Robust Moving Object Extraction and Tracking Method Based on Matching Position Constraints

Tetsuya OKUDA††, Nonmember, Yoichi TOMIOKA††, Member, and Hitoshi KITAZAWA*, Fellow

SUMMARY Object extraction and tracking in a video image is basic technology for many applications, such as video surveillance and robot vision. Many motion object extraction and tracking methods have been proposed. However, they fail when the scenes include illumination change or light reflection. In order to maintain shape constancy, several methods have been proposed: Horn-Schunk [1], Lucas-Kanade [2], SIFT feature tracking [3], and exclusive block matching [4]. However, it is impossible to obtain the flow that constitutes the correspondence of feature points accurately using only appearance information. The appearance of an object can vary because of illumination changes or light reflection as it moves, as shown in Fig. 1. Moreover, if an area of an object is uniform in color, the extracted flows on the object can intersect each other. Therefore, we should consider the shape information when obtaining the correspondence of feature points.

In order to maintain shape constancy, several methods that minimize the energy function expressed by Eq. (1) have been proposed.

\[
E = \sum_{p} d(p, p') + \lambda \sum_{p,q} s(p, p', q, q').
\]  

1. Introduction

For many applications in which video image analysis is implemented, such as video surveillance and robot vision, object extraction and tracking is a basic requirement. In particular, to achieve action recognition, it is necessary to obtain a detailed motion vector between consecutive frames. Many motion object vector extraction methods have been proposed: Horn-Schunk [1], Lucas-Kanade [2], SIFT feature tracking [3], and exclusive block matching [4]. However, it is impossible to obtain the flow that constitutes the correspondence of feature points accurately using only appearance information. The appearance of an object can vary because of illumination changes or light reflection as it moves, as shown in Fig. 1. Moreover, if an area of an object is uniform in color, the extracted flows on the object can intersect each other. Therefore, we should consider the shape information when obtaining the correspondence of feature points.

In order to maintain shape constancy, several methods that minimize the energy function expressed by Eq. (1) have been proposed. Here, \( p \) and \( q \) are feature points in the current frame and \( p' \) and \( q' \) are those in the previous frame, which are matched with \( p \) and \( q \), respectively. The similarity is calculated from the data term \( [d(p, p')] \) and smoothness term \( [s(p, p', q, q')] \). The former represents visual feature similarity, and the latter represents the shape similarity [5]. We can obtain the correspondences of the blocks by solving the minimization problem in Eq. (1) [6]. However, this problem cannot be solved strictly because we have to solve the quadratic assignment problem. Therefore, it is approximated by using the steepest descent method [7], a genetic algorithm (GA) [8], successive convexification [9], and the iteration of visual feature optimization and shape optimization [5].

In this study, we propose a new method for robustly obtaining the correspondence of feature points by applying matching position constraints on a cost matrix when the object shape does not change suddenly. Moreover, we show a condition of the object translation and rotation that allows us to consider the matching position on a line.

In the next section, we explain exclusive block matching as a basis for the proposed method. In Sect. 3, we introduce a theory for the alignment of matching positions. In Sect. 4, we describe a method for applying the matching position constraint for exclusive block matching. In Sect. 5, we show the experimental results of flow extraction and compare them with those of previous methods.

2. Exclusive Block Matching [4]

We explain the exclusive block matching method, which is a basic component of the proposed method. Exclusive block matching is a method of detecting the flow by obtaining the correspondence of the blocks between two successive frames; the procedures of exclusive block matching are summarized in Fig. 2. First, we divide the images of the prev-
vious frame and the current frame into blocks of $8 \times 8$ pixels. Then, we generate a cost matrix by calculating the similarity between the current and previous frame’s blocks. We extend the cost matrix as shown in Fig. 3 for matching to the background image (Bg) and appearance of the new object (Create). A current frame’s block that resembles background image block matches a diagonal element of Bg matrix. HSV histogram and HOG [10] are employed as the feature of each block, and the Bhattacharyya distance is employed for calculating the similarity. The optimum correspondence of the block is determined such that the total cost is minimized, by solving the linear assignment problem.

3. Alignment of Matching Positions

For each motion of an object, the matching positions of a cost matrix, as shown in Fig. 3, are arranged in a certain pattern depending on the motion. In particular, when the object shape is not changed significantly and the translation and rotation are small, the matching points are arranged approximately in a certain line, as shown in Fig. 4. By using this matching position constraint, robust tracking can be achieved. In this section, we analyze the patterns of matching positions for basic movements: translation and rotation.

For a cost matrix, the row index $i$ represents the $i$-th block of the current frame in raster scanning order. Similarly, the column index $j$ represents the $j$-th block of the previous frame. When the $i$-th block of the current frame is matched with the $j$-th block of the previous frame, element $c(i, j)$ is a matching position on the cost matrix. An object can be deemed to consist of some blocks, and they move together while the object moves. For example, when an object is magnified, as shown in Fig. 5 (a), the matching positions of the blocks, indicated by “x” in Fig. 5 (a), are arranged in a line on the cost matrix, as shown in Fig. 5 (b). It should be noted that this line appears not on the screen but on the cost matrix.

Since the matching positions are sparse on the cost matrix, we convert the cost matrix by collecting them in the compartment of the upper left hand corner of the cost matrix to facilitate understanding that the matching positions are arranged in a certain figure. The collected matching pattern depends on the block scanning direction for generating the initial cost matrix. The cost matrix generated by raster scanning in the X-Y (Y-X) direction, as shown in Fig. 6, is called X-Scan, (Y-Scan). For X-Scan (Y-Scan), the row and column of a matching position correspond to the x-coordinates (y-coordinates) of its current and previous frame’s blocks, respectively. X-Scan (Y-Scan) show alignment of matching positions clearly, although the information of y-direction (x-direction) disappear in X-Scan (Y-Scan). Then, we solve the assignment problem on the original (not collected) cost matrix.

We assume that a CG synthesized object is moving, and we calculate its screen coordinates to show the alignment of the matching positions. We show the simulation results of the X-Scan and Y-Scan pattern when an object moves to the right in Fig. 7, and when an object approaches the camera in Fig. 8. Different colors are assigned to the matching positions according to the face to which the corresponding block belongs. We see that the matching positions are arranged approximately on lines for both X-Scan and Y-Scan.
means matching position and corresponds to the same letter in the X-Scan
tation in the world coordinate.
the same relational expression between their x-coordinates
relational expression is approximately linear. In the follow-
y-coordinates) in the current and previous frames, and the

Fig. 6  Generation of X-Scan and Y-Scan. Each letter in the cost matrix
means matching position and corresponds to the same letter in the X-Scan
(Y-Scan).

Fig. 7  X-Scan and Y-Scan for an object moving right.

Fig. 8  X-Scan and Y-Scan for an object approaching the camera.

In other words, all the matching points of an object satisfy
the same relational expression between their x-coordinates
(y-coordinates) in the current and previous frames, and the
relational expression is approximately linear. In the follow-
ing, we discuss the relational expression that is satisfied for
two kinds of basic movement: parallel displacement and ro-
tation in the world coordinate.

We define that $X_t = [X_t, Y_t, Z_t]^{T}$ is the world coordi-
nate of a point $q$ at time $t$, $x^t = [h_t x, h_t y, h_t]^T$ is the screen
coordinate that corresponds to $q$, $P$ is a projection matrix,
and $PA = P[R[t]]$ is a perspective projection matrix. That is,

$$x_t = PAX_t.$$  \hspace{1cm} (2)

For ease of explanation, we assume that the camera po-
sition is fixed at the origin of the world coordinate, the cam-
ner angle does not vary ($A = I$), and

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$ \hspace{1cm} (3)

where $f$ is the focal distance. From Eq. (2), the screen coor-
dinate $x_t, y_t$ can be expressed as

$$x_t = fX_t/Z_t,$$ \hspace{1cm} (4)

$$y_t = fY_t/Z_t.$$ \hspace{1cm} (5)

Here, we denote the time of the previous frame by $t$ and
the time of the current frame by $t + \Delta t$. We assume that $\Delta t$
is sufficiently small. The screen coordinate of current frame
can be expressed as:

$$x_{t+\Delta t} = x_t + \frac{dx_t}{dt} \Delta t;$$ \hspace{1cm} (6)

$$y_{t+\Delta t} = y_t + \frac{dy_t}{dt} \Delta t.$$ \hspace{1cm} (7)

Therefore, we can obtain the correspondence between
current frame and previous frame by calculating of the
screen coordinates $x$, $y$, and its time derivative $dx_t/dt, dy_t/dt$.

Here, an object is composed of multiple points. We
consider that many points on the object move at the same
time. Then, their matching positions form a certain figure
on the cost matrix. We assume that an object is a cuboid. $X_t$
$Y_t, Z_t$ have ranges represented by

$$\begin{align*}
X_a & \leq X_t \leq X_b, \\
Y_a & \leq Y_t \leq Y_b, \\
Z_a & \leq Z_t \leq Z_b.
\end{align*}$$ \hspace{1cm} (8)

$X_t$ is expressed by

$$X_t = B(t)X_0,$$ \hspace{1cm} (9)

where, $X_0 = [X_0 Y_0 Z_0]^{T}$ is the location at time $t = 0$, and
$B(t)$ is the affine transformation that includes translation and
rotation.

(1) Parallel displacement

Assuming that the movement of the object constitutes only
parallel displacement, we can write

$$B(t) = \begin{bmatrix} 1 & 0 & 0 & v_{xt} \\ 0 & 1 & 0 & v_{yt} \\ 0 & 0 & 1 & v_{zt} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$ \hspace{1cm} (10)

where $v_x, v_y$, and $v_z$ are the moving velocity of the object for
the X, Y, and Z axis, respectively.

By substituting Eq. (10) into Eq. (9), point \( \mathbf{X}_t = [X_t, Y_t, Z_t] \) is represented as

\[
\begin{bmatrix}
X_t \\
Y_t \\
Z_t
\end{bmatrix} = \begin{bmatrix}
X_0 + v_xt \\
Y_0 + v_yt \\
Z_0 + v_zt
\end{bmatrix}.
\]

From Eqs. (4) and (5), \( x_t \) and \( y_t \) are written as

\[
x_t = \frac{(X_0 + v_xt)f}{Z_0 + v_zt},
\]
\[
y_t = \frac{(Y_0 + v_yt)f}{Z_0 + v_zt}.
\]

Substituting these into Eqs. (6) and (7), we obtain

\[
x_{t+\Delta t} = x_t + \Delta t\left(\frac{f v_z}{Z_t} - \frac{f X_t v_x}{Z_t^2}\right),
\]
\[
y_{t+\Delta t} = y_t + \Delta t\left(\frac{f v_y}{Z_t} - \frac{f Y_t v_z}{Z_t^2}\right).
\]

First, we consider the points that have the same Z-coordinate as in the front face of the cuboid. Eliminating \( X_t \) and \( Y_t \) from Eqs. (14) and (15) using Eqs. (4) and (5), we obtain

\[
x_{t+\Delta t} = \left(1 - \frac{v_z \Delta t}{Z_t}\right) x_t + \frac{f v_x \Delta t}{Z_t},
\]
\[
y_{t+\Delta t} = \left(1 - \frac{v_y \Delta t}{Z_t}\right) y_t + \frac{f v_z \Delta t}{Z_t}.
\]

These equations show that the matching positions are located on a line on the cost matrix for the multiple points that have the same Z-coordinate, because \( x_{t+\Delta t} \) and \( y_{t+\Delta t} \) for the points satisfy the same linear expression.

Second, we consider the points that have the same X or Y-coordinate as in the side, top, and bottom faces of the cuboid. We transform these equations using Eqs. (4) and (5) as

\[
x_{t+\Delta t} = \left(1 + \frac{v_x \Delta t}{X_t}\right) x_t - \frac{v_x \Delta t}{f X_t} x_t^2,
\]
\[
y_{t+\Delta t} = \left(1 + \frac{v_y \Delta t}{Y_t}\right) y_t - \frac{v_y \Delta t}{f Y_t} y_t^2.
\]

\( x_{t+\Delta t} \) and \( y_{t+\Delta t} \) are represented as a quadratic function of \( x_t, y_t \). A parabola appears on the X-Scan if the sample points have the same X-coordinate. Similarly, a parabola appears on the Y-Scan if the sample points have the same Y-coordinate. In particular, when the object is at a distance from the camera and its movement is sufficiently small, the parabola has a gentle curve. We can approximate the parabola by a line because of the quantization of blocks. Then, we calculate the condition of the object’s moving speed and distance for linear approximation. We can rewrite Eqs. (18) and (19) as

\[
x_{t+\Delta t} = a_x x_t + b_x + E_x,
\]
\[
y_{t+\Delta t} = a_y y_t + b_y + E_y,
\]

where, \( a_x, b_x, a_y, \) and \( b_y \) are constant and \( E_x \) and \( E_y \) are non-linear terms.

We define \( \max(\cdot) \) and \( \min(\cdot) \) as its maximum and minimum value when \( X_t, Y_t, Z_t \) is in the range represented by Eq. (8), respectively. It is necessary that \( E_x \) and \( E_y \) satisfy the following equations for the linear approximation.

\[
\max(E_x) - \min(E_x) < \frac{\delta}{2}
\]
\[
\max(E_y) - \min(E_y) < \frac{\delta}{2}.
\]

where \( \delta \) is the block size on the screen.

Thus, \( E_x \) and \( E_y \) of Eqs. (18) and (19) are

\[
E_x = -\frac{v_x \Delta t}{f X_t} x_t^2,
\]
\[
E_y = -\frac{v_y \Delta t}{f Y_t} y_t^2.
\]

Substituting Eqs. (4) and (5) and removing the screen coordinates, we obtain

\[
E_x = -\frac{\Delta t f X_t}{Z_t^2},
\]
\[
E_y = -\frac{\Delta t f Y_t}{Z_t^2}.
\]

Therefore, from Eq. (22), the condition of linear approximation is

\[
\frac{\frac{v_x \Delta t}{Z_t} X_b}{Z_b} - \frac{v_x \Delta t F X_a}{Z_a} < \frac{\delta}{2}
\]
\[
\frac{\frac{v_y \Delta t}{Z_t} Y_b}{Z_b} - \frac{v_y \Delta t F Y_a}{Z_a} < \frac{\delta}{2}.
\]

here, we assume that \( X_a, X_b, Y_a, Y_b, Z_a, Z_b > 0 \).

We show an example of the conditions under which a figure formed by the matching positions becomes a line in the cost matrix. We devide the screen into 40 \times 30 blocks. As shown in Fig. 9, we assume that a camera whose scene width is equal to the distance from camera. The angle of view is about 53.1 degrees. In this case, the scene width is 40\( \delta \), therefore, \( f = 40\delta \). We assume that the moving object is a cube the length of the sides of which is 1 m. We define the range of the object we can see as \( X = 5 \) m, \(-0.5 \) m \( \leq Y \leq 0.5 \) m, \( 10 \) m \( \leq Z \leq 11 \) m. In this case, we see Fig. 9 The scene where we show the condition under matching positions align.
that the condition of linear approximation is $|v_c| \Delta t < 1.25$ m.

(2) Rotation around the X-axis and Y-axis

Here, we explain the matching position alignment for rotation around the Y-axis. For rotation around the X-axis, we can write similar equations. When the object rotates around the Y-axis, we can write it using

$$B(t) = \begin{bmatrix} \cos \omega t & -\sin \omega t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \sin \omega t & 0 & \cos \omega t & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (28)$$

where, $\omega$ is the rotation speed.

Substituting Eq. (28) into Eq. (9) and converting in a manner similar to that used for the parallel displacement, we obtain

$$x_{t+\Delta t} = x_t - \frac{\omega \Delta t}{f} x_t^2 - f \omega \Delta t,$$

$$y_{t+\Delta t} = y_t - \frac{\omega \Delta t}{f} x_t y_t. \quad (29)$$

This includes non-linear terms $x_t^2$ and $x_t y_t$. We adapt linear approximation in the same way as Eqs. (18) and (19). Using Eqs. (4) and (5), the non-linear terms $E_x$ and $E_y$ of Eqs. (29) and (30) are written as

$$\frac{-\omega \Delta t}{f} x_t^2 = -\frac{\omega \Delta t f X_t^2}{Z_t^2}, \quad (31)$$

$$\frac{-\omega \Delta t}{f} x_t y_t = -\frac{\omega \Delta t f X_t Y_t}{Z_t^2}. \quad (32)$$

The condition under which matching positions align on a line is the same as Eq. (22), and we obtain the following condition.

$$\max \left( -\frac{\omega \Delta t f X_t^2}{Z_t^2} \right) - \min \left( -\frac{\omega \Delta t f X_t^2}{Z_t^2} \right) < \delta \frac{2}{2} \quad \text{and}$$

$$\max \left( -\frac{\omega \Delta t f X_t Y_t}{Z_t^2} \right) - \min \left( -\frac{\omega \Delta t f X_t Y_t}{Z_t^2} \right) < \delta \frac{2}{2}. \quad (33)$$

We show an example of conditions under which a figure formed by the matching positions becomes a line in the cost matrix. We consider the same object as that introduced in the example of parallel displacement. Here, we obtain $\max(X_t^2/Z_t^2) = 25/100$, $\min(X_t^2/Z_t^2) = 25/121$, $\max(X_t Y_t/Z_t^2) = 0.25/100$, and $\min(X_t Y_t/Z_t^2) = 0.25/100$. In this case, we see that the condition of linear approximation is $|\omega| \Delta t < 14.3$ deg.

(3) Rotation around the Z-axis

For the situation where the object rotates around the Z-axis, we can write

$$B(t) = \begin{bmatrix} \cos \omega t & -\sin \omega t & 0 & 0 \\ \sin \omega t & \cos \omega t & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (34)$$

Substituting Eq. (34) into Eq. (9) and transforming it in the manner as that used for parallel displacement, we obtain

$$x_{t+\Delta t} = x_t - \omega \Delta t y_t,$$

$$y_{t+\Delta t} = y_t + \omega \Delta t x_t. \quad (35)$$

From this, we see that $y_t$ is necessary for representing $x_{t+\Delta t}$ and $x_t$ is necessary for representing $y_{t+\Delta t}$. Therefore, $x_{t+\Delta t}$ and $y_{t+\Delta t}$ are expressed by the following equations.

$$\left\{ \begin{array}{l} x_{t+\Delta t} = a x_t + b y_t + c \\ y_{t+\Delta t} = d x_t + e y_t + f \end{array} \right. \quad (37)$$

These equations represent not a line but a plane in four-dimensional space. However, we call it an equation of a line to avoid confusion.

4. Solving Linear Assignment Problem to Satisfy Matching Position Constraint

In this section, we explain the method for obtaining the solution of the linear assignment problem that satisfies the matching position constraint. We solve the linear assignment problem using the Hungarian method[11]. The Hungarian method is executed as follows.

(1) Subtract the minimum value of each row from each element of the row.

(2) If we can choose $n$ independent zeros ($n$: matrix size), which means every row and every column includes only one zero, the position of the zeros indicates the matching positions.

(3) Cover all zeros by the fewest possible horizontal or vertical lines.

(4) Subtract the minimum value of uncovered elements from them and add the value to the elements that are covered with both horizontal and vertical lines; go to (2).

Figure 10 shows an example of the Hungarian method for $n = 4$.

We can guess that the minimum value of each row tends to the solution of the assignment problem. They appear as zero-cost after we subtract the minimum value of each row from each element of the row. Therefore, if multiple zero-cost elements located on a line, we consider they
are correct matching positions. For the proposed method, we extract lines including as many zero-cost elements as possible on the cost matrix. We modify the values of the cost matrix based on the extracted lines.

Zero-cost elements that are on an extracted line and have no other zero-cost elements in the same row and same column must be determined as matching positions. We select them as a matching solution. By omitting them from the assignment, we can reduce the calculation time.

“Nonzero-cost elements on an extracted line” and “zero-cost elements near an extracted line” are considered to have been caused by the shape distortion of an object or an appearance change caused by an illumination change or reflection. We cannot distinguish whether or not the object has in fact transformed. This is the trade-off between the appearance change and shape change of the object. We should consider it an appearance change when the matching position is near an extracted line because the shape of an object does not change suddenly. Therefore, we decrease the cost of non-zero elements on an extracted line” to raise the matching possibility. An element that is not on an extracted line and has low color similarity should not be a matching position. Therefore, we make its cost infinity so that it is not chosen.

The procedure of the proposed method is described below, and illustrated in Fig. 11.

1. Subtract the minimum value of each row from each element of the row.
2. On the cost matrix, extract the lines that cover as many zero-cost elements as possible using RANSAC [12].
3. Mark the zero-cost element on the line as (a).
4. Decrease the cost of non-zero elements on the line by $\Delta c$ and mark as (b).
5. An element the distance of which from the line is less than or equal to 1 and the color cost (distance) of which is less than a predefined threshold $th$ is marked (c).
6. Set the cost value of an element that is not marked as infinity.
7. For each row that has only one marked element, if there is no marked element in the same column, determine the element as a matching position, and exclude the row from the assignment problem.
8. Solve the assignment problem for the remaining rows using the Hungarian method.

We use RANSAC as a line extraction method. If the number of “zero-cost elements” that align on a continuous line is four and over, we consider them a meaningful line for the matching position constraint. The extracted lines should be continuous for 24 neighbor blocks on both X-Scan and Y-Scan. 

5. Experimental Results

We show the flow extraction results of the proposed method for static and moving camera images. The parameters in (4) and (5) of the proposed method procedure in Sect. 4 are fixed values. We set the value $\Delta c$ in (4) is 50, and the value $th$ in (5) is 50, which are one a ninth of Create (appearance of the new object) threshold 450, where similarity is normalized to 0 to 1000. We see that these values are not sensitive to scenes for these experiments.

A comparison of the results for the static camera images in Figs. 12 and 13 shows that the proposed method extracted more precise flows than exclusive block matching without matching position constraint. The results for a moving camera shown in Figs. 14 and 15 demonstrate that by applying the proposed matching position constraint, the flows are appropriately extracted. The moving object and
the background are separated appropriately by the flows.

We also show an application for template matching. For template matching, instead of the previous frame, a template is used. The matching position constraints are applied as for object tracking. In Table 1, we compared the proposed method with six existing methods in [14]–[18], and [8] for David Indoor [19]. We define that matching is successful when all blocks of the template match to the target. The proposed method and the methods in [18] and [8] achieved the highest accuracy.

Moreover, in Table 2, we show a comparison with the method in [8] for six data including David Indoor. In these experiments, we also adopted Kalman Filter [13] to predict the object location. We reduced the cost of the blocks near the object position predicted by the Kalman filter. By using the Kalman Filter, the number of successful frames is improved for three data: David Outdoor, Sylvester, and Road Bridge. As compared with the method in [8], the proposed method realized more robust template matching. Examples of the template matching results are shown in Fig. 16. For template matching, both the method in [8] and the proposed method adopt HOG Context [8] as a block feature. As we see from the matching result in Fig. 16, the matching blocks preserve the shape of the target in the proposed method. Moreover, matching is succeeded in scenes David Outdoor and Sylvester that include the frame which has shaded target. It means that the matching position constraint is robust for illumination change. However, matching is failed
in scene Car 4. It is considered that illumination change is too strong.

In Table 3, we show the calculation time per frame for four methods: exclusive block matching without matching position constraint, the method in [8], the proposed methods with and without Kalman Filter. The calculation time of the proposed method is comparable with the original exclusive block matching, and it is shorter than that of the method in [8].

We also evaluate the performance of the proposed method by Rui Yao’s criteria [20]. We use the overlap ratio, center location error, and success frame for evaluation. Overlap ratio is defined as $R_{\text{overlap}} = \frac{\text{Area}(B_T \cap B_{GT})}{\text{Area}(B_T \cup B_{GT})}$, where $B_T$ is the tracking bounding box and $B_{GT}$ is the ground truth bounding box. If the overlap ratio is larger than 0.5, the matching is considered to be successful. We compare ‘Without matching position constraint’, ‘GA’, ‘Proposed method’ and ‘Proposed method with K.F.’. The results are shown in Tables 4, 5, and 6. The proposed method generated more accurate results than without matching position constraint and GA.

### Table 4 Comparison of overlap ratio. “without MCP” is “matching position constraint”.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Without MCP</th>
<th>GA</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Indoor</td>
<td>0.716</td>
<td>0.747</td>
<td>0.750</td>
</tr>
<tr>
<td>David Outdoor</td>
<td>0.604</td>
<td>0.619</td>
<td>0.695</td>
</tr>
<tr>
<td>Sylvester</td>
<td>0.579</td>
<td>0.609</td>
<td>0.702</td>
</tr>
<tr>
<td>Car4</td>
<td>0.526</td>
<td>0.316</td>
<td>0.614</td>
</tr>
<tr>
<td>Backpacker</td>
<td>0.350</td>
<td>0.478</td>
<td>0.763</td>
</tr>
<tr>
<td>Road Bridge</td>
<td>0.660</td>
<td>0.655</td>
<td>0.684</td>
</tr>
</tbody>
</table>

### Table 5 Comparison of average center location errors (pixel). “without MCP” is “matching position constraint”.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Without MCP</th>
<th>GA</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Indoor</td>
<td>13.07</td>
<td>9.09</td>
<td>7.07</td>
</tr>
<tr>
<td>David Outdoor</td>
<td>18.30</td>
<td>16.43</td>
<td>7.66</td>
</tr>
<tr>
<td>Sylvester</td>
<td>19.23</td>
<td>15.91</td>
<td>8.68</td>
</tr>
<tr>
<td>Car4</td>
<td>24.77</td>
<td>21.91</td>
<td>10.27</td>
</tr>
<tr>
<td>Backpacker</td>
<td>21.56</td>
<td>22.69</td>
<td>5.46</td>
</tr>
<tr>
<td>Road Bridge</td>
<td>15.98</td>
<td>15.59</td>
<td>12.68</td>
</tr>
</tbody>
</table>

### Table 6 Comparison of number of success frames (Overlap ratio is greater than 0.5). “without MCP” is “matching position constraint”.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Without MCP</th>
<th>GA</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Indoor</td>
<td>629</td>
<td>670</td>
<td>704</td>
</tr>
<tr>
<td>David Outdoor</td>
<td>84</td>
<td>87</td>
<td>105</td>
</tr>
<tr>
<td>Sylvester</td>
<td>864</td>
<td>940</td>
<td>1207</td>
</tr>
<tr>
<td>Car4</td>
<td>358</td>
<td>346</td>
<td>559</td>
</tr>
<tr>
<td>Backpacker</td>
<td>64</td>
<td>50</td>
<td>109</td>
</tr>
<tr>
<td>Road Bridge</td>
<td>48</td>
<td>49</td>
<td>58</td>
</tr>
</tbody>
</table>

### 6. Conclusion

We showed that the matching positions are located on a line on the cost matrix and that the method of calculation that retains the shape through exclusive block matching is effective. In the experiment, our method achieved more accurate tracking results in a shorter time than the previous method for static and moving camera images. For practical application, we should improve the calculation time. This remains for future work.
References


Tetsuya Okuda received his B.S. degree in Electrical and Electronic Engineering from Tokyo University of Agriculture and Technology, Japan, in 2013. He is currently pursuing a master’s course at the university. His research interests include image processing.

Yoichi Tomioka received his B.E., M.E., and D.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2005, 2006, and 2009, respectively. He was a research associate at Tokyo Institute of Technology until 2009. He was an assistant professor in the Division of Advanced Electrical and Electronics Engineering at Tokyo University of Agriculture and Technology until 2015. Since 2015, he has been an associate professor in the School of Computer Science and Engineering at University of Aizu. His research interests include image processing, security systems with mobile robots, VLSI package design automation, and combinational algorithms. He is a member of IEEE and IPSJ.

Hitoshi Kitazawa received his B.S., M.S., and Ph.D. degrees in Electronic Engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1974, 1976, and 1979, respectively. He joined the Electrical Communication Laboratories, Nippon Telegraph and Telephone Corporation (NTT), in 1979. Since 2002, he has been a professor at the Tokyo University of Agriculture and Technology. His research interests include VLSI CAD algorithms, computer graphics, and image processing. He is a member of IPSJ.