2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

A Portable and Low-Cost 3D Tracking System Using Four-Point Planar Square Calibration

Pekka Peltola, Miika Valtonen and Jukka Vanhala
Department of Electronics
Tampere University of Technology
Tampere, Finland
{pekka.peltola, miika.valtonen, jukka.vanhala}@tut.fi

***Abstract*—A practical indoor tracking system should be accurate, affordable, portable and easy-to-use. However, most of the current tracking applications lack in at least one of these properties. To fill this gap, we present such a 3D tracking system. Built around two to four wireless Nintendo Wii Remotes, the system can simultaneously track up to four infrared LEDs carried by multiple users. The set up and calibration takes only a few minutes. A semiautomatic calibration method is used, where the user places the infrared LED marker sequentially on the four corners of a freely defined square in the tracking area. Within a cube with an edge length of 2.5 m, the system was found to be able to track both a static and a moving target with an accuracy of 3 cm. In addition to discussing the developed system, this paper presents computational methods for improving the positioning accuracy by altering the assumed optical centers of the Wii Remote cameras, which vary due to intrinsic differences in them.**

Keywords-Wii Remote; indoor positioning; tracking; Four-point calibration; infrared

I. INTRODUCTION

Our modern daily life consists increasingly of interaction between humans and computers. In many applications, user tracking offers new opportunities in location based services. The need for tracking is especially evident in applications of virtual reality. Additionally, in art performances, exhibitions and games this information deepens the feeling of immersion. In future devices, the new and more agile interfaces will replace the older and cumbersome. The Wii Remote, or Wiimote as it is often called, is a recent example of this trend. Though, not all the possibilities that it offers have been utilized yet. Many research groups and individuals have been creating new alternative uses for the Wiimote. The interactive whiteboard by Johnny Chung Lee is a good example [1].

Diverse spectrum of different techniques have been used in the existing indoor positioning applications. The ultrasound and the capacitive positioning techniques often require more manual configuration and calibration than the optical tracking systems. Expensive and exact timing components are required in radio-frequency applications in calculations of e.g. the relation between received signal strength and distance. In these systems the tracked tag needs to bear higher processing capabilities, which results in reduced usability.

In this paper we present a cheap and easily movable tracking system. Using two to four Wiimotes we can detect the 3D positions of one or more simple tracking objects consisting of one to four infrared light emitting diodes (IR LEDs). The positioning application runs on a regular laptop.

Connected to a laptop, the minimal setup consists of two Wii Remotes, a Bluetooth adapter and a controller or a tag which holds the tracked IR LED. The cost of this installation for home use would then be approximately 300$ excluding the price of the laptop. By replacing the Wiimotes with new infrared camera modules the price would be even lower, since the Wiimote has features that are not utilized. For a larger scale scenario, for example in an office space with multiple rooms, the price would follow the number of the installed camera modules.

A Wiimote provides 2D location data of one to four of the brightest infrared dots that are inside the field of view. In order to acquire three dimensional positioning data of the IR LED, a stereo vision system is required. The first step in setting up a stereo vision system is the camera calibration. Within the calibration and measurement stages there are many variables that can contribute to the total positioning error. However, a Wiimote is still a more precise tracker in comparison to other techniques.

We implemented the positioning system for a tracking space of 2.5×2.5×2.5 meters. The positioning error is analyzed within this cubical space. Conducted positioning test measurements are examined and the latency in our set-up is measured. The positioning error inside the cube is compensated by the represented principal point correction method and by 3D position data mapping.

In the next chapter the related research is examined. The third chapter discusses the characteristics of a multi camera vision system. Our system is presented in the fourth chapter. The fifth chapter describes the evaluation methods of the designed tracking system. The results and discussion chapter assesses the properties of the developed system. Finally, the conclusions chapter sums up the research.

978-1-4673-1954-6/12/$31.00 ©2012 IEEE

II. Related Work

The ultrasound based DOLPHIN tracking system, presented in [2], had an accuracy of 15 cm. In order to reach the precision level of optical systems more research is needed. This applies also within the capacitive systems.

In [3], the stance and the height of the tracked person were extracted by measuring the capacitance change, with an accuracy of 9.1 cm. But if the height calculation is required to be more accurate than 10 cm, the capacitive system does not yet serve the purpose.

Although the capacitive, RF and ultrasound techniques are able to track an object behind obstacles and do not necessarily need a line of sight, however, reducing the reflections and noise requires more research to achieve better precision. These technologies still lack in reliability in comparison with the optical systems.

Kinect requires a great deal of image processing resources. The Wiimote is thus the computationally lighter option among the commercial optical tracking devices.

The Wiimote camera based applications can be divided into two groups according to the technique used with the target. The first, an active technique, uses an IR LED as a part of the object being tracked. In the second option, the object has a passive marker that reflects the infrared light coming from a separate source. For example, in [4], the IR LEDs were attached to the spectacles in a head tracking application and into a pen in another, drawing application. As in [5], a 16 W infrared illuminator was used with reflective markers. [1, 4-11]

The company, Natural Point [6], has a commercial solution for headtracking up to 1.5 meters, which enables an additional control dimension e.g. for 3D shooting games. The product prices are around 150$. As well, Advanced Realtime Tracking GmbH provides products with capabilities of infrared tracking reaching up to 6 meters [7].

In [1], Johnny Chung Lee presents his Wiimote applications, including headtracking, of which the interactive whiteboard appears to be the most promising. Lee also mentions the 3D motion tracking as a possible future project. Our system implements this idea; and in addition the developed system can identify separate IR LEDs and recognize gestures performed with an IR LED. [1]

Other similar applications include a musical instrument controller or a dance move recognizer, brought up in [8]. A rehabilitation system fixed to a table is shown in [9]. This system and the system that follows a robotic arm in [10], utilize two Wiimotes, enabling a stereo vision tracking scheme.

An example of a hybrid stereo vision tracking system was designed in [11]. Here the Wiimote and a conventional video camera make up the stereo vision system. The position of an unmanned aerial vehicle (UAV) was tracked. The 2D tracking data of the IR LEDs on the UAV is combined with the 2D position data of white markers.

III. Tracking Characteristics

Since the release of the Wii in 2006, the unpublished features of the Wii Remote have been reverse engineered by many. Wiimote has a camera from PixArt with an infrared pass filter. The PixArt camera resolution is 128×96 pixels and with interpolation the Wiimote gives the IR LED 2D position data with a resolution of 1024×768; The refresh rate being 100 Hz. [1, 8]

The overall positioning error in a stereo vision system is a cumulative result of the multiple steps performed on the data. The internal camera properties affect the error accumulation as well. The finite camera resolution defines one of these limits.

In [12], the range error is mentioned to dominate over the horizontal or the vertical error in a stereo vision system. This can be compensated by proper positioning of the two cameras. In [13], three error sources are listed. First are considered the errors in the world coordinates of the calibration points. This corresponds to the manual IR LED placement errors in our calibration square. The second is the error in image coordinates during the calibration phase. The third is the error in image coordinates during the measurement stage, adding up to the overall positioning error. [13]

The combined error, or the space inside which the measured position is most likely located, can be depicted as a polyhedron of uncertainty [13]. The probability of the actual position can be thought to be uniform within this volume [12]. The polyhedron is formed within the given limits and uncertainties of the error sources. In [14], the errors are also divided into three categories. These are the errors in the camera model, in image processing and the camera calibration errors. These relate to the error listing made in [13].

In stereo vision systems the calibration phase of the cameras is an important step that will define the tracking capabilities of the system. Zhang describes an easy way to calibrate a camera by using a plane with a recognizable pattern on it [15]. Using corner detection on the known planar pattern, the position of the camera can be extracted from the image. By using multiple images of the same planar pattern in different orientations, the position, orientation and the intrinsic features, like the radial distortion of the camera, can be derived more accurately. These methods can be found in the Matlab Camera Calibration [16] and the Computer Vision Toolbox. The derived camera intrinsic properties matrix used in the virtual model of the tracking set-up tries to reproduce the intrinsic properties of the real Wiimote camera. Every Wiimote has individual deviation in these intrinsic properties. [17, 18, 19]

In [20], lighting is mentioned for the edge detection in the image processing step. In general, the light sources and the reflection conditions have to be taken into account when designing a stereo vision system for a particular space. For the Wiimote, the infrared light sources and reflecting surfaces have to be considered. The environment in our set-up did not have reflecting surfaces. A proper filtering step has to be added to the system in order to discern the wanted light sources.

IV. TRACKING SYSTEM

One of the goals that were set in developing our positioning system was the ease of use. This means a minimal time spent with setting the tracking system up and running. In addition, we did not want to attach the Wiimotes onto any fixed stands.

The camera placement defines, in part, the positioning accuracy of the system. In terms of [21], the camera placement is a geometric design problem, where smallest reconstruction error of the system is searched by altering the spatial position and orientation of a finite set of cameras. The goal is to find an effective solution for helping the expert and non-expert alike in camera placement. The camera installation, tracking object and environment spatial variables can be simulated with software like EPOCA. EPOCA utilizes genetic algorithms. [21, 22]

Depending on the camera type and under different illumination conditions, the desired features can be recognized in the images. The tracked object shape affects the tracking precision, resulting in that the optimal placement of the cameras for a certain object may not work for different objects. In general, the larger convergence (genetic or angular) between two cameras recording the same feature helps in getting a more accurate positioning result of the feature.

In [22], the constraints for the optimal camera placement are discussed. Visibility in the tracking area is perhaps the first thing to consider. The amount of obstructions in the field of view should be minimized. The optimal number of the cameras depends on the space where the system will be implemented. The more there are cameras for a certain measured feature the more there is data and redundancy to be used for achieving more accurate positioning results. In many home surveillance systems budget dictates the number of cameras.

The convergence angle for spherical targets and cameras in an imaging system is recommended to be more than 20 – 30 degrees for better resolution [22]. We placed the cameras on the corners of a cube with a mutual angle of 90 degrees between the Wiimotes. Using these angles the depth in one Wiimote corresponds to a horizontal coordinate in adjacent Wiimotes. The orientation of the Wiimotes was considered also. We tracked a simple IR LED that was carried by the user. The placing of the cameras close to the ceiling minimizes the occlusion of the IR LED by the user or other objects within the room. The Wiimotes face the center of the cube so that the cube corners fit inside the field of views. This arrangement is favored by photogrammetrists. [21, 22]

Our tracking system consists of a laptop that is used with two to four Wiimotes to track one to four infrared light emitting diodes. The horizontal field of view angle for a Wiimote is approximately 41 degrees and the vertical is 31 degrees [18]. We prepared a measurement space of 2.5×2.5×2.5 meters. To cover this space the Wiimotes were placed 4.5 meters away from the edges of the cube and at a height of 2.0 m. The top to down orthographic view of the measurement space is shown in the Fig. 1. The origin is at the corner of the cube on the floor level.

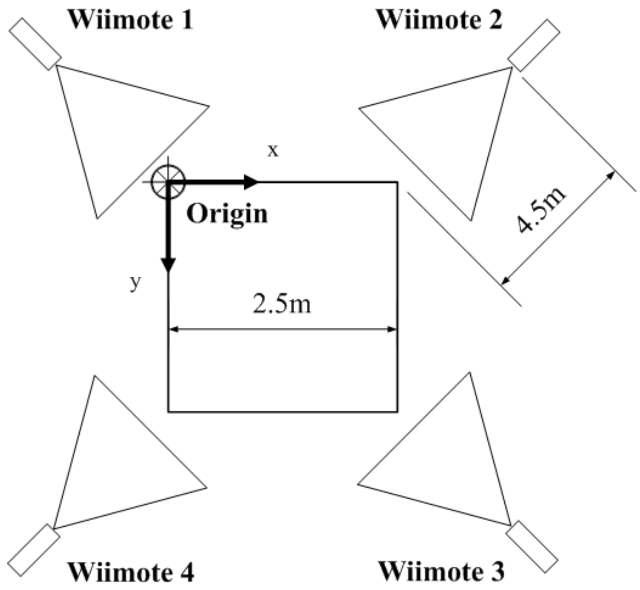

Figure 1. The top to down orthographic view depicts the cubical measurement space of 2.5×2.5×2.5 meters. The origin and the Wiimote positions, horizontal field of views and the image planes are shown. The positive z-axis points into the paper (into the floor).

A. System Architecture

The Wii Remotes are connected to a laptop using Bluetooth. USB Bluetooth adapter from MSI and the Wiimotes are compliant with the Windows 7 drivers. The laptop used was an ASUS X70I. The data acquisition from the Wii remotes was implemented with the wiiuse API [23].

The tracking prototype testing program was developed using Qt Creator. OpenGL was used to visualize the real time tracking and to construct the virtual model of the measurement space. The program has three main components – the connection, the calibration and the measurement stages. Matlab communicates with our program in the calibration phase.

B. System Procedures

The connection is the first task involving a check-up that the IR LED is seen by the Wiimotes. An adequate field of view over the whole intended tracking area can be ensured at this stage.

The second task is the calibration. The pinhole camera projection matrix is

$$P = K\,[\,R \mid t\,]. \tag{1}$$

The matrix K consists of the camera intrinsic parameters. These are the pixel focal lengths in terms of pixel dimensions, the principal point (both with x and y components) and the skew coefficients. The extrinsic features R and t define the rotation and the translation of the camera. [24]

A ray starting from a 3D point inside the field of view can be thought to travel through the image plane to the camera centre in the pinhole camera model. The point on the 2D image plane is a perspective projection of the 3D point. [24]

Our program gathers first the 2D data of the 4 calibration points from the Wiimotes. Then the data is sent to a Matlab session. The initial intrinsic properties matrix

$$K = \begin{bmatrix} 1306.3 & 0 & 535.0 \\ 0 & 1302.3 & 401.7 \\ 0 & 0 & 1 \end{bmatrix}, \qquad (2)$$

was taken from [17] and [18]. The properties, x – component of the focal length (1306.3) and the y – component (1302.3), the x – component of the principal point (535.0) and the y – component (401.7) can be altered in the program. This intrinsic matrix is also sent to the Matlab session. The extrinsic data R and t is then calculated in the Matlab session and sent back to the program. We modified the Matlab code for the extraction of the extrinsic camera parameters for our system. It minimizes the projection error using the Levenberg-Marquardt algorithm. [16, 24]

The calibration method uses a planar square target and can be called semiautomatic, because of the active involvement of the user in the process. The square is defined by moving the IR LED sequentially to all of the four corners on the floor. The direction of the z-axis depends on the sequential corner recording order. We placed the IR LED clockwise to the four corners on the floor, which results in the positive z axis to point downwards (into the paper in Fig. 1). For the measurements the origin is marked in the Fig. 1 which is also the first recorded IR LED position.

The side length of the calibration square can be adjusted by defining it in the program (here 2.5 m). The calibration square image on the Wiimote image plane is a perspective projection. In [19], multiple stereo vision image pairs for the Matlab Camera Calibration Toolbox were used in the calibration phase of the cameras. Our extrinsic features calibration uses only one image of the planar square calibration pattern. It is easy to implement and takes little time to measure in the room-size-cube. The initial intrinsic matrix is defined in our program and can be changed. The skew and the distortion coefficients are omitted in our measurements.

The third stage is the measurement stage, where the pin-hole camera model is used. The tracking algorithm first constructs rays using the calibration results and the recorded 2D IR LED positions. Then it calculates the closest point between the constructed rays.

The calibrated cameras are placed into the OpenGL virtual model. Image planes are placed according to the camera attitudes and are shown in the virtual model. The tracked IR LED produces a perspective projection points on each image plane of the Wiimotes. Rays are constructed with a starting point at the calculated Wiimote center points. The rays go through the points on the individual Wiimote image planes.

The measured IR LED position is taken to be the average closest point between the rays from all Wiimotes. In an ideal situation the rays would intersect at one exact point. Normally, this does not happen, instead they are separated according to the errors accumulated from the calibration step and by the errors in the measurement stage. Additionally, the errors in the virtual model add up in the total positioning error. For example, the horizontal field of view angle is not exactly 41 degrees as it is supposed in the virtual model.

Within other variables such as the defined square side length, an error is accumulated throughout the stages. Especially the manual placement error of the IR LEDs during the calibration stage adds into the total positioning error. The errors in different phases of the tracking process are discussed also for example in [14].

The measurement results are written into a file that can be opened and analyzed in Matlab afterwards.

C. Principal Point Correction

A simple method was implemented and tested that moves the predefined principal points. After the first calibration step, the position of the origin is measured. The distance error between the ideal origin in the virtual model and the measured origin can be projected individually on each Wiimote image plane. The individual principal points are then moved in the direction of this difference vector by one fourth of its length on the image plane. Then each camera is recalibrated with the new principal points. This process is iterated until the difference of the virtual model and the measured origin is within the predefined limits.

We used 14 recalibration rounds in all our principal point correction measurements. The Fig. 2 shows the difference vector components in the xy –plane for two Wiimotes.

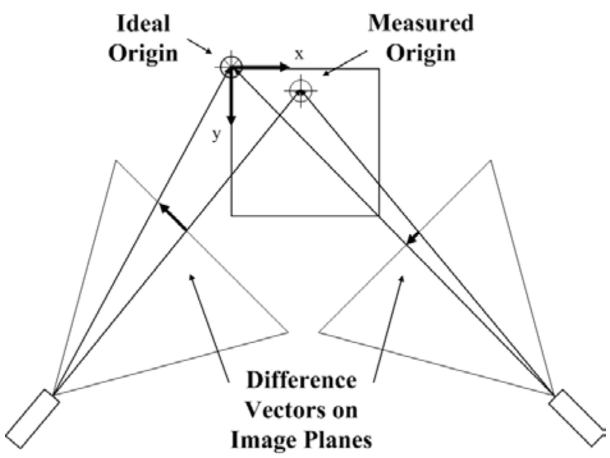

Figure 2. The principal point correction. The tracking system measures the origin. The difference vector between the ideal origin and the measured origin is projected onto individual Wiimote image planes. The principal points are then moved one fourth of the length of this resulting difference vector on the image plane towards this difference vector direction.

V. Evaluation Approach

The tracking system was evaluated with the tests described in this chapter. The results are discussed in the next chapter.

We assembled a calibration grid with 16 IR LEDs shown in the Fig.3. Fig.4 shows the more powerful infrared LED, a 3.1 W OSTAR IR LED with a spectral emission peak at 940 nm. The half power irradiation angle of this IR LED is 70 degrees. A small copper cone reflection plate was placed on top of the OSTAR IR LED, which reflects the infrared light sideways (Fig. 4).

A. *Distance, Latency and Identification*

First we measured the detection distance of the OSTAR IR LED with one Wiimote. The Wiimote was kept in a small constant circular motion and was taken further away from the IR LED. When the measured position did not change within the defined time limit, the distance was measured and was taken as the detection distance. At this distance the irradiance was no longer powerful enough. The time limit criterion for detection was 0.5 seconds. The detection distance depends on the irradiance of the IR LED and thus also on the current that was driven through the IR LED.

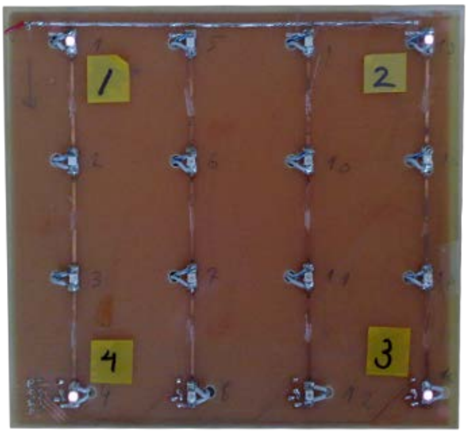

Figure 3. The calibration board with 16 infrared LEDs forming nine smaller squares. The test grid is used with the Matlab Camera Calibration Toolbox and was also used in the identification measurement. The IR LED with the number 1 is the origin. The x axis will be between IR LEDs 1 and 2, when the sequential calibration is made clockwise according to the IR LED numbering.

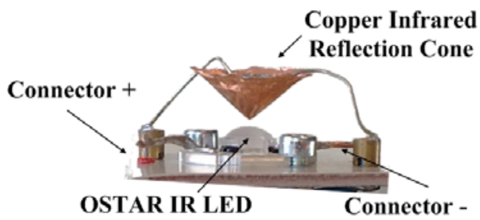

Figure 4. The OSTAR IR LED with the copper reflection cone on top. The evaluation measurements in the cube space of 2.5×2.5×2.5 meters were measured using this IR LED. The cone is attached with thin wires in order to block the infrared irradiation as little as possible.

The latency was measured in the second measurement session. Here the tracking system was set up for a smaller space of 1.0×1.0×0.5m and the OSTAR IR LED was tracked. An infrared laps counter was placed at the positive x axis. The OSTAR IR LED was put into a constant circular motion in the xy-plane with a radius of 40cm. The centre of the circle was at the origin. The laps counter interrupted the positioning system program every round and the interrupt time and Wiimote positioning coordinates were recorded into a file. The positioning program process was a high priority process in Windows 7.

The same setup was used for the IR LED identification testing. In the third measurement four LEDs were blinking on the calibration grid board. The first IR LED was on 90% of the time, the second IR LED was on 60% of the time, the third 40% and the fourth 20% of the period. The four IR LEDs were blinking synchronously so that the first IR LED was put on first, then the second, the third and the fourth. The IR LEDs were turned off in the reverse order. Already with three points we can define the orientation of the object on which the IR LEDs are blinking.

B. *System Evaluation Using Two and Four Wiimotes*

We performed altogether 20 measurement sessions; 10 sessions using two Wiimotes and 10 sessions using four Wiimotes. The two Wiimotes were located at the Wiimote positions 1 and 2 in the Fig. 1 for the cases when measuring with two Wiimotes. In all of the 20 measurement sessions, altogether 10 points within the cube space were measured. These were all the corners of the cube and in addition the centre point on the floor at (1.25m, 1.25m, 0m) and a point inside the cube at (1.25m, 1.25m, -1.50m). We assess the maximum manual placement error to be 1 cm for the measurement points on the floor and 3 cm above the floor. This error is the manual placement error of the IR LED to the desired measurement point within the cube.

The measurements consisted of five cases, with each case consisting of two measurement sessions. In the first case that we denote with C1, the intrinsic matrix (2), presented earlier, was used. The principal point correction was not applied. In the second case, C2, an intrinsic camera matrix acquired by the Matlab Camera Calibration Toolbox was used for each Wiimote. This matrix for each Wiimote was calculated from 12 images of the calibration grid taken with the Wiimote. The principal point correction was not used in this case either.

In the cases C3, C4 and C5 the principal point correction method was used. In the case C3, the initial matrix was the same matrix as in the case C1. Likewise the initial matrix in C4 was the same that was used in the case C2. For the case C5 the principal point was set to (512, 384). The focal lengths used in case C5 were the same as with the cases C2 and C4.

In each measurement session, three circles were measured with centre points at (1.0m, 1.0m, -0.23m), (1.5m, 1.5m, -0.23m) and (1.25m, 1.25m, -0.95m). The OSTAR IR LED traveled the circular track with constant velocity having a radius of 75 cm. The error of this dynamic test of the tracking system is brought up in the next chapter.

VI. RESULTS AND DISCUSSION

The most accurate measurement had a positioning error below 3 cm. Without the principal point correction the measured cube was shifted in space. The cube retained its form in the measurements, although slight deformations could be noticed.

A. Distance, Latency and Identification Results

The detection distance measurement results are shown in the Fig. 5. This distance is shown by the current flowing through the IR LED. Based on these results we chose to feed 560 mA through the OSTAR IR LED in the tracking system evaluation measurements. The 10 meters range was sufficient to cover even the furthest corner of the cube.

The latency measurement result is shown in the Fig. 6. The red circle is the fitted track that the IR LED traveled along. The blue crosses mark the spots where the measured values of the Wiimote tracking system were at the instant when the IR LED crossed the positive x axis. We can define the time between the actual crossing of the x axis and the crossing in the virtual model from the angles. When the cycle time was 1.4883 s, we get the latency for our system to be approximately between 280 – 320 ms, which is relatively high. The replacement of the Bluetooth connection with wired connections would shorten the latency, although this is not desired of a portable system. Changing and comparing the fastest Bluetooth adapter and software environment are the most obvious next steps. By using only two Wiimotes the connection could also function faster. This will be left for future research.

The circle measurement values of the cases for both two and four Wiimotes comply with the laps counter values that are seen in the Fig. 6. We assessed the manual placement error of the circle track to be also 3 cm. And indeed, the errors in the dynamic circle tests were very close to the errors acquired with the static tests of the cube measurement points.

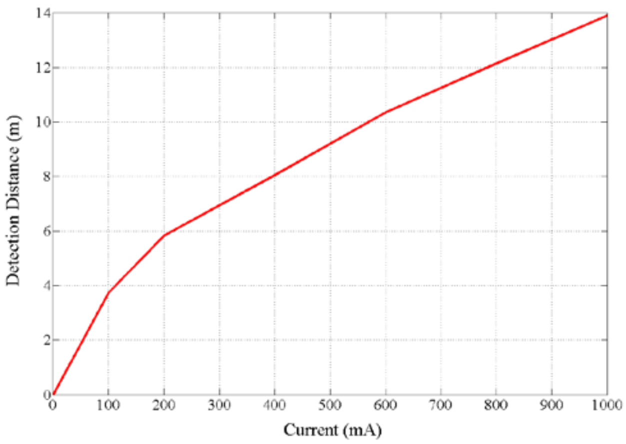

Figure 5. The detection distance was measured using the OSTAR IR LED. A copper cone on top of the IR LED reflected the infrared light sideways. Wiimote was in constant small circular motion and was taken further away from the IR LED. The infrared irradiation detected by the Wiimote is proportional to the current driven through the IR LED. The detection distance is presented in relation to this current.

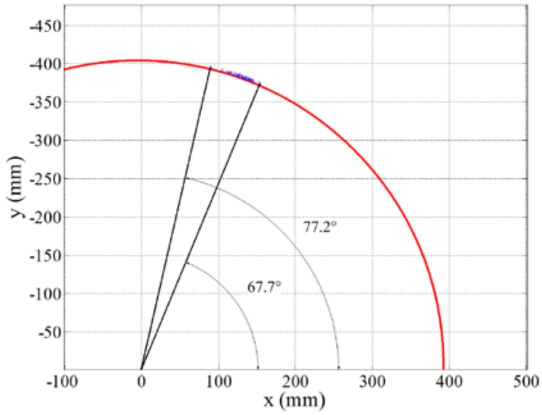

Figure 6. In the latency measurement the OSTAR IR LED was tracked. The red circle is the fitted track constructed from all of the measured values. The blue (x) values show the position in the virtual model at the instant when the actual IR LED passed the positive x axis. The two angles show the segment inside which all these values reside. The circulating IR LED cycle time is 1,4833 s. From these we derive the latency of approximately 280 – 320 ms.

The identification of the IR LEDs was successful in all test cases, although the refresh rate slows down depending on the length of the blinking period. The detection of the minimum blinking period was left for future research. The prerequisite for the first IR LED identification application program is that all the LEDs are inside the field of vision. Also the IR LED blinking synchronization is required.

The maximum number of IR LEDs for one Wiimote is only four. This restricts the amount of objects that can be tracked simultaneously. The orientation of only one object can be calculated with this system. Here we assume that minimal three points are needed for tracking the object orientation and that the object is not symmetric.

B. Two and Four Wiimote System Evaluation Results

The Fig. 7 shows the cumulative probability distribution of the positioning error in the measurement sessions where two Wiimotes were used. There were altogether 20 positions measured in each case, which we denoted by C1, C2, C3, C4 and C5.

We see that without the principal point correction the positioning error is between 15 cm and 50 cm in the cases C1 and C2. With the principal point correction the error is between 0 – 7 cm in the cases C3, C4 and C5.

The difference after using the principal point correction is best depicted in Fig. 8. There are three cubes drawn of which the red dashed line represents the ideal measurement cube. The green cube with complete line is the cube measured without the principal point correction. The green cube is the measurement case C2, in which the intrinsic camera matrix was extracted from the 12 images using the Matlab Camera Calibration Toolbox. The last blue cube is the measurement case C4, where the principal point correction method was used. After using the principal point correction method, the cube of the case C2 is shifted very close to the ideal red cube.

The positioning error of over 6 cm originates from the closest cube corner to the reader. We can clearly see that it has the most error of the blue cube corners. This corner is furthest from the two Wiimotes that are located in the opposing corners. The two Wiimotes were placed at the positions of Wiimotes 1 and 2 in the Fig. 1. The closest corner to the reader, with the most error is the Wiimote 4 corner in the Fig. 1.

It is possible to compensate the positioning error with 3D data remapping if we know the cube shift and deformation. Even the worst positioning errors caused by the false principal points can be compensated by remapping and e.g. using tensor mathematics [24]. If this is the case and the cube corresponding points are well known, for the same camera configuration, the system can be recalibrated very accurately.

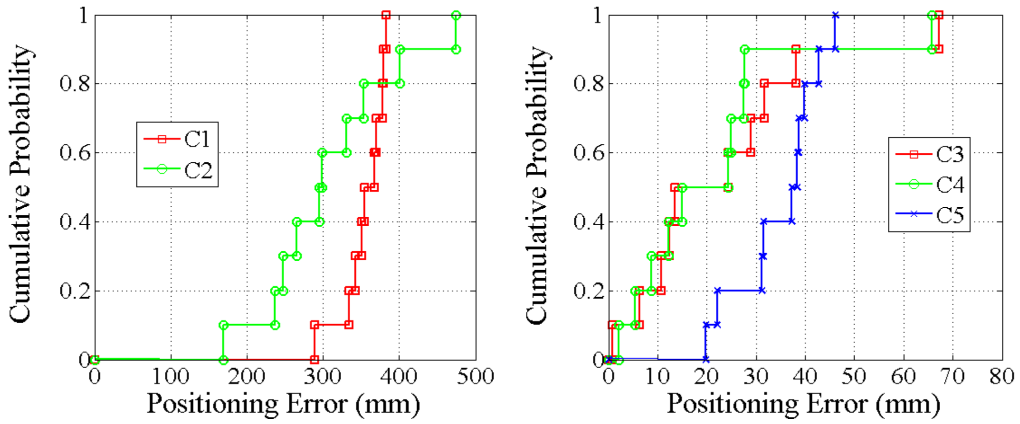

Figure 7. The cumulative probability distributions of the positioning errors for the cases where **two** Wiimotes were used. On the left the principal point correction method was not used. On the right the principal point method was used. In the cases C1 and C3 the intrinsic camera matrix, taken from [17], was used. In the cases C2 and C4 the matrix was acquired by the Matlab Camera Calibration Toolbox. For the case C5 the principal point of the C4 case was changed to value (512, 384).

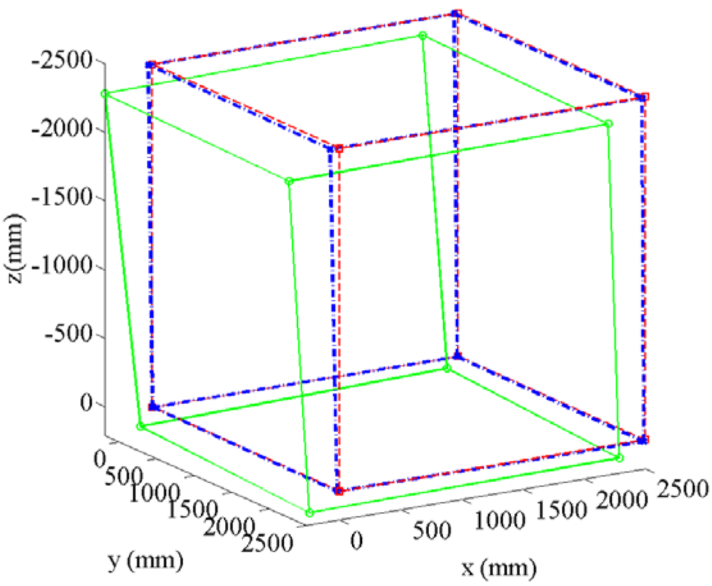

Figure 8. The ideal cube, formed from the ideal measurement points, is shown with the red dashed line. The green cube depicts the measurement case C2 where the principal point correction was not used. For the blue cube C4 the principal point correction was applied. Two Wiimotes were used in the measurements described in this figure. The two Wiimotes were placed 4.5 meters from the corners that are in the far right and far left. We see the effect of the wrong principal point values with the green shifted cube. This cube is also slightly deformed. Principal point correction was used 14 times in the measurements of the blue cube.

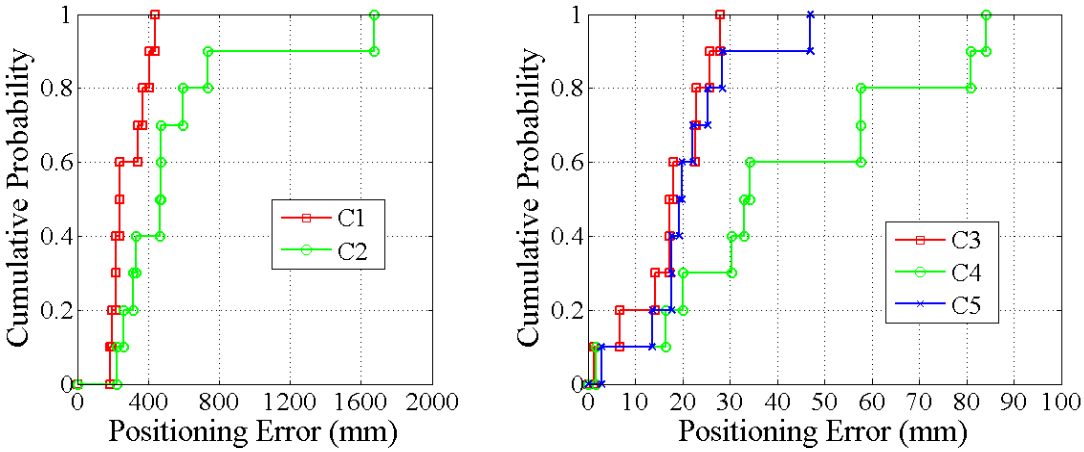

Figure 9. The cumulative probability distributions of the positioning errors for the cases where **four** Wiimotes were used. On the left the principal point correction method was not used. On the right the principal point method was used. In the cases C1 and C3 the intrinsic camera matrix, taken from [17], was used. In the cases C2 and C4 the matrix was acquired by the Matlab Camera Calibration Toolbox. For the case C5 the principal point of the C4 case was changed to value (512, 384). The most accurate positioning results were acquired in the case C3 (2.8 cm).

In the Fig.9 are shown the measurement cases for the sessions where four Wiimotes were used. In this set-up the cases C2 and C4 are especially erroneous. When only 12 images were given for the Matlab camera calibration toolbox the error limits for the principal points were as high as ±40 pixels. This error seemed to accumulate all the way to the positioning error and this can also be seen in the green graphs of Fig. 9. Clearly only 12 images constructed with the Wiimote and the test rig are not enough for the extraction of the intrinsic parameters in Matlab Camera Calibration Toolbox. In [19], thousands of image pairs were collected in order to achieve ±1% tolerance within the intrinsic parameters.

As well, the 14 recalibration rounds were not enough to fully compensate for the wrong principal point values in the measurement case C4. The C4 error value was close to 8.5 cm while in comparison the C3 and C5 values were under 4.8 cm.

Especially the initial matrix (2), taken from [17], proved to be good with the four Wiimote system measurements. Here the positioning error was only 2.8 cm. These preliminary results indicate slightly that adding more Wiimotes to the system increases the total positioning accuracy. The larger amount of cameras recording a certain feature increases the redundancy and thus, in positive test cases, the positioning accuracy. It is also good to point out here that the principal point values after the principal point correction were always only ±5 pixels away from the value (512, 384).

VII. CONCLUSIONS

We have examined the characteristics of the constructed positioning system. The tests for identifying separate IR LEDs were successful. It is possible to detect the orientation of one object with three or four IR LEDs. The positioning system latency is within a tolerable level for many applications.

The principal point correction method succeeds in compensating and correcting the differing initial principal points. Even without 3D data remapping or time consuming calibration methods the presented system is able to position with an accuracy of 3 cm. It becomes difficult to discern whether the positioning error is caused by manual placement errors or by errors in the virtual model of the tracking scheme. In future experimentations more precise reference measurement framework is needed. For many tracking applications this level of accuracy is sufficient.

If the distortion coefficients are taken into account, the principal point method has to be revised. The method accuracy depends on the 2D data location of the origin on the individual image planes. At different 2D points the compensation works slightly different. More points could be used in the method, at least the three other points of the calibration square that were measured during the calibration phase.

One applicable idea for the future is the usage of lenses. A lens on the Wiimote camera could widen the field of view enabling closer positioning of Wiimotes to the tracking area. An omnidirectional camera in the center of a room at the ceiling could cover the whole tracking space. A drawback would be the degraded resolution.

The system can easily be extended to cover larger areas. This is mostly a programming and image processing task. The system implementation into an office could offer interesting alternative location based services. The gestures performed with the controller could vary between rooms. The maximum possible amount of Wiimotes is one of the limiting factors.

ACKNOWLEDGMENT

The authors wish to thank H. Raula, K. Palovuori, M. Jukola, M. Rajala, N. Narra and the reviewers for the comments, help and support in the research.

REFERENCES

[1] J. C. Lee, “Hacking the Nintendo Wii Remote,” IEEE Pervasive Computing, vol. 7, no. 3, pp. 39-45, 2008.

[2] Y. Fukuju, M. Minami, H. Morikawa, and T. Aoyama, ”DOLPHIN: An autonomous indoor positioning system in ubiquitous computing environment,” Software Technologies for Future Embedded Systems, pp. 53-56, 2003.

[3] M. Valtonen, L. Kaila, J. Mäentausta, and J. Vanhala, ”Unobtrusive human height and posture recognition with a capacitive sensor,” Journal of Ambient Intelligence and Smart Environments, vol. 3, no. 4, pp. 305-332, 2011.

[4] L. Bharath, S. Shashank, V. S. Nageli, S. Shiravastava, and S. Rakshit, “Tracking method for human computer interaction using Wii Remote,” Emerging Trends in Robotics and Communication Technologies, pp. 133-137, 2010.

[5] S. De Amici, A. Sanna, F. Lamberti, and P. Bralio, ”A Wii Remote-based infrared-optical tracking system,” Entertainment Computing, vol. 1, no. 3-4, pp. 119-124, 2010.

[6] http://www.naturalpoint.com/trackir/, NaturalPoint Inc., (20.10.2012)

[7] http://www.ar-tracking.com/home/, Advanced Realtime Tracking GmbH, (20.10.2012)

[8] C. A. Wingrave, B. Williamson, P. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. J. LaViola Jr., ”The Wiimote and beyond: spatially convenient devices for 3D user interfaces,” IEEE Computer Graphics and Applications, vol. 30, no. 2, pp. 71-85, 2010.

[9] S. Attygalle, M. Duff, T. Rikakis, and J. He, ”Low-cost, at-home assessment system with Wii Remote based motion capture,” Virtual Rehabilitation, pp. 168-174, 2008.

[10] T. Petrič, A. Gams, A. Ude, and L. Žlajpah, ”Real-time 3D marker tracking with a WIIMOTE stereo vision system: application to robotic throwing,” Robotics in Alpe-Adria-Danube Region, pp. 357-362, 2010.

[11] H. Vargas, E. Preza, and R. Velázquez, ”A combined Wiimote-camera tracking system for small aerial vehicles,” Electronics, Robotics and Automotive Mechanics Conference, pp. 226-231, 2009.

[12] S. D. Blostein, and T. S. Huang, ”Error analysis in stereo determination of 3-D point positions,” Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 6, pp. 752-765, 1987.

[13] R. S. Ramakrishna, and B. Vaidyanathan, ”Error analysis in stereo vision,” Computer Vision, vol. 1351, pp. 296-304, 1997.

[14] A. Belhaoua, S. Kohler, and E. Hirsch, ”Estimation of 3D reconstruction errors in a stereo-vision system,” Modeling Aspects in Optical Metrology II, vol. 7390, pp. 73900X-1-73900X-10, 2009.

[15] Z. Zhang, ”A flexible new technique for camera calibration,” vol. 22, no. 11, pp. 1330-1334, 2000.

[16] Matlab Camera Calibration Toolbox - http://www.vision.caltech.edu/bouguetj/calib_doc/, J.-Y. Bouquet (20.10.2012)

[17] A. Yildiz, A. Akay, and Y. S. Akgul, ”Wii Remote calibration using the sensor bar,” Computer Vision and Pattern Recognition Workshops, pp. 7-12, 2011.

[18] S. Hay, J. Newman, and R. Harle, ”Optical tracking using commodity hardware,” Mixed and Augmented Reality, pp. 159-160, 2008.

[19] A. Boyali, and M. Kavakli, ”3D and 6 DOF user input platform for computer vision applications and virtual reality,” Innovations in Intelligent Systems and Applications, pp. 258-263, 2011.

[20] A. Belhaoua, S. Kohler, and E. Hirsch, ”Error evaluation in a stereovision-based 3D reconstruction system,” Journal on Image and Video Processing, vol. 2010, no. 2, 2010.

[21] G. Olague, and R. Mohr, ”Optimal 3D sensor placement to obtain accurate 3D points positions,” Pattern Recognition, vol. 1, pp. 8-10, 1998.

[22] G. Olague, ”Autonomous photogrammetric network design using genetic algorithms,” Applications of evolutionary computing, vol. 2037, pp.353-363, 2001.

[23] http://sourceforge.net/projects/wiiuse/, Programming library written in C for Wii Remotes, M. Laforest (20.10.2012)

[24] R. Hartley, and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge Univ. Press, 2003.