5-2015

Creating Abstract Motion Sculptures Through Simulation

Jonathan M. Barry
Clemson University

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Recommended Citation
https://tigerprints.clemson.edu/all_theses/2079

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.
Abstract

Inspired by artists of the late 19th-century and early 20th-century, such as Étienne-Jules Marey, Marcel Duchamp and Anton Bragaglia, this thesis describes the development of dynamic motion sculptures abstracted from motion capture data combined with simulation and procedural modeling techniques. The methods described in this thesis present a robust, simulation-driven, and artist friendly process for manipulating animated point data into concrete and controllable structures. SideFX Software’s 3D development package, Houdini, was used for nearly all aspects of the production. The node-based workflow streamlined the process of building, testing, and integrating custom tools and provided a system for procedural modeling and data-driven shading. A virtual museum tour showcases the sculptures generated from performance capture of a contemporary dance.
Acknowledgments

I would like to thank my committee: Dr. Jerry Tessendorf for his suggestions and critique during the production process, Dr. Donald House for his ideas and recommendations to blend art and simulation, and Dr. Sophie Jörg for her expertise in the motion capture process.

I would like to thank Tony Penna and Dr. Andrea Feeser, whose artistic background and experience became an invaluable asset in the design process. I would also like to thank Brooke O’Friel for her hard work choreographing and performing the dance and Mandy Morland for her assistance in the motion capture process and data cleanup.

Finally, I thank my family and friends who have supported and encouraged me through all of my endeavors.
# Table of Contents

Abstract ................................................................. ii
Acknowledgments ......................................................... iii
List of Figures ........................................................... v
1 Introduction ............................................................ 1
2 Background ............................................................. 2
3 Design ................................................................. 11
4 Implementation ......................................................... 16
5 Results ................................................................. 25
Appendix ................................................................. 45
References ............................................................... 45
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Head-spring, a flying pigeon interfering by Eadward Muybridge [13]</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Chronophotographic gun from La Nature, 1882 [6]</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>Somersault over an obstacle by Étienne-Jules Marey, 1884 [12]</td>
<td>5</td>
</tr>
<tr>
<td>2.4</td>
<td>Nude Descending a Staircase by Marcel Duchamp [8]</td>
<td>6</td>
</tr>
<tr>
<td>2.5</td>
<td>Photodynamism by Anton Bragaglia</td>
<td>8</td>
</tr>
<tr>
<td>2.5.a</td>
<td>Multiple exposures of baseball player, 1913 [3]</td>
<td>8</td>
</tr>
<tr>
<td>2.5.b</td>
<td>The Cellist, 1913 [4]</td>
<td>8</td>
</tr>
<tr>
<td>2.6</td>
<td>Applications of motion capture</td>
<td>9</td>
</tr>
<tr>
<td>2.6.a</td>
<td>Sprinter doing motion capture for Nike’s Sports Research Lab [14]</td>
<td>9</td>
</tr>
<tr>
<td>2.6.b</td>
<td>Performance capture for Dawn of the Planet of the Apes, (2014) [16]</td>
<td>9</td>
</tr>
<tr>
<td>2.7</td>
<td>Results from Line Drawing as a Dynamic Process, D. House and M. Singh [10]</td>
<td>10</td>
</tr>
<tr>
<td>2.7.a</td>
<td>Input Model</td>
<td>10</td>
</tr>
<tr>
<td>2.7.b</td>
<td>Generated Line Drawing</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Examples of interior design of modern art museums</td>
<td>11</td>
</tr>
<tr>
<td>3.1.a</td>
<td>Rumansky Art Centre, Slovakia [1]</td>
<td>11</td>
</tr>
<tr>
<td>3.1.b</td>
<td>Modern Art Museum in Fort Worth, Texas [17]</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Final design of museum</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Materialization of motion sculpture</td>
<td>13</td>
</tr>
<tr>
<td>3.4</td>
<td>“Light Painting” long-exposure photography</td>
<td>14</td>
</tr>
<tr>
<td>3.4.a</td>
<td>LEDs on Roomba [2]</td>
<td>14</td>
</tr>
<tr>
<td>3.4.b</td>
<td>Light Painting [7]</td>
<td>14</td>
</tr>
<tr>
<td>3.5</td>
<td>Examples of sculptures inspired by light painting</td>
<td>15</td>
</tr>
<tr>
<td>4.1</td>
<td>Dancer in motion capture suit on performance stage</td>
<td>16</td>
</tr>
<tr>
<td>4.2</td>
<td>Motion capture data represented as (A) points, (B) curves, and (C) mesh.</td>
<td>17</td>
</tr>
<tr>
<td>4.3</td>
<td>Interface for initializing spring constants on the control points</td>
<td>19</td>
</tr>
<tr>
<td>4.4</td>
<td>Drawing of motion trails with and without the controller system</td>
<td>20</td>
</tr>
<tr>
<td>4.4.a</td>
<td>Without the PID controller system</td>
<td>20</td>
</tr>
<tr>
<td>4.4.b</td>
<td>With the PID system (high natural frequency)</td>
<td>20</td>
</tr>
<tr>
<td>4.4.c</td>
<td>With the PID system (low damping)</td>
<td>20</td>
</tr>
<tr>
<td>4.4.d</td>
<td>With the PID system (high damping)</td>
<td>20</td>
</tr>
<tr>
<td>4.5</td>
<td>Drawing of motion trails with and without velocity scaling</td>
<td>21</td>
</tr>
<tr>
<td>4.5.a</td>
<td>Constant trail width</td>
<td>21</td>
</tr>
<tr>
<td>4.5.b</td>
<td>Trail width based on velocity of pen</td>
<td>21</td>
</tr>
<tr>
<td>4.6</td>
<td>Using OpenVDB to re-mesh a sculpture</td>
<td>23</td>
</tr>
<tr>
<td>4.6.a</td>
<td>Input Mesh</td>
<td>23</td>
</tr>
<tr>
<td>4.6.b</td>
<td>Levelet volume</td>
<td>23</td>
</tr>
<tr>
<td>4.6.c</td>
<td>Output Mesh</td>
<td>23</td>
</tr>
<tr>
<td>4.6.d</td>
<td>Close-up</td>
<td>23</td>
</tr>
<tr>
<td>4.7</td>
<td>Attribute-based shading</td>
<td>24</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------</td>
<td>----</td>
</tr>
<tr>
<td>4.7.a</td>
<td>Input Mesh</td>
<td>24</td>
</tr>
<tr>
<td>4.7.b</td>
<td>Attribute Visualization</td>
<td>24</td>
</tr>
<tr>
<td>4.7.c</td>
<td>Output Mesh</td>
<td>24</td>
</tr>
<tr>
<td>4.7.d</td>
<td>Close-up</td>
<td>24</td>
</tr>
<tr>
<td>5.1</td>
<td>Small Sculpture 1</td>
<td>25</td>
</tr>
<tr>
<td>5.2</td>
<td>Small Sculpture 2</td>
<td>26</td>
</tr>
<tr>
<td>5.3</td>
<td>Small Sculpture 3</td>
<td>26</td>
</tr>
<tr>
<td>5.4</td>
<td>Small Sculpture 4</td>
<td>27</td>
</tr>
<tr>
<td>5.5</td>
<td>Small Sculpture 5</td>
<td>27</td>
</tr>
<tr>
<td>5.6</td>
<td>Large Sculpture 1</td>
<td>28</td>
</tr>
<tr>
<td>5.7</td>
<td>Large Sculpture 2</td>
<td>28</td>
</tr>
<tr>
<td>5.8</td>
<td>Large Sculpture 3</td>
<td>29</td>
</tr>
<tr>
<td>5.9</td>
<td>Large Sculpture 4</td>
<td>29</td>
</tr>
<tr>
<td>5.10</td>
<td>Large Sculpture 5</td>
<td>30</td>
</tr>
<tr>
<td>5.11</td>
<td>Shot 1 - Introduction to the museum environment</td>
<td>31</td>
</tr>
<tr>
<td>5.12</td>
<td>Shot 2 - The first sculpture begins forming</td>
<td>32</td>
</tr>
<tr>
<td>5.13</td>
<td>Shot 3 - Close-up of first sculpture</td>
<td>33</td>
</tr>
<tr>
<td>5.14</td>
<td>Shot 4 - The second sculpture begins forming</td>
<td>34</td>
</tr>
<tr>
<td>5.15</td>
<td>Shot 5 - The second sculpture finishes building</td>
<td>35</td>
</tr>
<tr>
<td>5.16</td>
<td>Shot 6 - Traveling to the other side of the room as the third sculpture materializes</td>
<td>36</td>
</tr>
<tr>
<td>5.17</td>
<td>Shot 7 - Spiral through the third sculpture</td>
<td>37</td>
</tr>
<tr>
<td>5.18</td>
<td>Shot 8 - Wide shot as two more sculptures appear</td>
<td>38</td>
</tr>
<tr>
<td>5.19</td>
<td>Shot 9 - The final sculptures finish construction</td>
<td>39</td>
</tr>
<tr>
<td>5.20</td>
<td>Shot 10 - Final view of the full exhibit.</td>
<td>40</td>
</tr>
<tr>
<td>5.21</td>
<td>Sculpture generated without spring system (left) and with (right)</td>
<td>41</td>
</tr>
<tr>
<td>5.22</td>
<td>Sculpture generated from particle system</td>
<td>42</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This thesis presents a method for creating artistic, sculptural interpretations of motion by abstracting and manipulating data through dynamic simulation. Using spring dynamics coupled with motion capture data, a series of sculptures were constructed from a contemporary dance routine.

The dance was choreographed and performed by Brooke O’Friel, who has been classically trained in both jazz and ballet, and has also studied more contemporary styles such as tap, hip-hop, and musical theatre. With inspiration from dance, theatre, and modern art movements, a short animation was created to showcase the sculptures as a museum exhibit. Throughout the duration of the short, several sculptures are constructed and materialize, giving a small glimpse at the dancer’s performance.

The process of creating these sculptures utilizes SideFX Houdini’s robust procedural workflow. Various tools were developed for manipulating the shape, color, and animation of the structures. Furthermore, in addition to motion capture data, the tools and workflow that were developed can be applied to any animated point data (e.g. keyframe animation, particle simulation) and provides freedom and control for artist-driven creation of abstract motion sculptures.
Chapter 2

Background

Motion and its interpretations have long been subjects of interest in both science and art. In media and visual arts, the capture of motion and its reconstruction has fundamentally stayed the same for the past 100 years: a series of images is viewed in quick succession, each with slightly varying poses, giving the illusion that a single object is transitioning from one pose to the next.

Eadweard Muybridge was an innovative photographer active in the late 1880s who became well known for his motion studies and pioneering techniques for capturing movement. He developed an array of cameras with very short exposure times that could be sequentially triggered, producing a series of consecutive images. These images, when viewed individually, represented a single instance in time, but as a collection, accurately portrayed movement as a whole. Figure 2.1 shows a series of photographs from Muybridge’s *Animal Locomotion* collection which showcases what he was able to achieve with his 12 camera setup. The collection of more than 20,000 photographs, along with the rest of his work, has been accepted by scholars and artists as an exceptional study in the mechanics of human and animal motion [18].

Muybridge’s motion studies have been studied extensively by researchers and artists alike because of its discrete and clear representation of motion. His innovations with photographic technology and imaging techniques also pushed the art form to new ways of visualizing motion.
Figure 2.1: *Head-spring, a flying pigeon interfering* by Eadward Muybridge, 1887 [13].
Contemporary with Muybridge, Étienne-Jules Marey, who also studied photography and movement, developed a “chronophotography gun” (Figure 2.2) which consisted of a spinning shutter and a single exposure plate [6]. This allowed him to quickly capture a series of images like Muybridge had done, but expose them on a single piece of film. This was advantageous because Marey could easily trace the trajectories of the animal’s locomotion and map it to anatomically accurate systems of bones and muscles.

He experimented with gestural styles of capturing the movement, and focused less on the individual poses. Instead, he tried to accentuate the flowing arcs and transitions from one pose to the next. In Figure 2.3, the motion of a human form is merely suggested by the flowing lines and blurred shapes. Additionally, the photograph illustrates an interesting effect that Marey was able to achieve, the overlay of multiple poses in a single frame. This abstract imagery would inspire ideas in two important European modern art movements: Cubism and Futurism, both of which broke away from the analytical portrayals of movement and tried to use motion to provoke a sensation or emotion.
Figure 2.3: *Somersault over an obstacle* by Étienne-Jules Marey, 1884 [12]
The French Cubist movement began in the early 1900s and developed artistic visions of time and space. Objects were broken down into segmented geometric pieces and reconstructed in abstracted ways to juxtapose the three-dimensional subject matter of the real world with the two-dimensional space of the canvas. In regards to the fourth dimension, time, French philosopher Henri Bergson wrote that “life is subjectively experienced as a continuous forward movement in time, with the past flowing into the present and the present merging into the future” [9]. Much in the same sense that dimensionality can be broken down and viewed simultaneously, time could also be broken down and reconstructed.

Figure 2.4: *Nude Descending a Staircase, No. 2* by Marcel Duchamp, 1912 [8].

In the early 1910s, Marcel Duchamp began developing his own interpretations of motion in regards to time and space. *Nude Descending a Staircase, No. 2* (Figure 2.4), one of his most famous works, depicts a nude woman descending a staircase and portrays her motions as an overlay of several
Duchamp was greatly influenced by the work of both Muybridge and Marey, combining their deconstruction of movement and time with Cubism’s deconstruction of shape and space.

While the French Cubist painters were dealing with fragmentation, Italian Futurists were picking up where Muybridge and Marey left off with a movement known as Photodynamism. Anton Bragaglia, who grew up working in close proximity to film and cinematography, was a pioneer of Photodynamism, in which photography tightly embraced technology not as an analytical tool for breaking apart time and space, but as a means to an artistic end, using time as a brush on the spatial canvas. In his manifesto _Fotodinamic Futurista_, Bragaglia explains his intentions [5]:

> We, certainly, are not concerned with the aims and characteristics of cinematography and chronophotography. We are concerned not with the precise reconstruction of a movement, which has already been broken up and analyzed, but only with that element of a movement which produces sensation, the memory of which still palpitates in our consciousness.

> We despise the precise, mechanical, icy reproduction of reality, avoiding it at all costs, because for our purposes it is a harmful and negative element, whereas for cinematography and chronophotography it is the very essence. They in turn overlook the trajectory, which for us is the most essential value.

Comparisons can be made between Bragaglia’s photodynamism and Marey’s chronophotography. However, Bragaglia was clear that while they are technologically similar, artistically-speaking there is much more effort put into provoking a feeling or sensation by the viewer, expressing the metaphysical essence of movement. Figure 2.5 shows a few examples of Bragaglia’s work. Though subjective, there appears to be more consideration in composition and story telling, not just showcasing a motion. The intensity of the baseball being thrown and the musicality of the performer are felt through Bragaglia’s use of composition to focus on individual aspects of the movement.
a. *Multiple exposures of baseball player, 1913* [3]


Figure 2.5: Photodynamism by Anton Bragaglia
Movement is a means of communication as well as artistic expression, whether captured as a series of moving images or a single still frame. With modern technology, we can go beyond capturing a figure’s motions on canvas or film. Using an array of infrared camera’s and small markers attached to the actor, motion capture technology (also called performance capture) allows us to collect large amounts of accurate data representing movement, joint rotations, and acceleration. Similar to Muybridge’s work, the data is both scientifically and artistically intriguing. It is used on athletes (Figure 2.6.a) for medical diagnosis, analyzing the repercussions of high impact sports [14]. In entertainment (Figure 2.6.b), it allows actors to copy live performances, emotions, and gestures to digital characters. In these scenarios, the data is manipulated and optimized by researchers and data processing tools or through collaboration of performers and animators and, in the end, is applied to the study of human mechanics or to a digital humanoid character.

a. Sprinter doing motion capture for Nike’s Sports Research Lab [14]

b. Performance capture for Dawn of the Planet of the Apes, (2014) [16]

Figure 2.6: Applications of motion capture
The main idea behind this project stems from considering how artists such as Marey, Duchamp, and Bragaglia might work with performance capture to emulate their artistic vision and interpretations of space, time, shape, and trajectory. In computer graphics and visual effects there is a notion of blurring the line between technology and art. In fact, there have been many attempts to “simulate” art in a computational setting. In the paper *Line Drawing as a Dynamic Process* (D. House and M. Singh) [10], a procedure is discussed that aims to simulate the gestural nature of line drawings using physics simulation and mathematical rendering techniques. In the paper, line drawings are simulated as complex tracings of three-dimensional objects with considerations taken for drawing speed, stroke thickness and smoothness, pen pressure, and contour. See Figure 2.7 for one of the results from their system.

![Figure 2.7](image-url)

Figure 2.7: Results from *Line Drawing as a Dynamic Process*, D.House and M.Singh [10]

The setup described in the paper consisted of a spring-mass-damper system attached to a control point, that when traced along the surface exhibited varying speeds and momentum conservation. Their simulation is projected back into two-dimensions so as to be used as a drawing on a flat surface. Inspired by their method, I used a similar setup attached to motion capture data to map trajectories of movement similar to Marey and Bragaglia. Specific details of the implementation will be covered in Section 4 of this paper.
Chapter 3

Design

Creating a coherent set of sculptures and placing them in a space that fit with the artistic nature of the exhibit, yet brought to mind a sense of staging and performance was an important consideration in the design process for the final animation. Keeping in mind the artistic influences, there was a desire to create an environment that resembled an art gallery or museum. Primarily, the room would be very minimal and simplistic as to not distract from the sculptures. However, the sculptures themselves needed to feel like they belonged there and had been put on display. The large sculptures, placed on large platforms, would occupy the perimeter of the room, while smaller platforms would fill the wide-open center. Figure 3.1 shows a few examples of modern art museums and the simplicity of their design and Figure 3.2 shows the final design of the museum.

Figure 3.1: Examples of interior design of modern art museums
Figure 3.2: Final design of museum
Furthermore, because the motion capture data was from a dance performance, I wanted to infuse some feeling of a stage and theatrical lighting elements. The wide open area in the middle would fill up as the dance progressed and tight spotlights would guide the narrative from pose to pose. Inspiration was taken from “Caught” [15], a dance choreographed by David Parsons, in which the performer moves around the stage and is illuminated by spot lights and strobe lights, seemingly freezing time and taking a snapshot of each pose. For the museum, small platforms were placed in the center of the room. As the fly-through progresses, each sculpture would materialize, tracing the motion of the dancer. This building process creates a finished motion sculpture while showcasing the movement and origin of its design. (Figure 3.3)

![Figure 3.3: Materialization of motion sculpture](image)
In regards to building and designing the sculptures, they needed to have a basis in reality, meaning they should appear to be built out of common materials (concrete, metal, ceramic, glass, etc.). Conversely, it was decided that, for the sake of visual intrigue, they should not necessarily be structurally possible in that segments need not be connected or supported in any way and could be suspended in mid-air.

Other inspiration came from long-exposure photography similar to what Marey and Bragaglia had developed. A style known as “light painting” (Figure 3.4) uses LEDs, or other light sources, in conjunction with a long-exposure, creates smooth, streaking lines of light within a darkened environment, as if painting with light.

![Image](images/fig3_04.png)

Figure 3.4: “Light Painting” long-exposure photography

These ideas were applied to the performance capture data using a custom tool set (described in Section 4) to produce a one minute and forty second long animation. The museum tour showcases a variety of sculptures, each capturing a specific moment in the dance. Figure 3.5 shows some results.
Figure 3.5: Examples of sculptures inspired by light painting
Chapter 4

Implementation

The first step in the process was gathering the data. Motion capture was used to retrieve movement data of a choreographed dance. The motion capture process consisted of placing 60 retroreflective markers into a specific configuration on the performer’s body using a motion capture suit (Figure 4.1). The markers are positioned over major joint positions and pivot points to capture an accurate representation of the performer’s movement. The capture stage is surrounded by 12 cameras that emit flashes of infrared light. These lights illuminate the markers attached to the suit, producing a reflection back to the motion capture cameras. When calibrated to the stage, the cameras work together to reconcile the reflections to accurately determine spatial location of all the markers on the performer. This constitutes a virtual 3-D interpretation of the dancer’s performance.

Figure 4.1: Dancer in motion capture suit on performance stage
The motion capture pipeline uses Vicon Blade for capturing and Vicon IQ for post-processing and cleanup. These software packages match the movement of the markers to a virtual skeleton’s joint rotations. Though the captured performance was recorded at 120 frames per second, the skeleton was exported to Maya where the animation was re-baked to 60 frames per second as the high resolution was not needed for the applications of this project. Once the skeleton has the performance attached as a series of keyframes, it can be applied to a variety of scenarios (Figure 4.2). For my purposes, I translated the animated skeleton into any number of points on the skeleton (A), a series of curves (B), or a mesh (C), each having its advantages and disadvantages. For example, the points or curves give a very lightweight way of working with the data and can be very robust, but clarity of the human form can be lost.

![Figure 4.2: Motion capture data represented as (A) points, (B) curves, and (C) mesh.](image)

There was a strong desire to not simply use the motion capture data, but to manipulate and abstract the data to get something more visually inspired by the work of Marey, Duchamp, and Bragaglia. While the dance was very well choreographed and performed, the motions are still constrained to the limits of the human body. Incorporating other dynamic properties, keeps the suggestion of the human movement while creating something much more fluid and free. The spring-mass-damper system described in Line Drawing as a Dynamic Process [10] is a 2-D application...
that is easily expanded to a 3-D environment. There is a set of control points and a corresponding set of pen points. There is a one-to-one mapping between the two sets and connecting each pair is a spring-mass-damper system. The spring has a defined constant that tries to reduce the error (separation) between the pen and the controller, while the pen has a defined mass allowing the inertia to conserve the pen’s momentum and drive it away from the controller. Generally speaking, this is called a linear proportional, integral, derivative (PID) controller. We can calculate the acceleration of the pen in terms of the spring properties and the error present in the system. The error is defined as the distance between the controller and pen: \( e = p - \tau \) where \( p \) is the position of the pen and \( \tau \) is the position of the controller. Equation 4.1 gives the acceleration of a pen with mass \( m \) and error \( e \):

\[
\frac{d^2 p}{dt^2} = -\frac{1}{m} \left( ke + d \frac{de}{dt} + c \int e \, dt \right) \tag{4.1}
\]

where \( k \), \( d \), and \( c \) are the proportional gain, derivative gain, and integral gain respectively. From the equation it is clear the proportional gain is the correctional force most directly associated with the error at any given point in time, while the derivative gain is a smoothing factor that reduces large changes in the error. For simplicity, we can drop the integral term, which tries to drive the amount of error to zero, but, in this case, did not drastically effect the overall system dynamics.

Also, we will assume that each pen has a unit mass of 1. With a consideration for the properties of the springs, we can rewrite the equation as follows:

\[
\frac{d^2 p}{dt^2} = -\omega^2 e - 2\zeta \omega \frac{de}{dt} \tag{4.2}
\]

where \( \omega \) is the natural frequency of the spring and \( \zeta \) is the damping ratio. This gives a much better interface into the forces applied to each pen.

SideFX Houdini was the primary software package that was used once the motion capture data was collected. Houdini uses a procedural, node-based workflow that is very robust and allows for fast, efficient changes, optimization, and experimentation. A system was built so that the motion capture data is used as the control points and a corresponding set of pen points is driven by the data. A useful feature of Houdini particles is the ability to store any number of attributes on a per-point basis and manipulate point attributes and positions through the use of VEX code. The particle system allowed easy processing of control points and creation of corresponding pen points. The PID equations were implemented within a Point Wrangle node in Houdini as VEX code running
on each of the point pairs. The Appendix includes a simplified version of this code as well as a brief explanation of the VEX language.

Additionally, the spring’s natural frequency and damping forces could be varied across the control points so that each pair behaved differently. Using a set of NURBS curves to store the motion capture data, the curves could be re-sampled and attributes added as needed. Because each point on the curve has a defined position, attributes could be added based on their parametric length along the curve. Figure 4.3 shows the interface used to setup the spring constants on the control points.

Figure 4.3: Interface for initializing spring constants on the control points
Generating the motion trails was a simple process. As the pen points moved via the simulation, each one emitted a trail of points that, when connected together, constituted a curve that traced the trajectory through 3-D space. Figure 4.4 gives a comparison between the unmodified motion capture data and the data output from the simulated PID controller system.

![Diagram](image-url)

a. Without the PID controller system  

b. With the PID system (high natural frequency)

c. With the PID system (low damping)  

d. With the PID system (high damping)

Figure 4.4: Drawing of motion trails with and without the controller system

The curves were turned into polygonal models using extrusion and re-meshing. Furthermore, the points on the curve were given attributes to control size, color, and twist. This provided a convenient interface for controlling and changing the output geometry. In many cases, it was useful to use the velocity of the pen point as a scaling factor for the width of the trail. Figure 4.5 shows that varying width with the velocity produces the desired gestural feel.
a. Constant trail width

b. Trail width based on velocity of pen

Figure 4.5: Drawing of motion trails with and without velocity scaling
The robust data manipulation tools and procedural modeling methods allowed for relatively quick visual development of the sculptures in a museum setting. Additional tools were developed to further manipulate the shape, color, and formation of the sculptures. The sculptures were categorized into stationary and moving. The stationary ones would be static within the museum. Without animation concerns, these could be modeled from the simulation data and manipulated further to produce interesting shapes and structures. Data was layered from different times of the performance and noise patterns were introduced into the trails. Additionally, mesh refinement could make the sculptures more realistic and organic by removing non-manifold geometry and smoothing the topology.

One toolkit that was especially helpful was OpenVDB. OpenVDB is a volumetric modeling and manipulation library that uses optimized data structures for fast processing. Within these complicated and interweaving trails, there were often inter-penetrations. To smooth these transitions and reshape the geometry, I used OpenVDB to convert the mesh into a levelset volume and then re-mesh it back into a polygonal mesh. This process can create a very large number of polygons and was therefore not suitable for the moving sculptures. However, the output mesh, though geometrically dense, produces a much more natural looking sculpture. Figure 4.6 shows this process.

As discussed in the Design section of this thesis, another inspiration was light painting. The moving sculptures were a natural candidate for mimicking the flowing lines present in long-exposure photography while maintaining the essence of the capture performance. Combined with the structural formation, intertwined streaks of light add interest to the structure and contours of the motion sculptures.

Because of the concrete nature of the sculptures and a desire to anchor them in reality, they needed to feel imperfect, as if they were carved and built by the hands of an artist. Houdini’s procedural nature allowed for geometry based attributes to drive shading parameters, such as color, displacement, noise, and specular. This provided a quick and easy way of adding organic breakup to the surfaces (Figure 4.7).
Figure 4.6: Using OpenVDB to re-mesh a sculpture
Figure 4.7: Attribute-based shading

a. Input Mesh
b. Attribute Visualization
c. Output Mesh
d. Close-up
Chapter 5

Results

The finished animation is one minute and forty seconds long with a total of ten shots. There are five large stationary sculptures placed around the perimeter of the room, while five smaller sculptures are constructed throughout the duration of the short. Though generated from various segments of the same dance routine, each sculpture was given a unique style and color palette.

Figure 5.1: Small Sculpture 1
Figure 5.2: Small Sculpture 2

Figure 5.3: Small Sculpture 3
Figure 5.6: Large Sculpture 1

Figure 5.7: Large Sculpture 2
Figure 5.8: Large Sculpture 3

Figure 5.9: Large Sculpture 4
Choosing the parts of the dance to use for construction of each sculpture was a creative process. Overall, the design was driven by experimentation with shape, color, and silhouette. Furthermore, using specific parts of the body (i.e. legs, arms, hands, etc.) proved to be especially effective in creating abstract, yet coherent structures. The inspiration gathered from the techniques and ideas presented in Cubism and Futurism helped to place the sculptures in an artistic context and together the sculptures represent an exhibit.

Though the cinematography creates the sensation of walking around and exploring the museum as the sculptures are forming, the camera’s movement is not restricted to a person’s point-of-view. Additionally, light and shadow play a key part in bringing interest to the composition and guiding the narrative. Spotlights fade in and illuminate the empty platforms to cue the next sculpture. Lights focused on the large stationary sculptures create interweaving shadows on the walls of the room.
Figure 5.11: Shot 1 - Introduction to the museum environment
Figure 5.12: Shot 2 - The first sculpture begins forming
Figure 5.13: Shot 3 - Close-up of first sculpture
Figure 5.14: Shot 4 - The second sculpture begins forming
Figure 5.15: Shot 5 - The second sculpture finishes building
Figure 5.16: Shot 6 - Traveling to the other side of the room as the third sculpture materializes
Figure 5.17: Shot 7 - Spiral through the third sculpture
Figure 5.18: Shot 8 - Wide shot as two more sculptures appear
Figure 5.19: Shot 9 - The final sculptures finish construction
Figure 5.20: Shot 10 - Final view of the full exhibit.
The methods discussed in this thesis can be applied to a variety of motions to create interesting sculptures. The tools are robust and can work with any animated set of points, whether it is performance capture, keyframe animation, or dynamic simulation. The trail system can help visualize precise trajectories of objects, but when paired with the spring simulation, creates an abstract and artistic interpretation of that motion. Even the simplest animation can create complex and intricate sculptures. Figure 5.21 shows two sculptures generated from animating the transformation of a basic sphere: one with the dynamic point-constraint system and one without. Both are interesting, however the simulated sculpture has a much more organic and varying structure. Figure 5.22 shows a sculpture generated from a small particle system.

Figure 5.21: Sculpture generated without spring system (left) and with (right)
Figure 5.22: Sculpture generated from particle system
Appendix

The Houdini Point Wrangle nodes provide a process for writing custom code (in VEX) to manipulate points and geometry. The code is written for a single point and is then processed over the entire collection of input points. It is a convenient way to add and change point attributes. Below is the VEX code that implements the PID system described in Section 3 of this thesis.

In the code below, the ch("..") refers to a channel added to the interface of the node. This allows the artist to update and change parameters without directly changing the code. The @ symbol (in most cases) denotes a parameter or attribute that is directly stored on the points themselves. These attributes can be modified within the simulation and the values can later be used to drive other processes, such as shading. The variables @OpInput1, @OpInput2, etc. refer to the node streams connected to the Point Wrangle’s inputs. The first input is set of points on which the code is being run and the other inputs can be other geometry that is referenced inside the code. In this case, input 1 is the set of pen points, input 2 is the control points, and input 3 is the set of pen points at the previous frame. The system variable @ptnum is the point number currently being processed. This is used for looking up a specific point in a large data set.
```cpp
/** GET THE SYSTEM PARAMETERS **/
float deltaT = ch("timescale");
vector externalForce = set(ch("extForcex"), ch("extForcey"), ch("extForcez"));

/** INITIALIZE VARIABLES **/
v@e = {0,0,0};
v@v = {0,0,0};
float zeta = point(@OpInput2, "zeta", @ptnum) * ch("scale_zeta");
float omega = point(@OpInput2, "omega", @ptnum) * ch("scale_omega");

/** GET THE POSITIONS OF THE PEN AND CONTROL POINTS **/
vector ctrl_position = point(@OpInput2, "P", @ptnum);
vector pen_position = point(@OpInput3, "P", @ptnum);

/** CALCULATE THE ERROR AND CHANGE IN ERROR **/
vector e_current = pen_position - ctrl_position;
v@e = e_current;
f@error = length(e_current);
vector e_prev = point(@OpInput3, "e", @ptnum);
vector e_change = (e_current - e_prev) * (1.0/deltaT);

/** CALCULATE THE PROPORTIONAL AND DERIVATIVE GAINS **/
vector proportional = -1.0 * omega * omega * e_current;
vector derivative = 2.0 * zeta * omega * e_change;

/** UPDATE THE ACCELERATION AND VELOCITY VECTORS **/
vector accelVec = (proportional - derivative) + externalForce;
@accel = accelVec;
f@acceleration = length(@accel);

vector velVec = point(@OpInput3, "v", @ptnum) + accelVec*deltaT;
@v = velVec;
f@speed = length(@v);

/** UPDATE THE PEN POSITION **/
@P = pen_position + velVec*deltaT;
```
References


