3D Tracking using multiple Nintendo Wii Remotes

A Simple Consumer Hardware Tracking Approach

David Scherfgen
Bonn-Rhein-Sieg University of Applied Sciences
Department of Computer Science
Grantham-Allee 20
53757 Sankt Augustin, Germany
david.scherfgen@h-brs.de

Rainer Herpers
University of New Brunswick
Faculty of Computer Science
540 Windsor Street
Fredericton, NB, Canada
rainer.herpers@h-brs.de

ABSTRACT

An easy to build and cost-effective 3D tracking solution is presented, using Nintendo Wii Remotes acting as cameras. As the hardware differs from usual tracking cameras, the calibration and tracking process has to be adapted accordingly. The tracking approach described could be used for tracking the user's motions in video games based upon physical activity (sports, fighting or dancing games), allowing the player to interact with the game in a more intuitive way than by just pressing buttons.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies*; I.4.1, I.4.8, I.4.9 [**Image Processing and Computer Vision**]: Digitization and Image Capture, Scene Analysis, Applications—*Camera calibration, Imaging geometry, Tracking*

General Terms

Algorithms, Measurement, Experimentation

Keywords

Tracking, Triangulation, Nintendo Wii Remote, Camera Calibration

1. INTRODUCTION

Since the launch of Nintendo's Wii game console, video games demanding physical activity from the player have become very popular. But the motion information reported by the Wii Remote controller is relative and not very accurate. This contribution explores the potential of using the built-in infrared cameras of Wii Remotes to achieve more accurate, yet still very cost-effective 3D tracking of objects or the user's motions [1]. Full 3D tracking can be used for creating novel gaming experiences [4].

2. SETUP AND CALIBRATION

The Wii Remotes are placed in such a way that each point in space where tracking is required can be seen by at least two cameras. It should be noted that the field of view is relatively narrow (about 45° horizontally). The objects to be tracked are fitted with infrared LEDs (preferably with a wide beam angle) or infrared reflectors. The Wii Remote camera seems to be most sensitive to wavelengths around 940 nm.

For 3D object tracking, the cameras need to be calibrated. Camera calibration yields *intrinsic* and *extrinsic* camera parameters. Intrinsic parameters are characteristical constants for a given camera and describe how points in space are transformed to 2D image coordinates. They include the *focal length*, the *principal point* (image center) and *lens distortion*. Extrinsic parameters specify the camera's *position* and *orientation* in space [2].

Calibration is typically done by pointing the camera at a predefined calibration pattern (e.g. a chess board) and then detecting the pattern in the image. All camera parameters can then be calculated. Unfortunately, the Wii Remote's camera image can't be accessed directly. Camera information is limited to the position and approximate size of up to four infrared dots, which are detected by the Wii Remote itself. We use a simple calibration board with infrared reflectors and illuminate groups of them (Figure 1). In future, this will be done automatically with infrared LEDs.

Figure 1: Calibration Setup. The infrared-reflecting dots are illuminated using an infrared light source.

3. TRACKING

In the camera image, the infrared LEDs or reflectors appear as blobs. The Wii Remote detects up to four blobs in the image (1024×768 pixels) and returns their approximate center point position and size. The results can be queried and are then sent to the tracking PC via Bluetooth [3].

Given a point P, whose 3D position is to be determined, and its projected position P_1' on one camera's image plane, a ray can be constructed that originates from the camera position C_1 and passes through P_1'. P is located at some unknown distance along that ray. Once a second camera located at C_2 sees the point at P_2' on its image plane, a second ray can be constructed. P is then calculated by finding the two rays' intersection point (Figure 2). This is called *triangulation* [2].

In practice however, the rays don't intersect, because of limited camera resolution, errors in the point detection process and numerical inaccuracy. One approach is to find a point with minimal average distance to all rays. The average distance provides an error measure. Alternatively, one can try to find a point that minimizes the deviation that occurs when reprojecting it onto the cameras' image planes and comparing the coordinates to the actually observed coordinates.

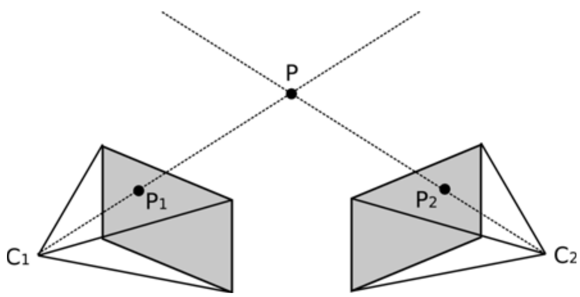

Figure 2: Triangulation. Given the observed 2D point positions P_1 and P_2, the 3D point position P can be approximated.

However, when tracking multiple points, it is not clear which points correspond to which in the camera images. This correspondence problem has to be solved before triangulation can be applied, since triangulation obviously relies on targeting the rays at the same point in space. Otherwise the error gets large. This can be exploited by testing all possible correspondences and then selecting the one that produces the lowest error. The brute force approach is affordable, since the number of points and cameras is low.

In an additional step, the points have to be identified reliably over time. This is done by minimizing the distances between each point's actual position and its predicted position, which is calculated using its approximate velocity.

4. IMPLEMENTATION

The tracking software is divided into a calibration tool, the tracker server and tracker clients.

The calibration tool calibrates the Wii Remotes, that is, it calculates intrinsic and extrinsic camera parameters using the *OpenCV* library. It works for any planar calibration pattern, which will then define the tracking coordinate system.

The tracker itself runs as a server application. Upon startup, it reads the data produced by the calibration tool and then periodically queries all the Wii Remotes. The 3D coordinates for each point are determined using triangulation and filtered for jitter reduction.

Clients (e.g. games) can connect to the tracking server and receive the 3D point positions including approximate errors.

5. PRELIMINARY RESULTS

Using two cameras at a distance of about two meters, the precision achieved lies in the range of millimeters. This result was obtained by estimating the position of various infrared LED at known coordinates.

The Wii Remote's tracking precision is convincing, considering its low price. Nevertheless, the pixel positions reported by the hardware often jitter, therefore a smoothing filter should be applied, which introduces additional latency. A 4-sample moving average filter was found to be a good compromise.

Due to the Wii Remotes' narrow field of view, they have to be placed relatively far away from the tracking area, meaning that much space is required, or they have to be used in greater numbers. Additionally, only four points can be tracked per camera. When more than four points are visible, it is unclear which of them will be detected. However, if a lower tracking frequency is acceptable, more points can be tracked by turning groups of four LEDs on and off periodically. This would require additional hardware for synchronization.

6. CONCLUSION

3D tracking allows for more intuitive gaming experiences and can be achieved using only cost-effective hardware like the Wii Remote and some infrared LEDs. As the blob detection is done in hardware, the tracking process is simple and doesn't need much computational effort.

The 3D tracking system could be used in physical gaming. Reflective infrared markers or LEDs would be attached to the user's head, hands, joints and/or hand-held objects, allowing their movements to be tracked. This would allow the player to directly control his avatar in an intuitive way, for example in fighting, sports (tennis, bowling) and dancing games. Other games could recognize gestures or track the user's head in order to adjust the perspective accordingly, increasing immersion.

7. REFERENCES

[1] S. Attygalle, M. Duff, T. Rikakis, and J. He. Low-cost, at-home assessment system with wii remote based motion capture. *Virtual Rehabilitation, 2008*, pages 168–174, Aug. 2008.

[2] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, 2003.

[3] J. Lee. Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, 7(3):39–45, July-Sept. 2008.

[4] J. Yim, E. Qiu, and N. Graham. Experience in the design and development of a game based on head-tracking input. In *Future Play 2008*, pages 236–239, 2008.