5-2019

Multi-Color Diffusion Simulation of Dye Over Folded Fabrics

Horst R. Evans
Trinity University, horst.evans@gmail.com

Horst Evans

Multi-Color Diffusion Simulation of Dye Over Folded Fabrics

Trinity University Computer Science
Bachelor of Science Thesis
April 2019

TRINITY UNIVERSITY

Abstract

Horst Evans: Multi-Color Diffusion Simulation of Dye Over Folded Fabrics
Bachelor of Science Thesis
Trinity University
Bachelors Degree Program in Computer Science
April 2019

Creating a unique dyed shirt requires accurately simulating a fabric model, implementing the ability to fold the fabric, encoding the process of dye diffusion through the fabric, and allowing for different color channels of dye to mix. These goals require an adherence to the physical properties and laws that govern the process of diffusion and a suitable representational model that this diffusion is conducted on and through.

This paper presents a model for the representation of fabric that is comprised of two layers woven together in a weave pattern. Using Ficks Second Law of Diffusion and properties of the fabric and threads, we calculate the rate of diffusion for each cell of the fabric. This calculation is done over each color channel to allow for the full spectrum of dye colors to be realized. Using a relationship matrix, the fabric can be folded over itself, and the dye will diffuse over this fold into the layer on the other side. Most of the parameters involved in defining the type of fabric can be modified to allow for a large range of visual expression in the fabric. The results support the effectiveness and veracity of the model.

Keywords: Dye Diffusion, Fabric Modeling, Tie-Dye Art, Rendering.

Contents

1 Introduction 1
1.1 Importance 1
1.2 Goals 2
2 Previous Works 3
3 Terms and Definitions 7
3.1 Tie-Dye 7
3.2 Diffusion 8
3.3 Thread 8
3.4 Fiber 9
3.5 Weave 9
3.6 Weft and Warp 11
3.7 Tortuosity 11
3.8 Porosity 12
4 Environment 13
5 Methods: Data Structures 15
5.1 Diffusion Cell 15
5.2 Cloth Cell 16
5.3 Layer 16
5.4 Cloth Model 17
5.5 Run Simulation 18
6 Methods: Algorithm Design 19
6.1 Fick's Second Law of Diffusion 19
6.2 Tortuosity 21
6.3 Diffusion Coefficient 21
7 Visualization 22

7.1 Implementation System 22
7.2 Parameters and Variables 22
7.3 Setup Function 23
7.4 Draw Function 25
7.5 Dyeing The Fabric 26
7.6 Saving the Image 26
7.7 Folding 27
8 Results 28
8.1 Code Repository 35
9 Next Steps 36
10 Conclusion 38
References 39

List of Figures

3.1 A Tie-Dye shirt, twisted and dyed [2] 7
3.2 Diffusion of particle through a volume [6] 8
3.3 A small selection from a plain weave pattern. The threads alternate over under [7] 9
3.4 Visualization of Gaps in Fabric 10
3.5 The three classic weave patterns: plain, twill, sateen [4] . . . 10
3.6 The Weft and Warp in fabric [10] 11

4.1 Processing Logo [21] 14
4.2 Eclipse Logo [20] 14

5.1 A general layout of the structure from [16], which provided a solid basis for this project 17

6.1 A simple vertical fold [5] 20

8.1 Blue Diffuse Incorrect Parameters 28
8.2 Color Diffuse Incorrect Parameters 29
8.3 $\triangle d = .005$ 29
8.4 $\triangle d = 1$ 30
8.5 $\triangle d = 1.3$ 30
8.6 $\triangle d : 1.75$ 31
8.7 Triangle Shape 32
8.8 Square Shape 32
8.9 Large Circle and Fabric 33
8.10 Modified Tortuosity Values ($1- > .005$) 34

1

Introduction

The realm of simulation operates on the foundation of a strict adherence to the laws of physics that govern the world in and around us. For clothes to be colored, the threads must be dyed. For the threads to be dyed, the dye must diffuse through the fabric. This act of diffusion has strict rules that it must follow, which have been described by Adolf Fick in his Laws of Diffusion.

Tie-Dye uses these laws. It is an art form that involves folding or tying fabric into a desired shape and dyeing the result to create interesting patterns of dye on the cloth. In this artistic endeavor, simple colors mix to form more complex ones, and dye is transferred across the folds and tied fabric created by the artist. Therefore, in the simulation of this art form, it follows that we must accommodate the manipulation of fabric, and color mixing.

1.1 Importance

The importance of this paper to the field of simulation is that it creates more realistic representations of textiles which can be applied to many fields for a variety of uses. While we chose to model the Tie-Dye textile art style, this system can be applied to many other textiles and styles by altering a few parameters such as tortuosity, weave pattern, cell distance, thread properties, and porosity. Other styles include shirt printing and vat dyeing. It could also be used to prototype designs in fashion, create detailed textures for games, and more.

1.2 Goals

The overall goal of this research is to create a model and system that allows for the creation of accurate Tie-Dye images. However, this begets several subordinate aims:

- We must create a representation for fabric that can be dyed multiple colors and that can be folded over itself.
- We must simulate an accurate-to-physics diffusion of dye through the fabric.
- The interaction of the two previous items requires that the fabric has physical properties that affect how the dye diffuses.
- We must also create a method to adjust various aspects of the diffusion process, such as how long the diffusion process lasts, how much dye is placed on the fabric, the size of the fabric, etc.
- Finally, we must create a method of visualization for all of the previous requirements that allows for a time step visualization of the diffusion process and the creation of a final image.

The successful fulfillment of all of these aims is described in this paper, as are the results of our efforts.

2

Previous Works

The field of artistic simulation is not new, and it has a solid body of previous work. It is, however, quite broad, and covers a wide range of topics, from watercolor to painting with ink wash.

The work of Liu et al. [11], focuses on the simulation of hand-painted patterns. They break down the simulation into three primary components:

1. the cloth grid
2. the contour grid and interaction areas (pattern)
3. and the dye filling

The cloth grid is a two dimensional model with three classes of grid cell: warp, weft, and gap. Each grid cell can be on the top layer or the bottom layer. In addition, they define five types of position relations between two cells. I: warp and warp. II: weft and weft. III: warp and weft. IV: warp/weft and gap. V: gap and gap. These relations are used in calculating diffusion rates.

The creation of the contour grid and interaction areas is done by taking a color image, and using boundary extraction algorithms to create a contour image. This is then mapped onto a grid of the same size as the cloth to get the interaction areas.

The simulation is calculated using the Navier-Stokes equation. Boundary restriction is also done to adhere to the pattern. This paper heavily influenced the data structure implemented.

Sakurai et al. [17] specializes in the fabric side of the problem. Their paper focuses on the accurate simulation of woven textiles with fuzz. The

fuzz is generated by transforming polygonal chains that represent individual fuzz strands.

Xua et al. [22] propose a generic pigment model for both dry and wet painting and the gradient in between. They use a brush model to simulate the tip of the brush on the paper. They utilize a formulae to simulate paint moving between mediums, and another to simulate the diffusion after application. They also handle nuanced physics (evaporation, brush tip bundle). Rendering is done by utilizing the Kubelka-Munk model.

Chu et al. [1] discuss simulation of ink on paper. They used the Lattice-Boltzmann Equation for the simulation of ink diffusion. The LBE approach uses a particle kinetic model to divide the simulation into a regular lattice. They use a three-layer paper model, with the layers being surface, flow, and fixture. Ink is deposited on the surface layer, diffuses through the flow layer and into the fixture layer, where it does not move. They also consider a brush model as a deposit source. This 3-layer model is a different approach to the problem of diffusion.

Curtis et al. [3] concerns itself with realistic rendering of watercolor style. They implemented a 3-layer model, with the layers being the shallow-water layer, the pigment-deposition layer, and the capillary layer, with the capillary layer in charge of actual diffusion. Diffusion is handled through the rate of fluid movement, a relatively simple diffusion method. The paper also extensively covers effects. This paper provides a unique and in depth look at the physical nuance of a simulation, and provides a good basis for diffusion.

Simulation of ink on paper in diffuse ink painting style was done by Kunii et al. [8]. They break up the style into three zones, initial zone, black border, and gray zone. Using a cartesian coordinate system, they derive a mathematical formalisation of the unique properties of the dye. This formalisation was then used to derive simple computer simulations.

Shamey et al. [18] modeled simulation of dye in cloth via dyebath as opposed to plug flow. They take into account three major phenomena involved in mass-transfer dyeing: diffusion, dispersion, and convection. Diffusion is done via molecular diffusion and Ficks second law. Dispersion is dependant on flow velocity and Convection is dependant on bulk-flow,

with equations given to simulate both. This paper provides an insight into a more complex look at dye simulation modeling, but lacks a physical simulation component.

Work has previously been done on creating a suitable representation of fabric that can support dye. Morimoto et al. [15] has previously done work on the design and implementation of a two-layered cloth model that supports single color dyes. They represent the cloth as a 2 layered model comprised of cells. Each cell is either a weft or warp cell, and has a transportability and maximum dye capacity value. Diffusion is modeled by the functions for capillary flow. They identify 3 layers of transport:

1. between layers
2. parallel to layer direction
3. perpendicular to layer direction

This paper provides good fundamental support for the idea of a fabric representation.

Morimoto and Kenji Ono [14] propose a new cloth model to be used for dye simulation, especially in folded 3D cloth geometries. It utilizes a Voronoi diagram for cloth gathering and locally applied geometric operations for multiple dyed patterns. The diffusion is done through a 3D model.

In a different paper by the same pair [12], they address the dyeing of 3-D cloth shapes, such as folded pieces of fabric. This is done by taking the 2-layer, 5 relation model of their previous working and adding in a new relationship of the contact cells.

In yet another paper by this prolific duo also dealing with folded fabric [13], they utilized a 3-D graph to simulate the folding relationships. The diffusion equation is then applied to this graph, which connects folded areas together. It also takes into account pressed areas vs connected areas, as pressed areas have a different diffusion coefficient.

Our implementation was heavily influenced by Morimoto et al. [16] and their work on the simulation of dye on cloth. It utilizes a cloth model

and a diffusion algorithm. The cloth model is made up of the weft layer and the warp layer, with each cell being a weft cell, a warp cell, or a gap cell. Each main cell is subdivided into several diffusion cells. For diffusion, Ficks second law is used. They also include tortuosity as an important factor, with 3 types, and the 3rd type having 3 separate sub-parts that define relationships between cells (different layer, cell and gap, different directions within a cell). To get a visual representation, the model is run for a set number of time steps. It would most likely be written to an image, although this is not implicitly stated. This in-depth paper covers a solid basis for which to expand upon.

3

Terms and Definitions

This paper touches on a broad range of topics, from folk art to diffusion physics to the components of fabric. These terms will be used throughout this paper, so a working definition is required to help the reader become familiar with the ideas relevant to this paper, especially those definitions in fields far from the reader's normal environment of study.

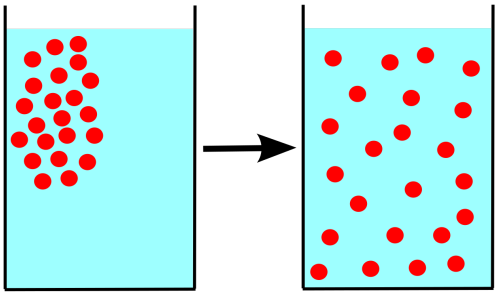
Figure 3.1 *A Tie-Dye shirt, twisted and dyed [2]*

3.1 Tie-Dye

Tie-Dye is an art form that uses folds and dyeing techniques to apply color to fabric, often in unique shapes. The cloth is often folded or knotted first,

and then subsequently dyed to have color diffuse through the fabric and over the folds, creating intracite designs and patterns. Normally cotton is used, as cotton readily absorbs water better than other fabrics, such as polyester.

***Figure** 3.2 Diffusion of particle through a volume [6]*



3.2 Diffusion

Diffusion is a process by which liquids propagate from regions of high concentration to regions of low concentration. In the case of dye on fabric, the liquid diffuses from the area of application to the surrounding fabric, carrying with it the dye that binds itself to the fabric. However, the rate of diffusion is dependent on multiple factors, such as the relative concentration levels of liquid in surrounding areas, the porosity of the fabric, the tortuosity of the fabric, the shape of the weave, and the size of the threads.

3.3 Thread

The fabric is the medium through which the dye must diffuse. This fabric is made up of individual threads woven together to create the final textile. The properties of the threads, such as thickness and porosity, can drastically alter how the dye will diffuse through it, and significantly impact the outward appearance.

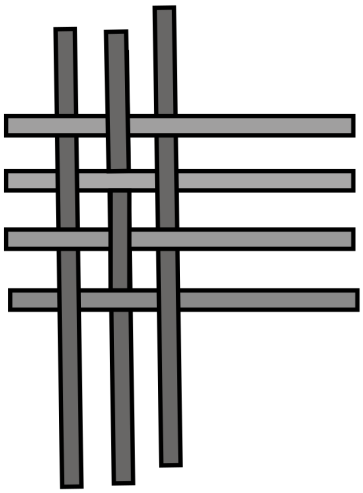
3.4 Fiber

Each thread is composed of fibers. The fibers are the medium through which the dye diffuses. A single thread is likely to have many fibers within them that compose the thread.

3.5 Weave

These threads are woven together in a pattern. The traditional pattern, and the one used throughout this paper, is the plain weave, or criss-cross pattern. Plain weave has two layers of threads. Each layer has multiple threads that are parallel to each other, but the layers themselves are perpendicular. The layers are woven between each other, such that a thread from the first layer goes over one thread of the second layer and then under the next. The next thread in the first layer repeats this pattern, but reversed, so if the first thread started by going over, then this next thread will start by going under.

Figure 3.3 *A small selection from a plain weave pattern. The threads alternate over under [7]*



The pattern of the weave can significantly affect the final shape of the

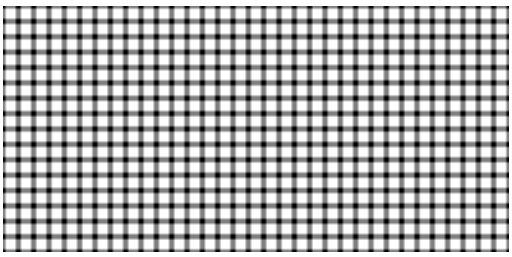
dye, so that weave pattern should be taken into consideration when selecting a fabric. We have chosen the plain weave pattern for several reasons:

One, it is easy to implement, because it is incredibly straightforward and only requires knowing whether the previous thread starts by going over or under.

Two, it is fairly recognizable as a pattern. This simple pattern is a standard in the textile industry and should be easily recognizable to those who are not experts or are not learned in fabrics.

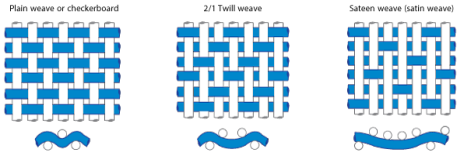
Three, it has a consistent and extremely symmetrical shape, allowing for nearly identical shapes to appear when dye is applied, regardless of where on the fabric the dye is placed. With other weaves such as twill or sateen, the placement of the dye can dramatically affect how the final image will appear, but with the plain weave, the image can only be affected by an offset of one thread.

Figure 3.4 *Visualization of Gaps in Fabric*



Each weave will have small portions of space between the threads, called gaps. These gaps affect how the dye diffuses throughout the fabric. The size of the gaps and the size of the threads are independent.

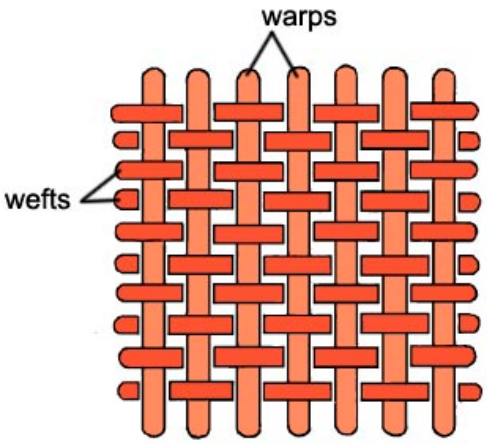
Figure 3.5 *The three classic weave patterns: plain, twill, sateen [4]*



3.6 Weft and Warp

Weft and Warp merely refer to the directions in which the threads of a specific layer are going. In this paper, weft refers to longitudinal or horizontal threads while warp refers to vertical or latitudinal threads. These separate layers are woven together to make the weave pattern, which as a whole comprises the fabric upon which the dye is applied.

Figure 3.6 *The Weft and Warp in fabric [10]*



3.7 Tortuosity

Tortuosity refers to how twisted the fabric is. Twist constricts flow, making it more difficult to diffuse a liquid through the fabric. There are three major degrees of twist, and tortuosity is the product of these three values.

1. There is the twist of the thread along its length, like a rope. This is the same for all threads in the fabric.
2. The second form of twist refers to how the thread is positioned in relation to the weave pattern and how the thread itself is oriented.
3. The third kind of twist is from how two different fibers are oriented in relation to one another. There are five different possible orientations between fibers, each with the possibility of having a different degree of twist.

I : There are two fibers in different layers, i.e. one fiber is in the weft and the other is in the warp.

II : There are two fibers in the same layer and they lie parallel to each other (i.e. they are in the same thread).

III : There is one fiber and a gap.

IV : There are no fibers, and two gaps.

V : There are two fibers in the same layer and they lie perpendicular to each other (i.e. they are in different threads).

3.8 Porosity

Porosity is concerned with how much empty space is present in a thread. The more porous something is, the more space it has and the more easily fluid can diffuse through it. It is assumed that porosity is constant throughout the fabric.

4

Environment

This thesis was written utilizing the Processing 3 programming language. Processing is a Java based language designed for simulations. It was chosen for its ease of use, open source nature, and because of prior university experience by the writer. To quote their website:

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

Most importantly, Processing handles the visual engine itself. The user only interacts with the visual engine through a setup() function, a draw() function, a pixel array for image manipulation, and a few other select library functions, allowing the user to spend the majority of their time developing their idea instead of fiddling around in the back-end, such as attempting to create an looping draw function or a mechanism to modify or create images.

Figure 4.1 *Processing Logo [21]*

Figure 4.2 *Eclipse Logo [20]*

5

Methods: Data Structures

This section describes the components that will hold the dye and allow for diffusion to occur. They comprise a representation of the fabric, and it is on these structures that the Fick's Second Law of Diffusion will be run. This implementation was based of the work of Morimoto et al. [16].

5.1 Diffusion Cell

The diffusion cell is the smallest unit in the model. It is used to calculate diffusion, and is the unit referred to by the diffusion equation. Each diffusion cell has a vertical position, horizontal position, and the layer it is in. The layer is either the weft or the warp. The diffusion cells have a reference to their parent cloth cell. The parent cloth cell passes a lot of this information down onto the child cell. It also gives the child cell information on whether the cell is up or down in the weave.

The diffusion cell has 3 channels for color: red, green, and blue, that determine the overall appearance of the dye (future works include converting this to a CMYK system). Each individual channel is a ratio with values in the range [0-1], where 1 indicates that the diffusion cell is completely full of that color, meaning that no other dye can enter. By extension of this premise, the sum of all three channels cannot exceed 1, as this would indicate an amount of dye present in the cell that exceeds the maximum amount physically capable of being held by the cell. The diffusion cell also

has a flag for being a gap. If the cell is not a gap, then it is part of a thread.

5.2 Cloth Cell

The cloth cell is the fundamental building block for the fabric data structure and for threads. They are strung together into threads and then these threads are woven together in a plain weave, making the fabric. Every cloth cell is also sub-divided into a number of diffusion cells, the exact amount depends on how thick the thread is and how large the gaps between threads are. These variables are defined by the user and used to calculate an upper and lower bound for the x and y range of diffusion cells within the cloth cell. Everything outside of this range is flagged as a gap cell. It takes into account any discrepancies between the width of the fabric versus the height of the fabric by looking at which layer the cloth cell is in. If the fabric is wider than it is tall for example, then the weft layer threads will be longer than the warp layer, but there will be more threads in the warp layer. Every cloth cell is given a unique ID that is used to identify it within a specific layer. This ID is also passed down to the child diffusion cells, allowing these child cells to have a reference to identify their parent cell. These diffusion cells are stored in a two-dimensional array. The cloth cell also knows which layer it is in, weft or warp, and whether it is up or down.

5.3 Layer

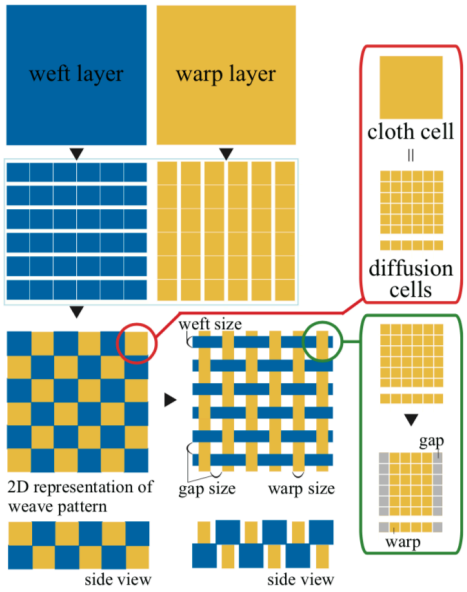
A layer holds all parallel threads in a single direction. There are two layers, a weft layer and a warp layer. The weft layer holds all horizontal threads, the warp layer holds all vertical threads. Each of these threads has an array of cloth cells that are strung together to form the thread. This array is constructed by first calculating the number of cells required for a single thread in the weft and for a single thread in the warp, which is done by taking the width (or height) of the fabric, which is defined by the user in the main class of the system, and dividing that number by the size of the thread to get the number of cells. We then iterate over these numbers, creating cloths cells equal to the product of the number of cells in the

width and the number of cells in the height. Each cloth cell is given a unique id, and they alternate being flagged as up or down, to simulate the plain weave alternating pattern. The first cell in each thread must also alternate between up and down, so two flags are actually used, the second determining whether the first cell in a thread starts as flagged up or flagged down. Since the x and y values for each cell is greater than one (assuming the thread size is greater than one) then each cloth cells x and y value is equal to the top left diffusion cells value, and each value is a multiple of the thread size. The values in between are distributed amongst the child diffusion cells.

5.4 Cloth Model

The cloth model comprises the entire sheet of fabric. It is very simple and holds the two layers, weft and warp. The cloth model mostly exists for simplification of code in the main class.

Figure 5.1 *A general layout of the structure from [16], which provided a solid basis for this project*



5.5 Run Simulation

The main class holds all of the user defined variables that determine how the fabric looks and functions. These variable are the width of the weft fibers, the width of the warp fibers, the size of the gaps between the fibers, the overall width of the fabric, the overall height of the fabric, and the distance between the individual diffusion cells from a physics standpoint (in mm). The main class also holds all run-time variables that affect how the dye diffuses through the fabric. These are the tortuosity values (including the 5 different forms of tortuosity-3, I-V) that represent how twisted the threads are and how much flow is restricted, the porosity of the thread that limits how much dye can move through at any given time, the $\triangle time$ that dictates how much time in seconds has elapsed during a single call of the diffusion function, the max volume ratio which limits how much of a diffusion cell can be filled up, the concentration of the dye which indicates the starting concentration of the dye solution, and fold multiplier which inhibits the transfer of dye across the fold in fabrics.

6

Methods: Algorithm Design

6.1 Fick's Second Law of Diffusion

Ficks Second Law of Diffusion is used to simulate dye diffusion in our fabric model. It returns the ratio of dye that enters or leaves a given diffusion cell based on the surrounding cell dye concentrations. This diffusion equation assumes that the cloth is wet so that the concentration of dye material is the main variable factor. The equation is based off of the one given by [16] in their paper:

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x}(D\frac{\partial \phi}{\partial x}) + \frac{\partial}{\partial y}(D\frac{\partial \phi}{\partial y}) + \frac{\partial}{\partial z}(D\frac{\partial \phi}{\partial z}))$$

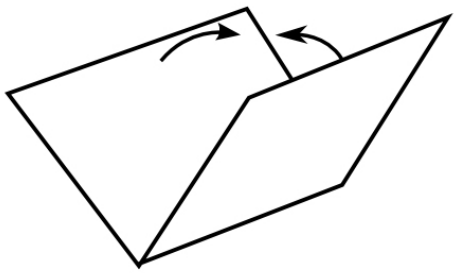
where ∂ indicates a partial derivative, ϕ is the diffusion density, t is time, D is the diffusion coefficient, x is the displacement along the x-axis, y is the displacement along the y-axis, and z is the layer (weft / warp) which can be thought to be analogous to the z-axis.

Following [16] again, we discretize the equation to get the change in concentration for a specific cell over the time interval $\triangle t$, which is:

$$\frac{\triangle\phi_{i,j}}{\triangle t} = \frac{1}{\triangle d}(D_{i+1,j,l} * \frac{\phi_{i+1,j,l} - \phi_{i,j,l}}{\triangle d} + D_{i-1,j,l} * \frac{\phi_{i-1,j,l} - \phi_{i,j,l}}{\triangle d} +$$
$$D_{i,j+1,l} * \frac{\phi_{i,j+1,l} - \phi_{i,j,l}}{\triangle d} + D_{i,j-1,l} * \frac{\phi_{i,j-1,l} - \phi_{i,j,l}}{\triangle d} +$$
$$D_{i,j,(l+1)/2} * \frac{\phi_{i,j,(l+1)/2} - \phi_{i,j,l}}{\triangle d} + fm * D_{f_i,f_j,(l+1)/2} * \frac{\phi_{f_i,f_j,(l+1)/2} - \phi_{i,j,l}}{\triangle d})$$

where i is the x index of the diffusion cell, j is the y index of the diffusion cell, l indicates the layer that the diffusion cell is in, $\triangle$d is the distance interval between diffusion cells, and D is the diffusion coefficient between two diffusion cells, one at (i, j) and the other at the given offset of (i, j). ϕ represent a diffusion cell at the given indices. The second term (e.g. $\frac{\phi_{i+1,j,l} - \phi_{i,j,l}}{\triangle d}$) calculates the actual change in concentration. Since this model supports color, we need to calculate this term three times to handle each of the color channels. Each of these terms is then multiplied by the diffusion coefficient. If the sum of these products exceed 1, then each individual term is divided by the sum to normalize the cell back down to 1. These three terms are the result of the equation, are are used to update the fabric.

Figure 6.1 *A simple vertical fold [5]*



The equation also takes into account folds in the cloth. The sixth term handles this fold interaction. The fold multiplier fm limits the amount of dye that can cross over this fold. The fold point is generated by referencing a fold matrix f that gives the corresponding point, if any, that lies above or below the folded diffusion cell.

6.2 Tortuosity

Tortuosity gets incorporated into the algorithm by using the total tortuosity value for the cell. This value is the product of the three kinds of tortuosity previously described in this paper ($t1 * t2 * t3$). This total value is used in the Diffusion Coefficient equation as a term.

6.3 Diffusion Coefficient

The diffusion coefficient is defined as follows:

$D_0() * porosity * tortuosity$

Where porosity is the ratio of void space to solid thread space in the fabric, and is defined during initialization, and tortuosity is calculated via its product function. $D_0()$ is the diffusion coefficient of free water, and is equivalent to 1.93 (mm^2/h).

7

Visualization

Now that the structure for the model of the fabric is completed and the diffusion equation is decided upon, all that is left is a concrete implementation.

7.1 Implementation System

This simulation model was completed in Processing3. Processing3 has many helpful built-in libraries that allowed us to focus on the implementation side of the model and diffusion equation and not on the back-end support systems such as creating a looping draw() function, finding a method to display the results, image library support, and more. Processing3 is a java based language, and so the project only required two imports: *processing.core.PApplet* and *processing.core.PImage*.

7.2 Parameters and Variables

Before the fabric can be instantiated and the dye poured on the cloth, we must define several global parameters. All of these values are defaults for figure 8.6 in the results.

All tortuosity values need to be defined:

$t1 = 1, t2 = 0.47$

$t3 = \{I, II, III, IV, V = 1\}$

These values create a smooth gradient within the shape of the placed dye that does not place significance on the relationship between diffusion cells. Most notably gaps are not taken into account, as all values in t3 are 1. The value of t3 is one of the five possible options. Since the default for all options of t3 is one, t3 will always be 1. Since t3 handles gap relations between cells, with these defaults, gaps will not impact the diffusion of the dye.

Next, many of the general parameters need to have values;

$porosity = 0.5$ (ratio of void volume),

$\triangle t = 0.0005$ (hours), $\triangle d = 1.75$ (nm).

Properties of the threads need to be defined as well: the default widths of the thread for both the weft and warp are

$thread\_weft = 6$ and $thread\_warp = 6$.

We also need to define a size for gaps $gap\_size = 1$. The maximum size of the thread needs to be defined as well; $width = 200$ pixels and $height = 100$ pixels is a suitable default. It can be the case that the width of the threads do not evenly divide into the width and height. For example, 100 is not evenly divided by 6. This default was chosen on purpose to make sure that functionality addressed in a later section works. The last few parameters are related to the draw loop of the function. The number of iterations to run is $max\_iter = 100$. The number iterations to apply dye is $dye\_iter = 0$. The interval at which the system updates the display and saves a copy of that image in the display is $iteration\_mod = 10$. The final parameter indicates the shape to print: e.g. $shape = circle$.

7.3 Setup Function

The setup function is in charge of initialization of most of the data structures used to construct the fabric. After the fabric is initialized, we must

manipulate it by folding and dyeing, and create structures related to the diffusion equations. Before any of this can occur, we must adjust the width and height to be a multiple of the thread weft/warp. The width and height is updated to be the closest multiple smaller than or equal to the defined number:

$w = w - (w\%thread\_weft\_size)$;

$h = h - (h\%thread\_warp\_size)$;

The updated values are printed to the console.

Now onto the fabric. We instantiated a new Cloth_Model. The called cloth model then creates two layers, which create threads of cloth cells, which sub-divide into diffusion cells. We must also instantiate multidimensional arrays used to store the results of the Ficks second law diffusion equation. After this is called on all diffusion cells, this array is used to update the fabric. This array mechanism is done so that the results of the diffusion equation on one cell does not affect the equation call on later cells. There is one array for each layer, and are indexed by x index for the fabric, y index for the fabric, and color index.

double[][][] rates top;

double[][][] rates bot;

Next we initialize the fold vector. The fold represents the action of taking the fabric data structure and folding over this vector so that the dye applied to one layer will leak through to the other. It is initialized by giving it a linear equation, ax+b which is interpreted into a vector. The fold creates a relationship matrix that gives the corresponding point, if one exists, to any given point over the fold. For example, if the default state is folded over its vertical median (x=100) then point (50,50) has the corresponding point (150,50) in the fold matrix.

Now that the fabric has been created and manipulated, we need to dye it. This is done via the dye function, which will be discussed in more depth below. The basic premise is that it applies a patch of dye to the fabric in the declared shape, the default of which is a circle. This also handles any color mixing that might happen from overlapping shapes of different

colors.

With our fabric folded and dyed, we need to instantiate an Image, which is used by Processing3 to view the diffusion during runtime and to export to a saved .jpg file.

7.4 Draw Function

The draw function is in charge of running and handling all clock based events. It is divided into 3 main sections: the diffusion iteration section, the screen update section, and end of simulation section.

The diffusion iteration sections main job is to run Ficks second law on the entire fabric. It should only do this while the iteration clock is less than the maximum number of runs. First, Ficks second law is run on every diffusion cell in the fabric, in both the weft layer and the warp layer. The results of these runs are stored in the rates arrays. After all the equations are run, the fabric must be updated with these values. Since Ficks second law gives an amount of dye entering or leaving as a positive or negative ratio of a whole, the associated color channel of a diffusion cell can be summed with its corresponding color value in one of the rates matrices. The iteration counter is updated and then, if there is still dye to be reapplied to a location, that section is redyed and the dye iteration counter is incremented.

The screen updates sections main job is to update the screen and save a copy of the current state of the fabric to a .jpg file. This section only triggers if the iteration counter is equal to a multiple of the iteration mod counter (the default is 10). A format string using a naming convention conducive to numerical sorting is made combining the iteration number and the shape. Using this string, save_image is called, and the screen is updated with a processing3 library call to image(). Save image will update the pixel[] array that processing uses for images and then save to a .jpg.

The end simulation section is in charge of creating a final image save and stopping the draw() loop from repeating indefinitely. It only triggers once the maximum number of iterations has been reached and completed.

Save image is once again called, and the process is ended via exit().

7.5 Dyeing The Fabric

The dye function applies a patch of dye in a specified shape to the cloth (the default is a circle). The function currently supports three shapes: a circle, a square, and a right triangle, and takes in an x position, a y position, a size variable, and a color.

The square loops from i=(x to x+size), and j=(y to y+size), and fills the requested color channel (red, green, blue) of the diffusion cells within that range with dye. The function then checks to see if adding color overflows the diffusion cell, and proceeds to normalize those values if so.

The triangle is created in an almost identical manner, the only difference is that the triangle loops from i=(x to x+size) and j=(y+i to y+size).

The circle is created by drawing a line from the center of the circle to all 360 degrees of the circle. To save time, we will create a visited list to prevent channel overwriting and normalization on cells that have already been dyed.

If no shape is specified, an error message is printed and the process is ended.

7.6 Saving the Image

Save image is in charge of creating a .jpg file using an input filename. Processing3 uses a one-dimensional pixel array to represent a two-dimensional image. Doing some simple math, the function loops over every diffusion cell in the fabric and updates the pixel array for the image to be the average of the two diffusion cells at the same index in the weft and warp. The pixel also takes into account rgb color format, which is the default, by having the color channel be equal to:

1-(diffusion-cell-1.color_amount + diffusion-cell-2.color_amount) / 2;

This allows for an empty cell to be white, a cell with all channels filled to be black, and a cell with only red filled to be red.

After the pixel array is updated, we use image.save(filename.jpg) to create an image of that specific iteration stage.

7.7 Folding

The fold function is in charge of populating the relationship matrix that points to the corresponding fold point. At the moment the system only supports a fold over the vertical median. A line is drawn that can be seen before the first iteration is printed to the display screen. After that, all the points are looped over, and for any given point its fold point is found. For a vertical median fold, the corresponding point for an arbitrary point (x,y) is (w-x,y). For example, fold[x][y] gives us the diffusion cell at index[w-x][y], where w is the width of the fabric.

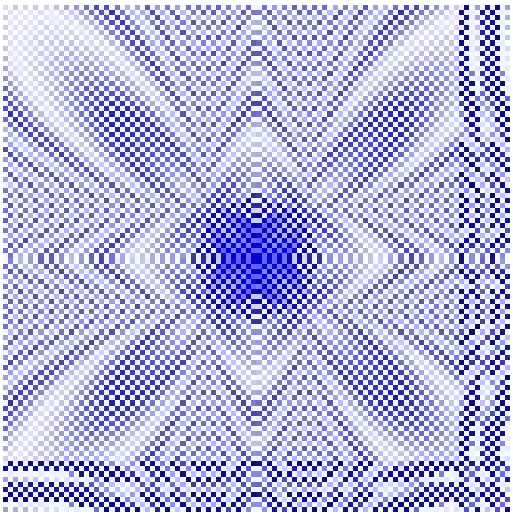
8

Results

This project has been an iterative process in development. Early attempts were not as successful as later ones.

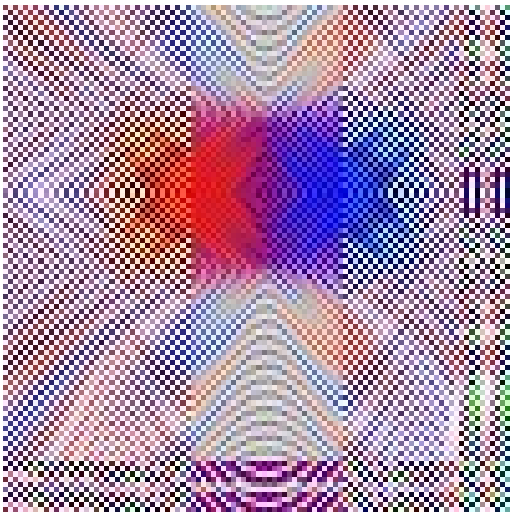
Early on, the parameters were not quite correct. In the following figures, the value for $\triangle d$ is too small, creating extremely rapid diffusion. The artifact in the corner is a result of not properly resizing the fabric during creation, and the system attempts to diffuse over partial threads. These same issues make the color mixing of two squares a psychedelic mess.

Figure 8.1 *Blue Diffuse Incorrect Parameters*



These issues were soon addressed, and $\triangle d$ was brought into a more acceptable value range. The distance between cells has an extreme influence

Figure 8.2 *Color Diffuse Incorrect Parameters*



over the appearance of the final render, and heavily affects how that render appears. Different lengths were experimented with, and their results are as follows:

Figure 8.3 $\triangle d = .005$

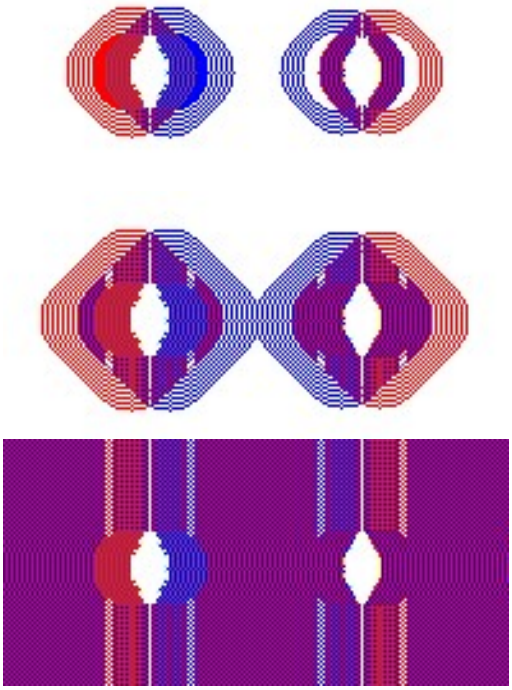


Figure 8.4 $\triangle d = 1$

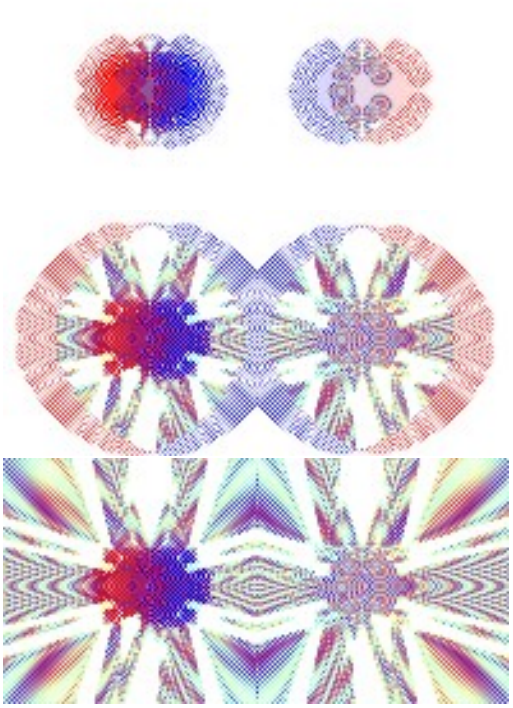
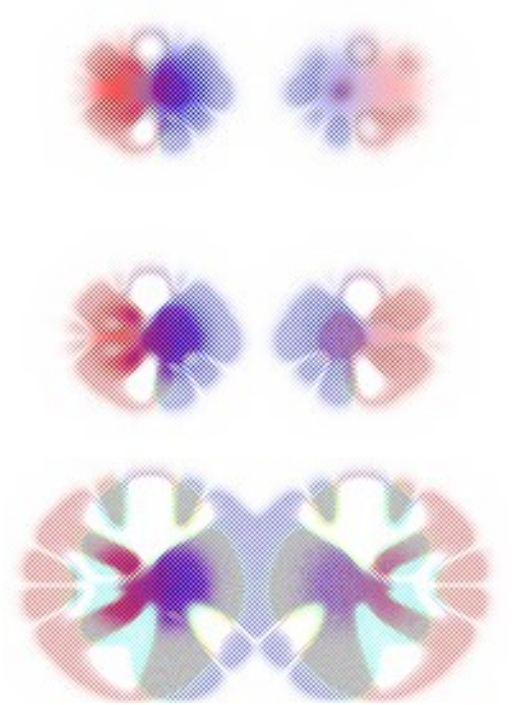


Figure 8.5 $\triangle d = 1.3$



The final value we settled on was 1.75, which gives a nice, simple diffusion without distortion. A similar view as previous for this final result is in the following figure.

Figure 8.6 $\triangle d : 1.75$

In this figure, we can see that our model is working as intended. This model has two circles of different colors applied on the left side, one red and one blue. Where they overlap, the color channels mix, creating a purple. Over time, these circles of dye diffuse slowly outwards, with a very faint ring that denotes the outer edge of diffusion.

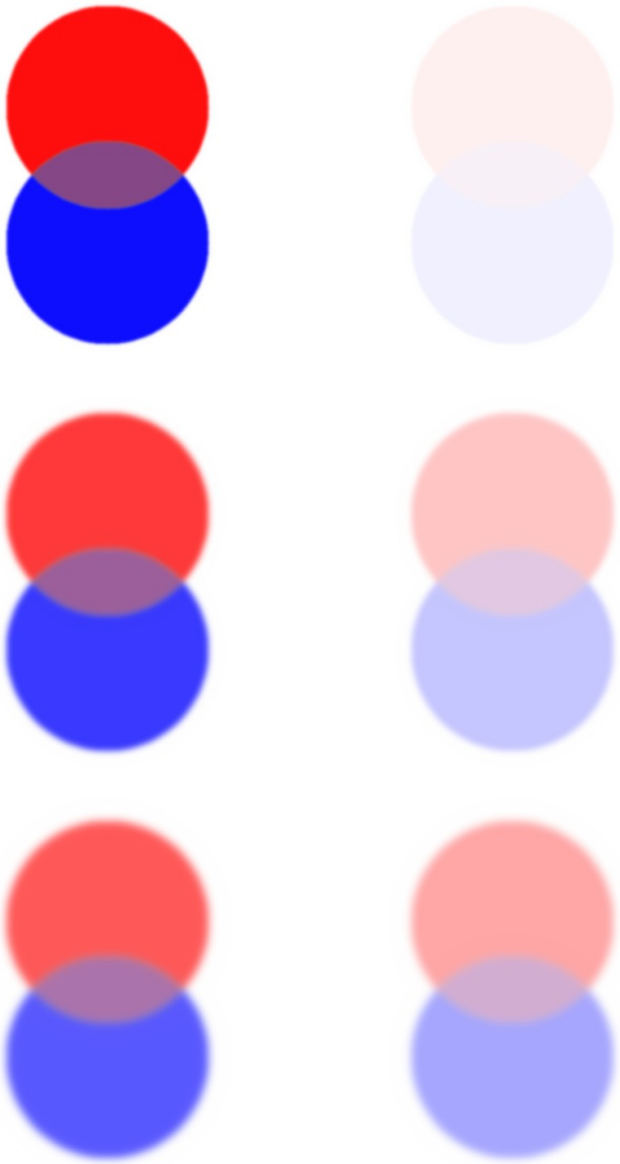
Folding is also working correctly. This fabric was folded over the vertical midpoint, left on to of right, and dye was only applied to the left side (which would be the top of the folded structure). The reflected circle pair on the right side of the renders is a result of that dye diffusing over the fold into the bottom layer of fabric.

Figure 8.7 *Triangle Shape*

Figure 8.8 *Square Shape*

The system also supports two other shape types, a triangle and a square. Like the circle, the can be specified with a width and a starting position. As we can see, both of these shapes also diffuse, mix colors, and traverse folds correctly.

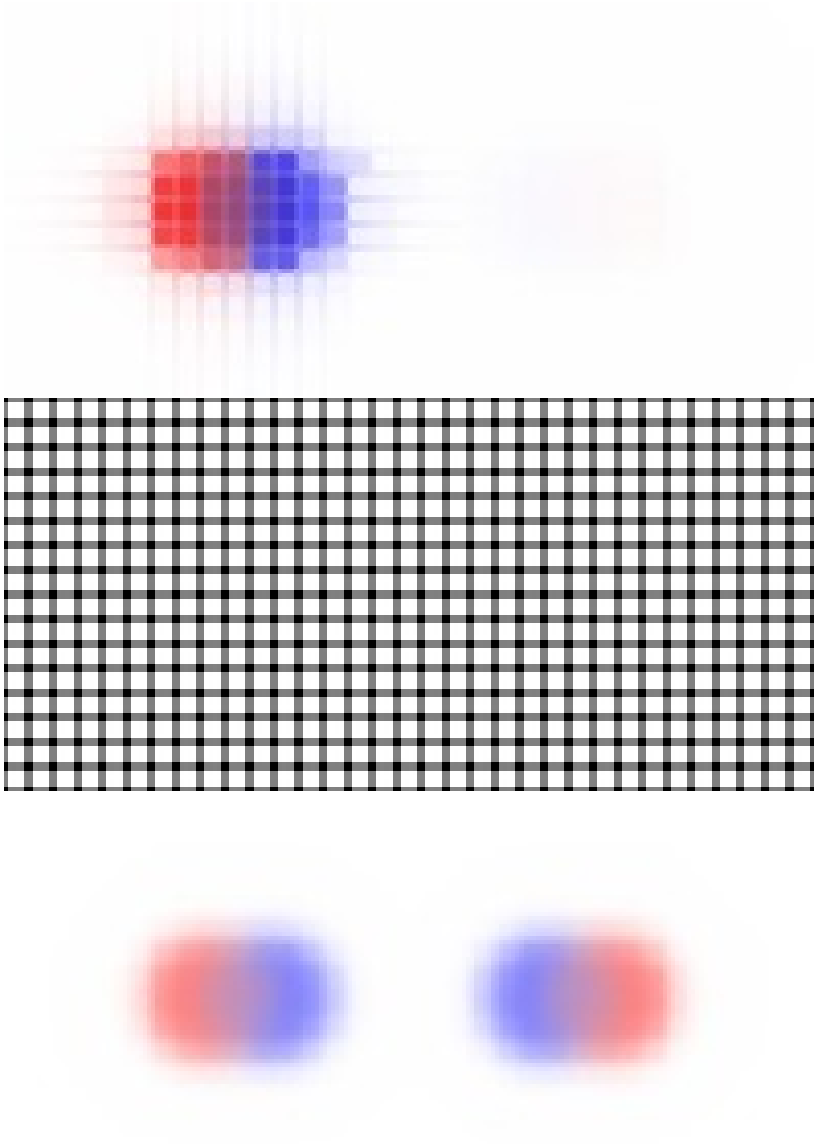
Figure 8.9 *Large Circle and Fabric*



The model was also tested on significantly larger sizes. Figure 8.9 was run on a fabric size 5 times larger than the previous figures, with the patches of dye being increased in size as well.

As we can see, while the model still works correctly, the diffusion is more localized and the overall effect being less impressive.

Figure 8.10 *Modified Tortuosity Values* (1− > .005)



This final figure has all variables the same, except for the tortuosity values. The values of I, II, and III from the third tortuosity relationship, which measures how two different cells are oriented to one another, have been modified. As a refresher, I is the case where two cells are in different layers, II is the case where they are in the same layer and parallel, and III is where one cell is a fiber and the other is a gap. Previously, these specific

tortuosity values were not taken into consideration, as they were all 1.

In this simulation run, these tortuosity values were changed from 1 to .005f. However, since the tortuosity value "I" limits layer diffusion, the dye has trouble diffusing through the layers. If the fold multiplier is adjusted to compensate, the third image is generated.

The middle image is a reference to the location of the gaps in the fiber of this size.

The gaps are clearly visible in this diffusion, as they are now taken into account by III. These renders show us that the tortuosity algorithms working correctly and that gaps are properly implemented in the fabric model.

It appears that our model is successfully simulating the diffusion of dye throughout our fabric.

8.1 Code Repository

All of the code from this paper can be found on a GitHub repository located online at "https://github.com/horst-evans/Thesis-Tie-Dye.git". It is a public repository. We invite the reader to clone or download the repository and make any modifications or improvements that they see fit.

9

Next Steps

While much has been accomplished in this paper, there is still much more to be done. In terms of improving the major systems and components covered by this paper, there are several aspects which we could improve. Much of this would be expanding on existing functions. They include:

1. Allowing for considerably more complex initial dye shapes.
2. Default presets of different weave patterns, such as twill and sateen, to see how these more complex weaves affect diffusion with all else being equal.
3. Apply different base parameters, adjusting values such as tortuosity and porosity.
4. Improving the 3-D manipulation of the fabric from beyond simple folds. Ideally, this would support complex actions such as twisting and pulling up on a fabric, which creates a sort of spiral column shape from the fabric, which is considerable more complex than a fold.
5. Manipulate the properties of the dye itself. As it is right now, the physical properties of the dye is not taken into consideration, beyond the color it would impart to the fabric after it has successful diffused to that section. Factors such as viscosity, concentration, and even molecular structure could all affect how the dye diffuses and adheres to the fabric.

6. The ability to account for pressing elements, such as rubber bands or clamps. Tie-Dye often uses tools like these to help force the cloth to retain its shape, but they also have the dual use of limiting flow. Dye will not readily diffuse over an area that is being squeezed by a rubber band or pressed on by a clamp. From an artistic standpoint this limits the dye to certain areas, preventing mixing, but from a simulation standpoint it affects how the dye diffuses and how the fabric is deformed by these pressing elements.

7. Changing the color system to be a CMYK base. While RGB is nice for displaying color on the screen, real dye operates as subtractive color, instead of additive. CMYK would be more realistic.

8. Taking into account the possibility of threads and gaps stretching or contracting during the dyeing process. This could have subtle effects on the appearance of the dyed cloth.

9. Having more nuanced tortuosity manipulation. Tortuosity can have significant impacts on the appearance of they dye, as we have seen in the results. Fine manipulation of these variables would help create a more accurate fabric model

All of these would make excellent improvements on the current state of the simulation of dye diffusion through fabric and making accurate renderings of Tie-Dye.

10

Conclusion

We attempted to design a method for the simulation of tie-dye fabrics through modeling dye diffusion through fabric in a physics accurate manner. Such an approach requires a complex, interwoven fabric model that is capable of folding and an algorithm for diffusion, in our case Fick's Second Law of Diffusion. As we can see from the results, specifically figure 8.6, our model can accurately simulate real world tie-dye. Diffusion over time appears to be distributing at the correct rate, as is seen also in figure 8.6, and folds appear to be working correctly.

Bibliography

[1] Nelson S.-H. Chu and Chiew-Lan Tai. "MoXi: Real-time Ink Dispersion in Absorbent Paper". In: *ACM SIGGRAPH 2005 Sketches*. SIGGRAPH '05. Los Angeles, California: ACM, 2005. DOI: 10.1145/1187112.1187186. URL: http://doi.acm.org/10.1145/1187112.1187186.

[2] Wendy Copley. *Tie-Dye*. [Online; accessed April 10, 2019]. 2009. URL: https://www.flickr.com/photos/wendycopley/3795762235.

[3] Cassidy J. Curtis et al. "Computer-generated Watercolor". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 421–430. ISBN: 0-89791-896-7. DOI: 10.1145/258734.258896. URL: https://doi.org/10.1145/258734.258896.

[4] Filmedia Everspring. *Weaving Method of Filter Cloth*. [Online; accessed April 10, 2019]. 2015. URL: http://www.best-filter.com/wp-content/uploads/2016/06/Filter-Fabrics-Weaven.png.

[5] Heron2. *Processing Logo Clipped*. [Online; accessed April 10, 2019]. 2006. URL: https://commons.wikimedia.org/wiki/File:Valley-fold.jpg.

[6] JrPol. *Diffusion*. [Online; accessed April 10, 2019]. 2008. URL: https://commons.wikimedia.org/wiki/File:Diffusion.svg.

[7] KDS444. *Plainweave*. [Online; accessed April 10, 2019]. 2013. URL: https://en.wikipedia.org/wiki/File:Plainweave.svg.

[8] T. L. Kunii, G. V. Nosovskij, and T. Hayashi. “A Diffusion Model for Computer Animation of Diffuse Ink Painting”. In: *Proceedings of the Computer Animation*. CA ’95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 98–. ISBN: 0-8186-7062-2. URL: http://dl.acm.org/citation.cfm?id=791214.791456.

[9] Tosiyasu L. Kunii, Gleb V. Nosovskij, and Vladimir L. Vecherinin. “Two-Dimensional Diffusion Model for Diffuse Ink Painting”. In: *International Journal of Shape Modeling* 7 (2001), pp. 45–58.

[10] Kydd. *Warp and Weft*. [Online; accessed April 19, 2019]. 2015. URL: http://heathenhistory.co.uk/wp-content/uploads/2015/12/warpswefts.jpg.

[11] Shiguang Liu and Di Chen. “A Computational Approach to Digital Hand-painted Printing Patterns on Cloth”. In: *Multimedia Tools Appl.* 75.8 (Apr. 2016), pp. 4577–4591. ISSN: 1380-7501. DOI: 10.1007/s11042-015-2492-x. URL: http://dx.doi.org/10.1007/s11042-015-2492-x.

[12] Yuki Morimoto and Kenji Ono. “Computer-generated Tie-dyeing Pattern”. In: *ACM SIGGRAPH ASIA 2009 Posters*. SIGGRAPH ASIA ’09. Yokohama, Japan: ACM, 2009, 36:1–36:1. DOI: 10.1145/1666778.1666814. URL: http://doi.acm.org/10.1145/1666778.1666814.

[13] Yuki Morimoto and Kenji Ono. “Computer-generated Tie-dyeing Using a 3D Diffusion Graph”. In: *Proceedings of the 6th International Conference on Advances in Visual Computing - Volume Part I*. ISVC’10. Las Vegas, NV, USA: Springer-Verlag, 2010, pp. 707–718. ISBN: 978-3-642-17288-5. URL: http://dl.acm.org/citation.cfm?id=1939921.1940001.

[14] Yuki Morimoto and Kenji Ono. “New Cloth Modeling for Designing Dyed Patterns”. In: *ACM SIGGRAPH ASIA 2010 Posters*. SA ’10. Seoul, Republic of Korea: ACM, 2010, 11:1–11:1. ISBN: 978-1-4503-0524-2. DOI: 10.1145/1900354.1900366. URL: http://doi.acm.org/10.1145/1900354.1900366.

[15] Yuki Morimoto and Reiji Tsuruno. “Cellular Modeling of Dye Stain on Cloth”. In: *ACM SIGGRAPH 2004 Posters*. SIGGRAPH ’04. Los Angeles, California: ACM, 2004, pp. 25–. ISBN: 1-58113-896-2. DOI:

10.1145/1186415.1186445. URL: http://doi.acm.org/10.1145/1186415.1186445.

[16] Yuki Morimoto et al. "Visualization of Dyeing Based on Diffusion and Adsorption Theories". In: *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*. PG '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 57–64. ISBN: 0-7695-3009-5. DOI: 10.1109/PG.2007.68. URL: https://doi.org/10.1109/PG.2007.68.

[17] Kaisei Sakurai and Kazuo Matsufuji. "A Procedural Modeling of Woven Textiles with Fuzz". In: *ACM SIGGRAPH ASIA 2009 Posters*. SIGGRAPH ASIA '09. Yokohama, Japan: ACM, 2009, 58:1–58:1. DOI: 10.1145/1666778.1666836. URL: http://doi.acm.org/10.1145/1666778.1666836.

[18] Renzo Shamey, Xiaoming Zhao, and Roger H. Wardman. "Numerical Simulation of Dyebath and the Influence of Dispersion Factor on Dye Transport". In: *Proceedings of the 37th Conference on Winter Simulation*. WSC '05. Orlando, Florida: Winter Simulation Conference, 2005, pp. 2395–2399. ISBN: 0-7803-9519-0. URL: http://dl.acm.org/citation.cfm?id=1162708.1163125.

[19] Chung-Ming Wang and Ren-Jie Wang. "Image-Based Color Ink Diffusion Rendering". In: *IEEE Transactions on Visualization and Computer Graphics* 13.2 (Mar. 2007), pp. 235–246. ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.41. URL: http://dx.doi.org/10.1109/TVCG.2007.41.

[20] Wdwdbot. *Eclipse-Luna-Logo*. [Online; accessed April 10, 2019]. 2018. URL: https://es.wikipedia.org/wiki/Archivo:Eclipse-Luna-Logo.svg.

[21] Woodmath. *Processing Logo Clipped*. [Online; accessed April 10, 2019]. 2010. URL: https://commons.wikimedia.org/wiki/File:Processing_Logo_Clipped.svg.

[22] Songhua Xu et al. "A Generic Pigment Model for Digital Painting". In: *Comput. Graph. Forum* 26 (2007), pp. 609–618.

[23] Tian-ming Yao and Tian-ding Chen. “Black-ink painting synthesis based on brush modeling and ink diffusion algorithm”. In: *Proceedings of 2008 3rd International Conference on Intelligent System and Knowledge Engineering, ISKE 2008* (Nov. 2008). DOI: 10.1109/ISKE.2008.4731104.