Local Feature based Multiple Object Instance Identification using Scale and Rotation Invariant Implicit Shape Model

Ruihan Bao, Kyota Higa, Kota Iwamoto

Information and Media Processing Laboratories, NEC Corporation

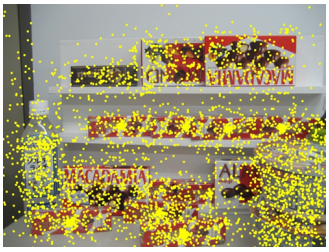
Abstract. In this paper, we propose a Scale and Rotation Invariant Implicit Shape Model (SRIISM), and develop a local feature matching based system using the model to accurately locate and identify large numbers of object instances in an image. Due to repeated instances and cluttered background, conventional methods for multiple object instance identification suffer from poor identification results. In the proposed SRIISM, we model the joint distribution of object centers, scale, and orientation computed from local feature matches in Hough voting, which is not only invariant to scale changes and rotation of objects, but also robust to false feature matches. In the multiple object instance identification system using SRIISM, we apply a fast 4D bin search method in Hough space with complexity $O(n)$, where n is the number of feature matches, in order to segment and locate each instance. Futhermore, we apply maximum likelihood estimation (MLE) for accurate object pose detection. In the evaluation, we created datasets simulating various industrial applications such as pick-and-place and inventory management. Experiment results on the datasets show that our method outperforms conventional methods in both accuracy (5%-30% gain) and speed (2x speed up).

1 Introduction

Locating and identifying multiple objects in an image is important for robotics [1, 2] and automation [3]. Furthermore, it also attracts attentions for industrial applications such as inventory management and planograms [4, 5]. Figure 1(a) shows an example of multiple object identification. In such applications, instead of understanding general object classes [6], recognizing specific object instances and there poses (e.g. its location, orientation and relative scale) is of interest. Though the definition of the object instance is varied in researches [4, 7, 8], in this paper we are interested in the problem like [2, 4], in which an instance is a particular object example that has identical texture (i.e. appearance) with the database object.

For object instance detection, local feature based methods using SIFT [9] and SURF [10] are very popular. A classic process includes feature extraction (i.e. keypoint detection and local descriptor generation), feature matching, and geometric verification by Hough transform or RANdom SAmple Consensus

(a)Example of multiple object identification



(b)Results of object center estimation

Database image

Fig. 1. Multiple object identification.

(RANSAC). When there are few or no repeated instances in the query image, the problem can be simply treated as detecting objects by identifying true feature matches from false feature matches caused by background or irrelevant objects in the foreground. However, in the case when many repeated instances are present in an image (e.g. images of production lines or store shelves), in addition to identifying true matches from false matches, since all instances generate true feature matches, it is also crucial to segment those correct feature matches individually and locate each instance accordingly.

In order to locate and identify each instance in an image containing multiple object instances, [2, 4, 11] propose methods that cluster keypoint coordinates in query images. Specifically, the method in [4] applies windows to locate possible positions of object instances. In contrast, [11] applies graph based method using Markov Random Field (MRF) on the feature matches to segment object instances. Finally, in [2], a scalable and low latency object recognition system called MOPED is introduced. The system locates object instances by roughly clustering keypoints coordinates using mean-shift after feature matching, and then applying coarse-to-fine object detection steps using RANSAC iteratively. In order to improve the speed for the process, the system is carefully implemented by taking advantages of parallel computing technology such as OpenMP and GPU. These methods, however, have a common problem. Since keypoints of an instance are sparsely distributed and do not form dense clusters for each instance, it is therefore very difficult to achieve high detection accuracy by clustering on keypoint coordinates in complex scenes (e.g. many repeated instances or cluttered background).

Alternatively, [1, 3, 5] propose Hough voting based methods. These methods allow each feature match to cast a vote for the common object center position estimated using keypoint scale, orientation and the coordinates, and then locating object instances by clustering object centers using mean-shift [1, 3] or grid

voting [5]. Methods employing object centers are effective in locating instances since object centers are more densely clustered in Hough space. However, when applied to real world applications, these methods are still problematic due to large number of false matches. Figure 1 (b) illustrates the difficulties in detecting multiple objects in a complex scene, in which we plot the object centers computed from local features using methods in [1, 3, 5]. It shows that though the estimated object centers from true feature matches form clusters, they are overwhelmed by object centers estimated from false matches, thus making it difficult to locate individual instance. In [9], a 4D Hough voting method is introduced combined with an iterative outlier removal scheme. However, as will be discussed later, the object location vote (instead of object center vote) in [9] is very sensitive to both scale changes and rotation of objects. When repeated instances are present, location votes from different instances will be excessively overlapped in the Hough space, making it hard to differentiate and locate objects by Hough voting. Therefore, this method is not suitable for multiple instance detection.

Recently, in a related area of object category detection, Implicit Shape Model (ISM) [6] has been proposed and received a lot of attentions. It successfully combines feature matching, codebook learning and Hough voting into the same framework and produces promising results. ISM adopts scale invariant object centers in Hough voting and extends the voting space to include scale changes as the third dimension. Nevertheless, ISM is not invariant to object rotation and thus can only be used under the assumption that all objects in query images have no rotation.

In this paper, we apply the idea of ISM to instance identification and extend it to accommodate scale and rotation changes by proposing Scale and Rotation Invariant Implicit Shape Model (SRIISM). Specifically, we compute object centers using keypoint scale, orientation and coordinates so that they are invariant to object centers compared to original ISM. In addition, we add object scale and orientation votes to make the Hough voting more robust to false matches compared with conventional methods [1, 3, 5]. This is equivalent to weighting object centers according to the distribution of object rotation and scale. The main contributions of this paper are:

(1) We propose a method called SRIISM that models the joint distribution of object centers, scale and orientation in Hough voting. The proposed method is not only invariant to object scale changes and rotation, but also very robust to false matches caused by cluttered background and irrelevant objects.

(2) We apply the model of SRIISM and develop a system for multiple object instance identification, which includes 4D Hough voting, fast 4D bin search of complexity $O(n)$, and pose estimation using maximum likelihood estimation (MLE). The system is tested on datasets simulating various applications such as pick-and-place and inventory management, and we show that superior performance in both speed and accuracy can be achieved.

The paper is organized in the following way. In section 2, the proposed model of SRIISM is discussed. In section 3, the details of the multiple object in-

stance identification system using SRIISM is introduced. Finally, the evaluation datasets and experiment results are described and discussed in section 4.

2 Scale and Rotation Invariant Implicit Shape Model (SRIISM)

In this section, we introduce the Scale and Rotation Invariant Implicit Shape Model (SRIISM) for instance identification, inspired by ISM [6, 12]. Different from the original ISM using visual words which can be seen as a special type of local feature, our model uses original local features extracted from images. Let f_j be the observed local feature (represented by descriptor) in the query image and l_j be the associated parameters (feature pose) of the feature, which is the 2D coordinates, orientation and scale of the feature. l_j can be easily obtained from scale and rotation invariant local features such as SIFT and SURF. Let $p(O, x)$ be the probability of the presence of object O with pose x. x includes object center, orientation and scale. We denote a local feature entry as D_i, which contains the local descriptor as well as associated coordinates, orientation and scale. The SRIISM then computes the probability of $p(O, x)$ by marginalizing through local features ($p(O, x, f_j, l_j)$) in an query image, i.e

$$p(O, x) = \sum_j p(O, x, f_j, l_j) \tag{1}$$

$$= \sum_j p(f_j, l_j) p(O, x | f_j, l_j) \tag{2}$$

Assume that the prior term $p(l_j, f_j)$ over features and feature pose are uniformly distributed, we marginalize again for the feature entries (D_i) in the database, and get the following equations,

$$p(O, x) \propto \sum_j p(O, x | f_j, l_j) \tag{3}$$

$$= \sum_{i,j} p(O, x | D_i, f_j, l_j) p(D_i | f_j, l_j) \tag{4}$$

$$= \sum_{i,j} p(O, x | D_i, l_j) p(D_i | f_j) \tag{5}$$

$$= \sum_{i,j} p(x | O, D_i, l_j) p(D_i | f_j) p(O | D_i) \tag{6}$$

From Eq. (4) to Eq. (5), we used the fact that $p(D_i | f_j, l_j) = p(D_i | f_j)$, which means that an observed local feature f_j is matched to the feature entries D_i only by its local feature. Moreover, $p(O, x | D_i, f_j, l_j) = p(O, x | D_i, l_j)$ is based on the fact that object pose x is only inferred by the feature coordinates, scale and orientation from query images (l_j) and database images (D_i). Finally, applying

Bayes rule to $p(O, x|D_i, l_j)$ and assuming that $p(O|D_i, l_j) = p(O|D_i)$ (i.e. coordinates, scale and orientation of local features are independent to the presence of objects), we obtain Eq. (6).

In Eq. (6), $p(x|O, D_i, l_j)$ is the probabilistic Hough vote. The original ISM defined voting elements as object center and object scale, computed from 2D coordinates and scales. Instead, we propose the following voting elements that can elegantly handle scale and rotation changes to the object.

$$\begin{bmatrix} x_{obj} \\ y_{obj} \end{bmatrix} = \begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} - \frac{s_{img}}{s_{db}} \times \begin{bmatrix} \cos\theta_{obj} & -\sin\theta_{obj} \\ \sin\theta_{obj} & \cos\theta_{obj} \end{bmatrix} \left(\begin{bmatrix} x_{db} \\ y_{db} \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} \right) \tag{7}$$

$$s_{obj} = s_{img}/s_{db} \tag{8}$$

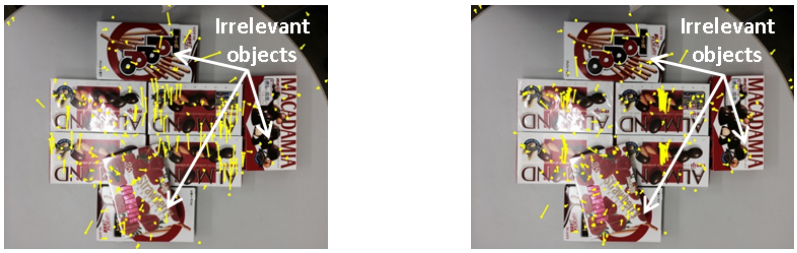
$$\theta_{obj} = \theta_{img} - \theta_{db} \tag{9}$$

Here, $x = (x_{obj}, y_{obj}, s_{obj}, \theta_{obj})$ is the proposed 4D scale and rotation invariant Hough vote for object center, scale and orientation. $(x_{img}, y_{img}, s_{img}, \theta_{img})$ and $(x_{db}, y_{db}, s_{db}, \theta_{db})$ are the 2D coordinates, scales and orientations for local features from query image and database, respectively. (x_c, y_c) are registered object centers (or any reference points) from the database images. Figure 2 (b) shows an example of the proposed Hough vote. In practice, since scale votes computed by taking ratio (Eq. (8)) are sensitive to even small changes in divisor, we thus take the logarithm of the values to convert the computation from division to the substraction.

In Eq. (6), the term $p(D_i|f_j)$ is the matching quality between feature f_j and database entry D_i. One way to define this probability is to assign matching score based on the feature distance. A more general treatment in object identification [5, 9] is to perform exhaustive search between features f_j of query image and each database image, then find the closest D_i in the feature space for further processing. This is equivalent to assigning $p(D_i|f_j)$ to 1 if D_i is matched to f_j, and otherwise to 0.

Finally, term $p(O|D_i)$ represents the confidence of making inference of object O when observed D_i. One way is assigning term frequency inverse document frequency (tf-idf)[13, 14] to this probability. For simplicity, we assume local features have the same chances to be observed in each database object, thus we assign $1/M$ to this probability, where M is the number of database objects.

Here, we would like to compare our method with that in [9]. In [9], a Hough transform based method is mentioned in which object location, scale and orientation are used in Hough voting. Though the calculation of location vote is not clearly specified, as written in [15], the location is computed as the difference of 2D coordinates of keypoints. This means that the estimated location is easily affected by scale change and rotation of the object, thus the estimated locations from each keypoint will scatter in the voting space. For multiple object identification, especially when repeated instances are close together, this scattering will cause excessive overlap in the Hough space, making it extremely difficult to identify and locate each object instance (shown in Fig.2 (a)). In our method, we



(a)Hough votes computed in [9] (b)Hough votes computed by this paper

Database image

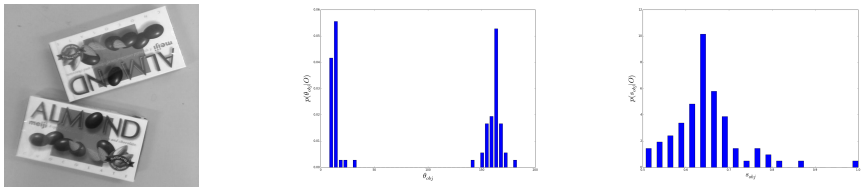
Fig. 2. Example of 4D Hough votes proposed in [9] (a) and in this paper (b) when instances in query images have different rotation and scales compared to objects in database images. The root position of the arrow represents the object location (object center), direction of the arrow represents object orientation and the length of the arrow represents object scales. Notice that our Hough votes shown in (b) are clustered in object centers, and scale and orientation are consistent.

instead estimate the object center position, which is calculated by using keypoint scale and orientation in addition to the 2D coordinates (in Eq. (7)), so that it is invariant to the scale change and rotation of the object. Thus the estimated object center position is much more consistent among keypoints, forming a much tighter cluster in the voting space (shown in Fig.2 (b)). This is very effective for multiple object identification, since it can help to accurately identify and locate each object instance.

It is also important to compare our model with other multiple object identification methods in [1, 3, 5]. In those methods, only the coordinates of object center computed by Eq. (7) are used for Hough voting, whereas in our method we additionally use scale s_{obj} and orientation θ_{obj} of the object. In order to illustrate the difference, we denote object centers as (x_{obj}, y_{obj}), object scale as s_{obj}, object orientation as θ_{obj}. Since object center, scale and orientation are independent, the joint distribution of object centers, scale and orientation can be written as,

$$p(O, x_{obj}, y_{obj}, s_{obj}, \theta_{obj}) = p(O, x_{obj}, y_{obj})p(O, s_{obj})p(O, \theta_{obj}) \tag{10}$$

The term $p(O, x_{obj}, y_{obj}, s_{obj}, \theta_{obj})$ on the left-hand side is our proposed joint distribution for Hough voting containing object centers, scale and orientation, and the term $p(O, x_{obj}, y_{obj})$ on the right-hand side is the object center vote used in the conventional methods. Therefore, by modeling the joint distribution of



(a) Query image (b) Orientation distribution (degree) (c) Scale distribution

Fig. 3. Example of scale distribution and orientation distribution for multiple object identification.

object centers, scale and orientation instead of object centers alone, our method can be seen as assigning weights to object centers by the distribution of object orientation $p(O, \theta_{obj})$ and scale $p(O, s_{obj})$, while conventional methods implicitly assume uniform distribution of scale and orientation. Figure 3 shows an example of distribution of the orientation and scale computed by Eqs. (8) and (9) from a query image. It shows that the distribution of scale and orientation exhibit bell shape property (peaked at the object scales and orientation). Thus, this distribution can help to perform more accurate Hough voting.

3 Multiple object instance detection and identification

3.1 Overview

In this section, we explain how to apply the model of SRIISM to detect and identify multiple object instances in query images. Figure 4 shows the block chart of the proposed system to detect multiple objects. For object images in database, we extract local features and create feature entries D_i, including local descriptors, scale and orientation extracted at keypoints. For each object, we also save its object center position (x_c, y_c) to the database.

For the query image, local features are first extracted and matched with features of the database. After feature matching, scale and rotation invariant Hough voting are carried out in the 4D space (Algorithm 1). Then a fast 4D bin search is employed and object pose (represented by bounding box) are recovered using maximum likelihood estimation. Finally, post processing is performed by which over detection is removed.

3.2 Scale and rotation invariant Hough voting

Algorithm 1 illustrates how to compute the $p(O, x|f_j, l_j)$ in Eq. (3). After feature matching, each matched feature pair votes for the possible object center, scale and orientation of the objects. Since the probability should be summed to one, we assign the term $p(x|O, D_i, l_j)$ to $1/N_f$, where N_f is the number of local features in the query image.

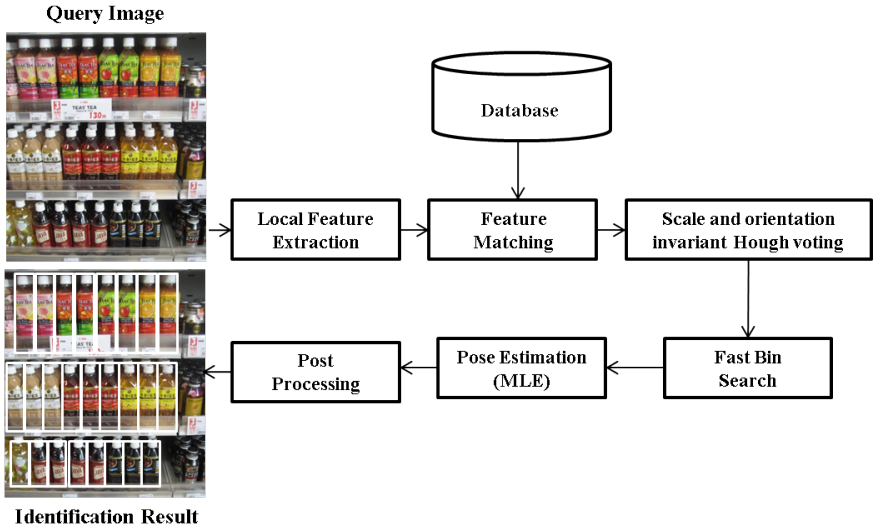


Fig. 4. Overview of the proposed algorithm.

Algorithm 1 4D Hough voting

for all features in query image (f_j, l_j) **do**
  for all feature entries in database D_i **do** //compute scale and rotation invariant vote according to Eqs. (7),(8),(9)
    $x \leftarrow (x_{obj}, y_{obj}, s_{obj}, \theta_{obj})$
    $p(x|O, D_i, l_j) \leftarrow 1/N_f$ //N_f is the number of features in query images
    $p(O, x|f_j, l_j) \leftarrow p(x|O, D_i, l_j)p(D_i|f_j)p(O|D_i)$.
  end for
end for

3.3 Fast 4D bin search

After Hough voting is carried out, we then detect the possible position (represented by object center) and associated pose of the object. Conventionally for this purpose, methods such as in [1] apply density estimation methods such as mean-shift to locate object instances and recover their poses. Nevertheless, these methods using mean-shift are not only time-consuming (takes $O(n^2)$, where n is the number of feature matches), but also can only recover at most four degree-of-freedom approximation of the object pose (position, scale and orientation). This is not enough for applications requiring more accurate object pose such as robotic vision. Therefore, we propose a fast 4D bin search method (takes $O(n)$) to first locate objects in the Hough space and then apply maximum likelihood estimation directly on feature matches to recover affine or higher order poses (6 degree-of-freedom or more) of the object.

In order to find the possible feature matches of objects, we divide Hough space into 4D bins, namely, for object centers (x_{obj}, y_{obj}), scale s_{obj} and orientation

θ_{obj}. Then we select bins that have scores larger than a threshold. Here, the scores of bins are defined by the following equation,

$$S(O,k) = \sum_{x_m \in V(k)} p(O, x_m) = \sum_{x_m \in V(k)} \sum_{j} p(O, x_m | f_j, l_j) \tag{11}$$

where, $S(O,k)$ means the score of the k-th bin for object O. Simply put, the score for each bin is the summation of Hough votes falling into it ($x_m \in V(k)$).

In practice, we found that it is inefficient to use 4D array to store bin votes when carrying out Hough voting. Because 4D array contains large number of bins (e.g. typically containing several million bins), it takes a lot of time in the final step to search for the candidate bins that are above the threshold. But in fact, there is only limited number of matches producing Hough votes compared to the large number of bins, thus most bins are empty. Using this property, we employ map structure (associative array) to store only non-empty bins. Specifically, when a 4D Hough vote is computed from a feature match, it is quantized and converted to a unique key representing the index of the bin. Then the probabilistic votes associated with the key is incremented if the entry for the key exists in the associative array, otherwise the entry for the key is created with an initial vote. Finally, our method iterates through all the entries and those bins with votes above threshold are selected. This implementation performs much faster than using 4D array since it does not search through empty bins.

In the experiment, bins for object centers (in pixel), orientation (in radians) and scale (ratio in logarithmic scale) are equally partitioned. Furthermore, in order to reduce the influence caused by quantization errors, we allow those bins to be overlapped (e.g. 50%).

3.4 Object pose estimation

Once candidate bins are selected based on the bin scores, we estimate the 6 degree-of-freedom (or more) object pose by maximum likelihood estimation (MLE).

Our method uses the local feature coordinates of feature matches belonging to selected bins. We denote the coordinate of a feature match of query and database by $t = [x_t, y_t, 1]$ and $q = [x_q, y_q, 1]$. Then their relationship, given an affine model $A \in R^{3\times3}$ can be expressed as:

$$t = Aq + \xi \ . \tag{12}$$

Here ξ is the error term due to the image noise. We assume that ξ has the form of Gaussian distribution with zero mean and variance of σ^2, that is, $\xi \sim N(0, \sigma^2)$, we can then write the likelihood term for t given q and A,

$$p(t|q, A, \sigma) \sim N(Aq, \sigma^2) \tag{13}$$

which is also a Gaussian distribution with the mean Aq and variance σ^2.

Given all the match pairs $(t_n, q_n)_{n=1,2,3,...}$ in a bin, we can apply maximum likelihood estimation to find out the affine pose for the objects.

$$A = \arg\max_A \prod_n p(t_n|q_n, A, \sigma^2) \tag{14}$$

In order to solve Eq. (14), we can take the logarithm of the likelihood function and reformulate it to an equivalent form:

$$A = \arg\min_A \sum_n \|t_n - Aq_n\|^2 \tag{15}$$

Then Eq. (15) can be easily solved using least square solver.

In practice, we found that when the threshold for the bin score is set low, the bin may contain votes from false matches. Therefore, the estimated results are not the correct object poses. In order to solve this problem, we apply additional verification method for bins containing few Hough votes.

Assume affine pose A are estimated, it can be decomposed into $A = TrR_2SR_1$ using SVD [16], where Tr,R_i and S are translation, rotation and scale matrix. We then compare the product of scale and rotation matrix ($Q = R_2SR_1$) from affine model with the value of $\overline{s}_{obj}R(\overline{\theta}_{obj})$, in which $\overline{s}_{obj}$ and $\overline{\theta}_{obj}$ are average scale and orientation of Hough votes in each grid and $R(\cdot)$ is the rotation matrix. When the elements of two matrix are not agreed to an extent, we re-set the score of $S(O, k)$ to 0. The mathematical explanation of this process is to add a prior term to A (i.e. $p(A)$) in MLE according to the evidence from Hough votes.

3.5 Post processing

Finally, we annotate detected objects by projecting bounding boxes using estimated affine model. Since nearby bins for the same object produce overlapping bounding box, we keep the one with the maximum bin score if they are overlapped. In order to compute the overlapped area, we employ Sutherland-Hodgman algorithm [17] to find out corresponding vertices of the overlapping polygons and then apply cross product to compute corresponding areas.

4 Experiment

In order to evaluate the proposed method, we reproduced experiment from related papers [1, 3, 5] and added real world datasets taken from supermarket and convenience stores.

The first dataset (shown in Fig.5) simulates pick-and-place applications for industrial automation. We evaluated our method for repeated instance detection tasks similar to [3], where repeated instances have different rotation and scale.

(a)High texture objects (b)Medium texture objects (c)Low texture objects

Fig. 5. Examples of objects used in repeated instance dataset (dataset 1).

Fig. 6. Examples of cluttered environment dataset (dataset 2).

Furthermore, the datasets also contain partial occlusions. In order to test the performance to the objects with various texture levels, we divided objects into three sets according to the texture level they have (high texture, medium texture, and low texture). In this experiment, we collected 87 query images (512×384 pixels) with 556 instances in all. In such tasks (e.g. industrial automation), false positive (i.e. over detection or false detection) rate must be kept low. Therefore, we evaluated the detection rate (recall) under the condition that the precision is 100% (by chosing proper threhold during Hough voting), and compared with conventional methods.

In the second dataset (shown in Fig.6), we reproduced the task of [1, 5], in which multiple objects (include repreated instances) are to be detected in a cluttered environment. The dataset also includes a collection of challenging real world images taken from the super market (see Fig.6). In order to test the robustness to the perspective changes for our proposed method, we took query images at different perspectives. We also included occlusions in the query images. The database contains 221 objects. For the query images, there are 28 images in total, which the width and height are ranging from 2000 to 3000 pixels, and the task is to identify total of 502 objects (targets) from over a thousand of objects (irrelevent objects).

4.1 Experiment settings

In all experiments, we compared the proposed method with conventional methods using mean-shift (denoted as 2D mean-shift) [1, 3] and grid voting (denoted as 2D gridvoting+RANSAC) [5] on object centers. Moreover, in order to show that our method is independent to local descriptors used, we implemented and tested our method using SIFT and BRIGHT [18], which is a binary local descriptor used in [5]. In order to count the true positive, we applied 50% overlapping criterion, that is, for correct detection (true positive), its bounding box should be at least 50% overlapped with that of ground truth. In addition, we also pose constraints when counting true positives, that is, scale and rotation of the bounding box should be consistent with ground truth.

The experiment has been conducted on Windows7 PC with Core i7-2700K CPU@3.50GHz.

4.2 Experiment Results

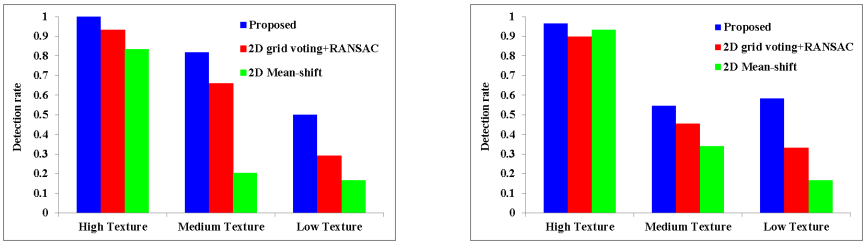
Figure 7 shows detection results for the repeated instance detection tasks (dataset 1). It shows that the proposed method outperforms object center based methods for all three types of objects (high texture, medium texture and low texture). The result also shows that our method in overall achieves better performance both for SIFT and BRIGHT. Especially, our method outperformed object center based methods by 30% for low texture objects. This is because while objects with low texture generate only few correct feature matches, they are easily contaminated by the false matches and resulted in detection failures when applying conventional methods.

Figure 8 shows results on the cluttered environment dataset (dataset 2). In order to compare our method with conventional methods, we compute the recall and precision rate for all three methods (proposed, 2D mean-shift and 2D grid voting+RANSAC). It shows that our method has the best performance among all three methods. At 95% precision rate, our methods is 5% better in recall rates compared to 2D grid voting methods both for SIFT and BRIGHT as local descriptor.

Figure 9 shows the average processing time (matching with one database image) between the proposed method and the conventional methods using object centers with mean-shift and grid voting [5] on the cluttered environment dataset (dataset 2). It shows that proposed method in total works twice as fast as that of conventional methods, and six to senven times faster for the process after feature matching. This is because our 4D Hough voting is robust to false matches so we apply non-iterative affine estimation combined with a geometric consistency check in the final step, while methods such as [5] have to iteratively apply RANSAC to remove outliers.

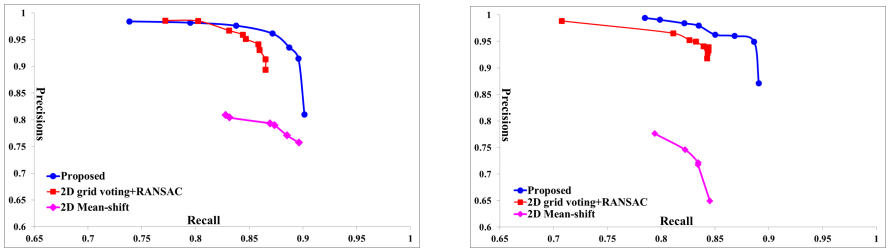
5 CONCLUSION

We proposed a Scale and Rotation Invariant Implicit Shape Model (SRIISM), and developed a local feature matching based system using the model to ac-



(a)SIFT (b)BRIGHT

Fig. 7. Results on the repeated instance detection (dataset 1).



(a)SIFT (b)BRIGHT

Fig. 8. Results on the cluttered environment dataset (dataset 2).

curately locate and identify large numbers of object instances in an image. In the proposed SRIISM, we model the joint distribution of object centers, scale, and orientation computed from local feature matches in Hough voting, which is not only invariant to scale changes and rotation of objects, but also robust to false feature matches. For the multiple object instance identification system using SRIISM, we apply a fast 4D bin search method in Hough space with complexity $O(n)$, where n is the number of feature matches, in order to segment and locate each instance. Futhermore, we apply maximum likelihood estimation (MLE) for accurate object pose detection. In the evaluation, we created datasets simulating various industrial applications such as pick-and-place and inventory management. Experiment results on the datasets showed that our method outperforms conventional methods in both accuracy (5%-30% gain) and speed (2x speed up). In the future works, we will extend our research to the non-rigid object identification by considering more flexible local transformation models.

References

1. Zickler, S., Veloso, M.: Detection and localization of multiple objects. In: Humanoid Robots, 2006 6th IEEE-RAS International Conference on. (2006) 20–25

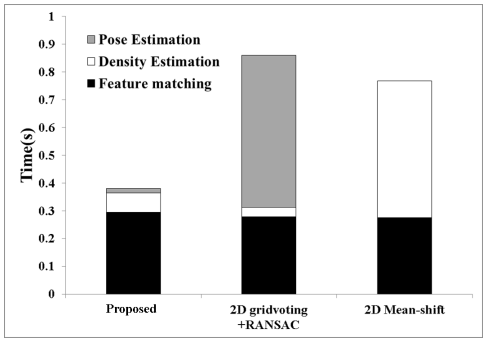


Fig. 9. Matching time between proposed method and conventional methods (dataset 2).

2. Collet, A., Martinez, M., Srinivasa, S.S.: The moped framework: Object recognition and pose estimation for manipulation. The International Journal of Robotics Research (2011) 0278364911401765
3. Piccinini, P., Prati, A., Cucchiara, R.: Real-time object detection and localization with sift-based clustering. Image and Vision Computing **30** (2012) 573 – 587
4. Lin, F.E., Kuo, Y.H., Hsu, W.H.: Multiple object localization by context-aware adaptive window search and search-based object recognition. In: Proceedings of the 19th ACM International Conference on Multimedia. MM '11, New York, NY, USA, ACM (2011) 1021–1024
5. Higa, K., Iwamoto, K., Nomura, T.: Multiple object identification using grid voting of object center estimated from keypoint matches. In: Image Processing (ICIP), 2013 20th IEEE International Conference on. (2013) 2973–2977
6. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. International journal of computer vision **77** (2008) 259–289
7. Liu, M.Y., Tuzel, O., Veeraraghavan, A., Chellappa, R.: Fast directional chamfer matching. 2013 IEEE Conference on Computer Vision and Pattern Recognition **0** (2010) 1696–1703
8. Barinova, O., Lempitsky, V., Kholi, P.: On detection of multiple object instances using hough transforms. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 1773–1784
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
10. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110** (2008) 346 – 359
11. Wu, C.C., Kuo, Y.H., Hsu, W.: 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, New York, NY, USA, ACM (2012) 969–972
12. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 1038–1045

13. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE (2003) 1470–1477
14. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2911–2918
15. Perona, P.: David lowe's recognition system (2004)
16. Korman, S., Reichman, D., Tsur, G., Avidan, S.: Fast-match: Fast affine template matching. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 1940–1947
17. Sutherland, I.E., Hodgman, G.W.: Reentrant polygon clipping. Commun. ACM **17** (1974) 32–42
18. Iwamoto, K., Mase, R., Nomura, T.: Bright: A scalable and compact binary descriptor for low-latency and high accuracy object identification. In: Image Processing (ICIP), 2013 20th IEEE International Conference on. (2013) 2915–2919