and introduced the concept of a dimensionless number, the Reynolds number, which helps to characterize different flow regimes, such as laminar flow and turbulent flow, even when they occur in very different settings [19]. Through the studies, we are able to conclude some characteristics that turbulent flows exhibit, such as irregularity, chaotic, and seemingly random behaviors, diffusion, dissipation, three-dimensionality and time dependence. Nevertheless, the Navier-Stokes equations cannot be solved analytically except for a few simple cases. We must resort to numerical solutions in efforts to predict flow behaviors, but such computations can easily become unstable, especially at high Reynolds numbers, while turbulence occurs. Predicting flow behaviors remains a fundamental challenge for engineers, who seek out the engineering solutions to real world problems, such as the construction of buildings and aircrafts, and weather forecasting.

While the complex nature and intellectual challenge of modeling turbulence continuously drives the science and engineering community, it also enthralls the artists with its sheer beauty and mysterious form. There has never been a lack of regard for the beauty and splendor of turbulence in the field of fine arts. Since Leonardo da Vinci, many artists have attempted to bring their visions of natural flows onto canvas: the seventeenth century Japanese ukiyo-e artist Katsushika Hokusai depicted enormous, powerful ocean waves through a series of prints; Chinese painter Tao Shi from the same period drew an analogy between water waves and mountain ranges in his studies of turbulent forms [3]. Their depictions of turbulence are shown in figures 2.2 and 2.3.

Nevertheless, there is one name in the history of art with which we particularly associate turbulence – Vincent van Gogh. In 1890, the post-Impressionist painter died after suffering from self-

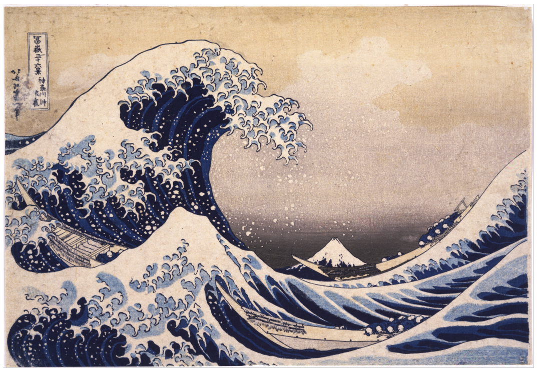


Figure 2.2: Katsushika Hokusai, The Great Wave Off Kanagawa, 1830-1833

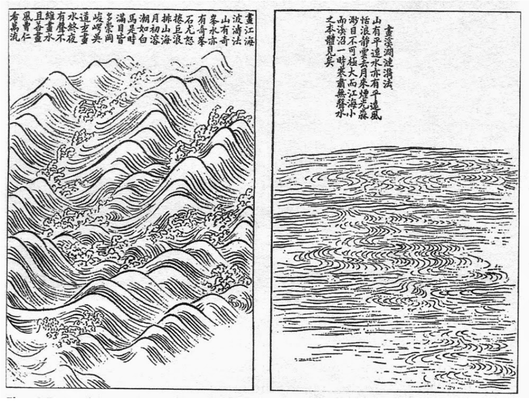


Figure 2.3: Tao Shi, included in "The Mustard Seed Garden Manual of Painting" by Gai Wang, 17th c

inflicted gunshot for two days, leaving the world with an astonishing number of invaluable artworks, which have been extensively studied by art historians and scientists. The years before his death, van Gogh went through a period of intense suffering from mental illness, leading him to admit himself into the Saint-Paul asylum in Saint-Remy [17]. Nevertheless, this period saw him at the height of his artistic capabilities and creative power. From his room at the asylum, he produced a series of paintings, including "The Starry Night", which was probably his most well-known painting. It depicts the view from his bedroom window just before sunrise, with the glorious moon, the swirling winds, the ebbs and flows of glimmering stars, and the idealized village mixed in a mysterious motion [17].

More than a hundred years later, with the NASA Hubble Space Telescope, scientists were able to see "the eddies of distant clouds of dust and gas around a star", shown in figure 2.5, which reminded them of van Gogh's paintings and inspired them to conduct further studies of the paintings [17]. They discovered that the patterns of fluid flow structures presented in "The Starry Night", which was painted during a period of intense suffering from mental illness, closely resemble the structure and characteristics of turbulent flows in nature. Although it is uncertain whether van Gogh's turbulent mental state somehow allowed him to perceive and represent "one of the most supremely difficult concepts nature has ever brought before mankind" [17], an accurate depiction of a scientific concept exhibited by the artist is evident, and it could expectedly shed light on the qualitative and descriptive side of turbulence research for the science community.

Moreover, recent studies applying the Wndchrm method, which uses a large set of numerical

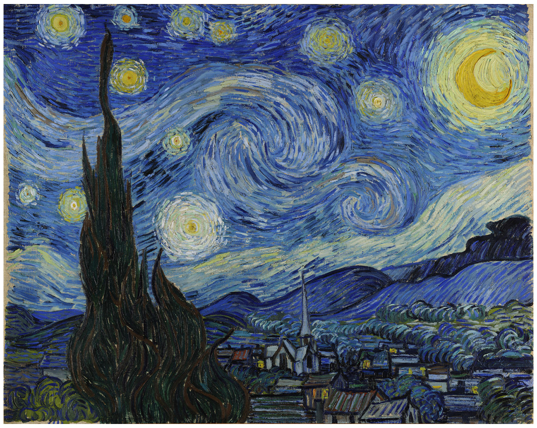


Figure 2.4: Vincent van Gogh, The Starry Night, 1890

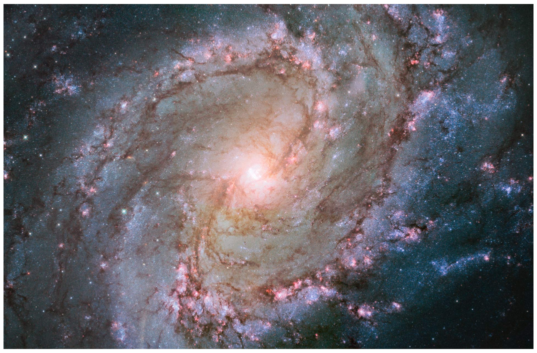


Figure 2.5: The scatterings of bright stars and thick dust that make up the barred spiral galaxy Messier 83, captured by NASA Hubble Space Telescope

content descriptors, have shown that there is a strong similarity between Vincent van Gogh's and Jackson Pollock's painting techniques [15]. Jackson Pollock (1912-1956) was an influential American painter and a leading figure in the post-war movement of Abstract Expressionism. He was often identified with his unique painting techniques, which have been recognized as a "crucial advancement in the evolution of modern art" [18]. His works were gotten done by unrolling a large canvas on the ground and allowing constant streams of paint to drip onto the canvas and splash on it. Instead of standing in front of an easel, Pollock would move around or stand on the canvas. Thus, even though the finished works are presented on a flat canvas, he introduced a third dimension when it comes to the process of making art. The canvas becomes "an arena in which to act" [14], and because of this, the painter is no longer in contact with the canvas, but through air.

Pollock also adopted ideas from previous art movements such as surrealism and cubism, featuring elements like chance and abstraction in his works. His action painting demonstrated an interplay of physical movements and emotional expressions. Furthermore, as spontaneous as the brushstrokes and Pollock's movements may seem, they are in complete control just as he once argued "I can control the flow of paint. There is no accident" [10]. His statement was proved by science. Computer analysis indicates that his paintings are composed of a distinct fractal pattern. The beauty and subtle order underneath the seemingly random gestures and paint flows of Pollock echo those patterns found in nature, such as in trees, clouds, mountains and coastlines [18]. Further comparison even concludes that "the fractal dimensions increased steadily through the years", meaning that he refined his techniques over years [18].

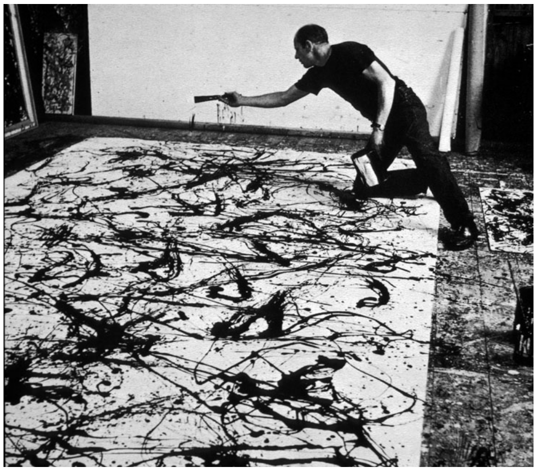


Figure 2.6: Pollock in action

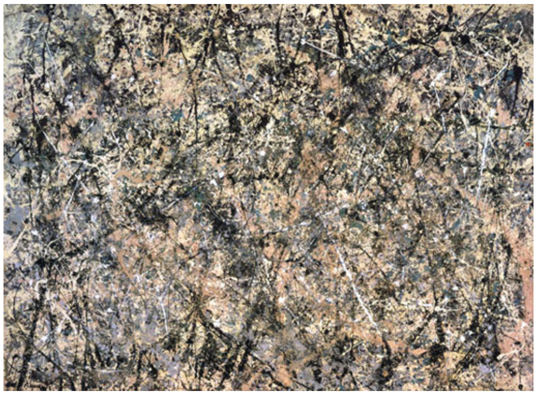


Figure 2.7: Jackson Pollock, Number 1 (Lavender Mist), 1950

Pollock led a dramatic life just as van Gogh did. He struggled with alcoholism for most of his life and was killed in a single-car accident while driving under the influence. The swirling lines and drips of paint unfolded the struggle with emotional turmoil of two great artists, as well as their fight to control the chaos and freely express emotions and their passion for life.

The main idea behind this thesis is to explore the interplay between science and art, how science contributes to a deeper understanding of art and seeks inspiration from art, and how art unravels scientific concepts. Specifically it provides artists with a framework to examine the natural gesture of fluid flows and the balance between chaotic property changes in turbulent flows and artistic control.

Chapter 3

Implementation

The implementation of this thesis consists of three stages: gathering data from GPU-based air flow simulation, creating particle system simulations in the commercial software package Houdini, and creating an art-directable framework for composing paintings that naturally carry the energy of turbulent flow.

3.1 GPU-Based Air Flow Simulation

This thesis utilizes data generated from a GPU-based air flow simulation. The simulation code used in this study was designed by Brenden Roberts, Joshua Levine and Robert Geist from the School of Computing at Clemson University. It simulates high Reynolds number flows across arbitrary geometries, and it allows output of position and velocity information.

This design employs Lattice-Boltzmann(LB) methods, alternatives to traditional Computational Fluid Dynamics(CFD) methods, which have provided success in modeling fluid flows [13].

A one-dimensional example using the LB method is modeling heat flow in a rod. Let ρ(x,t) denote the energy in a rod at site x at time t. Let f_i(x,t) denote the energy at site x, time t flowing in direction i=±1. Assume lattice spacing λ, and time step τ. Also assume that, due to molecular collisions at each time step at each site, some fraction, σ, of the energy continues flowing in the same direction, and fraction 1-σ in reverse direction. The fundamental update at any site x is thus:

$$f_i(x + i\lambda, t + \tau) = \Omega_i \cdot f_i(x, t) \tag{3.1}$$

where

$$\Omega = \begin{pmatrix} \sigma - 1 & 1 - \sigma \\ 1 - \sigma & \sigma - 1 \end{pmatrix}$$

Using a Taylor expansion of (3.1), one can show that as $(\lambda, \tau) \to 0$, the limit of (3.1) is then

$$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial x^2}$$

which is the one-dimensional diffusion equation, with diffusion coefficient

$$D = (\frac{\lambda^2}{\tau})(\frac{\sigma}{2 - 2\sigma})[13]$$

The Lattice-Boltzmann methods can be extended to solve differential equations in higher dimensions, in particular, the Navier-Stokes equations which models fluid flows:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \nu \nabla^2 \vec{u}$$

where $\vec{u}$ describes fluid velocity, p the pressure, ρ the density, ν the viscosity, and $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ is the gradient operator.

Instead of considering only two directions, as in the one-dimensional case, the most common model for three-dimensional cases is the so-called D3Q27 model, which uses 27 directions, consisting of the vectors from a lattice point to each neighboring lattice point, including itself, in a cube of unit radius λ [13]. Thus, a 27x27 matrix is needed for the fundamental update.

With simplification introduced by Chen et al. and Qian et al., $\Omega_i \cdot f(\vec{r}, t)$ can be replaced with $-\omega \cdot (f_i(\vec{r}, t) - f_i^{\text{eq}}(\vec{r}, t))$, where $f_i^{\text{eq}}(\vec{r}, t) = \rho \cdot w_i \cdot (1 + 3\vec{c_i} \cdot \vec{u} + \frac{9}{2}(\vec{c_i} \cdot \vec{u})^2 + \frac{3}{2}\vec{u} \cdot \vec{u})$, where ω is a relaxation rate, $\vec{c_i}$ is direction i, and w_i is a link weight that is based on link length 0, 1, $\sqrt{2}$, or $\sqrt{3}$. f_i^{eq} is called Maxwell-Boltzmann equilibrium [5][11]. Therefore, the fundamental update becomes:

$$f_i(\vec{r} + \lambda \vec{c_i}, t + \tau) = f_i(\vec{r}, t) - \omega \cdot (f_i(\vec{r}, t) - f_i^{\text{eq}}(\vec{r}, t))$$

In the GPU implementation, the fundamental update is performed synchronously across the nodes in a grid, with one computational thread per grid node. The update is split into two steps, a

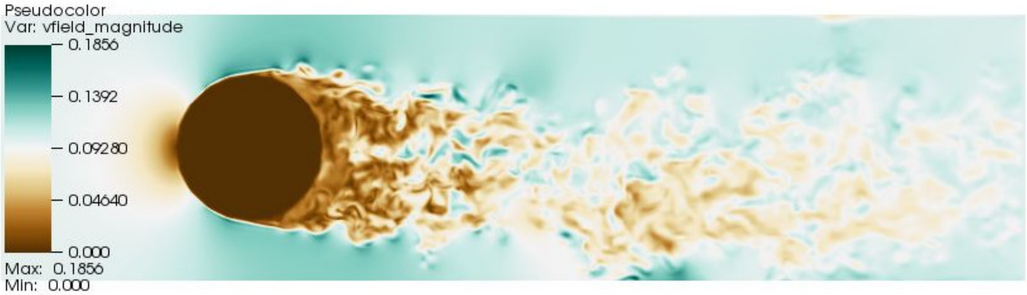


Figure 3.1: Visualization slice of air flow over a sphere

cascade or collision step

$$\widetilde{f}_i(\vec{r}, t) = f_i(\vec{r}, t) - \omega \cdot (f_i(\vec{r}, t) - f_i^{\text{eq}}(\vec{r}, t))$$

and a streaming step,

$$f_i(\vec{r} + \lambda\vec{c_i}, t + \tau) = \widetilde{f}_i(\vec{r}, t)$$

so that the first step is completely local, and so that optional collision steps of fluid with hard surface boundaries may be inserted into the update between cascade and stream [13].

Changes in surface area and fluid properties can result in distinct flow behaviors. Thus, it is essential to choose a geometry, usually represented by a .obj file, that would generate “interesting” flow patterns. Visualization tools such as visIt can be used to quickly visualize a vector field, and so that effects of various geometries can be explored quickly. Figure 3.1 shows the visualization of a slice of the air flow over a sphere at time 10000(ms).

The flow data from a simulation of air flow over a sphere was captured between a starting and an ending timestamp at fixed time intervals. The GPU program outputs the velocity information $\vec{v}$=(v_x, v_y, y_z) at each lattice point $\vec{p}$=(p_x, p_y, p_z) as one line. Since the dimension of the grid was set to 704*192*192, the maximum size for the available hardware, the output file at one timestamp consisted of 704*192*192=25,952,256 lines. Therefore, over a chosen time range, a file sequence was generated with each file representing the simulation data at one timestamp. The naming convention was vfield.#, where # represents the time in milliseconds.

3.2 Particle System Simulation

Houdini was the primary tool used in this stage. Developed by Side Effects Inc., it is an animation and visual effects package that provides a procedural, node-based workflow for various stages of a production, such as animation and lighting. Because it is procedural, it allows a high level of flexibility and productivity, especially when dealing with dynamic simulations. One example is to create a mountain-like terrain. A simple procedure would be wiring a *Grid*, a *Mountain* and a *Color* node inside a *Geometry* node. Modification to the geometry can be done by adding, editing, by-passing or deleting nodes and any change will cascade through the network. For example, by-passing the *Mountain* node will direct data to flow from *Grid* directly to *Color*, thus resulting in a flat terrain. The example is demonstrated in Figure 3.2(a)-(d).

The particle system simulation in this section utilizes Houdini's various geometry nodes and a *DOP Network*, which stands for dynamic operator network. It contains a dynamic simulation and provides simulation-wide controls. The first step was importing and processing data from the air flow simulation. Digital assets and operator types in Houdini extend its level of flexibility by allowing users to build customized tools. Users are able to encapsulate their work in a node or run a code snippet to modify the incoming data, and more importantly, create artist-friendly interfaces [2]. A Python "Geometry Object Operator" was built to parse the incoming data. It allowed user-defined values for:

1. Velocity Scale(*vscale*), which defines the scaling factor for the original velocity

2. Offset(*offset*), which defines the offset in the file sequence

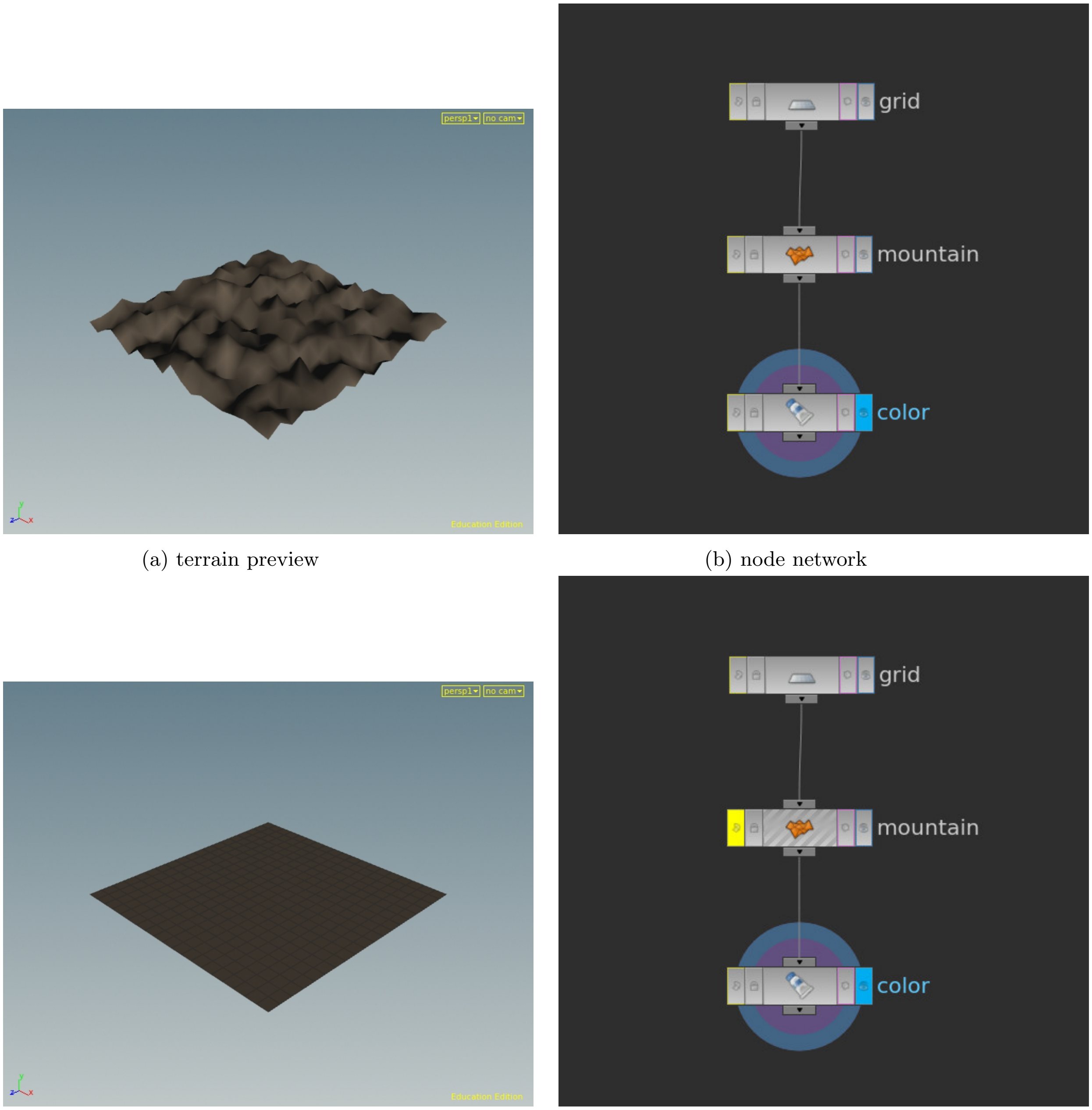
3. Starting Time(*stime*), which defines the starting time of the particle simulation

4. Interval(*interval*), which defines the time interval of the air flow simulation

5. Time Scale(*tscale*), which defines the time interval of the particle simulation

The operator was used to calculate the number in the file name based on the current frame number with the following equation:

$$number = int(\frac{frame}{tscale}) \cdot interval + stime + offset \cdot interval,$$



(a) terrain preview

(b) node network

(c) terrain preview with mountain node disabled

(d) node network with mountain node disabled

Figure 3.2: A terrain created by wiring three nodes

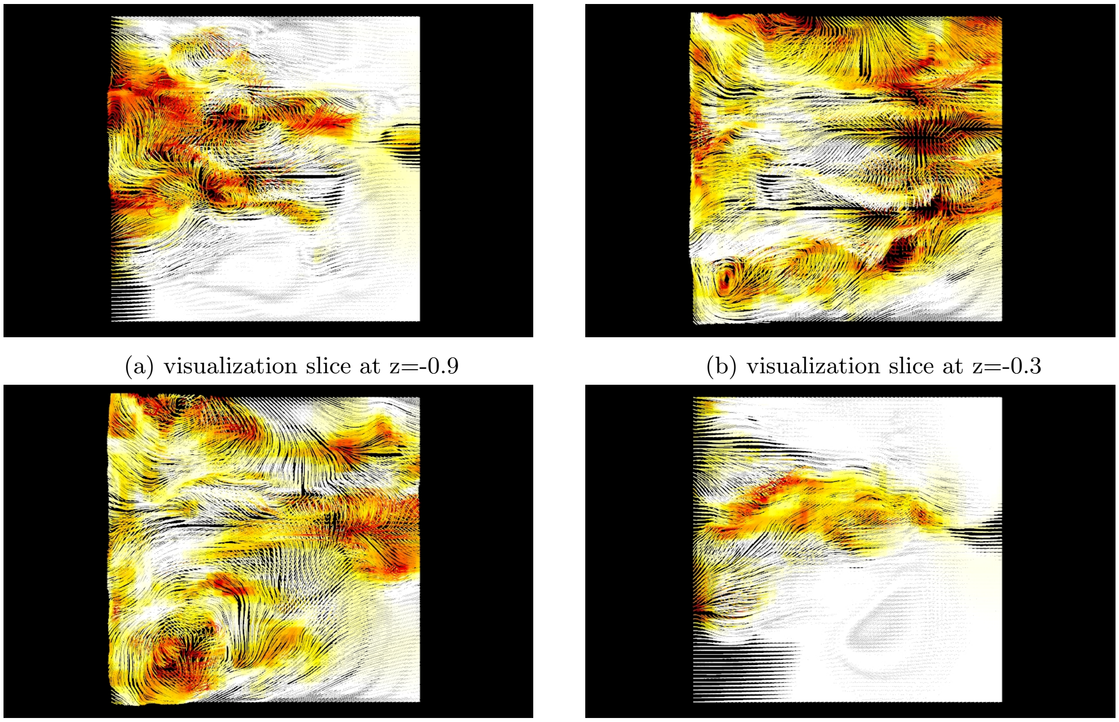
which led to the corresponding file in the file sequence from the air flow simulation. It then created a point cloud with velocity information(u_x, u_y, u_z) attached to each grid point(p_x, p_y, p_z). With the air flow simulation data imported, the next step was to convert the point cloud into a volume. A volume in Houdini is defined as a box composed of smaller boxes, called the "voxels". Each voxel can store a number, a vector, or a matrix. A *Volume From Attrib* node was created to transfer the velocity attribute to a volume. The resulting geometry is a set of three volumes that represent the x, y, and z components of the velocity field.

Before the velocity field data was incorporated into the particle simulation, visualization served as a key technique to validate and preview the data presented in the file sequence. Flow visualization is an important branch of scientific visualization; its goal is to "present the behavior of simulation data in a meaningful manner from which important flow features and characteristics can be easily identified and analyzed" [9]. Common approaches and techniques for visualizing flows of fluid and gas, specifically the velocity information associated with the flows, include *streamlines*, *pathlines*, and *streaklines*. In this thesis, streamlines were used to visualize the velocity field represented by the velocity volumes. Streamlines are lines that are tangent to the velocity vector field V, which can be defined as solutions to an ordinary differential equation [4]:

$$\frac{d\vec{p}(s)}{ds} \times V(\vec{p}(s)) = \vec{0},$$

where $\vec{p}(s)$ denotes a parametric representation of a streamline and $V(\vec{p})$ denotes the velocity at position $\vec{p}$ in velocity vector field V. The velocity field was evaluated and visualized as slices at positions z = -0.9, -0.3, 0.3 and 0.9. The results are shown in figure 3.3(a)-(d).

The next step was to create "brushstrokes" in a particle simulation with the given velocity field. In 1983, William T. Reeves first introduced the concept of particle system as a method for "modeling fuzzy objects such as fire, clouds and water" [12]. A particle system maintains particle states, controls particle creation and death, and governs system dynamics usually by means of numerical integration. The particles carry attributes such as position, velocity, mass, age, size, and color.



(a) visualization slice at z=-0.9

(b) visualization slice at z=-0.3

(c) visualization slice at z=0.3

(d) visualization slice at z=0.9

Figure 3.3: Visualization of the velocity field using streamlines

3.2.1 System Dynamics

Let V($\vec{p}$) denote the velocity at position $\vec{p}$ in a velocity vector field V, and Δt be the time step. The position of a particle after advection $\vec{p}_{\text{advected}}$ is then given by Euler integration:

$$\vec{p}_{\text{advected}} = \vec{p} + V(\vec{p}) \cdot \Delta t$$

System dynamics were maintained with a Houdini *DOP Network*, in which particles are advected by the volumes that represent the velocity field. The node *POP Advect by Volumes* advects particles with specified volumes. In this case, it took the emitter volume as source, generated particles from the source and advected particles with the velocity volume. It was then fed into a *POP Solver* along with a *POP Object*. Particles generated from the emitter were given a random initial velocity to add more variations to their motion. Some other controls over the simulation included particle birth rate, lifespan, life variance and velocity scale. With control of constant activation, geometries representing individual brushstrokes can be created with specified starting time and end time.

3.2.2 Collision

Since the particles were moving in three-dimensional space, a collision object, which acted as a virtual canvas, was needed to catch the splashes of the brushstrokes. A collision detection was set up in the same dynamics network as the particle simulation. A geometry with user defined position and dimensions was referenced inside the network as a static rigid body. It was fed into a *Static Solver*, which was then merged with the *POP Solver* that simulated particle behaviors. Figure 3.4 shows the network view of the particle system setup.

3.2.3 Wedge of Brushstrokes

A wedge of brushstrokes was generated based on different positions of the emitter. The particles would start from different positions and directions in space and exhibit different particle behaviors. Thus, it would generate brushstrokes with different shapes and movements. This was done with a *Wedge* node in Houdini, which renders the same geometry with different settings. For example, the x, y, z values referencing the position of the emitters were set to their corresponding range and the number of steps were set to 5, yielding a total of 5^3=125 different settings. Assuming

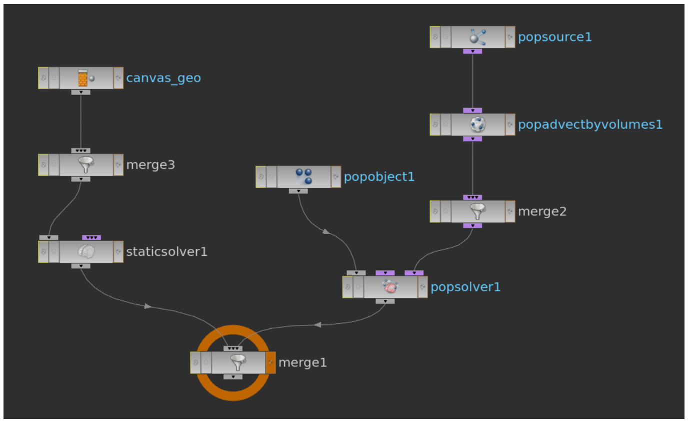


Figure 3.4: Network view of the particle simulation

the time range is 1-100, a total of 125*100=12500 geometry files (.bgeo files) would be written to disk, in other words 125 distinct brushstrokes over a time range of 1-100 would be generated. Figure 3.5 shows a list of example brushstrokes from one wedge.

3.3 Designing an Art-Directable Workflow

A great amount of the work in the development of this thesis was dedicated to automating procedures and creating an art-directable framework for artists to explore the balance between the nature of turbulent flows and aesthetic concepts. This section summarizes when and how artists can interject their own aesthetic judgments in this process.

3.3.1 Constructing the Canvas and the Emitter

The simulation network was customized so that users are able to control the size (sx, sy, sz) and position (px, py, pz) of the canvas in order to determine the position and domain of the collisions. The size and position of the emitters can be set the same way, or they can be handled by a wedge as mentioned in previous section, if multiple brushstrokes are desired.

3.3.2 Viewing Individual Brushstokes

Loading geometry files that represent brushstrokes from disk can be done easily with Houdini's *File* node. Artists are able to view the collision between each brushstroke and the canvas over time. One problem was to keep only the particles that were in contact with the canvas at each

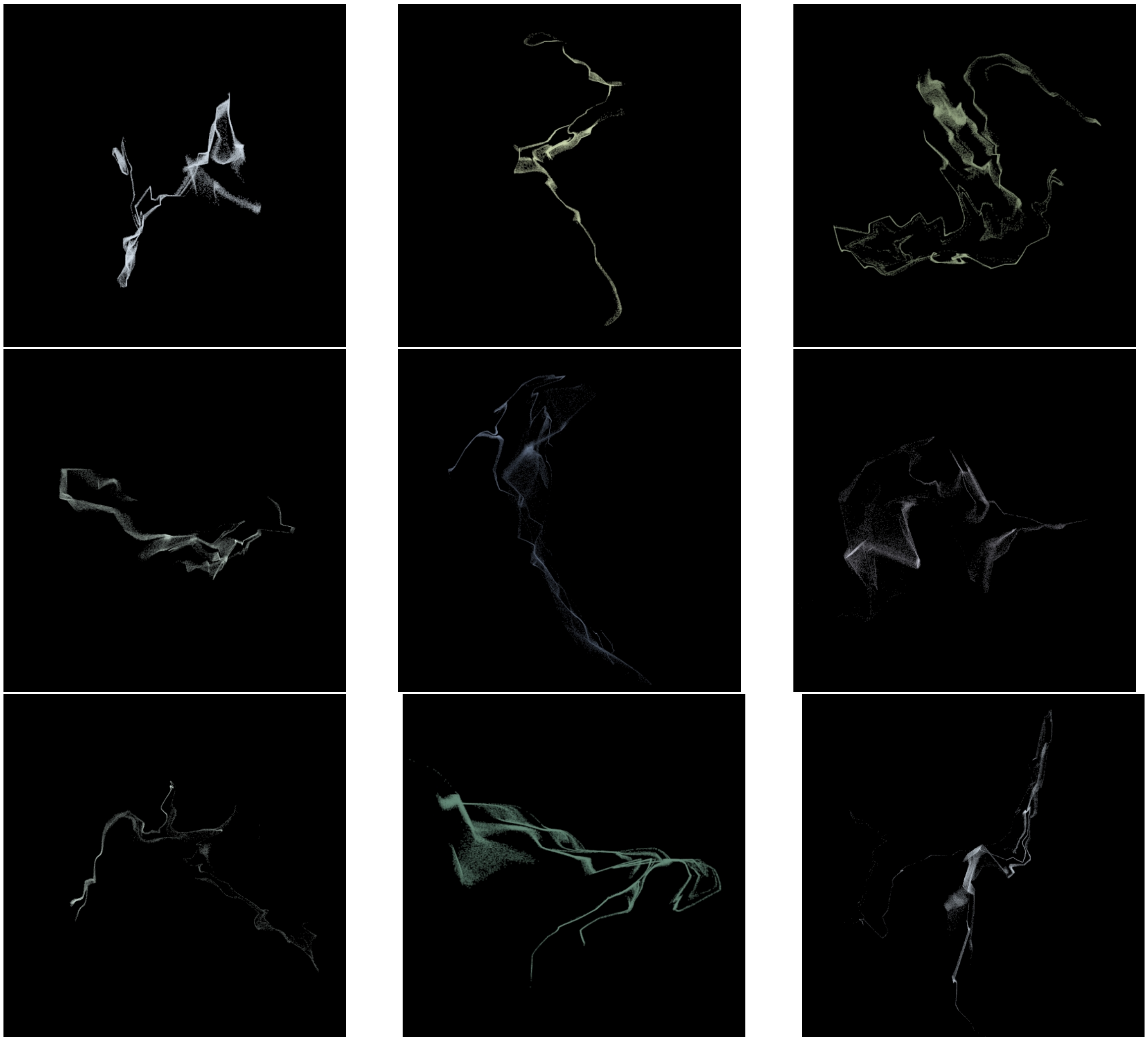


Figure 3.5: Brushstrokes with varying shapes

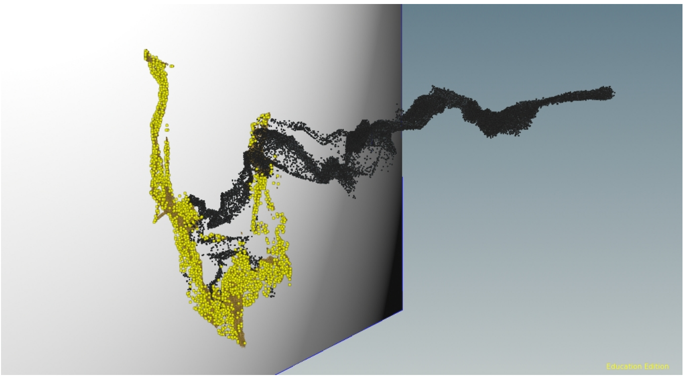


Figure 3.6: Particles on canvas being highlighted

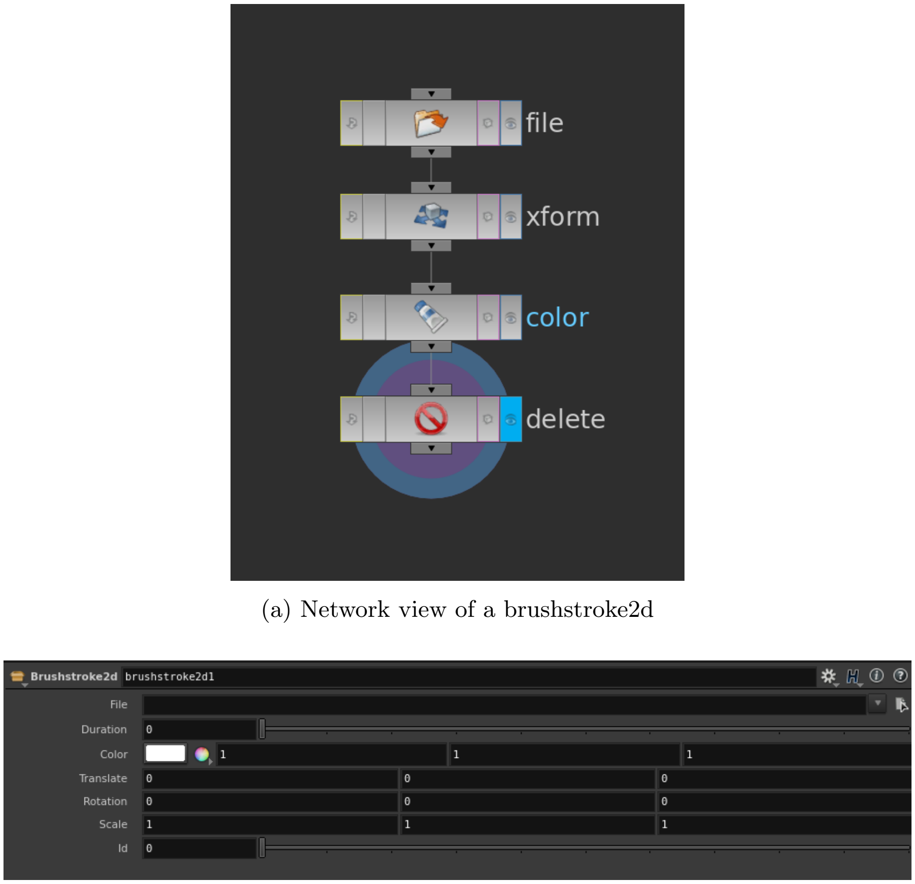
frame. To achieve this, a *Group* node was used to select the group of points that fall within a three-dimensional bounding box, coupled with a *Delete* node, which deletes all other particles, so that only the particles on canvas are kept. Figure 3.6 shows the particles on canvas being highlighted. Artists can create a list of *File* nodes that represent different file sequences and keep those for future manipulation. They can choose to have a brushstroke only at one particular frame as well.

3.3.3 Manipulating and Composing the Brushstrokes

After the file sequences are selected, the final step is to create the painting by manipulating and composing the brushstrokes. The list of controls that artist has are duration, color, translation, rotation, scale, and offset.

The listed controls require different nodes added to each geometry, and thus a repetition in terms of wiring nodes. A Digital Asset "brushstroke2d" was created to ease the process. Each set of nodes required to control an individual brushstroke was encapsulated in a subnetwork and then converted to a Digital Asset. Also, all the control parameters were promoted to the highest-level so that artists could control multiple parameters without navigating through networks. Figure 3.7 demonstrates the brushstroke2d subnetwork and the high-level controls provided for artists.

The framework thus presents an alternative in creating art. It emphasizes components that might be invisible to or ignored by artists, for example, the physics that govern the movements of the paint, and it makes those direct and accessible to artists so that they can explore the "randomness" in nature, and at the same time exercise their own aesthetic judgments. The controls at different levels add layers of flexibility to artistic creation. With the same set of brushstrokes generated, artists could produce a wide range of final compositions. Even with similar compositions, the painting



(a) Network view of a brushstroke2d

Brushstroke2d brushstroke2d1

File

Duration 0

Color 1 1 1

Translate 0 0 0

Rotation 0 0 0

Scale 1 1 1

Id 0

(b) High-level controls of a “brushstroke2d”

Figure 3.7: brushstroke2d Digital Asset

process could yield significant differences.

Chapter 4

Animation

The potential of the framework described in this thesis is demonstrated through a short animation, which consists of three shots. The first shot shows brushstrokes moving through three-dimensional space and splashing onto a two-dimensional canvas; the second shot illustrates the process of making a painting; and the last shot shows the canvas rotating in space.

The building blocks of the final painting are individual brushstrokes. While the brushstrokes were generated from turbulent flows with unpredictable behaviors, the painting was composed with deliberation in terms of the placement and timing of the brushstrokes. It is based on the ideas from "all-over painting", which refers to "a canvas covered in paint from edge to edge and from corner to corner, in which each area of the composition is given equal attention and significance" [1]. The technique was first developed by Ukrainian-American artist Janet Sobel, whose work was shown at Peggy Guggenheim's gallery in the 1940's and admired by Jackson Pollock [16]. Figures 4.1 and 4.2 show two examples of all-over paintings.

With "controlled randomness" as the central concept of this thesis, it is important to engage principles coherent to those of Pollock's underlying the design of the final painting. As Pollock stated on his artistic process, a few things were considered fundamental and vital – first and foremost is directness, since there is no drawing or color sketch made before the painting; second is the large scale of the painting, which makes the painter feel like part of it; and lastly, there is no beginning nor end of the painting [10]. A set of criteria was thus developed to guide the placement of brushstrokes based on these principles:

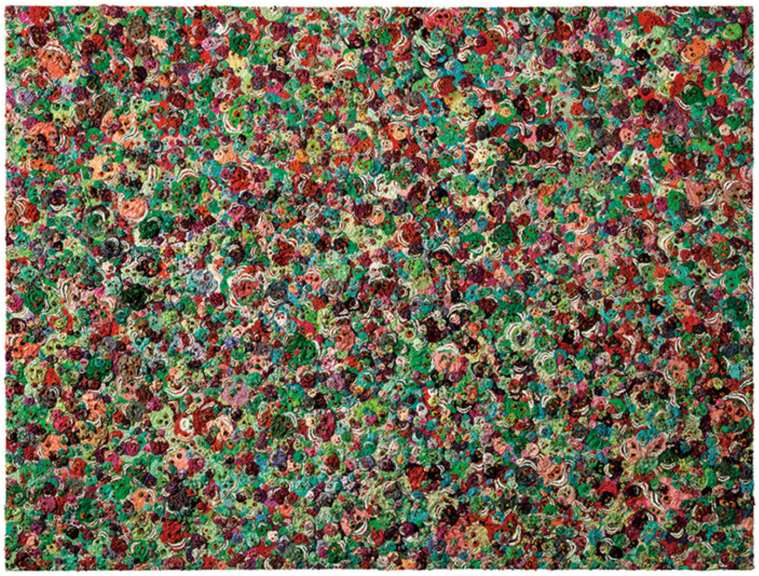


Figure 4.1: Huan Zhang, Poppy Field No. 12, 2012

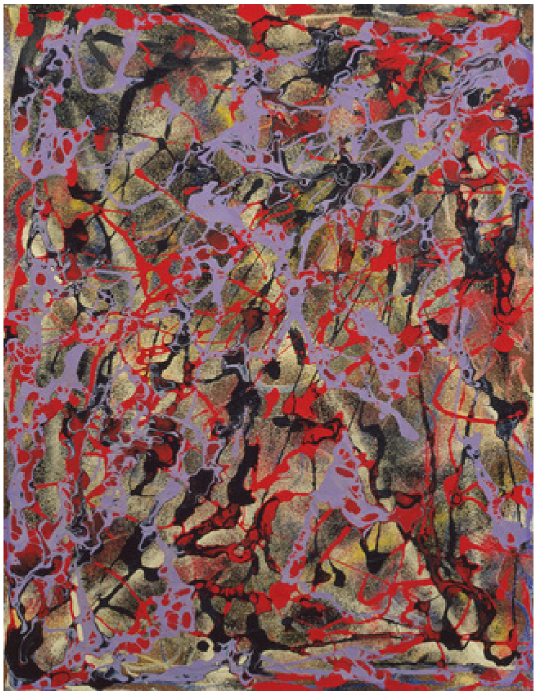


Figure 4.2: Janet Sobel, Untitled, 1946

- The brushstrokes are divided into several groups. A short break is inserted between each group, echoing the non-constant process of making a painting.

- The duration of each brushstroke in a particular group falls within a certain range.

- The brushstrokes appear roughly clockwise or counterclockwise within each group, mimicking an artist walking around a canvas on the floor.

- The brushstrokes in the same group share the same color; color varies in different groups.

- There is no clear figure-ground relationship – the canvas is covered evenly with no particular region catching more or less attention.

Figure 4.3 shows the progression of brushstrokes on canvas in the final animation.

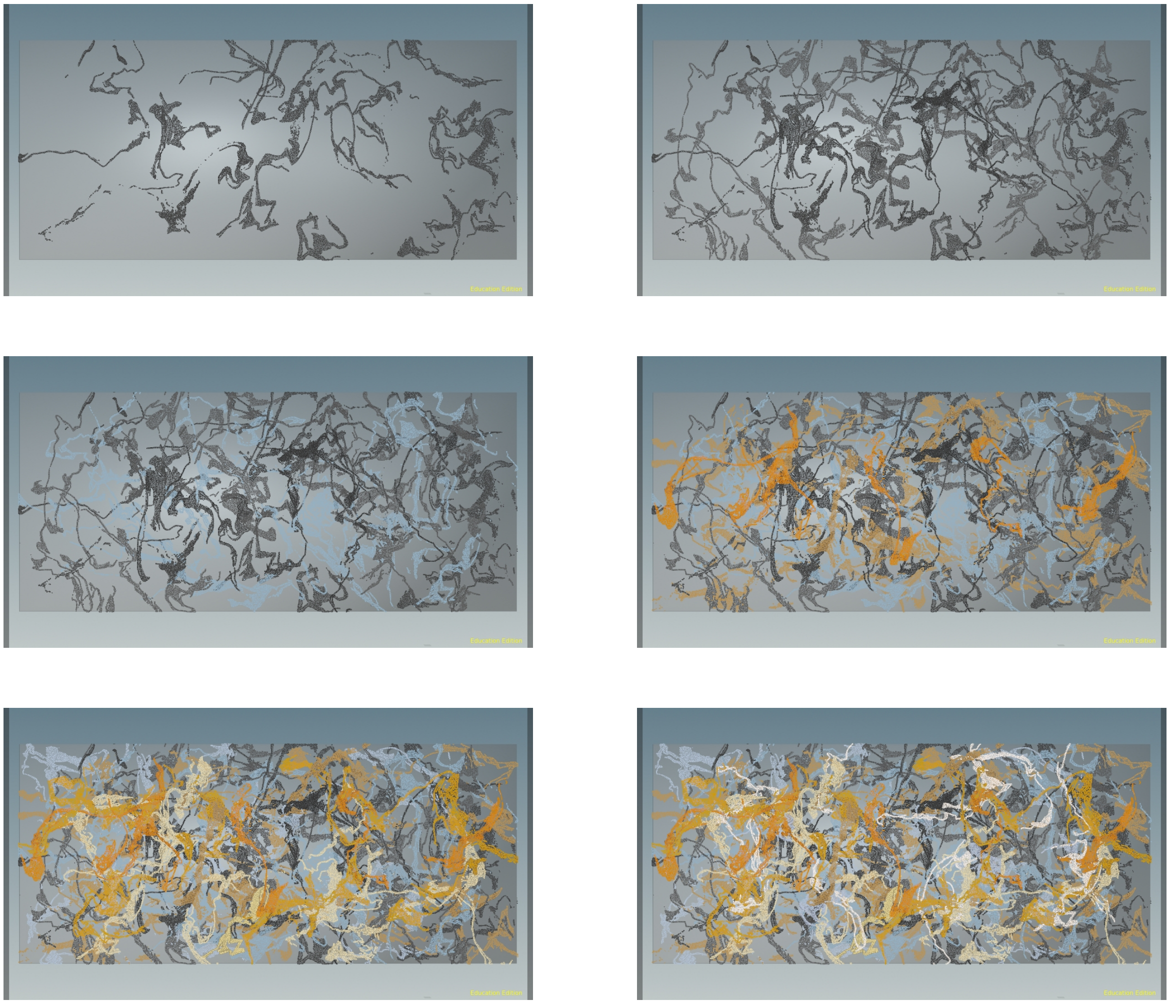


Figure 4.3: Progression of brushstrokes

Chapter 5

Shading and Rendering

As an important component of constructing the final look of any images, shading creates a depth perception in 3d models. Two shaders were created to display the surface properties of the paint and the canvas.

The canvas shader was created with a “Mantra Shader Builder” in Houdini, which is a “high-level shader that can contain one or more sub-shaders, such as surface shaders and displacement shaders” [2]. A constant color was given as the surface color. Displacement was generated by adding 3d Perlin noise to each point position and displacing the point along the surface normal. A rest position node was added to ensure that the noise follows the geometry as it moves.

One of the challenges in shading was to add physicality to the brushstrokes. Since the brushstrokes were particle-based, with each particle having an alpha attribute, it was easy to capture

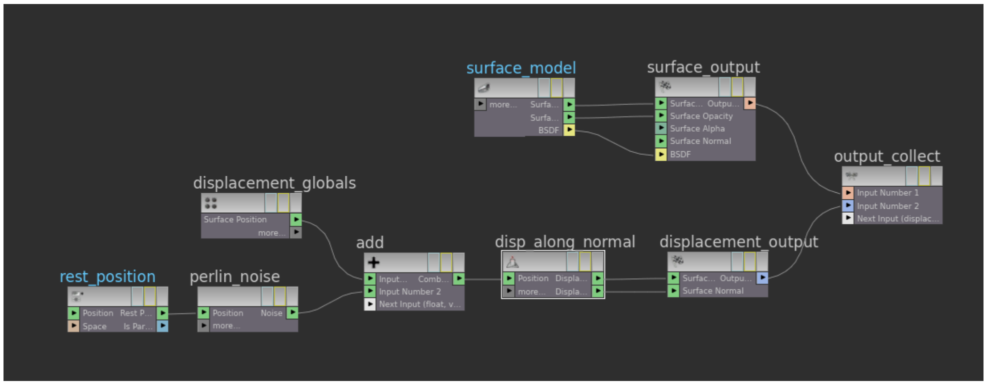


Figure 5.1: Network view of the canvas shader

Figure 5.2: Canvas rendered with no displacement

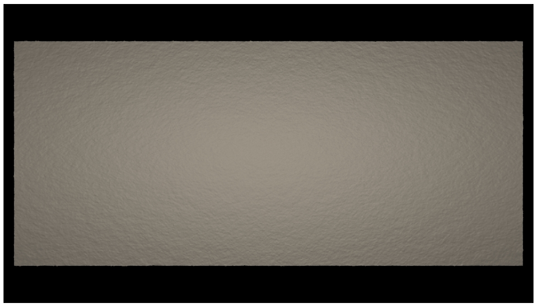


Figure 5.3: Canvas rendered with a displacement amount of 0.2

the thickness and clustering property of each brushstroke in the alpha channel. Nevertheless, with no normal associated with each particle, the brushstrokes were rendered as flat surfaces, and so the surface was not perceived as painted. The solution was to generate a bump map, which is a technique used in computer graphics to create an illusion of depth on a surface. That is to say, the effects are fake – there is no modification of or additional resolution added to the original geometry. Bump maps are usually grayscale images with the color value at each pixel indicating the amount of bump at each point. Therefore, the height at each pixel was calculated by counting the number of hits by brushstrokes. An easy way to represent that information is to use the value from the alpha channel. Figure 5.4 shows an example of a bump map converted from the alpha channel of a rendering. The opacity map was generated with the expression c==0?0:(c<th?(1-th+c):1) applied to each of the R, G, B channels, where c is the channel value, and th is the threshold. Figure 5.5 shows an opacity map converted from figure 5.4. Together with a color map, the two maps were then attached to a "Mantra Surface Shader" as a shader for paint.

The final step was rendering. Instead of rendering the original geometry, which consisted of over 150 brushstrokes and 6 million points, a grid with the same dimension as the canvas in x and y direction was created. Therefore, the paint can be rendered by attaching the paint shader to the grid and rendering the grid. Figures 5.6-5.11 show the use of maps to create the final paint effects.

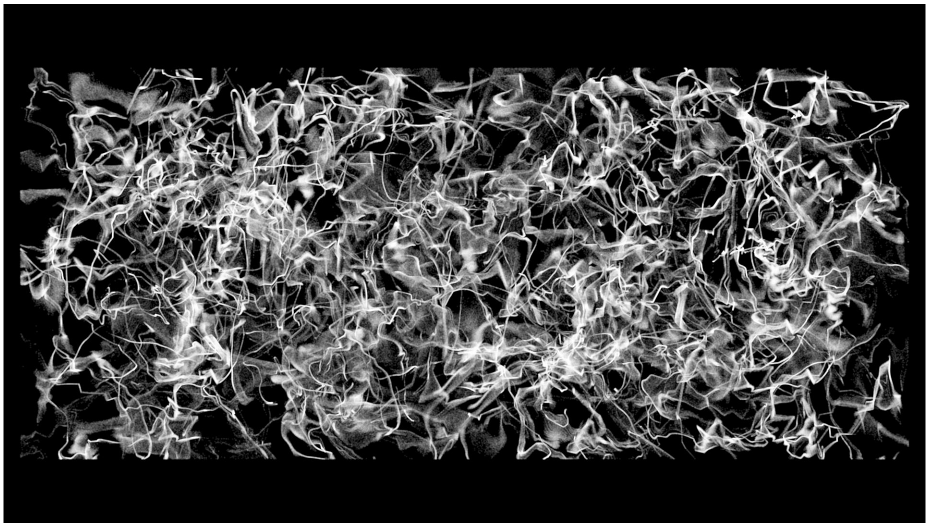


Figure 5.4: Bump map generated from alpha channel of a rendering

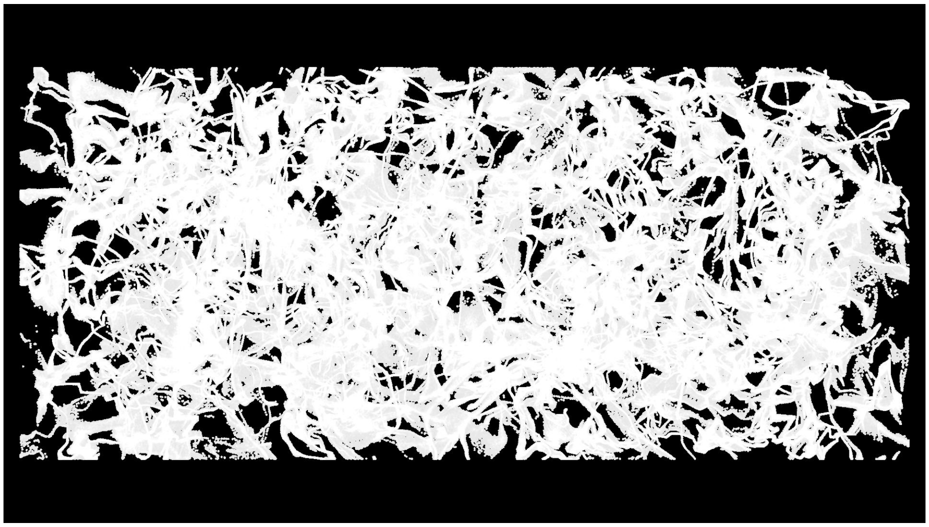


Figure 5.5: Opacity map converted from the bump map

Figure 5.6: Rendering only the grid

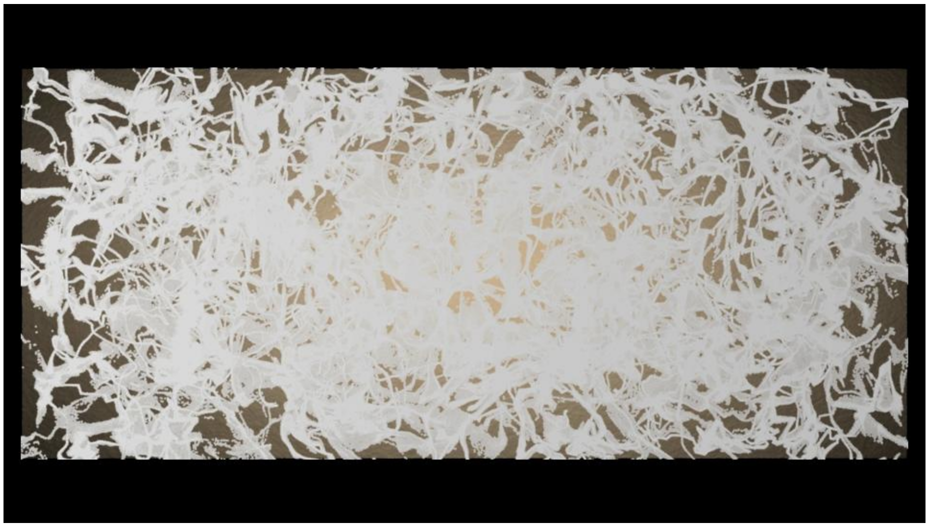


Figure 5.7: Adding the opacity map

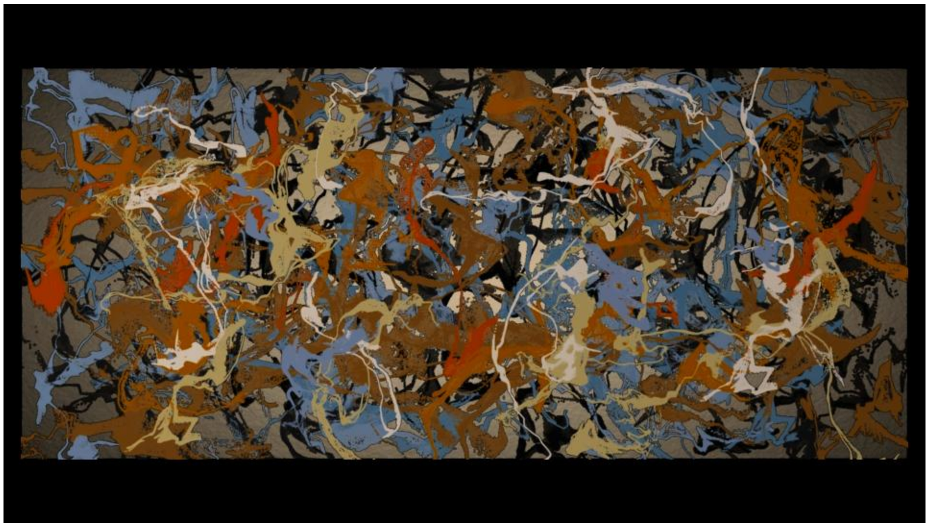


Figure 5.8: Adding the color map

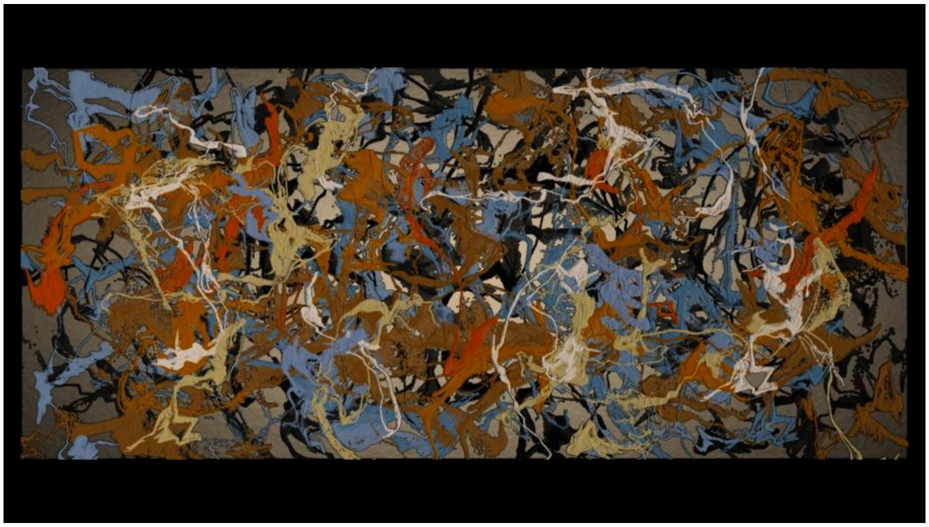


Figure 5.9: Adding the bump map with a scale of 0.2

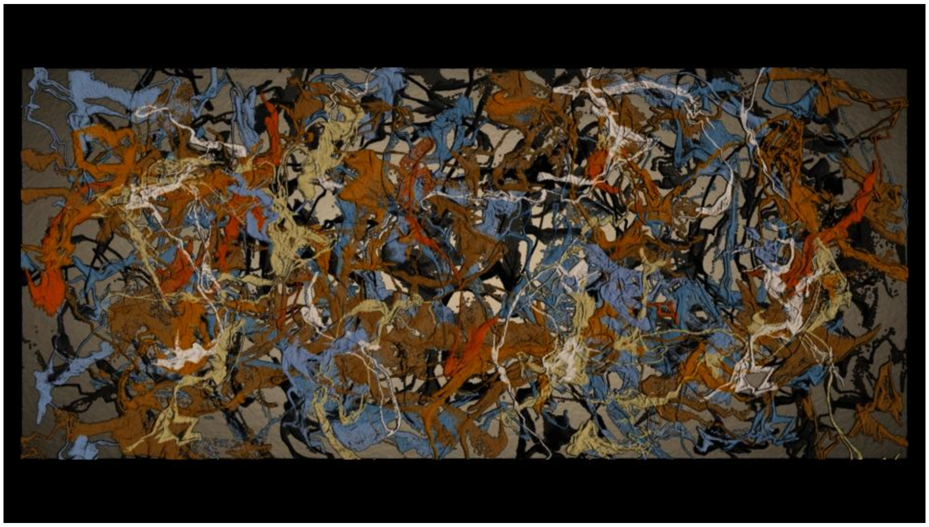


Figure 5.10: Adding the bump map with a scale of 0.5

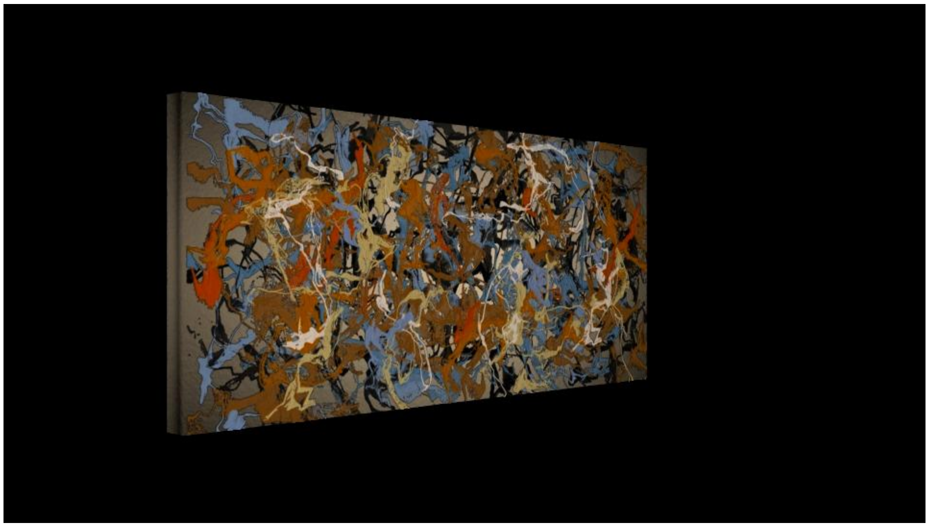


Figure 5.11: Rotated view of the canvas

Chapter 6

Conclusions and Future Work

This thesis has presented a technique to create simulation-driven brushstrokes. A pipeline was built to import turbulence data into Houdini and process it for particle simulations. In addition, a framework has been implemented for creating paintings that are composed of individual brushstrokes.

Within the framework, artists are able to visualize and explore information embedded in air flow simulation, and more importantly, exercise their judgment based on aesthetic concepts when creating paintings. This thesis emphasizes the scientific and subconscious components of art creation, and makes them visible and accessible for artists. A short animation was created to demonstrate possible outcomes. While the composition showed in the animation was heavily influenced by the concepts associated with all-over painting and Jackson Pollock's action painting, there are many aspects that artists can experiment with, especially the criteria for placing the brushstrokes, to convey different artistic concepts and styles. They can also explore different velocity fields, and insert a virtual canvas at different positions and directions inside a velocity field to compare the results. The canvas is not restricted to have a rectangular shape so that artists can design canvases with special shapes as well.

Furthermore, several improvements and extensions can be added to the framework to allow more artistic controls over the creation of a painting. First, the Digital Asset "brushstroke2d" in Houdini allows brushstrokes to grow from one end to the other. Control points could be added to allow them grow from an arbitrary point to both ends, which would offer more artistic freedom in terms of expression and style development. Second, material properties, such as viscosity, are worth experimenting with in shader writing. For example, a very thin and fluid material can be applied to simulate ink and a glowing material can be applied to simulate luminous paint. Also, time-dependent factors can be added to the animation. For example, due to gravity, the beginning of a brushstroke would appear thicker than the end.

Lastly, fractal analysis has been used in authentication of original works that are suspected to have been created by Jackson Pollock [18]. It is a natural extension of this thesis to explore whether dynamic fractal analysis of images might be used to supplement the creative processes described here.

Bibliography

[1] Allover painting. `https://www.moma.org/cef/abex/html/know_more9.html`.

[2] Houdini 14.0 documentation. `http://www.sidefx.com/docs/houdini14.0/`.

[3] Philip Ball. The scientific problem that must be experienced. In *Nautiluss*, volume 15. NautilusThink, 2014.

[4] George Keith Batchelor. *An introduction to fluid dynamics.* Cambridge university press, 2000.

[5] Shiyi Chen, Hudong Chen, Daniel Martnez, and William Matthaeus. Lattice boltzmann model for simulation of magnetohydrodynamics. *Physical Review Letters*, 67(27):3776, 1991.

[6] Wladimir Lyra. Elliptic and magneto-elliptic instabilities. In *EPJ Web of Conferences*, volume 46, page 04003. EDP Sciences, 2013.

[7] Algirdas Antano Maknickas. On global solution of incompressible navier-stokes equations. In *Proceedings of the World Congress on Engineering*, volume 2. Citeseer, 2014.

[8] JM McDonough. Introductory lectures on turbulence physics, mathematics and modeling. 2004.

[9] Tony McLoughlin, Robert S Laramee, Ronald Peikert, Frits H Post, and Min Chen. Over two decades of integration-based, geometric flow visualization. In *Computer Graphics Forum*, volume 29, pages 1807–1829. Wiley Online Library, 2010.

[10] Hans Namuth and Paul Falkenberg. Jackson pollock, 1951.

[11] YH Qian, Dominique d'Humières, and Pierre Lallemand. Lattice bgk models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

[12] W. T. Reeves. Particle systems: a technique for modeling a class of fuzzy objects. *ACM Trans. Graph.*, 2(2):91–108, April 1983.

[13] Brenden Roberts, Joshua Levine, and Robert Geist. Gpu-based simulation on high reynolds number flows across intricate geometries. preprint(2015).

[14] Harold Rosenberg. The american action painters. *Art News*, 51(8):22, 1952.

[15] Lior Shamir. Computer analysis reveals similarities between the artistic styles of van gogh and pollock. *Leonardo*, 45(2):149–154, 2012.

[16] Roberta Smith. Art in review; janet sobel. `http://www.nytimes.com/2002/02/15/arts/art-in-review-janet-sobel.html`.

[17] Natalya St. Clair. The unexpected math behind van gogh's "starry night", 2014.

[18] Richard P Taylor, Adam P Micolich, and David Jonas. Fractal analysis of pollock's drip paintings. *Nature*, 399(6735):422–422, 1999.

[19] Arkady Tsinober. *An informal introduction to turbulence*, volume 63. Springer Science & Business Media, 2001.

[20] Akiva Yaglom. The century of turbulence theory: The main achievements and unsolved problems. In *New trends in turbulence Turbulence: nouveaux aspects*, pages 1–52. Springer, 2001.