An approach for Semi-Autonomous Recovery of Unknown Objects in Underwater Environments

Mario Prats, José J. Fernández and Pedro J. Sanz
Computer Science and Engineering Department
University of Jaume-I, 12071 Castellón, Spain
e-mail: mprats@uji.es

Abstract—Autonomous grasping of unknown objects by a robot is a highly challenging skill that is receiving increasing attention in the last years. This problem becomes still more challenging (and less explored) in underwater environments, with highly unstructured scenarios, limited availability of sensors and, in general, adverse conditions that affect in different degree the robot perception and control systems. This paper describes an approach for semi-autonomous grasping and recovery on underwater unknown objects. A laser stripe emitter is attached to a robot forearm that performs a scan of a target of interest. This scan is captured by a camera and the partial 3D structure of the scene is recovered. A user then indicates the part where to grab the object, and the final grasp is automatically planned on that area. The methods herein presented are tested and validated in real conditions in a water tank.

I. INTRODUCTION

Exploration of the oceans and shallow waters is attracting the interest of many companies and institutions all around the world, in some cases because of the ocean valuables resources, in others because of the knowledge that it houses for scientists, and also for rescue purposes. Remote Operated Vehicles (ROV’s) are currently the most extended machines for doing these tasks. In this context, expert pilots remotely control the underwater vehicles from support vessels. However, due to the high costs and control problems involved in ROV-based missions, the trend is to advance towards more autonomous systems, i.e. Autonomous Underwater Vehicles (AUV’s), and one of the most challenging problems in the migration from operated to autonomous vehicles is to automate grasping and manipulation tasks.

Only a few projects are related with autonomous underwater manipulation. The earlier achievements date back to the 1990’s, when the UNION project [1] validated in simulation conditions coordinated control and sensing strategies for increasing the autonomy of intervention ROVs. Simultaneously, the European project AMADEUS [2] demonstrated the coordinated control of two 7 degrees of freedom (DOF) arms submerged in a water tank. The ALIVE project [3] demonstrated the capability to autonomously navigate, dock and operate on an underwater panel. It is worth mentioning the SAUVIM project [4], that recently demonstrated the autonomous recovery of seafloor objects with a 7 DOF arm. Finally, the CManipulator project [5] shows some videos on its website where an underwater manipulator autonomously grabs objects and plugs connectors.

In this paper we present our approach for grasping unknown objects underwater, either with recovery purposes, or for other applications that require a firm grasp (e.g. fixing the vehicle to an underwater structure). Grasping objects generally requires knowing at least some partial 3D structure. Different methods exist for recovering 3D information of a given scene. They can be generally classified, according to the used sensor, into sonar-based, laser rangefinders, vision-based and structured light. Sonar-based methods are the most extended approach in the underwater robotics community, because of the good properties of sound propagation in the water. However, these are more suited for long distances and not for the short range typically required for manipulation ($\approx 1m$). Laser rangefinders are rare underwater, probably because of the light absorption problem and floating particles. Stereo vision is the cheapest alternative, although not useful on turbid waters, on untextured floors or in the darkness. Structured light, however, is also a cheap alternative, can work on untextured grounds on short distances, can emit in the wavelengths that are less absorbed by the water, and offer a good accuracy even in the darkness, although they need to be combined with a camera for doing
triangulation. In this paper we combine a laser emitter with a vision system in order to recover the 3D structure of unknown objects lying on the seafloor. The reconstructed 3D point cloud is then used for planning a grasp that is executed autonomously by the robot, after a simple user indication.

Laser scanning systems have been widely used in ground applications for 3D reconstruction. For instance, [8] presented a hand-held device composed of an already calibrated laser stripe emitter and a CCD Camera. Most of the existing work is focused on how to accurately detect the laser stripe in the image [9] and on camera-laser calibration [10], [11]. In underwater environments, laser scanning has been normally used for inspection of subsea structures [12], [13], [14]. To the best of the author’s knowledge, this is the first approach where a laser scan is performed with an underwater arm and combined with a grasp planner for object recovery purposes.

A. Considered scenario and structure of the paper

The scenario illustrated in Figure 1 is considered. An underwater robotic arm (in our case the CSIP Light-weight ARM 5E [7]) is attached to an AUV. An underwater camera (in our case a Bowtech 550C-AL) is placed near the base of the arm and facing downwards, i.e. looking towards the ground. A laser stripe emitter (in our case the Tritech SeaStripe) is assembled on the forearm of the manipulator, and can emit a laser stripe visible in the camera field of view (depending on the arm joint configuration). It is assumed that the AUV has reached a pose where the target of interest is visible in the arm joint configuration. It is assumed that the AUV has laser stripe visible in the camera field of view (depending on the arm joint configuration). It is assumed that the AUV has reached a pose where the target of interest is visible in the image, and that this pose is known with respect to a global fixed frame W. The paper is then focused on the laser scan process, 3D reconstruction and grasping of the target object.

This article is organized as follows: Section II describes the algorithm used for detecting the laser stripe in the image; Section III introduces the 3D reconstruction process from the scan; User-guided grasp planning on the 3D point cloud is presented in Section IV; Finally, some results and conclusions are included in Sections V and VI.

II. LASER PEAK DETECTION

One of the first problems when dealing with 3D reconstruction based on structured light is to detect the projected light patterns in the image. In the case of a laser stripe, this process is called laser peak detection, and different approaches with variable robustness and accuracy have been presented in the literature [15].

The problem is to detect those pixels in the image that best correspond to the projected laser stripe. A state-of-the-art approach is adopted. It is composed of the following two steps:

1) The image is segmented using a color reference similar to the laser stripe color. The euclidean distance (in RGB space) between each pixel color and the reference color is computed, and only those pixels with a distance lower than a threshold are considered. Please note that underwater vehicles normally operate on dark environments that offer an advantage in this case. In fact, segmentation of a laser stripe on a dark background is easier than doing it in light conditions. The experiments presented in this paper were done under normal lighting. Results should be better in the darkness of a real ocean scenario.

2) For each row in the segmented image, the peak is computed in sub-pixel accuracy as the centroid of the segmented pixels.

Figure 2 shows a laser scan of an amphora and a bottle in a water tank, and the camera view used for the detection process. These are the two application examples that will be used throughout the paper. The recovery of an amphora and other ancient objects can be of interest for underwater archaeology. Grasping bottles may have interest for harbour cleaning or even for treasure hunting ¹.

III. 3D RECONSTRUCTION

For those pixels in the image that correspond to the laser stripe, 3D information can be recovered by triangulation. Assuming that the input image is undistorted, a pixel in the image with row and column coordinates r and c, defines a line in projective coordinates given by the column vector \( l = (l_x, l_y, 1) = \left( \frac{r - c_0}{p_x}, \frac{r - c_0}{p_y} \right) \), being \( c_0, r_0, p_x \) and \( p_y \) the camera intrinsic parameters, i.e. the principal point in pixels and the focal length to pixel size ratio. If a pixel in the image belongs in addition to the projected laser plane, the intersection of the camera ray with the laser plane gives the 3D coordinates of the point.

The line defined by the camera ray can be expressed with its parametric equation as:

\[
P = (X, Y, Z) = \lambda l
\]  \( \text{(1)} \)

¹In July 2010 two bottles of champagne (the older dating from 1841) were found in a shipwreck in Finland. They were recovered by expert divers and sold for $44000 and $35000. Underwater wine cellars are also receiving increasing attention.
Fig. 3. The reconstructed point cloud seen from the camera frame. Red color indicates more proximity to the camera. Note the little number of points reconstructed on the bottle, due to the refractive nature of the glass material. In fact, only points in the label were recovered.

If, in addition, a 3D point, \( P \), belongs to the laser plane, it holds that:

\[
(P - P_0)^T n = 0;
\]

where \( n \) is the plane normal given in camera coordinates, and \( P_0 \) is a 3D point that belongs to the plane. Merging equations 1 and 2 leads to:

\[
\lambda = \frac{P_0^T n}{1^T n}
\]

and the final 3D coordinates of the point are given by \( P = \lambda l \).

In order to compute these equations it is necessary to know the laser plane equation (given by the point \( P_0 \) and a normal \( n \)) in camera coordinates. As the laser emitter is attached to the robot arm, the camera-laser relationship is computed from the camera external calibration with respect to the arm base, and the arm kinematics. Being \( C_M^B = [C_R^B \ C_t^B] \) the homogeneous matrix representing the camera frame with respect to the arm base frame, \( B_M^F \) the relationship between the arm base frame and the forearm frame (given by the arm kinematics), and \( F_M^L \) the laser pose with relative to the forearm (given in our case by a CAD model), the laser-camera transformation is computed as \( C_M^L = C_M^B B_M^F F_M^L \).

Then, the plane normal and point can be expressed in camera coordinates as:

\[
n = C_R^L (1 \ 0 \ 0)^T \quad P_0 = C_t^L
\]

Figure 3 shows the point clouds corresponding to the amphora and the bottle scans. Missing 3D points are due either to projected shadows or a wrong segmentation of the laser stripe. Note that the scans were performed in light conditions. The amount of reconstructed points would increase in the darkness, where laser peak detection is easier.

IV. GRASP PLANNING

A supervisory grasping strategy is adopted: the approach is to let a human indicate the most suitable part for grasping, and then automatically plan a grasp on that part. In fact, in some underwater robotic applications such as archeology, the selection of the part where to grab an object is crucial in order to avoid any damages of the item. Therefore, in our approach it is a human who indicates the most suitable part for grasping, and the robot just plans a grasp around that area. The main novelty with respect to the existing approaches is that the grasp will be later performed autonomously, and not remotely controlled by the human. The user just indicates a grasp region, but the control is performed autonomously.

A. User specification & point cloud filtering

The user input is a pair of antipodal grasp points clicked on a 2D view of the object. The projection of those 2D points define two 3D lines that are used to define a 3D volume (with a fixed width).

As an example, Figure 4(top) shows a case where a grasp for the amphora is specified on the rope (see also Figure 2, right). The final point cloud (Figure 4, top right), contains a small number of points that approximately describe the volume of the object around the area indicated by the user. The bottom part of Figure 4 shows the equivalent process for the bottle.

B. Setting the grasp frame

In order to set the position and orientation of the grasp, a cartesian frame, \( G \), is defined (relative to the world frame, \( W \)) according to the following steps:
C. Inverse kinematics

After the grasp to perform has been specified, it is necessary to compute a suitable vehicle position on top of the object so that the required grasp is feasible. This can be done by computing the inverse kinematics of the whole arm-vehicle kinematic chain. The obtained vehicle pose will be sent to the vehicle navigation system and will need to be reached before starting the grasping action.

Our approach is to adopt a classical iterative inverse Jacobian method where the Jacobian is computed in a specific way in order to exploit the kinematic redundancy of the whole system. We first describe the redundancy management techniques that we consider, and then give more details on the inverse kinematics solver, followed by some examples.

1) Management of redundancy: In the context of this work, we consider a 4 DOF vehicle with an attached robotic arm. With an arm of more than two DOF, the whole kinematic system is redundant, thus allowing the robot to reach a given cartesian pose with many different joint configurations. This allows to establish preferences of some configurations over others.

In this work we adopt the Resolved Motion Rate Control (RMRC) approach for manipulator control [16], and the well-known gradient projection method (GPM) for joint redundancy management [17]. The general approach is to project a secondary task into the nullspace of the first task, as follows:

\[
\dot{q}_i = J_{Ei}^+ \dot{x}_E + (I - J_{Ei}^+ J_{Ei}) e_j
\]

where \( \dot{x}_E \) is a desired arm end-effector velocity, \( \dot{q}_i \) is the corresponding joint velocity, \( J_{Ei}^+ \) is the pseudo-inverse of the arm Jacobian at the end-effector, \( (I - J_{Ei}^+ J_{Ei}) \) is the nullspace projector, and \( e_j \) is a cost vector, which is normally computed as the gradient of a cost function \( h(q) \), i.e. \( e_j = \frac{\partial h}{\partial q} \). There are many different possibilities of cost functions: increase manipulability, minimize energy consumption, etc. [17]. In our case, we define \( e_j \) in order to accomplish the secondary task of keeping the arm posture near an equilibrium configuration that (i) minimizes unbalancing effects on the vehicle, (ii) guarantees that the end-effector is far from the workspace limits, and (iii) guarantees that the end-effector is in the field of view of the vehicle camera (situated in the bottom part of the vehicle and facing downwards, see Figure 1). In general, our cost vector adopts the following expression:

\[
e_j = \lambda_j (q_e - q)
\]

where \( q_e \) is a desired joint equilibrium configuration, \( q \) is the vector with current joint values and \( \lambda_j \) is the gain of the secondary task.

2) Grasp redundancy: In our approach, the grasping action is specified as moving a hand frame, \( H \) (attached to the gripper, and given relative to the end-effector frame, \( E \)) towards a desired relative positioning with respect to the grasp frame, expressed as \( H M_{G_e}^T \). Constrained and free degrees of freedom in this relationship can be indicated. For the constrained DOF, the hand frame must completely reach the desired relative pose with respect to the grasp frame. However, for free degrees of freedom, there is no particular relative pose used as reference. Instead, the robot controller can choose a suitable relative pose, according to different criteria such as manipulability, joint limit avoidance, etc. We refer to these directions as grasp-redundant DOF. A grasp with grasp-redundant DOF is called an under-constrained grasp.

3) Inverse kinematics of a joint and grasp-redundant vehicle-arm system: We assume that the kinematic model of the vehicle-arm system is known, and includes \( n \) degrees of freedom, the closest and the farest points of the filtered point cloud relative to the camera are found. Let \( c = (c_x, c_y, c_z) \) and \( f = (f_x, f_y, f_z) \) be these points. From them, the local object height is computed as \( h = f_z - c_z \).

Let \( e \) be the maximum amount (in meters) the hand can envelope the object. This depends on the hand geometry, and can be generally set to the maximum distance between the fingertips and the palm. In the case of the ARM 5E hand, it is set to \( e = 0.05 m \). For larger hands, more enveloping may be desired. From the object height, \( h \), and the maximum enveloping distance, \( e \), the final grasp depth is computed as the minimum of both, i.e. \( gd = \min(h, e) \).

After computing all the previous parameters, the grasp frame position is set to \( C t_G = (c_x, c_y, c_z + gd) \).

The grasp frame orientation is set so that the local Z axis corresponds to the camera Z axis, and Y axis is parallel to the vector that joins the two points indicated by the user. X axis is then set according to the right-hand rule.

Finally, the grasp frame is expressed with respect to a world fixed frame, \( W \), as \( W M_G = W M_{C_G} C M_G \), being \( W M_C \) the transformation matrix that relates the camera frame and the world frame at the moment the laser scan is performed.

Fig. 5. A simulation showing far and close views of the computed grasps on the point clouds and the inverse kinematic solutions.
freedom (DOF), four of which correspond to the underwater vehicle 3D position and yaw angle, and the rest to the arm, i.e. \( \mathbf{q} = (\mathbf{q}_v, \mathbf{q}_a) \), being \( \mathbf{q}_v = (x, y, z, \alpha) \), and \( \mathbf{q}_a = (q_1, q_2, \ldots, q_{n-4}) \). This constitutes a \( n \) DOF kinematic system for which the forward kinematic model can be computed following standard techniques, leading to \( \mathbf{x}_E = FK_{\text{va}}(\mathbf{q}) \).

The inverse kinematics can be computed following an Inverse Jacobian method [18], leading to \( \mathbf{q} = IK_{\text{va}}(\mathbf{x}_E) \). These methods are based on Newton-Rhapson numerical algorithms for solving systems of nonlinear equations. One of the main advantages is that the same RMRC controller (equation 3) that is used for real-time control can be used here for computing an inverse kinematics solution on redundant systems. The general approach is to build a virtual joint configuration \( \hat{\mathbf{q}} \) (for which an initial estimate is known) and control the corresponding virtual end-effector pose towards the desired cartesian pose, \( \mathbf{x}_E \). More concretely, the following equation is used to update the solution until convergence (i.e. \( \Delta \mathbf{x}_E \approx 0 \)):

\[
\Delta \mathbf{q} = J^*_E \Delta \hat{\mathbf{x}}_E + (I - J^*_E J_E) \mathbf{e}_j \tag{4}
\]

where \( \Delta \hat{\mathbf{x}}_E = \lambda_a (\mathbf{x}_E^* - FK_{\text{va}}(\hat{\mathbf{q}})) \). The number of redundant DOF can be further increased if grasp-redundant DOF are also considered. In the following we extend the previous expression in order to exploit under-constrained grasps.

First, as the execution of the grasp is controlled at the hand frame, \( H \), a new jacobian is computed from the end-effector jacobian, as:

\[
\mathbf{J}_H = E W_{H}^{-1} J_E \tag{5}
\]

being \( E W_{H} \) the twist transformation matrix associated to \( E M_H \). A modified jacobian is then computed for exploiting grasp-redundant DOF, as \( J^*_H = S_c^* J_H \), being \( S_c^* \) a diagonal selection matrix that selects the degrees of freedom necessary for the grasp. This removes grasp-redundant DOF from the main task. Then, equation 4 is transformed into:

\[
\Delta \mathbf{q} = J^*_H \Delta \hat{\mathbf{x}}_H + (I - J^*_H J_H) \mathbf{e}_j \tag{6}
\]

With this new expression, \( \mathbf{e}_j \) is projected not only on the joint redundant DOF, but also on the grasp redundant ones. Therefore, more DOF are available for the secondary task, allowing the robot to perform the main task, while effectively performing auxiliary motion. Secondary tasks acting on preferred configurations for the grasp-redundant DOF could also be defined by projecting them into the joint space.

4) Grasp inverse kinematics: The goal of the grasping action is to match the hand frame, \( H \), with the grasp frame, \( G \). It is therefore possible to apply the inverse kinematic model to the grasp frame origin, \( W^g \mathbf{g} \), given in an absolute frame \( W \), as \( \mathbf{q} = IK_{\text{va}}(W^g \mathbf{g}) \), leading to a suitable \( \mathbf{q}_v \) and \( \mathbf{q}_a \) for the given task.

Figure 5 shows a virtual visualization of the grasps planned using these methods.

V. RESULTS

The two application examples that have used throughout the paper have been executed with a real underwater arm in a water tank. The Light-weight ARM 5E was assembled into a fixed structure and placed underwater. The objects were manually placed on the floor in a way they were in the field of view of the camera and inside the workspace of the arm. For each object, a laser scan was performed by moving the elbow joint until the laser stripe covered all the image.

It is worth mentioning that the problem of local vehicle motion is ignored in this work, i.e. the arm base is fixed during the scan process. In practice, it is very likely that the vehicle suffers from little disturbances that affect its pose. Dealing with such base motion is outside the scope of this paper and will be addressed in future work. In fact, visual odometry [19] or local tracking techniques previously developed in our group [20] could be used for detecting such motion.

Figure 6 shows the grasping process on an amphora and a bottle with the real arm, according to the user specification, 3D
reconstruction, planning and inverse kinematics described in previous sections of this paper. Due to the reduced size of the robotic gripper, the grasp on the amphora had to be performed on a rope and succeeded. The bottle, however, slipped through the gripper before lifting it. We believe this is due to the reduced size of the gripper, its geometry that does not allow for parallel grasps, and the lack of a soft contact with high friction. Note that the grasp pose was reached successfully. In the near future there are plans for improving the gripper so that a wider range of grasps are feasible.

VI. CONCLUSION

This paper has presented a new approach for semi-autonomous recovery of unknown objects with underwater robots. First, the object 3D structure is recovered by using a laser stripe emitter combined with a camera. The laser emitter is mounted on a robot arm that is in charge of performing the scanning motion and relating the laser plane with respect to the camera pose. A 3D point cloud is generated after the scanning process, and combined with a grasp region indicated by an operator on a 2D view of the target object. This allows to reduce the number of 3D points and to delimit the grasping area. After that, a grasp planner computes a suitable grasp on the reduced point cloud, and an inverse kinematic solver computes a suitable vehicle-arm pose for reaching the object at the desired grasp pose. All the previous methods have been validated both in simulated and real conditions in a water tank. The proposed approach could be applied for recovery missions in the sea with autonomous vehicles.

ACKNOWLEDGMENT

This research was partly supported by Spanish Ministry of Research and Innovation DPI2011-27977-C03 (TRITON Project), by the European Commissions Seventh Framework Programme FP7/2007-2013 under Grant agreement 248497 (TRIDENT Project), by Foundation Caixa Castelló-Valencia ACOMP/2012/252.

REFERENCES