

To appear in the *Journal of Experimental & Theoretical Artificial Intelligence*
Vol. 00, No. 00, Month 20XX, 1–15

Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects

D. Fornas^{a*}, J. Sales^a, A. Peñalver^a, J. Pérez^a, J. J. Fernández^a, R. Marín^a and P. J. Sanz^a

^a*Department of Computer Science and Engineering, Jaume I University
Av. Vicent Sos Baynat, s/n, 12071 Castelló de la Plana, Spain*

(Received November 2014)

This article presents research on the subject of autonomous underwater robot manipulation. Ongoing research in underwater robotics intends to increase the autonomy of intervention operations that require physical interaction in order to achieve social benefits in fields such as archaeology or biology that can not afford the expenses of costly underwater operations using Remote Operated Vehicles (ROV). Autonomous grasping is still a very challenging skill, especially in underwater environments, with highly unstructured scenarios, limited availability of sensors and adverse conditions that affect the robot perception and control systems. To tackle these issues, we propose the use of vision and segmentation techniques that aim to improve the specification of grasping operations on underwater primitive shaped objects. Several sources of stereo information are used to gather 3D information in order to obtain a model of the object. Using a RANSAC segmentation algorithm, the model parameters are estimated and a set of feasible grasps are computed. This approach is validated in both simulated and real underwater scenarios.

Keywords: Underwater autonomous manipulation; grasp specification; point cloud; 3D reconstruction; shape fitting; UWSim underwater simulator

1. Introduction

Research on autonomous robotic intervention on land has recently made some valuable advances. In contrast, the current state-of-the-art in underwater intervention is at a very primitive stage where most systems are tele-operated by an expert user with complex interfaces. These operations are very expensive and only the oil industry can really afford the use of Remote Operated Vehicles (ROV). It is desirable to increase the autonomy levels for Underwater Intervention Missions, where the level of human interaction may vary from minimal participation to total domination with a control loop (Hexmoor, McLaughlan, & Tuli, 2009). In this regard, development of cheaper Autonomous Underwater Vehicles (AUV) may provide answers to important challenges in fields such as archaeology, biology or even other emerging applications like fish farms maintenance. Nowadays, the only available platforms in these fields are tele-operated or vision only autonomous systems without interaction capabilities. These restrictions make very difficult for this fields to carry out even the more simplistic operations. For this reason, development of new technologies and methodologies will allow, for example, archeologists to increase and maintain underwater historical, artistic and cultural heritage. Biology is

*Corresponding author. Email: dfornas@uji.es

another science that will be benefited as this will enable the study of new species or the taking of natural samples at higher depths.

Autonomous underwater interventions present lots of challenges, the autonomous manipulation tasks being one of the biggest. In this context there exist only a few research projects in autonomous manipulation. In the field of underwater intervention it is worth mentioning previous projects like AMADEUS (Marani, Angeletti, Cannata, & Casalino, 2000), in which two underwater robot arms were mounted on a cell to demonstrate bi-manual skills. Nevertheless, this functionality demonstrated on a fixed frame, was never demonstrated on board an AUV, and hence with a mobile base. TRIDENT (Sanz, Ridaó, et al., 2013), demonstrated the capability to autonomously survey an area and recover an object from the seafloor, but still with some interaction with a human operator and its operations were restricted to shallow waters. In the context of this project, a framework to grasp objects with the user interaction just for the grasp planning phase was presented in Prats, Fernández, and Sanz (2012). This approach, focused on increasing autonomy using 3D data, is further developed with this framework. At the moment, only the ongoing PANDORA (FP7-PANDORA, 2012) project is running in the underwater intervention context funded by European Commission. This project proposes a learning by demonstration approach where a human operator teaches the system how to turn a valve through a set of trials. After the learning stage, the system generalizes this information to turn a valve autonomously in similar conditions.

The present research has been conducted within the Spanish Project TRITON (Palomeras et al., 2014), which is in its last year. The two scenarios considered in TRITON, which will be carried out by an autonomous underwater vehicle with a robotics arm, are the following:

- The vehicle docks to an underwater permanent observatory and it performs operations on an intervention panel, turning a valve and plugging/unplugging a connector. The structure is considered as a well known object, thus limiting the amount of information needed from the operator and increasing its autonomy (Sanz, Pérez, et al., 2013).
- The robot performs a search and recovery intervention of a known object (black-box, amphora, etc.). This paper is focused in the recovery phase of this last scenario.

With respect to segmentation and modelling of characteristics for object manipulation it is worth mentioning some advances that use 3D environment information to determine a grasp posture. In more controlled land scenarios it is possible to obtain an almost perfect object model, for example, using a rotating table and stereo cameras (Ciocarlie et al., 2011). In the addressed scenario, however, only a single view of the object can be easily obtained and part of the object is unknown. There is an important effort in the 3D information processing field to extract geometric features of an object. In particular, the use of the RANSAC (RANdom SAmple Consensus) (Schnabel, Wahl, & Klein, 2007) method, that estimates parameters of a mathematical model from a set of observed data which contains outliers. The present paper is inspired by García (2009), which uses RANSAC in a 3D representation of an object in order to determine the best available grasp. In Huebner, Ruthotto, and Kragic (2008), the decomposition of complex objects in a series of primitive shapes with which a better grasp can be determined is described. These methods are not directly applicable because there is less 3D information available to marine sensors.

In this paper we present a method able to perform grasping tasks more autonomously in the constrained, yet realistic, problem of grasping cylindrical objects like an amphora, a bottle or a pipe. More generally, it allows to autonomously grasp unknown objects that resemble primitive shapes in the way that the amphora can be considered a cylinder

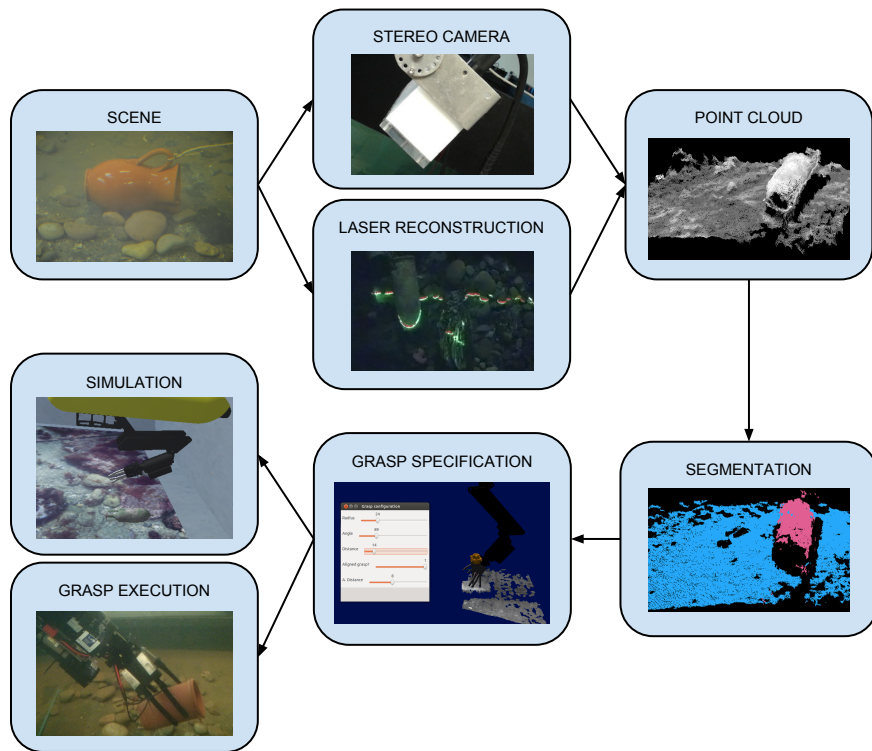


Figure 1. Framework steps from image acquisition to grasp execution.

or an airplane black box resembles a cuboid for the purpose of manipulation. Grasping objects generally requires at least some partial 3D information, which is gathered using stereo vision and laser reconstruction (Prats, Fernández, & Sanz, 2012). The obtained point cloud is then used for planning a grasp, that is then executed fully autonomously by the robot. The framework overview can be seen in Figure 1.

This article is organized as follows: the next section describes the considered scenarios and setups; Section 3 briefly outlines the 3D point cloud acquisition and RANSAC shape fitting algorithm; Section 4 describes the grasp specification with the analytical model of the object and the specification interfaces; Section 5 summarizes the execution step; Section 6 shows the results and finally, further work and conclusions are included in Section 7.

2. Experimental setup

To develop and test the proposed framework, several scenarios have been considered. In the first place, a simulated environment has been used to develop the proposed algorithms using UWSim, an underwater simulator (Prats, Pérez, Fernández, & Sanz, 2012), because it allows working with robots in underwater environments within the ROS (Robot Operating System) architecture (Quigley et al., 2009). The main scenario is illustrated in Figure 2. The mechatronics consists of a virtual model of an underwater robotic arm attached to the Girona 500 AUV (Ribas, Ridaó, Magi, Palomeras, & Carreras, 2011). The arm has 7-DOF (degrees of freedom) and was developed by GraalTech (named here GT-arm) in the context of the TRIDENT project (Sanz, Ridaó, et al., 2013). The end-effector is a jaw to allow a parallel grip. In this way, only an approach vector is needed to specify the grasp. However, this solution has less flexibility than using a dexterous

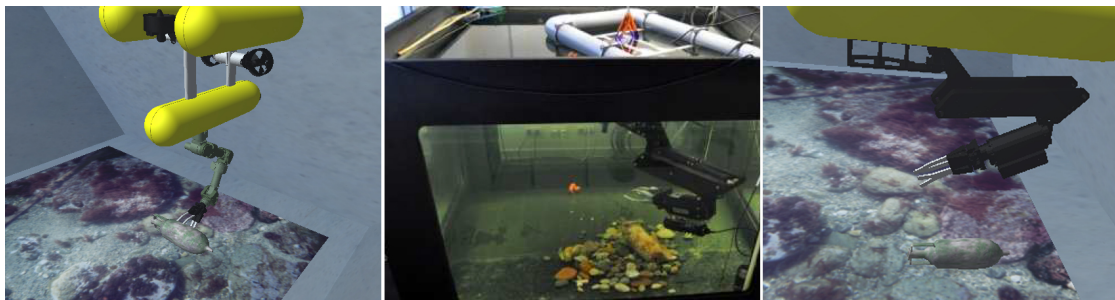


Figure 2. Simulation environment that reproduces the CIRS pool with the GT-arm attached to the Girona 500 AUV (left). Real water tank conditions used for the experiments. The ARM5E arm is attached to a floating platform that is placed in the water in a fixed position (centre). Simulation scene using the ARM5E arm (right).

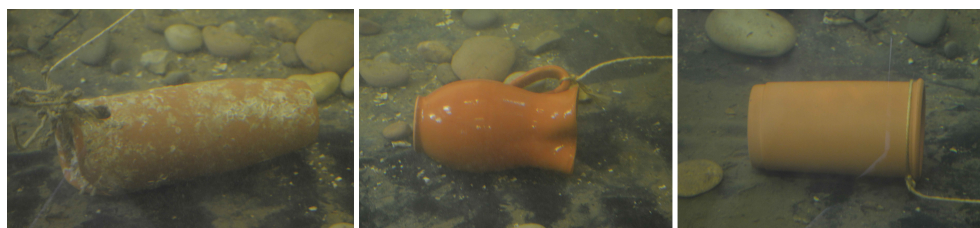


Figure 3. Cylindrical objects used during execution trials. For reference purposes they are named as objects A, B and C.

hand. The vehicle is floating in the CIRS (Centre d'Investigació en Robòtica Submarina, University of Girona) simulated pool and the target object is an amphora lying on a textured floor which simulates the seafloor. This vehicle-arm configuration is not physically available for experimental validation, but its features are optimal for testing purposes in simulation because the arm is redundant and can reach most positions in the desired orientation.

The experimental setup, that is also considered, is based on a 4-DOF underwater robotic arm, in our case the CSIP Light-weight ARM 5E (Fernández et al., 2013), attached to a floating vehicle prototype that remains static (Figure 2). The real scenario consists of a 2m x 2m x 1.5m water tank. The physical target objects (Figure 3) are also cylindrical objects lying on a planar surface surrounded with stones. This arm has also been used in the aforementioned simulation scenario fixed to the Girona 500 AUV to compare the difference between the two arms.

In the real testbed, there exist two vision systems placed looking towards the ground. The first one consists of a Videre Stereo Camera located near the base of the arm inside a sealed case. The second one consists of a monocular underwater camera (Bowtech 550C-AL). This camera is used in conjunction with a laser projector (Tritech SeaStrip) attached to the forearm to obtain a 3D representation of the scene (Prats, Fernández, & Sanz, 2012). As can be seen in Figure 4, a laser stripe is projected and visually segmented to know the 3D coordinates of each point.

It is worth mentioning that the simulated vehicle has access to a simulated version of the same sensors that the real system has, i.e. stereo vision, monocular cameras and a laser projector (Figure 5). Thanks to that, it is possible to simulate the same sources of 3D data that could be used in a real system.



Figure 4. Laser stripe projection over the scene. Camera point of view with laser (green) and its visual segmentation (red) in the left. System overview in the right.



Figure 5. The onboard camera is an example of the simulated sensors.

3. 3D Reconstruction and Segmentation

The first step in order to grasp an object consists of information acquisition about the environment. As stated previously, it can be gathered either using laser stripe reconstruction or a stereo camera (real or virtual in UWSim). The algorithms have been tested using both sources, building a single point cloud which can be processed using Point Cloud Library, PCL (Rusu & Cousins, 2011).

This processing consists of using filters included in PCL: downsampling the point cloud, in order to decrease the number of points using only the most relevant ones; and applying to it an outlier filter, in order to remove the points that are potential wrong values caused by spurious particles or optical reconstruction errors. Decreasing the number of points decreases the computation time of the following steps and increases the robustness of the overall method. With this relevant point cloud, a RANSAC algorithm, described in Schnabel et al. (2007), is used twice to separate the object from the background. First, the background plane is detected with a RANSAC plane fitting algorithm and the resulting parameters are used to remove the plane inliers from the original point cloud. In the next step, other RANSAC algorithm is used to obtain the cylinder parameters associated to the searched object (that is supposed to resemble a cylinder). These algorithms are parameterized to allow fitting quality and performance control. The main parameters are distance to the plane threshold in case of the plane segmentation, and distance to the cylinder and maximum radius in case of the cylinder. The maximum number of iterations can be adjusted too.

The result of these steps is a set of inliers that represent the detected amphora points and the analytical parameters: a point in the obtained model axis, the axis direction and the cylinder radius. This process has been tested separately from the final execution using clouds extracted with all the previous point cloud sources with different degree of success. In general, stereo cameras perform better with good light conditions such as the experimental water tank or the simulation scene while laser reconstruction benefits from darker conditions where the contrast with the green ray is higher (like in deep sea operations or the water tank with lights off and windows closed). Stereo cameras generally produce more dense point clouds, which require more computation time. Different segmentations from virtual and real stereo cameras are shown in Figures 11-15 and will be further explained in the Results section.

4. Grasp specification

Using the cylinder model and the corresponding points (also called inliers) obtained with the RANSAC algorithm, a grasp posture can be specified. To avoid errors, the grasp pose is computed using the most significant points of the cylinder inliers (the 90% nearest to the centre points). The middle point of the cylinder axis is used as a starting point. Then, taking into account the amphora radius and desired approach distance and angle, the grasping end-effector frame is moved away from the starting position. These free variables allow the computation of different grasp frames around the cylinder axis. Two different possibilities appear: (1) using these variables to allow the end-user to set up a grasp with an easy and quick interface or (2) using them to autonomously maximize grasp characteristics such as angle with the floor and stability. While the first approach is really appealing, sometimes the loss of autonomy of the second option causes a huge robustness increase, as the user is the one who decides whether a grasp pose is good enough or not. This differs from the approach used in Sanz, Peñalver, et al. (2013) where the user was selecting a pair of points within a 2D image and then a 3D grasp was automatically generated.

In our first approach, the user modifies the grasp pose in a 3D space, until he considers that it is in a desirable position. This could be quite a difficult task if the interface allows the user to set the grasp in a completely free 3D space. The use of the obtained analytical model to set the grasp approach vector, permits the user to move the end-effector around the cylinder axis and place it in the desired pose really quickly. Two possible grasp configurations are shown in Figure 6.

To increase the system feedback, this interface shows to the user the grasp that he wants to set and the actual position that the arm can reach. After the grasp to be performed has been specified, it is necessary to check whether it is feasible or not. This can be done by computing the inverse kinematics of the whole arm kinematic chain and calculating its reachability. Our approach is to adopt a classical iterative inverse jacobian method. This method is used to compute the nearest reachable pose. If the desired pose is within the workspace of the arm, the end position will be very accurate while the orientation will often be different from the desired orientation.

These kinematic computations are carried out immediately after any movement of the grasp posture, showing the desired grasp goal as well as the reachable grasp posture in a color coded way (Figure 7). Moreover, it is possible to show the user the arm configuration that it would reach. With all this information, the user can decide if the end grasp posture is correct and if the arm is in an optimal configuration with respect to the following task or if it is good for the vehicle stability.

Following the second idea of a more autonomous interface, an interface that generates and evaluates a manifold of grasp postures was also developed (see Figure 8). This method

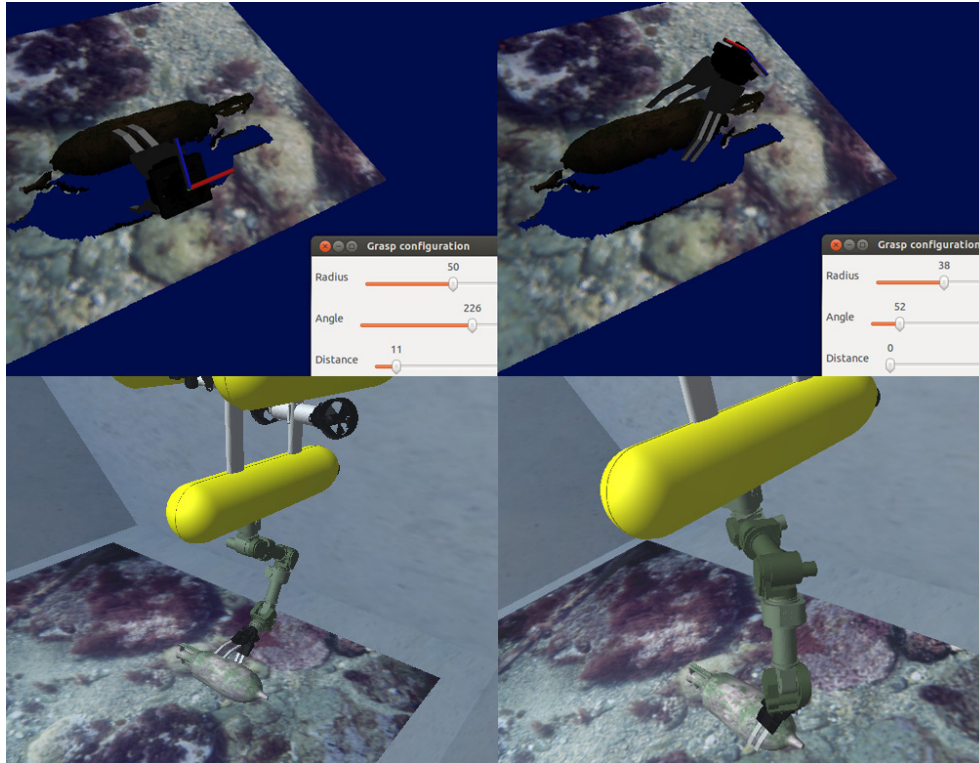


Figure 6. Grasp specification interface showing the two types of grasp: side grasp (top left) and front grasp (top right). UWSim showing the GT-arm in the obtained grasp poses from two grasp specification processes (bottom).



Figure 7. The interface shows the desired pose in orange, the reachable approach pose in green and the pose where the jaw will close in blue (left). It is also possible to see the full arm configuration (right).

generates a manifold of grasp postures using the same distance and angle variables that the user can set up manually with the previous interface. This list is then ranked taking into account the arm geometry and using measures that will be explained below. Then a weighted score is computed in order to sort the grasp list and show the user only the grasps with a higher score. The weights have been estimated based on the importance of these measures although a future study of its impact is recommended. Finally the user can select one among the highest ranked. This method is quicker and easier for the user, but at the same time allows a certain amount of control over the grasping pose. The

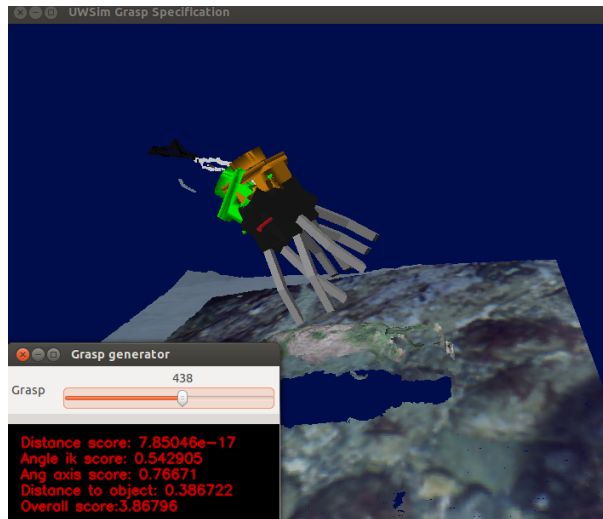


Figure 8. The interface shows the color coded poses and the computed scores.

aforementioned scores could be shown also in the first user centered interface to show how good a grasp is in terms of scores.

This technique uses the following measures to rank the grasp poses:

- Angle of the approach vector z axis with the cylinder axis. This angle should be near to 45 degrees to allow to lift the object properly. This measure is computed over the reachable grasp pose.
- Distance of the grasp frame to the cylinder center. It should leave enough space between the gripper and the object. This measure is also computed over the reachable grasp pose.
- Distance and angle between the initial desired pose and the reachable grasp pose that is the one that is evaluated. These measures correspond to reachability and should be minimized.

5. Grasp execution

When the grasp frame is selected and reachable, it can be executed. This step has been performed in both virtual and real hardware. Figure 6 shows the arm configuration for two possible grasp approaches (side and frontal grasp) in the virtual scenario. With the 7-DOF GT-arm, there are enough degrees of freedom to reach different positions with completely different orientations, nevertheless, the use of the ARM5E arm constrains the possible approach vectors to a great extent. This makes it almost impossible for the user to set orientation variations without including the vehicle in the kinematic chain. For this reason, simulated and real experiments using the limited arm are carried out considering an adequate vehicle-object relative position.

The grasp is executed moving the end-effector directly towards the object center. Again, the ARM5E cannot always perform a straight motion. The developed interfaces also show the end position of the gripper where it closes around the object.

The grasp simulation provided by UWSim is limited in its current development stage, providing only contact physics. To overtake this, the simulator offers a capability named virtual object picker, that attaches the object to the inner part of the gripper when a given distance threshold is reached. Although it does not use friction physics yet to perform the grasp, it lets the user visualize how it would perform in a real scenario prior

to the real execution. To increase the capabilities of UWSim in this sense is out of the scope of this paper.

In the virtual experiments, the simulator acts as a controller to move the arm. The use of a real arm also requires the use of advanced controllers that issue commands to the joint actuators. The low-level control architecture for both systems was implemented in C++ and makes use of ROS for inter-module communications. The separate kinematic module accepts either cartesian or joint information (i.e. pose, velocity) and it is capable of computing kinematics and issuing commands to the virtual and real arm interchangeably. In the specification stage, velocity control is used to move the end-effector towards the desired pose. Then, in the execution stage, cartesian velocity towards the center of the object is applied. When it reaches the end pose, it closes the gripper, using as feedback the electric current of the arm (with real hardware) or virtual sensors (in simulation). Finally, the robot is commanded to a folded configuration to carry the object. This can be done with both the UWSim and the real hardware using the same framework.

6. Results

This section shows the segmentation, specification and execution results of this research. The grasp specification framework presented in this paper has been tested with various image sources and has been used to execute different grasps with the described vehicle in both simulation and real environments.

6.1. Execution and specification results

The main results are the grasp execution capabilities in the real testbed, as they demonstrate that this framework is well suited for underwater environments and that can be used even with a limited real arm. Three different objects have been grasped (see Figure 9), even when their shapes are not completely cylindrical. A complete execution can be seen in an online video¹, where the robot approaches from the specified position to the object and carries the objects. These trials give us a sense of the integration level of the perception, specification, user interaction, kinematics and control systems of the robot with this framework. The execution in the real scenario demonstrated that this framework is well suited for underwater environments and can be used with a limited arm at least in restricted scenarios.

Table 1 summarizes the segmentation results obtained through these trials, showing also the results of the simulated amphora for reference. It can be seen that the grasped objects results are very different, but the objects were grasped properly thanks to the user supervision. The plane segmentation is an easy task that was always performed successfully (as the scenario is almost the same) and relatively quickly. On the contrary, cylinder segmentation success varies with the quality of the cloud (from 70% to 18.9%). Table 2 shows the physical properties of the objects, which are strongly related with the amount of points obtained by the stereo camera that represent the object (from 12.5% of the cloud to only 4%). For this reason, bigger objects are represented by more points and are easier to segment. Another factor that affects the reconstruction is texture. Objects B and C have a uniform surface so they are reconstructed worse. Moreover, the object C has a matt surface and is even harder to reconstruct (light reflections improve reconstruction for object B). For this reason, its segmentation (that was done with an external light source to improve the reconstruction) is the worst of all. It can also be seen that the

¹Grasping execution of three different objects: <http://youtu.be/xRPw2TAH8Fg>

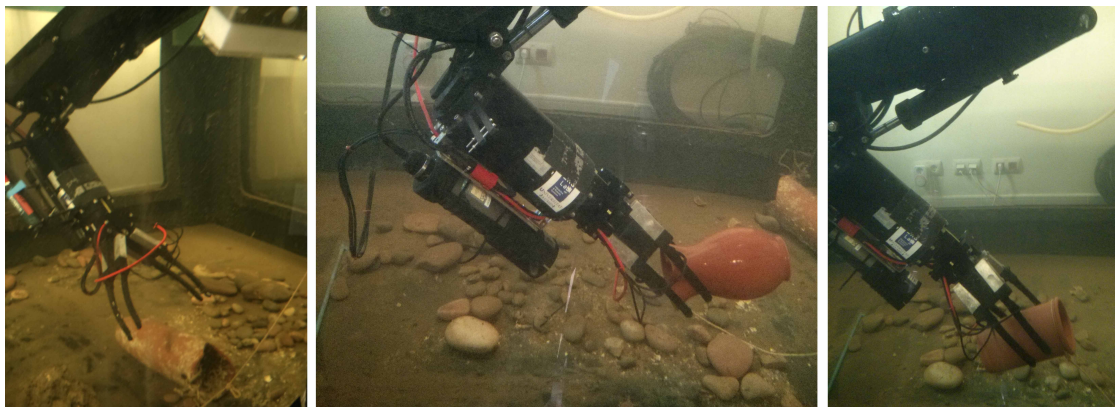


Figure 9. Three different objects grasped with this framework and the ARM5E arm. For reference purposes they are named as objects A, B and C.

Table 1. Segmentation accuracy and execution time for plane and cylinder segmentation. Three different real objects and a virtual object are compared.

Objects	Object A	Object B	Object C	Sim. Obj.
Cloud points	144,529	146,760	148,401	642,583
Object points (groundtruth)	18,085	9,012	5,933	12,953
Points corresponding to the object (%)	12.5	6.1	4.0	2.0
Segmented plane points	121,550	124,624	135,196	620,561
Plane seg. succes (%)	96.1	90.5	94.9	98.6
Plane seg. time (s)	0.12	0.12	0.10	0.35
Segmented cylinder points	12,657	4,215	1,113	11,774
Cylinder seg. succes (%)	70.0	46.8	18.8	90.9
Cylinder seg. time (s)	0.12	0.41	2.12	0.10

Table 2. Physical properties of the segmented objects.

Objects	Object A	Object B	Object C	Sim. Obj.
Main surface property	Textured	Glossy	Matt	Textured
Light conditions	Daylight	Daylight	Daylight+lamp	Daylight
Cylinder height (cm.)	36	23	21	52
Cylinder weight (cm.)	16	14	13	16
Surface (cm ² .)	3,619	2,023	1,715	5,228
Points/surface ratio	4.99	4.45	3.46	2.48

segmentation time for the cylinder model is inversely proportional to the difficulty of the segmentation (measured by the segmentation success).

The grasp specification interface used for the experiments (Figure 6) can be seen in an online video². As described in the previous sections, the grasp specification interface allows the visualization of the grasp possibilities proposed for a cylindrical object, where a non expert user can guide the system to a proper grasp starting position. The online reachability check with the real hardware³ shows to the user the configuration of the arm for the desired end-effector pose (Figure 10).

6.2. Segmentation results

The segmentation stage has been analyzed with real and virtual images from both the stereo camera and the laser reconstruction method. Several point clouds have been segmented in order to analyze the parametrization of the RANSAC algorithm. Depending

²Grasp specification interface with online reachability test: <http://youtu.be/cXUz1Wx9Ycg>

³Grasp specification interface with real hardware: <http://youtu.be/hsztMWqbn9U>

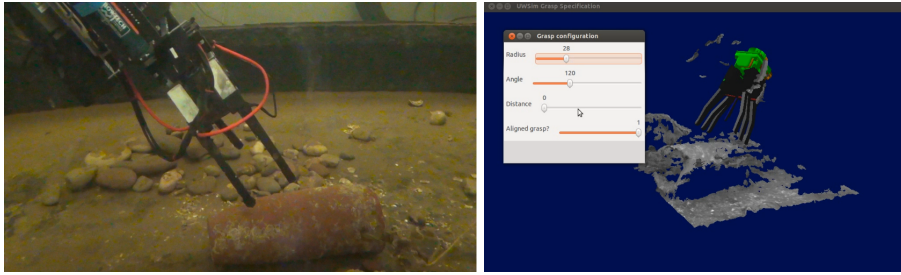


Figure 10. Online grasp specification. The AMR5E arm (left) is moved the user specified position (right).

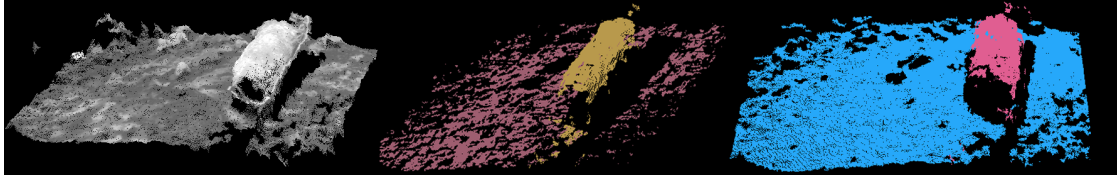


Figure 11. Point clouds from the Videre Stereo Camera. Segmentation with 10 cm. to the cylinder threshold (center). Segmentation with 5 cm. threshold (right).

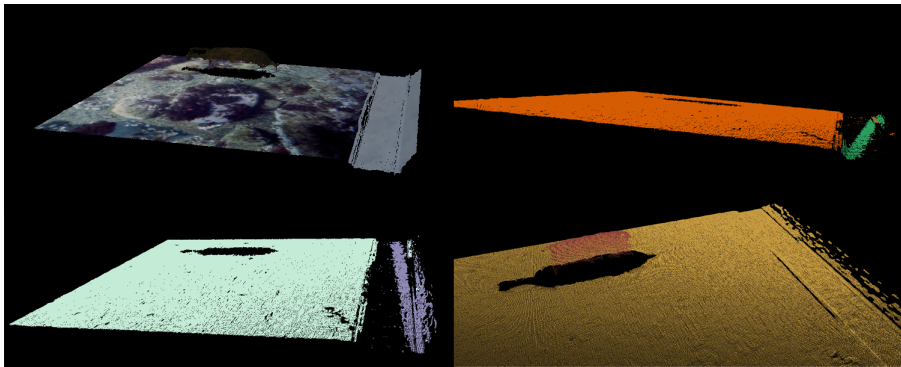


Figure 12. Point clouds from the virtual stereo camera: extreme scenario where accurate segmentation parameters are needed.

on the distance to the plane threshold, distance to the cylinder threshold and maximum cylinder radius, a better segmentation can be obtained. This step is critical for the final grasp posture quality.

In Figure 11, the point cloud obtained with the real stereo camera is segmented with good results and varying the threshold results in some point of the floor considered as part of the cylinder. A similar effect can be seen in the simulator experiment (Figure 12), where, with a certain plane threshold, some part of the pool floor is not considered as part of the plane, resulting in an error in the following cylinder segmentation. This, though, can be considered as an extreme case because the floor corner has lots of points and its curved shape fits well in the cylinder shape.

With regard to the laser reconstruction, Figure 13 shows that this method generates more occlusions than stereo vision due to the fact that some areas lightened by the laser are not seen by the camera and some areas cannot be covered by the laser. In this context, a lower distance to cylinder threshold allows the algorithm to segment only the points that are really part of the object. Another possibility could be to increase the plane threshold in order to extract more plane inliers.

Although the laser reconstruction within the UWSim (Figure 14) obtains worse point clouds than the virtual stereo vision, the segmentation results are still excellent because

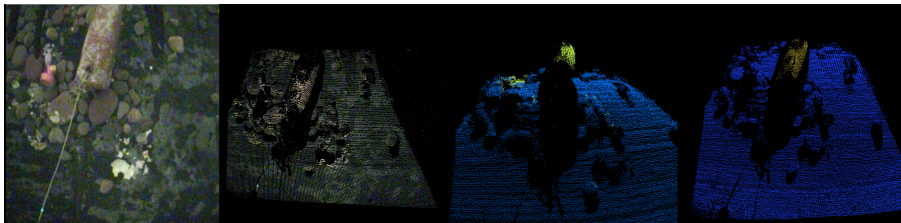


Figure 13. Point clouds from real laser reconstruction and segmentation with varying cylinder segmentation thresholds.

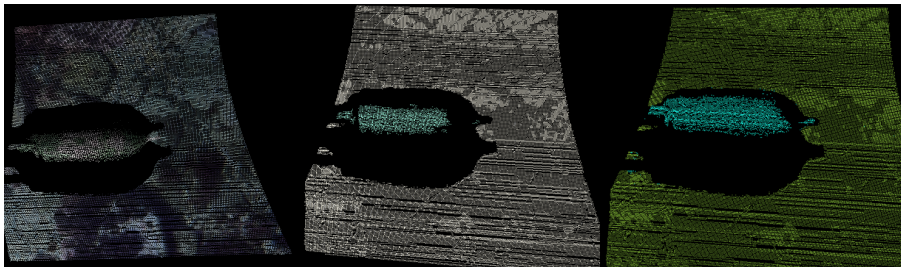


Figure 14. Point clouds from virtual laser reconstruction and segmentation with different cylinder segmentation thresholds.

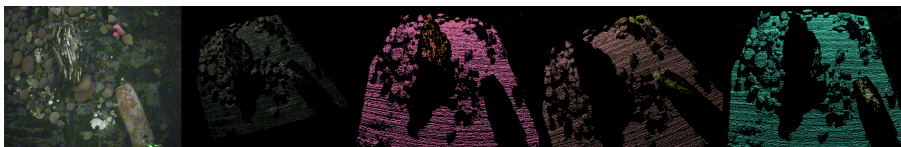


Figure 15. Point clouds from real laser segmentation of a scene with a clay jar and a wooden trunk. Low thresholds (center) lead the algorithm to segment bigger objects instead of cylindrical shaped ones.

Table 3. Plane segmentation time and percentage of found inliers using a 10 cm threshold with RANSAC.

Source	Num. of points	Seg. Time (s)	% of plane inliers
Real Laser	51,971	0.058	77
Real Stereo	307,200	0.257	59
Virtual Stereo	786,432	0.414	78
Virtual Laser	60,326	0.043	90

the cylindrical shape is well preserved by the reconstruction.

The previous results seem to indicate that it is better to use a higher threshold to assure that the cylinder is found. However, if two different objects are placed in the scene and the threshold is not high enough, the object with more points can be segmented instead of the smaller more cylindrical object (Figure 15).

With respect to the segmentation execution time, the following results show that it is not possible to use them in real time if the object is moving. The number of iterations of the algorithm constrains the upper limit of time, which in this case has been up to 0.4 seconds. For this analysis, a distance to the plane threshold of 10 cm and 5 cm has been considered for various executions. Tables 3 and 4 show the point cloud size, the mean segmentation time and the mean inliers percentage. In Table 3, where a higher threshold is used, the relationship between the point cloud size and execution time is almost linear. However, in Table 4, execution time depends on the segmentation complexity. When the threshold is lowered, the time spent is generally higher. On the contrary, with more percentage of inliers, the computing time is lessened.

Table 4. Plane segmentation time and percentage of found inliers using a 5 cm threshold with RANSAC.

Source	Num. of points	Seg. Time (s)	% of plane inliers
Real Laser	51,971	0.270	47
Real Stereo	307,200	1.066	37
Virtual Stereo	786,432	0.397	77
Virtual Laser	60,326	0.056	82

Table 5. Cylinder segmentation time using RANSAC depending on the threshold.

Source	10 cm threshold		5 cm threshold	
	Plane outliers	Time (s)	Outliers	Time (s)
Real Laser	11,536	1.385	27,497	1.810
Real Stereo	124,850	0.476	193,518	4.800
Virtual Laser	5,860	0.022	10,688	0.080
Virtual Stereo	171,560	0.220	174,087	0.180

Finally, Table 5 shows the results of the next step, which consists in the cylinder segmentation for both plane segmentation cases, showing the number of points after removing the plane inliers. The results are very different between reconstruction sources. With real sources, a looser threshold removes more points from the original cloud, thus making the segmentation quicker. Using virtual sensors, though, cylinder segmentation is always relatively fast because points adjust very well to the model, even more in the case of the virtual laser segmentation where the amount of points is lower. These results indicate that the segmentation process is flexible but not fast enough to be executed at a reasonable frequency like 5 Hz. In this case, other tracking techniques should be used to follow the object movements.

With regard to the execution time, it is noticeable that the maximum number of iterations imposes a limit to this time. However, that limit is not proportional only to the maximum execution time, as each iteration consumes a variable amount of time depending on the cloud sizes (because the full cloud is traversed at each iteration) and the analytic model to be estimated. Moreover, if a model is found with a low error, the maximum number of iterations is not reached. For example, using the cloud with two objects, which has over 130,000 points, the segmentation time for 400 iterations is 3.8 seconds, while for 50 iterations the segmentation time is 1 second and it is still able to find a suitable model. The plane segmentation can be done in 1 second for 100 iterations or in 0.12 seconds for 10 iterations. In both cases, increasing the number of iterations over 400 and 100 respectively, doesn't increase the execution time because the model is found before reaching the limit.

7. Conclusions and future work

After very successful research achievements through previous projects like TRIDENT, following a semi-autonomous strategy, the ongoing project TRITON follows a more autonomous strategy. This paper presents a new framework to improve the autonomy of manipulation of unknown structured objects in underwater environments. The main application field considered has been archaeology, where the use of the presented framework and the future developments yet to come could provide significant benefits in order to discover new artifacts or recover very valuable objects easily. The experimental validation has been focused on grasp tasks in the constrained, yet realistic, problem of grasping unknown cylindrical objects such as an amphora, a pipe, a barrel or a bottle; although this framework can be further developed to recognize other primitive shapes. These shapes

could be spheres, cuboids or even more complex models that can be approximated with a set of primitive shapes as shown in Huebner et al. (2008). With that flexibility, the system could be capable of specifying the grasp of other objects without the need of knowing the exact model of the object but only its approximate shape. However, this flexibility will have the drawback of more computation time in these time sensitive tasks.

The steps to obtain a model, specify a grasp and execute it have been described and further validated in real and practical scenarios, but as new strategies arise, they have to be tested carefully. The real testbed for the experiments has been shown in Figure 2. The optical devices and segmentation algorithms have been extensively tested, and real trials have been performed, thus demonstrating the use of this specification procedure.

With these experiments, other issues have arisen, such as the grasp execution control that can be improved using tactile sensor feedback. With respect to the supervised interface, it is interesting to remark that future developments will deal with learning from the user's decisions to autonomously get an adequate grasp. Another possibility would be the use of reinforced learning based on the success or quality of the grasp.

Regarding other future improvements, the use of coupled AUV-Arm kinematic chain will generate a system that would have more than six available degrees of freedom so that all the framework potential could be exploited. Kinematic solvers for this robot should prefer the use of the arm instead of the vehicle to reach a certain configuration in an energy efficient way.

Acknowledgement

This research was partly supported by Spanish Ministry of Research and Innovation DPI2011-27977-C03 (TRITON Project), by Foundation Caixa Castelló Bancaixa PI-1B2011-17, by Universitat Jaume I PhD grants PREDOC/2012/47 and PREDOC/2013/46, and by Generalitat Valenciana PhD grant ACIF/2014/298.

References

- Ciocarlie, M., Pantofaru, C., Hsiao, K., Bradski, G., Brook, P., & Dreyfuss, E. (2011, June). A side of data with my robot. *Robotics Automation Magazine, IEEE*, 18(2), 44-57.
- Fernández, J., Prats, M., Sanz, P. J., García, J., Marín, R., Robinson, M., ... Ridao, P. (2013). Grasping for the seabed: Developing a new underwater robot arm for shallow-water intervention. *Robotics Automation Magazine, IEEE*, 20(4), 121-130.
- FP7-PANDORA. (2012). *Persistent Autonomy through learNing, aDaptation, Observation and Re-plAnning (PANDORA)*. <http://persistentautonomy.com/>.
- García, S. (2009). *Fitting primitive shapes to point clouds for robotic grasping* (Master's thesis). Royal Institute of Technology, Stockholm, Sweden.
- Hexmoor, H., McLaughlan, B., & Tuli, G. (2009). Natural human role in supervising complex control systems. *Journal of Experimental & Theoretical Artificial Intelligence*, 21(1), 59-77.
- Huebner, K., Ruthotto, S., & Kragic, D. (2008, May). Minimum volume bounding box decomposition for shape approximation in robot grasping. In *IEEE International Conference on Robotics and Automation* (p. 1628-1633).
- Marani, G., Angeletti, D., Cannata, G., & Casalino, G. (2000, June). On the functional and algorithmic control architecture of the amadeus dual arm robotic workcell. In *World Automation Congress*. Hawaii, USA.

- Palomeras, N., Peñalver, A., Massot-Campos, M., Vallicrosa, G., Negre, P. L., Fernández, J. J., ... Palomer, A. (2014, Sept). I-AUV docking and intervention in a subsea panel. In *Intelligent Robots and Systems (IROS 2014), IEEE/RSJ International Conference on* (p. 2279-2285).
- Prats, M., Fernández, J. J., & Sanz, P. J. (2012). Combining template tracking and laser peak detection for 3D reconstruction and grasping in underwater environments. In *Intelligent Robots and Systems (IROS 2012), IEEE/RSJ International Conference on* (p. 106-112).
- Prats, M., Pérez, J., Fernández, J. J., & Sanz, P. J. (2012, Oct). An open source tool for simulation and supervision of underwater intervention missions. In *Intelligent Robots and Systems (IROS 2012), IEEE/RSJ International Conference on* (p. 2577-2582).
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., ... Ng, A. Y. (2009). ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*.
- Ribas, D., Ridao, P., Magi, L., Palomeras, N., & Carreras, M. (2011, June). The Girona 500, a multipurpose autonomous underwater vehicle. In *IEEE/OES OCEANS 2011* (p. 1-5). Spain.
- Rusu, R., & Cousins, S. (2011, May). 3D is here: Point Cloud Library (PCL). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (p. 1-4).
- Sanz, P. J., Peñalver, A., Sales, J., Fornas, D., Fernández, J. J., Perez, J., & Bernabé, J. A. (2013, Oct). GRASPER: A multisensory based manipulation system for underwater operations. In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Manchester, UK: IEEE.
- Sanz, P. J., Pérez, J., Peñalver, A., Fernández, J. J., Fornas, D., & Sales, J. (2013, Sep). GRASPER: HIL simulation towards autonomous manipulation of an underwater panel in a permanent observatory. In *OCEANS'13 MTS/IEEE conference*. San Diego, CA.
- Sanz, P. J., Ridao, P., Oliver, G., Casalino, G., Petillot, Y., Silvestre, C., ... Turetta, A. (2013). TRIDENT: An european project targeted to increase the autonomy levels for underwater intervention missions. In *OCEANS'13 MTS/IEEE conference*. San Diego, CA.
- Schnabel, R., Wahl, R., & Klein, R. (2007, June). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2), 214-226.