machines

MDPI

Article

Robust Object Positioning for Visual Robotics in Automatic Assembly Line under Data-Scarce Environments

Yigong Zhang [1,†], Huadong Song [2,†], Xiaoting Guo [2] and Chaoqing Tang [3,*]

[1] China Quality Certification Centre, Beijing 100070, China
[2] SINOMACH Sensing Technology Co., Ltd., Shenyang 110043, China
[3] China Belt and Road Joint Laboratory on Measurement and Control Technology, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
* Correspondence: billtang@hust.edu.cn
† These authors contributed equally to this work.

Abstract: Object positioning is a basic need for visual robotics in automatic assembly lines. An assembly line requires fast transfer to new object positioning tasks with few or no training data for deep learning algorithms, and the captured visual images usually suffer from partial missing and cropping and environmental lighting interference. These features call for efficient and robust arbitrary shape positioning algorithms under data-scarce and shape distortion cases. To this end, this paper proposes the Random Verify Generalised Hough Transform (RV-GHT). The RV-GHT builds a much more concise shape dictionary than traditional GHT methods with just a single training image. The location, orientation, and scaling of multiple target objects are given simultaneously during positioning. Experiments were carried out on a dataset in an automatic assembly line with real shape distortions, and the performance was improved greatly compared to the state-of-the art methods. Although the RV-GHT was initially designed for vision robotics in an automatic assembly line, it works for other object positioning mechatronics systems, which can be modelled as shape distortion on a standard reference object.

Keywords: machine vision; Hough transform; object positioning; shape distortion; automatic assembly line

Citation: Zhang, Y.; Song, H.; Guo, X.; Tang C. Robust Object Positioning for Visual Robotics in Automatic Assembly Line under Data-Scarce Environments. *Machines* **2022**, *10*, 1079. https://doi.org/10.3390/machines10111079

Academic Editors: Antonios Gasteratos and Praneel Chand

Received: 18 October 2022
Accepted: 10 November 2022
Published: 16 November 2022

1. Introduction

Object positioning is crucial to many visual robotic systems, such as sorting robot arms, transfer robot arms, assembly robots, etc. These applications require not only the information of an object's location, but also its orientation, scale, and the number of instances for operation. Although deep learning has shown impressive accuracy in shape recognition and localisation, in dealing with precise orientation and scale information, it suffers from poor performance without a comprehensive training dataset [1]. On the other hand, many applications are frequently transferred to a totally new operation target. Deep learning methods require days or even months of training data collection to guarantee good performance in comprehensive situations, which is a serious interruption to normal production and cannot meet the online efficiency requirement. An alternative deterministic method is the Generalised Hough Transform (GHT), a shape detection method [2–5], which uses the contour of a template shape to build a codebook, known as the R-Table. In the detection step, votes are cast for the presence of the shape in the Hough space by looking up the R-Table. Since the R-Table is a one-to-many mapping between indices and votes, the shape can be identified through majority voting. Local information stored to generate the R-Table can be expanded into interest points [6], image patches [7], or regions [8]. The GHT extends the Hough transform, which can only be used to detect shapes that have analytical models such as line segments, circles, ellipses, etc. [9–11], to arbitrary shapes.

The conventional GHT is known [3,12] to suffer from two fatal limitations: (a) There is a high computational cost when the input shape is rotated or scaled. If the rotation or scaling of the target shape is different from the template, common practice is to transform the R-Table template to all its potential rotations and scaled sizes, with a certain step. Not only does this approach come at a high cost, but the detection resolution is limited to the step size. Some authors find their own rotation/scale invariant features [13–17], but these methods are either too sensitive to noise or limited to shape constraints. Another category of approaches uses supervised training and matching for task-adapted R-Tables called Hough forests [18]. The primary Hough forest cannot give the rotation information; some extended works [19,20] augmented the training dataset with rotated copies of the template or trained a classifier common to "all" orientations. (b) There is poor performance when the target shape is similar, but not identical to the input template; this is a common case in automatic assembly and packaging lines. For example, in the demo images shown in Figure 1, the contour of the target shapes is polluted when there is light interference from the environment or the components themselves have a high reflective rate or shape fluctuation. For precise manufacturing applications, other visual modalities such as infrared and X-ray are also common. These visual images are much more fuzzy than visible light images, which easily leads to shape distortion. The R-Table template in Hough methods is made up of standard shapes, and any distortion in the target will result in votes, which contaminate the ground truth in conventional methods. This phenomenon is exacerbated with the increase in the number of the one-to-many mappings in the R-Table.

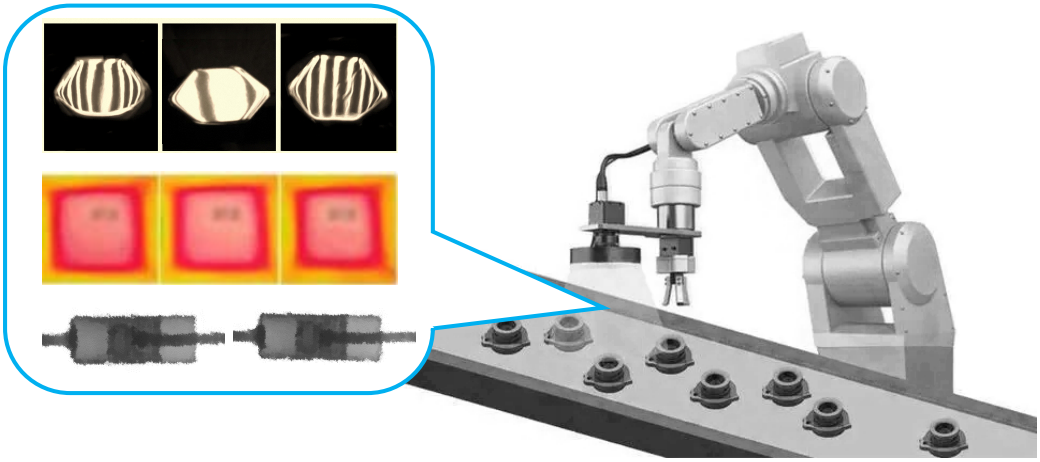


Figure 1. Example images of different modalities of visual robotic systems in automatic assembly lines. From top to bottom are visible light, infrared, and X-ray.

In order to address both issues described above, we introduce a novel computer vision algorithm called the Random Verify GHT (RV-GHT). It has low complexity, yet achieves state-of-the-art accuracy. Our main contributions are as follows:

- This paper provides one efficient and robust algorithm for arbitrary shape detection under data-scarce cases for visual robotics in automatic assembly lines. The proposed method can be trained with even just a single image to build the R-Table, while deep learning methods require a large dataset to cover different rotations/scalings, and such a dataset usually takes a long time to collect for new detection targets.
- To handle the bottleneck of poor performance under shape distortion for the traditional GHT methods, a series of anti-distortion procedures was designed; the major one is the random verify process. The difficulty of the one-to-many mapping problem for traditional GHT is greatly reduced, and this leads to a great accuracy improvement.
- To avoid the time-expensive iterative multi-dimensional voting process for traditional GHT methods, this paper designs one single-shot voting scheme to obtain the scaling factor, shape centre, and target rotation from 0 to 2π simultaneously.

- To effectively lower the search overhead in detection, a new R-Table is designed, which has the most concise form compared to the existing GHT methods. Besides, a new minimum point set selection algorithm that reduces the number of indices both in R-Table construction and in the detection process is designed.
- The validation results illustrate a much better accuracy than other algorithms under shape distortion cases. The overall efficiency is similar to the state-of-the art method.

2. Related Work

Efficient GHT algorithms: The concept of the GHT was introduced in the 1980s [2], but has been subject to a number of further refinements in the presence of rotation and scaling. The R-table typically records the position of all (or most) boundary points of a template shape relating to a fixed reference point [12].

Several authors have put effort into improving the overall efficiency in the presence of scaling and rotation. One category is putting the scaling and rotation into the R-Table [21], but this method costs too much storage space for the R-Table compared to methods that use invariant features. Commonly considered invariant features are invariant interest points [22,23], local curvature [15] and pole–polar triangles [24–28]. Methods based on interest points find invariant interest points on both the target and the template image [29] using detectors such as SIFT, SURF, HoG, etc. The results of such algorithms are sensitive to background noise. Additionally, any outlying interest points or a general insufficiency of interest points decreases the overall performance. Another category of the GHT is based on curvature. It performs poorly for shapes that consist of line segments rather than curves.

A third category of the GHT is based on pole–polar triangles. These algorithms are more robust to noise and have been more popular in practical applications. Pole–polar-triangle-based methods can be categorized into pixel-based and non-pixel-based methods. Pixel-based methods calculate an index for every edge pixel. The advantage is that they are rich in index information, but are time-consuming. Ser et al. [24] proposed a dual-point GHT (DP-GHT), which uses pixel pairs with the same slope to form the triangle. This method does not work for shapes that lack pixels with the same slope. Chau and Siu [25] proposed an improved version of the DP-GHT called the GDP-GHT (also called the RG-GHT), which uses index pairs with a constant angle difference. The DP-GHT is a special case of the GDP-GHT. They further extended the algorithm to accommodate shapes with multiple constant angle differences [28].

By contrast, non-pixel-based methods approximate the target shape using blocks [30–32], lines or circles [26], or polygons [16]. This approach involves a fast index calculation at the cost of representation errors of the template shape. If the target is identical to the approximation, the algorithm cannot distinguish between the two. This approach may also lack sufficient indices to vote for the true configuration. Jeng and Tsai [26] used half lines and circles in the R-Table. The detection process involves separate scaling-invariant and orientation-invariant cell incrementing strategies. This works well only when the rotational angle and scaling are in a fairly restricted range. Yang et al. [16] proposed a polygon-invariant GHT (PI-GHT), which employs the local pole–polar triangle features based on polygonal approximation, using dominant points from the edges as the index. Due to limited invariant feature quantity, the results are too sensitive to the quality of the polygonal approximation, which means the algorithm cannot identify shapes that are the same as their approximation. Ulrich et al. [27] introduced a hierarchical strategy that splits the image into tiles; each tile has a separate R-table. However, in the absence of invariant features, this is just an improved, but inefficient brute force search. Kimura et al. [30] proposed a fast GHT (FGHT), which splits the image into small sub-blocks and approximates the slope of each block. In order to reduce the influence of noise on estimating block slope, Reference [31] proposed a generalized fuzzy GHT (GFGHT). The overall process is much the same as the FGHT, with the exception of a fuzzy voting process, which gives a block pair's centre point more weighting in the voting if it is closer to the centre of the shape. The serious limitation of the latter two methods is the increased complexity:

if N blocks satisfy the distance threshold, the index pair is C_N^2. To reduce the complexity of FGHT methods, Chiu et al. [32] proposed a fast random GHT (FRGHT), which uses a random pair selection procedure in index searching and weighting. This method greatly reduces the index number, at the cost of reduced robustness.

In summary, the pixel-based subcategory of using the pole–polar triangle as an invariant feature has the richest index and, hence, is the most robust to noise. The state-of-the-art GDP-GHT methods and PI-GHT methods are not limited to shape constraints; however, the high number of one-to-many mappings from the index of the R-Table to a configuration increases the time complexity and the presence of incorrect votes. In the case of both pixel-based and block-based methods, the number of index pairs can be unacceptably high. The FRGHT reduces the number of indices required by the GFGHT at the cost of a sacrifice in robustness.

GHT algorithms that are robust to shape distortion : Few approaches deal with the challenge of shape distortion directly. The most popular methods rely on the fuzzy voting concept, by weighting each vote instead of simply accumulating every single one. For example, Xu et al. [33] weighted the voting in the GHT by how important the corresponding ship part was. They gave the fore and the poop deck a higher weight because they are most different from the rest of the contour. The Generalised Fuzzy GHT [31] (GFGHT) gives more weight to index pairs whose middle pixel is closer to the shape centre. The reference point concept in the GFGHT is a good effort to remove incorrect votes. However, these methods still only have a limited ability to deal with shape distortion.

3. The Proposed Random Verify GHT

The proposed RV-GHT method has a similar outline to other GHT methods, as shown in Figure 2: it extracts local index information, builds an R-Table offline, and then applies detection online. There are mature methods for the contour extraction and slope calculation processes. The idea proposed in this paper designs a new R-Table and detection process, which efficiently deal with rotation and scaling under shape distortion. While building the R-Table, we used pairs of contour pixels with a constant slope difference in order to generate rotation- and scaling-invariant features as indices. In the detection process, each index is simultaneously used to vote for location, rotation, and scaling and uses a Gaussian kernel function for voting. The rotation is only described as the angle of rotation, but no actual centre of rotation was identified. Furthermore, during the detection step, a random verify scheme helps to eliminate ambiguities in the one-to-many mappings, thereby reducing the influence of distortion or partial missing.

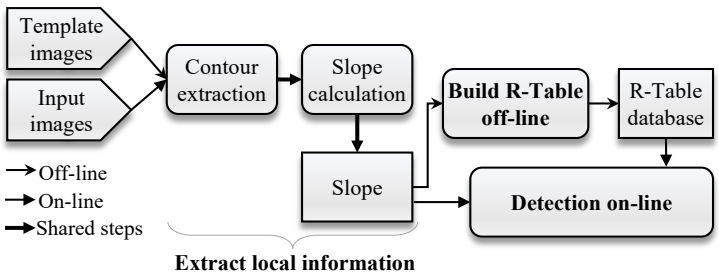
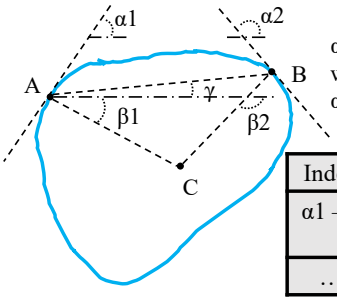


Figure 2. The overall diagram of the RV-GHT.

3.1. A New and Concise R-Table in RV-GHT

The R-Table stores the projection between local information and the standard template shape. Based on the contour slope, a contour pixel pair AB with constant-convexity-augmented slope difference θ is chosen as two vertices of the pole–polar triangle, as is shown in Figure 3. θ can be any value set by the user, but needs to be the same for building the R-Table process and the detection process. Therefore, Points A and B and the intersection point of their respective slope lines form a pole–polar triangle.

Based on Points A and B, the index in the R-Table is the difference between the slope of A (denoted as $\alpha 1$) and the slope of Line AB (γ). This index is invariant to rotation and scaling because the rotation amount for $\alpha 1$ is always the same as that of γ.



$\alpha 1 - \alpha 2 = \theta$,
where θ is a constant;
$\alpha 1, \alpha 1, \beta 1, \beta 2, \gamma \subset [0, \pi]$

R-Table

Index	Item			
$\alpha 1 - \gamma$	$\alpha 1$	Distance of AC	$\beta 1$	$\beta 2$
…	…	…	…	…

Figure 3. R-Table in the proposed RV-GHT.

The items of the resulting R-Table are also shown in Figure 3. To give an accurate description of shape location, a reference point (centre point) C is allocated. This point can be set at any location by the user. Some special settings can help with future processing, e.g., the centroid or the start and end points of a simple curve. Given Point C, it is possible to calculate the slope of Lines CA (denoted as $\beta 1$ in Figure 3) and CB ($\beta 2$). All items are angles except the third column for the length of Segment AC, because all related angle information can be derived from the stored angles.

Each row of the R-Table is made up by searching iteratively for constant slope difference pairs such as AB for $0 \leq \alpha 1 < \pi$. As is shown in Algorithm 1, the algorithm aims to compress the length of the R-Table while preserving as much information as possible during the searching process. Firstly, the length of AB must be greater than a threshold. This threshold can be as small as 10 pixels just to avoid the estimation error for γ. Secondly, it is popular for a shape contour to show a group of consecutive pixels that has the same slope. In conventional versions of the GHT, all these pixels will generate index pairs, which is highly redundant. Instead, we propose the following minimum point set selection scheme in order to cut down redundancy. Our algorithm takes as the input a point set that has the same local information. A separate input is a distance threshold that defines the minimum distance between two points. The output is a point set whose distance between any two points in the set is greater than the defined threshold. The minimum point set can be obtained after $N - 1$ to $\frac{(N-1)N}{2}$ times of distance calculations and comparisons, where N is the number of contour pixels.

Algorithm 1: Find the minimum point set, the pairwise distance of which is above a threshold.

Input: Point set that has same local information, $\mathbf{P}$.
A distance threshold, Dth
Output: Minimum point set, $\mathbf{P}_m$

```
Choose any point in the set as the seed, P_1;
Initial minimum point set, P_m={P_1};
for any remaining point in P, P_x do
    if all distances of P_m P_x > Dth then
        Add P_x to P_m
    else
        Remove P_x from P.
    end
end
```

For some shapes, the contour pixel pairs that satisfy $\alpha 1 - \alpha 2 = \theta$ are not unique. For example, there are 4 contour pixels whose slope is $\alpha 1$ (Point A in Figure 3) and 3 for $\alpha 2$ (Point B in Figure 3). In such cases, every A is matched with every B to construct 12 pairs. This results in an over-complete R-Table. This R-Table ensures the scheme in the *Find AB pairs* section can always find the corresponding record under multiple candidates for A and B.

3.2. Detection Process of RV-GHT

The detection process is shown in Figure 4. The principal contours of the whole target image are first obtained by running, for example, a Canny edge detector, followed by calculating the slope for the resulting contours. Given these pre-processing steps, the RV-GHT finds an index pair first. With an R-Table for a template shape, the primary centre (point C in Figure 3), rotation, and scaling can be found. The random verify gives a weighting factor for each configuration. Lastly, the multi-instance classification process gives the final centre, rotation, and scaling for each instance. The rest of this section introduces each stage in detail.

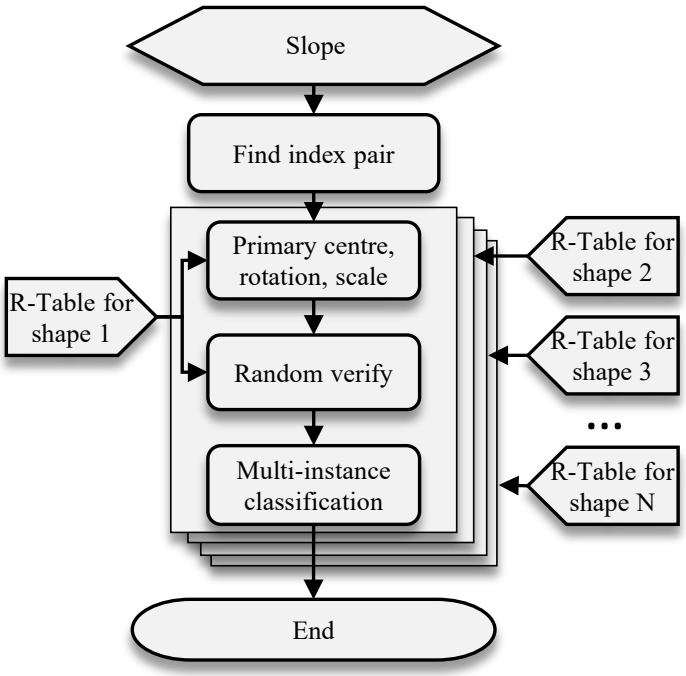


Figure 4. The detection process of the RV-GHT.

3.2.1. Find AB Pairs

The first step is to find edge pairs such as A and B in Figure 3. To this end, the slope for all contour pixels (denoted **G**) is computed first. Algorithm 2 details the searching process. The example given in the last paragraph of Section 3.1 generates 12 index pairs. As every index pair will generate a vote later, if there are too many points with the same slope, brute force searching of this large set can greatly increase the complexity of the detection process. This motivates why Step 5 of Algorithm 2 limits the number of output pairs. Within the same example discussed above, Step 5 of the algorithm only generates four pairs: A_1B_1, A_2B_2, A_3B_3, and A_4B_x, where x is a random value within $\{1, 2, 3\}$. This eliminates about 60% of the redundant votes for this example whilst guaranteeing that every A and B participated in the vote.

Algorithm 2: Find the AB pair from the edge image.

Input: The slope for all contour pixels, **G**; Constant angle, θ
Output: ABPair
ABPair = {empty};
for $i = 0$ *to* π **do**
  Find all points of $\mathbf{G} = i$, and compress with Algorithm 1 as $A_1, A_2, \ldots, A_m$;
  Find all points of $\mathbf{G} = i - \theta + \pi H(\theta - i)$, and compress with Algorithm 1 as $B_1, B_2, B_3, \ldots, B_n$, where H(.) is the Heaviside step function;
  Assume $m > n$ and $m/n = p \ldots q$, NewABPair = $A_1B_1, A_2B_1, \ldots, A_pB_1, A_{p+1}B_2, A_{p+2}B_2, \ldots, A_{2p}B_2, \ldots, A_{m-q+1}B_{x1}, A_{m-q+2}B_{x2}, \ldots, A_mB_{xq}$, where $x_1, x_2, .., x_q$ is a random permutation for the sequence from 1 to q; if $m \leq n$, just swap the index of AB;
  ABPair = ABpair $\cup$ NewABPair;
end

3.2.2. Find Primary Centre, Rotation, Scaling

This section shows how each pair AB is mapped onto votes for the centre, rotation, and scaling factor.

Centre: A line intersection method is proposed to find the centre. Using Figure 3 as an example, the coordinates of centre C can be calculated in terms of $\beta 1$, $\beta 2$, the R-Table (denoted as **R**), and the coordinates of $A(x_A, y_A)$ and $B(x_B, y_B)$.

Lines AC and BC can be written as linear entities as:

$$\begin{aligned} y_A &= a_A x_A + b_A \\ y_B &= a_B x_B + b_B \end{aligned} \tag{1}$$

where a_A, a_B, b_A, and b_B are parameters for the linear function. Then, the coordinates of C are:

$$C = \left(\frac{b_B - b_A}{a_A - a_B}, a_A \frac{b_B - b_A}{a_A - a_B} + b_A \right) \tag{2}$$

To find the coordinates in Equation (2), firstly, the index in the R-Table can be calculated as:

$$I = \alpha 1 - \operatorname{atan}\left(\frac{y_A - y_B}{x_A - x_B} \right) - \pi H(x_A - x_B) \tag{3}$$

where $H(x) = \frac{d}{dx} \max\{x, 0\}$ is the Heaviside step function. Therefore, $a_A = \tan(\mathbf{R}_{I,4})$ and $a_B = \tan(\mathbf{R}_{I,5})$, where $\mathbf{R}_{i,j}$ is the i^{th} row and j^{th} column in the R-Table. Then, b_A and b_B can be derived from (1).

Rotation : With the index I in (3), the rotation bringing the template to the target shape is:

$$r = \alpha 1 - \mathbf{R}_{I,2} \tag{4}$$

which is the current slope of A substituting the original value of the template stored in the R-Table.

Scaling : With the index I in (3), the scaling factor by which the template is multiplied to obtain the target shape can be calculated as the current distance AB divided by the original value of the template stored in the R-Table.

$$s = \frac{\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}}{\mathbf{R}_{I,3}} \tag{5}$$

3.2.3. Random Verify Scheme for Centre, Rotation, and Scaling Refinement

This paper proposes the random verify to assign a weighting to each configuration to deal with the one-to-many mapping problem: if a certain configuration is voted for,

the voted configuration should have a high intersection ratio with the target edge image, even in the presence of distortions. Firstly, given the R-Table and a configuration (centre, rotation, scaling), the random verify takes the shape from that given configuration back to the image space by Equation (6a,b). We denote $(\hat{x}, \hat{y})$, where I is a random row index in the R-Table and $\mathbf{c} = (c_x, c_y)$, r, and s are the candidate centre coordinate, rotation, and scaling, respectively.

$$\begin{aligned} \hat{x} &= c_x - s\mathbf{R}_{I,3}\cos(\beta 1 - r) \\ &= c_x - s\mathbf{R}_{I,3}\cos(\mathbf{R}_{I,4} - r) \end{aligned} \tag{6a}$$

$$\hat{y} = c_y - s\mathbf{R}_{I,3}\sin(\mathbf{R}_{I,4} - r) \tag{6b}$$

Secondly, for a given configuration, every row in the R-Table can be projected back to a point in the image space. The random verify involves randomly selecting rows from the R-Table for back projection. Figure 5 illustrates a case where the target shape presents distortion from the template. Two candidate configurations are randomly projected back into the image space with 5 and 7 pixels, respectively, shown as green and blue dots in the figure. To deal with shape distortion, each projected pixel is expanded to a square area $\pm R$ pixels away. Even in the presence of distortion, the expanded area is more likely to intersect the target shape. Intersection is considered to take place when the total number of pixels in the expanded range is above a set threshold. We denote the intersection percentage as $\tau, (0 \leq \tau \leq 1)$ to indicate whether the current configuration corresponds to a found shape. This random verify process, which returns a weighting, is denoted as $w = f_r(s, r, \mathbf{c}, \mathbf{R}, \mathbf{E}, N_r, R)$, where $\mathbf{E}$, N_r, and R are, respectively, the edge image of the target shape, the number of random projection pixels, and the expansion radius in pixels. We define its relation to τ as Equation (7).

$$w = \tau^k, k \in \mathbb{R} > 1 \tag{7}$$

Only configurations with w greater than a threshold are retained. An intersection percentage τ closer to 100% is trusted more than lower percentages, and the output weighting is set as a power of τ. Therefore, the configurations closer to the target are more likely to be found, whereas the wrong configurations are quickly eliminated. For example, for $k = 2$ and $\tau = 0.3$, more than 12 incorrectly voted configurations can override the true configuration because $1/0.3^2 \approx 11.1$. Note that τ = 0.3 is already a seriously polluted image. Conversely, this can happen for the "wrong" configuration to obtain $\tau = 1$ accidentally and for the true configuration to only obtain $\tau = 0.8$ due to shape distortion. In this case, if k is too large, this wrong configuration needs too many true votes to be eliminated. When $k = 3$, two true configuration votes can override the wrong votes because $1/0.8^3 \approx 1.95$. Therefore, we recommend k to be 2 or 3.

Due to the strong wrong vote compression ability, it is not necessary to gather a rich index to make the true configuration stands out. The benefit is that if most of the index is lost or polluted in the target image, the true configuration still can stand out; if the target image is less polluted, downsampling the multiple index found in the R-Table will not influence the final results, but will reduce the time consumption greatly, which means this random back projection process deals with the traditional one-to-many mapping problem in the R-Table and distortion efficiently.

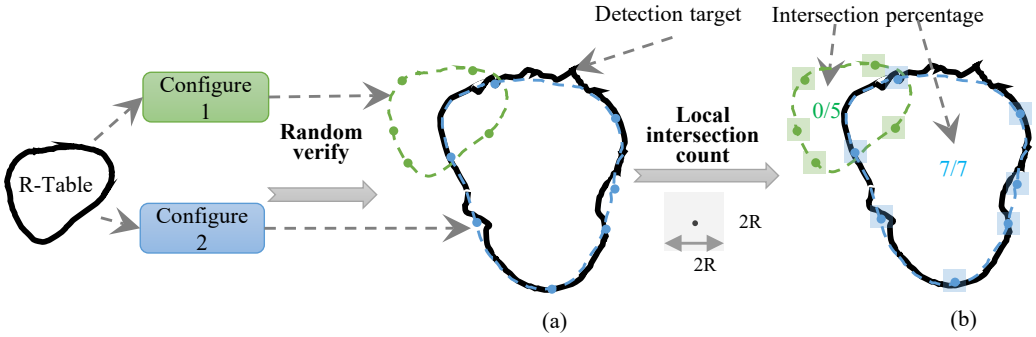


Figure 5. Diagram for the random verify. The green and blue dashed lines in (**a**,**b**) are the fully back-projected shape for Configurations 1 and 2. For the random verify, the configurations are projected back onto the target image with some random rows in the R-Table, so there are some random points that correspond to Configurations 1 and 2 in (**a**). Then, an intersection percentage of each configuration can be calculated with an expanded range for each point, as shown in (**b**).

3.2.4. Multi-Instance Classification

The multi-instance case features in many real applications. The target edge image may vote for the existence of several instances of the template shape. Each pair of edge pixels corresponds to one specific configuration and to a weight, if the index generated by the pair can be found in the R-Table. These configurations can be stored in a matrix **M**:

$$\mathbf{M} = \begin{bmatrix} \mathbf{c}_1 & r_1 & s_1 & w_1 \\ \mathbf{c}_2 & r_2 & s_2 & w_2 \\ \cdot & \cdot & \cdot & \cdot \\ \mathbf{c}_n & r_n & s_n & w_n \end{bmatrix} \tag{8}$$

where n is the total number of votes.

Due to noise or distortion, the centres stored in **M** usually do not coincide, but cluster discretely in the voting space. To estimate the precise location and strength of each shape's appearance, a Gaussian kernel function is used in the voting process.

$$P_{x_i y_i} = P_{x_i y_i} + w_i \exp\left(-\frac{(x - x_i)^2}{2\sigma_c^2} + \frac{(y - y_i)^2}{2\sigma_c^2}\right) \tag{9}$$

where $i = 1, 2, 3, \ldots, n$. $P_{x_i y_i}$ is a patch in the voting space, which has (x_i, y_i) as the patch centre. Likewise, the latter centre and rotation voting also use a Gaussian kernel for accumulation, as shown in (10).

$$P_{r_i} = P_{r_i} + w_i \exp\left(-\frac{(x - r_i)^2}{2\sigma_r^2}\right) \tag{10}$$

The overall algorithm for this multi-instance is given in Algorithm 3. Given the configuration matrix in Equation (8) and a cut-off threshold T, the algorithm returns the multi-instance centre coordinates and the corresponding strength, rotation, and scaling. The algorithm votes for the centre first, followed by searching for the multi-instance centres. It simply searches for peaks above the cut-off part of the max vote value, and the maximum is the corresponding strength. After obtaining a centre, it votes for a rotation and scaling for each centre. If different objects use the same angle difference, several target shapes can be detected after obtaining the relevant index. This process lends itself to parallelisation, which can be effectively processed by hardware [34].

Algorithm 3: Multi-instance classification.

Input: Configuration matrix, **M**; cut-off threshold $0 < T \leq 1$
Output: TureCentre, Strength, Rotation, Scale

$\mathbf{V}_c = \mathbf{0}$ is a 2D voting space for centre, which has the same size as the input image; TrueCentre = {empty}; Strength = {empty};
for *each row of* **M do**
  Vote $\mathbf{V}_c$ with Gaussian kernel as Eqn. (9);
end
C_{max} = max value in $\mathbf{V}_c$;
while *max value in* $\mathbf{V}_c \geq C_{max}T$ **do**
  NewTrueCentre = the max value position;
  TrueCentre = TrueCentre ∪ NewTrueCentre;
  Strength = strength ∪ max value in $\mathbf{V}_c$;
  Set the patch of the true centre in $\mathbf{V}_c$ as 0;
end
Initialise the 1D rotation and scale voting $\mathbf{v}_{rx}$ = 0; $\mathbf{v}_{sx}$=0; x = 1,2,3,. . . is the number of TrueCentres;
for *each row of* **M do**
  if **c** *is in certain range of the i-th TrueCentre* **then**
    vote $\mathbf{v}_{ri}$ and $\mathbf{v}_{si}$ with Gaussian kernel as Eqn. (10);
end
find the max value position in $\mathbf{v}_{rx}$ and $\mathbf{v}_{sx}$ as rotation and scale.

4. Experiments and Discussion

The RV-GHT algorithm resolves the two fatal issues for the traditional GHT, i.e., the high computational cost in the presence of rotation or scaling and the failure to find target shapes that are similar, but not identical to the template shape. The experiments in this section validate our efficiency and accuracy claims.

Comparisons were made against the most efficient of the GHT algorithms in the literature, namely the PI-GHT, GDP-GHT, GFGHT, and FRGHT. The GDP-GHT uses edge pixel pairs with a constant slope difference to calculate the R-Table index and, for each pair, stores the displacement vector in the centre. To improve the overall performance, our algorithm uses the scaling factor obtained from *AB*, *AC*, and *BC* in order to remove some wrong projections for the GDP-GHT. The GFGHT, PI-GHT, and FRGHT divide the images into small blocks and store a reference point in the R-Table. The FRGHT randomly chooses a block as a seed and iteratively looks for a subsequent point that is further than a distance threshold and that has never been chosen before; the found point *B* in the last iteration is used as Point *A* in the next iteration; this index chain is also linked to the vote weighting.

4.1. Dataset

Existing publicly available datasets are too simple for efficiency and robustness testing because: (1) there is no real case shape distortion; (2) there is no interference from ambient light or reflective surfaces; (3) existing shapes are relatively regular, which goes against real applications. Our dataset was built to feature ten radically different shapes, shown in Figure 6. Theses shapes were chosen to be relevant to a variety of potential real applications. For example, `S1` refers to a challenging, highly reflective object positioning; `S2` is a challenging case for accurate rotation and scaling detection due to the rich one-to-many mapping in the R-Table. `S8`, `S9`, and `S10` represents infrared, ultrasound, and X-ray inspection applications in manufacturing, respectively. Other source shapes are some common components' positioning in automatic assembly and packaging line.

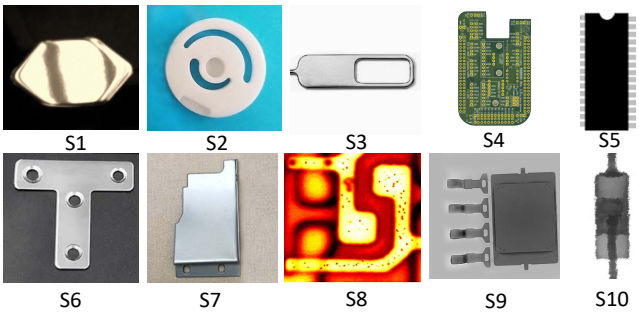


Figure 6. The source shapes in the test dataset, numbered from S1 to S10.

4.2. Accuracy Analysis

Even in the absence of shape distortion, the one-to-many mapping pollutes the voting, especially when rotation and scaling are also present. The corresponding template for source images in the test dataset are shown in Figure 7 by Canny edge detection. Compared to deep learning methods, which require a huge dataset for pre-training, this method only requires a single typical image to build a template contour shape. Building a simple, yet unique template for each source shape is preferred, which helps to improve efficiency. Every shape is assigned a centre of coordinates; the position and dimension illustrated in Figure 6 were set as templates. The R-Tables for each algorithm and every shape are generated before the detection process.

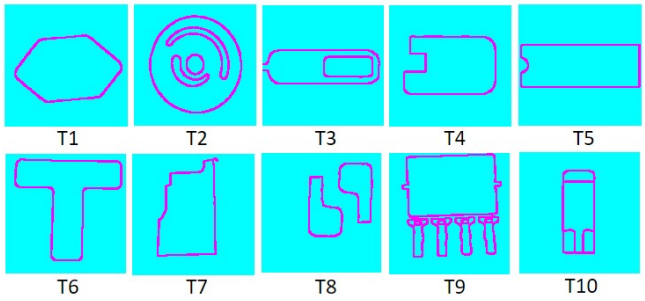


Figure 7. Template for objects in the test dataset, numbered from T1 to T10.

4.2.1. Qualitative Accuracy Performance

To have a first impression of the accuracy of the RV-GHT in dealing with one-to-many mapping, the first experiment involved the most ambiguity-prone shape, S8 for shape centre and shape and S2 for rotation and scaling. Both shapes were transformed with an anti-clockwise rotation of 53° and a scaling factor of 0.73. The constant angle difference for our RV-GHT and the GDP-GHT were set as 60°; the block size of the GFGHT and FRGHT was set to 10 pixels. The distance threshold for the found pair was 10 and 60 pixels, respectively, for our RV-GHT and the GFGHT/FRGHT. Accumulation with one pixel for each vote is too sensitive to noise or distortion: any noise or distortion will cause position misalignment to affect the voting. A more robust scheme is voting with a Gaussian kernel, as shown in Equations (9) and (10). To make the vote fair, in our experiments, we allowed the use of the Gaussian kernel voting scheme, even though the various algorithms did not originally make use of such a scheme.

The voting results for the centre, rotation, and scaling for these two challenging shapes are shown in Figures 8–10, respectively. The proposed RV-GHT had the best peak voting. Due to shape S8 having a serious one-to-many mapping, the GDP-GHT method found many centres that were nowhere near the ground truth, as shown in Figure 8. The GFGHT and FRGHT had a cleaner vote space because the check point removed some wrong votes, but the resulting centres still did not intersect with the ground truth due to the representation error using blocks. Smaller blocks can improve the accuracy, but will

greatly increase the index number. Figures 9 and 10 show that the sidelobe for the RV-GHT was much narrower than its counterparts, although all methods detected the configuration correctly; this also means a better confidence level for our method. In the presence of background noise, the peak for the rotation and scaling votes of other methods can easily be defeated.

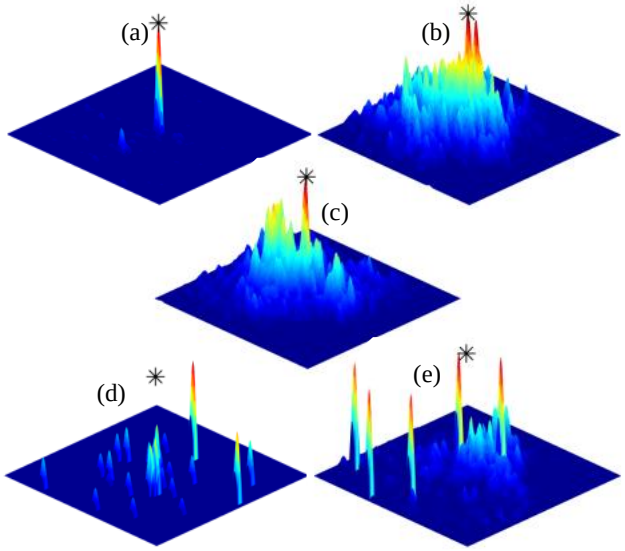


Figure 8. Centre voting for S8 (**a**) for the proposed RV-GHT, (**b**) the GDP-GHT, (**c**) the PI-GHT, (**d**) the GFGHT, and (**e**) the FRGHT under the configuration of rotation = 53° and scaling = 0.73. The ground truth centre is labelled as a black star near the peak.

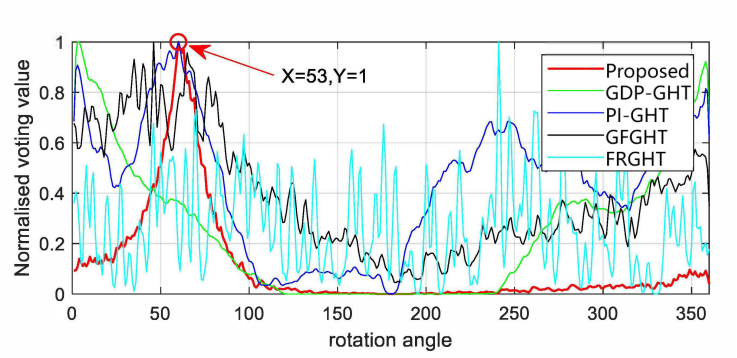


Figure 9. The voting space for rotation for the most-challenging shape S2 with the configuration of rotation = 53° and scale = 0.73.

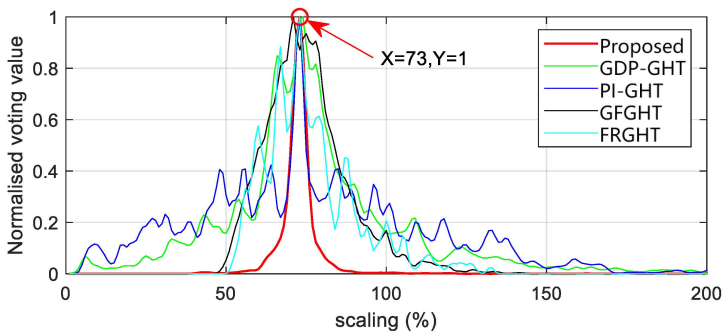


Figure 10. The voting space for scale for the most-challenging shape S2 with the configuration of rotation = 53° and scale = 0.73.

Figure 11 shows the final detection results of some typical shapes. The proposed method had the cleanest voting space (shown as the white cloud in Figure 11), and the positioning results were accurate. Other methods were influenced by the one-to-many mapping seriously.

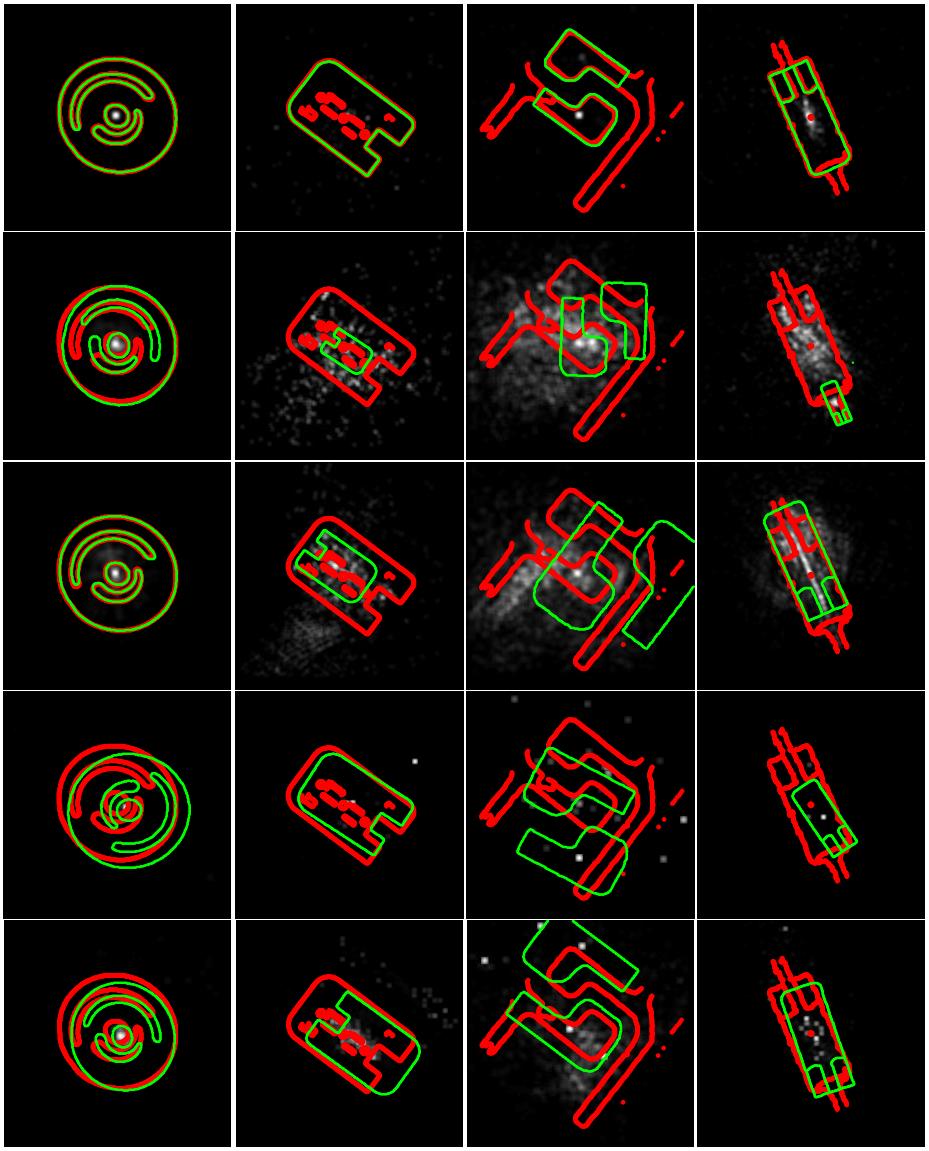


Figure 11. Examples of positioning results for some typical shapes (columns) for the proposed RV-GHT, the GDP-GHT, the PI-GHT, the GFGHT, and the FRGHT (in top-down order). The red and green contour are the source image contour and detection results, respectively. The white clouds are the centre voting results.

4.2.2. Qualitative Multi-Instance Detection Performance

Figure 12 shows our algorithm's multi-instance detection ability. Multi-instance detection is not a key argument for this paper, so a sample image containing multiple instances of three shapes was made just to show that our proposed method can deal with it. The detection targets were `T1` and `T4`, and `T7` was the interference. Only the results of the RV-GHT are given because other GHT methods do not contain a procedure to deal with multiple instances for different rotations or scalings. The outlier shapes generated almost no votes in the voting space, which shows that our method is robust to background noise. All targets were successfully detected even if they had different rotations and scaling factors.

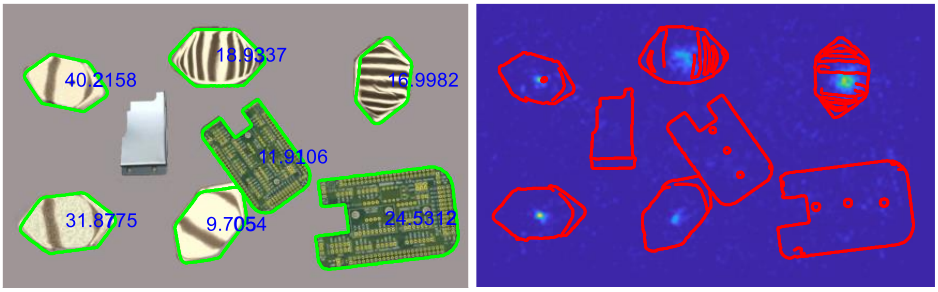


Figure 12. Multi-instance positioning results for shapes `T1` and `T4` (**left**) and the centre votes' response (**right**).

4.2.3. Quantitative Analysis of Key Parameters for Dealing with Shape Distortion

This experiment analysed key parameters in the proposed RV-GHT for dealing with shape distortion. Shape distortion is one of the major challenges of GHT methods, i.e., when the target shape is similar, but not identical to the template, the performance becomes unacceptably poor. Our proposed method uses a series of procedures to deal with shape distortion including line intersection to find the centre, the random verify, and Gaussian kernel voting. The two key parameters that influence the performance are the back projection point number N_r and the expansion radius R in Figure 5.

Based on the same dataset shown in Figure 6, different levels of salt and pepper noise were added manually to the template shapes. Figure 13a illustrates some distorted versions for `S2`. Under different salt and pepper noise densities (from 0 to 0.35 with a stepsize of 0.05), there were partial missing, shape distortion, and noise incurred. The *accuracy* curves under such settings are given in Figure 14 using Monte Carlo simulation. Every template shape in Figure 6 was tested and transformed to a random rotation and scale, and every noise density was repeated 50 times to obtain the average performance. *Accuracy* is defined as the rate of total true positives (TPs) and true negatives (TNs) for all test samples (N), i.e., $accuracy \equiv (TP + TN)/N$. The true positives for centres, rotation, and scaling are defined, respectively, when the detected results are within 5 pixels, within 5°, and within a factor of 0.05 of the ground truth. These thresholds do not need to be strictly so; other values that are not too relaxed will not influence the conclusion. Users should set them according to the error tolerance of their task.

For centre, rotation, and scale accuracy, increasing N_r improved the ability of compressing the one-to-many mapping, so $0 \leq N_r \leq Number\ of\ rows\ in\ R{-}Table$. $N_r = 50$ had slightly better accuracy than $N_r = 15$ pixels, but will increase the computation time. As for R, $R = 5$ showed the best performance. The principle behind this is that, when R is close to 0, the algorithm will give more weight to the identical part and use it to detect the shape. When R increases from 1, the shape distortion tolerance improved, but it decreased the ability of compressing the one-to-many mapping. Increasing R will not keep improving the tolerance for shape distortion. If $R = 0$ or R approaches or becomes greater than half of the image size, the algorithm loses the ability of compressing the one-to-many mapping totally. However, if there is not an absolutely identical part in a real application, setting R as 5 to 10 is a better choice than 1 to allow shape distortion.

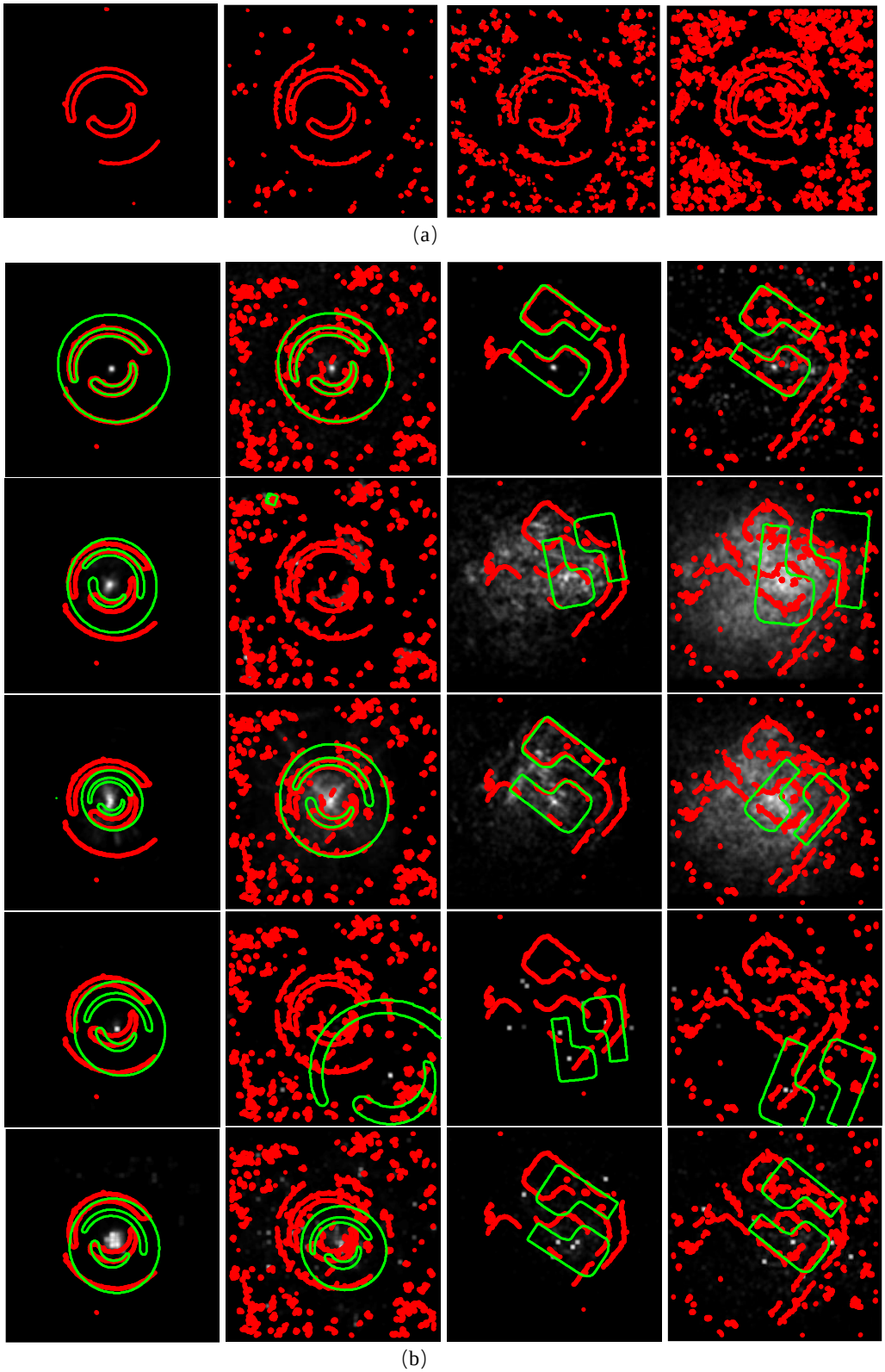


Figure 13. (**a**) S2 under salt and pepper noise densities of 0.1, 0.2, 0.3, and 0.4 (left to right). (**b**) Demo of positioning results for the proposed RV-GHT, the GDP-GHT, the PI-GHT, the GFGHT, and the FRGHT (in top-down order) for S2 and S8 with a noise density of 0.1 and 0.3.

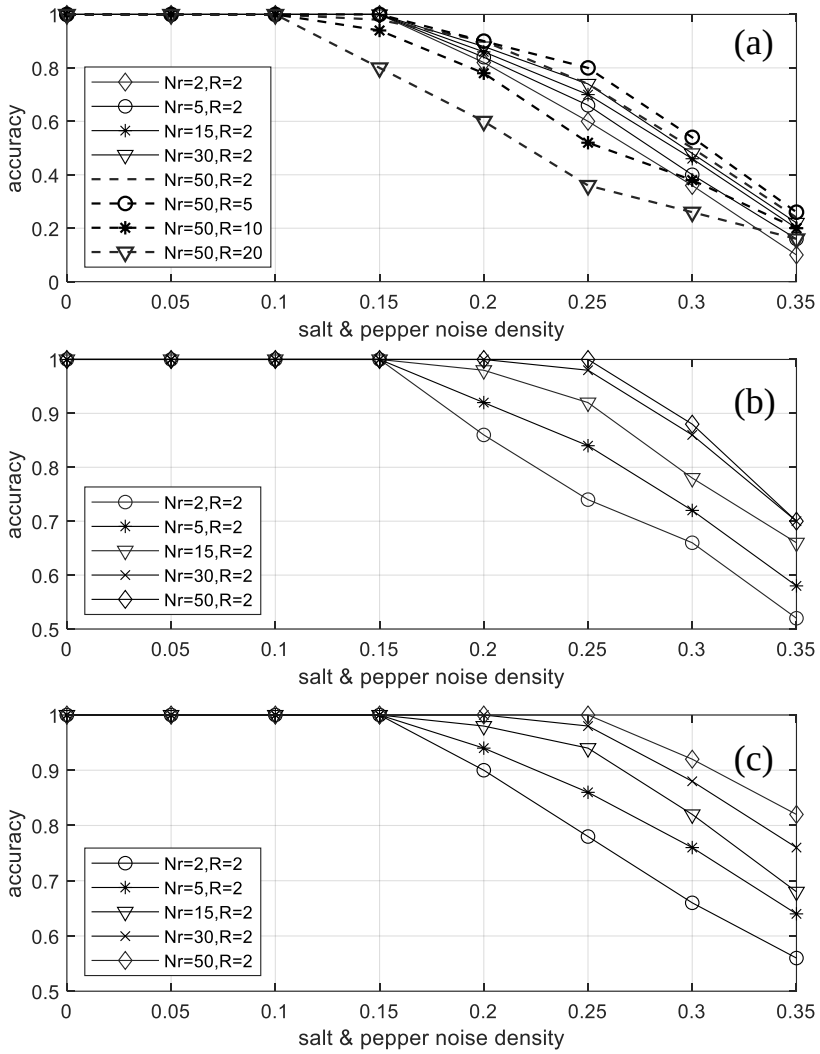


Figure 14. Accuracy vs. salt and pepper noise density for (**a**) centre, (**b**) rotation, and (**c**) scale under various R and N_r in Figure 5 for the proposed RV-GHT using the Monte Carlo method.

4.2.4. Quantitative Accuracy Performance under Shape Distortion

This section compares our proposed method with other counterparts under various levels of distortion. We set $N_r = 15$ and $R = 5$ according to the analysis in Section 4.2.3. Some of the detection results are shown in Figure 13b. Salt and pepper noise brought partial shape loss, shape distortion, and interference. Only the proposed RV-GHT handled all these cases successfully.

For a quantitative study, the more comprehensive evaluation metrics of *precision* and *recall* were used because *accuracy* ignores the false negatives (FNs) and false positives (FNs), where $precision \equiv TP/(TP + FP)$, $recall \equiv TP/(TP + FN)$. *Precision* describes how much ground truth there is within the predicted truth, and *recall* describes how much ground truth has been extracted. The *precision* vs. *recall* curves for centre, rotation, and scaling are shown in Figure 15. The same Monte Carlo method and setting as in Figure 14 were used. N_r and R were set as 15 and 5, respectively. Figure 15 shows that the proposed method outperformed all other state-of-the art methods greatly in the accuracy for centre, rotation, and scaling detection.

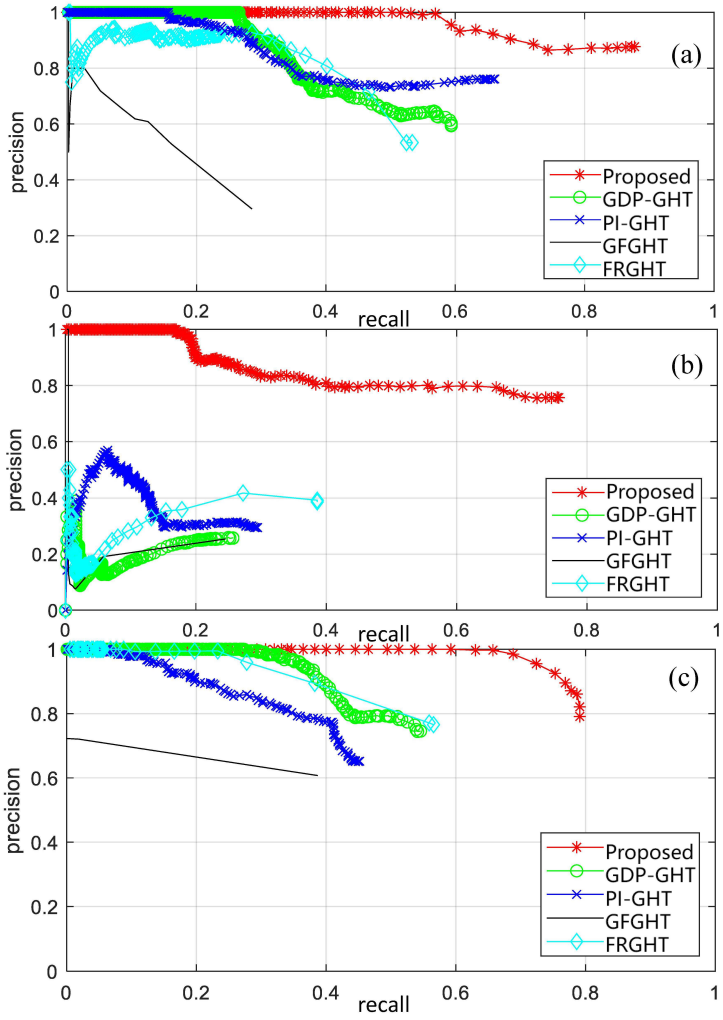


Figure 15. Precision vs. recall for (**a**) centre, (**b**) rotation, and (**c**) scaling estimation using the Monte Carlo method.

4.3. Efficiency Analysis

The time complexity of the traditional GHT method is $N_r N_s N_{idx}$, where the respective numbers refer to all possible rotations, scalings, and indices computed for the target image. Efficient GHT algorithms eliminate the iteration for rotations or scaling, preferably both, leading to a time complexity of around N_{idx}. To further analyse the complexity of efficient GHT methods, this paper recalls the processes that are common to these algorithms. During the detection process, after obtaining the index for the target image, the algorithm will look up this index in the R-Table. When one-to-many mappings are present in the R-Table, an index may correspond to multiple rows, each of which will correspond to a projection and will yield the calculation of a centre, rotation, and scaling triplet, followed by the accumulation for each parameter.

The calculation of these parameters can vary greatly with the particularities of each shape. This makes it nearly impossible to find a closed-form equation for the time complexity. However, from the description of our method, it should be clear that introducing the random verify will greatly reduce the accumulation number by downsampling and by removing some of the wrong voting. Other GHT methods count towards the accumulation of all the relevant rows in the R-Table. In our method, whilst the number of random verify points also increases the overall time, those extra back projections improve the accuracy of the results at the expense of a small amount of extra computation.

Our time efficiency tests were carried out on the dataset from Section 4.1, on a laptop with a Core i7-7500u CPU. Each shape was considered in a series of rotational positions so as to obtain the average runtime. No scaling or shape distortion were applied. The overall runtimes are given in Table 1. As expected, the random verify point number (N_r) caused the overall time to increase. The normalised runtime efficiency shared the same trend for all shapes, which had a gain factor of only around 0.6. Although the proposed method was not the most time-efficient one, it still outperformed the others in most cases when $N_r = 5$; this setting already showed a great improvement in accuracy, and $N_r \geq 15$ did not improve the overall efficiency much.

Table 1. Runtime efficiency for all shapes in the dataset for different algorithms in seconds.

Shape No.	1	2	3	4	5	6	7	8	9	10	Average
Pro (5)	0.0238	0.0244	0.0130	0.0104	0.0133	0.0466	0.0480	0.0601	0.0675	0.0414	
	=1	=1	=1	=1	=1	=1	=1	=1	=1	=1	1
Pro (15)	0.0417	0.0449	0.0229	0.0178	0.0200	0.0705	0.0767	0.0903	0.1264	0.0676	
	=1.8	=1.8	=1.8	=1.7	=1.5	=1.5	=1.6	=1.5	=1.9	=1.6	1.67
Pro (30)	0.0565	0.0773	0.0372	0.0306	0.0309	0.1407	0.1533	0.1349	0.1714	0.1068	
	=2.4	=3.2	=2.8	=3.0	=2.3	=3.0	=3.2	=2.3	=2.5	=2.6	2.73
Pro (50)	0.0834	0.1043	0.0688	0.0360	0.0527	0.2015	0.2192	0.2071	0.2529	0.1578	
	=3.5	=4.3	=5.3	=3.5	=4.0	=4.3	=4.6	=3.4	=3.7	=3.8	4.04
GDP-GHT	0.0497	0.0689	0.0092	0.0069	0.0141	0.1077	0.0899	0.1824	0.0252	0.0677	
	=2.1	=2.9	=0.7	**=0.66**	=1.1	=2.3	=1.9	=3.0	=0.37	=1.6	1.66
PI-GHT	0.0234	0.0820	0.0062	0.0113	0.0015	0.0733	0.1249	0.0782	0.0489	0.0615	
	=0.98	=3.4	**=0.48**	=1.1	**=0.12**	=1.6	=2.6	=1.3	=0.73	=1.5	1.37
FGHT	0.0259	0.0560	0.0149	0.0098	0.0244	0.0306	0.0189	0.0302	0.0224	0.0085	
	=1.1	=2.3	=1.1	=0.95	=1.8	=0.66	=0.39	=0.5	=0.33	=0.21	0.941
FRGHT	0.0209	0.0212	0.0109	0.0081	0.0066	0.0081	0.0095	0.0203	0.0078	0.0012	
	=0.88	**=0.87**	=0.84	=0.79	=0.49	**=0.17**	**=0.2**	**=0.34**	**=0.12**	**=0.03**	0.472

Note: The proposed RV-GHT with different random verify numbers is listed as **Pro (N_r)**. Normalised time is given bellow "=" in each cell. The last column is the average normalised time.

5. Conclusions

Efficient and robust component positioning is a basic need for visual robotic systems in automatic assembly and packaging lines. The proposed RV-GHT method addresses the challenges present in current GHT methods of low overall efficiency and poor accuracy when dealing with shape distortion.

Pixel pairs with a constant slope difference are used to index the R-Table. A new, reduced-size R-Table was proposed, which has only five columns. This is an improvement on the data storage, for example nine columns in the GFGHT or eight columns in the GDP-GHT. Its rows are further compressed by a minimum point set selection algorithm.

During the detection process, an index pair search algorithm obtains a minimum index without losing information. These schemes significantly reduce the search overhead without losing integrity. Subsequently, a line intersection method is used to find from the R-Table the primary centre and the corresponding rotation and scaling, which is more robust to shape distortion than traditional methods that simply use the displacement vectors. Then, a random back projection scheme assigns a weighting to each configuration. The random back projection significantly reduces the number of wrong votes, thereby enabling the overall algorithm to downsample the one-to-many mappings, obtaining a linear gain in efficiency.

The runtime was very low compared to the other GHT algorithms, placing our method in the state-of-the-art, simultaneously achieving a ground-breaking improvement in accuracy in distortion cases.

This paper targeted positioning tasks in visual robotic systems in automatic assembly and packaging lines, where the view angle is usually set at a constant or similar point. To benefit wider industrial applications, we will push this method to deal with 3D shape detection where the view angle becomes a key issue, but this is beyond the scope of this article.

Author Contributions: Conceptualization, Y.Z. and C.T.; methodology, Y.Z. and C.T.; software, Y.Z. and X.G.; validation, H.S.; formal analysis, Y.Z.; investigation, X.G.; resources, H.S.; writing—original draft preparation, Y.Z. and C.T.; writing—review and editing, H.S. and C.T.; supervision, C.T.; project administration, X.G.; funding acquisition, C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, Grant Numbers U21A20481 and 62103154; SINOMARCH, Grant Number ZDZX2021-4; Huazhong University of Science and Technology, Grant Numbers 2021XXJS097 and 20220819; the Science and Technology Research Project of PipeChina, Grant Number WZXGL202104.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the reported results by the authors can be sent by e-mail.

Acknowledgments: The authors would like to thank Irina Voiculescu of Oxford University, who helped with the English editing on an earlier version of this paper.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
2. Ballard, D.H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **1981**, *13*, 111–122. [CrossRef]
3. Mukhopadhyay, P.; Chaudhuri, B.B. A survey of Hough Transform. *Pattern Recognit.* **2015**, *48*, 993–1010. [CrossRef]
4. Ahmad, R.; Naz, S.; Razzak, I. Efficient skew detection and correction in scanned document images through clustering of probabilistic hough transforms. *Pattern Recognit. Lett.* **2021**, *152*, 93–99. [CrossRef]
5. Conti, C.; Romani, L.; Schenone, D. Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform. *Pattern Recognit.* **2018**, *74*, 64–76. [CrossRef]
6. Leibe, B.; Leonardis, A.; Schiele, B. Robust object detection with interleaved categorization and segmentation. *Int. J. Comput. Vis.* **2008**, *77*, 259–289. [CrossRef]
7. Gall, J.; Lempitsky, V. Class-specific Hough forests for object detection. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1022–1029. [CrossRef]
8. Payet, N.; Todorovic, S. Hough Forest Random Field for Object Recognition and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1066–1079. [CrossRef]
9. Xu, Z.; Shin, B.; Klette, R. Accurate and Robust Line Segment Extraction Using Minimum Entropy With Hough Transform. *IEEE Trans. Image Process.* **2015**, *24*, 813–822. [CrossRef]
10. Jia, Q.; Fan, X.; Luo, Z.; Song, L.; Qiu, T. A Fast Ellipse Detector Using Projective Invariant Pruning. *IEEE Trans. Image Process.* **2017**, *26*, 3665–3679. [CrossRef]
11. Lu, C.; Xia, S.; Shao, M.; Fu, Y. Arc-Support Line Segments Revisited: An Efficient High-Quality Ellipse Detection. *IEEE Trans. Image Process.* **2020**, *29*, 768–781. [CrossRef]
12. Hassanein, A.S.; Mohammad, S.; Sameer, M.; Ragab, M.E. A survey on Hough transform, theory, techniques and applications. *Int. J. Comput. Sci. Issues* **2015**, *12*, 139–156.
13. Jia, Q.; Fan, X.; Liu, Y.; Li, H.; Luo, Z.; Guo, H. Hierarchical projective invariant contexts for shape recognition. *Pattern Recognit.* **2016**, *52*, 358–374. [CrossRef]
14. Yang, C.; Wei, H.; Yu, Q. A novel method for 2D nonrigid partial shape matching. *Neurocomputing* **2018**, *275*, 1160–1176. [CrossRef]
15. Siang De, M.; Xing, C. Hough transform using slope and curvature as local properties to detect arbitrary 2D shapes. In Proceedings of the 9th International Conference on Pattern Recognition, Rome, Italy, 14–17 November 1988; pp. 511–513. [CrossRef]
16. Yang, H.; Zheng, S.; Lu, J.; Yin, Z. Polygon-Invariant Generalized Hough Transform for High-Speed Vision-Based Positioning. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 1367–1384. [CrossRef]
17. Yu, Y.; Guan, H.; Zai, D.; Ji, Z. Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-Hough-forests. *ISPRS J. Photogramm. Remote Sens.* **2016**, *112*, 50–64. [CrossRef]
18. Gall, J.; Yao, A.; Razavi, N.; Gool, L.V.; Lempitsky, V. Hough Forests for Object Detection, Tracking, and Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2188–2202. [CrossRef]

19. Zhang, X.; Mu, R.; Chen, K.; Yang, Y.; Chen, Y. Intelligent Hough Transform With Jaya to Detect the Diameter of Red-Hot Circular Workpiece. *IEEE Sens. J.* **2021**, *21*, 560–567. [CrossRef]
20. Dong, H.; Prasad, D.K.; Chen, I.M. Object Pose Estimation via Pruned Hough Forest With Combined Split Schemes for Robotic Grasp. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 1814–1821. [CrossRef]
21. Chen, H.; Gao, T.; Qian, G.; Chen, W.; Zhang, Y. Tensored Generalized Hough Transform for Object Detection in Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3503–3520. [CrossRef]
22. Aguado, A.S.; Montiel, E.; Nixon, M.S. Invariant characterisation of the Hough transform for pose estimation of arbitrary shapes. *Pattern Recognit.* **2002**, *35*, 1083–1097. [CrossRef]
23. Chen, H.Y.; Lin, Y.Y.; Chen, B.Y. Robust feature matching with alternate hough and inverted hough transforms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2762–2769.
24. Ser, P.K.; Siu, W.C. A New Generalized Hough Transform for the Detection of Irregular Objects. *J. Vis. Commun. Image Represent.* **1995**, *6*, 256–264. [CrossRef]
25. Chun-Pong, C.; Wan-Chi, S. Generalized dual-point Hough transform for object recognition. In Proceedings of the 1999 International Conference on Image Processing (Cat. 99CH36348), Kobe, Japan, 24–28 October 1999; Volume 1, pp. 560–564. [CrossRef]
26. Jeng, S.C.; Tsai, W.H. Scale- and orientation-invariant generalized hough transform-a new approach. *Pattern Recognit.* **1991**, *24*, 1037–1051. [CrossRef]
27. Ulrich, M.; Steger, C.; Baumgartner, A. Real-time object recognition using a modified generalized Hough transform. *Pattern Recognit.* **2003**, *36*, 2557–2570. [CrossRef]
28. Chau, C.P.; Siu, W.C. Adaptive dual-point Hough transform for object recognition. *Comput. Vis. Image Underst.* **2004**, *96*, 1–16.
29. Li, Y.; Xia, R.; Huang, Q.; Xie, W.; Li, X. Survey of Spatio-Temporal Interest Point Detection Algorithms in Video. *IEEE Access* **2017**, *5*, 10323–10331. [CrossRef]
30. Kimura, A.; Watanabe, T. Fast generalized Hough transform: Rotation, scale and translation invariant detection of arbitrary shapes. *IEICE Trans.* **1998**, *81-D-II*, 726–734.
31. Suetake, N.; Uchino, E.; Hirata, K. Generalized fuzzy Hough transform for detecting arbitrary shapes in a vague and noisy image. *Soft Comput.* **2006**, *10*, 1161–1168. [CrossRef]
32. Chiu, S.H.; Wen, C.Y.; Lee, J.H.; Lin, K.H.; Chen, H.M. A fast randomized generalized hough transform for arbitrary shape detection. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 1103–1116.
33. Xu, J.; Fu, K.; Sun, X. An invariant generalized Hough transform based method of inshore ships detection. In Proceedings of the 2011 IEEE International Symposium on Image and Data Fusion, Tengchong, China , 9–11 August 2011; pp. 1–4.
34. Geninatti, S.R.; Boemo, E.I. Real-Time Reconfigurable Processor to Detect Similarities in Compressed Video Using Generalized Hough Transformation. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2932–2946. [CrossRef]