



Fast Circular Object Localization and Pose Estimation for Robotic Bin Picking

Linyao Luo¹, Yanfei Luo², Hong Lu¹, Haowei Yuan³, Xuehua Tang³,
and Wenqiang Zhang²(✉)

¹ Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University,
Shanghai, People's Republic of China

² Shanghai Engineering Research Center for Video Technology and System,
School of Computer Science, Fudan University,
Shanghai, People's Republic of China
wqzhang@fudan.edu.cn

³ Shanghai Electric Group CO., LTD. Central Academe, Shanghai, China

Abstract. Detecting and localizing objects in three-dimensional space is essential for robotic manipulation. One practical task is known as “bin-picking”, where a robot manipulator picks objects from a bin of parts without any assistance of an operator. For such a task, vision-based object detection and location can be a cost-effective solution. In this paper, we propose a fast and robust approach for picking flanges in a crowd condition. We present a continuous edge detector improved from Canny and a fast ellipse detector based on randomized hough transformation to obtain the outer contours of flange. And then we have implement several picking experiments to verify our proposed approach is fast and robust in practical environment.

Keywords: Bin-picking · Mono vision · Randomized hough transform

1 Introduction

One of the key challenge in highly automated robot-aided manufacturing is the capability to automatically identify and locate parts, thus the robot can grasp and manipulate them in an accurate and reliable way. In general, parts are randomly placed inside a bin or in a conveyor belt, so one needs sophisticated perception systems to identify and precisely locate the searched objects. Usually, this perception task is referred as the “bin-picking” problem, and it has been widely studied in the last decades due to its strong impact in the flexibility and productivity for manufacturing companies.

Vision systems for recognition and localization of objects, based on standard cameras and 2D image analysis, have been widely used in industrial automation for many years. A vision-based recognition system for planar object has been

proposed in [1], where a set of invariant features based on geometric primitives of the object boundary are extracted from a single image and matched against a library of invariant features computed from the searched objects models, generating a set of recognition hypothesis. Hypothesis are then merged and verified to reject false recognition hypothesis. In [2], Rahardja and Kosaka presented a stereo vision-based bin-picking system that, starting from a set of model features selected by an operator, search for easy to find “seed” features (usually large holes) to roughly locate the searched objects, and then look for other, usually small, “supporting” features used to disambiguate and refine the localization. In [3], the Generalized Hough Transform (GHT) is used for 3D localization of planar objects, the computational complexity of the GHT is here reduced by uncoupling parameter detection. Shroff et al. [4] presented a vision-based system for specular object detection and pose estimation: authors detect a set of edge features of the specular objects using a multi-flash camera that highlights high curvature regions, a multi-view approach is exploited to compute the pose of the searched object by triangulating the extracted features. An overview of general vision-based object recognition and localization techniques can be found in [5], along with a performance evaluation of many types of visual local descriptors used for 6 DoF pose estimation.

2 Target Location

A large number of industrial parts are almost circular shapes like flanges, thus we will focus on perform an experiment on the flanges. In following sections, we will explain our core algorithms of the mono vision system in several subsection: edge detection, ellipse extraction and pose refinement.

2.1 Edge Detection

Traditional edge detectors like Canny [6], Sobel can extract edge pixels, but meanwhile include much noise. As an object contour is usually continuous, we propose a method of fast continuous edge detection that divides into three steps: compute gradient, find candidate points and extract continuous edges.

The first step is to compute gradient image. The gradient of each pixel is computed as the same algorithm as Canny. However gradient directions are divided into 4 major directions that denote as $C1, C2, C3, C4$, because we do not carry about the accurate gradient direction. The regions of 4 major direction are defined as

$$C1\left\{\left(-\frac{\pi}{8}, \frac{\pi}{8}\right) \cup \left(\frac{7\pi}{8}, \frac{9\pi}{8}\right)\right\}, C2\left\{\left(\frac{\pi}{8}, \frac{3\pi}{8}\right) \cup \left(\frac{9\pi}{8}, \frac{11\pi}{8}\right)\right\}, \\ C3\left\{\left(\frac{3\pi}{8}, \frac{5\pi}{8}\right) \cup \left(\frac{11\pi}{8}, \frac{13\pi}{8}\right)\right\}, C4\left\{\left(\frac{5\pi}{8}, \frac{7\pi}{8}\right) \cup \left(\frac{13\pi}{8}, \frac{15\pi}{8}\right)\right\}$$

Then candidate points need to be found in this step. In order to detect continuous edge, we start with the candidate points. Since candidate points

regarded as seeds and extend to a whole edge, we expect that the distribution of these candidate points is dispersed.

As a edge pixel has prominent value in the gradient image, we extract the likely candidates based on its gradient value. However considering the effect of illumination variety, we adopt the local maximum gradient searching to find candidates. A pixel will be brought into candidate point set if it has the local maximum gradient value in $k \times k$ neighborhood. The choice of k depends on the object distribution density that high value k will result in less candidates and sparse distribution, and low value k in more noise. Therefore when we use a high value k to search candidate points, we add a likely local maximum strategy that if the local maximum p_{max} of a $k \times k$ patch is not in the $\frac{k}{2} \times \frac{k}{2}$ neighborhood of the second largest point p_{sec} , the p_{sec} will be involved in candidate set (Fig. 1).

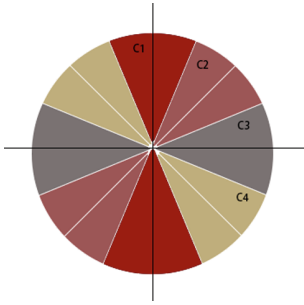


Fig. 1. Regions $\{C_1, C_2, C_3, C_4\}$ represent 4 major gradient directions respectively.

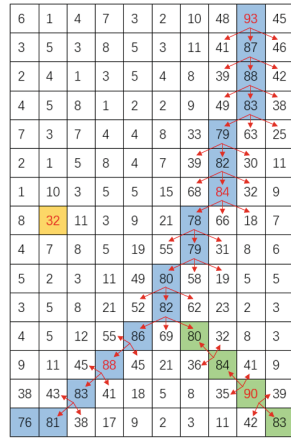


Fig. 2. Process of continuous edge detection.

After candidate points obtained, we start at these points to implement continuous edge extraction. Above all, pixels on a continuous edge are satisfied the following conditions: adjacent in vertical direction of gradient (adopt 8-neighbor judgement) gradient values are quite close; gradient directions are quite close. The detection process is shown in Fig. 2. and result in Fig. 3. Obviously, our proposed continuous edge detection approach includes less noise than Canny detector as shown in Fig. 3.

2.2 Hough-Based Ellipse Extraction

As we know that 5 points determine a ellipse in a plane. That means the time complexity of extracting a ellipse from n points is $O(n^5)$ when implementing randomized hough transform (RHT_5) in [7]. In the crowd industrial environment, the process of RHT_5 is time-consuming in randomly sampling 5 points.

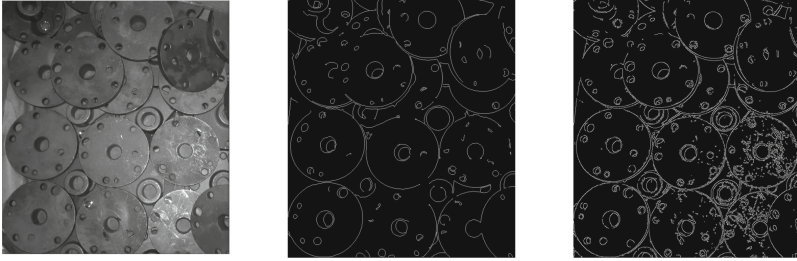


Fig. 3. Edge image. Left: original image. Middle: edge image by our proposed approach. Right: classical Canny edge detection.

A great many invalid samples and accumulations included makes the algorithm poor performance even almost fail in limit time.

For the reasons given above, we propose a improved RHT with 3 points. First, we get a long axis of ellipse L_a determined with 2 points p_1, p_2 that is randomly chosen from edge point set V . The center O of ellipse, long radius r_a and inclination angle θ_a can be computed as

$$O = \left(\frac{p_1 + p_2}{2}\right), \tag{1}$$

$$r_a = \frac{\|p_1 - p_2\|_2}{2}, \tag{2}$$

$$\theta_a = \tan^{-1}\left(\frac{p_1^x - p_2^x}{p_1^y - p_2^y}\right), \tag{3}$$

Second, the sum of distances between p_3 and focuses f_1, f_2 is equal to the length of long axis, then we have

$$\|p_3 - f_1\|_2 + \|p_3 - f_2\|_2 = 2r_a. \tag{4}$$

We can get the focus coordinates

$$f_1^x = O_x - \cos |\theta| \sqrt{r_a^2 - r_b^2} \tag{5}$$

$$f_1^y = O_y - \sin |\theta| \sqrt{r_a^2 - r_b^2} \tag{6}$$

$$f_2^x = O_x + \cos |\theta| \sqrt{r_a^2 - r_b^2} \tag{7}$$

$$f_2^y = O_y + \sin |\theta| \sqrt{r_a^2 - r_b^2}. \tag{8}$$

The short radius r_b can also be obtained as

$$r_b = \sqrt{\frac{r_a^2 \delta^2 - r_a^2 \gamma^2}{r_a^2 \gamma^2}} \tag{10}$$

where

$$\delta = \|O - p_3\|_2, \quad \gamma = \sin |\theta|(O_y - p_3^y) + \sin |\theta|(O_x - p_3^x). \quad (11)$$

At last step, after collecting all parameters that an ellipse needed $\{O, r_a, r_b, \theta\}$, we set an accumulator to count how many points $p_i \in V$ fit the ellipse we obtained. It will be accepted as a valid ellipse when the count of points exceeds a threshold n_{thresh} . In practical experiment, we get rid of some too long or too short long radius r_a in first step, in order to accelerate the process. The pseudo-code of RHT_3 can be described below.

Algorithm 1. RHT_3 Ellipse

```

1: Point Set  $V\{p_i\}, i = 1, 2, \dots, N$ 
2: for  $p_1, p_2$  in  $V$  do
3:   if  $Flag[p_1] = 1$  or  $Flag[p_2] = 1$  then
4:     continue
5:   end if
6:   if  $Distance(p_1, p_2) > max_a$  or  $Distance(p_1, p_2) < min_a$  then
7:     continue
8:   end if
9:   compute  $\{O, a, \theta\}$ 
10:  init accumulator  $Acc := 0$ , perimeter  $Pe := 0$ 
11:  for  $p_3$  in  $V$  do
12:    if  $Flag[p_3] = 1$  then
13:      continue
14:    end if
15:     $d := Distance(p_3, O)$ 
16:    if  $d > a$  then
17:      continue
18:    end if
19:    compute  $b$ 
20:     $Acc[b] := Acc[b] + 1$ 
21:  end for
22:   $b_{max\_acc} := argmax(Acc)$ 
23:  if  $Acc[b_{max\_acc}] / Pe[b_{max\_acc}] > min\_vote\_rate$  then
24:    obtain ellipse  $EP\{O, a, b_{max\_acc}, \theta\}$ 
25:    for  $p_t$  on  $EP$  do
26:       $Flag[p_t] := 1$ 
27:    end for
28:  end if
29: end for

```

2.3 Pose Refinement

In this section, we show how the pose will be estimated with the ellipse function and how to make the pose more accurate.

Euler Angle. In this paper, we use euler angle to describe object’s 3D pose. The image coordinate system is defined that top-left corner used as origin, right direct as X axis, down as Y axis and inside as Z axis. A object pose is consist of positions $\{Pos_x, Pos_y, Pos_z\}$ and rotations $\{Rot_x, Rot_y, Rot_z\}$. However, we ignore the Z-axis rotation Rot_z in our experiment because it has no effect on picking step, and the Pos_z can only be computed in calibration. Therefore, in the section, we only need to obtain the positions $\{Pos_x, Pos_y\}$ and rotations $\{Rot_x, Rot_y\}$. We define the order of rotation about axis as X, Y, Z. The euler angle [8] can be calculated as below, and we will not show the detail derivation process.

$$\{Pos_x, Pos_y\} = O \tag{12}$$

$$Rot_x = \cos^{-1} \frac{c}{a} \tag{13}$$

$$Rot_y = \cos^{-1} \frac{d}{\sqrt{(d \sin \alpha)^2 + (a \cos \alpha)^2}} \tag{14}$$

In above formula, a, b, O is respectively the long radius, short radius and center of an ellipse, and c, d is the Y-intercept and X-intercept.

Mirror Problem. Obviously, the outer contour of flange is always symmetric, thus we encounter the mirror problem that we are not able to distinguish the correct rotation direct from the mirror direct (as shown in Fig. 4.). In term of this issue, we propose a method to recognize the correct rotation direct, which can also improve the accuracy of fitting the flange for the ellipse. We find noisy points focuses on one side of ellipse in a Canny edge image with a low threshold as shown in Fig. 5., because of the flange thickness effect. We check noise distribution of each $\epsilon \times \epsilon$ patch centered by the point in outer contour, and then regard those points with top 25%-35% density of noise distribution as the outliers.

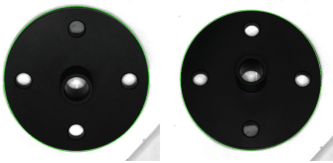


Fig. 4. Mirror condition. We find no difference between the out contours of left and right image because of symmetrical geometry.

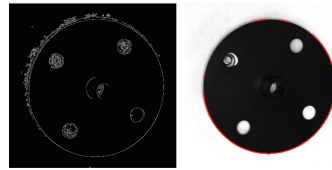


Fig. 5. Noise distribution. Noise points are always converge in one side where the flange is blocked up. Red points are the correct points we obtained.

Actually our method not only imply which rotation direction is accord with the fact, but also make the step of ellipse fitting more accuracy when discarded outliers. The front-view contour and side-view contour are shown in Fig. 6.



Fig. 6. Actual outline. Red edge is the actual flange front outline, and green edge is a side-view contour. (Color figure online)

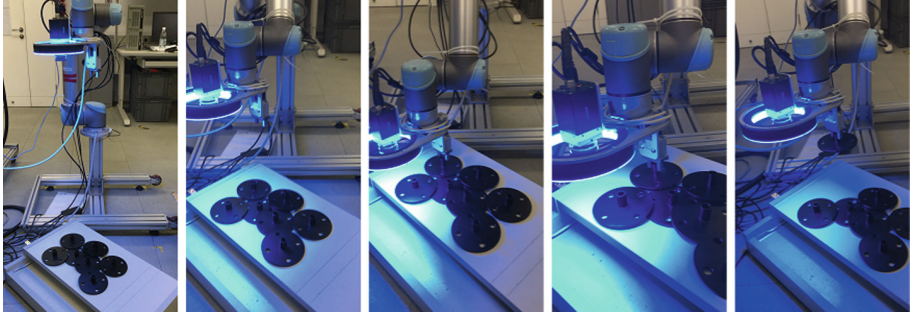


Fig. 7. Picking process

3 Experiment

3.1 Strategy

For the sake of accurate picking, a flange will be always located twice. For each flange we implement RHT_3 on the first image to obtain a rough position, on which camera will be moved. We stop the camera just above the flange, and then take another image for pose refinement. The strategy is showed in the following steps (Figs. 7 and 8).

1. Take the first image I_0 . Implement continuous edge detector (Sect. 2.1) and RHT_3 (Sect. 2.1) to find all ellipse in I_0 . The ellipse E_1 with most integrated contour will be picked next, and the center position C_1 is obtained;
2. Move camera to C_1 just above E_1 ;
3. Take another image I_1 , and compute refined pose Pos_1 of E_1 by using the method proposed in Sect. 2.3, and meanwhile find the rough position C_2 of next flange.
4. Pick up the flange on pose Pos_1 by robot manipulator;
5. If the next flange not found, stop picking process. Otherwise, C_2 will be regarded as C_1 , and then go to Step 2.

3.2 Experimental Result

In our experiment, we have test the proposed algorithms in these environments: single target and multi-targets randomly placed.

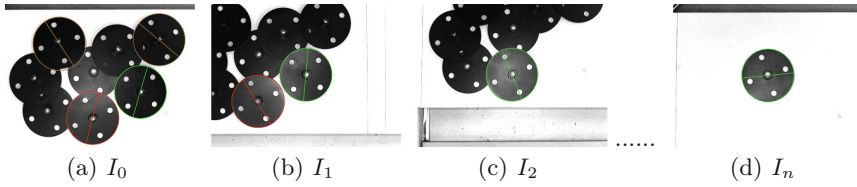


Fig. 8. Camera view

In single target test, we elevate one side of a flange deliberately with some specific angles, in order to test the accuracy of pose refinement. It is shown in Table 1. that the translation error is almost less than 2 mm and the angle error is less than 3.5°. Specially, we find a small rotation angle will result in a quite big error by pose estimation. This is because $\cos^{-1} \theta$ function is steep decrease around $\theta = 1$, and we have used \cos^{-1} to calculate Rot_x, Rot_y . However, it does not affect our picking performance, since we can pick it up as well by regarding a small angle as zero.

Table 1. Single target error

Angle	Translation error		Angle error	
	X (mm)	Y (mm)	X (deg)	Y (deg)
4.50	0.51	0.48	3.51	2.88
9.42	1.78	1.34	2.17	2.24
13.29	1.65	1.76	1.05	1.27
17.42	1.82	1.20	1.17	0.74
21.54	1.78	1.76	1.06	0.85
Mean	1.51	1.31	1.79	1.48

In multi-target test, we place several flanges on platform at random, and then record successful times among 50 attempts of picking. In order to test one-time success rate, the robot will bring the flange back automatically to experiment platform after picking up. Of course, the returned position is almost randomized. We do each task 5 times and obtain the average number of successful picking times as shown in Table 2.

In addition, we test the performance of practical bin-picking task that picking all the flanges on platform with the strategy in Sect. 3.2. In this task, success rate of attempts and time consuming of algorithm will be recorded in Table 3. For each task, we also employ the average value of 5 times experiments.

Table 2. Multi-target attempts success rate

Task	Number of flanges	Times of attempts	Success rate (Canny + RHT_5)	Time (Canny + RHT_5)	Success rate (Our approach)	Time (Our approach)
1	5	50	94.0%	1320 ms	97.6%	337 ms
2	10	50	87.8%	8359 ms	93.2%	953 ms
3	15	50	74.6%	26170 ms	88.4%	2754 ms
4	20	50	-	Time out	81.2%	7713 ms

Table 3. Aattempts for Picking All the Flange

Task	Number of flanges	Times of picking
1	5	5.4
2	10	11.2
3	15	18.8
4	20	23.8

4 Conclusion

A mono vision system for picking crowded flanges has been presented in this paper. The core of the system is the location algorithm which is demonstrated to be robust, fast and accurate. At first we implement a continuous edge detection in order to suppress noise in preprocessing stage, and then put forward a RHT_3 approach to dramatically accelerate the process of ellipse extraction. At last subtly, we make advantage of noise distribution around edge points to solve the mirror problem and to further improve the accuracy of results.

Acknowledgement. This work was supported in part by National Natural Science Foundation of China (No. 81373555), and Shanghai Committee of Science and Technology (14JC1402200 and 14441904403) for funding.

References

1. Rothwell, C.A., Zisserman, A., Forsyth, D.A., Mundy, J.L.: Planar object recognition using projective shape representation. *Int. J. Comput. Vis.* **16**(1), 57–99 (1995)
2. Rahardja, K., Kosaka, A.: Vision-based bin-picking: recognition and localization of multiple complex objects using simple visual cues. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems 1996, IROS*, vol. 3, pp. 1448–1457 (1996)
3. Cozar, J.R., Guil, N., Zapata, E.L.: Detection of arbitrary planar shapes with 3D pose. *Image Vis. Comput.* **19**(14), 1057–1070 (2001)
4. Shroff, N., Taguchi, Y., Tuzel, O., Veeraraghavan, A.: Finding a needle in a specular haystack. In: *IEEE International Conference on Robotics and Automation*, pp. 5963–5970 (2011)

5. Viksten, F., Forssen, P.E., Johansson, B., Moe, A.: Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In: IEEE International Conference on Robotics and Automation, pp. 1139–1146 (2009)
6. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 679–698 (1986)
7. Inverso, S.: Ellipse detection using randomized hough transform. Final Project Introduction to Computer Vision (2006)
8. Slabaugh, G.G.: Computing Euler Angles from A Rotation Matrix (1999)