Analysis of gesture-based scroll control for a living room photo browser

Hongchan Choi, Sukwon Chung, Tyler Crimm, Jieun Oh, Matt Sencenbaugh
{ hongchan, quantasw, tman7000, jieun5, msencenb } @ stanford.edu

Mar 9, 2012

1 INTRODUCTION

In this study, we asked users to scroll through a linear (1-D) horizontal menu of items, labeled from 1 to 50, using our newly designed "swipe" gesture. This gesture, similar to turning pages on a book and analogous to a finger-swipe gesture used on multi-touch devices (but instead, using the entire fore-arm), is a modification from our previous "elbow-pivot" gesture, which users found to be unintuitive based on our Prototype I presentation. This new swipe gesture allows users to navigate forward (swiping using right hand/arm with right-to-left motion) and backward (swiping using left hand/arm left-to-right motion), at three different scrolling speeds that results in translation of either 1, 3, or 10 items in the cover flow.

We had two overarching motivations for the user study on our living room photo browser, specifically dealing with navigating a cover flow menu. We believed that adding physics and animation into the GUI (e.g. velocity-dependent scroll acceleration) would not only provide a natural "feel" to the gesture, but also allow users to utilize scrolling behaviors they are already familiar with. Second, we felt that having an auditory feedback – in lieu of (or as an alternative to) having a real-time visual display of the skeleton – could enhance system feedback. To test these assumptions, we addressed the following questions in our study:

1. Is having animated physics in our graphical user interface helpful for users to navigate the menu to select desirable items? If so, how is it helpful? How do users feel the system without it?

2. Is having auditory feedback in GUI helpful for users? What's the best level or scope of the auditory feedback? (event-based sonification that's triggered only when a gesture is recognized vs. continuous sound that maps to the position of relevant joints)

3. What are the optimal boundary (threshold) values calculated from the swipe gesture according to which we can categorize the gesture into three distinct speeds (slow, normal, fast) – resulting in scrolling of 1, 3, or 10 items?

We hypothesized that (1) displaying animated transitions with physics can convey critical information such as where users are in the menu and what the available action are, and that (2) visual or auditory implication of "gesture-activating" elements can more efficiently direct users to appropriate actions and further provide feedback on how the system is registering the user's gestures.

2 Methods

Participants

We recruited 9 subjects, many of whom currently live in a family setting. The following is a more detailed demographics:

- s1: living on campus, a PhD student in Computer Music, 30s, male, from Canada

- s2: living on campus, a housewife, 30s, female, from Korea
- s3: living on campus, a PhD student in Management Science and Engineering, 20s, female, from Korea
- s4: living off campus, a PhD student in Electrical Engineering, 30s, male, from Korea
- s5: living on campus, a husband, 40s, male, from the US
- s6: living on campus, a wife, 40s, female, from the US
- s7: 11 year old daughter, female, from the US
- s8: living on campus, age 21, female, undergraduate student
- s9: living on campus, age 20, male, undergraduate student

Participants were recruited by cupcakes and coffee. :)

System Setup

Our setup was a mockup GUI based on C# XNA framework, with auditory feedback implemented using the ChucK programming language (http://chuck.cs.princeton.edu/). These two components are connected using the low-latency Open Sound Control protocol (http://opensoundcontrol.org/). Note: For the purpose of this user study only, we isolated and encapsulated the "scroll menu" functionality from our full NodeJS + front-end architecture to simplify debugging and parameter changing. We tested different design variants of our system according to the following 2x2 matrix:

variable 2: auditory feedback	variable 1: animated transitions and physics: no	variable 1: animated transitions and physics: yes
no	(1) no animated transitions, no physics no auditory feedback	(3) animated transitions, physics no auditory Feedback
yes	(2) no animated transitions, no physics auditory feedback	(4) animated transitions, physics auditory feedback

Environments

The study was conducted in the family living room of the participant's home. For undergraduate participants, study was conducted in a dorm lounge.

Tasks

For each of the four conditions (1, 2, 3, 4 described above in the 2x2 matrix):

- We show participants a linear menu of items labeled from 0 through 49. The default item in the center of screen is 25.
- We ask participants to navigate to a "far away" number, say 49, and select that item.
- From there, we ask participants to navigate to a "close" number, say 46, and select that item.
- Repeat the navigate-select task several times, for a range of distances.

Data Collected

We will record the following data for each subject, condition, and task:

- subject ID
  - condition: 1, 2, 3, 4
    - distance and direction instructed to scroll (e.g. "right" 10 items, if going from 25 to 35)
      - time it took from starting gesture to selecting the correct item
      - number of "under" shooting (i.e. didn't quite reach the target item)
      - number of "over"shooting (i.e. went to far, and passed the target item)
      - "velocity" of the swipe gesture(s) used
    - follow-up questions/ survey to inquire experience

To ensure that we capture all the data needed, our study sessions were video-taped.

After test, we additionally requested users to take three different speed scrolling motions (fast/normal/slow) to record the value defined by the velocity metric function:

$$\text{Value} = 400 \times \frac{\text{hand displacement in } x\text{direction}}{\text{length from shoulder to hip}} \times \frac{1}{\text{number of frames used to move hand}}$$

This is the velocity metric we used to detect user's swiping velocity. To make the metric independent from user's height, we normalized the hand displacement value with the length from shoulder to hip. By doing this, tall users should move their hand more than short users. To calculate average velocity, we divided the normalized hand displacement value by the number of frames used to move hand. The number "400" is the hand-picked constant to make most of final values in the range of [0, 150].

3 RESULTS

3.1 Number of trials for different settings

In this section, we divided velocity metric values into three ranges. If [Value $>$ 120], the motion will be detected as fast motion and results in translation of 10 items in the cover flow. If [120 $\geq$Value $>$100], the motion will be detected as normal speed motion and 3 items will be translated. If [100 $\geq$Value], it will be detected as slow speed motion and only one items will be translated. The numbers, 100 and 120, are hand-picked and we will suggest more appropriate number in the subsection 3.2.

- s1: living on campus, a PhD student in Computer Music, 30s, male, from Canada

	Animation	Sound	1st (2~5 items)	2nd (7~9 items)	3rd (18~22 items)
1	No	No	30->32 2 trials	32->42 1 trial	42->21 5 trials
2	No	Yes	21->17 4 trials	17->27 1 trial	27->47 9 trials
3	Yes	No	47->43 4 trials	43->33 10 trials	33->10 9 trials
4	Yes	Yes	10->14 4 trials	14->24 4 trials	24->44 2 trials

- s2: living on campus, a housewife, 30s, female, from Korea

	Animation	Sound	1st (2~5 items)	2nd (7~9 items)	3rd (18~22 items)
1	No	No	42->40 2 trials	40->30 1 trial	30->37 11 trials
2	No	Yes	37->33 2 trials	33->42 9 trials	42->23 4 trials
3	Yes	No	23->25 7 trials	25->11 4 trials	11->33 7 trials
4	Yes	Yes	33->27 9 trials	27->13 10 trials	13->37 9 trials

- s3: living on campus, a PhD student in Management Science and Engineering, 20s, female, from Korea

	Animation	Sound	1st (2~5 items)	2nd (7~9 items)	3rd (18~22 items)
1	No	No	25->30 3 trials	30 -> 37 5 trials	37->17 7 trials
2	No	Yes	17->21 5 trials	21->30 5 trials	30->50 5 trials
3	Yes	No	50->46 2 trials	46->38 4 trials	38->18 2 trials
4	Yes	Yes	18->22 2 trials	22->31 5 trials	31->3 5 trials

- s4: living off campus, a PhD student in Electrical Engineering, 30s, male, from Korea

	Animation	Sound	1st (2~5 items)	2nd (7~9 items)	3rd (18~22 items)
1	No	No	25->28 3 trials	28->37 4 trials	37->15 4 trials
2	No	Yes	15->18 1 trial	18->26 3 trials	26->48 3 trials
3	Yes	No	48->45 1 trial	45->36 6 trials	36->16 2 trials
4	Yes	Yes	16->18 2 trials	18->27 2 trials	27->49 4 trials

- s5: living on campus, a husband, 40s, male, from the US

	Animation	Sound	1st	2nd	3rd
1	No	No	21->35 4 right 1 left	35->13 4 left 1 right	35->13 4 left 1 right
2	No	Yes	49->35 2 left 3 right	35->20 5 left 4 right	20->40 3 left 7 right
3	Yes	No	40->8 6 left 1 forward	40->8 6 left 1 forward	33->45 3 forward 0 back
4	Yes	Yes	45->28 2 left 3 right	28->30 2 right	30->20 2 right

- s6: living on campus, a wife, 40s, female, from the US

	Animation	Sound	1st	2nd	3rd
1	No	No	10 ->24 0 left 9 right	24->20 2 left 2 right	20->35 1 left 10 right
2	No	Yes	35->49 0 left 13 right	49->41 3 left 1 right	41->42 0 left 1 right
3	Yes	No	42->30 2 left 6 right	30->20 1 left 0 right	20->40 0 left 5 right
4	Yes	Yes	40->35 2 left 1 right	35->47 1 left 7 right	47->27 2 left 0 right

- s7: 11 year old daughter, female, from the US

	Animation	Sound	1st	2nd	3rd
1	No	No	20->31 5 right 1 left	31->15 2 right 3 left	15->38 4 right 1 left
2	No	Yes	38->35 1 left 0 right	35->22 2 left 3 right	22->26 0 left 4 right
3	Yes	No	26->36 4 right 0 left	36->49 4 right 0 left	49->0 0 right 7 left
4	Yes	Yes	0->13 1 left 5 right	13->25 1 left 3 right	25->10 2 left 5 right

- s8: living on campus, age 21, female, undergraduate student

	Animation	Sound	1st	2nd	3rd
1	No	No	27->33 0 left 4 right	33->31 1 left 6 right	31->41 1 left 5 right
2	No	Yes	41->30 2 left 3 right	30->21 3 left 0 right	21->40 1 left 9 right
3	Yes	No	40->42 1 left 3 right	42->20 3 left 2 right	20->33 2 left 4 right
4	Yes	Yes	33->30 1 left 5 right	30->20 1 left 0 right	20->49 0 left 14 right

- s9: living on campus, age 20, male, undergraduate student

	Animation	Sound	1st	2nd	3rd
1	No	No	49->30 4 left 0 right	30->34 0 left 4 right	34->23 2 left 2 right
2	No	Yes	23->15 2 left 3 right	15->38 2 left 9 right	38->35 1 left 0 right
3	Yes	No	35->10 4 left 1 right	10->21 0 left 5 right	21->30 1 left 3 right
4	Yes	Yes	30->49 0 left 8 right	49->45 2 left 2 right	45->1 6 left 0 right

[Note] Unfortunately, for subjects **s5 - s9**, we were missing detailed source videos, and the test were also conducted without variating the distance to the target. So our analysis relied heavily on data we collected from subjects **s1 - s4**.

There were a few limitation to analyze the data quantitatively.

- We only had 9 different users, and this is not a sufficient amount of data to conclude something very meaningful.
- Since our user samples include various ages and various backgrounds, everyone acts quite differently.
- As user can see the system's reaction immediately, without realizing it, user keeps learning how to use the system and shows better performance as time goes on. So to get meaningful data, it is inappropriate to repeat the same test with the same user several times. (For this reason, we should have presented the 4 different environments of our 2x2 design in a randomized order, instead of always proceeding in same order.)

Instead, we made some interesting qualitative observations, the most notable of which we note below:

- Critical incident: In the middle of user test, a preschooler jumped in front of Kinect several times and it stopped the gesture recognition of the system. However, the user presence icon at the upper-right corner is automatically turned off (to gray) when the user is not in the right place and it was easy enough for users to guess that the system was not registering their gestures when the icon changed color from green to gray. (See item 6 in the Discussion section below for implications.)
- Usage observations: Not a single person we user tested had to be told twice about how to use the swipe gesture after they were instructed on how to do it. Some of them found how to use the system themselves without instructions.
- Without showing optimal gestures (swiping horizontally), users made swiping gestures differently. Most of them seemed to do a pretty straight swipe (efficient gestures since we only use the movement in the x direction) but some users made an arc-like swipe [s1, s2, and s6].
- Users also had different habits on the motion after ending swiping gestures. Some users raised their both hands first (above the spine) and use one hand to make a gesture and put it back to the gesture starting position (above the spine). Others only raised appropriate hand to make a gesture and then lowered the hand (below the spine) after finishing gestures. However, as long as their intended gesture was detected, they maintained the way of gesturing.
- User quotes: " I felt like he should run" $\rightarrow$ comment on real-time continuous audio feedback
- Notes form talk-aloud protocol: 11 year old was great at it despite having no previous knowledge of Kinect. While watching her father use the system she even suggested that he should swing his arm faster in order to move ahead or backward more (which is how we implemented it) without any prompting from us.
- Besides that most people we interviewed seemed to do a pretty straight swipe with an "arc up" on the way back to the starting point only exception is the older women [s6] who held her hand out in front of her and tried to "nudge" it in a really choppy way. She was acting kind of like a robot and not fluid.

3.2 Value from velocity metric

[Note] Unfortunately, we were missing data for subjects **s5 - s9**. So our analysis only relied on data we collected from subjects **s1 - s4**.

As we already defined in the previous section 2, we recorded value from our own velocity metric:

$$\text{Value} = 400 \times \frac{\text{hand displacement in } x\text{direction}}{\text{length from shoulder to hip}} \times \frac{1}{\text{number of frames used to move hand}}$$

We requested users to take different speed motions (fast/normal/slow), each time we randomly chose the speed and the hand to be used. We repeated to get ten values for each speed (conducted in the random mixed order). Mean was calculated using eight values excluding the maximum value and the minimum value.

- s1: living on campus, a PhD student in Computer Music, 30s, male, from Canada

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Mean
Fast	152	108	138	135	101	148	130	105	140	128	129
Normal	62	81	65	92	85	95	84	98	74	78	81.75
Slow	59	29	17	23	36	30	34	25	21	29	28.4

- s2: living on campus, a housewife, 30s, female, from Korea

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Mean
Fast	102	95	113	127	136	121	140	131	98	101	116.1
Normal	86	96	78	80	96	117	91	84	92	86	88.9
Slow	56	50	60	43	29	40	44	35	39	42	43.6

- s3: living on campus, a PhD student in Management Science and Engineering, 20s, female, from Korea

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Mean
Fast	68	94	69	136	121	106	125	94	77	94	96.4
Normal	52	18	72	64	68	83	84	76	95	81	72.5
Slow	39	10	3	36	23	52	33	45	49	37	34

- s4: living off campus, a PhD student in Electrical Engineering, 30s, male, from Korea

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Mean
Fast	70	60	138	73	164	126	118	135	170	122	119
Normal	58	109	83	89	57	71	77	92	80	83	79.1
Slow	62	54	52	47	56	68	34	39	28	42	48.3

This test shows that recorded metric for different speeds quite differ from user to user. In this test, we were able to use data from only different users.

Based on the values we got, we will calculate two boundary values α and β which will divide the possible values into three ranges. Therefore, (0, α], (α, β], and (β, ∞) will be corresponding to slow, normal, and fast speed, respectively.

For user i, let $\bar{F}_i$, $\bar{N}_i$, and $\bar{S}_i$ correspond to the mean values from fast motion, normal motion, and slow motion, respectively. Then, α and β were calculated in the following way:

$$\alpha = \frac{\text{Min}\left(\bar{N}_1, \bar{N}_2, \bar{N}_3, \bar{N}_4\right) + \text{Max}\left(\bar{S}_1, \bar{S}_2, \bar{S}_3, \bar{S}_4\right)}{2} = \frac{72.5 + 48.3}{2} = 60.4$$

$$\beta = \frac{\text{Min}\left(\bar{F}_1, \bar{F}_2, \bar{F}_3, \bar{F}_4\right) + \text{Max}\left(\bar{N}_1, \bar{N}_2, \bar{N}_3, \bar{N}_4\right)}{2} = \frac{96.4 + 88.9}{2} = 92.7$$

Therefore, based on our four samples user test, it is better to divide the metric values using the range (0, 60], (60, 93], and (93, ∞). This is just one suggestion and we can only say that the ranges, defined by these values, might perform better than the ranges we used for user test (0,80],(80,120], and (120, ∞] because $\alpha = 80$ and $\beta = 120$ were handpicked values without any user test.

However, the number of users (4) is too small to get the statistically meaningful result. Also, our users already experienced previous relevant test, which can train users to perform in a certain way.

It will require lots of user testing data to find optimal α and β. If time permits and if we can do more tests, it will be really nice to calculate optimal values with more samples. We will be able to even find a more appropriate metric function. However, since we only have limited time and there will be much more important things to do, we will use our α and β, found in this user testing, and let users to be trained with these values. We already found that people got used to and did better to control the speed after few trials with immediate feedbacks.

4 DISCUSSION

Presence of animation and auditory feedback did not make as significant impact on user performance as we had originally thought; however, users were especially conscious of the auditory feedback, and generally found the event-based feedback helpful. It was somewhat surprising at first to see users' disliking the continuous feedback, but their reactions made sense considering the perceivable latency that we could not avoid in implementing this feature.

We must be careful, however, to not infer too much from our limited user testing setup. One of the greatest threats to external validity is the unintended game-like nature of our experimental task, of navigating to a specified number on the menu. Granted, this task was intentionally chosen and designed to keep things simple in order to minimize confounding factors. (That is, had we used "real", visually-complex photographs, the amount of time and effort taken to find a desired photo would likely be affected by many more factors than the core variables we want to test for: gesture control, animation, and auditory feedback). However, because the task became so simplified for the purpose of our user testing, users seemed to be ultra-focused on getting to the right number quickly, as opposed to browsing and scanning nearby items in the menu (which is what would actually happen in a real intended application setting). With this in mind, we have put together the major implications of our user study in the following section.

5 IMPLICATIONS

In this section, we summarize the implication of our study results on our next design iteration, especially with respect to improving gesture recognition, auditory feedback, and visual feedback.

1. Processing requirement

   We learned through the user testing that running the Kinect application with ChucK (for audio) was too much computation for older, less powerful laptops, resulting in too much latency for auditory and visual feedback. Since performance optimization is not our priority, we will compromise by using the best machines we have for our project demo and by removing computationally costly (but perhaps less useful) features like having a continuous auditory feedback.

   On a related note, we decided to change the smoothing parameter in our Kinect application to send un-smoothed data (offering us lowest latency possible) for our various computations, and then we would smooth (low-pass filter) the sensor value ourselves for the portions that need smoothing.

2. Gesture: intuitiveness over precision

   Based on our informal testing from Prototype I, we had re-designed our gesture for scrolling through the menu, from **"elbow-pivot"** gesture to a **swipe** gesture. The reason for this change was that we found out from Prototype I that the elbow-pivot gesture was not intuitive and required us to spend a long time teaching the gesture to users.

   Thankfully, our new swipe gesture turned out to be quite intuitive: even users who had never interacted with a Kinect was able to figure our gesture with a small hint from us "to swipe". What was even more powerful was that many users figured out by themselves that the effort put into swiping (based on the velocity of the fore-arm movement during the gesture) impacted the amount of scroll. Not a single person we user tested had to be told twice about how to use the swipe gesture after they were instructed on how to do it.

   A possible trade-off for using the swipe gesture instead of "elbow-pivot" is that it is now more difficult for the users to communicate their intended magnitude of swipe to the system: "elbow-pivot" uses the absolute angle deviation of the form-arm from the vertical (and this would be the same and unambiguous for all users), but our new swipe gesture calculates magnitude of swipe based on average velocity of the swipe movement over several frames, which translates to different amount of "effort" for different users. However, we still feel that the naturalness or intuitiveness offered by the swipe gesture is worth sacrificing some precision in magnitude control.

3. Animation: a keeper

   Surprisingly, many users reported that they did not notice the difference between having and not having variable 1 (animated transition and physics), even though the visual difference is significant in the videos we recorded of the testing sessions [see s1_1.mov for without animation and s1_3.mov for with animation, linked from Milestone 6 in our project homepage: https://ccrma.stanford.edu/∼jieun5/cs247/P4/P4.html]. As shown in the videos, when the animation variable is present, items in the cover flow menu translate in a continuous manner, rather than jumping to the destination without showing transitional items.

   We hypothesize that this "failure" to recognize animation may have to do with the nature of the experimental setup and the tasks we asked our subjects to perform. As mentioned above, this prompt made the task feel like a game testing speed of navigation, rather than accurately simulating browsing or searching photos. Consequently, we feel that the user paid more attention to the content (number) of the currently-centered item, rather than scanning the cover flow menu below – which is where the core animation is happening.

   At the same time, subject s1 and s2 described how the "speed of scroll" helped them realize that their speed of swipe gesture affected the amount of scroll; this implies that these subjects were using animation as a visual cue, at least unconsciously or subconsciously. In any case, because including animation is not costly and did not seem to have a negative effect on navigation performance, we decided to keep this variable in our next design iteration.

4. Auditory feedback: Continuous feedback is no-good, but keep and improve event-based feedback

   Unlike animation, subjects were always aware of the presence versus absence of variable 2: auditory feedback. To recap, auditory feedback comprised of (1) continuous feedback on the absolute position of users' hands and (2) event-based feedback upon recognition of the swipe gesture. Subjects' attitude towards aspects of the auditory feedback that they liked or disliked were somewhat controversial. But for the most part, people disliked the continuous feedback because of its unpleasant timbre (that made one subject feel "like he should run", reminding him of a Hollywood movie effect) and noticeable latency (especially when testing on a slower laptop), but found the event-based feedback helpful conveying that their swipe gesture has been recognized.

   We decided to keep the event-based sound feedback, but improve on its effects for the next iteration by having the sound directly convey the number of menu items transposed (e.g. by the number of "dings", mimicking the effects of a spinning Wheel of Fortune). We decided to remove the continuous sound feedback because of its unpleasant timbre and the unavoidable latency that defeated its original purpose (of offering a continuous feedback to users that the absolute position of their hands are being detected and monitored by the Kinect).

5. Visual skeletal feedback: Potential to-add

   In place of the continuous sound feedback, we are considering including a simplified visual skeletal feedback in the front-end interface of our final application – with the purpose of conveying to users how the system is perceiving users' motions.

   Because our application architecture is based on a web-application powered by a web server that communicates with the C# Kinect application, we do not have immediate access to 3D skeletal position rendering on the front-end. By transmitting key skeletal positions (such as the upper-body including arms) continuously using low-latency methods such as OSC, it should be feasible to render a simplified skeletal positions as a visual feedback on our front-end application. Time permitting, we hope to include this in our final product.

6. "Detecting" icon: Potential to-add

   Finally, our user testing module had a small icon on top-right corner of screen that stayed green if skeletal detection was functioning properly, and turned gray otherwise. This proved to be quite useful as when the user moved out of the sweet spot or when a preschooler jumped in front of the Kinect several times during user testing, the user could tell from the icon's gray color that his gesture was

no longer being detected. We hope to add this functionality in our web prototype for the next design iteration.