

# Robust Semantic Template Matching Using a Superpixel Region Binary Descriptor

Hua Yang<sup>✉</sup>, *Member, IEEE*, Chenghui Huang, Feiyue Wang, Kaiyou Song<sup>✉</sup>, and Zhouping Yin, *Member, IEEE*

**Abstract**—Almost all conventional template-matching methods employ low-level image features to measure the similarity between a template image and a scene image using similarity measures, such as pixel intensity and pixel gradient. Although these methods have been widely used in many applications, they cannot simultaneously address all types of robustness challenges. In this paper, with the goal of simultaneously addressing the various challenges, we present a robust semantic template-matching (RSTM) approach. Inspired by the local binary descriptor, we propose a novel superpixel region binary descriptor (SRBD) to construct a multilevel semantic fusion feature vector for RSTM. SRBD uses a new kernel-distance-based simple linear iterative clustering method to extract the stable superpixels from the template image. Then, based on the average intensity difference between each superpixel region and its neighbors, the dominant gradient orientation of each superpixel can be obtained, and the semantic features of each superpixel can be described as the dominant orientation difference vector, which is coded as the rotation-invariant SRBD. In the offline matching phase, the fusion semantic feature vector of RSTM combines the multilevel SRBD features with different numbers of superpixels. In the online matching phase, to cope with rotation invariance, a marginal probability model is proposed and applied to locate the positions of template images in the scene image. Moreover, to accelerate computation, an image pyramid is employed. We conduct a series of experiments on a large dataset randomly selected from the MS COCO dataset to fully analyze the robustness of this approach. The experimental results show that RSTM simultaneously addresses rotation changes, scale changes, noise, occlusions, blur, nonlinear illumination changes, and deformation with high time efficiency while also outperforming the previous state-of-the-art template-matching methods.

**Index Terms**—Template matching, image matching, image feature, superpixels descriptor.

## I. INTRODUCTION

TEMPLATE matching [1], [2] is a popular and active research topic in image processing and computer vision fields and has been widely used in applications such as vision positioning [3], face recognition [4], visual tracking [5], object recognition [6], scene recognition [7] and robotic vision. Given

a template image and a scene image, a template-matching method is employed to find the most similar candidate area in the scene image, called a matched region. To meet practical vision-based positioning requirements, such as robotic pick-and-place operations, an ideal template-matching method should simultaneously be robust to a number of challenges, such as rotation invariance, scale invariance, noise, occlusion, blur, nonlinear illumination and deformation.

In recent decades, various template-matching methods have been proposed and developed to address these challenges. These methods can be classified into three main groups, namely, those based on pixel intensity, pixel gradient and local invariant features.

Pixel-intensity based methods include the sum of absolute differences (SAD) [8], the sum of squared differences (SSD) [9], the sequential similarity detection algorithm (SSDA) [10], the successive elimination algorithm (SEA) [11], the multilevel SEA (MSEA) [12], image moment invariants (IMIs) [13], normalized cross correlation (NCC) [14] and matching by tone mapping (MTM) [15]. The SAD, SSD, SSDA, SEA and MSEA methods use the sum of the pixel intensity differences between the template image and the candidate window of the scene image as the discriminative function for the similarity calculation; therefore, these methods are not suitable for the challenge of illumination change. To overcome this limitation, both NCC and MTM methods have been proposed that use cross-correlation calculations of normalized pixel intensity rather than the sum of gray intensity differences. The NCC and MTM methods are both robust against blur, noise and linear illumination changes, but NCC is not suitable for nonlinear illumination changes. MTM can suit nonlinear illumination changes because it can be viewed as a generalization of the NCC for nonlinear mappings. However, NCC and MTM are not suitable for occlusion.

Pixel-gradient based methods, such as the generalized Hough transform (GHT) [16], hierarchical modified GHT (M-GHT) [17], polygon-invariant GHT (PI-GHT) [18], shape-based matching (SBM) [19], gradient location-orientation histogram (GLOH) [20] and scale-invariant template matching using a histogram of dominant gradients (SITM-HDG) [7], utilize the gradient directions of edge pixel points as the feature vectors. Even when the illumination changes nonlinearly, the gradient direction of each edge pixel point remains unchanged. Therefore, the feature vector based on gradient direction is invariant to nonlinear illumination change. The

Manuscript received April 21, 2018; revised December 3, 2018; accepted January 12, 2019. Date of publication January 17, 2019; date of current version April 10, 2019. This work was supported in part by the National Science Foundation of China under Grants 51875228, 51475193, and 51327801, and in part by the Major Project Foundation of Hubei Province under Grant 2016AAA009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Raja Bala. (*Corresponding author: Hua Yang.*)

The authors are with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: huayang@hust.edu.cn).

Digital Object Identifier 10.1109/TIP.2019.2893743

1057-7149 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html) for more information.

similarity measurement used for this method counts paired similar edge points with the same gradient direction; thus, pixel-gradient methods are also highly robust to both clutter and occlusion problems. However, the edge pixel point gradient amplitude is severely influenced by blur and the edge pixel point gradient direction is sensitive to noise. Thus, pixel-gradient methods are unsuitable for template matching with noise or blur under a low signal-to-noise ratio (SNR).

Local-invariant methods are based on features such as best-buddies similarity-robust template matching (BBS) [21], the scale-invariant feature transform (SIFT) [22], very fast SIFT (VF-SIFT) [23] and the speeded-up robust feature (SURF) [24]. BBS is based on counting best-buddies pairs, whose points are mutual nearest neighbors and whose similarity is based on a statistical score. Therefore, despite being robust to clutter and occlusions, BBS is susceptible to noise and blur because it extracts information based on point units. SIFT, VF-SIFT and SURF first extract feature keypoints in the image scale space. Then, they establish feature descriptors based on the local regions around these keypoints. Finally, they match the feature points extracted from the template image to the ones extracted from the scene image. The use of feature points enables these types of methods to address the challenges of rotation invariance, scale invariance, linear illumination change, viewpoint change, occlusion, cluster and affine transformation. At present, local invariant methods outperform other template-matching methods and are widely used in image processing and computer vision fields for matching natural scene images that have sufficient feature keypoints. However, they are very sensitive to noise and blur, because the local-region feature descriptor is based on pixel-gradient orientation. They are also typically time-consuming due to the feature point extraction in the image scale space. Moreover, this type method does not perform well on some simple shapes with few characteristics—for example, smooth curves or regions without sufficient feature points.

Overall, the present template-matching methods utilize only low-level image features to calculate similarity, yet some (such as SBM) are already widely used in robotics and manufacturing equipment. Each template-matching method has specific advantages and drawbacks that—while suitable for a particular application—cannot simultaneously deal with all the possible challenges, particularly occlusion, blur and noise under a lower SNR. In applications that require visual positioning, such as robot vision and machine vision, the captured image is usually degraded and a variety of occlusions, noise, blur or even deformation issues can occur because industrial production environments are notoriously difficult. Therefore, there is a significant demand in areas such as high-performance robotics and manufacturing equipment to find a more robust template-matching method under low SNR, particularly for pick-and-place vision-based positioning.

To meet this demand, in this study, a novel superpixel region binary descriptor (SRBD) is proposed as a multi-level semantic feature for robust template matching. The superpixels are extracted by KD-SLIC from a template image, and the dominant superpixel orientation is estimated based on the intensity differences between the current superpixel

and its neighborhoods. Then, the corresponding dominant orientation difference vector between the current superpixel and its neighborhoods can be calculated. A rotation-invariant SRBD can be obtained by coding the orientation difference vector to one binary vector. Furthermore, a robust semantic template-matching method (RSTM) is introduced that uses the SRBD feature. RSTM employs a single marginal probability model to measure similarity and can simultaneously address rotation changes, scale changes, noise, occlusion, blur, non-linear illumination changes, and deformation.

The remainder of this paper is organized as follows: In Section II, related works concerning superpixels and local binary feature are briefly introduced. In Section III, SRBD is proposed in detail. RSTM, which is based on SRBD, is introduced in Section IV, and its computational complexity is discussed. Section V presents a set of experiments that demonstrate the performance of the proposed RSTM algorithm in terms of robustness and computational time. Finally, Section VI concludes the paper.

## II. RELATED WORKS

In this section, existing superpixel extraction methods are introduced in detail and local binary descriptors are presented and discussed.

### A. Superpixels

The concept of superpixels was first introduced by Ren and Malik [25]. A superpixel groups pixels into perceptually meaningful atomic uniform regions that can be utilized to replace the rigid pixel-grid structure of digital images. It is both more convenient and more effective to extract feature vectors based on regions rather than on pixels. The present methods for superpixel generation are widely used as preprocessing steps in many image processing and computer vision tasks to reduce the computational complexity, including segmentation [26], tracking [27], object localization [28], depth estimation [29] and body model estimation [30].

In recent decades, numerous superpixel generation methods have been proposed to improve the three most important superpixel characteristics: boundary adherence, uniform intensity and compactness (COM). These methods can be divided into three main groups [31]: graph-based methods, gradient-ascent based methods and *K*-means based methods.

Graph-based methods model an image as a graph by treating pixels as nodes and the similarity between a current pixel and its neighboring pixels as an edge weight. Then, an energy function is defined on that graph and solved to generate the superpixel regions. The most popular graph-based methods include the normalized cuts algorithm (NCut) [32], superpixels lattice algorithm (SL) [33] and graph-based approach (GB) [34]. NCut generates regular, visually pleasing superpixels; however, the method is time consuming, and its boundary adherence is not ideal. SL produces a regular grid of superpixels, but the output quality and speed are influenced by the pre-computed boundary maps. GB performs rapidly, but the superpixels produced by GB have irregular sizes and shapes.

Gradient-ascent-based methods generate superpixel regions based on image gradient information. These methods include the watershed approach (WS) [35], mean shift approach (MS) [36], quick shift approach (QS) [37] and so on. WS is highly effective but often generates irregular superpixels when the number and COM of its superpixels are not set appropriately. MS is relatively robust in practice but is very slow, and the generated superpixels are irregular and oversegmented. While QS is faster than MS, the number of superpixels it generates is uncontrolled.

$K$ -means-based methods utilize the rough initialization of cluster centers to generate coarse superpixels and then refine these clusters until a convergence condition is met. Simple linear iterative clustering (SLIC) [31] is the most popular  $K$ -means based method, and it is computationally more efficient than previous methods. The majority of SLIC's superpixels have regular sizes and shapes, and they adhere well to the boundaries.

In general, SLIC is quite effective, and it can achieve a satisfactory tradeoff between superpixel COM and adherence to object boundaries. However, the superpixels resulting from the conventional SLIC method do not always adhere well to the boundaries. Therefore, a modified SLIC, namely, KD-SLIC, is introduced in this study to extract more stable superpixels from template images.

### B. Local Binary Descriptor

The local binary descriptor is one of the most active research areas in the fields of image processing and computer vision. Local binary descriptors are widely used in object classification and detection, object tracking, image retrieval and so on. A local binary descriptor encodes the local properties of the image feature point neighborhood as a single numerical vector. Many local binary descriptors have been proposed and developed in the past decade.

Hand-crafted local binary descriptors compare a pixel's intensity with the intensities of its neighborhood pixels to create the binary codes. The local binary pattern (LBP) approach [38] and its extensions use the differences in a pixel's intensity compared with its neighborhood pixels to represent the textural characteristics of images. The local intensity order pattern (LIOP) [39] encodes both the local intensity ordinal information of each pixel and the overall intensity ordinal information by dividing a local patch into subregions. The binary robust independent elementary feature (BRIEF) [40] compares the pixel intensities of random pairs of points with a local patch to perform fast calculations of the binary vectors, but it is not rotation-invariant. To overcome this BRIEF limitation, oriented fast BRIEF (ORB) [41] was proposed to address both scale and orientation invariance by using scale pyramids and orientation operators. Binary robust invariant scalable keypoint (BRISK) [42] utilizes a circular sampling pattern instead of the random sampling pattern of BRIEF. Fast retina keypoint (FREAK) [43], which was inspired by the retina of the human visual system, uses a retinal sampling pattern to improve BRISK's performance. These hand-crafted descriptors are both fast and

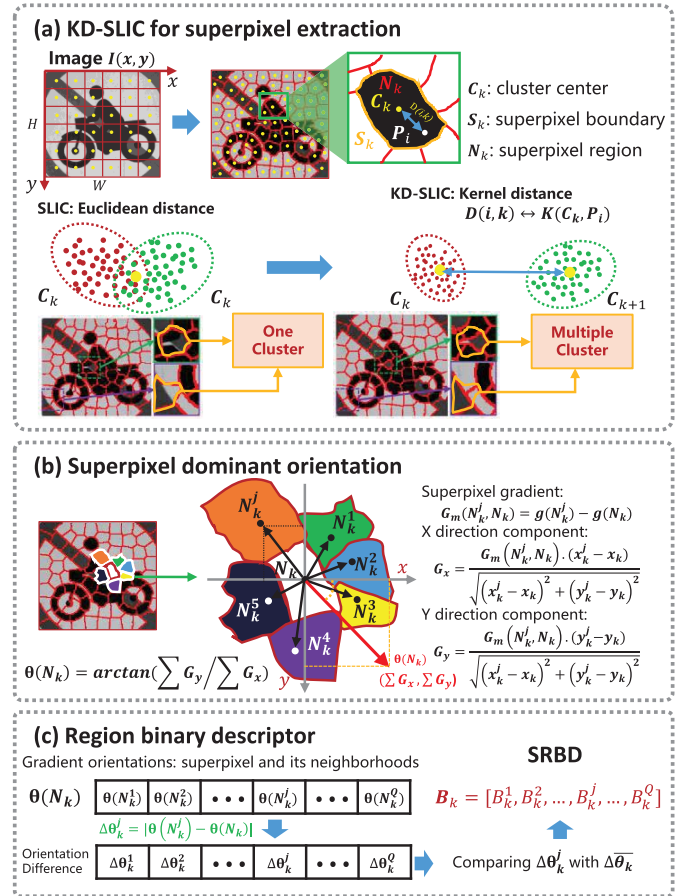


Fig. 1. Flowchart of the SRBD process. (a) illustrates that KD-SLIC based on exponential distance metric adheres to the boundaries better than SLIC; (b) illustrates the procedure of calculating the dominant superpixel orientation with the average intensity of the superpixel region; (c) illustrates the procedure of calculating an SRBD with the dominant orientations.

effective; however, they are susceptible to image noise and image transformation because they use only a simple intensity comparison.

In this study, inspired by the properties of local binary descriptors, a superpixel region binary descriptor is proposed that simultaneously addresses all the major challenges involved in template matching. While a local binary descriptor encodes an image patch to binary code based on the intensity differences between a central pixel and its neighborhood pixels, the superpixel region binary descriptor encodes the relationship between central superpixels and neighborhood superpixels; thus, it includes more semantic information and is more robust.

### III. SUPERPIXEL REGION BINARY DESCRIPTOR

In this section, the novel mid-level image descriptor SRBD is introduced in detail. The proposed SRBD descriptor, shown in Fig.1, utilizes the kernel-distance based simple linear iterative clustering (KD-SLIC) method to extract stable superpixels. Then, the dominant orientation of each superpixel is determined using the gradient information of its neighborhood superpixels. The dominant orientation difference vector between each superpixel and its neighborhoods can be directly



calculated; this vector is then utilized to code the superpixel region as the binary descriptor.

#### A. KD-SLIC for Superpixel Extraction

SLIC is the most popular and widely adopted of the current superpixel extraction methods because of its practicality. However, the superpixels resulting from conventional SLIC sometimes do not perform well because they adhere to the weak gradient edges due to the simple linear Euclidean distance metric used in SLIC. This problem is illustrated in Fig.1(a), which shows that the superpixels extracted by conventional SLIC do not adhere well to the boundaries. In general, the more stable the superpixel extraction process is, the better the superpixel descriptor is. Therefore, in this study, an improved KD-SLIC method is employed for superpixel extraction.

The KD-SLIC algorithm is a type of K-means clustering method, the solution process of which can be viewed as an expectation-maximization (EM) algorithm [44]. The function of the KD-SLIC algorithm is to divide the pixels of the entire image into K clusters, where each cluster represents a superpixel region. As a type of K-means algorithm, the KD-SLIC algorithm consists of three steps, namely, initialization, label assignment and update, which correspond to the initialization, expectation and maximization steps of the EM algorithm.

In the KD-SLIC initialization step, the image  $I(x, y)$  is divided into  $K$  regular grid subsets called clusters, where  $x \in (1, \dots, W)$  and  $y \in (1, \dots, H)$  indicate the horizontal and vertical coordinates of the image respectively, and the image resolution is  $W \times H$ . The grid interval is calculated by the following equation:

$$R_s = \sqrt{\frac{W \times H}{K}}. \quad (1)$$

The initial location of each cluster center is the center of the regular grid. The feature vector of the  $i$ th image pixel  $P_i$  can be constructed as  $[L_i, a_i, b_i, x_i, y_i]$ , where  $[L_i, a_i, b_i]$  represents the color values of the  $i$ th image pixel. To adapt the method for grayscale images, we define  $L_i = a_i = b_i = \eta_i$ , where  $\eta_i$  is the gray value of the  $i$ th image pixel. Thus, the feature vector of the initial cluster is  $C_k = [L_k, a_k, b_k, x_k, y_k]$ . Then, each cluster center is moved to the location corresponding to the lowest pixel gradient magnitude within a local  $5 \times 5$  neighborhood to avoid centering a superpixel on an edge or on a noisy pixel.

In the label assignment step, each pixel is associated with the nearest cluster center and for every cluster center, only pixels in a  $2R_s \times 2R_s$  region are searched. The purpose of this step is to update the category to which each pixel belongs, which is equivalent to the expectation step in the EM algorithm. The procedure is performed by means of a distance measure  $D$ . To enhance the discriminative power of each pixel, we define a novel kernel distance  $D(i, k)$  between each pixel feature vector  $P_i$  and the cluster-center feature vector  $C_k$ , which—compared with the original k-means algorithm—restrains the clustering procedure to a compact local displacement of the original cluster borders.  $D(i, k)$  can be viewed

as an exponential formula with quadratic terms that penalizes points far from the cluster center. The equations are as follow:

$$\begin{cases} \kappa_L(i, k) = e^{-\frac{(L_i - L_k)^2}{2\sigma^2 R_L^2}} \\ \kappa_a(i, k) = e^{-\frac{(a_i - a_k)^2}{2\sigma^2 R_a^2}} \\ \kappa_b(i, k) = e^{-\frac{(b_i - b_k)^2}{2\sigma^2 R_b^2}} \end{cases} \quad (2)$$

$$\begin{cases} d_c(i, k) = \sqrt{\kappa_L(i, k) + \kappa_a(i, k) + \kappa_b(i, k)} \\ d_s(i, k) = \sqrt{e^{-\frac{(x_i - x_k)^2 + (y_i - y_k)^2}{2\sigma^2 R_s^2}}} \end{cases} \quad (3)$$

$$D(i, k) = d_c(i, k) + d_s(i, k) \quad (4)$$

where  $d_c(i, k)$  denotes the kernel distance in CIELAB color space, and  $d_s(i, k)$  indicates the kernel distance in 2D image coordinate space. Here,  $R_L = L_{max} - L_{min} + 1$  is the  $L$  component value range, where  $L_{max}$  is the maximum  $L$  value,  $L_{min}$  is the minimum  $L$  value, and  $R_a = a_{max} - a_{min} + 1$  is the  $a$  component value range, where  $a_{max}$  is the maximum  $a$  value and  $a_{min}$  is the minimum  $a$  value.  $R_b = b_{max} - b_{min} + 1$  is the  $b$  component value range, where  $b_{max}$  is the maximum  $b$  value and  $b_{min}$  is the minimum  $b$  value, and  $R_s$  is the maximum 2D image coordinate distance. Here,  $L_{max}$ ,  $L_{min}$ ,  $a_{max}$ ,  $a_{min}$ ,  $b_{max}$  and  $b_{min}$  can be directly calculated from the image  $I(x, y)$ .

The purpose of the update step is to update the mean vector of each cluster center, which is equivalent to the maximization step in the EM algorithm. After all the image pixels  $I(x_i, y_i)$  have been assigned to the closest cluster centers  $C_k$ , the new cluster centers can be updated by the mean pixel feature vector of all the image pixels belonging to the same cluster  $N_k$ . Both the clustering and updating operators of KD-SLIC are conducted iteratively until the residual error of the cluster center locations is below a threshold,  $\varepsilon$ , or the maximum number of iterations,  $\gamma$ , has been reached. In this paper, the threshold  $\varepsilon$  is set as  $10^{-6}$  and the iteration maximum number  $\gamma$  is set as 10.

Fig.1(a) shows a comparison of the results for SLIC and KD-SLIC. The superpixels extracted by KD-SLIC adhere to the boundaries better than the do those extracted by SLIC because SLIC uses the linear Euclidean distance metric in both CIELAB color space and 2D image coordinate space. In contrast, the proposed KD-SLIC method classifies one cluster, which has slightly different pixel-feature distance differences in Euclidean space, into multiple clusters in Kernel space. The KD-SLIC method treats the various Euclidean distances in both CIELAB space and image coordinate space differently according to the kernel function; therefore, the similar pixels in both CIELAB color space and the adjacent pixels are easily grouped into one compact superpixel.

#### B. Superpixel Dominant Orientation

As we know, pixel intensity is sensitive to illumination changes; however, the intensity difference between each pixel and its neighborhood, called a pixel gradient, is robust to illumination changes. However, this pixel feature is a low-level image feature that is severely affected by noise and

blur. Inspired by this property, in this study, the concept of superpixel dominant orientation is introduced in detail.

A pixel gradient can be calculated using forward difference, backward difference, or central difference because its neighborhood is fixed and clear. In contrast, the spatial position relationships between each superpixel and its neighborhoods are not fixed; they vary based on each superpixel's location. Let  $C_k, k \in (1, \dots, K)$  denote the cluster center location of the  $k$ th superpixel region  $N_k$ , and let  $N_k^j, j \in (1, \dots, J_k)$  indicate the  $j$ th neighborhood of the superpixel region  $N_k$ . Here,  $J_k$  is the maximum number of neighborhoods of  $N_k$ , and the average intensity of all image pixels belonging to the same superpixel region  $N_k$  is  $g(N_k)$ . Then, the gradient between a superpixel and its neighboring superpixel is defined by the following equation:

$$G_m(N_k^j, N_k) = g(N_k^j) - g(N_k). \quad (5)$$

After the gradient magnitudes  $G_m(N_k^j, N_k)$  of the superpixel region  $N_k^j$  are obtained, all the  $J_k$  gradient magnitudes are sorted in descending order to select the important neighborhoods. To avoid the effects of nonlinear illumination changes, when  $J_k$  is sufficiently large, only the top  $Q = 5, Q \leq J_k$  neighborhood superpixels are selected to estimate the dominant orientation of the superpixel region  $N_k$ . When  $J_k$  is small ( $J_k < 5$ ), the top  $Q$  ( $Q = J_k$ ) nearest superpixels to  $N_k$  are selected to estimate the dominant orientation of the superpixel region  $N_k$ . Here, we define the  $q$ th selected superpixel as  ${}^qN_k, q \in (1, \dots, Q)$ . Then every selected gradient is projected into horizontal and vertical components. The distance between a superpixel and its neighbor is used to determine the ratio of the components. The following formulas are used:

$$G_x(N_k^j, N_k) = \frac{G_m(N_k^j, N_k) \cdot (x_k^j - x_k)}{\sqrt{(x_k^j - x_k)^2 + (y_k^j - y_k)^2}} \quad (6)$$

$$G_y(N_k^j, N_k) = \frac{G_m(N_k^j, N_k) \cdot (y_k^j - y_k)}{\sqrt{(x_k^j - x_k)^2 + (y_k^j - y_k)^2}} \quad (7)$$

where  $(x_k, y_k)$  denotes the cluster center of the superpixel region  $N_k$  and  $(x_k^j, y_k^j)$  is the cluster center of its  $j$ th neighbor superpixel region  $N_k^j$ . Then, as shown in Fig.1(b), the dominant orientation  $-\frac{\pi}{2} \leq \theta(N_k) \leq \frac{\pi}{2}$  of superpixel region  $N_k$  can be derived by the following equation:

$$\theta(N_k) = \arctan \left( \frac{\sum_{q=1}^Q G_y({}^qN_k, N_k)}{\sum_{q=1}^Q G_x({}^qN_k, N_k)} \right) \quad (8)$$

where  $\arctan(\cdot)$  denotes the arc tangent function. Specifically, when  $\sum_{q=1}^Q G_x({}^qN_k, N_k) = 0$ , the dominant orientation  $\theta(N_k)$

is defined as follows:

$$\theta(N_k) = \begin{cases} -\frac{\pi}{2}, & \text{if } \sum_{q=1}^Q G_y({}^qN_k, N_k) < 0 \\ \frac{\pi}{2}, & \text{if } \sum_{q=1}^Q G_y({}^qN_k, N_k) > 0 \\ 0, & \text{if } \sum_{q=1}^Q G_y({}^qN_k, N_k) = 0. \end{cases} \quad (9)$$

### C. Region Binary Descriptor

The dominant orientation  $\theta(N_k)$  of the superpixel region  $N_k$  is not a rotation-invariant feature; it varies with the rotation angle. However, the dominant orientation relationship between each superpixel and its neighborhoods is utilized in this study to construct a rotation-invariant robust feature. Here, as shown in Fig.1(c), the orientation difference  $\Delta^q\theta_k$  between each superpixel  $N_k$  and its neighborhood  ${}^qN_k, q \in (1, \dots, Q)$  can be calculated as follows:

$$\Delta^q\theta_k = |\theta({}^qN_k) - \theta(N_k)| \quad (10)$$

where  $|\cdot|$  is the absolute operation function. The absolute operation, which emphasizes the magnitude of the orientation difference rather than the sign, is not influenced by small orientation difference variations and generates robust descriptors. The result is the orientation difference vector  $\Delta\theta_k = [\Delta^1\theta_k, \dots, \Delta^Q\theta_k]$ , which is not sensitive to rotation changes. To reduce the computational complexity and memory consumption, inspired by the concept of local binary descriptors, the proposed SRBD  $B_k$  of the superpixel  $N_k$  can be constructed as a binary code as follows:

$$B_k = [{}^1B_k, \dots, {}^qB_k, \dots, {}^QB_k] \quad (11)$$

$${}^qB_k = \begin{cases} 1, & \text{if } \Delta^q\theta_k \geq \overline{\Delta\theta_k} \\ 0, & \text{if } \Delta^q\theta_k < \overline{\Delta\theta_k} \end{cases} \quad (12)$$

where  $\overline{\Delta\theta_k}$  is the average value of the orientation difference vector  $\Delta\theta_k$ .

Compared with low-level image features, such as pixel-intensity and pixel-gradient, SRBD utilizes the average intensity of all pixels belonging to the same superpixel region to denote the intensity of the superpixel. This approach makes it robust to noise and blur. The operator in which only the top  $Q$  neighborhoods are selected and used to calculate the superpixel dominant orientation means that SRBD can resist nonlinear illumination changes. The orientation difference vector employed to construct the binary code gives the SRBD feature rotation-invariance. Moreover, both the superpixel dominant orientation and the orientation difference vector are used to represent the relationships between one local region and others. Therefore, when the size of the superpixel region is sufficiently large, SRBD becomes a mid-level semantic image feature, which explains why it is robust even under a low SNR.

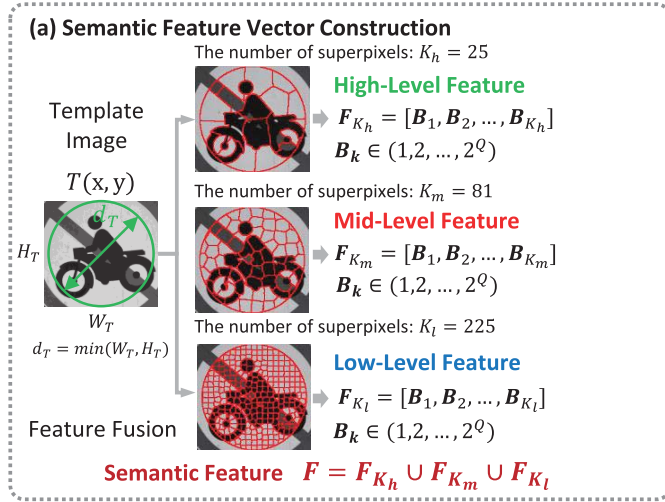


Fig. 2. Semantic feature vector construction. We obtain superpixels of different sizes by changing parameter  $K$  of the KD-SLIC algorithm. Then, the corresponding high-level, mid-level and low-level SRBD sets are calculated and jointed into the final semantic feature.

#### IV. ROBUST SEMANTIC TEMPLATE MATCHING

The proposed RSTM algorithm is introduced in detail in this section. The SRBD is employed to construct the semantic feature vector of a template image, where the number of superpixels  $K$  determines the semantic information level of the image feature. However, it is time-consuming to implement KD-SLIC for the entire scene image or for each candidate window of the scene image; therefore, the proposed RSTM algorithm extracts superpixels only from the template image, during the off-line phase of RSTM. In the on-line phase of RSTM, a Bayesian marginal probability model is established to perform the similarity measurement between the template and each candidate window. To further improve the speed of the matching procedure, an image pyramid strategy is employed. Finally, the computational complexity of RSTM is analyzed and discussed.

##### A. Semantic Feature Vector Construction

After the template image  $T(x, y)$ ,  $x \in (1, \dots, W_T)$ ,  $y \in (1, \dots, H_T)$  is provided, we implement KD-SLIC to extract the superpixels during the off-line RSTM phase, where  $W_T$  and  $H_T$  are the width and height of the template, respectively. To construct the rotation-invariant feature vector, only a circular region of the template image is considered and processed via KD-SLIC. The diameter  $d_T$  of that circular region is determined as the minimum value between  $W_T$  and  $H_T$ , as shown in Fig.2. In general, the number of superpixels  $K$  is important for extracting image semantic features: the larger  $K$  is, the smaller the superpixel region  $N_k$  is.

The KD-SLIC algorithm initializes positions of the cluster centers by dividing the original image into a regular grid to determine the number of superpixels that are expected to be generated. Therefore, the value of  $K$  is generally in the form of the square of  $n$ ,  $n \in (1, 2, \dots)$ . In this study, the number of superpixels  $K_h$  is set to 25 to obtain high-level image features.

To obtain mid-level image features,  $K_m$  is set to be 81. Finally, to obtain low-level image features,  $K_l$  is set to 225. Thus, the total number of superpixels used in RSTM is  $K = K_h + K_m + K_l$ . For features at each level, each superpixel is coded as one  $Q$ -bit binary vector. Then, the binary vector is recorded as the corresponding decimal number  $B_k \in (0, \dots, 2^Q - 1)$ . Finally, the semantic feature vector of the template image is obtained by fusing the features at different levels using the following equations:

$$\begin{aligned}
 F_{K_h} &= [B_1, B_2, \dots, B_{K_h}, \dots, B_{K_h}] \\
 F_{K_m} &= [B_1, B_2, \dots, B_{K_m}, \dots, B_{K_m}] \\
 F_{K_l} &= [B_1, B_2, \dots, B_{K_l}, \dots, B_{K_l}] \\
 F_K &= F_{K_h} \cup F_{K_m} \cup F_{K_l}.
 \end{aligned} \tag{13}$$

Clearly, the SRBD descriptor describes only the local information, such as texture, between each superpixel and its neighbors. However, the semantic feature vector  $F_K$  combines different level image features based on the number of superpixels  $K$ ; thus, the vector can express some image semantic information.

##### B. Similarity Measurement Model

As stated above, implementing KD-SLIC on the entire scene image or for each candidate window in the on-line phase of RSTM is time consuming. Moreover, for a given object, the superpixels extracted from the template image  $T(x, y)$  may sometimes be different than the ones extracted from the candidate window  $I_c(x, y)$  of the scene image  $I(x, y)$ , due to influences from the scene image's background information. Therefore, calculating the similarity between the semantic feature vector  $F_K$  of the template image and the ones of the candidate window directly, using a distance measurement such as Euclidean distance, is unsuitable. Moreover, the proposed RSTM algorithm extracts superpixels from the template image only during the RSTM off-line phase. In this study, a marginal probability distribution model via all superpixel codes  $B_k \in F_K$  is established based on Bayesian theory. This model is then employed to estimate the similarity between the template image and the candidate window of scene image.

During RSTM's off-line phase, we construct a curve function using the superpixel codes to estimate the probable rotation angles. We choose the center point  $(P_x, P_y)$  of the template image as the rotation center. The superpixel boundary  $S_k$  of the superpixel region  $N_k$  is rotated  $\alpha_p$  degrees using the distance between the cluster center  $C_k$  and the rotation center point  $(P_x, P_y)$  as the rotation radius. Here,  $\alpha_p$  is equal to  $p$  degrees,  $p \in (1, \dots, 360)$ . Thus, for each superpixel region  $N_k$ , there are 360 approximate-superpixel regions  $\hat{N}_k^p$ , the boundary of which is  $\hat{S}_k^p$ . Then, for each approximate-superpixel region  $\hat{N}_k^p$ , the corresponding binary code of each region  $B(\hat{N}_k^p)$  can be calculated via SRBD, and the result is transformed into a decimal number. All 360 codes form a relationship curve function  $\phi_k(\alpha_p)$ ,  $\phi_k(\alpha_p) = B(\hat{N}_k^p)$ , where  $\alpha_p$  is the independent variable of  $\phi_k(\alpha_p)$ , as shown in Fig.3. The curve function  $\phi_k(\alpha_p)$  denotes the relationship between the superpixel boundary  $\hat{S}_k^p$  of the approximate-superpixel



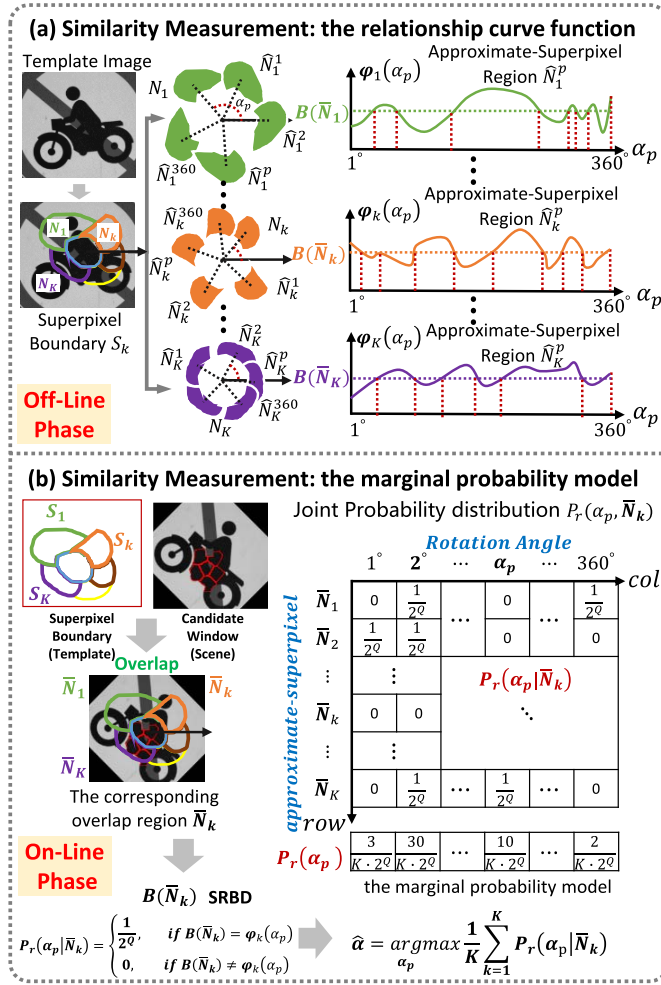


Fig. 3. Similarity measurement model. (a) Construction of the relationship curve function. For every superpixel boundary  $S_k$ , 360 corresponding SRBD codes are calculated and form the relation curve function. (b) Similarity measurement based on the margin probability model. For every sliding position, the similarity can be calculated by the joint probability model  $P_r(\alpha_p | \tilde{N}_k)$ .

region  $\tilde{N}_k^p$  and the reference point  $(P_x, P_y)$  of the template image under different rotation angles  $\alpha_p$ . In other words, if the region binary code  $B(N_k)$  of a superpixel region  $N_k$  is given, the probable rotation angle  $\alpha_p$  can be estimated based on the curve function  $\varphi_k(\alpha_p)$ .  $K$  superpixels are extracted from the template image; therefore, there are also  $K$  curve functions  $\varphi_k(\alpha_p)$ —all calculated during the off-line phase of RSTM. Thus, each curve function indicates the relationship between a superpixel and the reference point under different rotation degrees, as shown in Fig.3. The off-line phase procedure is shown in Algorithm 1.

In the RSTM on-line phase, the scene image  $I(x, y)$  is divided into a set of candidate windows  $I_c(x, y)$ . In a scene image, the target object is similar to the template; thus, their superpixels are similar. Consequently, the superpixel boundary  $S_k$  of the template image is reused for the candidate window and KD-SLIC does not need to be conducted. RSTM covers only each candidate window  $I_c(x, y)$  of the scene image and uses the superpixel boundary  $S_k$  from the template image. Then, the corresponding overlapped region  $\tilde{N}_k$  of the

#### Algorithm 1 Off-Line Phase

**Input:** The template image  $T(x, y)$ .

**Output:**  $S_k$  and curve functions  $\varphi_k(\alpha_p)$ ,  $k \in [1, K]$ ,  $p \in [1, 360]$ .

- 1: Extract  $N_k$  and  $S_k$  by KD-SLIC.
- 2: Set  $(P_x, P_y)$  as the rotation center.
- 3: **for**  $p = 1$  to 360 **do**
- 4:   **for**  $k = 1$  to  $K$  **do**
- 5:     Obtain  $\tilde{N}_k^p$  by rotating  $N_k$  around the rotation center by the rotation angle  $\alpha_p$ .
- 6:   **end for**
- 7:   **for**  $k = 1$  to  $K$  **do**
- 8:     Calculate  $\varphi_k(\alpha_p)$ .
- 9:   **end for**
- 10: **end for**
- 11: Return curve functions  $\varphi_k(\alpha_p)$  and  $S_k$ ,  $k \in [1, K]$ .

candidate window has a region boundary  $\tilde{S}_k$  the same as the superpixel boundary  $S_k$  of the template image. There are  $K$  superpixel boundaries  $S_k$ ; therefore, there are  $K$  corresponding overlapped region boundaries  $\tilde{S}_k$ , and both have the same region topological spatial relationships. Next, for the  $k$ th corresponding overlap region  $\tilde{N}_k$ , its SRBD code  $B(\tilde{N}_k)$  is calculated via SRBD. Finally,  $K$  SRBD codes  $B(\tilde{N}_k)$  are estimated from the candidate window of scene image.

Let  $P_r(\alpha_p, \tilde{N}_k)$  denote the joint probability distribution of the  $k$ th corresponding overlapped region  $\tilde{N}_k$  with respect to the rotation angle  $\alpha_p$  in the candidate window  $I_c(x, y)$ . Then, the marginal probability distribution  $P_r(\alpha_p)$  of the rotation angle of the template image with respect to the candidate window can be derived from the joint probability distribution  $P_r(\alpha_p, \tilde{N}_k)$  as follows:

$$P_r(\alpha_p) = \sum_{k=1}^K P_r(\alpha_p, \tilde{N}_k) = \sum_{k=1}^K P_r(\alpha_p | \tilde{N}_k) \cdot P_r(\tilde{N}_k) \quad (14)$$

where the probability distribution  $P_r(\tilde{N}_k)$  of the  $k$ th corresponding overlap region  $\tilde{N}_k$  is assumed to be evenly distributed. Then, the marginal probability distribution  $P_r(\alpha_p)$  of the rotation angle can be derived from Eq.14 as follows:

$$P_r(\alpha_p) = \frac{1}{K} \sum_{k=1}^K P_r(\alpha_p | \tilde{N}_k) \quad (15)$$

where  $P_r(\alpha_p | \tilde{N}_k)$  denotes the conditional probability of the  $k$ th corresponding overlap region  $\tilde{N}_k$  rotated by  $\alpha_p$  degrees,  $\alpha_p \in (1^\circ, 2^\circ, \dots, 360^\circ)$ , and it can be predicted based on the curve function  $\varphi_k(\alpha_p)$ , as shown below:

$$P_r(\alpha_p | \tilde{N}_k) = \begin{cases} \frac{1}{2^\circ}, & \text{if } B(\tilde{N}_k) = \varphi_k(\alpha_p), \\ 0, & \text{if } B(\tilde{N}_k) \neq \varphi_k(\alpha_p). \end{cases} \quad (16)$$

The similarity measurement function  $F(T, I_c)$  between the template image  $T(x, y)$  and the candidate window  $I_c(x, y)$  of the scene image  $I(x, y)$  can be considered to be the maximum probability  $P_r(\hat{\alpha}_p)$ , and is derived from both Eq.15 and Eq.16

**Algorithm 2** On-Line Phase

An example of estimating the similarity between the template image and a candidate window.

**Input:**

Curve functions  $\varphi_k(\alpha_p)$  and  $S_k$ ,  $k \in [1, K]$ ,  $p \in [1, 360]$ ,  
a candidate window  $I_c(x, y)$ .

**Output:**

The similarity  $P_r(\hat{\alpha}_p)$  between  $T(x, y)$  and  $I_c(x, y)$ ,  
and the predicted rotation angle  $\hat{\alpha}_p$ .

```

1: Initialize all  $P_r(\alpha_p) = 0$ ,  $p \in [1, 360]$ .
2: for  $k = 1$  to  $K$  do
3:   Obtain  $\bar{N}_k$  by using  $S_k$ .
4:   Calculate  $B(\bar{N}_k)$ .
5:   for  $p = 1$  to  $360$  do
6:     if  $B(\bar{N}_k) == \varphi_k(\alpha_p)$  then
7:        $P_r(\alpha_p) = P_r(\alpha_p) + \frac{1}{2Q}$ .
8:     end if
9:   end for
10: end for
11:  $P_r(\hat{\alpha}_p) = 0$ ,  $\hat{\alpha}_p = 0$ .
12: for  $p = 1$  to  $360$  do
13:   if  $P_r(\alpha_p) > P_r(\hat{\alpha}_p)$  then
14:      $P_r(\hat{\alpha}_p) = P_r(\alpha_p)$ .
15:      $\hat{\alpha}_p = \alpha_p$ .
16:   end if
17: end for
18:  $P_r(\hat{\alpha}_p) \leftarrow \frac{P_r(\hat{\alpha}_p)}{K}$ .
19: return  $P_r(\hat{\alpha}_p)$  and  $\hat{\alpha}_p$ .
```

as follows:

$$F(T, I_c) = P_r(\hat{\alpha}_p) = \frac{1}{K} \sum_{k=1}^K P_r(\hat{\alpha}_p | \bar{N}_k) \quad (17)$$

$$\hat{\alpha}_p = \arg \max_{\alpha_p} \frac{1}{K} \sum_{k=1}^K P_r(\alpha_p | \bar{N}_k). \quad (18)$$

The similarity function  $F(T, I_c)$  utilizes the total  $K$  corresponding overlap regions to estimate the template probability, and it has already considered the rotation case in which  $\hat{\alpha}_p$  is the predicted rotation angle. Therefore, the similarity function has robust rotation invariance due to its statistical property. In the RSTM implementation, to rapidly calculate the similarity, a voting table can be established to count the conditional probability  $P_r(\alpha_p | \bar{N}_k)$  and the marginal probability  $P_r(\alpha_p)$ . The procedure is shown in Algorithm 2, which estimates the similarity between a template image and a candidate window from the scene image.

In addition, to address object scale changes, RSTM utilizes different sliding window sizes corresponding to different scales, in which the corresponding overlap region  $\bar{N}_k$  is scaled by the corresponding scale value. All the similarity scores calculated at different scales are compared, and the scale value with the highest similarity score is adopted as the template scale used in the scene image.

In general, the sliding window method is utilized to divide the scene image  $I(x, y)$  into numerous candidate windows

$I_c(x, y)$ . These windows slide from left to right and top to bottom in a pixel-by-pixel manner. The similarity measurement function is applied to each candidate window, and the best-matched region is selected as the high score of the similarity function. In this study, the image pyramid is employed to speed up the proposed RSTM algorithm. Both the scene image and the template image are downsampled via a Gaussian pyramid.

**C. Computational Complexity Analysis**

In this subsection, our method's computational complexity is analyzed. For the convenience of the analysis, we consider that the scale value is 1 without applying the image pyramid. The width and height of the template image are  $W_T$  and  $H_T$ , respectively, while the width and height of the scene image are  $W_S$  and  $H_S$ , respectively. The proposed approach also adopts the sliding window search strategy; hence, there are  $(W_S - W_T + 1)(H_S - H_T + 1)$  window searches. During similarity measurement, there are  $360 \cdot K$  probability calculations for each similarity calculation, where  $K$  is the number of superpixels. Thus, the computational complexity of the proposed RSTM algorithm is  $O(360 \cdot K \cdot (W_S - W_T + 1) \cdot (H_S - H_T + 1))$ .

**V. EXPERIMENTAL RESULTS**

The robustness of the proposed RSTM is evaluated using the MS COCO dataset [45] and RSTM is compared with four classic algorithms (SBM, M-GHT, NCC, and SURF) and three recent algorithms (SITM-HDG, MTM, and BBS). To more fully analyze the performances of these methods, rotation change, scale change, noise, occlusion, blur, nonlinear illumination change and deformation are simultaneously considered in our experiments. In addition, because the number of superpixels  $K$  in the template image is the main influencing factor on RSTM's performance, we analyze and discuss the parameter  $K$ .

**A. Performance Analysis of the Parameter  $K$** 

The SRDB parameter  $K$  denotes the number of superpixels that should be extracted from the template image. In general, the larger  $K$  is, the smaller the size of each superpixel is and the less semantic information SRDB will contain. Thus, the performance of RSTM is affected by the parameter  $K$ . The effect of  $K$  is evaluated and discussed by conducting the following experiment.

Fig.4(a) shows the original image. The template image  $T(x, y)$ , with  $89 \times 82$  pixels is cropped from the original. Fig.4(b) shows the scene image  $I(x, y)$  with  $461 \times 346$  pixels, which is degraded from the original image by the addition of Gaussian white noise with a mean of 0 and a variance of 0.1. To investigate the relationship between the RSTM runtime and parameter  $K$ , the number of superpixels  $K$  in the template image is varied from 25 to 961 with a step size of  $8 \times n$ ,  $n \in (3, 4, \dots, 15)$ . We define the relative time  $t_K^r$  as follows:

$$t_K^r = \frac{t_K}{t_{25}} \quad (19)$$



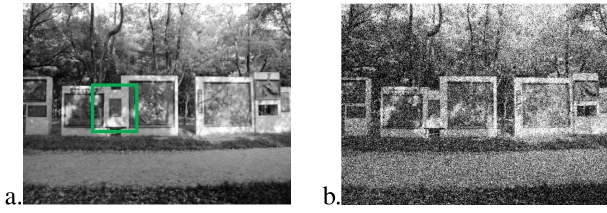


Fig. 4. The scene images used for the parameter experiment: (a) the template image (denoted by the green box); (b) the scene image with noise.

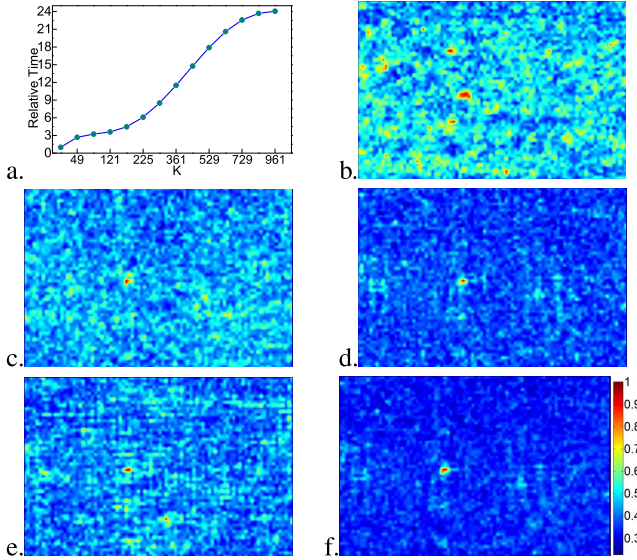


Fig. 5. The experimental results of varying parameter  $K$ : (a) the relative time and  $K$ ; (b) the probability map of  $K_h = 25$ ; (c) the probability map of  $K_m = 81$ ; (d) the probability map of  $K_l = 225$ ; (e) the probability map of  $K_{ll} = 361$ ; and (f) the probability map of the fusion feature ( $K = K_d + K_m + K_l$ ).

where  $t_K$  denotes the RSTM runtime with  $K$  superpixels, and  $t_{25}$  denotes the RSTM runtime when  $K = 25$ .

Fig.5(a) shows the relationship curve between the relative time  $t_K^r$  of RSTM and the value of parameter  $K$ . It can be observed that larger  $K$  values result in longer relative time  $t_K^r$ . When the  $K$  is larger than 225,  $t_K^r$  increases dramatically. Fig.5(b)–(d) shows the probability maps (the similarity measurement function) for  $K_h = 25$  (the high-level image feature),  $K_m = 81$  (the mid-level image feature) and  $K_l = 225$  (the low-level image feature), respectively. It shows that the probability maps of RSTM using both  $K_h = 25$  and  $K_m = 81$  are cluttered with no prominent peak. Consequently, it is very difficult to find the template from the scene image because the proposed RSTM is based on a Bayesian statistical model that requires sufficient data. The situation when  $K_l = 225$  is much better than that when  $K_h = 25$  or  $K_m = 81$ , although considerable clutter still exists in the probability map. Fig.5(e) shows the results when  $K_{ll} = 361$ . This result indicates that even when the number  $K$  is sufficiently large, the peak of the probability map is not prominent and is present because the size of the superpixel region is very small and, thus, sensitive to noise.

Therefore, in this study, we constructed a fused semantic feature  $K = 25 + 81 + 225$  by combining the high-level

semantic feature with the mid- and low-level semantic features. The probability map for this fused semantic feature is shown in Fig.5(f). It can be observed that the fusion semantic feature not only suppresses the influence of noise effectively and highlights the probability peak, but also has higher time efficiency than the situation when  $K_{ll} = 361$ , because  $K = 25 + 81 + 225 < K_{ll}$ .

### B. Robustness

To more fully analyze the robustness of RSTM, this experiment investigates various types of challenges to the template image, namely, rotation changes, scale changes, noise, occlusions, blur, nonlinear illumination changes and deformation. For comparison purposes, we also include four popular classic algorithms (SBM, M-GHT, NCC and SURF) and three recent algorithms (SITM-HDG, MTM and BBS). Here, the evaluation dataset is built by randomly selecting 100 images from the MS COCO dataset as original images. Then, to implement the challenges, each original image is transformed into 30 images, each by a different transformation or degradation method. A total of 7 template-matching challenges are considered in this study; therefore, the evaluation dataset consists of 21,000 images. Fig.6 shows an example image “truck” and the corresponding images containing the various challenges.

To quantitatively analyze the robustness of the compared methods, the intersection-over-union (*IoU*) metric is adopted, which is calculated as follows:

$$IoU = \frac{area(R_c \cap R_g)}{area(R_c \cup R_g)} \quad (20)$$

where  $area(\cdot)$  denotes the pixel area of  $(\cdot)$ ,  $R_c$  is the candidate bounds and  $R_g$  indicates the ground truth bounds. *IoU* ranges from 0 to 1; the more robust a method is, the higher its *IoU* score is.

1) *Rotation Change*: In this test, each original image is rotated from  $0^\circ$  to  $360^\circ$  degrees in steps of  $12^\circ$  degrees. Thus, the rotation subdataset includes 3,000 rotated images. These dataset images are divided into three categories based on rotation angle: *RotationLevel1* ranges from  $0^\circ$  degree to  $108^\circ$  degrees; *RotationLevel2* ranges from  $120^\circ$  to  $228^\circ$  degrees; and *RotationLevel3* ranges from  $240^\circ$  degree to  $348^\circ$  degrees. Note that SBM, NCC and MTM are not rotation-invariant methods; therefore, differently rotated template images—from  $0^\circ$  degree to  $360^\circ$  degrees at a step size of  $1^\circ$  degree—are constructed to allow these methods to adapt to rotation changes.

Fig.7 (a) shows the performances of the tested methods on the rotation change test. The average *IoU* values of RSTM, SBM, M-GHT, NCC, MTM and SURF are larger than 0.99 for all three levels, but the average *IoU* values of SITM-HDG and BBS reach no higher than 0.7, which is unsatisfactory in practical applications.

The proposed RSTM directly estimates an accurate rotation angle for the template using the probability distribution of the superpixels. SBM, NCC, and MTM can indirectly address rotation change by rotating the original template image into different rotated template images. M-GHT constructs a separate R-table for every possible orientation to address rotation



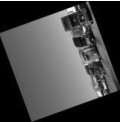
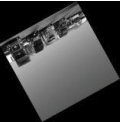













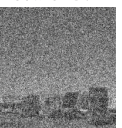




















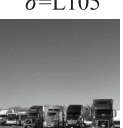

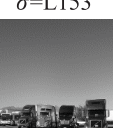

Type	level1		level2		Level3	
1 Rotation	 angle=0	 angle=108	 angle=120	 angle=228	 angle=240	 angle=348
2 Scale	 scale=0.5	 scale=1.4	 scale=1.5	 scale=2.4	 scale=2.5	 scale=3.4
3 Noise	 density=0.02	 density=0.2	 density=0.22	 density=0.4	 density=0.42	 density=0.6
4 Occlusion	 length=2	 length=20	 length=22	 length=40	 length=42	 length=60
5 Blur	 r=0.4	 r=4	 r=4.4	 r=8	 r=8.4	 r=12
6 Nonlinear illumination	 $\sigma=D104$	 $\sigma=D65$	 $\sigma=L105$	 $\sigma=L147$	 $\sigma=L153$	 $\sigma=L195$
7 Deformation	 angle=1	 angle=10	 angle=11	 angle=20	 angle=21	 angle=30

Fig. 6. The test image “truck” and its challenge images: In the 1st row, “angle” is the rotation angle of the challenge image, and the green box is the template. In the 2nd row, “scale” is the scale value of the interference image. In the 3rd row, “density” is the noise density of the salt and pepper noise. In the 4th row, “length” is the side length of the square occlusion area. In the 5th row, “ $r$ ” is the radius of the disk defocussing model for defocus blur. In the 6th row, “ $\sigma$ ” is the sigma of the Gaussian kernel used to simulate nonlinear illumination. In the 7th row, “angle” is the rotation angle of the rotating deformation model.

change, while SITM-HDG uses the pixel gradient orientation to estimate the similarity between the template image and a candidate region. As we know, the gradient orientation varies with the rotation angle, which causes SITM-HDG to be sensitive to rotation change. In addition, the experimental results show that the popular SURF is a rotation-invariant method, due to its use of rotation-invariant feature points. However, BBS utilizes image patches selected from the template image and scene image to calculate the similarity, and these image patches are sensitive to rotation change; therefore, BBS cannot address rotation change very well. Therefore, among the tested methods, only RSTM, SBM, NCC, MTM and SURF are robust against rotation change.

2) *Scale Change*: In this test, each original image is zoomed at scales ranging from 0.5–3.4 in steps of 0.1. Thus, the scale change subdataset test also includes 3,000 scaled images divided into three categories based on scale size: *ScaleLevel1*

ranges from 0.5 to 1.4; *ScaleLevel2* ranges from 1.5 to 2.4; and *ScaleLevel3* ranges from 2.5 to 3.4. Note that SBM, NCC and MTM are not scale-invariant methods; therefore, differently scaled template images are constructed to help these methods adapt to the scale change.

The average *IoU* scores of those methods for the three scale levels are shown in Fig.7 (b). It can be observed that the average *IoU* of RSTM, SBM, NCC, MTM, SITM-HDG and SURF are greater than 0.98 for all scale levels, which means that these methods all perform well for scale changes. In contrast, both M-GHT and BBS are sensitive to scale changes; their average *IoU* scores decrease as the scale level increases from *ScaleLevel1* to *ScaleLevel3*.

Here, RSTM, SBM, NCC and MTM utilized differently sized template images to assess similarity at different scale sizes; therefore, they can address scale change. In general, SURF is a scale-invariant method, because the feature points

used in SURF are extracted in the scale space and are scale-invariant. Similar to the rotation change results, M-GHT is not able to identify the template location in the scene image at scale sizes other than 1. The patches of BBS are not rotation-invariant, which means it is unable to handle scale changes. Thus, among the tested methods, only RSTM, SBM, NCC, MTM and SURF are robust to scale changes, although SITM-HDG is robust against large-scale changes.

3) *Noise*: In this test, the salt/pepper noise is considered to evaluate the robustness of the tested methods. Each original image is degraded by adding different salt/pepper noise levels. The salt/pepper noise images are generated at different noise densities  $d_{sp}$  ranging from 0.02 to 0.6 with a step size of 0.02. The noise density affects approximately  $d_{sp} \cdot \text{Num}(\mathbf{I})$  pixels, where  $\text{Num}(\mathbf{I})$  is the number of pixels in the image  $\mathbf{I}$ . Here, *Salt/pepperLevel1* denotes a noise density between 0.02 and 0.2; *Salt/pepperLevel2* denotes a noise density between 0.22 and 0.4; and *Salt/pepperLevel3* denotes a noise density between 0.42 and 0.6. Consequently, the salt/pepper noise subdataset consists of 3,000 images.

Fig.7 (c) shows the performances of the tested methods when addressing salt-and-pepper noise. At all three noise levels, the average *IoU* scores of RSTM, SBM, NCC and MTM exceed 0.96, which indicates they are robust against noise. It can be observed that the average *IoU* scores of M-GHT, SITM-HDG, SURF and BBS decrease as the noise level increases. The average *IoU* score of M-GHT decreases from 0.924 to 0.271, the scores of SITM-HDG decrease from 0.834 to 0.215, the scores of SURF decrease from 0.867 to 0.139, and the scores of BBS decrease from 0.877 to 0.282.

RSTM utilizes the average intensity of a superpixel region instead of the single-pixel intensity; thus, it performs very well when subject to the noise. NCC and MTM both use pixel-intensity information to calculate the cross-correlation similarity; thus, they are also able to address the noise. SBM utilizes the difference in the gradient orientation of the template image pixel and the scene image pixel to calculate the similarity function. Even under the influence of noise, the gradient orientation difference changes slightly; therefore, SBM is also robust against the noise. M-GHT, SITM-HDG and SURF utilize pixel-gradient orientation to estimate similarity. Gradient orientation is robust to slight noise levels but cannot address strong noise levels, because the gradient calculation is influenced by noise. In addition, these methods employ SSD, which cannot reduce the noise influence, to calculate the similarity between two patches in the BBS. Thus, only four of the tested methods, RSTM, SBM, NCC and MTM, are robust against noise.

4) *Occlusion*: In this test, each original image is covered by another object with different side lengths of a square occlusion area, ranging from 2 pixels to 60 pixels with a step size of 2. The subdataset for the occlusion test consists of 3,000 scaled images divided into three levels: *OcclusionLevel1*, from 2 pixels to 20 pixels; *OcclusionLevel2*, from 22 pixels to 40 pixels; and *OcclusionLevel3*, from 42 pixels to 60 pixels.

Fig.7 (d) shows the average *IoU* scores of the tested methods on the occluded images. The results show that RSTM, SBM, M-GHT, SURF and BBS are good at addressing

occlusion, followed by SITM-HDG. The average *IoU* scores of RSTM, SBM, M-GHT, SURF and BBS exceed 0.9, and the scores of SITM-HDG are above 0.7 at all three levels. However, as the occlusion level changes, the average *IoU* scores of NCC decrease from 0.978 to 0.368, while the MTM scores decrease from 1 to 0.62619. This result means that both NCC and MTM are unsatisfactory at addressing occlusion, and it occurs because they both utilize global intensity information to calculate the cross-correlation value as the similarity value. Thus, when occlusion is present, the similarity value is seriously affected.

However, RSTM, SBM and M-GHT employ a local similarity accumulation strategy that estimates the ratio between the number of similar local features and the total number of local features extracted from the template image. Therefore, they can all address occlusion easily. Despite the presence of occlusion, when sufficient feature points can be extracted, SURF also performs well. BBS and SITM-HDG also use nonoccluded regions to construct corresponding patches or features and they also use local features when determining similarity; therefore, they adapt well to occlusion. In general, RSTM, SBM, M-GHT, SURF and BBS are robust against occlusion, while NCC and MTM cannot cope well with occlusion.

5) *Blur*: The blur test uses the defocus blur to evaluate the robustness of the compared methods. Each original image is blurred and degraded into 30 images. The subdataset for the defocus blur test includes three levels of blur based on the radius of the disk defocus model: *DefocusLevel1* ranges from 0.4 to 4 pixels; *DefocusLevel2* ranges from 4.4 to 8 pixels; and *DefocusLevel3* ranges from 8.4 to 12 pixels. The step size is 0.4 pixels. The resulting defocus blur subdataset includes 3,000 images.

Fig.7 (e) shows the average *IoU* scores of these methods for defocus blur. The average *IoU* scores of RSTM, NCC and MTM are slightly higher than 0.94 and are followed by SBM and M-GHT, whose average *IoU* scores are never less than 0.7. However, as the blur level increases, the average *IoU* scores of SITM-HDG, SURF and BBS decrease dramatically. The average *IoU* score of SITM-HDG decreases from 0.909 to 0.349. SURF's average *IoU* scores also dramatically decrease from 0.981 to 0.362. Finally, BBS's average *IoU* scores also dramatically decrease from 0.957 to 0.257.

By taking advantage of the blur-invariant semantic feature, RSTM easily adapts to the blur. NCC and MTM both rely on global intensity information, which is slightly influenced by blur; therefore, they are robust against blur. However, because pixel gradient orientation is seriously influenced by blur and causes the corresponding relationships to be messy, SBM and M-GHT always skew the accurate location of template image, particularly at higher blur levels. BBS uses the SSD method to find the corresponding image patches, and the SSD is affected by blur; therefore, BBS cannot address blur effectively.

In addition, the gradient magnitude decreases as the blur level increases, but there are insufficient stable gradient points with large gradient magnitudes in SITM-HDG; thus, it is also unsuitable for blur, especially at higher blur levels. Because extracting feature points is impossible under high blur levels, SURF cannot address blur effectively. In general, RSTM,



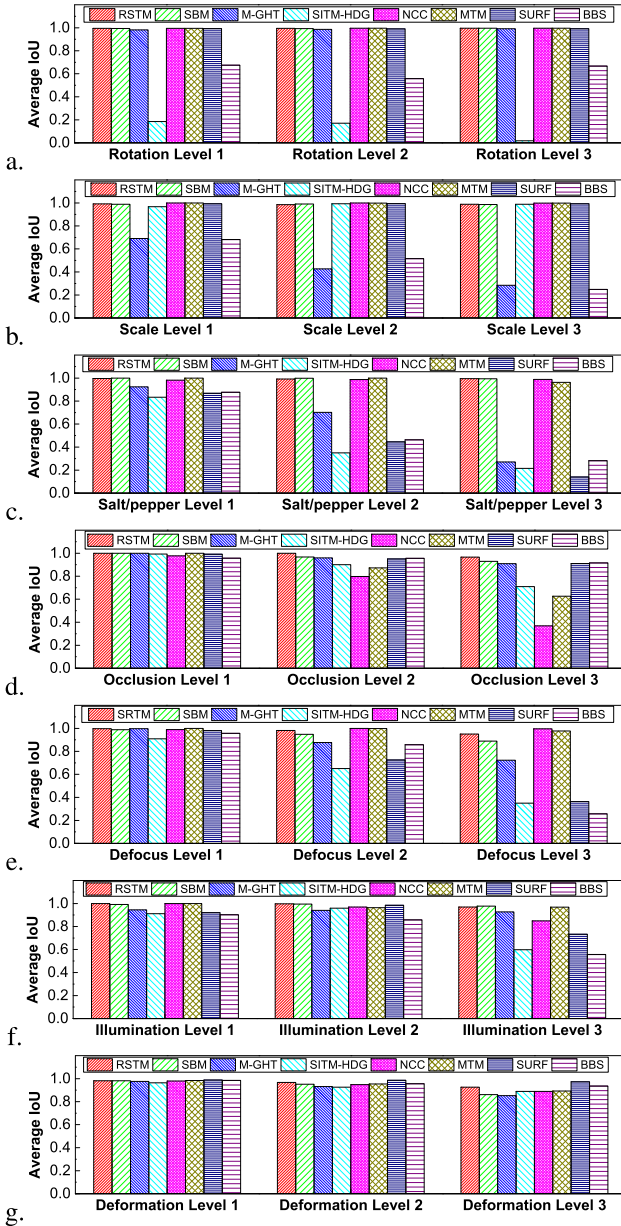


Fig. 7. The average *IoU* in different robustness tests. (a) Rotation test; (b) scale test; (c) noise test; (d) occlusion test; (e) blur test; (f) nonlinear illumination test; (g) deformation test.

NCC and MTM are the most robust against blur, followed by M-GHT and SBM, while SITM-HDG, SURF and BBS are easily affected by blur.

6) *Nonlinear Illumination*: In this test, a Gaussian kernel with different sigma values is applied to each original image to simulate non-linear illumination. The subdataset for nonlinear illumination test consists of 3,000 images divided into three levels. *IlluminationLevel1* is “dark illumination” using a dark intensity signal with a sigma varying the range from 104 to 65, with a step size of  $-3$ . *IlluminationLevel2* and *IlluminationLevel3* are “bright illumination.” The sigma of *IlluminationLevel2* ranges from 105 to 147, and the sigma of *IlluminationLevel3* ranges from 153 to 195, both with a step size of 6.

The performances of the methods under these three nonlinear illumination levels are shown in Fig.7 (f). RSTM, SBM,

M-GHT and MTM achieve the best performances, averaging *IoU* values greater than 0.92. The average *IoU* score of NCC decreases from 1 to 0.849, varying with the illumination level. The average *IoU* score of SITM-HDG decreases from 0.911 to 0.598 with the illumination level. The lowest average *IoU* scores of SURF and BBS are 0.733 and 0.557, respectively.

The SRDB feature of RSTM is based on the dominant superpixel orientation, which suppresses the influence of the nonlinear illumination changes. SBM and M-GHT are based on pixel gradient orientation, which is also robust to nonlinear illumination. NCC is influenced by the nonlinear illumination because the normalized correlation operator can just detect the influence of linear illumination. MTM can be considered as a variant of NCC that it is more robust to nonlinear illumination due to its nonlinear tone mappings. High illumination levels influence the feature point extraction method used in SIFT. BBS utilizes local patches to calculate similarity. Thus, when nonlinear illumination is present in the scene image, the patches change. Therefore, BBS cannot find the corresponding patches using SSD. SITM-HDG relies on pixel gradient magnitude to build the dominant gradient, but the gradient magnitude is unstable under nonlinear illumination. In general, only RSTM, SBM, M-GHT and MTM perform well under nonlinear illumination.

7) *Deformation*: In this test, each original image is deformed by a rotating deformation model whose rotation angle varies from  $1^\circ$  to  $30^\circ$  degrees. The subdataset for this test contains 3,000 images divided into three levels: *DeformationLevel1* ranges from  $1^\circ$  to  $10^\circ$  degrees, *DeformationLevel2* ranges from  $11^\circ$  to  $20^\circ$  degrees, and *DeformationLevel3* ranges from  $21^\circ$  to  $30^\circ$  degrees. The step size is  $1^\circ$ .

Fig.7(g) shows the average *IoU* scores of these methods under deformation. From *Level1* to *Level3*, the average *IoU* scores of SURF are 0.988, 0.985 and 0.973, respectively; those of RSTM are 0.981, 0.967 and 0.925, respectively, and those of BBS are 0.985, 0.956 and 0.936, respectively. Analogously, the average *IoU* scores of SBM, M-GHT, SITM-HDG, NCC and MTM decrease as the deformation level increases; their average *IoU* scores at *Level3* are 0.861, 0.853, 0.889, 0.889 and 0.892, respectively.

SURF is based on local feature points, which are barely influenced by deformation. BBS is based on local patches, which are also barely influenced by the deformation. RSTM utilizes the image semantic information, which is robust against deformation to a certain extent. The other methods can cope with slight deformations but are unable to resist severe deformations. In general, SURF has the best performance under deformation, followed by BBS and RSTM.

8) *Discussion*: The average *IoU* scores of the methods across the entire experiment to test robustness are shown in Fig.8. The average *IoU* scores of RSTM, SBM, M-GHT, SITM-HDG, NCC, MTM, SURF and BBS are 0.984, 0.965, 0.852, 0.677, 0.913, 0.925, 0.717 and 0.648, respectively. In this study, a lowest average *IoU* score for a method above 0.90 means the method is suitable for practical application and can be considered as robust. RSTM achieved the best robustness under all the types of challenges, followed

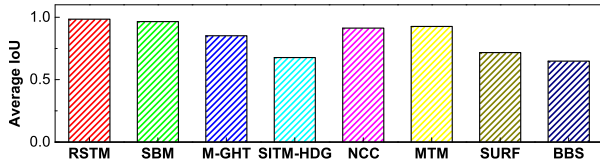


Fig. 8. The average *IoU* scores of RSTM, SBM, M-GHT, SITM-HDG, NCC, MTM, SURF and BBS.

TABLE I  
THE ROBUSTNESS OF RSTM, SBM, M-GHT, SITM-HDG, NCC, MTM, SURF AND BBS

Method	RSTM	SBM	M-GHT	SITM-HDG	NCC	MTM	SURF	BBS
Rotation	✓	×	✓	×	×	×	✓	×
Scale	✓	×	×	✓	×	×	✓	×
noise	✓	✓	×	×	✓	✓	×	×
Occlusion	✓	✓	✓	×	×	×	✓	✓
blur	✓	×	×	×	✓	✓	×	×
Illumination	✓	✓	✓	×	×	✓	×	×
Deformation	✓	×	×	×	×	×	✓	✓

TABLE II  
THE AVERAGE RUNTIMES OF RSTM, SBM, M-GHT, SITM-HDG, NCC, MTM, SURF AND BBS DURING THE ROBUSTNESS EXPERIMENTS

Method	RSTM	SBM	M-GHT	SITM-HDG
time(ms)	25.3652	7.8183	5.6910	18.8470
Method	NCC	MTM	SURF	BBS
time(ms)	43.2588	44.6467	1.9296	128.0228

by SBM. NCC and MTM have similar anti-interference ability and performed better than did M-GHT, SITM-HDG and BBS.

In addition, to fully evaluate the robustness of the eight compared methods, the relationships between the challenges and the methods are shown in Table I. However, SBM, NCC and MTM are able to cope with rotation and scale changes only when we construct different rotating template images and scaling template images for them; thus, these methods are not robust to rotation and scaling, even though their average *IoU* scores are high. Table I reveals that RSTM has the best overall robustness compared with the other methods.

### C. Runtime

In general, template matching is widely used in automation, for instance, for robot pick-and-place operations. To meet the needs of these real-time applications, the computation complexity of template matching should be sufficiently low. We recorded and analyzed the runtimes of the tested methods during the previous robustness experiments. All the methods were implemented on a platform consisting of an Intel 3.30-GHz Core i5 CPU with 4 GB of RAM and a Windows 10 64-bit OS.

The average runtime for the tested methods are shown in Table II. RSTM is faster than NCC, MTM and BBS; however, it is slower than SBM, M-GHT and SITM-HDG. Nevertheless, RSTM's time consumption is acceptable for most real-time applications, and it is highly robust to all the various challenges.

## VI. CONCLUSIONS

In this paper, we presented a robust semantic template-matching method (RSTM) that uses a novel superpixel region binary descriptor (SRBD). A kernel-distance based simple linear iterative clustering method is developed and employed to extract stable superpixels. Then, to address all types of robustness challenges, a dominant orientation difference vector is constructed based on the differences between the current superpixel and its neighborhoods. The constructed vector is coded as the SRBD. A fused feature combining the high-, mid- and low-level semantic features is used for template matching. To cope with rotation invariance, the marginal probability model is proposed and applied to locate the positions of the template image in scene image, and to minimize the runtime, the image pyramid is used.

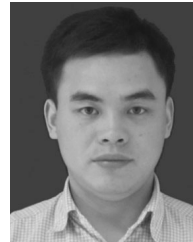
To evaluate its robustness, RSTM is compared with both the most popular and the most recent methods. A series of experiments are carried out on a large dataset randomly selected from the MS COCO dataset. The experimental results show that RSTM simultaneously addresses rotation changes, scale changes, noise, occlusions, blur, nonlinear illumination changes and deformation while maintaining high efficiency. From the results, we can conclude that the proposed RSTM outperforms the state-of-the-art template-matching methods.

## REFERENCES

- [1] M. A. Treiber, *An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications*. Springer, 2010, pp. 1–10.
- [2] K. Grauman and B. Leibe, “Visual object recognition,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*. San Rafael, CA, USA: Morgan & Claypool, 2011, p. 577.
- [3] B. Zhang, H. Yang, and Z. Yin, “A region-based normalized cross correlation algorithm for the vision-based positioning of elongated IC chips,” *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 3, pp. 345–352, Aug. 2015.
- [4] Y. Wang, Y. Y. Tang, and L. Li, “Correntropy matching pursuit with application to robust digit and face recognition,” *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1354–1366, Jun. 2017.
- [5] J. Wang and Y. Yagi, “Many-to-many superpixel matching for robust tracking,” *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1237–1248, Jul. 2014.
- [6] A. Hill, C. J. Taylor, and T. F. Coates, “Object recognition by flexible template matching using genetic algorithms,” in *Proc. Eur. Conf. Comput. Vis.*, vol. 588, no. 12, 1992, pp. 852–856.
- [7] J. Yoo, S. S. Hwang, S. D. Kim, M. S. Ki, and J. Cha, “Scale-invariant template matching using histogram of dominant gradients,” *Pattern Recognit.*, vol. 47, no. 9, pp. 3006–3018, 2014.
- [8] M. J. Atallah, “Faster image template matching in the sum of the absolute value of differences measure,” *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 659–663, Apr. 2001.
- [9] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.
- [10] D. I. Barnea and H. F. Silverman, “A class of algorithms for fast digital image registration,” *IEEE Trans. Comput.*, vol. C-21, no. 2, pp. 179–186, Feb. 1972.
- [11] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105–107, Jan. 1995.
- [12] X. Q. Gao, C. J. Duanmu, and C. R. Zou, “A multilevel successive elimination algorithm for block matching motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 501–504, Mar. 2000.
- [13] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” *Pattern Recognit.*, vol. 26, no. 1, pp. 167–174, 1993.
- [14] J. P. Lewis, “Fast template matching,” *Pattern Recognit.*, vol. 10, no. 11, pp. 120–123, 1995.
- [15] Y. Hel-Or, H. Hel-Or, and E. David, “Matching by tone mapping: Photometric invariant template matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 317–330, Feb. 2014.

- [16] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.
- [17] M. Ulrich, C. Steger, and A. Baumgartner, "Real-time object recognition using a modified generalized Hough transform," *Pattern Recognit.*, vol. 36, no. 11, pp. 2557–2570, 2003.
- [18] H. Yang, S. Zheng, J. Lu, and Z. Yin, "Polygon-invariant generalized Hough transform for high-speed vision-based positioning," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1367–1384, Jul. 2016.
- [19] C. Steger, "Occlusion, clutter, and illumination invariant object recognition," *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.*, vol. 34, no. 3/A, pp. 345–350, 2002.
- [20] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
- [21] S. Oron, T. Dekel, T. Xue, W. T. Freeman, and S. Avidan, "Best-buddies similarity—Robust template matching using mutual nearest neighbors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1799–1813, Aug. 2017.
- [22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [23] F. Alharwarin, D. Ristić-Durrant, and A. Gräser, "VF-SIFT: Very fast SIFT feature matching," in *Proc. Joint Pattern Recognit. Symp.* Berlin, Germany: Springer, 2010, pp. 222–231.
- [24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
- [25] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct. 2003, pp. 10–17.
- [26] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, Aug. 2004.
- [27] L. Wang, H. Lu, and M.-H. Yang, "Constrained superpixel tracking," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1030–1041, Mar. 2018.
- [28] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 30, no. 2, Sep./Oct. 2009, pp. 670–677.
- [29] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 49–65, 2007.
- [30] G. Mori, "Guiding model search using segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2005, pp. 1417–1423.
- [31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [32] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [33] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [34] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [35] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.
- [36] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [37] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 705–718.
- [38] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, 2009.
- [39] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proc. Int. Conf. Comput. Vis.*, vol. 23, no. 5, 2011, pp. 603–610.
- [40] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, vol. 63, no. 14, 2010, pp. 778–792.
- [41] L. Zhuo, Z. Geng, J. Zhang, and X. G. Li, "ORB feature based Web pornographic image recognition," *Neurocomputing*, vol. 173, no. P3, pp. 511–517, 2016.
- [42] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 58, no. 11, Nov. 2012, pp. 2548–2555.

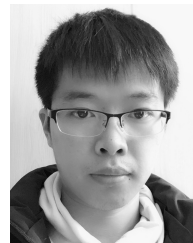
- [43] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina key-point," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 157, no. 10, Jun. 2012, pp. 510–517.
- [44] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
- [45] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.



**Hua Yang** received the B.S. and M.S. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2008, respectively, and the Ph.D. degree from Hiroshima University, Higashihiroshima, Japan, in 2011.

He has been with Hiroshima University as a Research Associate from 2011 to 2012 and as an Assistant Professor since 2012. He is currently an Associate Professor with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. His current research

interests include high-speed vision and its applications (object recognition, detection and tracking, particle image velocity, and dynamic-based vision inspection).



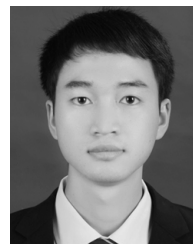
**Chenghui Huang** received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2015, where he is currently pursuing the M.S. degree with the State Key Laboratory of Digital Manufacturing Equipment and Technology.

His current research interests include object recognition and high-speed vision.



**Feiyue Wang** received the B.S. degree in mechanical design, manufacturing and automation from Wuhan University, Wuhan, China, in 2017. He is currently pursuing the M.S. degree with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan.

His current research interests include image match and high-speed vision.



**Kaiyou Song** received the B.S. degree in mechanical design, manufacturing and automation from Wuhan University, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan.

His current research interests include image segmentation, object detection, and deep learning.



**Zhoupeng Yin** received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1994, 1996, and 2000, respectively.

He has published two monographs, three chapters in English books, and more than 30 papers in international journals, such as the *IEEE TRANSACTIONS*, the *ASME Transactions*, and *Computer-Aided Design*. His current research interests include machine vision, flexible electronics, and electronic packaging.