Melange:

A Computational Fluid Dynamics Audiovisual Instrument

by

Kurt William Kaminski

UNIVERSITY OF CALIFORNIA

Santa Barbara

Melange:

A Computational Fluid Dynamics Audiovisual Instrument

A Thesis submitted in partial satisfaction of the requirements for the degree

Master of Science

in

Media Arts and Technology

by

Kurt William Kaminski

Committee in charge:

Professor Theodore Kim, Chair

Professor JoAnn Kuchera-Morin

Professor Curtis Roads

June 2017

The thesis of Kurt William Kaminski is approved.

JoAnn Kuchera-Morin

Curtis Roads

Theodore Kim, Committee Chair

June 2017

Melange:

A Computational Fluid Dynamics Audiovisual Instrument

by

Kurt William Kaminski

ACKNOWLEDGEMENTS

I would like to thank the members of my committee for their support and insight during my time at MAT.

I would also like to thank:

Members of the AlloSphere Research Group and the Graphics Group including Keehong Youn, Pablo Colapinto, Tim Wood, Kenny Kim, Hannah Wolfe, Karl Yerkes, and Andrés Cabrera for their inspiration and support both creatively and technically regarding computer graphics and digital audio,

Juan Manuel Escalante for his creative wisdom and thoughtful critique,

Mark Hirsh for assisting with video documentation,

My friends, family, and loved ones for their endless support, encouragement, and patience,

The artists and scientists who came before me for which this work would otherwise not be possible.

ABSTRACT

Melange:

A Computational Fluid Dynamics Audiovisual Instrument

Kurt William Kaminski

The convergence of GPUs and spatial sensors fosters the exploration of novel interactive experiences. Next generation audiovisual synthesis instruments benefit greatly from such technologies because their components require significant computing resources and robust input methods. One technique that shares these requirements is physical simulation. The expressive potential of real-time physical simulation is rarely used in the domain of visual performance.

This Masters document describes Melange, an audiovisual instrument that maps gestural input to a highly evocative real-time fluid dynamics model for synthesizing image and sound. Using general-purpose GPU computing and a structured light depth sensor, different visual and sonic transformations of fluid flow are explored as an interactive computational substance.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

The convergence of GPUs and spatial sensors fosters the exploration of novel interactive experiences. Next generation audiovisual synthesis instruments benefit greatly from such technologies because their components require significant computing resources and robust input methods. One technique that shares these requirements is physical simulation. The expressive potential of real-time physical simulation is rarely used in the domain of visual performance.

This Masters document describes Melange, an audiovisual instrument that maps gestural input to a highly evocative real-time fluid dynamics model for synthesizing image and sound. Using general-purpose GPU computing and a structured light depth sensor, different visual and sonic transformations of fluid flow are explored as an interactive computational substance.

1.1 Motivations and Significance

"If, when a musical instrument sounds, someone would perceive the finest movements of the air, he certainly would see nothing but a painting with an extraordinary variety of colors." [15].

The motion of liquid and gas has inspired human creativity for thousands of years. Prehistoric *Homo sapiens* produced throwing spears some 400,000 years ago [30], suggesting an empirical relationship with wind resistance. With agriculture came efforts to control the flow of water, and around the same time, the first boats were constructed [20]. Our efforts to manipulate and understand fluid flow is fundamental to human civilization.

Leonardo da Vinci was captivated by fluid motion, making several detailed drawings of flow and becoming the first to articulate flow visualization—lines representing the movement of particles in a fluid over time. Vincent van Gogh and Hokusai explored powerful and expressive representations of fluid mechanics in their work. Many performance artists incorporate fluid dynamics into their work.

The expressive potential of fluid simulation is rarely used in the domain of audiovisual performance, however. This is primarily because physical simulation is computationally expensive and difficult to control. Implementing accurate fluid mechanics involves solving many large differential equations. Jos Stam's paper [29] on real-time fluid simulation is nearly 20 years old, yet remains one of the most efficient ways of making interactive fluids on the computer. Fluid flow is highly variable, so often an artist will want to restrict its behavior in some unique or explicit way. Doing so interactively and intuitively introduces additional complexity to both the interface and the simulation. The problem ultimately lies in producing and controlling simulated fluid material in real-time.

1.2 Design Goals

In order to build an expressive tool based on fluid simulation, we adapted design criteria from Golan Levin [17], Lance Putnam, and JoAnn Kuchera-Morin's [27] research on audiovisual instruments.

First, sound and image need to be created together in real-time. This is perhaps the most important design criteria because it is how we see and hear the work. Highly responsive, low latency output is crucial for live performance.

Second, the physical simulation should be the basis for sound and image. The reason for this is twofold: to capture the interesting and often surprising dynamics of a physical substance, and to encourage the development of a system in which sound and image are equally malleable. Levin conveys the importance of audiovisual instruments whose sonic and visual components do not overwhelm each other. Striving for equal amounts of control over each component leads to expressive performances in each domain.

Third, the control interface needs to be instantly knowable yet indefinitely masterable. In the same way that striking a piano key allows anyone to understand what a piano does, its accessibility does not preclude it from complex expressions.

Finally, to foster a meaningful link between the user and the material, human gestures will map to physical parameters in the simulation. As our bodies are complex and imperfect systems, movement patterns can lead to interesting, spontaneous results when coupled with a dynamic medium. The subtle adjustments by, for example, a horn players lips alters the air in distinct ways that imparts personality and emotion into the work. By mapping gestures to simulation parameters, we hope to evoke similar amounts of depth in the visual and sonic products of the instrument.

2. Background and Related Work

2.1 Visual Music

Visual music, also known as color music, ocular music, or music for the eyes, is the tradition of synchronizing abstract images and sounds. The history of visual music is long and varied. A complete look at its development is beyond the scope of this paper. For a more detailed analysis of the history of visual music, see [17]. We instead will focus on visual music examples that support real-time creation of audiovisual material. Many of these can be considered audiovisual instruments with robust input methods and expressive output. Some however, such as Mary Hallock Greenewalt's *Visual Music Phonograph*, and Kurt Hentschläger and Ulf Langheinrich's *Model 5*, do not have sophisticated controls, but nonetheless synchronize sound and image in real-time. This section is organized into two parts: visual music in the pre-computational era, and visual music after the proliferation of digital computing. Computational methods offer many benefits to the artist with respect to timing and non-linear editing of content, so it is prudent to identify visual music efforts before such affordances.

2.1.1 Pre-computational Visual Music

The first recorded instance of synchronizing abstract imagery with music was by Louis-Bertrand Castel and his *clavecin oculaire*. Beginning with very primitive working examples, the final iteration was a massive 144 key harpsichord—twice the size of a grand piano—that raised individual shutters to expose colored panels of tissue paper to candlelight [25]. There are no surviving diagrams of his color organ. Castel, inspired by Kircher's *Musurgia Universalis*, was less eager to construct working examples himself and more interested in the theory of relating tones and colors. Castel thought that marrying the two would reveal a hidden order in the universe where communication between sonic and visual senses were fluid, allowing a deaf person to enjoy music by seeing [25].

Others were interested in an absolute audiovisual relationship. Isaac Newton, in 1704, published the first edition of *Optiks*—his theory of light—which proposed seven colors that coincided with seven musical notes and seven days of the week [2]. In 1893, painter Alexander Rimington created a color organ that could control the quality of light through a church organ-like interface. He too believed the optical spectrum directly mapped to musical notes. Rimington imagined converting the standard repertoire would be performed in color and that one day musicians would begin to write color and music scores in tandem [28].

In the 1900s, electronics allowed for precise, synchronized control systems to be developed. Mary Hallock-Greenewalt's *Visual Music Phonograph* used a phonograph record to control lights. Thomas Wilfred's *Clavilux* used a handheld controller to manipulate colored glass discs illuminated by filtered light.

With the proliferation of film media, cinema was developed as a language to unfold image and sound over time to tell stories. Experimentation with the physicality of film and its relationship with audio was advanced by Oskar Fishinger, Norman McLaren, and Daphne Oram. These artists all studied the effect of marking directly on tape to alter its audiovisual output in unexpected, but nonetheless coordinated, ways [18]. It is here where audiovisual content begins to emerge from a single material or process rather than from a fusion of two separate actions. The power of electronics for audiovisual work lies in its ability to decode sound and image simultaneously, leading to increasingly sophisticated material and more intimate levels of control over it.

2.1.2 Computational Visual Music

Toward the end of the 20th century, computation came to dominate the audiovisual design process. Contemporary visual music examples use digital computing to synthesize and manipulate new kinds of audiovisual material. Here, there is a shift from 18th century notions of a true link between sound and color to evoking sound and color from a shared substance.

At the same time, new methods of controlling the material were developed. For example, the human body became an input device in Myron Kruger's *Videoplace* (1975) and David Rokeby's *Very Nervous System* (1986). Kurt Hentschläger and Ulf Langheinrich used the electronic music technique of granular synthesis in live video performance for *Model 5*.

Golan Levin's *Audiovisual Environment Suite* is a collection of 5 works that attempt to satisfy his design criteria for audiovisual instruments, some of which were adopted for Melange [16]. Interestingly, all the instruments had their shortcomings. No one instrument stood out as perfectly embodying the principles he laid out for himself. One of the instruments, *Floo*, implements a low-resolution fluid dynamics simulation. But this was before general purpose GPU computing, so its input methods and overall fidelity were fairly limited compared to today. While these instruments may have aged, his efforts to classify audiovisual systems are of lasting importance.

In 2016, JoAnn Kuchera-Morin, with collaborators Lance Putnam and Luca Peliti, began performing *The Hydrogen-Like Atom*, a synthesizer that can visualize and sonify quantum equations and information. It was built for the AlloSphere, a 10-meter diameter data visualization capsule with a 54-speaker audio system. *The Hydrogen-Like Atom* is controlled by a MIDI keyboard and multitouch video display.

2.2 Fluid Dynamics

2.2.1 A Brief History of Fluid Dynamics

The scientific study of fluid dynamics started with Archimedes and his work on hydrostatics around 250 BC. Significant progress was stalled for almost two thousand years until Leonardo da Vinci found the least resistive “streamlined” shape [21]. The mathematical description of fluid mechanics began with Galileo in the 1600s. Many mathematicians began working on the problem after that, with calculus providing the tools necessary to describe fluid motion. Eventually, in the 1840s, Claude-Louis Navier and George Gabriel Stokes produced the Navier-Stokes equations which accurately model viscous fluid. Unfortunately, numerical implementations in the 1900s from mathematicians such as Geoffrey Taylor were driven by the race to develop atomic bombs [6]. To the author’s knowledge, we have managed to avoid such applications with Melange.

2.2.2 Fluid Dynamics in Art

Fluid dynamics has been a major influence on human creativity and art. The Chauvet Cave in southern France is one the most significant prehistoric art sites ever discovered, with dozens of detailed paintings made 35,000 years ago in complete darkness. Fire, a textbook example of fluid motion, was required to illuminate the works. The noisy flicker of flames and the shadows they produce are thought to have played a role in the compositions [10].

A very small sampling of art over the last century reveals widely varying interpretations and invocations of fluid dynamics. Leonardo da Vinci made several detailed drawings of the behavior of water around objects which became the first examples of flow visualization. Da Vinci was a strong advocate for the study of fluid motion [5]. Vincent van Gogh's "Starry Night" features plausible turbulent fluid structures that play a central role in the piece [32]. Iannis Xenakis used the statistical mechanics of gas molecules as a compositional tool in "Pithoprakta" [4]. Tony Martin was part of a movement of "liquid light" audiovisual concerts mixing fluids on an overhead projector. Martin did the first light shows for The Grateful Dead, Jefferson Airplane, and Morton Subotnick [22]. Peter and Chris Parks used high speed macro photography of fluid reactions as key elements in Darren Aronofsky's film "The Fountain" [24]. Finally, Ned Kahn invokes fluid flow as a kinetic, sculptural medium in many of his large-scale installations [13].

2.2.3 Fluid Dynamics in Contemporary Media Art

Computational fluid dynamics as a creative material is a relatively recent development. At this point, it is worth mentioning visual effects for film and computer animation. The visual effects industry is a notable example of widespread creative use of fluid motion for many elements that are crucial to modern, multi-million-dollar blockbuster films. These can include things from dust kicked up from a character running across a field to explosions, crashing waves, clouds, and even the characters themselves. While much work has gone into increasing the believability of these elements and to providing artist control over them, their output is strictly visual and far from real time. It is not uncommon for a simulation to take 24–48 hours to calculate for ten seconds of footage.

For examples of real-time uses of fluid dynamics, we turn to experiential installations. Memo Akten mapped a user's body movements to velocity and color inputs in his projection piece "Body Paint" [1]. Vincent Houzé projected abstract, non-realistic renderings of fluid dynamics into a smoke-filled installation space in "Lull" [11]. In 2017, John Gerrard rendered a smoke simulation that formed the shape of a flag and composited it onto a live broadcast of an empty field in Texas for his piece "Western Flag" [8]. This field was the site of the first major oil discovery in history, symbolizing our continued dependence on oil. Importantly, none of these works generated audio from fluid dynamics.

The author's own work involves instrumentation and natural processes. In 2014, he created a MaxMSP instrument that used the Milankovitch cycles to produce sound. Milankovitch cycles attempt to model how much energy the Earth receives over a period of about 50,000 years due to its irregular orbit. In 2015, in collaboration with Juan Manuel Escalante, we created Lukidus. Lukidus used a biological microscope modified to be controllable from a computer to perform a variety of phenomena including crystal growth and mixing fluids with different densities.

3. Design

Melange can be subdivided into three main components:

- Interaction
- Computational Substance
- Visualization and Sonification

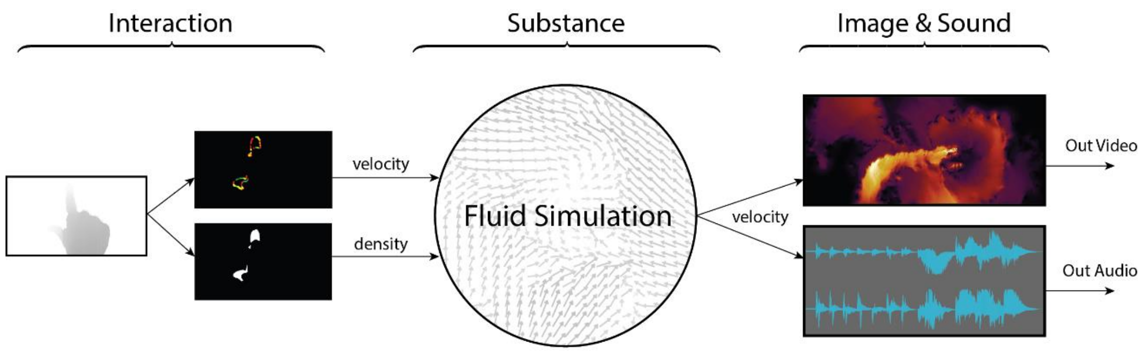

Figure 1. Simplified data flow schematic for Melange.

A depth image from a depth sensing camera is first transformed into velocity and density fields which are injected into fluid simulation. From the fluid's velocity, images and sounds can be produced. Although the graph in Figure 1 shows data flowing linearly, there are instances where data flow is non-linear, such as introducing feedback from sonification as velocity impulses into the fluid simulation. This is explained in detail on page 29.

This chapter will first discuss the nature of fluid simulation and how it is achieved in Melange. In doing so we will introduce some important terms and concepts that lead to an understanding of how the fluid is visualized and sonified. Finally, with knowledge of what the system can do, we will describe how to control it and modify its parameters.

3.1 Tools

Melange was made almost entirely within TouchDesigner. TouchDesigner is a visual programming application for Windows and Macintosh operating systems used in live performance and installations. It wraps OpenGL structures into nodes that are patched together, not unlike programs such as MaxMSP or PureData. TouchDesigner supports execution of C++ plugins and native OpenGL shader code.

C++ was used to write a custom plugin to retrieve depth map information from the RealSense SR300 depth camera. It returns a 32-bit floating point texture at a framerate specified by the user, depending on resolution. The resolution and framerate in Melange is 640 x 480 at 60 frames per second. The plugin also allows for control over parameters specific to the SR300 such as accuracy, laser projector power, filtering options, and motion-range tradeoff.

GLSL is a shading language that can be used for parallel programming by leveraging graphics primitives such as texture buffers. GLSL is very portable, with support on all popular devices including smart phones. Shaders do the heavy computational lifting of Melange. Despite Melange existing in TouchDesigner, its main functionality could easily be adapted to any framework with OpenGL and GLSL support.

Finally, Python was used to glue various parts of the system together. TouchDesigner has a Python front end with every node wrapped in a Python class. It also allows for scripting using Python. Some examples of how Python is used in Melange include passing MIDI information to different parts of the system, generating perceptual ramps, and converting notes to frequencies.

3.2 Fluid Simulation Methodology

3.2.1 Fluid Flow as a Computational Substance

Fluid flow is a useful substance on which to base audiovisual material. Flow can be described by a field of velocity vectors. Velocity is neither sound or image, but can be transformed into them in different ways. As described earlier, producing both domains from a shared source helps to ensure that the sound or the image do not take precedent over each other.

Fluid dynamics also has a large parameter space. For example, the rate of diffusion, amount of curl, and boundary conditions all lead to very different results. Since velocity is a field in multiple dimensions, most of these parameters can be spatialized.

Turbulence provides infinite variability. The chaotic changes in pressure and velocity in fluids ensures constantly varying, dynamic evolution through time.

3.2.2 Navier-Stokes Equations

The Navier-Stokes equations describe viscous fluid motion over time. They were discovered in the 1840s and are still used today to model the weather, ocean currents, water flow in a pipe, and air flow around the wing of an aircraft. They aid in automotive and aircraft design, cardiovascular study, air pollution assessment, and many other fields.

$$\underbrace{\frac{\partial u}{\partial t}}_{\text{Variation over time}} = \underbrace{-(\mathbf{u} \cdot \nabla)\mathbf{u}}_{\text{Advection}} - \underbrace{\frac{1}{\rho}\nabla p}_{\text{Pressure}} + \underbrace{\nu \nabla^2 \mathbf{u}}_{\text{Diffusion}} + \underbrace{\mathbf{F}}_{\text{External force}}$$

$$\underbrace{\nabla \cdot \mathbf{u} = 0}_{\text{Continuity}}$$

Figure 2. The Navier-Stokes equations.

An important term to point out is advection. Advection is the process in which a fluid's velocity transports both other quantities, and itself, in the fluid. If ink is dropped into a bowl of water, we can say that the ink is advected along the water's velocity field. In this equation, the advection term represents the velocity pushing itself around just as it pushes the ink.

Pressure is an acceleration term that represents the build-up of forces across the field. Pressure is responsible for making the swirls that are so common in fluids, and can be thought of as the thing that makes fluids "slosh" around. Diffusion describes how resistive a fluid is to flow, or in other words, its viscosity. A relevant example is vegetable oil as opposed to syrup—the syrup has a lower diffusion rate, or higher viscosity, than the vegetable oil.

Finally, the external force is where we can add velocity from other objects. This is where user interaction can affect the system.

The second equation is the continuity equation and ensures the conservation of mass. It is coupled to the first equation though the pressure term and is responsible for creating vortices. These equations assume incompressibility, which means that the volume of any subregion of the fluid is constant over time. The continuity equation factors in divergence, which is the net change in velocity across an area surrounding a point in the field. Ensuring the fluid always has zero divergence enforces incompressibility.

3.2.3 Grid-Based Implementation

The way that we implement the Navier-Stokes equation is along a grid. The grid assumes that a particle of fluid exists at the center of every cell. Instead of pushing a particle to a new position, we update the particle's velocity using the implicit method developed by Jos Stam [29].

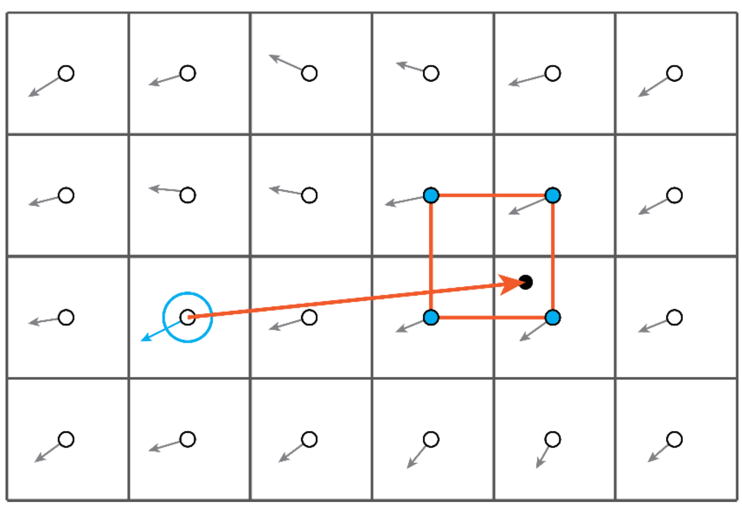

Figure 3. Grid-based advection.

Using its current velocity, we "trace" the particle back in time to find where it would have been one frame ago, drawn in the figure as the long arrow pointing to the black dot. This previous position would likely not fall directly on the center of a cell, but somewhere between them, so we take the four closest points and interpolate them to get the new velocity. This approach ensures that the simulation is stable, even for very large time steps.

In addition to being stable, Stam's method can be implemented on the GPU, making it very computationally efficient [9]. This allows for high fidelity simulations in real-time.

At any point, obstacles or boundaries can be added or removed from the simulation. Interaction with boundaries changes when the influence of pressure is modified.

A number of fields are advected in addition to velocity. Density is a field that tracks some local amount of material as it moves around and dissipates in the fluid. Color can be treated the same way by advecting each red, green, and blue channel of an image together, which is discussed on page 23.

Buoyancy attempts to model velocity currents that are a result of the interaction between density and an additional field, temperature. These currents can be seen in the weather, oceans, and things like coffee.

Finally, vorticity simply means the amount of *rotation* around a point in the fluid. Due to the grid-based simulation method, small scale rotations and the interesting structures that come with them can be lost. *Vorticity confinement* is a technique for restoring these finer motions [7].

3.3 Visualization

3.3.1 Velocity Ramps

Velocity may be visualized in many ways. One of the easiest and most direct ways of seeing the velocity field is by mapping its magnitude, or speed, to a color ramp.

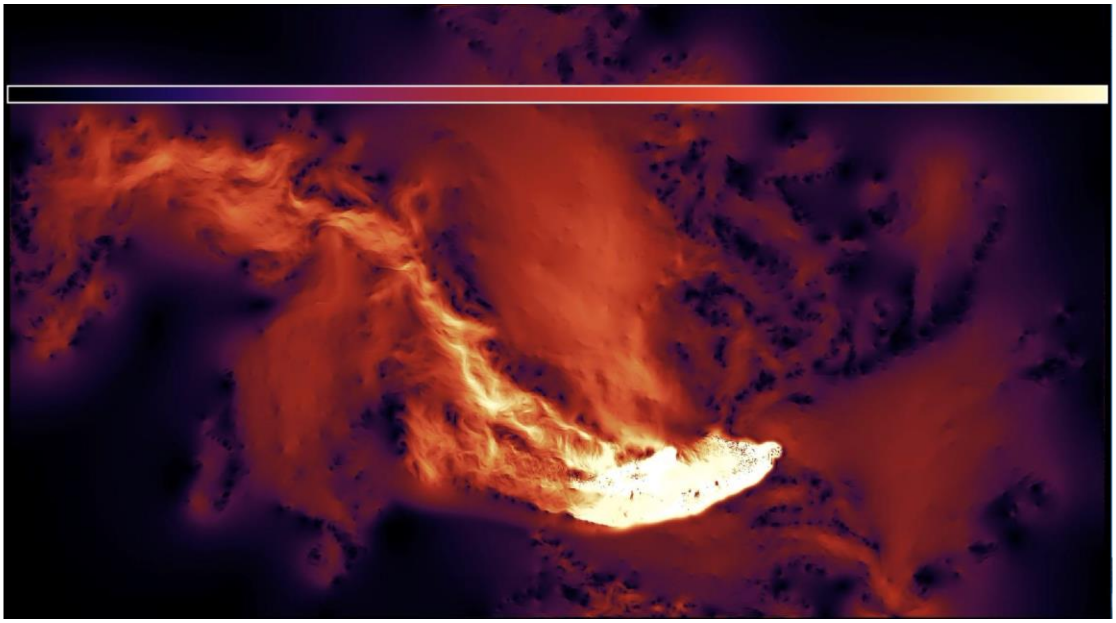

Figure 4. Velocity color ramp with source gradient.

The ramp used in Figure 4 is shown at the top of the image. Bright, warm colors represent faster velocity, and dark, cool colors represent slower velocity. It is important to note that the color ramp can vary in brightness, hue, and saturation, but if there is a sufficient difference in the values across the ramp, it is still possible to gain an understanding of how fast the fluid is moving in different areas.

The different color ramps that are used in the project and how they are selected by the user are explained on page 38.

3.3.2 Color Field

A field of solid colors can be advected and directly visualized as well, like the ink-in-water example mentioned before.

Figure 5. Advecting RGB color channels.

Figure 5 depicts three color channels—red, green, and blue—being advected together in the fluid. By representing each color channel as an individual field, it is possible to achieve mixing effects and painterly results.

Each color channel is a signed 16-bit floating point value, so it is possible to deposit negative color values into the field. This has the effect of aggressively darkening the field while increasing local saturation. Since depositing color is an additive process, negative values can be a useful performance and compositional tool to prevent the image from washing out or becoming too bright.

3.3.3 Particle Advection

Velocity can also be abstracted and not directly mapped to an image for an implicit visualization. With particle advection, hundreds of thousands of points are deposited into the fluid. They inherit and interpolate values from the velocity field, moving along with its currents. Their collective movement highlights the flow structures.

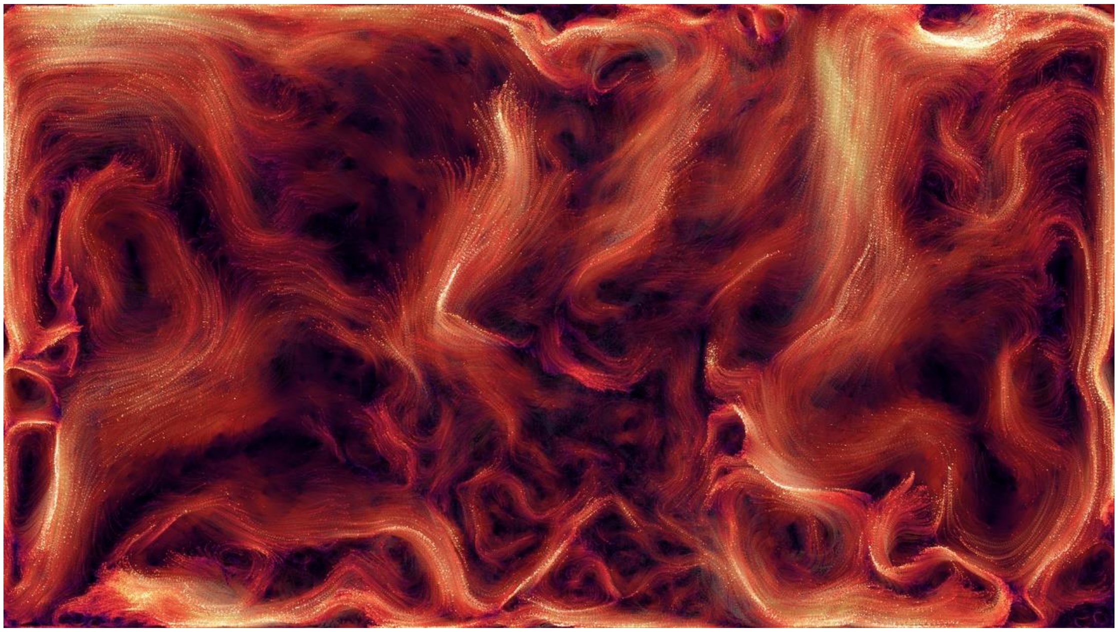

Figure 6. Particle advection with flow lines.

Figure 6 shows particles being advected in a fluid field. Their color is set using the same color ramp used in Figure 4 to visualize magnitude, only now it is applied to each individual particle.

Particles can be considered massless, uniform bodies. These particles tend to aggregate around strong currents and result in sharp, defining flow lines. Particles can also be given mass and drag to alter their behavior. This makes some particles more resistant to flow than others, so their distribution in the field becomes more varied than massless particles.

Finally, the path of each particle may be traced by not clearing the final image with each rendering. In Figure 6, the particle paths have been traced with trails that fade out over time. This produces flow lines similar to what da Vinci was depicting in his studies of flowing water sketches.

3.3.4 Geometry Instancing

We can also instance geometry onto the simulation grid and modulate it based on properties of the velocity field.

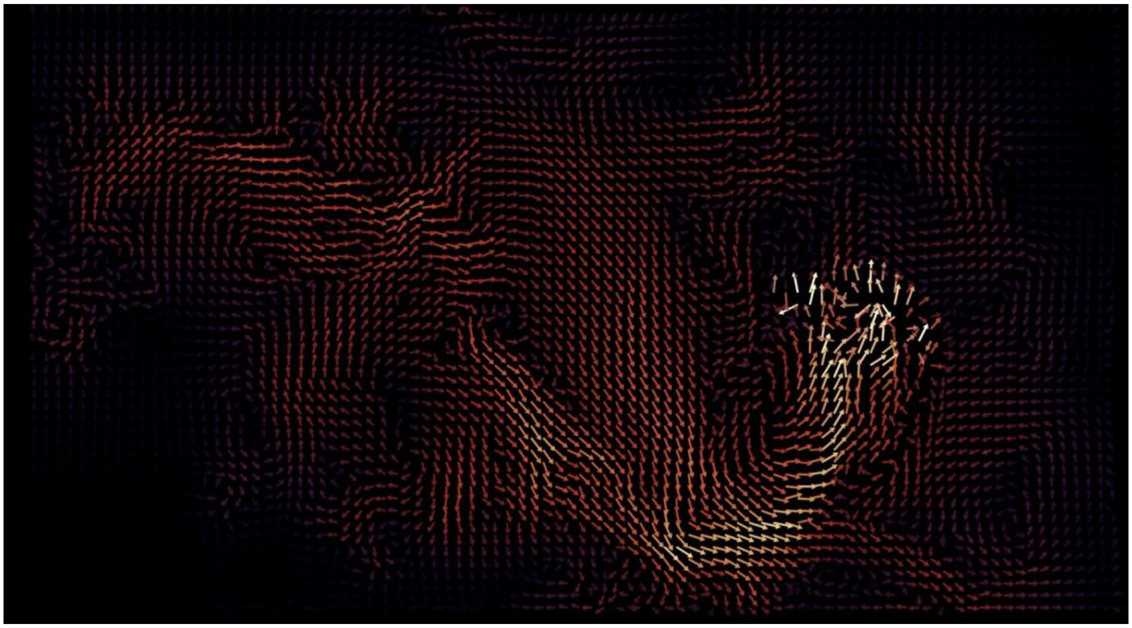

Figure 7. Instancing geometric arrows.

In Figure 7, three dimensional arrows made from elongated cubes and cones are copied onto the velocity grid. They rotate in the direction of their underlying cell's velocity, and scale to be longer based on the magnitude of the velocity. The velocity grid has been downscaled by a 16^{th} of its original resolution. If the original resolution was used, the arrows would cover the image so densely it would be an indecipherable mass of color, or the arrows would be so small that a much larger resolution would be required to render them all at a reasonable scale.

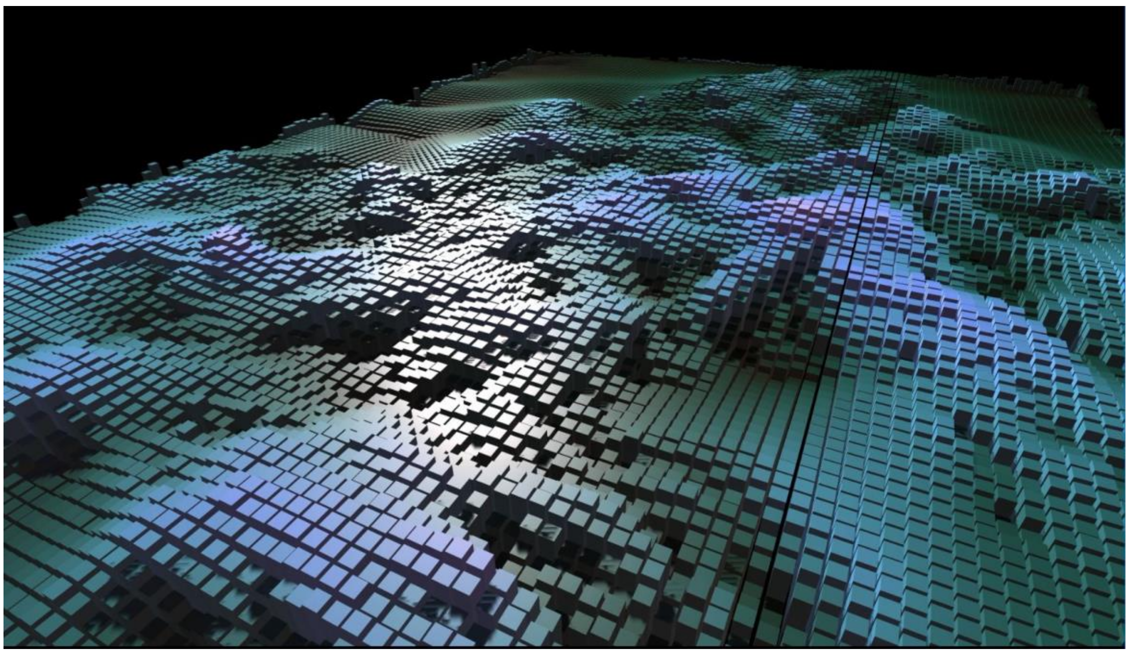

Figure 8. Instancing boxes.

Figure 8 depicts another style of rendering with boxes instanced onto each grid cell. Their height scales with the speed of the velocity field. The scene is rendered with lighting, so the longer shadows from the taller boxes indicate areas with greater differences in velocity and generally higher turbulence.

3.4 Sonification

3.4.1 Raster Scanning

Like visualization, the velocity field may be sonified in many ways. One technique we implemented was raster scanning. This reads the velocity field line-by-line, treating each cell value as an audio sample. The horizontal velocity component was put into the left channel, and the vertical component in the right channel. This gives us direct and highly responsive sonification of fluid flow that covers the whole field.

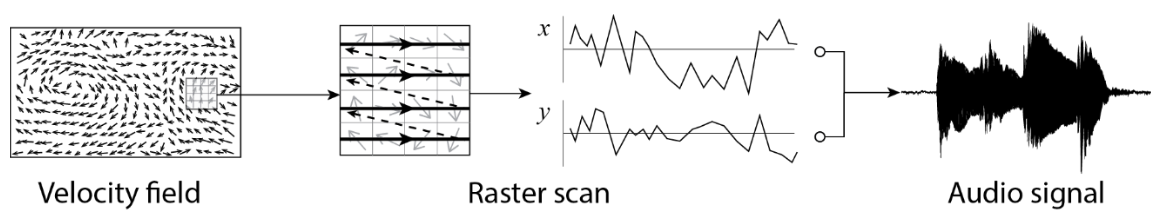

Figure 9. Raster scanning.

In Melange, the velocity field is scaled down by an eighth to maintain real-time data transfer from the GPU to system memory. Recall that all velocity field calculations are executed on the GPU through texture buffers, so it is necessary to copy these values to system memory for audio processing.

Because the user may wish to choose a different resolution of the fluid field, it is necessary to consider the length of the resulting 1d array to maintain consistent acoustic results when scanning the field. TouchDesigner's Audio Oscillator node expects logarithmic pitch control—a value of one will increase the pitch by one octave. To do this, we divide the size of this array by the sample rate, then divide the desired frequency by this number, then take the binary logarithm of that to produce the logarithmic pitch value. Frequency and amplitude are selected by the user in different ways which is discussed on page 36.

A downside to raster scanning is that the waveform can be very noisy due to a lot of local variation in the velocity. Even after downscaling the velocity field—and interpolating it in the process—the waveform is extremely noisy. This makes timbre difficult to shape and manipulate.

Another reason the waveform can be noisy is due to raster scanning a fluid field with confined boundary conditions. As the scan reaches the end of one row of cells, the beginning of the next row is appended to the end of the previous row. With confined borders, these cells have nothing to do with each other spatially. This can be avoided by using periodic boundaries, however.

Since the velocity field is a texture buffer, image operations may be performed on the field before it is read to system memory. Blurring the velocity image removes noise from the resulting sound. It produces a similar acoustic effect as setting a cutoff on the magnitude of the frequency spectrum, as per Ryan McGee's FFT cutoff filter used in "Voice of Sisyphus" [19].

It is possible to adapt raster scanning to behave like a waveshaper by scanning over the waveform at different rates, multiplying them by different values, and averaging them together. This can produce sounds like an electric guitar.

3.4.2 Scanned Synthesis

Raster scanning heavily relies on post processing to produce musical sounds. As per this project's design goals, we were more interested in the effects of physical simulation than chaining post processing effects together. Were there other sonification models that could be used to help evoke more musical qualities from the velocity field?

Scanned synthesis reads a slowly transforming wavetable at audio rate to produce sound. In Melange, the wavetable is a spring-mass simulation of a one-dimensional string that is agitated by the fluid simulation. The difference between the string's rest state and its current position becomes the waveform which is scanned at audio rate. Frequency and amplitude are selected by the user in different ways, which are discussed on page 36.

Scanned synthesis was first described by Bill Verplank in 1999 [33]. The author was unaware of Verplank's work and arrived at this technique independently, inspired by conversations with JoAnn Kuchera-Morin and her work on "The Hydrogen Atom".

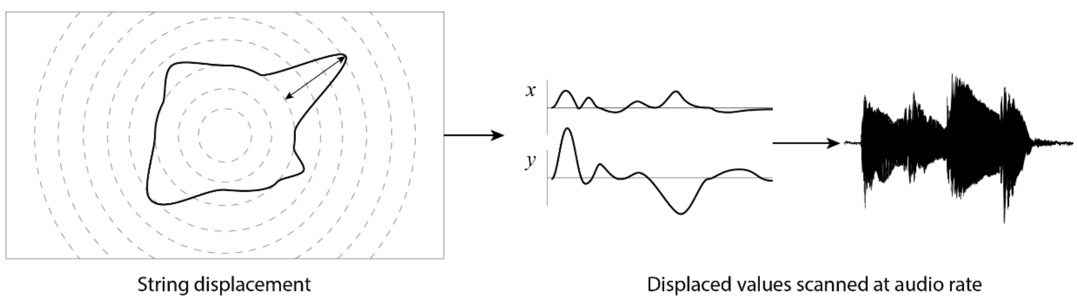

Figure 10. Scanned synthesis.

The string model is a one-dimensional connection of masses and springs. In Figure 11, M is a spring mass, x is the position, C is the spring connecting the mass to its rest state, D is damping to its rest state, and f is the force of the mass. Since our strings are circular, M_0 connects to M_N. It is prudent to note that the spring simulation is implemented on the GPU via GLSL fragment shaders, making it very computationally efficient. The width of the texture that the spring simulation operates on is the number of samples in each string and the height is the total number of strings. In Melange, this equates to a 512 x 12 texture buffer.

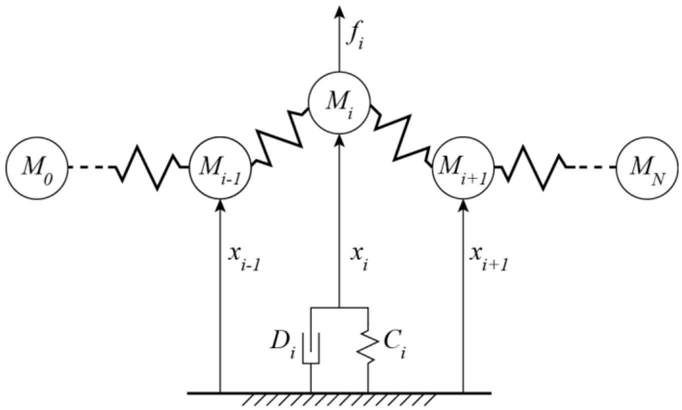

Figure 11. Spring-mass model.

Scanned synthesis provides many more tools to shape the sound compared to raster scanning. Parameters for controlling how tight or loose the springs are can produce different responses such as plucked or bowed sounds. It is also possible to choose what rest shape the strings take. In Melange, concentric rings emanating from the center of the field were chosen. The rest shape could also be a series of straight lines like a guitar or harp. They could take on any unusual shape. A complete taxonomy of string rest shapes and their varying sonic qualities is outside the scope of this project, but the author encourages this work to be explored. Rings provide the benefit of smoother waveform shapes because samples at the beginning of the waveform will meet up with samples at the end of the waveform. This helps avoid noise.

Finally, another interesting feature of this implementation of scanned synthesis is that the velocity of the springs may be rendered as a two-dimensional image and fed back as a velocity impulse to the fluid simulation. This results in feedback effects that, when balanced by the user as to avoid exponential increases in velocity, can produce desirable visual and sonic results.

3.5 Interaction

Musical instruments offer an ideal interface for performance of not only sound, but audiovisual material as well. Musical instruments usually possess these features which are conducive to expressive output [12]:

1. There is interaction with a physical object.
2. Coordinated hand and finger motions are crucial to audio output.
3. The sonic reaction is instantaneous.

We used these features to guide interface decisions made for Melange.

3.5.1 Hardware Controllers

Figure 12 shows the two controllers used in the project— a depth sensing camera and a small MIDI controller.

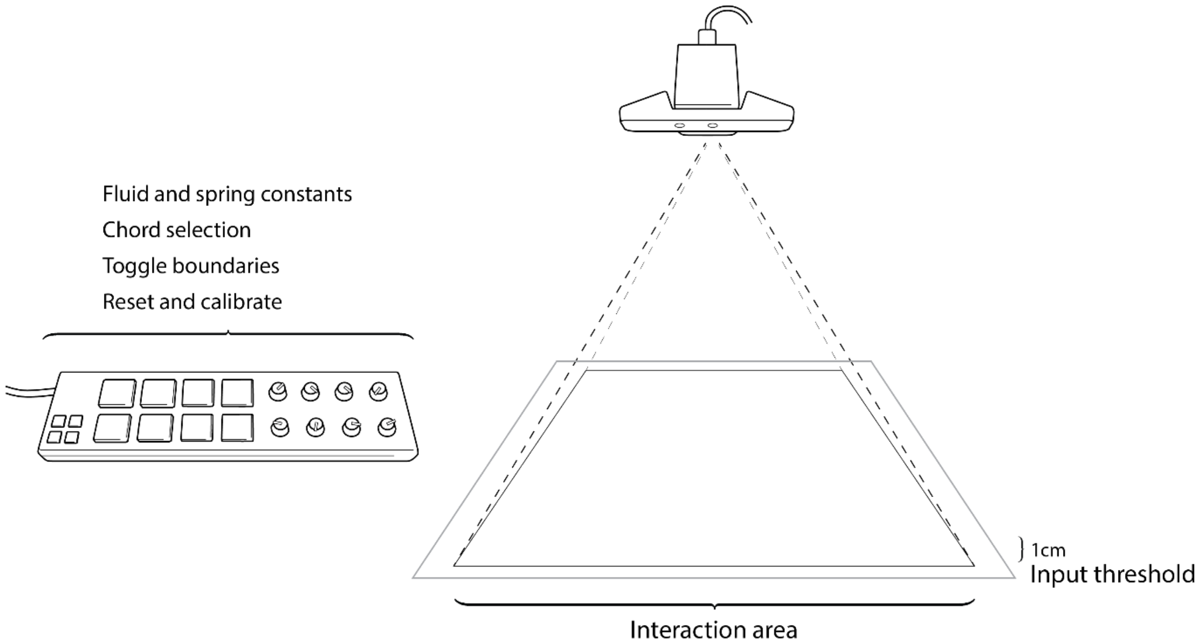

Figure 12. Hardware interface for Melange.

In Melange, a surface or table top is treated as a virtual interaction field onto which audiovisual material can be deposited. Anything that crosses the 1cm threshold—roughly the width of a finger—is added to the simulation. This satisfies the need for tactile feedback with a physical object.

Solid material that cross into the input threshold produces an image we call the contact texture. From this texture, simple blob detection may be performed to create additional parameters for controlling pitch or color.

A small MIDI controller was used for precise value control over various parameters of the system such as velocity diffusion rate and spring damping.

3.5.2 Depth Map Filtering

The depth map that is retrieved from the SR300 depth camera is inherently noisy. It is necessary to filter and stabilize the texture to facilitate predictable input and less stochastic results from the optical flow algorithm. Care must be taken to not smooth the input to the point of eliminating the subtle hand and finger motions of the performer.

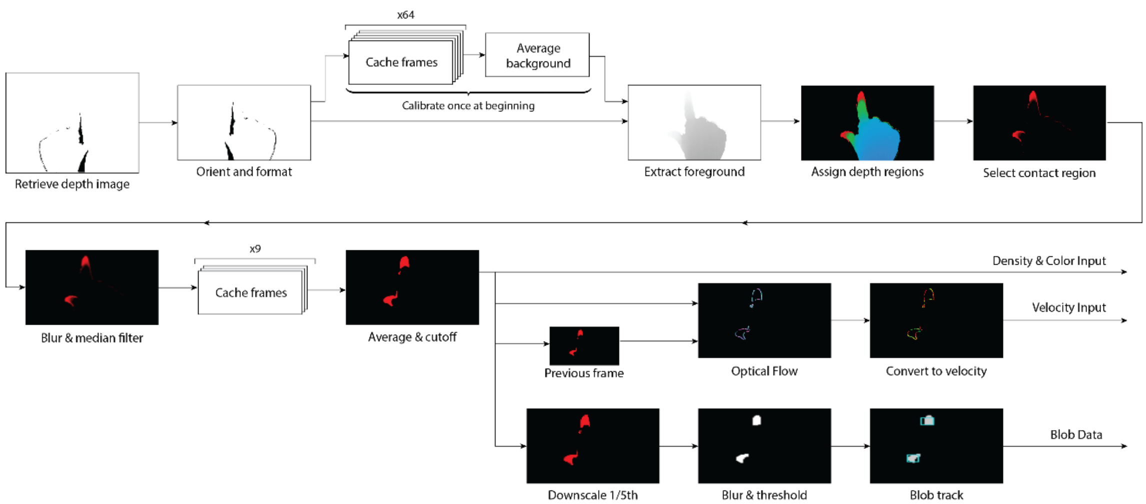

Figure 13. Depth map filtering.

The schematic in Figure 13 depicts steps taken in filtering the depth map. The depth image is first compared to an image of the background that has been stored in memory to extract foreground objects. Next, data below the input threshold is removed to produce a contact texture. The contact texture is blurred using a Gaussian kernel and filtered using a 3x3 median filter to smooth its edges. Then, a nine-frame cache is stored which is averaged. Values below a set threshold is thrown out, further stabilizing the texture.

The contact texture is then processed using an optical flow algorithm implemented on the GPU [26]. The algorithm compares the location of edges from the current and previous frames to produce velocity vectors. These are used as impulses in the simulation.

The contact texture is downscaled by one tenth and send to TouchDesigner's native blob detection node which invokes OpenCV blob tracking routines. Blob tracking attempts to identify consistently distinct features in an image. It returns the center and dimension of each blob which are combined in different ways for image and sound control.

3.5.3 Control Signal Experiments

Blob tracking was chosen to control how pitches are selected. This was based on trial and error with different control signals.

First, we tried analyzing different components of the fluid such as pressure, divergence, and vorticity to drive the pitch changes. This felt too limiting since it locked in certain states of the fluid to a particular frequency range. It was also hard to predict what the note would be.

The RealSense SR300 depth camera has some powerful tools for analyzing hand images and generating a bone system with three-dimensional finger joint positions and rotations. Using those as control values was the next attempted approach. Unfortunately, the camera expects to be mounted on a monitor and facing directly at users, where their hands would be palm-forward, interacting with their computer. The way it is configured in Melange has the hand in the opposite direction, so the hand feature algorithm would bend the fingers backwards and give unusable results. The second image in Figure 14 is doing just that—the camera is above and the users fingers are curling downward to the bottom of the image, but the algorithm assumes they are going the other way.

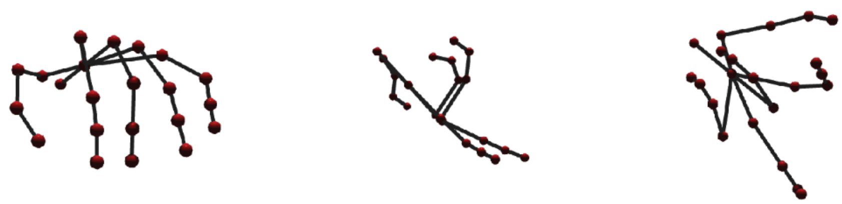

Figure 14. Hand feature tracking results.

We did not want to sacrifice touching a physical surface, so blob detection was the final choice. Feeding the blob tracking algorithm a downscaled version of the contact texture, fairly accurate real-time touch position information from any object can be retrieved, not just fingers. In Melange, different objects such as stones, rulers, and drink coasters were all used to manipulate the fluid.

3.5.4 Chord Progression

To really explore musical potential, it was important to control a chord progression while also having the freedom to break away from it. Mimicking the radial layout of the strings' rest state, the field is divided into regions for selecting chords, as seen in Figure 15.

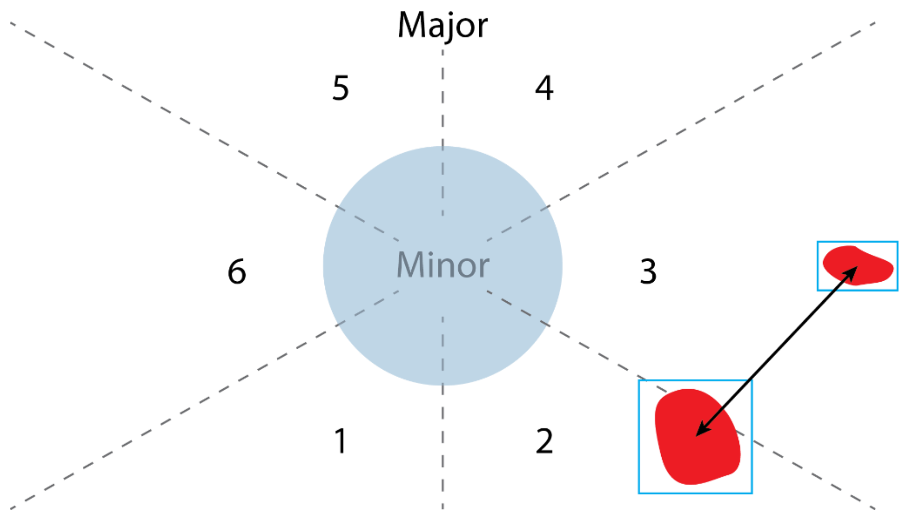

Figure 15. Chord progression trigger regions.

The chords advance counter clockwise around the center and are chosen based on the first blob that's detected. The chord selection is held until the contact region is empty and no more blobs are present. This prevents any one area of the image to being locked to a specific chord because the gesture can start in one area and move to the other, maintaining that note.

The area in the center shifts each region's chord to its minor. If the maximum distance between blobs is above a set threshold—shown in Figure 15 as the black arrow pointing to the two blobs—all the chord indices advance by six, making it a twelve-chord progression in total. Originally there were twelve regions, but the thinner slices sometimes made it difficult to get consistent picks. Finally, the number of blobs determines the octave. Each blob adds three whole steps, so two blobs adds a perfect octave.

It is important to note that this is just one possible tuning of the instrument. There are likely as many tunings for a system such as this as there are composers.

3.5.5 Color Scale

Each chord is associated with a color ramp. As mentioned in the background section, many have investigated the relationship between musical notes and hues.

Alexander Rimington started his color scale at C with dark red and progressing through the spectrum. Rimington emphasized that the colors were just approximations. His color organ had lamps that could be adjusted for hue, saturation, and brightness, so many gradations other than was printed in his "Colour Music" pamphlet could be achieved [28].

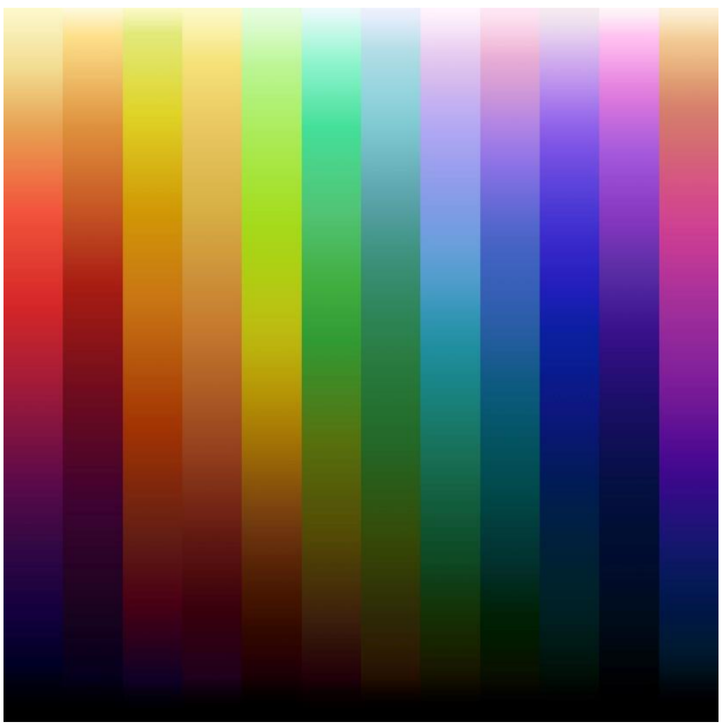

Figure 16. Color scales in Melange.

The mapping used in Melange is depicted is depicted in Figure 16. This scale is an inspired by Rimington's scale perceptual color ramps. Perceptual color ramps attempt to evenly distribute colors according to their perceived luminosity. These ramps have been increasingly popular in data visualization in the last few years due to their legibility [31]. As discussed on page 38 in the velocity color ramps section, the colors are brighter where the fluid is faster, and darker where it is slower. If a single note is played and not a chord, the hue range of each gradient is restricted to the dotted red line, but the value range is maintained.

4. Results

4.1 Contributions

The following is a list of contributions made public on the TouchDesigner forums and available as open source code on the author's GitHub web page [14].

- Fluid.tox received a fair amount of positive feedback from the community. It has been used in other projects, which I discuss in the following section.
- SenseTOP is a plugin written for TouchDesigner that allows users to control features specific to the SR300 depth camera such as laser projector power and filtering options.
- Spring-Mass Sonification is a simplified version of the scanned synthesis technique used in Melange.

4.2 User Adoption

Yea Chen used it to animate fire-like effects emanating from a dancer [3]. Fire is essentially a smoke simulation where some of the density glows, so this effect is not too far off. Chen also combined it with edge detection and optical flow. She started with the example file that was shared by the author which by default applies the velocity of the mouse cursor as an impulse force. Prema Paetsch extended the fluid component to look like more realistic flames by adding different frequencies of noise textures to the velocity feedback loop [23].

4.3 Milestones

4.3.1 Time Differential

The first iteration of the instrument was "Time Differential" in 2015. This work had three major differences with Melange:

1. The fluid simulation was written as a CUDA plugin. CUDA is a parallel programming language for NVIDIA graphics cards. It allows for very advanced mathematical operations and the potential for achieving optimal GPU performance, but the author ultimately found both the longer development time and less portability offered by GLSL shaders to be hindering the project.
2. This was a purely visual instrument—it did not produce any sound. At the time of the show, the sonification techniques were still in development not ready for exhibition.
3. The input system was very confusing. The big difference from Melange was that "Time Differential" used a touch sensitive trackpad instead of a camera-based system. The trackpad gives very accurate multitouch position information, but it is not a graphical input method, so the system could not produce unique shapes on its own.

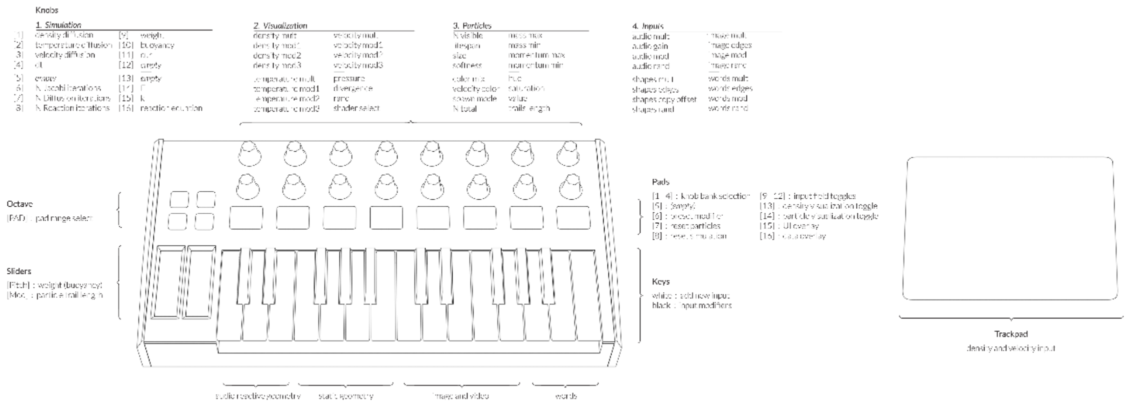

Figure 17. Hardware interface for Time Differential.

The author tried to compensate for the lack of a graphical input method by using a MIDI keyboard with twenty-five keys, sixteen knobs, eight pads, and four different modes it could be in, which each one completely changing the layout of the keyboard. The keys triggered shapes to appear at fixed locations for adding density and velocity. While some interesting, precise patterns could be made, the expressions were largely pre-defined. Melange still supports these kinds of inputs due to its image-based framework, but the focus of Melange is more on playing the fluid itself rather than triggering image samples.

4.3.2 Melange

Next is this project, which made its debut at the End of Year Show in May 2017. It was also used at a performance downtown at the Santa Barbara Center for Art, Science and Technology.

4.3.3 Cirrus

Finally, Cirrus is a real time generative artwork being installed in downtown Chicago. It runs on four computers and outputs to a 150-foot long, twenty-foot high LED wall. The installation uses the same fluid core as Melange with an additional reaction-diffusion term coupled to the density field. These image in Figure 18 is from an initial test done in March 2017. It is set to run continuously for one month in October 2017.

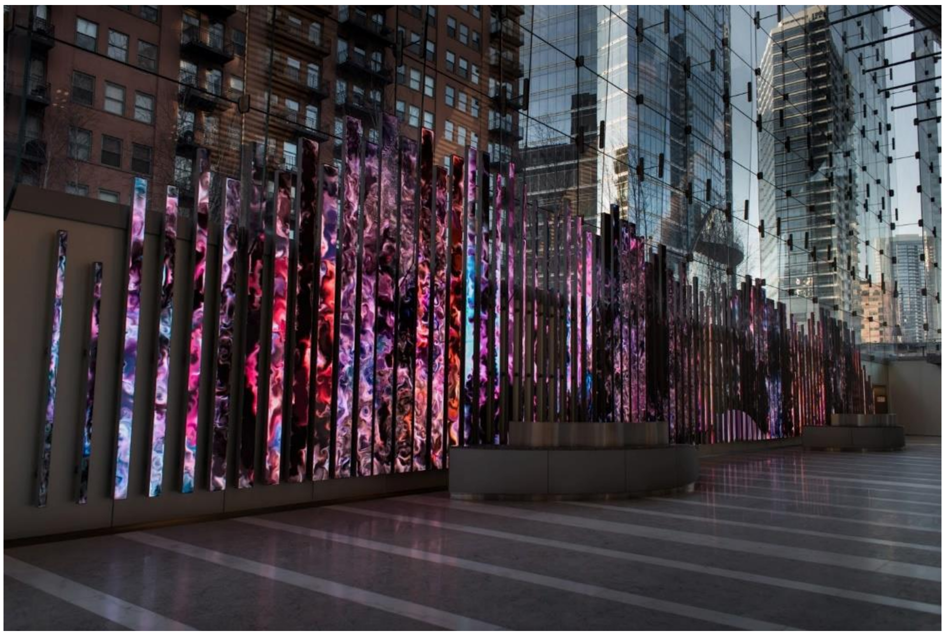

Figure 18. Cirrus installation in Chicago.

4.4 Evaluation

The evaluation criteria in Figure 19 are from Golan Levin’s attempts to review his audiovisual work [17].

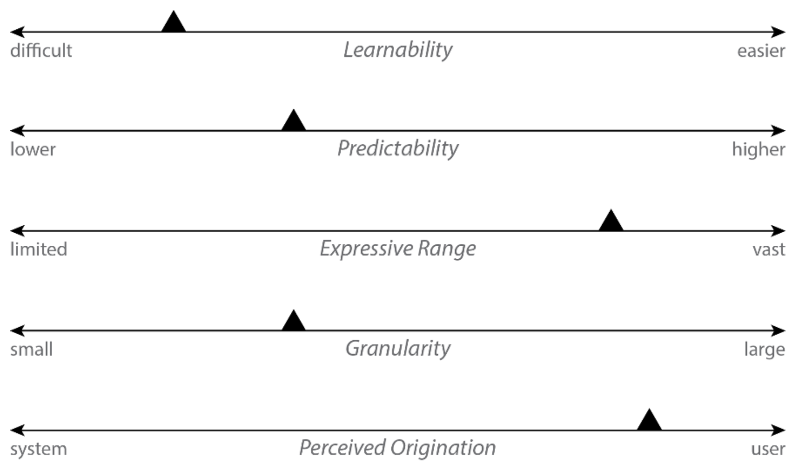

Figure 19. Melange evaluation.

The learning curve of Melange is fairly steep. Despite the primary input being easy to understand—where the performer touches the table is where it agitates the fluid—the sensitivity and parameter space of the system make producing subtle, evolving works a bit challenging.

Since the fluid is turbulent, its exact behavior can be difficult to predict, especially when factoring in the dynamics of the spring simulation.

The system has a wide range of expressive potential due to the variability of fluids. The precise, graphical input method allows for any two-dimensional composition to affect the system.

Small-scale turbulent patterns and delicate, wispy particles support fine levels of visual granulation. These are translated to sonic granularity through tight, undampened string settings.

Finally, the degree to which the performer’s actions determine the outcome is high. Aside from the dynamics inherent in the physical simulation, the composition is determined by the path the user has taken in depositing and manipulating the material.

5. Conclusion

5.1 Future Directions

There are many directions we could go from here. The system can be expanded by adding a sequencer for input. The idea would be to record not the resulting sound and image, but rather record gestures for arrangement and playback into the fluid simulation.

Another extension would be to simulate six fields and arrange them into a cube map. Extensions of the fluid to consider cube map boundaries would be required, but this would allow for spherical rendering in the AlloSphere or virtual reality.

Adding the third spatial dimension to the simulation is not a big stretch to implement. The challenge would be in how to control it, efficiently visualize it, and meaningfully sonify it.

Set and group theory for music would give the chord progressions a mathematical basis. This could potentially be coupled with simulation data in a useful way.

5.2 Expressive Physical Modeling

Ultimately, the author wishes to apply lessons learned from developing Melange to other physical processes. I believe treating physical phenomena as an inexhaustible, infinitely variable computational substance is a valuable conceptual framework for building audiovisual instruments.

By facilitating the creative expression of these complex systems, we can breathe a human voice into the laws of nature and aid in our quest in understanding them. The author hopes this work will inspire the same in others.

6. References

1. M. Akten. (2009) Memo Akten: Body Paint. [Online]. http://www.memo.tv/bodypaint/

2. C. Brebbia, C. Greated, and M. Collins, *Colour in Art, Design & Nature*. Southampton, UK: WIT Press, 2014.
3. Y. Chen. (2017, January) Finally, Optical Flow + Fluid. [Online]. https://www.facebook.com/yea.chen.9/videos/vb.1349932443/10212075283581266/

4. I. Chrissochoidis, S. Houliaras, and C. Mitsakis, "Set theory in Xenakis' EONTA," in *International Symposium Iannis Xenakis*, Athens, 2005, pp. 241-249.

5. L. da Vinci, "Studies of flowing water, with notes," The Royal Collection, London, 1510-13.

6. P. N. Edwards, "Entangled histories: Climate science and nuclear weapons research," *Bulletin of the Atomic Scientists*, vol. 68, no. 4, July 2012.

7. R. Fedkiw, J. Stam, and H. W. Jensen, "Visual Simulation of Smoke," in *Proceedings of SIGGRAPH*, Los Angeles, 2001.

8. J. Gerrard. (2017) Western Flag. [Online]. http://westernflag.johngerrard.net/

9. M. J. Harris, "GPU Gems," in *GPU Gems*. Boston, MA: Addison-Wesley Professional, 2004, ch. 38.

10. W. Herzog, "Cave of Forgotten Dreams," Ministère de la Culture et de la Communication, Paris, Film 2010.

11. V. Houzé. (Folativ: Lull) 2015. [Online]. http://www.foliativ.net/lull/

12. A. Hunt, "Radical User Interfaces for Real-time Musical Control," University of York, York, PhD Thesis 1999.

13. Ned Kahn. (2012) Ned Kahn Studios. [Online]. http://nedkahn.com/

14. K. Kaminski. (2017) Kurt Kaminski's GitHub Repositories. [Online]. https://github.com/kamindustries

15. A. Kircher, *Musurgia Universalis*. Rome, 1650.

16. G. Levin. (2000) Flong. [Online]. http://www.flong.com/projects/aves/

17. G. Levin, "Painterly Interfaces for Audiovisual Instruments," Massachussets Institute of Technology, Boston, M.S. Thesis 2000.

18. Maura McDonnell. (2002) Colour and Sound: Visual Music. [Online]. http://homepage.eircom.net/~musima/visualmusic/visualmusic.htm

19. R McGee, J. Dickinson, and G. Legrady, "VOICE OF SISYPHUS: AN IMAGE SONIFICATION MULTIMEDIA INSTALLATION," in *The 18th International Conference on Auditory Display*, Atlanta, 2012.

20. C. Meide, *The Dugout Canoe in the Americas: An Archaeological, Ethnohistorical, and Structural Overview*. Tallahassee: Florida State University, April 1995.

21. Y. Nakayama, *Introduction to Fluid Mechanics*. Woburn: Butterworth-Heinemann, 1999.

22. R. Oppenheimer. (2009, April) Maximal Art: The Origins and Aesthetics of West Coast Light Shows. [Online]. https://rhizome.org/editorial/2009/apr/15/maximal-art-the-origins-and-aesthetics-of-west-coa/

23. P. Paetsch. (2017, March) TouchDesigner - Realtime fire. [Online]. https://vimeo.com/209054124

24. Chris Parks. (2007) Chris Parks Art. [Online]. http://www.chrisparksart.com/thefountain.htm

25. J. Peel, "The Scale and the Spectrum," *Cabinet*, no. 22, Summer 2006.

26. princemio. (2013, November) Optical Flow GPU Version - based upon Andrew Benson's solution. [Online]. https://github.com/princemio/ofxMIOFlowGLSL

27. L. Putnam, J. Kuchera-Morin, and L. Peliti, "Studies in Composing Hydrogen Atom Wavefunctions," *Leonardo*, vol. 48, no. 2, pp. 158-166, April 2016.

28. A. Rimington, "A New Art: Colour-Music," Messrs. Spottiswoode & Co., London, Pamphlet 1895.

29. J. Stam, "Stable Fluids," in *SIGGRAPH*, Los Angeles, 1999, pp. 121-128.

30. H. Thieme, "Lower Palaeolithic hunting spears from Germany," *Nature*, vol. 385, pp. 807-810, 1997.

31. S. van der Walt and N. Smith. (2015) Mpl Colormaps. [Online]. https://bids.github.io/colormap/

32. V. Van Gogh, "Starry Night," Museum of Modern Art, New York City, 1889.

33. B. Verplank and M. Matthews, "Scanned Synthesis," in *ICMC*, Berlin, 2000.