## Article

# Gait Analysis Using Floor Markers and Inertial Sensors 

Tri Nhut Do and Young Soo Suh *<br>School of Electrical Engineering, University of Ulsan, Mugeo 2-Dong, Nam gu, Ulsan City 680-749, Korea; E-Mail: trinhutdo@gmail.com

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

Received: 1 December 2011; in revised form: 31 January 2012 / Accepted: 31 January 2012 /
Published: 7 February 2012


#### Abstract

In this paper, a gait analysis system which estimates step length and foot angles is proposed. A measurement unit, which consists of a camera and inertial sensors, is installed on a shoe. When the foot touches the floor, markers are recognized by the camera to obtain the current position and attitude. A simple planar marker with 4,096 different codes is used. These markers printed on paper are placed on the floor. When the foot is moving off the floor, the position and attitude are estimated using an inertial navigation algorithm. For accurate estimation, a smoother is proposed, where vision information and inertial sensor data are combined. Through experiments, it is shown that the proposed system can both track foot motion and estimate step length.


Keywords: gait analysis; image processing; inertial sensors; position estimation

## 1. Introduction

Gait analysis [1] involves the measurement of temporal/spatial characteristics (step length and walking speed), kinematics and kinetics. Gait analysis is used for medical purposes, sport analysis, and also for entertainment purposes. An instrumented walkway, where pressure sensors are located on the floor, can be used to measure step length and foot pressure [1]. A vision-based motion tracking system can measure step length and foot angles accurately [2]. However, both systems have rather limited walking ranges (usually less than 10 m ).

Recently, inertial sensor based systems are becoming popular. In [3], inertial sensors are installed on a leg and gait phases are identified by computing angles of leg segments. In [4], inertial sensors are
used to estimate upper-limb orientation. More relevant results are given in [5,6], where inertial sensors are installed on a shoe or a leg and the foot movement is estimated using inertial navigation algorithms. Step length is estimated in [5] and a foot movement is estimated in [6]. These techniques can estimate step length or foot movement without range limitations. However, due to double integration errors, the accuracy tends to degrade as time goes by.

In this paper, a new gait analysis system is proposed. A measurement system consisting of a camera and inertial sensors is installed on a shoe. Fiducial markers are printed on paper and placed on the floor. Fiducial markers are used mainly in virtual reality systems and there are many types of such markers [7-9]. In this paper, we use simple markers since there are no concerns that complicated backgrounds might mistakenly be recognized as markers.

When a foot is on the floor, the position and attitude are estimated using a camera by recognizing the markers. When a foot is moving, its movement is estimated using inertial sensors. In an inertial-based foot estimation system, Kalman filters are usually used to estimate the position and attitude of the foot. For better accuracy, a smoother is used to combine vision information and inertial sensor data. It will be shown that the accuracy of the proposed system is better than that of inertial sensor-only estimation as described in [5].

One possible application of the proposed system is the clinical gait assessment of patients, where long walking ranges are desirable. Another application is parameter estimation of some pedestrian navigation algorithms [10]. In those algorithms, relationship between step length and one step walking time needs to be identified; thus step length and one step walking time should be accurately measured. We note that step length measurement using vision only [11] and partial combinations of vision and inertial sensors are reported in [12]. In this paper, vision and inertial sensor data are tightly coupled in the proposed algorithm. The proposed system can be applied only for the flat ground. For example, the proposed system cannot be used for stairs.

The paper is organized as follows. In Section 2, an overview of the proposed system is given. In Section 3, how to estimate the position and attitude is explained. In Section 4, an inertial navigation algorithm combining inertial sensor data and vision data is described. A smoothing algorithm [13] consisting of a forward filter and a backward filter is used. In Section 5, experiment results are given. Conclusions are presented in Section 6.

## 2. Gait Analysis System

The gait analysis system consists of a sensor unit placed on a shoe and fiducial markers located on the floor (Figure 1). A sensor unit consists of a camera (point grey Firefly MV FFMV-03MTM) and inertial sensors (XSens MTi inertial measurement unit).

A fiducial marker used in this paper is similar to ARTag markers [7-9]. Each marker is a nine by nine grid of quadrilateral cells of 1.5 millimeter edges and thus the marker size is $13.5 \mathrm{~mm} \times 13.5 \mathrm{~mm}$. The intra-marker space is 4.5 mm . The marker size is determined so that there are at least two markers in the camera image. We note that the height of a camera from the markers is about 85 cm and camera field of view is about $54.5 \times 45 \mathrm{~mm}$.

Figure 1. Gait Analysis System.


The whole cells are blank or colored in black in order to make them bitonal cells allowing either gray scale or color cameras to be used. The cell in black represent 1s, while the cells in white represent 0 s (Figure 2). The border cells are colored in black and are one cell width wide, followed by blank closed-chain cells. The digital coding system used to identify the marker consists of a five by five grid of bitonal cells including the blank central-cross cells, a black origin cell and three blank corner cells in the interior of a marker. A marker structure and its twelve bits digital coding system inside are illustrated in Figure 2.

Figure 2. Fiducial Marker Structure (a) and a marker ID sample (b).


The code bits are distributed along the quadrilateral central-cross in the order from left to right, and from top to bottom (the bit corresponding to 0 position in Figure 2(a) is the least significant bit and 11 is the most significant bit). The origin quadrilateral cell and three blank corner cells are used to detect the least significant bit regardless of the relative orientation between a camera and a marker. For example, the marker ID in Figure 2(b) is 000110101000 in a binary number and 424 in a decimal number.

Usually, the digital coding system is encoded in the marker containing a marker ID, checksum and error correcting codes. However, in our application, markers are seen by the camera in predetermined sequences not random sequences as in other virtual reality applications. Thus, in order to increase the recognition speed and number of markers, a simple coding system is used. Neither checksum nor error correcting codes is used. Totally, there are $2^{12}=4096$ unique markers (marker ID 0~4095). With 4096 unique markers, the total length is 16.9 m . If longer walking range is needed, markers can be repeated so that marker ID 0 reappears after the marker ID 4095.

A planar landmark system is generated by $N$ fiducial markers which are composed of a four by M grid of markers as in Figure 3. The marker ID is encoded from zero to ( $N-1$ ).

Figure 3. Planar Landmark System.


There are four coordinate frames in this paper. The world coordinate frame is based on the marker plane: $x$ and $y$ axes of the world coordinate frame constitute the plane where markers are placed. The origin of the world coordinate frame coincides with the center of the first marker. Foot positions are expressed in the world coordinate frame.

The navigation coordinate frame is used in an inertial navigation algorithm. The origin of the navigation coordinate frame is the same as that of the world coordinate frame. The $z$ axis of the navigation coordinate frame coincides with the local vertical. Unlike a usual navigation coordinate frame (where the $x$ axis is in the direction of the magnetic north), the $x$ axis lies on the $x-z$ plane of the world coordinate frame. If markers are placed on the perfect plane, the inclination angle of the floor is zero and two coordinate frames are the same.

The camera coordinate frame is placed at the pinhole of the camera, where the $z$ axis is perpendicular to the image plane. Three axes of the body coordinate frame coincide with those of inertial sensors. In this paper, it is assumed that the three axes of the camera coordinate frame and the body coordinate frame are the same.

One walking step is illustrated in Figure 4, where normal walking is considered. When a person is walking, a foot touches the ground for a short time (usually about $0.1 \sim 0.3$ seconds) almost periodically. Even if a shoe looks like it is moving all the time during walking, a shoe is on the ground and not moving for a short time. This short interval is called the zero velocity interval. We also define moving intervals, which refer to an interval when a foot is moving. Thus one normal walking step consists of a zero velocity interval and a moving interval (see Figure 4). How to divide walking steps into a zero velocity interval and moving interval is given in Figure 4. In medical gait analysis [1], one walking step consists of seven gait phases: loading response, mid-stance, terminal stance, pre-swing, initial swing, mid-swing, terminal swing. The zero velocity interval exists between the loading response phase and terminal stance phase.

Figure 4. A Zero velocity interval and a moving interval of a walking step.


During zero velocity intervals, the position and attitude of a foot are estimated using markers on the floor (see Section 3). During moving intervals, the position and attitude are estimated using inertial sensors (see Section 4).

## 3. Position and Attitude Estimation Using Markers

When a foot is not moving on the floor, markers in camera images are recognized through an image processing process. The Canny algorithm, the most popular algorithm for edge detection, is applied to detect whole contours from the original captured image (Figure 5(a,b)). The quadrilateral contours are then detected based on the geometric properties (Figure 5(c)). Other contours are eliminated. Two quadrilateral contours whose central coordinates are close to each other are grouped. A line equation is computed for each side of a quadrilateral using the least squares method. For each marker, eight lines are computed (four lines for an outer quadrilateral and four lines for an inner quadrilateral). Based on these lines, the inner quadrilateral is partitioned into $7 \times 7$ partitions to check the interior cells (see Figure 2). The interior cells of grouped contours are converted into a binary coding using the following adaptive threshold:

$$
\text { threshold }=\frac{1}{640 \times 480}\left(\sum_{i=1}^{480}\left(\sum_{j=1}^{640} I_{i j}\right)\right)
$$

where $I_{i j}$ is the gray image intensity level of the $(i, j)$ pixel. The image size is $640 \times 480$. If a binary coding satisfies the constraint in Figure 2, the group is identified as a marker and a marker ID is computed. If the constraint is not satisfied, the group is abandoned. From the origin cell (see Figure 2), the orientation of a marker can be identified. In Figure 5(d), identified markers with their IDs and their four corners are given, where intersections of four lines of an outer quadrilateral are defined as four corners of a marker.

Figure 5. Image processing steps including (a) Origin gray scale image; (b) Detected contours using Canny edge detector; (c) Quadrilateral contours; and (d) Marker IDs and four corners.


Position ( $\hat{r}_{\text {vision }}$ ) and attitude ( $\hat{C}_{\text {vision }}$ ) of a camera with respect to the world coordinate frame are determined using four corners of a marker (outer corners in Figure 5(d)). Let $r_{w}$ be a point expressed in the world coordinate and $r_{b}$ be the same point in the body coordinate. Then the relationship between $r_{b}$ and $r_{w}$ is given by:

$$
r_{w}=\hat{C}_{\text {vision }} r_{b}+\hat{r}_{\text {vision }} .
$$

A vector $\hat{r}_{\text {vision }}$ represents the position of a camera in the world coordinate. A rotation matrix $\hat{C}_{\text {vision }}$ represents a rotation matrix transforming a body coordinate (=camera coordinate) into a world coordinate.

It is known that the position and attitude can be computed if there are at least four points [14,15]. Thus position and attitude can be computed if there is at least one marker with four corners in the image. If there is more than one marker, more points can be used for the position and attitude computation; then the estimation error becomes smaller. We used an algorithm in [14] to estimate position $\hat{r}_{\text {vision }}$ and attitude $\hat{C}_{\text {vision }}$. Position information is directly used in Section 4. Only yaw information in $\hat{C}_{\text {vision }}$ is used in Section 4 since pitch and roll information can be computed from inertial sensors.

## 4. Position and Attitude Estimation Using Inertial Sensor Data

When a foot is moving, the position and attitude are estimated using an inertial navigation algorithm. To simplify the inertial navigation algorithm, it is assumed that the navigation coordinate frame and the world coordinate frame are the same. This assumption is satisfied if markers are placed on a completely flat floor.

A quaternion $q=\left[\begin{array}{llll}q_{0} & q_{1} & q_{2} & q_{3}\end{array}\right]^{T} \in R^{4}$ is used to represent rotation between the navigation coordinate frame and the body coordinate frame [16,17]. The rotation matrix $C(q)$ corresponding to quaternion $q$ is defined by:

$$
C(q)=\left[\begin{array}{ccc}
2 q_{0}^{2}+2 q_{1}^{2}-1 & 2 q_{1} q_{2}+2 q_{0} q_{3} & 2 q_{1} q_{3}-2 q_{0} q_{2} \\
2 q_{1} q_{2}-2 q_{0} q_{3} & 2 q_{0}^{2}+2 q_{2}^{2}-1 & 2 q_{2} q_{3}+2 q_{0} q_{1} \\
2 q_{1} q_{3}+2 q_{0} q_{2} & 2 q_{2} q_{3}-2 q_{0} q_{1} & 2 q_{0}^{2}+2 q_{3}^{2}-1
\end{array}\right]
$$

Gyroscope output ( $y_{g}$ ) and accelerometer output ( $y_{a}$ ) are given by:

$$
\begin{aligned}
& y_{g}=\omega+v_{g} \\
& y_{a}=C(q) \tilde{g}+a_{b}+v_{a}
\end{aligned}
$$

where $\omega$ is the body angular rate, $\tilde{g}$ is the gravitational acceleration vector, $a_{b}$ is the acceleration due to movement, $v_{g}$ is the gyroscope measurement noise, and $v_{a}$ is the accelerometer measurement noise. The sensor noises $v_{g}$ and $v_{a}$ are assumed to be zero mean white Gaussian noises satisfying:

$$
\mathrm{E}\left[v_{g}(t) v_{g}(s)^{T}\right]=\lambda_{g} I \delta(t-s), \quad \mathrm{E}\left[v_{a}(t) v_{a}(s)^{T}\right]=\lambda_{a} I \delta(t-s), \quad \mathrm{E}\left[v_{a}(s) v_{g}(t)^{T}\right]=0
$$

The sampling frequency of inertial sensors is 100 Hz and the sampling frequency of vision data is 30 Hz . A discrete time system is based on the sampling period of inertial sensors; that is, the sampling period $T$ for the discrete time system is $T=0.01 \mathrm{~s}$. For example, $y_{g, k}$ in the discrete time means $y_{g}(k T)$ in the continuous time. Vision data is not synchronized with inertial data. When vision data is available at the continuous time $t$, its discrete time index is computed by $k=$ floor $(t / T)$ where floor $(U)$ is the largest integer not larger than $U$.

### 4.1. Zero Velocity Interval Detection

Since the velocity of a foot cannot be measured directly, the zero velocity intervals are determined from inertial sensor data and vision data. To determine whether a discrete time $k$ belongs to a zero velocity interval, firstly the following condition are tested:

$$
\begin{array}{rr}
\left\|y_{a, i}-y_{a, i-1}\right\|<S_{a}, & k-\frac{N_{a}}{2} \leq i \leq k+\frac{N_{a}}{2} \\
\left\|y_{g, i}\right\|<S_{g}, & k-\frac{N_{g}}{2} \leq i \leq k+\frac{N_{g}}{2} \tag{1}
\end{array}
$$

where $S_{a}, S_{g}, N_{a}$ and $N_{g}$ are constant parameters. The first condition means that difference between consecutive accelerometer values should be small. The second condition means that gyroscope values should be small. In addition to Equation (1), there is an additional condition for zero velocity determination, which requires difference between two consecutive images should be small. Recall that vision data are not available for each discrete time $k$ since the sampling frequency of a camera is lower than that of inertial sensor data. Here vision data means that the image coordinates of marker corners. Difference between two images is measured by computing the average movement of image coordinates of marker corners.

Given a discrete time $k$, let $k_{1} \leq k$ and $k_{2}>k$ be discrete time indices at which time vision data are available. That is, vision data are available at $k_{1}$ and $k_{2}$ and there are no vision data in $\left(k_{1}, k_{2}\right)$. Let $p_{i, k_{1}} \in R^{2}$ and $p_{i, k_{2}} \in R^{2}\left(1 \leq i \leq N_{k}\right)$ be the image coordinates of marker corners, which exist both at $k_{1}$ and $k_{2}$ discrete time images. The same index $i$ represents the same corner and $N_{k}$ is the number of marker corners appearing in the both images. The small image movement is defined as follows:

$$
\begin{equation*}
\frac{1}{N_{k}} \sum_{i=1}^{N_{k}}\left\|p_{i, k_{1}-} p_{i, k_{2}}\right\|<S_{\text {image }} \tag{2}
\end{equation*}
$$

In summary, discrete time $k$ is determined to belong to a zero velocity interval if Equations (1) and (2) are satisfied.

### 4.2. Initialization of an Inertial Navigation Algorithm

We use the position estimate in Section 3 as an initial position estimate. For an initial attitude estimate, we could have used the attitude estimate ( $\hat{C}_{\text {vision }}$ ) from vision data. Instead we combine the attitude estimate ( $\hat{C}_{v i s i o n}$ ) and accelerometer output $\left(y_{a}\right)$, where pitch and roll are estimated using $y_{a}$ and yaw is estimated from $\hat{C}_{\text {vision }}$. In this way, we can obtain a correct attitude estimate even if the floor is not completely flat and thus the navigation and world coordinate frames are not exactly the same.

An initial attitude estimate is computed by finding $C(q)$ satisfying the following:

$$
\begin{gather*}
y_{a}=C(q) \tilde{g}  \tag{3}\\
\hat{C}_{v i s i o n}\left[\begin{array}{l}
1 \\
0 \\
0
\end{array}\right]=C(q)\left[\begin{array}{l}
1 \\
0 \\
0
\end{array}\right] \tag{4}
\end{gather*}
$$

The TRIAD algorithm [18] is used to compute $C(q)$. We note that in the TRIAD algorithm pitch and roll are estimated using Equation (3) and yaw is estimated using Equation (4). Thus $\hat{C}_{\text {vision }}$ only affects yaw in $C(q)$. We have not used magnetic sensors since there could be large magnetic disturbances indoors, which degrades the accuracy of yaw estimation.

### 4.3. Basic Equations for an Inertial Navigation Algorithm

Let $v_{n}$ be the velocity of a foot in the navigation coordinate frame. Basic equations for $q, v_{n}$ and $r_{n}$ are given by:

$$
\begin{align*}
\dot{q} & =\frac{1}{2} \Omega(\omega) q \\
\dot{v}_{n} & =C(q)^{T} a_{b}  \tag{5}\\
\dot{r}_{n} & =v_{n}
\end{align*}
$$

where $\Omega(\omega)$ is defined by:

$$
\Omega(\omega) @\left[\begin{array}{cccc}
0 & -\omega_{1} & -\omega_{2} & -\omega_{3} \\
\omega_{1} & 0 & \omega_{3} & -\omega_{2} \\
\omega_{2} & -\omega_{3} & 0 & \omega_{1} \\
\omega_{3} & \omega_{2} & -\omega_{1} & 0
\end{array}\right]
$$

In this paper, we estimate $q, v_{n}$ and $r_{n}$ by combining a forward filter and a backward filter [19]. An indirect filter is used for a forward and a backward filter. Firstly $q, v_{n}$ and $r_{n}$ are estimated using Equation (5); let $\hat{q}, \hat{v}_{n}$ and $\hat{r}_{n}$ be corresponding estimated values. Let $\tilde{q}_{e}, v_{e}$ and $r_{e}$ be errors of these estimated values, which are defined by:

$$
\begin{aligned}
& \tilde{q}_{e}=\hat{q}^{*} \otimes q \\
& v_{e}=v_{n}-\hat{v}_{n} \\
& r_{e}=r_{n}-\hat{r}_{n}
\end{aligned}
$$

where $\otimes$ represents the quaternion multiplication. Assuming that $\tilde{q}_{e}$ is small, we can approximate $\tilde{q}_{e}$ as follows:

$$
\tilde{q}_{e}=\left[\begin{array}{c}
1 \\
q_{e}
\end{array}\right]
$$

where $q_{e} \in R^{3}$. In an indirect filter, $q_{e}, v_{e}$ and $r_{e}$ are estimated. Let a state $x$ be defined by:

$$
x=\left[\begin{array}{l}
q_{e} \\
v_{e} \\
r_{e}
\end{array}\right] \in R^{9}
$$

The differential equation is given by:

$$
\begin{equation*}
\dot{x}(t)=A(t) x(t)+w(t) \tag{6}
\end{equation*}
$$

where:

$$
A=\left[\begin{array}{ccc}
-\left[y_{g} \times\right] & 0 & 0 \\
-2 C(\hat{q})^{T}\left[y_{a} \times\right] & 0 & 0 \\
0 & I & 0
\end{array}\right], \quad w=\left[\begin{array}{c}
-0.5 v_{g} \\
-C(\hat{q})^{T} \eta_{a} \\
0
\end{array}\right] .
$$

For a vector $p \in R^{3},[p \times]$ is defined by:

$$
p=\left[\begin{array}{ccc}
0 & -p_{3} & -p_{2} \\
p_{3} & 0 & -p_{1} \\
-p_{2} & p_{1} & 0
\end{array}\right] .
$$

Note that $\mathrm{E}\left[w(t) w(t)^{T}\right]=Q \delta(0)$, where:

$$
Q=\left[\begin{array}{ccc}
0.25 \lambda_{g} I & 0 & 0 \\
0 & \lambda_{a} I & 0 \\
0 & 0 & 0
\end{array}\right] .
$$

### 4.4. Forward Filter

A forward filter processes data from the beginning to the end of data. In a forward filter, $\hat{q}_{f, k}, \hat{v}_{f, k}$ and $\hat{r}_{f, k}$ are first computed using the discretized approximation of Equation (5): subscript $f$ represents a forward filter and $k$ represents the discrete time index. The following equations are used to compute $\hat{q}_{f, k}, \hat{v}_{f, k}$ and $\hat{r}_{f, k}$ :

$$
\begin{gathered}
\hat{q}_{f, k+1}=\left(I+\frac{3}{4} \Omega_{k} T-\frac{1}{4} \Omega_{k-1} T-\frac{1}{6}\left\|y_{g, k}\right\|_{2}^{2} T^{2}-\frac{1}{24} \Omega_{k} \Omega_{k-1} T^{2}-\frac{1}{48}\left\|y_{g, k}\right\|_{2}^{2} \Omega_{k} T^{3}\right) \hat{q}_{f, k} \\
\hat{v}_{f, k+1}=\hat{v}_{f, k}+0.5 T\left(C\left(\hat{q}_{f, k+1}\right)^{T} y_{a, k+1}+C\left(\hat{q}_{f, k}\right)^{T} y_{a, k}\right)-T \widetilde{g} \\
\hat{r}_{f, k+1}=\hat{r}_{f, k}+0.5 T\left(\hat{v}_{f, k+1}+\hat{v}_{f, k}\right)
\end{gathered}
$$

where $\Omega_{k}$ is defined by $\Omega_{k}=\Omega\left(y_{g, k}\right)$. In the computation, $\hat{q}_{f, k}$ is normalized each time.
The estimation errors in $\hat{q}_{f, k}, \hat{v}_{f, k}$ and $\hat{r}_{f, k}$ are estimated using a forward Kalman filter, where the process model is given in Equation (6). During the zero velocity interval, the fact $v_{n, k}=0$ is used in the measurement equation. If vision data are also available, $\hat{r}_{\text {vision }, k}$ and $\hat{C}_{\text {vision }}$ are used in the measurement update. For example, a measurement equation using $v_{n, k}=0$ and $\hat{r}_{\text {vision }, k}$ is given by:

$$
z_{k}=\left[\begin{array}{c}
0-\hat{v}_{f, k} \\
\hat{r}_{\text {vision }, k}-\hat{r}_{f, k}
\end{array}\right]=\left[\begin{array}{ccc}
0 & I & 0 \\
0 & 0 & I
\end{array}\right] x_{k}+v_{k} .
$$

The measurement noise $v_{k}$ is assumed to satisfy:

$$
\mathrm{E}\left[v_{k} v_{k}^{T}\right]=\left[\begin{array}{cc}
r_{v} I & 0 \\
0 & r_{r} I
\end{array}\right]
$$

where $r_{v}$ and $r_{r}$ are scalar constants. Similarly to the initialization process, only yaw information in $\hat{C}_{\text {vision }}$ is used. We used the same technique in [20] for yaw updating. In [20], magnetic sensors are used for yaw updating. Replacing the magnetic sensor outputs with $\hat{C}_{\text {vision }}\left[\begin{array}{lll}1 & 0 & 0\end{array}\right]^{T}$, we can obtain a yaw updating algorithm.

The error covariance of the forward Kalman filter is denoted by $P_{f, k}$ :

$$
P_{f, k}=\mathrm{E}\left[\left(x_{f, k}-\hat{x}_{f, k}\right)\left(x_{f, k}-\hat{x}_{f, k}\right)^{T}\right] \in R^{9 \times 9} .
$$

### 4.5. Backward Filter

A backward filter processes data from the end of data to the beginning. Similarly to the forward filter, $\hat{q}_{b, k}, \hat{v}_{b, k}$ and $\hat{r}_{b, k}$ are first computed (subscript $b$ represents a backward filter) :

$$
\begin{aligned}
& \hat{q}_{b, k}=\left(I-\frac{3}{4} \Omega_{k} T+\frac{1}{4} \Omega_{k+1} T-\frac{1}{6}\left\|y_{g, k}\right\|_{2}^{2} T^{2}-\frac{1}{24} \Omega_{k} \Omega_{k+1} T^{2}+\frac{1}{48}\left\|y_{g, k}\right\|_{2}^{2} \Omega_{k} T^{3}\right) \hat{q}_{b, k+1} \\
& \hat{v}_{b, k}=\hat{v}_{b, k+1}+0.5 T\left(C\left(\hat{q}_{b, k+1}\right)^{T} y_{a, k+1}+C\left(\hat{q}_{b, k}\right)^{T} y_{a, k}\right)-T \tilde{g} \\
& \hat{r}_{b, k}=\hat{r}_{b, k+1}+0.5 T\left(\hat{v}_{b, k+1}+\hat{v}_{b, k}\right)
\end{aligned}
$$

Their errors are estimated in the backward filter, where the state space model is given by:

$$
\dot{x}_{b}=-A x_{b}-w
$$

where the state definition of $x_{b}$ is the same as that of $x_{f}$.
The measurement equations are the same as those in the forward filter. The error covariance of the backward Kalman filter is denoted by $P_{b, k}$ :

$$
P_{b, k}=\mathrm{E}\left[\left(x_{b, k}-\hat{x}_{b, k}\right)\left(x_{b, k}-\hat{x}_{b, k}\right)^{T}\right] \in R^{9 \times 9}
$$

### 4.6. Smoother

The smoother combines the forward filter output and the backward filter output. Due to an indirect filter structure, it is difficult to derive an optimal smoother. In this paper, we use a suboptimal smoother, which only uses diagonal matrices of $P_{f, k}$ and $P_{b, k}$.

Diagonal matrices of $P_{f, k}$ and $P_{b, k}$ are given by:

$$
P_{f, k}=\left[\begin{array}{ccc}
P_{q_{f}, k} & * & * \\
* & P_{v_{f}, k} & * \\
* & * & P_{r_{f}, k}
\end{array}\right], P_{b, k}=\left[\begin{array}{ccc}
P_{q_{b}, k} & * & * \\
* & P_{v_{b}, k} & * \\
* & * & P_{r_{b}, k}
\end{array}\right]
$$

where * parts denote the off-diagonal matrices and not used in the smoother. The smoothed quaternion is computed by using the quaternion averaging algorithm in [21]. The smoothed quaternion $q_{s}$ is computed by solving the following optimization problem:

$$
\hat{q}_{s, k}=\min _{q} q^{T}\left(\Xi\left(q_{f, k}\right)^{T} P_{q_{f}, k}{ }^{-1} \Xi\left(q_{f, k}\right)+\Xi\left(q_{b, k}\right)^{T} P_{q_{b}, k}{ }^{-1} \Xi\left(q_{b, k}\right)\right) q
$$

where:

$$
\Xi(q)=\left[\begin{array}{cccc}
-q_{1} & q_{0} & -q_{3} & q_{2} \\
-q_{2} & q_{3} & q_{0} & -q_{1} \\
-q_{3} & -q_{2} & q_{1} & q_{0}
\end{array}\right]
$$

The smoothed velocity and position estimates ( $\hat{v}_{s}$ and $\hat{r}_{s}$ ) are given by:

$$
\begin{aligned}
& \hat{v}_{s, k}=\left(P_{v_{f}, k}{ }^{-1}+P_{v_{b}, k}{ }^{-1}\right)^{-1}\left(P_{v_{f}, k}{ }^{-1} \hat{v}_{f, k}+P_{v_{b}, k}{ }^{-1} \hat{v}_{b, k}\right) \\
& \hat{r}_{s, k}=\left(P_{r_{f}, k},{ }^{-1}+P_{r_{b}, k},-1\right)^{-1}\left(P_{r_{f}, k}{ }^{-1} \hat{r}_{f, k}+P_{r_{b}, k}{ }^{-1} \hat{r}_{b, k}\right) .
\end{aligned}
$$

## 5. Experimental Results

In this section, experimental results to verify the proposed method are given. The camera is calibrated to obtain its intrinsic parameters [22]. The inertial sensor unit is calibrated using the algorithm in [23]. The sampling rate of camera and inertial IMU sensors are 30 fps and 100 Hz , respectively. The computation is done off-line using Matlab. The thresholds used for zero velocity detection are given as follows:

$$
N_{a}=0.1, S_{a}=20, N_{g}=0.05, S_{g}=20, S_{\text {image }}=30
$$

where these thresholds are decided by trial and error. Various combinations of thresholds are tried and the above thresholds are chosen since they identify zero velocity intervals well. In this process, true zero velocity intervals are manually identified by inspecting inertial sensor data.

The initial values used for the indirect Kalman filter in the experiment are given as follows:

$$
\begin{gathered}
P_{q_{f}, 0}=P_{q_{b}, 0}=0.001 I, P_{v_{f}, 0}=P_{v_{b}, 0}=0.001 I, P_{r_{f}, 0}=P_{r_{b}, 0}=0.008 I \\
r_{v}=0.004, \mathrm{r}_{r}=0.008, R_{\text {yaw }}=0.01 I
\end{gathered}
$$

where $P_{q_{f}, 0}, \quad P_{v_{f}, 0}, \quad P_{r_{f}, 0}$ are the initial values of forward Kalman filter error covariance matrices, $P_{q_{b}, 0}, \quad P_{v_{b}, 0}, \quad P_{r_{b}, 0}$ are the initial values of backward Kalman filter error covariance matrices, respectively.

### 5.1. Table Experiment: Comparison with the Digitizer Output

A shoe is moved back and forth between two positions A and B several times along to the $Y_{w}$ axis direction as illustration in Figure 6. The movement is tracked using Microscribe G2X digitizer, whose output is considered as a true value. The accuracy of G2X model is up to 0.23 mm in a 1.27 m sphere workspace.

Figure 6. Table experiment ( $Y_{w}$ axis moving experiment).


Since the movement is mainly along the $Y_{w}$ axis, only the $Y_{w}$ axis velocity and position are plotted in Figure 7. The first graph of Figure 7 shows the estimated velocity using the forward filter, the backward filter and the smoother while the second graph shows the position estimates. The third graph shows the zero velocity detection results, where the value 1 indicates that the corresponding discrete time belong to a zero velocity interval. In the second graph, assuming the smoother estimates are accurate, we can see the error in the forward filter increases as the moving time increases. Also we can see the error in the backward filter increases as the moving time backwardly increases. In the final graph of Figure 7, $Y_{w}$ axis position estimated by the smoother, the vision, and the digitizer are given. The RMS difference between digitizer data and smoothed estimation is $4.8 \pm 9.1 \mathrm{~mm}$ (mean $\pm$ standard deviation).

Figure 7. Results of a table experiment ( $Y_{w}$ axis moving experiment).


### 5.2. Walking Experimental Results

A person wearing the shoe walked along a planar marker system path (Figure 1). The length of the path is 33.8 meters. We note that this length can be easily extended by using more markers. Euler angles of a shoe are given in Figure 8.

Figure 8. Floor experiment (Euler angles estimation).


Note that quaternion is used to represent attitude. Euler angles are transformed from quaternion estimates for visualization. In the attitude estimation, there are almost no differences between the forward and backward filter estimates. This is due to the fact that attitude errors are almost periodically reset during zero velocity intervals. Velocity and position estimation results are given in Figures 9 and 10.

Figure 9. Floor experiment (velocity estimation).


Figure 10. Floor experiment (position estimation).


Note that there are large differences between the forward and backward filters. The errors are probably due to inertial sensor errors (bias and scaling factor). We can see the velocity and position estimation errors can be reduced by using the smoother.

The position estimates from vision is compared with smoothed position estimates in Figure 11. Note that the position estimated from vision is mostly available during zero velocity intervals. During moving intervals, marker recognition becomes difficult due to motion related image blurring.

Figure 11. Floor experiment (Comparison of smoother estimated and vision estimated positions).


From Figure 10, step length can be computed. One walking step consists of a zero velocity interval and a moving interval (see Figure 4). The accuracy of the step length estimation is evaluated by one-step experiment using the ruler as a reference (see Figure 12). A marker pen is attached on the shoe. Its tip touches on the ruler when the shoe is on the ground. Step length measured by the ruler (by measuring two dots) is considered as a true value.

The results of 20 one-step experiments are listed in Table 1. The error between the measurement using a ruler and the smoothed estimation is in range $0.5 \sim 4.1 \mathrm{~mm}$. RMS difference and the worst case error are given in Table 2.

Table 1. Step length estimation (20 steps).

|  | Step \#1 | Step \#2 | Step \#3 | ... | $\begin{gathered} \hline \text { Step } \\ \# 18 \\ \hline \end{gathered}$ | $\begin{gathered} \hline \text { Step } \\ \# 19 \end{gathered}$ | $\begin{gathered} \text { Step } \\ \# 20 \end{gathered}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Measurement by a ruler[mm] | 792 | 804 | 841 | $\ldots$ | 779 | 800 | 814 |
| Estimation by a smoother[mm] | 793.6 | 802.2 | 839.6 | $\ldots$ | 778.5 | 797.8 | 810.0 |
| Error[mm] | 1.6 | 1.8 | 1.4 | $\ldots$ | 0.5 | 2.2 | 4.0 |

Table 2. Step length estimation error.

| Mean of step length error [mm] | 1.99 |
| :--- | :--- |
| Standard deviation of step length error [mm] | 1.25 |
| Maximum value of step length error [mm] | 4.10 |

Figure 12. One walking step length measurement.


The RMS error between the measurement by a ruler and the smoothed estimation is $1.99 \pm 1.25 \mathrm{~mm}$ (mean $\pm$ standard deviation). They are smaller than the one ( $4.8 \pm 9.1 \mathrm{~mm}$ ) in Section 5.1 since the step length is computed using estimates during zero velocity intervals. During zero velocity intervals, positions are compensated from the vision and thus position estimates are more accurate than those during moving intervals. Step length RMS error in [5] is $34.1 \pm 2.1 \mathrm{~mm}$, where an optical tracker output is used as a true value. The proposed system is more accurate since markers are used to compensate position and attitude errors.

## 6. Conclusions

A gait analysis system combining a reliable fiducial marker system on the floor and a inertial sensor unit was proposed. The system can track foot motion and measure step length on flat ground. The position errors tend to become larger during the moving intervals and smaller during the zero velocity intervals since vision data are used to reduce position and attitude errors. The step length RMS error is $1.99 \pm 1.25 \mathrm{~mm}$, which is smaller than that of an existing inertial sensor only system. Commercial optical trackers such as the Vicon unit is more accurate than the proposed system-they are usually sub-millimeter level, however, they have rather limited walking ranges. On the other hand, the walking range of the proposed system can be easily extended.

## Acknowledgments

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (No. 2011-0021978).

## References

1. Whittle, M.W. Gait analysis: An introduction, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2007; p. 255.
2. Wang, F.; Stone, E.; Dai, W.; Skubic, M.; Keller, J. Gait Analysis and Validation Using Voxel Data. In Proceedings of 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009), Minneapolis, MN, USA, 2-6 September 2009; pp. 6127-6130.
3. Liu, T.; Inoue, Y.; Shibata, K. Development of a wearable sensor system for quantitative gait analysis. Measurement 2009, 42, 978-988.
4. Hyde, R.A.; Ketteringham, L.P.; Neild, S.A.; Jones, R.J.S. Estimation of upper-limb orientation based on accelerometer and gyroscope measurements. IEEE Trans. Biomed. Eng. 2008, 55, 746-754.
5. Martin Schepers, H.; van Asseldonk, E.H.; Baten, C.T.; Veltink, P.H. Ambulatory estimation of foot placement during walking using inertial sensors. J. Biomech. 2010, 43, 3138-3143.
6. Sabatini, A.M. Quaternion-based strapdown integration method for applications of inertial sensing to gait analysis. Med. Biol. Eng. Comput. 2005, 43, 94-101.
7. Fiala, M. Artag Revision 1, a Fiducial Marker Sytem Using Digital Techniques; National Research Council Publication 47419/ERB-1117: Ottawa, Canada, 2004.
8. Fiala, M. Artag, a Fiducial Marker System Using Digital Techniques. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, USA, 20-25 June 2005; Volume 2, pp. 590-596.
9. Fiala, M. Comparing Artag and Artoolkit Plus Fiducial Marker Systems. In Proceedings of IEEE International Workshop on Haptic Audio Visual Environments and their Applications, HAVE 2005, Ottawa, ON, Canada, 1-2 October 2005; pp. 148-153.
10. Shin, S.H.; Park, C.G.; Kim, J.W.; Hong, H.S.; Lee, J.M. Adaptive Step Length Estimation Algorithm Using Low-Cost Mems Inertial Sensors. In Proceedings of SAS 2007-IEEE Sensors Applications Symposium, San Diego, CA, USA, 6-8 February 2007.
11. Do, T.-N.; Suh, Y.-S. Foot Motion Tracking Using Vision. In Proceedings of 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), Seoul, Korea, 7-10 August 2011; pp. 1-4.
12. Park, S.K.; Suh, Y.S.; Do, T.N. The Pedestrian Navigation System Using Vision and Inertial Sensors In Proceedings of ICROS-SICE International Joint Conference 2009, Fukuoka, Japan, 18-21 August 2009; pp. 3970-3974.
13. Suh, Y.S.; Do, T.N.; Ro, Y.S.; Kang, H.J. A Smoother for Attitude Estimation Using Inertial and Magnetic Sensors. In Proceedings of 2010 IEEE Sensors, Kona, HI, USA, 1-4 November 2010; pp. 743-746.
14. Lu, C.-P.; Hager, G.D.; Mjolsness, E. Fast and globally convergent pose estimation from video images. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 610-622.
15. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 1981, 24, 381-395.
16. Phillips, W.F.; Hailey, C.E.; Gebert, G.A. Review of attitude representations used for aircraft kinematics. J. Aircr. 2001, 38, 718-737.
17. Kuipers, J.B. Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerocpace, and Virtual Reality; Princeton University Press: Princeton, NJ, USA, 1999.
18. Shuster, M.D.; Oh, S.D. Three-axis attitude determination from vector observations. J. Guid. Control 1981, 4, 70-77.
19. Brown, R.G.; Hwang, P.Y.C. Introduction to Random Signals and Applied Kalman Filtering, 3rd ed.; John Wiley \& Sons Inc.: New York, NY, USA, 1997.
20. Suh, Y.S. Orientation estimation using a quaternion-based indirect Kalman filter with adaptive estimation of external acceleration. IEEE Trans Instrum. Meas. 2010, 59, 3296-3305.
21. Gramkow, C. On averaging rotations. Int. J. Comput. Vis. 2001, 42, 7-16.
22. Zhang, Z. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 1330-1334.
23. Won, S.P.; Golnaraghi, F. A triaxial accelerometer calibration method using a mathematical model. IEEE Trans. Instrum. Meas. 2010, 59, 2144-2153.
© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).
