

## A FAST RANDOMIZED GENERALIZED HOUGH TRANSFORM FOR ARBITRARY SHAPE DETECTION

SHIH-HSUAN CHIU<sup>1,\*</sup>, CHE-YEN WEN<sup>2</sup>, JUN-HUEI LEE<sup>1</sup>  
KUO-HUNG LIN<sup>1</sup> AND HUNG-MING CHEN<sup>1</sup>

<sup>1</sup>Department of Materials Science and Engineering  
National Taiwan University of Science and Technology  
No. 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan

\*Corresponding author: schiu@mail.ntust.edu.tw  
{ D9104301; D9304301; D9704004 }@mail.ntust.edu.tw

<sup>2</sup>Department of Forensic Science  
Central Police University  
No. 56, Shujen Rd., Takang Village, Kueishan Hsiang, Taoyuan 33304, Taiwan  
cwen@mail.cpu.edu.tw

Received September 2010; revised January 2011

**ABSTRACT.** *The well-known arbitrary shape detection technology, generalized Hough transform (GHT) has the drawbacks of heavy computations (one-to-many or 1-to- $n$  mapping) and storage requirements (voting space and entry number). Some  $n$ -to-1 mapping approaches have been proposed for improving the performance of GHT, such as the FGHT (fast generalized Hough transform), ADPHT (Adaptive dual-point Hough transform) and GFHT (generalized fuzzy Hough transform). The  $n$ -to-1 mapping approaches use  $n$  feature points as one set to produce one increment of the vote in the accumulator array. Although the  $n$ -to-1 mapping approaches can efficiently reduce the spurious voting, the improvement for the heavy computations is limited due to redundant mapping. In this study, we propose the fast randomized generalized Hough transform (FRGHT), which uses a randomized waypoint strategy to choose feature line segments randomly and consecutively. With this strategy, not only the required entry number of the table to avoid redundant mapping can be reduced dramatically, but also the relationship between sets can be found to reduce the spurious voting. The experimental results of FRGHT show better performance than the previous modified GHT's (FGHT and GFHT) in voting efficiency, less computation costs and storage requirements (entry number).*

**Keywords:** Generalized Hough transform, Random waypoint strategy, Arbitrary shape detection

**1. Introduction.** Pattern recognition has been applied in many fields. For example, H. Benitez-Perez and A. Benitez-Perez proposed a two-stage method for feature extraction and classification for fault diagnosis patterns [1], Ohno and Murao constructed an image retrieval system by using a similarity measuring method based on reference vectors [2], and Miyata et al. constructed a road sign recognition system based on color feature extraction and its components analysis by using dynamic image processing [3]. Shape detection is crucial in pattern recognition. The generalized Hough transform (GHT) is a well-known technology for arbitrary shape detection [4,5]. The main process of the GHT is a kind of a brute voting strategy for evidence gathering [6,7]. On a template image ( $T$ ), we first choose a reference point ( $R$ ) inside the target shape. For each point on the shape of  $T$ , we compute the geometric arrangements (e.g., gradient, orientation and vector) based on its relative position to  $R$ . We build the reference table (R-table) by setting the gradient as the entry, and the orientation and vector as the indexes. For

the voting process of shape detection, the contour point in the input image ( $I$ ), which has the same entry information, will vote for all possible transformations (i.e., scaling, rotation and translation) by computing the indexes. The most voted-for cell and its related parameters specify the most likely transformation relationship between  $T$  and  $I$ . However, spurious votes or wrong evidences may be generated during the voting process. For each entry, only one index correctly presents the transformation, and the other ones are wrong evidences. Unfortunately, wrong evidences are rich in practical applications. In other words, the 1-to- $n$  mapping for a shape point produces the increment of  $n$  cells that may be spurious. We can see the main drawbacks of the GHT are heavy computations (1-to- $n$  mapping) and storage requirements (voting space and entry number).

Several strategies are proposed for improving the brute voting: 1) the randomized strategy: a so-called randomized generalized Hough transform (RGHT) was proposed by combining both the GHT and RHT [8]. Although the RGHT uses the randomized strategy to improve the GHT, it still falls into the category of the 1-to- $n$  mapping approach. 2) The geometric arrangement strategy for reducing spurious votes: Tsai proposed an improved generalized Hough transform [9]. The improved GHT has two properties: a) employ the circle fitting method as the constraint to eliminate false matches of points as many as possible; b) employ the center position of a circle and object edge points to form a vector for estimating the rotational angle and translations. However, the circle approximation is insufficient for arbitrary shape. Some  $n$ -to-1 mapping approaches based upon the geometric arrangement strategy have been proposed. The strategy uses geometric invariant properties and finds  $n$  feature points on a predefined pattern as a feature set to produce only one increment in the accumulator array. The predefined patterns can be any significant one, such as vertical forms [10], parallel forms [11,12] and triangle forms [13-18]. Obviously, the invariant features of geometric arrangement strategy can effectively reduce the amount of wrong evidence gathering [11].

Among previous mentioned forms, the pole-polar triangle is the most popular one since the building of the feature sets is based on computation (intersection of pole-polar relationship) instead of brute-searching in particular geometric relationship [11]. Ser and Siu proposed a dual-point generalized Hough transform (DPGHT) [14]. They examined the index angle with one pre-defined relation (e.g., parallel gradient directions). The performance is dependent on the pre-defined relation. However, it is not easy to find a well relation for arbitrary shapes. Chau and Siu proposed an improved version of the DPGHT [16]. They used statistics to find the best characteristic angle as the relation. However, only one characteristic angle can be employed for voting process. For some complex shapes, more than one relation will be needed to describe the shapes. Chau and Siu proposed the adaptive dual-point Hough transform (ADPHT), which provided a way to find multiple characteristic angles sequentially [17]. The ADPHT utilized the concept that one entry per index is allowed to be stored inside the R-table to make the number of entries per index as small as possible. It results in reducing the spurious votes, while reducing the computational time for voting process. However, the ADPHT still uses the point-based framework. This framework requires a time-consuming process to search the characteristic angle, and it may not suitable for real-time applications [17].

Kimura and Watanabe proposed the fast generalized Hough transform (FGHT) based on the geometric arrangement strategy [15], which was the extension of the chord-tangent transform [13]. The FGHT divides the contour image (the template image,  $T$ ) into several sub-blocks and then approximates the partial contour by a line segment in each block. The "pole-polar triangle" of the  $n$ -to-1 mapping approach is then applied to estimate the entry for the extended C-table. Each entry consists of two angles inside the pole-polar triangle form. In addition, the FGHT uses the third line segment as a check to get rid

of spurious votes. However, the voting process of the FGHT may be dispersed while the vague noise appears near to the shape of objects. Its improved version, the generalized fuzzy Hough transform (GFHT) was proposed to be suitable for noisy shape detection [18]. The GFHT uses the Gaussian model as a membership function and collects all votes in the fuzzy region. This fuzzy strategy improves the efficiency of the FGHT in noisy and vague images. Although the FGHT and GFHT can reduce the spurious votes, the redundant sets are still unavoidable when building the extended C-table. In this study, the fast randomized generalized Hough transform (FRGHT) is proposed. It is an  $n$ -to-1 mapping based method that can reduce the spurious votes. Besides, it has the following important properties:

- (1) A novel “waypoint strategy” concept is applied for constructing entries randomly and consecutively: it reduces the required storage space of the extended C-table.
- (2) The “consecutive entry matching” approach is applied for voting process: it can avoid redundant mapping dramatically; meanwhile, it can improve voting efficiency and reliability.

This paper is organized as follows: Section 2 briefly summarizes the main scheme of the FGHT; and Section 3 describes the analysis of the phenomena of the spurious votes and redundant mapping of two line-based GHT’s (the FGHT and GFHT); Section 4 introduces the proposed method (the FRGHT); Section 5 demonstrates the experimental results and discussions; and finally, the conclusions are shown in Section 6.

**2. The Fast Generalized Hough Transform.** Since the concept of the proposed method (FRGHT) comes from the fast generalized Hough transform (FGHT), we summarize three main steps of the FGHT in this section: “Line segment approximation”, “Creation of the extended C-table” and “Shape Recognition.”

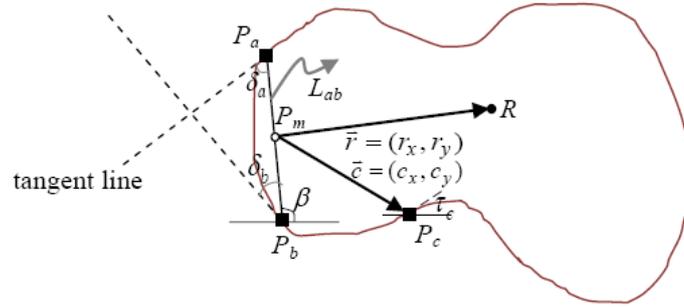
**2.1. Line segment approximation.** Let  $T$  and  $I$  be a template and an input contour image, respectively. Each of them is divided into blocks ( $B \times B$  pixels). Then, we approximate the contour in each block region by a line segment. The sets of line segments are defined as  $\{T_i | i = 1, 2, \dots, N\}$  in  $T$  and  $\{I_j | j = 1, 2, \dots, M\}$  in  $I$ .

**2.2. Creation of the extended C-table.** We arbitrarily choose a point from  $T$  as the reference point,  $R = (R_x, R_y)$ . In general,  $R$  can be the gravity center of the template contour. For each line segment pair ( $T_a$  and  $T_b$ ), we find their middle points ( $P_a$  and  $P_b$ ) as feature points. If the Euclidean distance  $L_{ab}$  between feature points  $P_a$  and  $P_b$  is longer than the predefined threshold  $L_{th}$ , they will be applied to construct the extended C-table. Each element ( $q_n$ ) of the table is represented by six indexes as:

$$q_n = \{ \delta_a(n), \delta_b(n), L_{ab}(n), \beta(n), \tau_c(n), \vec{c}(n), \vec{r}(n) \}, \quad (1)$$

where  $n = 1, \dots, Q$ , and  $Q$  is the entry number of the extended C-table. In Figure 1, it shows the geometric arrangement of the indexes and the structure of the extended C-table. The definitions of the six indexes are summarized as follows:

- (1)  $\delta_a, \delta_b$ : The interior angles formed by two line segments whose middle points are  $P_a$  and  $P_b$ , respectively.
- (2)  $L_{ab}$ : The Euclidean distance between  $P_a$  and  $P_b$ , i.e., the length of  $\overline{P_a P_b}$ .
- (3)  $\beta$ : The angle of  $\overline{P_a P_b}$ .
- (4)  $\tau_c$ : The angle of the tangent line at  $P_c$ , where  $P_c$  is called the check point.
- (5)  $\vec{c}$ : A vector from  $P_m$  to  $P_c$ , where  $P_m$  is the middle point of  $\overline{P_a P_b}$ .
- (6)  $\vec{r}$ : A vector from  $P_m$  to  $R$ , where  $P_m$  is the middle point of  $\overline{P_a P_b}$ .



(a)

$q_n$	Entry	Connecting line		Check point		Vector to $R$
		Length	Angle	Angle	Vector ( $\bar{c}$ )	( $\bar{r}$ )
$q_1$	$\delta_a(1), \delta_b(1)$	$L_{ab}(1)$	$\beta(1)$	$\tau_c(1)$	$(c_x(1), c_y(1))$	$(r_x(1), r_y(1))$
$q_2$	$\delta_a(2), \delta_b(2)$	$L_{ab}(2)$	$\beta(2)$	$\tau_c(2)$	$(c_x(2), c_y(2))$	$(r_x(2), r_y(2))$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

(b)

FIGURE 1. The illustration of geometric arrangement of a triangle defined by pole-polar relationship [18]: (a) a template image ( $T$ ); (b) the structure extended C-table of (a) that including entry:  $\delta_a$  and  $\delta_b$ , connecting line:  $L_{ab}$  and  $\beta$ , check point:  $\tau_c$  and  $(c_x, c_y)$ , and vector  $(r_x, r_y)$  to  $R$  (reference point)

**2.3. Shape detection.** The shape detection is executed by a voting process in the parameter space. In general, the parameter space is represented by a 4-dimensional accumulator array,  $A(v_x, v_y, k, \theta)$ , which is regarded as a voting array. The four parameters describe the possible transformation of  $T$  and  $I$ :  $(v_x, v_y)$  is the correspondent coordinate of reference point in  $I$ ,  $k$  is the scaling factor and  $\theta$  is the rotation degree between  $T$  and  $I$ . The voting process includes four main steps: 1) Comparing with all entries to check entry matching; 2) Calculating possible transformation for  $k$  and  $\theta$ ; 3) Checking process and 4) Voting process for the  $A(v_x, v_y, k, \theta)$ .

(1) Comparing with all entries to check entry matching

For each line segment pair ( $I_a$  and  $I_b$ ) in  $I$ , we find their middle points ( $P_a^I$  and  $P_b^I$ ) and compute their parameters to check whether “entry matching”. The correlative data of the middle points of  $I$  are summarized as follows:

$\delta_a^I, \delta_b^I$ : The interior angles formed by two line segments whose middle points are  $P_a^I$  and  $P_b^I$  in  $I$ , respectively.

$L_{ab}^I$ : The Euclidean distance between  $P_a^I$  and  $P_b^I$ , i.e., the length of  $\overline{P_a^I P_b^I}$ .

$\beta^I$ : The angle of  $\overline{P_a^I P_b^I}$ .

The “entry matching” is found when either of following conditions is:

$$|\delta_a^I - \delta_a| \leq \Delta_{th} \text{ and } |\delta_b^I - \delta_b| \leq \Delta_{th}, \tag{2}$$

or

$$|\delta_a^I - \delta_b| \leq \Delta_{th} \text{ and } |\delta_b^I - \delta_a| \leq \Delta_{th}, \tag{3}$$

where  $\Delta_{th}$  is a predefined tolerance error between two angles.

(2) Calculating the possible transformation for  $k$  and  $\theta$

While “entry matching”, we obtain the following possible transformation:

$$\theta = \beta^I - \beta, \tag{4}$$

$$k = L_{ab}^I / L_{ab}, \tag{5}$$

where  $\theta$  and  $k$  denote possible scaling factor and rotation degree, respectively.

(3) Checking process

The possible transformations for scaling and rotation factors are confirmed by geometric invariant matching with the check point (i.e.,  $P_c$  in the table). A new point,  $P_c^I$ , and its coordinates  $(c_x^I, c_y^I)$  are computed by the following geometry transformation:

$$\begin{pmatrix} c_x^I \\ c_y^I \end{pmatrix} = k \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \begin{pmatrix} x_m^I \\ y_m^I \end{pmatrix}, \tag{6}$$

where  $(x_m^I, y_m^I)$  is the middle point of  $\overline{P_a^I P_b^I}$ . If a line segment exists near to  $P_c^I$  whose coordinate is  $(c_x^I, c_y^I)$  in  $I$ , and the angle of the tangent line at  $P_c^I$ ,  $\tau^I$ , satisfies the following condition:

$$|\tau^I - \tau_c - \theta| \leq \Delta_{th}. \tag{7}$$

Again,  $\Delta_{th}$  is a predefined tolerance error between two angles. The probable scaling and rotation parameters are obtained and continue the following voting process.

(4) Voting process

For this parameter set, we compute the relative reference point,  $(v_x, v_y)$ , in  $I$ :

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = k \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} r_x \\ r_y \end{pmatrix} + \begin{pmatrix} x_m^I \\ y_m^I \end{pmatrix}. \tag{8}$$

A vote will be cast for the parameter set in the accumulator array. In the FGHT, the vote value is equal to one, i.e.,  $A(v_x, v_y, k) = A(v_x, v_y, k) + 1$ .

**3. The Awkward Problem of the Fast Generalized Hough Transform.** The major time costs for the FGHT and GFHT are from redundant mapping and spurious voting. Since either method encounters the same problems, we use the FGHT as an example to explain it:

- (1) Redundant mapping: in the FGHT, we pick up any two sub-blocks (approximated by line segments) in the template image  $T$  as a set to build the extended C-table. We use Figure 2(a) as an example. Let the template image  $T$  be divided into  $N$  sub-blocks:  $P_1, P_2, \dots, P_N$ . We pick a sub-block pair as a set to create information (entries and indexes) in the extended C-table; i.e., the  $(P_1, P_2)$  form a pair; similarly,  $(P_3, P_4)$  and  $(P_5, P_6)$  can be used to form two pairs, and we will totally have  $\binom{N}{2}$  pairs (i.e.,  $\binom{N}{2} = N(N - 1)/2 \approx N^2/2$  entries) in the table. In the shape detection, if the information created from a pair (for example,  $P_a^I$  and  $P_b^I$  in Figure 2(b)) of an input image is found to match exactly the information of one set in the C-table, the voting process should be continued until going through the whole table. That means we have  $\binom{N}{2} - 1$  redundant pairs for entry matching; i.e., redundant mapping.
- (2) Spurious voting: in an ideal shape detection process, we should find just one set in the C-table to match the pair,  $(P_a^I$  and  $P_b^I)$ , in the input image  $I$ . However, it is not the case in practical applications. When the entry number of the extended C-table is big, the chance of different sets having same (or very close) entry  $\delta_a$  and  $\delta_b$  is increased. Taking Figure 2(a) as an example, only the pair,  $(P_1, P_2)$ , is exactly matching with the pair,  $(P_a^I, P_b^I)$ . However, there are two redundant sets,  $(P_3, P_4)$  and  $(P_5, P_6)$ , with the same (or very close) entry  $\delta_a$  and  $\delta_b$  to the pair  $(P_a^I, P_b^I)$ . More time is required in the checking process to verify the indexes; meanwhile, the risk of spurious voting is increased.

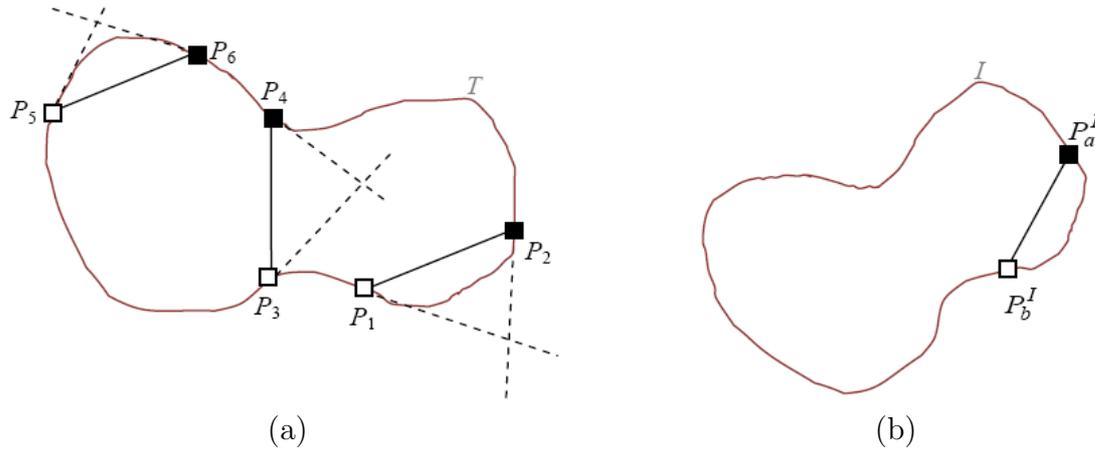


FIGURE 2. The illustration of the redundant mapping with similar entries: (a) a template image ( $T$ ) with similar entries:  $(P_1, P_2)$ ,  $(P_3, P_4)$  and  $(P_5, P_6)$ ; (b) an input image,  $I$ : only the pair  $(P_a^I, P_b^I)$  can match exactly with the pair  $(P_1, P_2)$ , and the other two pairs,  $(P_3, P_4)$  and  $(P_5, P_6)$ , are redundant mapping for spurious voting

Once the entry number of the extended C-table is reduced, the redundant sets and spurious voting will be improved efficiently. If the relationships between sets are found, we can reduce the spurious voting. In this paper, we extend the FGHT method to efficiently reduce redundant mapping by using randomized waypoint strategy.

**4. The Fast Randomized Generalized Hough Transform.** In this paper, we propose the fast randomized generalized Hough transform (FRGHT), which uses a randomized waypoint strategy to choose feature line segments randomly (randomized point-picking) and consecutively (consecutive entry matching). With this strategy, not only the required entry number of the table to avoid redundant mapping can be reduced dramatically, but also the relationship between sets can be found to reduce the spurious voting. The “randomized point-picking” is applied to modify the creation of the extended C-table step (Subsection 2.2), and the “consecutive entry matching” is used to modify the shape detection step (Subsection 2.3):

(1) Randomized point-picking

Step 1: **Initialization**

- 1) Start from the top of the extended C-table; i.e., let  $n = 1$ .
- 2) While  $P_a$  and  $P_b$  are too close to each other, we may not find a proper pole-polar triangle because of  $\delta_a \approx 0$  and  $\delta_b \approx 0$  (see Figure 1). So, we define a minimum distance criterion for  $P_a$  and  $P_b$ . By using four points on the top, bottom, left and right points of the template image, the smallest window  $W$  with four vertexes ( $v_1, v_2, v_3$  and  $v_4$ ) is easily obtained to cover the target object in  $T$ , and the vertexes:

$$W = \{v_i = (x_i, y_i) | i = 1, 2, 3 \text{ and } 4\}. \quad (9)$$

Instead of picking all pairs, we randomly pick a sub-block as a seed,  $P_a$ , be accompanied with another sub-block,  $P_b$ , which is also randomly picked and satisfied the following condition:

$$D(P_a, P_b) \geq d \cdot \max[D(P_a, v_i)], \quad (10)$$

where  $i = 1, 2, 3$  and  $4$ , and  $D(\bullet)$  denotes the Euclidean distance between two feature points; i.e.,  $D(P_a, v_i) = \|P_a - v_i\|$  and  $D(P_a, P_b) = \|P_a - P_b\|$ . Since the two closed feature points (the Euclidean distance is small) in input image ( $I$ ), it is easy to get spurious voting while entry matching. The distance criterion in Equation (10) is applied to sift the two feature points which their distance is small. From the empirical results, the value of  $d$  is suggested between  $0.4$  and  $0.6$ .

**Step 2: Set replacement**

We use the information obtained from  $P_a$  and  $P_b$  to build the entry and indexes of one set of the table. For the next set ( $n$  is accumulated by one), we replace  $P_a$  with  $P_b$  and find a new  $P_b$ . The new  $P_b$  is randomly picked from other unpicked sub-blocks and must satisfy Equation (10).

**Step 3: Iteration**

Repeat Step 2 till we cannot find a new  $P_b$  at that iteration.

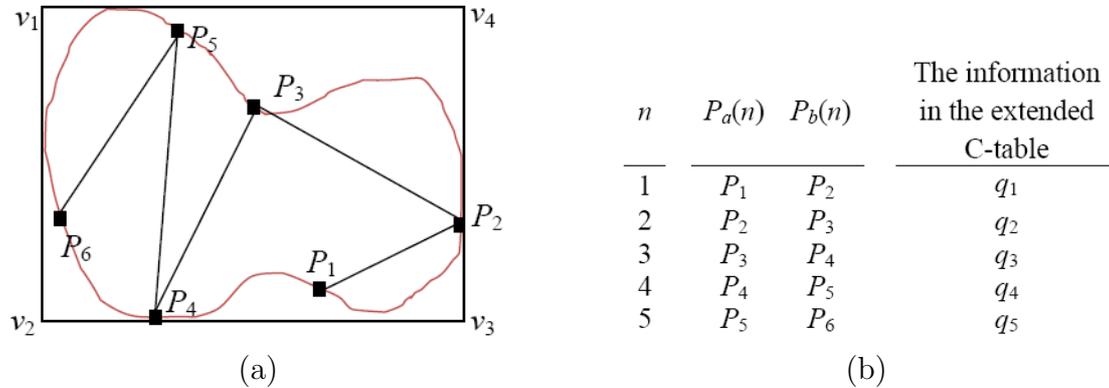


FIGURE 3. The illustration of the proposed FRGHT: (a) the randomized waypoint strategy for building the information within the extended C-table in (b), where  $P_a$  and  $P_b$  denote the feature points of the line segment pair in  $n$  set, and  $q_n$  denotes the geometric arrangement information (six indexes) that been defined in Equation (1) and illustrated in Figure 1

We use Figure 3 as an example. At Step 1, we find the smallest window  $W$  to cover the target object. We pick  $P_1$  as  $P_a$  and find another sub-block  $P_2$  as  $P_b$  to satisfy Equation (10). At Step 2, we use the information obtained from pair,  $(P_1, P_2)$ , to build the entry ( $\delta_a$  and  $\delta_b$ ) and indexes ( $L_{ab}, \beta, \tau_c, (c_x, c_y)$  and  $(r_x, r_y)$ ) of the set  $n = 1$ . Let  $n$  be accumulated by one, i.e.,  $n = 2$ . We replace  $P_a$  with  $P_2$  and find a new  $P_b$ , i.e.,  $P_3$ ; i.e., the pair,  $P_a(n = 2)$  and  $P_b(n = 2)$ , are  $P_2$  and  $P_3$ , respectively. We repeat Step 2 and find the entry and indexes of the set  $n = 2$ . We repeat Step 2 till we cannot find a new  $P_b$  at that iteration. In this case, assume we cannot find a new  $P_b$  for the iteration  $n = 5$ , the process is stopped. We totally find five sets (i.e.,  $Q = 5$ ) for the extended C-table.

(2) Consecutive entry matching

The “Shape detection” of the proposed method is similar to that of the FGHT, except we use an adaptive vote weights ( $w$ ), instead of a constant value.

**Step 1: Entry matching**

From the input image, we pick a feature pair ( $P_a^I$  and  $P_b^I$ ) and process the “entry-matching (Subsection 2.3)” to find a possible matched set ( $P_a(n)$  and  $P_b(n)$ ) in the table with Equations (2) and (3). The possible transformations,

$k$  and  $\theta$  are computed with Equations (4) and (5). Initially, the vote weights are set to be one; i.e.,  $w = 1$ .

**Step 2: Consecutive checking process**

We obtain the vector  $(c_x, c_y)$  from  $P_a(n+1)$  to  $P_b(n+1)$ . Then, the coordinate of the next consecutive point is computed,

$$\begin{pmatrix} N_x \\ N_y \end{pmatrix} = k \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \begin{pmatrix} Base_x \\ Base_y \end{pmatrix}, \quad (11)$$

where  $(Base_x, Base_y)$  is the coordinate of  $P_b^I$ . If a feature point with the coordinate  $(N_x, N_y)$  in  $I$  is found,  $P_N^I$ , and the angle of the tangent line at  $P_N^I$ ,  $\theta_N$ , satisfies the following condition:

$$|\theta_N - \theta_b - \theta| \leq \Delta_{th}, \quad (12)$$

where the index  $\theta_b$  is the angle of the tangent line at  $P_b(n+1)$  and  $\Delta_{th}$  is a predefined tolerance error of two angles. We let  $w$  be accumulated by one (i.e.,  $n = n + 1$ ) and repeat Step 2 till Equation (12) is un-satisfied.

**Step 3: Voting**

We compute the reference point  $(v_x, v_y)$  in  $I$  for this parameter set with Equation (8). The accumulated vote will be cast for the parameter set in the accumulator array, i.e.,  $A(v_x, v_y, k, \theta) = A(v_x, v_y, k, \theta) + w$ . Please be noted  $w$  is always equal to one in the FGHT. We let  $w$  be an adaptive vote weights.

**Step 4: Iteration of checking process**

We reset the vote weights be one; i.e.,  $w = 1$ . Let  $n = n + 1$  and go to Step 2 till all sets in the table are checked.

**Step 5: Iteration**

Go to Step 1 till all feature pairs are picked.

**5. Experiments and Discussions.** In this study, all experiments are performed on Windows Vista Operation System with Intel Core 2 Duo CPU (2.20 GHz) and executed (C++ programming) by using Borland C++ Builder 6.0. Synthetic and realistic images with  $640 \times 480$  pixels are applied to test the performance of the proposed method, including storage requirements and computation analysis. In the synthetic image experiment, a synthetic image with an artificial created clipper is used as the target object, see Figure 4(a). Then, we impose different levels of pepper noises on the image, for example 15% noise in Figure 4(b), for noise tolerance tests. In the realistic image experiment, a sketched

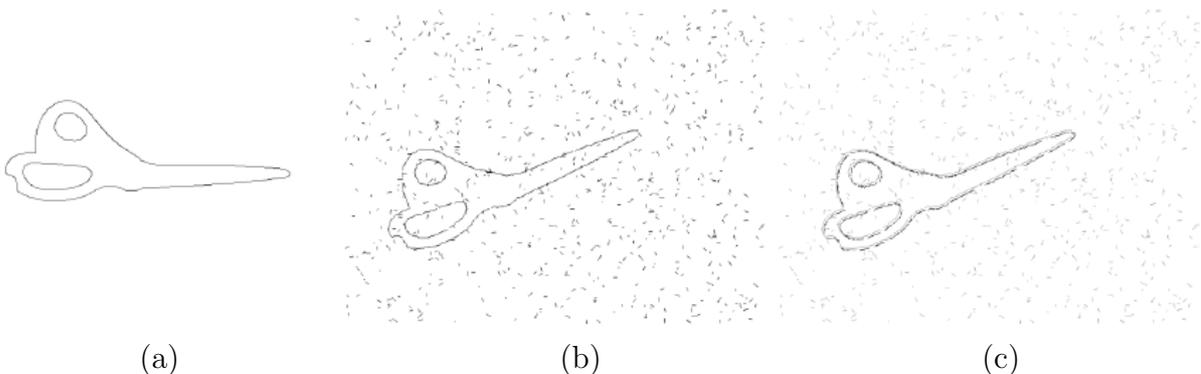


FIGURE 4. The synthetic image test with adding noises: (a) a template image,  $T$ ; (b) an input image,  $I$  ( $640 \times 480$ ), with adding 15% noise; (c) the detection result of (b) by using the proposed FRGHT

kite template is used, see Figure 5(b), to create the extended C-table and detect a realistic kite target image, see Figure 5(a).

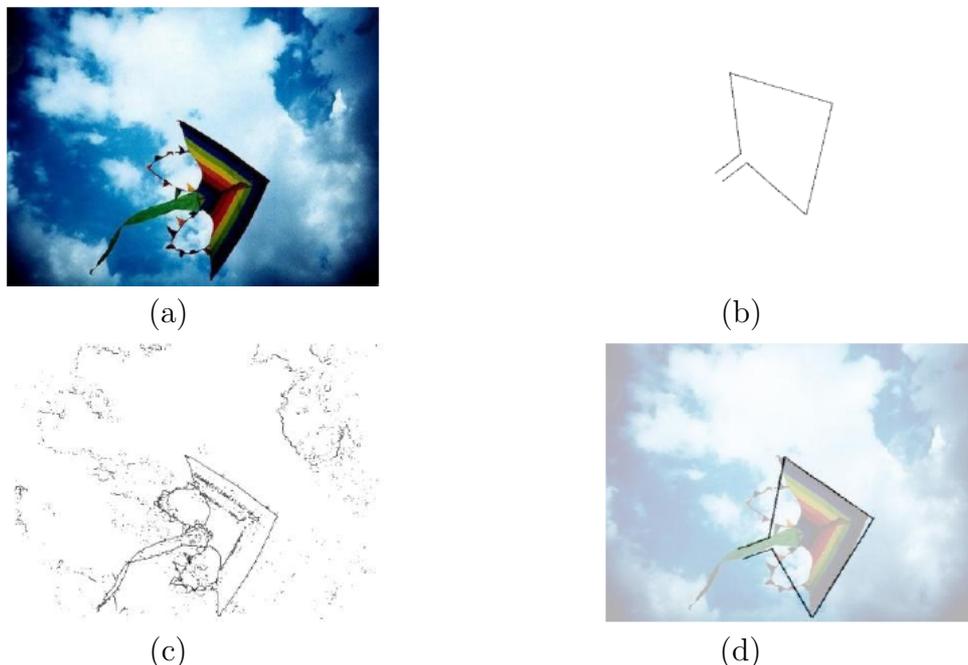


FIGURE 5. The realistic image test: (a) a kite image ( $640 \times 480$ ); (b) a template image,  $T$ ; (c) an input image,  $I$ : the edge of (a); (d) the detection result by using the FRGHT. The  $C_{FRGHT}$  in this experiment, is about  $3.18 (\times 10^5)$ , and the execution time is about 0.25 seconds for detection.

TABLE 1. The parameters settings of the GHT's. The “ $\times$ ” denotes the method does not need the parameter.

Parameters	Methods		
	FGHT	GFHT	FRGHT
$B$	8	8	8
$\Delta_{th}$ (degree)	10	10	10
$L_{th}$ (pixels)	6	6	$\times$
$d$	$\times$	$\times$	0.4

Table 1 shows the parameter settings of the three methods in our experiments.  $B$  denotes the block size, and it is set as 8 in all experiments based on previous work [18].  $\Delta_{th}$  is a tolerance error value described in Equations (2), (3), (7) and (12).  $L_{th}$  is a predefined threshold for sifting the short line segment.  $d$  in Equation (10) is used to sift two feature points with a short Euclidean distance between them. From the empirical results, the suggested value of  $d$  is between 0.4 and 0.6. In this study, all experiments are set as  $d = 0.4$ .

**5.1. Storage requirements.** The storage requirements include two parts: 1) the extended C-table and 2) the accumulator array:

- (1) Memory size of the extended C-table:  $N$  feature points in  $T$  (template image) are assumed. In the worst-case scenario,  $N - 1$  pairs to be picked and built into the

extended C-table during the “randomized waypoint” strategy:

$$Q_{\text{FRGHT}} = N - 1. \quad (13)$$

For the same  $T$ , the  $Q$ 's of the FGHT and GFHT,  $Q_{\text{FGHT}}$  and  $Q_{\text{GFHT}}$ , in the extended C-table can be approximated as:

$$Q_{\text{FGHT}} = Q_{\text{GFHT}} = \binom{N}{2} \approx \frac{N^2}{2}. \quad (14)$$

Comparing the  $Q$  of the FRGHT to  $Q_{\text{FGHT}}$  and  $Q_{\text{GFHT}}$ , as seen in Equations (13) and (14), we find that the ratio of  $Q_{\text{FRGHT}}$  to  $Q_{\text{FGHT}}$  and  $Q_{\text{GFHT}}$  can be estimated as:

$$\frac{Q_{\text{FRGHT}}}{Q_{\text{FGHT}}} \approx \frac{Q_{\text{FRGHT}}}{Q_{\text{GFHT}}} \approx \frac{2}{N}, \quad (15)$$

where  $N$  is the number of feature points in  $T$ . In practical applications,  $N$  is significantly greater than 2 ( $N \gg 2$ ). It can be seen that the FRGHT requires the least entry number than those of the FGHT and GFHT.

For example, a  $640 \times 480$  template image, most 4,800 feature points (i.e.,  $N = 4,800$ ) be used to construct  $q_n$  (please refer to Equation (1)) while  $B = 8$ . Each  $q_n$  contains nine 8-byte variables. The memory size required of the proposed FRGHT in the extended C-table is approximated as 0.33 megabyte (i.e.,  $(4800 - 1) \times 9$  variables  $\times$  8 bytes  $\approx$  0.33 megabyte). On the other hand, the memory size required of the FGHT and GFHT in the extended C-table is approximated as 790 megabyte (i.e.,  $\binom{N}{2} \times 9$  variables  $\times$  8 bytes  $\approx$  790 megabyte). We can find that the memory size of the proposed FRGHT is far less than that of the others.

- (2) Memory size of the accumulator array: The accumulator array of the FRGHT is four-dimensional array. The memory size of the accumulator array can be approximated as:

$$\frac{S_x}{B} \times \frac{S_y}{B} \times \frac{S_k}{k_r} \times \frac{S_\theta}{\theta_r}, \quad (16)$$

where  $S_x \times S_y$  is the template image ( $T$ ) size.  $k_r$  and  $\theta_r$  are the resolutions of scaling factor ( $S_k$ ) and rotation factor ( $S_\theta$ ), respectively. For example, if the size of  $T$  is  $640 \times 480$  pixels,  $S_x = 640$  and  $S_y = 480$ . Assume the transformation of  $T$  and  $I$  for scaling and rotation factors are predefined from 0.50 to 2.50 and from  $0^\circ$  to  $179^\circ$ , respectively. The resolutions of them are set as  $k_r = 0.01$  and  $\theta_r = 1^\circ$ , respectively. We can obtain  $S_k = 2.01$  ( $= 2.50 - 0.50 + 0.01$ ) and  $S_\theta = 180^\circ$  ( $= 179^\circ - 0^\circ + 1^\circ$ ). From Equation (16), we can obtain about  $S = 1.73 \times 10^8$  cells in the accumulator array, while  $B = 8$ . Since each cell is defined as a 4-byte integer, the memory sizes of all three methods are approximated as 660 megabyte.

**5.2. Computation analysis.** In the “Consecutive entry matching”, the worst of the computation of the FRGHT for entry matching can be estimated as:

$$C_{\text{FRGHT}} = \binom{M}{2} \cdot Q_{\text{FRGHT}} = \frac{M(M-1)}{2} \cdot (N-1) \approx \frac{M^2 N}{2}, \quad (17)$$

where  $M$  and  $N$  denote the numbers of feature points in  $I$  (input image) and  $T$  (template image), respectively. The computational times of the FGHT for entry matching,  $C_{\text{FGHT}}$ ,

is:

$$C_{\text{FGHT}} = \binom{M}{2} \cdot Q_{\text{FGHT}} \approx \frac{(MN)^2}{4} \tag{18}$$

Comparing the computational time of the FRGHT to  $C_{\text{FGHT}}$ , as seen in Equations (17) and (18), we obtain the ratio:

$$\frac{C_{\text{FRGHT}}}{C_{\text{FGHT}}} \approx \frac{2}{N}, \tag{19}$$

where  $N$  is the number of feature points in  $T$ . In practical applications,  $N$  is significantly greater than 2 ( $N \gg 2$ ). It can be seen that the FRGHT requires less computational times than the FGHT.

Similarly, for the GFHT, since the number of feature points is calculated in a fuzzy region for vote collection, we use an integer  $e \geq 1$  to estimate the feature point number within the fuzzy region. The computation time of the GFHT for entry matching,  $C_{\text{GFHT}}$ , is:

$$C_{\text{GFHT}} = \binom{M}{2} \cdot Q_{\text{GFHT}} \cdot e \approx \frac{(MN)^2 \cdot e}{4}. \tag{20}$$

Comparing the computational time of the FRGHT to  $C_{\text{GFHT}}$ , as seen in Equations (17) and (20), we obtain the ratio:

$$\frac{C_{\text{FRGHT}}}{C_{\text{GFHT}}} \approx \frac{2}{N \cdot e}. \tag{21}$$

Again,  $N \gg 2$  and  $e \geq 1$ . It still can be seen that the FRGHT requires less computation (for entry matching) than the GFHT.

**5.3. Experiments of synthetic and realistic images.** Figure 4 is used as an example to show the comparison. Since the FRGHT, FGHT and GFHT are line-based methods, we add “line segment noise” to test them. All parameters are shown in Table 1. Figure 4(a) shows the template image ( $T$ ), and Figure 4(b) shows the input image ( $I$ ) with adding 15% “line segment noise”. Figure 4(c) shows the detection result using the proposed FRGHT. Moreover, we add noise from 3% to 24% to test all methods, and the experimental result is shown in Figure 6. Even in adding heavy noise (i.e., 24%), the FRGHT still uses the least iterations for entry matching. The main reason for that is the FRGHT uses the least entry number, i.e.,  $Q_{\text{FRGHT}} = 132$  in Table 2.

TABLE 2. The experimental result of  $Q$ 's (Entry number) of the three methods in the synthetic and realistic images

Experiments	Methods		
	FGHT	GFHT	FRGHT
Figure 4(a)	22,273	22,273	132
Figure 5(b)	5,207	5,207	65

Similarly, a realistic image is used in Figure 5(a): a sketched kite image. Figures 5(b) and 5(c) show the template image ( $T$ ) and the edge of the kite image (be treated as the input image,  $I$ ), respectively. The detection result is shown in Figure 5(d), and the  $C_{\text{FRGHT}}$  in this experiment only needs about  $3.18 \times 10^5$  iterations for entry matching (please be

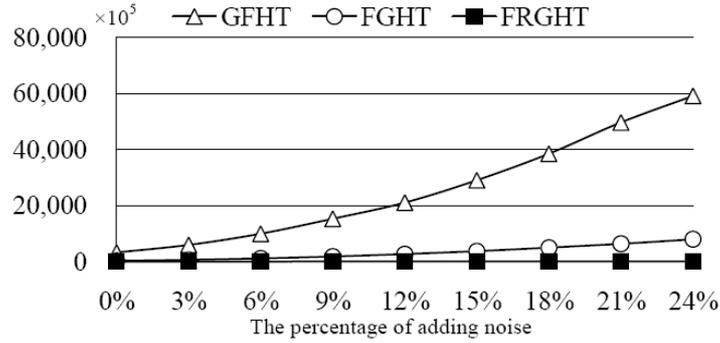


FIGURE 6. The computational times of the FGHT, GFHT and FRGHT in the synthetic image test (See Figure 4. The  $C_{\text{FRGHT}}$ 's in this experiment are about 1.13, 2.37, 4.39, 7.07, 9.66, 12.8, 16.68, 19.41 and 25.17 ( $\times 10^5$ ), respectively. The computation of the FRGHT is far less than that of the FGHT and GFHT.

noted that the execution time is approximated as 0.25 seconds). The proposed FRGHT not only can detect the target kite, but also can efficiently sift redundant mapping for entry matching to decrease the execution time required. The main reason for that is the FRGHT uses the least entry number, i.e.,  $Q_{\text{FRGHT}} = 65$  in Table 2.

**5.4. Discussions of the spurious voting.** Two indexes are used for estimating the efficiency of reducing spurious votes: 1) voting efficiency,  $V_{\text{eff}}$  and 2) voting entry reliability,  $R_{\text{ve}}$ .

(1) Voting efficiency  $V_{\text{eff}}$

The voting efficiency,  $V_{\text{eff}}$ , is defined as the ratio between the number of correct votes and the number of total votes [17]. The higher  $V_{\text{eff}}$  means higher voting efficiency. We redo the synthetic image test in Figure 4 and show the result of voting efficiency in Table 3; obviously, the proposed FRGHT has the highest voting efficiency than the FGHT and GFHT.

TABLE 3. The experimental result of the synthetic image test (see Figure 4: the voting efficiency,  $V_{\text{eff}}$  (%), of the three methods

Adding noise (%)	Methods		
	FGHT	GFHT	FRGHT
0	0.119	0.134	3.874
3	0.065	0.077	1.410
6	0.041	0.044	0.725
9	0.026	0.031	0.638
12	0.017	0.021	0.220
15	0.013	0.014	0.134
18	0.010	0.011	0.095
21	0.007	0.008	0.131
24	0.006	0.007	0.068

(2) Voting entry reliability  $R_{\text{ve}}$

While computing the total computational times in Equation (17), the possible entry matching times may include picking noised object (or non-target). That will increase the computation times and decrease the voting efficiency simultaneously. We use the

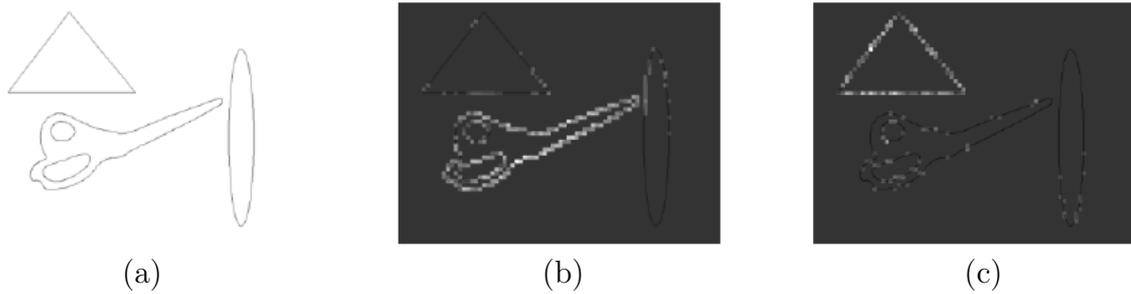


FIGURE 7. The result of voting entry reliability ( $R_{ve}$ ): (a) the clipper is the target (template) in the input  $I$ ; (b) the  $R_{ve}$  images of the proposed FRGHT. The higher illumination denotes more entry matching times are obtained, and we can see the bright pixels are mostly on the target clipper; (c) if we treat the triangle as our target (template), it still can be seen that the bright pixels are mostly on the target triangle.

following ratio to denote the reliability of voting entry:

$$R_{ve} = \frac{\text{the entry matching times of target pairs}}{\text{the entry matching times of noise pairs}}. \quad (22)$$

The higher value of  $R_{ve}$  means the higher vote weights on the target object in the voting process. In Figure 7(a), there are three objects (clipper, triangle and ellipse) in the image. Assume our target is the clipper. Figure 7(b) shows the experimental results of entry matching times of the possible pairs of the proposed FRGHT. The higher illumination denotes more entry matching times are obtained. While the  $R_{ve}$  value of the FRGHT is about 13.74, the  $R_{ve}$ 's of the FGHT and GFHT are respectively about 1.51 and 1.66, which are much less than that of the FRGHT. While the triangle is used in Figure 7(a) as the target (template) and repeat the experiment, most of bright pixels are on the target triangle, in Figure 7(c).

**6. Conclusions.** In this paper, we propose the fast randomized generalized Hough transform (FRGHT), which uses a randomized waypoint strategy to choose feature line segments (its middle points) randomly and consecutively. With this strategy, a feature pair only provides one entry in the extended C-table. Moreover, consecutive entry matching can reduce the effect of spurious voting. The experimental result shows that the FRGHT needs less computation and storage (the entry number) requirements than the FGHT and GFHT.

Although the FRGHT obtain the higher voting efficiency than the previous modified GHT by using consecutive entry matching, it still requires a four-dimensional array for voting process. In the future work, we will reduce the array dimension to improve the voting process.

## REFERENCES

- [1] H. Benitez-Perez and A. Benitez-Perez, The use of ARMAX strategy and self organizing maps for feature extraction and classification for fault diagnosis, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4787-4796, 2009.
- [2] A. Ohno and H. Murao, A similarity measuring method for images based on the feature extraction algorithm using reference vectors, *International Journal of Innovative Computing, Information and Control*, vol.5, no.3, pp.763-772, 2009.
- [3] S. Miyata, A. Yanou, H. Nakamura and S. Takehara, Road sign feature extraction and recognition using dynamic image processing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4105-4114, 2009.

- [4] P. P. Roy, U. Pal and J. Lladós, Seal object detection in document images using GHT of local component shapes, *Proc. of the ACM Symposium on Applied Computing*, Sierre, Switzerland, pp.23-27, 2010.
- [5] C. H. Chung, S. C. Cheng and C. C. Chang, Adaptive image segmentation for region-based object retrieval using generalized Hough transform, *Pattern Recognition*, vol.43, no.10, pp.3219-3232, 2010.
- [6] D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition*, vol.13, no.2, pp.111-122, 1981.
- [7] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 2nd Edition, Newnes, New York, USA, 2002.
- [8] F. F. Ping and S. L. Wing, Randomized generalized Hough transform for 2-D grayscale object detection, *Proc. of International Conf. of Pattern Recognition*, Vienna, Austria, pp.511-515, 1996.
- [9] D. M. Tsai, An improved generalized Hough transform for the recognition of overlapping objects, *Image and Vision Computing*, vol.15, no.12, pp.877-888, 1997.
- [10] P. Tipwai and S. Madarasmı, A modified generalized Hough transform for image search, *IEICE Transactions on Information and Systems*, vol.E90-D, no.1, pp.165-172, 2007.
- [11] E. Montiel, A. S. Aguado and M. Nixon, Improving the Hough transform gathering process for affine transformations, *Pattern Recognition Letters*, vol.22, no.9, pp.959-969, 2001.
- [12] A. S. Aguado, E. Montiel and M. Nixon, Invariant characterisation of the Hough transform for pose estimation of arbitrary shapes, *Pattern Recognition*, vol.35, no.5, pp.1083-1097, 2002.
- [13] T. E. Dufresne and A. P. Dhawan, Chord-tangent transformation for object recognition, *Pattern Recognition*, vol.28, no.9, pp.1321-1332, 1995.
- [14] P. K. Ser and W. C. Siu, A new generalized Hough transform for the detection of irregular objects, *Journal of Visual Communication and Image Representation*, vol.6, no.3, pp.256-264, 1995.
- [15] A. Kimura and T. Watanabe, Fast generalized Hough transform: Rotation, scale and translation invariant detection of arbitrary shapes, *IEICE Transactions*, vol.J81-D-II, no.4, pp.726-734, 1998 (in Japanese).
- [16] C. P. Chau and W. C. Siu, Generalized dual-point Hough transform for object recognition, *Proc. of IEEE International Conf. on Image Processing*, Kobe, Japan, pp.560-564, 1999.
- [17] C. P. Chau and W. C. Siu, Adaptive dual-point Hough transform for object recognition, *Computer Vision and Image Understanding*, vol.96, no.1, pp.1-16, 2004.
- [18] N. Suetake and E. Uchino, Generalized fuzzy Hough transform for detecting arbitrary shapes in a vague and noisy image, *Soft Computing*, vol.10, no.12, pp.1161-1168, 2006.