Study in User Preferred Pen Gestures for Controlling a Virtual Character

By

Shusaku Hanamoto

A Project

submitted to

Oregon State University

in partial fulfillment of the requirements for the degree of

Master of Science

1/05/2006
©Copyright by Shusaku Hanamoto

January 5, 2005

All Rights Reserved
AN ABSTRACT OF THE PROJECT OF

Shusaku Hanamoto for the degree of Master of Science in

Computer Science presented on 1/05/2006.

Title: Study in User Preferred Pen Gestures for Controlling a Virtual Character

Controlling a virtual character with a pen input device is difficult. Pen input devices require freeform gestures and users are not confined to particular mapping of a key or a button that is exactly repeatable. This is a problem since an intuitive motion gesture for one user might not be intuitive for another user. In this paper, we explore user preferred input gestures for character control through user experiments. Most previous pen input gesture sets are based on the preference of the developer. Our goal is to try to find common pen gesture features for common commands through user experiments. For our experiment, we have chosen navigational motions that are common for controlling a character in a virtual world. The users were asked to make gestures for a set of navigational motions according to their intuition. We then analyzed the gesture data and outlined some gesture design guidelines as well as compared the resulting gestures to those used in existing applications that use pen devices for character control.
DEDICATION

I dedicate this project to my family, especially to my parents, for their undying support, love, and sacrifice.
# TABLE OF CONTENTS

| 1 | INTRODUCTION ........................................................................... | 1 |
| 2 | OVERVIEW ............................................................................. | 3 |
| 3 | BACKGROUND .......................................................................... | 5 |
| 4 | EXPERIMENT ........................................................................... | 7 |
|   | 4.1 Experiment Goal .............................................................. | 7 |
|   | 4.2 Experiment Setup .............................................................. | 8 |
|   | 4.3 Experiment Method ............................................................ | 8 |
| 5 | ANALYSIS .................................................................................. | 10 |
|   | 5.1 Aerial Perspective ............................................................. | 10 |
|   | 5.2 Following Perspective .......................................................... | 14 |
|   | 5.3 First-person Perspective ...................................................... | 19 |
|   | 5.4 Discussion ........................................................................... | 21 |
| 6 | COMPARISONS ........................................................................... | 25 |
| 7 | DESIGN APPROACH ..................................................................... | 29 |
|   | 7.1 Fundamental Movement Design ............................................. | 29 |
|   | 7.2 Advanced Movement Design ................................................ | 30 |
| 8 | SUMMARY .................................................................................. | 32 |
|   | 8.1 Conclusion ........................................................................... | 32 |
|   | 8.2 Future Work ........................................................................ | 33 |
| 9 | ACKNOWLEDGMENTS .................................................................. | 35 |
| 10| BIBLIOGRAPHY .......................................................................... | 39 |
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Camera orientations used in the study. (left) Aerial view. (middle) First-person view. (right) Following view</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Sequence of the character moving forward in the Aerial view. The rightmost image shows the user preferred stroke for this motion</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Top row shows the Turn right character motion with its user defined gesture. Bottom row shows the Turn left character motion with its user defined gesture</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>The graph shows the cluster result of the Aerial view user gestures. Each bar indicates the cluster groups. For the primary and secondary cluster group, the cluster center is shown</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>Sequence of a character performing Forward motion in the Following view</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>The Turn Right motion of the character. The arrows are the facing direction of the character at each frame</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>(Left) Turn right motion + Forward motion in the Aerial view. (Right) Turn right motion + Forward motion in the Following view. The starting point for the stroke is marked as a triangle and the end point is marked as a circle</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Top row shows the character performing a Right motion. Bottom row shows the character performing a Forward motion. Although the facing direction is different, the gesture for Left motion is identical. The white triangle indicates the start of the gesture and the gray circle indicates the end</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Average size of the Following motion gestures given by the users. (Top Left) Jump. (Top middle) Turn Left. (Top Right) Turn Right. (Bottom Left) Crouch. (Bottom middle) Flip Left. (Bottom Right) Flip Right</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>A character performing Flip motions. (Top row) Flip Left motion. (Bottom row) Flip Right motion</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>The graph shows the cluster result of the Following view user gestures. Each bar indicates the cluster groups. For the primary and secondary cluster group, the cluster center is shown</td>
<td>18</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>12</td>
<td>Sequence of a character performing Forward motion in the First-person view</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td>(Top left) Left motion in First-person view. (Bottom left) Left motion in</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Following view. (Top right) Turn Right motion in First-person view. (Bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td>right) Turn Right motion in Following view</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>The graph shows the cluster result of the First-person view user gestures.</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Each bar indicates the cluster groups. For the primary and secondary cluster</td>
<td></td>
</tr>
<tr>
<td></td>
<td>group, the cluster center is shown</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Velocity illustration for various gestures. (Left) Aerial view Crouch walk</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>gesture. (Second Left) Aerial-view Forward gesture. (Second Right)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Following view Crouch walk gesture. (Right) Following view Forward gesture.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The starting point of the gesture is indicated by the white triangle and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the ending point is indicated by the gray circle</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Sketch comparison. (Left) Sketch used in motion doodles [21]. (Right)</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Sketch from our study</td>
<td></td>
</tr>
<tr>
<td>A’s</td>
<td>Aerial View Primary Cluster Center Strokes</td>
<td>36</td>
</tr>
<tr>
<td>T’s</td>
<td>Following View Primary Cluster Center Strokes</td>
<td>37</td>
</tr>
<tr>
<td>F’s</td>
<td>First-person View Primary Cluster Center Strokes</td>
<td>38</td>
</tr>
</tbody>
</table>
Study in User Preferred Pen Gestures for Controlling a Virtual Character

1. Introduction

The pen-tablet PC was introduced to society in the late 1990’s. Today, the technology has reached a level where data and even games can be controlled in the palms of our hands. Unfortunately, pens are not widely used even though they have been around for years. One of the possible reasons for the lack of widespread use is the limitation on degrees of freedom that a pen can represent. Its counterparts, such as the keyboard and the game console controllers are flexible in representing the desired user input. The limitation can be seen by controlling a virtual character in a 3D environment. Both the keyboard and the game console controllers have the necessary degrees of freedom to move a character in 3D. For example, these controllers have been used effectively to allow users to move the character on a flat plane and make it jump vertically off of the plane. With a pen as an input device, the user is capable of moving the character on a plane because the device is primarily 2D, but it can be difficult to make the character jump, which requires a mapping to the 3rd dimension.

Ambiguity in the input is another reason for the lack of pen input applications. With a pen gesture as an input, the chance of multiple users having the same exact gesture for a particular command is small. For instance, if a command for executing a program is represented with a letter ‘R’, it is almost impossible to predict that every user will write the letter exactly the same in order to execute a program. There are gesture recognition algorithms [1,6, 7,13,16,21] that will recognize gestures within a certain
threshold. Even then, the gestures outside of the threshold may not get registered, leading to frustration for the user.

Although pen input is not yet as popular as its standard keyboard & mouse counterpart, there has been recent success. The handheld personal digital assistant (PDA) is becoming more like a desktop computer with its faster processors, gigabytes of memory, and internet capability. In the PDA, a stylus integrated Smartphone and even a stylus wrist watch [4] have been introduced to the commercial market. In addition, Nintendo has released their version of a stylus gaming device, Nintendo DS [11], which introduced a new dimension of game control to the industry. With the stylus devices continuing to gain recognition, it becomes important that we begin to examine how to design pen based interfaces for various applications.

Recent tablet PC devices are capable of capturing pressure, tilt, and hover of the pen in addition to position and time. These additional features could be used to increase the pen’s degrees of freedom. However, using all of these attributes could make things complicated for the user and increase the length of the learning process. So, how can we improve the quality of the control? In this paper, we intend to explore user preferences for character control gestures. Obviously human motion is complex and it is impossible to represent all human motion with a pen-based input. Our aim is not to study skeletal motions, but to focus on navigation and tasks motions that maybe typical for controlling a 3D virtual character in an action game or other virtual worlds [14].
2. Overview

Our goal is to identify control gestures that make intuitive sense to the users. One of the main requirements is that the user’s learning process for the gestures must be kept to a minimum. This can be achieved if the gestures are similar to the instinctive pen-input strokes presented by the user.

We have selected fifteen motions related to character navigation: (1) move forward, (2) move backward, (3) move left, (4) move right, (5) jump, (6) turn left, (7) turn right, (8) flip left, (9) flip right, (10) crouch, (11) move right and move forward, (12) move backward and move left, (13) crouch walk, (14) turn right and move forward, and (15) turn left and move backward. We presented the user with combination movements (11 - 15) besides the individual movements (1 – 10) to observe if there are differences in gestures between the two. We chose a set of motions typical of 3D applications, such as games and virtual simulation that require locomotion control. Note that the move left and the move right motions are performed with a side stepping motion. (This applies to these motions in the combination movements)

This paper also presents possible design guides for controlling character locomotion with a pen input device. We will indicate possible solutions for developing effective character control as well as objectionable elements that could create problems for users. Finally, we will discuss same existing input applications and how the gesture set compares to the find in our experiments.

The remainder of the paper is organized as follows: In Section 3 we will present the related work. In Section 4, we will describe the experiment. In Section 5, we will
explain the data uncovered by the experiment. In Section 6, we will compare existing pen input gestures sets to these identified in our study. Finally, in Section 7, we will propose some simple design guidelines and conclude in Section 8.
3. Background

The idea of using sketch gestures in computer graphics has been explored for several years. Zeleznik et al. developed SKETCH [15], which allowed a user to rapidly model primitive shapes from sketched gestures. Igarashi presented the sketching interface Teddy [9] for creating 3D geometric shapes from 2D freeform strokes. More recently, Davis et al. [3] developed an interface for drawing an animation of a character. An artist sketches a key frame of a character and the system determines and reconstructs possible poses from it. We are also interested in sketched gestures. However, our interest is understanding the appropriate set of gestures for character locomotion control.

Controlling the motion of a character can be created by sketching a path on the ground for the character to follow [5,9,21]. Thorne et al. [21] introduced an interface for sketching the character motion. Animations such as flip, hop, jump, and top-toe walking are played back according to the pen gestures drawn by the user. These gestures are incorporated into the path itself thus the character can execute desired animations while following the path. In this work, the authors designed the gesture set based on their own ideas of the correct set of motions. Our goal is to try to understand what gestures make the most sense to users.

Oshita [13] proposed a technique for interactive control using a pen-input device. Besides the position, the pressure and the tilt of the pen are used to animate more complex motions for the character. For instance, the character can bend and stretch according to the tilt direction and angle of the pen. The character can also jump and crouch according to the pressure of the pen. Although their research has similarities, our
goal is different. Our study focuses on identifying user preference in the pen gestures rather than the actual control of the character. More recently, Oshita [12] developed an interface for controlling various human motions with strokes. By representing the link between the virtual character and target objects with strokes, Oshita was able to intuitively control the character performing complex motions.

Current commercial pen-tablet software, such as the Nintendo Super Mario 64 DS [18], attempt to use a pen control for the character, but many gamers have concerns with the difficulty of controlling the character [19,20]. We hope to begin to shed some light on how to design gesture sets to better accommodate user control with pen devices.
4. Experiment

The experiment was conducted in the Interactive Graphics and Visions Lab at Oregon State University. Thirty one adults were recruited to participate in this study. Each experiment lasted approximately 20 minutes.

4.1. Experimental Goal

Successful input devices such as the keyboard or game controllers generally map key presses or button presses to user intentions. These mappings are often fairly intuitive. For example, the up-arrow key on the keyboard and up button on the game controller generally makes the character move forward or upward. On the other hand, a pen input device generally presents ambiguities. Unlike a key on the keyboard or a button on the game controller, users are not bound to a fixed input stroke. The user has freedom to draw unique gestures for a particular motion. Thus, a gesture for moving the character forward may vary from user to user. In order to eliminate ambiguities, there should be some constraint on the input. However, constraints lead to undesirable gesture input that does not correlate well with the intentions of some users. Nevertheless, we must identify a gesture set for pen input devices if we are to allow users to control virtual characters.

Our aim was to try to uncover a set of control gestures for virtual characters that relate well with user intentions. Instead of drawing conclusions from research, we preferred to uncover answers from field studies. We hypothesized that if we analyzed large number of user defined gestures for specific character motion sequences, we could identify similarities within the gestures. We hoped to find answers that could be used to guide the development of gesture sets for controlling character locomotion.
4.2. Experimental Setup

In our experiments, users were asked to draw a gesture stroke on a tablet that best represents the movement of a virtual character. The movement of the character is pre-defined. Motions are played back on the monitor placed in front of the user and the user records pen gestures as if they are controlling the character in real-time.

We used a Toshiba Portege 1.7 GHZ, 512 MB RAM tablet PC for the recording device. A simple application written in Visual C-Sharp was created to interactively record the strokes of the user.

The simulation video was created using the animation package, Face Poser, and the game level editor, Hammer, which was included in the Half-Life2 engine (HL2) [22]. The motions were chosen from the motion library included in the HL2 game development kit.

4.3. Experimental Methods

Once the subject was prepared to start the study, the motion video was played in the following order: move forward, move backward, move right, move left, jump, turn right, turn left, crouch, flip left, flip right, move right and move forward, move backward and move left, crouch walk, turn right move forward, and turn left and move forward.

We also incorporated different perspective views (Figure 1); first-person, aerial, and...
following views of the motion. We anticipated that there would be some disparity in gesture stroke for the same motion when viewed from different perspectives. For each video, there was a short description explaining the motion, and then the motion sequence was played twice. The purpose of this method was to reduce bad data. Although there was an explanation of the motion before it was played, the users did not know how the character moved or what the exact motion looked like. Playing the motion only once could surprise the user which could lead to delayed and erratic strokes that would not represent the user’s intended stroke. Furthermore, in games and other environment where a user is controlling a virtual character, there are no surprises in the actual control of the character. The user knows and anticipates the character movement before executing the control. Thus, playing back the motion twice reduces erratic strokes and gives us a better chance of recording good data.

Each motion gesture stroke is stored in a file as position, speed, and pressure samples.
5. Analysis

For each motion, the collected gestures are clustered into groups according to position and velocity samples. We did not use the pressure information. All of the gestures were drawn with the same pressure and did not show any further information. We use a subpixel contour matching [17] algorithm as a measure of the cost difference between the strokes. The algorithm matches two curves or contours using dynamic programming. As it computes the matching, a measurement of the deformation between the two curves is obtained, which is used as our cost value. The costs are used to construct a cost matrix, that is then used in a subtractive clustering method [2]. This method determines the number of clusters and the cluster centers for $K$ elements. The method is an efficient algorithm that considers each data point as a potential for being a cluster center. The potential is considered as a distance function between two data points. The cluster group is computed based on the density of the data, and the cluster center is calculated based on the density of the surrounding data points.

5.1. Aerial Perspective

For the aerial perspective (Figure 1, left), the camera was located above in the back and to the right of the character. In this view, we found that most motion gesture drawn along the character’s motion path in respect to the camera view. Figure 2

![Figure 2: Sequence of the character moving forward in the Aerial view. The right-most image shows the user preferred stroke for this motion.](image)
demonstrates the preferred stroke made by the user for Forward motion. The preferred stroke is defined as the cluster center for the largest cluster for that motion. We can clearly see the relation between the character’s motion path on the screen and the stroke. Turning motions, however, were not tied to the character’s path. In Figure 3, the top row shows the Turn right motion with its users preferred stroke. The bottom row is the Turn left motion. The left most figure displays the initial facing direction of the character in each motion. The figure on its right displays the final facing direction of the character along with the arc between the initial and final facing direction. This arc is consistent with the user defined stroke. A similar connection can be found in all user gestures for the aerial view.

**Figure 3:** Top row shows the Turn right character motion with its user defined gesture. Bottom row shows the Turn left character motion with its user defined gesture. The starting point of the user defined gestures on the right most column is shown as a white square, and the ending point is shown as a gray circle. The dotted arrow follows the orientation of the character’s facing direction before and after the turn.
Figure 4 shows the result from clustering the user gesture data in the Aerial perspective. The dominance of the primary cluster group was apparent. In each of the tested motions, the primary cluster group enveloped over 50% of the user gestures and was actually greater than 70% for the majority of the clusters. For instance, the primary cluster for Turn Right motion included 93.5% of the 31 strokes with the gesture #17 being the cluster center (see Figure A-6).

The primary cluster center for directional motions (Forward, Backward, Left, and Right), which are illustrated in Figure A-1 through A-4, have common similarities. The length of the stroke is fairly uniform. It also happens that the length of the stroke is approximately equal to the length (in pixels) of the side of the brick wall shown in Figure 2. Since the motion path for the directional motion is roughly identical to the length of

![Aerial - Position & Velocity](image)

**Figure 4**: The graph shows the cluster result of the Aerial view user gestures. Each bar indicates the cluster groups. For the primary and secondary cluster group, the cluster center is shown.
one of the side wall, the length of the strokes is relative to the amount of displacement of the character in the screen.

Besides the directional motions, the user defined gestures for other motions resulted in smaller strokes when compared to the other two perspectives. Primary cluster centers strokes for the Jump, Turn, Crouch, and Flip motions are illustrated by Figure A-5 through Figure A-10, note that they are small in size when compared to the gestures for directional displacement. In fact, most of the recorded gesture strokes for these motions were approximately the same size as the cluster center stroke. We believe this was caused by the amount of body movement in these motions. Looking back at Figure 1, there is a significant difference in character size between the Aerial perspective and the Following perspective. Thus, the character motion in the Following view is more dramatic then the character motion in the Aerial view. This may explain the smaller strokes for the Jump, Turn, Crouch, and Flip motions. The directional motion gestures were relatively the same in length for all perspectives. This is because the actual distance traveled by the character does not change among the perspectives. In Figures A-13 and A-14, a similar pattern is observed. The Turn portion of the gesture is small and the length of the Forward motion gestures is similar to that in Figures A-3 and A-4.
5.2. Following Perspective

In Following view, the camera position was located right behind the character. As expected, the gestures captured for this view were relative to the camera position and orientation. Figure 5 shows the character performing the Forward motion. Unlike the Aerial view, the stroke is straight, which pertains to the character moving straight into the screen.

One of the distinct differences between the Aerial view and Following view is the facing direction of the camera along the character’s facing direction. In the Following view, the camera always turns with the facing direction of the character, shown in Figure 6. This caused the user to draw gestures relative to the camera’s facing direction. In the Aerial view, the camera was placed at a certain offset above the character looking down with a fixed direction. This influenced the users to draw gestures that were relative
to the camera’s view, except the Turn motion gestures (Figure 3). For the Turn motion, the gestures generally started from the character’s initial facing direction and ended at the

![Figure 7: (Left) Turn right motion + Forward motion in the Aerial view. (Right) Turn right motion + Forward motion in the Following view. The starting point for the stroke is marked as a square and the end point is marked as a circle.](image)

final facing direction. But, all other motion gestures in the Aerial view were sketched along the path of the character movement relative to the camera. The gestures in Figure 7 are gestures from the same motion sequence viewed from different perspectives. The sequence shows Turn right motion followed immediately by the Forward motion. In the Aerial view (Figure 7, Left), the change of facing direction clearly persuaded the user to make the arc curve from character’s starting facing direction to the final facing direction. This characteristic is also found in the Following view (Figure 7, Right). The difference is in the Forward motion gesture. In the Aerial view (Figure 7, Left), although the character’s turn changes the facing direction, the gesture is not drawn in respect to the new facing direction. The character is indeed performing a Forward motion based on its facing direction, but the user defined Forward gesture is performed along the path of the
character movement. In Figure 8, the top row shows the Right motion sequence with the character’s facing direction towards the Northeast, while the bottom row shows the Forward motion (Forward motion part of the Turn right motion + Forward motion sequence) with the character’s facing direction towards the Southeast. Although the facing directions of the character are different, the gestures are identical. In the

![Figure 8:](image)

Following view, the facing direction after the Turn right motion is straight into the scene, illustrated in Figure 6 (left). Naturally, the subsequent Forward motion translates to a straight stroke from bottom to top of the gesture area, shown in Figure 5. The combined gesture should reveal a curve and a straight line, which corresponds to Figure 7 (Right). In the Following view, the camera’s movement along the facing direction of the character played an important role in determining the shape of the gesture.

As mentioned in the Aerial view section, the character’s body displacement is related to the length and the size of the gesture. In the Following view, Jump, Turn, Crouch, and Flip motion gestures are larger than in the Aerial view. Although the
primary cluster center strokes for these motions are approximately the same size as the Aerial strokes, most other strokes are larger. Figure 9 shows the average stroke size in the Following view given by the users for these strokes. It is clear that the gesture size

![Figure 9: Average size of the Following view motion gestures given by the users. (Top Left) Jump. (Top middle) Turn Left. (Top Right) Turn Right. (Bottom Left) Crouch. (Bottom middle) Flip Left. (Bottom Right) Flip Right.](image)

for the Following view is much larger than the gestures in the Aerial-view.

In the Following view, the distinction between the two Flip motion gestures is apparent. Close inspection reveals Flip Left gesture being a circular shape compared to Flip Right gesture being a flat oval shape. This difference in the gesture shape may be caused by the difference in the flip sequences, as shown in Figure 10. We have

![Figure 10: A character performing Flip motions. (Top row) Flip Left motion. (Bottom row) Flip Right motion.](image)
repeatedly mentioned that the body displacement of the character in the screen affects the outcome of the gesture. We also believe that the shape of the gesture in Flip motions is affected by the displacement of the character in the screen. The motion sequence of the Flip Left and Flip Right is not symmetric. Flip Left motion had a full extension of the character’s body when it performed the flip. One the other hand, Flip Right motion was performed with the character in a tucked position. The full-circular characteristic of the Flip Left motion gesture (Figure 9, bottom middle) relates well with the full extension of the characters body. The tucked-in roll motion of the character in Flip Right motion is well described by the squished oval shape gesture (Figure 9, bottom right). We see a similar characteristic in the Aerial view, shown in Figure A-9 and A-10. This example is in agreement with an hypothesis that the user relates their gesture stroke with the displacement of the character’s body in the scene.

Overall, the primary cluster gesture group dominated each motions tested in the Following view. By observing Figure 11, Flip Right motion was the only motion that

![Third - Position & Velocity](image)

**Figure 11:** The graph shows the cluster result of the Following view user gestures. Each bar indicates the cluster groups. For the primary and secondary cluster group, the cluster center is shown.
had its primary cluster group below 50%. The secondary cluster group for the Flip Right motion, with its cluster center stroke similar to the gesture (Bottom Right) in Figure 8 in size and shape, contributed to 35.5% of the total gestures in the motion. In both Following and Aerial views, Flip motions display disparity in the gesture data compared to the other motions. In the Aerial view, the primary cluster group for Flip Left motion consisted of 74.2% of the gestures and 51.6% for Flip Right motion. In the Following view, Flip Left motion comprised 54.8% of the gesture and 45.2% for the Flip Right.

The cause of disparity in flip motion data could be due to the intricate motion of the character, quickness of the motion, and/or the motions were not familiar to the users.

### 5.3. First-person Perspective

There is a good chance that the user is inexperienced in controlling the character with a pen in the First-person perspective compared with the other two views. Also, in the First-person view, there is no physical representation of the character’s body in the screen. The only reference the users have for the motion is the camera movement and its facing direction. Thus, we hypothesized that the gesture datasets would reveal random results. But instead, the performed datasets exhibited a discernible outcome. The First-person motion gestures were similar to the Following gestures in many ways. Figure 12 illustrates the character performing the Forward motion in the First-person view.

![Figure 12: Sequence of a character performing Forward motion in the First-person view.](image)
Although the character is not displayed, the amount of distance the character traveled is the same for the motion sequence. Thus, the stroke length between the Following and the First-person view of the Forward motion is very similar. For instance, the Forward motion cluster centers for the Following view and the Aerial view showed only 22% difference in length, and for the Backward motion cluster centers, the difference was 8%. This similarity can be seen in all directional motions (Forward, Backward, Left, and Right). Besides the stroke length, the stroke direction is also similar. Comparing the Following figures T-1 through T-4 and First-person figures F-1 through F-4, we see that the direction of the strokes is nearly identical. This is the result of the First-person view’s character facing direction and the camera movement performing exactly the same as the Following view. Figure 13 shows this similarity. Figure 13 also compares the Turn Right motions. The gesture for these motions, shown in Figure T-6 and Figure F-6,

![Figure 13:](image)

are also nearly identical.

The result of the First-person view motion gestures has shown that it is possible to identify an intuitive gesture set for character control in the First-person perspective. The users were able to create meaningful gestures even if the character body was not present in the screen. As long as the camera motion and the facing direction of the character
exist, the users are able to represent the gesture for the motion. Figure 14 illustrates that for each motions, the primary cluster centers comprised 55% of the user gestures and greater than 70% for the majority of the clusters.

**Figure 14:** The graph shows the cluster result of the First-person view user gestures. Each bar indicates the cluster groups. For the primary and secondary cluster group, the cluster center is shown.

### 5.4. Discussion

From our experiments, we have found that it is possible to identify a “common” gesture features for each basic locomotion task from various view configurations. The most important motion feature appears to be the overall displacement of the body as it performs the motion. For Walk, the key motion feature is the direction of the character heading. For Jump, it is the upward and downward motion of the body. The arm and leg movement during the motion was not apparent in most gestures. However, the overall displacement of the body is not the only feature that is useful in determining the user defined gestures. In the Aerial view, the position of the character during the motion...
sequence played an essential role in determining the gesture outcome. In the Following and the First-person view, the camera movement along the character’s facing direction guided the shape of the gesture.

The velocity information showed a relationship between the gesture speed and the character speed. Figure 15 illustrates the velocity information for Crouch walk motion and Forward motion. The lighter color indicates the slower velocity and the darker color indicates the higher velocity. For the Crouch walk motion, both Aerial and Following views showed high velocities for the initial crouching gesture, but the slowed down for the forward walking gesture. On the other hand, Forward motion gesture for both Aerial and Following views showed high velocities. This characteristic was consistent with the character’s slow movement in the crouch walk sequence and faster movement in the forward walking sequence.

In Forward motion gestures shown in Figure 15, the transition of speed is shown. The gestures initially started slow, quickly attained high speed, and slowed down again towards the end. This transition was common among the directional motion gestures; Forward, Backward, Left, and Right. At first, we believed that this was consistent with
the velocity change of the character, but we also realized that this could be a side effect for sketching. In order to clarify this ambiguity, further research much be conducted.

One of the compelling results we’ve found is the length of the stroke was dependent on the displacement of the body motion of the character. For instance, in the Jump motion, the body of the character travels farther in the Following view than in the aerial view. This is caused by the character’s body size between the views. Since the camera is located farther away in the aerial view than the Following view, the character appears smaller. As a result, the user’s Jump gesture in Following view is longer than the Jump gesture in aerial view.

The experiment also revealed interesting patterns in flip motion gestures. The character motions for the right and left flip are different. For the left flip motion, the character performs a full-cartwheel, which is a full extended body motion. On the other hand, the right flip is a tucked roll motion, which is a contracted body motion. The users distinguished between these motions by sketching the left flip with large full-circular stroke, and squished elongated ellipse stroke for the right flip. In addition to the shape of flip gesture, the initial position of the stroke did not reflect the facing direction of the character. For example, if we are looking down at the character that is facing up towards the top of the page, the gesture for left flip does not necessarily start from 12 o’clock position and move to the left. The gesture started at a random position, but the direction of the stroke is consistent with the direction of the flip.

Unlike the flip gestures, the turn gestures indicated a close relation between the initial facing direction of the character and the initial position of the gesture. Applying the same example above, the left turn gesture starts from the 12 o’clock position to the
final facing direction. The turn gesture could be a source of ambiguity when compared to the flip gesture. Although we only tested a quarter turn (90°), the gesture exhibited a distinct relation between its arc length and the degree of turn. From this, we expect the full turn gesture to have circular characteristic, which could present an ambiguity when compared to the flip gestures.
6. Comparison with Existing Interfaces

The results of the study revealed some interesting relationships between gestures, character, and camera, as well as differences when compared to gesture strokes used in previous researches.

In 1996, Zeleznik et al. [15] presented a way for the user to generate and edit geometric shapes using gesture strokes. The *SKETCH* interface defined distinct gestures for selected shapes. These gestures are built according to the important visual features of the shape. For instance, three perpendicular lines created a cube. Gestures were also defined for editing. The resizing required over-sketching of one edge with two parallel lines drawn in opposite directions. The reasoning behind each gesture, the important visual feature, is in agreement with what we found in our studies. In our studies, most of the users preferred gestures that reflected the important motion feature of the character, the direction of the body movement. For example, in jump motion, the important motion feature is the body moving vertically up and down. The users related this feature with an up and down spike-like gesture. Our finding suggested that the important motion feature, in general, is helpful in determining the shape of the gesture. Although our finding is similar to Zeleznik’s reasoning for how they built their gesture, the important visual feature, how we arrived at this reasoning is different. We focused our study on user preference. From this, our findings suggest that motion gesture could be described by the principle body direction. This may not apply to the *SKETCH* interface, where the reasoning behind each gesture was not described according to the user’s intuition.

In the *Motion Doodles* [21] interface, the application contains a gesture vocabulary for 18 motions. The user gesture stroke is compared to these vocabularies
and a resolved motion from its recognition process is applied to the character. Walk, Jump (in place stomp), and flip are motions defined in the vocabulary. For Walk, the gesture is described as a stroke with a series of parabolic curves wedged against each other in a horizontal manner (Figure 16, top-left). Jump (in place stomp) is described as a vertical spike similar to an up-side down “V” (Figure 16, middle-left). Flip is expressed as either a Back or a Front Flip. For Front Flip, the stroke is defined as a bell-shaped curve starting from left to right with two loops in it. Back Flip is represented as a bell-shaped curve starting from right to left with a single loop in it (Figure 16, bottom-left).

Our study has shown that the users prefer using a straight line for the walking motion. Only 1 out of 31 strokes illustrated a curvy characteristic. This may suggest that the users are more concerned about the overall movement of the character than how the character is walking. For the Jump gesture, the study showed that the users described the motion as a vertical spike. This is consistent with the in-place jump stroke used in the Motion Doodle vocabulary. For the flips, our study used a right flip and a left flip, which are equivalent of Motion Doodle’s front flip and the back flip respectively. In both cases, the flip strokes from our studies

![Figure 16: Sketch comparison. (Left) Sketch used in motion doodles [27]. (Right) Sketch from our study.](image)
showed similar characteristics to the flip strokes defined in the vocabulary. Users favored the loop stroke to represent Flips. For the left flip, the stroke started from the right to left. The right flip started from the left to right. Figure 16 shows the stroke comparisons. However, users rarely used multiple loops for a flip.

Kolhoff et al. [10] developed a method to control the gait animation of a virtual character using two pens. Each pen represents one leg of the character. The user moves the character by striding over the tablet surface with pens. In our study, the users opted to use the main body displacement as the control variable for the character. This could be the result of using only one pen, or it could also be the result that the users relate better with the main body displacement than the leg motions. Because we did not supply multiple input pens, we cannot comment on preferences for a single or double pen input.

In Pen-to-Mime, Oshita [13] used pen pressure for two purposes; jump and duck. The determining factor is the speed. If the pen is pressed slowly, the character ducks, but if the pen is pressed and released quickly, the character jumps. The height of the jump is based on the pressure level at the moment when the pen is released. However, our study indicated that using pressure information for the control is not necessarily a natural approach. We expected the pressure information to have relevance in all of our tested motions, especially for jump and duck (crouch), however, the data did not reflect any relationship between the pressure and the motion. The users choose to use spike-like gesture stroke to represent both jump and duck motions. There are probably many explanations for this finding. One explanation could be that we are better at controlling the direction of the pen tip than controlling the pressure of the pen. This could be the reason why many users preferred to write out the jump and duck motions instead of using
the pressure. Besides the pressure, velocity information was also used in Oshita’s interface. The velocity of the pen determined the velocity of the character. This decision is intuitive and our result supports this decision. In most of our motions, the users moved the pen with a speed relative to the character speed on the screen. For example, the normal walking motion and crouch walking motion have two very different speeds, and our result from the study reflect these differences in the velocity profiles of the gestures.

Oshita [12] recently proposed an interface for controlling complex humanoid motion with a pen stroke. The stroke gestures are made for intuitive and interactive motion controls. By connecting the character’s body part and an object with a stroke, a motion is generated according to the context of the scene. For example, by sketching a stroke between the character’s hand and a ball on the ground, the character picks up the ball. The control is intuitive and effective as it is shown by their user experiment. This approach is clearly aimed at higher level task specification.
7. Design Approach

This section describes possible design options for developing primary motion control for a virtual character with a pen based input method. Indeed, there are applications whose aim are to demonstrate complex humanoid motion [12], in which case our implications may not apply to them.

An important consideration in designing a pen based input control for virtual character is simplicity. We hypothesize that, one could take simplicity into account by developing the control according to the user’s intuitive preference. Our study has shown common features exhibited by the users that could assist the design of pen-input character control. We have broken this section into two parts; fundamental movement design and advanced movement design.

7.1. Fundamental Movement Design

The motion described here are directional motions and rotational motions. The directional motions are move forward, move backward, move right and move left. The rotational motions consists of turn right and turn left. These are inevitable basic motions for controlling a virtual character. Without these motions, the character would not be able to freely move around in a virtual world.

For the directional motions, we propose the controlling gesture should possess a straight line characteristic. The direction of the stroke should determine the walking direction of the character in its facing direction. We also propose the speed of the character be related to the speed of the stroke it was drawn by.
In the case of rotational motions, the gesture should resemble a curve characteristic. The direction of the curve should determine the direction of the turn. We also recommend that the turning stroke be initiated at a position with respect to the facing direction of the character. For instance, if we are looking at a character from above and it is facing up towards the top of the page, and if the character performs a left turn, the gesture stroke, in general, should resemble a counter-clockwise curve path from 12 o’clock to the final facing direction. The speed of the turn should reflect the speed of the stroke.

There are a couple of design choices that may cause difficulties for users. One of which is to incorporate features outside of straight lines for walking motion. Although using wavy or blocky characteristic strokes may represent the foot stepping of the character, hardly any user used this technique. The users instinctively selected to use the straight line approach. Another possibly confusing approach is to use pen pressure information. We have found that in all of our tested motions, the user’s information on pen pressure was inconclusive. We do not address pen pressure for this reason and would not recommend its use for character motion control.

7.2. Advanced Movement Design

The advanced movements consist of jump, crouch and flip. In the jump motion, a design should have the stroke reflect a spike characteristic sketched in an up and down manner. This should apply to first, Following, or aerial perspectives. The design should also incorporate size. The size of the spike should indicate the strength of the jump. The longer the stroke, the higher the character should jump. In jump motion, the pen pressure
information may appear relevant, but we suggest avoiding the use of pressure. Only a single subject employed a method that used pressure primarily for the gesture.

The crouch motion should have a downward stroke characteristic. In our study, most of the users used the downward gesture to represent the crouch. This feature was consistent in all of the perspectives. However, this may cause an ambiguity with the walk backward gesture. Resolving this similarity could be difficult and may require additional research.

The flip motion should have a loop. Depending on the direction of the stroke, left or right flip should be selected. Unlike the turn, the orientation of the stroke is irrelevant to the facing direction of the character. The loop and its shape are the relevant features. A variety of flips can be represented by the deformation of the loop. For example, flat and compressed loop can symbolized a tucked flip, or a roll. A full circular loop could, for example, describe a cartwheel.
8. Summary

We presented a study for determining the user preference for controlling a character using a pen input. By collecting gestures directly from the user, we were able to analyze and hypothesize about the most intuitive gestures for controlling a character using a pen. We believe our findings will assist in developing an instinctive control system that relates well with users.

8.1. Conclusions

For the motions we have tested, the user defined gestures were described based on several factors. The overall displacement of the character’s body during a motion was one of the factors. The diminutive motions of the arms and legs in motions were not discernible in any gestures. The camera movement along the facing direction of the character greatly determined the gesture shape in the Following view and the First-person view. Unlike Following and First-person views, the gestures in the Aerial view did not associate the facing direction, except in Turn motion. The character could be facing North or South, and if it moves to the left, the gesture shows a left motion. In the Turn motion, the arc characteristic of the gesture approximately matches the angle the character turn from the initial facing direction. Another factor was the velocity. The gestures associated the character’s speed with its gesture speed. In slower character motion such as the crouch walking, the gestures demonstrated slower velocity, and in high velocity motions such as the forward walking, the gestures showed higher velocity. The pressure information did not demonstrate any significant attribute. All of the gestures were sketched with the same pressure and did not show any other information.
This study presents a first step in beginning to understand the proper gestures for character control. The input mechanism is essential in any environment a user interacts with, especially in an environment where a user controls a virtual character. Seamless and intuitive input is required to make users feel as if they are the character in the virtual world. With a pen as an input, this becomes more difficult since the pen stroke is a freeform gesture. Each user has their way of illustrating a particular control. Unfortunately, the current pen input mechanisms are built around the preference of the developer. This approach requires that users learn and adapt to the developer’s design. Instead, we approach this problem from the user’s perspective. Through investigating numerous pen gestures from the users, we were able to study common characteristics that can be used to develop an input system.

This project is a step towards understanding the effective pen input mechanism for controlling a virtual character. More studies must be conducted in order to verify the data and ultimately design and test a character control gesture set.

8.2 Future Work

There are several cases we have overlooked. In a hand held device, such as the Nintendo DS [15], the dimension of the touch screen is much smaller then the touch screen used in this study. Since the users may not have the space needed to complete their strokes, this could lead to different results. For example, in Forward motion, the user may have to make several small strokes instead of one large stroke. There is also Turn motion expansion. The motion could be expanded to test higher degrees of turn to get different results. Another case we did not test is a propulsive jump. In a game environment, jump motion usually involves directional momentum. Our study only
focused on an in-place jump. There is also the Flip motion case we did not test for the First-person view. We hope to revisit these cases in the future.
9. Acknowledgements

I would like to express my deepest gratitude to my parents for their endless support and inspiration throughout my life. Without them, I could have never gotten this far physically, mentally, and educationally. The sweat and tears they’ve poured out in order for me to achieve my goal in every area of my life, especially in academics, I thank them with all of my heart. I’m indebted to have my brother and my sister by my side to lift me up and encourage me to walk the path I’ve set forth.

I want to thank my advisor Dr. Ron Metoyer for his support and guidance. Since my undergraduate studies, he has inspired and motivated me to further my knowledge in Computer Graphics. His persistent guidance and encouragement has shaped and refined my capability as a student and as a person. I would like to thank Dr. Eugene Zhang for his passion and support in teaching me new material. His support and insightful contributions have always been priceless. I also like to thank Dr. Bailey for expanding my knowledge and equipping me with new tools in Computer Graphics. I express my gratitude to these three distinguished professors for their time and efforts to serve on my committee. Thank you very much.

I would also like to thank all of my friends at Corvallis, Hawaii, and Japan. Their enduring support has been my strength and perseverance.

Lastly, I want to thank all of the participants who have made this project successful. Thank you for all of your time and efforts.
Aerial View Primary Cluster Center Strokes

Figure A-1: #17 for Forward motion.
Figure A-2: #28 for Backward motion.
Figure A-3: #23 for Left motion.
Figure A-4: #23 for Right motion.

Figure A-5: #25 for Jump motion.
Figure A-6: #17 for Turn Right motion.
Figure A-7: #21 for Turn Left motion.
Figure A-8: #10 for Crouch motion.

Figure A-9: #14 for Flip-left motion.
Figure A-10: #16 for Flip-right motion.
Figure A-11: #4 for Right motion + Forward motion.
Figure A-12: #23 for Backward + Left motion.

Figure A-13: #14 for Crouch walk motion.
Figure A-14: #15 for Turn Right + Forward motion.
Figure A-15: #24 for Turn Left + Forward motion.
Following View Primary Cluster Center Strokes

Figure T-1: #21 for Forward motion.
Figure T-2: #5 for Backward motion.
Figure T-3: #10 for Left motion.
Figure T-4: #7 for Right motion.

Figure T-5: #24 for Jump motion.
Figure T-6: #10 for Turn Right motion.
Figure T-7: #10 for Turn Left motion.
Figure T-8: #29 for Crouch motion.

Figure T-9: #14 for Flip-left motion.
Figure T-10: #14 for Flip-right motion.
Figure T-11: #22 for Right motion + Forward motion.
Figure T-12: #8 for Backward + Left motion.

Figure T-13: #14 for Crouch walk motion.
Figure T-14: #27 for Turn Right + Forward motion.
Figure T-15: #22 for Turn Left + Forward motion.
First-person View Primary Cluster Center Strokes

Figure F-1: #19 for Forward motion.
Figure F-2: #9 for Backward motion.
Figure F-3: #29 for Left motion.
Figure F-4: #4 for Right motion.

Figure F-5: #8 for Jump motion.
Figure F-6: #8 for Turn Right motion.
Figure T-7: #7 for Turn Left motion.
Figure F-8: #18 for Crouch motion.

Figure F-9: #6 for Right + Forward motion.
Figure F-10: #18 for Backward + Left motion.
Figure F-11: #22 for Crouch walk motion.
Figure F-12: #5 for Turn Right + Forward motion.

Figure F-13: #22 for Turn Left + Forward motion.
BIBLIOGRAPHY


4. Fossil Wrist PDA. http://www.fossil.com/jump.jsp?itemID=2208&itemType=CATEGORY


19. Super Mario 64 DS User review 1.  

20. Super Mario 64 DS User review 2.  
[http://www.1up.com/do/reviewPage?cId=2019506&did=2](http://www.1up.com/do/reviewPage?cId=2019506&did=2)


22. Valve Software.  