Estimation of the Posture of Mobile Depth Sensor for Detecting the Ground and Walls in Terms of Initial Values*

Masahiro Tanaka†

The author previously applied Extended Kalman Filter for estimating the posture of depth sensor attached to a walking person or a mobile vehicle in conjunction with the surface of the plane, where the posture of the sensor was included in the state vector of the model. However, it could not adapt to situations when the initial value of Extended Kalman Filter was not appropriate. The initial value of the Extended Kalman Filter strongly affects the filtering results, especially for the model when the domain of the state vector has several groups. Different starting point leads to different results. In this paper, setting method of the initial value of the algorithm is proposed. Trial and error approach has proven to work well and has a good property to adapt to various situations. By defining the reset conditions differently, it is shown that the system can detect different surfaces.

1. Introduction

The author has been developing a mobile alarming system for a blind person, an elderly person on a mobility scooter and unmanned mobile vehicles such as a mobile robot by using a depth sensor[1–5]. The system is intended to raise an alarm for dangerous situations that can be detected by a depth sensor attached to the mobile body. The dangerous situations include the cases when there are obstacles on the road or holes exist on the road. In the former case, objects are located at higher positions than the surface of the road, and in the latter case, some part of the surface lacks and the detected object to its direction is located at lower position than the surface.

Depth sensor makes it easy to detect them. By transforming the polar coordinate of every pixel captured by a depth sensor into the Cartesian Coordinates of the world, it is possible to check the vertical position in the world coordinate for every pixel. Since judgment is based on the vertical level compared to the surface, it is crucial to detect the surface precisely.

The view model may be available from the specification of the camera. Furthermore, if we assume that the attachment of the sensor is with known pitch and roll angles at a known height, it is easy to express the relation between the polar coordinate from the sensor and the Cartesian coordinate of the environment. Thus we can figure out the Cartesian coordinate from the polar coordinate data. However, we have to cope with the fact that the posture always changes. Since the polar coordinate is based on the camera and the relation between the camera coordinate and the world coordinate depends on the camera posture that varies according to the posture of the mobile body, the problem requires a solid mathematical model.

The author first considered the case when only the pitch angle was subject to vary[1]. This happens when a depth sensor is attached to a mobility scooter and it goes straight on a rough passage. The author also considered the case when the vehicle turns[2]. When the sensor is attached to the basket of the mobility scooter, the roll angle changes as the driver tilts the handlebar. The height of the sensor position also changes in this case. For walking persons, this is obviously necessary.

Next, an estimation scheme based on Extended Kalman Filter was proposed, where the problem was formulated by a nonlinear state space model[4,5]. The state vector to be estimated was expressed as a 3-dimensional vector consisting of the pitch and roll angles and the height, where the range information from the sensor to the ground point excluding the data apart from the ground level is used as the observation data.

When the recursive procedure of Extended Kalman Filter is applied, the property of the algorithm depends on the initial values of the posture estimates. The author used a fixed initial value to this problem in the past research, but there were many cases when the system could not adapt to the actual pose easily.

---

* Manuscript Received Date: July 2, 2018

The material of this paper was partially presented at the 49th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS’17) which was held in November, 2017.

† Konan University; 8-9-1 Okamoto, Higashinada, Kobe-city, Hyogo 658-8501, JAPAN

Key Words: posture estimation, depth sensor, extended Kalman filter, initial condition.
In such cases, the sensor was intentionally rotated or moved up and down.

This paper proposes to use random initial values as a trial and error basis, where the goodness of the value is measured in one cycle of evaluation. The usefulness of this algorithm is shown in the experimental results.

2. Model

In this section, the model is described to be referenced in the discussion later. The model that is almost the same as was proposed in [4,6].

2.1 Coordinates

2.1.1 World Coordinate

The world coordinate is expressed by the Cartesian coordinate with \((x_0, y_0, h)\). The origin can be determined according to the property of the sensor. The case of Kinect is described in [6].

2.1.2 Camera Coordinate

\(X\) is the forward distance, \(Y\) is the distance to left, and \(Z\) is the height. The depth sensor captures the range \(r_{ij}\) for every direction of the view, where \(i\) is the number of vertical position from the top, and \(j\) is the number of horizontal position from the left.

2.1.3 Relation Between the Coordinates

Suppose the posture of the sensor is affected by Euler angle and shift. We assume that the camera is set with pitch angle \(\alpha\), roll angle \(\psi\) at position \((x_0, y_0, h)\) with these parameter values assumed to be known.

Fig. 1 shows the relation described above.

\[
\begin{bmatrix}
x_i \\
y_i \\
z_i
\end{bmatrix} = R_x(\psi)R_y(\alpha)r_{ij}\begin{bmatrix}
q_1^{(1)} \\
q_2^{(2)} \\
q_3^{(3)}
\end{bmatrix} + \begin{bmatrix}
x_0 \\
y_0 \\
h
\end{bmatrix}
\]

(1)

where \(R_x(\psi)\) and \(R_y(\alpha)\) are rotation matrices around axes \(x\) and \(y\), respectively. The symbols \(\{q^{(n)}_i|n = 1,2,3\}\) are constants at the pixel \((i,j)\) whose values are determined according to the property of the sensor.

2.2 Derivation of Observation Equation

Observation equation needs to be based on something whose shape and/or positions are known. For the case when the sensor is attached to a mobile body, a large flat area such as the floor or the walls can be the candidates. At any point on the floor, \(z = 0\). A wall can be treated as the floor if the sensor is rotated by 90°. Other objects are very difficult to be used in a mobile environment.

In this system, relatively a large number points of the data lie on the floor/ground or a wall. Consider, for example, the floor/ground case.

If \(r_{ij}\) reflects a point on the floor or the ground, the height of the position at \((i,j)\), or \(z_{ij}\) should be zero. Hence, by setting \(z_{ij} = 0\) in equation (1), we have a nonlinear equation:

\[
r_{ij} = -\frac{q_3}{\beta}
\]

(2)

where

\[
\beta = c_{ij}^{(1)}\sin q_1 \cos q_2 + c_{ij}^{(2)} \sin q_2 + c_{ij}^{(3)} \cos q_1 \cos q_2
\]

and the state vector \(q\) is defined as

\[
q = \begin{bmatrix}
q_1 \\
q_2 \\
q_3
\end{bmatrix} = \begin{bmatrix}
\alpha \\
\psi \\
h
\end{bmatrix}
\]

(3)

Since the relation above is on the same instant, the time index \(k\) is omitted for simplicity of notation.

Note that there might be a large number of points which reflect the floor/ground. Although it is not necessary to use all of the floor/ground points, by collecting a sufficient number of \((i,j)’s\) (the amount of such pixels is written as \(N\)), we have the following nonlinear observation equation:

\[
r = f(q) + e
\]

(4)

where \(e\) is the observation noise.

To apply Extended Kalman Filter, it is necessary to derive Jacobian of \(f(q(k))\). It can be derived as follows.

\[
H = \frac{\partial f}{\partial q} = \begin{bmatrix}
H_{1,1} & H_{1,2} & H_{1,3} \\
\vdots & \vdots & \vdots \\
H_{N,1} & H_{N,2} & H_{N,3}
\end{bmatrix}
\]

(5)

whose size is \(\tilde{N} \times 3\), and the function form can be derived from equation (2) as

\[
H_{ij,1} = \frac{\partial f_{ij}}{\partial q_1} = \frac{q_3}{\beta^2} c_{ij}^{(1)} \cos q_1 \cos q_2 - c_{ij}^{(3)} \sin q_1 \cos q_2
\]

(6)
\[ H_{ij,2} = \frac{\partial f_{ij}}{\partial q_2} = \frac{q_3}{\beta^2} - c_{ij}^{(1)} \sin q_1 \sin q_2 + c_{ij}^{(2)} \cos q_2 \]

\[ H_{ij,3} = \frac{\partial f_{ij}}{\partial q_3} = -\frac{1}{\beta} \]

Since the equation holds only for the points on the floor/ground, the points above or below the ground should not be included in this observation data.

Our technique is to estimate \( r_{ij} \) by using the currently estimated variables \( q \) in equation (4), and if it deviates largely from the observed \( r_{ij} \), the observation data on this point is excluded in the observation vector. Hence, the size of the output \( \tilde{N} \) is not always the same. We denote pixels on the floor as “valid data”.

The observation data may still be too large, since \( \tilde{N} \) is a valid portion of the number of pixels, e.g. 640 × 480 for Xtion Pro Live (which is our current sensor), which is much more than needed. Thus, we apply this system by using the valid portion of (640/16) × (480/16) which is enough for this problem.

2.3 State Transition Equation

As is already defined, the state vector \( q \) consists of the sensor’s angles and the height. Since we want to establish a broad range of sensor movement, we simply define the state transition model by

\[ q(k+1) = q(k) + w(k) \]  \hspace{1cm} (9)

where \( k \) is the discrete time corresponding to the estimation cycle, and \( w(k) \) is a Gaussian random vector with mean zero and covariance \( Q(k) \). By using equations (4) at time \( k \) and the state equation (9), an Extended Kalman Filter can be applied.

2.4 State Estimation Scheme

The state estimation scheme consists of the following steps:

Step 1. Set initial values. Give \( \hat{q}(0) \) and \( \hat{P}(0) \).

Step 2. Repeat this step.

Step 2.1 Selection for observation inclusion. For the selected points (e.g. (640/16) × (480/16) points), evaluate the estimate of \( r_{ij} \), and the point whose difference between the observation and the estimate of it is small enough is included in the observation.

Step 2.2 Apply Extended Kalman Filter. The state variables are time-updated by the following algorithm.

\[ \hat{q}(k) = \hat{q}(k-1) \]  \hspace{1cm} (10)

\[ \hat{P}(k) = \hat{P}(k-1) + Q(k) \]  \hspace{1cm} (11)

followed by the observation-update:

\[ \hat{q}(k) = \hat{q}(k) + K(k)(r(k) - f(\hat{q}(k))) \]  \hspace{1cm} (12)

\[ \hat{P}(k) = \hat{P}(k) - K(k)H(k)\hat{P}(k) \]  \hspace{1cm} (13)

\[ K(k) = \hat{P}(k)H(k) \times [H(k)\hat{P}(k)H(k)^T + R(k)]^{-1} = [\hat{P}(k) + H(k)^T R^{-1}(k)H(k)]^{-1} \times H(k)^T R^{-1}(k) \]  \hspace{1cm} (14)

where \( H(k) \) is Hessian (equation (5)) evaluated by using \( q(k) \). We prefer the second formula of equation (14) because of the matrix size. \( R(k) \) is a covariance matrix of \( e \). Since it can be assumed to be diagonal, the inverse of it does not need a matrix of size \( \tilde{N} \times \tilde{N} \).

Step 2.3 Evaluate every point in the data. By applying the updated posture value, the level of all points are re-calculated and judged.

Step 2.4 Count upper and lower data off the floor. Count the portion of the data off the floor level. Based on this judgment, the alarm may be dispatched.

Step 2.5 Check the reset condition. If any of the reset condition is not satisfied, go to Step 1 to reset the initial value of the state.

3. Determination of Initial Values

3.1 Theoretical Initial Value

Extended Kalman Filtering algorithm can be derived as a Bayes estimate. It has a recursive formula, and at every time \( k \), first the state vector \( q(k-1) \) is assumed to be generated from the Gaussian distribution \( N(q(k-1), \hat{P}(k-1)) \). Then, based on the time update equation, the prediction of \( q(k) \) can be computed as a Gaussian distribution \( N(q(k), \hat{P}(k)) \).

After getting the observation \( r(k) \), the Bayes estimate of \( q(k) \) is modified as \( N(q(k), \hat{P}(k)) \). This sequence is repeated.

The algorithm does not tell us any concrete idea how we should determine \( \hat{q}(0) \) and \( \hat{P}(0) \). In familiar textbooks of Kalman Filter such as [8–11], the initial values of this algorithm is simply \( \hat{q}(0) \) and \( \hat{P}(0) \), where there are no descriptions about the determination of these values.

3.2 Setting Values in the Author’s Previous Papers

In [2–5], the initial values of the sensor posture was assumed to be the same due to the same assumption of the usage of the system, i.e. it was attached to a human body or a mobile robot/vehicle with 20–40 [deg] depression angle to detect the floor or the ground. In [5], walls can also be detected, but they still remain based on the ground (or wall) detection first.

Changing the initial values can adapt to some cases. The author previously set the initial value as Table 1.
Table 1  The initial values in our previous papers

<table>
<thead>
<tr>
<th>mean</th>
<th>roll [rad]</th>
<th>height [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi/6$</td>
<td>0</td>
<td>1000</td>
</tr>
</tbody>
</table>

In this framework, the author showed various possibilities. Based on the estimated posture of the sensor, it becomes possible to compute the normal vector directions. Fig. 2 shows points whose normal vectors are approximately parallel to the floor, with two colors indicating left (yellow) or right (magenta) directions. Furthermore, the largest plane in the figure is parametrized and painted by cyan color ([7] (see Fig.2)).

![Image](image)

Fig. 2  Colors indicate facing left (yellow) and right (magenta), top (red) and parameterized primary wall (cyan)

It may be further possible to detect the secondary, tertiary surfaces, but everything here is based on the floor/ground detection. If the system fails to detect the floor, any wall cannot be detected, either. The new algorithm shown below will be useful in such cases.

### 3.3 Numerical Examples for our Previous Algorithm

When the real posture of the sensor is not close to the initial setting value of the algorithm, it takes a long time to adapt, or fails. In the following, this will be shown in numerical examples.

Table 2 shows experimental results when the real depression angle was 30°, roll angle was 0° and the height was 720mm.

We carried out two cases. Case 1 is when the initial values of $\hat{q}(0)$ was set nearly to the true angle, i.e.

$$\hat{q}_{0,1} = \hat{\alpha}_0 \sim N(25\pi/180,1)$$  \hspace{1cm} (15)

and in Case 2,

$$\hat{q}_{0,1} = \hat{\alpha}_0 \sim N(20\pi/180,1)$$  \hspace{1cm} (16)

The other two state values were set in the same way as

$$\hat{q}_{0,2} = \hat{\psi}_0 \sim N(0,1)$$  \hspace{1cm} (17)

$$\hat{q}_{0,3} = \hat{h}_0 \sim N(700,300^2)$$  \hspace{1cm} (18)

where the angles are denoted by [rad] and the height is by [mm]. Note that $N(\mu,\sigma^2)$ indicates a Gaussian distribution with mean $\mu$ and variance $\sigma^2$ and the sample was randomly generated from this distribution.

For each case, 50 trials were carried out whose results are summarized in Table 2. “No hit” indicates a case when the reset condition was consecutively carried out until the limit of time (10[sec]). The experimental situation was within a small room with walls and doors in the upper part of the view and the floor in the lower part of the floor. The time was measured by the program automatically.

![Image](image)

Table 2  The property of our old algorithm

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;no hit&quot; counts within 10 sec.</td>
<td>6/50</td>
</tr>
<tr>
<td>mean time to hit among hit cases [sec]</td>
<td>0.958</td>
</tr>
</tbody>
</table>

From this result, we can see that it is very difficult to get the possible initial value of the algorithm when the assumed initial values are unknown or largely deviated. Note that, in real situations when the sensor is attached to a human body, the person can change its posture by manual perturbation to hit. So, this problem is not recognized so much as is shown in the Table 2.

However, if the initial value was set to the values in Table 3, it adapted to the correct situation in a few seconds. Thus, the author’s previous algorithm had a strong limitation of usage.

Table 3  The initial values set for looking at the wall

<table>
<thead>
<tr>
<th>mean</th>
<th>roll [rad]</th>
<th>height [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi/2$</td>
<td>0</td>
<td>1000</td>
</tr>
</tbody>
</table>

4.  Proposal of Initial Value Setting for Some Problems

In this section, a new algorithm for initial value setting is proposed.

The Step 1 and Step 2.5 of Section 2.4 are given by the following algorithm.

**Step 1. Set initial values.** Randomly generate variables in each range uniformly:

$$\alpha_0 \leq q_{1,0} \leq \bar{\alpha}_0$$  \hspace{1cm} (19)

$$\psi_0 \leq q_{2,0} \leq \bar{\psi}_0$$  \hspace{1cm} (20)
\[ h_0 \leq q_{3,0} \leq \bar{h}_0 \]  

A diagonal matrix \( \hat{P} \) is defined by considering the change of the variables in one cycle.

**Step 2.5 Check the reset condition.** At Step 2.5, check the inequalities (19)-(21) if they are satisfied or not, as well as the following inequalities:

\[
N_{\text{valid}} \geq N_{\text{valid}} \quad (22)
\]
\[
N_{\text{far}} \leq N_{\text{far}} \quad (23)
\]

where \( N_{\text{valid}} \) is the portion of the plane surface data in the acquired amount of depth data, and \( N_{\text{far}} \) is the portion of the farther data than the obtained ground/wall. If the posture parameter values are near to the truth, large portion of the data can be judged to lie on the plane surface. So, this value can be used for the goodness of the posture parameters (state values).

These constants \( N_{\text{valid}} \) and \( N_{\text{far}} \) should be determined according to the target problem, and some preliminary experiment to determine such parameters need to be carried out. If not all the conditions are satisfied, the current estimate of the state is considered to be invalid, and the algorithm is reset.

### 4.1 Numerical Examples for the Proposed Algorithm

Here, we come back to the initial setting problem explained in section 3.2, where the initial guess of the posture of the sensor was important and difficult. In our new algorithm explained above in section 4., we check this point again and its result.

We made 50 trials when the initial guess of the unknown posture was near to the true value (Case 1) and was different substantially (Case 2), respectively. For each case, the results are summarized in Table 4. All the conditions except the following values are the same as the one for Table 2. \( N_{\text{valid}} \) was set to 0.4, and \( N_{\text{far}} \) was set to 0.1.

<table>
<thead>
<tr>
<th>Item</th>
<th>Prog1</th>
<th>Prog2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_0 )</td>
<td>( 20^\circ )</td>
<td>( 70^\circ )</td>
</tr>
<tr>
<td>( \beta_0 )</td>
<td>( 50^\circ )</td>
<td>( 110^\circ )</td>
</tr>
<tr>
<td>( \psi_0 )</td>
<td>( -30^\circ )</td>
<td>( -30^\circ )</td>
</tr>
<tr>
<td>( \psi_1 )</td>
<td>( 30^\circ )</td>
<td>( 30^\circ )</td>
</tr>
<tr>
<td>( h_0 )</td>
<td>( 20 \text{ cm} )</td>
<td>( 20 \text{ cm} )</td>
</tr>
<tr>
<td>( h_0 )</td>
<td>( 400 \text{ cm} )</td>
<td>( 500 \text{ cm} )</td>
</tr>
<tr>
<td>( N_{\text{valid}} )</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>( N_{\text{far}} )</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

In this table, we can see that the property of hitting the good initial condition between the Cases 1 and 2 is not much different, and hence this algorithm can be said “robust” for initial value setting. By comparing Table 4 (new algorithm) to Table 2 (old algorithm), the detecting speed for Case 1 is worse. However, no hit cases in Table 4 is smaller than Table 2. By comparing Case 2 in Table 4 to Table 2, the number of instances of “no hit” case has been greatly reduced. In other words, only the new algorithm can be used when the initial guess of the posture is very much uncertain.

### 4.2 For a Pedestrian

This system is intended to be applied to blind persons for walking. Blind person uses a white cane to feel the pressure and the sound when it touches some object.

Akita Seiko Co. Ltd. and Akita Prefectural University developed an electronic white cane\(^1\). It has a shape of a traditional white cane with ultra sound to detect objects, and vibration is used to notice the danger to the person.

Our system is completely supplemental to the white cane. We do not propose replacing a white cane by our system.

The system for those who need this system such as blind persons should be equipped with the detection ability of the objects on the floor and also the approaching walls or doors. In this application, there exist two distinguished features: to scan the floor, the depression angle \( \alpha \) is always around \( 20^\circ \sim 40^\circ \), and the height from the floor is at around the level of sensor attached to the body. Also, it is more important to detect obstacles or holes than measuring the posture of the sensor.

Here we run this program with two settings.

**Prog1** The program for detecting the floor needs to be used with a depression angle, e.g. \( 20 \) [deg.]. In this case, the height of the sensor oscillates within a few centimeters.

**Prog2** The program for detecting the wall or the door needs to be used with a depression angle, e.g. \( 90 \) [deg.]. In this case, the height of the sensor decreases as the person approaches the wall/door.

\(^1\)http://www.jst.go.jp/pr/announce/20110530-2/index.html
Now we explain about the figure. In the top row, two values are indicated. The left one (surf) indicates the portion of the pixels judged to be on the primary surface. In Fig. 4, it is 0.29. The right one (val) indicates the portion of valid data in the image. If the sensor is too close to some object, there are few valid data. If this value is too small, a string “few valid data” appears to the right of this number.

In the second row, 4 numbers are indicated. The left one is the FPS (Frames Per Second). The second one is the estimated $\alpha$. The third one is the estimated $\psi$. The fourth one is the estimated height $h$. The right one is the estimated height $h$.

In the figure, primary surface (i.e. wall or floor) is not painted on the original image and can be seen gray, red pixels are the part which are closer than the primary surface. If the primary surface is the floor, the red part indicates some objects upper than the floor. So, a walking person must pay attention to it. Blue pixels are the part which are farther than the primary surface. If the primary surface is the floor, the blue part indicates something lower than the floor, e.g. holes, ditches, and so on. In the room, usually blue part does not appear with exception at the stairs going down.

If the primary surface is the wall (cases of Figs. 5 and 6), red part shows something closer than the wall. Blue part shows the area beyond the wall.

By looking at these figures, we can see that the primary planes were detected in different ways. In Fig. 3, the primary surface was the floor, and in Fig. 5, it was the door just in front of the sensor.

In Figs. 4 and 6, the result when the sensor approached the door is shown. This can be seen in the values of $h$. In Fig. 5, the distance to the wall is 1800.2 mm, while in Fig. 6, it is 813.4 mm.

### 4.3 As a Measuring Device for Height or Distance in the Cluttered Site

This system can be used to measure the distance to the wall when the wall has various disturbances or there are some annoying objects moving in front of the wall. Also, it can be used to measure the distance to the ground from a flying vehicle, roller coaster or lift which is not necessarily stable.

<table>
<thead>
<tr>
<th>Table 6</th>
<th>Parameters used in inequalities (19)-(23)</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>value</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>$-50^\circ$</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>$50^\circ$</td>
</tr>
<tr>
<td>$\psi_0$</td>
<td>$-30^\circ$</td>
</tr>
<tr>
<td>$\psi_1$</td>
<td>$30^\circ$</td>
</tr>
<tr>
<td>$h_0$</td>
<td>$d \times [\cos \gamma \pi / 180], 1 - r$</td>
</tr>
<tr>
<td>$h_1$</td>
<td>$d \times [\cos \gamma \pi / 180], 1 + r$</td>
</tr>
<tr>
<td>$N_{valid}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$N_{far}$</td>
<td>0.2</td>
</tr>
</tbody>
</table>

The initial values are indicated in Table 6. We carried out 4-case experiments where $\gamma$ is the possible maximal angle of the sensor direction and the vertical line, and $d$ denotes the distance from the sensor to one of the pre-defined center point of the depth image (e.g. $(iW/4, jH/4); i, j = 1, 2, 3$) selected randomly. The domain of a random number $r$ is chosen appropriately depending on the environment. Now $\gamma = 50^\circ$
Table 7  Measured results

<table>
<thead>
<tr>
<th>No.</th>
<th>place</th>
<th>this system</th>
<th>real value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>table → ceiling</td>
<td>2,016</td>
<td>2,039</td>
</tr>
<tr>
<td>2</td>
<td>table → ceiling (inclined by 45°)</td>
<td>1,949</td>
<td>(2,039)</td>
</tr>
<tr>
<td>3</td>
<td>cluttered floor</td>
<td>2,325</td>
<td>2,321</td>
</tr>
<tr>
<td>4</td>
<td>cluttered wall</td>
<td>4,471</td>
<td>4,424</td>
</tr>
<tr>
<td>5</td>
<td>floor (sofa)</td>
<td>561</td>
<td>(948)</td>
</tr>
<tr>
<td>6</td>
<td>floor (sofa) wall</td>
<td>944</td>
<td>948</td>
</tr>
</tbody>
</table>

“Real value” in this table was measured by a laser measure LD-500 (STABILA).

This system can be applied to measure the height of the sensor position in cluttered areas. Estimating the camera height and posture over a cluttered area is an example.

The results are shown in Table 7. The experiments 1 and 2 are measuring the same height with different sensor angles, and the target photos are shown in Figs. 7 and 8. Even when the sensor is inclined, the distance to the ceiling was almost correctly measured.

Next, the measured distance to the cluttered floor is shown as No.3 of Table 7 showing the scene in Fig. 9. This shows that this system is also valid to situations where the floor is cluttered and the sensor posture is correctly estimated.

Number 4 of Table 7 and Fig. 10 is the case of measuring the distance to a far wall (far in the sense of this sensor’s spec.).

If the sensor’s direction is perpendicular to the ground, the distance between the sensor and the ground is just the height $h$. Same discussion holds to the wall case.
If the direction of the sensor and the plane has the angle $\gamma$, the distance between the sensor and the plane is $h \sin \gamma \pi / 180$. Since the angle is unknown, random angle is adopted, and further random distance taken from the interval $[-r, r]$ is added.

We show the last situation. When there are actually no holes, we can utilize the decision result judged to be a hole or a ditch, or “far” area. By resetting using the amount of far area larger than the threshold, the system can search the farthest plane. This is effective when there are large plane nearer than the background and we want to find the latter one. Number 3 and 4 (corresponding to Figs. 11 and 12) are the case when we want to measure the distance to the floor, but a sofa has appeared in a large part in the view. Since the floor can be seen slightly in the bottom, the system has captured the sofa surface first (Fig. 11). When the floor was seen as a bit larger part (Fig. 12), the floor was captured as the objective surface.

5. Conclusions

This paper discussed about the initial value of Extended Kalman Filter with a decision in it for sensor posture estimation in our application system. The proposed method is a trial and error approach by using random numbers including a little information of the unknown variable, and its evaluation is based on the decision rate.

By using the proposed approach, the applicability of this system has been extended.

Acknowledgments

This research has been partially supported by JSPS KAKENHI Grant Number #16K01581, and also supported in part by MEXT, Japan.

This paper is a revised and expanded version of the paper “On the Initial Guess of Posture of Mobile Depth Sensor for Detecting the Primary Surface in the Environment” included in the proceedings of SSS’17[12].

References


Author

Masahiro Tanaka (Member)

Masahiro Tanaka was born in May, 1956. He graduated from the Department of Applied Math. and Physics, Kyoto University in 1979, and he got degrees of M. Eng. in 1981, D. Eng. in Control Engineering in 1990 both from Kyoto University. He was with Shimadzu Corporation from 1981, Shiga University from 1983, Okayama University from 1990, and since 1999, he has been a professor at Konan University, Kobe, Japan, where he is currently at Department of Intelligence and Informatics. He was a research scholar of IIASA from 1988 to 1989. His scientific fields include neural networks and deep learning, genetic algorithm, computer vision, state estimation, robotics and visual sensor applications. He is a member of SICE, IEEE, RSJ, IEE, JSME and JSEC.