Analysis of IR-based Virtual Reality Tracking Using Multiple Kinects

Srivishnu Satyavolu*
Dept. of Computer Science
Univ. of Minnesota Duluth

Gerd Bruder†
Dept. of Computer Science
Univ. of Würzburg

Pete Willemsen‡
Dept. of Computer Science
Univ. of Minnesota Duluth

Frank Steinicke§
Computer Science
Univ. of Würzburg

Figure 1: Multiple Kinects are arranged in a circular manner. Circled Kinect is the primary tracker. Kinects circled with dotted lines are providing IR interference.

Abstract

This article presents an analysis of using multiple Microsoft Kinect Sensors to track users in a VR system. This article focuses on using multiple Kinect sensors to track infrared points for use in virtual reality applications. Multiple Kinect sensors may serve as a low-cost and affordable means to track position information across a large lab space in applications where precise location tracking is not necessary. We present our findings and analysis of the tracking range of a Kinect sensor in situations in which multiple Kinects are present. Overall, the Kinect sensor works well for this application and in lieu of more expensive options, the Kinect sensors may be a viable option for very low-cost tracking in virtual reality applications.

1 Introduction

This article analyzes the capability of Microsoft's Kinect sensor to act as a tracking system in virtual reality applications. The introduction of the Kinect sensor instigated much interest across the VR community, primarily due to the rich array of sensors for capturing 3D scene information, at a very low cost of approximately U.S.$150. Professional tracking systems tend to be quite expensive, when compared with the price of the Kinect. We present our efforts using multiple Kinect sensors to provide an inexpensive, infrared position tracking system for virtual reality. We analyze the Kinect's ability to robustly track an infrared marker across the space of a 7m x 10m lab, focusing on the accuracy of position data acquired over distance and in the presence of multiple Kinect sensors that potentially can cause severe infrared interference.

*e-mail: saty0010@d.umn.edu
†e-mail:gerd.bruder@uni-wuerzburg.de
‡e-mail:willemsn@d.umn.edu
§e-mail:frank.steinicke@uni-wuerzburg.de

2 Related Work

The Kinect has seen wide-spread adoption well outside the traditional game console market. Since its launch in late 2010, the Kinect has been used in numerous projects integrating with either the Microsoft-provided Kinect SDK or OpenNI, through the open-source Linux driver. A Kinect sensor is a motion-sensing input device that comes with the Microsoft Xbox 360 console. The sensor contains a RGB camera, a structured infrared (IR) light projector, an infrared camera, and microphones. Some background information on the device is available from both Microsoft and other sources [6]. Kinects need to be accurately calibrated to achieve robust depth estimates [4, 2]. Microsoft's KinectFusion system demonstrated robust acquisition of depth data for 3D reconstruction [3].

3 Tracking Positions with Kinects

The overall objective of this work is to understand the efficacy of using multiple Microsoft Kinects as tracking devices for monitoring a user's position and a user's skeletal system in a VR application. While a single Kinect is capable of tracking an IR light within the lab, multiple Kinects afford different views of users within the tracked space that are not possible using a single Kinect. Combining multiple views can strengthen skeletal tracking mechanisms provided by APIs such as Microsoft's Kinect SDK or the OpenNI framework. However, these APIs' effective ranges are limited to about 2 to to 3 meters, which could limit usable mobility in VR applications, where more typical tracking ranges from 10 to 12 meters in size. Understanding the tracking range of the Kinect is an important factor to consider for using Kinects for position tracking.

The operational range of skeletal tracking APIs is limited, likely because the depth resolution of the Kinect decreases with increase in distance from the Kinect. Hence, it becomes difficult to estimate skeletal joint positions as users get farther away. The tracking system described in this paper exploits the fact that, even though it's difficult to estimate the entire set of skeletal positions after a certain range, it's still possible to get reliable depth values enough to get an estimate of the user's position across a large lab space. To get reliable position tracking, we use a single IR marker attached to the user's head that can be tracked over a large space.

While this article focuses on IR tracking, we specifically test the sensitivity of the tracking to IR interference from multiple Kinects. The reason why interference from multiple Kinects may be an issue involves using multiple Kinects to improve skeletal tracking. With multiple Kinects, IR projectors will potentially project structured IR light into the scene and possibly into other Kinect IR sensors. In theory, this *interference* could cause problems with the depth values from the Kinect. Some research has been done to reduce interferences by using a Kinect-shuttering approach [5, 1]. Our experiments specifically focus on testing the Kinect-based IR tracking system against interference from multiple Kinects.

3.1 Segmenting Depth

Marker world positions are obtained by using the Kinect IR depth image to get the pixel position, (x_{im}, y_{im}) of the IR marker and thus, obtain the depth of that pixel, d_{im} from the corresponding depth image. Raw depth values typically range from [0, 2047], with 2047 representing invalid depths. The image data is converted into the

Figure 2: Background depth image (left), depth image before background subtraction (center), depth image after background subtraction (right).

world location of the marker using the following equations:

$$z_M = 0.1236 * tan(d_{im}/2842.5 + 1.1863)$$

$$x_M = (x_{image} - cx_d) * z_{world}/fx_d$$

$$y_M = (y_{image} - cy_d) * z_{world}/fy_d$$

where (x_M, y_M, z_M) is the position of the IR marker in world coordinates, and fx_d, fy_d, cx_d, and cy_d are the intrinsic parameters of the depth camera. These equations were obtained from Stephane Magnetat's and Nicholas Burrus' posts to on-line forums discussing the Kinect. A series of OpenCV image morphological operations are performed on the extracted foreground depth and IR images to remove any external noise. These images are used to extract the position and depth information of the IR marker. Unfortunately, the IR marker creates it's own small circle of IR interference that makes it difficult to precisely identify it in the IR image. Moreover, interference from the other Kinects does cause problems for detecting the IR marker. In these situations, our software potentially can get confused between the actual IR marker and the bright IR interference from the additional Kinects. To overcome these issues, we apply neighborhood locality and static background separation techniques to retrieve the actual IR marker position.

Background depth information of the entire scene is captured by merging depth images of the same scene obtained over time. The acquisition of the background depth image is done once upon starting the tracker system. The background image is used to extract foreground images in real time by a simple background separation technique using OpenCV. Each IR image from the Kinect is analyzed in real time to find the brightest pixel in the foreground (in this case, our IR marker). Figure 3.1 illustrates the segmented depth information during this process. Pixel depth is calculated using a locality of neighborhood principle to estimate the IR marker depth value by examining the surrounding pixels outside the small circle of interference of the IR marker. Using a 15 by 15 pixel neighborhood window to calculate the a mean of all the valid depths worked well for this implementation.

3.2 Experimental Setup

Our experimental setup consists of 5 Kinects mounted on chairs of same height, with one being the primary Kinect (Figure 1) while the remaining introduce interference. They are all arranged to provide tracking centered around a 7m by 7m space. We used a WorldViz IR marker mounted on a cart as the tracked point. The depth sensor of the primary Kinect is calibrated using the intrinsic parameters of the Kinect calculated using RGBDemoV0.4, which is Nicholas Burrus' implementation of OpenCV's standard chessboard recognition techniques. The tracked space consists of 30 manually placed points, that are roughly a meter apart from each other. At each point, 1000 samples of the IR marker positions are collected which are then averaged to get the final estimate of the tracked position. This is done both with and without interference from the other Kinects. The entire step is repeated for all the 30 points. The jitter in these IR marker positions is then analyzed over all the 1000 samples with and without interference.

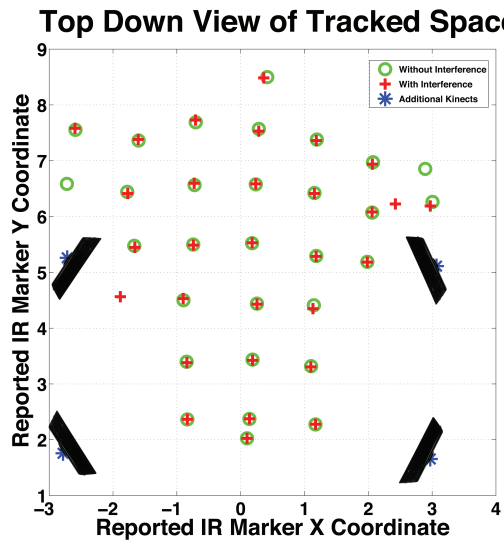

Figure 3: Comparison of the Kinect tracking data for points acquired with Kinect-based interference and without interference.

4 Results and Conclusion

Figure 4 compares the tracked positions with and without interference. The green circles represent the estimates of the points measured without interference, while the red crosses represent the estimates of the points measured with interference from other Kinects. Standard deviations of reported pixel positions were nearly zero in all cases except for some points with increased interference. The mean and standard deviations of a single point with maximized interference was a mean depth value of 1016.0 (SD=89.6723). Without interference at this location, the mean depth value was 1039.6 (SD=1.0484). Additionally, when compared with our WorldViz PPTH system the Kinect was roughly off by 3cm on average.

We observed that interference is not a major concern for IR tracking with the Kinect. Thus, we believe that based on our initial experiments, the Kinect can serve as a low cost tracking system. With multiple Kinects, it should be possible to implement a more functional tracking system that allows multiple users to interact using gestures across VR lab spaces.

References

[1] K. Berger, K. Ruhl, C. Brümmer, Y. Schröder, A. Scholz, and M. Magnor. Markerless motion capture using multiple color-depth sensors. In *Proceedings of Vision, Modeling and Visualization (VMV)*, pages 317–324, 2011.

[2] C. D. Herrera, J. Kannala, and J. Heikkilä. Accurate and practical calibration of a depth and color camera pair. In *Proceedings of International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 437–445, 2011.

[3] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*, pages 559–568, 2011.

[4] K. Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS Workshop on Laser Scanning 2011*, 2011.

[5] Y. Schröder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, and M. Magnor. Multiple kinect studies. Technical Report 09–15, ICG, TU Braunschweig, 2011.

[6] J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. Technical report, CTU Prague, 2011.