IR MOTION TRACKING AS A STANDARD INPUT DEVICE

Eureka
Volume 2, Number 1 (2011)

Michael Cimolini
Department of Computing Science, University of Alberta

Abstract

As our computing needs change and the availability of advanced input systems begin to increase we find that we are coming to the point where current interfaces are beginning to become outdated. Touch interfaces are becoming abundant, but there is only so much that one can do with a 2D input. This project tries to show that there is a potential for IR based interface systems to replace standard mouse input in current as well as future interfaces. Our results show that both the accuracy and robustness of the IR system designed herein are equivalent to that of standard mouse input.

Introduction

As the pricing of IR based tracking systems decrease, we find that they are now beginning to become affordable for consumer use. As the programs we use become more complex and our requirements from our systems increase we are starting to see a shift away from conventional mouse and keyboard input towards new input systems such as touch systems and motion tracking. In this experiment we look at the accuracy, robustness, and ease of use of an IR based tracking system for use as an input device. We compare the use of the IR system versus the mouse in two tests, the first with a program designed specifically for these tests, and the second with just the use of the standard Windows environment. The goals of these experiments are to show that an IR system could potentially replace mouse input in future interfaces and to show that the IR systems show the same reliability and accuracy as a standard mouse.

Related Work

There has been some work in this field, but as far as I have been able to determine there has been no work specifically using an IR tracking system to track rigid bodies, or work to determine how close in accuracy the IR system and mouse are. Work by Vlasic et. al. [3] has shown the use of a motion capture system for everyday use that shows a high degree of accuracy and reliability. This system uses ultrasonic sensors in combination with accelerometers to track motion. The novelty of this approach is that the ultrasonic sensors track the position of the accelerometers, giving them a reference frame and helping correct for drift. Jacanovic and MacKenzie [1] created a system using optical tracking to control mouse cursor movement. This system is similar in design to the IR desktop application in this experiment, however, it uses a printed marker which is tracked by a camera to control the system based on head position. Takase and Sawada [2] developed a system similar to the 3D brick breaker application of this experiment; however they used optical tracking to track a finger and hand as the control device for their system, instead of the IR based tracking of this experiment.

System Design

IR System

The IR tracking system used for these experiments was the natural point OptiTrack system using their Arena software system. This system uses six infrared cameras to track IR reflective markers. For this experiment all six cameras were calibrated and used, with a calibration area of approximately 2m x 3m x 2.5m. This calibration was completed using the inbuilt calibration tools in Arena. For the tracked object, three reflective markers were used. These markers were arranged on the supplied rigid body marker in a triangle pattern. This marker array was then attached to the back of a wrist brace so as to be worn on the right hand with the markers on the back of the hand. The Arena system software was used for motion tracking, with the streaming settings active. Only rigid body data was streamed which, for this experiment, meant that only data from the markers which were used to track the users hand were transmitted over the network.

Network Application

As Arena streams data through a network as a server, a client application was needed to capture and use this data. To this extent natural point created the NatNet SDK. This is a software development kit designed specifically to build client and server programs for use with the OptiTrack. The SDK is written for use in a C or C++ developing environment. For this experiment the SDK was used to create a client which would capture and interpret the data from Arena. This client was programmed in C++. The client program consists of two parts. The first interacts with the server, receiving data in a secure and reliable way through the use of inbuilt function in the NatNet SDK. The second part interprets this data from the Arena co-ordinates into screen co-ordinates. This data is then passed along to the user interactive program through either a socket connection as in the case of the 3D brick breaker game, or directly as in the case of the IR desktop application.

3D Brick Breaker

This was the first of the two applications designed for the IR motion capture system. This application is very similar to the application developed by Takase and Sawada [2]. The premise and design for the game itself has been around for many years and is an adaptation of the original pong style games. The object of the game is to destroy all of the blocks while keeping the ball in the playing area. The game itself is programmed in Vizard, which is a development environment and extension for the Python programming

language. As such this application was connected to the network application through the use of the Windows socket API so that data could be passed from the network application. The game uses the inbuilt physics and collision detection systems inherent in the Vizard libraries for physics simulation in game. The brick model, paddle model, and world model used in the game were modeled in 3DS Max and exported using a plug-in available on the WorldViz website, WorldViz being the developer of Vizard. These models were exported in the .DAE format for easy importing into the Vizard environment. The ball model used is a model that comes packaged with the Vizard program as part of the Duck Court example. The game itself consists of 72 randomly arranged blocks, with the blocks location being static between runs. The blocks were laid out in a way such that there was no overlap between blocks. Two nearly identical instances of the game were created, with the only difference between the two being the control method. One system was designed to be controlled with the IR tracking system and the other with the mouse. Mouse movement was mapped natively to the paddle, such that the movement of the mouse directly correlated to the movement of the paddle. For the IR controlled system, the mapping to paddle movement was much the same as with the IR desktop application; that being with reaching up mapping to upward movement of the paddle, down to downward movement, left to leftward movement and right to rightward movement. The sensitivity of the system was calibrated for the IR such that a motion of 0.75m was necessary to move from either the left to the right of the screen, or from the top to the bottom. Score was calculated as the number of blocks destroyed minus the number of balls missed. A ball was considered missed when it had traveled passed the paddle in the negative z-axis, with the z-axis being perpendicular to the plane of the monitor.

IR Desktop

This was the second of two applications designed to take advantage of the IR motion capture system. This application is designed to control the mouse in a standard windows environment. This application was written in C++, again using the NatNet SDK. It uses the Windows API to emulate mouse movement and mouse clicks. The application receives data from Arena through the network application. Movement of the mouse is controlled by movement of the tracked rigid body in the x and y co-ordinates of the screen, with the x-axis being measure perpendicular to the standing user, and the y-axis being measured parallel to the standing user, such that reaching up would move the mouse cursor up, left moving the cursor left, down moving the cursor down, and right moving the cursor right. The z-axis was measured perpendicular to the x and y-axis, with reaching forward, away from the user, being the positive z-axis, and reaching behind being the negative z-axis. The z-axis was used to recreate mouse movements, with a movement of approximately 5cm in the positive z-axis in a period of 0.5seconds performing a left mouse click. For this experiment only left mouse clicks were used, however it would be possible to simulate right mouse clicks in a similar fashion. The sensitivity of the system was calibrated such that it would require a movement of 0.5m to move the cursor from the top of the screen to the bottom of the screen, and a movement of 1m would move the cursor from the left of the screen to the right of the screen.

Experiment

For the experiment, participants were randomly assigned to complete one of the two experiments first and then, after a 5 minute rest, were asked to complete the other experiment. For all participants, except for in one case, the same camera calibration results and rigid body marker calibration were used. For the one exception, the tracking volume had to be lowered as the height of the participant caused the markers to be lost while reaching towards the bottom of the screen. In total, twelve participants completed the experiment, with ten completing the IR desktop portion and the full twelve completing the 3D brick breaker portion. Throughout the experiment users were monitored for soreness or discomfort in their shoulders or arms, as these are noted as being related to gorilla arm syndrome, a common problem with interfaces requiring the arm to be raised for extended periods of time. Users were also asked to complete a survey at the end of the experiment rating their ability to continue to use the system for extended periods of time. This was used in a qualitative measure, not a quantitative measure, as a general assessment to determine the need for future experiments focusing on this issue.

3D Brick Breaker

For this experiment two metrics were measured, these being time and score. Time was measured from when the first ball was launched to when the last brick was destroyed. It was used as a gross estimate more so then a solid metric as the time taken is variable based on the luck of bouncing of the ball. Also the system was designed to change the launch angle of the ball based on the position of the mouse cursor, which may have had some effect on time as well. Score was used to calculate the number of errors that being the number of times the ball passed off screen. This was calculated by subtracting the score from seventy two, the total number of blocks in the game. Participants were assigned to complete the game using either the IR tracking system or the mouse input first and then, after a three minute rest, were asked to complete the game a second time using the other input device. Users were assigned to these two groups in a pseudo random fashion, with six users being placed in each group. Each participant was given the same instruction at that start of each game, that being the objective of the game and the controls. Each participant was given two minutes to become familiar with the mouse controls, or four minutes with the IR controls. During the four minutes with the IR controls, the users were asked to touch each of the four corners of the play area to determine that the IR tracking system was properly calibrated and that they would indeed be able to reach the entire playing area. In cases where this was not possible, the game was restarted to recalibrate the tracking and position of the paddle. Due to a small glitch in the physics engine of the game, from time to time the ball

would pass through the walls of the environment. When this occurred, a new ball was launched by clicking the mouse. This did not deduct points from the users score. During the participants playing of the IR controlled game, the mouse would be moved at random to adjust the position that the next ball would be fired at to emulate the way in which the ball would be fired in the mouse controlled game.

IR Desktop

For this experiment three metrics were measured, with these being time, errors, and number of clicks. Time and clicks were the major comparison for this experiment, with these being directly related to the speed and ease of use of the system. Time was measured from when the first instruction was read, until the last instruction had been completed. Number of clicks was recorded as the number of times the left mouse button was clicked in the case of the mouse, or as the number of forward presses made by the participant in the case of the IR system. Errors were measured as the number of times that the IR system lost tracking during the runs; this was used as a measure of robustness for the system. As with the first experiment, users were again assigned to two groups, with each group starting with one of the two control methods and then, after a ten minute rest, being asked to complete the experiment with the other input method. The reason for the extended rest in this case was due to the users having to complete the same set of tasks, so a ten minute time interval was used to try to cause the participants to forget the tasks they had completed in the first run. During this time period users were given a tour of the lab in hopes of aiding in causing them to forget the tasks they had completed. There were seven tasks that the users were asked to complete, all being standard Windows tasks. These tasks were read to the participants in order, with each subsequent task after the first only being read after the user had completed the previous task. The tasks were as follows: open an internet browser, open any bookmarked webpage, close the internet browser, open Notepad, close Notepad, open the Trashcan, and close the Trashcan. The point of these tasks was to cause the participants to interact with the largest screen area possible. To this extent the icon for the Trashcan was placed in the bottom right corner of the screen, and the icon for the internet browser was placed in the top right. As the user needed to reach the top left corner to close the internet browser, and the bottom left to open notepad, all corners of the screen were explored. The Notepad and Trashcan windows were set to open in the middle of the screen, allowing for exploring of the middle of the screen as well. For each trial, users were asked to begin with the cursor centered in the middle of the screen, such that the starting position would not be a confounding effect on the experiment, as well as allowing for a more thorough navigation of the centre of the screen. For both runs of the experiment, the screen resolution was lowered to 1440x900, with the experiment being run on a 22” 16:10 aspect screen.

Results

3D Brick Breaker

The statistics between the mouse and IR inputs were compared for both time and errors for this experiment, with the recorded results listed in Table I. of the appendix.

The mean time for the IR input was 04:21.7 minutes, and 03:51.7 minutes for the mouse input. For these results, we get a t-value of 1.306262 versus a t-critical value of 2.200985, with 11 degrees of freedom. This gives a p-value of 0.218117, meaning that there is no statistical difference between the times for the two input methods. A similar comparison of the errors leads us to a mean of 10.16667 for the IR and 9.916667 for the mouse. For this comparison we have a t-value of 0.166317 versus a t-critical value of 2.200985with 11 degrees of freedom. This gives a p-value of 0.870925, meaning that there is no statistical difference between the errors. For both of these measurements we find that we are unable to reject the null hypothesis that there is no difference between controlling this program with the IR system or with the mouse.

IR Desktop

The statistics between the mouse and IR inputs were compared for time, errors, and clicks for this experiment, with the recorded results listed in Table II. of the appendix.

The mean time for the IR input was 01:23.5 minutes and 00:21.1 minutes for the mouse input. For these results, we get a t-value of 6.722904 versus a t-critical value of 2.262157with 9 degrees of freedom. This leads to a p-value of 8.63^-5, meaning that there is a statistical difference between these means. For the clicks we get a mean of 32.8 for the IR input and 14.2 for the mouse input. This leads to a t-value of 9.1194 versus a t-critical value of 2.262157 with 9 degrees of freedom. This leads to a p-value of 7.66^-6, meaning that there is a statistical difference in the mean value of clicks for the two input methods. The third statistic, errors, had a mean value of 0.6 for the IR input and 0 for the mouse input. This leads to a t-value of 2.25 versus a t-critical value of 2.262157 with 9 degrees of freedom. This leads to a p-value of 0.051003, meaning that there is no statistical difference in the number of errors for the two input methods, but it is close.

Conclusion

A number of conclusions can be drawn from this experiment. The first is that when it comes to direct control of the cursor, an IR system can be just as reliable and accurate as a mouse. This is shown most notably in the 3D brick breaker experiment, where neither time needed to complete the game, nor number of errors showed any statistical difference for the two systems. This means that our null hypothesis that the two systems are the same holds true. As this experiment only showed differences in the ability to control the cursor we see that the accuracy of the two systems is highly comparable. The fact that there was also shown to be no statistical difference in the number of errors in the IR system versus the mouse in the IR desktop experiment shows that an IR system can be just as reliable as the mouse, even when used in a standard Windows environment for which the mouse was designed

and optimized. From the IR desktop experiment we do, however, see a statistical difference in both time and number of clicks required. As we have noted previously that accuracy was no different for the mouse and IR system, we can attribute the difference in time to the clicking performed in the IR desktop experiment. This is mainly due to the design of the system. During the forward pressing motion of the hand to perform clicks in with the IR system, it was often noted to be the case that this would cause the mouse pointer to move, causing the participant to miss their intended target, leading to both an increase in clicks and, as a result, time taken to complete the experiment. Implications that can be drawn from these results are thus; an IR system can be as accurate and reliable as a mouse, though some conventions will need to be taken to augment the IR system with a more robust and accurate method of performing clicks than was presented in this research.

Future Work

3D Brick Breaker

Any future work on the 3D brick breaker game would likely focus on depth perception. Through the trials it was noted that participants had difficulty tracking the exact location of the ball due to a lack of depth cues. Possible improvements to this system would be the inclusion of shadows for the ball, bricks, and paddle on all four walls of the environment allowing the users eyes to track these to determine relative depths. The other would be to modify the program to work with a 3D display which would then add depth perception to the program by the nature of the display.

IR Desktop

Future work for this system would focus on finding a better method to control clicking the left and right button. Methods that have been noted thus far would be using tracking of a second hand and having gestures from that hand control clicking, while having the main hand control movement. This would also allow for a form of multi touch, as two points could then be tracked. A second idea would be to add some sort of button system to the tracked hand or as a separate device to allow for clicking with the buttons of that device instead of through the use of gestures.

General Program

There is quite a bit that can be done as future work with the system in general. The first would be to add more tracking markers to the rigid body to increase the accuracy of tracking and to decrease errors. The second would be to look into possibilities to reduce the number of cameras used in tracking to decrease the size of the system, as well as decreasing the price. The third would be looking into the design of other games for the system as there is definitely some possibility for use of a system such as this for physical rehabilitation.

References

[1] Javanovic, R., & MacKenzie, I. S., MarkerMouse: Mouse Cursor control using a head-mounted marker. *Proceedings of the 12th International Conference on Computers Helping People With Special Needs, ICCHP, 2010,* pp.49-56. Berlin: Springer.

[2] Takase, H., & Sawada, H., Gestural interface and the intuitive interaction with virtual objects. *ICCAS-SICE, 2009*, pp.3260-3263.

[3] Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., & Popović, J. (2007). Practical motion capture in everyday surroundings. In *International Conference on Computer Graphics and Interactive Techniques* (3rd ed., Vol. 26, Article 35). San Diego, California: ACM SIGGRAPH.

Appendix

TABLE I.

3D Brick Breaker Results

Brick Breaker									
IR					Mouse				
Time	*Score*	*Bricks Left*	*Errors*	*First*	*Time*	*Score*	*Bricks Left*	*Errors*	*First*
04:21.8	59	0	13	Y	02:39.6	67	0	5	N
06:32.2	63	0	9	Y	03:06.3	56	0	16	N
04:10.8	60	0	12	N	04:27.6	60	0	12	Y
05:30.9	54	0	18	N	04:23.4	61	0	11	Y
04:55.1	61	0	11	N	03:33.3	66	0	6	Y
04:22.2	62	0	10	Y	05:31.7	54	0	18	N
04:02.7	64	0	8	Y	03:06.8	64	0	8	N
04:03.4	67	0	5	Y	03:04.8	64	0	8	N
03:52.7	65	0	7	N	03:55.2	62	0	10	Y
03:56.7	62	0	10	N	04:45.0	63	0	9	Y
02:41.3	68	0	4	N	03:25.3	66	0	6	Y
03:51.1	57	0	15	Y	04:20.8	62	0	10	N

TABLE II.

IR Desktop Results

Windows							
IR				Mouse			
Time	*Errors*	*Clicks*	*First*	*Time*	*Errors*	*Clicks*	*First*
01:18.7	0	37	Y	00:18.8	0	19	N
02:46.3	2	39	N	00:32.8	0	14	Y
01:06.0	2	29	N	00:22.3	0	12	Y
01:35.3	1	32	Y	00:15.6	0	13	N
00:54.8	0	22	N	00:19.8	0	15	Y
01:06.6	0	36	N	00:20.8	0	14	Y
01:19.5	0	31	Y	00:20.3	0	14	N
01:11.5	1	36	Y	00:17.7	0	13	N
00:52.6	0	24	N	00:16.7	0	14	Y
01:43.8	0	42	Y	00:25.9	0	14	N