Sensors **2015**, *15*, 15888-15902; doi:10.3390/s150715888

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Foot Pose Estimation Using an Inertial Sensor Unit and Two Distance Sensors

Pham Duy Duong and Young Soo Suh *

Department of Electrical Engineering, University of Ulsan, Namgu, Ulsan 680-749, Korea;
E-Mail: duyduongd2@gmail.com

* Author to whom correspondence should be addressed; E-Mail: yssuh@ulsan.ac.kr;
Tel.: +82-52-259-2196; Fax: +82-52-259-1686.

Academic Editor: Gert F. Trommer

Received: 22 May 2015 / Accepted: 30 June 2015 / Published: 3 July 2015

Abstract: There are many inertial sensor-based foot pose estimation algorithms. In this paper, we present a methodology to improve the accuracy of foot pose estimation using two low-cost distance sensors (VL6180) in addition to an inertial sensor unit. The distance sensor is a time-of-flight range finder and can measure distance up to 20 cm. A Kalman filter with 21 states is proposed to estimate both the calibration parameter (relative pose of distance sensors with respect to the inertial sensor unit) and foot pose. Once the calibration parameter is obtained, a Kalman filter with nine states can be used to estimate foot pose. Through four activities (walking, dancing step, ball kicking, jumping), it is shown that the proposed algorithm significantly improves the vertical position estimation.

Keywords: inertial sensor; distance sensor; foot pose estimation; Kalman filters

1. Introduction

Foot pose (position and orientation) estimation is used in many areas, such as gait analysis [1–3], exergaming [4,5] and pedestrian navigation systems [6]. The most accurate method to estimate foot pose is using an optical motion tracker. However, optical motion tracking is rather expensive, and it can only capture the motion in a limited space, which is determined by the number of cameras.

Recently, inertial sensor-based foot pose estimation methods are increasingly used, since the sensor can be attached to a shoe, which does not require any infrastructure in an environment and, thus, removes the space constraint imposed by optical motion trackers. As inertial sensors are becoming less expensive and smaller, inertial sensor-based motion tracking (ISBMT) is expected to be more popular.

The basic principle of an ISBMT is integration of gyroscope output (which gives the orientation) and double integration of accelerometer outputs (which gives the position). In the process, the sensors noises are also integrated. For example, if there is a nonzero accelerometer sensor bias term, it affects the position estimation error proportional to the square of elapsed time [7]. Furthermore, initial orientation error also significantly affects the position estimation error, since the gravitational acceleration is not correctly removed during the position estimation [7].

In foot pose tracking, the error increase can be mitigated using zero velocity updating [8,9]. Whenever a foot touches the ground, we know the velocity of the foot is zero. Using this zero velocity information, the estimation error can be significantly reduced [6].

Another approach to reduce the estimation error is to use a smoother [10]. The smoother uses both prior and posterior data to estimate the current state. This combination reduces the estimation error. One drawback of a smoother is that its computation cannot be done online and, thus, is not suitable for applications, such as gaming.

The pose estimation error also can be reduced by using additional sensors. In [11], a pressure sensor is attached on a shoe to accurately detect the zero velocity intervals, which increases the estimation accuracy. In [12], an ultrasound range sensor is used to measure the distance to walls. The ultrasound sensor does not improve foot pose estimation, but improves the location accuracy for indoor pedestrian navigation. In [13], a camera is used to read markers on the floor, which gives the absolute position and orientation of a foot. One disadvantage of this approach is that the markers must be installed on the floor, so the experiment space is limited by the number of makers. In [14], a camera and infrared LEDs are used to measure the relative pose between two feet.

In this paper, we present a methodology to improve the accuracy of foot pose estimation by attaching low-cost distance sensors on a shoe in addition to an inertial sensor unit. Since the distance sensor gives height information, it can help to improve height estimation accuracy. The improved accuracy could be helpful in gait analysis and exergaming applications. The foot pose estimation algorithm is implemented using a Kalman filter. To combine inertial sensors and distance sensors, the relative pose between two sensors should be known. This pose parameter is included in the proposed Kalman filter.

The paper is organized as follows. In Section 2, the system hardware and coordinate systems are introduced. In Section 3, dynamic equations of the Kalman filter for ISBMT are given. In Section 4, measurement equations of the Kalman filter are provided. In Section 5, the proposed algorithm is tested, and the estimated positions are compared with optical tracker measurement values. The discussion is given in Section 6.

2. System Overview

An inertial measurement unit (three-axis accelerometers and three-axis gyroscopes, Xsens MTi) is attached on a shoe as in Figure 1. The sampling frequency of the inertial sensors is 100 Hz. Two distance sensors (VL6180 [15]) are also attached on a shoe, where symbols A and B are used to distinguish the two sensors. The distance sensor VL6180 measures the distance by measuring the time-of-flight of infrared light, and the measurement range is up to 20 cm. This sensor is most often used as a proximity sensor in smartphones. The sampling frequency of the distance sensors is 33.33 Hz.

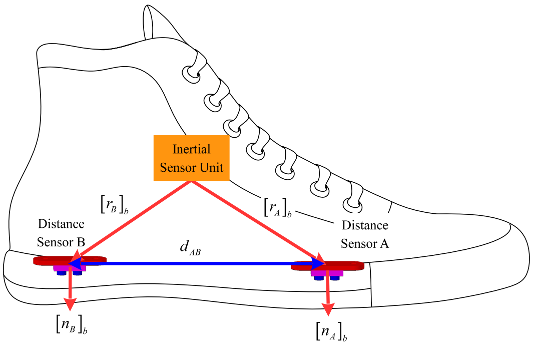

Figure 1. Inertial and distance sensors on a shoe.

Two coordinate systems are used in this paper: the body and world coordinate systems. The three axes of the body coordinate system coincide with the three axes of the inertial sensor unit. The z axis of the world coordinate system is in the direction of the local gravitational field: the z axis is pointing upward. The x and y axes are chosen arbitrarily. The origin of the world coordinate system is assumed to be on the floor.

The relative position and orientation of a distance sensor with respect to the inertial sensor unit are required in the foot pose estimation algorithm. As shown in Figure 1, the positions of two distance sensors are denoted $[r_A]_b \in R^3$ and $[r_B]_b \in R^3$, which are the position coordinates in the body coordinate system. The notation $[p]_b$ ($[p]_w$) for a vector $p \in R^3$ is used to emphasize that a vector is represented in the body (world) coordinate system. When there is no confusion, $[p]_b$ (or $[p]_w$) is just expressed by p. The pointing direction of the distance sensor is denoted by a unit vector $n_A \in R^3$ and $n_B \in R^3$.

The distance sensor parameters (r_A, r_B, n_A, n_B in Figure 1) could be determined using a ruler and a protractor. It would be, however, not easy to determine the parameters with high accuracy. Thus, the small errors in the parameters are estimated in the Kalman filter. We model the parameters as follows:

$$\begin{aligned} r_A &= \hat{r}_A + \bar{r}_A \\ r_B &= \hat{r}_B + \bar{r}_B \\ n_A &= \hat{n}_A + \bar{n}_A \\ n_B &= \hat{n}_B + \bar{n}_B \end{aligned} \tag{1}$$

where $\hat{r}_A \sim \hat{n}_B$ are initial estimated parameter values (most likely measured values using a ruler and a protractor) and $\bar{r}_A \sim \bar{n}_B$ are errors in the parameter estimation.

The distance between two distance sensors is denoted by $d_{AB} = \|r_A - r_B\|$, which is measured by a ruler. This scalar quantity can be measured more accurately comparing with other vector parameters, which are rather difficult to measure and sometimes require guesswork. Thus, the error of d_{AB} is not estimated. The measured value of d_{AB} is denoted by z_{AB}:

$$z_{AB} = d_{AB} + v_{AB} \tag{2}$$

where v_{AB} is the measurement noise.

3. Kalman Filter for ISBMT

In this section, the foot pose tracking algorithm is given.

Let $r \in R^3$ and $v \in R^3$ denote the position and velocity of the inertial sensor unit in the world coordinate system. Let $q \in R^4$ be the quaternion [16] representing the rotation relationship between the body and world coordinate systems. Let $C(q) \in SO(3)$ be the rotation matrix corresponding to the quaternion q.

3.1. Basic Pose Equations

The basic equations for r, v and q are given by [17]:

$$\begin{aligned} \dot{q} &= \tfrac{1}{2}\Omega(\omega)q \\ \dot{v} &= C(q)'[a]_b \\ \dot{r} &= v \end{aligned} \tag{3}$$

where $[a]_b \in R^3$ is the acceleration expressed in the body coordinate system and $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}'$ is the angular velocity of the body coordinate system with respect to the world coordinate system. The symbol $\Omega(\omega)$ is defined by:

$$\Omega(\omega) \triangleq \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

3.2. Numerical Integration

The basic principle of ISBMT is that q, v and r can be computed by numerically integrating Equation (3) [7]. To do that, we need to know ω and $[a]_b$, which can be measured using gyroscopes and accelerometers.

The gyroscope output ($y_g \in R^3$) and accelerometer output ($y_a \in R^3$) can be modeled as follows:

$$\begin{aligned} y_g &= \omega + v_g \\ y_a &= [a]_b + C(q)[\tilde{g}]_w + v_a \end{aligned} \tag{4}$$

where $[\tilde{g}]_w = \begin{bmatrix} 0 & 0 & g \end{bmatrix}' \in R^3$ is the local gravitational vector in the world coordinate system and g is the magnitude of the gravitational acceleration. The measurement noises v_g and v_a are assumed to be uncorrelated zero mean white Gaussian.

We can integrate Equation (3) by replacing ω by y_g and replacing $[a]_b$ by $y_a - C(q)\tilde{g}$. The numerical integration algorithm is given in [18]. Let the integrated values be denoted by $\hat{q}$, $\hat{r}$ and $\hat{v}$.

Since $\omega \neq y_g$ and $[a]_b \neq y_a - C(\hat{q})\tilde{g}$, there are errors in $\hat{q}$, $\hat{r}$ and $\hat{v}$, which are denoted by $\bar{q} \in R^3$, $\bar{r} \in R^3$ and $\bar{v} \in R^3$:

$$\begin{aligned} \bar{q} &= \begin{bmatrix} 0_{3\times 1} & I_3 \end{bmatrix} (\hat{q}^* \otimes q) \\ \bar{r} &= r - \hat{r} \\ \bar{v} &= v - \hat{v} \end{aligned} \tag{5}$$

where $\otimes$ denotes the quaternion multiplication and q^* is the conjugate quaternion of $q \in R^4$. The definition of $\bar{q} \in R^{3\times 1}$ is from the assumption that the estimation error of $\hat{q}$ is small, and thus, the following is satisfied [19]:

$$q \approx \hat{q} \otimes \begin{bmatrix} 1 \\ \bar{q} \end{bmatrix} \tag{6}$$

The errors $\bar{q}$, $\bar{r}$ and $\bar{v}$ along with the parameter errors $\bar{r}_A \sim \bar{n}_B$ in Equation (1) are estimated in the Kalman filter. Combining nine states in Equation (5) and 12 states in Equation (1), we have the following state for a Kalman filter:

$$x = \begin{bmatrix} \bar{q} \\ \bar{r} \\ \bar{v} \\ \bar{r}_A \\ \bar{r}_B \\ \bar{n}_A \\ \bar{n}_B \end{bmatrix} \in R^{21\times 1} \tag{7}$$

Once the calibration parameters $\hat{r}_A \sim \hat{n}_B$ are estimated, the parameter error terms $\bar{r}_A \sim \bar{n}_B$ in Equation (7) can be removed for fast computation.

The dynamic equation of x is given by:

$$\dot{x}(t) = A(t)x(t) + w(t) \tag{8}$$

where:

$$A(t) = \begin{bmatrix} [-y_g\times] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 & 0 & 0 \\ -2C(\hat{q})'[y_a\times] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad w(t) = \begin{bmatrix} -\frac{1}{2}v_g \\ 0 \\ -C(\hat{q})'v_a \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The dynamic equations for $\bar{q}$, $\bar{r}$ and $\bar{v}$ are from the result in [20]. The derivatives of $\bar{r}_A \sim \bar{n}_B$ are zero, since $r_A \sim n_B$ are constant parameters.

4. Measurement Equation of the Kalman Filter

Two measurement equations are used in the Kalman filter. One is the measurement from distance sensors (Section 4.1), and the other is the measurement equation using the zero velocity intervals (Section 4.2).

4.1. Distance Sensor Output and Parameters

The distance sensor outputs $z_A \in R$ and $z_B \in R$ can be modeled as follows:

$$\begin{aligned} z_A &= d_A + v_A \\ z_B &= d_B + v_B \end{aligned} \tag{9}$$

where d_A and d_B are true distance values of sensors Aand B, respectively. Symbols v_A and v_B denote measurement noises of A and B, respectively. We note that d_A and d_B are not the same as the heights of distance sensors A and B if they are not perpendicular to the floor. The heights can be obtained from d_A and d_B if we know the orientation of distance sensors.

Assuming that the floor is flat and the origin of the world coordinate system lies on the floor, the following is satisfied:

$$\begin{aligned} \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] [r]_w &= -\left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] C(q)' \left[r_A + n_A d_A\right]_b \\ \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] [r]_w &= -\left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] C(q)' \left[r_B + n_B d_B\right]_b \end{aligned} \tag{10}$$

In Equation (10), $-\left(r_A + n_A d_A\right)$ is a vector (in the body coordinate system) from a point (intersection of line n_A and floor plane) on the floor to the inertial sensor. By pre-multiplying $C(q)'$ (the rotation matrix from the body coordinate system to the world coordinate system), the third component of $C(q)'\left(r_A + n_A d_A\right)$ is the height of the inertial sensor.

Now, we relate Equation (10) to the state Equation (7). Note that [21]:

$$C(q)' = C(\hat{q})'(I + 2[\bar{q}\times]) \tag{11}$$

Inserting Equations (1), (5), (9) and (11), into the first equation of Equation (10), we have the following:

$$-\left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] C(\hat{q})'(I + 2[\bar{q}\times])\left(\hat{r}_A + \bar{r}_A + (\hat{n}_A + \bar{n}_A)(z_A - v_A)\right) = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] (\hat{r} + \bar{r}) \tag{12}$$

Assuming that the error terms ($\bar{q}$, $\bar{r}_A$, $\bar{n}_A$, $\bar{r}$) and the noise term (v_A) are small, we can ignore the product terms: for example, $[\bar{q}\times]\bar{r}_A$ is ignored. By ignoring the product terms, we obtain:

$$\tilde{z}_A = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] \left(2C(\hat{q})'[(\hat{r}_A + \hat{n}_A z_A)\times]\bar{q} - \bar{r} - C(\hat{q})'\bar{r}_A - C(\hat{q})' z_A \bar{n}_A + C(\hat{q})'\hat{n}_A v_A\right) \tag{13}$$

where:

$$\tilde{z}_A = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] \left(\hat{r} + C(\hat{q})'(\hat{r}_A + \hat{n}_A z_A)\right) \in R$$

Thus, the measurement $\tilde{z}_A$ is related to the state x as follows:

$$\tilde{z}_A = H_A x + C(\hat{q})'\hat{n}_A v_A \tag{14}$$

where:

$$H_A = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccccccc} 2C(\hat{q})'[(\hat{r}_A + \hat{n}_A z_A)\times] & -I_3 & 0 & -C(\hat{q})' & 0 & -C(\hat{q})' z_A & 0 \end{array}\right] \in R^{1\times 21}$$

Similarly, we can derive a measurement equation for the distance sensor B:

$$\tilde{z}_B = H_B x + C(\hat{q})'\hat{n}_B v_B \tag{15}$$

where:

$$\tilde{z}_B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (\hat{r} + C(\hat{q})'(\hat{r}_B + \hat{n}_B z_B)) \in R$$

$$H_B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2C(\hat{q})'[(\hat{r}_B + \hat{n}_B z_B)\times] & -I_3 & 0 & 0 & -C(\hat{q})' & 0 & -C(\hat{q})' z_B \end{bmatrix} \in R^{1\times 21}$$

In addition to Equations (13) and (15), some constraints are added in the measurement equations. The first constraint is n_A and n_B should be unit vectors since they represent direction vectors. The second constraint is that $\|r_A - r_B\| = d_{AB}$.

The constraint (n_A and n_B should be unit vectors) is expressed as:

$$\|n_A\| = \|\hat{n}_A + \bar{n}_A\| = \hat{n}'_A \hat{n}_A + 2\hat{n}'_A \bar{n}_A + \bar{n}'_A \bar{n}_A = 1 \tag{16}$$

From $\|\hat{n}_A\| = \hat{n}'_A \hat{n}_A = 1$ and $\bar{n}'_A \bar{n}_A \approx 0$ (assuming that $\bar{n}_A$ is small), we have the following approximation:

$$\hat{n}'_A \bar{n}_A = 0 \tag{17}$$

Thus, constraints on n_A and n_B are imposed through the following measurement equations:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = H_{constraint} x + v_{constraint} \tag{18}$$

where:

$$H_{constraint} = \begin{bmatrix} 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & \hat{n}'_A & 0_{1\times 3} \\ 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & \hat{n}'_B \end{bmatrix} \in R^{2\times 21}$$

The noise term $v_{constraint} \in R^{2\times 1}$ is an artificial noise reflecting the fact that the constraint Equation (17) is an approximation.

Inserting Equation (2) into the constraint $\|r_A - r_B\| = d_{AB}$, we have:

$$z_{AB} = \|\hat{r}_A + \bar{r}_A - (\hat{r}_B + \bar{r}_B)\| + v_{AB} \tag{19}$$

Since it has the nonlinear relationship, an extended Kalman filtering technique is used for this measurement.

$$\tilde{z}_{AB} = H_{AB} x + v_{AB} \tag{20}$$

where:

$$\tilde{z}_{AB} = z_{AB} - \|\hat{r}_A - \hat{r}_B\| \in R$$

$$\begin{aligned} H_{AB} &= \begin{bmatrix} 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & \frac{\partial d_{AB}}{\partial \bar{r}_A} & \frac{\partial d_{AB}}{\partial \bar{r}_B} & 0_{1\times 3} & 0_{1\times 3} \end{bmatrix} \\ &= \begin{bmatrix} 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & \frac{(\hat{r}_A - \hat{r}_B)'}{\|\hat{r}_A - \hat{r}_B\|} & \frac{-(\hat{r}_A - \hat{r}_B)'}{\|\hat{r}_A - \hat{r}_B\|} & 0_{1\times 3} & 0_{1\times 3} \end{bmatrix} \in R^{1\times 21} \end{aligned}$$

In summary, Equations (14), (15), (18) and (20) are used as measurement equations for the Kalman filter if distance sensor data z_A and z_B are available. In this case, the measurement equation is given by:

$$z = \begin{bmatrix} \tilde{z}_A \\ \tilde{z}_B \\ 0_{2\times 1} \\ \tilde{z}_{AB} \end{bmatrix} = \begin{bmatrix} H_A \\ H_B \\ H_{constraint} \\ H_{AB} \end{bmatrix} x + \begin{bmatrix} C(\hat{q})'\hat{n}_A v_A \\ C(\hat{q})'\hat{n}_B v_B \\ v_{constraint} \\ v_{AB} \end{bmatrix}$$

4.2. Zero Velocity Updating

In addition to distance sensors, zero velocity updating is used in a measurement equation. If we know when a foot is not moving (for example, when a foot is on the ground), the velocity error in the ISBMT algorithm can be reset, which significantly reduces foot pose estimation errors. Since velocity sensors (such as a Doppler velocity sensor in [22]) are not used, the zero velocity intervals are detected indirectly, where many zero velocity detection algorithms have been proposed [8,23].

In this paper, we use a simple zero velocity detection algorithm. Let Z_m be a set of all discrete time indices belonging to the zero velocity intervals. The discrete time k belongs to Z_m if the following conditions are satisfied:

$$\begin{array}{ll} \|y_{g,i}\| \leq B_g, & k - \frac{N_g}{2} \leq i \leq k + \frac{N_g}{2} \\ \|y_{a,i} - y_{a_{i-1}}\| \leq B_a, & k - \frac{N_a}{2} \leq i \leq k + \frac{N_a}{2} \end{array} \tag{21}$$

where N_g and N_a are even number integers.

During zero velocity intervals, we assume the following:

$$v = 0_{3\times1} \tag{22}$$

Inserting Equation (5) into Equation (22), we have the following measurement equation in the Kalman filter.

$$z_v = H_v x + v_{zero} \tag{23}$$

where:

$$z_v = 0_{3\times1} - \hat{v} \in R^{3\times1}$$

$$H_v = \left[\begin{array}{ccccccc} 0_{3\times3} & 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{array} \right] \in R^{3\times21}$$

The noise term $v_{zero} \in R^{3\times1}$ is an artificial noise reflecting the fact that the true velocity may not be zero (the velocity may be almost zero, but not exactly zero).

5. Experiments and Results

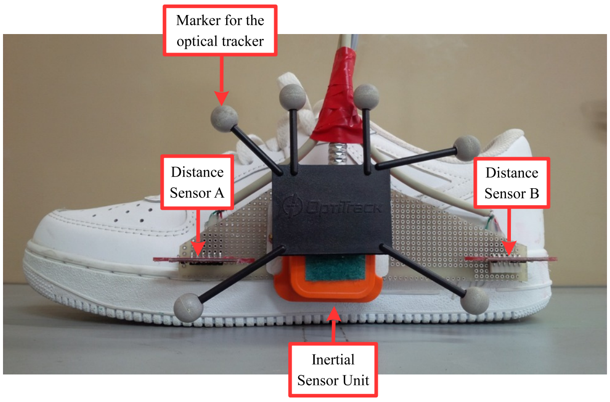

Figure 2. A shoe for the experiment.

The proposed algorithm is tested with several foot movement experiments. The foot position is measured with an inertial sensor unit (Xsens MTi) and two distance sensors (VL6180) as in Figure 2. The estimated positions are compared with the positions obtained using an optical motion tracker (Optitrack six Flex 13 camera system), which is considered as a ground truth.

The parameters used in the proposed algorithm are summarized in Table 1. All noises are assumed to be uncorrelated white Gaussian noises.

Table 1. Parameters in the proposed algorithm.

Parameter	Value	Related Equations
initial value of $\hat{r}_A$	$[\ -0.008\quad 0.080\quad 0\ ]'$	(1)
initial value of $\hat{r}_B$	$[\ -0.006\quad -0.062\quad 0\ ]'$	(1)
initial value of $\hat{n}_A$	$[\ 1\quad 0\quad 0\ ]'$	(1)
initial value of $\hat{n}_B$	$[\ 1\quad 0\quad 0\ ]'$	(1)
z_{AB}	0.142	(2)
$\mathrm{E}\{v_g v_g'\}$	$0.00017 I_3$	(4)
$\mathrm{E}\{v_a v_a'\}$	$0.0083 I_3$	(4)
P_{init} = initial value of $\mathrm{E}\{xx'\}$	$P_{init}(2:3, 2:3) = 0.001 I_2(^*)$, $P_{init}(6,6) = 0.0001$, $P_{init}(10:21, 10:21) = 0.000001 I_{12}$ (all unspecified elements are zero)	(7)
$\mathrm{E}\{v_A v_A'\}$	0.0004	(9)
$\mathrm{E}\{v_B v_B'\}$	0.0004	(9)
$\mathrm{E}\{v_{constraint} v_{constraint}'\}$	$0.00000001 I_2$	(18)
$\mathrm{E}\{v_{AB}^2\}$	0.000001	(20)
$\mathrm{E}\{v_{zero} v_{zero}'\}$	$0.0001 I_3$	(23)

* $P_{init}(2:3, 2:3)$ represents a $R^{2\times 2}$ matrix consisting of second and third rows and column elements.

Four motions (walking, dancing steps, kicking in a football, jumping) are tested. The estimated position is compared with two other inertial sensor-only pose estimations. In the first inertial sensor-only pose estimation, the same algorithm is used, except for the distance sensors: that is, Equations (14) and (15) are not used in the Kalman filter. We call this method "K.F. (zero velocity updating)". In the second inertial sensor-only pose estimation, the height updating is added in the first inertial sensor-only pose estimation. If we assume that the floor is flat, the foot height during the zero velocity intervals (that is, when a foot is on the ground) should be the same. Thus, in addition to zero velocity updating, the z axis value of r is updated to the initial z axis value during each zero velocity interval. We call this method "K.F. (zero velocity + height updating)"

In Figures 3–5, the estimated positions (walking case) by three methods (the proposed method and the two inertial sensor-only methods) are given along with the position by an optical tracker. Since the optical tracker coordinate system is different from the world coordinate system, the position data are translated and rotated. Furthermore, the inertial sensor data and optical tracker data are not synchronized at the hardware level. The data are synchronized by maximizing the cross-correlation.

We can see that x and y axis position estimations are similar in all three methods. This is not surprising, since the distance sensor only gives the z axis position (height) information. In the z axis position estimation, the proposed method gives better results (see the third graphs of Figures 3–5 around 4.6, 6 and 7 s). Among inertial sensor-only estimations, the Kalman filter (zero velocity + height updating) is better, since its height is compensated during the zero velocity intervals.

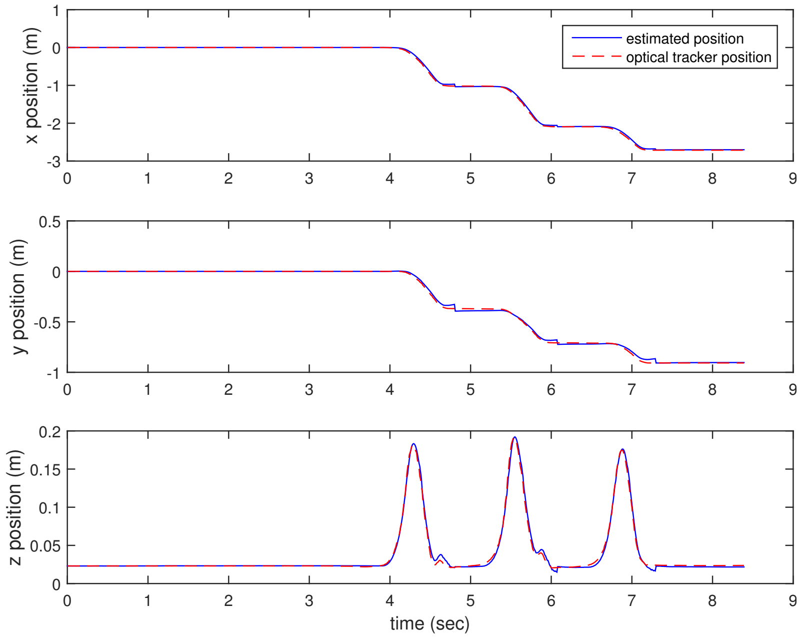

Figure 3. Walking position estimation: proposed method.

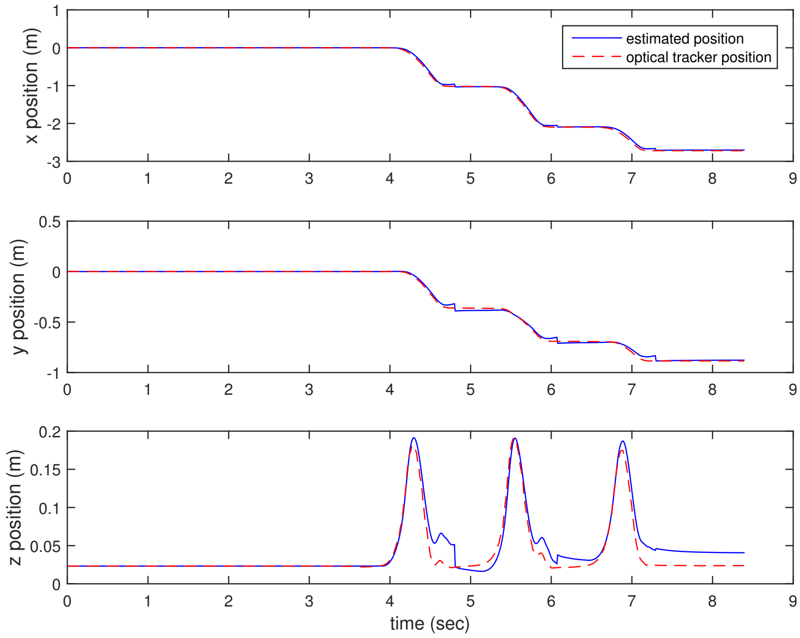

Figure 4. Walking position estimation: K.F. (zero velocity updating). K.F., Kalman filter.

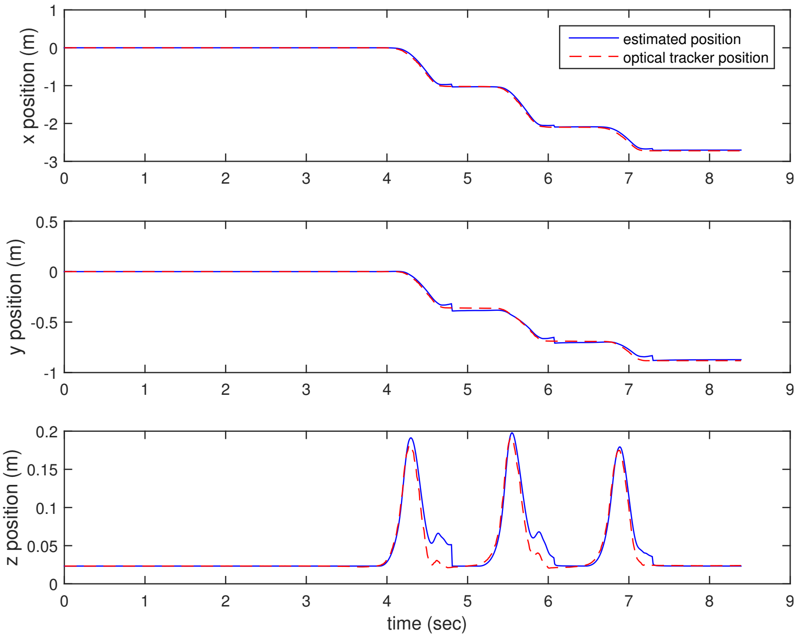

Figure 5. Walking position estimation: K.F. (zero velocity + height updating).

Another example is given in Figures 6 and 7, where a ball kicking action is done. The x and y position estimation results are similar, both by the proposed method and K.F. (zero velocity + height updating). In this case, the position errors are quite large, presumably due to the fact that there is a rather long moving interval (2.8∼4-s interval) and a quick movement (large sensor values). In the z axis position estimation, we can see that the proposed method gives a significantly better result.

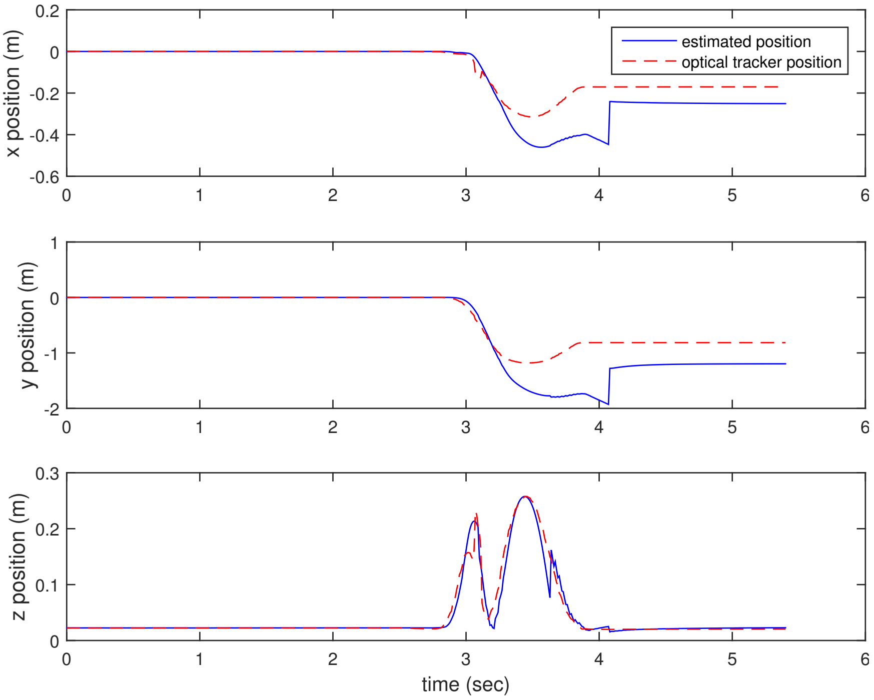

Figure 6. Ball kicking estimation: proposed method.

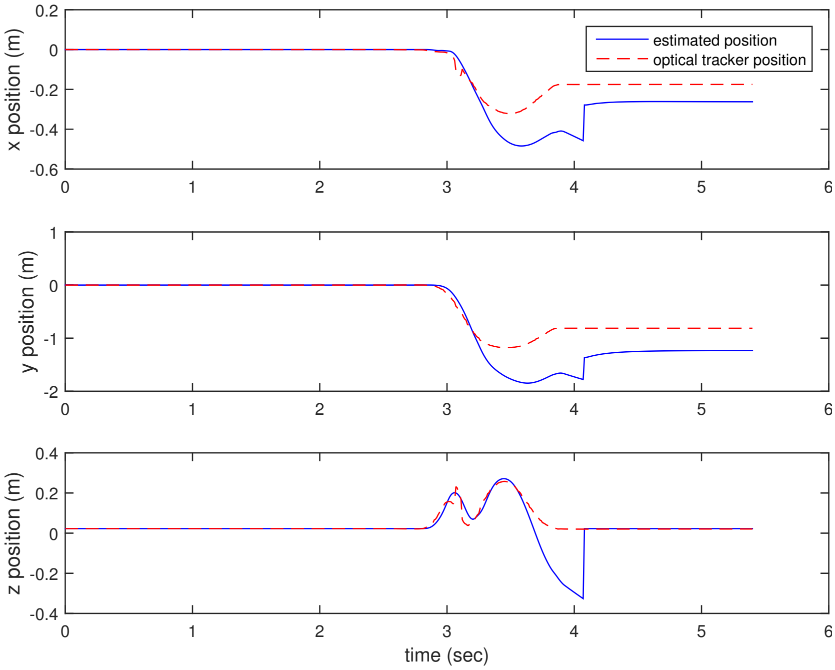

Figure 7. Ball kicking estimation: K.F. (zero velocity + height updating).

To compare the result quantitatively, RMS (root mean square) position errors are computed. The x axis RMS position error is defined by:

$$E_{RMS,x} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\hat{r}_{x,k} - \hat{r}_{optical,x,k})^2}$$

where $\hat{r}_{x,k}$ is the x axis estimated position at the discrete time k, $\hat{r}_{optical,x,k}$ is the x axis position measured by the optical tracker and N is the total number of data. Similarly, we can define $E_{RMS,y}$ and $E_{RMS,z}$

The RMS error results for four different activities are given in Table 2. In general, there is no significant improvement in the x and y axis position estimation.

Table 2. RMS position error comparison of different methods (unit: m).

Motion Type	**Estimation Method**	$E_{RMS,x}$	$E_{RMS,y}$	$E_{RMS,z}$	**Sum**
walking	proposed method	0.0208	0.0118	0.0030	0.0356
	K.F. (zero velocity)	0.0243	0.0131	0.0121	0.0495
	K.F. (zero velocity + height updating)	0.0247	0.0130	0.0095	0.0472
dancing steps	proposed method	0.0231	0.0798	0.0084	0.1113
	K.F. (zero velocity)	0.0265	0.0805	0.0397	0.1466
	K.F. (zero velocity + height updating)	0.0269	0.0810	0.0266	0.1345
ball kicking	proposed method	0.0816	0.3513	0.0120	0.4449
	K.F. (zero velocity)	0.0871	0.3551	0.0714	0.5136
	K.F. (zero velocity + height updating)	0.0877	0.3583	0.0685	0.5145
jumping	proposed method	0.0529	0.0705	0.0188	0.1423
	K.F. (zero velocity)	0.0630	0.0807	0.0313	0.1750
	K.F. (zero velocity + height updating)	0.0637	0.0808	0.0244	0.1689

Once the calibration parameters are computed, the dimension of the Kalman filter state reduces to nine from 21, since $\bar{r}_A \sim \bar{n}_B$ can be removed from the state. The proposed algorithm is used with this nine-state Kalman filter. In one Kalman filter, the initial calibration parameters in Table 1 are used. In another Kalman filter, the estimated calibration parameters in Equation (24) (which are supposed to be more accurate) are used. The position estimation results of two nine-state Kalman filters are given in Table 3, where the result of the 21-state Kalman filter is repeated from Table 2 for an easy comparison. The nine-state Kalman filter with calibrated parameters gives slightly better results, except for the kicking case. Since the difference between the initial and calibrated parameters is small, the RMS error difference is also small. The calibration of parameters is a one time process, and after the calibration, we can estimate the position using the nine-state Kalman filter.

In the walking data, the calibration parameters are estimated in the Kalman filter. The calibration parameters from walking data are given by:

$$\hat{r}_A = \begin{bmatrix} -0.0124 \\ 0.0806 \\ 0.000 \end{bmatrix}, \quad \hat{r}_B = \begin{bmatrix} -0.0033 \\ -0.0611 \\ -0.0006 \end{bmatrix}, \quad \hat{n}_A = \begin{bmatrix} 1.0000 \\ 0.0001 \\ 0.0000 \end{bmatrix}, \quad \hat{n}_B = \begin{bmatrix} 1.0000 \\ 0.0002 \\ -0.0001 \end{bmatrix} \tag{24}$$

Table 3. RMS position error comparison of different calibration parameters (unit: m).

Motion Type	Estimation Method	$E_{RMS,x}$	$E_{RMS,y}$	$E_{RMS,z}$	Sum
walking	proposed method	0.0208	0.0118	0.0030	0.0356
	with fixed initial parameter	0.0244	0.0119	0.0034	0.0397
	with fixed estimated parameter	0.0201	0.0119	0.0029	0.0350
dancing steps	proposed method	0.0231	0.0798	0.0084	0.1113
	with fixed initial parameter	0.0362	0.0852	0.0089	0.1303
	with fixed estimated parameter	0.0196	0.0745	0.0083	0.1024
ball kicking	proposed method	0.0816	0.3513	0.0120	0.4449
	with fixed initial parameter	0.0764	0.3495	0.0120	0.4380
	with fixed estimated parameter	0.0853	0.3506	0.0120	0.4479
jumping	proposed method	0.0529	0.0705	0.0188	0.1423
	with fixed initial parameter	0.0630	0.0807	0.0313	0.1750
	with fixed estimated parameter	0.0486	0.0655	0.0188	0.1329

6. Conclusions

There are many inertial sensor-based foot pose estimation algorithms. In this paper, the position estimation is improved additionally using distance sensors. The distance sensor has a 20-cm range limitation. However, this limitation does not degrade estimation accuracy much, since a foot is over the 20-cm range only for a short time, unless the foot is in the air intentionally. Longer range sensors exist, but they are either less accurate in short ranges or more expensive.

The inertial motion estimation algorithm is proposed using a 21-state Kalman (nine states for pose tracking and 12 states for calibration parameters). After the parameters are calibrated, the nine-state Kalman filter can be used to estimate the foot pose.

The proposed algorithm is tested for four activities: walking, dancing steps, ball kicking and jumping. It is shown in Table 2 that there is significant improvement in the z axis position estimation compared with inertial sensor-only foot pose estimation. On the other hand, there is no improvement in the x and y axis, since the distance sensor only gives the height information.

The proposed algorithm can be used in gait analysis (which requires a foot pose estimation), motion-based gaming (such as a soccer game) and exergaming (game-based exercises).

Acknowledgments

This work was supported by the 2015 Research Fund of University of Ulsan.

Author Contributions

Young Soo Suh contributed to main algorithm design. Pham Duy Duong implemented the algorithm in Matlab and did all the experiments. Both authors wrote the paper together.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Sabatini, A.M.; Martelloni, C.; Scapellato, S.; Cavallo, F. Assessment of walking features from foot inertial sensing. *IEEE Trans. Biomed. Eng.* **2005**, *52*, 486–494.
2. Tao, W.; Liu, T.; Zheng, R.; Feng, H. Gait analysis using wearable sensors. *Sensors* **2012**, *12*, 2255–2283.
3. Suh, Y.S. Inertial sensor-based smoother for gait analysis. *Sensors* **2014**, *14*, 24338–24357.
4. Maillot, P.; Perrot, A.; Hartley, A.; Do, M.C. The braking force in walking: Age-related differences and improvement in older adults with exergame training. *J. Aging Phys. Act.* **2014**, *22*, 518–526.
5. Alshurafa, N.; Xu, W.; Liu, J.J.; Huang, M.C.; Mortazavi, B.; Roberts, C.K.; Sarrafzadeh, M. Designing a robust activity recognition framework for health and exergaming using wearable sensors. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1636–1646.
6. Foxlin, E. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comput. Gr. Appl.* **2005**, *25*, 38–46.
7. Titterton, D.H.; Weston, J.L. *Strapdown Inertial Navigation Technology*; Peter Peregrinus Ltd.: London, UK, 1997.
8. Park, S.K.; Suh, Y.S. A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems. *Sensors* **2010**, *10*, 9163–9178.
9. Xu, Z.; Wei, J.; Zhang, B.; Yang, W. A robust method to detect zero velocity for improved 3D personal navigation using inertial sensors. *Sensors* **2015**, *4*, 7708–7727.
10. Colomar, D.; Nilsson, J.; Handel, P. Smoothing for ZUPT-aided INSs. In Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation, Sydney, Australia, 13–15 November 2012; pp. 1–5.

11. Bebek, O.; Suster, M.A.; Rajgopal, S.; Fu, M.J.; Xuemei, H.; Cauvusoglu, M.C.; Young, D.J.; Mehregany, M.; van den Bogert, A.J.; Mastrangelo, C.H. Personal navigation via high-resolution gait-corrected inertial measurement units. *IEEE Trans. Instrum. Meas.* **2010**, *59*, 3018–3027.
12. Girard, G.; Cote, S.; Zlatanova, S.; Barette, Y.; St-Pierre, J.; Van Oosterom, P. Indoor pedestrian navigation using foot-mounted IMU and portable ultrasound range sensors. *Sensors* **2011**, *11*, 7606–7624.
13. Do, T.N.; Suh, Y.S. Gait analysis using floor markers and inertial sensors. *Sensors* **2012**, *12*, 1594–1611.
14. Hung, T.N.; Suh, Y.S. Inertial sensor-based two feet motion tracking for gait analysis. *Sensors* **2013**, *13*, 5614–5629.
15. STMicroelectronics. VL6180x (Proximity and Ambient Light Sensing (ALS) Module) Datasheet. Available online: http://www.st.com/web/en/resource/technical/document/datasheet/DM00112632.pdf (accessed on 6 June 2015).
16. Kuipers, J.B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*; Princeton University Press: Princeton, NJ, USA, 1999.
17. Nam, C.N.K.; Kang, H.J.; Suh, Y.S. Golf swing motion tracking using inertial sensors and a stereo camera. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 943–952.
18. Savage, P.G. Strapdown inertial navigation integration algorithm design Part 1: Attitude algorithms. *J. Guid. Control Dyn.* **1998**, *21*, 19–28.
19. Markley, F.L. Multiplicative vs. additive filtering for spacecraft attitude determination. In Proceedings of 6th Cranfield Conference on Dynamics and Control of Systems and Structures in Space, Baltimore, MD, USA, 8 September 2004; pp. 467–474.
20. Suh, Y.S.; Park, S. Pedestrian inertial navigation with gait phase detection assisted zero velocity updating. In Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, 2009; pp. 336–341.
21. Maley, J.M. *Multiplicative Quaternion Extended Kalman Filtering for Nonspinning Guided Projectiles*; Army Research Laboratory, Adelphi, MD, USA, 2013.
22. Hawkinson, W.; Samanant, P.; McCroskey, R.; Ingvalson, R.; Kulkarni, A.; Haas, L.; English, B. GLANSER: Geospatial location, accountability, and navigation system for emergency responders. In Proceedings of Position Location and Navigation Symposium (PLANS), Myrtle Beach, SC, USA, 23–26 April 2012; pp. 98–105.
23. Skog, I.; Nilsson, J.O.; Handel, P. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In Proceedings of International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010; pp. 1–6.