

Using a Kinect Interface to Develop an Interactive 3D Tabletop Display

Llyr ap Cenydd, Chris J. Hughes, Rick Walker and Jonathan C. Roberts

School of Computer Science, Bangor University

Abstract

While display technology has advanced significantly in recent years, interaction techniques are still tied to the mouse and keyboard paradigm. While for many tasks this may still be appropriate, allowing systems to recognise and respond to user gestures and motions has enormous potential for natural methods of interaction with virtual media. Traditional methods for pose recognition involve using cameras to track the position of the user. This can be very challenging to complete accurately in a variety of environments where objects may be occluded and the lighting conditions can change. Further, to accurately determine the depth of objects in a scene requires a much more complicated and carefully calibrated system. In this research we prototyped a 3D tabletop display and explored the Kinect game controller as a possible solution to tracking the pose and gesture of a user interacting with our display.

1. Introduction

There are several commercially available tabletop displays - for example, the Microsoft Surface (www.microsoft.com/surface/) - which provide 2D displays with full multi-touch interaction, allowing the user to select, drag and interact with items on a virtual desktop. These displays have been used with great success as collaborative tools where multiple users can interact with a single environment [IHHCI0]. However, with the advances in 3D technology, it has now become feasible to implement a 3D tabletop that allows the user to place and manipulate objects directly in 3D.

Although there has been significant work toward understanding the boundaries of using 3D technology in a tabletop display [GW07], it remains a challenge to identify the pose and gestures of a user and also to register their position to the virtual objects in the 3D world. In this research we prototyped a simple 3D tabletop display. As the user is working within a 3D projected space we use a Kinect device to segment the position of the users hands, regardless of the lighting conditions. The Kinect also gives us a depth position, facilitating the registration of the user's hands in the virtual environment. As a proof of concept we developed a simple game for our display allowing the user to drop vir-

tual cubes onto the table, where a physics engine models the behaviour and interaction of the cubes.

2. Hardware Setup

Our 3D tabletop display consists of a white, reflective surface. A digital 3D projector is used to project onto the tabletop, using a mirror to extend the throw of the beam. The Kinect is mounted in line with the mirror, and aimed directly downwards to ensure that it has a clear view of the user and the workspace (see Figure 1).

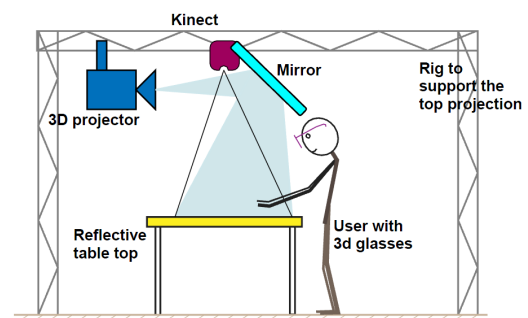


Figure 1: The hardware setup for our 3D tabletop display.

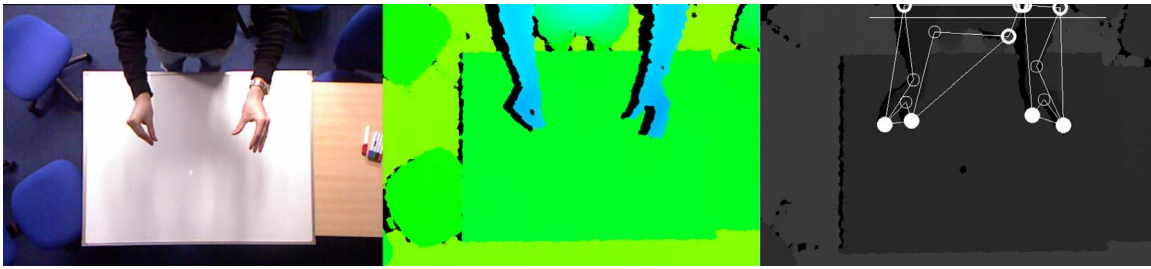


Figure 2: The Kinect provides two views from its camera of the user and work area. Left shows a normal camera view and centre a depth view, mapped here to a colour scale. The arms and hands can be clearly distinguished in the depth image, since they are at a different depth to the background. In the right image, we identify the user finger position by finding key points in the view.

3. User Tracking Using Kinect

Tracking hand position is a common task in computer vision applications. Typically, a camera image is segmented and features identified. Difficulties often arise in distinguishing the hand from the background in the image, and lighting and colour of clothing must be carefully controlled.

Microsoft's Kinect sensor offers a solution to these problems, by combining an RGB camera with a structured-light depth sensor in a commodity device. The device returns both an RGB colour image and a depth map for the scene. Using the depth values makes it easy to separate hand regions from the background as shown in Figure 2. Contouring algorithms can then be used on the depth map to give the location of the hands.

4. Software Implementation

Our software implementation used the CL-NUI platform driver (codelaboratories.com/kb/nui/) to interface with the Kinect. Given the device's reconstructed depth image, OpenCV [BK08] was used to track the user's hands and fingers using a convex hull approximation method. The two or four most prominent points were identified in each frame (see Figure 2), which allowed detection of finger pinching or clapping gestures. The extracted coordinates were projected into a 3D scene created using the Ogre graphics engine [Jun06], and rendered in stereo using Nvidia's 3D vision. At each coordinate a semi-transparent sphere was drawn, in effect floating at the user's fingertips.

Finally a simple gesture tracker was implemented to detect whether the user was separating clasped fingers or hands as shown in Figure 3. These gestures result in PhysX (www.nvidia.com/object/physx_new.html) rigid-body cubes being generated out of thin air at the detected location, with initial velocity influenced by the user's motion.

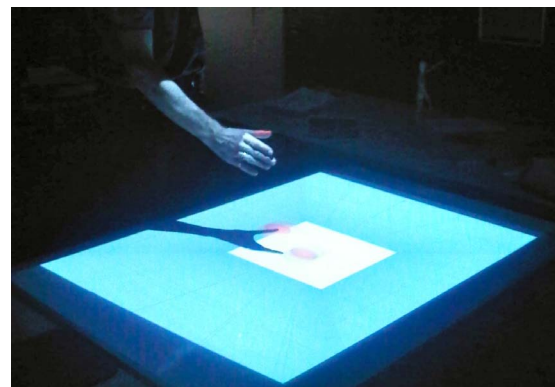


Figure 3: A user dropping cubes onto the 3D tabletop display using a gesture recognised by the Kinect

5. Conclusions

Pairing the Kinect controller with a 3D projector allows construction of an inexpensive and flexible system for interacting with a 3D image. The Kinect simplifies the tasks of identifying pose and gesture considerably, and the supplied depth information allows true 3D interaction with the projected image. In our system, we added some simple interactions with a physics engine, but many more complex gestures can be detected. It is anticipated that many uses can be found for such a system.

References

- [BK08] BRADSKI G., KAEHLER A.: *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.
- [GW07] GROSSMAN T., WIGDOR D.: Going deeper: a taxonomy of 3d on the tabletop. In *Tabletop* (2007), pp. 137–144.
- [IHH10] ISENBERG P., HINRICHS U., HANCOCK M., CARPENDALE S.: Digital Tables for Collaborative Information Exploration. *Tabletops-Horizontal Interactive Displays* (2010), 387–405.
- [Jun06] JUNKER G.: *Pro OGRE 3D programming*. Apress, 2006.