A Novel and Efficient Distance Detection Based on Monocular Images for Grasp and Handover

Dianwen Liu[1], Pengfei Yi[1(✉)], Dongsheng Zhou[1,2], Qiang Zhang[1,2], Xiaopeng Wei[2], Rui Liu[1], and Jing Dong[1]

[1] Dalian University, Dalian, People's Republic of China
[2] Dalian University of Technology, Dalian, People's Republic of China

Abstract. Robot grasping and human-robot handover (HRH) tasks can significantly facilitate people's production and life. In these tasks, robots need to obtain the real-time 3D position of the object, and the distance from the object to the camera plane is the critical information to get the object position. Currently, depth camera-based distance detection methods always need additional equipment, which results in more complexity and cost. In contrast, RGB camera-based methods often assume that the object's size is known or the object is at a fixed height. To make distance detection more adaptive and with low cost, a novel and efficient distance detection method based on monocular RGB images is proposed in this paper. With a simple marker, the method can estimate the object's distance in real-time from the pixel information obtained by a general, lightweight target detector. Experiments on the Baxter robot platform show the effectiveness of the proposed method, where the success rate of the grasping test reaches 87.5%, and the success rate of the HRH test goes 84.7%.

Keywords: Monocular RGB image · Distance detection · Grasping · Human-robot handover

1 Introduction

Grasping and human-robot handover (HRH) have a broad prospect of application. Grasping is one of the classic tasks in robotics, widely used in various industries of people's production and life. For example, industrial robots can accomplish the pick-and-place task, which is laborious for human laborers, and domestic robots can assist disabled or older people in their daily grasping tasks [17]. In addition, HRH [11] has become a research hotspot of cooperative robotics in recent years [2]. For example, industrial robot assistants can fetch or pass tools to human workers to increase efficiency in factories. Service robots can fetch or pass objects that older or disabled adults are needed to help them live independently.

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 642–658, 2021.
https://doi.org/10.1007/978-3-030-92635-9_37

To achieve grasping or HRH, robots must locate the 3D position of the object quickly and accurately. It's worth noting that in HRH tasks, unlike grasping tasks where the object is placed on a table of constant height, the object will be held in the human hand, and its height will change, which will make it more difficult for robots to locate the object.

The distance from the object to the camera plane, namely depth information of the object, is the key information for robots to locate the object. Currently, most robots are equipped with RGB cameras. Still, the object is difficult to be located from RGB cameras directly because 2D images obtained by RGB cameras lack depth information in space. Existing analytic-based object distance detection algorithms [3,7,8] are fast but restricted, which are difficult to adapt to complex and changeable robot applications. For example: (1) The object must be placed on a fixed horizontal plane to calculate the distance, which makes the algorithm unsuitable for HRH tasks; (2) In some algorithms, the size of the object must be known, which limits its applicability to a variety of objects. As a result, researchers need to provide additional distance sensors to robots in practical tasks. With the development of depth cameras, the current mainstream method is to use the depth information obtained by depth cameras to calculate the distance between the object and robot [14,19,20]. This approach is practical, but there are some problems: (1) Due to the high price of depth cameras, large-scale deployment in a factory environment will significantly increase the cost; (2) The hardware performance and technical level of depth cameras will seriously restrict its practical application effect; (3) When additional depth cameras are loaded, their coordinate system needs to be embedded very precisely into the robot system, which adds the complexity of device deployment.

For human beings, they can easily judge the distance of the object and complete grasp or handover tasks, which is the effect of complex human visual mechanism: (1) For objects of the same size, the size of the object perceived by human eyes is inversely proportional to the distance. (2) For multiple objects of different sizes, people can quickly judge the distance between different objects and themselves, which benefits from the rich prior knowledge acquired by birth, enabling people to know the size difference between objects in advance judge the distance. (3) Rich prior knowledge can help people quickly estimate the size of the unknown object by comparing it with the known object to judge the distance of the unknown object.

Inspired by human vision, this paper proposes a novel two-stage unknown object distance detection algorithm based on monocular RGB images. In the first stage, a known size marker is used as prior knowledge to calculate the distance of the unknown object on the desk. In the second stage, the object's distance on the desk is used to calculate the object's distance at any position. The algorithm uses a general, lightweight object detector to obtain the pixel information of the object in real-time, establishes a geometric-imaging model, and calculates the distance of the object on the desk and in the air successively. It meets precision and real-time performance; simultaneously, it is cheap and easy to be deployed.

2 Related Works

There are three ways to estimate the distance from the object to the camera plane: RGB-D-based methods, analytics-based methods, and model-based methods. Firstly, recent works of grasping and HRH tasks are introduced in Sect. 2.1, where the distance is obtained by RGB-D camera. Afterward, the traditional analytics-based distance detects methods are discussed in Sect. 2.2. Lastly, the model-based methods are introduced in Sect. 2.3, which use deep neural networks to detect the distance of the object in RGB images directly.

2.1 RGB-D-Based Methods

Robot grasping is a classic task. In recent years, most studies have completed the grasping tasks by connecting the external RGB-D camera to obtain the 6D target pose [18] or infer the 6D grasping pose of the end-effector [9,10,16]. In these works, the object distance is calculated from point cloud information by the k-means clustering algorithm. Zhang et al. [21] estimated grasping points through RGB images and realized grasping on the Baxter robot platform, however in actual grasping tasks, instead of using the RGB camera on the robot's head, they use an external depth camera.

HRH has become an important research topic. Rosenberger et al. [14] used object detection, human segmentation, hand segmentation, grasping point detection to realize the handover of a variety of objects while ensuring the safety of human partners. Yang et al. [20] divided the gestures of people grasping objects into seven categories, and each category corresponds to a grasping trajectory. In their subsequent research [19], they proposed an HRH method for transferring objects of arbitrary shape, size, and stiffness. In these implementations, objects are also located by the RGB-D camera.

2.2 Analytic-Based Methods

The analytical-based algorithms use the principle of optical imagery and geometry to directly solve the distance between the object and the camera plane from RGB images. The distance can be calculated from a simple optical imaging model when the object's size is known. For example, Pathi et al. [12] implemented a distance detection algorithm for people, in which the pixel distance from human ear to clavicle is obtained through a human pose estimation algorithm, and then using optical imaging model to calculate the distance between human and cameras. The process assumes that the actual distance from the ear to the clavicle is the same for different adults and takes it as a known size.

The challenge is to detect the distance of unknown objects using RGB images. Ricardo et al. [13] proposed a handover method based on binocular vision. This method configures two RGB cameras to obtains the distance of the object through multi-view geometry [1]. Krishnan et al. [8] proposed a complex logarithmic mapping (CLM) method, but it required objects to be placed on and

perpendicular to the optical axis, so it is not widely used. Chiang et al. [3] proposed a geometric triangulation method for measuring the object distance, in which some hard-to-measure information in the workspace needs to be taken as known quantities. Jmzad [7] implemented a soccer robot that uses a distance detection algorithm with variable pitch angles to locate the ball. Still, the algorithm also needed to measure some hard-to-measure information in advance, and the ball must be on the ground.

2.3 Model-Based Methods

With the development of deep learning, some researchers use neural networks to solve monocular distance detection problems. Zhu et al. [22] proposed a distance detection model for specific objects, which is used for vehicle distance detection in the process of autonomous driving. This method is only suitable for specific objects (size range is fixed) and unsuitable for indoor multi-object robot tasks. Haseeb et al. [6] also proposed a distance detection model for road scenes and set the average size of different categories of targets. This method requires manual measurement to mark large datasets. Griffin et al. [5] proposed a target depth estimation model, which requires the relative movement of the target and camera and maps the pixel transformation of the target to distance. Still, this method is only suitable for the circumstances of a comparatively fixed object pose when object location is unchanged. Posture changes, the pixel will also undergo significant changes, resulting in a drastic change of detection distance.

To sum up, the model-based distance detection method is more flexible. It has high accuracy but slow reasoning speed and poor generalization, making it difficult to use for complex and changeable robot tasks. Moreover, it needs to take up extra space, train the network, and download many data sets, which is inefficient. Using RGB-D cameras to obtain the object distance can meet the requirements of robot tasks in speed and accuracy. Still, the cost is high, and the RGB-d camera's hardware performance and technical level seriously restrict the practical application effect. Analytic-based methods have some advantages of high efficiency, low cost, and easy deployment. In theory, the effect of these methods is less affected by the camera hardware performance and more depends on the rationality of the algorithm. Unfortunately, existing analytics-based algorithms have significant limitations and cannot be applied to robotic tasks. Therefore, our goal is to design a novel analysis-based algorithm to detect the distance between the unknown object and the camera plane using the monocular image. It can be successfully applied to grasping or HRH tasks while retaining the advantages of traditional algorithms overcoming some limitations.

3 Method

The implementation process of our method is as follows (see Fig. 1): Firstly, the robot obtains the image of the workspace through a head-fixed camera (internal parameters are known). Then a general detector is used to obtain the 2D

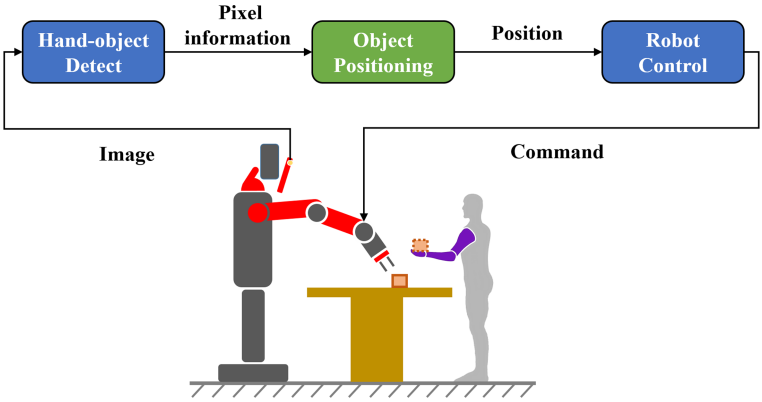


Fig. 1. General diagram of our method

pixel information of the hand and the object. Next, the 2D pixel information is transformed into the 3D position of the object through the object positioning module. Finally, the robot control module receives the 3D position of the object and sends commands to control the manipulator to move towards the target, to implement grasping or HRH.

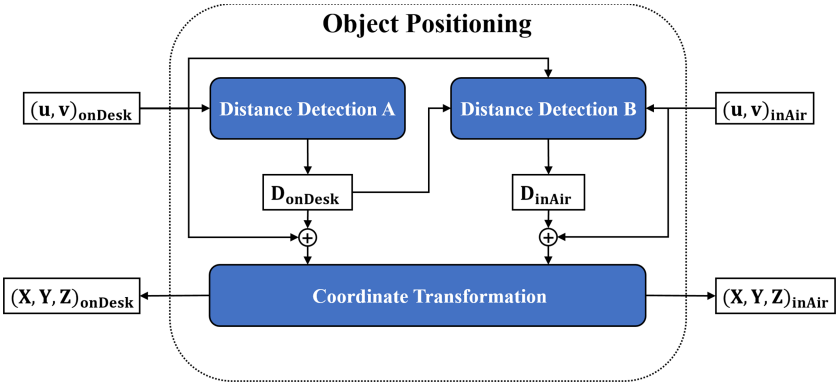


Fig. 2. Overview of object positioning module, where $(u, v)_{onDesk}$, D_{onDesk}, $(X, Y, Z)_{onDesk}$ respectively represent the pixel coordinates of the object on the desk, the distance to the camera, and world coordinates; $(u, v)_{inAir}$, D_{inAir}, $(X, Y, Z)_{inAir}$ respectively represent the pixel coordinates of the object in the air, the distance to the camera, and world coordinates.

In this section, we describe the object positioning module (see Fig. 2) in detail, which is composed of three sub-modules: (1) Distance Detection A, which is used to detect the distance of the object on the desk; (2) Distance Detection B, which is used to detect the distance of the object in the air. (3) Coordinate

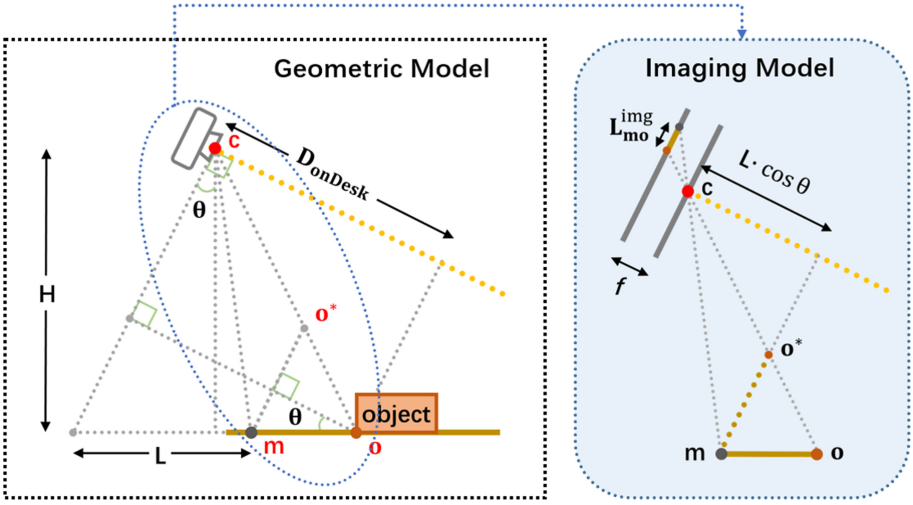


Fig. 3. Geometric-imaging model of workspace, in which the object is on the desk

conversion, which converts the object pixel coordinates to world coordinates using the acquired distance between the object and the camera plane. We focus on Distance Detection A (Sect. 3.1) and Distance Detection B (Sect. 3.2). In the grasping task, only Distance Detection A and Coordinate Transformation are needed to locate the object. In HRH, Distance Detection A is first used to get the object's distance on the desk, and then Distance Detection B and Coordinate Transformation are used to locate the object's location in the air.

3.1 Distance Detection A

This sub-module calculates the distance of the object to the camera plane when the object is on the desk (D_{onDesk}). We abstract the side view of the workspace as a geometer-imaging model (see Fig. 3): the height of camera $\boldsymbol{c}$ to the tabletop is H, the horizontal distance of the marker $\boldsymbol{m}$ to the camera plane is L, the camera inclination angle is $\boldsymbol{\theta}$, the focal length is f, and object is abstracted as a particle $\boldsymbol{o}$. Before calculation, workspace is simply marked: the pixel information of marker $Pix_{mark}(\Delta u_{mark}, v_{mark})$ is obtained and taken as a known quantity, where v_{mark} represents the v-axis pixel coordinates at the bottom of marker and Δu_{mark} represents the pixel width at the bottom of the marker. The essence of this process is, at the position where the vertical distance to the camera is H, and the horizontal distance to the camera plane is L, the pixel information projected by a line segment of length W in the camera is fixed. This pixel information can be regarded as a size prior information in the current workspace and used as a reference of the unknown object. Therefore, only one mark is required when the workbench height is fixed; it needs to be re-marked when its height is changed. The distance of the object on the desk can be expressed as:

$$D_{onDesk} = distanceA(Pix_{mark}, Pix_{object}) \tag{1}$$

where $distanceA(\cdot)$ is a distance calculation function of the object on the desk, and Pix_{object} is the pixel information of the object on the desk, which the detector can obtain. Next comes the concrete reasoning of Eq. 1. To be specific, an optical imaging model is first established according to the distance between the object $\boldsymbol{o}$ and the marker $\boldsymbol{m}$: Next comes the concrete reasoning of Eq. 1.

$$\frac{L_{mo^*}}{L \cdot \cos\theta} = \frac{L_{mo}^{img}}{f} \tag{2}$$

where $L_{mo}^{img}(cm)$ is the projection length of line segment mo in the camera, which is equivalent to the projection length of virtual image mo^* in the camera; $L_{mo^*}(cm)$ is the length of line segment mo^*. In addition, an optical imaging model is established according to the width W of the marker:

$$\frac{W}{L \cdot \cos\theta} = \frac{W^{img}}{f} \tag{3}$$

where $W(cm)$ is the actual width of the marker and $W^{img}(cm)$ is the projection width of the marker in the camera. In the geometric model, triangle similarity theorem is used to obtain:

$$\frac{L_{mo'}}{H \cdot \sec\theta} = \frac{D_{onDesk} \cdot \sec\theta - L}{D_{onDesk} \cdot \sec\theta} \tag{4}$$

From Eqs. 2, 3 and 4, it can be deduced that:

$$D_{onDesk} = \frac{L}{\sec\theta - \frac{W}{H} \cdot \frac{L_{mo}^{img}}{W^{img}}} \tag{5}$$

In addition, we can easily find that:

$$\frac{L_{mo}^{img}}{W^{img}} = \frac{\Delta v_{mo} \cdot \mathrm{d}v}{\Delta u_{mark} \cdot \mathrm{d}u} \tag{6}$$

where Δv_{mo} is the pixel length of line segment mo in image, Δu_{mark} is the pixel width of marker. $\mathrm{d}u$, $\mathrm{d}v(cm)$ is the length of each pixel in direction of u and v axis respectively. In general, $\mathrm{d}u = \mathrm{d}v(cm)$, so Eq. 5 can be changed to:

$$D_{onDesk} = \frac{L}{\sec\theta - \frac{W}{H} \cdot \frac{\Delta v_{mo}}{\Delta u_{mark}}} \tag{7}$$

wherein, θ can be set by itself or returned by robot topic, Δv_{mo} and Δu_{mark} can be directly obtained by pixel information Pix_{mark} and Pix_object. So D_{onDesk} can be solved. Equation 7 is equivalent to Eq. 1.

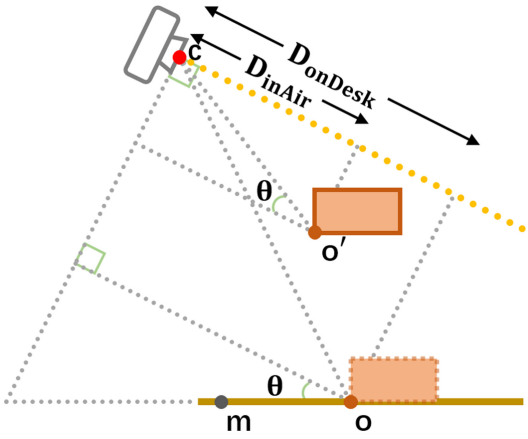


Fig. 4. Geometric-imaging model of workspace

3.2 Distance Detection B

This submodule uses the previous information to calculate the distance from the object in the air to the camera plane (D_{inAir}):

$$D_{inAir} = distanceB(Pix'_{object}, Pix_{object}, D_{onDesk}) \tag{8}$$

where $distanceB(\cdot)$ is a function of the distance of the object in the air, Pix'_{object} is the pixel information when the object is in the air. Next comes the concrete reasoning of Eq. 8. Specifically, we use the previously obtained information of the object on the desk to calculate the distance D_{inAir} (see Fig. 4). We set up optical imaging models for the two states of the object on the workbench and in the air:

$$\frac{W_{object}}{D_{onDesk}} = \frac{W^{img}_{object}}{f} \tag{9}$$

$$\frac{W_{object}}{D_{inAir}} = \frac{W^{img}_{object'}}{f} \tag{10}$$

where $W_{object}(cm)$ represents the actual width of object, $W^{img}_{object}(cm)$ is projection width in the camera of the object on the workbench, and $W^{img}_{object} = \Delta u_{object} \cdot \mathrm{d}u$, where Δu_{object} is pixel width of object on the workbench. $W^{img}_{object'}(cm)$ is the projection width in camera of the object in the air: $W^{img}_{object'} = \Delta u'_{object} \cdot \mathrm{d}u$, in which u'_{object} is the pixel width of object in the air. According to Eq. 9 and Eq. 10:

$$D_{inAir} = D_{onDesk} \cdot \frac{\Delta u'_{object}}{\Delta u_{object}} \tag{11}$$

The essence of Eq. 11 is to find a scale-invariant of an object, to connect the pixel information of an object in different states. We choose object width as the scale-invariant and use the bounding box width to approximately replace the pixel width of the object. Equation 11 is equivalent to Eq. 8.

After obtaining the distance D (D_{onDesk} or D_{inAir}), we can convert 2D pixel coordinates (u, v) to 3D world coordinates (X, Y, Z):

$$\begin{cases} (x_c, y_c, z_c)^T & = D \cdot K^{-1} \cdot (u, v, 1)^T \\ (X, Y, Z, 1)^T & = T^{-1} \cdot (x_c, y_c, z_c, 1)^T \end{cases} \tag{12}$$

where $K_{3\times3}$ is the camera's internal parameters matrix, and $T_{4\times4}$ is the camera's external parameters matrix, (x_c, y_c, z_c) is the coordinates of the object in the camera coordinate system.

4 Experiments and Results

To evaluate the effectiveness of our approach, a set of experiments are performed on a Baxter robot platform. First, introduce our experimental equipment in Sect. 4.1 and preliminary work experiments in Sect. 4.2. Afterward, the process and results of the robot grasping experiment are reported in Sect. 4.3. Then the process and results of HRH experiments are reported in Sect 4.4. After that, the time cost of our method is fully described in Sect. 4.5. Lastly, the weaknesses of the proposed distance detection algorithm are qualitatively discussed in Sect. 4.6.

4.1 Experimental Equipment

A Baxter robot is used in the experiment, with an RGB camera (1280 × 800) on its head, a laptop computer with an NVIDIA 1050 Ti graphics for image processing, and robot control. A workbench is placed within the effective working range of the manipulator in front of the robot. A worker is present in the scene. Ten objects with different materials, shapes, and sizes (see Fig. 5) are used as targets, including five large-size objects and five small-size objects to verify the applicability of our method to objects with different attributes.

4.2 Preliminary Work

For the Baxter workspace (see Fig. 6), actually, we measured the distance L' from the marker to the Y-O-Z plane of the world coordinates instead of measuring L. Because L' is easier to be measured and L can be calculated by L':

$$L = L' - (X_{camera} - H \cdot \tan\theta) \tag{13}$$

where X_{camera} is the distance from the camera to the world coordinate Y-O-Z plane.

Fig. 5. Objects used in the task, the unit of size is cm. (a) Large objects (L-objects), in order: plastic box (15.3 × 10.8 × 4.3), square carton (10.0 × 10.0 × 6.0), long carton (11.6 × 10.0 × 4.6), adhesive tape (10.4 × 5.0^2), Long tin box (15.4 × 9.4 × 3.0). (b) Small objects (S-objects), in order: rubber (3.1 × 2.1 × 1.2), small cap (0.9 × 2.3^2), small ball (3.8^3), U-disk (4.3 × 2.9 × 1.6), cosmetic box (3.5 × 3.0^2).

We get the camera's parameter matrix ($K_{3\times3}$ and $T_{4\times4}$) and correct the camera's distortion. Then the workspace is measured and marked to obtain $H = 58(cm)$, $W = 6.3(cm)$, $L' = 52(cm)$, $Pix_{mark}(\Delta u_{mark}, \Delta v_{mark}) = (52{,}790)$.

For the object detector, we chose YOLOv5 [4] to meet the real-time requirements and trained it with a 100K Frame-level dataset [15], in which only objects in contact with hands are labeled. After training, the detector takes a single RGB image as input, outputs the bounding box of the hand and objects that contact the hand. Objects that do not contact the hand cannot be detected. In addition, we filter the detection results to exclude the cases of the hand touching the tabletop, touching itself, and misdirecting the robotic arm as the hand, which can lead to task failure or cause danger (see Fig. 7).

4.3 Grasping Tests

The diagram of grasping tasks is shown in Fig. 8. Clamping jaws of different specifications (9.5 cm–13.5 cm, 2 cm–6 cm) are used to perform grasping tests on five large and five small objects (L-object and S-object) successfully. Objects are placed on the workbench at any location. The worker randomly touches the object and drives the robot to grab it; When the manipulator is moving, the worker constantly adjusts the location of other objects (only change the location, not change the object posture). 200 experiments are conducted on large objects and small objects respectively (total of 400), and each of which resulted in one of the following four conditions:

- Success: Baxter successfully grabs the object and puts it back where it was.
- Drop: Baxter grabs the object, but it drops out and is not put back in place.

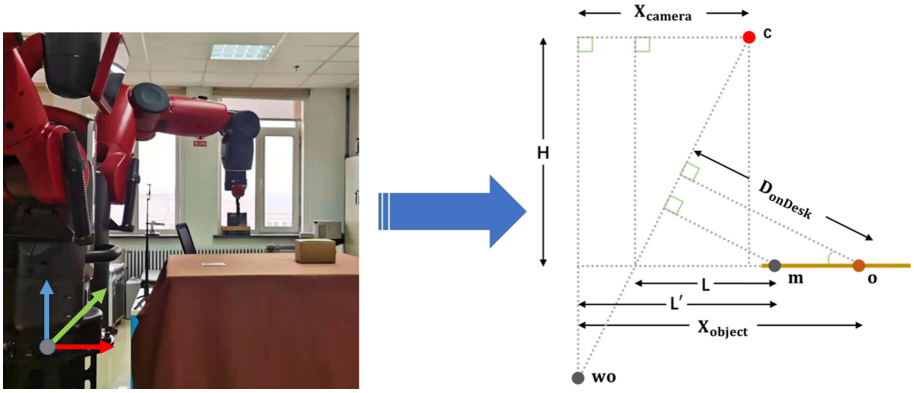


Fig. 6. Real workspace is abstracted as a geometric model, where Colored labels are world coordinates

Table 1. Success rates of the grasping tests about grasp L-objects (Large objects) and S-objects (Small objects).

	Success	Drop	Inaccuracy	Unreachable
L-objects	91.5%	0%	6.5%	2%
S-objects	83.5%	5%	9%	2.5%
Total	87.5%	2.5%	7.75%	2.25%

- Inaccuracy: Baxter cannot accurately move its arm to the object location and fails to grasp.
- Unreachable: Baxter gets the object's location but can't move the arm to it.

Table 1 shows the result of grasping: The overall grasping success rate is 87.5%, among which the grasping success rate of large objects reaches 91.5%, while that of small objects is only 83.5%, this is because small objects are much smaller in size than large objects and are difficult to grab. Large objects do not fall, while small objects fall 5% of the time, this is because when there is a slight deviation, large objects can be grabbed firmly, but small objects cannot be grabbed firmly. Failed to fetch in 7.75% of cases: (1) Most of the time is after the start moving, the change in the object location; (2) Part of the situation because of the two objects get too close, makes the detector produce error detection, which results in location errors; (3) For small objects, some deviations make the robot arm cannot reach the object grasping range, resulting in fetching failure. In 2.25% of cases, the manipulator cannot reach the target position because only the left manipulator is used, and the angle of the end-effector is fixed perpendicular to the working platform. Still, such grasping failure has nothing to do with the accuracy of our algorithm.

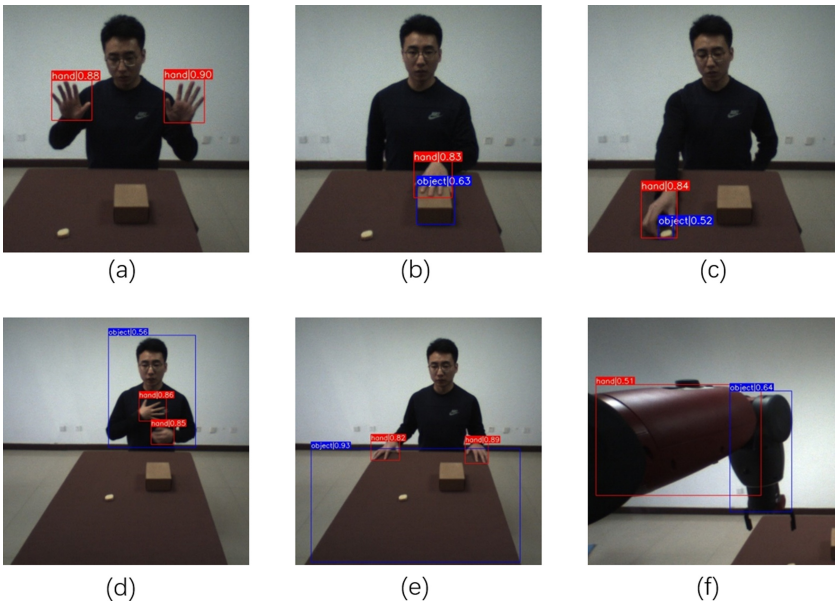


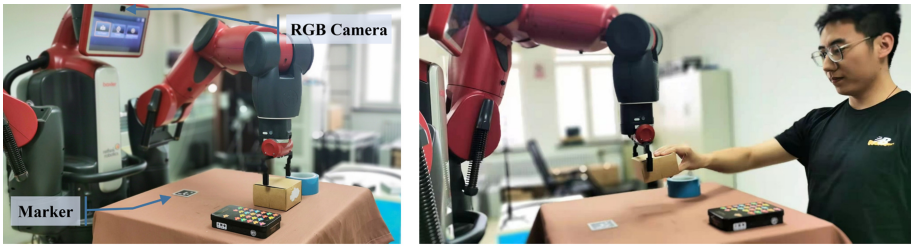
Fig. 7. Detection effect. (a) The hand does not make contact with the object, only detects the hand. (b) When hands touch objects, both hands, and objects are detected. (c) Even small objects can be detected. (d) Hands touching the body makes people the target of detection, which will lead to dangerous interactions (e) Touch the workbench, and the workbench will be detected, which will cause the task to fail. (f) The robot arms are wrongly detected as the human hand.

4.4 Human-Robot Handover Tests

The diagram of Human-robot handover tests is shown in Fig. 8. Only large objects are used (small objects cannot be used for handover). Three objects are placed on the workbench, and 10 participants in the experiment (including six males and four females). Workers randomly grab objects on the workbench for handover, take the objects to different positions in the workspace and keep the object posture relatively fixed. Each worker carried out 30 handover tests a total of 300 times, and the results of each test can be divided into the following three conditions:

- Success: Baxter successfully catches the object and returns it to the human.
- No hand: The detector cannot detect a hand, end the handover in advance.
- Inaccuracy: Mechanical cannot move to the object location accurately and get it.
- Unreachable: Baxter is unable to move its arm to the positions of the object.

HRH results are shown in Table 2: The success rate of experiments is 84.7%; 1.3% of the cases does not detect the hand, which is because the hand is blocked by the object and cannot obtain the interactive information, which led to the



(a) (b)

Fig. 8. The schematic diagram of the experiment scene. (a) robot grasping task. (b) human-robot handover task.

Table 2. Success rates of the HRH tests about grasp L-objects (Large objects).

	Success	No hand	Inaccuracy	Unreachable
L-objects	84.7%	1.3%	9.7%	4.3%

handover failure. In 9.7% of cases, the detection is not accurate, one part is because the failure of the handover due to the change of the location of the object when the manipulator moved to the object, another part is due to the inaccurate positioning, which causes the failure of the handover due to the inability of the manipulator to move to the handover accurately. In 4.3% of cases, the robotic arm cannot reach the location. Here are two possible situations: (a) The object is located correctly but beyond the effective workspace and cannot be reached; (b) The object is in the effective workspace but is wrongly positioned outside the effective workspace and cannot be reached. There is no excellent criterion to distinguish between the two conditions.

4.5 Time Cost

To prove that our method is efficient, has a low delay, and can meet the real-time requirements of the robot, we conducted an additional 40 experiments (20 grasping experiments and 20 HRH experiments). We counted the following five time indicators respectively:

- IAP-ATM: Image acquisition average time, the average time required to pull an image from a video stream and preprocess it.
- MD-ATM: Model detection average time, the average time from image input to YOLOv5 model to output detection results.
- DC-ATM: Distance calculation average time, the average time required to calculate the distance of the object and locate its position in space.
- RM-ATM: Robot movement average time, the average time it takes for the robot to move from start to finish the task.
- T-ATM: Total average time, the total average time it takes from the moment the image is pulled to the robot completes the task.

Table 3. Average time cost of IAP-ATM (image acquisition average time), MD-ATM (model detection average time), DC-ATM (distance calculation average time), RM-ATM (robot movement average time), and T-ATM (Total average time). Time measured in s.

	IAP-ATM	MD-ATM	DC-ATM	RM-ATM	T-ATM
Grasping	4.416 s	8.353×10^{-2} s	7.328×10^{-5} s	28.83 s	33.33 s
HRH	3.595 s	8.306×10^{-2} s	7.419×10^{-5} s	35.02 s	38.70 s
Total	4.006 s	8.330×10^{-2} s	7.374×10^{-5} s	31.93 s	36.02 s

Table 3 shows the time cost of our approach. We can see obviously that the main factors affecting time cost are robot moving time, the network delay to some extent, also affect the time cost. The model of reasoning and distance calculation takes very little time, hardly can be ignored, this shows that our proposed distance detection algorithm is low latency, can satisfy the real-time requirements of robot tasks. The factors that restrict the real-time completion of tasks are the robot's own moving speed and network delay. In grasping and HRH tasks: (1) The distance detection time is almost the same, which indicates that our distance detection method is very stable. (2) The difference in image pulling time is nearly 1s, caused by network fluctuation. (3) The difference in the moving time of the robots is about 7 s, which is because the trajectories and distances of the two robots are different.

4.6 Qualitative Results and Future Work

Besides the reported evaluation, additional internal tests are conducted to identify weaknesses that should be addressed in future work. During these tests, (1) Change the position of the object to traverse the entire workspace. (2) Change the posture of the object. These internal tests do not have a strict evaluation procedure but are intended to verify the adaptability of the proposed algorithm to object position and pose.

The further tests reveal some limitations of our approach. (see Fig. 9): (1) Because the height information of the object is ignored, when the object is very high, the success rate of grasping and transferring will decrease. (2) When calculating the distance of objects in the air, we choose the width at the lowest point of the object as the size-invariant, so the object must be facing the robot and keep its relative posture unchanged. When the object rotates, the accuracy of distance detection will be affected; (3) We simply use the width of the object bounding box from the detector to replace the object width approximately. When the object deviates from the central axis of the robot, there will be an error.

These limitations are essentially caused by the loss of spatial information in 2D RGB images. The object detection model can only obtain 2D position information of the object, cannot obtain the size and pose information of the object. In future work, we will obtain more abundant information of the object through a real-time monocular-based object 6D pose estimation model of the

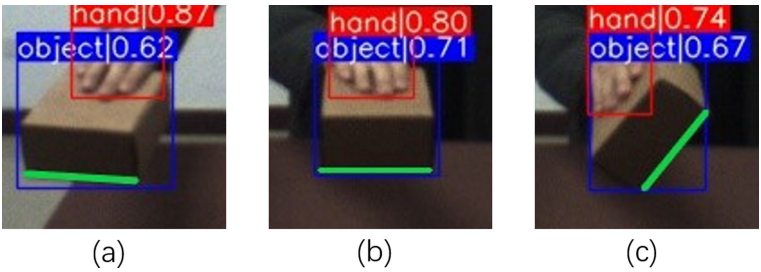


Fig. 9. The green line is the object's width, and the blue box is the bounding box. (a) When the object is facing the robot and on the central axis, the width of the bounding box is approximately equal to the object's width. (b) After changing the object's pose, the width of the bounding box is not equal to the object's width. (c) When the object is on the optical axis, the width of the bounding box is not equal to the object's width. (Color figure online)

object based on RGB image to further improve the current distance detection algorithms and the practicability of the algorithm in practical tasks.

5 Conclusion

This paper proposes a novel two-stage unknown object distance detection algorithm based on monocular RGB images, and a marker of known size is used as prior knowledge. Compared with existing RGB-D-based and analytics-based methods, the proposed algorithm has more adaptive, higher efficiency with low cost meets the requirements of real-time, accuracy, and effectiveness in robot tasks. It is successfully used for robot grasping and HRH tasks.

But the algorithm also has some limitations in actual use. For example, the object posture needs to be relatively fixed, and the algorithm is unsuitable for irregular objects. The root cause of these limitations is that the RGB camera loses the 3D information. A simple object detector can only obtain the pixel position information, which cannot determine object size and pose. There are some other restrictions: two-stage although distance detection algorithm is effective to solve the monocular RGB image the distance of the object under the air, the flexibility to drop, and the space parameter measurement method is more simple than before, but still affect the practicability of the proposed algorithm. These two points, to a certain extent, restrict the practical application of the algorithm. We will focus on real-time monocular-based object 6D pose estimation technology and grab point estimation technology. We will use them to obtain more rich information about objects from RGB images to address the limitations of the need for fixed object orientation and the inability to grasp irregular objects. We will also further improve the proposed distance detection algorithm to improve the practicability of practical tasks.

Acknowledgement. This work was supported in part by the Key Program of NSFC (Grant No. U1908214), Special Project of Central Government Guiding Local Science and Technology Development (Grant No. 2021JH6/10500140), Program for the Liaoning Distinguished Professor, Program for Innovative Research Team in University of Liaoning Province, Dalian and Dalian University, the Scientific Research fund of Liaoning Provincial Education Department (No. L2019606), Dalian University Scientific Research Platform Project (No. 202101YB03), and in part by the Science and Technology Innovation Fund of Dalian (Grant No. 2020JJ25CY001).

References

1. Andrew, A.M.: Multiple view geometry in computer vision. Kybernetes **30**(9/10), 1333–1341 (2001)
2. Bauer, A., Wollherr, D., Buss, M.: Human-robot collaboration: a survey. Int. J. Hum. Robot. **5**(1), 47–66 (2008)
3. Chiang, Y.-M., Hsu, N.-Z., Lin, K.-L.: Driver assistance system based on monocular vision. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS (LNAI), vol. 5027, pp. 1–10. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69052-8_1
4. Glenn, J., Alex, S., Jirka, B.: Ultralytics/YOLOv5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations (2021). https://doi.org/10.5281/zenodo.4679653
5. Chiang, Y.-M., Hsu, N.-Z., Lin, K.-L.: Driver assistance system based on monocular vision. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS (LNAI), vol. 5027, pp. 1–10. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69052-8_1
6. Haseeb, M., Guan, J., Ristic-Durrant, D.: Disnet: a novel method for distance estimation from monocular camera. In: 10th Planning. Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS (2018)
7. Jamzad, M., Foroughnassiraei, A., Chiniforooshan, E.: Middle sized soccer robots: Arvand. In: Robot Soccer World Cup (RoboCup) 1999. vol. 1856, pp. 61–73. Springer (1999). https://doi.org/10.1007/3-540-45327-X_4
8. Krishnan, J.V.G., Manoharan, N., Rani, B.S.: Estimation of distance to texture surface using complex log mapping. J. Comput. Appl. **3**(3), 16 (2010)
9. Mousavian, A., Eppner, C., Fox, D.: 6-DoF GraspNet: variational grasp generation for object manipulation. In: International Conference on Computer Vision (ICCV) 2019, pp. 2901–2910. IEEE/CVF (2019). https://doi.org/10.1109/ICCV.2019.00299
10. Murali, A., Mousavian, A., Eppner, C.: 6-DOF grasping for target-driven object manipulation in clutter. In: International Conference on Robotics and Automation (ICRA) 2020, pp. 6232–6238. IEEE (2020). https://doi.org/10.1109/ICRA40945.2020.9197318
11. Ortenzi, V., Cosgun, A., Pardi, T.: Object handovers: a review for robotics. IEEE Trans. Robot. **37**, 1855–1873 (2021)
12. Pathi, S.K., Kiselev, A., Kristoffersson, A.: A novel method for estimating distances from a robot to humans using egocentric RGB camera. Sensors **19**(14), 3142 (2019)
13. Ricardo, S.M., Konstantinos, C., Apostolos, M.: Benchmark for human-to-robot handovers of unseen containers with unknown filling. IEEE Robot. Autom. Lett. **5**(2), 1642–1649 (2020)

14. Rosenberger, P., Cosgun, A., Newbury, R.: Object-independent human-to-robot handovers using real time robotic vision. IEEE Robot. Autom. Lett. **6**(1), 17–23 (2021)
15. Shan, D., Geng, J., Shu, M.: Understanding human hands in contact at internet scale. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020). https://doi.org/10.1109/CVPR42600.2020.00989
16. Vohra, M., Prakash, R., Behera, L.: Real-time grasp pose estimation for novel objects in densely cluttered environment. In: International Conference on Robot and Human Interactive Communication (RO-MAN) 2019, pp. 1–6. IEEE (2019). https://doi.org/10.1109/RO-MAN46459.2019.8956438
17. Vyas, D.R., Markana, A., Padhiyar, N.: Robotic grasp synthesis using deep learning approaches: a survey. In: Sahni, M., Merigó, J.M., Jha, B.K., Verma, R. (eds.) Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy. AISC, vol. 1287, pp. 117–130. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9953-8_11
18. Wang, C., Xu, D., Zhu, Y.: Densefusion: 6d object pose estimation by iterative dense fusion. In: Conference on Computer Vision and Pattern Recognition (CVPR) 2019, pp. 3338–3347. IEEE/CVF (2019). https://doi.org/10.1109/CVPR.2019.00346
19. Yang, W., Paxton, C., Arsalan, M.: Reactive human-to-robot handovers of arbitrary objects (2020)
20. Yang, W., Paxton, C., Cakmak, M.: Human grasp classification for reactive human-to-robot handovers. In: International Conference on Intelligent Robots and Systems (IROS) 2020, pp. 11123–11130. IEEE/RSJ (2020). https://doi.org/10.1109/IROS45743.2020.9341004
21. Zhang, H., Lan, X., Bai, S.: Roi-based robotic grasp detection for object overlapping scenes. In: International Conference on Intelligent Robots and Systems (IROS) 2019. pp. 4768–4775. IEEE/RSJ (2019). https://doi.org/10.1109/IROS40897.2019.8967869
22. Zhu, J., Fang, Y.: Learning object-specific distance from a monocular image. In: International Conference on Computer Vision (ICCV) 2019, pp. 3838–3847. IEEE/CVF (2019). https://doi.org/10.1109/ICCV.2019.00394