# A Fast Position Estimation Method for a Control Rod Guide Tube Inspection Robot with a Single Camera

Jae C. Lee, Jun H. Seop, Yu R. Choi, Jae H. Kim
*Korea Atomic Energy Research Institute, 150. Duckjin dong, Yusong gu, Taejon, jclee2@kaeri.re.kr*

## 1. Introduction

One of the problems in the inspection of control rod guide tubes using a mobile robot is accurate estimation of the robot's position. The problem is usually explained by the question "Where am I?"[1,2]. We can solve this question by a method called dead reckoning using odometers. But it has some inherent drawbacks such that the position error grows without bound unless an independent reference is used periodically to reduce the errors.

In this paper, we presented one method to overcome this drawback by using a vision sensor. Our method is based on the classical Lucas Kanade algorithm for on image tracking. In this algorithm, an optical flow must be calculated at every image frame, thus it has intensive computing load. In order to handle large motions, it is preferable to use a large integration window. But a small integration window is more preferable to keep the details contained in the images.

We used the robot's movement information obtained from the dead reckoning as an input parameter for the feature tracking algorithm in order to restrict the position of an integration window. By means of this method, we could reduce the size of an integration window without any loss of its ability to handle large motions and could avoid the trade off in the accuracy. And we could estimate the position of our robot relatively fast without on intensive computing time and the inherent drawbacks mentioned above.

We studied this algorithm for applying it to the control rod guide tubes inspection robot and tried an inspection without on operator's intervention.

## 2. Dead Reckoning and Image Feature Tracking

The two key methods of our position estimator are dead reckoning and feature tracking. In this section, the two techniques are described.

### *2.1 Dead Reckoning Position Estimating*

Dead reckoning is the process of estimating a robot's position by advancing a known position using the direction, speed and distance to be traveled. Our robot is a type of differential drive mobile robot. The incremental travel distance for the left and right wheel, $\Delta U_L$ and $\Delta U_R$ can be computed by counting the pulses from the corresponding encoders. Incremental linear displacement of robot's center-point, denoted by $\Delta U_i$, is:

$$\Delta U_i = \frac{1}{2}(\Delta U_L + \Delta U_R) \tag{1}$$

And a robot's incremental change of orientation is:

$$\Delta \Theta_i = \frac{1}{b}(\Delta U_L - \Delta U_R) \tag{2}$$

Where b is the wheelbase of the robot, ideally measured as the distance between the two contact points for the wheel and the floor. The robot's new position and orientation are computed as follows:

$$\begin{aligned} \Delta\Theta_i &= (\Theta_{i-1} + \Delta\Theta_i) \\ x_i &= x_{i-1} + \Delta U_i \cos\Theta_i \, . \\ y_i &= y_{i-1} + \Delta U_i \sin\Theta_i \end{aligned} \tag{3}$$

Where $x_i$ and $y_i$ are the relative position of the robot's center-point at instant i.

### *2.2 Lucas Kanade Feature Tracking*

We will explain the classical Lucas Kanade feature tracking algorithm briefly[3,4]. Let A and B be two grayscaled 2D images. The two quantities A(**x**) and B(**x**) are the grayscale value of the two images at location $\mathbf{x} = [x\ y]^T$, where x and y are the two pixel coordinates of an image point **x**. Images A and B will be referenced respectively as the first and the second image. We consider an image point $\mathbf{u} = [u_x\ u_y]^T$ on the first image A. To track the motions of the image point **u** resulting from the robot's movement, we have to find the location $\mathbf{v} = \mathbf{u} + \mathbf{d} = [u_x+d_x,\ u_y+d_y]^T$ on the second image B such that A(u) and B(v) are similar. The vector $\mathbf{d} = [d_x\ d_y]^T$ is the optical flow at **x** and represents the image velocity at **x**. We defined **d** as being the vector that minimizes the similarity function ε defined as follows:

$$\varepsilon(\mathbf{d}) = e(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (A(x,y) - B(x+d_x, y+d_y))^2. \tag{4}$$

Where $w_x$ and $w_y$ are the integers to represent the aperture. At the optimal **d**, the first derivative of ε with respect to **d** is zero:

$$\frac{\partial \varepsilon(\mathbf{d})}{\partial \mathbf{d}} = [0 \quad 0] \tag{5}$$

After an expansion of the derivative, we obtain:

$$\frac{\partial \varepsilon(\mathbf{d})}{\partial \mathbf{d}} \approx -2 \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (A(x,y)-B(x+d_x,y+d_y)) \begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} \end{bmatrix} \tag{6}$$

We substitute B by its first order Taylor expansion about point $\mathbf{d} = [0\ 0]^T$:

$$\frac{\partial \varepsilon(\mathbf{d})}{\partial \mathbf{d}} \approx -2 \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left(A(x,y)-B(x,y)-\mathbf{d}\cdot\begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} \end{bmatrix}\right)\begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} \end{bmatrix} \tag{7}$$

The quantity of A(x,y)-B(x,y) can be interpreted as the temporal image derivative at point $[x\ y]^T$:

$$\delta I(x,y) \doteq A(x,y) - B(x,y). \tag{8}$$

The matrix $[\frac{\partial B}{\partial x}\ \ \frac{\partial B}{\partial y}]$ can be interpreted as the image gradient vector:

$$\nabla \mathbf{I} = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \doteq \begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} \end{bmatrix}^T \tag{9}$$

Equation (3) can be written as follows.

$$\frac{1}{2}\frac{\partial \varepsilon(\mathbf{d})}{\partial \mathbf{d}} \approx \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (\nabla \mathbf{I}^T \bar{d} - \delta I)\nabla \mathbf{I}^T,$$

$$\frac{1}{2}\left[\frac{\partial e(\mathbf{d})}{\partial \mathbf{d}}\right]^T \approx \mathbf{G}\mathbf{d} - \mathbf{b} \tag{10}$$

where

$$\mathbf{G} = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{11}$$

and

$$\mathbf{b} = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \begin{bmatrix} \partial I I_x \\ \partial I I_y \end{bmatrix} \tag{12}$$

Therefore, the optimum optical flow vector is

$$\mathbf{d}_{opt} = \mathbf{G}^{-1}\mathbf{b}. \tag{13}$$

This equation is valid only if the matrix G is not singular.

## 3. Proposed Method

The robot's position after a given length of time is defined by equation (3). And the robot's movement will create the differences for the image B compared to image A in equation (4) as vector **d**. For a clarity purpose, we changed the name of the displacement vector **d** to **t** as in the following equation:

$$\mathbf{t} = \lfloor t_x \quad t_y \rfloor = \lfloor d_x - x_i \quad d_y - y_i \rfloor \tag{14}$$

where $x_i$ and $y_i$ are obtained from equation (3). And image point **u** is changed as follows:

$$\mathbf{p} = \lfloor u_x \cos\Theta_i \quad -u_y \sin\Theta_i \rfloor. \tag{15}$$

Then equation (4) is can be rewritten as follows:

$$\varepsilon(\mathbf{t}) = e(t_x, t_y) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x,y) - B(x+t_x, y+t_y))^2. \tag{16}$$

After the same processes as described in section 2.2, the optical flow vector **t** can be calculated. And it is used to correct the $x_i$ and $y_i$ in equation (3) to reset the accumulated errors. The modified process is as follows:

a. Calculate the robot's next position by equation (3).
b. Calculate the vector **t** that minimizes equation (16).
c. Reset the accumulated errors by vector **t**
d. Robot's orientation can be taken by using an independent absolute sensor like compass.

## 3. Conclusion

In this paper, we proposed a method that combines two techniques, an odometer based dead reckoning and a vision based image tracking. The two key components to any position estimator are accuracy and robustness. It is expected that our method satisfies both components satisfy by combining the two complementary methods.
Also, there are some restrictions to our method. Because $u_x$ and $u_y$ are integers, a type of interpolation is needed in equation (15). And this method can only be useful if the robot's movement plane and image plane are parallel. Inspection of reactor control rod guide tubes is this very circumstance. These problems will be further studied.

## REFERENCES


[1] J. Borenstein, H. R. Everett, and L. Feng Hantz, Sensors and Methods for Mobile Robot Positioning, The University of Michigan, p. 14, 1996.

[2] A. Agent, The Advantages of Absolute Encoders for Motion Control, SensorsIn Physics Research, p.19, 1991.

[3] J. Y. Bogust, Pyramidal Implementation of the Lucas Kanade Feature Tracker, Intel Corporation Microprocessor Research Lab. Documentation from Internet.

[4] J. Shi and C. Tomasi, Good Features to Track, Proceedings of IEEE Computer Society Conference Computer Vision and Pattern Recognition, P. 593, 1994.