Entertainment Computing 12 (2016) 9–27

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

A low-cost infrared-optical head tracking solution for virtual 3D audio environment using the Nintendo Wii-remote ☆

Yashar Deldjoo [a,b], Reza Ebrahimi Atani [b,*]

[a] Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden
[b] Department of Computer Engineering, University of Guilan, P.O. Box 3756, Rasht, Iran

ARTICLE INFO

Article history:
Received 29 September 2014
Revised 21 September 2015
Accepted 31 October 2015
Available online 10 November 2015

Keywords:
Head tracking
Rigid body
Degrees of freedom
Virtual 3D audio system
3D audio rendering
Wii-remote

ABSTRACT

A virtual audio system needs to track both the translation and rotation of an observer to simulate a realistic sound environment. Current existing virtual audio systems either do not fully account for rotation or require the user to carry a controller at all times. This paper presents a three-dimensional (3D) virtual audio system with a head tracking unit that fully accounts for both translation and rotation of a user without the need of a controller. The system consists of four infrared light-emitting diodes on the user's headset together with a Wii-remote to track their movement through a graphical user interface. The system was tested with a simulation that used a pinhole camera model to map the 3D-coordinates of each diode onto the two-dimensional (2D) camera plane. This simulation of 3D head movement yields 2D coordinate data that were put into the tracking algorithm and to reproduced the 3D motion. The results from a prototype system, assembled to track the 3D movements of a rigid object were also consistent with the simulation results. The tracking system has been integrated into an Ericsson 3D-audio system and its effectiveness has been verified in a headtracked virtual 3D-audio system with real-time animating graphical outputs.

1. Introduction

The human auditory system plays a major role in our three-dimensional (3D) spatial awareness and in extracting information from our environment. It is through hearing that we localize sound sources and identify and follow the movement of such sources around us. Human sound perception is based on binaural hearing, which results in the same sound reaching each ear at a slightly different time and with different intensity. The first phenomenon is called the interaural time difference (ITD) and the second the interaural intensity difference (IID). Further, sound waves interact with the torso, head, and especially the external ear or pinna, becoming further altered before reaching the eardrums. These interactions modify the frequency content of the signals by reinforcing some frequencies and attenuating others depending on the sound's direction of arrival. Therefore, the frequency spectrum reaching one ear will be slightly different from that reaching the other. The brain uses the IID, ITD, and spectral difference (spatial cues) between signals received by the ears to determine the location of the source [1,2].

Virtual 3D audio technologies use specialized filters known as head-related (HR) filters to render sound. Because HR filters contain all the information needed to locate a sound source, they can artificially spatialize sounds if the appropriate HR filters are known, a process known as binaural synthesis. HR filters are unique for every position and angle of incidence and are usually measured for sound sources in many locations relative to the head to obtain a database of hundreds of HR filters. Using HR filtering of the source signal, a virtual audio system can simulate 3D wave propagation that triggers the spatial hearing of the listener. As shown in Fig. 1, the listener experiences a 3D audio environment when listening to a generated sound signal, which is different from the 3D environment in which he or she actually exists in.

A major problem for virtual 3D audio systems is head movement [1,2]. This changes the direction of arrival (DOA), and hence the path of each sound wave to each eardrum, thereby changing the spatial information the brain needs to correctly locate the source. Therefore, to have a realistic virtual 3D audio system, head movements have to be tracked. When the listener's orientation is not tracked, the simulated 3D audio space moves with the head movements of the listener, which is not realistic. A dynamic virtual

☆ This paper has been recommended for acceptance by Ryohei Nakatsu.

* Corresponding author.
E-mail addresses: yashar.1984.deldjoo@ieee.org (Y. Deldjoo), rebrahimi@guilan.ac.ir (R.E. Atani).

http://dx.doi.org/10.1016/j.entcom.2015.10.005
1875-9521/

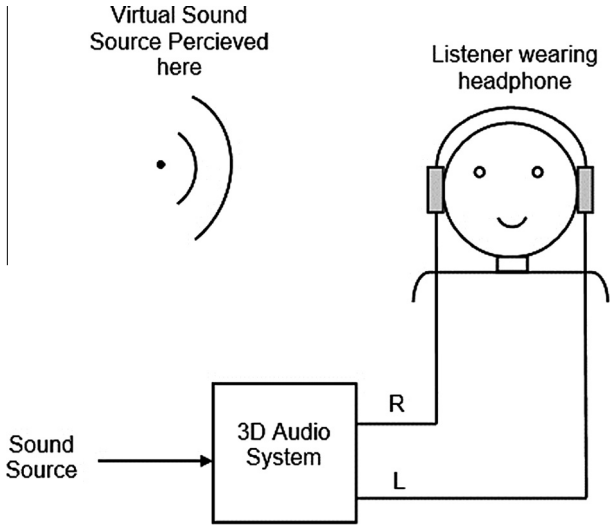


Fig. 1. 3D audio system using binaural synthesis.

3D audio system in which the user's movement alters the sound environment requires the following:

1. Head tracking (HT) with six DOFs to account for both position and orientation.
2. Fast processing of the tracking algorithm to make changes in virtual sound properly correspond with changes in the listener's position and orientation. Ideally the brain should not detect any delay between change in position/orientation and change in sound properties.
3. Accuracy in detecting events, specifically the type of event. For example, an upward–downward translation should not be reported as a rotation around the *y* or *x* axis (Fig. 2).

Visual HT systems can be categorized into two classes: marker-free systems and marker-based systems. Marker-free systems only exploit optical sensors to measure movements of the human head. They acquire the information directly from the recorded images without trying to build a 3D representation of the user's head. Building such systems usually involves tracking of facial features

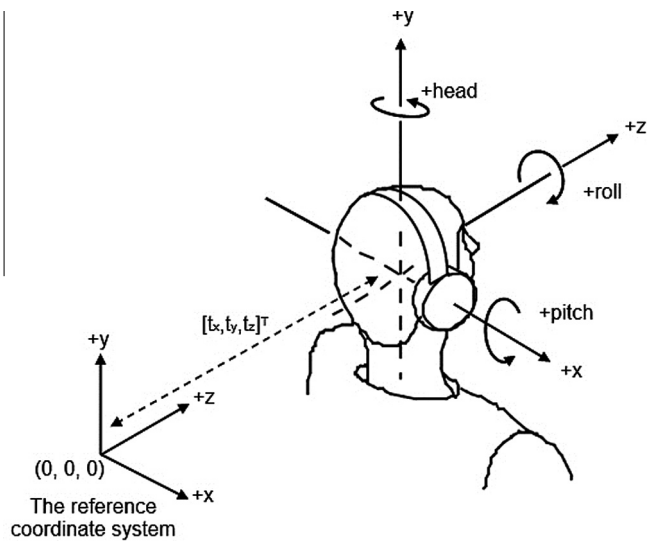


Fig. 2. Axis rotations (pitch, yaw and roll) and translation specified by a translation vector $[t_x, t_y, t_z]^T$ in a HT system.

that are non-rigid and subject to various articulation and deformation due to muscle contraction and relaxation and usually results in movements in high degrees of freedoms (DOFs). These models are also known as appearance-based methods and are classified in 2D or 3D tracking methods [3–5]. On the other hand, marker-based systems apply identifiers which are placed upon the human head to capture movements. These systems build up a 3D model that represents human head and then try to estimate the 3 parameters that define the translation and 3 parameters that define the rotation with respect to reference coordinate system shown in Fig. 2. This results in tracking systems with 3-Degrees-Of-Freedoms (3DOF) or 6DOF in which the latter is considered as the most complete representation of a rigid object in three-dimensional space. The marker-based systems are referred to as model-based systems [4–6].

Some 2D and 3D approaches have been developed for estimating head orientation. The 2D approaches [3] compare each new head image with a set of reference templates and then use the closest-matching template as the head pose. The advantages and disadvantages of such systems are discussed in [7]. Lee [8] has developed a 3D HT system that produces view-dependent images. This system uses a head-mounted sensor bar with a Nintendo Wii remote (Wiimote™) at the base of the display. It assumes that the observer is always looking into the middle of the screen, and does not account for head rotation. Kreylos' system [9] uses a direct rigid-body transformation to yield an approximate solution to the orientation problem, but that system requires the user to carry a controller and uses the controller's inertial data to achieve 6DOF tracking. Other devices that track all the DOFs similarly require the user to carry or wear a controller, which is inconvenient, cumbersome, and difficult to implement at home [10].

Wiimote is used in several other applications such as automated assembly simulation, head position tracking for gaze point estimation, home assessment of Parkinson's disease and as a tracking system for braille readers, etc. In [11] a low cost and simple location management system using the Wii remote controller and infrared LEDs is proposed in which a Wiimote controller is placed on a mobile robot pointing upward toward a number of IR LEDs placed on the ceiling. In [12] a Wiimote-based motion capture system is developed for automated assembly simulation. In [13] a pair of Nintendo Wiimote imaging sensors is used to create a stereo camera for 6DOF position tracking of the headset for eye gaze estimation. Application of the Nintendo sensor-rich data for building of a home-based assessment of Parkinson's disease (PD), known as WiiPD is presented in [14]. Another example is shown in [15] where a Wiimote and a refreshable braille display are used to build a cheap an easy-to-use finger tracking system for studying braille reading. A low-cost high accuracy 3D tracker is implemented in [16] using the Wiimote to detect the pose of a target. It applies the triangulation techniques to build the 3D location of the markers. In Fig. 3, the main criteria for design of a Wii-Based HT solution is presented. As can be seen, the problem of HT is an optimization problem on several parameters. In general, one of the main advantages of Wii-based HT systems is the low cost of building such systems. The next important factor is the precision of tracking which in the case of rigid objects can be performed with a precision of 3DOF to maximum 6DOF. On the other hand, there is a growing trend on using multiple Wii-motes to create a stereo-vision based system using triangulation technique which makes it easier to bring 6DOF tracking into reality. The camera mode is yet another determining factor in that the camera can be used in two stationary or non-stationary modes for the task of HT.

This research aims to solve the virtual 3D HT simulation system by developing a visual tracking system based on infrared positioning that updates the listener's orientation vectors without the cumbersome need of carrying a somewhat heavy Wii-remote

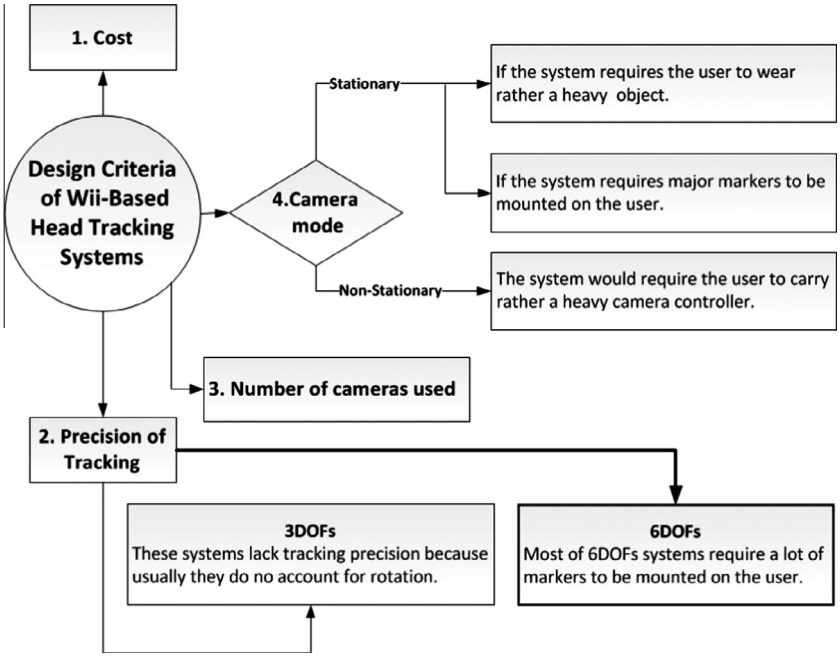


Fig. 3. Design criteria of Wii-based HT systems.

controller. The system enables the DOAs for all sources to be updated correctly and the appropriate HR filters to be used in the 3D audio system. This keeps the virtual sound sources fixed in space from the perspective of the listener, which creates a realistic 3D audio space in a way that is convincing and comfortable. In Table 1, a comparison of different papers that use the Nintendo Wii-emote in integration with an application is presented. In each row, the advantage of each system is labeled in bold. Ideally the solution should have the following characteristics: (1) low cost, (2) offers highest achievable precision, i.e. 6DOF of tracking (3) uses less camera which leads to less complexity and cost of the solution, (4) uses a stationary camera, in order that the user does NOT need to carry a controller which is inconvenient, (5) uses the IR illuminators in a convenient manner, and (6) applies the tracking in integration with an application (e.g. VR application).

The main contribution of this work is that it offers a very low-cost and affordable HT solution with fairly accurate performance. The proposed system uses only a single Wiimote (and four IR illuminators) for the task of HT which compared to the works [12,25,28,29] is less expensive, more precise (i.e. 6DOF tracking) compared to the works [24–27,29,30,18,8], does NOT require the user to carry the controller all the time as in [9] and applies the tracking solution in integration with an application (e.g. in virtual reality) as opposed to [25,17,27] where the authors do NOT show the applicability of the proposed solution in integration with an application. To date, the author knows of no virtual 3D simulation system that is able to track head movements fully with minimal burden to the observer.

The rest of paper is structured as follows: Section 2 presents the theoretical methods and models. Then Section 3 details the integration and algorithms adopted, while Section 4 reports the system implementation and testing. Concluding remarks are given in Section 5. For the benefit of the interested reader, the Appendix gives a tutorial on using the prototype system.

2. Theoretical background

The following section provides a theoretical background to the mathematical models in further details.

2.1. Motion capture

3D movement of a rigid object can be modeled with a reference rigid object rotated and translated to give the transformed rigid body. During a camera recording, the camera detects the transformed rigid body and projects its points on its image plane.

Table 1
Comparison of solutions that use the Nintendo Wii emote for a VR application.

No.	Paper	Cost	Precision	# of Cameras	Camera mode	IR Illuminators	Integration with an application
1	[12]	High	**6-DOF**	8	**Stationary**	2-IR LEDs set	**Assembly simulation**
2	[4]	**Low**	1-DOF	**1**	**Stationary**	1-IR LED	**Head tracking for use in MRI**
3	[25]	High	3-DOF	2	**Stationary**	**4-LEDs**	No
4	[26]	**Low**	2/3-DOF	**1**	**Stationary**	2-LED	**VR**
5	[17]	**Low**	**6-DOF**	**1**	**Stationary**	8-LEDs on a circlet	No
6	[27]	**Low**	2/3-DOF	**1**	**Stationary**	1-LEDs on a finger	No
7	[28]	Mid	**6-DOF**	4	**Stationary**	4-IR LED + IR bacon	**VR**
8	[29]	Mid	3-DOF	4	**Stationary**	Not given	**Assessing surgical skills**
9	[30]	**Low**	1-DOF	**1**	**Stationary**	1-LED	**Interactive picture navigating**
10	[9]	**Low**	**6-DOF**	**1**	Moving	4-IR LED + Accelerometer data	**Wiimote Input Device Flight Gear flight simulator**
11	[8]	**Low**	2/3-DOF	**1**	**Stationary**	2-IR LED	**VR Display**

2.2. Camera model

As shown in Fig. 4 this work uses the classical pinhole camera model. This model is a basic mathematical model that does not consider all the optical effects present in an image. However, it is sufficient for capturing motion in many image processing applications. The camera's coordinate system is defined with the origin at the focal point of the camera. The focal point is on the positive side of the z-axis, which points toward the camera's optical axis. The focal length, f, is the distance from the focal point to the image plane [19,20].

The relationship between the 3D point $M = [X, Y, Z]^T$ and its projection $m = [x, y]^T$ is given by similar triangles:

$$\begin{aligned} \frac{x}{X} &= \frac{f}{-Z} \rightarrow x = -f\frac{X}{Z} \\ \frac{y}{Y} &= \frac{f}{-Z} \rightarrow y = -f\frac{Y}{Z} \end{aligned} \tag{1}$$

In real applications, the inverse problem is of interest, that is, to obtain the coordinates of M from its image m. This problem is an underdetermined problem in which Z stands as a free parameter [17]:

$$\begin{aligned} X &= -Z\frac{x}{f} \\ Y &= -Z\frac{y}{f} \end{aligned} \tag{2}$$

2.3. Camera parameters

For a pinhole camera model, the mapping process depends on the following parameters [21]:

1. The intrinsic camera matrix $K \epsilon \Re^3$ contains the internal parameters of the camera. These parameters define the optical, geometric and digital characteristics of the viewing camera such as focal length and lens properties [19,20]. They can be approximated using camera calibration techniques and are assumed to be known and constant during tracking.
2. The external parameters of the camera, $[R|t] \epsilon \Re^{3\times4}$, also known as the camera pose, use a rotation R and translation t to relate 3D points in the world coordinate system to that in camera coordinate system and allow transformation between the two. These parameters determine positions of objects in 3D space independent of the camera location. These parameters are not known and must be determined through pose estimation techniques.

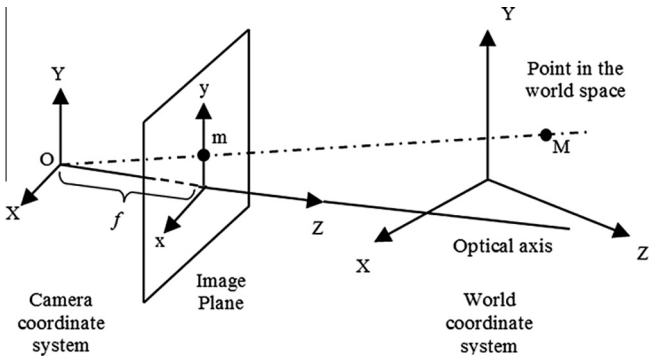


Fig. 4. Pinhole camera model.

2.4. Pose estimation method with the perspective n-point problem

Under a homogeneous representation that defines a 3D point $P \epsilon \Re^4$ in the world coordinate system and its 2D mapped image $q \epsilon \Re^3$, the motion-capturing process can be expressed in the compact form [21]:

$$q = K[R|t]P \tag{3}$$

Pose can be formulated as a minimization problem:

$$\min_{R,t} \|q - K[R|t]P\|_p \tag{4}$$

where the norm is an Lp-norm ($p \geqslant 1$). The process entails searching for the camera pose $[R|t]$ that best satisfies Eq. (4) under a known calibration K. With a sufficient number of 2D image data, pose estimation algorithms can be classified roughly into two main classes [21]:

1. Perspective n-point (PnP): This algorithm attempts to estimate the parameters that define the solution space of all valid poses.
2. Direct rigid-body transformation (DRBT): This algorithm attempts to directly estimate the camera pose.

From our experimental evaluations, we select a robust version of the PnP method with $n = 4$ (P4P) for final implementation. The perspective n-point (PnP) problem is defined as follows: We are given a set of 3D points $\{P_i\}_{i=1}^n$ with coordinates (X_i, Y_i, Z_i) known in some world coordinate system. Let the points be projected onto the image plane of the camera to give a set of 2D points $\{q_i\}_{i=1}^n$ with known coordinates (x_i, y_i) in the camera coordinate system. It is desired to evaluate the unknown camera-point distances $\{r_i\}_{i=1}^n$ as depicted in Fig. 5.

Each pair of correspondence $P_i \rightarrow q_i$ & $P_j \rightarrow q_j$ gives a constraint on the unknown camera-point distances r_i and r_j according to the law of cosines:

$$r_i^2 + r_j^2 - r_i r_j \cos\theta_{ij} = d_{ij}^2 \tag{5}$$

where d_{ij} is the inter-point distance between the ith and jth points known to us as one of the parameters of the reference model, and θ_{ij} is the 3D viewing angle subtended at the camera center by the ith and jth points. The cosine of the viewing angle can be directly evaluated from the image plane data:

$$\cos\theta_{ij} = \frac{\langle q_i, q_j \rangle}{|q_i||q_j|} \tag{6}$$

Eq. (5) is quadratic with two unknowns. When three features (q_1, q_2, q_3) are available, the cosine relation is formulated for three pairs: (q_1, q_2) (q_1, q_3), and (q_2, q_3), and the system of expanded equation will have three quadratic unknowns with three equations:

$$\begin{cases} r_1^2 + r_2^2 - r_1 r_2 \cos\theta_{12} = d_{12}^2 \\ r_1^2 + r_3^2 - r_1 r_3 \cos\theta_{13} = d_{13}^2 \\ r_2^2 + r_3^2 - r_2 r_3 \cos\theta_{23} = d_{23}^2 \end{cases} \tag{7}$$

With four points available, the system of equations will become over-determined. In general, projection from 3D to 2D space is a nonlinear operation. The developed set of equations in perspective n-point problem is a bilinear system that can be solved iteratively by using an optimization algorithm. In the optimization algorithm, we minimize the residual distance $(f_i(r) - d_{jk}^2)$ over the unknown camera-point distance $\boldsymbol{r}$:

$$f_i(\boldsymbol{r}) = r_j^2 + r_k^2 - r_j r_k \cos\theta_{jk} \rightarrow \min_r \sum_{i=1}^{n} \|f_i(\boldsymbol{r}) - d_{jk}^2\|^2 \tag{8}$$

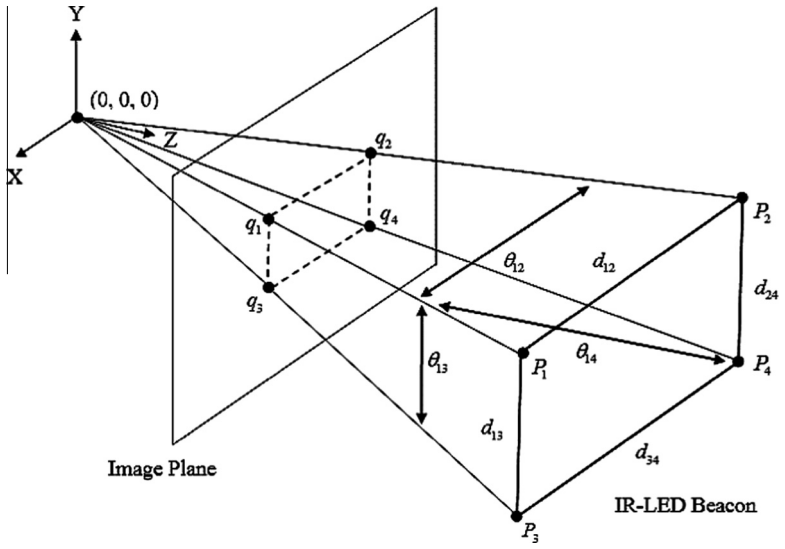


Fig. 5. Projections of four reference points on the camera's image plane; PnP method for $n = 4$, P4P.

The problem can be solved using Gauss–Newton or Levenberg–Marquardt algorithms.

3. Solution specifications and design

We integrate the HT system with the 3D virtual audio system developed by Ericsson Research. This paper proposes a novel HT system that uses Nintendo's Wiimote controller as the main tracking device. The Wiimote has a built-in optical infrared (IR) camera that can immediately determine the locations of four light sources in a 2D image plane. Therefore, by mounting four IR diodes on the listener's headset, we can use the Wiimote to determine both the position and orientation of the listener. In addition to IR tracking, the Wiimote also incorporates a three-axis linear accelerometer for motion sensing. However, the Wiimote-based HT system developed for this research is purely optical and is built by measuring the (x,y) coordinates (2D data) from the IR camera and reconstructing the 3D model using pose estimation techniques. We use all four available points from the Wii-remote to recover pose with all six DOFs.

3.1. Integration

With HT in place, the user's head acts as an input and enables the 3D audio system to generate discernible directional sounds based on the movements of the four designated target points attached to the listener's headphone (Fig. 6). These four points are monitored by a graphical user interface (GUI). In such a dynamic virtual environment, sound sources, listener and scene objects may all be moving, and the listener can naturally interact with animated graphics.

The proposed solution will entail a listener wearing a headphone with a number of IR-LEDs mounted on it. The Wiimote camera views the IR diodes and reports dots' location on its image plane to the computer via Bluetooth connection. The head-tracking algorithm calculates the head pose from the image plane data. The results are then reported to a 3D audio rendering engine for

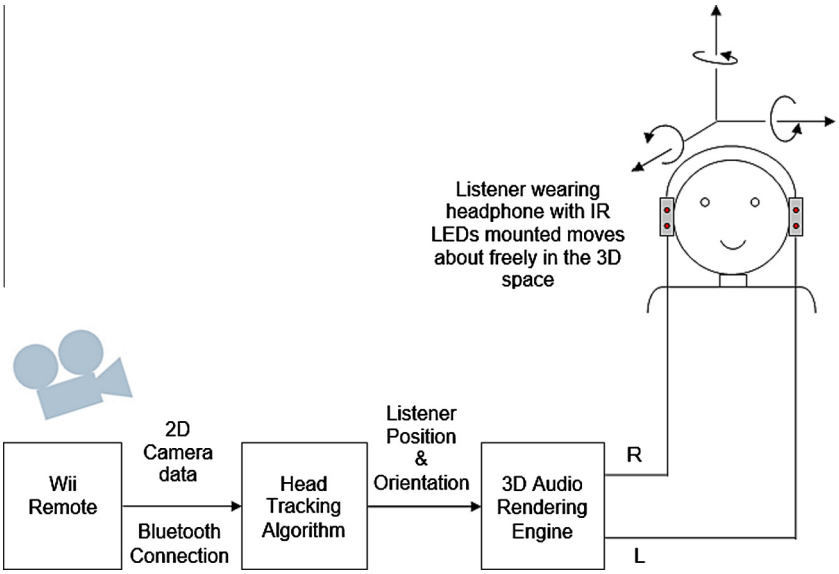


Fig. 6. The architecture of the proposed HT solution.

updating binaural signals and introducing the spatial 3D cues in the final audio signal. The flowchart in Fig. 7 illustrates the different stages involved in developing the Wii-remote-based HT system:

3.2. Constraints

The limitations of using the Wiimote are:

1. The maximum number of trackable markers is $n_{max} = 4$. The complexity of the system developed and the accuracy of the tracking results of marker-based tracking systems depends on the number of tracked sources.
2. The data are quantized.The camera uses interpolation to reach a higher resolution. As a result, the raw data contain quantization noise that can propagate through tracking equations and result in spread and intensification of the tracking error in the achieved result. This makes it very important to choose a HT algorithm that is robust against the quantization noise.

The camera's FOV in the horizontal plane is 22.5 m in each direction and the maximum trackable distance with regular LEDs is around 2 m. As a result, the operator's 3D movements would be constrained if only one camera is used for tracking.

3.3. Selecting pose estimation algorithm

The Wii camera reports and updates four pairs of coordinates every 20 ms, and the 3D head pose has to be estimated accordingly. The types of nonlinearities in the equations of the PnP method can be solved fairly accurately and rapidly with standard iterative methods, and thus it was selected as the core tracking engine because of its good performance.

3.4. Design of the tracking algorithm

The complete information flow in the HT algorithm is illustrated in Fig. 8. First, the listener or observer of the 3D audio environment wears four visual on-body sensors. The sensors are IR light emitting sources attached to the listener's headphone or eyeglasses. The moving target points (markers) are detected by the camera. The camera can report a maximum of four 2D points at each time. The data are neither ordered nor associated with any of the points on the physical IR-plane. Therefore, the randomly numbered data have to be reordered to correspond with each of the points on the tracking IR plane. If one point is missing, it can be recovered with a prediction method. If more than one point is missing, it is not possible to estimate pose with six DOFs; only three DOFs will be possible.

Once all markers are detected, tracking is achieved with the PnP method with $n = 4$. A smoothing filter can be used to remove noise from the computed data. The clean data are then decomposed into the real position and orientation of the user's head:

3.5. Feature registration

During the tracking, the camera detects the moving markers as bright spots in its 2D image plane. These spots are projections from 3D space to 2D space as defined by the camera projection equations. After the Bluetooth connection is established and the controller and the computer are paired, during tracking the following information about each detected feature is reported:

$$(i \quad x \quad y) \quad i \in [0-3] \quad x \in [0-1023] \quad y \in [0-768] \tag{9}$$

where i is the label of the detected IR-LED and (x, y) are the 2D Cartesian coordinates of the features on the camera image plane. The LED labeling is done arbitrarily by the driver software *Wiiuse*,

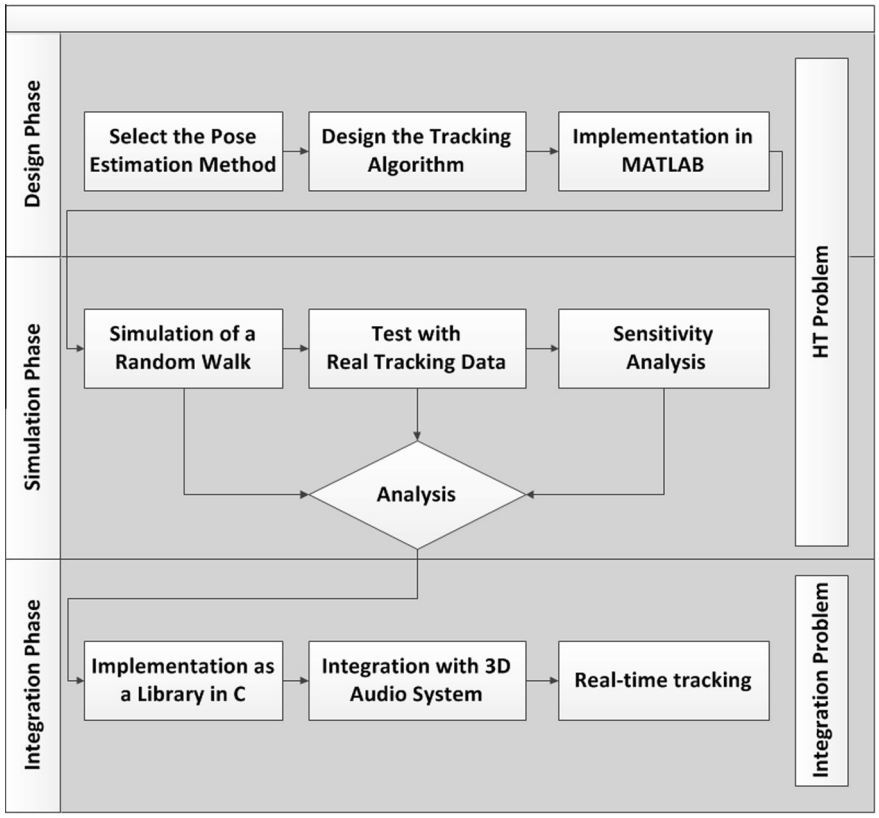


Fig. 7. Different stages involved in designing the Wii-Remote-based head-tracking system.

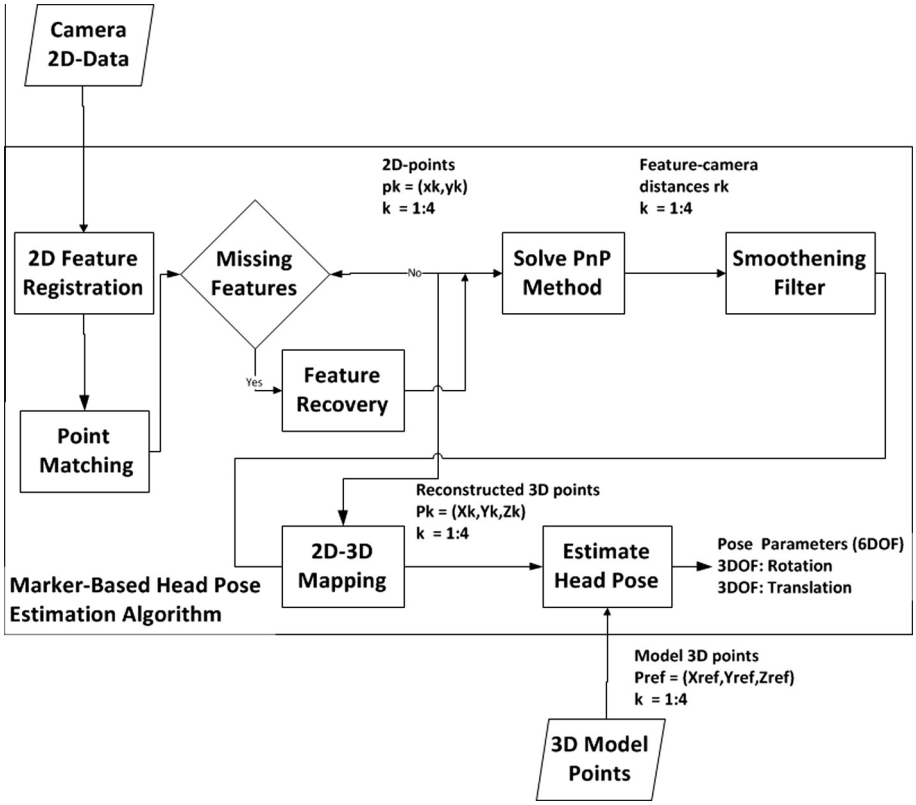


Fig. 8. Information flow in the proposed HT algorithm.

which maintains the same labeling format for as long as the four points remain constantly within the camera's field of vision. The labeling format has to be changed to suit our own configuration. This is because the priori saved distances. d_{ij} have to be matched with the right distances in the P*n*P method so that the calculations are valid.

3.6. Point matching

Projected points can be matched to their representative 3D points on the physical IR plane by using the shape of the infrared pattern. The four LEDs mounted on the operator's headphone form a rectangular shape. When the camera is at rest i.e. relatively horizontal and orthogonal to the IR-plane, by reading and comparing the x–y coordinates we can label the points. The labeling procedure is done only once, at the beginning of the tracking, because the updates are reported every 20 ms and it is assumed that in such a short time similar points will remain close to each other (i.e. two consecutive data packets appear identical). This fact could be used to choose the smallest distance between point i in the update k and all the other points in update $k+1$; the smallest distance matches the corresponding point in group $k+1$. This should be done for all the other points in group k as Fig. 9 shows.

3.7. Feature recovery

Three points is the minimum required to describe the object pose with six DOFs. Below that threshold, only three DOFs can be calculated. Over that threshold, the excess of points can be used to improve the accuracy of the evaluated tracking results. Although our HT algorithm is designed to function with four inputs, it is possible to solve a reduced version of P4P and solve a P3P algorithm to calculate pose data with six DOFs. Once P3P is solved, we recover the three features in 3D space. By knowing our configuration model, we predict the fourth point and the algorithm can proceed. Solving the P3P algorithm has some numerical constraints; therefore, it is suggested to avoid situations where a point is occluded for a long period of time.

3.8. Solving the P4P problem

The first step to evaluate the six pose parameters is to evaluate the system of over-determined equations in Eq. (5). In this work we select the Gauss–Newton algorithm to solve this set of nonlinear equations.

Given the distances d_i between the dots and calculated cosines of the angles c_i, the basic idea is to minimize the residual distances on both sides of the expanded equation set (six equations with $n = 4$). A starting value r_i^0 is required to initialize the algorithm. To improve the efficiency of the algorithm in terms of speed and convergence, the converged iteration results are fed back into the algorithm as the next initial values. Later we will present an analysis of the sensitivity of the computed location parameters to the existence or lack of feedback.

According to the Gauss–Newton method, the Jacobean matrix should be computed at each iteration. The Jacobian matrix here is a 6×4 matrix and is used to solve a least-square problem. In MATLAB, the backslash (\) operation finds the least-square solution to the system of over-determined (or under-determined) equations using QR decomposition. The operation automatically handles the rank deficiency problems. In C, however, we programmed the classical Schur algorithm to solve the least-square problem. The algorithm converges in a few iterations.

3.9. Smoothing filter

A smoothing filter is implemented to remove noise from the computed data and detect possible outliers during the computations

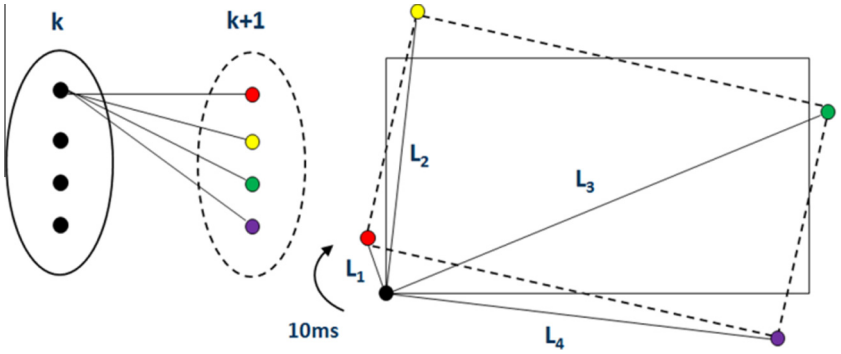


Fig. 9. Point matching at each update. We choose the smallest distance L1 < L2 < L3 < L4.

that may have resulted from the quantization noise introduced by the camera. We use a median filter of size five as the smoothing filter to detect the outliers (high-frequency components) and replace the value of a sample by the median of values in the neighborhood of that sample. A median filter is a nonlinear filter used to perform a high degree of noise reduction. The median filter is suitable because it is not sensitive to the level of noise data.

3.10. 2D–3D mapping

The next step after computing the camera-point distances is to calculate the 3D coordinates of the target points. The goal is to match the 2D feature points $q_i = (x_i, y_i)$ with the corresponding 3D target $P_i = (X_i, Y_i, Z_i)$. This problem is illustrated in Fig. 10, where f is the camera's focal length and Vi is the vector representing the point directions originating from the camera center. The 2D feature points lie in the direction of this vector, and thus the correspondence between 2D and 3D can be derived according to the following formulas:

$$\begin{aligned} X_i(\text{mm}) &= \frac{x_i(pixel)}{\sqrt{x_i^2 + y_i^2 + f^2}} |r_i(\text{mm})| \\ Y_i(\text{mm}) &= \frac{y_i(pixel)}{\sqrt{x_i^2 + y_i^2 + f^2}} |r_i(\text{mm})| \\ Z_i(\text{mm}) &= \frac{f(pixel)}{\sqrt{x_i^2 + y_i^2 + f^2}} |r_i(\text{mm})| \end{aligned} \tag{10}$$

3.11. Estimate head pose

So far we have constructed the moving rigid body and acquired its 3D coordinates $P_i = [X_i, Y_i, Z_i]^T$. A reference model 3D rigid body $P_i^{ref} = [X_i^{ref}, Y_i^{ref}, Z_i^{ref}]^T$ is known and saved from the beginning. By comparing these two data sets, we obtain $\boldsymbol{R}$ and $\boldsymbol{T}$ that rotate and translate the reference model $P_i^{ref} = [X_i^{ref}, Y_i^{ref}, Z_i^{ref}]^T$ to match the recovered model $P_i = [X_i, Y_i, Z_i]^T$:

$$P = RP^{ref} + T \tag{11}$$

The problem of finding R and T is known as the absolute orientation problem. Given two point sets P_i and P_i^{ref} for $i = 1 : N$ related by:

$$P_i = \underbrace{RP_i^{ref} + T}_{P_i^{tr}} + N_i \tag{12}$$

where N_i the noise vector, find the R and T that minimize the following least-square formulation:

$$\Sigma^2 = \sum_{i=1}^{N} \| P_i - (RP_i^{ref} + T) \| \tag{13}$$

The solution can be acquired using an iterative or non-iterative algorithm. We use a non-iterative algorithm based on singular value decomposition (SVD) to solve the least-square problem. The solution is obtained by decoupling rotation and translation and solving for them individually in two steps.

It is shown in [22] that if $\hat{R}$ and $\hat{T}$ are the solutions to Eq. (14), then $\overline{P} = \overline{P}^{tr}$ where:

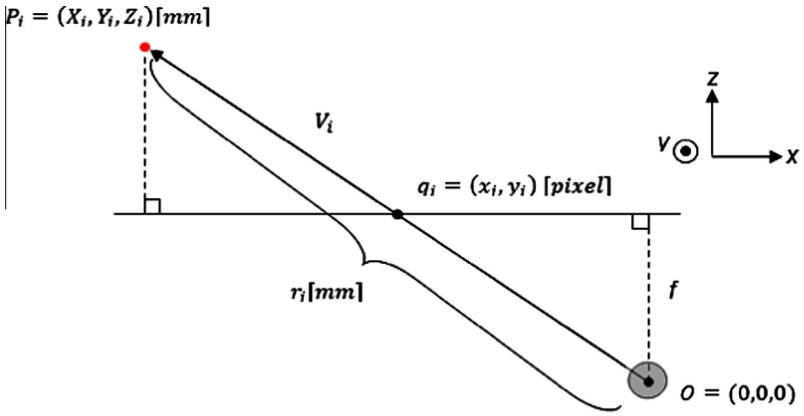


Fig. 10. 2D–3D mapping.

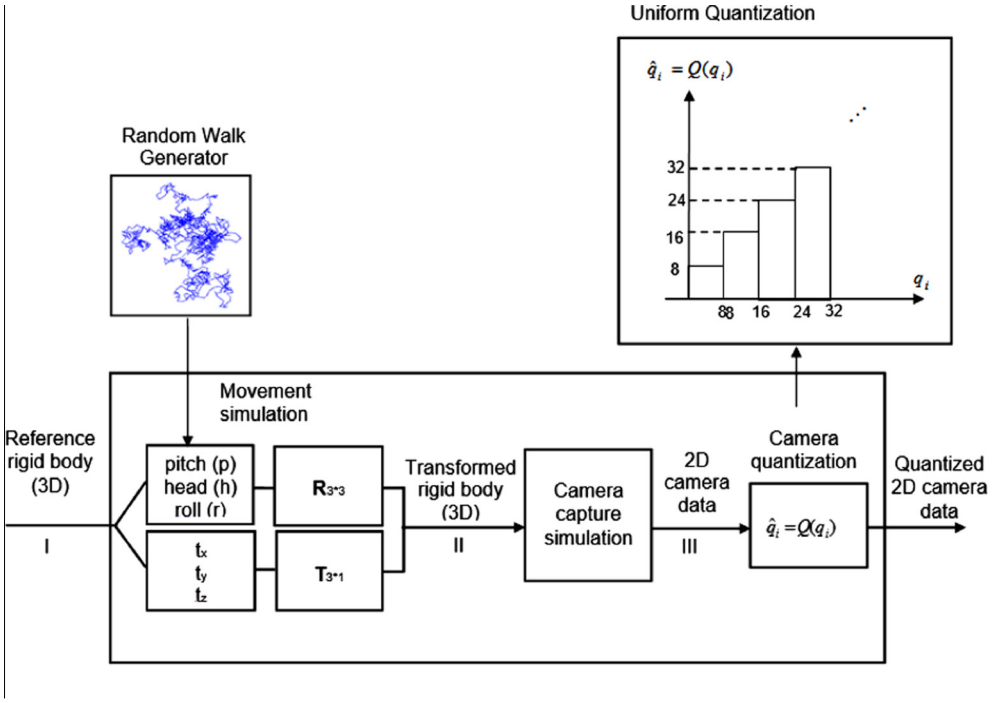
$$\begin{aligned}\overline{P} &= \frac{1}{N}\sum_{i=1}^{N} P_i \\ \overline{P}^{ref} &= \frac{1}{N}\sum_{i=1}^{N} P_i^{ref} \\ \overline{P}^{tr} &= \frac{1}{N}\sum_{i=1}^{N} P_i^{tr} = \frac{1}{N}\sum_{i=1}^{N}(RP_i^{ref} + T) = \widehat{R}\overline{P}^{ref} + \widehat{T}\end{aligned} \tag{14}$$

If we assume

$$\begin{aligned}Q_i &= P_i - \overline{P} \Rightarrow P_i = Q_i + \overline{P} \\ Q_i^{ref} &= P_i^{ref} - \overline{P}^{ref} \Rightarrow P_i^{ref} = Q_i^{ref} + \overline{P}^{ref}\end{aligned} \tag{15}$$

then Eq. (14) is equivalent to

$$\begin{aligned}\Sigma^2 &= \sum_{i=1}^{N} \|P_i - RP_i^{ref} + T\| \\ &= \sum_{i=1}^{N} \left\|(Q_i + \overline{P}) - (\widehat{R}(Q_i^{ref} + \overline{P}^{ref}) + \widehat{T})\right\| \\ &= \sum_{i=1}^{N} \left\|(Q_i + (\widehat{R}\overline{P}^{ref} + \widehat{T})) - (\widehat{R}(Q_i^{ref} + \overline{P}^{ref}) + \widehat{T})\right\| \\ &= \sum_{i=1}^{N} \left\|(Q_i - \widehat{R}Q_i^{ref})\right\|\end{aligned} \tag{16}$$



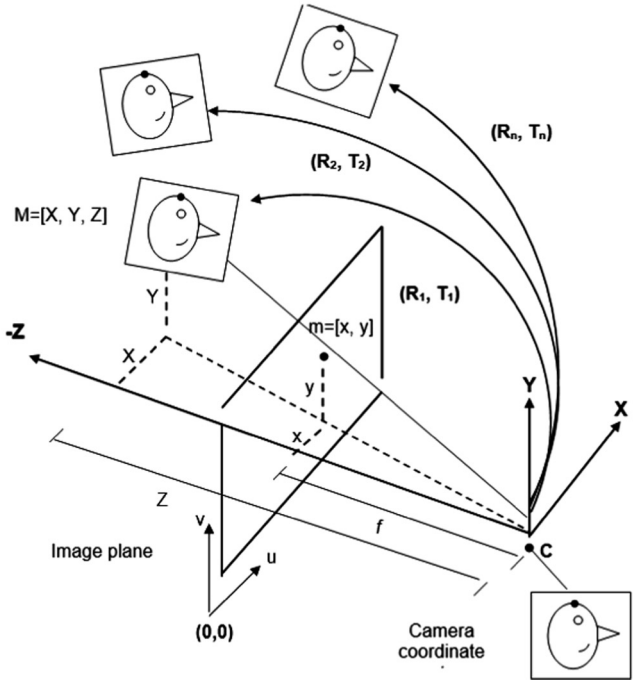


Fig. 11. Simulation part 1: simulation of the camera capture (top), and graphical illustration of random walk and corresponding 3D-to-2D mapping for each random movement (down).

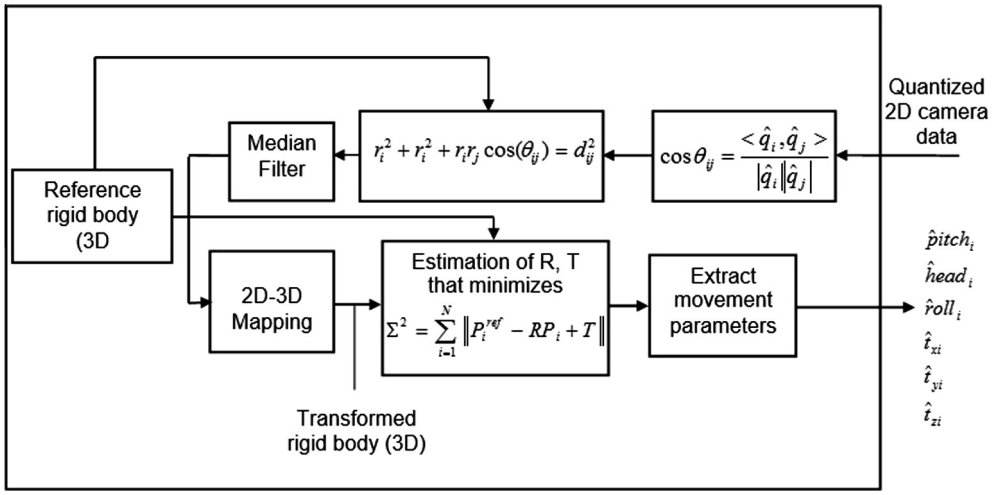


Fig. 12. Simulation part 2: evaluation of pose based on simulated capture data.

The least-square problem should be solved to find the $\hat{R}$ that minimizes Ψ^2 in Eq. (17). Then $\hat{T}$ is found through:

$$\hat{T} = \bar{P}^{tr} - \hat{R}\bar{P}^{ref} \tag{17}$$

4. Experimental results and evaluation

4.1. Simulation

We simulated a known rigid body (e.g. human head) centered at the origin rotated and translated in arbitrary directions, giving a transformed rigid body. Each pair of transformation parameters (R_i, T_i) gives a set of projections on the camera image plane defined by the camera projection equations in the pinhole camera model. These are the raw data captured by the camera. We took the camera quantization effect into account to obtain a realistic model of the camera capture. We applied a uniform quantization of level 8 to the raw data to model the quantization effect of the camera. Quantization is the primary source of inaccuracy in the measurements, and it is important to determine its effect on measurement results. As shown in Fig. 11, through this procedure, we can simulate the raw data and use our HT algorithm on simulated data. Fig. 12 illustrate the process of estimating the angles and translations from the image plane data. The quantization function is a level-8 uniform quantization that imitates the function of the camera in a simulated framework.

Fig. 13 shows the plots of estimated translations and angles. In rows Fig. 13(a) and (c), the black curves show the real data used to move the rigid body in space (phase 1 of the simulation) and the blue curves show the estimated parameters from the image plane data (phase 2 of the simulation). The green curves show the estimated parameters when the smoothing filter is applied. The bottom plots in rows (b) and (d) depict the error consisting of the differences between phase 1 and 2 of the simulation.

As can be seen from plots in row (b) and (d), the error fluctuates with small margin around zero which implies our proposed camera model presented in Fig. 11 conforms to the Wiimote camera behavior with great precision. The proposed algorithm shows a robust performance in the presence of quantization noise and is able to track the operator's translation and rotation with low tracking error. On the other hand, by observing the tracking curves and the corresponding error plots, it could be inferred that the tracking error is relatively proportional to the absolute value of coordinates and shows a deviation of less than 5% from the ideal result, which is not a significant deviation. Fig. 13(e) and (f) compare the error of the translation plots with level-1 and level-8 quantization. It should be noted that the level of error significantly decreases with level-1 quantization, which supports the hypothesis that inaccuracies are mostly related to quantization noise.

4.2. Evaluation based on real data

This section presents the physical implementation of the HT system. We used four infrared light-emitting diodes (IR-LEDs) as markers because the Wii camera functions with IR radiation and receives maximum power at 940 mm. We used LEDs (OP165W 0.5 mW 940 nm 3.1 mm) with beam angles of about 45 deg, which is relatively wide compared to conventional IR-LEDs. The LED is commercially available and inexpensive. *Wiiuse* is a library written in C (our HT software is also compiled as a library in C). The Wiiuse library offers a clean and light API that is single-threaded and non-blocking. Wiiuse can connect to several Wiimotes. It supports motion sensing, IR tracking and other features. Further, Wiiuse is an open-source library that supports Windows and Linux.

The prototype HT system uses the headphones of the operator and only requires the LEDs to be mounted on the headphone's frame without the need for additional devices like eyeglasses, which only increase the complexity of the system. We used a regular wireless headphone to include the infrared LEDs. The LEDs were connected in parallel and powered by 1.2 V AA batteries. We started building our LED pattern on a solderless border ("IR beacon"). The final unit had the LEDs mounted on the headphone. The IR-beacon has the following specifications:

1. It forms a rectangle with four diodes, each at one of its corners.
2. d is the distance vector that the algorithm would require in initializing the algorithm and is as follows:

$$d = [d12 = 127\text{ mm}, d23 = 127\text{ mm}, d34 = 129\text{ mm}, d14 = 132\text{ mm}, d13 = 27\text{ mm}, d24 = 33\text{ mm}]$$

3. The focal distance is approximated by f = 1328 [5]

The tracking trajectory is as follows:

1. The target rigid object was first moved to the left for a few mm and back to the initial position.
2. It was then translated to the right for few mm and then translated back to its initial position.

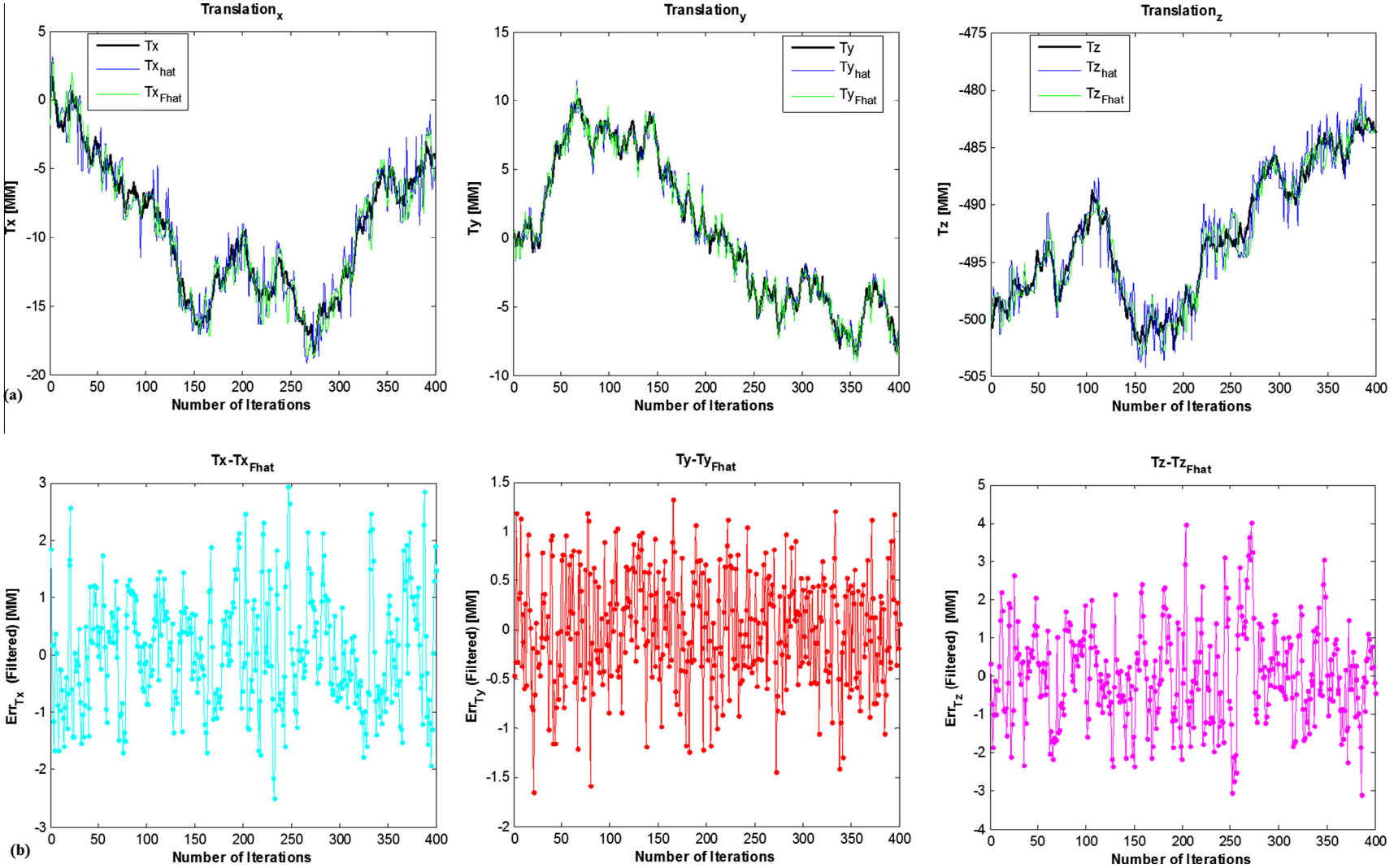


Fig. 13. (i) Top: Translation plots with level-8 quantization (the given path and estimated result) Bottom: The (translation) error plots. (ii) c: Rotation plots with level-8 quantization (the given path and estimated result) d: The (rotation) error plots. (iii) Comparison of the errors of the translation plots with level-1 quantization (e) and level-8 quantization (f).

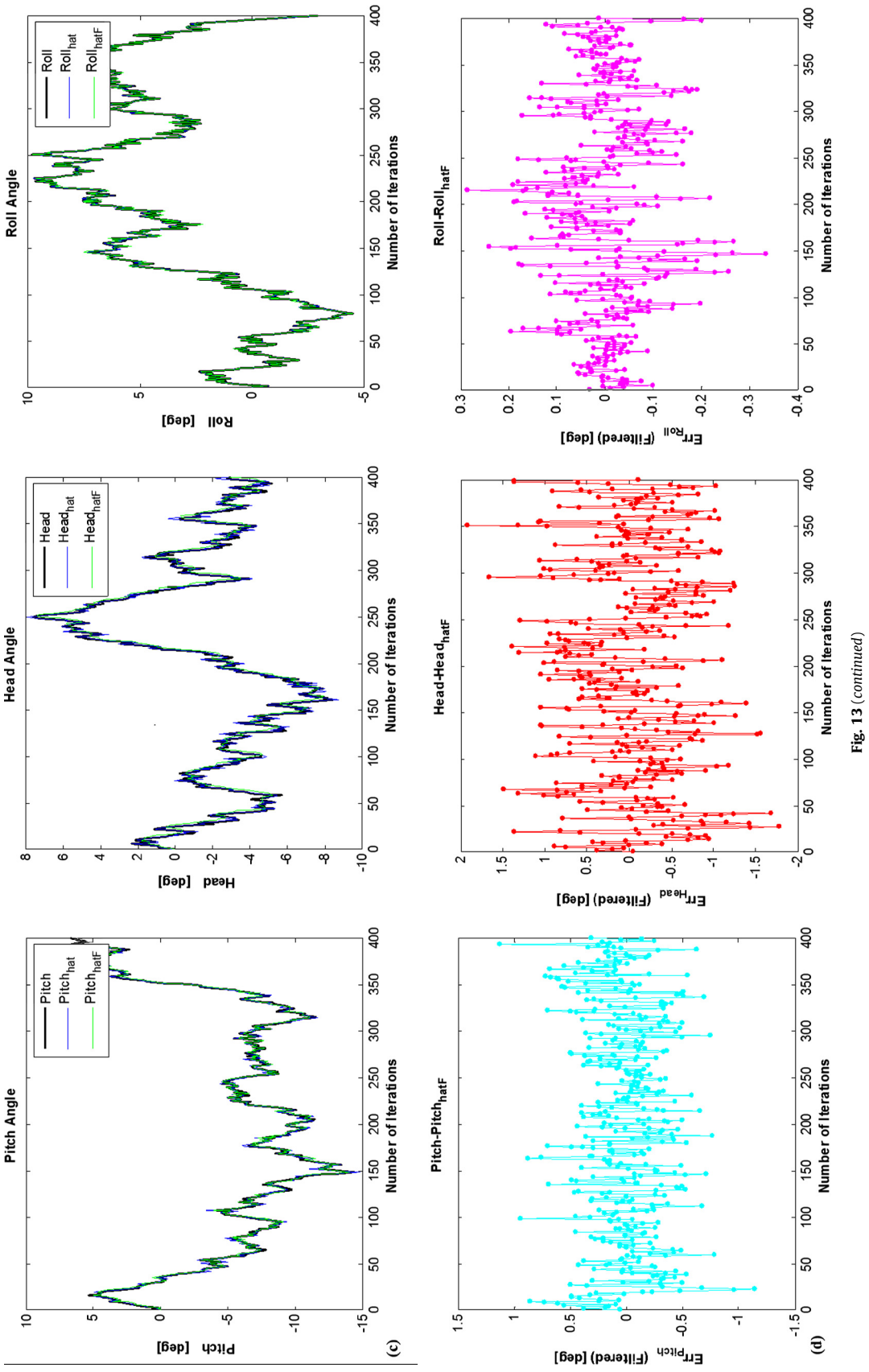


Fig. 13 (*continued*)

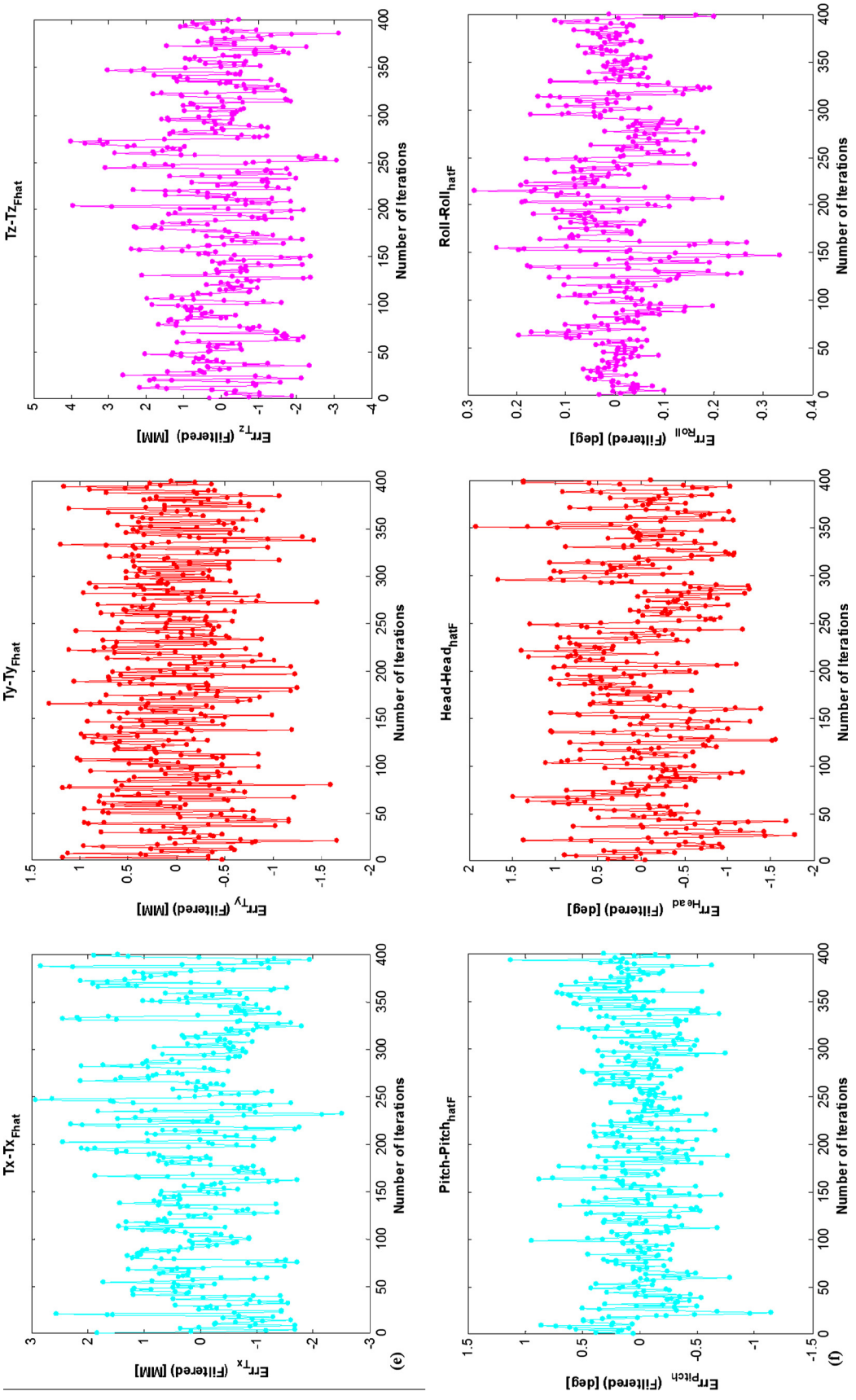


Fig. 13 (*continued*)

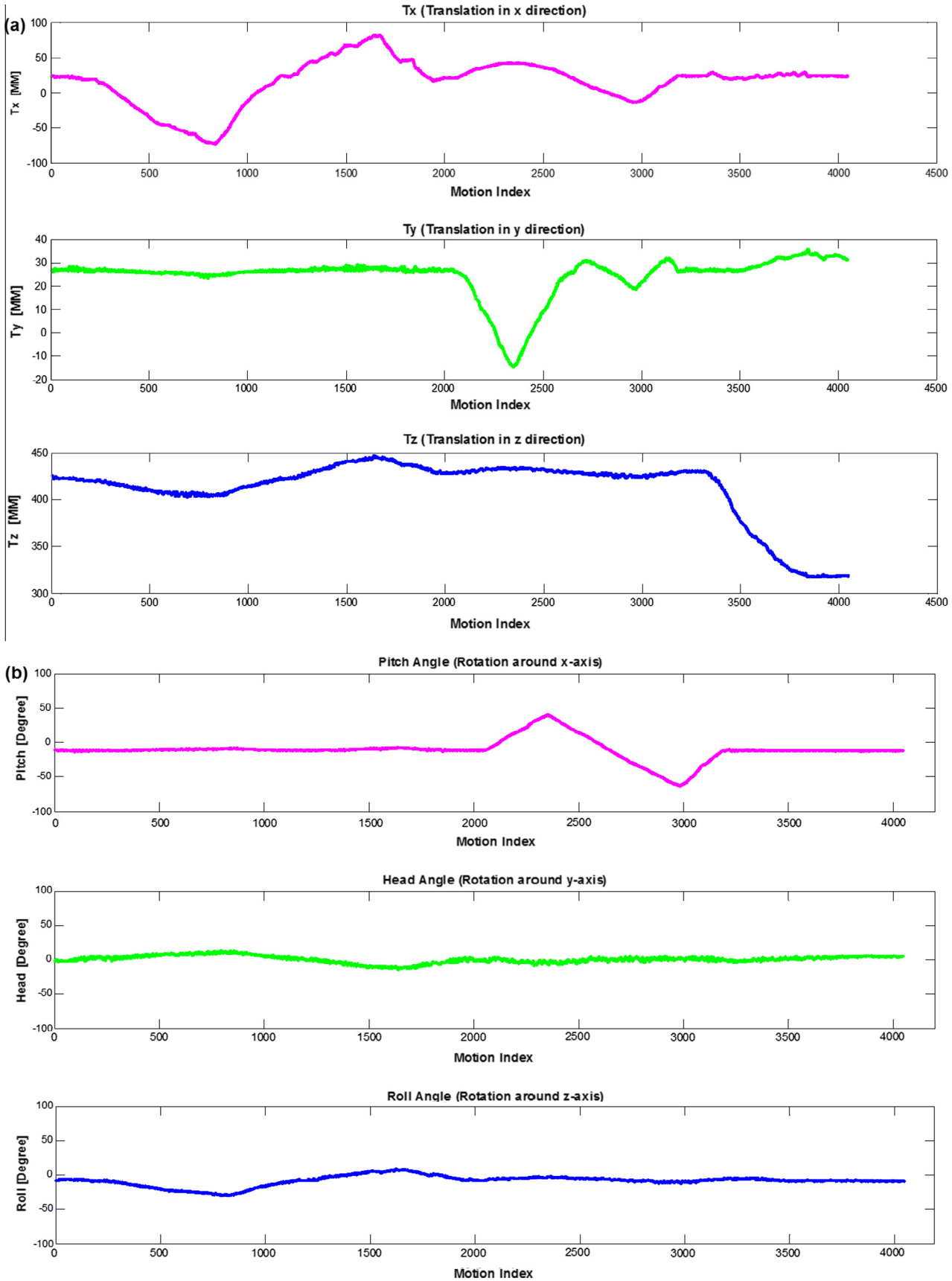


Fig. 14. Translation and rotation plots of the entire rigid body.

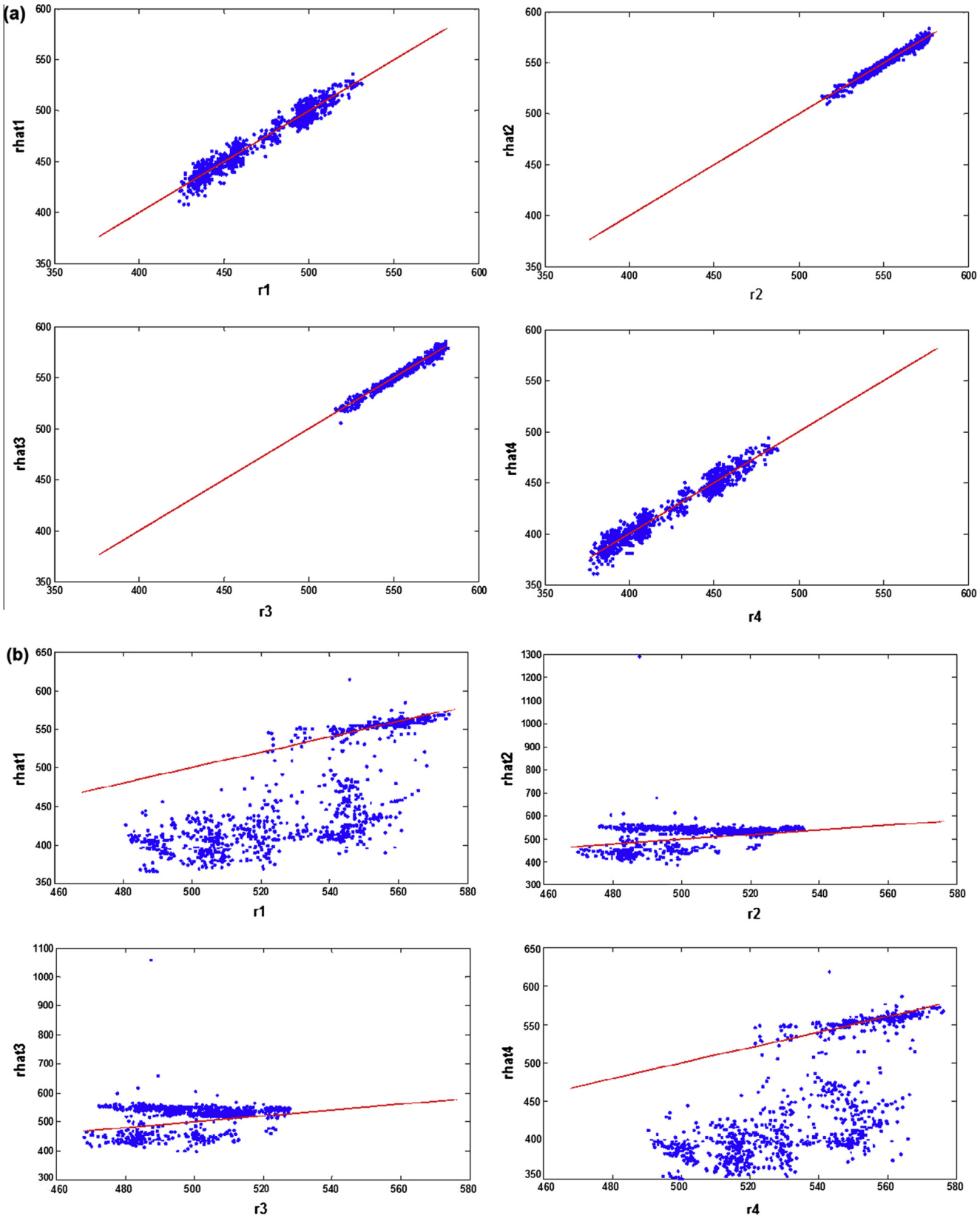


Fig. 15. Scatter plots for radius evaluation with (Block a) and without feedback (Block b) Divergence of the estimation caused by not meeting a non-coplanar arrangement (Block c).

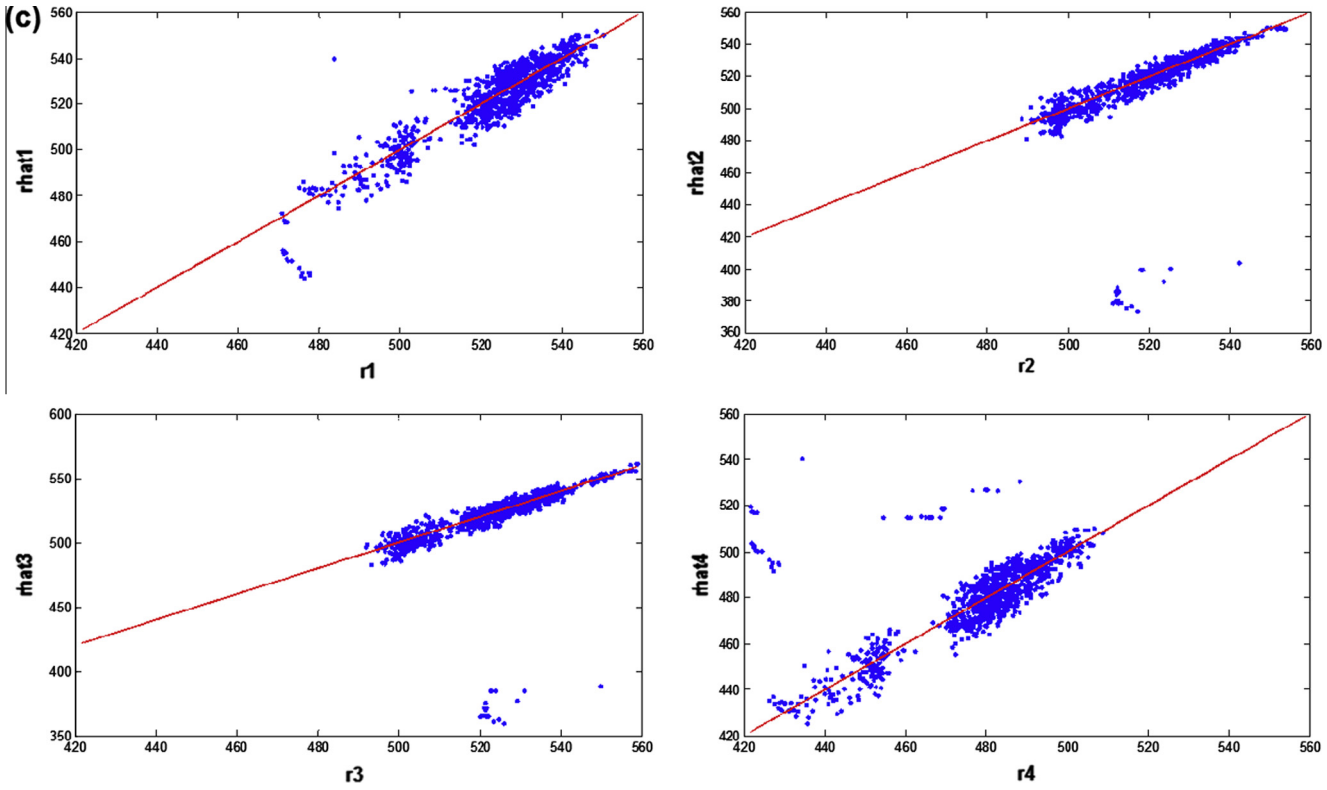


Fig. 15 (*continued*)

3. Then the rigid object was rotated around its *x*-axis (the pitch angle). The rotation was first done clockwise and then counter-clockwise back to the original state.
4. A similar translation was performed in the *z*-direction in the next stage.

The HT system detected the following movements:

1. Movement in the *x*-direction of about 80 mm in negative direction and back to the original position (from motion indices (MIs) 250–1200)
2. Movement in the *x*-direction of about 80 mm in positive direction and back to the original position (from motion indices 1200–1900)
3. Rotation around the *x*-axis (pitch) first counterclockwise for a maximum of 30 deg and back to its original position (MI 2100–2750) and then clockwise and back to the initial state (MI 2750 to 3250)
4. Movement in the *z*-direction (toward the camera) of about 150 cm

By comparing the results of real captured data with the simulation results, we can observe how the estimations follow the ground truths and respond to the head movement of the operator both in translation and orientation. We performed similar experiments for each of the pure rotations (pitch, roll and head) and translations (*x*, *y*, and *z*) and the results did not disagree with what we expected and were in agreement with the original trajectory. The results in Fig. 14 show the proposed tracking algorithm's applicability in real time.

4.3. Sensitivity analysis

We also investigated the sensitivity of the solution to the following:

1. The initial value of the tracking iterative algorithm.
2. The rigid-body configuration, more specifically the placement of the IR-LEDs on the IR plane.

To have a realistic simulation, we performed a sensitivity analysis in the presence of simulated noise. The tracking results in each iteration are quite sensitive to the initial value of the iterative solution with either the Levenberg–Marquardt or Gauss–Newton method. We found that if the evaluated results are not fed back into the algorithm as the next iteration initial guess, the algorithm has a high chance of divergence. This intuitively makes sense, because the result of tracking at two consecutive frames are very close to each other, and if the last iteration results are selected as the next iteration initial values, we are guiding the iteration algorithm correctly. Fig. 15(a) and (b) compares the difference in tracking results when the feedback exists or does not exist.

Motivated by Kreylos [5], we performed another experiment to determine whether the results are truly sensitive to the arrangement of the LEDs. The results of our experiments, indicated that the four mounted LEDs must not be coplanar. If they do, the iterative algorithm may diverge. To prove this hypothesis, we made several experiments with diodes forming coplanar or semi-coplanar patterns, and the results were quite interesting as the divergence was often noticeable. The explanation is that co-planarity of the diodes leads to a vanishing gradient, and makes the system unstable. Fig. 15(c) shows an example of divergence due to co-planarity. This phenomenon also explains why three diodes are more susceptible to divergence, because three diodes always form a plane.

With four diodes however, the co-planarity problem could be alleviated by simply mounting one of the diodes higher than the others (e.g. 3–4 cm above). Because of the camera's limited FOV of 45 deg and tracking distance of about 2 m, the algorithm may get confused and lose the target if the IR-beacon remains outside

the FOV of the camera for a significant time. Two possible solutions are recommended to tackle this problem:

1. Mount the Wiimote device on special servomotors known as Pan and Tilt servos to rotate the device so that the camera target is always in the camera's FOV.
2. With two cameras, stereovision techniques can be used to extract 3D information. In this case, triangulation determines the 3D pose of points using camera parameters. Using two Wiimote controllers and a stereovision algorithm can also remedy the camera's limited FOV and improve the robustness of the tracking algorithm.

Other sources of IR light, such as sunlight or incandescent light bulbs can cause detection problems. To remedy this limitation, we used fluorescent light, which emits little IR light. We are now able to conduct real-time tracking with the prototype we developed, which integrates the tracking algorithm and Wii device with Ericsson's 3D audio system. The Appendix gives a tutorial on using this prototype.

5. Conclusion

This paper has demonstrated the theoretical basis, design and implementation of a novel low-cost visual HT system for 3D spatial interaction in a virtual 3D audio environment based on the Nintendo Wiimote. The proposed HT system can track the full six DOFs of the operator (i.e. both translation and orientation) and deliver the result to a virtual 3D audio system for binaural rendering.

1. We have successfully developed a HT and integrated it into a 3D audio system for synchronized visual and auditory cues. Our merged auditory, visual and human interaction unit capture and hold user's attention and provide him or her with a strong feeling of immersion. Such a system can be used in multimodal applications including 3D games or 3D websites to facilitate user interaction.
2. We used two novel methods to solve the tracking problem (i.e. tracking with six DOFs using only four feature points). The P4P method was selected for implementation in the HT algorithm. During the evaluations, the algorithm was never taken to the next stage until it had received a "Pass" score from the previous stage. In particular, in the presence of quantization noise, the algorithm was tested via simulation (of a random walk), via sensitivity analysis, using real generated data and real time in integration with a virtual 3D audio system. The algorithm has demonstrated a robust behavior through various stages, and we have discovered its sensitivity to a few parameters.
3. The camera functions at a rapid refresh rate of 100 Hz. Using the Schur algorithm to solve the LS problem, convergence is accomplished in a few iterations.
4. The HT solution has been developed as a dynamic library in C. The user can easily modify the configuration data in a .txt file. The library takes the input information from the .txt file, evaluates the head pose and delivers the results to a particular HT application. The solution is therefore programmed to work with any desired configuration and can be easily coupled to other HT applications. This allows its usage in a wide spectrum of audio/video applications.
5. The proposed 6DOF HT system is a simple as no major image processing algorithms are required and inexpensive as it only requires a Wiimote.

Acknowledgments

The authors are very grateful to the multimedia department of Ericsson Research, in Stockholm, Sweden for their support of this work.

Appendix A. Tutorial on tracking with the virtual 3D audio system

A.1. Software routines

The HT software has been compiled in C as it is fast and can be used on most platforms and is easily accessible to other applications. Further, the Wii remote's driver is also written in C, making integration with the HT library easy and convenient. An external function is linked to the HT library to access the functionalities inside the library. This function contains an initialization function, *void InitPoseEst ()*, that initializes the library with the initial parameters required for head pose evaluation. This is normally done once at the beginning of the program call and the values are used during the tracking. The external function includes another function called void *GetPosePosOrient ()*, which is the main function responsible for calculating the head pose. This function evaluates the position of the rigid body as well as the front and up vector associated with 3D rigid-body movement. As soon as the new 2D data are available, *GetPosePosOrient ()* is called and then evaluates the output parameters and overrides the parameters in a data structure shared by the 3D Audio Engine. The 3D Audio engine collects the new data and updates the audio scenes associated with the listener's location in 3D space.

Initialization is done by clicking the Play button in the GUI (Fig. 18). This activates the *WiiRemoteThread*, which keeps polling the Wii remote at regular intervals, and initializes the Wiiuse library. Then the configuration file is read by the library that contains the information about camera focal length, distance between LEDs in the IR-beacon and initial radii. The parameters are used to initialize the *InitPoseEst ()* function. The next call is to connect to the Wii remote. When the connection is established, the HT application enables the IR tracking by setting the flag START IR_TRACKING, which is activated by pressing the up arrow on the Wii remote. This causes the library to call *wiiuse_set_ir ()*, which in return activates the IR tracking. To get the data from the Wiiuse library, we need a function to poll the library regularly. This is implemented by placing the polling function inside a while (1) loop which is activated/deactivated by setting START_IR_TRACK/STOP_IR_TRACK flags. This keeps polling the Wiiuse at regular intervals. The polling continues for as long as the application runs as shown in Fig. 16.

The developed program is multithreaded in which each thread performs certain tasks concurrently with the other threads. These threads may share the same memory storage. One of these threads is the *WiiRemoteThread*, which communicates with the Wii remote and collects the raw IR data from the device every 20 ms. These raw data are then interpreted as pose data, and the results are stored in a memory storage named *ListenerPose* Data. *ListenerPose* Data are global variables shared by two threads. To avoid the simultaneous use of common data, mutual exclusion (Mutex) is used. Mutex acts like a LOCK to the data memory and protects the data from being corrupted because of issues such as mutual concurrent use, or from Thread TimeOut. The key to the lock is only available to the threads sharing the same memory storage, *WiiRemoteThread* and *SceneUpdateThread* (Fig. 17).

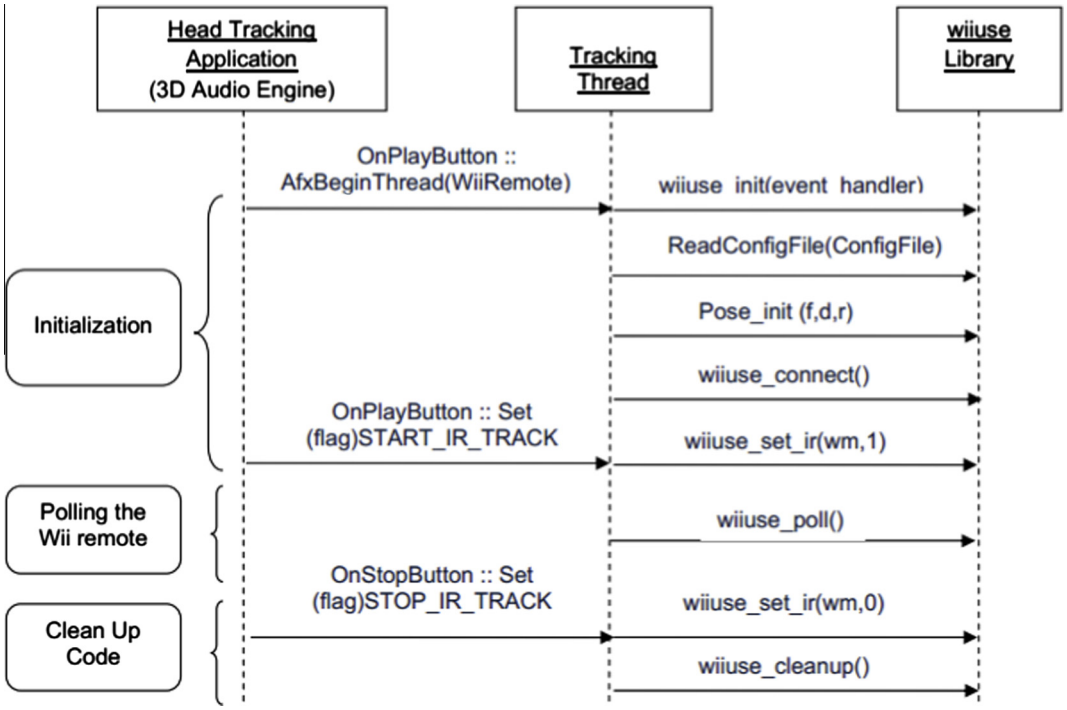


Fig. 16. Sequences of function calls for the HT application.

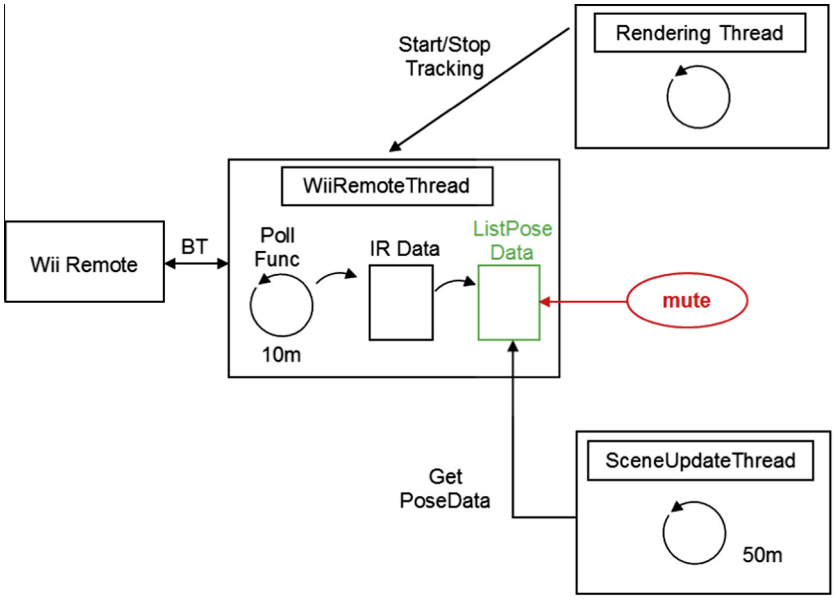


Fig. 17. Threads involved in the head-tracked 3D audio rendering engine.

A.2. Controls

Here is how to use the HT integrated with the 3D audio system as tested in Ericsson's laboratory. Nine buttons are located below the main menu. The two leftmost are the play and stop buttons, which start and stop the scene rendering. The three buttons labeled 1, 2 and 3 can enable or disable the corresponding sound source, but only when the manual scene has been chosen. For all other scenes the number of sound sources is predefined. To the right of the source buttons are a headphone button and a loudspeaker button, which define the type of output device. Headphones are chosen by default. The two rightmost buttons, 3D and Mono, can be used to toggle between mono and 3D audio rendering. The GUI also shows three different views over the simulated scene: the horizontal plane, the frontal plane and the median plane. When the manual path is chosen for a sound source, the position of the source can be changed by dragging the source with the mouse. The position of the listener can be moved irrespective of which scene has been chosen. At the bottom of the GUI, the x, y and z positions of the sources and the listener are shown.

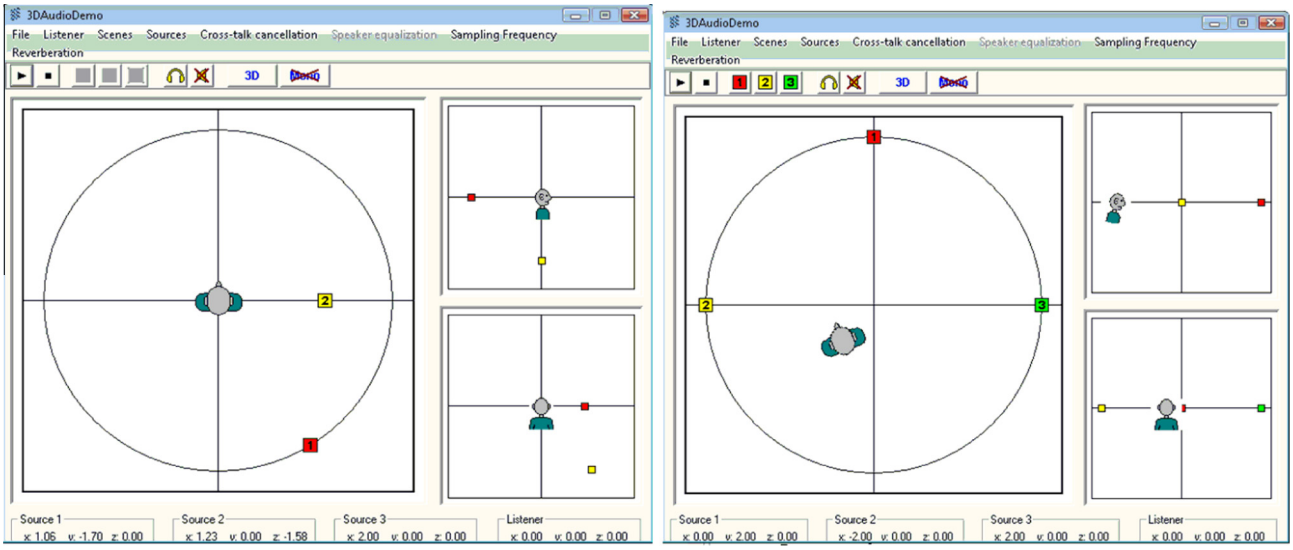


Fig. 18. Virtual 3D audio scene configured in the circle mode. Two sources are moving along the x–y and x–z circles around the listener (left); with the HT system integrated, both the listeners and the source can move freely in the 3D space (right).

For instance, in Fig. 18 left, source 1 is selected as a singing source moving along the x—y circle while source 2 is an ambulance making a siren sound while moving along x—y. The listener is at the center of the plane. The spatial audio scene for the listener is very convincing, and as soon as the operator runs the system the listener hears the scene moving around him circularly as it would be heard in reality. Without an HT mechanism, the 3D audio system assumes the listener to be fixed, and so does not change the binaural signals in a way that corresponds to the listener's movement. However, the proposed system removes this restriction, and allows the listener to move and rotate freely in 3D space (Fig. 18 right). Synchronized with the user's movement, the listener's images translate and rotate in different 2D planes in such a way that the movement corresponds fully with the listener's position and orientation in 3D space. The movements produce changes in the audio signal sent to the listener's headphone, creating user-dependent 3D audio scenes. Further details can be found in [23].

References

[1] J. Blauert, Spatial Hearing: The Psychophysics of Human Sound Localization, MIT Press, Cambridge, Massachusetts, 1999.
[2] S. Rumsey, Spatial Audio, Focal Press, Oxford, 2001.
[3] D.M. Gavrila, 3-D model-based tracking of humans in action: a multi-view approach, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 1996, pp. 73–80.
[4] H. Zhoua, H. Hu, Human motion tracking for rehabilitation—a survey, Biomed. Signal Process. Control 3 (1) (2008) 1–18.
[5] W. Zhao, R. Chellappa, P.J. Philips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.
[6] D.M. Gavrila, L.S. Davis, Towards 3-D model-based tracking and recognition of human movement: a multi-view approach, in: International Workshop on Automatic Face- and Gesture-Recognition, 1995.
[7] E. Murphy-Chutorian, M. Trivedi, Head pose estimation in computer vision: a survey, IEEE Pattern Anal. Mach. Intell. 31 (4) (2007) 607–626.
[8] J. Lee, Wii Remote projects. <interrefhttp://johnnylee.net/projects/wii/urlhttp://johnnylee.net/projects/wii/>. [Accessed 7 January 2014].
[9] O. Kreylos, Wiimote Hacking, <http://graphics.cs.ucdavis.edu/~okreylos/ResDev/Wiimote/index.html> [Accessed 7 January 2014].
[10] W.G. Gardner, 3D Audio and Acoustic Environment Modeling, Wave Arts, Arlington, 1999.
[11] B. Tas, N. Altiparmak, A.S. Tosun, Low cost indoor location management system using infrared Leds and Wii Remote controller, in: International Conference on Virtual Rehabilitation, 2009, pp. 111–116.
[12] W. Zhu, A.M. Vader, A. Chadda, M.C. Leu, X.F. Liu, J.B. Vance, Wii Remote-based low-cost motion capture for automated assembly simulation, Virt. Real. 17 (2) (2013) 125–136.
[13] C. McMurrough, Jonathan Rich, Vangelis Metsis, An Nguyen, Fillia Makedon, Low-cost head position tracking for gaze point estimation, in: International Conference on Pervasive Technologies Related to Assistive Environments, 2012.
[14] J. Synnott, WiiPD—objective home assessment of Parkinson's disease using the Nintendo Wii remote, IEEE Trans. Inform. Technol. Biomed. 16 (6) (2012) 1304–1312.
[15] I. Aranyanak, R.G. Reilly, A system for tracking braille readers using a Wii Remote and a refreshable braille display, Behav. Res. Methods 45 (1) (2013) 216–228.
[16] S. Hay, J. Newman, R. Harle, Optical tracking using commodity hardware, in: International Symposium on Mixed and Augmented Reality, 2008, pp. 159–160.
[17] M. Ubila, D. Mery, R.F. Cadiez, Head tracking for 3D audio using the Nintendo Wiimote, in: International Computer Music Conference, San Francisco, 2010, pp. 494–497.
[18] Y. Matsumoto, N. Sasao, T. Suenaga, T. Ogasawara, 3D model-based 6-DOF head tracking by a single camera for human–robot interaction, in: IEEE International Conference on Robotics and Automation, 2009, pp. 3194–3199.
[19] G. Medioni, S. Kang, Emerging Topics in Computer Vision, Prentice Hall, New Jersey, 2004.
[20] E. Trucco, A. Verri, Introductory Techniques for 3D Computer Vision, Prentice Hall, New Jersey, 1998.
[21] T. Nöll, A. Pagani, D. Stricker, Markerless camera pose estimation – an overview, in: Visualization of Large and Unstructured Data Sets. IRTG Workshop, Dagstuhl Publishing, Germany, 2010.
[22] T.S. Huang, S.D. Blostein, E.A. Margerum, Least-squares estimation of motion parameters from 3D point correspondence, in: IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, Florida, USA, 1986, pp. 24–26.
[23] Y. Deldjoo, Wii Remote Based Head Tracking in 3D Audio Rendering, Master's thesis, Chalmers University of Technology, 2009.
[24] Modroño, Cristián, Antonio F. Rodríguez-Hernández, Francisco Marcano, Gorka Navarrete, Enrique Burunat, Marta Ferrer, Raquel Monserrat, José L. González-Mora, A low cost fMRI-compatible tracking system using the Nintendo Wii remote, J. Neurosci. Methods 202 (2) (2011) 173–181.
[25] Jongshin Kim, Kyoung Won Nam, Ik Gyu Jang, Hee Kyung Yang, Kwang Gi Kim, Jeong-Min Hwang, Nintendo Wii remote controllers for head posture measurement: accuracy, validity, and reliability of the infrared optical head tracker, Invest. Ophthalmol. Vis. Sci. 53 (3) (2012) 1388–1396.
[26] Ryan A. Pavlik, Judy M. Vance, A modular implementation of Wii remote head tracking for virtual reality, in: ASME 2010 World Conference on Innovative Virtual Reality, American Society of Mechanical Engineers, 2010, pp. 351–359.
[27] L. Bharath, S. Shashank, V.S. Nageli, Sangeeta Shrivastava, S. Rakshit, Tracking method for human computer interaction using Wii remote, in: 2010 International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT), IEEE, 2010, pp. 133–137.
[28] Stefano De Amici, Andrea Sanna, Fabrizio Lamberti, Barbara Pralio, A Wii remote-based infrared-optical tracking system, Entertain. Comput. 1 (3) (2010) 119–124.
[29] Ranil Sonnadara, Neil Rittenhouse, Ajmal Khan, Alex Mihailidis, Gregory Drozdzal, Oleg Safir, Shuk On Leung, A novel multimodal platform for assessing surgical technical skills, Am. J. Surg. 203 (1) (2012) 32–36.
[30] Ross C. Williams, Finger tracking and gesture interfacing using the Nintendo® wiimote, in: Proceedings of the 48th Annual Southeast Regional Conference, ACM, 2010, p. 11.