Performance Measurements for the Microsoft Kinect Skeleton

Mark A. Livingston* Jay Sebastian† Zhuming Ai‡ Jonathan W. Decker§

Naval Research Laboratory

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies;

1 Introduction

The Microsoft Kinect for Xbox 360 ("Kinect") provides a convenient and inexpensive depth sensor and, with the Microsoft software development kit, a skeleton tracker (Figure 2). These have great potential to be useful as virtual environment (VE) control interfaces for avatars or for viewpoint control. In order to determine its suitability for our applications, we devised and conducted tests to measure standard performance specifications for tracking systems. We evaluated the noise, accuracy, resolution, and latency of the skeleton tracking software. We also measured the range in which the person being tracked must be in order to achieve these values.

2 Measurements

We conducted our tests on a machine configured with Windows 7 Ultimate (Service Pack 1) equipped with two Intel Core2 6600 2.4 GHz processors and 3.0 GB of usable RAM (4.0 GB total) in a 32-bit architecture. All tests used the 20-point skeleton data stream (Figure 2, center) and were conducted in the expected environment for our training applications: a laboratory with standard fluorescent lighting fixtures. We expect our users to be looking at the monitor which shows a desktop VE; thus we mounted the Kinect sensor above the monitor, and our subjects were generally facing the monitor at all times during the data acquisition. Subjects did not wear any special clothing and varied in skin color from very light to dark. We did not observe any differences in the Kinect's performance with respect to clothing or skin color, but this was not a focus of our test. We tested with one, two, and three users present, although only two skeletons may be tracked.

2.1 Range

We need to know how close and how far a user can be from the imaging sensor in order to be a tracked skeleton. The angular range of the device is $57^\circ \times 43^\circ$ (horizontal $\times$ vertical). This can be extended vertically by using the software controls available for a tilt motor; this has a range of 54°. Microsoft recommends an "optimal range" of 1.2–3.5m from the sensor. Our tests showed that a skeleton could be acquired 0.85–4m from the camera. An acquired skeleton would be lost moving outside these bounds. Data beyond 4m appeared impossible to retrieve. An inspection of the values returned by the depth stream revealed the same result; any depth pixel that should have held a value beyond 4m instead held zero. Since other libraries for the Kinect have been known to sense well beyond that distance, albeit not very precisely [2], it seems that the Kinect for Windows SDK limits depth tracking to 0.85–4m. However, the skeleton data at both extremes of this range tended to be erratic and unreliable. For this reason, Microsoft's optimal range of 1.2–3.5m was used for further testing.

*e-mail: mark.livingston@nrl.navy.mil

†e-mail: sebastianjay@yahoo.com

‡e-mail: zhuming.ai@nrl.navy.mil

§e-mail: jonathan.decker@nrl.navy.mil

2.2 Noise

Tracker noise causes a rendered avatar to jitter on the screen. To quantify the noise required a user to stand still while being tracked; we used support structures hidden from the depth imaging system (behind the user) to hold the user still. We took 1000 samples of the central position for a tracked skeleton standing 2.0m from the sensor and computed mean position and standard deviation (sd). We found 3D noise at 1.2m to be 1.3mm with a sd=0.75mm; at 3.5m, we found 6.9mm, sd=5.6mm. Figure 1 shows an exponential performance curve within the optimal depth range. Noise differed by dimension: x averaged 4.1mm, y 6.2mm, and z (depth) 8.1mm. The right wrist (31.0mm) and right hand (22.6mm) exhibited more noise than other joints. (The next highest was the right foot: 8.2 mm.) One would expect joints near the edge of the image to be higher in noise; they may partly disappear from the frame and still be considered tracked. While the wrist and hand are at the end of the skeleton, they were well within the imager's field of view. We did not see an effect of the number of people in view.

2.3 Accuracy

We chose to measure relative accuracy rather than absolute accuracy, for which we would require a calibration of the arbitrary coordinate system of the Kinect to a world reference system. Our applications require this type of relative tracking for gesture recognition. Our reference was a visibly straight wooden meter stick positioned 2m from the sensor, running approximately along the x axis. We affixed a marker to a user's wrist to give a consistent position relative to the physical skeleton and placed this marker along the meter stick. We took 25 samples per point to reduce the effects of noise and measured distances between 100mm and 500mm. The average error in these tests was 5.6mm, with a standard deviation of 8.1mm. To test scaling of accuracy with additional users in view, we used a long metal bar, 250mm segments, and a tape measure for reference. Error grew from 1.4mm with one user to 1.8mm with two users to 2.4mm with three users. These values were repeated through multiple tests; no differences were found with respect to dimension, including in depth.

2.4 Latency

We devised a relative measurement of latency using the USB mouse. We were able to poll our mouse at 125Hz, implying a minimum measure of 8ms; other authors found previous Windows systems to be as much as 20ms [3, 4]. We found the position of a hand touching the mouse and then programmed our application to record the time when the hand reached that position after having not been at that position. The program also recorded the time when the mouse button was pressed. The difference of these two times gives us relative latency to the mouse, to which we add the 20ms estimate as a worst-case scenario. This measurement succeeds in giving us an accurate estimate on the basis of several conditions. The application is only waiting for these two events (button press

and hand in the desired position). No other user applications were running on the computer. A closed-loop system such as a pendulum or other visually repeating mechanism would require a system that could sense when the camera image shows the hand at the correct position; this in turn would imply that the latency of the camera and the display of its image is also being measured. A test was conducted with two skeletons using a simple pendulum (a hand moving shoulder-to-shoulder) and found similar results as we describe [1].

When the program was running at its normal frame rate of 30 Hz, the relative latency was found to be 106 ms on average, with a standard deviation of 23 ms and a maximum of 156 ms. When the program was running more slowly, the relative latency was found to average 202 ms, with a standard deviation of 26 ms and a maximum of 270 ms. Program speed depended largely on the number of pixels tracked (with or without a skeleton) as part of a human form and showed a quadratic relationship. We re-ran this test bringing the right hand down to the left, in which the mouse was held and thus the button hit. With a single skeleton to track, the program generally maintained a 30 Hz update rate, but it would on occasion drop. With two or three users, the frame rate was generally between 18-20 Hz. The results show slightly higher measurements than our initial test, likely owing to the variability we experienced in the frame rate. With one skeleton, we saw mean latency of 146ms (maximum 243ms); with two skeletons, mean of 234ms (maximum 386ms); with three users (two skeletons), mean of 205ms (maximum 490ms).

2.5 Resolution

To establish resolution, we looked for error values less than the accuracy measurement with standard deviation less than the noise measurement; we did this separately for depth (z) and one lateral dimension (x). Measurements were taken 2m from the sensor using the protocol of the relative accuracy test, with distances of 1-5mm. We found lateral resolution of 3mm (0.086°), in agreement with PrimeSense, makers of the depth sensor. Our measured depth resolution, 2mm, was much better than the specification of 10mm. The skeleton computations use segmented regions of many pixels and appear to synthesize depth resolution through averaging or interpolation across the skeleton. The skeleton tracking may "track" (actively recognize) or "infer" (interpolate by humanoid spatio-temporal constraints) the joints. Although all our measurements were with the relevant joint position reported as tracked, it appears that we benefited from some form of interpolation.

3 Discussion

We asked whether the Kinect was suitable for the type of gesture recognition appropriate for our applications. Our initial evaluation of the basic performance characteristics of the device and its accompanying software toolkit give us reason for optimism. VEs wishing to use the Kinect to acquire gestural commands can use these measurements to guide the placement of the sensor relative to the user's position. We did not acquire data appropriate for all applications and displays; applications like ours that use a single projection screen or desktop VE can use these measurements to assist in the design of gestural controls that will be accurately discerned.

The range of the device appears to be sufficiently large to provide room for a user to move and perform gestures. The noise within the optimal depth range appears tolerable to permit gesture recognition, although clearly one would prefer that the user stay as close to the sensor without exiting the optimal range. The accuracy of the skeleton tracking appears sufficient to support arm and hand gestures. Although the error is an order of magnitude greater than many high-end commercial tracking systems, we see it as an inexpensive and reasonable solution for our applications. However, the accuracy will not enable gestures that require recognition of finger positions or movements.

The latency is perhaps the most problematic of the performance characteristics. The best conditions produced a latency of 106 ms relative to the USB mouse on our system, or approximately 125 ms of end-to-end latency. This is notably larger than many commercial tracking systems frequently used in VEs. With multiple users or a single user close to the sensor, the number of pixels being processed for tracking of human forms increased, and the latency increased correspondingly. We experienced maximum latencies of nearly 500 ms, which is large enough to be disturbing to a user. Also, we note the conflict between desiring to be close to the sensor to reduce noise, but far from the sensor to reduce latency.

As with any structured light system to recover depth, the Kinect's performance can degrade in difficult lighting conditions. Bright fluorescent lighting can increase the noise, but we have seen it track in dark rooms. We noticed numerous spurious "recognitions" of skeletons on inanimate objects, but generally speaking, the software properly recognizes humans in the environment and tracks the skeleton. We plan to investigate how multiple Kinect devices could be used in tandem, as well as how the motor control system could help expand the range.

Acknowledgements

The authors wish to thank Joel Alejandre. This work was supported in part by the Office of Naval Research Expeditionary Maneuver Warfare and Combating Terrorism Department and by the ONR/ASEE Science and Engineering Apprenticeship Program.

References

[1] B. D. Adelstein and S. R. Ellis. Personal communication, July 2011.

[2] N. Crock. Mathnathan: Kinect depth vs. actual distance. `http://mathnathan.com/2011/02/03/depthvsdistance/`, Feb. 2011. Last accessed 09 August 2011.

[3] R. R. Plant, N. Hammond, and T. Whitehouse. How choice of mouse may affect response timing in psychological studies. *Behavior Research Methods*, 35(2):276–284, 2003.

[4] L. Ramadoss and J. Y. Hung. A study on universal serial bus latency in a real-time control system. In *IEEE Industrial Electronics*, pages 67–72, Nov. 2008.

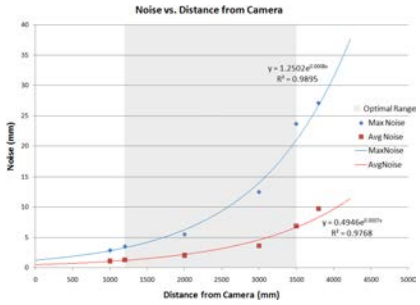

Figure 1: The mean and maximum noise as a function of distance from the sensor showed an exponential fit. The shaded region denotes the optimal depth range.

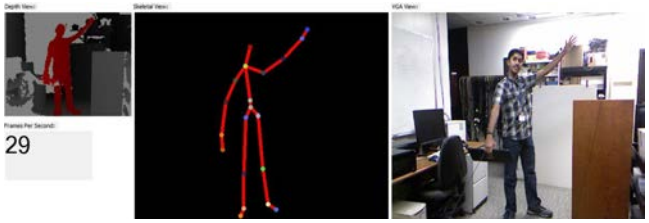

Figure 2: Three data streams provided by the Kinect for Windows SDK, from left: depth with user index, skeleton, and video.