Tesseract— 3D pointing device

Jayasurya Salem Sudakaran

with Prof. Michael Kipp
Interaction Engineering

Winter semester 2019/2020
Augsburg University of Applied Sciences

Resources related to the project
https://drive.google.com/open?id=1HY7u_w85KKmiMUskoj-FL_9ccPuNFgsp

Abstract

Tesseract is a pointing device for three-dimensional virtual spaces. The aim of the project was to implement a novel but practical solution to navigation in three-dimensional Human-Computer Interfaces.

Tesseract presents a 3D physical space within which the user can operate with six degrees of freedom (translation and rotation in X-Y-Z axes). A ball is suspended in a hollow cubic frame with elastics— the user moves and twists it in physical space which is mapped to the virtual 3D space. After every use, the elastics bring back the ball to its central position in the cubic space, which is the true zero.

Page	**Content**

Motivation

3D virtual spaces and objects, in modeling software like Blender and Cinema 4D, are handled and manipulated on 2D displays using 2D pointing devices, like a mouse. The third dimension, depth, is handled through inconsistent implementations— keyboard shortcuts or rotating the virtual space itself. This approach leads to a lot of problems— 3D manipulation is limited to the 2D plane it is viewed in and making even simple objects like spring coils in 3D is not possible by hand. Mathematical functions or merging two different shapes are the only ways to go, and it presents a steep learning curve.

Related work

There have been attempts to solve this problem, although most implementations address only a part of it. Here are some of the projects that served as a beginning point for Tesseract—

1. Martinot, François & Plenacoste, Patricia & Chaillou, Christophe. (2004). The DigiTracker, three degrees of freedom pointing device. 99-104. 10.2312/EGVE/EGVE04/099-104.

Digitracker implements translating and rotating the 3D space by implementing a rig that's controlled by the pointing finger.

This model is intuitive and easy to grasp but doesn't realize the idea into a more wholesome control. This paper explores a 3D pointing device but lacks the other functions of a mouse— it exclusively implements a 3D controller to be used in addition to a mouse and so Tesseract is expected to build upon the ideas DigiTracker to be more comprehensive and intuitive to use.

2. Space Navigator— https://www.3dconnexion.com/spacemouse_compact

Space navigator implements a joystick-like device that can move as well as rotate along the X-Y plane and implements the third axis (Z) by enabling the joystick control up and down.

While the space navigator is the industry standard for 3D pointing devices, the device is not equally unbiased on all three axes (X and Y are easier than Z) and the complicated learning curve for panning vs. tilting is a disadvantage especially for beginners and casual users. Tesseract tries to eliminate this by trying to map physical space to the virtual 3D space.

User group

The idea of the project is to cater less to technology-oriented people who can work around indirect solutions (who also liked the idea of Tesseract) and more to people like visual artists who rely on their sense of action-output.

In general, Tesseract is aimed to be a device that provides a more direct relation to the input action and the output onscreen, bypassing the limitations of a 2D based system.

Scenario

1. **3D Software**

    a. **Rotation & pan—** Basic 3D camera movements.

    b. **3D curves—** Making a curve that has components along all three axes can be made by just moving the ball in 3D space, as compared to switching between multiple planes with a mouse.

    c. **Quick shapes—** Some 3D shapes, say like that of a spring coil could be really complex to make with a mouse, but can be relatively easier with Tesseract as the user only has to make the same shape in physical space while holding the ball.

2. **Video games**

    a. **In-game camera—** In games where navigation and viewing angles both need 3D controls, while movement could be controlled with a joystick/WASD, viewing, thereby aiming, could be done with Tesseract.

Physical prototype

The prototype was put together very quickly, so as to test and refine the interaction models. While the roughness of the prototype negatively affected the precision of the tool, threshold amounts were provided in the software to negate the effects to a large extent. The components are as follows—

1. **Ball as the controller—** Wooden ball with a good-enough size to be held by hand is suspended with elastic strings from all eight corners of the frame.
2. **Hollow cubic frame—** A wooden frame of 25cm x 25cm x 25cm provides the physical space for the controller to move around.
3. **Phone camera—** A smartphone mounted to the back of the frame was used as the tracking device.
4. **Visual marker—** a preprocessed image that is tracked in 3D space by the camera.

Implementation

There are a lot of ways the device can be implemented, exactly which must be used depends on the software used and the kind of work done. Some are as follows—

1. The mouse is mapped to the cursor, and a button/clutch on the ball can be implemented for the selection of the object to be manipulated. This will provide deep control in the 3D space as the user can easily use it as the primary pointing device, even in 2D interfaces.
2. The mouse is secondary, a 2D mouse selects the object and the 3D mouse only takes care of translation and rotation— this might help in separating concerns between 2D and 3D manipulation.

Interaction system

For both translation and rotation, I began with a one to one mapping with a clutch to enable movement. The user presses a button on the ball and moves it around to move the object/cursor exactly as much as the ball moves. This is similar to a 2D mouse where moving 0.1cm on the table moves the cursor an equivalent amount (0.1cm x conversion factor) of distance on the screen. This posed two major problems—

1. To move longer distances or rotate to higher angles, the constraints of the minimal space inside the frame and the elastics would be restrictive. One solution might be to let go of the ball to get back to zero every time but to move 10cm, the user might have to move the ball in the same direction for 2cm five times.
2. To get a curved path, the user has to return to zero after every small movement, to switch to a different angle.

The next idea was to map the movement of the ball to the speed of movement of the cursor in the same direction. To move the cursor 4cm (considering conversion factor of 1) in the X-axis, the user moves the ball 1cm in the X-axis for four seconds. This works similar to a gas pedal, where how much it is pressed determines the speed of the car and how long it's pressed is how long the car keeps moving. This provides a solution to both problems mentioned above. As long as the user wants movement in one of the eight sections of a 3D space (as divided by the three axes), they can move it around in one continuous stroke.

Technology

Two applications were made, both in Unity.

Android
I used a smartphone that ran an Android-Unity app. Vuforia AR library was used to preprocess the marker image and track it in 3D space. Vuforia identifies visual elements on the image and trains the app to identify them.

TCP over IP
The android app sends the tracking data through local wifi over TCP connection and is received by the windows machine.

Windows
The desktop counterpart was also a Unity app. To test and refine the prototype, a custom 3D app was built, as it gave full control over the manipulation of objects. The data is received over the TCP connection and is mapped to the cursor/object.

User testing scenario

The user tests were designed in three stages—

1. Move a cube to place on top of another cube in 3D space.
    a. Evaluation criteria: Time taken
    b. 2D mouse v. Tesseract
2. Move a cube to place on top of another cube in 3D space.
    a. Evaluation criteria: Time taken
    b. 3D space navigator v. Tesseract
3. Make a 3D cut on a cube
    a. Evaluation criteria: Precision, time taken

Preliminary tests for 1 and 2 with a few users have been conducted and helped in understanding key points that helped change the interaction method as mentioned above..

Future scope

1. While the implementation now is restricted to custom 3D software, OS level drivers can be implemented.
2. More rigorous testing should be done to refine the device.
3. In 2D interfaces like file managers, the depth can be used to open and close folders and files can be grabbed and pulled out of folders and put in different ones (cut and paste). Furthermore, the operating interface can itself be modified to support 3D functions.
4. The design of a cursor that moves in 3D space that tells information to the user must be explored. Extending that, interfaces that are spread out in 3D spaces must be explored.

Resources

Videos, screenshots and the code for the applications can be found in the following Google Drive folder—

https://drive.google.com/open?id=1HY7u_w85KKmiMUskoj-FL_9ccPuNFgsp