Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 6937852, 12 pages
http://dx.doi.org/10.1155/2016/6937852

Research Article

Dual-Layer Density Estimation for Multiple Object Instance Detection

Qiang Zhang,[1,2,3] Daokui Qu,[1,2,3] Fang Xu,[1,2,3] Kai Jia,[1,2,3] and Xueying Sun[2,4]

[1]*State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, No. 114 Nanta Street, Shenhe District, Shenyang 110016, China*
[2]*University of Chinese Academy of Sciences, No. 19A Yuquan Road, Beijing 100049, China*
[3]*SIASUN Robot & Automation Co., Ltd., No. 16 Jinhui Street, Hunnan New District, Shenyang 110168, China*
[4]*Department of Information Service and Intelligent Control, Chinese Academy of Sciences, No. 114 Nanta Street, Shenhe District, Shenyang 110016, China*

Correspondence should be addressed to Qiang Zhang; zhangqiang@sia.cn

Received 8 May 2016; Revised 19 July 2016; Accepted 1 August 2016

Academic Editor: Luis Payá

This paper introduces a dual-layer density estimation-based architecture for multiple object instance detection in robot inventory management applications. The approach consists of raw scale-invariant feature transform (SIFT) feature matching and key point projection. The dominant scale ratio and a reference clustering threshold are estimated using the first layer of the density estimation. A cascade of filters is applied after feature template reconstruction and refined feature matching to eliminate false matches. Before the second layer of density estimation, the adaptive threshold is finalized by multiplying an empirical coefficient for the reference value. The coefficient is identified experimentally. Adaptive threshold-based grid voting is applied to find all candidate object instances. Error detection is eliminated using final geometric verification in accordance with Random Sample Consensus (RANSAC). The detection results of the proposed approach are evaluated on a self-built dataset collected in a supermarket. The results demonstrate that the approach provides high robustness and low latency for inventory management application.

1. Introduction

With the development of robotics, humanoid robots have been introduced in innumerable applications. Among the available functionalities of the humanoid robot, specific object detection has attracted increasing attention in recent years. Inventory management, autosorting, and pick-and-place system are typical applications. Unlike single-object detection, multiple-instance detection is a more challenging task. In this paper, we focus on the goal of multiple object instance detection for robot inventory management and propose an effective approach to achieve this goal.

Multiple object instance detection is a complex technology that encounters a variety of difficulties. First, diversities of species, shapes, colors, and sizes of objects make it difficult to accomplish the fixed goal. Moreover, target objects appear different in different environments. For example, changes in scale, orientation and illumination increase uncertainty and ambiguity for identification. Additionally, multiple instances can affect the verification procedure.

There are two representative types of techniques for object instance detection: the training and learning-based approach and the template-based approach. The latter approach includes an extensive range of template forms, such as edge boxes [1], patches [2], and local features. Local feature matching based object detection method has received considerable attention from researchers because of its notable advantages in overcoming a portion of the deficiencies caused by scale, rotation, and illumination changes. Scale-invariant feature transform (SIFT) [3] was proposed by Lowe in 2004 and has been widely applied in many situations due to its robustness. A new approach, called PCA-SIFT [4], was proposed to simplify the calculations and decrease storage space. The main concept of PCA-SIFT is dimension reduction. In 2005, Mikolajczyk and Schmid proposed the gradient location and orientation histogram (GLOH) [5]. The GLOH is a SIFT-like

descriptor that uses a log-polar transformation hierarchy rather than four quadrants. The original high dimensionality of its descriptor can be reduced using PCA. In 2008, Bay et al. developed a prominent method known as speeded up robust features (SURF) [6] based on improvements in the construction of SIFT features. In [5, 7], the performances of local feature descriptors, such as SIFT, PCA-SIFT, GLOH, and SURF, were compared. According to [5, 7], PCA-SIFT and SURF have advantages in terms of speed and illumination changes, whereas SIFT and GLOH are invariant to rotation, scale changes, and affine transformations.

Feature matching is a basic procedure in object detection. Feature matching is typically performed by comparing the similarity of two feature descriptors. In fact, raw matches often contain a large number of mistakes; thus, false match elimination is necessary. The classical approaches are the ratio test [3], bidirectional matching algorithm [8], and RANSAC [9]. In addition, a remarkable method based on scale restriction [10, 11] was proposed. This method first estimates a dominant scale ratio using statistics after prematching. Then, features are reextracted from the high-resolution image at an adjusted Gaussian smoothing parameter according to the dominant scale ratio. After refined matching, feature pairs that do not conform to a certain scale ratio restriction are rejected. This method is adopted in our work due to its high performance. In addition, we provide a new approach to gaining access to the dominant scale ratio. In 2010, Arandjelović and Zisserman [12] considered that the Hellinger kernel leads to superior matching results compared to Euclidean distance in SIFT feature matching.

Lin et al. [13] used a key point coordinate clustering method for duplicate object detection. Regions of interest are detected using an adaptive window search. Wu et al. [14] reported an improved graph-based method to locate object instances. In [15], Collet et al. proposed a scalable approach known as MOPED. The framework first clusters matched features and generates hypothesis models. Potential instances can be found after an iterative process for pose refinement. However, the key point coordinates obtained from the clustering results in [13–15] might be unreliable because the key points are sparsely distributed. Alternatively, approaches based on Hough voting were proposed and applied in [16–18]. The Hough voting based approach locates possible instances according to feature mapping and density estimation. Specifically, the method in [16] applies mean-shift in the voting step. Similarly, grid voting was adopted in [19]. Although Hough voting is an effective approach for multiple object instance detection, the clustering radius for mean-shift or grid voting should be preset by experience, which leads to low adaptability and accuracy.

In this paper, we present a new architecture that improves multiple object instance detection accuracy by considering the adaptive selection of the optimal clustering threshold and a cascade of filters for false feature match elimination. The contributions of our work are as follows:

(i) We propose an architecture for multiple object instance detection based on dual-layer density estimation. The first layer calculates an optimal clustering threshold for the second layer and applies a constraint for the next scale restriction-based false match elimination. The second layer aims to detect all candidate object instances. The proposed strategy can reduce the possibility of mismatch and improve detection accuracy. Compared to traditional methods which need to set the threshold manually, the proposed adaptive clustering threshold computation method leads to stronger environmental flexibility and higher robustness.

(ii) We introduce a new method to compute and verify the value of the dominant scale ratio between the training image and query image. Rather than using a histogram statistical method for matched features, the value is derived from the first layer of the density estimation. Then the value is tested by an approximate one which is obtained based on the homography matrix. According to our experiments, the proposed method is more robust for dominant scale ratio estimation compared to the conventional methods.

The remainder of this paper is organized as follows. Section 2 describes the proposed architecture according to our particular application background. Details of the proposed method are discussed in Section 3. A variety of experiments are designed to evaluate our approach. The experimental methodology, results, and discussions are presented in Section 4. Finally, Section 5 summarizes our contributions and presents conclusions.

2. Framework Overview

In this section, we provide an introduction to the background of our work and briefly explain the proposed architecture.

Our work develops a service robot for a supermarket. The purpose of the robot is to count the goods before the start of business and provide feedback to the staff to ensure adequate supplies. Because no standard database exists for our specific application, we created a database for 70 types of man-made products to evaluate our algorithm.

The lighting conditions in the supermarket are generally uniform, and thus we collected training images for each item under same lighting conditions. One image was obtained from the front and another 24 were captured from 24 different directions. The frontal object image serves for object recognition, and all 25 sequence images were used to build a sparse 3D model for recovering pose of the identified object. All training images were captured at the distance which is approximately equal to the minimum safe distance between the robot and shelves. This sampling method can ensure that the training image has more details. To validate our architecture, the training database was divided into three sets based on the density of textures. The set with the highest density of textures contains 20 types of products, the set with a medium density of textures has 30 types, and the set with the lowest density of textures includes 20 types. For each object, there were 2 to 40 instances in the scene image.

Our proposed method is based on local features which can provide information about scale and rotation, and SIFT,

SURF, and PCA-SIFT are three alternatives. According to [5, 7], SIFT has better performance in scale and rotation change than SURF and PCA-SIFT; thus SIFT is used in our work although it is time-consuming. The proposed framework is based on SIFT feature extraction and feature matching by considering the specific application background. The framework consists of two phases: the offline training phase and the online detection phase. A graphic illustration of the proposed approach is shown in Figure 1. To make our algorithm more explicit, we make selected arrangements in advance. First, the term *key point* refers to a point with 2D coordinates, and the point is detected by SIFT theory. The term *descriptor* represents a 128-dimensional SIFT feature vector. The term *feature* consists of a description vector and the scale, orientation, and coordinate of the SIFT point.

In the offline phase, as shown in Figure 1(a), an initial value of the Gaussian smoothing parameter is given in advance. The SIFT features are extracted from the training images for certain objects. Reference vectors between all key points and the object center are computed to locate the object centroid. All features are stored in a retrieval structure to reduce time overhead during detection. On the other hand, we created a sparse 3D model for each object with a standard Structure from Motion algorithm [20] and each 3D point was associated with a corresponding SIFT descriptor.

The online detection phase is a dual-layer density estimation-based method. The first layer exists for two purposes: to compute the dominant scale ratio between the training image and query image (Figures 1(b)–1(e)) and to calculate a reference clustering threshold for the second layer of density estimation (Figures 1(f)–1(i)). At the beginning of feature extraction for the query image, an initial value of the Gaussian smoothing parameter is given, the same as in the training phase. All descriptors extracted from the video footage are matched to their nearest neighbors in the database (Figure 1(b)), and the key points are projected to their reference centers (Figure 1(c)). A valid object center with a maximum density value can be found using kernel density estimation (Figure 1(d)). Considering that object instances in our applications have nearly the same scale, the dominant scale ratio and an effective clustering threshold are computed accordingly (Figure 1(e)). The second layer of density estimation detects all possible instances. First, the feature template is reconstructed based on the initial value of the base scale and the calculated dominant scale ratio (Figure 1(f)). The majority of false feature matches are removed by a cascade of filters based on the distance ratio test and scale restriction (Figure 1(g)). The key point projection and 2D clustering methods are applied to find all candidate object centers (Figure 1(h)). The final geometric verification procedure can eliminate incorrect detection results and determine each instance's pose (Figure 1(i)).

3. Description of the Proposed Method

In this section, we introduce our work in detail in accordance with the aforementioned architecture. The schematic diagram for the offline training phase and the flowchart of the online detection are shown in Figures 2 and 4, respectively.

3.1. Offline Training: Template Generation and Retrieval Structure Construction. Indeed, the proposed method can be applied in conjunction with any scale and rotation invariant features. As is described in Section 2, SIFT is applied in our work for its robustness. To create templates for all types of object instances, frontal images of the targets must be captured. As noted in Section 2, the light conditions in our application are relatively invariant. In addition, we assume that all object instances face front outward. SIFT is able to work properly under these conditions. Thus, we can collect one frontal image for each type of product for object recognition. Besides, for the following object pose estimation, a sparse 3D model for each object was created (as shown in Figure 3), and thus 24 other images were captured at approximately equally spaced intervals in a circle around each object. According to SIFT theory, the Gaussian smoothing parameter should be given first. Suppose that the initial value is set to $\sigma_{\text{TrainInit}} = \sigma_o$. In this work, σ_o is a fixed value, as is described in Section 4, and the SIFT feature extraction takes place.

We assume that the number of features for a specific object is n. Each SIFT feature descriptor is a 128-dimensional vector f_i, where $i = \{1, 2, \ldots, n\}$. Similarly, the scale of the feature is s_i, the principle orientation is θ_i, and its coordinate is $c_i(x_i, y_i)$. Coordinate differences v_i between each SIFT key point $c_i(x_i, y_i)$ and the related object centroid $c_o(x_o, y_o)$ are calculated according to the following:

$$v_{io} = \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \end{bmatrix}. \quad (1)$$

Feature matching is a subprocedure in our multiple object instance detection architecture. The process is used to find the most similar feature in the dataset based on a distance measurement. In our work, the Hellinger distance measurement is applied due to its robustness according to [12]. Feature matching is typically a time-consuming process. The construction of an effective retrieval structure is necessary for speeding up the detection phase. Two types of effective retrieval methods are currently available: tree-based methods and hashing-based methods. The randomized kd-tree [21, 22], hierarchical k-means tree [21, 22], and vocabulary tree [23] are typical representatives of tree-based methods. Local sensitive hashing (LSH) [24, 25] and SSH [26] are two representative hashing-based methods. In all of the feasible methods, near-optimal hashing algorithms [27] have proven to be highly efficient and accurate, and this method was chosen for our work. Construction of multiple independent trees to form a forest is necessary to reduce the false negative and false positive rates.

3.2. Online Multiple Object Instances Detection

3.2.1. Feature Extraction for Query Image and Feature Matching. During online detection, the system first obtains access to a new captured video frame. SIFT key points are detected and descriptors are extracted in the same manner as the first part of offline procedure. The Gaussian smoothing parameter is also set as $\sigma_{\text{Query}} = \sigma_o$. Then, the near-optimal

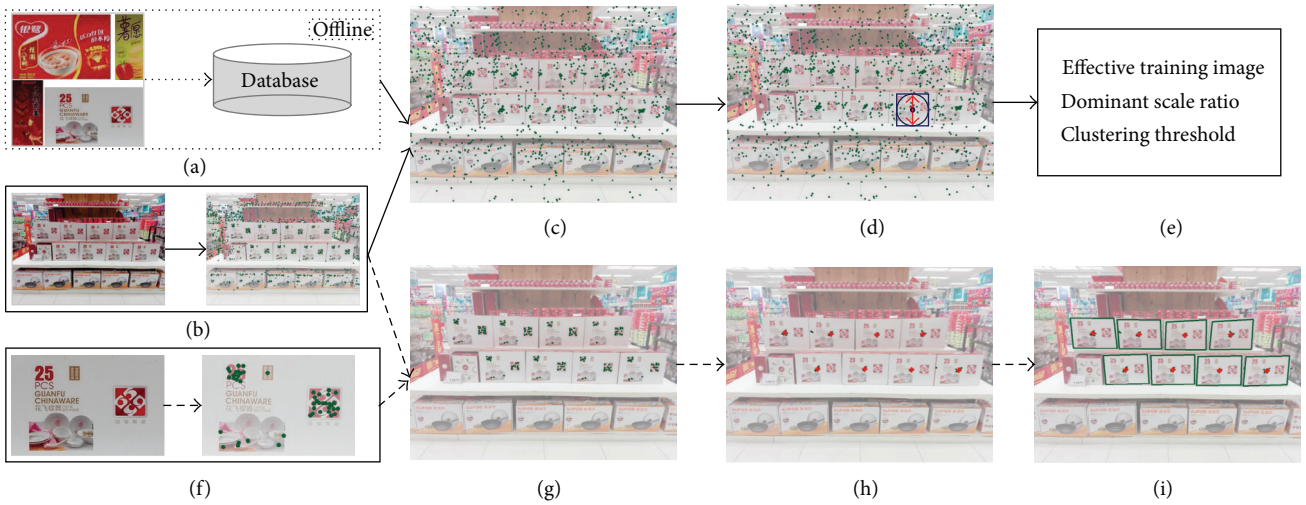


Figure 1: Overview of the proposed framework: (a) offline phase for constructing the retrieval structure; (b)–(e) first layer of density estimation: (b) local feature detection, (c) feature matching and key point mapping, (d) first layer of density estimation, and (e) intermediate results; (f)–(i) second layer of density estimation: (f) feature template reconstruction, (g) false matching result elimination, and (h) clustering for candidate instances detection; (i) geometric verification.

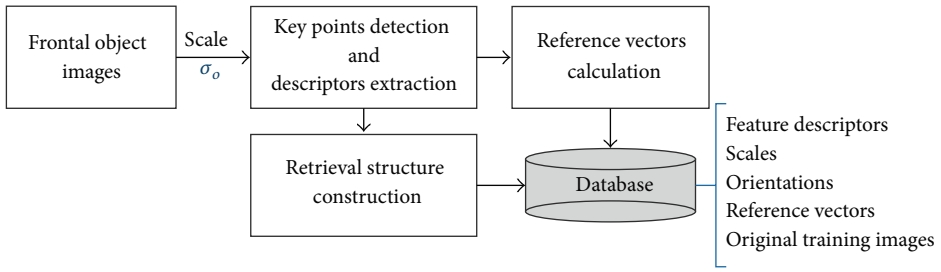


Figure 2: Offline training procedure.

Figure 3: 3D sparse model of packing box from 25 images.

hashing algorithm takes effect. During feature matching, low discriminable matches are discarded based on ratio test of distances between the nearest neighbor and second nearest neighbor, which was proposed in [3].

3.2.2. Key Points Projection and Object Center Estimation. The principle of key point projection is illustrated in Figure 5. In Figure 5, the left part is the training image, and the right part is the query image. Regarding the middle part, the solid region is a matched patch from the query image, and the area formed by dotted lines is assumed to be the ideal case in which there is only similarity transform. Assume that the matching pair of features is f_i and f_j, where f_i is from the database and f_j is from the query image. The key points corresponding to these two features are $p_i(x_i, y_i)$ and $p'_j(x'_j, y'_j)$. As for a plane object, the center $\widehat{c'_{oj}}(\widehat{x'_{oj}}, \widehat{y'_{oj}})$, related to f_j, can be estimated according to (2)–(5).

In the formulas, s'_j and θ'_j are the corresponding scale and orientation of feature f'_j. Similarly, s_i and θ_i are related to feature f_i in the training image. For each pair of matching features, there is a normalized deflection angle ε_j between the normal vector of an object surface and camera optical axis for each matched features. According to (5), the estimated centers would be located in a small range of areas around the real center when the training image is the exact image corresponding to the ordered object instance and ε_j has an extremely small value.

$$\theta = \theta'_j - \theta_i. \tag{2}$$

As shown in Figure 5, reference centers are distributed in small areas. Then, the problem of determining the center

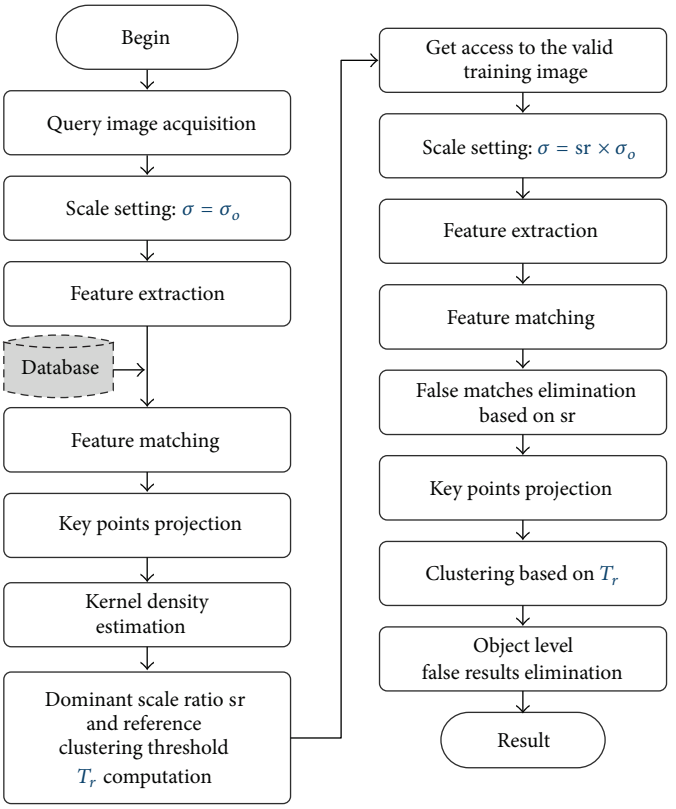


Figure 4: Online detection flowchart.

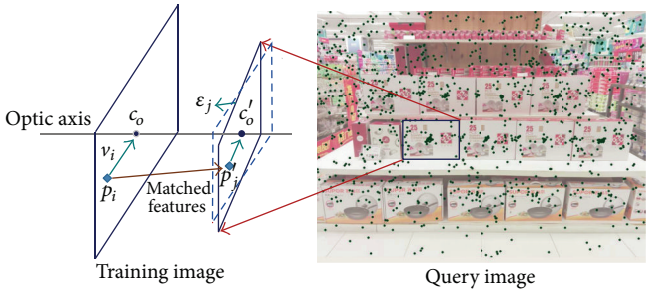


Figure 5: Key points projection principle diagram.

coordinates is converted into a density estimation problem. The first layer of density estimation aims to find one of the valid centers in the query image. Object center estimation is a crucial problem. A two-stage procedure-based adaptive kernel density estimation method, elaborated in [28], is employed to improve the precision. Only those density values associated with the mapped key points are calculated to speed up the process. The point with the highest density value is saved. Although this point may be not the exact center, it is a typical approximation. Thus, the mapped point is identified as a valid center. Simultaneously, the exact training image can be obtained. As is illustrated in Figure 6, the blue point is the obtained object center.

$$\begin{bmatrix} \widehat{x'_{oj}} \\ \widehat{y'_{oj}} \end{bmatrix} = \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \frac{s'_j}{s_i} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times v_i \times \cos\varepsilon_j \tag{3}$$

$$\begin{aligned} &= \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \frac{s'_j}{s_i} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times v_i \\ &\times \left(1 - \frac{\varepsilon_j^2}{2!} + \frac{\varepsilon_j^4}{4!} - \cdots\right) \end{aligned} \tag{4}$$

$$\begin{aligned} &= \underbrace{\begin{bmatrix} x'_{oj} \\ y'_{oj} \end{bmatrix}}_{\text{RealCenter}} \\ &\underbrace{+ \frac{s'_j}{s_i} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times v_i \times \left(-\frac{\varepsilon_j^2}{2!} + \frac{\varepsilon_j^4}{4!} - \cdots\right)}_{\text{DistributionRange}}. \end{aligned} \tag{5}$$

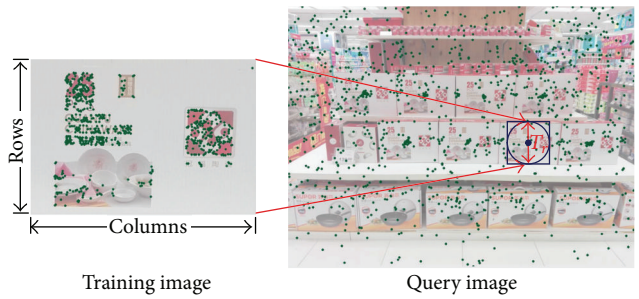


Figure 6: Reference clustering threshold calculation.

3.2.3. Dominant Scale Ratio Estimation and Scale Restriction-Based False Matches Elimination. The dominant scale ratio serves two purposes: false match elimination and calculation of a reference clustering radius for the second layer of density estimation. In contrast to the conventional methods in [10, 11], the dominant scale ratio in our work can be derived according to (6) based on the assumption that the estimated center has a typical scale ratio value. In (6), sr is the oriented scale ratio, s'_m is the scale of the key point related to the estimated object center, and s_n is the scale of the matched key point in the training image.

$$\mathrm{sr} = \frac{s'_m}{s_n}. \tag{6}$$

Once the valid center is found, the points that support the center are recorded. These points are used to calculate the homography matrix H_o for the pattern. The matrix is shown in (7). Because the minimum safe distance between the robot and the shelves is far enough, which means the camera on the robot is far from the targets, the actual homography is sufficiently close to affine transformation. Then the dominant scale ratio sr′ can also be computed according to (8). Then, sr′ is used to verify sr. Only if the value of sr is approximate to sr′, the value of sr is confirmed to be correct. We use (9) to assess the similarity between the two values.

$$H_o = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}, \tag{7}$$

$$\mathrm{sr}' = \sqrt{|h_{11} \times h_{22}| + |h_{12} \times h_{21}|}, \tag{8}$$

$$\left| \frac{\mathrm{sr} - \mathrm{sr}'}{\min(\mathrm{sr}, \mathrm{sr}')} \right| < 15\%. \tag{9}$$

To find all possible object instances, a SIFT feature-based template of the ordered object must be reconstructed (see Figure 1(f)). The Gaussian smoothing factor is to be set based on the dominant scale ratio and is adjusted in accordance with (10). A new retrieval structure is constructed after SIFT features are detected. Then, features obtained from the query image above are matched to the new dataset. Due to the aforementioned preprocessing, the amount of SIFT features in the newly constructed database is reduced compared to offline training phase. Thus, the time overhead of the matching process is greatly reduced.

$$\sigma_{\mathrm{TrainAdjust}} = \mathrm{sr} \times \sigma_o. \tag{10}$$

The strategy of feature matching disambiguation here is a cascade of filters. These filters can be divided into the ratio test algorithm (proposed in [3]), scale restriction-based method (presented in [11]), and geometric verification-based approach. The ratio test and scale restriction methods use the following matching process. The geometric verification takes effect after clustering. After this series of filters, most of false matches can be eliminated.

3.2.4. Reference Clustering Threshold Computation and Candidate Object Instances Detection. Traditional methods for detecting multiple object instances, such as mean-shift and grid voting, are based on density estimation. However, these methods have the same disadvantage that the bandwidth must be given by experience. For example, in [16], the clustering threshold was set to a specific value. In [19], the voting grid size was set to the value associated with the size of the query image. Nevertheless, this approach may still lead to unreliable results. For our specific application occasion, the clustering threshold can be estimated based on the size of training image and the aforementioned dominant scale ratio. Before the clustering threshold is finally determined, a reference clustering threshold should be computed automatically. Here, the reference clustering threshold can be estimated based on (11). In the formula, T_r is the reference clustering threshold, sr is the oriented scale ratio, and rows and cols are the numbers of rows and columns in the training image, respectively. As noted above, the mapped key points are located in small regions around real centroids. Therefore, the clustering threshold Th can be finalized in line with (12), in which k is a correction factor. According to our repeated experiments described in Section 4, we provide a recommended value for k. Candidate object instance detection is based on the second layer of density estimation. Grid voting is employed here due to its high precision and recall.

$$T_r = \begin{cases} \mathrm{sr} \times \mathrm{rows}, & \text{if rows} < \text{cols} \\ \mathrm{sr} \times \mathrm{cols}, & \text{otherwise,} \end{cases} \tag{11}$$

$$\mathrm{Th} = k \times T_r. \tag{12}$$

3.3. Object Level False Result Elimination. In the procedure for eliminating false detection results, we first calculate the homography matrix for each cluster. Then, four corners of the training image are projected onto four new coordinates. As a result, a convex quadrilateral in accordance with the four mapped corners is produced. Here, we provide a simple but effective way to assess whether the system has obtained correct object instances, and error detections are eliminated. The criterion is as follows:

$$c_{\min} \le \frac{\mathrm{Area\,(Quadrilateral)}}{\mathrm{sr}^2 \times \mathrm{Area\,(TrainingImage)}} \le c_{\max} \tag{13}$$

(a) (b) (c)

Figure 7: Examples of objects with different texture levels: (a) high texture; (b) medium texture; (c) low texture.

In (13), Area(Quadrilateral) is the area of the convex quadrilateral derived from each candidate object instance. Area(TrainingImage) is the area of the training image. According to (13), if the detection is accurate, the ratio coefficient between the area of the quadrilateral and the training image is approximate to sr^2. The threshold c_{min} and c_{max} should be set before verification.

Finally, for each cluster, the features are matched to the 3D sparse model created in the offline training procedure. A noniterative method called EPnp [29] was employed to estimate pose for each object instance.

4. Experiments

4.1. Experimental Methodology. We are developing a service robot for the detection and manipulation of multiple object instances, and there is no standard database for our specific application. To validate our approach, we created a database for 70 types of products with different shapes, colors, and sizes in a supermarket. Objects to be detected were placed on shelves with the front outside. All images were captured using a SONY RGB camera. The resolution of the camera was 1240×780 pixels. To comprehensively evaluate the accuracy of the proposed architecture, the database was divided into three sets according to the texture level of the objects. Figure 7 shows examples of objects with different texture levels.

We designed three experiments to evaluate the proposed architecture. The first experiment was to verify whether the scale ratio calculation and false elimination method were feasible. The second one was to examine whether the proposed clustering threshold computation method was effective. The last experiment was to comprehensively evaluate the performance of the proposed architecture. These three experiments were designed as follows:

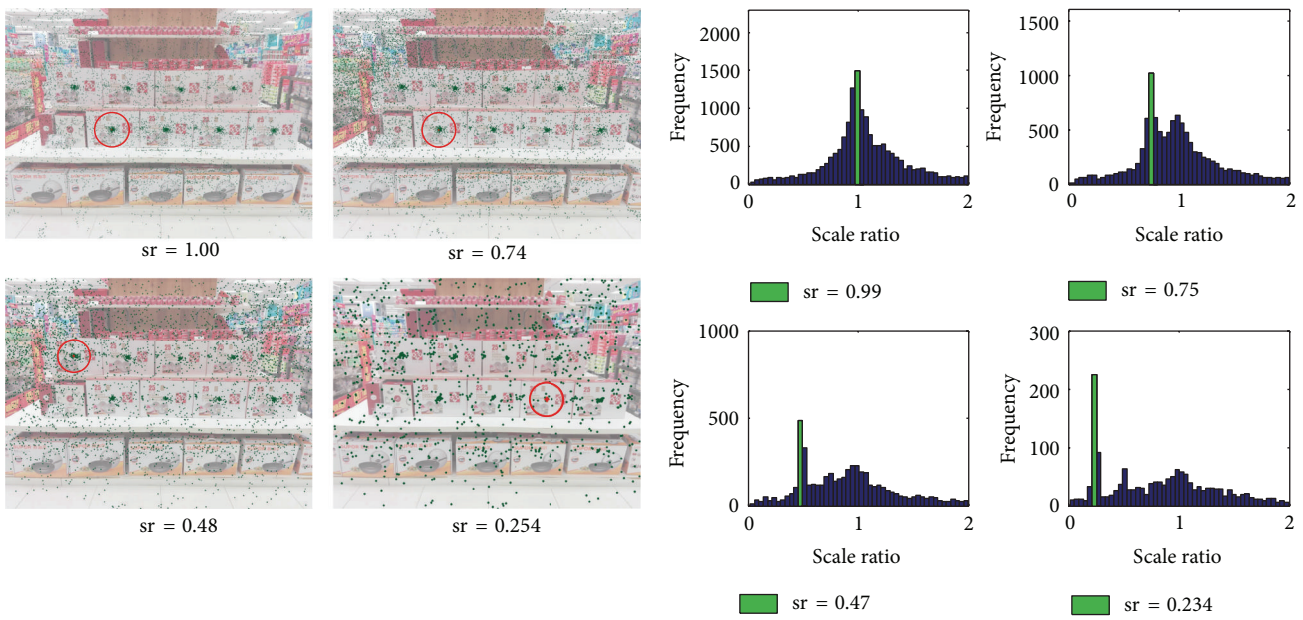
(i) Experiment I: for each training image in the database, we acquired an image considering that the object instance in the image had the same scale as the training image. Then, the captured images were downsampled. The size of the resampled images were 100%, 75%, 50%, and 25% of the original size. We calculated the dominant scale ratios based on the conventional histogram statistics and proposed method separately. Then, the accuracy of both values was compared. The feature matching and key point projection results with and without false elimination were also recorded and compared.

(ii) Experiment II: we first calculated a clustering threshold according to (14). Then, we tested the performance of the conventional methods (mean-shift and grid voting) based on changing the clustering threshold continuously. Here, an approximate nearest neighbor searching method was employed to speed up mean-shift. Because the thresholds could not be directly compared in different experiments, we used the multiple of the computed threshold in different experiments to express the new value. In (14), CR is the bandwidth for mean-shift, GS is the grid size for grid voting, and k_{MS} and k_{GV} are the coefficients. We chose an optimal threshold value according to the experimental results. In the experiment, the threshold ratio parameters were sampled as $k_{MS} = k_{GV} = \{2.6, 2.4, 2.2, 2.0, 1.9, 1.8, 1.7, 1.6, 1.4, 1.2, 1.0, 0.8\}$.

$$\begin{aligned} \mathrm{CR} &= \frac{1}{2} \times k_{\mathrm{MS}} \times T_r, \quad \text{using mean-shift}, \\ \mathrm{GS} &= k_{\mathrm{GV}} \times T_r, \quad \text{using grid voting}. \end{aligned} \tag{14}$$

(iii) Experiment III: we compared the proposed method with the conventional grid voting on three types of datasets. The experimental conditions of the conventional grid voting were as follows: width and height of the grid are 1/130 of the width and the height of the query image, and the voting grid had an overlap of 25% of size with an adjacent grid. The performances of the proposed method and the conventional grid voting were expressed in terms of the accuracy (precision and recall) and computational time.

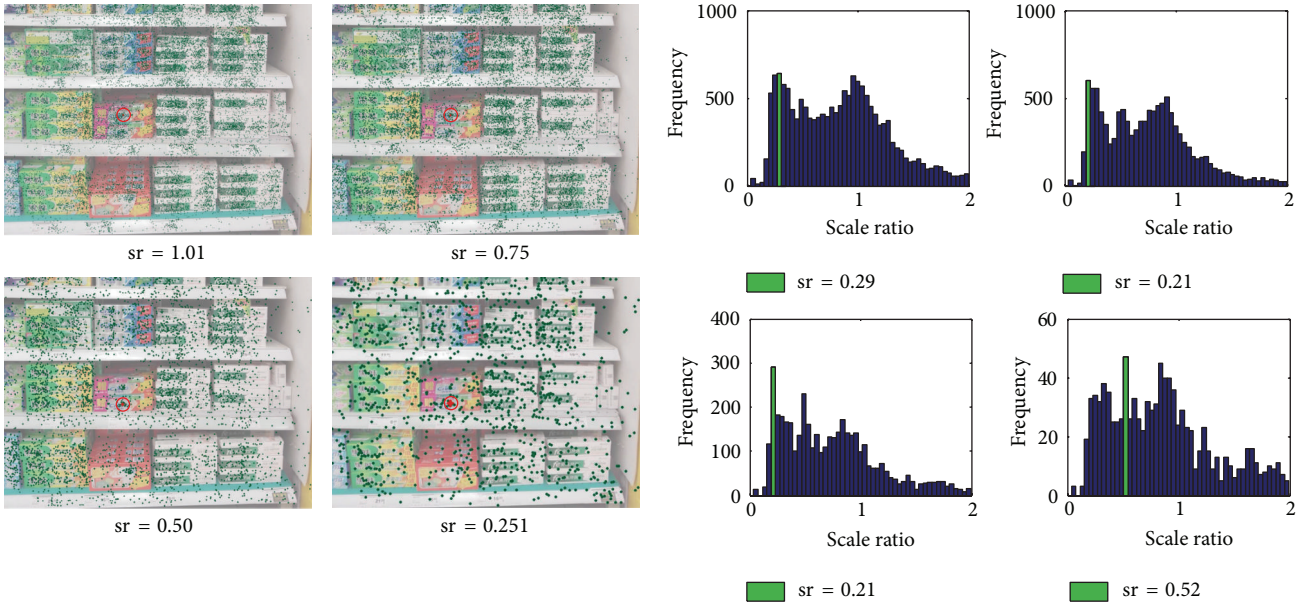
In all the experiments, the parameters for SIFT feature extraction and the threshold for feature matching were set as the default values in [3]. In particular, the initial Gaussian smoothing parameter was set as $\sigma_o = 1.6$, and the default threshold on key point contrast was set to 0.1. In the verification procedure in our experiments, thresholds c_{min} and c_{max} were set as 0.8 and 1.2, respectively. In our work, all of the experiments have been conducted on Windows 7 PC with Core i7-4710MQ CPU @ 2.50 GHz and 8 GB RAM.



(a) Center estimation and dominant scale ratio computation by proposed method

(b) Dominant scale ratio computation by conventional histogram statistic

Figure 8: The first example of dominant scale ratio computation.



(a) Center estimation and dominant scale ratio computation by proposed method

(b) Dominant scale ratio computation by conventional histogram statistic

Figure 9: The second example of dominant scale ratio computation.

4.2. Experimental Results and Analysis

4.2.1. Results of the Dominant Scale Ratio Computation and Scale Restriction-Based False Match Elimination. Figures 8 and 9 display the results of two examples for computing the dominant scale ratios. Figures 8(a) and 9(a) are the results of the proposed method, whereas Figures 8(b) and 9(b) are the results of the conventional method. The reference scale ratios are 100%, 75%, 50%, and 25% in these figures. In Figures 8(a), 8(b), and 9(a), the calculated results are close to the reference values. However, in Figure 9(b), the results obtained by the conventional method are not reliable. The reason for the error in Figure 9(b) is that the background noise is too severe and the extracted features may have nearly the same

(a) (b) (c)

Figure 10: Raw matching results: (a) training image; (b) feature matching; (c) key points projection.

(a) (b) (c)

Figure 11: Matching results with false matches elimination: (a) training image; (b) feature matching; (c) key points projection.

scale ratio. The proposed method evaluates the dominant scale ratio depending on the distribution and relationship of key points; therefore, the result is more reliable.

Figure 10 shows that the raw matching results without scale-constrained filtering exhibit a large number of false matches. The matching results based on scale-constrained filtering are shown in Figure 11, with fewer outliers present. Scale restriction-based template reconstruction and elimination of false matches lead to the best optimum results (Figure 12). Most of the false matches are eliminated and lay a good foundation for the subsequent clustering. Figures 10–12 illustrate the effectiveness of the proposed filters.

4.2.2. Results of Clustering Threshold Estimation. Figures 13(a)–14(b) show the performance of the methods using mean-shift and grid voting. The brown curve in Figure 13(a) describes the accuracy of grid voting, and the blue one describes accuracy of mean-shift. Figure 13(b) illustrates the true positive rate versus false positive rate of mean-shift and grid voting as the discrimination threshold changes. Points in both Figures 13(a) and 13(b) were sampled based on different clustering threshold ratios, as detailed in the experimental methodology. The threshold ratio values decrease gradually from left to right. Besides, coordinates surrounded by circles are related to the precalculated threshold. Figures 14(a) and 14(b) show the average value and standard deviation of computational time for mean-shift and grid voting based on different thresholds.

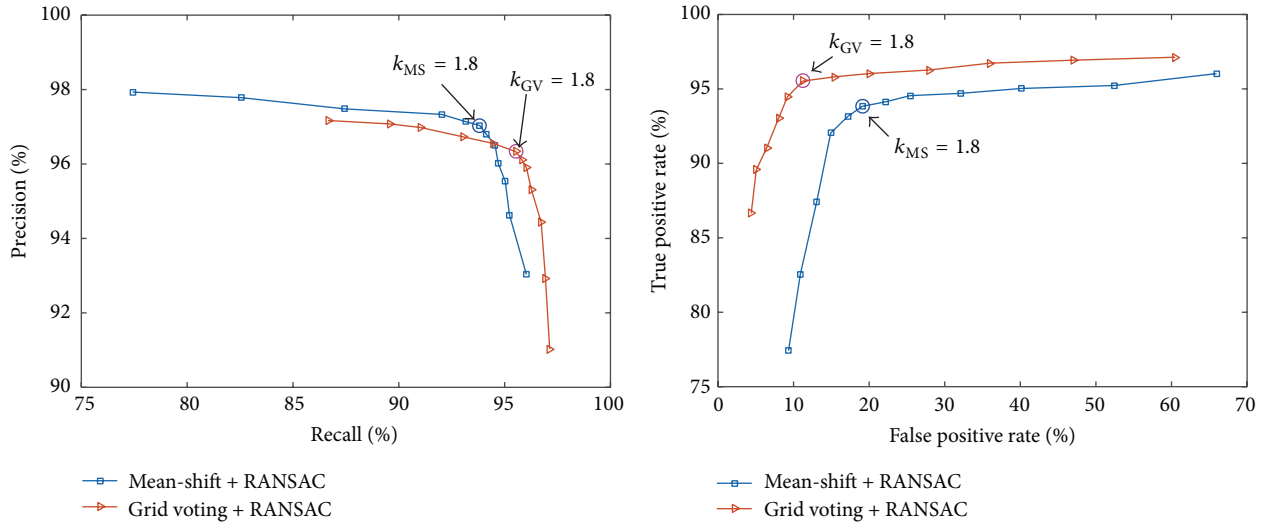
As shown in Figure 13(a), the precision decreases and the recall increases as the threshold is decreased. In Figure 13(b), both the true and false positive rates increase as the threshold is decreased. Figure 13(a) shows that grid voting has a better performance than mean-shift in recall as a whole and Figure 13(b) indicates that grid voting has a better performance in accuracy than mean-shift. According to Figures 13(a) and 13(b), k_{MS} and k_{GV} corresponding to the inflection point are both 1.8. As shown in Figure 14(a), the time cost for feature matching and ANN-based mean-shift clustering remains relatively stable. However, a smaller threshold ratio leads to a higher time cost for geometric verification because the number of clusters increases. As shown in Figure 14(b), the computational time for clustering using grid voting is considerably shorter than when using mean-shift, but the verification time becomes longer due to the clustering errors. According to the results of the feasibility validation, clustering radius $k_{MS} = 1.8$ for mean-shift and $k_{GV} = 1.8$ for grid voting are optimized preset parameters for the detection of multiple object instances in inventory management.

4.2.3. Performance for Different Object Instance Detection Based on the Proposed Architecture. Table 1 shows the average results of different levels of textures using the proposed method and grid voting. The precision and recall were recorded. The computational times for feature extraction, raw matching, density estimation, template reconstruction-based rematching, clustering, and geometric verification were documented separately. Figure 15 shows the results of two examples using the proposed method.

According to Table 1, different levels of texture density will lead to different accuracies and computational times.

(a) (b) (c)

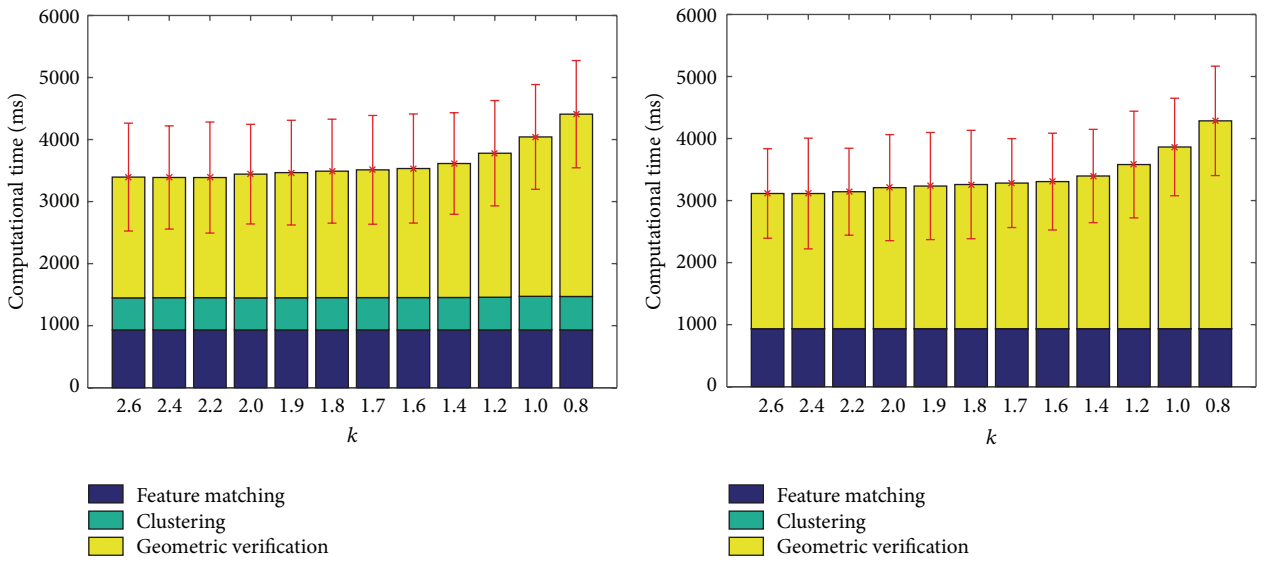
FIGURE 12: Matching results based on template reconstruction and scale restriction: (a) training image; (b) feature matching; (c) key points projection.



(a) Accuracy of mean-shift and grid voting

(b) True positive rate versus false positive rate of mean-shift and grid voting

FIGURE 13: Accuracy performance using mean-shift and grid voting.



(a) Computational time for mean-shift

(b) Computational time for grid voting

FIGURE 14: Computational time statistics.

(a) (b) (c)

Figure 15: Results of two detection examples.

Table 1: Average results for different levels of texture using proposed method and grid voting.

Texture level	Methods	Accuracy (%)		Computational time (ms)						
		Precision	Recall	Feature detection	Raw match	Density estimation	Rematch	Clustering	Geometric verification	Total
High	Proposed	97.6	96.8	1027	379	479	526	3	522	2936
	Grid voting	96.2	96.3	1027	379	0	0	4	2595	4005
Medium	Proposed	96.4	95.8	941	220	191	246	3	866	2467
	Grid voting	95.7	95.4	941	220	0	0	4	2033	3198
Low	Proposed	92.1	93.6	586	94	72	119	4	1054	1929
	Grid voting	91.6	91.9	586	94	0	0	3	1345	2028

Precision and time overhead increase with increases in the texture density. Although the first layer of density estimation and template reconstruction-based rematching take some computational time, the geometric verification latency is greatly reduced compared to the conventional method because the adaptive threshold is more reasonable than the judgment based simply on the size of the query image. Table 1 indicates that the proposed architecture can accurately detect and identify multiple identical objects with low latency. As can be seen in Figure 15, most of object instances were detected. However, objects marked as "A" in Figure 15(a), "B," "C," and "D" in Figure 15(b), and "F," "H," and "G" in Figure 15(c) were not detected and objects marked as "E" were a false detection result. Reasons for these errors are the reflection of light (in Figure 15(a)), high similarity of objects (the short bottle marked as "E" is similar to the high one in Figure 15(b)), translucent occlusion (three undetected yellow bottles marked as "B," "C," and "D" in Figure 15(b)), and error clustering results ("F," "G," and "H" in Figure 15(c)).

5. Conclusions

In this paper, we introduced the problem of multiple object instance detection in robot inventory management and proposed a dual-layer density estimation-based architecture for resolving this issue. The proposed approach is able to successfully address the multiple object instance detection problem in practice by considering dominant scale ratio-based false match elimination and adaptive clustering threshold-based grid voting. The experimental results illustrate the superior performance our proposed method in terms of its high accuracy and low latency.

Although the presented architecture performs well in these types of applications, the algorithm would fail when applied to more complex problems. For example, if object instances have different scales in the query image, the assumptions made in this paper will be no longer valid. Further more, the accuracy of the proposed method will be greatly reduced when there is a dramatic change of illumination or the target is occluded by other translucent objects. In our future work, we will focus on improving the method for solving such complex problems.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank Shenyang SIASUN Robot Automation Co., Ltd., for funding this research. The project is supported by The National Key Technology R&D Program, China (no. 2015BAF13B00).

References

[1] C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," in *Proceedings of the European Conference*

on Computer Vision (ECCV '14), Zurich, Switzerland, September 2014, pp. 391–405, Springer, Cham, Switzerland, 2014.

[2] S. Hinterstoisser, S. Benhimane, N. Navab, P. Fua, and V. Lepetit, "Online learning of patch perspective rectification for efficient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, Anchorage, Alaska, USA, June 2008.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. II506–II513, Washington, DC, USA, July 2004.

[5] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[7] L. Juan and O. Gwun, "A comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing*, vol. 3, no. 4, pp. 143–152, 2009.

[8] Q. Sen and Z. Jianying, "Improved SIFT-based bidirectional image matching algorithm. Mechanical science and technology for aerospace engineering," *Mechanical Science and Technology for Aerospace Engineering*, vol. 26, pp. 1179–1182, 2007.

[9] J. Wang and M. F. Cohen, "Image and video matting: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, 2008.

[10] Y. Bastanlar, A. Temizel, and Y. Yardimci, "Improved SIFT matching for image pairs with scale difference," *Electronics Letters*, vol. 46, no. 5, pp. 346–348, 2010.

[11] J. Zhang and H.-S. Sang, "SIFT matching method based on base scale transformation," *Journal of Infrared and Millimeter Waves*, vol. 33, no. 2, pp. 177–182, 2014.

[12] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2911–2918, San Francisco, Calif, USA, June 2012.

[13] F.-E. Lin, Y.-H. Kuo, and W. H. Hsu, "Multiple object localization by context-aware adaptive window search and search-based object recognition," in *Proceedings of the 19th ACM International Conference on Multimedia ACM Multimedia (MM '11)*, pp. 1021–1024, ACM, Scottsdale, Ariz, USA, December 2011.

[14] C.-C. Wu, Y.-H. Kuo, and W. Hsu, "Large-scale simultaneous multi-object recognition and localization via bottom up search-based approach," in *Proceedings of the 20th ACM International Conference on Multimedia (MM '12)*, pp. 969–972, Nara, Japan, November 2012.

[15] A. Collet, M. Martinez, and S. S. Srinivasa, "The MOPED framework: object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1284–1306, 2011.

[16] S. Zickler and M. M. Veloso, "Detection and localization of multiple objects," in *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 20–25, Genova, Italy, December 2006.

[17] G. Aragon-Camarasa and J. P. Siebert, "Unsupervised clustering in Hough space for recognition of multiple instances of the same object in a cluttered scene," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1274–1284, 2010.

[18] R. Bao, K. Higa, and K. Iwamoto, "Local feature based multiple object instance identification using scale and rotation invariant implicit shape model," in *Proceedings of the 12th Asian Conference on Computer Vision (ACCV '14), Singapore, November 2014*, pp. 600–614, Springer, Cham, Switzerland, 2014.

[19] K. Higa, K. Iwamoto, and T. Nomura, "Multiple object identification using grid voting of object center estimated from keypoint matches," in *Proceedings of the 20th IEEE International Conference on Image Processing (ICIP '13)*, pp. 2973–2977, Melbourne, Australia, September 2013.

[20] R. Szeliski and S. B. Kang, "Recovering 3D shape and motion from image streams using nonlinear least squares," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '93)*, pp. 752–753, IEEE, New York, NY, USA, June 1993.

[21] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the 4th International Conference on Computer Vision Theory and Applications (VISAPP '09)*, pp. 331–340, Lisboa, Portugal, February 2009.

[22] M. Muja and D. G. Lowe, "Fast matching of binary features," in *Proceedings of the 9th Conference on Computer and Robot Vision (CRV '12)*, pp. 404–410, IEEE, Toronto, Canada, May 2012.

[23] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 2161–2168, IEEE, New York, NY, USA, June 2006.

[24] B. Matei, Y. Shan, H. S. Sawhney et al., "Rapid object indexing using locality sensitive hashing and joint 3D-signature space estimation," *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 28, no. 7, pp. 1111–1126, 2006.

[25] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2130–2137, Kyoto, Japan, October 2009.

[26] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3424–3431, IEEE, San Francisco, Calif, USA, June 2010.

[27] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '06)*, pp. 459–468, Berkeley, Calif, USA, October 2006.

[28] B. W. Silverman, "Density Estimation for Statistics and Data Analysis, Chapman & Hall, London—New York, 1986, 175 pp., £12.," *Biometrical Journal*, vol. 30, pp. 876–877, 1988.

[29] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.