Pose-Estimation and Reorientation of Pistons for Robotic Bin-Picking

Jianhua Su, Zhi-Yong Liu, Rui Li and Hong Qiao

State Key Lab. of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences

Purpose - Picking up pistons in arbitrary poses is an important step on car engine assembly line. We usually use vision system to estimate the pose of the pistons and then guide a stable grasp. However, a piston in some poses, e.g., the mouth of the piston faces forwards, is hardly to be directly grasped by the gripper. Thus, we need to reorient the piston to achieve a desired pose, i.e., let its mouth face upwards, for grasping.

Design/methodology/approach - This paper aims to present a vision-based picking system that can grasp pistons in arbitrary poses. The whole picking process is divided into two stages. At localization stage, a hierarchical approach is proposed to estimate the piston's pose from image which usually involves both heavy noise and edge distortions. At grasping stage, several steps robotic manipulations are designed to enable the piston to follow a nominal trajectory to reach to the minimum of the distance between the piston's center and the support plane. That is, under the design input, the piston would be pushed to achieve a desired orientation.

Findings - A target piston in arbitrary poses would be picked up from the conveyor belt by the gripper with the aid of vision.

Practical implications -The designed robotic bin-picking system using vision is an advantage in terms of flexibility in car manufacturing industry.

Originality/value – The theoretical and experimental analyses for picking of piston is proposed.

Keywords - Vision-guided bin-picking; Ellipse detection algorithm; Reorient

Paper type - Research paper

1. Introduction

Flexible and high-speed handling or picking up of workpieces increases interests in low-cost robot grasping system. Such system requires low-cost grippers, faster sensing and reliable operations. In general, workpieces are randomly placed inside a bin or on a conveyor belt. Then, a robot has to precisely estimate the pose of object for grasping. Thus, the vision-based bin-picking has been widely studied due to it is considered as an innovative method to improve manufacturing process.

1.1 Related Work

The objective of the vision-based bin-picking is to identify and estimate the pose of workpieces with standard cameras images, and then select a suitable grasping configuration for manipulations. Li and Lee (Li and Lee, 1996) presented a vision guidance robot grasping system for food handling in which the visual information was formulated to detect the pose of the food and then guide a stable grasp. Calisle *et al.* (Calisle *et al.*, 1994) used the 2-D vision to detect the position and orientation of a workpiece on the conveyor, and then a three-pin pneumatic gripper was employed to reorient the part to achieve a desired pose. Sanz *et al.* (Sanz *et al.*, 2005) addressed the applications of lettuce processing and packing of plastic parts with a parallel-jaw gripper. They used the 2-D vision system to identify the shape of the object and then find the grasping region on the contour. Morales *et al.* (Morales *et al.*, 2006) designed a robot system to grasp non-modeled planar extruded objects. In their work, the image processing system extracted the contour of the object along with its location. Taylor *et al.* (Taylor *et al.*, 1994) retrieved the geometric information from the analysis of image. Then, they selected a more favorable grasp on the rim of the object, such as the apparent antipodal grasps on the rim. Macias and Wen (Macias and Wen, 2014) developed a vision guidance system for picking-up and stacking block, where they used binary markers to aid in block identification and localization. Pretto *et al.* (Pretto *et al.*, 2013) developed a bin-picking system to locate planar objects using a single camera. In their work, a voting scheme was presented to select the candidate object, and an iterative optimize-and-score procedure was designed to select the best match. Holz *et al.* (Holz *et al.*, 2014) investigated picking up objects with a mobile robot. In their work, the object models were learned from single scans in an offline phase, and then the objects could be detected in the scene using a probabilistic graph matching method. Liu *et al.* (Liu *et al.*, 2012) proposed a fast directional chamfer matching (FDCM) algorithm to identify and estimate the pose of an object for picking it up.

Dupuis *et al.* (Dupuis *et al.*, 2008) developed a vision guidance robot system for picking up of randomized con-rod, where a 3-D vision system was used to obtain a topographical map of the pile surface. In Ogawa's work (Ogawa *et al.*, 2014), a robot system with stereo vision is

developed for picking up coil springs, in which the highlights made by illumination were used to identify and estimate the pose of a target coil spring. Sansonia *et al.* (Sansonia *et al.*, 2014) developed a 3-D pattern matching method for picking up applications, where a fast camera acquired a blade projected by a laser source. Buchholz et al. (Buchholz *et al.*, 2013) discussed picking up arbitrary objects with a standard 3-D sensor, where the localization algorithm was an enhancement of existing surface based localization techniques. Domae *et al.* (Domae *et al.*, 2014) used a 3-D depth sensor for picking-and-placing of piled objects randomly placed in a bin, in which the grasping configuration was estimated on a single depth map.

1.2 The purpose of this work

Pistons are the key parts of a car engine, and robotic assembly of piston is a very challenge task in manufacturing. In our novelty developed piston assembly system (as described in Section 4, Figure 10), the pose of the target piston would be firstly identified and estimated from a pile of pistons randomly placed on the conveyor belt, and then be grasped and placed into the fixture.

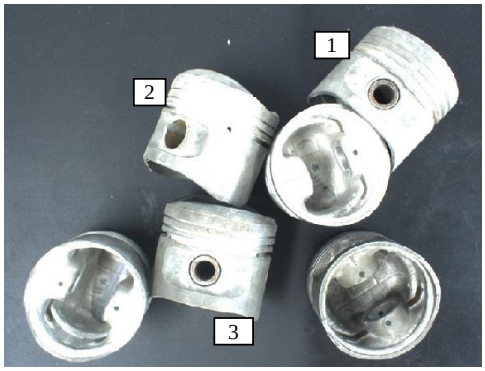


Figure 1: The six automobile pistons should be picked up by the gripper. The original image contains six pistons in which three of them, as marked 1,2,3, are placed facing forwards.

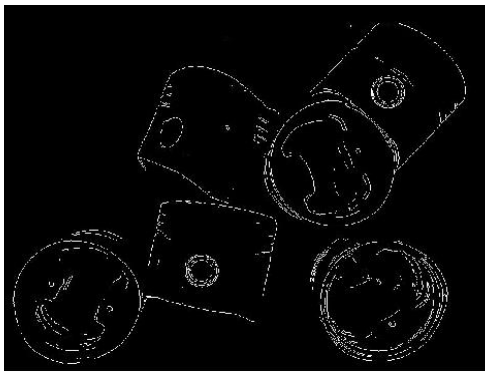


Figure 2: The edge map is obtained by sobel operator. In fact, detection of multiple ellipses in a real environment with heavy noise remains a very challenging task.

Recognition and pose estimation of the parts are mainly based on 2-D or 3-D vision techniques. The pose estimation with 2-D vision system is ideal for a planer part whose three dimensions are negligible. There are also some works on the pose estimation of 3-D parts with 2-D vision system. For example, based on real-time template recognition, Hinterstoisser *et al.* (Hinterstoisser *et al.*, 2012a) proposed an approach for real-time detection of texture-less objects, in which the templates can both be built and matched quickly. They (Hinterstoisser et al., 2012b) also discussed the 6 degrees-of-freedom pose estimation in real-time by using template-based LINEMOD approach and a Kinect. Rios-Cabrera and Tuytelaars (Rios-Cabrera and Tuytelaars, 2013) detected multiple specific 3D objects based on the LINEMOD template-based method in which they learned the template online and speed up the detection based on cascades. Brachmann *et al.* (Brachmann *et al.*, 2014) discuss the estimate of the 6D Pose of specific objects from a single RGB-D image. The key concept is a learned, intermediate representation in form of a dense 3D object coordinate labelling paired with a dense class labelling. Bonde *et al.* (Bonde *et al.*, 2014) presented a learning-based instance recognition framework from single view point clouds. They use a soft label Random Forestto learn discriminative shape features of an object and use them to classify both its location and pose.

In this work, we use 2-D vision to estimate the pose of the pistons by detecting the circular holes. However, it is a challenge work to identify the circular holes due to both heavy noise and edge distortions. And, the pistons under some poses, e.g., facing forwards as shown in Figure 1 (e.g., pistons-1,2,3), are hardly to be stretched by the two-pin gripper directly (we can only grasp the inner cavity of the piston by stretching actions as described in Figure 3). Thus, we need to reorient the piston to achieve a desired pose, i.e., facing upwards, for picking.

Similar to our work are Rao's (Rao and Goldberg, 1995) and Calisle's (Carlisle *et al.*, 1994). They utilized vision information to determine the initial pose of a polygonal part and used pneumatic gripper to reorient polygonal parts to achieve desired poses. However, there are some differences between theirs and ours:

a) They assumed that each part was dropped onto the conveyor belt in isolation, but we discuss the multiple parts are randomly placed on the conveyor belt.

b) Their squeeze-grasp actions to reorient a polygon were realized in a two-dimensional space (to eliminate the uncertainties of one rotational angle), but our planning actions are designed in a three-dimensional space to eliminate the uncertainties of the pose of the piston.

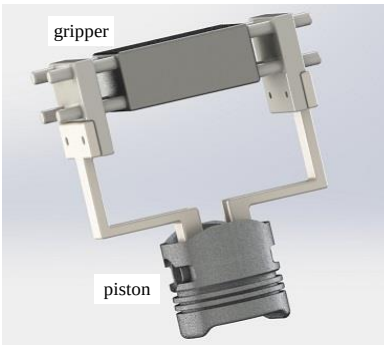


Figure 3: The piston should be squeezed in the inner cavity to ensure the gripper should not damage the surface of the piston.

In summary, there are two difficulties for picking up the pistons.

a) A target piston is needed to be searched from multiple parts placed on the conveyor belt, as shown in Figure 2. The circular holes of the pistons are projected as ellipses in the image. But, it is hard to identify the circular holes due to both heavy noise and edge distortions.

b) Some pistons possibly face to forwards (as shown in Figure 1) on the conveyor belt; hence, it is difficult to grasp those pistons (e.g., the Piston-1, 2, 3 in the Figure 1). Therefore, we should use the pneumatic gripper to manipulate those parts to achieve the desired pose, i.e., let it face upwards.

In this work, we develop a bin-picking system that allows picking up pistons in arbitrary poses. The 2-D vision information is used to estimate the rough pose of a part; then, the gripper reorients the part to achieve the desired pose and picks it up.

The major contributions of this paper are:

- We develop a methodology that uses a pneumatic gripper and 2-D vision information for picking multiple pistons under arbitrary poses.
- The rough pose of the parts, which are in the environment that usually involve edge distortions, are estimated based on a hierarchical approach
- The pose uncertainties of the pistons are eliminated by several steps robotic manipulations.

The rest of the paper is arranged as follows: In section 2, we introduce the proposed hierarchical approach for pose-estimation of the pistons, and then the grasping strategy is discussed in Section 3. In Section 4, the experiments of vision-based picking of piston are given to show the efficiency of the proposed strategy.

2. Identification and Pose-estimation of Pistons

The identification of ellipses generally includes the preprocessing of edge detection in the original image and then followed by target detection in the edge map, using fitting based, voting based, or arc finding based approaches etc.

For an input image, its edge map is firstly obtained by an edge detection operator. The subsequent step of neat curve segmentation requires thin edges, so a thinning algorithm is generally needed to make the edge with one pixel width. Based on our previous work (Liu and Hong, 2009), this section presents a more efficiency method of the detection of pistons, which is divided into three steps: curve segmentation and fitting, candidates finding, and target estimation. The whole detecting process is described by Figure 4.

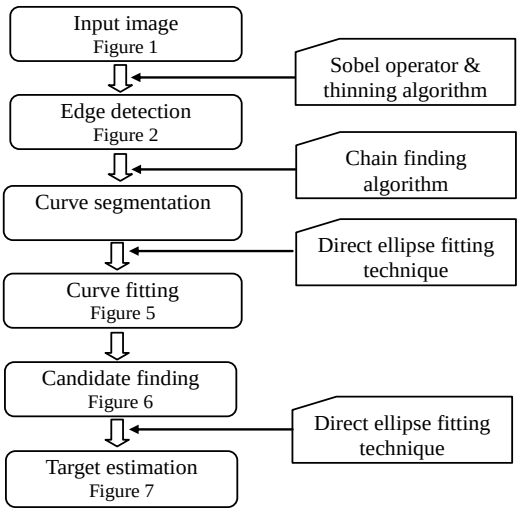


Figure 4: The flowchart of the detection algorithm

2. 1 Curve segmentation and fitting

In this step, all of the neat curve segments in the edge map are obtained, and an ellipse fitting is then performed on each of them. Finally, each neat curve is described by seven parameters with five corresponding to the ellipse and two for beginning and ending direction of the curve segment. The parameters provide a basis for the subsequent candidate finding.

(a) ***Neat curve segmentation***

Neat curve means that it does not include any branches. To find neat curves in binary edge map, we use the region growing technique (Kanatani and Ohta, 2002) that groups all of the linked pixels together as one chain. By the chain finding operator and by removing too small curves (shorter than 100 pixels here), there are totally 97 neat curves found in the edge map, as shown in Figure 5.

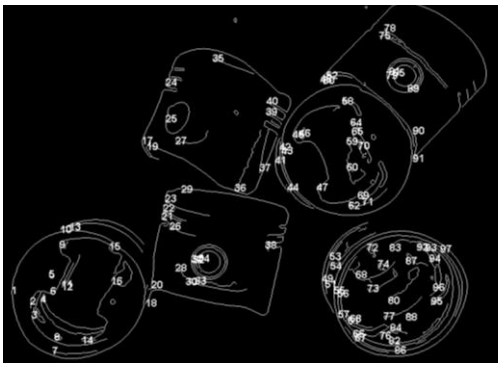


Figure 5: Totally 97 neat curves found in the edge map

(b) ***Ellipse fitting on each curve***

Once the neat curves are obtained, an ellipse fitting is performed on each curve. There are several choices for this purpose, such as the mean square error minimization or maximum likelihood based fitting methods, or even the Hough transform (HT), whose computational load is reduced greatly compared with on the original whole edge points. In literature, the direct ellipse fitting method is probably the one with the least computational complexity. As the curve is not a small elliptical segment as frequently encountered, we use the direct fitting technique (Fitzgibbon et al., 1999) to output the target ellipse.

Such obtained curve is usually named as principle curve. In order to evaluate whether two different curves belong to the same candidate or not in the subsequent step, five fitted parameters are calculated for each curve, including its orientation with respective to the ellipse center.

The orientation with respective to the ellipse center is denoted by two angles, i.e., it beginning direction θ_1 and ending direction θ_2. Based on the two angles the integrity I can be then directly calculated. The orientation and integrity are added to evaluate to what degree and in which direction the curve is an integral ellipse.

In summary, each curve is finally parameterized by a seven-dimensional vector.

$$CG = [x, y, a, b, \phi, \theta_1, \theta_2]$$

where x, y, a, b, ϕ denote the center, axes length, and orientation of the fitted ellipse, and θ_1, θ_2 denote the beginning and ending direction of the curve segment, respectively. It should be noted that one ellipse maybe denoted by two different vectors, e.g., the two vectors $CG1$ and $CG2$ with $a2=b1$, $a1=b2$, $\phi_1= \phi_2+\pi/2$ denote the same ellipse.

2.2 Candidates finding

The objective of the candidates finding is to provide a proper initialization for the final target estimation. A curve is generally regarded as an integral ellipse when it forms the main parts of the ellipse, e.g., the curves no. 1 and no. 97 shown in Figure 5. Thus, we can get five ellipses as shown in Figure 6 (as marked by blue circles).

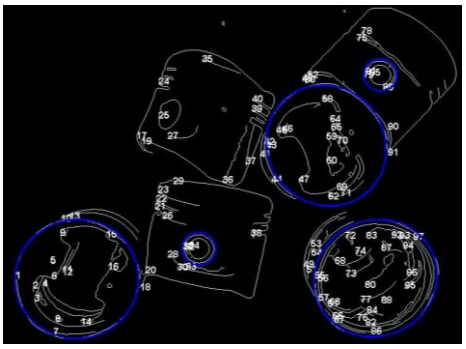


Figure 6: The fitting result of a curve segment.

After some ellipses are directly obtained from the curves, we could detect whether there contain other piston from the remaining curves. Theoretically, *CG* of two different curves that belonging to the same target satisfies that:

- Both of the first five elements are identical;
- Their orientations are complementary to form an integral ellipse.

Since the five parameters of an ellipse are not measured with the same metric, it is also hardly to judge whether two curves are close or not by a simple vector similarity measure. Therefore, below we use a candidate finding strategy that takes the five parameters of each curve into account in a comprehensive way.

Intuitively, the curves that are complementary to compose an ellipse should be grouped together as a candidate. The complementarity between curves (to compose an ellipse) is measured by several factors, including their positions, sizes, directions, that is, the curves that are complementary to form an ellipse should satisfy:

(1) They are close enough,

(2) They are with similar size, and their directions are complementary.

(a) Size

The size similarity between curve c_i and c_j is simply measured by their axes, that is, curve c_i and c_j are similar in size if the following conditions are satisfied:

$$\eta_0 < \frac{b_i}{a_j} < \eta_1, \quad \eta_0 < \frac{a_i}{b_j} < \eta_1, \text{ if } \quad \frac{\pi}{4} < |\phi_i - \phi_j| < \frac{3\pi}{4} \tag{1}$$

$$\eta_0 < \frac{a_i}{a_j} < \eta_1, \quad \eta_0 < \frac{a_i}{a_j} < \eta_1, \text{ otherwise} \tag{2}$$

where $0 < \eta_0, \eta_1 < 1$ are an empirical threshold.

(b) Direction

To measure the direction complementarity, we simplify the beginning and ending of the curve into a binary eight-dimensional vector corresponding to the eight directions, e.g., the direction of curve shown in Figure 6 is denoted by c_i =[0,0,0,1,0,0,0,0], c_j =[0,0,0,0,0,1,0,0]. Then the direction complementarity of curve c_i to c_j is defined as follows:

$$OC_{ij} = \frac{\mu_{j-i}}{\mu_{i+j}} = \frac{1}{2} = 0.5 \tag{3}$$

where μ_{j-i} denotes the number of effective directions of c_j different from c_i, and μ_{i+j} denotes the total number of effective directions of curve c_i and c_j. Thus, the direction of c_j is complementary to c_i in direction if:

$$OC_{ij} \le \beta \tag{4}$$

where β is an empirical threshold.

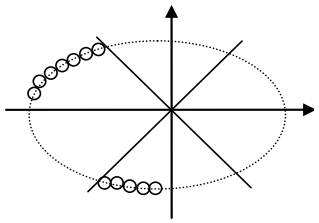


Figure 7: The fitting result of a curve segment, where the direction of c_j is complementary to c_i.

(c) Group evaluation

Once the curves with similar size and complementary directions are grouped together, we need to check the group to be a qualified candidate or not. Thus, we calculate its integrity, i.e., by the number of effective directions covered by the grouped curves. If *IG* is bigger than a predefined threshold, the group is taken as a qualified candidate:

$$IG > \gamma \tag{5}$$

where γ is an empirical threshold.

2.3 Target estimation

Once candidates are found in the previous step, we finally distinguish the pixels belonging to each target. In this step, we still use the direct fitting technique to find the best matched target of each edge pixel.

We usually regard two ellipses with similar center represent one piston. We then can remove the redundant ellipse by a neighboring field F_i around its center. The neighboring field is defined as an elliptical field with the similar center and orientation. Other ellipses whose centers within F_i are regarded as close enough to the ellipse. Thus, we can finally obtain the six ellipses that represent six pistons as shown in Figure 8.

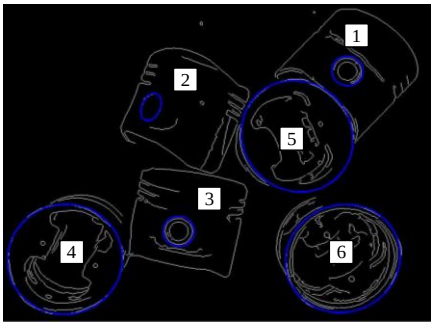


Figure 8: The finial fitting results in which the pistons no.1, no.2 and no.5 face forwards.

As shown in Figure 8, three pistons face forwards (i.e., no.1, no.2 and no.5). In order to pick the three pistons, we should firstly re-orient them to the desired pose as the piston facing upwards (i.e., no.3, no.4 and no.6). Following

section would discuss the strategy of reorient the piston to achieve the desired pose with the two-pin gripper.

3. Reorienting and grasping of target piston

Once the piston's initial pose is estimated based on the proposed algorithm, the gripper will then be guided to pick up a target piston from the conveyor belt. The key trouble in this stage is to squeeze the inner cavity of the pistons that are placed facing forwards (e.g., piston 1,2, and 3 in Figure 8). In general, we should reorient the pistons to achieve a desired pose for grasping, i.e., let the piston face upwards.

In Goldberg and his collaborator's work (Carlisle *et al.*, 1994; Rao and Goldberg, 1995), they reoriented the polygons in several steps manipulations and used the information of distance between the two parallel jaws upon closure to distinguish the orientation (the *intersection angle* θ between the edge of the polygon and the jaw of the gripper).

However, the orientation of the piston is decided by two rotational angles as described in Figure 9. Thus we need to design a sequence of pushing or grasping motions in a three-dimensional configuration space to reduce the uncertainties of the piston orientation, i.e., the angles θ_x and θ_y.

The whole grasping process is divided into two steps:

a) we design a set of reorientation actions to eliminate the uncertainties of θ_x and θ_y, that is, we push the piston to face upwards; then,

b) we pick the piston up.

3.1 Reorientation by pushing actions

We establish a *P*-coordinate frame fixed with the piston as follows. O_p is defined as the center of the end-surface of the piston. O_pX_p is parallel to axis of the piston pin hole, and O_pZ_p is along the axis of the piston. Assume that θ_x is the angle between Z_p-axis and X_s-axis, θ_y is the angle between Z_p-axis and Y_s-axis.

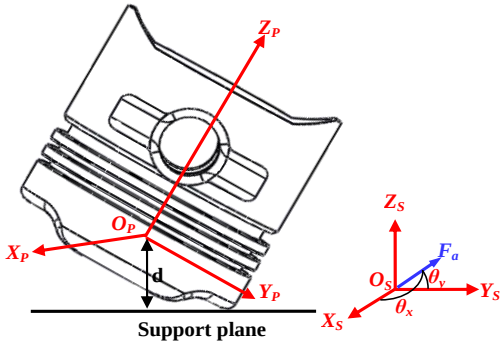


Figure 9: The *P*-coordinate frame.

Denote by $X=(\theta_x,\theta_y)$ the pose of the piston in the *P*-coordinate frame. The grasping system can be described as:

$$\frac{dX}{dt} = f\left(X, F(t), t\right) \tag{6}$$

Where $F(t)$ is the pushing force of the gripper.

We then define an energy function, E_p, of the dynamic system as:

$$E_p = F(t)\cdot d \tag{7}$$

where d is the distance of O_p to the support plane, as shown in Figure 9.

Normally, the energy function, E_p, can be regarded as a type of Lyapunov function. And, there exists a unique state X_0 of the dynamic system, which satisfies: for all X in the region $\|X- X_0\|<\varepsilon$ (ε is a positive number)

$$\begin{cases} E_p(X,t) > E_p(X_0,t), & X \neq X_0 \\ E_p(X,t) = E_p(X_0,t), & X = X_0 \\ dE_p(X,t)/dt < 0 \end{cases} \tag{8}$$

And, a special function, denoted by $g(X)$, exists in the configuration space under function E_p. We define the special function by

$$g(X) = d \tag{9}$$

where $X=(\theta_x,\theta_y)$ and $g(X)$ satisfies the following conditions:

$$\begin{cases} g(X) > g(X_0), & X \neq X_0 \\ g(X) = g(X_0), & X = X_0 \\ dg(X)/dt < 0 \end{cases} \tag{10}$$

Substituting Eq.7 and Eq.9 into Eq.8 gives

$$d\left(F(t)\cdot g(X)\right)/dt < 0 \tag{11}$$

That is,

$$dF(t)/dt\cdot g(X) + F(t)\cdot dg(X)/dt < 0 \tag{12}$$

Eq.12 is satisfied if

$$F(t) > 0 \text{ and } dF(t)/dt \le 0 \tag{13}$$

Therefore, under the pushing force F_a, the state of the system X, which satisfies $\|X- X_0\|<\varepsilon$, could converge to the stable state X_0.

The graph of Eq.8 (i.e., the function $g(\theta_x,\theta_y)=d$) is shown in Figure 10.

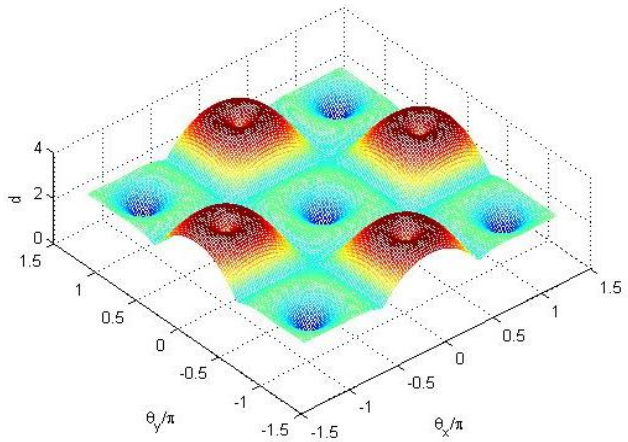


Figure 10: The 3D graph is drawn by setting h=4 and R=2 (h and R are the height and the radius of the piston) in which the piston is approximately represented by a circular cylinder

Any state of the piston in *P*-coordinate frame can be represented by a unique point in the configuration space (θ_x,θ_y,d), i.e., θ_x, θ_y and d can uniquely represent a contact state of a piston touching the support plane. And, at the bottom of the graph, we can obtain:

$$(\theta_x,\theta_y) = (0,0) \tag{14}$$

Thus, a pushing force can be generated by the gripper to enable the piston motion to follow a nominal trajectory. And, the piston would finally converge to the minimum of the graph. That is, under the pushing force, the piston would achieve the desired orientation.

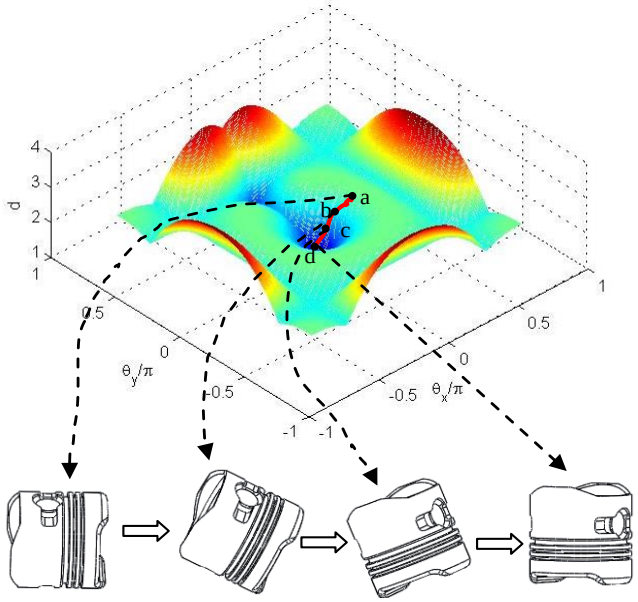


Figure 11: The correspondence between the red line in the configuration space (θ_x,θ_y,d) and the states of piston in the support plane. The minimum of the graph corresponds to the final orientation of the piston, that is, the piston is be pushed to face upwards.

Remark:

As shown in Figure 11, there exist some bow-like regions in the graph. The goal to achieve the desired pose of the piston is to move from state-*b* to state-*d* at the bottom of the bowl. Then no matter what the initial state is, with pushing force, it would eventually approach to the bottom of the bowl. In this case, the position of the piston can be treated as the state of a system, and the bowl as some constraints formed by the environment. Under the effect of gravity and pushing force, which is a state-independent input to the system, the state would finally converge to the goal region.

According to the constraints of Eq.13, an input force F_a should be designed as shown in Figure 9, where

$$F_a \cdot h > G \cdot h/4, \text{ if } \theta_x=0 \text{ or } \theta_y=0 \tag{14}$$

$$F_a \cdot h > G \cdot h/4 \cdot (\cos^2\theta_x + \cos^2\theta_y)^{1/2}, \text{ if } \theta_x \neq 0 \text{ and } \theta_y \neq 0$$

$$F_a \cdot (\cos^2\theta_x + \cos^2\theta_y)^{1/2} < F_\mu \tag{15}$$

$$F_\mu = \mu\left(-F_a \cdot (1-\cos^2\theta_x - \cos^2\theta_y)^{1/2} + G\right) \tag{16}$$

where G is the gravity force, F_μ is the frictional force and h is the height of the piston, r is the radius of the piston.

From Eq.15 and Eq.16, one can obtain that the piston would be pushed to face upwards when:

$$\frac{1}{4}G < F_a < \frac{\mu}{1+\mu}G \tag{17}$$

Obviously, F_a also satisfies the Eq.13 as:

$$F_a > 0 \quad \text{and} \quad dF_a/dt = 0$$

Figure 12 reports the definition of the distance d, and the input force F_a.

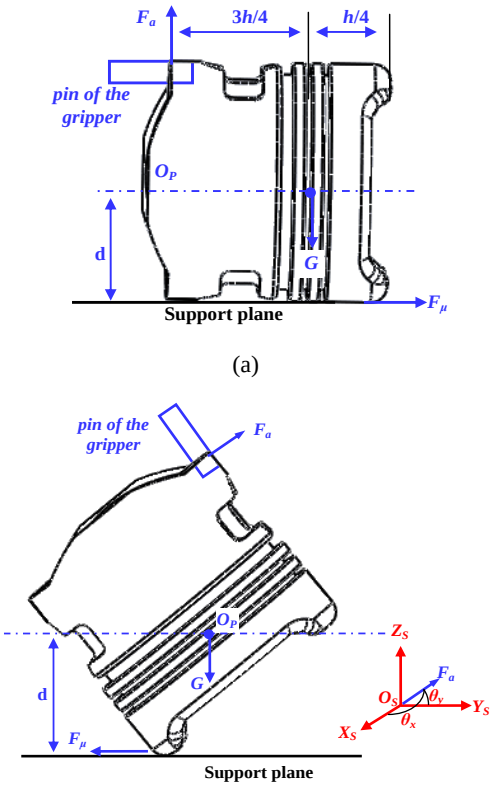


Figure 12: Design of the pushing action for the reorientation of the piston. In (a) the piston initially faces forwards, and in (b), the piston is pushed by the gripper.

3.2 Grasp by stretching actions

Once the piston has been pushed to face upwards, next is to stretch its inner cavity with the pins of the gripper. Note that we should design the pushing actions and the stretch actions for grasping of a piston facing forwards, e.g., pistons-1, 2 and 3 in Figure 8. But only the stretch action is needed for grasping of a piston facing upwards, e.g., pistons-4, 5 and 6 in Figure 8.

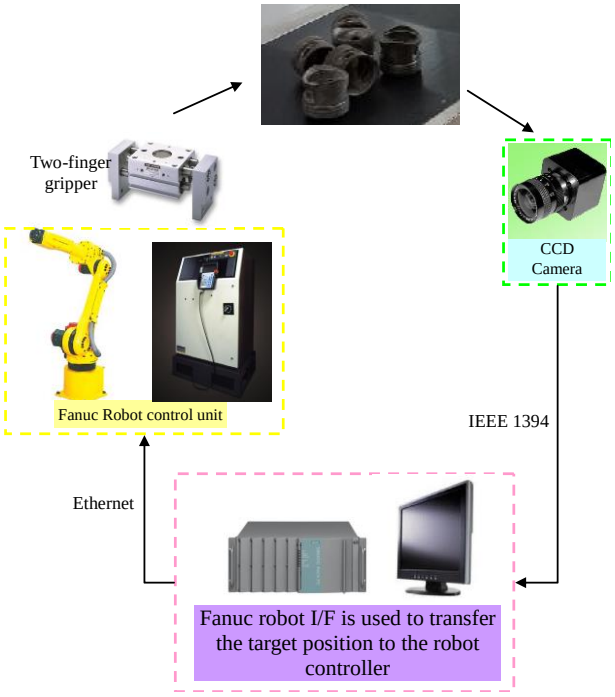
Following would conduct the experiment for picking the pistons under arbitrary poses.

4. Experiments

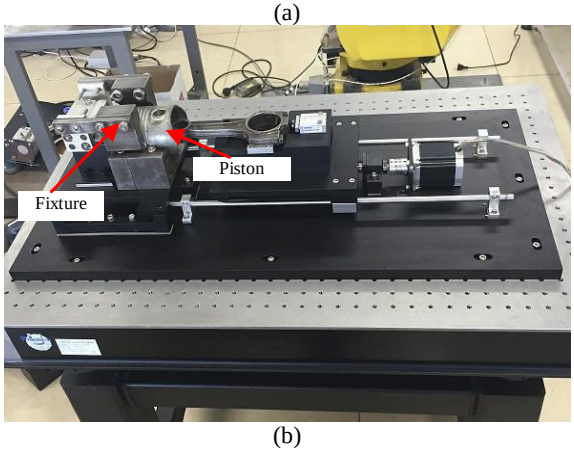
4.1 Experiment on picking up the pistons

A vision-based bin-picking system with a pneumatic two-pin gripper is developed for picking up the pistons from a conveyor belt as shown in Figure 13. The robot chosen for the picking up application is Fanuc M6i-B, which is a six-axis articulated robot for a variety of industrial applications. A pneumatic parallel-jaw gripper, SMC MHL-16D, is mounted on the robot to pick up the target piston. The vision system is composed of a computer and a color CCD camera (Allied Vision Technologies-Manta250) with 1264x1234 pixels.

The robot is controlled by a computer (with 9.0G memory and a 3.07 G Intel i7 CPU) in which the control program is written with Microsoft Visual C++. An ActiveX module, named Fanuc robot I/F, supplies a channel to send the position of the target from the computer to the robot controller.



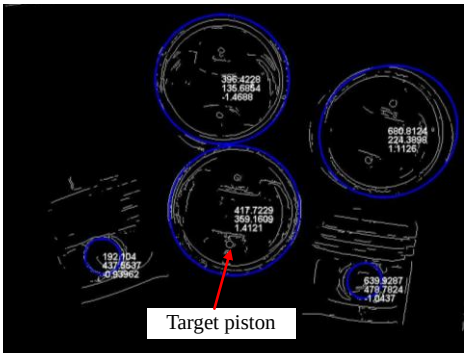
(a)



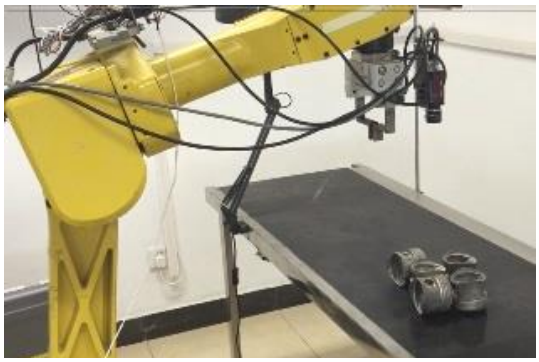
(b)

Figure 13: (a) The robotic bin-picking system. (b)The fixture holds the piston.

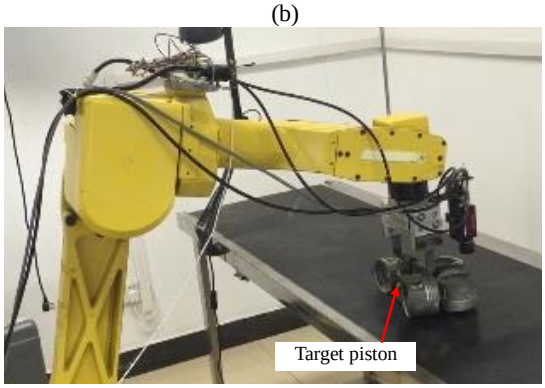
Figure 14 conducts an experiment that the gripper directly picks up a piston facing upwards. Figure 15 (a) shows the detected ellipses, in which the length of the major axis of ellipse could be used to distinguish whether the piston faces upwards or forward (large ellipses represent the pistons facing upwards and small ellipses describe the pistons facing forwards). Thus, we insert the pins of the gripper into the inner cavity of the piston facing upwards, then stretch and pick it up, as described by Figure 14(b), (c) and (d).



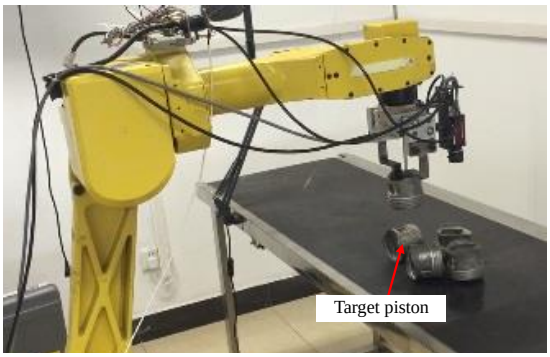
(a)



(b)



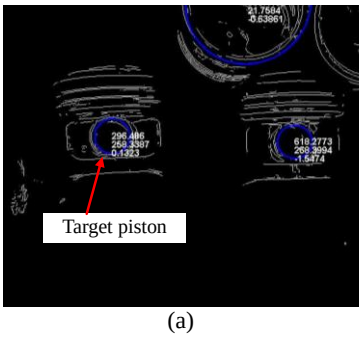
(c)



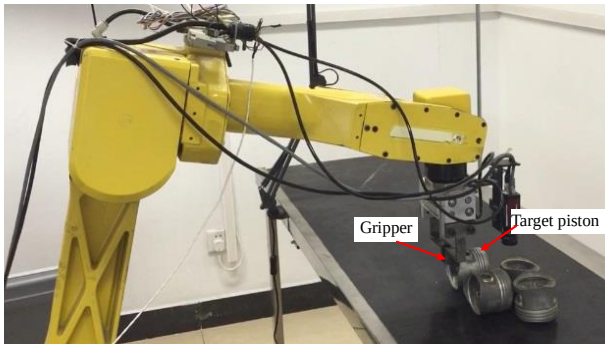
(d)

Figure 14: (a) reports the detected ellipses, in which the length of the major axis of ellipse could be used to distinguish the piston facing upwards or forwards. (b) reports that the pins of the gripper is inserted into the target piston. (c) shows that the gripper stretch the target piston and pick it up as (d).

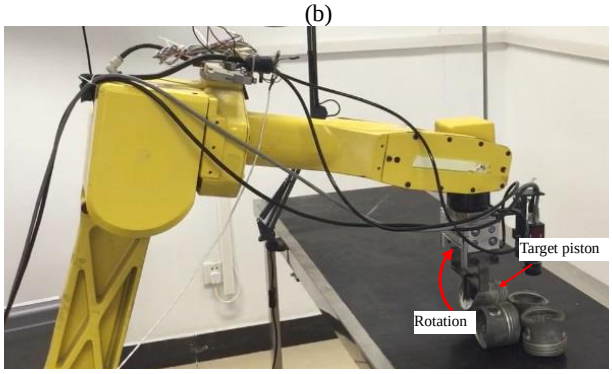
Figure 15 conduct an experiment that the gripper pushes a piston from facing forwards to upwards, and then pick it up. Figure 15 (a) shows the detected ellipses, in which the small ellipse represents the piston facing forwards. In order to ensure the robot could not interfere with the conveyor belt, we need to reorient the piston to face upward as shown in Figure 15(c-d).



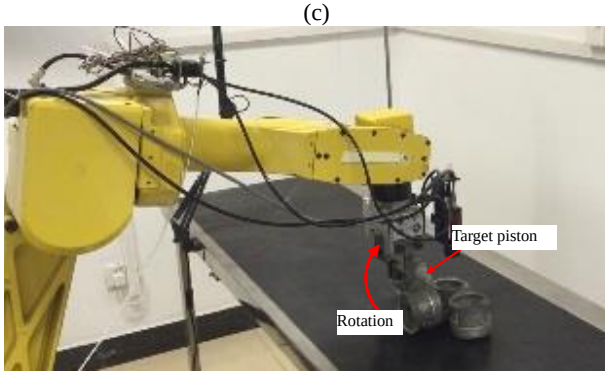
(a)



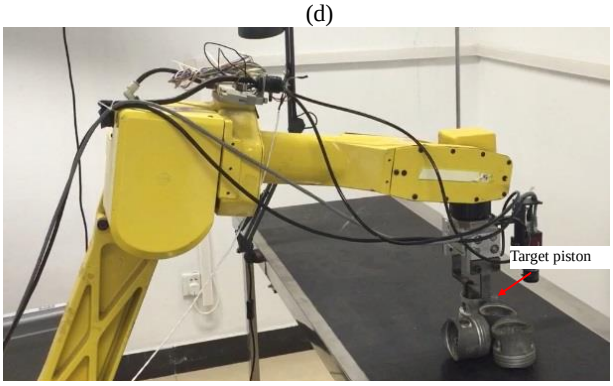
(b)



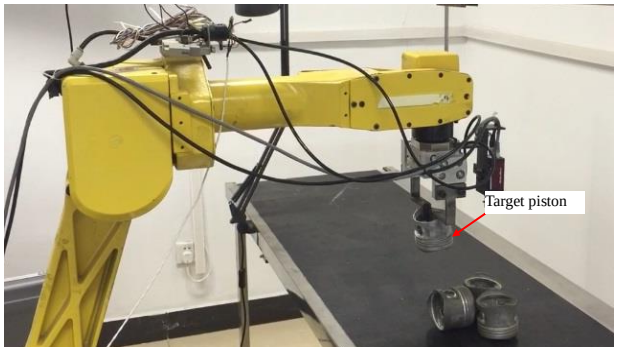
(c)



(d)



(e)



(f)

Figure 15: Picking up of a piston. (a) reports that we detect the initial pose of the pistons with the proposed ellipses detection algorithm. (b) reports that the two-pin gripper prepares to push the target piston. (c) reports that the gripper is pushing the piston. (d) shows the piston rotates. And, (e) reports the piston has been pushed facing upwards, and the gripper prepares to grasp it. (f) shows that the piston has been picked up.

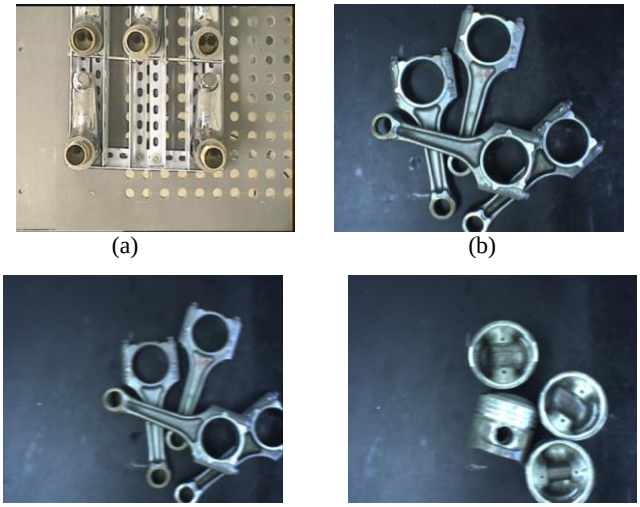
4.2 Experiments on the ellipses detection algorithm

We evaluate the proposed ellipse detection method by some other experiments on real images. Experimental simulations were done with the help of Matlab 7.0 (for our method and RHT based approach) and Visual studio 2013 (for Opencv) on a personal computer with 4.0G memory and a 2.27 G Intel i3 CPU.

We test the proposed method on some real images, i.e., water taps, rods, blurry rods and blurry pistons, as shown in the top four figures in Figure 16. The comparative results are shown in Table 1, and Figures 17, 18 and 19. For our algorithm and the other two approaches, all of the detected ellipses are shown in the image with the same rules in the final step. Thus, we can notice that our method and Opencv detected five ellipses in the water tap image, which contains five targets. On the other hand, RHT based approach detected six ellipse. Moreover, it is the slowest one among the three approaches. This is due to that, it is the voting based approaches, as the number of pixels increases, not only the computational load increases. Similar situation also happens for the remainder images. Our algorithm detected total 7 ellipses out from the rod image, 3 ellipses from the blurry rod image, and 2 ellipses from the blurry piston image, while the RHT based approach detected 3, 2 and 3, the *fitEllipse* function detected 9, 8 and 2, respectively. It is also noticed that our algorithm and Opencv are comparable on efficiency. The efficiency of our algorithm is dependent on several factors, such as the edge pixel number, number of candidates and targets, quality of the targets etc. Thus, although the number of targets in different images may differ greatly from each other, their time-consuming can be similar.

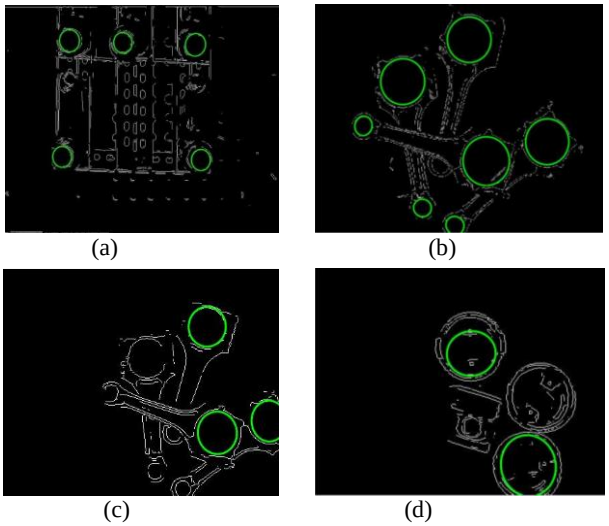
Table-1
Experimental results on four real images.

Approach	Times(s)				#Detected target			
	Water tap	Rod	Blurry rods	Blurry pistons	Water tap	Rod	Blurry rods	Blurry pistons
RHT	674.1	71.7	80.7	27.9	6	3	2	3
Opencv	0.54	0.74	0.58	0.47	5	9	8	2
Our method	4.24	4.5	0.6	1.1	5	7	3	2



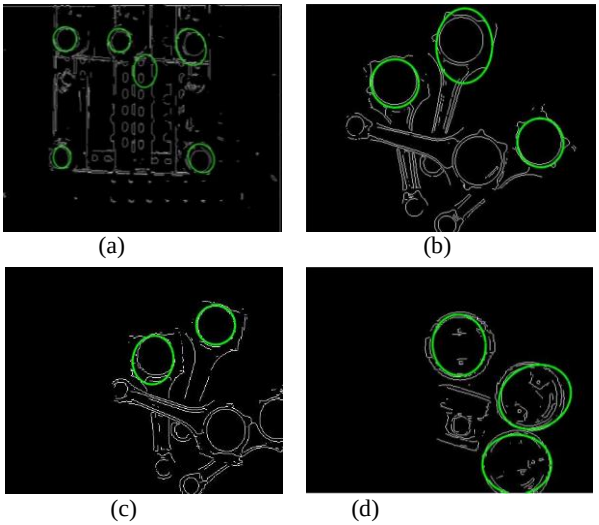
(a) (b) (c) (d)

Figure 16: The four real images: (a) five water taps, (b) four rods, (c) four blurry rods and (d) four blurry pistons.



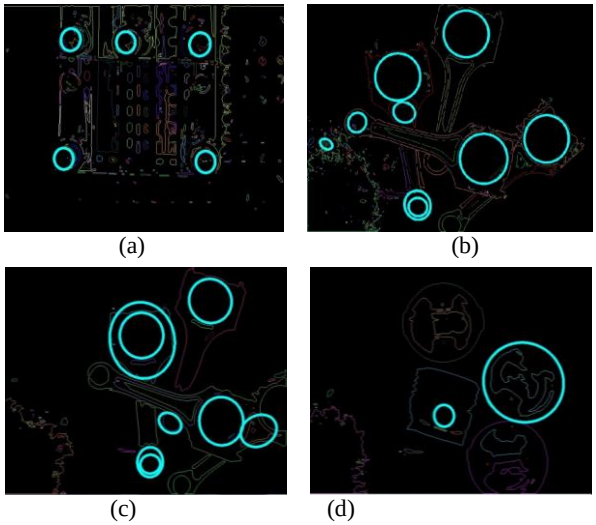
(a) (b) (c) (d)

Figure 17: The ellipse detection results by our algorithm. (a) 5 ellipses are detected; (b) 7 ellipses are detected in which 4 larger ellipses denoted the big ends while the 3 smaller ellipses denoted the small ends of the rods, and (c) detected out 3 ellipses; (d) detected out 2 ellipses.



(a) (b) (c) (d)

Figure 18: The ellipse detection results by RHT based approach. (a) 6 ellipses are detected; (b) 3 ellipses are detected from the 4 rods image, and (c) detected out 2 ellipses from 4 blurry rods image; (d) detected out 3 ellipses from 4 blurry piston image.



(a) (b) (c) (d)

Figure 19: The ellipse detection results by *fitEllipse* function of Opencv. (a) 5 ellipses are detected; (b) 9 ellipses including the 6 elliptical holes are detected from the rods image, and (c) 8 ellipses including the 3 elliptical holes are detected from the blurry rods image; (d) detected out two ellipses from four blurry piston.

5. Conclusion

Vision guidance picking of pistons from conveyor belt is a key step in the robotic piston-rod-bolt assembly system. Generally, the two stages involved are: localization, that is, how to find a target piston from a pile of parts randomly placed on the conveyor belt; and, grasping, that is, how to select the contact region that is suitable for grasping.

However, there exist some difficulties both in the localization and grasping. Firstly, the circular holes of the pistons are projected as ellipses in the image. But, it is hard to identify the circular holes due to both heavy noise and edge distortions. Secondly, the pistons possibly face to upwards or forwards on the conveyor belt; hence, it is difficult to directly grasp those pistons that face forwards.

In this paper, we addressed the issue of vision-guided robot system for picking pistons. We firstly present an ellipses detection algorithm to detect a target piston in arbitrary poses from a pile of pistons. The ellipses detection algorithm uses a hierarchical approach for detection of multiple ellipses in the environment that usually involve edge distortions. Then, we design squeezing push actions to reorient the orientation of the pistons facing forwards, i.e., let them from facing forwards to upwards. And, the stretch actions of the gripper should hold the piston with desired orientation.

Acknowledgement

This work was supported in part by National Science and Technology Major Project of China under grant number 2014ZX04013011, by Beijing Natural Science Foundation under grant number 4142056, and by National Natural Science Foundation of China under Grant 61210009.

The authors would also like to thank Yongbo Song and Ailong Yang for their help on the simulations and figures of the manuscript.

Reference

Li, Y.F., Lee, M.H. (1996), “Applying vision guidance in robotic food handling”, *IEEE Robotics & Automation Magazine*, Vol.3 No.1, pp. 4-12

Carlisle, B., Goldberg, K.Y., Rae, A.S. and Wiegley, J. (1994), “A

pivoting gripper for feeding industrial parts", *IEEE Int. Conf. on Robotics and Automation*, Vol.2, pp. 1650-1655
Sanz, P. J., Requena, A., J. Inesta, M. and Pobil, A.P. Del (2005), "Grasping the not-so-obvious: vision-based object handling for industrial applications" , *IEEE Robotics & Automation Magazine*, Vol.12 No.3, pp. 44-52,
Morales, A., Sanz, P.J., Pobil, A.P. Del, Fagg, A.H. (2006), "Vision-based three-finger grasp synthesis constrained by hand geometry", *Robotics and Autonomous Systems*, Vol.54, pp. 496-512
Taylor, M., Blake, A. and Cox, A. (1994), "Visually guided grasping in 3D", *IEEE Int. Conf. on Robotics and Automation*, pp.761-766
Macias, N. and Wen, J. (2014), "Vision guided robotic block stacking", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.779-784
Pretto, A., Tonello, S., Menegatti, E. (2013), "Flexible 3D localization of planar objects for industrial bin-picking with monocamera vision system", *IEEE Int. Conf. on Automation Science and Engineering*, pp.168-175
Holz, D., Nieuwenhuisen, M., Droeschel, D. (2014), "Active recognition and manipulation for mobile robot bin picking", *Volume 94 of the series Springer Tracts in Advanced Robotics*, pp.133-153
Liu, M.Y., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T.K., and Chellappa, R. (2012), "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *Int. J. of Robotics Research*, Vol. 31, No. 8, pp. 951-973
Dupuis, D.C., Léonard, S., Baumann, M.A., Croft, E.A., and Little, J. J. (2008), "Two-fingered grasp planning for randomized bin-picking," *Robotics: Science and Systems -Robot Manipulation: Intelligence in Human Environments*, pp.25-28,
Ogawa. T., Maeda, Y., Nakatani, S., Nagayasu, G., Shimizu, R., Ouchi, N. (2014), "Detection, localization and picking up of coil springs from a pile" , *IEEE Int. Conf. on Robotics and Automation*, pp. 3477 - 3482
Sansonia, G., Bellandia, P., Leonia, F., Docchiob, F. (2014), "Optoranger: A 3D pattern matching method for bin picking applications" , *Optics and Lasers in Engineering*, Vol. 54, pp. 222-231
Buchholz, D., Futterlieb, M., Winkelbach, S., and Wahl, F. M. (2013), "Efficient bin-picking and grasp planning based on depth data", *IEEE Int. Conf. on Robotics and Automation*, pp. 3245-3250
Domae, Y., Okuda, H., Taguchi, Y., Sumi, K., and Hirai, T. (2014), "Fast graspability evaluation on single depth maps for bin-picking with general grippers", *IEEE Int. Conf. on Robotics and Automation*, pp.1997-2004
Rao, A.S. and Goldberg, K.Y. (1995), "Manipulating Algebraic Parts in the Plane," *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 4, pp. 598-602
Liu, Z.Y. and Qiao, H. (2009), "Multiple ellipses detection in noisy environments: a hierarchical approach", *Pattern Recognition*, Vol. 42, No. 11, pp. 2421-2433
Kanatani, K. and Ohta, N. (2002), Automatic detection of circular objects by ellipse growing, in: Ninth Symposium on Sensing via Image Information, Yokohama
Fitzgibbon, A.W., Pilu, M., Fisher, R.B. (1999), "Direct least-squares fitting of ellipses", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21 No.5, pp. 476-480
Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V. (2012a), "Gradient response maps for real-time detection of textureless objects", *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Vol.4, No.5, pp.76-88
Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N. (2012b), "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes", *IEEE Int. Conf. on Computer Vision–ACCV*, pp. 548-562
Rios-Cabrera, R., Tuytelaars, T. (2013), "Discriminatively trained templates for 3D object detection: A real time scalable approach", *IEEE Int. Conf. on Computer Vision*, pp.2048-2055
Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C. (2014), "Learning 6d object pose estimation using 3d object coordinates", *13th European Conf. on Computer Vision*, Vol.8690, pp.536-551
Bonde, U., Badrinarayanan, V., Cipolla, R. (2014), "Robust Instance Recognition in Presence of Occlusion and Clutter", *13th European Conf. on Computer Vision*, Vol.8690, pp.520-535

Corresponding author

Jianhua Su can be contacted at: jianhua.su@ia.ac.cn