Real Time Six Degree of Freedom Pose Estimation Using Infrared Light Sources and Wiimote IR Camera with 3D TV Demonstration

Ali Boyali[1], Manolya Kavakli[1], and Jason Twamley[2]

[1] VISOR (Virtual and Interactive Simulations of Reality),
Research Group,
Department of Computing,
[2] Physics Department,
Macquarie University
Balaclava Road North Ryde NSW 2109, Australia
`{ali.boyali,manolya.kavakli,jason.twamley}@mq.edu.au`

Abstract. The goal of this paper is to present the development of a tracking technology to interact with a virtual object. This paper presents the general procedures of building a simple, low cost tracking system by using Wiimote (a remote of Nintendo game console) and the Open source Computer Vision (OpenCV) software library as well as interfacing the tracking system with an immersive virtual environment (Vizard). We used an iterative position and orientation estimation (POSIT algorithm) which is optimized as an OpenCV function for extracting position parameters. We filter out the noise in the coordinate values using Kalman filters. The orientation and translation of the tracked system are then used to manipulate a virtual object created in the virtual world of Vizard. Our results indicate that it is possible to implement an inexpensive and efficient application for interacting with virtual worlds using a Wiimote and appropriate digital filters.

Keywords: WiimoteIR tracking, 6 DOF pose estimation, OpenCV, 3D TV visualization.

1 Introduction

Tracking of moving objects has a wide variety of application areas including medical science, interactive entertainment, gaming, control technologies and military applications. Motion tracking can be performed either by tracking a moving object with a fixed capturing device or by capturing movement of a camera using stationary objects. The information related to motion in both cases is extracted and used according to the requirements of the systems.

One useful technology for optical motion tracking are Infra-Red (IR) cameras which can report the 2D projected coordinates of IR light sources. The reported 2D projected coordinates of a known specific 3D arrangement of IR sources are then

P. Sénac, M. Ott, and A. Seneviratne (Eds.): MobiQuitous 2010, LNICST 73, pp. 137–148, 2012.

processed to estimate the position and orientation of an object in three dimensional space (six Degree of Freedom – 6 DOF) using appropriate computer vision algorithms.

Pose estimation and gesture recognition brings distinct advantages to Human Computer Interaction (HCI) in comparison to conventional input devices. These techniques enable users to interact with computers in a more intuitive and natural way. Commercially available tracking systems do not appeal to ordinary computer users due to their lack of affordability and reduced availability [1]. Wiimote is a game input device that allows the user to manipulate objects in a virtual environment, with the capability of sensing gestures through its accelerometer and optical sensors (Fig. 1). It can measure yaw, pitch and roll orientations via built-in acceleration sensors, and report the coordinates of infra-red light sources with a resolution of 1024x768 pixels.

Fig. 1. Nintendo Wiimote controls [2]

Due to its unique features, affordability and availability, the Wiimote handset has attracted great attention in recent years. It has been used not only in computer games but also in HCI applications. For example, a head mounted IR LED arrangement can replace the ordinary mouse in order to enable people with limited mobility to surf the internet with minimal effort [3]. The click function can be realized with a pre-defined voice vocabulary in this system. In the study [4], Wiimote acceleration sensor readings are used to control a wheelchair with simple hand gestures. Low latency of the IR camera due to high refresh rates [5, 6] allows programmers to use Wiimote to control a virtual object [7], to navigate interactive maps [8], and build low cost interactive electronic whiteboards [9].

Our goal in this study is the development of an inexpensive 3DTV interactive environment for the user to interact with a virtual world, using head and ultimately, hand tracking.

We use a similar methodology for tracking IR light sources as presented in [5, 6], to demonstrate the potential use of such 3D tracking methods in medical imaging. The commonly used approach towards tracking objects using an IR camera and IR

LED (Light Emitting Diode) beacon is to extract orientation and translational movements from 2D image data. The structure of our tracked object is described in section 2 below. We present a method of pose estimation and its computer implementation in section 3. The software environment and coding are presented in section 4. We present the results of filtering in section 5. In section 6 we discuss our findings and directions for future studies.

2 IR Beacon Architecture

The Wiimote has a monochrome "camera" with a resolution of 128x96 pixels. On-camera proprietary Nintendo hardware post-processes the camera image and outputs the x-y coordinates of up to four infrared light sources at a refresh rate of 100 Hz. The coordinates of light sources are reported with a resolution of 1024x768 pixels via interpolation. There are scant details regarding the actual hardware makeup of the "camera" in the Wiimote (i.e. it is not clear whether it is an array device like a CCD chip in ordinary cameras), but it has a peak sensitivity in the IR though can also detect in the visible. In the Wiimote visible light is blocked with a plastic IR bandpass filter, which sharply attenuates light transmission below 900 nm. The combination of this IR filter and the spectral sensitivity of the "camera" results in the Wiimote being most sensitive to light around a wavelength of 940 nm [5, 6, and 10]. We conducted an experiment measuring the spectral properties of the IR bandpass filter as shown in Fig. 2.

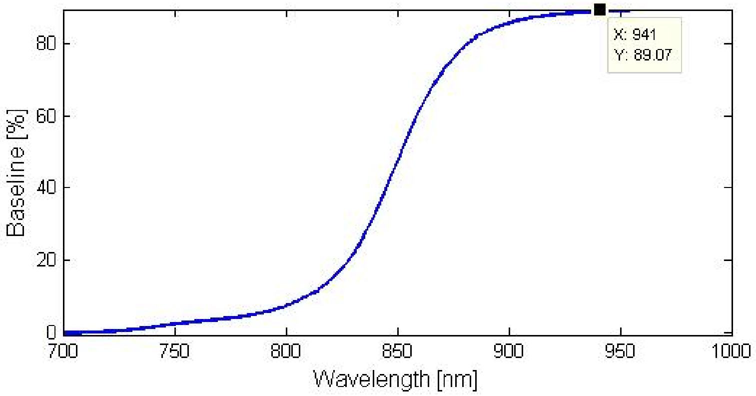
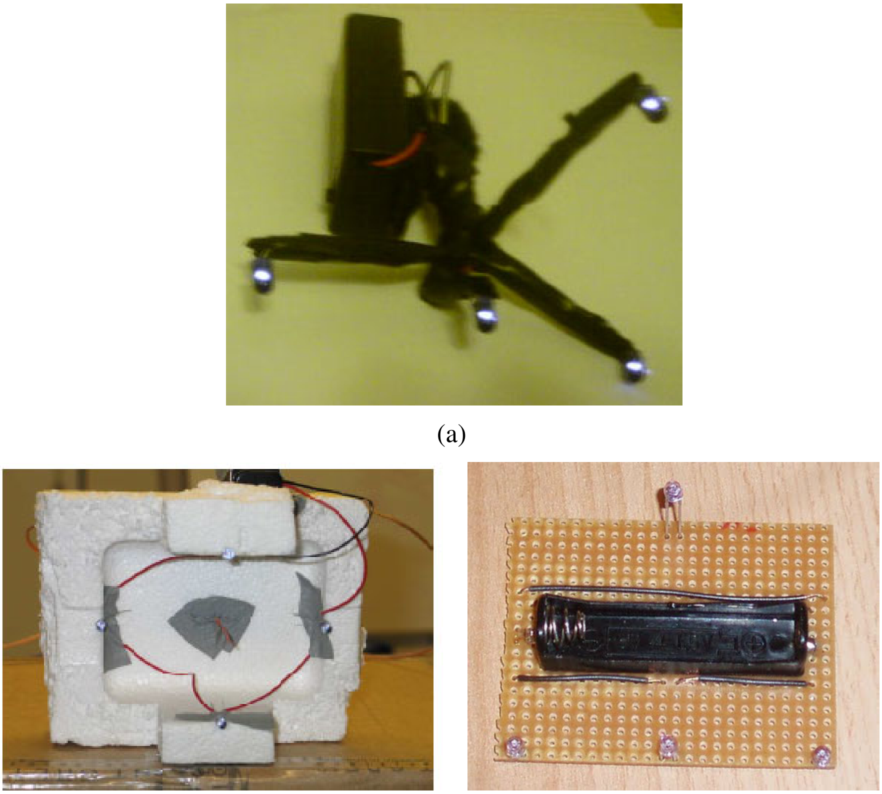


Fig. 2. Percentage transmission of light through the Wiimote's IR band-pass filter as a function of wavelength. We see a sharp attenuation below light of wavelength below 900 nm.

We used four LEDs which emit light at 940 nm wavelength in our LED beacon since the OpenCV pose estimation algorithm requires four non-planar points. A similar LED structure was used in the applications developed in [5] and [6] (Fig. 3, a, b and c).



(a)

(b)

(c)

Fig. 3. (a) LED arrangement used in this study, IR Beacons used in [1] and [5] (b and c)

We used four resistances in the circuit to exceed the current limit (100 mA) for LEDs (Fig. 4). Resistance values are calculated using the rated forward current of these LEDs for a specified voltage bias and we found 3V batteries and 4 Ohms resistances where sufficient for our application.

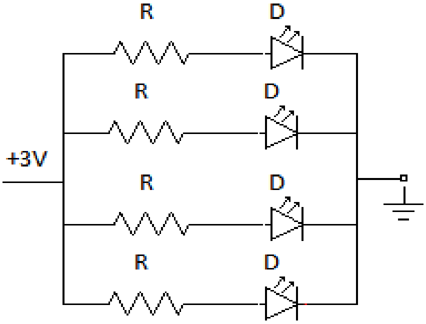


Fig. 4. Circuit diagram of our IR LED beacon where the LEDs (D) and associated resistances (R), are arranged in parallel and connected to a +3V voltage source

3 Methodology and Implementation

The Wiimote communicates bi-directionally via an on-board Broadcom 2042 Bluetooth driver chip [11]. This allows great flexibility in wireless communication between the device and the computer. There are freely available Wiimote libraries for different programming languages such as C#, Python, Matlab and C++. As OpenCV is a C++ based library we chose C++ as a development environment, and we used the Wiiuse library for the communication with Wiimote via Bluetooth.

There are a number of methods to estimate the orientations and positions of tracked objects, tagged by a constellation of tracking points [11, 12 and 13]. Among these methods the POSIT algorithm has some advantage since an initial pose estimation is not required [12]. However it suffers from a lack of accuracy when the tracking points lay on the same plane.

We used the POSIT function available in OpenCV to estimate the rotation matrix and translation vector of a tracked IR beacon. The OpenCV pose estimation method is also used in [5]. Alternatively one can solve a system of non-linear equations using the Levenberg-Marquardt minimization method to find the orientation matrix and translation vectors [6].

In the OpenCV C++ library, the cvPOSIT function takes the physical measurements of the configuration of lights on the object to be tracked (which are assumed to be pre-measured and fixed on the beacon to be tracked), the number of iterations, (or iteration sensitivity) as a termination criteria, and the camera-observed image points in 2D (as well as camera focal length in pixels). The function returns a rotation matrix and a translation vector. In nearly four or five iterations the process finds an estimate for the object's pose which is highly accurate [14, 15]. The use of extra non-coplanar points helps the algorithm to converge.

The camera focal length is an intrinsic parameter used to compute the z-coordinate of the 3D coordinate system and we model the optics using a pin-hole camera model. From this model the focal length can be found by using the camera calibration functions of OpenCV. A special IR beacon arrangement (a camera calibration pattern or a grid), whose geometry is known precisely (such as four LEDs positioned at eachcorner of a square), is required to use the camera calibration routines [14, 16]. However, camera calibration is beyond of the scope of this study and we used camera parameters as reported in [5, 6].

An important part of all pose estimation algorithms is the correct association between individual light points imaged by the camera and their respective sources on the object to be tracked. In [6] and [21], the Wiimote's linear accelerometer was used to obtain an initial point match and new poses were estimated successively with subsequent pose matchingbeing performed on a nearest neighborhood basis. In the study [5], three of the LEDs on the beacon were nearly collinear and thus the image points as seen by the camera lay also on a line in all poses. Thus the source and image point association can be achieved by applying co-linearity tests. We use a different geometrical approach to establish the association between the image and source points. Four different triangles can be constructed from four points (Fig 5). The triangle which has the biggest area excludes the point D, and this is true when viewed in any pose.

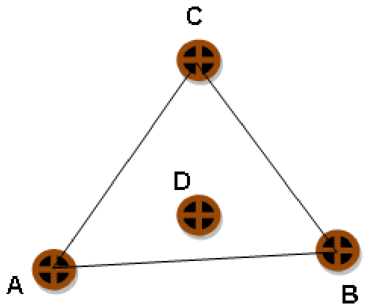


Fig. 5. The view of our IR LED arrangement from the z-axis direction

We can label point *C* since the vector *DC* constitutes the biggest angle in the combination of the vectors *AD*, *AB* and *DC*. Thus the point *C* can be distinguished from the others. Once the points, *C* and *D* are labeled, *A* and *B* can be labeled using a clockwise rotation check relative to point *C*.

The physical arrangements of IR LEDs on the beacon we use are given in Table 1. From our experiments the distances between the points are sufficient to track the device up to two meters away from the camera. The coordinates of point *A* are taken as the origin of the system. The other point's coordinates are then defined with respect to point *A*.

Table 1. Coordinates of the LEDs making up our IR beacon

Points/Coordinates [mm]	x	y	z
A	0	0	0
B	105	10	0
C	52	73	-10
D	54	13	7

3.1 Filtering Noisy Signal

Every digital measurement contains noise to some extent. Before processing the signals noise should be filtered out to increase the accuracy and to smooth the variation of the signal. Accuracy of six degree of freedom (6DOF) IR tracking with a single Wiimote IR camera was evaluated in [1, 6], by comparing the tracking performance with a more accurate 6DOF commercial tracking system.

In the study [1] the Wiimote tracking developed there had errors in estimating the coordinate and orientation parameters of the pose: x-y-z coordinates ~ 7-8 mm, yaw-pitch angles ~ 0.8 degree . Nearly same levels of accuracy are reported in [6]. The noise values as well as the offset in the *y* direction of the IR beacon are apparent in Fig. 6.

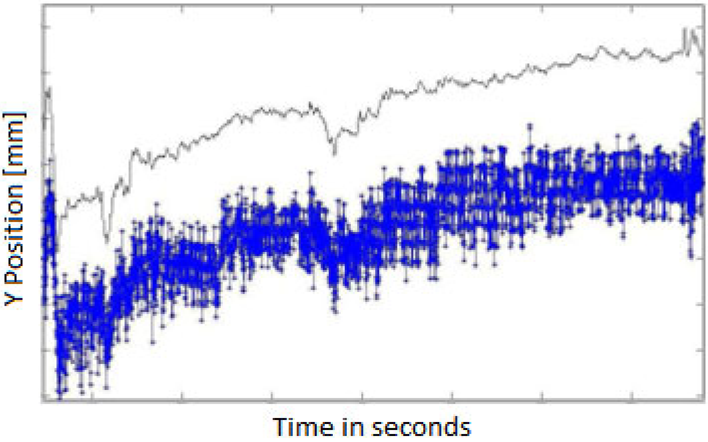


Fig. 6. Gauging the accuracy of the pose estimation. Graph plots the *y* coordinate in mm versus time at a stationary position taken from the study[6].

From these studies one observers that the Wiimote IR camera is not suitable for applications where precision is required, but it is sufficient for manipulating virtual objects or for use as a pointing device [1].

We can filter out the noise from the signal by applying a Kalman filter to the coordinates sent to the computer from the Wiimote. Kalman filtering is a powerful method to estimate the state of dynamical systems subject to random noise. It has been widely used in tracking systems to filter out noise from the signal [17]. Once the filter is initialized with initial state values and error covariance, the Kalman filter estimates the next state variables and error covariance in time by maximizing the posterior probability recursively and new measurements are used to update the estimation [14, 17].

Kalman filtering functions are available in the OpenCV library. We used these functions instead of writing the filtering class from scratch. Models of the system's dynamics and measurement are required for Kalman filtering.

Given a dynamic system at time step k, we can express the state and measurement models (1), (2) as follows

$$x_k = Fx_{k-1} + Bu_k + w_k \tag{1}$$

$$z_k = H_k x_k + v_k \tag{2}$$

Here, x_k and z_k represents the states and measured variables, u_k represents the control input, w_k and v_k represents the process and measurement noise respectively [14]. *F*, *B* and *H* are the matrices representing the state transition, control input and measurement matrices respectively.

State variables in our system for an individual LED position are

$$x_k = \begin{bmatrix} x \\ y \\ dx/dt \\ dy/dt \end{bmatrix}, \qquad F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

Here *dt* is the sampling time (refresh rate). Since we only measure the position, our measured variables will be x and y.

$$\mathrm{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{4}$$

The random variables w_k and v_k are independent from each other and they have a normal probability distribution [14, 18].

$$P(w) \sim N(0, Q) \tag{5}$$

$$P(v) \sim N(0, R) \tag{6}$$

$$P_k = E\left[e_k e_k^T\right] \tag{7}$$

Q, R and P represent process noise covariance, measurement noise covariance and error covariance respectively. An initial state and initial error covariance matrix should be assigned to start the filtering process. We assigned the coordinates of the LEDs as an initial state as if they are estimated in a virtual preceding step by using the OpenCV Kalman filter functions' state_post method. We assumed that our tracking system is linear and our model reflects the dynamics of the system accurately.

The underlying aim of this assumption is to smooth the visualization of the object since high accuracy is not needed for our intended use (3D pointer). Thus we kept the measurement and process noise covariance values at very low levels.

4 Visualization and Virtual Environment / 3D TV

In the visualization stage, we used the Vizard Virtual Reality engine developed by Worldviz. By using the Vizard SDK we created a plugin that receives our tracking data and makes it available within the Vizard environment. We could then easily attach the 6 DOF location of our physical sensor to either a graphical 3D model or to the virtual camera position.

In order to display our test platform in stereoscopic 3D, we used a recently released 3D television (Samsung UA55C7000). Vizard is able to output to the display in various 3D standards, one of which is to use side by side stereo images, a left eye scene and a right eye scene displayed side by side horizontally, or one above the other vertically. The 3D TV can then be manually set to the relevant side by side 3D stereo mode.

We were then able to see the results of our tracking device, which being a 6 DOF device moves in 3 dimensions and 3 orientations, on a screen capable of displaying the result in full (Fig. 7).

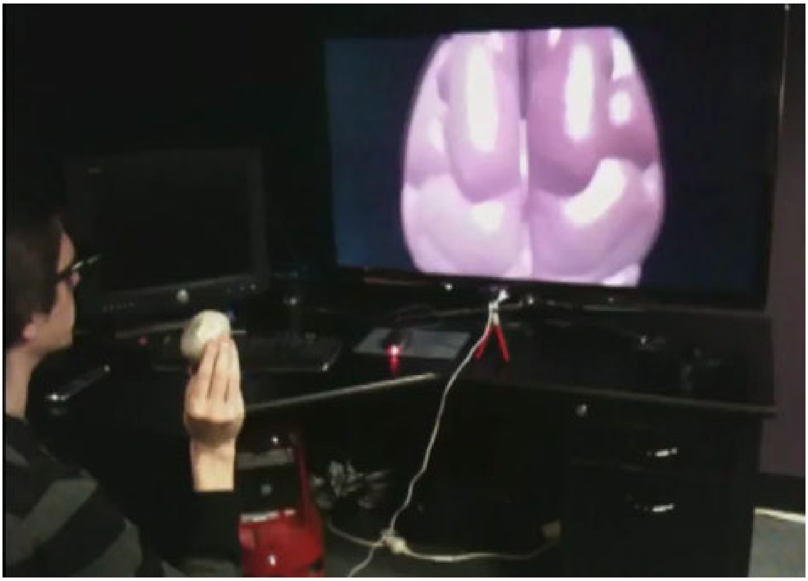


Fig. 7. Brain Model as a virtual object

5 Results

We used a Kalman filter to feed pose estimation model with a low level of noise, and not to allow the error level to increase after the use of applied functions. The effect of filtered coordinates on the Roll angle is seen in the Fig. 8.

The dotted line shows the roll angle computed using the Kalman filtered coordinates. There are eight coordinate values for four points. The coordinate values are integers and reported in the resolution of 1024x768 pixels. The errors occur

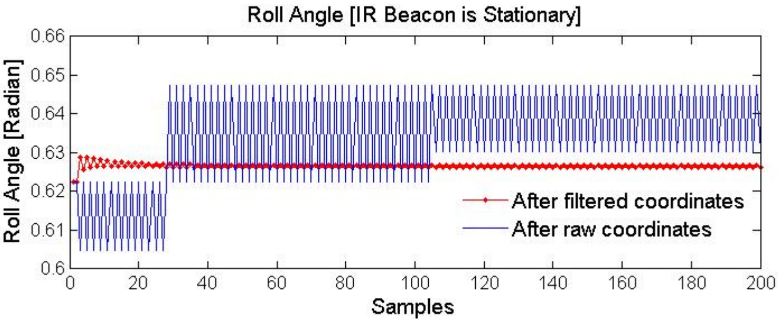


Fig. 8. Roll angle comparison computed using filtered and raw coordinates

jumping the coordinate values in the neighborhood of the LEDs' real position. The Kalman filter tries to estimate real position. The estimated coordinates have decimal values jumping between neighborhoods in small intervals.

Although the Kalman filter reduces error level in computed Euler angles, we still must use appropriate filters to smooth the visualization. In our experiments, we applied a moving-average filter to computed Euler angles for smoothing the visualization. The moving-average filter behaves as a low pass filter and attenuates the high frequencies. We continue to analyze the results, and design and test these and new filters for future phases of the project. A study of a first order low pass Butterworth filter on roll angle is given Fig. 9. As seen in the Fig. 9 Butterworth filter produces better results in smoothing the data.

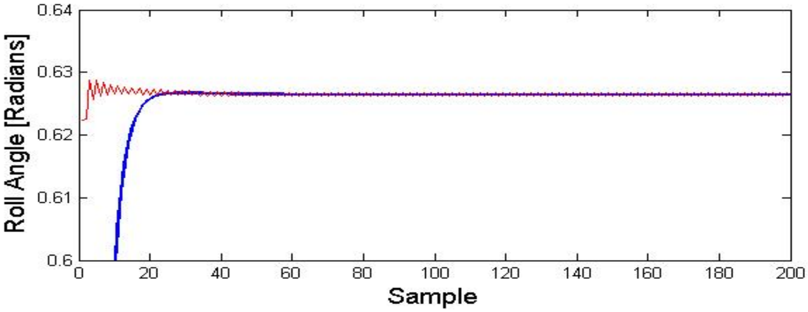


Fig. 9. Butterworth filtering after Kalman filter

6 Conclusion and Future Work

We demonstrated tracking and pose estimation of a moving IR beacon by using OpenCV and Wiiuse libraries. We built an IR LED beacon consisting of four non-coplanar LEDs and estimated the pose of this beacon using the OpenCV Posit function. Before calling cvPOSIT function to extract the rotation matrix and position vector of the device we applied a Kalman filter to smooth the visualization.

Movement of the manipulated object (in our study we used a 3D brain model) was very jittery before implementing the Kalman filter due to the very high level of noise associated with the Wiimote digitization hardware (also found by the studies [1, 12]). We suspect this noise arises from the actual low resolution of the Wiimote camera and the resulting hardware interpolation of coordinate values within the Wiimote camera.

Our results indicate that band pass filter algorithms can be used to achieve smooth pose estimation in virtual environments. Wiimotes provides an inexpensive and fast developing hardware platform for computer vision and gesture recognition studies, although they cannot be used in the application where precision and accuracy is essential. Future research will focus on developing greater pose accuracy, greater field of view and the ability to track many IR beacons simultaneously.

Acknowledgements. This project has been jointly sponsored through an Australian Research Council Discovery Grant, DP0988088 (Kavakli) 2009-2012 titled A Gesture-Based Interface for Designing in Virtual Reality and a Macquarie University Research Development Grant (Gilchrist, Twamley, Kavakli) 2008-2010 titled Immersive Virtual Scientific Collaboration. Special thanks to Richard Miller for the initial implementation of the IR tracking device, John Porte for the Vizard and 3D TV implementation and demonstrations and Peter Dekker for IR Filter measurements.

References

1. Chow, Y.W.: Low Cost Multiple Degrees of Freedom Optical Tracking for 3D Interaction in Head Mounted Display Virtual Reality. International Journal of Recent Trends in Engineering 1, 152–156 (2009)
2. Wronski, M.P.: Design and Implementation of a Hand Tracking Interface using the Nintendo Wii Remote, Dissertation, University of Cape Town (2008)
3. Azmi, A., Alsabhan, N.M., AlDosari, M.S.: The Wiimote with SAPI: Creating an Accessible Low-Cost, Human Computer Interface for the Physically Disabled. International Journal of Computer Science and Network Security 9(12), 63–68 (2009)
4. Duran, L., Carmona, M.F., Urdiales, C., Peula, J.M., Sandoval, F.: Conventional Joystick vs. Wiimote for Holonomic Wheelchair Control. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009. LNCS, vol. 5517, pp. 1153–1160. Springer, Heidelberg (2009)
5. Fleisch, T.: Wiimote Virtual Reality Desktop, `http://www.vrhome.de/vr-applications`
6. Vuong, P., Kurillo, G., Bajcsy, R.: Oliver Kreylos' Wiimote Tracking Algorithm and its Limitations, Summer, Research Report (2008), `http://phongvuong.com/wordpress/wp-content/uploads/2009/11/wiimotePaper1.pdf`
7. Dehling, E.: Using Multiple WiiMote Cameras to Control a Game, Student Project, `http://hmi.ewi.utwente.nl/tkisis/persoon/Eike%20Dehling`
8. Vural, G., Tekkaya, G., Erogul, C.: Using Head and Finger Tracking with Wiimote For Google Earth Control, Student Project, `http://www.ceng.metu.edu.tr/~erogul/ceng701/project/.../paper-draft.pdf` (accessed June 15 2010)
9. Wang, Z., Louey, J.: Economical Solution for an Easy to Use Interactive Whiteboard. In: Japan-China Joint Workshop on Frontier of Computer Science and Technology (2008)
10. Lee, J.: Hacking the Nintendo Wii Remote. IEEE Pervasive Computing 7(3), 39–45 (2008)
11. DeMenthon, D.F., Davis, L.S.: Model Based Object Pose in 25 Lines of Code. International Journal of Computer Vision 15(1-2), 123–141 (1995); Special issue: image understanding research at the University of Maryland
12. Daniel, G., Petersen, T., Kruger, V.: A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 706–715. Springer, Heidelberg (2009)
13. Campa, G., Mammarella, M., Napolitano, M.R., Fravolini, M.L., Pollini, L., Stolarik, B.: A comparison of Pose Estimation algorithms for Machine Vision based Aerial Refueling for UAVs. In: 14th Mediterranean Conference on Control and Automation, vol. 45(1), pp. 138–151 (2009); IEEE Transaction on Aerospace and Electronic System

14. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Sebastopol (2008)
15. Open Source Computer Vision Library Reference Manual, Intel Corporation (2009)
16. Dorfmuller-Ulhaas, K.: Optical Tracking, From User Motion To 3D Interaction, PhD Thesis, Vienna University of Technology (2002)
17. Chui, C.K., Chen, G.: Kalman Filtering with Real-Time Applications, 4th edn. Springer, Hielderberg (2009)
18. Welch, G., Bishop, G.: An Introduction to the Kalman Filter, `http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html`
19. Brindza, J., Szweda, J., Liao, O., Jiang, Y., Striegel, A.: WiiLab: Bringing Together the Nintendo Wiimote and MATLAB. In: Proceedings of the 39th IEEE International Conference on Frontiers in Education Conference, San Antonio, Texas, pp. 1373–1378 (2009)
20. Kavakli, M.: Gesture Recognition in Virtual Reality. Special Issue on Immersive Virtual, Mixed, or Augmented Reality Art of the International Journal of Arts and Technology (IJART) 1(2), 215–229 (2008)
21. Kreylos, O.: Research and Development Homepage, `http://idav.ucdavis.edu/~okreylos/ResDev/Wiimote/index.html`