Vision-Based Tracking with Extended Kalman Filter

T. Honda*, S. Takahashi*, H. Takauji**, S. Kaneko**

*Department of Intelligent Mechanical Systems Engineering, Kagawa University, Kagawa, Japan (e-mail: saru@eng.kagawa-u.ac.jp)
** Division of Systems Science and Informatics, Hokkaido University, Hokkaido, Japan (kaneko@ssi.ist.hokudai.ac.jp)

Abstract: In this paper, we introduce a tracking method for moving insect. Our method is robust image processing to change of illumination under orientation code matching. Further, based on an extended Kalman filter, it can track moving insects which are acting under the random motion. Through experiments, this paper shows effectiveness of our method.

Keywords:

1. INTRODUCTION

Vision-based tracking is an important research field in the medical area (see, Reference: Honda, T., et al. (2010)) and vehicle industry such as the intelligent transport system. However, image tracking in real environment has some problems. Especially, it is hard to detect the moving object by influences such that change of image brightness and occlusion of objects. In order to solve these subjects, there are methods such as the method of background difference (see, Reference: Wajima, W., et al. (2006)), and the method of template matching based on orientation code introduced by authors (see, References: Domae, Y., et al. (2005), Ullah, U., et al. (2001)). Background difference method is to see the difference after comparing the background image and the target image. It can be expected the fast processing time. But, it is not easy to detect the moving object with high accuracy. The other hand, since orientation code matching (OCM) uses the gradient of brightness at the pixel of feature point it is robust to illumination change of environment. However, when we treat OCM to detect moving object through time series images, there is a kind of problem. Since it occur the case where values of orientation code at some pixels are similar in matching area, there is a situation that moving object cannot be classified. In the past, the tracking method based on both OCM and motion vector has been proposed (see, Reference: Kagoike, K., et al. (2008)) in order to consider the above problem. However, the tracking accuracy is still low in the case where texture in background image is complicated. In this paper, we treat vision-based tracking by both OCM and extended Kalman filter (see, Reference: Katayama, T., (1999)) to estimate the position of moving object with high accuracy improving the past result (see, Reference: Kagoike, K., et al. (2008)). In here, let the moving object be insect such as ant and mosquito. Further, we use multiple templates for applying to complicated background texture.

In here, we show outline of this paper. In Chapter 2, notion concerning orientation code is given and the idea that extracts the position candidate area of moving insect is described. Further, for the candidate area above we seek the correct position of moving insect applying information such as position, speed, acceleration derived by extended Kalman filter in Chapter 3. Chapter 4 shows algorithm of our method which is introduced on Chapters 2 and 3. Through some of experimental results, we prove the effectiveness of proposed method in Chapter 5. Finally, Chapter 6 gives the short summary.

2. POSITION DETECTION BASED ON MULTI-TEMPLATE MATCHING AND ORIENTATION CODE

2.1 Orientation Code

Orientation code is the quantized value of the brightness gradient at certain pixel. It has unique feature at each pixel. In here, let notation \( I(i, j) \) denote the brightness at certain pixel \( (i, j) \). Then, the gradient \( \theta_{ij} \) of brightness is given by

\[
\theta_{ij} = \tan^{-1}\left( \frac{\nabla I_j}{\nabla I_i} \right)
\]  (1)

where \( \nabla I_i \) means the horizontal gradient and \( \nabla I_j \) shows the vertical gradient. Based on (1), orientation code \( c_{ij} \) is obtained by (2). The quantization width of gradient of brightness is decided by value of \( \Delta \theta \).

\[
c_{ij} = \begin{cases} 
\theta_{ij} & \text{if } |\nabla I_i| + |\nabla I_j| > \Gamma \\
N \frac{\Delta \theta}{2\pi} & \text{ otherwise}
\end{cases}
\]  (2)

where let the width \( \Delta \theta \) be 8, \( \Gamma \) is threshold value. In (2), if the gradient of brightness \( |\nabla I_i + \nabla I_j| \) at a certain pixel is...
smaller than $\Gamma$, it means that the pixel is removed. The other hand, $|V_l + V_I|$ is larger than $\Gamma$, the value of orientation code $c_{ij}$ is between 0 and 15.

2.2 Multi-templates Matching based on Orientation Code

In this section, we introduce a method of finding an area which exist the moving insect based on multi-template matching using orientation code. First, we fix the position ($i', j'$) of moving insect on initial image. Further, we seek the similar area on next time $k$th image by template data obtained from time $(k-U)$th image to time $(k-1)$th image. Namely, it means that we use multi-template. In here, let $U$ be an arbitrary number. Notation $k$ shows the sampling number. Let size of searching area on image be $M \times M$ (pixels $\times$ pixels). Then, value $D_{Tk}(ij)$ is defined as

$$D_{Tk}(ij) = \frac{1}{M^2} \sum_{cT,cg} d(cT,cg).$$  

(3)

Eq (3) means the correlation value between template area and the object area on time $T$. $cT, cg$ are orientation codes, respectively. In here,

$$d(a,b) = \left\{ \begin{array}{ll} \min |a - b|, & \text{if } a \neq N, b \neq N, \\ N, & \text{otherwise} \end{array} \right.$$  

(4)

When value $D_{Tk}(ij)$ is calculated, considering orientation code is cyclic, we obtain $D_{Tk}(ij)$ of corresponding pixels of template and target from (4). Then, using templates of $U$ pieces, we measure the correlation with target and each template. Further, we calculate it from $D_{k-1}$ to $D_{k-U}$ and then take average of between $D_{k-1}$ and $D_{k-U}$. We consider the case of false detection happened to be occurrence by high similarities of appearance between template in time $k-1$ and background texture. At this time, using more past time template than time $k-1$, similarity is down between template and background texture. Similarity is less down between template and moving object area than background area, because moving object in the area and moving object in the template have high correlation. Using average of multiple templates, we think we can get information of moving object in background texture. Then, average value $D'(k)$ is calculated from

$$D'(k) = \frac{1}{U} \sum_{k-U} D_{Tk}(ij).$$  

(5)

Further, we need to consider the frame late of processing. Then, search area of moving insect is limited as $H \times H$ (pixels $\times$ pixels).

3. METHOD OF POSITION DECISION BY STATE ESTIMATION

In this section, we show a method of choice of the moving insect position.

3.1 Position Estimation based on Extended Kalman Filter

We explain the method of calculate estimated position of the moving object. If collation position was ($i'_{k-1}, j'_{k-1}$) on time $k-1$, then our method calculates the estimation position ($i_k, j_k$) on time $k$. Then, this position is obtained by

$$\begin{align*}
  i_k &= i'_{k-1} + \dot{v}_{i,k-1} \Delta T + \ddot{a}_{i,k-1} \frac{1}{2} \Delta T^2 \\
  j_k &= j'_{k-1} + \dot{v}_{j,k-1} \Delta T + \ddot{a}_{j,k-1} \frac{1}{2} \Delta T^2
\end{align*}$$  

(6)

where $\dot{v}_{i,k-1}, \dot{v}_{j,k-1}, \ddot{a}_{i,k-1}, \ddot{a}_{j,k-1}$ represent the estimation of velocity and acceleration, respectively. These are calculated based on an extended Kalman filter. In here, let $\Delta T$ represent the sampling time. Suppose that $\Delta T = 0.3$. Now, we introduce an extended Kalman filter as follows. If input value of extended Kalman filter is observation value $y_{k-1}$, we substitute the collation position on time $k-1$ for $y_{k-1}$. Output is state of $\hat{x}_{k|k-1}$, and we substitute estimate value of position, velocity, and acceleration in time $k$. Now, input $y_{k-1}$ and output $\hat{x}_{k|k-1}$ are represented in (7) and (8).

$$\begin{align*}
  y_{k-1} &= [i'_{k-1}, j'_{k-1}] \\
  \hat{x}_{k|k-1} &= [\hat{i}_{k|k-1}, \hat{j}_{k|k-1}, \dot{v}_{i,k-1|k-1}, \dot{v}_{j,k-1|k-1}, \ddot{a}_{i,k-1|k-1}, \ddot{a}_{j,k-1|k-1}]^T
\end{align*}$$  

(7)

(8)

Then, an extended Kalman filter is presented by

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-2} + K_{k-1}(y_{k-1} - H\hat{x}_{k-1|k-2})$$  

(9)

$$K_{k-1} = P_{k-1|k-2}H^T(HP_{k-1|k-2}H^T + I_{2\times 2})^{-1}$$  

(10)

$$\hat{P}_{k|k-2} = P_{k|k-2} - K_{k-1}HP_{k-1|k-2}$$  

(11)

$$\hat{P}_{k|k-1} = F\hat{P}_{k-1|k-2}F^T + v_{\sigma o}$$  

(12)

$$F = \begin{bmatrix} I_{2\times 2} & p \cdot I_{2\times 2} & \frac{1}{2} p^2 \cdot I_{2\times 2} \\
O_{2\times 2} & I_{2\times 2} & p \cdot I_{2\times 2} \\
O_{2\times 2} & O_{2\times 2} & I_{2\times 2} \end{bmatrix}$$  

(13)

$$H = [I_{2\times 2} O_{2\times 2} O_{2\times 2}]$$  

(14)

$$G = [O_{2\times 2} O_{2\times 2} I_{2\times 2}]$$  

(15)

$$H = [O_{2\times 2} O_{2\times 2} I_{2\times 2}]$$  

(16)

where $K_{k-1}$ is Kalman gain, $\hat{P}_{k|k}$ represents the covariance of estimation error on time $k$. $v_{\sigma o}$ is the covariance ratio of process noise. $p$ presents the sampling time and it is different from $\Delta T$.

3.2 Relationship between Estimation Position and Candidate Position

First, we calculate the distance between the candidate position ($i'_k, j'_k$) and the estimated position ($i_k, j_k$). Depending on the distance, we fix the position of moving ant. Then, we use Euclidean distance to get both of the candidate position
of moving ant and the estimated position of it, respectively. In Fig. 1, the relationship between \((i_k, j_k)\) and \((i_k', j_k')\) is shown. Here, the evaluated value of distance is

\[
x_{k(ij)} = \frac{(i_k - i_k')^2 + (j_k - j_k')^2}{\sqrt{(i_k - i_{\text{max}})^2 + (j_k - j_{\text{max}})^2}} \tag{17}
\]

3.3 Decision Method of Position based on Evaluation Function

The integration equation \(E_{k(ij)}\) based on the evaluation function of both of the correlation value by the template agreement using \((5)\) and the distance value based on the estimated position calculated by \((17)\) is defined by

\[
E_{k(ij)} = \alpha \frac{2D'(ij)}{N} + (1 - \alpha) \cdot X_{k(ij)} \tag{18}
\]

where let \(\alpha\) be value of \([0, 1]\). If \(\alpha\) is close to 0, then the reliability of \(X_{k(ij)}\) is high. The other hand, if \(\alpha\) is close to 1, the reliability of \(D'(ij)\) is high. By using \((18)\), it is possible to track the moving ant without the false detection in various cases.

4. TRACKING ALGORITHM

In this chapter, we introduce the process flow of moving ant tracking. In here, we take an arbitrary point \((i_0, j_0)\) on the initial image and pickup the template area \(M \times M\) (pixels \(\times\) pixels) at a center of its point. According to the following algorithm, it can track the moving ant.

Take the target image on time \(k\).

Do multi-templates matching based on orientation code using \(U\) templates and calculate the evaluation value \(D'(ij)\) in \((5)\).

Define the correlation area of moving ant on time \(k-1\). Calculate the estimation velocity and the estimation acceleration on time \(k\) based on an extend Kalman filter in\((4)-(16)\).

Get the estimation position on time \(k\) based on each the estimation value and correlation point of time \(k-1\).

Calculate the distance value \(X_{k(ij)}\) in \((17)\).

Calculate the value \(E_{k(ij)}\) by using \(D'(ij)\) and \(X_{k(ij)}\).

Run\((b)-(f)\) processes at all pixels on the target image.

Repeat the above.

In our proposed method, we record the correlation position on time \(k\) and use next template matching on time \(k+1\). Each time template is renewal every frame and the oldest template in time \(k-U\) is throw away. If the first frame is defined as time 0 then there is little prior information. Namely, position estimation on time 1 based on an extend Kalman filter is difficult. Then, though we do not estimate the position, we use value of \(D'(ij)\) to evaluate. Further, we take the correlation position \((i_1, j_1)\) on time 1 and use the value of position \((i_1', j_1')\) as initial value of velocity \((v_{i_1}, v_{j_1})\) after time 2.

5. EXPERIMENTAL RESULTS

In here, we show the tracking result of moving ant using the movie obtained from video camera.

5.1 System Parameters

We define arbitrary parameters in proposal method. Value of \(\alpha\) shows the reliability between estimation and template matching. \(U\) is the number of template. \(S\) is size of search area. We see influence of their values in from Experiment 1 to Experiment 3 and consider each parameter that possible best to tracking accuracy. Center point of moving object \((i_{\text{TRUE}}, j_{\text{TRUE}})\) is obtained in each image frame by visual evaluation. Then we measure the distance between collation position \((i_k', j_k')\) and \((i_{\text{TRUE}}, j_{\text{TRUE}})\) and calculate sum of that distance each frame image. Fig. 2 shows the conceptual figure of measurement of tracking error. Fig. 3 is image of tracking target in this experiment. Fig. 3's image shows 50th frame, and many ants exist in place of marble texture. First, moving object place is pointed arrow in those figures. Then, illumination intensity of image is 1322 lx.

5.1.1 Experiment 1 - Reliability Value \(\alpha\) -

We inspect influence of arbitrary constant value \(\alpha\) influence of tracking accuracy. Value \(\alpha\) means constant value of decides reliability between estimate and template matching. This inspect is experimental value \(\alpha\) from 0 to 1. In here, proposal method is set up parameters as described below, horizontal image size is 640 pixels, and vertical image size is 480 pixels, size of search area \(S \times S\) is 151 pixels \(\times\) 151 pixels, size of collation area \(M \times M\) is 15 pixels \(\times\) 15 pixels, constant value \(p\) in \(F\) matrix in \((14)\) is 0.3, quantize width \(\Delta \theta\)
is $\pi/8$, noise threshold value $\Gamma$ is 20, noise covariance ratio \( v_{so} \) is 0.2 and the number of template $U$ is 2 pieces. We show Fig. 4 which is an experimental result of each value $\alpha$. According to Fig. 4, it seems that object tracking success without false recognized in case of setting value $\alpha$ is 0.5 to 0.9. Now we summarize the error in Table I. According to Table I, it is least tracking error in case of value $\alpha$ is 0.9. But, there could be estimate accuracy is down in case of target speed is suddenly change, so value $\alpha$ need to be large value for high reliability of template matching. Above experimental result, Exact tracking is possible with adjust reliability between position estimate and template matching to target moving object.

5.1.2 Experiment 2 - Number of Template $U$

We inspect influence of the number of template $U$ upon tracking accuracy. As the value of $U$ is increased, proposal method distinguish between moving object area and background area, but tracking accuracy could be down by change of moving object shape, and long process time is necessity. This inspects change $U$ at 1 in the interval from 1 to 10.

Proposed method is set up parameters as follows. Horizontal image size is 640 pixels, and vertical image size is 480 pixels, size of search area $S \times S$ is 151 pixels $\times$ 151 pixels, size of collation area $M \times M$ is 15 pixels $\times$ 15 pixels, constant value $p$ in $F$ matrix is 0.3, quantize width $\Delta \theta$ is $\pi/8$, noise threshold value $\Gamma$ is 20, noise covariance ratio $v_{so}$ is 0.2 and constant value $\alpha$ is 0.7. We show Fig. 5 that is experimental result of each number of template $U$. Object tracking success without false recognized in case of setting parameter $U$ is 2 to 10. Many templates have not bad influence on tracking accuracy in case of little shape change such as this experiment. Especially, it is least tracking error in case of $U$ by increase number of template $U$. It follows from Fig. 5 and Table II that any templates are not needed.

### Table 1. Total error.

<table>
<thead>
<tr>
<th>Value of $\alpha$</th>
<th>Total error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>13776.67</td>
</tr>
<tr>
<td>0.2</td>
<td>6906.09</td>
</tr>
<tr>
<td>0.3</td>
<td>6879.90</td>
</tr>
<tr>
<td>0.4</td>
<td>6879.90</td>
</tr>
<tr>
<td>0.5</td>
<td>479.94</td>
</tr>
<tr>
<td>0.6</td>
<td>486.89</td>
</tr>
<tr>
<td>0.7</td>
<td>489.02</td>
</tr>
<tr>
<td>0.8</td>
<td>471.93</td>
</tr>
<tr>
<td>0.9</td>
<td>456.73</td>
</tr>
<tr>
<td>1.0</td>
<td>4594.33</td>
</tr>
</tbody>
</table>

### Fig. 5. Graph of total error.

5.1.3 Experiment 3 – Size of Search Area $S$-

We inspect influence of search area size $S$ upon tracking accuracy. In here, proposed method is set up parameters as follows. Horizontal image size is 640 pixels, vertical image size is 480 pixels, size of collation area $M \times M$ is 11 pixels $\times$ 11 pixels, constant value $p$ in $F$ matrix is 0.3, quantize width $\Delta \theta$ is $\pi/8$, noise threshold value $\Gamma$ is 20, noise covariance ratio $v_{so}$ is 0.2 and constant value $\alpha$ is 0.7. 100 candidate points are picked up in ascending order. As you can see in Fig. 6 and Table III, object tracking success in case of setting parameter $S$ is 101, 201, 301, 401, 501, 601 (pixels) and search all pixels.

### Table 2. Total error.

<table>
<thead>
<tr>
<th>Template $U$</th>
<th>Total error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5050.16</td>
</tr>
<tr>
<td>2</td>
<td>461.79</td>
</tr>
<tr>
<td>3</td>
<td>415.34</td>
</tr>
<tr>
<td>4</td>
<td>410.30</td>
</tr>
<tr>
<td>5</td>
<td>425.06</td>
</tr>
<tr>
<td>6</td>
<td>416.72</td>
</tr>
<tr>
<td>7</td>
<td>423.05</td>
</tr>
<tr>
<td>8</td>
<td>421.70</td>
</tr>
<tr>
<td>9</td>
<td>431.54</td>
</tr>
<tr>
<td>10</td>
<td>438.08</td>
</tr>
</tbody>
</table>

### Fig. 6. Graph of total error.

5.2 Values of parameters for tracking

According to above Experiment 1 to Experiment 3, parameters for exactly tracking are presented as follows.

### Table 3. Total error.

<table>
<thead>
<tr>
<th>Search Area $S$</th>
<th>Total error</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>458.89</td>
</tr>
<tr>
<td>201</td>
<td>494.06</td>
</tr>
<tr>
<td>301</td>
<td>470.08</td>
</tr>
<tr>
<td>401</td>
<td>477.09</td>
</tr>
<tr>
<td>501</td>
<td>463.20</td>
</tr>
<tr>
<td>601</td>
<td>456.73</td>
</tr>
<tr>
<td>all pixels</td>
<td>456.73</td>
</tr>
</tbody>
</table>
5.3 Values of parameters for tracking

Experiment of Tracking to Moving Ant under low texture

We explain the case where tracking is success and compare with conventional method [7] which is tracking method based on orientation code matching and motion vector. Size of real image is $640 \times 480$. It is the same condition as experiment in Section 5.1. Image data is obtained by separating real-movie at each 30 frames par second. Illumination intensity of the image is $1322$ lx, Fig. 7(a) is experimental result of conventional method. This method is set up each parameters as described below, size of search area $S \times S$ is 201 pixels $\times$ 201 pixels, size of collation area $M \times M$ is 11 pixels $\times$ 11 pixels, quantize width $\Delta \theta$ is $\pi / 8$, noise threshold value $\Gamma$ is 20 and discriminate range $R = 60$. According to this experimental result, exactly tracking is difficult based on conventionally method. In the opposite Fig. 7(b) is represented this experimental result of proposal method, this method is set up each parameters as described below, size of search area $S \times S$ is 201 pixels $\times$ 201 pixels, size of collation area $M \times M$ is 11 pixels $\times$ 11 pixels, constant value $p$ in $F$ matrix is 0.3, quantize width $\Delta \theta$ is $\pi / 8$, noise threshold value $\Gamma$ is 20, noise covariance ratio $\nu_o$ is 0.2, constant value $\alpha$ is 0.7, and the number of template $U$ is 3 pieces. According to this experimental result, exactly tracking is possible from designated point in a first frame to 300th frame (about 10 seconds). Fig. 7(b) is last frame of experimental images. In Fig. 7(b), red dot represents trajectory of collation position. And green square represents search area, red square represents collation area. Fig. 7(c) is image of Fig. 7(b)’s orientation code information. In Fig. 7(c), black-dot presents trajectory of collation position. Black-square shows collation area. Our proposed method can be exactly tracking in case of orientation codes appear in the form of target moving object.

5.4 Experiment of Tracking to Moving Ant under High Texture

We explained the case of coursed false detection. This experiment used image that not appeared clearly moving object’s orientation code than before section. Fig. 8(a) is represented this experimental result. Fig. 8(a) is 21st frame image, in this image, red dot represents trajectory of collation position. Green square presents the search area and red square means false collation area.  Fig. 8(b) shows image of Fig. 8(a)’s orientation code information. In this image, black-dot represents trajectory of collation position. Black-square is false collation area. This experiments illumination intensity of the image is $2020$ lx. Proposal method is set up each parameter as described below, horizontal image size is 640 pixels, vertical image size is 480 pixels, and the size of search area $S \times S$ is 201 pixels $\times$ 201 pixels. The size of collation area $M \times M$ is 9 pixels $\times$ 9 pixels, constant value $p$ in $F$ matrix is 0.3, quantize width $\Delta \theta$ is $\pi / 8$, noise threshold value $\Gamma$ is 100, noise covariance ratio $\nu_o$ is 0.2, constant value $\alpha$ is 0.8 and the number of template $U$ is 10 piece. According to this experimental result, exactly tracking is possible from designated point in a first frame to 21st frame. But orientation code is not appeared clearly and false recognition is caused in 21st frame. Exactly tracking is difficult by proposal method, in case of orientation code of target object shape is not appeared clearly and many background orientation codes are appeared such as Fig. 8(b).
6. CONCLUSIONS

In this paper, we proposed a tracking method based on orientation code and an extended Kalman filter. Then, we consider the case where it cannot ignore the back ground texture on image. First, we explained multi-templates matching method with orientation code. Next we explained estimate of moving object position based on an extend Kalman filter, and we explained processing flow of proposal method. Final, experiment using image by video camera, and inspect problem and availability of proposal method. Future works are rethinking on equation of an extend Kalman filter for improvement of estimate accuracy, and rethinking on template matching method. We think of correspond to cannot obtained clearly orientation code of moving object shape, and estimate accuracy is down by in case of target object move high speed.

REFERENCES


