Hindawi
Mobile Information Systems
Volume 2022, Article ID 2884611, 10 pages
https://doi.org/10.1155/2022/2884611

Research Article

Autonomous Piloting and Simulation on Underwater Manipulations Based on Vision Positioning

Junli Wang ,[1,2] Shitong Wang,[1] and Wenhao Leng[3]

[1] *School of AI & Computer Science, Jiangnan University, Wuxi 214122, China*
[2] *Wuxi Vocational Institute of Commerce, Wuxi 214153, China*
[3] *China Ship Scientific Research Center, Wuxi 214082, China*

Correspondence should be addressed to Junli Wang; wangjunli@wxic.edu.cn

Received 14 May 2022; Revised 21 August 2022; Accepted 7 October 2022; Published 1 November 2022

Academic Editor: Chen Pengyun

ROV equipped with an underwater manipulator plays a very important role in underwater investigation, construction, and some other manipulations. Moving ROV precisely and operating an underwater manipulator to grasp and move some objects are frequent operations in underwater manipulations. At the same time, they also consume a lot of the physical strength of operators, which seriously degrades the efficiency of underwater manipulations. In this paper, a scheme of grasping the rod-shaped object autonomously is proposed. In the proposed scheme, two cameras are arranged on the ROV frame to form a stereo vision system, and then, the parameters of the position in space of the rod-shaped object are calculated from the stereo images. Accordingly, the ROV is driven, and the manipulator is controlled according to these parameters such that the end effector of the manipulator can clamp the rod-shaped object exactly. As a result, the task of capturing an object is completed autonomously. In this paper, images of the scene about underwater manipulations are simulated with the marine engineering simulation software Vortex Studio, and the position parameters of the rod-shaped cable in the scene are obtained by the algorithm proposed in this paper, and the displacement to move the ROV and the joint angles to operate the manipulator are obtained consequently. Therefore, the feasibility of autonomous capture underwater object is verified.

1. Introduction

For a long time, work class ROV (remote operated vehicle) has been widely used in marine scientific research, marine resource development, marine military security, and so on. It plays an important role in seabed operation, including pipe inspection, salvage of sunken objects, mine disposal, surface cleaning, valve operating, drilling, rope cutting and geological sampling, archaeological work [1].

Traditionally, work class ROV is teleoperated by at least 2 highly skilled operators, who stay in the surface mother ship or other shore-based control room. One in charge of piloting the ROV tries to move ROV as desired and keep as stable as possible by compensating for external motion disturbances (sea current, waves, tides) and ROV motion induced by the manipulator's reaction forces/moments, and the other operator performs the actual teleoperated manipulation task [1, 2]. During this procedure, the scene of underwater manipulations is captured by the camera installed on the ROV and transmitted to the control room, and the operators observe the video, analyze the scene situation, decide the motion of the ROV or manipulator, and then issue the motion control commands. Obviously, operating work class ROV requires many skills of operators and consumes a lot of labor strength. What is more, the cost of training a qualified operator to complete underwater manipulations is very high. Therefore, if some operations of ROV or manipulator can be done autonomously, the requirements of skill and labor intensity will be reduced, and consequently, the efficiency of underwater manipulations will be greatly improved.

In recent years, Schjølberg and Utne have graded the autonomy in ROV operations [3]. They considered that current ROV operations are mainly in levels *a*, *b*, and *c*–from direct control of equipment by personnel to remote control within the visual range, and then to remote control based on the remote video. The ROV LATIS [4] is the first project to demonstrate levels *d* and *e*, which is covered by future autonomous ROV operations. In level *d*, logic-driven vehicle control is semiautonomous control where some operations are performed through automatically generated wave points. In level *e*, logic-driven with goal orientation is that the operations are performed autonomously when high-level task instructions are uploaded. Trslic et al. studied the autonomous docking of work class ROVs [5]: vision-based pose estimation techniques are used to guide ROV for autonomously docking on both static and dynamic docking stations (Tether Management System–TMS). Several lights are mounted at the back of the TMS, and the position and pose of the ROV relative to the TMS are estimated through the image of the lights in the ROV camera. Here, attention was paid to the situation that some light markers are fully covered by the TMS frame or the tether, and the pose of the ROV cannot be estimated consequently. The pose of ROV in the new time point is estimated with the previous pose and the motion information. In this way, the problem of positioning continuity has been addressed.

Peñalver et al. studied ROV autonomous manipulations on an underwater panel mockup [6], where an underwater manipulator is automatically controlled to open/close a valve and to plug/unplug a hot-stab connector by using visually guided manipulations techniques. Markers were arranged on the manipulator and the panel, respectively, and then were observed simultaneously by the same camera on the frame of the ROV. By locating the markers in the coordinate system of the camera on the ROV, the relative position between the object to operate and the end effector of the manipulator can be determined. And, therefore, the manipulator will be controlled automatically to reach the target and consequently operate it—open/close a valve or plug/unplug a hot-stab connector. In order to estimate the joint angles of a hydraulically actuated manipulator for commercial use, Sivčev et al. [1] arranged a plane marker called AprilTag [7, 8] on the manipulator and then observed it with a camera on the frame of ROV—to determine the transformation matrix between the local coordinate system of the marker and the one of the camera first, and then derived the transformation matrix between the manipulator and the base of the manipulator. Finally, the joint angles of the manipulator were estimated consequently. At the same time, the position of the object to operate was determined in a similar way. To overcome the difficulty about the large delay of the visual servo system, they proposed a control solution between open loop and complete closed loop with variable steps approaching the target.

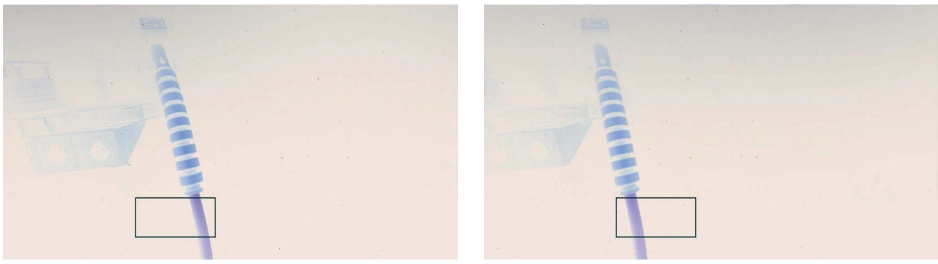
Kawamura et al. [9] proposed a control method on manipulator motion based on a calibration-free visual servo system. Some mark points were set up on the manipulator, and a stereo camera was arranged to locate the manipulator. Servo control was performed with the difference between the image of the marker point and that of the target position, and therefore, the accurate position of the target in the world coordinate system was avoided. However, a very important problem was ignored in [9], that is, how to obtain the imaging position of the target position in the camera, which will directly decide the feasibility of this method in engineering applications. In order to improve the autonomy of underwater investigation missions, García et al. [10] divided the underwater investigation task into multiple subtasks and introduced the techniques such as image segmentation and target recognition. Taking intelligent grasping as an example, the human-computer interaction and user interface were designed in detail.

With the evolution of the convolutional neural network (CNN), object detection in the underwater environment has gained a lot of attention. Naseer et al. [11, 12] employed a CNN-based detector to detect the Norway lobster Nephrops norvegicus burrows from underwater videos and proposed a detection refinement algorithm based on spatial-temporal analysis to improve the performance of generic detectors by suppressing the false positives and recovering the missed detections. The CNN-based detector is good at classifying images, but with less accuracy in positioning.

For a commercial underwater manipulator HLK-HD6W, a vision-based pose estimation algorithm was stated in [13]. Marker points were arranged on the arm link of the manipulator, and cameras were installed on the ROV body to image these markers, and consequently, the joint angles of the manipulator were estimated as the input data to automatically control manipulator. Based on this work, this study in this paper will develop a method to identify underwater rod-shaped objects and extract the position parameter of it, and then plan the motion path of underwater manipulator so as to autonomous grasp the rod-shaped object. This paper is organized as follows. In Section 2, how to identify rod-shaped object and extract its parameter from images is described in detail. In Section 3, the method about how to automatically control a 5-DOF manipulator to grasp rod-shaped object is stated. In Section 4, the identification and positioning of rod-shaped object and the motion control of the manipulator are simulated and analyzed. Finally, in Section 5, the feasibility of autonomously operating the manipulator to grasp an underwater object based on a visual image system is summarized.

2. Identification and Positioning of Rod-Shaped Objects

Images of an underwater rod-shaped object taken by a stereo camera with parallel optical axes are shown in Figure 1, respectively, from left camera and from right camera. In Figure 1, the selection boxes indicate regions of interest selected by manual intervention for processing, which will be described in detail later. In order to position the rod-shaped object in 3D space, first, the object needs to be segmented from the backgrounds in both the left and right images, and then the center line of the rod-shaped object is determined consequently; Then, the position of the center line in 3D space is calculated according to the difference of images of center lines between the left image and the right image, so as to provide input data for the automatic control of the manipulator.



(a)

(b)

Figure 1: Images of rod-shaped object in stereo camera system. (a) Image in left camera. (b) Image in right camera.

2.1. Identification of Rod-Shaped Object. In order to identify the rod-shaped object from an image, first, the edge detection method is taken to find the boundary between the rod-shaped object and the background. Then, the Hough transform is a token to detect the straight line among the boundary line. Finally, parallel line groups are expected to find from these lines, which may be the boundary lines on both sides of the rod-shaped object, and then, the center line of the rod-shaped object is estimated from the boundary line parameters.

Edge detection in digital image processing is actually a very classical problem. Many methods have been proposed until today, and they all deal with gradient calculation. These methods have been integrated into the open-source computer image library—OpenCV [14], where the Canny [15] method is particularly the commonly used one.

Further, the Hough transform algorithm [16] is often used to identify line parameters from boundary images. In the Hough transform, the line equation in the plane rectangular coordinate system is described as

$$u\cos\theta + v\sin\theta = \rho, \tag{1}$$

where u, v are the image coordinates in pixels, the origin is located in the upper left corner of the image, and vector $\overline{n}_l = (\overline{n}_x, \overline{n}_y)^T = (\cos\theta, \sin\theta)^T$ is the line's perpendicular direction. In other words, θ is the perpendicular angle, and ρ is the distance from the origin to the line in pixels.

Any point in a plane may belong to multiple lines. When the perpendicular direction is fixed, the distance from the origin to the line ρ can be calculated with (1). In line detection with the Hough transform, resolutions (region width) are set for the distance and direction parameters $(\Delta\rho, \Delta\theta)$, and then, every point in the image is visited, while the candidate line parameter is calculated for it, i.e., for every point (u_i, v_i)

$$\rho\left(\theta_j\right) = u_i\cos\theta_j + v_i\sin\theta_j, \quad \text{for } \theta_j = j\Delta\theta \in [0, \pi). \tag{2}$$

Taking $\Delta\rho$ as the distance resolution, the above formula can be rasterized.

$$\rho_j = \rho\left(\theta_j\right) - \varepsilon\Delta\rho, \quad \varepsilon \in [0, 1). \tag{3}$$

Therefore, the transform from the image pixel field to the Hough parameter field has been established. In the Hough parameter field, every point on the boundary edge votes for the potential parameter and results in a histogram graph on parameter $N(\rho_j, \theta_j)$, which stands for the number of pixels located in line with parameter (ρ_j, θ_j). Therefore, the parameter with the largest votes is the longest line. In the image library OpenCV, since line detection with Hough transform has been implemented, it is only needed to setup resolution for line parameters $(\Delta\rho, \Delta\theta)$ by using the library function, which results in parameter pair (ρ, θ) sorted in votes reversely.

The line parameters obtained directly by Hough line detection are discrete with resolutions $(\Delta\rho, \Delta\theta)$. This will bring some bad consequences: (1) the discretization of the direction of the line will bring larger error to it. (2) This error will make it possible for a line to be divided into multiple segments, which are identified as multiple parallel lines. When it is necessary to determine the line parameters more finely, this result may not meet such a requirement. However, simply refining the resolution parameters—increasing the resolution $(\Delta\rho, \Delta\theta)$—will reduce the number of votes for every grid in the parameter pair, which may reduce the effect of line detection. This is the question of Hough's transform in substance. Though there are multiscale Hough transform and progressive probabilistic Hough transform algorithm in OpenCV, the Hough transform cannot solve this question itself by tuning parameter resolutions.

Therefore, in this study, it is envisaged to collect edge points near the line originally detected by the Hough transform, and then, new line parameters are fitted from these points. This process of finding points and fitting new lines can be performed many times. On one hand, it modifies the parameter results obtained by the Hough line detection. On the other hand, it is also possible to combine the original multiple line segments into one, which becomes a longer straight-line segment. Even if the "straight line" is of a little bend, the parameter pair of ONE line will be fitted in this way.

For a candidate line parameter pair (ρ_c, θ_c), which is coarse one, edge points whose distance from the line is less than a threshold are collected. For every edge point (u, v), the distance from the line is calculated as

$$\delta = \left|u\cos\theta_c + v\sin\theta_c - \rho_c\right|. \tag{4}$$

If this distance is less than a threshold ε, the point is marked as an edge point corresponding to the candidate line, denoted as points set $(u_i, v_i), i = 1, 2, \ldots, n$. Now, it is necessary to fit the line according to the principle of minimizing the sum of square distances between the point and the line, i.e.,

$$\text{argmin: } \Psi(\rho, \theta) = \sum_{i=1}^{N} \frac{1}{2}\left(u_i \cos\theta + v_i \sin\theta - \rho\right)^2. \tag{5}$$

After differentiating the above object function $\Psi(\rho, \theta)$ with respect to its independent variables, and letting them be zeros, we have

$$\frac{\partial \Psi}{\partial \rho} = \sum_{i=1}^{N} \left(-u_i \cos\theta - v_i \sin\theta + \rho\right) = 0, \tag{6}$$

$$\frac{\partial \Psi}{\partial \theta} = \sum_{i=1}^{N} \left(u_i \cos\theta + v_i \sin\theta - \rho\right)\left(-u_i \sin\theta + v_i \cos\theta\right) = 0. \tag{7}$$

Reforming (6), we immediately have

$$\rho = \frac{1}{N}\sum_{i=1}^{N} u_i \cos\theta + \frac{1}{N}\sum_{i=1}^{N} v_i \sin\theta. \tag{8}$$

Substituting (8) into (7), we have

$$\tan 2\theta = -\frac{2\left(\sum_{i=1}^{N} u_i y_i - (1/N)\sum_{i=1}^{N} u_i \sum_{i=1}^{N} v_i\right)}{-\sum_{i=1}^{N} u_i^2 + \sum_{i=1}^{N} v_i^2 + (1/N)\left(\sum_{i=1}^{N} u_i\right)^2 - (1/N)\left(\sum_{i=1}^{N} v_i\right)^2}. \tag{9}$$

Therefore, parameters (ρ, θ) of line can be refined by (8 and 9).

After line parameters of both side boundaries of rod-shaped object, which are "parallel lines," have been obtained, the center line of rod-shaped object can be calculated by averaging the parallel lines.

In a plane, it is easy to calculate the direction vector of the line $\overline{d} = (\overline{d}_u, \overline{d}_v)^T$ from its normal vector $\overline{n}_l = (\overline{n}_u, \overline{n}_v)^T$, in the form of

$$\overline{n}_l \cdot \overline{d} = 0, \overline{d} = \left(\overline{d}_u, \overline{d}_v\right)^T = \left(\overline{n}_v, -\overline{n}_u\right)^T, \tag{10}$$

where the arbitrary direction of the line is picked, and the opposite direction is also appropriate. Therefore, the line may be expressed in the form of the parametric equation

$$p(u, v) = \left(u_A, v_A\right)^T + t\overline{d}, \tag{11}$$

where $(u_A, v_A)^T$ is the coordinate of any point on the line, and $\overline{d}$ is the direction vector of the line, $t \in R$ the independent parametric variable.

2.2. Positioning of Rod-Shaped Object. According to the method in the previous section, the position of the center lines of the rod-shaped object, respectively, in the left and right images can be obtained, which appear as straight lines. They actually are the projections of the same line from different perspectives. In order to determine the position of the rod-shaped object in space, the position parameters of the spatial line need to be recovered from the two projection lines. With the pinhole camera model, a point on the imaging plane corresponds to the ray starting from the optical center of the camera and passing through the imaging point in 3D space. And similarly, a straight line on the imaging plane corresponds to the plane determined by the camera optical center and the imaging line in 3D space.

If the virtual imaging plane of the camera is located on the plane $z = 1$ in the camera coordinate system, the projection relationship between a point $P(x, y, z)$ in 3D space of the camera coordinate system and the point $P_I(\overline{x}, \overline{y})$ in the imaging plane can be expressed as

$$\overline{x} = \frac{x}{z}, \overline{y} = \frac{y}{z}, \tag{12}$$

where x, y, z are the coordinates of point P in the camera coordinate system, and $\overline{x}, \overline{y}$ is the coordinates of the projection P_I on the imaging plane.

Given the pixel resolution of the imaging sensor, k, the coordinate of the point in the pixel coordinate system can be obtained as

$$u = k\overline{x} + u_c, v = k\overline{y} + v_c, \tag{13}$$

where u_c, v_c is the coordinate of the center of the image in the pixel coordinate system. Substituting (13) into (1), the line equation in the imaging plane may be expressed as

$$\overline{x}\cos\theta + \overline{y}\sin\theta = \frac{\left(\rho - u_c \cos\theta - v_c \sin\theta\right)}{k}. \tag{14}$$

For convenience, we introduce a new symbol $\overline{\rho}$ which is defined as

$$\overline{\rho} = \frac{\left(\rho - u_c \cos\theta - v_c \sin\theta\right)}{k}. \tag{15}$$

There is a line l in the imaging plane—the plane $z = 1$ in the camera coordinate system, which can be expressed in (1), as shown in Figure 2.

When the perpendicular direction vector $\overline{n}_l = (\overline{n}_x, \overline{n}_y)^T$ is the unit vector, i.e., $\overline{n}_x^2 + \overline{n}_y^2 = 1$, the intersection of the perpendicular and the line l, denoted as point A, can be expressed as

$$p_A = \left(\overline{\rho n}_x, \overline{\rho n}_y\right). \tag{16}$$

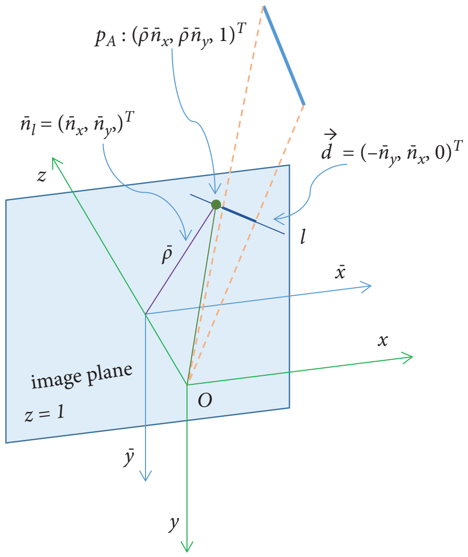

$p_A : (\bar{\rho}\bar{n}_x, \bar{\rho}\bar{n}_y, 1)^T$
$\bar{n}_l = (\bar{n}_x, \bar{n}_y,)^T$
$\vec{d} = (-\bar{n}_y, \bar{n}_x, 0)^T$
z
l
$\bar{\rho}$
$\bar{x}$
x
image plane
z = 1
O
$\bar{y}$
y

Figure 2: Image of line in imaging plane and local coordinate system of camera.

In the camera coordinate system $Oxyz$, the coordinates of the point A should be expended to 3 components, that is, $(\overline{\rho n}_x, \overline{\rho n}_y, 1)^T$. Meanwhile, the direction vector of a line, which can be derived from the direction of its perpendicular, resulting in $(-\overline{n}_y, \overline{n}_x)^T$, is located in plane $z = 1$ and expanded into the 3D vector in the camera coordinate system, resulting in $\vec{d} = (-\overline{n}_y, \overline{n}_x, 0)^T$. Line l, which is located in plane $z = 1$, and origin O of the camera coordinate system define a plane, where any line will project into the line l perspectively. Therefore, the normal vector of this plane can be derived from the vector product of the line l and the line from the origin to point A, as follows:

$$\begin{aligned} n_p &= d \times p_A \\ &= \left(-\overline{n}_y, \overline{n}_x, 1\right)^T \times \left(\overline{\rho n}_x, \overline{\rho n}_y, 0\right)^T \\ &= \left(\overline{n}_x, \overline{n}_y, -\overline{\rho}\right)^T. \end{aligned} \tag{17}$$

It should be noted that the normal vector expressed by the above formula is not a unit vector. Since the plane passes through the origin of the camera coordinate system, it can be expressed as the equation about the normal vector and the point A as follows:

$$p(x, y, z) \cdot n_p = 0 \cdot n_p, \tag{18}$$

where $p(x, y, z)$ is any point on the plane, which is defined in the camera local coordinate system. 0 is null vector in the local coordinate system, which may become nonzero vector after the above equation is transformed into a new coordinate system.

It is given that the transformation matrix and the offset vector from the left and right cameras to the global coordinate system are R_L, R_R and t_L, t_R, respectively. Then, in the left camera, the plane defined by the image line l_L and the optical center, denoted as plane p_L, can be expressed in the global coordinate system as

$$p_L(x, y, z) \cdot \left(R_L n_{p,L}\right) = t_L \cdot \left(R_L n_{p,L}\right), \tag{19}$$

where $n_{p,L}$ is the normal vector of the plane defined in the local coordinate system of the left camera. In a similar way, in the right camera, the plane defined by the image line l_R and the optical center, denoted as the plane p_L, can be expressed in the global coordinate system as

$$p_R(x, y, z) \cdot \left(R_R n_{p,R}\right) = t_R \cdot \left(R_R n_{p,R}\right). \tag{20}$$

These two planes intersect in space, as shown in Figure 3, and therefore, the intersection line l_c must be perpendicular to the normal vectors of both planes simultaneously. Subscript $\cdot_{\cdot,g}$ of the symbol in the figure stands for the vector described in the global coordinate system. Consequently, the direction vector of the intersection, d_c, can be derived by the vector product of the normal vectors of these two planes

$$d_c = \left(R_L n_{p,L}\right) \times \left(R_R n_{p,R}\right). \tag{21}$$

After obtaining the direction vector, it is necessary to determine a point on it to define the line l_c in space. Therefore, moving the point located in the optical center of the left camera—denoted by t_L, a distance λ along the perpendicular of the line l_c in the plane p_L, reaching a new point C, which is located in plane p_R. That is to say, the point C, located in both plane p_L and plane p_R, is located in the intersection line l_c.

$$\left(t_L + \lambda d_c \times \left(R_L n_{p,L}\right)\right) \cdot \left(R_R n_{p,R}\right) = t_R \cdot \left(R_R n_{p,R}\right), \tag{22}$$

where λ is an undetermined coefficient, which can be solved as

$$\lambda = \frac{t_R \cdot \left(R_R n_{p,R}\right) - t_L \cdot \left(R_R n_{p,R}\right)}{d_c \times \left(R_L n_{p,L}\right) \cdot \left(R_R n_{p,R}\right)}, \tag{23}$$

where the denominator is the scalar triple product of tree vectors

$$d_c \times \left(R_L n_{p,L}\right) \cdot \left(R_R n_{p,R}\right) = \left(R_L n_{p,L}\right) \times \left(R_R n_{p,R}\right) \cdot d_c. \tag{24}$$

It can be seen from (21) if the normal vectors of the two planes are not in parallel, d_c will not be zero and (23) is always valid; that is, the intersection of the two planes exists and keeps unique. When the center line of the observed rod-shaped object is in parallel with the line defined by the optical centers of the cameras in the stereo vision system, the two planes coincide. In this case, the normal vectors of the two planes are in parallel, and (23) is invalid. Consequently, positioning based on a stereo vision system fails.

After the coefficient λ is determined, the coordinates of the point C in the global coordinate system can be determined too.

$$p_C = t_L + \lambda d_c \times \left(R_L n_{p,L}\right), \tag{25}$$

where λ is shown in (23), and all the others are also known. Therefore, the intersection, which is the center line of the rod-shaped object, is determined in the parametric form in the global coordinate system.

$$l_c(x, y, z) = p_C + \mu d_c, \tag{26}$$

where μ is the independent parametric variable, whose value corresponds to every point on the line.

3. Scheme on Semiautonomous Manipulating and Automatic Piloting

3.1. Scheme on Semiautonomous Manipulating. Based on the theoretical method described in the previous section, ROV semiautonomous manipulating system can be designed to grasp rod-shaped objects. This study focuses on the combination of manual operation and machine intelligence, rather than the full-autonomous manipulating method only with the help of machine intelligence. Therefore, the traditional operation console and the corresponding control software are still set up to display the underwater manipulation scene observed by the camera and receive user operation commands. This system fully syncretizes human operation and machine automation so as to construct an efficient semiautonomous manipulating process about rod-shaped objects.

The whole process of clamping rod-shaped objects can be divided into the following steps:

(a) manually move the ROV to a suitable area, where the rod-shaped object can be observed completely by the stereo vision system on the ROV. During this procedure, the operation accuracy is not high, which is suited to be operated manually.

(b) accurately control the ROV to move with a small distance and operate the manipulator, such that the end effector of the manipulator clamps the rod-shaped object exactly. This procedure needs a very accurate operation, which requires operators to be skilledly trained and is generally the most time-consuming procedure.

(c) perform cutting or moving operations on the clamped rod-shaped object, including moving the ROV. Relatively, this is another operation procedure with low precision that is suited to be operated manually.

The whole process with manual and automatic operations is summarized in Figure 4.

Since it is the most time-consuming and skill-demanding procedure to accurately control the ROV and operate the manipulator to clamp the rod-shaped object, this study focuses attention on a scheme designed to automatically control during the procedure. To start the automatic control procedure on grasping the rod-shaped object, an operator only needs to select the approximate area where the rod-shaped object is located and point out the approximate spot to grasp the rod-shaped object, as shown in the following steps:

(a) manually move the ROV to a suitable area, where the rod-shaped object can be observed completely by the stereo vision system

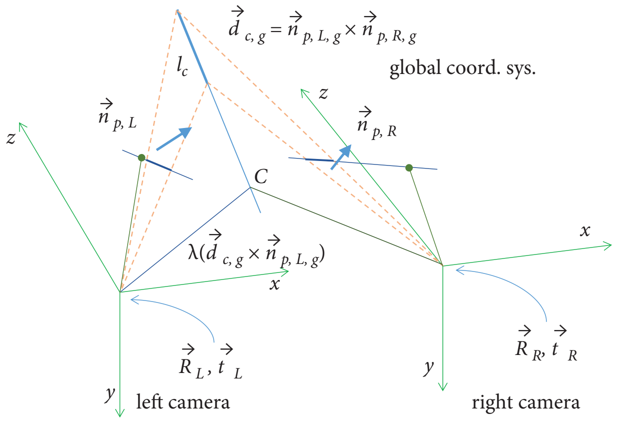


FIGURE 3: Intersection of planes corresponding to lines in images from stereo cameras.

(b) pick regions of interest for the target and grasping point on images, and then start the automatic control procedure, whose details are as follows:

  (i) pick the regions of interest for the rod-shaped object, respectively, in the images from the left and right cameras of the stereo vision system

  (ii) pick the approximate picking point on the rod-shaped object in the image from either the left or right camera

  (iii) push the "START" button on the user interface to start the automatically grasping procedure, during which a computer will process images, identify and locate the rod-shaped object, and then drive the manipulator to clamp the rod-shaped object

(c) continue the postgrasping operation, such as cutting, moving, which are operations with low accuracy, suited to be operated manually

In the above operation process, the procedure requiring the highest operation accuracy and expending the largest workload is done by computer-aided automatic control based on visual positioning. It can greatly reduce the requirements for professional skills of operators and the threshold of underwater manipulation. And consequently, it is of great value in engineering application. After the manipulator grasps the target object, the procedure of automatic operation terminates, and the underwater manipulation task turns to the manual operation mode to dispose of the target object or other intelligent workflow to finish further tasks, which is out of the scope of this study.

3.2. Automatic Piloting Method. Based on the method described in Section 2.2, the position parameters of the centerline of a rod-shaped object, which is a straight line in 3D space, can be obtained. During the automatic grasping rod-shaped object on the graphic user interface, after the operator points out the approximate area where the target object is located on and the approximate spot near the target object to clamp, the automatic control algorithm will search

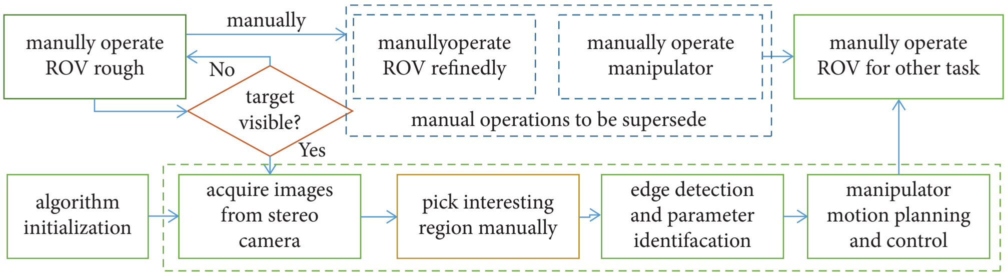

manully operate ROV rough
manually
No
target visible?
Yes
manullyoperate ROV refinedly
manually operate manipulator
manual operations to be supersede
manully operate ROV for other task
algorithm initialization
acquire images from stereo camera
pick interesting region manually
edge detection and parameter identifacation
manipulator motion planning and control

Figure 4: Some manual operations are superseded by autonomous manipulations.

for the grasping point on the rod-shaped object which is near the spot pointed out by the operator. It is assumed that the coordinate of the grasping spot M specified by the operator on the left image is $(\overline{x}_{M,L}, \overline{y}_{M,L})^T$. Then, the point on the image corresponds to a line in 3D space, which can be expressed in the left camera coordinate system as

$$p_L(x, y, z) = \xi d_M, \qquad d_M = \left(\overline{x}_{M,L}, \overline{y}_{M,L}, 1\right)^T, \tag{27}$$

where ξ is the parameter of the line equation in parametric form, and d_M is the direction vector of the line. Transforming the equation of the line from the left camera coordinate system to the global one can be done as follows:

$$p_g(x, y, z) = \xi (R_L d_M) + t_L, \qquad d_M = \left(\overline{x}_{M,L}, \overline{y}_{M,L}, 1\right)^T \tag{28}$$

This line specifies the grasping point expected by the operator, which is not accurate; that is to say, the real grasping point should be one near this line and will be the nearest one to the line.

The centerline of the rod-shaped object is described by (26), where the real grasping point, denoted as M', should be located. Furthermore, the point M' should be the nearest point to the line described by (28). And now, the line MM' is perpendicular simultaneously to the lines described by equations (26) and (28), whose direction vector is denoted as $\vec{d}_{\perp}$.

$$d_{\perp} = (R_L d_M) \times d_c. \tag{29}$$

Using the method of undetermined coefficients, the position of the grasping point can be expressed as

$$p_C + \mu d_c = \xi (R_L d_M) + t_L + \eta d_{\perp}, \tag{30}$$

where μ, ξ, η are the undetermined coefficients. It is a linear equation system with 3 unknown variables that can be solved as

$$\left[-d_c, (R_L d_M), d_{\perp}\right]\{\mu, \xi, \eta\}^T = p_C - t_L, \tag{31}$$

where the vectors $d_c, R_L d_M, d_{\perp}$ are linearly independent. The linear equation system in (31) has one unique solution, and accordingly, the three unknown variables μ, ξ, η are obtained. Only when $d_{\perp} = 0$, the lines determined by (26) and (28) are in parallel, and so there is no unique point pair with the minimum distance.

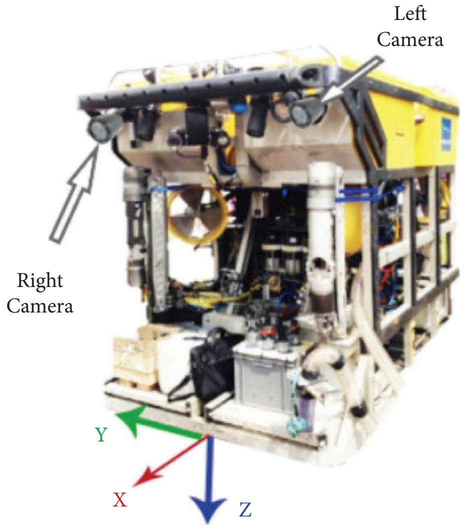


Figure 5: The positions of cameras and manipulator relative to ROV.

So far, the position of the destination at which the end effector of the underwater manipulator should reach has been obtained, which is expressed as $p_C + \mu d_c$ in the global coordinate system. To grasp the rod-shaped object, the end effector should clamp along the direction d_c at the specified position. Therefore, the end effector of the underwater manipulator is constrained in 5 degrees of freedom. As described in [13], the underwater manipulator model HLK has exactly 5 degrees of freedom in the end effector. Therefore, using this underwater manipulator, according to the destination of the end effector, it is possible to solve the angles of the joints of the underwater manipulator by inverse kinematics. Consequently, automatic control of the manipulator to grasp a rod-shaped object is achieved.

4. Simulation on Semiautonomous Manipulation

4.1. System Configuration. The tested ROV is equipped with a stereo vision system with parallel optical axes and underwater manipulator model HLK, shown in Figure 5. A satellite coordinate system is fixed on the ROV, so as to describe the configuration of the ROV, object to operation and the manipulator.

Table 1: Installation parameters of cameras and manipulator.

	Posi. x (mm)	Posi. y (mm)	Posi. z (mm)	Roll ϕ (°)	Pitch θ (°)	Yaw ψ (°)
Left camera	0	-500	-1000	90	0	90
Right camera	0	500	-1000	90	0	90
Manipulator	0	400	0	180	0	0

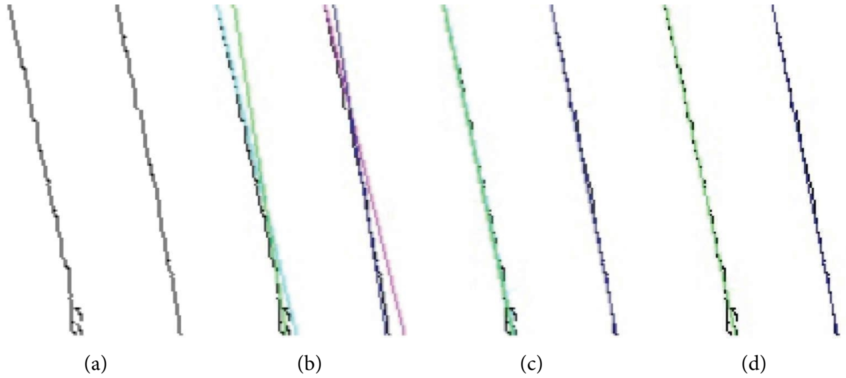


Figure 6: Process of detecting lines. (a) Edges of object. (b) Lines by hough. (c) Lines refined once. (d) Lines refined twice.

According to the general convention of camera optics, the origin of the camera local coordinate system is located at its optical center, and the z axis coincides with the optical axis and points to the object to observe. At the same time, x, y axes are parallel to the axes in the image coordinate system, respectively. In the ROV's satellite coordinate system, as shown in Figure 5, the installation position and attitude parameters of stereo vision system and manipulator are shown in Table 1.

Here, the distance of both optical axes of the stereo vision system is 1000 mm, which is located in the front and top of the ROV's frame and points ahead. The horizontal field of view of the camera is 63°, and the resolution of the image is 1920 × 1080.

4.2. Result of Simulation. While images shown in Figure 1(b) are taken as input, the image data in selected regions are converted from RGB color space to HSV color space and then are binarized by the hue component. Then, Canny edge detection is employed on the binarized image, resulting in Figure 6(a). Now, what we need to do is to find parallel line pair as the side boundary of the rod-shaped object from these points.

First, the Hough transform is employed to detect lines among the edges with the resolution parameters 4.5° in circumference and 2.5 pixels in radius. As a result, 4 straight lines have been detected, as shown in Figure 6(b). Due to the little bend in the edge of the object, two linear segments with different lengths are identified on both sides of the edges, and as a result, parameters of 4 lines are obtained. Second, the procedure to refine the line parameter described in the previous section is applied to collect the edge points adjacent to the lines with similar parameters and then fit the edge points into a line, respectively, which are shown in Figure 6(c). In this procedure, points in distance 3 pixels from the line are collected for fitting refined parameters. It can be seen that the edge lines on the same side have been identified as a coincident straight line in substance. In fact, refined again, the results will be better, as shown in Figure 6(d). During this process, the parameters of each straight line are shown in Table 2, where the parameters of each group are in the form (ρ, θ). However, for the sake of convenience, the direction of line is expressed in degrees rather than radians. Because of the errors of imaging and image processing, there is an angle about 0.72° between the "parallel lines."

Similarly, there are 3 lines detected from the image from the left camera, in which one edge is detected as two distinguished line segments with similar parameters. After refining the procedure, parameters of "more accurate" lines are obtained, as shown in Table 3.

In the process of refinement, 2 lines detected from a side edge have been combined into one new line, whose parameters are $(644\text{px}, -12.23^\circ)$. For the other side edge, it is one line for original detection, and the direction becomes more accurate after refinement.

Averaging is done for lines, which stand for the side edges of the rod-shaped object, as shown in Tables 2 and 3. Consequently, parameters for center lines of the rod-shaped object in the left and right cameras, respectively, can be obtained, as shown in Table 4.

Finally, with parameters of installation of the stereo vision system, the direction of the center line of the rod-shaped object in ROV's satellite coordinate system can be obtained as $(0.10, 0.02, 0.10)^T$, while the grasping point preferred by the operator is located at $(2.26, -0.40, -0.86)^T$ [m]. As described in [13], the underwater manipulator cannot reach the grasping point, whose maximum extension is about 1.5 m. To grasp an object, it is needed to move ROV additionally and be nearer to the

Table 2: Line parameters given by Hough line detection and refining procedure.

Line no.	Original Hough	Refined once	Refined twice
No. 1	(386.2500, −13.500)	(411.8512, −11.924)	(411.8512, −11.924)
No. 2	(438.7500, −13.500)	(476.4217, −11.198)	(476.4217, −11.198)
No. 3	(516.2500, −9.000)	(476.4217, −11.198)	(476.4217, −11.198)
No. 4	(466.2500, −9.000)	(411.9747, −11.917)	(411.8512, −11.924)

Table 3: Line parameters given by Hough line detection and refining procedure for left image.

Line no.	Original Hough	Refined once	Refined twice
No. 1	(621.2500, −13.500)	(642.1108, −12.326)	(643.9993, −12.227)
No. 2	(571.2500, −13.500)	(570.1830, −13.531)	(570.1830, −13.531)
No. 3	(706.2500, −9.000)	(651.5200, −11.831)	(643.9993, −12.227)

Table 4: Parameters of center lines in images from both cameras.

Camera no.	Parameters of line $(\rho\,[\text{pixel}], \theta)(^\circ)$
No.1: left camera	(607.0912, −12.879)
No.2: right camera	(444.1364, −11.561)

object. If ROV moves forward 1.6 m, left 0.4 m, and up 1.0 m, the ideal workspace of the underwater manipulator will cover the rod-shaped object to grasp, and the grasping point will be located at $(0.66, 0.0, 0.14)^T\,[\text{m}]$ in new situation. With inverse kinematics of the manipulator, to move the end effector of the underwater manipulator to grasp the object at the expected position, the angles of the joints of the manipulator can be solved as $(-16.2, -51.4, -78.7, 121.1, 110.0)\,[^\circ]$. Consequently, automatic controlling of the underwater manipulator to grasp rod-shaped object has been achieved.

5. Conclusion

Accurately moving the ROV and operating the underwater manipulator to grasp and place objects play an important role in underwater manipulations using a ROV equipped with underwater manipulators. When operators watch the scene of underwater manipulation on television, which is lack of 3d spatial information, it is difficult for operators to determine the relative position between the object and the end effector of the manipulator. And consequently, it is very difficult to operate the underwater manipulator to grasp the object underwater. To solve this problem, a scheme about autonomously grasping rod-shaped objects is proposed in this paper: first, a stereo vision system is arranged on the ROV frame to take a photo of the rod-shaped object to grasp. Then, the edge lines of the rod-shaped object in the images of respective cameras are detected, and the center line of the rod-shaped object is obtained. Furthermore, according to the installation parameters of the stereo vision system and the images in respective cameras, the position of the center line of the rod-shaped object in the ROV's satellite coordinate system is obtained. Finally, the joint angles of the manipulator to grasp the object are solved according to the relative position between the rod-shaped object and the ROV. When the object is out of the workspace of the underwater manipulator, it is also necessary to drive the ROV nearer first. In this way, automatic controlling the underwater manipulator to grasp rod-shaped object has been achieved.

In this paper, simulation software Vortex Studio [17], which is extensively used in marine engineering, is employed to simulate the scene of an ROV carrying an underwater manipulator to grasp cable underwater and generate the images which should be observed by the stereo vision system fixed in the front of ROV. Taking this image as the input, the relative position between the cable and the ROV is obtained successfully, and then, the ROV motion and the joint angles of the underwater manipulator to grasp the object are calculated. As a result, the feasibility of an autonomously operating underwater manipulator to grasp a rod-shaped object is validated.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) (81871373)and Research and Innovation TMDP of Wuxi Vocational Institude of Commerce (XTD202105).

References

[1] S. Sivčev, M. Rossi, J. Coleman, G. Dooly, E. Omerdić, and D. Toal, "Fully automatic visual servoing control for work-class marine intervention rovs," *Control Engineering Practice*, vol. 74, pp. 153–167, 2018.

[2] S. Sivčev, J. Coleman, E. Omerdić, G. Dooly, and D. Toal, "Underwater manipulators: a review," *Ocean Engineering*, vol. 163, pp. 431–450, 2018.

[3] I. Schjølberg and I. B. Utne, "Towards autonomy in rov operations," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 183–188, 2015.

[4] E. Omerdic, D. Toal, S. Nolan, and H. Ahmad, "Further Advances in Unmanned Marine Vehicles," *chapter ROV LATIS: Next generation smart underwater vehicle*, Inst of Engineering and Technology, London, UK, 2012.

[5] P. Trslic, M. Rossi, L. Robinson et al., “Vision based autonomous docking for work class rovs,” *Ocean Engineering*, vol. 196, Article ID 106840, 2020.

[6] A. Peñalver, J. Pérez, J. J. Fernández et al., “Autonomous intervention on an underwater panel mockup by using visually-guided manipulation techniques,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 5151–5156, 2014.

[7] E. Olson, “AprilTag: a robust and flexible visual fiducial system,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, Shanghai, China, May 2011.

[8] J. Wang and E. Olson, “AprilTag 2: efficient and robust fiducial detection,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, October 2016.

[9] A. Kawamura, M. Kubo, K. Yokoi, N. Sakagami, and S. Kawamura, “Motion control of underwater robotic arm using calibration-free visual servoing system,” in *Proceedings of the OCEANS 2015 - Genova*, vol. 1–5, Genova, Italy, May 2015.

[10] J. C. García, J. J. Fernández, P. J. Sanz, and R. Marín, “Increasing autonomy within underwater intervention scenarios: the user interface approach,” in *Proceedings of the Systems Conference*, San Diego, CA, USA, April 2010.

[11] A. Naseer, E. Nava Baro, S. D. Khan, and Y. Vila Gordillo, “Automatic detection of nephrops norvegicus burrows in underwater images using deep learning,” in *Proceedings of the 2020 Global Conference on Wireless and Optical Technologies (GCWOT)*, vol. 1–6, Malaga, Spain, October 2020.

[12] A. Naseer, E. Baro, S. D. Khan, and Y. Vila, “A novel detection refinement technique for accurate identification of nephrops norvegicus burrows in underwater imagery,” *Sensors*, vol. 22, no. 12, p. 4441, 06 2022.

[13] J. Wang, S. Wang, and W. Leng, “Vision positioning-based estimation method and its simulation studies on state of underwater manipulator,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–12, 2021.

[14] OpenCV Team, “Feature Detection,” 2021, https://docs.opencv.org/master/dd/d1a/group__imgproc__feature.html.

[15] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[16] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, London, UK, 1st edition, 2010.

[17] C. M. Labs, “Vortex Studio,” 2022, https://vortexstudio.atlassian.net/wiki/spaces/VSD21A/overview.