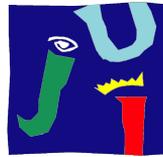


Universitat Jaume I
Escola de Doctorado



UNIVERSITAT
JAUME·I

Tesis Doctoral

APLICACIÓN DE LA VISIÓN
ARTIFICIAL A LA MANIPULACIÓN
ROBÓTICA EN EL CONTEXTO DE LA
INTERVENCIÓN SUBACUÁTICA

Autor: David Fornas García

Directores: Enric Cervera Mateu
Pedro J. Sanz Valero

Castellón de la Plana, Noviembre 2018



Programa de Doctorado en Informática

Escuela de Doctorado de la Universitat Jaume I

**Aplicación de la Visión Artificial a la Manipulación Robótica
en el Contexto de la Intervención Subacuática**

**Memoria presentada por David Fornas García para optar al
grado de doctor por la Universitat Jaume I**

David Fornas García Enric Cervera Mateu - Pedro J. Sanz Valero

Castellón de la Plana, Noviembre de 2018

Financiación recibida

Esta tesis ha sido financiada bajo la beca predoctoral de la Universitat Jaume I PREDOC/2013/46. Los recursos y materiales utilizados han sido proporcionados por los proyectos nacionales TRITON (DPI2011-27977-C03) y MERBOTS (DPI2014-57746-C3-1-R), financiados por el Ministerio de Economía y Competitividad.

A Sandra, mis padres y mis hermanos.

Agradecimientos

El resultado de 5 años de trabajo en el laboratorio Interactive and Robotic Systems Lab (IRS Lab) se ha plasmado en esta tesis. En primer lugar, me gustaría agradecer a mis directores Enric Cervera y Pedro Sanz, así como a Raúl Marín por su orientación a lo largo de este tiempo. Tampoco quiero olvidarme de Jose Vicente Martí, Gabriel Recatalà y Federico Thomas.

En segundo lugar, quiero dar las gracias a todos los compañeros del laboratorio que me han ayudado tanto, empezando por Javi con la mecatrónica, Toni con el robot, Javier con el simulador y Juan Carlos con las interfaces. Hemos aprendido mucho en nuestros experimentos en Gerona. También agradecer a grandes mentores como Ángel, Mario, Xavi, Marco, Jorge y Antonio. No quiero olvidarme de Diego, Majd ni por supuesto de Carlos, con el que llevo años en caminos paralelos. Finalmente, también quiero agradecer la ayuda de los compañeros de los laboratorios en Gerona, Islas Baleares y Viena, de los que he aprendido mucho.

Gracias Sandra por darme fuerzas en todo momento. A mi familia, que me ha estado alentando desde siempre. A mis amigos, que me han escuchado aún sin saber muy bien de qué les hablaba. Finalmente, y seguro que me olvido de alguien, agradecer a Sandra Catalán y Vicente Calpe, que con sus experiencias en el doctorado tanto me han ayudado.

Resumen

La manipulación robótica en un entorno submarino representa un gran desafío, ya que las condiciones físicas del medio acuático dificultan la percepción y el control de los robots. En los últimos años ha habido un notable avance en este ámbito gracias a la irrupción de los Vehículos Autónomos Submarinos de Intervención (*Intervention Autonomous Underwater Vehicles*, I-AUV). El uso de este tipo de vehículos resulta necesario para avanzar en la robótica submarina, ofreciendo funciones autónomas y menores costes respecto al uso de Vehículos Operados Remotamente (*Remotely Operated Vehicles*, ROV). El avance en este campo promete un notable impacto en la sociedad, facilitando operaciones en la industria del petróleo y el gas, en la investigación del océano, en la arqueología submarina o en las misiones de salvamento, aparte de los posibles usos militares o recreativos.

En esta tesis se describe un sistema integral para la manipulación de objetos guiada por visión en escenarios subacuáticos en el contexto de la arqueología. El sistema propuesto consiste en la reconstrucción en 3D del entorno, el modelado geométrico aproximado del escenario y los objetos que rodean al robot, la especificación y supervisión del agarre utilizando métodos de Interacción Humano-Robot (*Human-Robot Interaction*, HRI), y la ejecución del agarre. Se ha demostrado el uso de los diferentes componentes en simulación y en entornos reales de dificultad creciente. Así, se han realizado experimentos en el laboratorio Interactive and Robotic Systems Lab (IRS Lab) en un tanque de agua dulce, pruebas de integración en la piscina del Centro de Investigación en Robótica Submarina (CIRS) y experimentos de campo en el mar dentro del puerto de Sant Feliu de Guíxols.

El sistema desarrollado se ha adaptado a las necesidades y capacidades específicas del manipulador robótico ARM5E. Este manipulador se utiliza en este trabajo tanto de forma desacoplada como anclado a la plataforma móvil GIRONA500. Por otro lado, el sistema es totalmente modular y se puede configurar para utilizar otro I-AUV. Además, el *software* desarrollado se ofrece a la comunidad científica en forma de código libre en un repositorio público.

Las contribuciones de la presente tesis son: el sistema integral para la manipulación guiada por visión en entornos subacuáticos; los métodos de segmentación, estimación y seguimiento de objetos; el sistema de especificación del agarre supervisado; la interfaz de monitorización para intervenciones submarinas, y la aplicación de dicho sistema a la manipulación y a la limpieza de objetos parcialmente enterrados.

Abstract

Grasping is a major challenge for underwater robots due to the adverse effects of underwater physics on its perception and control systems. In the last few years, the use of Intervention Autonomous Underwater Vehicles (I-AUV) has pushed forward in this field by offering autonomous capabilities and lowering costs with respect to Remotely Operated Vehicles (ROV). The latest developments in this area are promising, with an impact in areas such as oil and gas industry, marine sciences, archaeology and rescue missions. Further applications include leisure or military uses.

This thesis describes an integral framework for visually guided manipulation of objects in underwater scenarios in the archaeology case of use. The proposed framework is used to perform the reconstruction of a 3D point cloud from the robot environment, compute the geometric model of the objects and background of the scene, specify and supervise the grasping through Human-Robot Interaction (HRI), and execute the grasping. The different components have been validated in simulation and in increasingly complex real scenarios. Experiments have been performed in the water tank of the Interactive and Robotic Systems Lab (IRS Lab), in the pool of the Centro de Investigación en Robótica Submarina (CIRS) with the integration of the vehicle and in seawater at the harbour of Sant Feliu de Guíxols.

This framework has been developed taking into account the restrictions and capabilities of the ARM5E manipulator. It has been used separately and attached to the GIRONA500 mobile platform. However, the system is modular and it can be configured to be used with other I-AUVs. Moreover, the software will be shared with the scientific community as free software in a public repository.

The contributions of this thesis are: the integral system for visually guided manipulation in underwater scenarios, the reconstruction, segmentation, pose estimation and tracking methods, the supervised grasp specification system, the monitoring interface for underwater interventions and the use of the system to manipulation and dredging in order to recover objects and clean buried artifacts.

Índice de contenidos

Acrónimos	xxv
Nomenclatura	xxvii
1. Introducción	1
1.1. Proyectos de investigación en el contexto de las intervenciones sub- marinas	3
1.2. Proyectos de investigación del IRS Lab	7
1.2.1. RAUVI	8
1.2.2. TRIDENT	9
1.2.3. TRITON	10
1.2.4. MERBOTS	10
1.2.5. TWINBOT	11
1.3. Contexto de la arqueología submarina	12
1.4. Objetivos	14
1.5. Estructura	15
2. Agarre guiado por visión: metodología, validación y arquitectura del sistema	17
2.1. Motivación	18
2.2. Estado del arte en la manipulación basada en visión 3D	18
2.2.1. Estimación y Especificación en 3D	20
2.3. Metodología del sistema de manipulación	21
2.4. Metodología de validación experimental	22
2.5. Definición de los escenarios de validación	23
2.5.1. Validación en simulación	23
2.5.2. Validación en el laboratorio IRS Lab	24
2.5.3. Validación en la piscina del CIRS	26
2.5.4. Validación en experimentos en el mar	26
2.6. Experimentos propuestos	27
2.7. Arquitectura <i>hardware</i> del sistema	28
2.7.1. Brazo robótico ARM5E	29
2.7.2. Plataforma móvil GIRONA500	35
2.7.3. Equipos de cálculo y visión	36
2.8. Arquitectura <i>software</i> del sistema	36

2.8.1. Simulación con UWSim	39
2.9. Discusión y conclusiones	41
3. Problemática inherente a la visión subacuática	43
3.1. Motivación	43
3.2. Visión artificial en el entorno subacuático	44
3.3. Sensores 3D comerciales	46
3.4. Reconstrucción estéreo	47
3.4.1. Algoritmos de correspondencia	49
3.5. Reconstrucción mediante segmentación láser	50
3.6. Mejora de la reconstrucción mediante filtrado	51
3.7. Comparativa de nubes de puntos	54
3.8. Registro de nubes de puntos	56
3.9. Calibración visual del manipulador	56
4. Estimación de la geometría de la escena	59
4.1. Motivación	59
4.2. Metodología de Segmentación y Estimación	60
4.2.1. Trabajos previos en manipulación subacuática	61
4.3. Segmentación mediante RANSAC	61
4.3.1. Obtención de características del objeto con RANSAC	62
4.4. Agrupamiento de objetos	64
4.5. Estimación de Simetrías	66
4.5.1. Búsqueda de Simetrías	68
4.6. Obtención de características mediante Momentos de Inercia	70
4.7. Obtención de características de una caja con RANSAC	71
4.8. Obtención de características mediante Supercuádricas	72
4.9. Seguimiento de objetos	74
4.10. Resultados y comparativa de algoritmos de Estimación	76
4.10.1. Resultados en simulación	76
4.10.2. Resultados de los experimentos de MERBOTS en la piscina	76
4.10.3. Resultados de los experimentos de MERBOTS en el mar	77
4.10.4. Resultados en el laboratorio IRS Lab	83
4.10.5. Comparativa de la caracterización del ánfora con RANSAC	92
4.10.6. Estimación utilizando otros sensores	95
4.10.7. Escenas de caracterización de múltiples objetos	96
5. Especificación del agarre	99
5.1. Motivación	99
5.2. Especificación del agarre	100
5.2.1. Especificación heurística	100
5.2.2. Generación y Clasificación de Hipótesis de agarre	101
5.2.3. Especificación analítica del agarre	103
5.3. Interfaces en la robótica submarina	106
5.4. Interacción Humano-Robot en MERBOTS	108
5.5. Interfaz para la supervisión de la manipulación	109
5.5.1. Especificación en Modo Geométrico	111
5.5.2. Especificación en Modo Interactivo	111

5.6. Especificación en los escenarios de intervención	116
6. Ejecución del agarre y Resultados	121
6.1. Motivación	121
6.2. Metodología de Ejecución de la tarea	122
6.3. Control del robot en el espacio de la tarea	124
6.4. Control del robot	125
6.4.1. Controlador basado en Prioridad de Tareas	127
6.5. Manipulación submarina: Aplicación para el desenterrado	127
6.6. Resultados de Ejecución	130
6.6.1. Experimentos de manipulación en el laboratorio IRS Lab	130
6.6.2. Experimentos de manipulación en el CIRS	132
6.6.3. Experimentos de manipulación en el mar	135
7. Conclusiones y Líneas de futuro	137
7.1. Conclusiones	137
7.2. Aplicaciones potenciales	139
7.3. Contribuciones	140
7.4. Publicaciones científicas	141
7.5. Líneas de futuro	142

Índice de figuras

1.1.	Las operaciones en el mar. De izquierda a derecha, de arriba a abajo, se llevan a cabo por buzos profesionales, ROV, AUV y I-AUV.	2
1.2.	De izquierda a derecha, el concepto de los proyectos SAUVIM, DEX-ROV y AMADEUS.	4
1.3.	Aproximaciones más recientes a la manipulación submarina. De izquierda a derecha y de arriba a abajo: el robot OCEAN ONE, el ROV AQUANAUT, el robot CRABSTER y el proyecto PANDORA.	5
1.4.	Escenarios de intervención de RAUVI (recuperación de una caja, izquierda) y TRITON (operación de un panel, derecha).	8
1.5.	El concepto del proyecto TRIDENT se divide en exploración (izquierda) e intervención (centro). El resultado de la recuperación de una caja negra se muestra a la derecha.	9
1.6.	El concepto del proyecto MERBOTS (izquierda) y el proyecto TWINBOT (derecha).	11
1.7.	Aplicaciones para la arqueología submarina en el proyecto Venus (izquierda) y un pecio en la isla de Chios (derecha).	13
2.1.	Metodología de validación experimental por fases.	23
2.2.	Escenarios de los experimentos simulados. El AUV GIRONA500 usa el brazo ARM5E. Escenario con un fondo arenoso y una vasija (izquierda); con el suelo texturizado y una vasija (centro); y con un modelo de caja negra (derecha).	24
2.3.	Experimentos en el tanque de agua del IRS Lab. El brazo robótico se acopla a una plataforma. La cámara estéreo se instala en la base y el proyector láser lineal en el brazo.	25
2.4.	Instalaciones del CIRS (izquierda). Centro de control con visión de la piscina (derecha).	26
2.5.	Los experimentos realizados en el mar tienen lugar en el interior del puerto de Sant Feliu de Guíxols.	27
2.6.	Objetos reales utilizados en el laboratorio. Se han utilizado cajas y objetos cerámicos con formas variadas.	28
2.7.	Brazo robótico eléctrico ARM5E con 4 GDL (izquierda). Efector final (derecha).	29
2.8.	Arquitectura de control del ARM5E.	30

2.9. Modelo 3D del brazo robótico con los sistemas de referencia de cada articulación.	31
2.10. Evolución del consumo del brazo ARM5E durante el procedimiento de calibración.	32
2.11. Evolución del consumo del brazo ARM5E durante el movimiento.	33
2.12. Evolución del consumo del brazo ARM5E durante la manipulación.	34
2.13. La mano UJI-ONE puede manipular objetos (izquierda) y herramientas tipo T (derecha).	35
2.14. El I-AUV compuesto por el GIRONA500 y el brazo ARM5E.	36
2.15. Estructura modular del sistema propuesto.	37
2.16. Estructura del repositorio de código libre ofrecido.	38
2.17. El simulador UWSim puede utilizar múltiples robots y manipuladores diferentes.	40
3.1. Efectos físicos que deterioran las imágenes subacuáticas: absorción (izquierda), dispersión y atenuación (derecha).	45
3.2. Imágenes tomadas con las cámaras DUO M (izquierda), Videre (centro) y Bumblebee2 (derecha).	47
3.3. Imágenes de la cámara virtual en diferentes escenas.	47
3.4. Reconstrucción 3D de las escenas en UWSim mediante visión estéreo.	48
3.5. Encapsulado de las cámaras estéreo Videre (izquierda) y Duo 3D (derecha) utilizadas en el laboratorio.	49
3.6. La reconstrucción láser utiliza una cámara y un proyector láser lineal.	50
3.7. Reconstrucción láser con imágenes reales (izquierda) y simuladas (derecha).	51
3.8. Filtro temporal de nubes de puntos con diferente iluminación. Resultado del filtrado (arriba, derecha).	52
3.9. Filtrado de nubes de puntos: nube original (arriba, izquierda), <i>Statistical Outlier Removal</i> (arriba, derecha), filtro bilateral (abajo, izquierda) y reescalado <i>voxel grid</i> (abajo, derecha).	53
3.10. Nube de puntos obtenida con la cámara estéreo Videre. Sensibilidad ISO 100 (arriba) e ISO 400 (abajo). De izquierda a derecha se usa <i>Block Matching</i> , <i>Block Matching</i> con acumulación y <i>Semi-Global Block Matching</i>	54
3.11. Nube de puntos obtenida con la cámara estéreo DUO M. Imagen de la cámara (izquierda), reconstrucción con <i>Block Matching</i> (centro) y con <i>Semi-Global Block Matching</i> (derecha).	55
3.12. Reconstrucción estéreo de los diferentes objetos presentados.	55
3.13. Detección de marcadores para determinar la posición del manipulador.	57
4.1. Componentes del módulo de Segmentación y Estimación.	60
4.2. Segmentación de un plano y un cilindro mediante RANSAC.	63
4.3. Resultado del agrupamiento por distancia Euclídea.	66
4.4. Aplicación de la simetría planar (izquierda) y axial (derecha). En rojo la fuente, en azul la proyección de los puntos sobre el eje o el plano y en verde los puntos generados.	67
4.5. Componentes del método de estimación de planos y ejes de simetría.	68
4.6. Estimación de la simetría planar en diferentes objetos.	70

4.7. Escena mejorada utilizando la estimación de simetría. En azul los puntos añadidos.	71
4.8. Visualización con UWSim de la estimación de la geometría de una maqueta de caja negra. Modelo (blanco) sobre la nube de puntos. . .	72
4.9. Evolución de las superelipsoides según sus parámetros ϵ_1 y ϵ_2	73
4.10. Varios clústeres dan lugar a diferentes ajustes de superelipsoides. . .	74
4.11. Algoritmos de seguimiento en 2D basados en: características (izquierda), seguimiento de bordes (centro) y plantillas (derecha).	75
4.12. Estimación de la vasija y la caja en simulación.	77
4.13. Posición estimada en la piscina del CIRS utilizando reconstrucción láser.	78
4.14. Posición estimada en la piscina del CIRS utilizando reconstrucción estéreo.	78
4.15. Estimación de la vasija en el CIRS con láser (arriba) y estéreo (abajo). .	79
4.16. Posición del objeto obtenida con marcadores (líneas sólidas) y posición obtenida utilizando la segmentación RANSAC (líneas discontinuas) durante la aproximación al objeto.	80
4.17. Resultados de la estimación de una vasija en el mar.	81
4.18. Posición de la estimación en el mar.	81
4.19. Objetos reales capturados con la cámara Videre izquierda, donde se aprecia la variedad de dimensiones y formas de los objetos.	83
4.20. Nube de puntos correspondiente a los objetos utilizados.	84
4.21. Resultados de la estimación utilizando el método RANSAC para cilindros. Conjunto de objetos C-L.	85
4.22. Resultados de la estimación utilizando el método RANSAC para esferas. Objetos I y J.	85
4.23. Resultados de la estimación utilizando el método RANSAC para cajas. Objetos A y B.	86
4.24. Resultados de la estimación utilizando el método SQ. Conjunto de objetos completo.	87
4.25. Resultados de la estimación utilizando el método PCA. Conjunto de objetos completo.	87
4.26. Resultados de estimación de la posición del objeto A.	88
4.27. Resultados de estimación de la posición del objeto B.	89
4.28. Resultados de estimación de la posición del objeto D.	89
4.29. Resultados de estimación de la posición del objeto E.	90
4.30. Resultados de estimación de la posición del objeto I.	91
4.31. Resultados de estimación de la posición del objeto J.	91
4.32. Tiempo de procesamiento para los algoritmos de RANSAC para cilindros y esferas.	93
4.33. Tiempo de procesamiento para los algoritmos de PCA, RANSAC para cajas (arriba), y SQ (abajo).	93
4.34. UWSim mostrando la detección del cilindro mediante RANSAC con un modelo del objeto (blanco) y la posición de agarre por defecto. Nubes de puntos de la cámara estéreo real (izquierda) y simulada (derecha, componente z codificada por color).	94
4.35. Segmentación mediante RANSAC de un objeto cilíndrico con la cámara DUO M (arriba) y la reconstrucción láser (abajo).	95

4.36. Reconstrucción 3D de las escenas con múltiples objetos.	96
4.37. Resultados de las escenas con múltiples objetos reconstruidas con SQ. 97	
5.1. Componentes del módulo de Especificación.	100
5.2. Componentes de la especificación analítica del agarre.	102
5.3. Especificación analítica del agarre con estimación de SQ (izquierda, centro). Análisis del agarre con OpenRave (derecha).	103
5.4. Muestreo de la superficie de los objetos y conos de fricción en los puntos de contacto.	105
5.5. Fases para la obtención de hipótesis de agarre en OpenRave con un manipulador diestro.	106
5.6. Interfaces gráficas utilizadas en los proyectos TRIDENT y SAUVIM. 107	
5.7. Interfaces <i>hardware</i> para controlar robots submarinos.	108
5.8. La interfaz muestra información de navegación, sensores, cámaras y controles para realizar la manipulación (izquierda). Escena de especificación en UWSim (derecha).	109
5.9. Interfaz de datos auxiliar utilizando RViz, la interfaz de monitorización de ROS.	110
5.10. Interfaz para la supervisión de la manipulación. Selección del método de estimación.	111
5.11. Especificación del agarre teniendo en cuenta la geometría del objeto. En blanco se superpone el modelo del objeto en UWSim.	112
5.12. Especificación del agarre teniendo utilizando marcadores interactivos en 3D. En blanco se superpone el modelo del objeto en UWSim.	112
5.13. Pestañas de control de la especificación. Control geométrico (arriba, izquierda), control con marcadores interactivos (derecha, arriba), control de la apertura del manipulador (abajo, izquierda) y control de la especificación con clasificación (abajo, derecha).	113
5.14. Configuración de parámetros de segmentación y estimación.	114
5.15. Uso de la GUI para introducir puntos de ruta que definen la trayectoria que debe seguir el robot.	115
5.16. Uso de la GUI en los experimentos de manipulación en el laboratorio IRS Lab.	116
5.17. Resultados de la especificación en el Modo Geométrico (arriba) y en el Modo Interactivo (abajo).	116
5.18. Resultados de la generación de agarres analítica con OpenRave para el objeto A.	117
5.19. Resultados de la generación de agarres analítica con OpenRave para el objeto E.	118
5.20. Resultados de la generación de agarres analítica con OpenRave para el objeto I.	119
5.21. Uso de la GUI para la especificación en la piscina del CIRS.	120
5.22. Uso de la GUI para la especificación en el mar.	120
6.1. Componentes del módulo de Ejecución.	123
6.2. Sistemas de referencia asociados a la tarea de manipulación.	124
6.3. Arquitectura de control del ARM5E en detalle.	126

6.4. Segmentación del objeto (izquierda, verde) y extracción del fondo (izquierda, azul). Trayectoria generada a partir de la nube de puntos 3D utilizando los métodos de proyección y <i>concave hull</i> (derecha, amarillo).	128
6.5. Obtención de la trayectoria de desenterrado para los objetos K y L (azul).	128
6.6. Estado final e inicial de la intervención de desenterrado.	129
6.7. Secuencia de ejecución de la manipulación para el objeto C en el laboratorio.	130
6.8. Experimentos de manipulación en el IRS Lab. Agarre de diferentes objetos.	131
6.9. Secuencia de ejecución de la manipulación para el objeto D en el laboratorio.	132
6.10. Trayectoria seguida por el manipulador respecto a la posición de agarre deseada en el laboratorio para el objeto <i>D</i>	132
6.11. Experimentos en el CIRS. El GIRONA500 dispone del ARM5E acoplado, cámara estéreo y sensor láser 3D. El SPARUS II se utiliza en el MERBOTS como apoyo.	133
6.12. Trayectoria del manipulador respecto a la posición de agarre deseada en la piscina.	134
6.13. Manipulación en el CIRS. Estimación, Especificación, Supervisión y Ejecución del agarre.	134
6.14. Experimentos en el mar. El AUV GIRONA500 dispone del ARM5E acoplado y cámara estéreo. El AUV SPARUSII se utiliza como apoyo.	135
6.15. Manipulación en el mar. Estimación, especificación, supervisión y ejecución del agarre.	136
6.16. Trayectoria del manipulador respecto a la posición de agarre deseada en el mar.	136

Índice de tablas

2.1. Dimensiones de los objetos.	28
2.2. Parámetros D-H de la cadena cinemática del brazo ARM5E.	31
4.1. Parámetros por defecto del algoritmo basado en RANSAC	64
4.2. Resultados de estimación de la vasija con RANSAC en MERBOTS.	82
4.3. Resultados de estimación de la vasija con PCA en MERBOTS.	82
4.4. Resultados de estimación de la vasija con SQ en MERBOTS.	83
4.5. Comparativa de la estimación con RANSAC en simulación y laboratorio.	94
5.1. Coste temporal del análisis de agarres	118

Acrónimos

AR Augmented Reality

ASC Autonomous Surface Craft

AUV Autonomous Underwater Vehicle

CCD Charge-Coupled Device

CIRS Centro de Investigación en Robótica Submarina

D-H Denavit-Hartenberg

FK Forward Kinematics

GDL Grados De Libertad

GUI Graphical User Interface

H-ROV Hybrid Remotely Operated Vehicle

HRI Human-Robot Interaction

I-AUV Intervention Autonomous Underwater Vehicle

ICP Iterative Closest Point

IK Inverse Kinematics

LiDAR Light Detection And Ranging

OSG Open Scene Graph

PCA Principal Component Analysis

PCL Point Cloud Library

RANSAC RANdom SAmples Consensus

RAUVI Reconfigurable Underwater Vehicle for Intervention Missions

ROS Robot Operating System

ROV Remotely Operated Vehicle

SQ Supercuádricas

ToF Time of Flight

UdG Universitat de Girona

UIB Universitat de les Illes Balears

UJI Universitat Jaume I

UVMS Underwater Vehicle-Manipulator System

UWSim UnderWater Simulator

Nomenclatura

$x_b = (x, y, z, \alpha, \beta, \gamma)$	Posición y orientación cartesiana del efector final en el sistema de referencia b
$\dot{x}_b = (\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma})$	Velocidad cartesiana del efector final en el sistema de referencia b
x^*, \dot{x}^*	Posición y velocidad cartesianas deseadas
$q = (q_1, q_2, q_3, q_4)$	Posición articular del robot
$\dot{q} = (\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4)$	Velocidad articular del robot
q^*, \dot{q}^*	Posición y velocidad articulares deseadas
J_k	Matriz jacobiana del robot respecto al sistema de referencia k
J_{\dagger}, J^+	Pseudoinversa de la matriz jacobiana
aMb	Matriz homogénea que representa la transformación del sistema de referencia b con respecto al sistema de referencia a
w, b, c, e, o, g	Sistemas de referencia del mundo (<i>world</i>), la base del robot, la cámara, el efector final, el objeto y la posición de agarre (<i>grasp</i>)
A, B, \dots, K, L	Conjunto de objetos utilizados en los experimentos
(a, b, c)	Dimensiones del objeto
(ϵ_1, ϵ_2)	Parámetros de control de forma de las supercuádricas

Introducción

Los descubrimientos oceánicos han intrigado por igual a científicos y exploradores, ya sea con la idea de estudiar la biología marina, el cambio climático o con la esperanza de descubrir recursos naturales y secretos enterrados a grandes profundidades. Los océanos cubren el 71 % de la superficie terrestre con una profundidad media de 3900 metros. Por este motivo, dichos objetivos han permanecido fuera del alcance del ser humano y lo han condicionado por mucho tiempo [Antonelli et al., 2008]. Las primeras exploraciones de los océanos se realizaron mediante vehículos tripulados, permitiendo descubrir relaciones fundamentales con los ecosistemas, la temperatura del planeta y el clima [Bigg et al., 2003]. Sin embargo, la aparición del Vehículo Operado Remotamente o Remotely Operated Vehicle (ROV), no tripulado, permitió alcanzar profundidades mucho mayores sin poner en riesgo al ser humano. Es en estas aguas donde todavía existen grandes misterios.

Los ROV han demostrado ser herramientas de gran valor a la hora de alcanzar grandes profundidades. Esta capacidad los ha convertido en la tecnología esencial para la industria *offshore* del petróleo y el gas en busca de nuevos recursos [Urabe et al., 2015] con el objetivo de desplegar y mantener estructuras y cableado a gran profundidad. Además, son capaces de acceder a zonas de gran interés científico, donde son capaces de recuperar información que es especialmente interesante para ciencias cuyos conocimientos de los océanos son todavía incompletos, como la geología, biología, química o arqueología. Para ello utilizan cámaras, sensores de todo tipo y batimetrías, que pueden ser después estudiadas.

No obstante, el uso de ROV está limitado al despliegue de buques de apoyo y el control remoto por parte de pilotos experimentados, haciendo uso de comunicaciones a través de cable e interfaces de control muy complejas. Estos vehículos son normalmente muy grandes y pesados, de modo que el coste logístico para transportarlos y manejarlos es elevado. Las complejas interfaces de control empleadas requieren el uso de pilotos habilidosos y adiestrados. Estas características hacen aumentar el coste de este tipo de operaciones de forma exponencial. Además, la necesidad de disponer de una umbilical introduce problemas de control adicionales y una limitación en su rango de actuación. Finalmente, la fatiga y estrés que sufren los pilotos son otros inconvenientes.

Estos vehículos resultan extremadamente caros de construir y operar, de modo que con el tiempo apareció el concepto de Vehículo Autónomo Submarino o

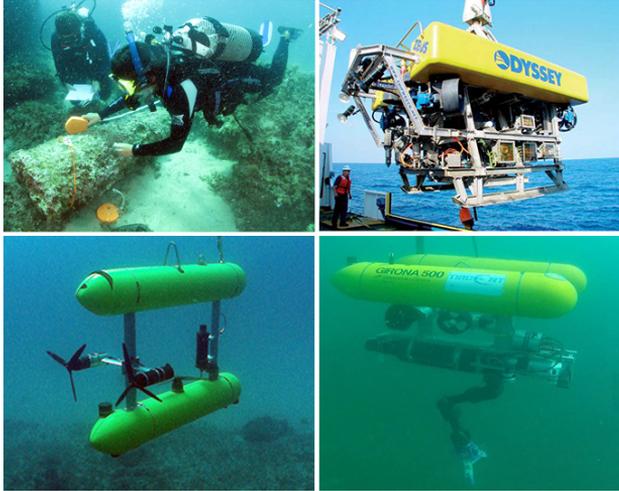


Figura 1.1: Las operaciones en el mar. De izquierda a derecha, de arriba a abajo, se llevan a cabo por buzos profesionales, ROV, AUV y I-AUV.

Autonomous Underwater Vehicle (AUV). Estos robots pretenden alcanzar mayores niveles de autonomía y reducir costes en las operaciones subacuáticas. Aunque originalmente el uso de un AUV es el de inspección y creación de mapas, como consecuencia de su utilidad aparece el Vehículo Autónomo Submarino de Intervención o Intervention Autonomous Underwater Vehicle (I-AUV), dotado de capacidad para interactuar con el medio para, por ejemplo, manipular objetos. Este escenario ofrece retos mucho mayores [Ridao et al., 2015], ya que estas tecnologías están todavía poco maduras. Es necesario mayor investigación para alcanzar niveles de autonomía que les permita ser usadas en un mayor rango de aplicaciones, como puede ser la arqueología subacuática. La Figura 1.1 muestra la evolución histórica de las misiones marinas desde las intervenciones llevadas a cabo por buceadores a poca profundidad, al uso de modernos AUV, pasando por los pesados ROV de trabajo hasta llegar a los I-AUV.

Este capítulo realiza la presentación del estado del arte sobre los temas tratados en esta tesis. En primer lugar se muestran los proyectos de investigación pioneros y los más recientes en el contexto de la manipulación submarina. En segundo lugar, se describen los proyectos desarrollados por el IRS Lab en este área. A continuación, se pone el énfasis en la utilización de los AUV y I-AUV en el ámbito de la arqueología submarina. Finalmente, se desarrollan los objetivos del trabajo y se concreta la estructura del documento.

En esta tesis se trata el uso de la visión para controlar y supervisar la manipulación que deben llevar a cabo los denominados I-AUV. Estos vehículos autónomos, a menudo cuentan con supervisión desde la superficie para simplificar y dotar de robustez e inteligencia a las intervenciones submarinas. Por ello, el componente de interacción con el usuario es también relevante en este trabajo.

1.1. Proyectos de investigación en el contexto de las intervenciones submarinas

Durante los últimos 20 años, los AUV se han convertido en la herramienta estándar para obtener mapas del fondo marino con sensores ópticos [Eustice et al., 2006] o acústicos [Paduan et al., 2009]. Las aplicaciones han sido múltiples desde inspección de presas [Ridao et al., 2010], geología marina [Escartín et al., 2008] a arqueología marina [Bingham et al., 2010]. Aunque se trata de una disciplina con un avance mayor que la de los vehículos con capacidad de manipulación, existen pocos vehículos autónomos comerciales. Sin embargo, se ha demostrado su uso para la realización de fotomosaicos en 2D [Ferrer et al., 2007] y mapas en 3D haciendo uso de *structure from motion* [Pizarro et al., 2009], cámaras estéreo [Johnson-Roberson et al., 2010] y escáneres láser [Massot-Campos et al., 2016], incluso utilizando múltiples vehículos [Campos et al., 2016].

Pese al avance de los AUV, existen aplicaciones que necesitan una mayor capacidad de interacción, como el mantenimiento de observatorios permanentes, tuberías sumergidas o redes de sensores cableadas. Como se ha comentado, estas tareas típicamente han requerido el uso de ROV, cuyo uso resulta extremadamente caro. Así, a mediados de los 90 surgieron los primeros proyectos relacionados con AUV como el AUV OTTER [Wang et al., 1995], el ODIN [Choi et al., 1994] o el proyecto UNION [Rigaud et al., 1998]. No obstante, los primeros vehículos autónomos capaces de realizar intervenciones en el mar aparecieron más tarde.

El proyecto AMADEUS [Lane et al., 1997], (Figura 1.2), que tuvo lugar entre 1993 y 1999 fue el primer intento de desarrollar un manipulador diestro para aplicaciones subacuáticas. Disponía de un efector final con 3 dedos hidráulicos diseñado para imitar los movimientos de una trompa de elefante con el objetivo de manipular objetos. Posteriormente, el proyecto añadió dos brazos robóticos eléctricos de 7 Grados De Libertad (GDL) trabajando de forma coordinada. Cada brazo pesaba 65 kg., medía 140 cm. y estaba sellado con aceite en su interior, permitiendo alcanzar 500 metros de profundidad. Se trata del primer manipulador eléctrico de 7 GDL capaz de obtener unos resultados similares a los de un brazo industrial. Se demostró el movimiento coordinado de los dos manipuladores para mover un objeto rígido en un tanque de agua.

El citado proyecto UNION [Rigaud et al., 1998], entre 1996 y 1999, se centró en desarrollar métodos para aumentar la autonomía e inteligencia de los ROV. Para ello, se desarrollaron sistemas de control coordinado y estrategias de percepción para el vehículo (el ROV VORTEX) y el manipulador (el Mitsubishi PA10). En este proyecto se estudió el control dinámico conjunto de la cadena vehículo-manipulador. Aunque únicamente fue validado en un entorno simulado, representa la primera ocasión en la que se ensambló un manipulador a un vehículo con el objetivo de realizar manipulación autónoma.

En [Evans et al., 2001] se presenta el proyecto SWIMMER (1999-2001), que plantea una configuración híbrida entre ROV y AUV, donde un AUV transporta al ROV cerca de una instalación en aguas profundas para que este pueda entonces actuar. En primer lugar, el AUV es capaz de acoplarse a una estación de forma autónoma. De esta forma, el ROV se conecta mediante umbilical a la estación, conectada a la superficie, por lo que puede ser teleoperado de forma tradicional.

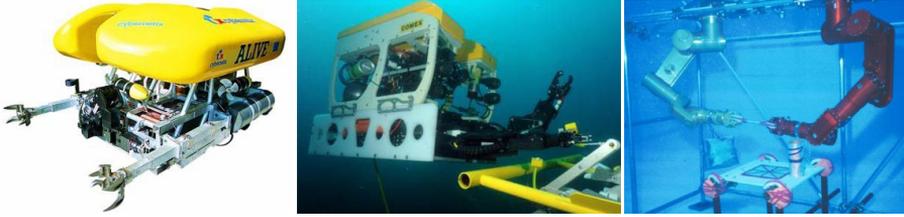


Figura 1.2: De izquierda a derecha, el concepto de los proyectos SAUVIM, DEXROV y AMADEUS.

El sistema demostró ser eficiente para inspección, mantenimiento y reparación de estaciones petrolíferas a gran profundidad, eliminando la necesidad de disponer de una umbilical de largo alcance.

En esta línea de intervención autónoma, el proyecto ALIVE ([Evans et al., 2003]) tenía el objetivo de desarrollar un vehículo diseñado para acoplarse en una estructura submarina desconocida de forma autónoma, similar a las utilizadas en la industria petrolífera. Una vez acoplado mediante dos brazos telescópicos, un brazo robótico de 7 GDL (Figura 1.2) podía operar el panel para girar válvulas o introducir conectores. Una vez acoplado, la tarea de manipulación se convierte en una tarea más convencional. Después de 4 años, en 2004, demostró la posibilidad de navegar, acoplarse y operar el panel abriendo una válvula como las que se pueden encontrar en la industria petrolífera. De acuerdo a estos trabajos se puede ver cómo ésta industria ha sido el motor de la innovación en el ámbito de la robótica submarina, aunque en la actualidad el desarrollo de estos robots se extiende a otras áreas.

La primera misión de manipulación de un objeto desde un I-AUV se realizó en el proyecto SAUVIM, descrito en [Yuh et al., 1998], que tuvo lugar entre 1999 y 2009 en la Universidad de Hawaii. El objetivo del vehículo desarrollado era el de recuperar misiles del fondo marino. Para ello, se concibió como un vehículo semi-autónomo con un brazo manipulador controlado por un supervisor en una base en superficie. El supervisor, controlaba tanto el brazo como el vehículo, pero el robot podía realizar la misión de forma autónoma con el uso de marcadores artificiales conociendo la posición aproximada del objeto. A diferencia de proyectos previos, el peso del robot (unas 6 toneladas) con respecto al brazo (65Kg. del brazo ANSALDO utilizado previamente en el proyecto AMADEUS) permite el control independiente de los dos sistemas, considerando el movimiento del brazo despreciable con respecto a la dinámica del vehículo. SAUVIM resultó el primer proyecto capaz de demostrar la manipulación autónoma de un objeto conocido a priori.

Posteriormente aparece el concepto de Vehículo Operado Remotamente Híbrido o Hybrid Remotely Operated Vehicle (H-ROV) [Fletcher et al., 2008] [Bowen et al., 2013] [Brignone et al., 2015]. Este tipo de vehículo es un AUV que puede utilizarse como ROV a través de una umbilical de fibra óptica. Trata de combinar los beneficios de ambas aproximaciones intentando reducir los costes asociados a los ROV clásicos. El ejemplo de uso más avanzado consiste en una intervención teleoperada de forma inalámbrica utilizando comunicaciones a través de luz LED desde una baliza conectada por umbilical al H-ROV [Farr et al., 2010]. Estos sistemas tratan de aumentar la autonomía de los robots manteniendo al mismo tiempo al usuario

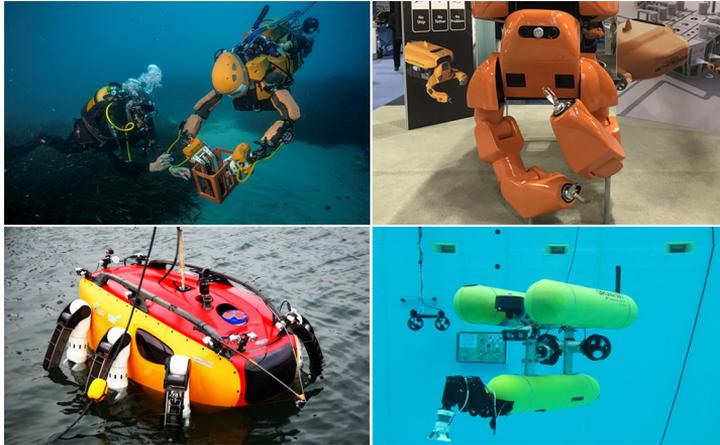


Figura 1.3: Aproximaciones más recientes a la manipulación submarina. De izquierda a derecha y de arriba a abajo: el robot OCEAN ONE, el ROV AQUANAUT, el robot CRABSTER y el proyecto PANDORA.

en el bucle de control.

Recientemente, el robot OCEAN ONE [Khatib et al., 2016] ha sido desarrollado trasladando conceptos de la robótica humanoide al escenario marino. Este proyecto propone el concepto de avatar que se puede ver en la Figura 1.3, el cual ofrece un alto nivel de autonomía para la interacción física al mismo tiempo que el experto puede tomar el control a través de una Interfaz Humano-Robot o Human-Robot Interaction (HRI) inmersiva e intuitiva, gracias a la similitud de diseño entre el robot y el ser humano. El robot dispone de capacidades de interacción autónoma y el usuario es capaz de instruir al robot mediante instrucciones de alto nivel cognitivo, al mismo tiempo que es capaz de tomar un control más directo. Una conexión mediante umbilical haría más complejo el sistema por lo que dispone de un ROV que se comunica a corta distancia con el robot y provee de una vista exterior del sistema, lo cual es muy útil también para el operador. Así, este proyecto resulta importante en el aspecto de robótica cooperativa.

Este proyecto se originó en la necesidad de disponer de un ROV nuevo para intervención que permitiera estudiar la barrera de coral del Mar Rojo en Egipto, ya que no existía un robot con las capacidades y delicadeza necesaria para interactuar con el coral como lo haría un buceador. Sin embargo, otro caso de uso del robot es, como en los proyectos desarrollados por el IRS Lab, el de la arqueología submarina y la recuperación de frágiles artefactos. El robot cuenta con dos brazos diestros equipados con sensores de fuerza, que llegan al piloto en forma de retroalimentación háptica. Así, el humano puede sentir si el robot está cogiendo el objeto con mayor o menor fuerza y actuar en consecuencia. Utilizando un concepto similar, el AUQUANAUT de Houston Mechatronics [Houston Mechatronics, 2018] es el primer robot comercial capaz de intercambiar entre modo AUV y modo ROV de acuerdo a las necesidades de la misión, al mismo tiempo que permite al usuario controlar el robot desde la distancia sin necesidad de un buque de despliegue. No obstante, este robot todavía no ha sido validado públicamente en una intervención

real.

Por otro lado, el CRABSTER CR200 [Yoo et al., 2015] es una alternativa a los AUV y ROV tradicionales capaz de soportar fuertes corrientes gracias a un concepto novedoso inspirado en los cangrejos. Este robot dispone de seis patas que le permiten desplazarse por el fondo marino como un crustáceo y, al mismo tiempo, generar menos partículas en suspensión que con el uso de propulsores. Respecto a la manipulación, este robot está equipado con dos manipuladores capaces de coger objetos, aunque el control es similar al de un ROV. Este robot fue diseñado para trabajar junto a un arqueólogo con el objetivo de examinar barcos hundidos en el siglo 12 en el Mar Amarillo. Sin embargo, en abril de 2014 fue utilizado tras el hundimiento de un ferry en Corea, donde el equipo científico desarrollador del CRABSTER elaboró un exhaustivo mapa del hundimiento durante 15 horas.

En la actualidad el concepto de manipulación cooperativa se está explorando de forma muy activa. La propuesta europea CRAUNIM, coordinada por el IRS Lab, propuso la cooperación entre dos I-AUV para manipular una tubería y realizar un ensamblado. De esta propuesta surgió el proyecto MARIS [Simetti et al., 2018], que se centraba en el desarrollo de robots capaces de interactuar con el medio submarino mediante la manipulación diestra. En él se demostró la recuperación de un objeto y se simuló el control cinemático para el transporte de una carga de forma cooperativa con dos vehículos en flotación libre (*free-floating manipulation*) [Simetti y Casalino, 2017].

El recién finalizado proyecto DEXROV [Gancet et al., 2015] propone aumentar la autonomía de los ROV para reducir los recursos logísticos necesarios para su despliegue. Así, el proyecto propone el uso de un H-ROV lanzado desde un buque con una tripulación mínima, permitiendo controlar el vehículo desde la costa a través de un enlace de satélite para reducir los costes de despliegue y control. En este caso, dispone de dos brazos diestros. Este tipo de metodología permite realizar intervenciones a gran profundidad con mayor facilidad. Pese a no ser un proyecto centrado en la manipulación autónoma, muestra la dirección del estado del arte en el que se buscan soluciones híbridas innovadoras que sacan provecho de lo mejor de ambos mundos. Por otro lado, el proyecto ROBUST [Simetti et al., 2017] propone el uso de un I-AUV en el contexto de la minería subacuática, aunque su principal objetivo es el mapeado mediante tecnologías muy novedosas.

La cooperación entre vehículos no solo se lleva a cabo estrictamente bajo el agua. En [Kaiser, 2013], se utiliza un deslizador o *wave glider* como vehículo de superficie para seguir al vehículo submarino y servir como plataforma de comunicaciones. Al mismo tiempo, un H-ROV trabaja en modo AUV para realizar un mapa de la zona para encontrar fugas en tuberías. Cuando encuentra una anomalía, realiza un barrido más cercano a la tubería y cambia a modo ROV para realizar una intervención previa al posible uso de un ROV. Este proyecto ha demostrado el uso de un enlace acústico/óptico de alta capacidad para la teleoperación del HROV NEREUS [Farr et al., 2010]. Por otro lado en [Chavez et al., 2017] se propone otro tipo de cooperación: entre un AUV y un buceador.

En la siguiente sección se describen proyectos de investigación recientes en los que ha participado el grupo IRS Lab, cuyas contribuciones van en la línea trazada por los proyectos pioneros descritos en esta sección. En paralelo a estos proyectos tuvo lugar el proyecto PANDORA. El objetivo principal fue proveer de autonomía persistente a los robots submarinos. Es decir, reducir la asistencia que necesitan los

robots [Lane et al., 2015], reconociendo cuándo se producen fallos y respondiendo a ellos a varios niveles de abstracción y con limitaciones temporales. Este planteamiento supone una novedad importante en un escenario tan imprevisible como es el subacuático. Los escenarios de validación son la inspección, la limpieza de una cadena y la interacción con una serie de válvulas. Se demostró el uso del robot para girar válvulas de forma autónoma en flotación utilizando un algoritmo de aprendizaje por demostración [Carrera et al., 2015]. Además, se demostró la realización de la misma tarea con un control cinemático basado en tareas [Cieslak et al., 2015] y el uso de técnicas de planificación en el estado del arte [Youakim et al., 2017].

1.2. Proyectos de investigación del IRS Lab

El trabajo de esta tesis se ha llevado a cabo como parte de la investigación del IRS Lab de la Universitat Jaume I (UJI) de Castellón. El grupo lleva realizando investigación en robótica submarina desde el año 2009, en el que inició el proyecto Reconfigurable Underwater Vehicle for Intervention Missions (RAUVI) (ver Figura 1.4), aunque había trabajado en manipulación robótica previamente como parte del Robotic Intelligence Laboratory (RobInLab) de la misma universidad. La experiencia del grupo se centra en manipulación [Sanz et al., 2005] y visión subacuática submarina [Pérez et al., 2015], así como robótica educacional [Pérez et al., 2017] e interfaces de usuario [García et al., 2017].

Principalmente el grupo ha realizado esfuerzos en la línea de manipulación robótica bajo el agua [Prats et al., 2012d] [Peñalver et al., 2015] [Sanz et al., 2012], aunque el grupo también se ha orientado a otros campos relacionados con el desarrollo y el control de robots submarinos autónomos. Otro campo de investigación, parte importante de este trabajo, es el caso de la HRI [García et al., 2017] [García et al., 2010b], en el que se desarrollan nuevas formas de controlar y supervisar robots con diferentes niveles de autonomía. También se han realizado esfuerzos en el campo de la visión artificial, como es el caso de esta tesis y el de la mejora de imagen submarina mediante *deep learning* [Perez et al., 2017]. Recientemente se ha abierto una línea de investigación en comunicaciones inalámbricas submarinas [Centelles et al., 2015]. En el grupo se desarrolló y se mantiene el simulador submarino UWSim [Prats et al., 2012c]. Este simulador se utiliza como herramienta esencial en el desarrollo de esta tesis y se detalla su uso a partir del Capítulo 2. Esta herramienta resulta esencial para integrar y verificar los algoritmos de percepción y control antes de utilizarlos en los robots reales.

El trabajo desarrollado en esta tesis se engloba en el contexto de los diferentes proyectos que ha realizado el grupo IRS Lab, además de coincidir en el tiempo y compartir algoritmos con varios de ellos. A continuación se detallan los proyectos realizados, haciendo hincapié en su aproximación a la manipulación subacuática. Todos estos proyectos son desarrollados en colaboración con la Universitat de Girona (UdG) y la Universitat de les Illes Balears (UIB), pues se han financiado por el Ministerio de Economía y Competitividad en su programa DPI y por la Comisión Europea bajo el programa FP7 como proyectos coordinados.

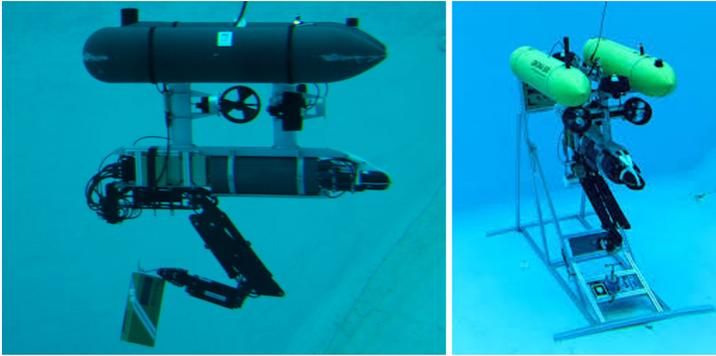


Figura 1.4: Escenarios de intervención de RAUVI (recuperación de una caja, izquierda) y TRITON (operación de un panel, derecha).

1.2.1. RAUVI

La experiencia en robótica subacuática del laboratorio IRS Lab empezó con el proyecto nacional RAUVI en 2009 (ver Figura 1.4). El objetivo principal de este proyecto fue el desarrollo del un AUV capaz de realizar misiones de intervención de forma autónoma en entornos submarinos. El caso de estudio propuesto fue la recuperación de un objeto del fondo marino como se describe en [Prats et al., 2012d]. Para cumplir con la misión se establecieron dos fases: exploración e intervención. Durante la fase de exploración se crea un mapa de la zona en busca de objetos de interés con información acústica y visual. Después, el vehículo emerge a la superficie, donde se recopila la información para preparar la intervención. Utilizando métodos HRI, basados en el uso de una Interfaz Gráfica de Usuario o Graphical User Interface (GUI), el usuario puede especificar la misión, definiendo el objeto que el vehículo debe recuperar. El proyecto demostró la búsqueda y recuperación de un modelo de caja negra de un avión utilizando un AUV y un brazo robótico con un gancho en piscina y en el mar. En este proyecto se desarrolló el AUV GIRONA500 [Ribas et al., 2012], que ha sido utilizado posteriormente en multitud de proyectos de investigación. El sistema fue capaz de forma autónoma de explorar la zona en busca de la caja y de recuperarla en la fase de intervención. Tras ALIVE y SAUVIM, RAUVI es el tercer proyecto capaz de demostrar capacidades de intervención autónomas y el primero en hacerlo con un vehículo ligero de menos de 200 kg.

Este proyecto supuso un punto de inicio de una línea de investigación muy importante en manipulación submarina. En este proyecto nacional se desarrolló el brazo eléctrico de 4 GDL que se utiliza en proyectos subsiguientes y en esta tesis [Fernández et al., 2013]. Durante la primera fase de la misión el robot obtiene un fotomosaico del área. El proceso desde el punto de vista de la manipulación empieza cuando el usuario selecciona de forma manual en 2D el objetivo y posición de agarre sobre el mapa generado. Finalmente, se realiza la recuperación autónoma del objeto. La manipulación del objeto se lleva a cabo utilizando un gancho que pasa por el asa de la caja, siendo el caso más sencillo de los que se presentan.

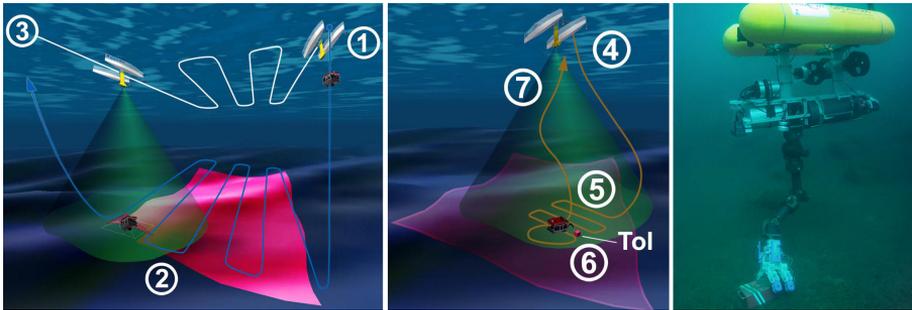


Figura 1.5: El concepto del proyecto TRIDENT se divide en exploración (izquierda) e intervención (centro). El resultado de la recuperación de una caja negra se muestra a la derecha.

1.2.2. TRIDENT

Posteriormente, el proyecto FP7 TRIDENT, financiado por la Unión Europea (FP7-ICT-2009-248497), tuvo el objetivo de desarrollar robots submarinos para permitir la manipulación diestra submarina multipropósito y autónoma [Sanz et al., 2012]. El proyecto, evolución de RAUVI, propone una nueva metodología para las intervenciones subacuáticas combinando tareas autónomas con supervisión, tal y como se propone en esta tesis para resolver una misión de búsqueda y recuperación.

El proyecto propuso un equipo heterogéneo de dos robots con capacidades complementarias formado por un Vehículo Autónomo de Superficie o Autonomous Surface Craft (ASC) y un I-AUV equipado con un brazo diestro de 7 GDL y una mano con 3 dedos. La misión que llevan a cabo se divide en la fase de exploración y la fase de intervención como se muestra en la Figura 1.5. Durante la exploración, los robots obtienen información para poder crear un mapa del entorno cuando vuelven a la superficie. Con la ayuda del mapa, el usuario selecciona el objeto deseado y la tarea a realizar. Entonces puede empezar la segunda fase, en la que se busca el objeto con el mapa obtenido y el I-AUV inicia la intervención multisensorial a través de una estrategia de control *free floating* en la que se utilizan de forma óptima todos los GDL del conjunto vehículo-manipulador.

En los experimentos finales en una piscina y en el mar (en el interior de un puerto) se demostró de forma desacoplada la capacidad de trabajar cooperativamente de los dos vehículos y la capacidad de intervención del robot ¹. Es un proyecto diferencial en el estado del arte ya que es el primero en ofrecer resultados de campo en el ámbito de la manipulación subacuática autónoma y multipropósito.

Para realizar la manipulación, el objeto fue localizado y seguido de forma autónoma utilizando una plantilla del objeto 2D obtenida de forma autónoma y el fotomosaico obtenido durante la fase de exploración previa. Esta plantilla fue detectada en 2D utilizando características o *features*. Después, el agarre del objeto se realizó con un brazo robótico de 7 GDL con una mano de 3 dedos [Prats et al., 2012b] utilizando un control basado en prioridad de tareas [Cieslak et al., 2015].

¹Vídeo de los experimentos finales de TRIDENT donde se muestra la manipulación autónoma diestra de un modelo de una caja negra de un avión <https://youtu.be/2qf7ukrUcCc>

1.2.3. TRITON

El proyecto nacional TRITON [Peñalver et al., 2015][TRITON, 2014] (Multisensory Based Underwater Intervention through Cooperative Marine Robots, DPI2011-27977-C03) tiene como objetivo la manipulación de un panel submarino, como el que se puede encontrar en la industria petrolífera o en un observatorio marino. El escenario propuesto es la intervención de un panel sumergido en el que se debe girar una válvula e interactuar con un conector (ver Figura 1.4), operaciones comunes en el mantenimiento de observatorios marinos sumergidos. Para ello el AUV GIRONA500 se tenía que acoplar al panel de forma autónoma para posteriormente realizar la manipulación, que podía requerir el desacople del vehículo. Este proyecto fue demostrado en piscina y en el mar ².

En cuanto a la metodología para resolver la manipulación, el problema se simplifica acoplando el I-AUV al panel utilizando retroalimentación visual. Gracias a esta técnica se reduce a mínimas perturbaciones la incidencia de las corrientes. El uso de un panel conocido permite saber la posición de los elementos a manipular de forma precisa mediante el uso de marcadores. Por otro lado, se utilizó el manipulador ARM5E que se utiliza en este trabajo. Debido a sus limitaciones cinemáticas, se utilizaron elementos *compliant*, que se adaptan a la manipulación. En concreto, el conector tuvo que ser modificado de forma que la inserción de un conector no fuese necesariamente en línea recta.

1.2.4. MERBOTS

Más recientemente, el proyecto MERBOTS (Multifunctional cooperative marine robots for intervention domains, DPI2014-57746-C3) [Sanz et al., 2017a], financiado por el Ministerio de Economía y Competitividad, tiene como objetivo la cooperación de tres vehículos heterogéneos para conseguir diferentes objetivos relacionados con la arqueología submarina de manera más segura, simple y económica. Para lograrlo se propone dos misiones bien diferenciadas. La primera consiste en la exploración o *survey* de forma cooperativa utilizando un AUV asistido por un vehículo de superficie que provee localización absoluta y comunicación. La segunda misión es la intervención cooperativa, que consiste en una intervención semi-autónoma realizada con un H-ROV asistido por el AUV SPARUS II [Carreras et al., 2013] que provee una visión externa durante la misión [García-Fidalgo et al., 2018]. El caso de estudio para este proyecto es la localización, el desenterrado y la manipulación de una vasija en el fondo marino. El proyecto, de tres años de duración, ha demostrado la misión de exploración y la manipulación de la vasija, tanto en una piscina como en el mar ³. El subproyecto MERMANIP [Sanz et al., 2015], responsabilidad del IRS Lab, agrupa los trabajos relativos a la manipulación de objetos.

En MERBOTS aparece el concepto de H-ROV como un robot para ayudar a los arqueólogos a recuperar objetos del fondo marino ofreciendo diferentes niveles de autonomía (Figura 1.6). El objetivo en este aspecto es desarrollar un sistema capaz de asistir a los arqueólogos en el trabajo de monitorización, caracterización, estudio, reconstrucción y preservación de los sitios arqueológicos a través de la supervisión de un experto humano. Una de las principales tareas es realizar operaciones de

²Vídeo de los experimentos finales del proyecto TRITON <https://youtu.be/xA2SGLi5TYg>

³Vídeo de los experimentos finales del proyecto MERBOTS <https://youtu.be/1xECxNb0-dQ>

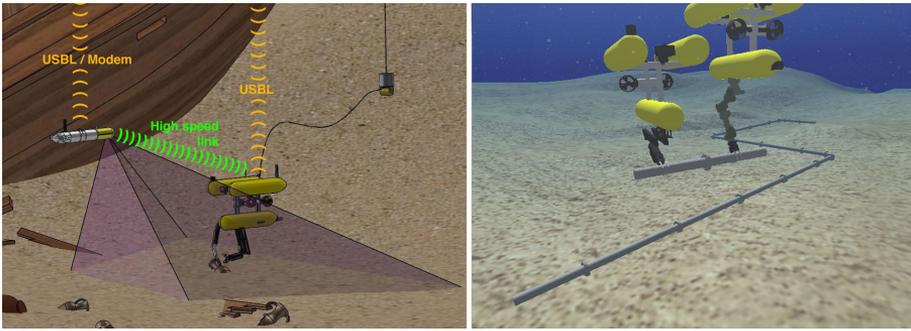


Figura 1.6: El concepto del proyecto MERBOTS (izquierda) y el proyecto TWINBOT (derecha).

dragado [Fernández et al., 2015] y manipulación de objetos. Antes del desenterrado, el H-ROV realiza una exploración cooperativa junto a otro vehículo para realizar una reconstrucción en 3D del entorno, de forma similar a la realizada en TRIDENT, en el que se obtenía un fotomosaico de la zona. Así, utilizando esta información se planifica la trayectoria de dragado. Posteriormente, se realiza la manipulación del objeto.

Parte de las técnicas propuestas en este proyecto relativas al segundo escenario forman parte de esta tesis y se explican con más detalle en los siguientes capítulos. Para realizar la manipulación, en primer lugar se realiza una reconstrucción en 3D del entorno del objeto utilizando una cámara estéreo o un novedoso método de reconstrucción láser [Palomer et al., 2018]. A continuación, el sistema es capaz de reconocer una vasija con forma cilíndrica por medio de diferentes técnicas: utilizando marcadores, utilizando una plantilla en 3D del objeto [Company-Corcoles et al., 2018] o mediante un método de segmentación RANdom SAMple Consensus (RANSAC) [Fornas et al., 2017], que es parte de este trabajo y se explica en más detalle en el Capítulo 4. Después, haciendo uso de la interfaz que se describe en el Capítulo 5 se realiza la especificación de la manipulación. Finalmente, se utiliza un algoritmo de control basado en prioridad de tareas [Peñalver et al., 2018] para guiar el vehículo a la posición de manipulación deseada.

1.2.5. TWINBOT

Como continuación del trabajo realizado, el laboratorio IRS Lab ha empezado recientemente el proyecto TWINBOT (2018), financiado por el Ministerio de Ciencia, Innovación y Universidades, en el que se propone un escenario de manipulación cooperativa utilizando dos I-AUV en un entorno con estructuras artificiales. Ambos vehículos deberán realizar dos misiones principales. La primera consiste en la manipulación y transporte de una tubería (ver Figura 1.6) de forma conjunta. La segunda es una tarea de ensamblado en la que un vehículo trata de acoplar una pieza a otra que sostiene el otro vehículo. Esta configuración propone múltiples desafíos como son la coordinación entre robots, la comunicación a corta distancia, el control de los manipuladores o la seguridad, entre otros.

En lo que respecta a la manipulación, resulta un escenario complejo ya que

se debe considerar cómo realizar la especificación del agarre en ambos vehículos, planificación del mismo teniendo en cuenta las colisiones entre vehículos y también el control sincronizado de los robots. La inclusión de dos vehículos genera en la necesidad de desarrollar estrategias de control distribuidas o de maestro-esclavo. Además, se propone también un control mediante fuerza para la tarea. Por otro lado, el escenario en visión resulta conocido, ya que una tubería no difiere mucho de un objeto como una vasija. Sin embargo, compartir esta información de forma precisa con otro robot supone un gran reto. En la línea de trabajo de esta tesis, disponer de una representación geométrica de la escena como la que se propone significaría reducir el ancho de banda necesario para compartir esta información entre robots.

1.3. Contexto de la arqueología submarina

En secciones previas se han introducido los diferentes avances de la robótica submarina desde la perspectiva de la capacidad de intervención autónoma. Los últimos años de arqueología subacuática en aguas profundas han demostrado que los arqueólogos también se pueden beneficiar en gran medida de las nuevas tecnologías en robótica [Allotta et al., 2015]. No obstante, los requisitos de este tipo de misiones proponen nuevos retos para los ingenieros que desarrollan estos sistemas [Bingham et al., 2010]. Los sumergibles tripulados tienen la ventaja de permitir a los científicos llegar físicamente a las profundidades y realizar una observación sistemática de los sitios de interés, así como recuperar artefactos de forma presencial [Sakellariou et al., 2007]. Sin embargo, debido a la complejidad y sus altos costes, su disponibilidad es muy limitada. Al igual que en otros ámbitos, el uso del ROV supuso un gran salto en la arqueología en sitios a gran profundidad. Con un suministro continuo desde un buque de apoyo, pueden disponer de gran variedad de sensores, pueden trabajar de forma ininterrumpida y pueden recuperar artefactos con manipuladores pesados pero muy precisos [Roman y Mather, 2010].

Los ROV generalmente obtienen la alimentación y se controlan vía umbilical desde la superficie. Esta solución permite disponer de autonomía ilimitada en términos de consumo y tiempo, pero limita su alcance y aumenta los costes de despliegue y utilización en función de la profundidad (necesitando buques de apoyo mayores). Aunque limitados en el tiempo de uso, los AUV evitan algunos de estos problemas con la supresión del cable umbilical y resultan con ello una herramienta más adecuada para investigaciones arqueológicas, reduciendo sustancialmente los costes. Como se ha indicado, ha habido un incremento en el uso del AUV durante la última década, tanto con aplicaciones comerciales como científicas. Los primeros usos de AUV en arqueología marina se pueden ver en [Mindell y Bingham, 2001], que muestra las aplicaciones de nuevas tecnologías al estudio de sitios arqueológicos, haciendo énfasis en las ventajas del uso de AUV. Más concretamente, [Foley et al., 2009] presenta una descripción detallada del estudio arqueológico de dos barcos de 2000 años de antigüedad. Debido a la profundidad de los pecios, que alcanza los 70 metros, no era adecuada la presencia de seres humanos. De este modo, utilizaron como alternativa un AUV para realizar mapas y estudiar los yacimientos. Estudios extensivos del uso del AUV y técnicas de arqueología a grandes profundidades se pueden encontrar en [Bingham et al., 2010, Roman y Mather, 2010]. En la Figu-

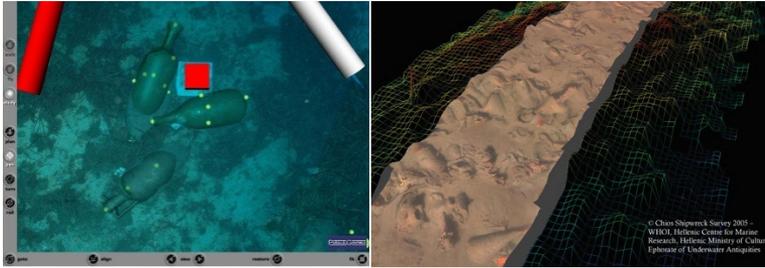


Figura 1.7: Aplicaciones para la arqueología submarina en el proyecto Venus (izquierda) y un pecio en la isla de Chios (derecha).

ra 1.7 se muestra la reconstrucción de un yacimiento arqueológico submarino en la isla de Chios, donde se combinan técnicas de batimetría en 3D y fotografía. Precisamente, la cartografía de yacimientos arqueológicos es una de las necesidades que pueden cubrir actualmente los AUV. El proyecto Venus [Drap et al., 2008] demuestra una aplicación de los AUV a la realización de fotomosaicos para obtener mediciones precisas y modelos 3D de objetos arqueológicos (ver Figura 1.7). Este paso resulta esencial antes de una intervención submarina, siendo uno de los objetivos principales de esta tesis.

La manipulación subacuática autónoma en escenarios arqueológicos es todavía más desafiante que las misiones de exploración. La razón más importante es la interacción con el entorno, ya que es susceptible de dañar tanto los robots como los objetos que se desean manipular. Existen avances significativos en este área como demuestra, por ejemplo, la búsqueda y recuperación de objetos o la intervención de paneles marinos descrita en el proyecto TRITON [Peñalver et al., 2015] que hacen pensar en nuevos usos de esta tecnología en aplicaciones como la arqueología. Aún así, debido a la fragilidad y el valor incalculable de los objetos que se manejan en estos escenarios hacen pensar que el concepto H-ROV mencionado anteriormente sea el más indicado ya que permite supervisar o controlar la manipulación y aprovechar sus funciones autónomas. Los H-ROV son esencialmente AUV, que pueden ser reconfigurados como ROV cuando utilizan una umbilical por fibra óptica o comunicaciones inalámbricas [Farr et al., 2010]. La demostración más avanzada de este concepto es el citado OCEAN ONE [Khatib et al., 2016]. Este robot fue probado satisfactoriamente en una misión de arqueología submarina en el lugar donde se encuentra el pecio de la *Lune* en Francia, manipulando una vasija a 91 metros de profundidad. No obstante, para desarrollar esta misión fueron necesarios gran cantidad de recursos, incluyendo un buque con equipamiento en la frontera del estado del arte para manipulación subacuática y despliegue del OCEAN ONE así como tres ROV de apoyo que le acompañaron.

Como se puede ver, en el estado del arte existen diferentes aproximaciones al problema que nos ocupa. En este trabajo se ha optado por la línea marcada por el proyecto MERBOTS, en la cuál se proveen soluciones semi-autónomas a los problemas de manipulación submarina. De esta forma, se permite la manipulación de diversos tipos de objetos que se pueden encontrar principalmente en el entorno de la arqueología submarina. El objetivo es, en primer lugar, proveer de una serie de

funciones que doten al sistema de operaciones autónomas de alto nivel. En segundo lugar, se ofrece al usuario la capacidad de monitorizar y supervisar el sistema mientras ejecuta estas operaciones autónomas o pasa a tomar control del vehículo. De esta forma, el usuario no interviene necesariamente en el control a bajo nivel del robot. Asimismo, es capaz de mejorar la robustez del sistema tomando decisiones sobre las operaciones que va realizando el robot mientras recibe solo la información relevante para la misión.

1.4. Objetivos

El objetivo principal de esta tesis es estudiar los problemas asociados a la visión artificial y la manipulación en el entorno subacuático. Para ello se ha desarrollado un sistema que mejora el estado del arte actual en cuanto a la manipulación con un I-AUV, enfocado a la utilización en sistemas reales en el contexto de las necesidades arqueológicas. Se ha desarrollado teniendo en cuenta los equipos disponibles, siendo posible su uso en cualquier vehículo para intervención. Esta metodología permite percibir el entorno utilizando técnicas de visión clásicas e introducir al usuario en el sistema de control del vehículo, ofreciendo diferentes niveles de autonomía. En el estado del arte existen multitud de alternativas para el guiado de la manipulación. Así, en esta tesis se ha optado por métodos contrastados para ofrecer soluciones tanto autónomas como supervisadas.

En el apartado experimental, se demuestra el trabajo realizado en diferentes entornos, desde un escenario de simulación que permite evitar algunas dificultades, hasta un entorno real en el mar. Estos pasos sucesivos permiten conocer las limitaciones del sistema y afrontar las dificultades del medio de manera progresiva, al mismo tiempo que se acaba demostrando el uso del sistema en un escenario creíble.

El objetivo principal se desglosa en los siguientes puntos o subobjetivos:

- **Establecer una metodología de trabajo incremental en la que el sistema se pruebe en simulación y en diferentes escenarios reales con dificultad creciente. Además, los algoritmos han de ser desarrollados para facilitar su uso en los diferentes escenarios.**
- **Desarrollar y evaluar el sistema de percepción adecuado para realizar la intervención basado principalmente en visión.**
- **Desarrollar algoritmos de procesamiento visual para el reconocimiento del entorno y la estimación de posición, orientación y geometría de los objetos. En este sentido, se proponen estrategias de detección clásicas para objetos cuyo modelo 3D no es conocido a priori, como son la segmentación mediante RANSAC y la estimación de Supercuádricas (SQ).**
- **Desarrollar un sistema de caracterización del agarre que permita determinar el agarre de los objetos de forma autónoma por medio de heurísticas y técnicas de análisis.**
- **Desarrollar un sistema de Human-Robot Interaction (HRI) que permita supervisar la especificación del agarre y monitorizar la manipulación.**

- **Implementar y demostrar la ejecución del sistema en simulación, en un tanque de agua en el laboratorio del IRS Lab, en la piscina del Centro de Investigación en Robótica Submarina (CIRS) y en el mar en el puerto de Sant Feliu de Guíxols. Este objetivo incluye la integración de los diferentes módulos que participan en la intervención.**

1.5. Estructura

Los diferentes temas tratados en esta tesis se presentan en 7 capítulos. En el **Capítulo 2** se explica la metodología del sistema de manipulación. Después se detalla el proceso de validación experimental llevado a cabo, orientado a realizar experimentos reales en el entorno submarino tras superar diferentes etapas intermedias. Además, se detalla la mecatrónica y sensores utilizados, así como el *software* desarrollado y la arquitectura del sistema de manipulación.

En el **Capítulo 3** se introduce la problemática de la visión submarina. Seguidamente, se detalla el uso de diferentes sensores y técnicas de reconstrucción en el robot, así como los métodos de filtrado utilizados. Además, se realizan distintas comparativas donde se observa la precisión de los diferentes métodos de reconstrucción y filtrado.

Las técnicas utilizadas para estimar la posición, orientación y forma de los objetos de la escena se muestran en el **Capítulo 4**. En este capítulo se detalla el uso de RANSAC, Análisis de Componentes Principales o Principal Component Analysis (PCA) y SQ para la estimación de la geometría de objetos cuyo modelo no es conocido. Además, se presentan técnicas de cálculo de simetrías y agrupamiento utilizadas para llevar a cabo la estimación. Finalmente, se muestran resultados en los diferentes escenarios propuestos.

A continuación, en el **Capítulo 5** se detalla la especificación del agarre a través de la interfaz HRI, desarrollada en el contexto de la tesis, así como las diferentes aplicaciones y componentes de la misma. Asimismo, se detalla el proceso de especificación autónoma del agarre de forma heurística y analítica. Después, se detalla el uso de la interfaz para supervisar el agarre generado y las posibilidades de monitorización de la ejecución que ofrece.

El **Capítulo 6** describe los componentes de la arquitectura de ejecución del agarre en el robot, detallando los sistemas de control del mismo. Como aplicación del sistema descrito, se muestra su uso en una intervención de limpieza de un objeto. Finalmente, se muestran los resultados de la ejecución de los experimentos en el laboratorio y en los experimentos del proyecto MERBOTS.

Finalmente, las conclusiones, contribuciones, potenciales aplicaciones, publicaciones y líneas de trabajo futuro fruto de esta tesis se exponen en el **Capítulo 7** para resumir el trabajo realizado.

Agarre guiado por visión: metodología, validación y arquitectura del sistema

El objetivo principal de este trabajo es la manipulación de objetos en el entorno subacuático a través del agarre guiado por visión. Este capítulo se centra en la descripción de la metodología utilizada para llevarla cabo, su validación experimental a través de diferentes escenarios y la arquitectura general del sistema.

En primer lugar se presenta la metodología utilizada para conseguir la manipulación. Como introducción se expone la motivación y se describe el estado del arte actual en manipulación robótica. A continuación, se describe la aproximación seguida en esta tesis y los diferentes elementos que interactúan.

A través de los escenarios de validación del sistema se describe la configuración experimental. Estos escenarios consisten en simplificaciones y variaciones de los casos de uso considerados en los proyectos TRIDENT y MERBOTS. A través de ellos, se han desarrollado algoritmos de visión y manipulación orientados a posibles aplicaciones en el campo de la robótica submarina. Este desarrollo se ha realizado con una complejidad creciente a medida que se han considerado escenarios que presentan mayores desafíos. En relación a estos escenarios, se describe la mecatrónica, cámaras, sensores y ordenadores utilizados, cuyo elemento principal consiste en un brazo robótico de 4 GDL con una pinza como efector final.

Finalmente, se describe cómo encajan en la arquitectura software los diferentes componentes, que se detallaran en capítulos siguientes, con respecto a visión, estimación, especificación y ejecución de la manipulación. Este *framework* permite el uso de diferentes módulos a la hora de resolver los escenarios propuestos con diferentes algoritmos de una forma que no se ha utilizado previamente en la robótica submarina.

2.1. Motivación

La manipulación robótica en escenarios subacuáticos resulta un gran desafío por los recursos que son necesarios para su desarrollo y las dificultades para probar los robots de forma controlada y sistemática. Es por ello que todos los elementos que conforman el sistema deben ser bien definidos y la metodología experimental ha de tener en cuenta que la dificultad de los casos de usos reales es extrema. Por ello, se debe desgranar el problema en componentes más sencillos introduciendo desafíos de forma progresiva. Además, el uso de algoritmos clásicos cuya robustez está demostrada resulta esencial para ensamblar un sistema capaz de funcionar en un entorno desestructurado e impredecible como es el submarino. Siguiendo este enfoque, en este capítulo se define la metodología, validación y arquitectura del sistema en detalle.

2.2. Estado del arte en la manipulación basada en visión 3D

En el Capítulo 1 se han introducido los diferentes trabajos del estado de arte en el ámbito submarino y cómo estos afrontan la manipulación. En esta sección se analiza el agarre guiado por visión en la robótica con el objetivo de aplicar los métodos que utilizan a la robótica submarina. Generalmente, las técnicas utilizadas en robótica submarina son mucho menos sofisticadas debido a las grandes limitaciones sensoriales y a los entornos más adversos. Adicionalmente, los robots submarinos deben considerar las interacciones del sistema vehículo-manipulador. De esta forma, se introduce una mayor complejidad en los sistemas de control.

La manipulación robótica ha sido tratada en el grupo IRS Lab desde que la robótica empezaba a dar sus primeros pasos [Sanz et al., 1998]. En este estado del arte se hace hincapié en el proceso de obtención de la posición y orientación del efector final del robot para manipular un objeto. Este proceso se conoce como **especificación, determinación o caracterización** del agarre. En [Sanz et al., 2005] se puede ver la determinación del agarre para objetos desconocidos en 2D bajo unas condiciones de trabajo muy bien definidas, como las que se pueden encontrar en un entorno industrial.

La especificación del agarre es un problema complejo debido al gran espacio de búsqueda resultante de todas las configuraciones del manipulador, tipos de agarres y propiedades del objeto. En la actualidad, existen multitud de taxonomías a la hora de clasificar los métodos de caracterización del agarre [Bohg et al., 2014]. El paradigma más común para realizar esta tarea es el basada en modelos o *model-based*, en el cual se usan leyes físicas para modelar la forma del objeto, los contactos, y las físicas. Existen multitud de trabajos que estudian el **análisis** de agarres (determinar las propiedades físicas de un determinado agarre) y la **síntesis** de agarres (el cálculo de agarres que cumplen determinadas propiedades deseables) [Shimoga, 1996]. No obstante, estas aproximaciones en la práctica no son capaces de proponer implementaciones en escenarios complejos o con incertidumbre, como pueden ser en entornos subacuáticos, porque se basan en asunciones difíciles de satisfacer.

En aplicaciones reales las escenas no son estáticas, los modelos son imprecisos y las posiciones de los objetos varían dinámicamente cuando el vehículo sufre corrientes marinas. Los objetivos se pueden mover también debido a las corrientes, la iluminación varía continuamente, aparecen partículas flotantes en la escena que causan problemas a los algoritmos de visión, etc. Con todo, la tendencia actual consiste en incorporar información sensorial a la planificación y síntesis del agarre, principalmente proveniente de la visión. Gracias a esta información, las técnicas basadas de análisis pueden ser aplicadas con éxito.

En general, se requiere una estimación robusta de la geometría 3D de un objeto para poder planificar la manipulación de forma adecuada [Schwarz et al., 2015][Aldoma et al., 2012][Rusu et al., 2009b]. En robótica el uso de sensores de profundidad basados en luz infrarroja es un estándar *de facto*. Esto se debe a que existen sensores económicos capaces de obtener reconstrucciones densas y robustas [Zennaro et al., 2015]. Los sensores de profundidad que cuentan con una cámara con la que registrar las imágenes se denominan sensores o cámaras RGB-D. Desafortunadamente, dichos sensores no están disponibles en el entorno submarino debido a las condiciones físicas del medio, que absorbe la luz infrarroja [Massot-Campos y Oliver-Codina, 2015][Duntley, 1963]. Las cámaras estéreo resultan una solución viable, aunque fallan al reconstruir objetos o entornos con poca textura. Los sistemas de reconstrucción basados en luz láser visible necesitan escanear activamente los objetos y tienen asociados otros problemas, que se exploran en el Capítulo 3. En general, los sensores submarinos están más limitados y obtienen menos precisión y resolución que sus contrapartidas generales [Massot-Campos y Oliver-Codina, 2015].

La obtención de la geometría de los objetos de la escena se puede realizar de formas muy diferentes atendiendo a la información que se conoce de antemano sobre objeto a manipular, dependiendo de si el modelo es conocido, familiar o completamente desconocido [Bohg et al., 2014]. En el primer caso, se dispone de precisos modelos 3D de los objetos antes de realizar la manipulación. Debido a que los objetos se consideran **conocidos**, se dispone de una fase previa en la que se aprende información sobre los objetos a manipular. En este paso, se recopila información para saber cómo reconocer los objetos y cómo manipularlos. Durante la ejecución, estos métodos deben ser capaces de recuperar esta información y utilizarla para realizar la tarea. En estos casos se utilizan a menudo técnicas basadas en aprendizaje automático [Schwarz et al., 2015]. En el ámbito submarino se han aplicado con éxito métodos de aprendizaje por demostración [Carrera et al., 2015].

En vez de considerar identidad exacta, se consideran objetos **familiares** aquellos en los que el objeto es similar a otros que se han encontrado previamente. Esta similitud puede darse a varios niveles. A bajo nivel se puede encontrar un objeto similar en forma, color o textura. A alto nivel se puede tratar de un objeto de la misma categoría, por ejemplo una taza. Las aproximaciones que se engloban en este tipo asumen que los nuevos objetos se pueden manipular de forma similar a la que se ha utilizado con objetos anteriores similares. El desafío en estos casos es definir la representación de los objetos, una métrica de similitud para identificar los objetos y un modo de transferir la experiencia de los objetos anteriores a los nuevos. Finalmente, existen aproximaciones que consideran los objetos totalmente **desconocidos**. En este caso, se debe estimar la geometría del objeto con la información sensorial disponible y utilizarla para caracterizar el agarre. No obstante,

algunos métodos ni siquiera obtienen un modelo del objeto concreto [Fischinger et al., 2015].

En esta tesis se profundiza en la manipulación de objetos cuyo modelo y geometría concreta no son conocidos a priori. En otros proyectos se utilizan modelos completos que no están disponibles en el contexto de la arqueología submarina, por ejemplo. Las metodologías aplicadas en este trabajo se basan en *familiaridad* en el caso del uso de RANSAC, ya que se busca una forma primitiva particular, y en objetos desconocidos en el caso de los algoritmos de SQ [Fornas et al., 2016] y de PCA.

2.2.1. Estimación y Especificación en 3D

Como se ha comentado, suele ser habitual disponer de un modelo completo del objeto a reconocer. Si se dispone de uno, se puede detectar su posición con soluciones robustas [Aldoma et al., 2012] basadas en *features* o características. El RTM Toolbox [Prankl et al., 2015] (Recognition, Tracking and Modelling) es un sistema de adquisición de modelos en 3D a partir de vistas parciales obtenidas mediante una cámara RGB-D¹. Dichos modelos pueden ser utilizados para estimar la posición del objeto y realizar un seguimiento. Una vez el modelo ha sido obtenido se puede realizar el seguimiento de múltiples maneras. Una aproximación similar ha sido aplicada paralelamente a este trabajo en el proyecto MERBOTS [Company-Corcoles et al., 2018]. Las técnicas basadas en modelos están muy avanzadas en el estado del arte actual, pero se basan en requisitos que no siempre son fáciles de satisfacer. Por este motivo se han considerado en mayor detalle las técnicas que no disponen de esta información a priori.

Existen gran variedad de métodos para segmentar un mapa de profundidad en puntos que pueden pertenecer a los mismos objetos [Nguyen y Le, 2013]. En muchas ocasiones este paso es esencial para estimar adecuadamente la geometría de un objeto. Además, se suele asumir que los objetos a manipular son objetos hechos por el hombre, que pueden ser aproximados mediante formas primitivas básicas. Por ejemplo, la aproximación de una caja se puede obtener de forma más sencilla que otras geometrías a partir de información sensorial. En [Huebner et al., 2008] se aproxima la geometría de diferentes objetos utilizando múltiples cajas a partir de una nube de puntos en 3D. Un grupo de características buscadas comúnmente son las simetrías [Taubin y Cooper, 1991], que sirven para obtener la geometría subyacente de objetos cuya observación es incompleta. En entornos domésticos, se han utilizado las simetrías para obtener reconstrucción en 3D basadas en la distribución Poisson [Bohg et al., 2011] o para estimar de forma heurística las partes no visibles de un objeto [Schiebener et al., 2016]. El uso de supercuádricas [Biegelbauer y Vincze, 2007] también ha sido estudiado en robótica para estimar objetos tras una fase previa de segmentación y estimación inicial.

Aunque existen multitud de metodologías para determinar la posición de un objeto, se hace hincapié en las técnicas orientadas a la posterior manipulación [Bohg et al., 2014][Bohg et al., 2011]. Debido al coste computacional que requieren determinados métodos de estimación, en ocasiones es suficiente con encontrar buenos candidatos para realizar el agarre [Miller et al., 2003]. A partir de la descomposición

¹El RTM Toolbox se puede encontrar en: www.acin.tuwien.ac.at/en/vision-for-robotics/software-tools/rtm-toolbox/.

de un objeto en elementos más sencillos, es posible obtener posiciones de agarre adecuadas [Huebner y Kragic, 2008]. Además, se buscan diferentes características dependiendo del manipulador a utilizar. Existen otras técnicas prometedoras que han sido aplicadas en la robótica y se centran en el uso de mapas de alturas del entorno para especificar y aprender agarres [Fischinger et al., 2015]. En este caso no se buscan objetos específicamente sino características locales. Es decir, áreas de la escena que pueden ser cogidas de forma efectiva.

La planificación del agarre basada en el conocimiento trata de simplificar este problema con un razonamiento más simbólico. A través de la geometría de los objetos y las capacidades del manipulador permite definir tareas mucho más complejas, como son abrir un armario o coger un libro de una estantería [Prats et al., 2010]. Dichas tareas, sin embargo, requieren del uso de manipuladores diestros cuya disponibilidad en el medio submarino es muy limitada [Prats et al., 2012b]. La aproximación más avanzada a la planificación del agarre en el entorno submarino se muestra en [Youakim et al., 2017]. En [Antonelli et al., 2008] se detallan los aspectos relativos al control más importantes a tener en cuenta en las intervenciones submarinas.

Se han presentado múltiples posibilidades a la hora de realizar la estimación de objetos y la especificación del agarre. En este trabajo se propone el uso de la segmentación basada en formas primitivas, como son los métodos basados en RANSAC y basados en agrupamiento de regiones. Por ejemplo, la segmentación basada en RANSAC es particularmente adecuada para escenarios arqueológicos, pues muchos objetos que se encuentran en dicho dominio tienen una parte manipulable cilíndrica o esférica (ánforas, vasijas, tazas...), pese a presentar partes con otras geometrías. Por otro lado, se propone el uso de supercuádricas para realizar una estimación de la geometría del objeto más adecuada. Utilizando estas aproximaciones, se propone la aplicación de la caracterización del agarre tanto heurística, como analítica [Rubert et al., 2018], siempre basada en la información sensorial obtenida (*data driven*). Muchos de los trabajos propuestos se focalizan en una de las fases de la manipulación, ya sea en la estimación inicial o en la especificación. Uno de los objetivos de este trabajo es el de integrar los algoritmos desarrollados para mostrar que pueden ser utilizados en un entorno real.

2.3. Metodología del sistema de manipulación

Como se ha visto en el capítulo anterior en las intervenciones submarinas se ha seguido un enfoque en el desarrollo muy orientado al control. No obstante, las últimas investigaciones proponen el uso de técnicas de visión y manipulación aplicadas a este ámbito. En base al estado del arte actual en manipulación robótica, se propone la aplicación al entorno submarino de los siguientes conceptos:

- Reconstrucción 3D de escenarios mediante visión estéreo y segmentación basada en láser.
- Segmentación, estimación y seguimiento de objetos mediante técnicas de RANSAC.
- Segmentación, modelado y seguimiento de objetos mediante estimación de SQ.

- Síntesis del agarre utilizando la geometría del objeto de forma heurística y de forma analítica.

2.4. Metodología de validación experimental

Debido a las dificultades que aparecen en el entorno subacuático, es necesario establecer una metodología de trabajo adecuada para poder obtener resultados experimentales satisfactorios. Siguiendo los conocimientos adquiridos en proyectos de investigación recientes, se ha seguido una metodología de validación experimental cíclica que trata de garantizar el éxito en los experimentos en el mar. Al mismo tiempo, da validez e interés a los experimentos realizados tanto en simulación como en entornos reales más sencillos, como pueden ser un tanque de agua o una piscina. El objetivo de esta validación es independiente del problema específico de intervención subacuática que se desea resolver y de la mecatrónica disponible. Esta metodología se desglosa en las siguientes fases, en orden creciente de complejidad:

- **Simulación** del sistema y los algoritmos mediante el simulador submarino UnderWater Simulator (UWSim) [Prats et al., 2012c]. Esta fase permite comprobar el funcionamiento de los algoritmos de una forma muy controlada. Introduciendo el ruido deseado en el sistema se puede comprobar su robustez. Además, permite comprobar la integración de los diferentes elementos *software* que tienen que interactuar para llevar a cabo la intervención.
- Pruebas de **interacción física** en un tanque de agua, permitiendo probar los algoritmos de visión y manipulación en un entorno real. De esta forma se prueban los algoritmos con el sistema brazo-mano real, así como con diferentes cámaras. Esta fase sigue siendo controlada en la medida que no interviene necesariamente el vehículo ni otros elementos como pueden ser las corrientes. En este trabajo en particular, es un escenario de una gran importancia ya que se pueden realizar experimentos sin restricciones, a diferencia de los escenarios de integración.
- Pruebas de **integración** en piscina en el CIRS de la UdG, que permiten probar el sistema brazo-mano montado en un AUV. En esta fase la integración con el vehículo ofrece más grados de libertad al sistema. Por ello, el control es más complejo y aparecen nuevos problemas asociados a la dinámica del agua y a la navegación. La integración de un mayor número de sistemas es también un reto, así como la aparición de un nuevo escenario y nuevas incertidumbres.
- Pruebas en el mar, que suponen el escenario de **validación final**. A esta fase se llega únicamente cuando todo ha sido probado satisfactoriamente en las anteriores fases. Se introducen nuevos elementos reales como son el agua salada (que afecta adversamente a la mecatrónica), las corrientes marinas, las partículas en suspensión del agua, el entorno desconocido, una mayor profundidad de trabajo, las condiciones de iluminación variables, el *flickering* debido a la luz solar, etc. Pese a que estas pruebas se realizan en un entorno acotado como es un puerto, ya suponen una complejidad muy grande y los sistemas que se consiguen validar en estas condiciones cuasi-reales se consideran un *benchmark* para dicho tipo de intervención.

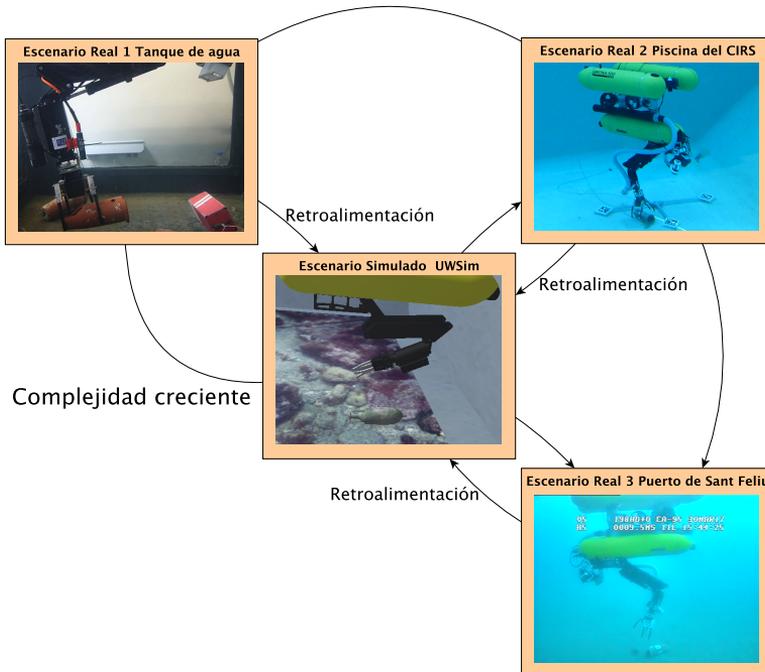


Figura 2.1: Metodología de validación experimental por fases.

Estas fases se resumen en la Figura 2.1. Esta metodología es cíclica en el sentido que los experimentos en fases avanzadas aportan retroalimentación a las fases anteriores, que pueden ser repetidas para introducir mejoras que no habrían podido ser anticipadas sin la realización de estos experimentos. Además, las dos primeras fases son de gran importancia ya que permiten detectar errores y posibles debilidades antes de realizar las intervenciones reales, de forma que se ahorran recursos y tiempo. Los algoritmos y arquitectura desarrollados en esta tesis han sido probados en diferentes estadios de validación. Por sus costes logísticos y humanos no es posible evaluarlos continuamente en los escenarios finales, que requieren esfuerzos conjuntos por parte de varias universidades. En esencia, el escenario de interacción física en el laboratorio del IRS Lab es el principal banco de pruebas en esta tesis.

2.5. Definición de los escenarios de validación

Siguiendo la metodología propuesta, los algoritmos desarrollados en este trabajo se han demostrado por fases en diferentes escenarios de validación. A continuación se describen en mayor detalle cada uno de los escenarios donde se llevan a cabo los experimentos.

2.5.1. Validación en simulación

Los algoritmos de visión y control del robot se han desarrollado en primer lugar utilizando simulaciones. El simulador submarino UWSim, que se describirá en de-

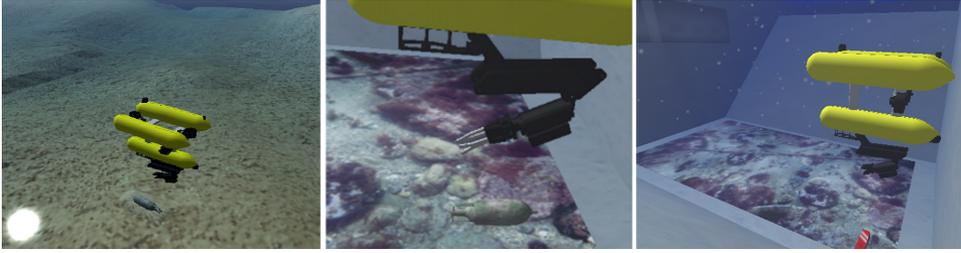


Figura 2.2: Escenarios de los experimentos simulados. El AUV GIRONA500 usa el brazo ARM5E. Escenario con un fondo arenoso y una vasija (izquierda); con el suelo texturizado y una vasija (centro); y con un modelo de caja negra (derecha).

talle más adelante, ofrece la posibilidad de simular actuadores y sensores. De esta manera, se puede obtener una aproximación realista del comportamiento del robot. Gracias a la flexibilidad del mismo, se han realizado pruebas en diversos escenarios disponiendo de forma virtual de la misma mecatrónica que los sistemas reales y la posibilidad de utilizar sensores análogos y también otros que no están disponibles. Esto es especialmente notable en el caso de la visión, donde los parámetros ópticos de las cámaras se pueden configurar totalmente. Así, se puede replicar el comportamiento de las cámaras reales y simular sensores más avanzados que los disponibles en el laboratorio. Por otro lado, es posible introducir ruido artificial en las imágenes generadas. En la Figura 2.2 se muestran algunos de los escenarios utilizados, que son los siguientes:

- Escenario de arqueología: este escenario simular un suelo arenoso en el mar. El fondo consiste en una textura que simula la arena. En el escenario se utiliza el I-AUV GIRONA500 y como objeto se dispone de una vasija. Este escenario presenta una variante en la que el objeto está parcialmente enterrado. Se utiliza en la aplicación del sistema a la limpieza de objetos parcialmente enterrados.
- Escenario en el CIRS de la UdG I: esta escena simula las instalaciones ubicadas en Gerona. El fondo consiste en un mosaico con textura que puede ser reconocido de forma excelente mediante visión estéreo. En el escenario se disponen I-AUV GIRONA500 y una vasija cilíndrica .
- Escenario en el CIRS de la UdG II: se trata de la misma escena anterior utilizando como objeto un modelo de caja negra. Al igual que en los escenarios previos se utiliza el I-AUV GIRONA 500 con el manipulador ARM5E.

2.5.2. Validación en el laboratorio IRS Lab

El principal escenario de validación en el contexto de esta tesis es el tanque de agua ubicado en el laboratorio del IRS Lab en las instalaciones UJI. Este tanque tiene unas dimensiones de 2m. de ancho, 2m. de largo y 1.5m de alto. Dispone de laterales de cristal que permiten pasar luz tanto del exterior como luz artificial.

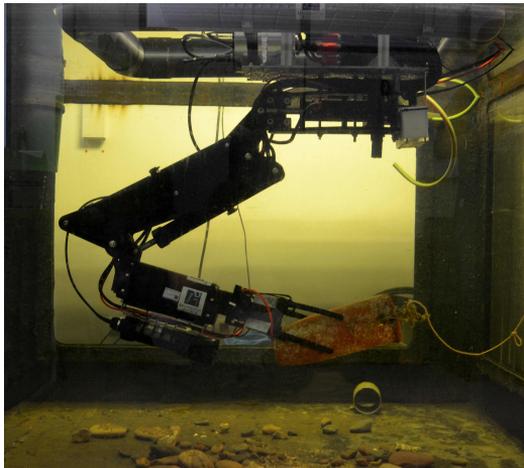


Figura 2.3: Experimentos en el tanque de agua del IRS Lab. El brazo robótico se acopla a una plataforma. La cámara estéreo se instala en la base y el proyector láser lineal en el brazo.

Como se puede ver en la Figura 2.3, esta configuración permite tener una visión directa del sistema mientras se está ejecutando, algo que no es posible en el mar.

En este laboratorio se dispone del brazo manipulador ARM5E, diferentes sensores y múltiples objetos que se describen en secciones subsiguientes. En este entorno se dispone de una plataforma flotante que permite utilizar el brazo de forma fija o en flotación para introducir en el sistema ruido en forma de perturbaciones. Además, las condiciones de iluminación se pueden variar y tomar mediciones con diferentes combinaciones de luz artificial y natural. Este escenario de validación es el más utilizado durante el transcurso de la tesis ya que es posible variar las condiciones de trabajo libremente y no se enmarca en un contexto de trabajo con otras universidades.

En los experimentos realizados en el laboratorio se pueden identificar los siguientes escenarios:

- Manipulación de una **caja negra**: en el escenario que se muestra en la Figura 2.3 se dispone el ARM5E en la plataforma fija y una maqueta a escala real de caja negra. Adicionalmente se dispone de otra caja con diferentes dimensiones y material.
- Escenario de manipulación **arqueológico**: en este escenario se utiliza la misma configuración del robot y se manipula un objeto similar a los que se pueden encontrar en un yacimiento arqueológico. Se dispone de diferentes objetos pero se considera un único objeto por escena.
- Escenario de manipulación con **múltiples objetos**: en este escenario se disponen múltiples objetos de diferentes formas.



Figura 2.4: Instalaciones del CIRS (izquierda). Centro de control con visión de la piscina (derecha).

2.5.3. Validación en la piscina del CIRS

Como parte de los proyectos TRITON y MERBOTS, se han realizado experimentos en las instalaciones del CIRS de la UdG. En este centro se encuentra el Laboratorio de Robótica Submarina del grupo de investigación Visión por Computador y Robótica (ViCoRob) de la misma universidad. Estas instalaciones disponen de una piscina de 5 metros de profundidad, 16 de longitud y 8 de anchura. Estas dimensiones permiten realizar una manipulación realista con el vehículo GIRONA500 con el brazo ARM5E acoplado. Además, como en el caso del tanque de agua, es posible tener una visión directa del robot durante el transcurso de los experimentos, ya que el espacio de trabajo dispone de una ventana que muestra la piscina (Figura 2.4). En estos experimentos se ha considerado un único escenario que es la manipulación de una vasija con el I-AUV GIRONA500.

En este escenario se ha evaluado el *framework* desarrollado para manipular una vasija mediante el uso de RANSAC. Posteriormente, se han utilizado los datos obtenidos durante los experimentos reales para evaluar el resto de métodos con un conjunto de datos realista a modo de *benchmark*. Con respecto a otros escenarios, en esta fase de validación se dispone de una reconstrucción 3D del entorno casi ideal mientras que en el aspecto del control se introduce la dificultad de lidiar con el sistemas de navegación y el control conjunto del manipulador y el vehículo. Además, la integración con otros robots y componentes también suponen un desafío para evaluar la modularidad del sistema.

2.5.4. Validación en experimentos en el mar

En relación a los experimentos en un entorno real, también se han realizado intervenciones en el mar en el contexto del proyectos MERBOTS (Figura 2.5). Durante los experimentos, se probaron las partes de visión y especificación del sistema propuestas en esta tesis. Adicionalmente, se tomaron datos reales de los robots para poder desarrollar mejores algoritmos de visión. Estos experimentos se llevaron a cabo en el puerto de Sant Feliu de Guíxols (Gerona). La ubicación del escenario dentro de un puerto reduce la influencia de las corrientes marinas, facilita el despliegue de los vehículos desde un pequeño barco y ofrece un fondo marino de 12 metros de profundidad y una superficie de arena. Al igual que en la piscina, el escenario único es el de la manipulación de una vasija. En este escenario se introducen multitud de desafíos como son: las corrientes de agua, la pérdida de capacidad de supervisión visual del robot, la reducción de la visibilidad y capacidad



Figura 2.5: Los experimentos realizados en el mar tienen lugar en el interior del puerto de Sant Feliu de Guíxols.

de supervisión, la imposibilidad de utilizar el sensor láser, la necesidad de recalibrar los sistemas de visión, etc. Algunos de estos desafíos se tratan en este trabajo, como los relacionados con la visión, mientras que otros están fuera del alcance de esta tesis.

2.6. Experimentos propuestos

Los proyectos que se han mencionado en la introducción proponen diferentes objetivos y casos de uso. El sistema desarrollado está enfocado a resolver problemas que pueden aparecer en la manipulación subacuática en entornos reales permitiendo avanzar en el estado del arte actual de la robótica submarina. Los casos de uso o problemas a resolver están directamente relacionados con estos proyectos y son los siguientes:

- **Manipulación de objetos arqueológicos:** como pueden ser ánfora, vasijas, tazas o culebrinas. Estos objetos, se consideran el objetivo principal de MERBOTS, centrado en la arqueología subacuática. En la Figura 2.6 se pueden ver algunos ejemplos utilizados en este trabajo, cuya forma predominante son los cilindros. En la Tabla 2.1 se detallan las dimensiones reales máximas de los objetos.
- **Limpieza de objetos arqueológicos:** aunque no es el experimento principal, se propone este caso como aplicación de las técnicas que se van a utilizar en la manipulación a un caso de uso diferente en el que se desea extraer la arena que cubre parcialmente un objeto.
- **Manipulación de una caja negra:** en la línea de proyectos anteriores, se considera la manipulación de la caja negra de un avión, como muestra de un objeto de fabricación actual, en contraposición a los objetos históricos, como vasijas antiguas. Este tipo de objetos, que ya se recuperaban con un gancho en el proyecto RAUVI, aparecen en los escenarios de validación de TRIDENT y TRITON. En la Figura 2.6 se puede ver el objeto *A*, una maqueta de una caja negra, y el objeto *B*, una caja de diferentes dimensiones y material.

Finalmente, cabe destacar que no todos los experimentos realizados son parte de los proyectos en los que participa el IRS Lab. Sin embargo, sí se guían por las

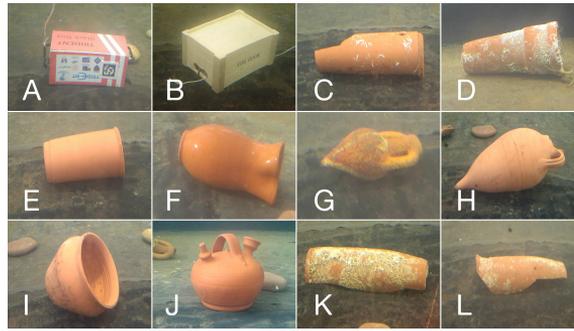


Figura 2.6: Objetos reales utilizados en el laboratorio. Se han utilizado cajas y objetos cerámicos con formas variadas.

Tabla 2.1: Dimensiones de los objetos.

Identificador	Longitud(mm.)	Anchura(mm.)	Altura(mm.)
A	300	160	135
B	280	193	125
C	340	140	140
D	350	180	180
E	200	120	120
F	260	135	135
G	180	90	90
H	470	250	250
I	270	270	200
J	220	195	195
K	350	140	140
L	350	140	140

necesidades que ofrecen estos proyectos, ya que proponen aplicaciones realistas en la línea del estado del arte y suponen un gran desafío para la robótica marina actual.

2.7. Arquitectura hardware del sistema

Un factor determinante para la capacidad de realizar intervenciones submarinas son los robots disponibles y su equipamiento. En esta sección se detalla la mecatrónica y sensores utilizados. En este sentido, se describe el brazo manipulador utilizado, el efector final del mismo, el AUV disponible en simulación y en los experimentos de integración, así como diferentes sensores de que disponen estos robots. Además, se explican los mecanismos de control de todos estos elementos. Por otro lado, se introducen los sistemas de visión y se describen en más detalle en el Capítulo 3.



Figura 2.7: Brazo robótico eléctrico ARM5E con 4 GDL (izquierda). Efector final (derecha).

2.7.1. Brazo robótico ARM5E

Los elementos primordiales para la manipulación son el brazo robótico ECA-CSIP Light-weight ARM5E [Fernández et al., 2013] y su efector final (Figura 2.7). Se trata de un brazo robótico eléctrico con 4 GDL desarrollado durante el proyecto RAUVI [Prats et al., 2012d]. Este manipulador se diferencia de los que se pueden encontrar en un entorno robótico normal en que no dispone de dedos articulados, cuya complejidad técnica en el medio marino es mucho mayor [Prats et al., 2012b]. Por el contrario, dispone de un manipulador con 1 GDL: apertura y cierre. En este trabajo, se utiliza con un efector final en forma de agarra. El ARM5E fue diseñado con el objetivo de acoplarse a AUV ligeros como el GIRONA500, gracias a su peso de 12Kg. en agua dulce y 29Kg. en el aire. Tiene una capacidad de carga de 12 Kg. totalmente extendido y es capaz de operar en profundidades de hasta 300 metros. Por lo tanto, sus especificaciones son adecuadas para los requisitos de las misiones propuestas.

Descripción de la mecatrónica

Mecánicamente, el manipulador se compone de 4 cilindros que contienen un motor eléctrico sin escobillas alimentado a 24V., excepto el último cilindro que contiene dos (siendo uno para controlar la apertura del efector final). Los motores proveen una fuerza lineal que se convierte en movimiento angular gracias al diseño de la estructura. Debido a este diseño, los motores tienen un espacio de trabajo limitado, pero el brazo es capaz de alcanzar distancias de hasta 1 m. Se definen las articulaciones desde la base del robot como *Slew*, *Shoulder*, *Elbow* y *Wrist*. La articulación *Slew* controla el movimiento del brazo en horizontal permitiendo un giro de 120 grados hacia un lado de la base. *Shoulder* y *Elbow* controlan el movimiento en vertical con un espacio de trabajo de 90 y 130 grados, respectivamente. *Wrist* es una articulación continua que gira el efector final sin límites mecánicos. Finalmente, el último motor es capaz de abrir y cerrar el efector final con una fuerza de 50kg por lo que en general es capaz de coger cualquier objeto que pueda introducirse correctamente entre sus garras. El brazo dispone de un cilindro con la electrónica de control a bajo nivel de las articulaciones que se comunica con el ordenador del robot mediante un enlace RS232.

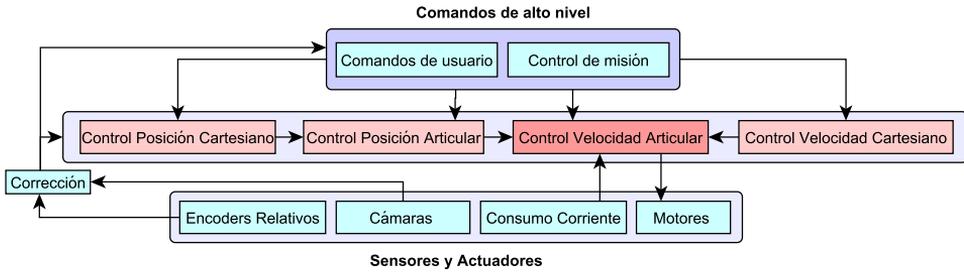


Figura 2.8: Arquitectura de control del ARM5E.

Controlador software

El brazo ARM5E es un prototipo basado en manipuladores que se utilizan en ROV ligeros. Por este motivo el sistema de control que ofrecen se limita al control en velocidad directo en bucle cerrado de cada articulación. Para ofrecer una funcionalidad mayor se ha desarrollado la arquitectura de control que se muestra en la Figura 2.8. Los comandos de alto nivel, que provienen del usuario o del propio sistema, se envían a un controlador en cascada. El controlador, a su vez, se comunica con el *hardware* para llevar a cabo los movimientos y obtener retroalimentación del robot. Estos componentes se detallan en el Capítulo 6.

Como se ha explicado, el control articular en velocidad se realiza internamente en la controladora del brazo, que se comunica mediante RS232. El control de la posición del efector final del brazo y el control cinemático se realiza en un cilindro estanco que dispone de un ordenador con conexión con la superficie o con el vehículo a través de Ethernet.

Como los brazos manipuladores que se instalan en ROV, está diseñado para ser utilizado de forma teleoperada, por lo que dispone únicamente de codificadores o *encoders* relativos de efecto Hall que pierden la posición del robot al perder la alimentación. Por ello, no es posible saber con precisión la posición absoluta de sus articulaciones. Esta es, junto con el limitado número de GDL, su principal debilidad. El brazo dispone de un controlador de velocidad en bucle cerrado integrado para cada articulación, mientras que la capa de control en posición se ha desarrollado a nivel de *software*. En cuanto a sensorización, ofrece la información de los citados *encoders* relativos y la corriente general consumida. Esta información es utilizada para determinar que se ha llegado al límite de una articulación, así como para determinar que un objeto ha sido cogido.

Cinemática directa

La Cinemática Directa o Forward Kinematics (FK) es un método matemático basado en ecuaciones cinemáticas que permite calcular la posición y orientación del efector final del robot x_b a partir de la posición angular de las articulaciones, donde b es la base de la cadena cinemática. La notación Denavit-Hartenberg (D-H) [Denavit y Hartenberg, 1955] se utiliza en robótica como convención para definir la cinemática de un robot.

En esta convención se sitúan sistemas de coordenadas en las articulaciones entre

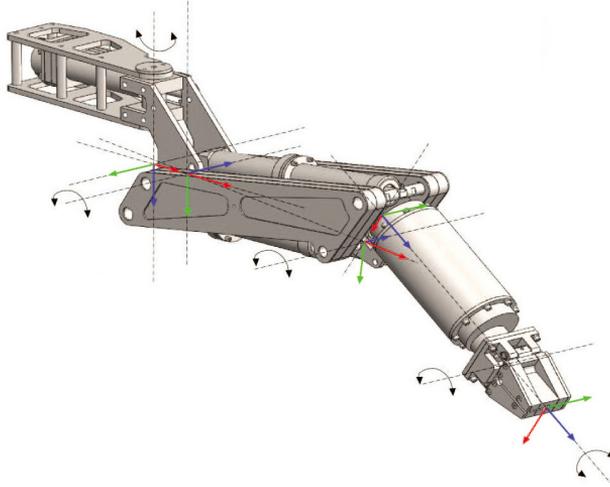


Figura 2.9: Modelo 3D del brazo robótico con los sistemas de referencia de cada articulación.

dos eslabones, de forma que se asocia una transformación al sistema de coordenadas de la articulación $[Z]$ y se asocia un segundo sistema de coordenadas al eslabón $[X]$. Así, se puede obtener la transformación relativa al efector final o cualquier articulación intermedia por medio de la concatenación $[T] = [Z1][X1][Z2][X2]...[Zn][Xn]$. Para definir las transformaciones $[Z]$ y $[X]$ la convención define los siguientes parámetros por cada articulación:

- θ : ángulo alrededor de la z previa. Es decir, ángulo entre la x previa y la nueva.
- d : desplazamiento a lo largo del z inicial hasta la normal común.
- a : longitud de la normal común, normalmente el radio alrededor del eje z inicial.
- α : ángulo alrededor de la normal común. Es decir, ángulo entre la z previa y la actual.

En la Figura 2.9 se muestra el robot en la posición inicial de las articulaciones cuyo valor angular es cero. En la Tabla 2.2 se muestran los parámetros D-H del robot. Las distancias entre enlaces son $a_1 = 0,08\text{m}$, $a_2 = 0,44\text{m}$, $a_3 = -0,08$, y d_4

Tabla 2.2: Parámetros D-H de la cadena cinemática del brazo ARM5E.

Articulación	θ	d	a	α
<i>Stew</i>	q_1	0	a_1	90
<i>Shoulder</i>	$q_2 + o_2$	0	a_2	0
<i>Elbow</i>	$q_3 + o_3$	0	a_3	90
<i>Wrist</i>	q_4	d_4	0	0

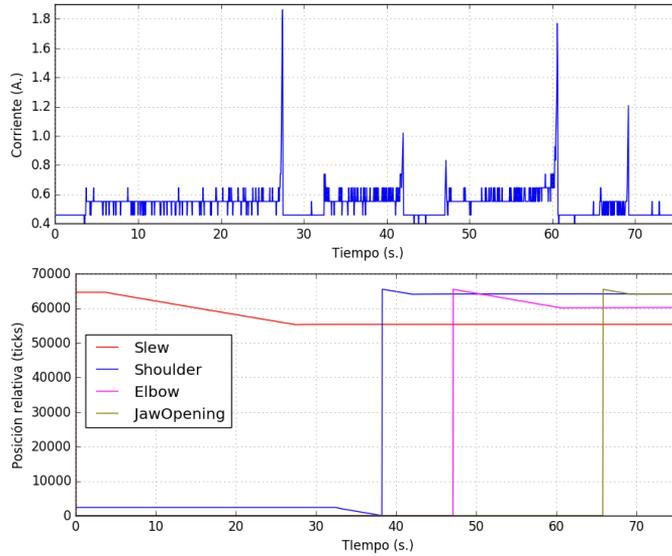


Figura 2.10: Evolución del consumo del brazo ARM5E durante el procedimiento de calibración.

$= 0,56$. Los valores $o_2 = 7.98$ y $o_3 = 127.02$ son desplazamientos que se aplican a la posición angular de las articulaciones (q_1, q_2, q_3, q_4) de acuerdo a la calibración del robot.

Cinemática inversa

La Cinemática Inversa o Inverse Kinematics (IK) es el método utilizado para calcular la configuración articular a la que se debe mover el robot para alcanzar una posición y orientación deseada. Existen aproximaciones clásicas para resolver este problema, tanto de forma geométrica como numérica [Angeles, 1985]. En este caso se ha utilizado un método numérico [Grudic y Lawrence, 1993] que utiliza un algoritmo iterativo para encontrar la solución. Esta alternativa brinda una mayor flexibilidad a la hora de utilizar el sistema con diferentes manipuladores. Esto es interesante puesto que en simulación se ha probado con otros brazos y la cinemática con el sistema completo también es diferente. Además, los métodos iterativos obtienen una solución aproximada cuando la posición deseada no es alcanzable. La posibilidad de obtener una solución aproximada resulta muy interesante al utilizar el ARM5E debido a su bajo número de GDL.

Por otro lado, los robots que incluyen al vehículo en la cadena cinemática o utilizan manipuladores con un mayor número de GDL requieren de sistemas capaces de controlar la redundancia del sistema [Siciliano, 1990]. En el Capítulo 6 se dan más detalles sobre el uso de la IK.

Calibración

Debido a la citada ausencia de codificadores absolutos, es necesario un proceso de calibración inicial. Este método consiste en mover las articulaciones a su límite

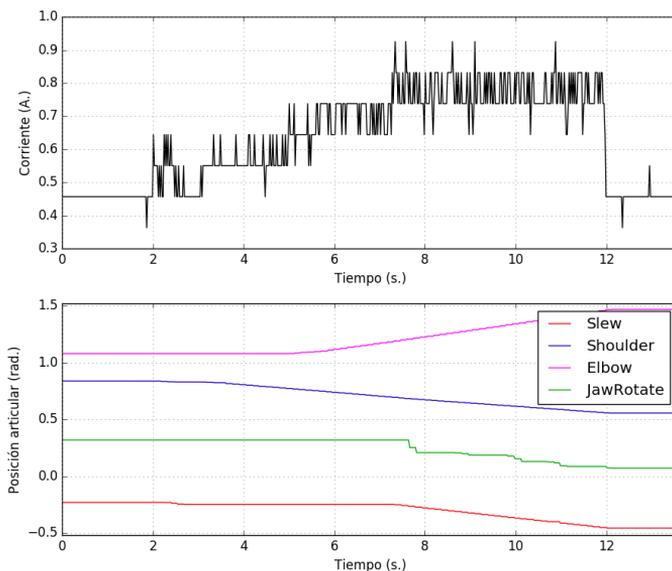


Figura 2.11: Evolución del consumo del brazo ARM5E durante el movimiento.

mecánico para inicializar la posición de los codificadores relativos. Sin embargo, tampoco dispone de sensores de final de carrera para determinar los límites del brazo. Por este motivo la principal fuente de información es la corriente consumida.

Para realizar la calibración se mueve cada articulación de forma individual hasta alcanzar el límite de las articulaciones, que se manifiesta a través de un incremento notable en la corriente consumida. La posición de la articulación *Wrist* se calibra de forma visual y manual utilizando marcas de alineamiento, ya que no tiene un límite físico. Los codificadores relativos envían al controlador el número de tics, que representa la posición de la articulación. Cuando se llega al límite se almacena el valor de número de tics actual. En cada momento se compara el valor actual con el inicial y se convierte en posición angular utilizando el modelo mecánico de las articulaciones [Fernández et al., 2013].

En las figuras 2.10 a 2.12 se muestra una gráfica con la evolución de la corriente consumida en el tiempo y otra gráfica con la variación de la posición de las articulaciones.

En la Figura 2.10 se muestra la respuesta de corriente al llegar al límite de las articulaciones. El valor de las articulaciones se muestra en tics, ya que la calibración está en proceso y no se conoce la posición absoluta.

Por otro lado, en la Figura 2.11 se puede ver la respuesta que tiene el uso de diferentes articulaciones de forma simultánea con respecto a la posición angular absoluta del brazo en radianes. Además, el valor de corriente se utiliza también para controlar el efector final y determinar que el objeto ha sido cogido de forma satisfactoria (Figura 2.12).

El valor obtenido mediante estos sensores tiene un error que se incrementa progresivamente por lo que resulta necesario realizar la calibración cada cierto tiempo. Además, el uso de la corriente consumida introduce errores en la posición

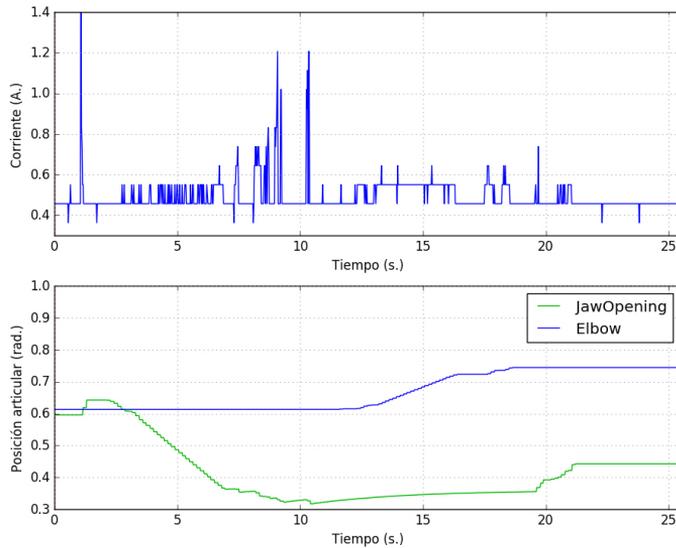


Figura 2.12: Evolución del consumo del brazo ARM5E durante la manipulación.

inicial del brazo. Esta corriente se ve afectada por factores externos y no puede considerarse que la respuesta sea idéntica. Estos problemas reducen la repetibilidad y precisión del sistema en gran medida. Este proceso, que además es lento y requiere supervisión, se complementa con el uso de la visión para la detección del efector final, como se explica en el Capítulo 3. Este tipo de problemas se compensan con el seguimiento visual del brazo y los objetos.

Efector final

El efector final original del brazo consiste en una articulación con forma de pinza. Este elemento es multipropósito, pues dispone de una inserción en forma de T que le permite realizar tareas como sostener una herramienta con forma de gancho [Prats et al., 2012d], sostener un conector [Palomeras et al., 2016] o girar una válvula [Palomeras et al., 2016]. Además, puede disponer de unas extensiones más largas, también referidas en este trabajo como garras, que le permiten manipular objetos. Esta aproximación, que se puede ver en las figuras 2.7 y 2.9, es la utilizada en esta tesis. Esta configuración permite manipular objetos con un tamaño medio como cilindros, cajas, esferas, etc. Aunque no se trata de un manipulador diestro, la capacidad resultante de agarres es relativamente elevada. Disponer de un efector final de más grados de libertad resultaría interesante pero es técnicamente complejo y caro. Esta limitación impide coger correctamente objetos pequeños que no ocupen el espacio mínimo entre los dedos de la pinza. Como alternativa al uso de uno de los dos efectores finales, en el contexto del proyecto TRITON se desarrolló la mano robótica UJI-ONE, capaz de alternar entre dos modos. Como se muestra en la Figura 2.13, es posible utilizarla para manipular objetos con sus garras o utilizar herramientas tipo T u operar válvulas.

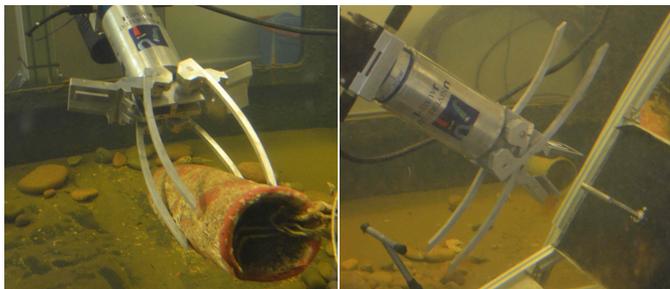


Figura 2.13: La mano UJI-ONE puede manipular objetos (izquierda) y herramientas tipo T (derecha).

2.7.2. Plataforma móvil GIRONA500

El I-AUV que se utiliza en los experimentos de integración está formado por el brazo manipulador ARM5E acoplado al AUV GIRONA500 [Ribas et al., 2012]. El GIRONA500 es un AUV ligero diseñado y desarrollado en la UdG que puede alcanzar una profundidad de 500 m. El vehículo dispone de tres cascos con forma de torpedo con las baterías en el casco inferior y la electrónica en los superiores. La distribución de masas del vehículo lo hace estable en el *roll* y el *pitch* por lo que resulta una excelente plataforma para la manipulación y exploración. Esta configuración dispone de 4 GDL: *surge*, *sway*, *heave* y *yaw*. Además, el GIRONA500 dispone de una gran cantidad de sensores: cámaras utilizadas para navegación, sensores inerciales (*Attitude and Heading Reference System*), sensores de presión, DVL (*Doppler Velocity Logger*), etc. Finalmente, el robot cuenta con varios equipos de procesamiento integrados e interconectados: un ordenador que controla el vehículo, uno que controla la manipulación y otro que procesa la visión. El robot se utiliza en estos experimentos conectado a superficie mediante un cable Ethernet, lo cuál permite monitorizar y supervisar la misión en todo momento a través de HRI. No obstante, el robot es técnicamente independiente y la alimentación proviene de un conjunto de baterías.

El conjunto, que se muestra en la Figura 2.14, ofrece 8 GDL para permitir una manipulación de objetos mucho mejor. La ejecución del agarre en esta configuración se controla mediante un algoritmo de control por tareas [Peñalver et al., 2018], que se detalla más adelante. El control de la manipulación del sistema vehículo-manipulador siempre se hace a nivel cinemático. El control dinámico del sistema es todavía un gran desafío en el estado del arte, ya que debe tener en cuenta de forma conjunta la dinámica e hidrodinámica del brazo y el vehículo. Sin embargo, estas técnicas están fuera del alcance de esta tesis.

En esta configuración, los algoritmos más complejos se ejecutan en el cilindro de visión, aunque normalmente los algoritmos de la tesis se desarrollan en un ordenador en superficie. Esto se hace así porque permite depurar mejor el sistema, disponer de un *hardware* de procesamiento más potente y científicamente es similar ya que el poder hacer este cálculo internamente depende principalmente de disponer de un presupuesto mayor. Esto es debido a que los equipos empotrados suelen tener una menor capacidad de procesamiento y aquellos que disponen de ella son muy costosos.

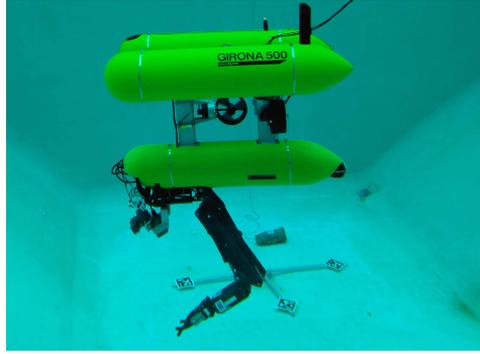


Figura 2.14: El I-AUV compuesto por el GIRONA500 y el brazo ARM5E.

2.7.3. Equipos de cálculo y visión

En los experimentos de laboratorio aquí descritos se utiliza como equipos de cálculo principal un ordenador con un Intel Core 2 Quad Q8200 a 2.33GHz de 4 núcleos con 8GB de RAM y una tarjeta gráfica NVIDIA GTX 280X, así como un ordenador con un Intel i7 4790 a 3.60 GHz. de 8 núcleos con 16GB de RAM y una tarjeta gráfica NVIDIA GTX960. Estos disponen de interfaces FireWire para el uso de las cámaras estéreo, que hacen uso de este interfaz. El control del brazo se realiza en un ordenador empotrado con un procesador VIA Nano U3300 a 1.2 GHz., 2GB de RAM y una tarjeta gráfica integrada. Aunque este equipo es más modesto, tiene un tiempo de respuesta adecuado para el control del brazo.

Con respecto a la visión, se dispone de dos cámaras estéreo (Videre Stereo Camera y Code Laboratories DUO M), así como un sistema de reconstrucción láser basado en un proyector láser y una cámara monocular. En los experimentos de integración se dispone de una cámara estéreo a color Point Grey Bumblebee 2 y un sistema de reconstrucción láser con una tecnología diferente. En el robot simulado se dispone de una cámara virtual de alta resolución a color. Los sistemas de visión utilizados en cada caso se detallan en el capítulo siguiente.

2.8. Arquitectura software del sistema

Los algoritmos desarrollados en el contexto de esta tesis tienen como objetivo la manipulación de objetos guiada por visión. En esta sección se describe la arquitectura general del sistema que se ha utilizado en el desarrollo de esta tesis con componentes modulares, de modo que se pueden intercambiar o modificar de forma sencilla. En la Figura 2.15 se muestran los diferentes módulos descritos a alto nivel y la interacción entre ellos. Estos módulos se explican en mayor detalle en los capítulos siguientes.

En primer lugar, el módulo de **Visión** artificial se encarga de obtener una reconstrucción 3D del entorno haciendo uso de los diferentes sensores disponibles. Además, trata de mejorar la reconstrucción integrando información de varios sensores o del mismo sensor en el tiempo. Después utiliza técnicas de filtrado de la

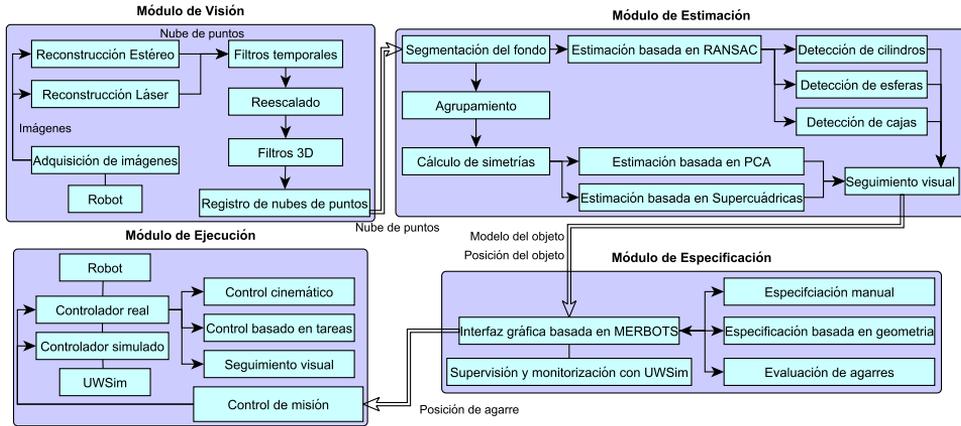


Figura 2.15: Estructura modular del sistema propuesto.

información para obtener el mejor resultado posible. El registro de nubes de puntos se utiliza para integrar información de múltiples vistas del escenario.

En segundo lugar, en el módulo de **Estimación** se realiza la segmentación de la escena para separar los objetos que se desean manipular del fondo de la escena. A continuación, se obtiene la estimación de la posición, orientación y características geométricas del objeto. Esta estimación se puede realizar con diversos algoritmos, por lo que se dispone de diversos módulos intercambiables entre sí que tienen diferentes ventajas y desventajas. Además de los métodos de estimación, en este módulo están incluidos los algoritmos de cálculo de simetrías y agrupamiento.

En el módulo de **Especificación** del agarre se obtiene la posición del efector final para la manipulación del objeto. Esta fase se realiza en función de la geometría del objeto encontrado y puede realizarse de forma completamente automática utilizando una clasificación de la posición de agarre o utilizando información que proporciona el usuario. En este aspecto, el módulo de HRI está interconectado con los módulos anteriores y permite al usuario supervisar el resultado de los módulos anteriores para validar o modificar la especificación del agarre. Además de los componentes de HRI, este módulo incluye la especificación heurística y analítica del agarre.

Una vez se ha validado la posición de agarre, el módulo de **Ejecución** se utiliza para realizar el control de la intervención. Este módulo realiza el control del brazo o el I-AUV a nivel cinemático para aproximar el efector final a la posición deseada y cogerlo. Después, se realiza el transporte del objeto. Este control también se lleva a cabo con un controlador u otro dependiendo del robot utilizado, ya sea en simulación o en real, y dependiendo de si el ARM5E se utiliza en conjunto con el AUV GIRONA500.

En la Figura 2.15 se muestran los diferentes módulos descritos a alto nivel y la interacción entre ellos. El software desarrollado está disponible de forma abierta para que pueda ser utilizado por cualquier miembro de la comunidad científica.

Como resultado de este trabajo los módulos desarrollados serán publicados como código libre en la plataforma GitHub para que puedan ser utilizados por cualquier miembro de la comunidad científica. En la Figura 2.16 se muestran los diferen-

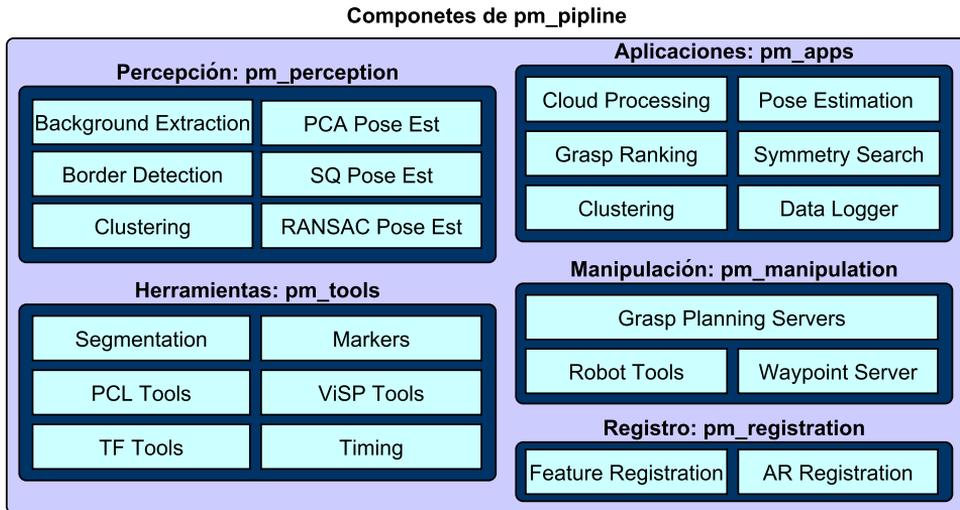


Figura 2.16: Estructura del repositorio de código libre ofrecido.

tes componentes del repositorio de código que se ofrecerán, llamado **pm_pipeline** (*Perception and Manipulation Pipeline*). El módulo **pm_apps** contiene ejemplos de uso, aplicaciones y test de las funciones desarrolladas. Por otro lado, el módulo **pm_manipulation** contiene las aplicaciones de planificación utilizadas principalmente para la manipulación, así como funciones para controlar el robot. El módulo **pm_perception** contiene los algoritmos de visión utilizados en los métodos mencionados previamente. En el módulo **pm_registration** se desarrolla una aplicación para el registro de nubes de puntos. Finalmente, el módulo **pm_tools** contiene funciones de uso habitual como conversiones de tipos, cálculos de tiempos y estadísticas, y *wrappers* sobre librerías como ViSP o Point Cloud Library (PCL). El trabajo desarrollado en esta tesis se ve también reflejado en el repositorio público del proyecto MERBOTS, que contiene la interfaz de usuario utilizada para la manipulación, así como en el repositorio de control del brazo robótico ARM5E del IRS Lab.

ROS y librerías externas

El desarrollo de un sistema complejo en el que intervienen tantos componentes sería imposible sin disponer de una sólida base sobre la que implementar los diferentes módulos. En este sentido, el uso de Robot Operating System (ROS) resulta esencial en el desarrollo de robots actuales [Quigley et al., 2009]. Definido como un meta sistema operativo, se trata de un *middleware* que se encarga de interconectar diferentes nodos utilizando *topics* y que ofrece multitud de herramientas útiles para el control y monitorización de robots. Además, dispone de una gran comunidad que ofrece paquetes que se pueden utilizar y modificar. En esta tesis se han utilizado múltiples paquetes de procesamiento de imágenes, calibración de cámaras, simulación, visualización y procesamiento de nubes de puntos, entre otros. En cada caso, se muestra cómo se han utilizado para adecuarlos al uso del robot

submarino, al mismo tiempo que se ponen a disposición los paquetes desarrollados en esta tesis.

En ROS, la mayoría de la comunicación entre los componentes se realiza mediante *topics*, que ofrecen un canal de comunicación entre dos nodos a través de mensajes con un tipo concreto. De esta manera es sencillo conocer las entradas y salidas que ofrece cada nodo. En ocasiones se utilizan herramientas más complejas como servicios o acciones, que ofrecen características similares pero que son más convenientes en ciertos aspectos. También se hace uso de las siguientes librerías:

- PCL [Rusu y Cousins, 2011] (Point Cloud Library): es una librería libre, independiente de gran escala para el procesamiento imágenes en 2D y nubes de puntos n-dimensionales. Es la herramienta más utilizada pues la mayoría de los algoritmos de visión interactúan con ella para el procesamiento de nubes de puntos en 3D.
- ViSP [Marchand et al., 2005] (Visual Servoing Platform): es una librería modular multiplataforma para el desarrollo de aplicaciones de *visual tracking* y *visual servoing* en tiempo real con múltiples aplicaciones en la robótica. De esta librería se utiliza principalmente sus funciones matemáticas y sus técnicas de detección sobre imágenes en 2D.
- KDL [Smits et al., 2011] (Kinematics and Dynamics Library): es una librería en C++ que permite calcular cadenas cinemáticas en tiempo real y se utiliza en el módulo de cinemática del ARM5E.
- QT [QT, 2018]: es un sistema de desarrollo de aplicaciones y Interfaces Gráficas de Usuario o GUI multiplataforma. La interfaz desarrollada en el contexto del proyecto MERBOTS que se utiliza para el control y manipulación del robot se desarrolla con QT.
- Eigen [Guennebaud et al., 2010]: es una librería de álgebra lineal escrita en C++ con plantillas. Se utiliza en este trabajo debido a sus excelentes capacidades de manipulación de matrices de transformación, vectores y los cálculos que se pueden realizar con estos tipos de datos.

2.8.1. Simulación con UWSim

En este trabajo se utiliza el simulador submarino UWSim [Prats et al., 2012c] para simular algunos de los casos de uso presentados, así como para mostrar información al usuario. Este simulador está disponible como software libre para la comunidad científica y ha sido utilizado en diversos proyectos financiados por la Unión Europea y laboratorios de investigación para simular intervenciones subacuáticas (ver Figura 2.17).

En el contexto de los proyectos de investigación realizados en el IRS Lab, UWSim ha crecido ofreciendo cada vez más funcionalidad, algunas desarrolladas en el contexto de esta tesis. Desarrollado en C++, utiliza Open Scene Graph (OSG) [Osfield et al., 2004] con *osgOcean* [Bale, 2012] como motor gráfico y *Bullet* [Coumans, 2012] como motor físico. UWSim es un proyecto en desarrollo activo disponible en <http://www.irs.uji.es/uwsim/>. Por otro lado, OSG es un motor gráfico utilizado en campos como la simulación visual, videojuegos, realidad virtual, visualización

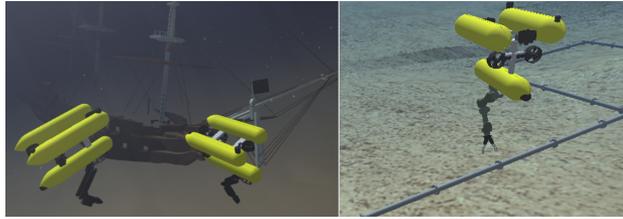


Figura 2.17: El simulador UWSim puede utilizar múltiples robots y manipuladores diferentes.

científica y el modelado. Está escrito también en C++ para múltiples plataformas. *osgOcean* es un proyecto de código libre que implementa renderizado de entornos submarinos en tiempo real utilizando OSG. *Bullet* es un motor de físicas libre y abierto que simula colisiones y dinámica. Se ha utilizado en videojuegos, efectos visuales para películas y es el motor de físicas de otros simuladores como Gazebo o V-REP. Este simulador se ha utilizado en numerosos proyectos como son TRITON, MERBOTS, MORPH [Kalwa et al., 2012], PANDORA y TRIDENT². UWSim utiliza las citadas librerías y dispone de funcionalidades para añadir a las escenas robots submarinos, sensores simulados y servir como interfaz con otros programas externos de control a través de ROS.

Configuración del simulador

El simulador UWSim permite la configuración de multitud de parámetros de la escena, objetos, robots y sensores. Es posible modificar parámetros relativos a la simulación del agua y del propio motor de UWSim. Por otro lado, como parte de la definición de cada robot se dispone de sensores, actuadores y cámaras. De esta forma, es posible desarrollar algoritmos para el robot sin tener que utilizar un robot real. A continuación se describen los principales elementos utilizados en este trabajo:

- **Cámara:** se trata de una cámara virtual totalmente parametrizada de forma que se puede definir su campo de visión, resolución y parámetros intrínsecos. Además, es posible añadir ruido gaussiano a las imágenes para simular imágenes imperfectas. En este trabajo se utilizan cámaras independientes y pares de cámaras para realizar reconstrucciones estéreo.
- **Proyector láser lineal:** este elemento consiste en un proyector láser lineal verde que se puede utilizar para realizar una reconstrucción 3D.
- **Herramienta de dragado:** consiste en un elemento que simula la succión de arena del fondo marino y se utiliza en el caso de uso de la limpieza de una ánfora para extraer la arena que la cubre parcialmente. Los objetos de la escena pueden ocultarse parcialmente especificando el porcentaje que se desea cubrir en el archivo de configuración de UWSim.

²UWSim en el proyecto TRIDENT: <https://www.youtube.com/watch?v=Hrj6wTw3bc>

- Sensor de fuerza: permite determinar la fuerza que se ejerce en un determinado punto.
- *Object picker*: este elemento consiste en un sensor de distancia que se puede añadir a las garras del robot y se dispara cuando el objeto está muy cerca de la garra. Este elemento es útil para la simulación visual de la manipulación en UWSim. Con respecto a la física, no permite realizar simulaciones realistas.

Se dispone de un motor de físicas que es capaz de detectar las colisiones entre diferentes objetos. Sin embargo, el proceso de determinación de las fuerzas y efectos resultantes sobre los objetos es muy complejo y no está completamente resuelto en este simulador. Como consecuencia es posible utilizar la simulación para observar los puntos de contacto del efector final y los objetos pero no es posible realizar una manipulación completa. Un módulo de dinámica está disponible como componente adicional. No obstante, éste trata los robots como un único objeto rígido de modo que no es capaz de simular las interacciones entre el AUV y el manipulador, ni con el manipulador y el entorno. Finalmente, el simulador dispone de un módulo de *benchmarking*. Este módulo permite realizar mediciones y comparar algoritmos [Pérez et al., 2015].

En el contexto de esta tesis, también se ha utilizado el simulador UWSim y los escenarios propuestos en robótica educativa [Pérez et al., 2017]. El simulador resulta una excelente herramienta educacional que permite introducir a los estudiantes en la problemática de la robótica y las particularidades del medio submarino. Este simulador se puede utilizar para realizar ejercicios, ofrecer escenarios *online* o para programar robots submarinos a través de lenguajes de programación visuales como Blockly [Trower y Gray, 2015].

2.9. Discusión y conclusiones

En este capítulo se ha presentado la metodología utilizada para realizar la manipulación y el sistema de validación experimental que se ha utilizado a la hora de desarrollar estrategias para la intervención subacuática. Este proceso iterativo en el que cada escenario es más complejo que el anterior ha resultado de ayuda a la hora de implementar algoritmos que sean capaces de superar los problemas que aparecen en entornos tan complejos y desestructurados.

Por otro lado se ha presentado la mecatrónica, cámaras, sensores y equipos utilizados. Se puede observar que los experimentos se deben adaptar a las posibilidades que ofrece el medio acuático así como a los robots y sensores disponibles para trabajar en él. Mientras que en la robótica en general se abaratan los costes y se mejoran los componentes a gran velocidad en el transcurso del tiempo, la robótica subacuática tiene que lidiar con una mecatrónica más limitada y costes mucho mayores. De hecho, sensores que en la robótica general son esenciales, como de profundidad basados en infrarrojos (también llamadas RGB-D si disponen de una cámara) y la mayoría de los sensores táctiles, ni siquiera son funcionales. En el capítulo siguiente se describen en mayor detalle estos problemas y las alternativas para paliarlos.

Problemática inherente a la visión subacuática

En este capítulo se tratan los desafíos a los que se enfrenta la visión submarina debido a las propiedades físicas del medio y, en particular, la reconstrucción 3D. El objetivo de este estudio es el de detallar la estrategia utilizada en este trabajo para obtener una representación 3D del entorno adecuada. Para ello, se realiza un estado del arte en visión y se muestran limitaciones que aparecen al trabajar bajo el agua, que restringen el tipo de sensores que pueden ser utilizados. A continuación, se habla sobre el estado del arte en cuanto a mejora de las reconstrucciones 3D mediante operaciones como filtrado o registro de nubes de puntos.

Finalmente, se explica el módulo de visión a un nivel más bajo que en el capítulo anterior. Este se subdivide en diferentes etapas: captura de imágenes 2D, reconstrucción 3D, obtención de nubes de puntos, filtrado para eliminar el ruido y reducción de resolución. Estos procesos preparan la información para posteriormente realizar la segmentación del fondo y los objetos, estimación de la posición, orientación y características de los objetos.

El trabajo de esta tesis en visión, ha sido desarrollado en parte en una estancia en el Vision for Robotics Group (V4R) supervisado por el profesor Markus Vincze en la Universidad Técnica de Viena (Austria). Durante la estancia, se utilizaron conjuntos de datos obtenidos en el entorno submarino para estudiar distintos algoritmos de reconstrucción, segmentación y seguimiento de objetos.

3.1. Motivación

En la actualidad, los robots son cada vez más capaces de determinar de forma autónoma cómo manipular un objeto. Sin embargo, estos casos se dan en entornos muy determinados (vaciar una caja o coger objetos de encima de una mesa), con una mecatrónica avanzada (brazos redundantes con manos diestras), se dispone de modelos completos de los objetos, y principalmente disponen de sensores 3D de gran precisión [Zennaro et al., 2015]. Por el contrario, en los escenarios propuestos solamente se tiene una idea aproximada del suelo marino (si es rocoso o arenoso, por ejemplo) y la forma aproximada de los objetos (si se trata de objetos cilíndricos

o se asemejan más a una caja). Así, uno de los grandes problemas es la percepción en 3D del entorno, ya que suele ser la fuente de información primaria para un robot manipulador.

Los desafíos del entorno marino en cuanto a percepción visual se deben a múltiples fenómenos. Pequeñas partículas reflejan la luz en dirección a la cámara, la luz que viaja a través del agua es absorbida reduciendo la iluminación del entorno, la luz se dispersa emborronando la imagen y se atenúa dependiendo de su longitud de onda alterando el color de las imágenes. Debido a ello, los sensores 3D que se utilizan en la actualidad no se pueden utilizar bajo el agua. Tanto los sensores que utilizan un patrón de luz proyectado como los que utilizan tiempo de vuelo hacen uso de luz infrarroja. Como se explica en las siguientes secciones, esta luz es la primera en ser absorbida y tiene un alcance de centímetros. Ante este problema, resulta esencial extraer el máximo rendimiento de los sistemas disponibles, como son las cámaras estéreo y la proyección de luz láser visible. Además, el correcto procesamiento de esta información es esencial para obtener reconstrucciones de calidad del entorno y los objetos que se hayan en él.

3.2. Visión artificial en el entorno subacuático

Resulta interesante considerar los problemas que aparecen en el medio subacuático. La turbidez hace que trabajar con cámaras sea una tarea compleja. Los algoritmos de visión clásicos de detección, seguimiento, reconstrucción en 3D y controladores guiados por visión sufren las consecuencias y deben ser adaptados. Además, esta degradación de la imagen es difícil de predecir ya que depende de las partículas en el medio, la iluminación ambiental y la profundidad, siendo los dos primeros factores variables. Por ello, los I-AUV deben estar preparados puesto que las condiciones de iluminación pueden cambiar en cualquier momento durante una misión de intervención. La interacción con el medio puede, en muchos casos, producir un cambio en la visibilidad.

Para entender los problemas del procesamiento de la imagen bajo el agua es necesario considerar la física de la propagación de la luz en el agua [Duntley, 1963]. Las propiedades físicas del medio causan una degradación en la imagen que no está presente en el aire. Estas imágenes submarinas se caracterizan por una deficiente visibilidad debido a que la luz se atenúa y dispersa de forma exponencial al atravesar el agua, resultando en imágenes con poco contraste y borrosas. Estos dos fenómenos limitan la visibilidad a 20 metros en agua clara y 5 metros o menos en aguas turbias. Este hecho dificulta la detección de objetos a manipular pese a la cercanía del robot al objeto.

La atenuación de la luz es causada principalmente por dos fenómenos: absorción, que reduce la energía de la luz, y dispersión, que cambia la dirección de las ondas de luz. La absorción y dispersión de la luz influyen en los sistemas de visión subacuáticos. La absorción reduce la cantidad de luz cuando el robot se encuentra a mayor profundidad o se aleja de la cámara, y los colores se pierden secuencialmente dependiendo de su longitud de onda. En la Figura 3.1, se muestra como el color azul es el que es capaz de viajar más tiempo por el agua debido a su longitud de onda más corta y, por tanto, más energética. El resultado es que en las imágenes submarinas predomina el color azul y, además, que la luz infrarroja no es capaz de

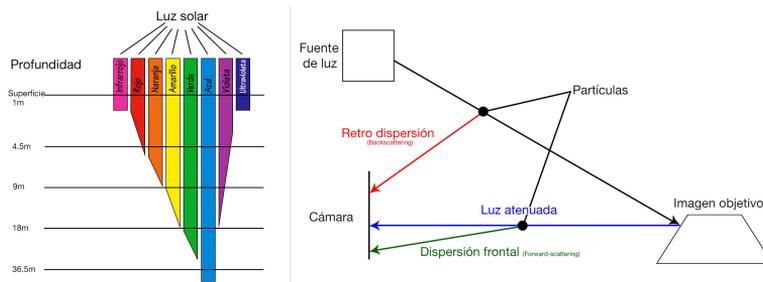


Figura 3.1: Efectos físicos que deterioran las imágenes subacuáticas: absorción (izquierda), dispersión y atenuación (derecha).

viajar distancias que permitan su uso en el agua.

La dispersión, como se muestra en la Figura 3.1, cambia la dirección de la luz. Se diferencian dos efectos diferentes. El *forward scattering* o dispersión frontal consiste en que la luz que se dirige de un objeto hacia la cámara se ve desviada produciendo emborronamiento en las características de la imagen. Por otro lado, en el *backward scattering* o retrodispersión, parte de la luz es reflejada por el agua hacia la cámara antes de alcanzar los objetos de la escena. Este método produce un velo que se superpone a la imagen limitando el contraste de las imágenes y mostrando borrosos los objetos a manipular.

Estos dos fenómenos son causados tanto por el agua como por otros componentes orgánicos disueltos en el agua y partículas flotantes. Su tipología y concentración es muy variable. Estas partículas, que pueden ser visibles en la cámara, absorben o desvían longitudes de onda diferentes dependiendo de sus características, causando los efectos mencionados.

La visibilidad se puede aumentar mediante iluminación artificial, pero los efectos de la dispersión (concretamente el *backward scattering*) son entonces más notables, sobretodo cuando la iluminación parte de cerca de la cámara. A parte de los problemas descritos, la iluminación suele no ser uniforme, produciendo más brillo en la parte central de la imagen y perdiendo iluminación progresivamente hacia los bordes, efecto conocido como viñeteo o *vignetting*. Además, se trata de fenómenos especialmente perjudiciales para la reconstrucción 3D con cámaras estéreo, pues estas se basan en la reconstrucción de la escena a través de características distinguibles en dos imágenes y la dispersión afecta de forma independiente a las dos cámaras. Este hecho produce diferencias notables en las imágenes y dificulta la búsqueda de correspondencias. Así pues, las imágenes subacuáticas se ven afectadas por la visibilidad limitada, bajo contraste, iluminación no uniforme, emborronamiento, partículas iluminadas, color azulado de la imagen y ruido. Por ello los algoritmos de visión se ven limitados por estos problemas, que deben tenerse en cuenta a la hora de realizar la detección de objetos y posterior manipulación. En este trabajo se utilizan métodos de estimación de la geometría de los objetos que tienen diferentes características. Una de ellas es la sensibilidad al ruido, siendo en ocasiones mejor un método tolerante al ruido que otro más flexible.

3.3. Sensores 3D comerciales

Pese a que existen multitud de alternativas orientadas a la robótica móvil a un coste excelente, actualmente se presentan muy pocas opciones en el mercado estrictamente submarino. En la robótica se pueden distinguir sensores estéreo, sensores de profundidad (también llamados RGB-D si disponen de una cámara con la que registrar imágenes), sensores de *tiempo de vuelo* o Time of Flight (ToF), y sensores de *detección por luz y distancia* o Light Detection And Ranging (LiDAR).

En el entorno submarino, debido a las restricciones presentadas, las mejores alternativas con un coste razonable son las cámaras estéreo y los sensores basados en luz láser visible, descartando de esta manera los sensores RGB-D basados en infrarrojos [Zennaro et al., 2015]. Además, existen otras alternativas como el uso de sonar o los escáneres *multibeam*, pero se trata de tecnologías más adecuadas para una escala mayor [Massot-Campos y Oliver-Codina, 2015], siendo su resolución insuficiente para la aplicación a la manipulación.

En el mercado podemos encontrar modelos como la cámara estéreo Teledyne Bowtech 3D-HD ¹. Sin embargo, debido a las certificaciones de profundidad que cumplen, resultan prohibitivas para su uso en investigación. Además, no se ofrecen interfaces sencillas para su integración en un sistema robótico, ya que se utilizan tecnologías propietarias y requieren módulos de procesamiento externos adicionales que no pueden ser integrados con facilidad en el interior de los AUV. Por otro lado, existen sistemas para buceadores que utilizan cámaras que graban para después procesar el vídeo, siendo también soluciones a medida difíciles de integrar en un sistema autónomo [Schmidt y Rzhanov, 2012][Williams et al., 2014].

Existen múltiples dispositivos capaces de proyectar patrones láser bajo el agua como el proyector láser lineal Trittech SeaStripe MKIII. No obstante, que conozca el autor, no existen alternativas comerciales que realicen la reconstrucción 3D del entorno a partir del patrón láser proyectado. De esta manera, la integración de los componentes, la segmentación de los patrones y la consiguiente reconstrucción deben ser realizados por el sistema que lo utiliza [Prats et al., 2012a][Palomer et al., 2018][Massot-Campos y Oliver-Codina, 2014].

Mientras que para el uso de ROV no son estrictamente necesarias este tipo de tecnologías, estos sensores son esenciales en la robótica submarina, donde la alternativa más utilizada es el encapsulado de una cámara estéreo [Rizzini et al., 2017]. En esta tesis se ha sondeado el mercado y puesto a prueba diferentes alternativas como la Videre Stereo Camera, Point Grey Bumblebee 2, Stereolabs ZED, Code Laboratories DUO M o el uso de dos cámaras IDS uEyE independientes. Para utilizar estos dispositivos es necesario encapsularlos adecuadamente [Chavez et al., 2017] e integrarlos con el resto del sistema. Por ello, son factores a tener en cuenta el tipo de conexión que presentan (Ethernet, USB2.0, USB3.0, FireWire...), sus dimensiones, su distancia entre centros focales o *baseline*, su distancia de trabajo y su coste.

¹Características de Teledyne Bowtech 3D-HD: <http://seatronics-group.com/equipment-sales/video/subsea-cameras/teledyne-bowtech-3d-sd-high-res-stereoscopic-camera/>



Figura 3.2: Imágenes tomadas con las cámaras DUO M (izquierda), Videre (centro) y Bumblebee2 (derecha).



Figura 3.3: Imágenes de la cámara virtual en diferentes escenas.

3.4. Reconstrucción estéreo

La fuente de información de 3D más utilizada por los I-AUV resulta el uso de cámaras estéreo. En base a los modelos disponibles mostrados en la sección previa, se describen las cámaras utilizadas en esta tesis así como sus características principales:

- Cámara estéreo Videre STH-DCAM: se trata de una cámara FireWire en blanco y negro compuesta por dos sensores Charge-Coupled Device (CCD) con una resolución de 640 x 480. Aunque es una resolución baja, resulta útil para la reconstrucción 3D cuyo computo es temporalmente costoso. Por otro lado, el sensor monocromático tiene una mayor sensibilidad y resulta ideal para la reconstrucción estéreo. Además, debido a la degradación del color bajo el agua, es una información menos relevante. El *baseline* es de 9cm., lo que hace que la distancia en la que el error es menor, aproximadamente a 1 m., sea ideal para la manipulación. Se utiliza en los experimentos en laboratorio del IRS Lab.
- Cámara estéreo Code Laboratories DUO M: se trata de una cámara estéreo monocolor ultra compacta con interfaz USB 2.0 y un *baseline* de 3cm. Sus pequeñas dimensiones (52x25x12mm.) la hacen ideal teóricamente para este entorno ya que es más sencilla de encapsular. Sin embargo, la distancia entre lentes hace que la precisión de esta cámara no sea la mejor. Por otro lado dispone de disparador global (ideal para sistemas en continuo movimiento), velocidad de captura variable y un ángulo de visión amplio. Se utiliza en los experimentos en laboratorio del IRS Lab.
- Cámara estéreo PointGrey Bumblebee 2: se trata de una cámara estéreo que dispone de dos sensores a color Sony ICX204 CCD con disparador global,



Figura 3.4: Reconstrucción 3D de las escenas en UWSim mediante visión estéreo.

velocidad de captura variable y una resolución de 1024 x 768. El *baseline* de estas cámaras es de 12cm., por lo que resulta adecuada para realizar manipulación. Está disponible únicamente en los experimentos de integración en la piscina del CIRS y en el mar.

- Cámara estéreo virtual: se trata de dos cámaras a color simuladas con el UWSim mediante las cuales se realiza la reconstrucción 3D. El *baseline* es de 12cm. y la resolución de 1280 x 720. Se utiliza en los experimentos simulados.

Como se ve, estos sensores tienen un *baseline* o separación entre los centros focales fija, mientras que otras soluciones comerciales permiten variar esta distancia. En la Figura 3.2 se muestra una captura con cada uno de los sensores reales utilizados. Por otro lado, la Figura 3.3 muestra imágenes virtuales generadas con el simulador. Con las cámaras simuladas se obtiene una reconstrucción 3D casi ideal (ver Figura 3.4). No obstante, como se verá en secciones posteriores, con el objetivo obtener una reconstrucción 3D adecuada, el uso de sensores reales debe estar acompañado de técnicas de filtrado y métodos tolerantes al ruido. Para poder utilizar estas cámaras en el sistema, debido a la arquitectura *software* elegida, se requiere el desarrollo de controladores adecuados para su control, así como un paquete de ROS que permita su interacción con el resto de componentes del sistema.

Por otro lado, las cámaras deben ser correctamente aisladas para su uso en el agua por lo que requieren de un encapsulado estanco ². Este encapsulado consiste en una carcasa y un elemento de metacrilato o vidrio que se interpone entre la lente y la escena. Este nuevo elemento óptico modifica la distancia focal de la cámara y normalmente introduce una pequeña degradación en la imagen, ya que la luz viaja a través de medios adicionales como son el agua, el propio metacrilato o vidrio, y el aire situado entre el sensor y la carcasa. En la Figura 3.5 se muestra el encapsulado de las cámaras utilizadas en este trabajo en los experimentos en el IRS Lab.

En general, el proceso de calibración de una cámara es un proceso necesario y crítico. Además, en el agua resulta más importante debido a que el índice de refracción del agua es diferente dependiendo de la salinidad del agua y el uso de carcasas estancas para las cámaras modifican los parámetros intrínsecos de las cámaras. Por esto, se hace necesario la recalibración de las cámaras con frecuencia. En este trabajo se ha utilizado el sistema de calibración disponible en ROS en el

²Sistema de visión estéreo por Ethernet de Sexton: <http://www.thesextonco.com/product/stereo-underwater-vision-system>

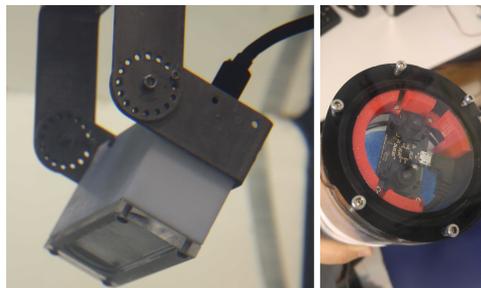


Figura 3.5: Encapsulado de las cámaras estéreo Videre (izquierda) y Duo 3D (derecha) utilizadas en el laboratorio.

paquete *camera_calibration* [Zhang, 2000] utilizando un dámara de calibración. Este sistema se utiliza para obtener los parámetros intrínsecos de las cámaras para poder corregir la distorsión de cada una de las imágenes. Simultáneamente, se obtiene la posición exacta entre las dos lentes.

3.4.1. Algoritmos de correspondencia

Una vez se ha obtenido un par de imágenes se debe realizar la reconstrucción 3D de la escena mediante triangulación. Este proceso puede ser realizado por *hardware* como en muchos sensores RGB-D. Sin embargo, en este caso se realiza mediante *software*. A la hora de realizar esta reconstrucción el paso más importante es proceso de *matching* o búsqueda de correspondencias, en el cuál se determinan los puntos de la imagen que corresponden con otros puntos de la otra imagen para posteriormente realizar la reconstrucción. En base a la distancia entre los puntos en coordenadas de la imagen y los parámetros de la cámara, se obtienen las coordenadas 3D del punto correspondiente.

El proceso de *matching* se puede realizar de forma local, semi-global o global [Scharstein y Szeliski, 2002]. Los algoritmos de *matching* local tratan de encontrar para cada punto de la imagen un punto que se le corresponda en la segunda imagen utilizando la misma estrategia en cada punto de la imagen. Los algoritmos que se utilizan por defecto se encuentran en el paquete de reconstrucción de ROS (*stereo_image_proc*). En primer lugar, el método *Block Matching*, implementado en OpenCV [Bradski, 2000], consiste en comparar la intensidad de cada píxel de la imagen con las intensidades de una ventana de píxeles en la otra imagen. Esta estrategia es muy rápida pero los resultados no son los mejores que se pueden obtener ya que la imagen resultante suele tener huecos debido a los píxeles de la imagen que no encuentran correspondencia. Por otro lado, el método *Semi-global Block Matching* [Hirschmuller, 2008], también implementado en OpenCV, resulta un método más avanzado en el estado del arte. Consiste en realizar un proceso de optimización teniendo en cuenta los píxeles adyacentes para obtener una reconstrucción más completa, aunque de forma significativamente más lenta. El resultado es una reconstrucción 3D más densa. Finalmente, las estrategias globales utilizan ambas imágenes en un proceso de optimización para obtener un mapa de disparidad todavía más denso y refinado [Geiger et al., 2010], tratando de encontrar este mapa mediante interpolación. Estos métodos son comparativamente mucho más

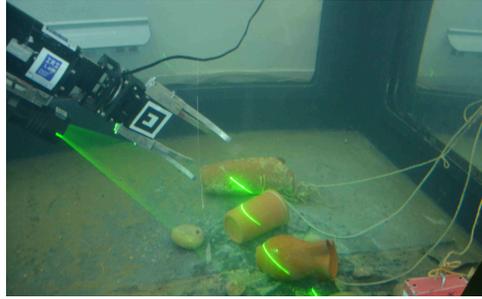


Figura 3.6: La reconstrucción láser utiliza una cámara y un proyector láser lineal.

costosos, por lo que su uso en tiempo de ejecución no es viable.

3.5. Reconstrucción mediante segmentación láser

El *framework* desarrollado en este trabajo puede hacer uso tanto de la visión estéreo como de la reconstrucción utilizando láser. Los sensores basados en infrarrojos son el estándar en la robótica actual. Como se ha indicado en la introducción, desafortunadamente la luz infrarroja no puede ser utilizada debido a que es absorbida en centímetros. No obstante, han aparecido en los últimos años alternativas que utilizan patrones luz láser verde [Prats et al., 2012a]. Esta luz sufre una menor absorción y es más sencilla de generar y de distinguir que la luz azul por las propiedades descritas en la introducción de este capítulo. Existen otros trabajos que utilizan luz estructurada en el medio acuático, ya sea creando patrones de luz multilínea [Massot-Campos y Oliver-Codina, 2014] o utilizando espejos y cámaras de alta velocidad [Palomer et al., 2018]. Esta última aproximación presenta unos resultados muy prometedores y se ha utilizado como fuente de reconstrucción estéreo en los experimentos realizados en la piscina en el contexto del proyecto MERBOTS.

Durante la realización de experimentos en el laboratorio del IRS Lab también se ha utilizado como origen de datos la reconstrucción láser. Para llevarla a cabo ello se utiliza el proyector láser lineal Tritech SeaStripe MKIII [Prats et al., 2012a]. La detección del láser puede ser realizada con la cámara Bowtech DIVECAM-550C-AL o mediante una de las cámaras del par estéreo (ver figuras 3.6 y 3.7). Para realizar la reconstrucción se proyecta el patrón láser y después se segmentan las imágenes de la cámara utilizando un detector basado en cambios de intensidad. Este detector compara cada punto de la escena con una referencia de color del láser obtenida previamente [Forest et al., 2004]. Después, se obtienen los puntos en 3D utilizando los parámetros de la cámara, la posición del proyector láser y el modelo del plano que proyecta. En [Peñalver et al., 2017] se muestra el uso de este método con la cámara acoplada al brazo. Esta técnica obtiene resultados diferentes, ya que el punto de vista de la cámara varía con el movimiento del brazo, generando reconstrucciones de los objetos más completas a una distancia configurable. Por otro lado, la precisión de la reconstrucción es inferior con respecto a la alternativa utilizando una cámara fija debido a que la precisión del brazo es limitada.

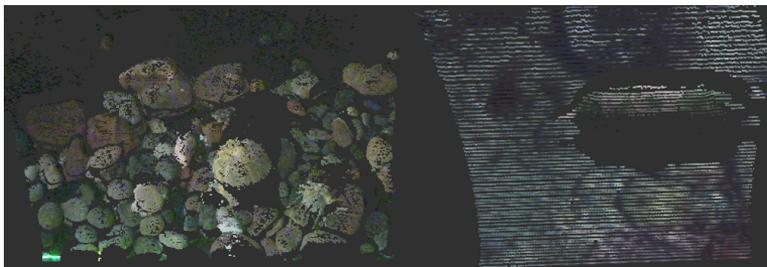


Figura 3.7: Reconstrucción láser con imágenes reales (izquierda) y simuladas (derecha).

En comparación a la reconstrucción estéreo, este tipo de métodos son más lentos puesto que requieren el movimiento físico del brazo durante unos segundos, mientras que la captura de imágenes estéreo depende del tipo de exposición. Por tanto, el vehículo tiene que permanecer quieto durante el proceso o tener en cuenta el movimiento del sistema con respecto al entorno [Prats et al., 2012a]. Otras alternativas que proponen un patrón que cubre parte de la imagen o el desplazamiento rápido del láser con un espejo no sufren estos problemas, aunque la detección del láser es más compleja en estos casos.

Respecto a la citada reconstrucción estéreo, los métodos basados en láser pueden utilizarse con una menor iluminación, incluso en la oscuridad, y son capaces de reconstruir objetos sin textura [Pérez et al., 2015]. Aunque son capaces de solucionar estos problemas que sí afectan a la visión estéreo, la presencia de partículas en el medio produce *backward scattering* de la luz proyectada. Esta luz reflejada llega a la cámara y es segmentada, de forma que aparece en la reconstrucción como ruido. Por este motivo, no se han utilizado sensores basados en láser en los experimentos en el mar.

3.6. Mejora de la reconstrucción mediante filtrado

Con el objetivo de mejorar la reconstrucción obtenida mediante las técnicas anteriores, se proponen una serie de medidas que ayuden a obtener resultados más precisos. Con sensores de profundidad de calidad este paso podría no ser tan necesario, sin embargo, como se muestra en la Figura 3.2, los sensores utilizados en el medio subacuático presentan una cantidad de ruido significativa. En esta sección presentan los algoritmos utilizados para preparar la información para su posterior utilización, mostrando el estado del arte en procesamiento de nubes de puntos. Como se ha explicado en el Capítulo 2, en este trabajo se ha utilizado PCL [Rusu y Cousins, 2011] para la manipulación de nubes puntos n -dimensionales, ya que incluye los tipos de datos y algoritmos adecuados para su tratamiento.

En primer lugar, se propone el uso de un filtro temporal en el que se utilizan diversas capturas en el tiempo para componer una nube de puntos con menos ruido. Así se, se realiza una integración de las nubes de puntos en varios instantes de tiempo, teniendo en cuenta el posible desplazamiento de la cámara. Aprovechando este filtrado, se ha propuesto un escenario con cambios de iluminación utilizando

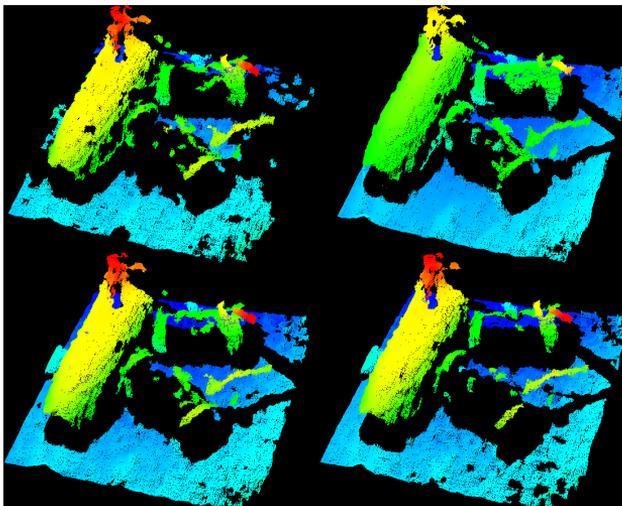


Figura 3.8: Filtro temporal de nubes de puntos con diferente iluminación. Resultado del filtrado (arriba, derecha).

diferentes niveles de iluminación de forma que en cada ocasión se obtienen una determinada cantidad de puntos. En la Figura 3.8 se muestran nubes de puntos obtenidos con los diferentes niveles de iluminación y la reconstrucción resultante de integrar diversas nubes de puntos en el tiempo. Esta solución se puede implementar utilizando luz artificial en el vehículo, permitiendo mejorar la reconstrucción, ya que dependiendo de la reflectancia de las superficies se pueden reconstruir mejor o peor. No obstante, el uso de luz artificial aumenta el citado *backward scattering*, introduciendo ruido. En este caso, se observa una mejora en la reconstrucción del objeto, obteniendo unas superficies más suaves y que representan mejor el objeto.

A continuación, se utiliza un filtro de paso que elimina los puntos cuyas coordenadas estén en valores fuera de los límites de la escena. Estos valores aparecen por errores de reconstrucción y búsqueda de correspondencias. Una vez se han eliminado estos puntos, que son claramente erróneos, el uso de filtros de tipo mediana y bilateral [Tomasi y Manduchi, 1998] permiten eliminar ruido que afecta a la nube de puntos resultante a costa de una disminución del detalle. El filtrado bilateral suaviza imágenes tratando de preservar bordes de la imagen, gracias a una combinación no lineal de puntos cercanos, de forma que la imagen final pierde menos detalle que con el filtro mediana.

Algunas de las irregularidades encontradas en las nubes de puntos se pueden resolver utilizando un análisis estadístico del vecindario de un punto, eliminando los puntos que no cumplen un determinado criterio. Para cada punto se calcula la distancia a sus vecinos. Asumiendo que la distribución de las distancias es gaussiana, se eliminan todos los puntos cuyas distancias medias estén fuera de las distancias definidas por la media y desviación estándar de la distribución. Este filtro se denomina *Statistical Outlier Removal* [Rusu et al., 2008]. Como se puede ver en la Figura 3.9, a costa de obtener puntos más fiables y robustos se eliminan puntos que pueden ser correctos. Por el contrario, el filtro bilateral obtiene puntos adicionales al tiempo que se modifican los puntos existentes en función de sus vecinos. Por ello,

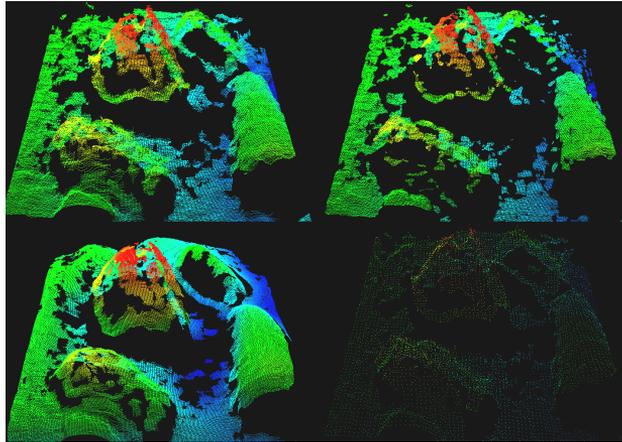


Figura 3.9: Filtrado de nubes de puntos: nube original (arriba, izquierda), *Statistical Outlier Removal* (arriba, derecha), filtro bilateral (abajo, izquierda) y reescalado *voxel grid* (abajo, derecha).

las aplicaciones de ambos filtros son diferentes.

Por otro lado, se utiliza un filtro *voxel grid* para subdividir el espacio en vóxeles de un determinado tamaño, de forma que se reduce la resolución al sustituir todos los puntos dentro del vóxel por su centroide. Este proceso se conoce como *downsampling* y también se hace referencia a él en este trabajo como reescalado. Este método aproxima mejor la nube puntos original que simplemente utilizar el centro del vóxel como representación, aunque es más lento. Este filtro se utiliza para reducir la cantidad de puntos de forma significativa, ya que este filtro preserva información suficiente como para obtener la estimación de los objetos a la vez que se reduce de forma drástica el tiempo necesario para procesar las nubes de puntos (ver Figura 3.9).

Alternativamente, se puede reducir el tamaño de una nube de puntos muestreando un número determinado de puntos en el espacio de las normales en un proceso denominado *Normal Space Sampling* [Rusinkiewicz y Levoy, 2001]. El resultado de este método es un subconjunto de puntos para los cuales la distribución de las normales es lo más uniforme posible con respecto a la nube de puntos original. De esta forma se obtiene un subconjunto que representa lo mejor posible las superficies originales. Esta aproximación es especialmente adecuada para técnicas de registro de nubes de puntos.

Cabe destacar que estos algoritmos se benefician del uso de nubes de puntos organizadas, es decir, nubes de puntos que están organizadas en una matriz como las imágenes originales, dividiéndose en filas y columnas. Los datos que provienen de cámaras estéreo o de ToF presentan este formato. En este trabajo se trata en todo momento de preservar la organización de las nubes de puntos. La ventaja principal de una nube de puntos organizada es la búsqueda de vecinos de forma rápida. Muchos de los algoritmos presentes en PCL requieren búsquedas de vecinos por lo que mantener esta organización supone una mayor eficiencia.

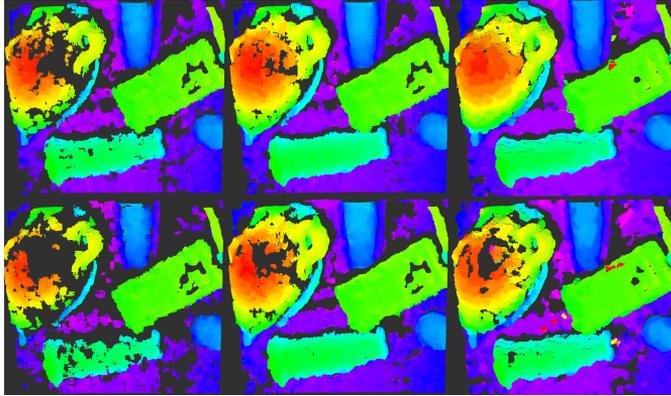


Figura 3.10: Nube de puntos obtenida con la cámara estéreo Videre. Sensibilidad ISO 100 (arriba) e ISO 400 (abajo). De izquierda a derecha se usa *Block Matching*, *Block Matching* con acumulación y *Semi-Global Block Matching*.

3.7. Comparativa de nubes de puntos

En la sección anterior se han mostrado las diferentes posibilidades a la hora de mejorar la calidad de una reconstrucción 3D. En las figuras 3.8 y 3.9 se puede ver cómo mejorar estas reconstrucciones. Sin embargo, es muy importante obtener una buena reconstrucción inicial. En [Pérez et al., 2015] se realiza una comparativa de la reconstrucción utilizando visión estéreo y segmentación láser mientras se varían las condiciones de iluminación. Se observa también el uso del simulador para realizar *benchmarking* con *Hardware-in-the-loop*, ya que se utiliza el simulador para evaluar un sensor virtual y el sensor real de la misma manera.

En la Figura 3.10 se muestra una comparativa cualitativa de la reconstrucción con la cámara Videre. En ella se muestra el uso de esta cámara con diferentes sensibilidades disponibles (ISO 100 e ISO 400). Con una sensibilidad ISO 100 los resultados son visiblemente mejores que con ISO 400. Mientras que la ISO más alta debería proporcionar mejores imágenes en un entorno con menos iluminación, en la práctica el aumento de la sensibilidad introduce ruido en la captura del sensor. Además, este ruido se da en ambas imágenes haciendo que la cantidad de ruido en la nube de puntos final sea la suma del ruido de las dos imágenes, impidiendo realizar la reconstrucción de cualquier punto en el que su punto correspondiente de la otra imagen se pierda por el ruido. Es decir, el ruido de las imágenes 2D no se traduce directamente en ruido en la nube de puntos si no que se genera un menor número de puntos, que acaban teniendo una precisión similar, puesto que son el resultado de la correspondencia entre pares de puntos libres de ruido.

En cuanto a los métodos utilizados, en la Figura 3.10 se muestra en primer lugar la reconstrucción utilizando *Block Matching* y después la misma reconstrucción con un filtro temporal de 1 segundo. En este caso, se aprecia una mejoría en la imagen puesto que los puntos que no habían sido encontrados previamente en una sola imagen a causa del ruido se pueden encontrar con este método. Sin embargo, cabe

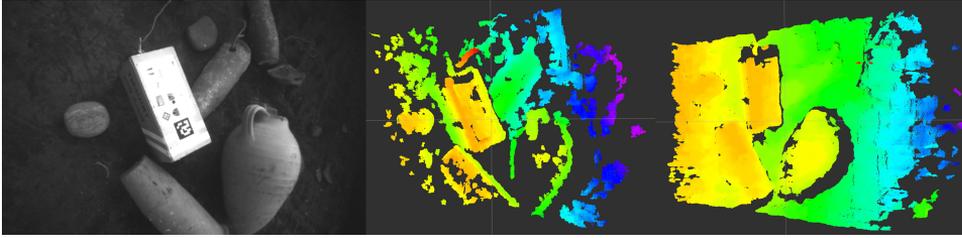


Figura 3.11: Nube de puntos obtenida con la cámara estéreo DUO M. Imagen de la cámara (izquierda), reconstrucción con *Block Matching* (centro) y con *Semi-Global Block Matching* (derecha).

destacar que esta captura se ha realizado con la cámara estática, que es el escenario ideal para este método. Finalmente, se muestra el resultado utilizando *Semi-Global Block Matching* [Hirschmuller, 2008]. Como se ve esta técnica presenta muchos menos *huecos* que las anteriores, pero en el código de color, que representa el eje Z, se aprecia como las superficies no son totalmente uniformes, por lo que la precisión es menor. En la Figura 3.12 se muestra una reconstrucción de los objetos utilizados en este trabajo utilizando *Block Matching* local. Como se puede ver, el tipo de superficie del objeto, el color, la iluminación y la textura del objeto tienen un impacto notable en la calidad de su reconstrucción.

Además de la cámara estéreo Videre, se ha utilizado en el laboratorio la cámara DUO M. En la Figura 3.11 se muestra la reconstrucción 3D utilizando dicha cámara. Se ha utilizado tanto el algoritmo de *matching* local como semi-global. La reconstrucción obtenida con correspondencias locales presenta una densidad mucho mayor, mientras que el uso del método semi-global ofrece un resultado aparentemente adecuado pero mucho menos preciso que la reconstrucción estéreo. Como se verá en el siguiente capítulo, este sensor no tiene calidad suficiente para realizar una estimación robusta.



Figura 3.12: Reconstrucción estéreo de los diferentes objetos presentados.

3.8. Registro de nubes de puntos

Una vez realizada la comparativa se puede ver que cada método tiene unas ventajas e inconvenientes diferentes. Por ello, se ha desarrollado un sistema que toma diversas imágenes capturadas con uno o más sensores para obtener una representación en 3D del entorno global. Este sistema se basa en el uso de un método conocido como registro de nube de puntos o *registration* [Holz et al., 2015] para unir varias nubes de puntos y obtener un mapa global de la escena. Este mapa se puede utilizar para disponer de una reconstrucción de un área mayor o para obtener diferentes puntos de vista de un objeto que aporten información adicional.

Para realizar este registro se ha recurrido al uso de marcadores. Pese a que es una simplificación del problema no asumible en muchos entornos, esta asunción es válida en este caso en el que los entornos arqueológicos normalmente disponen de marcas con mediciones en su superficie. También se ha utilizado como alternativa el uso de características o *features* como Fast Point Feature Histograms (FPFH) [Rusu et al., 2009a]. Estos dos métodos se utilizan para obtener una estimación inicial del alineamiento entre pares de nubes de puntos. A continuación, se utiliza Iterative Closest Point (ICP) [Holz et al., 2015] para perfeccionar la transformación entre las dos nubes. Este método es el más utilizado [Gelfand et al., 2003] y estudiado [Besl y McKay, 1992], por lo que parece natural utilizarlo en este contexto en el cual se necesita de algoritmos robustos.

El método basado en ICP se utiliza para obtener una transformación rígida (con vector de traslación T y matriz de rotación R) que alinea dos nubes de puntos a través de un refinamiento continuo. Se utiliza para reconstruir superficies 2D y 3D a partir de diferentes nubes de puntos, para localizar robots y en planificación de trayectorias. PCL implementa diferentes versiones de este algoritmo. Este procedimiento se repite dos a dos para cada nueva nube de puntos obtenida, de forma que se va construyendo un mapa global. Con una reconstrucción por pares cabe esperar que se introduzca progresivamente un error en el mapa. Es por esto que cada varias nubes de puntos se corrige la transformación entre el mapa y la nube de puntos actual.

3.9. Calibración visual del manipulador

Debido a que el proceso de calibración basado en los límites de las articulaciones es impreciso, se realiza un proceso de calibración visual con la que es posible conocer la posición del efector final con una precisión mayor (ver Figura 3.13). Este método consiste en la detección de un marcador mediante una herramienta de detección llamada ARUCO [Romero-Ramirez et al., 2018]. Estos marcadores son conocidos como *AR Markers*, marcadores para Realidad Aumentada o Augmented Reality (AR). Son ampliamente utilizados en aplicaciones interactivas, navegación, robótica y como referencia o *ground truth*. Alternativamente, en el laboratorio IRS Lab se han utilizado LEDs, de forma que es posible conocer la posición del manipulador en entornos de baja luminosidad [Peñalver et al., 2015].

Después de obtener la posición y orientación del efector final mediante marcadores (x_{marker}), se realiza la IK para obtener la posición estimada de las articulaciones en cada momento (q_{marker}). Esta información se utiliza para actualizar la posición

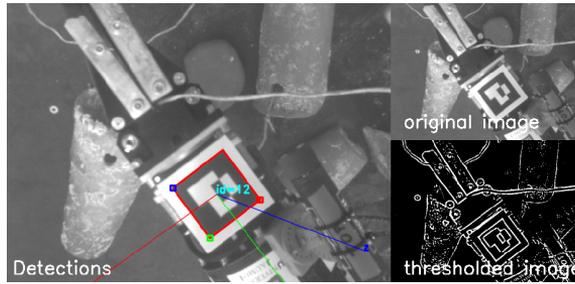


Figura 3.13: Detección de marcadores para determinar la posición del manipulador.

de las articulaciones estimada cada vez que se obtiene una detección. Cuando no se pueden obtener detecciones, los codificadores relativos siguen generando una posición en base a detecciones previas. Así, se obtiene la posición articular del robot (q_{fixed}) de manera más precisa. En el Capítulo 6 se muestra cómo se integra esta información con el controlador del robot.

Por otro lado, el uso de la detección de marcadores es muy útil en otros contextos. En primer lugar, se utiliza para detectar un objeto y generar una posición de referencia con la que poder comparar otros algoritmos. Este método ha sido utilizado en los experimentos de integración para evaluar las diferentes estimaciones de la vasija. En segundo lugar, estos marcadores también se utilizan en los sistemas de navegación del GIRONA500 para obtener un posicionamiento del robot de calidad. Pese a que este acercamiento no es válido en escenarios de navegación en mar abierto, como ya se ha explicado, esta metodología es aceptable en entornos arqueológicos donde los expertos colocan frecuentemente marcas visuales como sistema de referencia al tomar medidas y fotografías.

Estimación de la geometría de la escena

En este capítulo se describen las técnicas de visión utilizadas para obtener las características geométricas de la escena, compuesta por el fondo y uno o más objetos, con el objetivo de realizar posteriormente la manipulación. En primer lugar se introduce la motivación para el uso de estas técnicas, la metodología aplicada en este trabajo y el enfoque utilizado en trabajos realizados previamente en el medio subacuático. Después, se describe en detalle las técnicas para segmentar el fondo de la escena. El grueso del capítulo es la descripción de los métodos de estimación de la geometría de los objetos: basados en RANSAC, momentos de inercia y en SQ. Además, se explica el algoritmo de cálculo de simetrías y el uso de los métodos de agrupamiento, herramientas utilizadas para llevar a cabo la estimación.

Tras la descripción de las técnicas aplicadas, se detallan los experimentos realizados y resultados obtenidos. Se han evaluado estimaciones con conjuntos de datos obtenidos en el laboratorio, en la piscina del CIRS y en el mar. En los experimentos realizados en laboratorio, además, los resultados incluyen escenas con diversos objetos aislados, así como con múltiples objetos diferentes.

4.1. Motivación

El reconocimiento de objetos orientado a la manipulación es una tarea compleja con multitud de aproximaciones en la robótica actual [Bohg et al., 2014]. El desafío es mayor cuando no es posible utilizar un sensor adecuado, no se dispone de múltiples vistas de un objeto, aparecen oclusiones o el entorno es desconocido y desestructurado. La mayoría de estos condicionantes se dan en el medio acuático. Por este motivo, en este entorno se ha optado por una metodología que trate de encontrar la forma de manipular un objeto familiar o cuya forma aproximada se conoce pero no está disponible a priori un conocimiento perfecto de sus dimensiones o forma exacta. Este enfoque encaja en las necesidades de la arqueología submarina, ya que las dimensiones de los objetos pueden no ser conocidas de forma exacta hasta que se recupera el objeto. No obstante, se tiene un conocimiento sobre el tipo de objetos que se espera encontrar.

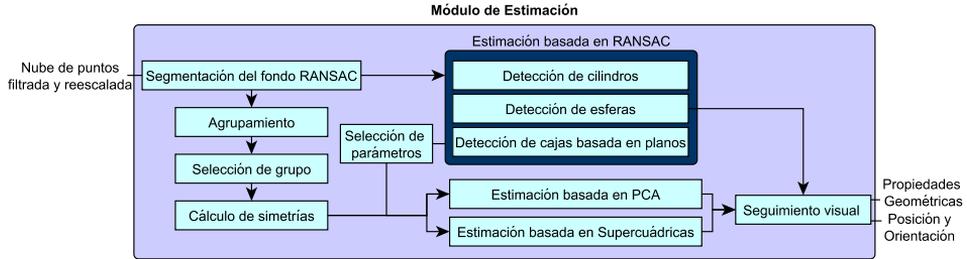


Figura 4.1: Componentes del módulo de Segmentación y Estimación.

El enfoque adoptado consiste en utilizar y ofrecer al usuario potencial del sistema diferentes métodos de estimación para determinar cómo manipular el objeto de interés de la forma más segura posible. Resulta interesante ver la tendencia especialmente en la robótica submarina de combinar la inteligencia humana con la necesidad de ofrecer funciones autónomas al usuario del sistema, que puede evitar así tomar el control directo del robot, ya que ésta es una tarea que ha demostrado ser difícil y peligrosa para la integridad del robot y los objetivos de la intervención.

4.2. Metodología de Segmentación y Estimación

Para obtener una posición de agarre correcta, la posición, orientación y forma del objeto debe ser obtenida primero. Inicialmente la escena se captura utilizando uno de los métodos descritos en el Capítulo 3. Después esta nube de puntos se procesa para reducir la cantidad de ruido y limitar su resolución. En la robótica de servicio en muchos de los trabajos en manipulación se asume que los objetos están dispuestos sobre una superficie plana como una mesa [Bohg et al., 2011]. En el presente trabajo, se asume que la superficie del suelo es aproximadamente plana. Sin embargo, en la mayoría de métodos utilizados no es un requisito esencial sino deseable. Por otro lado, en la concepción de los algoritmos se asume que hay un solo objeto en la escena, aunque de nuevo no todos los métodos presentados comparten esta limitación.

En la Figura 4.1 se pueden ver los diferentes componentes del módulo de Estimación. En este capítulo se propone el uso de técnicas de visión para segmentar objetos y calcular sus propiedades geométricas orientadas a la manipulación. Después de la obtención de la representación del mundo en forma de nube de puntos, ésta se segmenta para eliminar el fondo y obtener un modelo geométrico del objeto que aproxime su forma. Además, el sistema de visión está pensado para realizar un seguimiento del objeto durante la aproximación y manipulación del mismo. Los diferentes métodos desarrollados pueden ser aplicados a múltiples objetos de forma que es el usuario el que selecciona el método a utilizar.

Por otro lado, la parametrización es muy importante en este tipo de algoritmos. En cada uno de ellos se describen los parámetros que controlan cada algoritmo, ya que determinar sus valores adecuados es un paso necesario. Sin embargo, la obtención de los parámetros en este caso se ha realizado de forma experimental.

4.2.1. Trabajos previos en manipulación subacuática

En el contexto del proyecto RAUVI [Prats et al., 2012d], la manipulación del objeto se llevó a cabo utilizando técnicas de estimación basadas en imágenes. Como se ha explicado en la introducción, en primera instancia se procesaba la posición de agarre sin determinar la posición del objeto. Era el usuario quien indicaba dos puntos en una imagen 2D para que se proyectaran y convirtieran en una posición en 3D para el robot. Para ello, se obtenía la intersección entre la recta formada por estos puntos y la nube de puntos obtenida mediante reconstrucción láser.

En el proyecto TRIDENT [Sanz et al., 2012], cuyos objetivos eran más ambiciosos en otras áreas, se realizaba el seguimiento de la caja negra utilizando únicamente una plantilla en 2D obtenida previamente con la ayuda del usuario. Conociendo las medidas exactas de la caja, estas podían ser utilizadas para obtener la posición del objeto en 3D y la postura de agarre. En [Pérez et al., 2015] el usuario determina la posición de un modelo de caja negra conocido pulsando en las esquinas del objeto en una imagen en 2D.

Por otro lado, en el proyecto TRITON [Peñalver et al., 2015] se debía manipular un panel submarino cuyo modelo y posición eran conocidos a priori de forma aproximada. Así pues, en todas estas aproximaciones se realizó una simplificación de la caracterización del agarre, así como la necesidad recurrente de confiar en el criterio del usuario, con el objetivo de lograr manipular objetos en un entorno tan complejo.

Finalmente, en el proyecto MERBOTS [Sanz et al., 2017a] se realiza la manipulación con una la misma estrategia utilizada en este trabajo, y que consiste en obtener la posición, orientación y propiedades del objeto para después manipularlo. En el proyecto MERBOTS, sin embargo, solamente se considera la manipulación de un tipo de objeto: la vasija cilíndrica. En resumen, en trabajos anteriores se había utilizado una estrategia en la que se utilizaba información 2D para obtener una posición de agarre en 3D, mientras que en enfoques más recientes se opta por un proceso integral en 3D. De todos modos, en un entorno tan complejo sigue siendo necesario aplicar restricciones a los problemas a tratar.

4.3. Segmentación mediante RANSAC

De acuerdo a la metodología descrita, el primer paso una vez se tiene una reconstrucción 3D correctamente filtrada del entorno es el de discriminar los objetos del fondo. Pese a que para los algoritmos que utilizan RANSAC este paso no es estrictamente necesario, resulta una ventaja ya que reduce la complejidad del problema. Como se ha dicho, se asume que el fondo es relativamente plano o que dispone de textura y una superficie homogénea. Con esto en mente, se proponen dos estrategias para resolver la segmentación del fondo. La primera consiste en la utilización de un algoritmo iterativo denominado RANSAC [Fischler y Bolles, 1987] y la segunda en utilizar técnicas de agrupamiento, detalladas en secciones posteriores.

El método utilizado principalmente para la posterior estimación del objeto es RANSAC. Se trata de un algoritmo iterativo no determinista que sirve para calcular los parámetros de un modelo matemático a partir de un conjunto de datos

observados que contiene valores atípicos o *outliers*. Los datos consisten en *inliers*, es decir, datos cuya distribución se explica por un conjunto de parámetros del modelo, aunque pueden estar sujetos a ruido, y valores atípicos o *outliers*, que son datos que no encajan en el modelo. Los valores atípicos pueden provenir, por ejemplo, de valores extremos provocados por ruido, de mediciones erróneas o de hipótesis incorrectas sobre la interpretación de los datos. Los algoritmos que utilizan RANSAC también asumen que, dada un conjunto de *inliers* generalmente pequeño, existe un procedimiento que puede estimar los parámetros de un modelo que explica de manera óptima o se ajusta a esta información. Este método permite estimar modelos matemáticos muy diferentes. En esta tesis se ha utilizado para la estimación de modelos de cilindros, esferas y planos con su implementación de PCL [Rusu y Cousins, 2011].

En el caso que nos ocupa, se ha utilizado para eliminar el fondo el método de RANSAC para obtener el modelo matemático de un plano. Este algoritmo se utiliza habitualmente para detectar superficies en entornos cerrados como pueden ser paredes, suelos o mesas. Para la detección del plano se ha utilizado la forma Hessiana Normal que define el plano a través de 4 coeficientes con la siguiente ecuación general $ax + by + cz + d = 0$ de forma que en este caso el vector unitario normal del plano se define como $\hat{n} = (a, b, c)$.

La segmentación del fondo tiene una gran utilidad, ya que permite conocer la altura del vehículo con respecto al fondo marino y acelerar los cálculos posteriormente. El uso de sensores de presión resulta muy útil una vez conocida la profundidad a la que se encuentra el plano con respecto al vehículo. Una vez se ha separado el fondo se dispone de los potenciales objetos de forma aislada. Para los pasos siguientes se muestran diferentes alternativas.

En el caso de los métodos de segmentación mediante RANSAC, el siguiente paso consiste en la búsqueda del objeto entre los puntos que no son considerados el fondo. En este proceso se separan los puntos que son parte del objeto (*inliers*) de los que no lo son (*outliers*). Alternativamente, en el resto de métodos se realiza a continuación un proceso de agrupamiento. Después, se utiliza cada vez un único clúster de puntos para obtener la posición, orientación y geometría de cada clúster en vez de mostrar resultados para un único objeto.

4.3.1. Obtención de características del objeto con RANSAC

La primera técnica utilizada para la detección y manipulación de ánforas es la segmentación con RANSAC de objetos cilíndricos [Sanz et al., 2017b]. La ecuación implícita de un cilindro es $(x + a)^2 + (y + a)^2 = r^2$, de modo que al utilizarla con RANSAC se obtiene un cilindro de altura infinita. Así pues el resultado de este procesamiento son 7 coeficientes $(e_x, e_y, e_z, x, y, z, r)$, siendo (e_x, e_y, e_z) la dirección del eje del cilindro, (x, y, z) un punto perteneciente al eje y r el radio. La altura h del cilindro se calcula posteriormente utilizando los puntos pertenecientes al cilindro que han sido detectados mediante el algoritmo. Además, se utilizan únicamente el 90 % de los puntos para descartar ruido o posibles valores extremos. De esta forma se evita que un pequeño grupo de puntos detectados como parte del objeto erróneamente desplacen su centroide y empeoren la calidad de los agarres obtenidos. Toda esta información nos permite conocer con exactitud la posición, orientación y características (radio r y altura h) de cualquier objeto con forma cilíndrica. En



Figura 4.2: Segmentación de un plano y un cilindro mediante RANSAC.

la Figura 4.2 se muestra la segmentación del plano y el cilindro a través de un caso de uso ideal, puesto que ni el fondo ni la vasija presentan irregularidades notables.

La posición y orientación del objeto con respecto a la cámara (representado como cMo) se utiliza junto con los parámetros del modelo para realizar la especificación del agarre en el Capítulo 5. También se utiliza la transformación del objeto con respecto al mundo (representado como wMo), que se calcula utilizando la odometría del robot. Con el objetivo de obtener una representación homogénea en todos los métodos, se define la siguiente convención: el eje y del sistema de coordenadas del objeto debe estar alineado con el eje mayor del objeto. El eje z se debe calcular de forma que se minimiza el ángulo con la normal del plano del suelo para dirigir el eje z hacia el suelo

De la misma manera, se propone la utilización de RANSAC para la detección de un modelo esférico para objetos a través de su ecuación $x^2 + y^2 + z^2 = r^2$. En este caso se obtienen 4 coeficientes (x, y, z, r) que representan el origen de la esfera (x, y, z) y su radio r . Por otro lado, en los casos de uso presentados inicialmente se señala el caso de la manipulación de un modelo de caja negra. Por ello, sería deseable detectarla haciendo uso de RANSAC. No obstante, esto no es posible ya que un poliedro no dispone de una fórmula matemática directamente aplicable a con RANSAC como las utilizadas para esferas y cilindros. Más adelante se presentan alternativas para detectar cajas. Cabe destacar que con estos métodos es posible detectar parte de la forma de objetos más complejos. Por ejemplo, es posible segmentar la parte cilíndrica de una taza de forma que se podría utilizar esta parte del objeto sin tener en cuenta el asa para manipular el objeto. En [Huebner y Kragic, 2008] se propone una alternativa similar utilizando múltiples cajas para segmentar un objeto y obtener posiciones de agarre.

Como se ha avanzado en la introducción, un paso importante en este tipo de algoritmos es la parametrización de los mismos. En este paso se desea seleccionar los parámetros con los que se obtienen los mejores resultados posibles en un tiempo razonable. Los parámetros configurables en RANSAC son los siguientes:

- Umbral de segmentación para el fondo: el parámetro $Background_{threshold}$ es el umbral de distancia para que los puntos se consideren parte del modelo del plano utilizado para extraer el fondo.
- Número de iteraciones para el fondo: el parámetro $Background_{iterations}$ es el número de iteraciones máximas que utiliza el algoritmo. Es la condición de

Tabla 4.1: Parámetros por defecto del algoritmo basado en RAN-SAC

Parámetro	Segmentación del fondo	Segmentación del modelo
Umbral de segmentación	0.05	0.05
Número de iteraciones	100	20000
Tamaño máximo	-	0.15

terminación en caso de que los puntos del objeto no alcancen una distancia mínima al modelo.

- Umbral de segmentación para el modelo: el parámetro $Model_{threshold}$ es el umbral de distancia para que los puntos correspondientes se consideren parte del modelo del objeto utilizado, que puede ser un cilindro, esfera u otro plano en el caso de la extracción de una caja.
- Número de iteraciones para el modelo: el parámetro $Model_{iterations}$ es el número de iteraciones máximas que utiliza el algoritmo. Es la condición de terminación en caso de que los puntos del objeto no alcancen una distancia mínima al modelo.
- Tamaño máximo del modelo del objeto: el parámetro $Model_{sizeLimit}$ determina el tamaño máximo que puede tener el objeto. No se aplica en para la detección de cajas y limita el radio máximo del objeto en el caso de la segmentación RANSAC de cilindros y esferas. Este parámetro evita la degeneración del método pues un plano puede ser confundido con el lado de cilindro de radio muy grande.

En la Tabla 4.1 se pueden ver los valores por defecto utilizados en los experimentos. Estos valores se han obtenido de forma experimental en base a multitud de ejecuciones. Como se puede ver, el algoritmo de extracción del fondo necesita un número de iteraciones muy bajo ya que converge con rapidez debido a que la mayoría de los puntos 3D pertenecen al fondo. Existen trabajos que tratan de obtener este tipo de parámetros de forma automática en un proceso que se conoce como *autotuning*. Este proceso, sin embargo, es computacionalmente muy costoso. En consecuencia, la necesidad de ajustar los parámetros de forma manual es una de las debilidades de estos métodos. Por este motivo, la interfaz que se describe en el Capítulo 5 ofrece al usuario la posibilidad de de ajustarlos.

4.4. Agrupamiento de objetos

El *clustering* o agrupamiento es un método que consiste en agrupar una serie de puntos de acuerdo con un criterio. Este criterio puede ser distancia, similitud de color o de superficie. La cercanía se define en términos de una determinada función de distancia, como la euclídea. En esta tesis se ha utilizado el agrupamiento por distancia euclídea y el agrupamiento por crecimiento de regiones para obtener los subconjuntos de puntos correspondientes a diferentes objetos.

Los algoritmos de agrupamiento se utilizan para dividir una nube de puntos en partes más pequeñas o clústeres. El método más común es la división mediante *Nearest neighbors* y distancia euclídea de forma que sólo los vecinos más cercanos entre sí pertenecen a un determinado grupo o *clúster*. Esta separación se puede ver en la Figura 4.3, donde cada uno de los grupos de píxeles se muestra en un color. En el caso del agrupamiento por distancia euclídea, se debe configurar el parámetro de tolerancia que determina a qué distancia los puntos se consideran pertenecientes a otro clúster. En este caso se ha utilizado un valor de 0,02 obtenido de forma experimental. Además, de forma compartida con el siguiente método, los parámetros *minClusterSize* y *maxClusterSize* determinan el tamaño mínimo y máximo de los clústeres que pueden ser detectados. Puesto que es un valor en píxeles, debe ser determinado de forma experimental para cada cámara, pues el tamaño real de los objetos se relaciona con el número de píxeles en función de la resolución de la cámara y del paso de *downsampling* o reducción de la resolución. En los experimentos en el laboratorio, se ha utilizado un umbral de 300 puntos para que un clúster sea considerado un objeto. Pese a que el reescalado garantiza una resolución espacial uniforme, la densidad de puntos es menor en presencia de ruido. Por ello, en escenarios como la reconstrucción en el mar, es necesario considerar un umbral de tamaño de 150 puntos. Este hecho constata las dificultades que se encuentran cuando se utilizan algoritmos de demostrada eficiencia, pero con un conjunto de parámetros que puede variar en función de la situación.

En segundo lugar, también se ha utilizado el *Region Growing* o crecimiento por regiones, en el cual se unen entre sí puntos cercanos en términos de una condición de uniformidad de la superficie. El resultado de estos algoritmos son diferentes conjuntos de puntos que se consideran parte de la misma superficie. En la implementación utilizada se tiene en cuenta principalmente el ángulo entre las normales de los puntos y también la curvatura de la superficie en cada punto. Para ello, se utiliza un método de estimación de normales de la superficie. El algoritmo de cálculo de normales consiste en obtener por cada punto la normal del plano tangente a la superficie [Rusu, 2009]. Así, se utiliza un conjunto de k vecinos para aproximar el plano que determina la normal a ese punto.

El crecimiento por regiones es muy apropiado para determinar superficies que representan un mismo objeto. En algunos casos un objeto puede tener varias partes pero la parte a manipular tiene normalmente una superficie homogénea. El algoritmo empieza la búsqueda de regiones a partir del punto de menor curvatura para minimizar el número de grupos. Después selecciona iterativamente los puntos vecinos con una curvatura y normal similar. Cuando una región no puede crecer más, se vuelve a repetir el proceso hasta que no quedan más puntos sin agrupar. Entonces, se filtran las regiones que no contienen un número de puntos mínimo.

Los parámetros a configurar son la diferencia en el ángulo de las normales de cada punto al punto inicial del grupo y un umbral de curvatura máxima. Los valores utilizados han sido 8 y 10, respectivamente. Este método es muy interesante para agrupar puntos de objetos que puedan presentar una discontinuidad en su superficie por un error de visibilidad pero que claramente formen parte de la misma superficie. Por otro lado, no es posible agrupar partes de un mismo objeto que sean geoméricamente diferentes. La integración de estas dos características da lugar a un método muy adecuado para segmentación de objetos.

Otra alternativa a estos dos métodos es el agrupamiento basado en color, cuyo

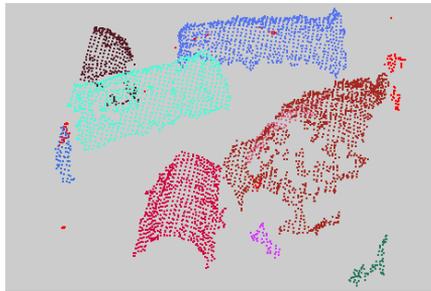


Figura 4.3: Resultado del agrupamiento por distancia Euclídea.

criterio para unir regiones es la similitud de color. Sin embargo, el resultado es generalmente peor en este contexto. Además, la mayoría de los experimentos se realizan con una cámara en blanco y negro, por lo que los colores se discriminan en menor medida. El uso de técnicas de agrupamiento por color resulta muy adecuado en el caso de fondos que sean homogéneos pero no planos, como pudiera ser un fondo arenoso.

Los algoritmos de agrupamiento se han utilizado en este trabajo para separar los diferentes objetos de la escena una vez eliminado el fondo. Sin embargo, también puede utilizarse el agrupamiento como paso inicial para eliminar el fondo de la escena. Esto es posible siempre que el fondo presente una continuidad y sea homogéneo.

4.5. Estimación de Simetrías

Los algoritmos de RANSAC tratan de calcular el modelo del objeto a partir del conjunto de datos de la escena. Sin embargo, en las ocasiones en las que se toma un único clúster como entrada del algoritmo de estimación y se considera que todos los puntos forman parte del objeto, es importante disponer de la mayor cantidad posible de información sobre el objeto. En este caso, la información relativa al objeto es su nube de puntos. Por este motivo, obtener información de simetrías ayuda a la correcta estimación de la geometría del objeto. Este paso es esencial para la estimación del modelo mediante SQ y PCA.

En el estado del arte actual el uso de planos de simetría resulta muy útil en aplicaciones que hacen uso de objetos hechos por el hombre cuando se considera que no se dispone del modelo completo del objeto [Bohg et al., 2011] [Schiebener et al., 2016]. Estas aproximaciones tratan de completar la información del objeto estimando una parte del objeto mediante planos o ejes de simetría. Aunque en estos entornos los planos o ejes de simetría suelen ser ortogonales al plano de soporte [Schiebener et al., 2016], en la aproximación utilizada aquí los objetos yacen en el fondo marino y se capturan aproximadamente desde una vista cenital. Por este motivo, la estrategia para obtener la información de simetrías es diferente. Normalmente, en el fondo marino los objetos tienden a reposar sobre su lado más largo de modo que su eje de simetría es paralelo al suelo.

Los objetos creados por el ser humano suelen tener un eje de simetría, siendo este hecho muy claro en los objetos arqueológicos cerámicos que se consideran en

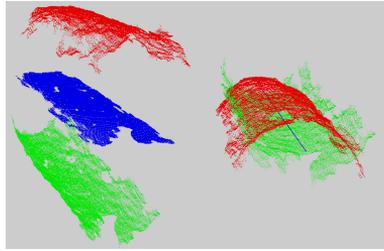


Figura 4.4: Aplicación de la simetría planar (izquierda) y axial (derecha). En rojo la fuente, en azul la proyección de los puntos sobre el eje o el plano y en verde los puntos generados.

este trabajo. Por otro lado, con el uso de planos de simetría también se obtiene una aproximación válida e incluso resulta una mejor opción para objetos como cajas. Por este motivo, se ha desarrollado un método para la estimación de partes ocultas de objetos mediante planos y ejes de simetría. En ambos casos, se han desarrollado dos estrategias para obtener el plano o el eje de simetría. La primera consiste en obtener el mejor plano o eje posible con la información de la escena. El segundo consiste en un proceso de búsqueda en el que se tiene en cuenta si los puntos resultantes tienen sentido. Se considera en primer lugar la aproximación para calcular el mejor plano o eje.

Para la estimación del mejor plano de simetría P_s se utiliza como origen del plano O_s el punto medio $pm_{o,f}$ entre el punto del objeto más lejano al fondo y el plano del fondo. La normal del plano n_s corresponde a la normal del fondo, previamente segmentado. Esta aproximación es rápida y resulta útil para la estimación del modelo del objeto. De hecho, ha demostrado ser suficiente para estimar correctamente la geometría de gran parte de los objetos utilizados. Sin embargo, no es la aproximación más precisa para algunos objetos. Especialmente, en el caso de los objetos cuyo eje o plano de simetría no es paralelo al plano del fondo, no se puede realizar la estimación correctamente con este algoritmo.

La estimación del eje de simetría E_s de una forma sencilla se realiza utilizando como origen O_s el centroide del objeto C_o y como vector director v_s la dirección del eje mayor del objeto. Dicho eje se obtiene utilizando el mayor eigenvector de la nube de puntos del objeto. Esta estrategia de inicialización tiende a obtener peores resultados que la anterior puesto que el centro de masas del objeto y origen del eje de simetría real está alejado del centroide de los puntos visibles del objeto. Es posible el cálculo de un punto inicial más adecuado teniendo en cuenta la distancia al fondo igual que en el caso del plano. Sin embargo, se propone una metodología de búsqueda capaz de obtener tanto el eje como el plano de simetría de forma más precisa.

En la Figura 4.4 se muestra el proceso de obtención de nuevos puntos utilizando las simetrías. Se muestra la nube de puntos del objeto original en rojo, la proyección de estos puntos sobre el eje y plano de simetría en azul, y la parte simétrica obtenida en verde. En este caso, el eje y plano obtenido simplemente ilustran la obtención de nuevos puntos y no son representativos de la posición y orientación obtenidos con el método descrito.

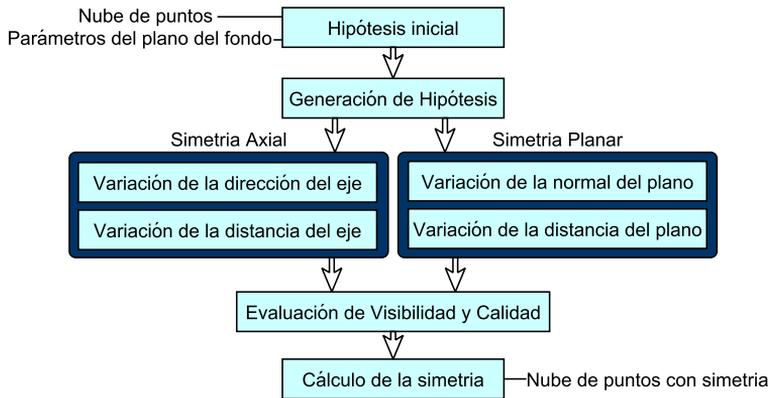


Figura 4.5: Componentes del método de estimación de planos y ejes de simetría.

4.5.1. Búsqueda de Simetrías

Como se ha dicho, la aproximación más sencilla es válida en el caso de objetos cuyo eje es totalmente paralelo al suelo. Por ello, se propone mejorar la estrategia buscando ejes o planos de simetría de forma que se reconstruya la forma del objeto con mayor fidelidad. Se ha desarrollado un sistema de puntuación que determina cómo de buena es la estimación de la simetría en base a condiciones de visibilidad y de volumen del objeto. En la Figura 4.5 se sintetiza el proceso de estimación. El algoritmo de búsqueda consiste en los siguientes pasos:

1. *Cálculo de Hipótesis inicial:* se toma como hipótesis del plano o eje inicial las descritas anteriormente, que se basan en la nube de puntos del objeto y los parámetros de detección del plano del fondo.
2. *Generación de Hipótesis de simetría:* se obtienen múltiples hipótesis a partir de la inicial variando su posición y orientación. Primero se muestrea el origen del plano O_s situándolo a diferentes distancias d_i entre el plano del fondo y el punto del objeto más lejano al fondo. Se realizan m pasos cada 5cm. Simultáneamente, se muestrean $n \times l$ rotaciones α_j y β_k de la normal del plano de simetría en el rango de -30° a 30° cada 2° . Así, se obtiene un conjunto de $m \times n \times l$ hipótesis. En el caso el eje de simetría, las variaciones α_j y β_k se realizan sobre el vector director del eje.
3. *Cálculo de la nube de puntos reflejada:* por cada hipótesis de simetría $s_{(i,j,k)}$ se calcula la nube de puntos reflejada a partir de la original. Este paso consiste en aplicar el plano o eje de simetría individualmente para cada punto.
4. *Evaluación de Visibilidad:* una vez se ha obtenido la nube de puntos reflejada se procede a evaluar la plausibilidad de esta estimación. Para ello se utiliza una métrica que consiste en evaluar la visibilidad de los nuevos puntos. En este caso se propone una función que debe ser minimizada. Por cada punto

reflejado p_R se traza un rayo en dirección a la cámara. Después se busca la intersección entre dicho rayo y un punto de la nube original p_I . Entonces, se pueden dar los siguientes casos:

- a) Si el rayo trazado por el punto no interseca con ningún punto de la nube original, este punto debería verse inicialmente. Por lo tanto, contradice la hipótesis y incrementa su puntuación en gran medida. El valor asignado a este incremento es la distancia al punto más cercano de la nube original que podría intersectar p_{NN} , multiplicada por una constante: $vis_1 = K_1 * dist(p_R, p_{NN})$.
- b) Si el punto está delante del objeto este debería verse inicialmente. Debido a que este hecho contradice la hipótesis, la puntuación se incrementa en función de la distancia entre los puntos, multiplicada por una constante: $vis_2 = K_2 * dist(p_R, p_I)$.
- c) Si el punto está muy cerca de su punto correspondiente, apoya la hipótesis de simetría y aumenta su puntuación de forma proporcional a la distancia. En este caso se incrementa la puntuación debido a que si se reduce el método degeneraría a buscar objetos con el mayor volumen posible. Dando un valor positivo con diferentes pesos a cada caso se trata de minimizar el volumen del objeto reconstruido al tiempo que se prioriza la visibilidad. El valor asignado a este incremento es la distancia entre ambos puntos multiplicada por una constante: $vis_3 = K_3 * dist(p_R, p_I)$.
- d) El punto está detrás del objeto. En este caso el punto ocupa espacio que no es visible normalmente, apoyando la hipótesis. En este caso se incrementa la puntuación el mínimo posible, sumando simplemente la distancia entre los puntos: $vis_4 = dist(p_R, p_3)$

Para obtener la puntuación se recorren los puntos reflejados y se aplica para cada uno el cálculo deseado. La constante de mayor valor es K_1 y K_2 es mayor que K_3 . De esta forma se priorizan los puntos con mejor criterio de visibilidad.

5. *Selección de la Hipótesis final*: se selecciona la hipótesis que obtiene una menor puntuación como la más adecuada. A través de este sistema se busca el mínimo global entre todas las posibilidades.

Tras la obtención de la mejor hipótesis en [Schiebener et al., 2016] se propone un método para completar el resto de los puntos del objeto utilizando el conocimiento sobre el tipo de objetos comunes. Se consideran los puntos correspondientes a la base de un objeto que está apoyado en una mesa y los puntos que corresponden a los lados del objeto visto desde la cámara. Estos puntos generalmente no se reconstruyen adecuadamente con sensores 3D. Por otro lado, en [Bohg et al., 2011] se utiliza la nube de puntos obtenida para estimar las normales de los nuevos puntos y realizar una reconstrucción 3D basada en Poisson [Kazhdan y Hoppe, 2013]. En este trabajo, se propone utilizar las nubes de puntos obtenidas como entrada para los algoritmos basados en SQ y PCA. Los métodos basados en RANSAC no se benefician del uso de simetrías puesto que la estimación no requiere tener los objetos aislados previamente mediante *clustering*.

El método de simetría planar aplicado a diferentes objetos se muestra en la Figura 4.6. Como se puede apreciar, en los objetos *A*, *C*, y *H* la estimación es coherente y sirve para obtener la forma completa del objeto. Sin embargo, el objeto *K* corresponde a una vasija partida por la mitad y el algoritmo aplicado (simetría planar utilizando la distancia al suelo para determinar la posición del plano) no es capaz de estimar el resto del objeto correctamente. El objeto *S*, que no ha sido descrito previamente, corresponde a la nube de puntos de una de las piedras del fondo. Este objeto no se estima correctamente debido a su geometría irregular, mostrando que los algoritmos desarrollados funcionan mejor al segmentar objetos regulares fabricados por el hombre. Así pues, el método funciona de la forma esperada con la mayoría de objetos del conjunto utilizado. En el caso de los objetos *K* y *L*, estos se asemejan al caso de que un objeto esté parcialmente enterrado. Actualmente el algoritmo no tiene en consideración esta posibilidad por lo que una posible línea de futuro sería detectar estos caso y aplicar la simetría correctamente.

Finalmente, la Figura 4.7 representa el resultado de la estimación de simetrías a una escena completa. En dicha escena se muestran los puntos obtenidos mediante este método sobrepuestos en azul. La escena contiene los objetos *A*, *C*, *H*, *KL* y *S*. Una vez acabado este proceso, se dispone de cada uno de estos objetos con su modelo completo y aislado del resto de la escena. En este momento se puede aplicar la estimación mediante Momentos de Inercia o SQ.

4.6. Obtención de características mediante Momentos de Inercia

Una vez se dispone de una reconstrucción del objeto aislada y completa, se pueden aplicar otros métodos de estimación. A continuación se muestra el método más sencillo para la obtención de un modelo básico de un objeto. Este método consiste en obtener los momentos de inercia [Taubin y Cooper, 1991] y la *minimum oriented bounding box* de un objeto, es decir la *bounding box* mínima que cubre el objeto. Esto resulta especialmente útil, ya que con el manipulador del que disponemos esto puede resultar suficiente para obtener el agarre correcto. De hecho es comparable a algunos de los métodos que se han utilizado en los trabajos previos en 2D, aunque pueda considerarse el método más simple. En este caso, antes de realizar el cálculo de los momentos de inercia se parte de una nube de puntos correspondiente al objeto. Es decir, una nube que ha sido obtenida mediante técnicas de agrupamiento partiendo de una nube filtrada y se han obtenido los puntos correspondientes a posibles simetrías.

Los momentos de inercia se pueden extraer utilizando Análisis de Componentes



Figura 4.6: Estimación de la simetría planar en diferentes objetos.

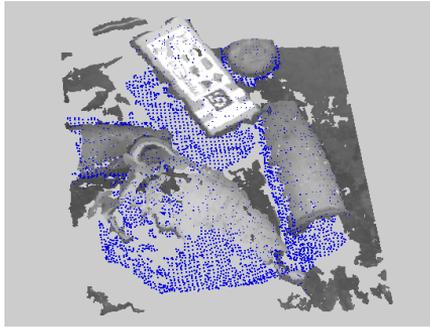


Figura 4.7: Escena mejorada utilizando la estimación de simetría.
En azul los puntos añadidos.

Principales o PCA, que es como se denomina este método en el resto del trabajo. Este algoritmo ha sido utilizado en robótica para determinar la posición y orientación de un objeto de forma aproximada [Artac et al., 2002] entre otros muchos usos. Esta técnica estudia la relación lineal entre las variables al convertir un conjunto de observaciones en otro conjunto de variables linealmente no correlacionadas. Los componentes principales se extraen mediante descomposición de valores singulares de la matriz de covarianza centrada en la nube de puntos. Tras realizar el cálculo de PCA se dispone del centroide $C = (x, y, z)$ de los datos de entrada, los eigenvalores $\lambda_1, \lambda_2, \lambda_3$ en orden descendente y sus correspondientes eigenvectores $v_{\lambda_1}, v_{\lambda_2}, v_{\lambda_3}$.

Con estos valores se obtiene la dirección y orientación del centro del objeto. Después, se obtiene el *bounding box* a través de puntos máximos y mínimos en cada eje. Esta información de geometría del objeto $G_o = (x, y, z, \rho, \psi, \theta, height, width, depth)$ se utiliza para determinar la manipulación del objeto como se describe en el Capítulo 5. A diferencia del resto de métodos, este método no utiliza parámetros que se deban configurar.

4.7. Obtención de características de una caja con RANSAC

El siguiente método utilizado consiste en la extracción de la geometría de cajas. Dado que RANSAC no es capaz de estimar directamente el modelo de una caja, se utiliza una aproximación basada en planos. Una vez estimada la posición del plano, se busca utilizando RANSAC un nuevo plano, que será la cara superior de esta caja debido a la configuración de las cámaras del robot. Después, se utiliza la estimación de la simetría para estimar la cara inferior de la caja. Finalmente, utilizando la estrategia de PCA propuesta en la sección anterior se pueden estimar la posición, orientación y dimensiones de la caja $G_c = (x, y, z, \rho, \psi, \theta, height, width, depth)$. En la Figura 4.8 se muestra la estimación del modelo de una caja negra obtenida mediante este método. Los parámetros de segmentación del plano de la caja son el número de iteraciones y el umbral de distancia al plano. El número de iteraciones es igual que para extracción del fondo, mientras que el umbral de distancia es de 0,04m., ligeramente más ajustado que el del fondo. En este trabajo se denomina



Figura 4.8: Visualización con UWSim de la estimación de la geometría de una maqueta de caja negra. Modelo (blanco) sobre la nube de puntos.

a este método estimación RANSAC basada en planos o RANSAC para cajas de forma abreviada.

4.8. Obtención de características mediante Supercuádricas

Finalmente, los objetos de la escena pueden ser caracterizados mediante el uso de supercuádricas (SQ) [Fornas et al., 2016]. De forma similar al caso anterior, se parte de un clúster aislado y filtrado. Además, se le aplica la estimación de simetrías para poder determinar la forma del objeto de forma correcta.

Las supercuádricas son una familia de formas geométricas que pueden representar una gran variedad de formas primitivas con un número relativamente pequeño de parámetros. Se asemejan a los elipsoides, pero con potencias arbitrarias. Estas formas fueron introducidas por primera vez en la comunidad de gráficos por computador [Barr, 1981] y fueron adaptadas posteriormente como una herramienta efectiva de modelado para aproximar la forma de un objeto. En general, las supercuádricas incluyen los superelipsoides y los supertoroides, pero por motivos prácticos sólo consideramos los superelipsoides. Éstos se pueden expresar en su forma canónica con la siguiente ecuación:

$$\left(\left(\frac{x}{a} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{b} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{c} \right)^{\frac{2}{\epsilon_1}} = 1 \quad (4.1)$$

Siendo en esta ecuación a , b y c los factores de escala a lo largo de los ejes principales. ϵ_1 es el factor de forma de la sección transversal de la superelipsoide en un plano ortogonal al plano XY y que contiene el eje Z. ϵ_2 es el factor de forma de la sección transversal de la superelipsoide en un plano paralelo al plano XY. Estos parámetros consideran la superelipsoide centrada en el origen. Si se considera una transformación general, el número total de parámetros requeridos para definir una superelipsoide es 11, siendo los 6 adicionales la traslación y rotación $(x, y, z, \rho, \psi, \theta)$. La solución paramétrica se puede expresar como:

$$x = a \cos(\theta)^{\epsilon_1} \cos(\gamma)^{\epsilon_2} \quad (4.2)$$

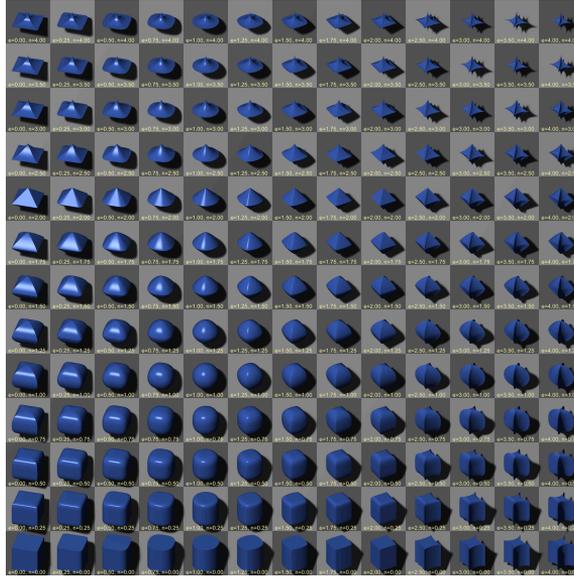


Figura 4.9: Evolución de las superelipsoides según sus parámetros ϵ_1 y ϵ_2 .

$$y = b \cos(\theta)^{\epsilon_1} \sin(\gamma)^{\epsilon_2} \quad (4.3)$$

$$z = c \sin(\theta)^{\epsilon_1} \quad (4.4)$$

Siendo $\theta \in [-\varphi/2, \varphi/2]$ y $\gamma \in [-\varphi, \varphi]$. A través de esta ecuación, se puede muestrear una superelipsoide de la que se conocen sus parámetros. En la Figura 4.9 se pueden ver las formas resultantes en función de los factores de forma ϵ_1 y ϵ_2 , siendo $a = b = c = 1$.

Teniendo esto en cuenta, el procedimiento para ajustar el modelo de la superelipsoide [Ichim, 2013] al objeto consiste en los siguientes pasos:

1. Se centra la nube de puntos en el origen de coordenadas.
2. Se calculan los ejes principales de la nube de puntos utilizando PCA y se alinean con los ejes de coordenadas.
3. Se da valor inicial a las variables a optimizar, siendo ϵ_1 y ϵ_2 igual a 1. a , b y c se inicializan utilizando los valores singulares obtenidos con PCA [Biegelbauer y Vincze, 2007].
4. A partir de esta inicialización se realiza un proceso de optimización local basado en el algoritmo Levenberg-Marquardt para obtener los parámetros finales [Katsoulas et al., 2008].

En primer lugar se utiliza PCA para obtener una estimación inicial de la posición y orientación del objeto. En segundo lugar se lleva a cabo un proceso de minimización.

El ajuste de diferentes nubes de puntos utilizando SQ se puede ver en la Figura 4.10. En esta ocasión, se ha utilizado como fuente de datos es sensor de profundidad Microsoft Kinect [Zennaro et al., 2015] como fuente de datos. Como se puede

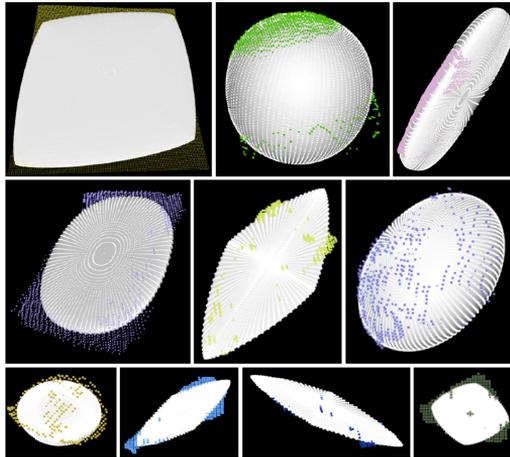


Figura 4.10: Varios clústeres dan lugar a diferentes ajustes de superelipsoides.

ver, este método se adapta a gran variedad de geometrías incluso en presencia de ruido.

Una vez obtenidos los 11 parámetros correspondientes al modelo se pueden utilizar los parámetros de posición y orientación $(x, y, z, \rho, \psi, \theta)$, así como los de escalado (a, b, c) para calcular el agarre sobre un objeto de la misma manera que en PCA. A partir de la fórmula del modelo y los parámetros obtenidos se obtiene un modelo en 3D. Este modelo se muestra al usuario en UWSim sobre la nube de puntos. En el contexto de una intervención supervisada, resulta tan importante ofrecer un buen sistema de especificación autónoma del agarre como disponer de información 3D adecuada para ayudar al usuario en la toma de decisiones. Como se verá en los resultados, este método es el más avanzado de los presentados y con el que se obtiene una representación de los objetos más flexible. Por otro lado, es el algoritmo más costoso con diferencia.

4.9. Seguimiento de objetos

En un entorno tan dinámico y variable como el que se presenta, no es posible asumir que la posición del objeto se obtenga una vez y no necesite ser actualizada. En proyectos anteriores, se han utilizado satisfactoriamente algoritmos de seguimiento de objetos en 2D basados en detección de características, bordes o plantillas [Prats et al., 2012a] (ver Figura 4.11). El seguimiento de objetos mediante detección de bordes permite conocer la posición y orientación de un objeto en 3D pero necesita un modelo preciso del objeto a priori. Los métodos basados en plantillas y características se utilizan para realizar el seguimiento en 2D de un objeto. No obstante, se pueden ampliar para determinar la posición en 3D siempre que se disponga de información sobre el precisa sobre el objeto, como sus dimensiones. Un buen ejemplo de ello es seguimiento mediante Efficient Second-order Minimization (ESM) [Benhimane y Malis, 2004] para el seguimiento de la caja negra, obteniendo

su posición en 3D gracias a que las dimensiones son conocidas. En el tipo de intervención propuesto en este trabajo, el modelo del objeto no es conocido a priori. Por ello se utilizan los métodos descritos en secciones anteriores con el objetivo de obtener la posición del objeto de manera continua.

En algunos casos, se puede llevar a cabo una segmentación más rápida durante la ejecución utilizando los parámetros ya conocidos. Esta metodología se sigue para la eliminación del fondo, puesto que una vez que el modelo del plano del fondo es conocido puede ser extraído con mayor velocidad, así como para la segmentación por RANSAC.

Con el objetivo de corregir una inicialización deficiente, una calibración imprecisa o una deriva en navegación del vehículo, se utiliza el paradigma de la manipulación guiada por visión. La geometría del objeto no se calcula continuamente, pero sí se vuelve a calcular la posición y orientación del objeto.

En el caso de la detección del cilindro, por ejemplo, se utiliza la profundidad del suelo para eliminar el fondo más rápidamente sin tener que obtener el plano del fondo. Después, se vuelve a calcular la posición y orientación del cilindro con la diferencia de que el radio del cilindro es entonces conocido. En el caso de la esfera, el radio es también conocido. De esta forma, se realiza el seguimiento durante la aproximación al objeto y durante la manipulación mientras el objeto es visible. Se utiliza un filtro temporal para eliminar los posibles errores de forma robusta y descartar detecciones que no sean coherentes y pudieran generar una orden que moviese el brazo a una posición alejada. Esta aproximación tiene en cuenta los problemas generados por el propio robot así como el posible movimiento del objeto debido a las corrientes o a la propia interacción física con el brazo. La corrección de la posición reduce la dependencia de una odometría del vehículo muy precisa.

En el proyecto MERBOTS se han propuesto métodos alternativos para detectar el objeto basados también en la detección de plantillas o *templates* en 3D [Company-Corcoles et al., 2018]. Mientras que en proyectos anteriores se realizaba la detección del objeto utilizando una plantilla del objeto en 2D obtenida con anterioridad, en este caso se obtiene una plantilla del objeto en 3D y sus características para luego hacer un seguimiento del mismo. En el primer caso se hacía uso de características de una imagen que incluye información principalmente de variación de intensidad o color, mientras que en 3D se utilizan características geométricas de los puntos.

Otro objetivo del seguimiento es el de seguir al objeto activamente durante la ejecución para determinar si éste ha sido cogido satisfactoriamente. Dependiendo del algoritmo utilizado y de las oclusiones producidas, el algoritmo puede no ser

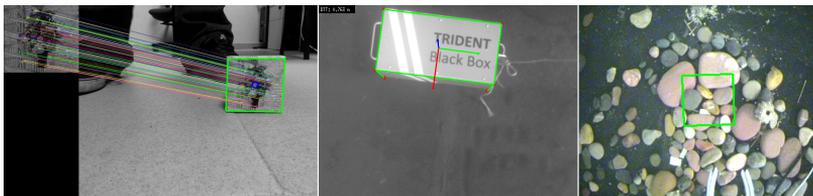


Figura 4.11: Algoritmos de seguimiento en 2D basados en: características (izquierda), seguimiento de bordes (centro) y plantillas (derecha).

capaz de continuar siguiendo al objeto.

4.10. Resultados y comparativa de algoritmos de Estimación

En esta sección se detallan obtenidos con el módulo de Estimación de características para las diferentes escenas consideradas. Los métodos descritos (variantes de RANSAC, PCA y SQ) se utilizan en diferentes entornos y se compara el resultado de aplicar estos algoritmos. En particular, se han utilizado escenas con un modelo de caja negra y con un ánfora en simulación, se han utilizado datos de los experimentos de manipulación del ánfora de MERBOTS (en la piscina del CIRS y en el mar) y finalmente se ha utilizado datos obtenidos en el laboratorio en escenas con uno o más objetos diferentes.

En esta comparativa se analizan los resultados tanto los métodos de estimación como de seguimiento, ya que en los algoritmos de RANSAC se obtienen las características del objeto para luego detectarlo continuamente, siendo clave la primera estimación del objeto. Sin embargo, en los métodos basados en SQ, PCA o detección de planos la estimación se realiza continuamente sin utilizar información previa.

En primer lugar se exponen los resultados simulados por ser el escenario más controlado. Después, se muestran los resultados en la piscina del CIRS y el mar porque se trata de experimentos limitados a un único objeto. Posteriormente, se detallan los resultados en el laboratorio del IRS Lab por ser los más extensos y variados.

4.10.1. Resultados en simulación

En los experimentos realizados en simulación, se dispone de una escena con una caja negra y una con una vasija. En ambas escenas la reconstrucción 3D se realiza utilizando la cámara estéreo virtual. En la Figura 4.12 se muestran diferentes estimaciones realizadas con reconstrucciones 3D a partir de imágenes del simulador. Ambos objetos han sido reconstruidos satisfactoriamente utilizando RANSAC (cada uno con su variante correspondiente), PCA y SQ. El ruido presente en las reconstrucciones se debe a la introducción ruido gaussiano en las imágenes con desviación estándar de 0,03 y a la presencia de partículas en suspensión simuladas. En comparación con las escenas reales el ruido es limitado y la calidad de la reconstrucción es buena. Por ello, los resultados son excelentes en todo momento y la posición estimada es muy estable. Por este motivo, este entorno es ideal como ayuda al desarrollo de algoritmos y como herramienta de evaluación inicial. Gracias al uso del simulador es posible realizar los experimentos en escenarios reales, más costosos y menos flexibles, con mayor facilidad.

4.10.2. Resultados de los experimentos de MERBOTS en la piscina

El primero de los experimentos reales que se va a tratar es el escenario de estimación de la geometría de una vasija en la piscina del CIRS. En estos experimentos

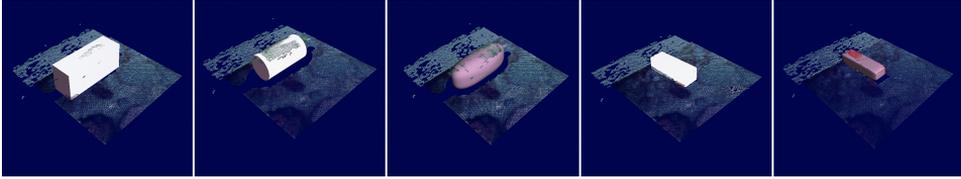


Figura 4.12: Estimación de la vasija y la caja en simulación.

se realizó la integración de la mecatrónica utilizada en el proyecto MERBOTS, siendo un banco de pruebas adecuado para los algoritmos de estimación. Durante los experimentos se registraron datos utilizando la cámara estéreo Bumblebee 2 que, como se ha detallado previamente, cuenta con una resolución de 1280 x 720 puntos, y un sensor basado en láser en la frontera del estado del arte [Palomer et al., 2018], con resolución ajustable. Utilizando ambos sensores, fue posible comprobar que los algoritmos son igualmente efectivos con diferentes fuentes de datos en un escenario complejo. Durante la realización de los experimentos se utilizó el método de estimación de cilindros mediante RANSAC disponible. Las nubes de puntos registradas se han utilizado posteriormente para evaluar los algoritmos basados en PCA y SQ.

La estimación utilizando reconstrucción basada en láser utiliza un conjunto de datos en los que el vehículo se mueve ligeramente con respecto al objeto. En la Figura 4.13) se aprecia este movimiento y cómo todos los algoritmos realizan detecciones siguiendo su trayectoria. Existen diferencias relativas en las posiciones de los centros, debido a que la forma estimada tiene dimensiones diferentes para cada algoritmo y, por tanto, sus centros no pueden coincidir. No obstante, se ve que los métodos obtienen estimaciones muy consistentes en este escenario. Debido a la claridad del agua, la visibilidad es excelente y la reconstrucción es casi ideal.

Por otro lado, en las imágenes utilizadas para la Figura 4.14, el vehículo está en un punto, se desplaza a otra posición y permanece unos instantes en ella. Los métodos de PCA y RANSAC para cilindros siguen la trayectoria de forma muy consistente, y se puede ver claramente la diferencia relativa entre ambos. El método de SQ obtiene unos resultados peores en este escenario. Particularmente, es debido a que no es capaz de aislar correctamente los puntos correspondientes al objeto y se introduce en la estimación ruido que no pertenece al objeto. Cabe destacar la importancia en el proceso de los algoritmos de agrupamiento y de estimación de simetrías.

En la Figura 4.15 se muestran visualmente las dimensiones, posición y orientación estimados de la vasija en el escenario del CIRS con datos provenientes de ambos sensores. Tanto la reconstrucción láser como la reconstrucción estéreo dan lugar a dimensiones similares.

4.10.3. Resultados de los experimentos de MERBOTS en el mar

En esta sección se muestran los resultados de estimación relativos al escenario más complejo disponible, es decir, el experimento de integración de MERBOTS en el mar. En este escenario se utilizó el método de RANSAC para la detección de cilindros, puesto que era el objeto de interés y el único algoritmo de detección en el momento. Además, los datos obtenidos en estos experimentos se utilizan

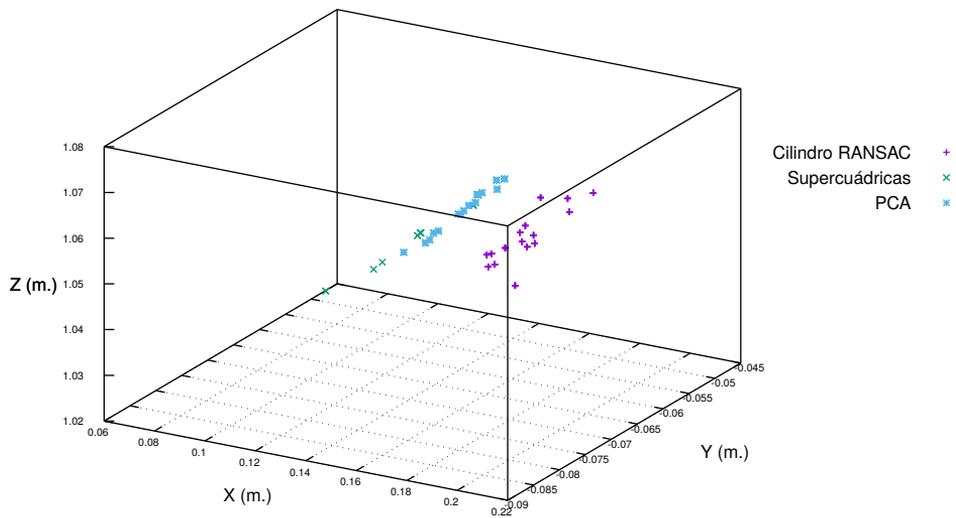


Figura 4.13: Posición estimada en la piscina del CIRS utilizando reconstrucción láser.

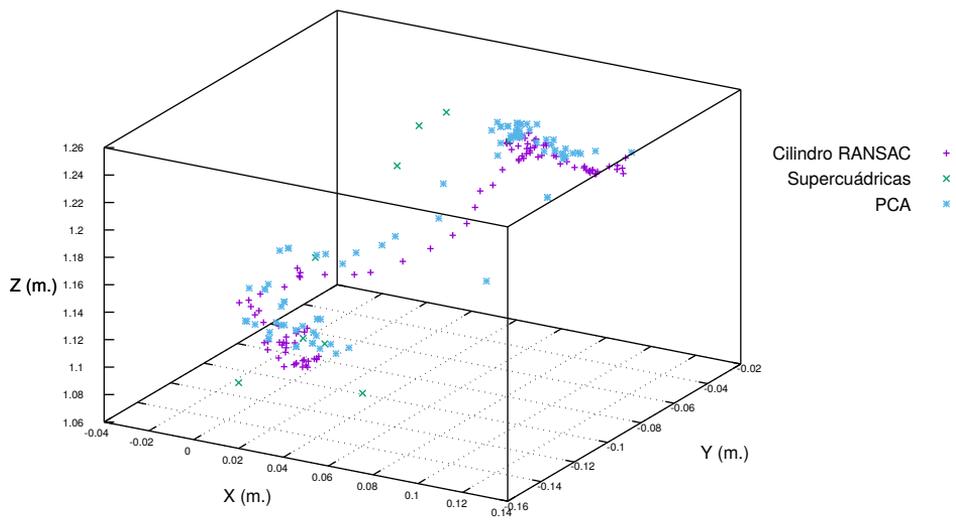


Figura 4.14: Posición estimada en la piscina del CIRS utilizando reconstrucción estéreo.

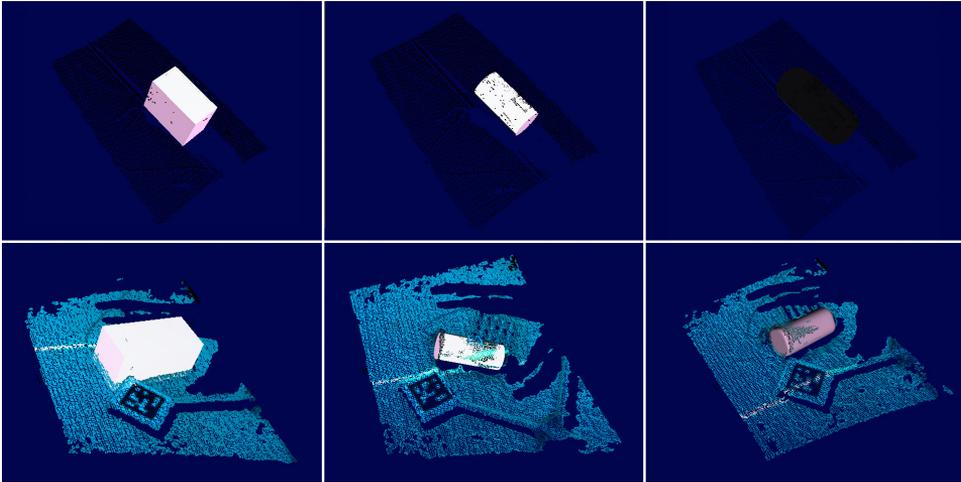


Figura 4.15: Estimación de la vasija en el CIRS con láser (arriba) y estéreo (abajo).

también para evaluar los algoritmos basados en PCA y SQ de manera similar a la evaluación de los experimentos en el CIRS. En este caso, se utiliza únicamente la cámara estéreo Bumblebee 2 para la reconstrucción 3D de la escena.

La calidad de la detección en la fase de estimación en este contexto ha sido evaluada con el uso de marcadores como referencia (*ground truth*). Se utilizaron 4 marcadores diferentes (*AR Markers* detectados mediante la librería ARUCO [Romero-Ramirez et al., 2018]) en 4 puntos del cilindro alrededor de su superficie con el objetivo de obtener la posición del objeto independientemente de su orientación. El uso de marcadores resulta muy robusto y adecuado como *ground truth* pero no es aceptable para objetos en el fondo marino porque no tienen ningún marcador. De igual manera, el modelo completo en 3D del objeto no suele estar disponible, haciendo difícil usar técnicas que confían en disponer de un modelo completo del objeto antes de la ejecución.

La diferencia relativa en posición y orientación con respecto a la obtenida mediante marcadores durante los experimentos de integración se muestra en la Figura 4.16. Esta estrategia de evaluación basada en marcadores está sujeta a errores en la posición. Sin embargo, se puede asumir que el error es comparativamente menor al error de estimación. Los datos mostrados pertenecen a una fase previa a la manipulación donde se localiza el objeto y el robot empieza a moverse hacia el objeto. El objeto permanece inmóvil mientras el robot se acerca y la posición se actualiza de forma progresiva. Se puede observar un error ligeramente mayor en el eje z . Esto es debido a que la profundidad es la dimensión más sensible al error cuando se utilizan sensores 3D. Además, la precisión de una cámara estéreo no es constante con la distancia, sino que tiene una distancia a la cuál la precisión es mayor y a distancias más cercanas o lejanas el error aumenta progresivamente.

Después de comparar únicamente las detecciones obtenidas durante los experimentos reales, se ha procedido a evaluar los métodos de estimación obtenidos con posterioridad utilizando las imágenes obtenidas por las cámaras. En la Figura 4.17

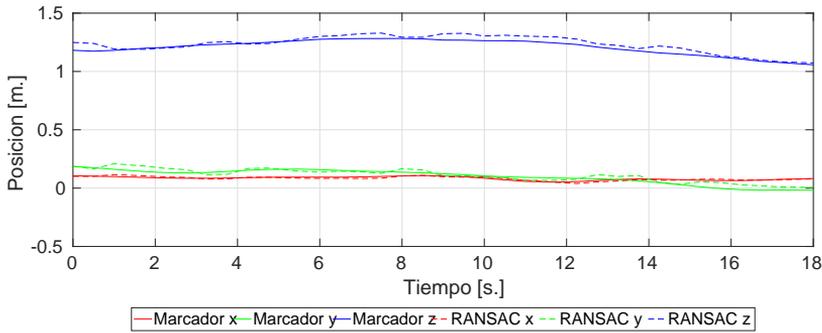


Figura 4.16: Posición del objeto obtenida con marcadores (líneas sólidas) y posición obtenida utilizando la segmentación RANSAC (líneas discontinuas) durante la aproximación al objeto.

se muestran resultados de los algoritmos de estimación de cilindros con RANSAC, PCA y SQ, calculados con el I-AUV en movimiento. Estos son los únicos métodos aplicables a este objeto. Los resultados obtenidos son suficientes para planificar el agarre del objeto, teniendo en cuenta la gran cantidad de ruido y la calidad de la reconstrucción en general, mucho peor que en el resto de escenarios. El algoritmo de PCA obtiene los peores resultados por su sensibilidad al ruido, mientras que en ocasiones SQ utiliza parte del objeto para estimar un supuesto objeto mucho mayor. Este efecto se aprecia también en los experimentos en el laboratorio y a efectos de manipulación se puede subsanar filtrando los agarres generados que no sean próximos a la nube de puntos real del objeto.

En la Figura 4.18, se puede ver la evolución de la posición del centro del objeto a lo largo del tiempo mientras el vehículo sigue una trayectoria cercana a él. Los algoritmos de RANSAC y PCA se ejecutan a una frecuencia alta por lo que se ve claramente la trayectoria trazada por el objeto (debido al movimiento del vehículo con respecto al objeto). El método basado en momentos de inercia obtiene una posición más errática, de nuevo a causa de la sensibilidad al ruido que modifica las dimensiones resultantes del objeto y, en consecuencia, su centro. Los resultados de SQ, debido a que su tiempo de ejecución es elevado, no están distribuidos uniformemente a lo largo de la trayectoria. En los puntos utilizados se puede observar cómo se alejan ligeramente de los puntos obtenidos con RANSAC. Esto se debe al efecto comentado anteriormente, que desplaza el centro del objeto.

En las secciones previas se han mostrado los experimentos de estimación del modelo de una vasija, en el contexto del proyecto MERBOTS. A continuación, se realizan pruebas más exhaustivas con un conjunto de objetos mayor. En las Tablas 4.2, 4.3 y 4.4 se muestra el número de puntos correspondientes al objeto y el tiempo necesario en los escenarios del mar, el CIRS y simulación para realizar la estimación de la posición, orientación y parámetros correspondientes al ánfora. En general, se puede ver cómo el número de puntos correspondientes al objeto es mayor en los métodos que utilizan estéreo en simulación y en la piscina, ya que estos datos presentan una mayor densidad. La reconstrucción estéreo en el mar obtiene un número de puntos mucho menor debido a la baja visibilidad en este entorno, mientras que

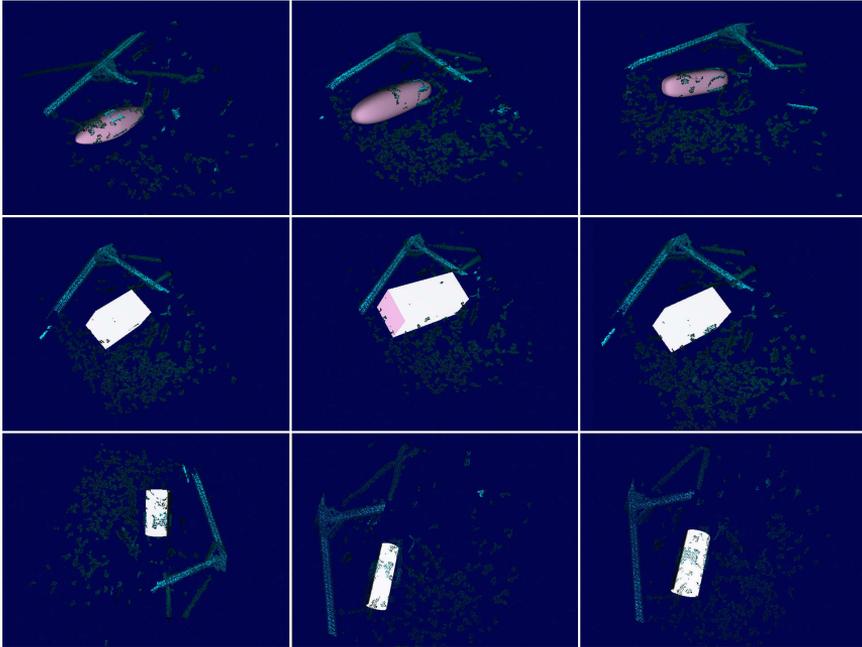


Figura 4.17: Resultados de la estimación de una vasija en el mar.

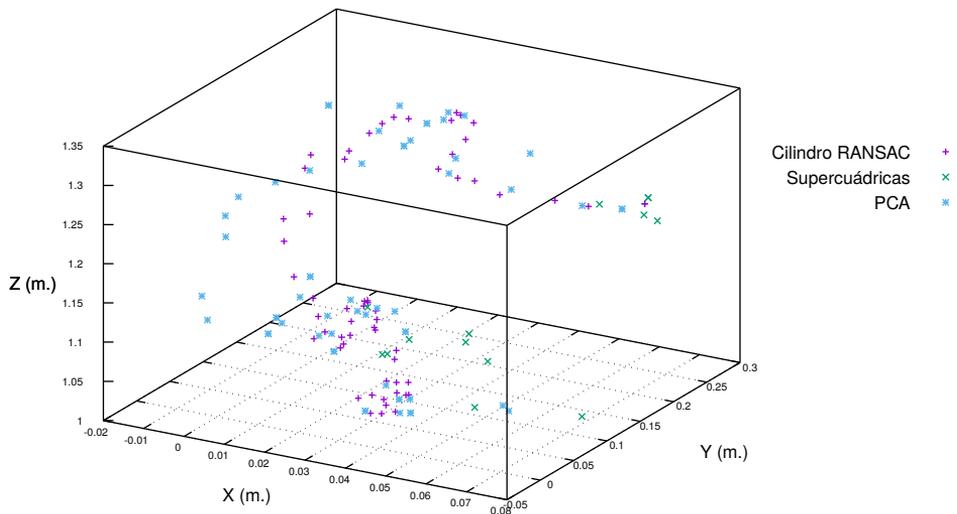


Figura 4.18: Posición de la estimación en el mar.

Tabla 4.2: Resultados de estimación de la vasija con RANSAC en MERBOTS.

	Núm.Puntos	σ	Tiempo (s.)	σ
Mar con estéreo	724,97	84,72	0,0793	0,0087
CIRS con estéreo	1310,99	36,07	0,1661	0,0279
CIRS con láser	351,31	15,62	0,0679	0,0089
UWSim con estéreo	1734,96	7,81	0,1669	0,0132

Tabla 4.3: Resultados de estimación de la vasija con PCA en MERBOTS.

	Núm.Puntos	σ	Tiempo (s.)	σ
Mar con estéreo	1873,37	428,69	0,0867	0,0087
CIRS con estéreo	3159,18	163,31	0,1905	0,0327
CIRS con láser	616,27	31,29	0,0674	0,01
UWSim con estéreo	3978,98	2,44	0,1936	0,01

la reconstrucción utilizando láser presenta el número de puntos resultantes más reducido. Esto es debido a que su resolución es variable y fue configurada para obtener una reconstrucción adecuada rápidamente. Sin embargo, es una cantidad de puntos suficiente para realizar la reconstrucción de calidad en un tiempo reducido. Aunque el resto de algoritmos realizan un proceso de *downsampling*, disponer de una nube de puntos de dimensiones adecuadas desde el principio permite obtener mejores resultados temporales.

Los tiempos de ejecución de los algoritmos basados en RANSAC (ver Tabla 4.2) y en PCA (ver Tabla 4.3) son proporcionales al tamaño de la nube y al número de puntos correspondientes al objeto. En el caso de RANSAC oscilan entre 0,07 y 0,017 segundos. Por otro lado, en PCA varían entre 0,07 y 0,19 segundos. Como se verá en secciones posteriores, estos resultados están en la línea de los resultados obtenidos en el laboratorio, ofreciendo un coste temporal suficiente para corregir la posición de agarre del robot durante la ejecución.

Finalmente, en la Tabla 4.4 se puede ver cómo la estimación basada en SQ está en un orden muy superior en cuanto a coste temporal, ya que los tiempos de ejecución en promedio oscilan entre 51,71s. en el mejor de los casos (reconstrucción láser en el CIRS) a 178,25s. en el peor (reconstrucción estéreo en el simulador). El tamaño de las nubes de puntos tiene una gran influencia en el coste temporal. Además, existe otro factor importante a la hora de determinar el tiempo de ejecución, como es la complejidad del modelo a estimar. Este matiz se puede ver a la hora de comparar la reconstrucción estéreo en el CIRS y en el simulador. Mientras que ésta última tiene un mayor número de puntos, el tiempo de ejecución es menor, debido principalmente a que la reconstrucción es más sencilla en un conjunto de datos sintético que en una nube de puntos real, con formas geométricas imperfectas y ruido.

Tabla 4.4: Resultados de estimación de la vasija con SQ en MERBOTS.

	Núm.Puntos	σ	Tiempo (s.)	σ
Mar con estéreo	1251,25	375,39	84,05	31,05
CIRS con estéreo	3072,12	159,76	178,25	0,023
CIRS con láser	642,08	3,26	51,71	8,26
UWSim con estéreo	3978,18	6,08	99,65	1,13

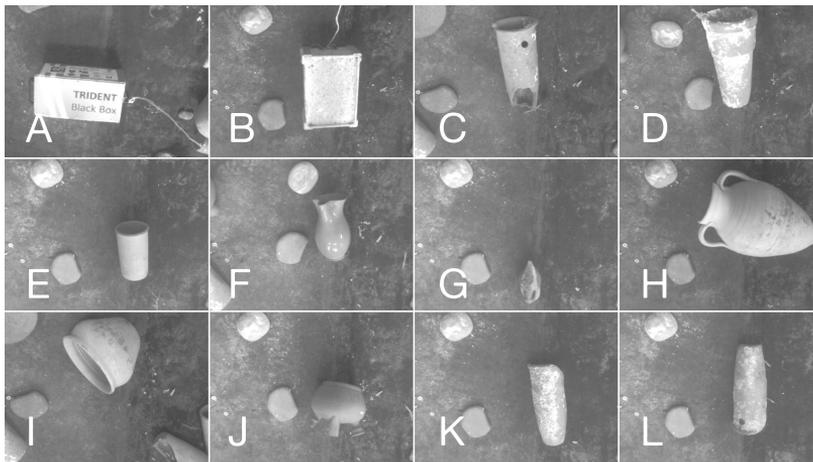


Figura 4.19: Objetos reales capturados con la cámara Videre izquierda, donde se aprecia la variedad de dimensiones y formas de los objetos.

4.10.4. Resultados en el laboratorio IRS Lab

En el laboratorio IRS Lab se han realizado los experimentos más extensos, ya que estos experimentos no están limitados en el tiempo como las pruebas de integración, que implican la coordinación con otros equipos de investigación. En primer lugar, se ha realizado la estimación de la posición y orientación de los objetos (ver Figura 4.19) utilizando los métodos disponibles: RANSAC, PCA y SQ. Además, el método RANSAC se presenta en sus variantes para formas cilíndricas, esferas y cajas. Esta diferenciación es importante ya que su implementación es diferente. Para verificar estos métodos se ha creado una escena para cada uno de los objetos, de forma que se pueden probar los métodos de forma aislada. Por otro lado, no todos los objetos pueden ser estimados por todos los métodos por lo que en cada caso se detallan los objetos utilizados. En el Capítulo 2 se han mostrado los objetos disponibles y sus dimensiones, donde se aprecia la variedad de tamaños y formas de los objetos (ver Figura 4.19).

Posteriormente, se ha realizado la estimación de la geometría de la escena en una configuración con múltiples objetos. En esta configuración se utiliza el método basado en SQ. El método basado en RANSAC puede ser utilizado iterativamente pero es de menor interés, ya que no es capaz de segmentar escenas con múltiples geometrías por su menor flexibilidad.

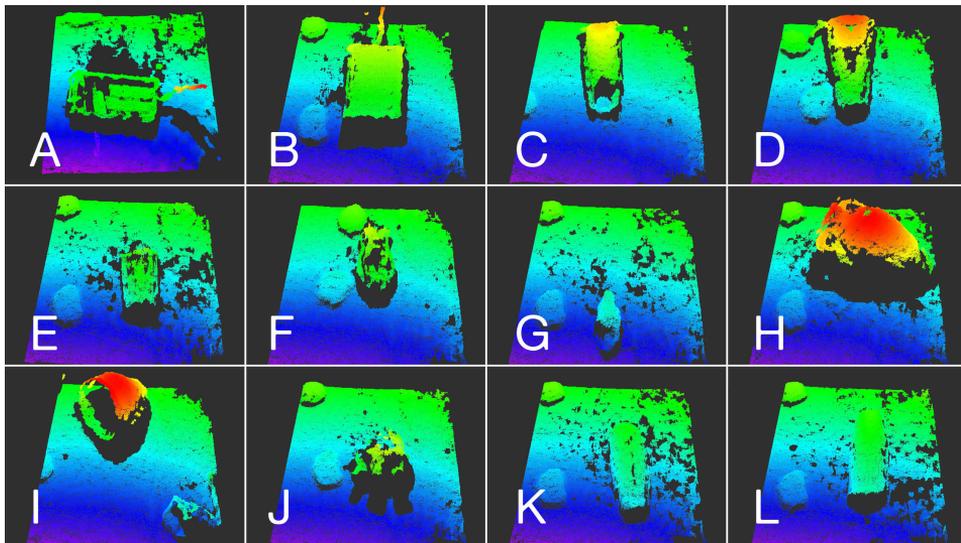


Figura 4.20: Nube de puntos correspondiente a los objetos utilizados.

Las imágenes en el laboratorio han sido tomadas con la cámara estéreo Videre descrita en el Capítulo 3. Esta cámara utiliza una resolución de 640 por 480 píxeles y se ha seleccionado el método de *block matching* local por su mayor rapidez, obteniendo un promedio de 214719 puntos en las diferentes escenas. Las capturas correspondientes a una de las cámaras se muestran en la Figura 4.19, mientras que el resultado de la reconstrucción 3D se muestra en la Figura 4.20. Se han etiquetado los objetos de la *A* a la *L* para hacer referencia a ellos en los resultados. Los objetos *K* y *L* corresponden a una vasija partida por lo que no son simétricos y se consideran un objeto diferente cuando están girados por lo que se identifican como *K2* y *L2*.

Cabe destacar que el sistema ha sido preparado para enviar por medio de ROS, tanto los resultados obtenidos (nubes de puntos calculadas, geometría de la escena y sistemas de referencia) como estadísticas del sistema. Así, se puede tanto enviar los resultados de la estimación a otros componentes del *framework* como guardar las estadísticas.

Resultados de estimación de la geometría

En primer lugar, en las figuras 4.21-4.25 se muestran los resultados correspondientes a los diferentes métodos de segmentación desarrollados. En este sentido, se ha optado por una representación visual de las estimaciones. Se ha considerado que esta representación permite analizar los resultados cualitativamente de forma más intuitiva que comparando las dimensiones obtenidas. Cada una de las figuras muestra para cada objeto, una escena de UWSim que muestra la reconstrucción 3D y el modelo del objeto estimado. Esta visualización es la misma que se ofrece al usuario del *framework* junto con la especificación del agarre.

Los resultados correspondientes a la segmentación mediante RANSAC se mues-

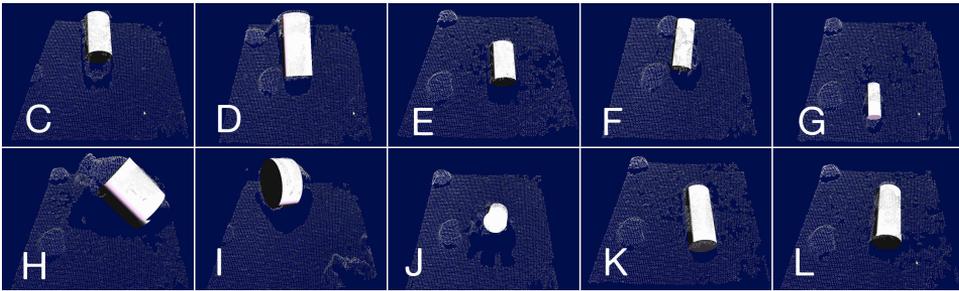


Figura 4.21: Resultados de la estimación utilizando el método RANSAC para cilindros. Conjunto de objetos C-L.

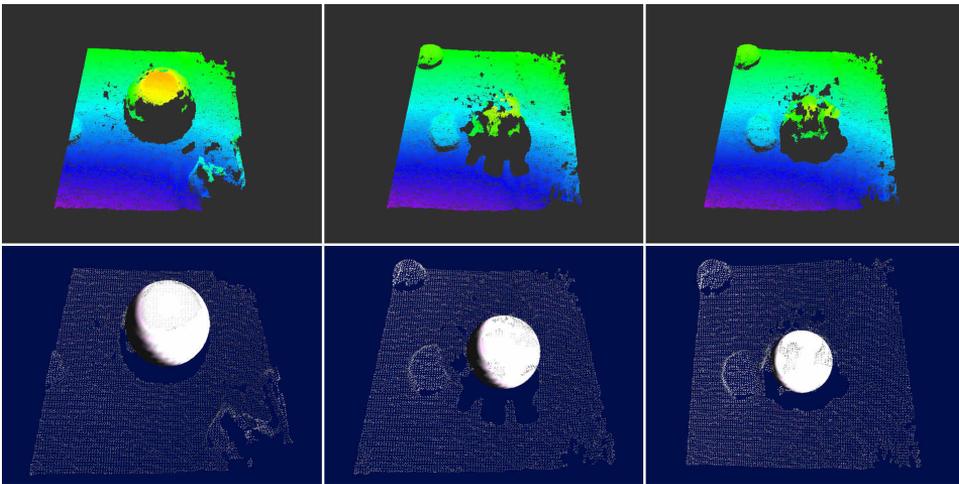


Figura 4.22: Resultados de la estimación utilizando el método RANSAC para esferas. Objetos I y J.

tran en las figuras 4.21-4.23. En el caso de los objetos cilíndricos (Fig. 4.21), se ha aplicado a todos los objetos excepto a las cajas (objetos *A* y *B*). Los resultados de segmentación son excelentes para los objetos cuya forma es más similar a un cilindro. Sin embargo, los objetos *H*, *I*, *J* presentan una forma ligeramente diferente y la reconstrucción obtiene la mejor forma posible. El objeto *J* es el único cuyo modelo es claramente incorrecto e impediría manipularlo correctamente sin la intervención del usuario.

El algoritmo de segmentación de esferas ha sido probado como se ve en la Figura 4.22. Por las características de los objetos, se ha podido probar únicamente con los objetos *I* y *J*. El objeto *I* tiene una forma adecuada cercana a una semiesfera, mientras que en el objeto *J* sus puntos se disponen de forma aproximadamente esférica. En el caso de este objeto, este modelo es más adecuado para obtener una posición de agarre aproximada que el modelo cilíndrico.

Los resultados de segmentación de cajas (Fig. 4.23) mediante extracción de planos con RANSAC muestran la estimación correspondiente a los objetos *A* y *B*, que son los únicos aplicables. Como se puede ver la reconstrucción es bastante acertada,

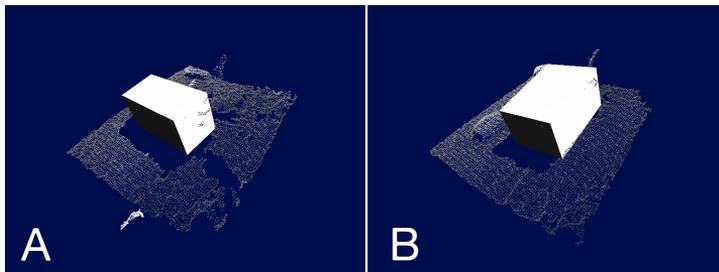


Figura 4.23: Resultados de la estimación utilizando el método RANSAC para cajas. Objetos A y B.

si bien la alineación no es perfecta debido a que el cómputo de la *bounding box* del objeto es bastante sensible al ruido. Por este mismo motivo, el modelo obtenido es ligeramente mayor que el tamaño real del objeto en algunas dimensiones. A efectos de especificación del agarre, esta diferencia supone una mayor apertura inicial del manipulador.

Por otro lado, en la Figura 4.24 se introduce la estimación de modelos mediante el método de SQ. Este algoritmo demuestra su flexibilidad siendo capaz de obtener los parámetros con éxito para todos los objetos. En general, los resultados son excelentes. En particular, los objetos *C* y *D* presentan una forma ligeramente cónica. Por este motivo el algoritmo considera los puntos como una parte de un objeto elíptico mayor. Este problema se resuelve en la fase de especificación al filtrar las posiciones de agarre que no están a una mínima distancia de la nube de puntos real. Esta solución no evita realizar a la generación del agarre con un modelo incorrecto, obteniendo agarres que serán filtrados posteriormente. El objeto *J*, de nuevo, no tiene una forma primitiva clara por lo que se reconstruye de la mejor manera posible. Finalmente, los objetos *K* y *L* se reconstruyen con una forma peculiar debido a que el algoritmo de estimación de simetrías no lidia correctamente con objetos *partidos por la mitad*, que no presentan simetrías o que están parcialmente enterrados.

En la Figura 4.25 se muestran los resultados correspondientes a la estimación mediante momentos de inercia. Al igual que en el caso previo, se ha utilizado el conjunto de datos completo. Este método sirve principalmente para comparar sus resultados con el resto. Al igual que el método de estimación de planos con RANSAC, la *bounding box* obtenida es, en la mayoría de los casos, ligeramente más grande que los objetos por el ruido alrededor de los mismos. Excepto para los objetos *H* e *I*, esta estimación es válida para obtener una posición y orientación de agarre inicial.

A continuación, se desea realizar un estudio más detallado de los resultados en cuanto a la posición de los objetos. Para la realización de estas pruebas se ha obtenido la posición del objeto continuamente a lo largo del tiempo con los algoritmos disponibles. Se ha obtenido un número de muestras similar en ambos algoritmos durante 10 segundos en todos los métodos excepto con SQ, cuyo coste temporal es mucho mayor. En dicho caso se han obtenido muestras hasta disponer de aproximadamente las mismas que en el resto de métodos. Las estimaciones se realizan sobre los mismos datos. Para ello, se utilizan los mismos 10 segundos de

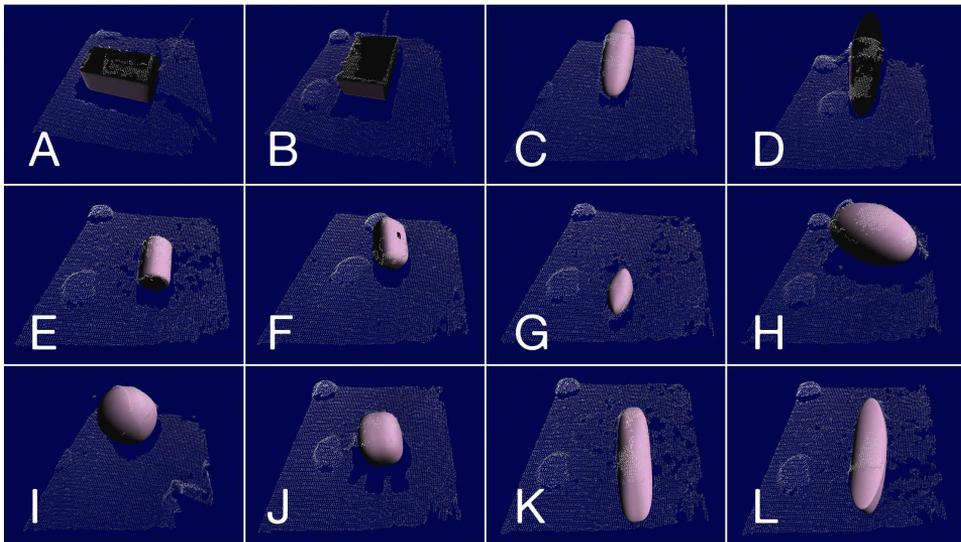


Figura 4.24: Resultados de la estimación utilizando el método SQ. Conjunto de objetos completo.

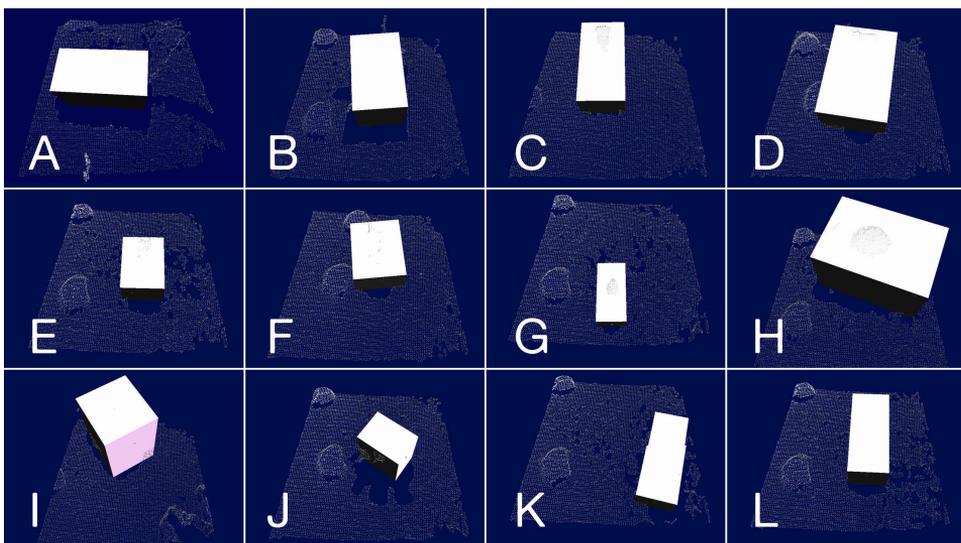


Figura 4.25: Resultados de la estimación utilizando el método PCA. Conjunto de objetos completo.

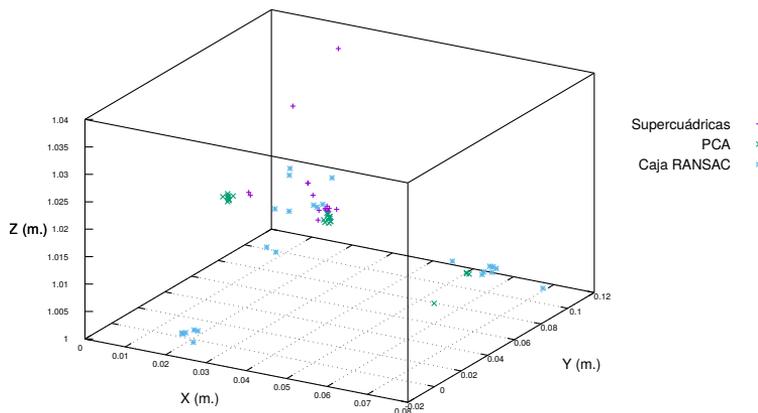


Figura 4.26: Resultados de estimación de la posición del objeto A.

datos de entrada reproducidos de forma cíclica.

En esta ocasión, el análisis de los resultados de los diferentes algoritmos se divide atendiendo a su forma real. Los objetos *A* y *B* son cajas, por lo que los algoritmos utilizados son los basados en estimación de planos con RANSAC, PCA y SQ. En los resultados de la estimación del primer objeto (ver Figura 4.26) se puede apreciar cómo la posición obtenida es más errática que en el segundo (ver Figura 4.27) para el caso de RANSAC. Esto se debe a que en el primero en ocasiones se detecta el lateral de la caja como plano principal, empeorando la detección. En el segundo objeto se puede ver cómo los métodos son mucho más consistentes. Las diferencias de posición entre ellos se deben a la forma de obtener los modelos en cada caso, produciendo leves desplazamientos en el centro de los objetos. En cualquier caso, excepto en la estimación del objeto *A* con RANSAC, los errores son reducidos.

El contexto en el que se sitúa este trabajo es el de la arqueología, y por este mismo motivo la mayoría de objetos utilizados presentan una forma más o menos cilíndrica. A tal efecto, se analizan los resultados de los objetos *D* (ver Figura 4.28) y *E* (ver Figura 4.29). La geometría de estos objetos se ha obtenido mediante RANSAC aplicado a cilindros, PCA y SQ. En el caso de RANSAC, el método consigue estimar la geometría con gran precisión continuamente para ambos objetos. El resultado es más estable para el objeto *D* que para el *E* con RANSAC y PCA porque tiene un mayor número de puntos y una superficie con textura que resulta más fácil de reconstruir. Sin embargo, al estimar el objeto *D* el método SQ sufre del citado problema de estimación que calcula un objeto mayor que el real, con lo que el centro se ve desplazado. De nuevo, las diferencias en posición entre algoritmos para un mismo objeto se deben a que las diferencias en cómo obtienen los modelos producen leves desplazamientos en el centro de los objetos.

Por último, los objetos *I* y *J* son los que presentan una forma más diferente al resto y han sido estimados también con la estimación RANSAC basada en esferas. En primer lugar, el objeto *I* (ver Figura 4.30) presenta notables diferencias entre los diferentes métodos. Esto es así porque cada uno obtiene una forma subyacente

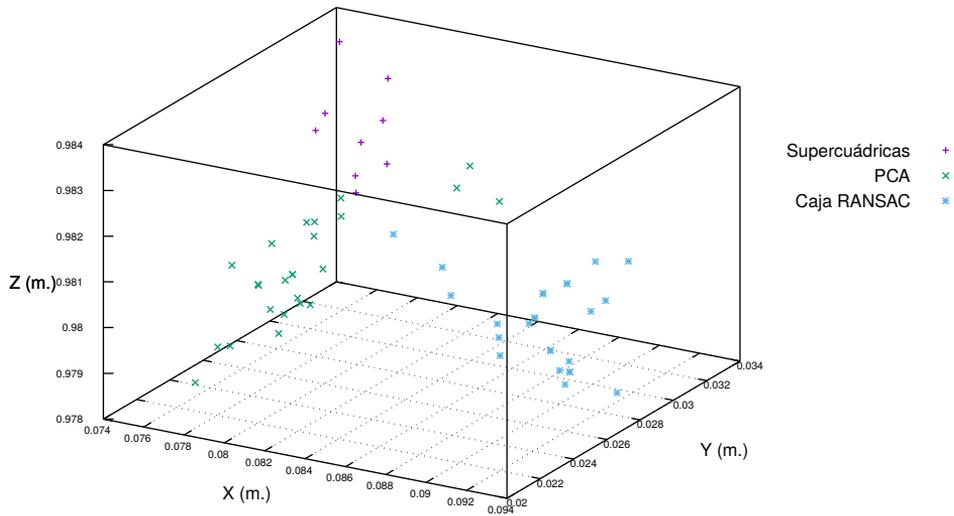


Figura 4.27: Resultados de estimación de la posición del objeto B.

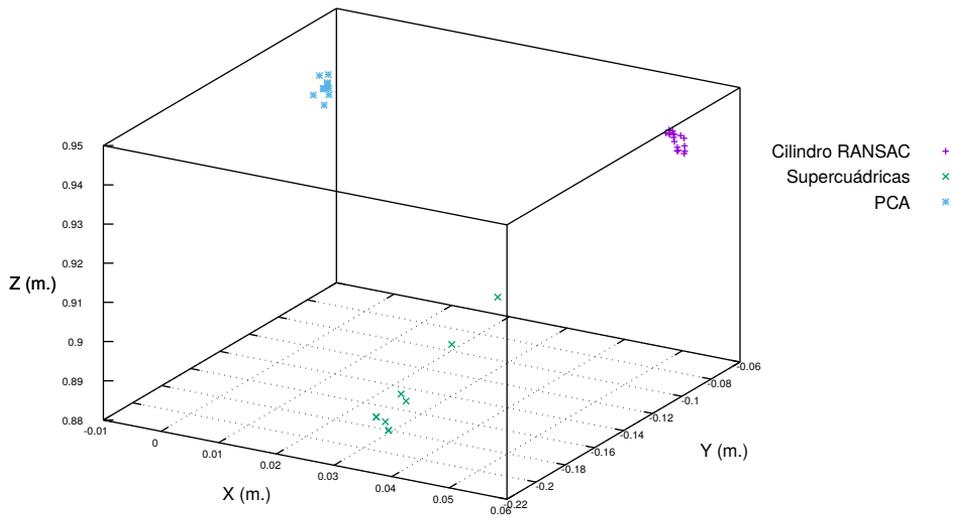


Figura 4.28: Resultados de estimación de la posición del objeto D.

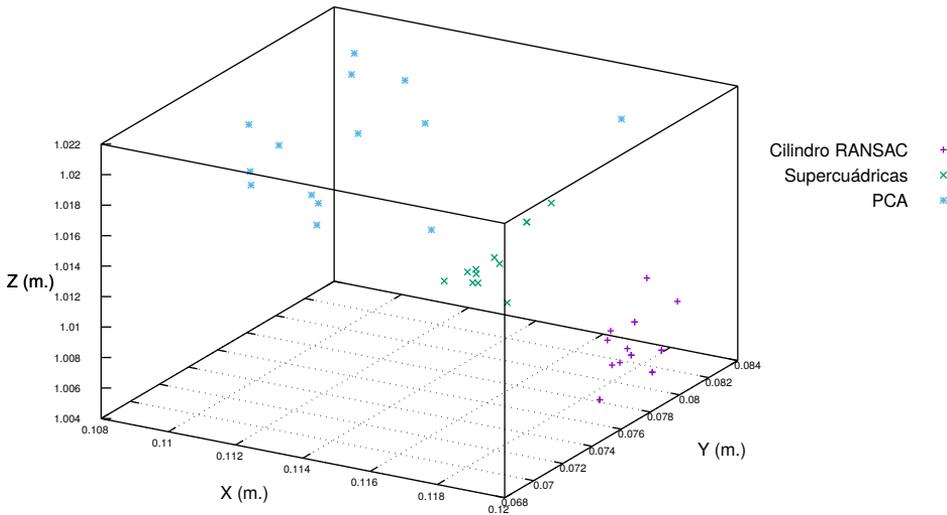


Figura 4.29: Resultados de estimación de la posición del objeto E.

diferente, con lo que el centro se desplaza. Además, el objeto presenta un número de puntos relativamente bajo y una cantidad de ruido notable. Por este motivo todos los algoritmos se ven afectados y en especial PCA, cuya tolerancia al ruido es baja. Por otro lado, el objeto *J* (ver Figura 4.31) se reconstruye de forma muy estable con todos los algoritmos excepto con RANSAC, que no es capaz en ningún momento de obtener correctamente la similitud del objeto con un cilindro y obtiene resultados cambiantes.

En base a estos resultados se concluye que cada algoritmo tiene sus puntos fuertes y débiles. El método basado en SQ es claramente el mejor en un número mayor de situaciones. No obstante, no funciona correctamente en ciertos casos extremos y su principal desventaja es el coste temporal. Los algoritmos que utilizan RANSAC funcionan de forma excelente cuando los objetos que se desea modelar tienen la geometría adecuada. Si no está presente la forma primitiva buscada, el algoritmo suele obtener resultados negativos y en ocasiones es difícil discernir programáticamente si se trata de una mala detección. El método que calcula los momentos de inercia mediante PCA es eficiente pero no es ni tolerante al ruido ni a una segmentación inicial incorrecta. Esta segmentación inicial incluye tanto la segmentación del fondo como el agrupamiento de nubes de puntos para segmentar cada objeto. Por ello, estos pasos son muy importantes para el correcto uso del algoritmo.

Análisis temporal de la estimación

Otro de los factores a analizar es el tiempo de proceso necesario para estimar la geometría de los objetos. En las figuras 4.32 y 4.33 se muestran los tiempos de proceso utilizados en promedio para estimar la geometría de los objetos en 100 iteraciones. Estos datos se presentan en función del número de puntos del objeto una

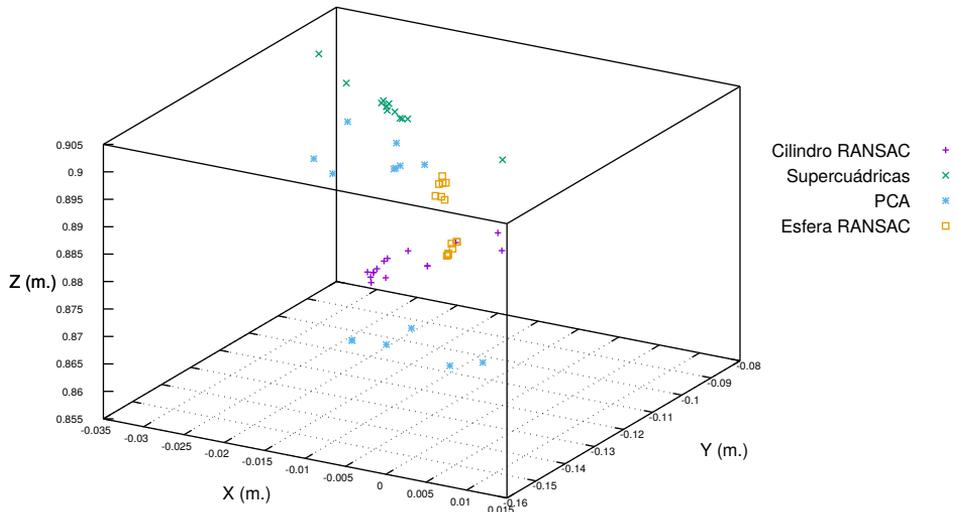


Figura 4.30: Resultados de estimación de la posición del objeto I.

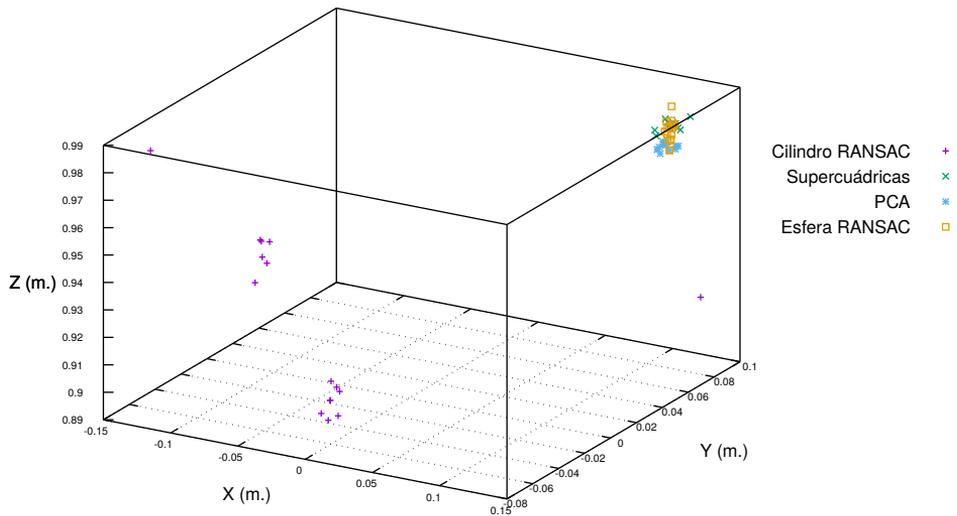


Figura 4.31: Resultados de estimación de la posición del objeto J.

vez segmentado de la escena. Este número de puntos es directamente proporcional al tamaño del objeto y la calidad de su reconstrucción. Esta calidad viene determinada, principalmente, por el tipo de superficie que dispone. Las figuras muestran visualmente la desviación estándar, cuyos valores resultan también interesantes para ver la estabilidad de cada método.

El primer dato destacable que podemos apreciar es la diferencia de magnitud entre el algoritmo basado en SQ, que necesita entre 40 y 240 segundos para procesar una sola estimación; y los tiempos del resto de algoritmos, que varían entre 0,085 y 0,16 segundos (ver Figura 4.33). Este orden de magnitud se debe a que el proceso de optimización que se lleva a cabo en el caso de SQ es mucho más complejo. Por esta razón, este método se puede utilizar una vez al inicio de la misión, pero no es realista realizarlo iterativamente, como sí se puede hacer con el resto de métodos. Asimismo, existe una pequeña diferencia entre los algoritmos de RANSAC para cilindros y esferas (ver Figura 4.32), que oscilan entre los 0,9 y los 0,12 segundos, y los de RANSAC para cajas y PCA, que requieren de 0,1 a 0,16 segundos. Esta diferencia de tiempo viene causada porque ambos métodos realizan adicionalmente un paso de agrupamiento.

Se puede ver que la estimación en todos los algoritmos, especialmente en el de SQ, tiene un coste temporal que aumenta en función del tamaño del objeto. Este dato resulta lógico ya que el número de puntos a procesar es mayor. Se observa, sin embargo, que existen objetos que se desvían ligeramente de esta relación. Si se observan los casos concretos como, por ejemplo, el objeto F con la segmentación RANSAC, se puede ver que los objetos que son más difíciles de segmentar para el algoritmo tardan comparativamente más y los que encajan perfectamente en un modelo tardan comparativamente menos. Esto es especialmente notable en el caso de RANSAC, en el que la velocidad de convergencia es muy importante.

En promedio, las nubes de puntos obtenidas durante los experimentos en el laboratorio cuentan con 214719 puntos. Tras el procesado utilizando un filtro de reescalado *Voxel Grid* con una resolución de 0,01 metros, se obtiene una nube con 11394 puntos en promedio, que acaban siendo 1632 tras la extracción del fondo. La operación carga tiene un coste temporal promedio de 0,024 s., el filtrado 0,022 s., la extracción del fondo 0,088s. y el cálculo de simetrías, con el método sin búsqueda, 0,006 segundos. Estos tiempos son despreciables con respecto a la estimación mediante SQ y son aceptables en el resto de casos, ya que el tiempo de procesamiento total se encuentra entre 0,16 y 0,085 segundos. No obstante, se puede ver cómo el tiempo de segmentación del fondo no es despreciable.

4.10.5. Comparativa de la caracterización del ánfora con RANSAC

Además de los experimentos individuales de segmentación de objetos, se ha comparado y analizado la estimación del cilindro mediante RANSAC en la escena de caracterización del ánfora en simulación y en el laboratorio de forma más detallada. Como se ha descrito previamente, en simulación se utiliza una cámara estéreo de alta resolución a color mientras que en el laboratorio se utiliza la cámara estéreo Videre, de menor resolución en blanco y negro. Esta diferencia propicia diferencias de tiempos como las que se han visto en secciones anteriores.

En la Figura 4.34 se puede ver la posición y orientación estimada del objeto

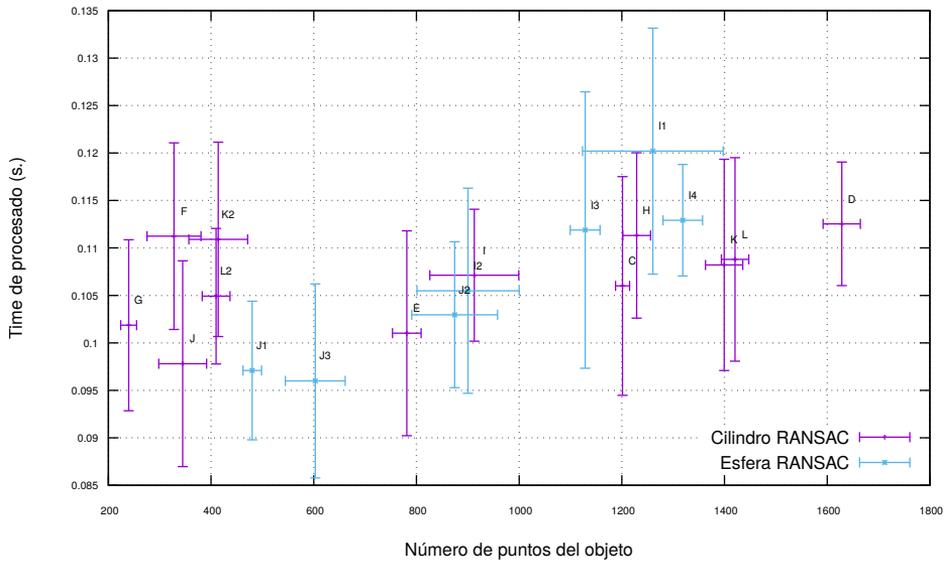


Figura 4.32: Tiempo de procesamiento para los algoritmos de RANSAC para cilindros y esferas.

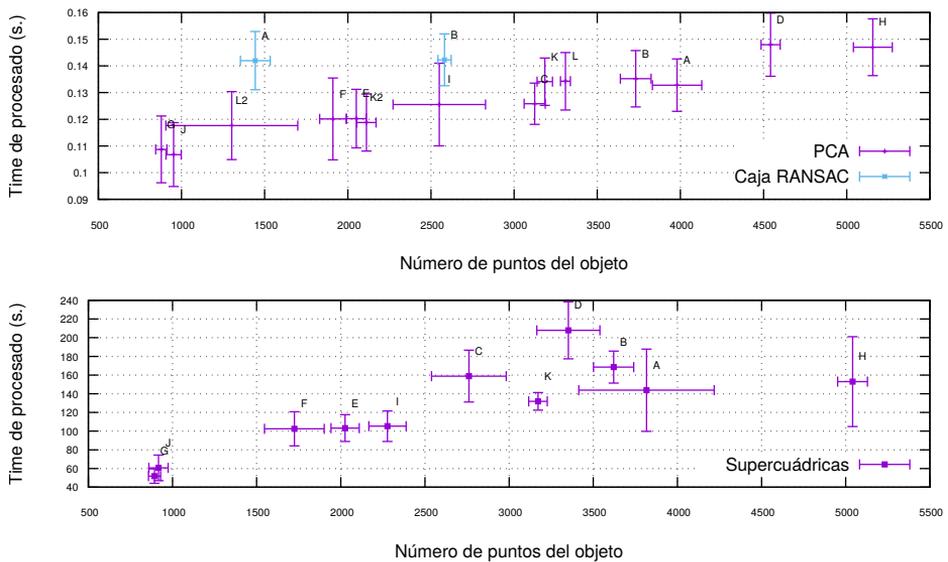


Figura 4.33: Tiempo de procesamiento para los algoritmos de PCA, RANSAC para cajas (arriba), y SQ (abajo).

Tabla 4.5: Comparativa de la estimación con RANSAC en simulación y laboratorio.

	Estéreo simulada	Estéreo real
Número de puntos	220000	140000
Puntos tras reescalado	23000	5000
Puntos tras segmentación	1762	1820
Tiempo medio de carga (ms.)	230	83
Tiempo medio de reescalado (ms.)	35	21
Tiempo de segmentación (ms.)	56	94
Tiempo de segmentación continua (ms.)	23	21
Tiempo total (ms.)	285	125

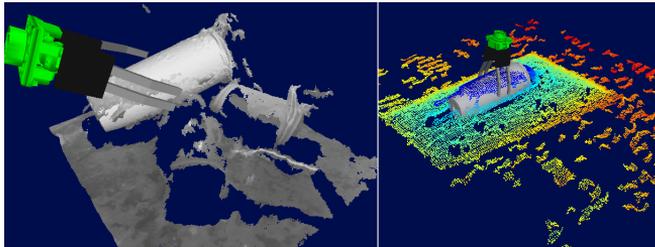


Figura 4.34: UWSim mostrando la detección del cilindro mediante RANSAC con un modelo del objeto (blanco) y la posición de agarre por defecto. Nubes de puntos de la cámara estéreo real (izquierda) y simulada (derecha, componente z codificada por color).

para la escena simulada y el laboratorio, así como las posiciones de agarre resultantes. En la Tabla 4.5 se muestra una comparativa del promedio de tiempos de carga, reescalado y segmentación para múltiples ejecuciones del algoritmo, diferenciando entre la primera segmentación y las siguientes, que realizan el seguimiento del objeto. Constatando los datos expuestos en secciones anteriores, el tiempo de carga es rápido con imágenes reales de la cámara Videre, de baja resolución, mientras que con las cámaras virtuales la resolución y el tiempo de computación son mayores [Fornas et al., 2015].

Este problema se puede minimizar reduciendo la resolución de las imágenes en el controlador a bajo nivel (utilizando *binning*, por ejemplo) o reduciendo la resolución de la nube de puntos antes de realizar la estimación (*downsampling*). Sin embargo, como se puede ver en la tabla, el algoritmo de reescalado también tiene un coste temporal significativo y no reduce los tiempos de carga de la nube de puntos.

En este experimento se ha evaluado también la estimación continua de la posición para realizar un seguimiento (*tracking*) del objeto mediante la segmentación del cilindro RANSAC. Se puede ver que aunque el tamaño de la nube de puntos simulada duplica a la real, el tiempo de segmentación en real es mayor. Esto se debe a que el algoritmo necesita un mayor número de iteraciones en presencia de ruido y puntos que no pertenecen al objeto (*outliers*). Sin embargo, las segmenta-

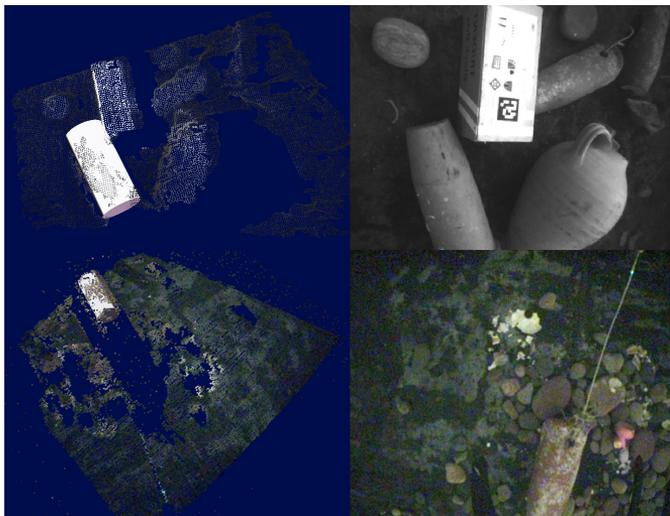


Figura 4.35: Segmentación mediante RANSAC de un objeto cilíndrico con la cámara DUO M (arriba) y la reconstrucción láser (abajo).

ciones que se realizan después de la primera son más rápidas en ambos escenarios. Como se ha comentado, el tiempo de carga es el cuello de botella del sistema para este algoritmo, ya que tarda 282 y 83 ms. en cargar la nube, comparado a los 23 y 21 ms. que tarda en segmentarlas. En base a estos resultados, se puede ver que el proceso puede realizar la estimación del objeto con una frecuencia media de 4Hz. en simulación y 8Hz. en el escenario real.

4.10.6. Estimación utilizando otros sensores

A parte de los métodos de obtención de nubes de puntos detallados en las secciones previas, se ha utilizado la cámara estéreo Code Laboratories DUO M para mostrar el funcionamiento del sistema de detección utilizando otras fuentes de información 3D. Como se puede ver en la Figura 4.35, el sistema es capaz de detectar objetos cilíndricos en una nube de puntos proveniente de este sensor. Con todo, no ha sido posible utilizar los métodos que hacen uso de PCA y SQ con estos datos debido a que la calidad de la nube de puntos es insuficiente. Es por ello que estos algoritmos no se pueden utilizar con sensores que no cumplen un mínimo de calidad, como en este caso. En este sentido, también se ha utilizado el métodos de estimación basado en RANSAC sobre nubes de puntos obtenidas a través de segmentación de láser con las técnicas desarrolladas en el IRS Lab [Prats et al., 2012a, Peñalver et al., 2017]. No obstante, estas nubes de puntos contienen igualmente gran cantidad de ruido y tampoco se han podido aplicar los métodos PCA ni SQ.

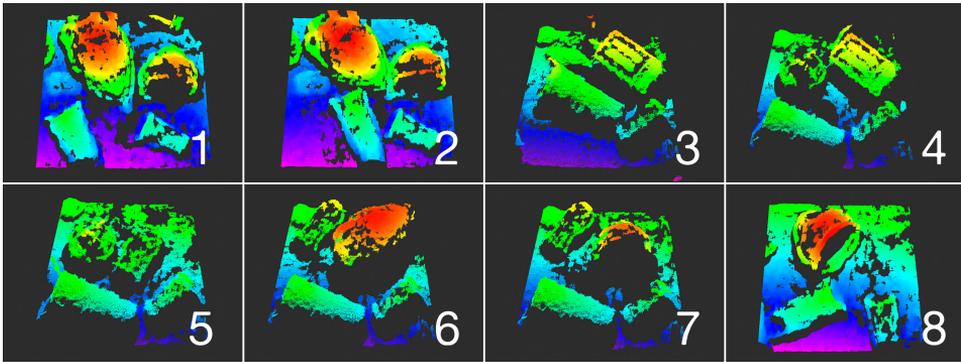


Figura 4.36: Reconstrucción 3D de las escenas con múltiples objetos.

4.10.7. Escenas de caracterización de múltiples objetos

Para evaluar la capacidad de los algoritmos para obtener la geometría de una escena completa se ha realizado la estimación sobre escenas con múltiples objetos. Los experimentos se han realizado en el laboratorio del IRS Lab con la cámara estéreo Videre. En cada escena se han dispuesto de forma arbitraria diferentes objetos de los presentados previamente. Además, estas escenas han sido registradas a diferentes distancias para comparar los resultados, puesto que la precisión de la reconstrucción es dependiente de la distancia. Como referencia, la Figura 4.36 muestra la reconstrucción 3D de ocho de las escenas utilizadas para demostrar la estimación con la cámara estéreo Videre.

Para realizar la estimación de escenas completas, se ha seleccionado el algoritmo de SQ por su demostrada flexibilidad. Los métodos basados en RANSAC son efectivos para modelos concretos, mientras que la técnica de PCA ha sido utilizada en este trabajo para comparar resultados y para obtener una estimación alternativa que pueda ser utilizada para la especificación del agarre. Así, tras la adquisición, reconstrucción y filtrado se inicia la estimación. En este caso, el algoritmo de SQ se aplica iterativamente a grupos de puntos o clústeres suficientemente grandes.

Los resultados del proceso se muestran en la Figura 4.37. En general, estos resultados son muy satisfactorios en el sentido que logran condensar la información de la escena de forma adecuada. No obstante, en el caso de algunos objetos concretos la estimación es mejorable. Por ejemplo, se ha visto en los resultados del laboratorio como el objeto D se reconstruye con una forma mucho mayor que la real. Puesto que la técnica aplicada es la misma, estos efectos se ven reflejados en la estimación de la escena completa. Finalmente, cabe destacar que el tiempo de ejecución viene determinado por el número de objetos de la escena, siendo el tiempo de estimación en promedio de un único objeto de 97,45 segundos en los escenarios del laboratorio, que utilizan la misma cámara.

Los resultados de los métodos de estimación para obtener la posición, orientación y geometría de los objetos han sido demostrados tanto de forma visual con información intuitiva, como de forma cuantitativa. De esta forma, se ha podido evaluar el principal elemento presentado en este trabajo. Los resultados son satisfactorios en el sentido de que se ha podido estimar correctamente múltiples objetos

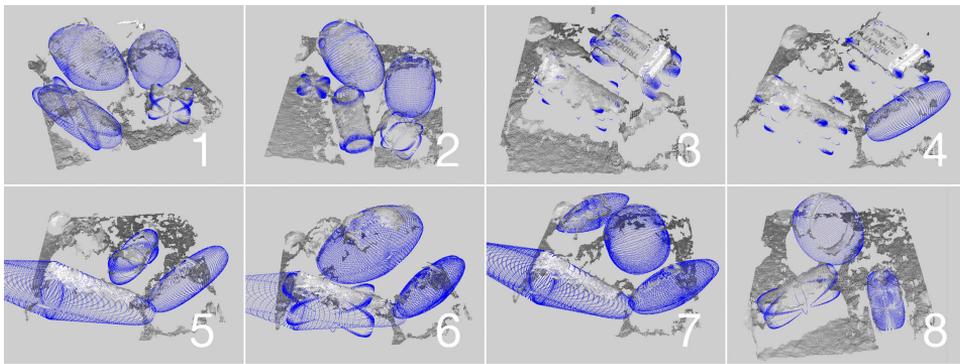


Figura 4.37: Resultados de las escenas con múltiples objetos reconstruidas con SQ.

en diversas condiciones.

Gracias a ello, ha sido posible conocer los puntos fuertes y debilidades de los algoritmos introducidos en este *framework*. En el Capítulo 5 se detalla el proceso de especificación del agarre partiendo de la información generada en este capítulo. Además, el siguiente capítulo también muestra el sistema HRI que permite al usuario iniciar la estimación, obtener información sobre la estimación y supervisar las tareas de especificación y ejecución.

Especificación del agarre

Una vez se ha obtenido la estimación de la geometría de la escena de intervención se puede iniciar la manipulación del objeto. En este capítulo se describe la metodología de especificación del agarre que supone el paso previo a la ejecución. En primer lugar se explica la motivación, el estado del arte y la metodología aplicada para especificar la posición y orientación que debe alcanzar el manipulador para alcanzar el objeto. A continuación, se detalla la HRI que permite supervisar y monitorizar la intervención en todo momento.

La especificación del agarre se realiza de forma supervisada. El sistema genera autónomamente un agarre por defecto, ya sea de forma heurística o analítica, para que el usuario decida utilizarlo o modificarlo. De esta manera, es el usuario en última instancia el que decide. Se introduce el estado del arte en HRI en entornos submarinos y la relación de la interfaz con el desarrollo del proyecto MERBOTS. Finalmente, se muestra en detalle la interfaz desarrollada, sus diferentes posibilidades de uso y los resultados obtenidos.

5.1. Motivación

En el entorno submarino, las operaciones de intervención representan un auténtico desafío con la tecnología actual. En particular, el desarrollo de sistemas completamente autónomos es, por el momento, una utopía. Como se ha visto en el Capítulo 1, los últimos trabajos realizados proponen estrategias dónde el control explícito del robot no es necesario como en un ROV. Por el contrario, se ofrecen funciones autónomas a diferentes niveles, desde un simple control en profundidad al guiado del robot mediante puntos de referencia en 3D. En el ámbito de la especificación del agarre contar con la supervisión de un usuario es de un gran interés, ya que permite dotar de robustez a este sistema. Esta ayuda es importante ya que se trata de manipular objetos cuya geometría exacta no es conocida a priori. En ocasiones, la especificación generada de forma autónoma puede no ser adecuada y su ejecución podría dañar los objetos. Estos objetos, además, pueden tener un valor arqueológico importante. La aproximación propuesta en este capítulo incide en estos aspectos. La estimación de la geometría de la escena se realiza de forma autónoma una vez el usuario ha seleccionado el método más adecuado. Análogamente,

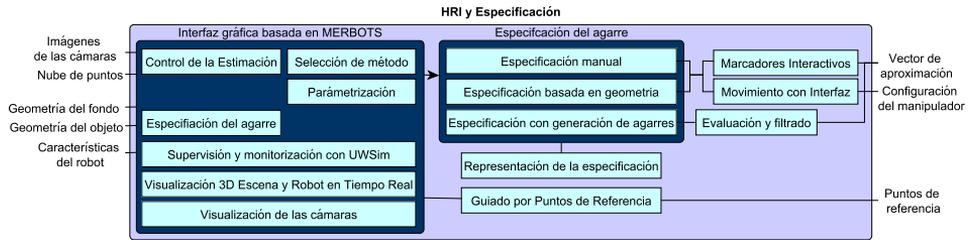


Figura 5.1: Componentes del módulo de Especificación.

el sistema realiza la especificación del agarre de forma autónoma, para después ofrecer la posibilidad al usuario de mejorar la especificación del agarre. Posteriormente se utiliza la interfaz para monitorizar su ejecución.

5.2. Especificación del agarre

El resultado del módulo de Estimación detallado en el Capítulo 4 es la geometría de la escena, que incluye el modelo aproximado, posición y orientación de cada uno de los objetos en la escena, así como información sobre el plano del fondo. A partir de esta información, en esta sección se describe cómo se obtiene la posición de agarre que deberá ejecutar el robot. Este proceso se denomina especificación del agarre. Los componentes de este módulo se muestran en la Figura 5.1.

La metodología de especificación propuesta se descompone en dos fases. Inicialmente se realiza la especificación del agarre de forma autónoma. En un segundo paso, la interfaz permite al usuario modificar dicha posición de agarre de acuerdo a sus conocimientos o crear una nueva más adecuada.

La especificación del agarre autónoma consiste en la determinación de la posición y orientación del efector final que permiten al robot manipular el objeto. Para determinarla se propone un método heurístico basado en la geometría del objeto y un modelo analítico en el cuál se utiliza el modelo obtenido para computar el mejor agarre generando un modelo 3D del objeto a partir de su geometría. En ambos métodos se propone un procedimiento de ordenación de agarres que permite la selección del mejor. Las limitaciones cinemáticas del brazo hacen que sea necesario filtrar estos agarres para determinar si se pueden alcanzar con la posición y orientación deseadas.

5.2.1. Especificación heurística

La especificación del agarre se realiza en primera instancia utilizando la geometría del objeto obtenida en el paso de estimación. Debido a que la manipulación de una vasija es el caso de uso básico de este trabajo, se describe en primer lugar la determinación de la posición de agarre para un objeto con forma cilíndrica. Como se ha mencionado anteriormente, el resultado del procedimiento de especificación es un vector de aproximación para el robot, es decir, una posición y orientación que el efector final debe alcanzar antes de empezar a cerrar las pinzas. Así, se desea obtener la posición de agarre (g , *grasp*) con respecto al sistema de coordenadas del

mundo(w , *world*): wMg . La matriz wMo representa la posición y transformación desde el mundo al centro del modelo del objeto, con el eje y alineado con el eje del cilindro y el eje z calculado de forma que se minimiza el ángulo con la normal del plano del suelo para dirigir el eje z hacia el suelo. Esta transformación se obtiene en el paso de estimación. Para generar el mejor agarre se obtiene la transformación entre el objeto y el agarre de forma heurística utilizando la dirección del eje del objeto, su centro y su radio. Esta transformación consiste en una traslación d a lo largo del eje y de wMo (eje del cilindro), una rotación α con respecto al eje x de wMo y un desplazamiento r a lo largo del eje z de wMo . Esta transformación se puede resumir como:

$$oMg = Trans_y(d) * Rot_x(\alpha) * Trans_z(r) \quad (5.1)$$

$$wMg = wMo * oMg \quad (5.2)$$

El desplazamiento r está basado en la longitud de las garras del manipulador y el radio del objeto, de forma que se obtenga una penetración suficiente en el objeto para manipularlo de forma satisfactoria. Este procedimiento se lleva a cabo de la misma manera con modelos esféricos, pero en este caso el desplazamiento d es 0. Es decir, únicamente se realiza la rotación y traslación hacia el exterior de la esfera según su radio:

$$oMg = Rot_x(\alpha) * Trans_z(r) \quad (5.3)$$

En el caso de la detección de objetos mediante la segmentación de un plano, los momentos de inercia o las supercuádricas, se toma como eje para el desplazamiento d el eje mayor del objeto de forma que la distancia máxima que se puede recorrer es la longitud de este eje. Después, la distancia al objeto viene determinada por la altura del mismo y la apertura de la mano θ_g por su anchura. Además, en el caso de la detección mediante supercuádricas, la anchura del objeto puede variar a lo largo del eje mayor o de máxima inercia.

Este procedimiento heurístico determina la posición de agarre en función de las variables descritas $oMg = F(d, \alpha, r)$. En la práctica, se ha determinado el valor de (d, α, r) de forma experimental, cuyo valor por defecto es (20cm., 75grados, 18cm.). Se han ejecutado agarres con estos parámetros con éxito en el laboratorio, la piscina del CIRS y en el mar. Sin embargo, se ha desarrollado un método para la generación y clasificación de posiciones de agarre a partir de esta función.

5.2.2. Generación y Clasificación de Hipótesis de agarre

Utilizando la función $oMg = F(d, \alpha, r)$ descrita, se define un método para generar de forma iterativa una serie de hipótesis de agarre. Este procedimiento heurístico es específico del manipulador ARM5E y el objetivo es encontrar la mejor posición de agarre posible teniendo en cuenta las limitaciones cinemáticas del manipulador. Este hecho se debe a que generalmente no es posible alcanzar las posiciones de agarre deseadas con la orientación deseada. Este algoritmo se trata de encontrar la mejor posición de agarre alcanzable por el manipulador. Para ello, se da valor a d a intervalos de K_d milímetros, a α a intervalos de K_α radianes y a r a intervalos de K_r

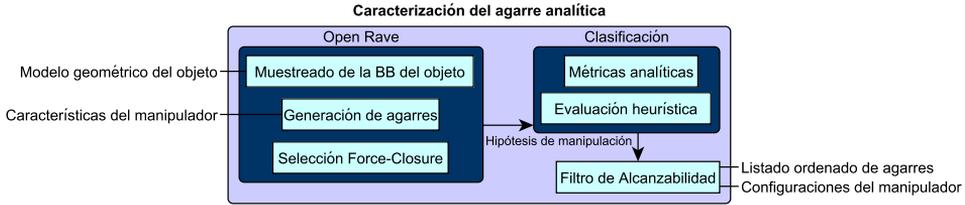


Figura 5.2: Componentes de la especificación analítica del agarre.

milímetros. Para cada una de las hipótesis de agarre obtenidas (oMg_{hp}) se calcula la alcanzabilidad de esta posición. Utilizando la cinemática del robot se obtiene la posición y orientación más cercana que es alcanzable por el robot (oMg_{ik}). Para realizar la ordenación de los agarres se determina la puntuación del agarre (S) de la siguiente manera: $S = C_1 * IK_{dist} + C_2 * IK_{ang} + C_3 * STD_{dist} + C_3 * STD_{ang}$. En esta fórmula se puede ver cómo se tienen en cuenta los siguientes criterios:

1. Distancia entre la posición generada y la alcanzable: $IK_{dist} = d(oMg_{hp}, oMg_{ik})$. Es el criterio más importante pues en ocasiones el robot no puede alcanzar una posición cercana a la generada y no tiene sentido utilizarla.
2. Distancia angular entre la orientación generada y la alcanzable: $IK_{ang} = angle(oMg_{hp}, oMg_{ik})$. Permite determinar una variación en orientación del manipulador aceptable.
3. Distancia con una posición preferida (STD_{dist}): utilizando la experiencia previa se determina que el manipulador es capaz de coger los objetos mejor formando un ángulo aproximado de 55 grados con el eje mayor del objeto cuando éste es paralelo al suelo. De esta forma el manipulador está alejado de sus límites articulares y se controla mejor.
4. Distancia angular con una orientación preferida (STD_{ang}): Debido a la longitud del efector final, se determina una distancia preferida del sistema de referencia del efector final al centro del objeto de forma que no esté demasiado cerca ni demasiado lejos para el manipulador en el momento del cierre.

Las constantes (C_1, C_2, C_3, C_4) determinan el peso de cada criterio en la clasificación. Los valores han sido obtenidos de forma experimental y son los siguientes: ($C_1 = 100, C_2 = 10, C_3 = 2, C_4 = 6$). La diferencia entre las distancias y los ángulos se deben a que se trata de magnitudes diferentes. La diferencia entre los criterios cinemáticos y los de preferencia se deben a que tiene una mayor importancia que la posición alcanzable sea similar a la posición deseada. Utilizando esta ordenación, el usuario dispone de un listado de hipótesis de agarre ordenadas de forma que puede tomar la mejor posición y disponer de alternativas de calidad. Este listado de agarres se expone al usuario utilizando la HRI. En la sección siguiente se describe el método de análisis de agarres basado en SQ y, posteriormente, las alternativas que tiene el usuario para configurar la posición de agarre.

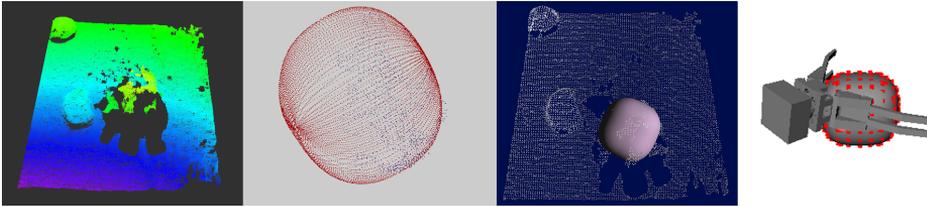


Figura 5.3: Especificación analítica del agarre con estimación de SQ (izquierda, centro). Análisis del agarre con OpenRave (derecha).

5.2.3. Especificación analítica del agarre

La especificación heurística utiliza el conocimiento experimental de cómo el brazo es capaz de coger el objeto, es decir, no se realizan cálculos sobre las superficies de contacto entre el objeto y el manipulador. En la práctica, esta metodología funciona debido a la introducción del usuario en el bucle de control, que es capaz de detectar cuándo un agarre no se ha calculado de forma adecuada. No obstante, se propone una caracterización avanzada que es más exhaustiva aunque tiene un coste computacional mayor. Así, en esta sección se describe la caracterización del agarre de forma analítica. Después de generar una serie de hipótesis de agarre utilizando modelos 3D, se validan los agarres con la condición de *force-closure* [Bicchi y Kumar, 2000]. Esta condición consiste en que la configuración de agarre obtenida debe restringir el movimiento del objeto en cualquier dirección, siendo capaz de resistir fuerzas externas [Antonelli et al., 2008]. Se calcula obteniendo la intersección de los conos de fricción entre el manipulador y el objeto en los puntos de contacto. Este cálculo resulta esencial en el uso de manipuladores diestros para obtener agarres de forma analítica [Bohg et al., 2014]. Las hipótesis obtenidas se pueden ordenar haciendo uso de métricas de calidad del agarre [Rubert et al., 2018]. Adicionalmente, en el caso del manipulador ARM5E, se deben filtrar las hipótesis mediante la función presentada en la sección previa.

A diferencia del método heurístico, en este caso se utilizan modelos 3D de los objetos y del efector final. Los modelos del objeto se obtienen utilizando el método de estimación basado en SQ. Además del modelo 3D del efector final, se utiliza su modelo cinemático. La estructura de la caracterización avanzada del agarre se muestra en la Figura 5.2. En resumen, se obtiene el modelo 3D del objeto, se muestrea su superficie para después obtener las hipótesis de agarre que cumplan la condición de *force-closure*, se evalúan los resultados utilizando métricas y se filtran los resultados teniendo en cuenta la alcanzabilidad. En la Figura 5.3 se muestran visualmente los pasos que se siguen en este proceso, desde la obtención del modelo mediante SQ, al resultado final de un agarre que cumple las propiedades deseadas, pasando por el muestreo de agarres, representado con flechas rojas sobre el objeto.

En primer lugar se utiliza la geometría de los objetos obtenidos en el capítulo anterior con el método de estimación de SQ para obtener una malla o *mesh* en un formato como ".ply", ".wrl" o ".stl". Además, se dispone de un modelo del efector final del manipulador ARM5E. Este modelo describe la estructura cinemática del manipulador, así como las mallas 3D que lo componen. Para determinar si un agarre

es adecuado o no, se realiza un análisis de los puntos de colisión entre las mallas correspondientes a los objetos y al efector final.

En segundo lugar, se muestrea la superficie del objeto para generar infinidad de posibles agarres (ver Figura 5.4). Para ello se calcula la **bounding box** del modelo 3D utilizado. A continuación se obtienen puntos en su superficie. El muestreo de los parámetros se puede configurar para obtener una mayor granularidad utilizando los siguientes conjuntos de parámetros:

- Número de hipótesis: la combinación de parámetros configurada genera un número determinado de posibles agarres. Después cada una de estas se comprueba hasta obtener el número de hipótesis deseadas que cumplen la condición de *force-closure*. El coste computacional del método aumenta de forma proporcional a este parámetro.
- *Anglerange*: permite controlar la orientación del manipulador con respecto al objeto en cada una de las posiciones. La mejor dirección de agarre no va necesariamente en dirección al centro del objeto, sino que puede ser necesario manipularlo con un determinado ángulo con respecto a su eje. Este parámetro permite especificar un rango orientaciones del manipulador y la resolución angular de muestreo, es decir, el ángulo entre las orientaciones de agarre evaluadas. Genera un mayor número de hipótesis iniciales. Sin embargo, no tiene un impacto en el coste computacional ya que se comprueban muestras hasta disponer del número de hipótesis deseado.
- *Deltaspac*: permite controlar la resolución espacial de muestreo, es decir la separación entre las posiciones de agarre evaluadas. Así, una menor distancia de muestreo genera un número mayor de muestras. Tampoco tiene impacto en el coste computacional.
- *Roll*: permite generar hipótesis de agarre con diferentes rotaciones en cada uno de los puntos muestreados. Se definen las rotaciones a evaluar a partir del ángulo mínimo, el ángulo máximo y la resolución angular que se desean utilizar.
- *Standoffs*: es un vector de distancias de partida al objeto. Con este parámetro es posible obtener agarres a diferentes distancias del objeto, pudiendo ejercer una fuerza mayor o menor dependiendo del tamaño del objeto.

Para cada una de las hipótesis de agarre generadas, el motor de OpenRave para *grasping* [Diankov, 2010] mueve el efector final hacia el objeto hasta que entra en contacto. Después, cierra el efector final si puede y obtiene los puntos de contacto. A continuación, el sistema determina si se trata de agarres con la propiedad de *force-closure* calculando los conos de fricción correspondientes. En la Figura 5.4 se muestran los conos de fricción para un agarre concreto. El proceso completo se ilustra en la Figura 5.5 utilizando un manipulador diestro. Dicho proceso se simplifica en este trabajo por no disponer de dedos articulados.

Una vez se han obtenido las hipótesis de agarre que cumplen la condición de *force-closure*, se propone usar el método propuesto por [Rubert et al., 2018] para obtener diversas métricas que permiten realizar una ordenación preliminar de las

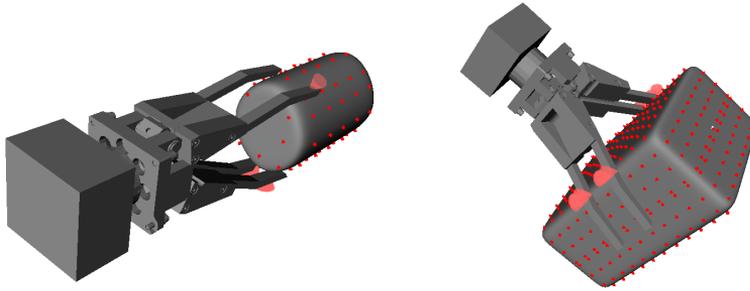


Figura 5.4: Muestreo de la superficie de los objetos y conos de fricción en los puntos de contacto.

hipótesis de agarre. Para calcular las métricas de agarre hace uso del *framework* OpenGrasp [Ulbrich et al., 2011], desarrollado sobre OpenRave [Diankov, 2010].

Las métricas propuestas en [Rubert et al., 2018] se orientan al uso de manipuladores diestros con múltiples articulaciones. Algunas métricas muy interesantes e intuitivas no se pueden calcular para un manipulador de 1 GDL, mientras que otras tienen poco sentido para dichos manipuladores. Por este motivo, se han utilizado las métricas Q_{A1} , Q_{A3} y Q_{B1} , que son las más útiles e intuitivas en este contexto. A continuación se describen dichas métricas, utilizadas para ordenar los agarres y obtener más información sobre ellos:

- Q_{A1} , Menor valor singular de G [Li y Sastry, 1988]: esta métrica mide cómo de lejos está la configuración de agarre (G) de una configuración singular, en la que se pierde la capacidad de soportar perturbaciones externas.
- Q_{A3} , Índice de isotropía del agarre [Kim et al., 2001]: esta métrica se utiliza para obtener una contribución de fuerzas uniforme entre las fuerzas aplicadas al objeto en cada punto de contacto.
- Q_{B1} , Distancia entre el centroide del polígono de contacto y el centro de masas del objeto [Ding et al., 2001]: esta distancia trata de minimizar los efectos gravitacionales y las perturbaciones durante el movimiento del robot. Por otro lado, la fricción con el agua es un factor muy importante a la hora de transportar un objeto en este medio. El uso de esta métrica ayuda también a limitar la influencia de dicha fricción durante el movimiento. Por ello, y debido a que es muy intuitiva, es la métrica más interesante del conjunto.

Así pues, se calculan las métricas propuestas para cada una de las hipótesis de agarre que cumplen la condición de *force-closure*. Las métricas propuestas se normalizan en el rango de 0 a 1, de forma que se pueden comparar fácilmente.

El resultado de este proceso es un conjunto de hipótesis de agarre válidas con sus métricas correspondientes. No obstante, estas hipótesis están descontextualizadas, ya que no tienen en cuenta la posición del objeto ni el manipulador en el escenario. Es decir, no se tienen en cuenta si la posición de agarre está libre de obstáculos, si está bajo el objeto o si la posición es alcanzable. Como se ha visto en los resultados de estimación, es importante eliminar aquellos agarres que estén

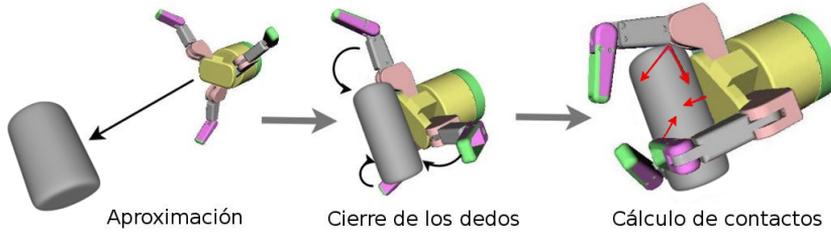


Figura 5.5: Fases para la obtención de hipótesis de agarre en OpenRave con un manipulador diestro.

en zonas del objeto que pese a ser parte del mismo no tienen puntos cercanos. Esto se debe a que la generación de SQ en algunos casos hace que el modelo represente algunas partes del objeto que no existen por imprecisiones en la forma o por ser el objeto completo realmente imposible de representar con un modelo de SQ. Por estos motivos, y debido a que se utiliza el manipulador ARM5E, tras este paso se realiza una clasificación de los resultados con la función que se ha descrito en la sección previa. Las hipótesis de agarre resultantes de este proceso se muestran al usuario a través de la HRI, que se describe en secciones siguientes.

5.3. Interfaces en la robótica submarina

Existe un gran número de misiones que puede llevar a cabo un robot de intervención submarino, en las citadas áreas como el petróleo y el gas, misiones de rescate, arqueología o biología. Estas operaciones han sido llevadas a cabo tradicionalmente con ROV, los cuáles utilizan una arquitectura maestro/esclavo que descarga toda la responsabilidad en el piloto, que sufre una gran fatiga cognitiva y estrés [Sheridan, 1992]. Por esta razón la HRI juega un papel muy importante, ya que es necesario aunar múltiples capacidades, desde la percepción hasta el control de las acciones del robot, de forma que el usuario sea capaz de llevarlas a cabo y entenderlas de manera intuitiva.

La evolución de los ROV hacia los AUV sacó al piloto del bucle de control directo de forma natural. Así, sus responsabilidades pasaron a ser las de supervisión y especificación. Los I-AUV, como nueva evolución de los AUV con capacidades de manipulación, abrieron nuevas posibilidades de interacción con el medio: operación de una válvula, manipulación de objetos, recuperación de muestras, etc. Aún así, ambos sistemas siguen teniendo problemas a la hora de controlar los robots y la manera que el usuario interactúa con el sistema. El piloto de ROV utiliza diferentes pantallas y mandos con múltiples botones para controlar el vehículo, lo que supone una desventaja en el sentido de la HRI. En ambos casos el uso de interfaces gráficas de usuario o GUI para controlar el robot se centra en usuarios expertos, que deben conocer tanto el robot como la misión a realizar perfectamente, debido a la complejidad de las pantallas de información. Así, el piloto de ROV debe prestar atención simultánea a las diversas pantallas y paneles de control para obtener la información sobre el robot, los sensores y las cámaras. Al mismo tiempo debe reaccionar a cualquier imprevisto. En el caso de los AUV, las interfaces de usuario

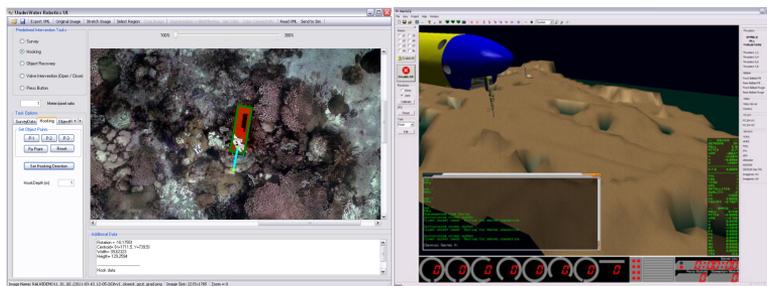


Figura 5.6: Interfaces gráficas utilizadas en los proyectos TRIDENT y SAUVIM.

son también complejas. El problema principal es la falta de retroalimentación al usuario, ya que no disponen de una conexión física con la superficie, y por tanto con el usuario. Fuera del contexto de las intervenciones submarinas, en [Chavez et al., 2017] se propone un interesante sistema de HRI mediante el cuál un AUV sigue a un buceador. Mientras el usuario imparte órdenes gestuales, el robot muestra información al usuario en una pantalla en tiempo real. Los usuarios de AUV, por el contrario, no pueden tener información presencial de la misión.

Los trabajos más relevantes en el contexto de HRI para robots subacuáticos se han realizado en proyectos de investigación como DVECS (*Distributed Virtual Environment Collaborative Simulator*) [Choi y Easterday, 2001], SAUVIM [Marani et al., 2009], RAUVI [Prats et al., 2012d] y PANDORA [FP7-PANDORA, 2015] (ver 5.6). En general, las interfaces de DVECS y SAUVIM fueron desarrolladas considerando un usuario experto. Estas interfaces eran complejas, dependientes del *hardware* concreto del robot y no presentaban ningún tipo de conexión con elementos hápticos o de inmersión. Por otro lado, TRITON [Peñalver et al., 2015], TRIDENT [Sanz et al., 2012] y PANDORA tratan de diseñar un sistema de HRI que pudiera ser utilizado por usuarios no expertos. Sin embargo, el mayor esfuerzo se realizó en el proyecto RAUVI, donde la interacción se diseñó teniendo en cuenta que el sistema pudiera ser utilizado por un usuario no experto [García et al., 2010a] y no depende de una plataforma robótica concreta. El robot OCEAN ONE [Khatib et al., 2016] desarrolló el concepto de avatar robótico teleoperado mediante visión e interfaces hápticas avanzadas como la que se muestra en la Figura 5.7. En esta figura se puede observar un centro de control de usuario desde dónde los AUV pueden ser controlados, utilizando información de sensores y otras fuentes de información. El enfoque en este caso es el de ofrecer una interfaz de control háptica con retroalimentación, mientras que la información que se muestra al usuario consiste en imágenes de las cámaras.

En el proyecto Venus [Drap et al., 2008], se propone el uso de un ROV para obtener un mapa del sitio georeferenciado. A partir de los datos obtenidos, se reconstruyen modelos 3D de los objetos y se presentan en una interfaz con realidad aumentada, en la que se solapan modelos y mediciones sobre las imágenes reales. Más recientemente, se ha utilizado el simulador UWSim [Prats et al., 2012c] en conjunción con las gafas de realidad virtual Oculus Rift para ofrecer una interfaz de control inmersiva [García et al., 2017]. Además, se ha experimentado con diferentes modos de control como los mandos de videojuegos e interfaces más avanzadas uti-



Figura 5.7: Interfaces *hardware* para controlar robots submarinos.

lizando Leap Motion, un sensor 3D con capacidad de detectar de forma precisa los gestos por parte del usuario [Weichert et al., 2013]. En estos trabajos se ha tratado de avanzar en el área de HRI en misiones submarinas a través de los siguientes conceptos:

- Ofrecer al usuario solamente la información que necesita dependiendo del contexto y la misión. De esta forma se minimiza la sobrecarga de información.
- Crear un sistema inmersivo en el que el usuario tenga la sensación de telepresencia.
- Diseñar interfaces de control gestual naturales. El objetivo es mejorar los sistemas de control tradicionales como el uso *joysticks* y teclados.

En las secciones siguientes se detalla la aproximación a la HRI seguida en el proyecto MERBOTS en general, y por este trabajo en particular. Se propone el uso de diferentes GUIs para permitir al usuario la supervisión y monitorización del sistema de acuerdo a los pasos que se han descrito en capítulos anteriores.

5.4. Interacción Humano-Robot en MERBOTS

Aunque la robótica y los AUV están alcanzando cotas de autonomía cada vez mayores, en el contexto de la recuperación de objetos arqueológicos se debe operar de forma robusta y segura para no dañar los valiosos objetos que se puedan encontrar. Teniendo en cuenta las consideraciones de la sección anterior, se hace necesario disponer de una interfaz que permita monitorizar y supervisar de forma conveniente las misiones de intervención arqueológicas. A este efecto, en el contexto del proyecto MERBOTS [Sanz et al., 2017a] se desarrolla una estrategia de HRI que consiste en el desarrollo de diversas interfaces para el control, supervisión y monitorización de la intervención de forma sencilla e intuitiva.

Las interfaces se integran con el resto de componentes de Visión, Estimación, Especificación y Ejecución, así como con ROS. Consisten en una ventana de control desarrollada con QT y una escena de supervisión en UWSim. Además de la herramienta de visualización RViz que se utiliza como fuente de información adicional.

La ventana de control principal dispone de diferentes pestañas que permiten monitorizar el robot GIRONA500 y el robot SPARUS II [Carreras et al., 2013] además de controlar la manipulación. Estas pestañas, que permiten controlar la

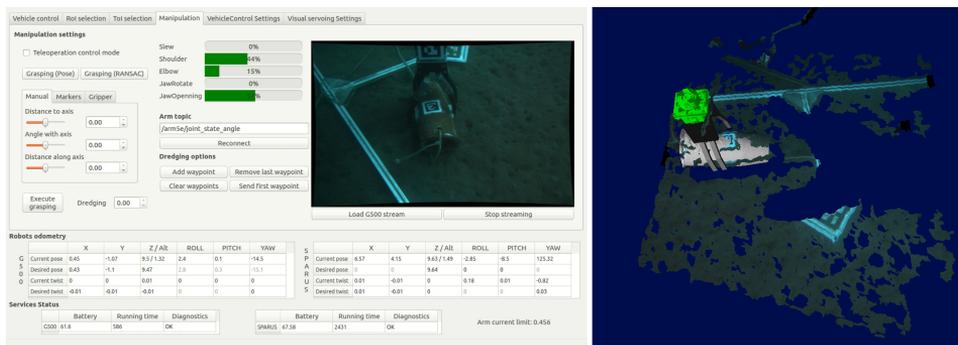


Figura 5.8: La interfaz muestra información de navegación, sensores, cámaras y controles para realizar la manipulación (izquierda). Escena de especificación en UWSim (derecha).

navegación, los parámetros del robot y la interacción entre ambos robots son parte del proyecto MERBOTS. Esta interfaz permite a un usuario controlar ambos robots de diferentes modos, así como monitorizar los parámetros de navegación como son la odometría e incluso visualizar las cámaras. Además, permite mover el robot a una determinada posición o configurar el SPARUS II para realizar un *station keeping* [García-Fidalgo et al., 2018] sobre el objeto que debe manipular el GIRONA500 con el brazo manipulador ¹. En las siguientes secciones se detalla la parte de la interfaz de MERBOTS para el control de la manipulación.

5.5. Interfaz para la supervisión de la manipulación

La ventana de control de la manipulación es el elemento principal de HRI de este trabajo junto con la escena de UWSim asociada (ver Figura 5.8). Consiste en una GUI desarrollada con QT [QT, 2018] que permite controlar todos los elementos de la intervención. Como ya se ha explicado, en ella se puede visualizar información de navegación, información para monitorizar la manipulación, controles para realizar la manipulación y la cámara del robot. La información relativa a la manipulación consiste en la posición de las articulaciones, la corriente consumida y los límites de las articulaciones.

La estimación de las características de la escena se realiza a través de un botón de esta GUI. Posteriormente se puede supervisar la caracterización del agarre. Cuando el usuario está satisfecho con la especificación puede proceder a la ejecución. En paralelo, se dispone de una escena en UWSim [Prats et al., 2012c] en la que se puede ver la posición de las articulaciones del robot en tiempo real y la nube de puntos 3D que se obtiene a través de la cámara. En el caso de haberse iniciado la estimación, también se muestra en el simulador un modelo 3D que representa la detección del objeto, así como la posición y orientación de agarre escogida utilizando un modelo 3D del efector final. De esta forma el simulador ofrece información del

¹Experimentos de *station keeping* de MERBOTS: <https://youtu.be/1xECxNb0-dQ>

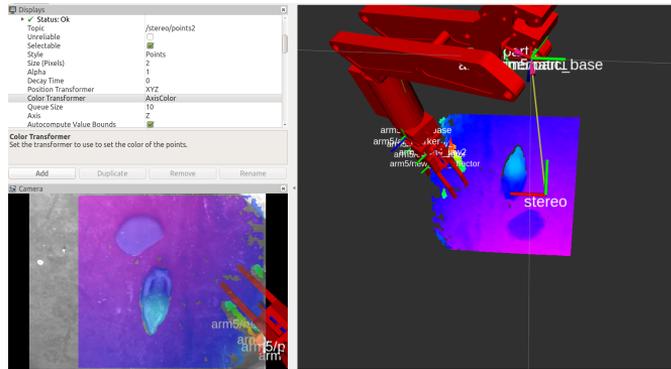


Figura 5.9: Interfaz de datos auxiliar utilizando RViz, la interfaz de monitorización de ROS.

entorno real y de la intervención a realizar, generando un entorno de Realidad Aumentada (AR) que facilita su tarea al usuario.

En adelante, se detalla cómo el usuario interactúa con el sistema para definir la manipulación. Además de la información disponible en la interfaz y en el simulador, se ha configurado la herramienta RViz para mostrar información adicional al usuario en caso de necesitarla. Esta ventana, que es totalmente opcional, muestra información como la estructura de sistemas de referencia (árbol de transformaciones de ROS [Quigley et al., 2009]) sobrepuesta sobre la imagen del robot, la nube de puntos 3D y la cámara del robot (ver Figura 5.9).

El primer paso a la hora de iniciar la intervención es la selección del método de estimación. El usuario selecciona entre los cinco métodos de estimación (ver Figura 5.10) descritos en el Capítulo 4 o elige la opción de obtener la posición desde una fuente externa, como puede ser un método de estimación basado en marcas o en plantillas. Esta opción se utilizó en MERBOTS para obtener la estimación con métodos alternativos. Después, inicia la estimación de la geometría de la escena. Una vez se ha calculado, se muestra en UWSim un modelo geométrico correspondiente al modelo estimado. En el caso de las estimaciones mediante RANSAC se muestra el modelo correspondiente (cilindro, esfera o caja), mientras que con el método de PCA se utiliza siempre una caja y con SQ se utiliza el modelo 3D obtenido a través de los parámetros del modelo matemático. Al mismo tiempo, se obtiene una posición y orientación por defecto del manipulador de forma autónoma. Excepcionalmente, en los experimentos de integración, que detectan una ánfora idéntica al objeto *C*, se utiliza como modelo visual un modelo 3D preciso del objeto obtenido mediante reconstrucción láser. Esta diferencia es únicamente visual y no es posible realizarla con objetos cuyo modelo es desconocido, como es el objetivo de este sistema.

Una vez el usuario dispone de una posición y orientación de agarre por defecto puede decidir utilizarla para realizar la manipulación, obtener una nueva posición utilizando la especificación avanzada analítica o modificar la posición obtenida. Si se desea ejecutar la tarea, el usuario pulsa el botón de inicio de la ejecución y la intervención pasa a ser controlada a través del módulo de Ejecución descrito en el Capítulo 6. En caso de que se desee utilizar la clasificación de agarres, puede

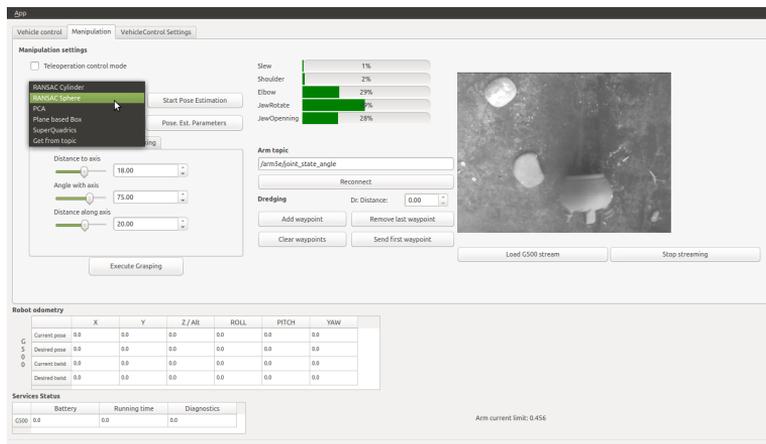


Figura 5.10: Interfaz para la supervisión de la manipulación. Selección del método de estimación.

hacerlo utilizando la pestaña de clasificación, que generará una lista de posiciones de agarre. En caso de aceptar la posición inicial o una obtenida mediante clasificación, esta puede ser ajustada de acuerdo a su criterio. La posición de agarre puede ser modificada teniendo en cuenta la geometría del objeto (Modo Geométrico) o utilizando marcadores interactivos en 3D (Modo Interactivo). En las siguientes secciones se detallan estos dos modos de funcionamiento de la interfaz.

5.5.1. Especificación en Modo Geométrico

Disponer de la geometría del objeto, incluyendo su posición, orientación y forma, resulta de gran utilidad para monitorizar y supervisar una intervención. El sistema de especificación del agarre utiliza esta información para generar una posición de agarre de forma autónoma. Después, se ofrece la posibilidad de modificar la posición generada de forma manual. Para ello la GUI dispone de controles deslizantes que permiten configurar los parámetros en la función $F(d, \alpha, r)$, descrita en el Capítulo 4. Así, se puede modificar la posición y orientación de agarre relativa al objeto (oMg) fácilmente. La Figura 5.11 muestra la pestaña utilizada (*Manual*) para utilizar el Modo Geométrico y la posición de agarre resultante en el simulador. Los controles *Distance to axis*, *Angle with axis* y *Distance along axis* permiten configurar los parámetros r , α y d , respectivamente. Este modo también puede ser utilizado para modificar la posición de agarre generada a través del método de clasificación heurística. Sin embargo, debido a que la posición generada mediante análisis no utiliza esta fórmula F , no es posible modificar los agarres generados analíticamente.

5.5.2. Especificación en Modo Interactivo

Como alternativa a la especificación basada en las propiedades geométricas, se pone a disposición del usuario un modo de interacción en 3D que hace uso de marcadores interactivos. Los marcadores interactivos son un elemento 3D utilizado

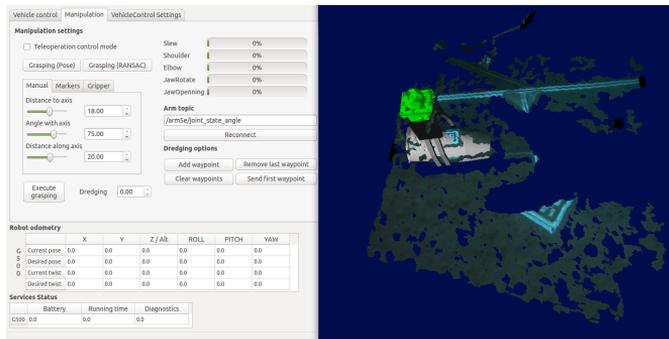


Figura 5.11: Especificación del agarre teniendo en cuenta la geometría del objeto. En blanco se superpone el modelo del objeto en UWSim.

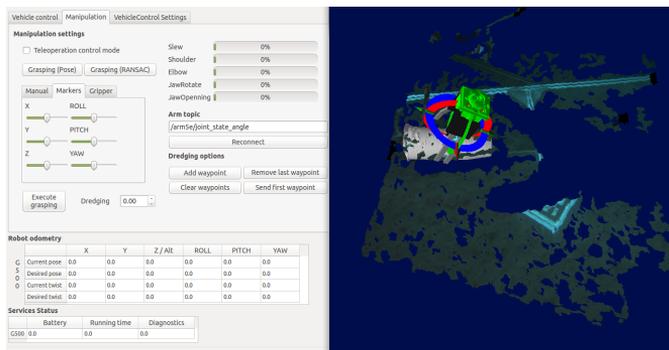


Figura 5.12: Especificación del agarre teniendo utilizando marcadores interactivos en 3D. En blanco se superpone el modelo del objeto en UWSim.

en la herramienta de visualización de ROS RViz que han sido desarrollados también para osgOcean y UWSim. Con ellos es posible interactuar con un objeto de la escena. Como se puede ver en la escena que se muestra en la Figura 5.12, se trata de un conjunto de controles que permiten controlar un objeto en 3D para rotarlo y moverlo en 3D de manera muy visual. Estos marcadores se utilizan cuando el usuario utiliza la pestaña *Markers*, entrando en el Modo Interactivo. En ese caso, la posición y orientación del manipulador se debe configurar de forma manual. Además de estos controles en UWSim, la interfaz cuenta con varios controles deslizantes (*X*, *Y*, *Z*, *ROLL*, *PITCH*, *YAW*) que se pueden utilizar para ajustar la localización con mayor precisión.

El uso de marcadores es muy intuitivo aunque requiere un mayor número de pulsaciones para configurar una posición. Por ello, este modo se puede utilizar para ajustar cualquier agarre que ha sido obtenido mediante el Modo Geométrico y mediante Clasificación de agarres heurística o analítica. Este modo de especificación puede ser más complejo para un usuario no experto, pero el usuario tiene una mayor capacidad de intervención. Alternativamente, también es posible realizar la especificación utilizando un *joystick* o mando de control.

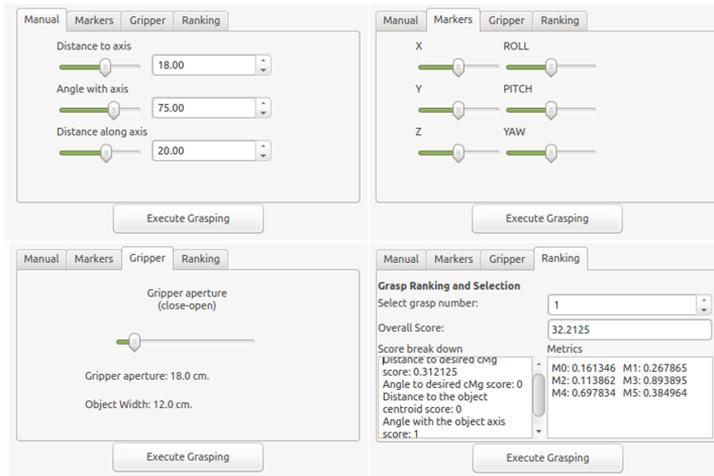


Figura 5.13: Pestañas de control de la especificación. Control geométrico (arriba, izquierda), control con marcadores interactivos (derecha, arriba), control de la apertura del manipulador (abajo, izquierda) y control de la especificación con clasificación (abajo, derecha).

Especificación basada en la Clasificación de agarres

La interfaz de Especificación ofrece otra alternativa para seleccionar la posición de agarre entre las hipótesis generadas mediante Clasificación de agarres. Como se muestra en la Figura 5.13 (*Ranking*), tanto el método de Clasificación de agarres heurístico como el analítico utilizan la misma pestaña de la interfaz. Para el método de estimación basado en SQ se realiza la clasificación analítica, mientras que para el resto de métodos se realiza la clasificación heurística.

En primer lugar el usuario puede elegir entre la lista de hipótesis de agarre ordenadas. Cada vez que se modifica el número de agarre seleccionado en la interfaz se muestra la posición y orientación correspondiente en la ventana de UWSim, mostrando al mismo tiempo la posición que el manipulador puede alcanzar realmente. Además, se muestra la puntuación heurística con la que ha sido clasificado (*Overall Score*) en base a los criterios detallados anteriormente y las métricas del sistema en caso de la clasificación analítica. Esta información es menos intuitiva que el resto de información visual y el resto de controles de la interfaz. No obstante, el objetivo de esta información es el de complementar la representación visual. El usuario puede, igualmente, basar su decisión únicamente en la información visual del agarre. Con estos datos el usuario puede iterar por la lista para encontrar el agarre que mejor se adapte a la situación. Una vez seleccionado, es posible editar esta posición y orientación antes de ejecutar el agarre. Para ello, el usuario debe cambiar a uno de los modos anteriores, que permiten editar la posición.

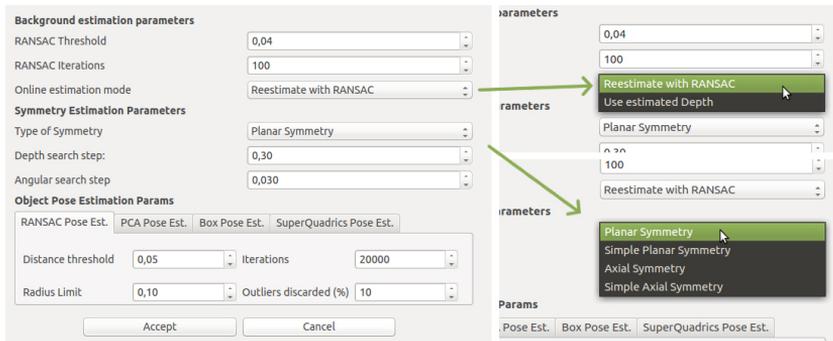


Figura 5.14: Configuración de parámetros de segmentación y estimación.

Controles adicionales

La GUI ofrece también la posibilidad de controlar la apertura inicial del manipulador. En la Figura 5.13 se muestra esta configuración, que es relevante porque una apertura del manipulador excesiva aumenta la probabilidad de contactar con obstáculos u otros objetos. El sistema ofrece una apertura por defecto en función de las dimensiones del objeto. En la práctica, no ha sido necesario modificar la apertura por defecto en ningún experimento.

Por otro lado, el usuario puede intervenir en la fase de estimación de la geometría modificando los parámetros de los algoritmos de estimación según sus necesidades. Como se ve en la Figura 5.14, además de modificar el método de estimación a aplicar se pueden modificar los siguientes parámetros:

- Parámetros de estimación del fondo: permite configurar el umbral de distancia y número de iteraciones para la segmentación RANSAC. Además, se selecciona el método de seguimiento, ya sea utilizando RANSAC sucesivamente o utilizando el modelo del plano existente para filtrar la nube de puntos rápidamente.
- Parámetros de estimación de simetrías: determinan los incrementos de distancia y ángulo para el algoritmo de búsqueda de simetrías. Un incremento menor de este valor aumenta el número de planos o ejes que se evalúan para obtener la simetría. Al mismo tiempo que se amplía el rango de búsqueda, también crece el tiempo de estimación. Además, se puede seleccionar si se utiliza simetría planar (por defecto) o simetría axial.
- Parámetros de estimación del objeto: ajuste de los parámetros de segmentación correspondientes a cada algoritmo. Los parámetros disponibles en cada método se describen en la sección correspondiente a cada método, en el Capítulo 4. En la Figura 5.14, por ejemplo, dispone de los parámetros de segmentación utilizados con RANSAC para la detección del cilindro y la esfera como son: umbral de distancia, número de iteraciones, límite de radio y porcentaje de *outliers* rechazados.

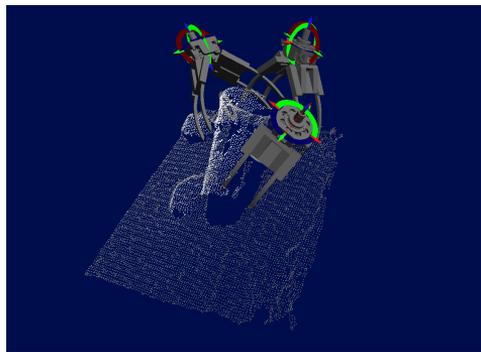


Figura 5.15: Uso de la GUI para introducir puntos de ruta que definen la trayectoria que debe seguir el robot.

Finalmente, la interfaz también dispone de una casilla para indicar que se quiere mover el robot de forma teleoperada, opciones para la conexión del brazo a través de ROS y opciones relativas a la misión de limpieza de objetos arqueológicos. Respecto a este tipo de intervención, el usuario puede utilizar la GUI de forma conjunta con el UWSim para definir y posteriormente seguir una serie de puntos de ruta. Gracias al uso de marcadores interactivos y el simulador UWSim, se pueden configurar varios puntos de ruta sobre la nube de puntos de la escena que el robot ha de seguir para eliminar los sedimentos alrededor de un objeto. La Figura 5.15 muestra un ejemplo de creación de puntos de ruta de forma manual en el espacio cartesiano. Cada punto de ruta que se crea se muestra utilizando un modelo 3D del manipulador. De esta manera, se puede ver de forma intuitiva las tres posiciones por las que pasará el manipulador una vez se inicie el proceso. Esta es una función ofrecida al usuario en el contexto del proyecto MERBOTS que se une a las múltiples funciones autónomas disponibles para el operador. Disponer de funcionalidades de alto nivel facilita el control de la misión. En el Capítulo 6 se demuestra el uso de técnicas desarrolladas en el contexto de esta tesis aplicadas a obtener una ruta de limpieza de forma autónoma.

Una vez la trayectoria de limpieza ha sido definida o el agarre totalmente especificado, el usuario inicia la ejecución, que transcurre de forma autónoma sin su intervención. El usuario puede ver el progreso de la misión a través de la cámara del robot, así como la configuración del robot junto con la reconstrucción 3D de la escena. En todo momento puede intervenir para tomar el control manual del robot o cancelar la misión.

Cabe destacar que la especificación del agarre se realiza una sola vez, definiendo la posición relativa deseada entre el objeto y el manipulador. De esta forma, si la posición y orientación del objeto cambia por cualquier motivo, la posición y orientación de agarre se actualiza consecuentemente. En los experimentos con el AUV, este método ha demostrado ser efectivo para corregir los efectos de las imprecisiones en la odometría del vehículo. Además, para prevenir falsas detecciones cuando existen oclusiones o una distancia inadecuada entre el objeto y la cámara, se configura un filtro sobre la posición del objeto que descarta detecciones inconsistentes.

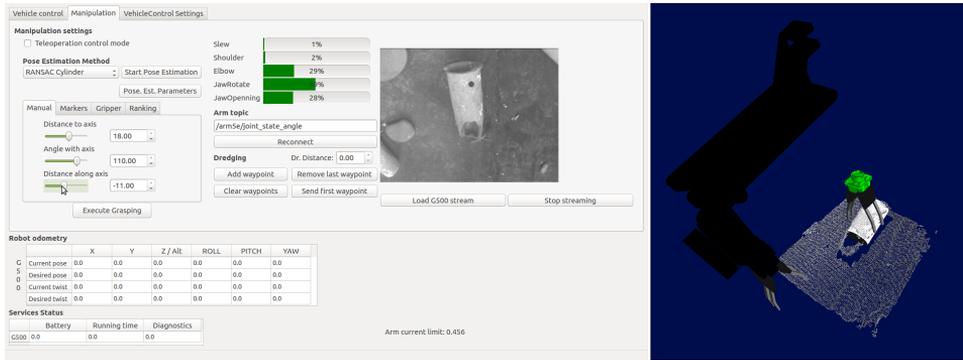


Figura 5.16: Uso de la GUI en los experimentos de manipulación en el laboratorio IRS Lab.

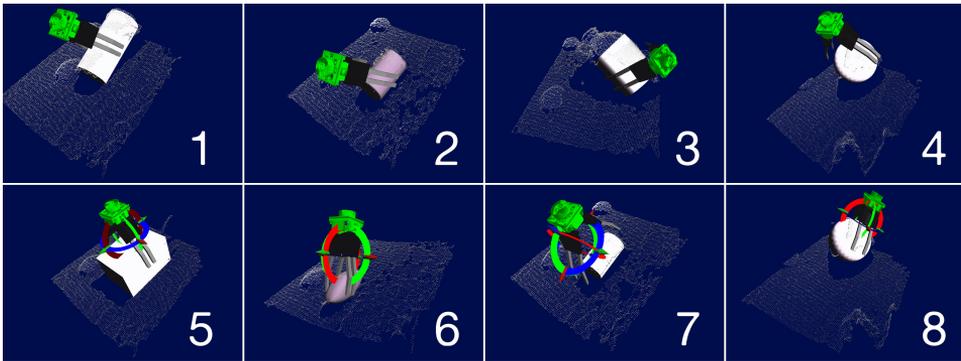


Figura 5.17: Resultados de la especificación en el Modo Geométrico (arriba) y en el Modo Interactivo (abajo).

5.6. Especificación en los escenarios de intervención

El módulo de Especificación ha sido utilizado en los diferentes escenarios de intervención: en el laboratorio, en simulación, en la piscina del CIRS y en los experimentos en el mar. Como se puede ver en la Figura 5.16, durante la realización de los experimentos en el laboratorio la interfaz muestra las imágenes de la cámara Videre e información relativa únicamente al manipulador, mientras que la información del vehículo permanece desactivada. En la escena en UWSim se puede ver la posición real del robot en ese momento, la reconstrucción 3D del entorno, las características del objeto estimadas y la posición de agarre que el usuario está configurando. En este escenario la GUI se ha utilizado para evaluar todos los algoritmos de estimación disponibles con todos los objetos en tiempo real.

En la Figura 5.17 se muestran resultados de la Especificación del agarre para varios objetos utilizando algoritmos diferentes. En la fila superior se muestran agarres definidos en Modo Geométrico. Los agarres 1 y 2, generados por defecto, son adecuados para el objeto. El agarre 3 también ha sido generado de forma autónoma, aunque a partir de un modelo del objeto que no es muy aproximado al objeto

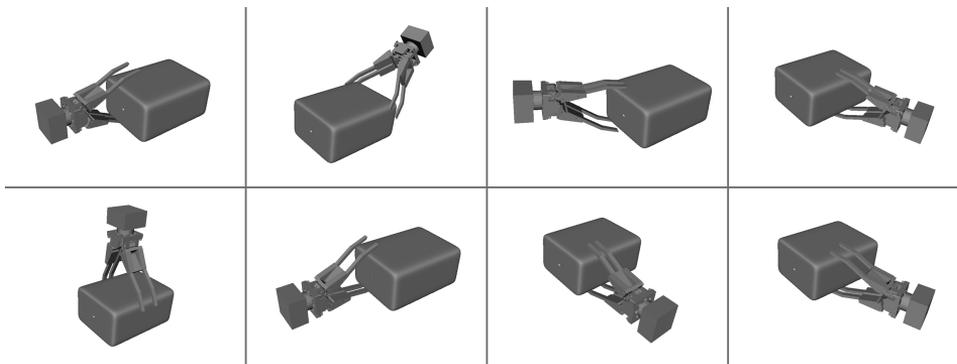


Figura 5.18: Resultados de la generación de agarres analítica con OpenRave para el objeto A.

real. En el caso 4, el agarre no es correcto debido a que con el modelo de la esfera es difícil determinar la dirección de aproximación adecuada para el agarre. Por otro lado, en la fila inferior se muestran los agarres en Modo Interactivo. Estos agarres (5, 6, 7 y 8) son definidos por parte del usuario de forma totalmente manual. Por ello parecen todos adecuados. Como se puede ver, este modo es muy flexible a la hora de manipular ciertos objetos. En el agarre 6 el usuario define correctamente el agarre sobre el objeto más pequeño del conjunto de objetos. En el agarre 7 se define una posición y orientación diferente a las generadas de forma autónoma ya que no se define a lo largo del eje de máxima inercia del objeto. En el agarre 8 se define de forma correcta un agarre que no pudo ser definido correctamente en el caso 4 de forma autónoma. Así, se utiliza el Modo Interactivo para corregir de forma manual errores en la generación autónoma del agarre que no se pueden corregir con libertad en el Modo Geométrico.

Por otro lado, el sistema de Especificación del agarre avanzado que hace uso de los modelos 3D también ha sido probado con todos los objetos. A modo de resumen, en las figuras 5.18, 5.19 y 5.20 se muestra la generación de agarres analítica para los objetos *A*, *E* y *I*, cuya geometría representa a grandes rasgos los tres tipos de objetos utilizados. Se muestran 8 agarres de un conjunto de 20 generados que cumplen la condición de *force-closure*. Como se aprecia en los agarres obtenidos las pinzas del manipulador no tienen la misma apertura. Este modelo no representa efector final porque este sólo tiene 1 GDL y ambas pinzas se mueven al unísono. Esta condición se ha relajado para que el *framework* OpenRave pueda encontrar agarres de forma satisfactoria. La forma en la que este funciona es que primero entra en contacto con un punto del manipulador y después cierra las pinzas. Por este motivo, para obtener la lista de agarres se descartan aquellos que por tener un ángulo diferente en cada pinza no pueden ser realizados. No obstante, la mayoría de los objetos pueden ser manipulados con los agarres generados.

Una de las desventajas del uso de métodos analíticos para obtener la posición de agarre es su coste temporal. En la Tabla 5.1 se muestra el coste temporal del análisis de agarres para cuatro objetos que representan las diferentes geometrías disponibles en el conjunto de objetos utilizados. Para cada uno de los objetos analizados se obtiene: el número de muestras generadas, el número de muestras para las cuáles se

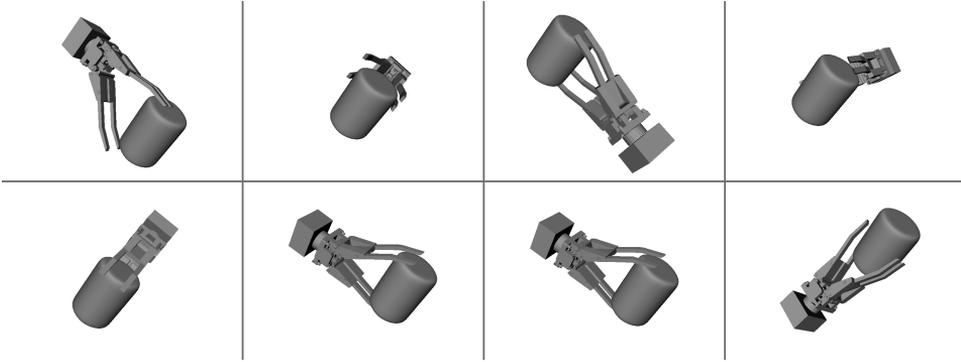


Figura 5.19: Resultados de la generación de agarres analítica con OpenRave para el objeto E.

Tabla 5.1: Coste temporal del análisis de agarres

ID	Núm. Muestras	Muestras Eval.	T. Análisis (s.)	T. Métricas (s.)
C	59920	1717	215,40	13
A	109760	2043	463,62	50
E	42560	686	101,73	17
H	126448	4101	578,62	13

ha evaluado el agarre para saber si cumple la condición de *force-closure*, el tiempo requerido para obtener los agarres y el tiempo para calcular las métricas para todos los agarres. Se han obtenido 100 agarres que cumplen la condición de *force-closure*. El número de muestras generadas en este experimento es elevado debido a los parámetros de muestreo elegidos, que son $Anglerange = 0.02$, $Deltaspacespace = 0.02$, $Roll = (0, \pi * 2, \pi/4)$, $Standoffs = (0.0, 0.02, 0.04, 0.05, 0.09, 0.12, 0.14)$.

Los resultados confirman que el coste temporal de la especificación analítica del agarre es elevado. Cabe destacar que los objetos más grandes (*A* y *H*) requieren significativamente más tiempo. Por un lado, dicho coste no es aceptable para un uso frecuente. No obstante, este coste está en la línea del coste para la estimación de los objetos utilizando SQ, que es el método utilizado en este caso. Así, para realizar una estimación inicial en conjunto con este método, este tiempo resulta aceptable. Comparativamente, el tiempo necesario para calcular las métricas es muy bajo por lo que es interesante obtenerlas en el mismo proceso.

Las hipótesis de agarre obtenidas se filtran posteriormente utilizando la geometría de la escena para descartar aquellos que sean físicamente imposibles. Esto se debe a que el análisis realizado no tiene en cuenta ni la posición del objeto ni la de la mano en el mundo real. Al trasladar estas posiciones de agarre al escenario de la intervención las posiciones deben estar en espacio libre. Además, haciendo uso de la IK del robot, se filtran los agarres que no sean alcanzables por parte del robot. Como se puede ver, se descartan hipótesis de agarre por diferentes motivos por lo que es necesario generar un número de agarres suficientemente alto.

La interfaz también ha sido probada realizando la especificación del agarre en una misión simulada con UWSim. De esta forma, se utiliza el UWSim tanto para

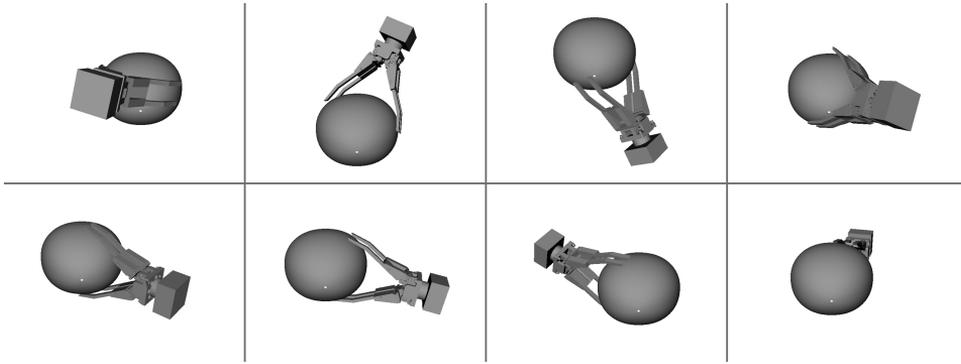


Figura 5.20: Resultados de la generación de agarres analítica con OpenRave para el objeto I.

controlar la misión, como para simular la escena. El simulador ha sido de gran utilidad para desarrollar los métodos de especificación y la interfaz de la misma forma que en el caso de los algoritmos de estimación.

Por otro lado, la GUI se utilizó en los experimentos de integración de MERBOTS para especificar la posición y orientación del manipulador en la piscina del CIRS. En esta configuración del robot, la pestaña de manipulación muestra los parámetros del brazo y las imágenes de la cámara Point Grey Bumblebee 2. Por otro lado, la interfaz muestra información de navegación en su parte inferior y dispone de otras pestañas para realizar el seguimiento de la misión desde el SPARUS II o configurar los parámetros de comunicación. En la Figura 5.21 se muestra el uso la interfaz en la piscina, dónde las cámaras disponen de una gran visibilidad. Durante estos experimentos se utilizó la GUI desarrollada para obtener la posición utilizando el método de RANSAC para cilindros en tiempo real y, alternativamente, para extraer la posición del objeto a través de una fuente externa.

Después, se utilizó la interfaz para verificar la posición de agarre generada de forma autónoma. Una vez comprobada la posición, se inició la ejecución del agarre de forma satisfactoria con la pulsación de un sólo botón. Posteriormente, se ha probado la interfaz con los datos grabados en estas misiones y otros métodos de estimación que no estaban disponibles en el momento de la realización de los experimentos, como son PCA y SQ. Esta evolución, se refleja en ambas figuras y en otras a lo largo de este trabajo, ya que en la versión más reciente (ver Fig. 5.16) se puede seleccionar entre los métodos disponibles en vez de tener que utilizar RANSAC exclusivamente. En estos experimentos, realizados en febrero de 2017, se utilizaron nubes de puntos procedentes tanto de la reconstrucción estéreo, utilizando una cámara Point Grey Bumblebee 2, como de la reconstrucción láser [Palomer et al., 2018].

Por último, la interfaz fue probada en experimentos en el mar en marzo de 2017. En dichos experimentos se utiliza el I-AUV en la misma configuración que en el CIRS. No obstante, la reconstrucción 3D basada en láser no estuvo disponible debido a la turbidez del agua. En consecuencia, los algoritmos solamente han sido probados con la cámara estéreo Bumblebee 2. Como se puede ver en la Figura 5.22, la visibilidad y reconstrucción del entorno es también peor en el puerto de Sant

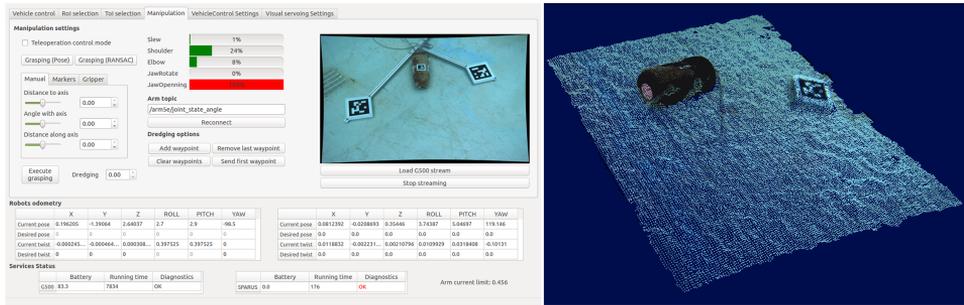


Figura 5.21: Uso de la GUI para la especificación en la piscina del CIRS.

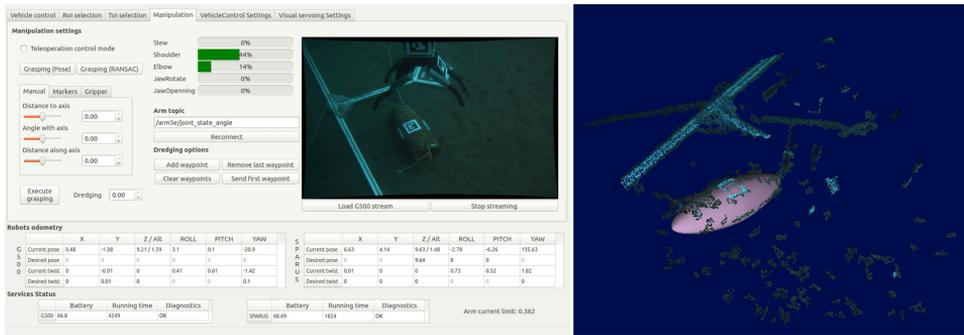


Figura 5.22: Uso de la GUI para la especificación en el mar.

Feliu de Guíxols. De la misma manera que en los experimentos en la piscina del CIRS, los experimentos se llevaron a cabo utilizando el método de segmentación mediante RANSAC. Igualmente, mientras que los datos obtenidos han sido utilizados para probar los métodos de segmentación y especificación aplicables a este objeto. Es decir, se han probado los métodos basados en PCA y SQ con datos de los experimentos en el mar.

En definitiva, la interfaz ha demostrado ser una herramienta de gran utilidad a la hora de supervisar y monitorizar las diferentes intervenciones realizadas en entornos reales en el contexto del proyecto MERBOTS, en simulación y en escenarios más controlados como el laboratorio del IRS Lab. El tanque de agua del laboratorio permite realizar un número de pruebas mucho mayor. Así, se ha utilizado la GUI en dicho entorno para obtener todos los resultados presentados en los Capítulos 4 y 6. Se puede observar cómo en los experimentos de integración la interfaz cobra una dimensión mayor, ya que interactúa con el brazo ARM5E, el GIRONA500 y el SPARUS II.

Ejecución del agarre y Resultados

Una vez el sistema dispone de toda la información del entorno se puede iniciar la fase de interacción física. En este capítulo se describe la ejecución de la manipulación y los resultados obtenidos en los escenarios de validación.

En primer lugar se detalla la metodología del módulo de Ejecución del agarre. Después, se detallan los diferentes modos de control disponibles en el sistema. Los modos de control ofrecen un conjunto de funcionalidades que sirven para llevar a cabo la misión y para que el usuario disponga de un conjunto de funciones de alto nivel.

Finalmente, se muestran los resultados de la manipulación de objetos arqueológicos en el laboratorio IRS Lab y en los experimentos realizados en el contexto del proyecto MERBOTS. Los experimentos en el laboratorio IRS Lab hacen uso del ARM5E en una plataforma fija mientras que los experimentos de integración del MERBOTS hacen uso del vehículo GIRONA500 con el manipulador ARM5E. Por otro lado, se explica una aplicación del *framework* desarrollado a una tarea diferente a la de manipulación, como es la de la limpieza de una ánfora.

6.1. Motivación

La ejecución de la manipulación supone la fase final del sistema de manipulación guiada por visión propuesto. Este paso sirve como validación de la metodología, en el sentido en que realizar la manipulación de forma satisfactoria indica que se han realizado los pasos previos de adecuadamente. En esta tesis se realiza la manipulación con una aproximación clásica al tiempo que se tratan de paliar los problemas que aparecen en el medio subacuático utilizando visión. En la robótica móvil existen en la actualidad multitud de opciones en cuanto a visión 3D, sensores táctiles y sensores de fuerza. Todos estos elementos mejoran la percepción y el control de las tareas de manipulación. Como se ha constatado en la introducción de este trabajo, en el medio submarino su disponibilidad es mucho menor. Por este motivo se utiliza la visión y la retroalimentación propia del manipulador a la hora de controlar la intervención.

Disponer de un sistema de control de la manipulación adecuado es de suma importancia para poder utilizar el robot de forma autónoma y poder utilizarlo también de forma teleoperada. Así mismo, es necesario definir las entradas del sistema, cómo se puede mover el robot y cómo se desarrolla el control de la misión. De esta forma, se pueden llevar a cabo diferentes casos de uso: mover el robot hasta coger un objeto, seguir una serie de puntos de ruta con el efector final o mover el manipulador de gracias a un mando de control. El desarrollo de estas funcionalidades permite al usuario disfrutar de un sistema más completo y fácil de utilizar.

6.2. Metodología de Ejecución de la tarea

El objetivo principal de este trabajo es el de desarrollar un sistema integral capaz de manipular objetos en el entorno submarino. Para llevarlo a cabo, se describe en esta sección la metodología para la ejecución de la tarea. En capítulos previos se han descrito las fases de Reconstrucción, Estimación y Especificación. El resultado de estas fases es la posición y orientación del efector final con respecto al objeto (oMg) para poder cogerlo. A partir de esta posición obtenida en la etapa de Especificación, la Ejecución se subdivide en las siguientes fases:

1. *Inicio*: durante la realización del procesamiento visual inicial, se lleva el brazo a una posición de inicio. Esta posición es siempre la misma y corresponde a una posición en la que el eje *Slew* está centrado con respecto a la base cinemática y la cámara. En esta configuración el resto de articulaciones están lejos de sus límites. Esta posición es: $q_{home} = (0, 0, 2.2, 0)$.
2. *Apertura*: se abre el efector final hasta el ángulo deseado. Este valor viene determinado por el tamaño del objeto y puede ser modificado por el usuario. Aunque podría ser un valor máximo fijo, un valor más ajustado a la forma del objeto permite evitar colisiones indeseadas.
3. *Aproximación*: se lleva el brazo a una velocidad moderada a las cercanías de la posición de manipulación a -10 cm. en línea recta de la posición deseada. El objetivo es aproximarse al objeto con una trayectoria adecuada. De esta forma se previene que el manipulador contacte con el objeto antes de estar en la posición deseada. Se utiliza un error de posición grande debido a que la precisión de la posición de aproximación no es importante.
4. *Agarre*: en esta fase se lleva el manipulador a la posición precisa de manipulación obtenida en la fase de Especificación. En este caso se permite un error de posición menor. En caso del ARM5E desacoplado se utiliza un error de posición de 2,5cm. Cuando se utiliza el GIRONA500 el error es mayor debido a las interacciones con el medio y los problemas dinámicos.
5. *Cierre*: se cierra el efector final hasta que el objeto ha sido cogido. Como se ha indicado previamente, se utiliza la retroalimentación de corriente del brazo para determinar que el objeto se ha cogido.

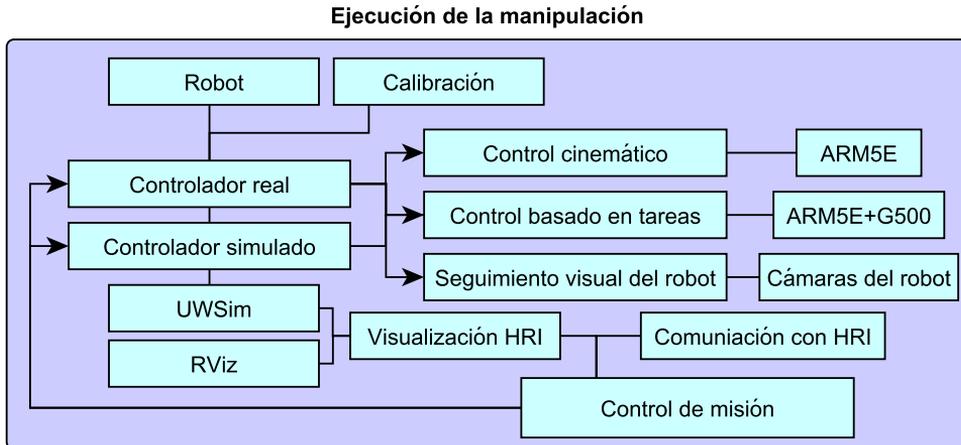


Figura 6.1: Componentes del módulo de Ejecución.

6. *Transporte*: se desplaza el objeto a la posición deseada. En los escenarios propuestos simplemente se eleva el objeto del suelo 15 cm. para después volverlo a dejar.
7. *Apertura final*: una vez el manipulador ha vuelto a la posición de agarre, se abre la mano para dejar el objeto.

Durante estas fases, la ejecución se controla utilizando la retroalimentación física del brazo y la retroalimentación visual de las cámaras. Para ello, se utiliza las lecturas de corriente del brazo, la información de posición de los codificadores relativos, la calibración del brazo y la información 3D de la visión para conocer en todo momento el estado del robot y la ubicación del objeto. Esto permite saber cuándo el objeto ha sido cogido correctamente o si ha ocurrido un error. Además, como parte de la HRI, se utiliza el simulador UWSim para mostrar la posición del brazo en tiempo real, durante y después de la ejecución de la operación. Como se ha comentado en secciones previas, se realiza el seguimiento del objeto en todo momento, por lo que si la posición del objeto y su orientación cambia, la posición de agarre deseada se actualiza en consecuencia. Este método ha demostrado ser de utilidad para combatir las incertidumbres del sistema debidas a las dificultades de la visión, las corrientes, las diferentes calibraciones necesarias, etc.

En la Figura 6.1 se muestran los diferentes componentes del módulo de Ejecución. Como se ha indicado, la ejecución se ha realizado de diferentes maneras en el mundo real: en el tanque de agua del IRS Lab, en la piscina del CIRS y en el puerto de Sant Feliu de Guíxols. En los experimentos en el laboratorio IRS Lab se ha utilizado el control cinemática del brazo únicamente. En los experimentos de MERBOTS, se ha utilizado un método basado en Prioridad de Tareas, desarrollado por Peñalver y Fernández [Peñalver et al., 2018]. En las secciones siguientes, se explica en mayor detalle la arquitectura del sistema de control del manipulador ya introducida en la Figura 2.8. En primer lugar se describen los sistemas de coordenadas utilizados en el espacio de la tarea. Después, se describe en mayor detalle el control del robot de diferentes modos. Posteriormente, se introduce el sistema

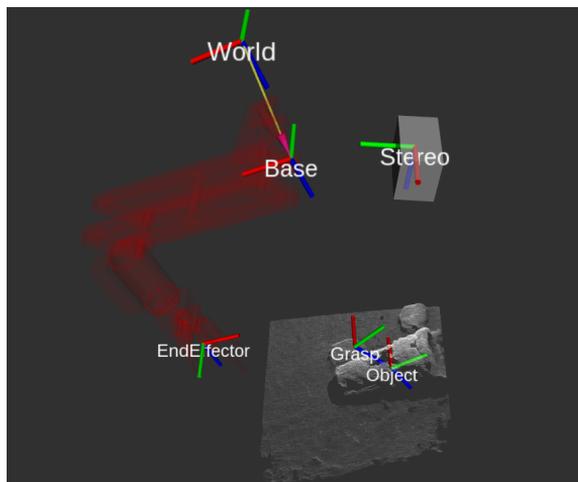


Figura 6.2: Sistemas de referencia asociados a la tarea de manipulación.

de control basado en prioridad de tareas, utilizado en los sistemas de control del I-AUV en los experimentos de integración.

6.3. Control del robot en el espacio de la tarea

Habitualmente la operación de un robot se define en el *espacio de la tarea* a través de posiciones y orientaciones en un espacio cartesiano, ya que resulta más natural [Prats et al., 2010]. De esta forma se define la posición de agarre del objeto en este trabajo. Sin embargo, la posición y orientación del manipulador se describe en el *espacio de las articulaciones*. Para definir la transformación entre dos sistemas de referencia cualesquiera en el espacio, en este trabajo se utiliza una nomenclatura con la siguiente forma aMb . Esta notación se puede leer como b con respecto a a y significa la transformación de b en el sistema de coordenadas a . Como se muestra en la Figura 6.2, en el sistema propuesto se utilizan las siguientes relaciones entre sistemas de referencia:

- wMb : Posición y orientación de la base del robot (*base*, b) con respecto del mundo (*world*, w). Esta transformación representa la posición del robot en el mundo y se utiliza principalmente en el caso de robots móviles. Es decir, se utiliza cuando se utiliza el brazo acoplado a un AUV. En este trabajo se considera la base del robot la base de la cadena cinemática del brazo. Cuando se utiliza el vehículo esta transformación se obtiene a través de la odometría.
- bMc : Posición y orientación de la cámara (*camera*, c) con respecto de la base del robot (*base*, b). Esta transformación determina la posición relativa entre la base de la cadena cinemática del brazo. Es muy importante para manipular los objetos de forma correcta puesto que la posición de agarre se determina en el sistema de referencia de la cámara. Puede ser calculada utilizando de forma empírica mediante calibración o de forma analítica. En este trabajo se

ha obtenido de forma analítica a través de los modelos 3D del robot ya que resulta más preciso debido a las limitaciones del brazo ARM5E.

- bMe : Posición y orientación del efector final del robot (*end effector*, e) con respecto de la base del robot (*base*, b). Esta transformación es la posición del manipulador con respecto a su base y se puede calcular utilizando la FK a partir de las articulaciones del robot.
- cMo : Posición y orientación del objeto (*object*, o) con respecto de la cámara estéreo (*camera*, c). Esta transformación representa la ubicación del objeto detectado. El módulo de Estimación tiene, entre otras funciones, el objetivo de determinar su valor. El sistema de referencia de o se sitúa en el centro del objeto, que viene determinado por el método de estimación. La orientación del los ejes del sistema de coordenadas se ha descrito en el Capítulo 4.
- cMg : Posición y orientación de agarre (*grasp*, g) con respecto de la cámara estéreo (*camera*, c). Esta transformación es el resultado del proceso de especificación y representa la ubicación a la que se debe mover el robot para poder manipular el objeto posteriormente. Se debe transformar en el sistema de coordenadas de la base del robot para poder calcular la IK.

Matemáticamente, estas transformaciones se representan mediante matrices homogéneas y pueden componerse e invertirse para obtener otras. Por ejemplo, para manipular un objeto es necesario disponer de la transformación entre la base del robot y la posición de agarre bMg . Para calcular la la posición de agarre con respecto a la base a partir de las transformaciones descritas se utiliza la ecuación $bMg = bMc * cMg$. Por otro lado, la posición de la base con respecto de la posición de agarre se obtiene utilizando la inversa de esta matriz $gMb = bMg^{-1}$.

6.4. Control del robot

La posición $x = (x, y, z, \alpha, \beta, \gamma)$ y velocidad $\dot{x} = (\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma})$ del efector final de una cadena cinemática k pueden ser obtenidas a partir de la posición de las articulaciones del robot $q = (q_1, q_2, q_3, q_4)$ y sus velocidades $\dot{q} = (\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4)$. Así, el cálculo se realiza a través de $x^k = FK(q)$ y $\dot{x}^k = J^k \dot{q}$, donde FK es la cinemática directa de la cadena y J^k la matriz Jacobiana del sistema en la configuración q . La FK del ARM5E ha sido descrita en el Capítulo 2.

Para llevar a cabo el mayor número posible de aplicaciones, el sistema debe estar dotado de una serie de primitivas de control que permita hacer diferentes tipos de peticiones al robot. Así, el control del brazo se puede realizar a diferentes niveles para alcanzar la posición o velocidad deseada. Estos modos de control se integran en cascada en único controlador en el módulo de Ejecución (ver Figura 6.3):

1. **Control cartesiano del efector final:** para llevar el efector final a la posición deseada x^* se realiza un cálculo de alcanzabilidad utilizando la IK. Después, se mueve el brazo a la posición articular obtenida q^* mediante el Controlador articular en posición.

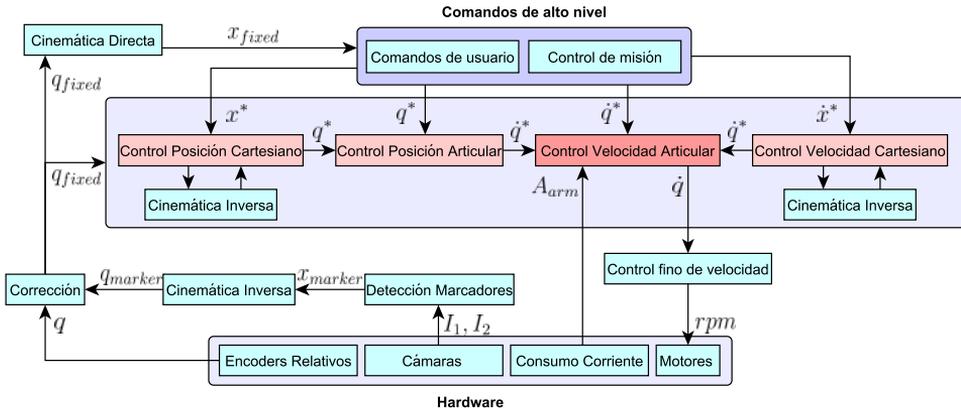


Figura 6.3: Arquitectura de control del ARM5E en detalle.

Un robot manipulador es redundante a nivel de articulaciones cuando el número de articulaciones disponibles es mayor que las necesarias para alcanzar una posición y orientación en el espacio cartesiano, es decir, 6 dimensiones. En la configuración disponible en el laboratorio se dispone de 4 GDL por lo que no es posible garantizar la alcanzabilidad de una posición con una determinada orientación. El controlador desarrollado sigue una estrategia *best effort* en la que se alcanza la posición deseada con la mejor orientación posible.

2. **Control cartesiano en velocidad del efector final:** la velocidad cartesiana deseada para el efector final (\dot{x}^*) se convierte en comandos de velocidad articular (\dot{q}^*) utilizando la pseudoinversa del Jacobiano (J^\dagger o J^+) del manipulador [Angeles, 1985][Grudic y Lawrence, 1993]. La velocidad articular y la velocidad cartesiana se relacionan mediante $\dot{x}_C = J_C \dot{q}$, donde J_C es la matriz jacobiana del robot respecto al sistema de referencia C , \dot{q} es el vector de velocidades articulares y \dot{x}_C es el vector de velocidades cartesianas del efector final en el sistema de referencia C . Cuando C es el efector final del robot, $\dot{x}_C = \dot{x}_e$ representa la velocidad cartesiana en el efector final dada la velocidad articular \dot{q} . Así, si la matriz jacobiana se da en el efector final y es invertible, el robot puede ser controlado a una velocidad cartesiana deseada \dot{x}_e^* mediante $\dot{q}^* = J_E^+ \dot{x}_e^*$, donde J_E^+ representa la pseudoinversa de la matriz jacobiana. Las velocidades articulares resultantes \dot{q}^* se envían al Controlador articular en velocidad.
3. **Control articular en posición:** antes de alcanzar la posición articular deseada q^* , se comprueba que los valores deseados están dentro de los límites de las articulaciones. Después, para alcanzar dicha posición q^* se obtiene la velocidad articular \dot{q}^* que reduce el error mediante un controlador estándar proporcional $\dot{q}(t) = K_p e(t)$, donde $e(t)$ es el error que se desea minimizar para cada articulación ($e = q^* - q$). Las velocidades articulares obtenidas \dot{q}^* se envían al Controlador articular en velocidad.
4. **Control articular en velocidad:** los comandos de velocidad q^* se verifican para determinar que las velocidades están en el rango de seguridad y se com-

prueba que este movimiento no superará los límites de las articulaciones. Una vez escaladas las velocidades para estar en el rango de seguridad del robot, se convierte la velocidad resultante \dot{q} en revoluciones por minuto (*rpm*) y se envía al brazo. El manipulador dispone de su propio controlador encargado de mantener la velocidad deseada para cada articulación. Como se puede ver, los comandos del resto de modos de control derivan en el envío de comandos de velocidad articular al robot mediante este controlador.

6.4.1. Controlador basado en Prioridad de Tareas

La capacidad de los robots para realizar tareas de forma diestra está directamente ligada a la redundancia del sistema [Siciliano, 1990]. La redundancia cinemática ocurre cuando un manipulador robótico tiene más GDL que aquellos requeridos para ejecutar una determinada tarea, como suele ocurrir en un Sistema Vehículo-Manipulador Submarino o Underwater Vehicle-Manipulator System (UVMS). La redundancia introduce en el sistema una mayor complejidad. No obstante, se puede utilizar para evitar singularidades, límites de las articulaciones, obstáculos, etc. En general, permite optimizar parámetros de rendimiento, lo que significa que el manipulador es capaz de obtener un mayor grado de autonomía.

La redundancia en los UVMS ha sido estudiada desde hace tiempo. Son conocidas varias aproximaciones que tratan con singularidades o el uso de tareas múltiples [Antonelli et al., 2008][Antonelli y Leonessa, 2008]. Sin embargo, es un punto de investigación de actualidad, y se han producido avances significativos manejando singularidades [Flacco y De Luca, 2014] y tratando con restricciones (unilaterales y bilaterales) en soluciones multitarea [Flacco y De Luca, 2015][Casalino et al., 2012].

Los sistemas de control presentados en la sección anterior se utilizan para el control del brazo ARM5E de forma desacoplada. Tanto en los experimentos reales del proyecto MERBOTS como en simulación, se utiliza un controlador basado en Prioridad de tareas [Peñalver et al., 2018]. El controlador por prioridad de tareas (*Task Priority*) tiene en cuenta los GDL del brazo al mismo tiempo que los del vehículo, estableciendo diferentes prioridades a diferentes tareas que se pueden definir. Este algoritmo en el estado del arte permite definir tareas secundarias a cumplir al mismo tiempo que se trata de cumplir otras. Estas tareas de menor prioridad pueden ser funciones como alejar las articulaciones de los límites, mantener el brazo en el campo de visión de una cámara o minimizar el uso de los GDL vehículo. Esta última tarea, por ejemplo, resulta muy interesante ya que permite reducir el consumo de energía y aumentar la precisión del sistema.

6.5. Manipulación submarina: Aplicación para el desenterrado

Con el objetivo de realizar la manipulación de objetos en el entorno submarino se ha estudiado el estado del arte y se han desarrollado los componentes del sistema presentados previamente. En la siguiente sección se presentan los resultados obtenidos con el sistema. No obstante, el sistema puede ser utilizado también para otros usos. En esta sección se propone una aplicación del *framework* a la limpieza de objetos que están parcialmente cubiertos de forma autónoma [Fernández et al.,

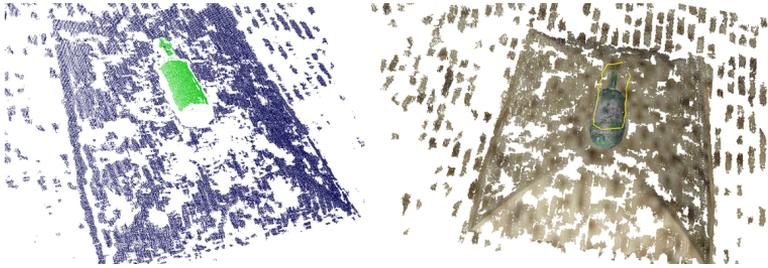


Figura 6.4: Segmentación del objeto (izquierda, verde) y extracción del fondo (izquierda, azul). Trayectoria generada a partir de la nube de puntos 3D utilizando los métodos de proyección y *concave hull* (derecha, amarillo).

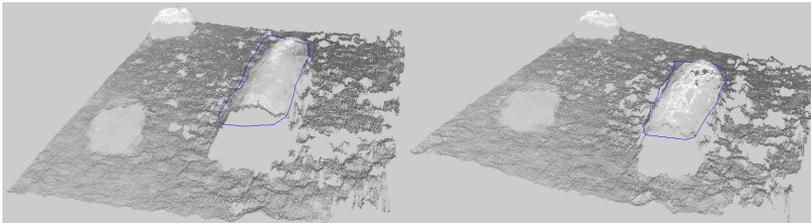


Figura 6.5: Obtención de la trayectoria de desenterrado para los objetos K y L (azul).

2015]. Esta aplicación es uno de los objetivos específicos del proyecto MERBOTS, que muestra un ejemplo de misión que puede querer realizar un arqueólogo. Para ello, se acopla una herramienta capaz de succionar el lodo que cubre la pieza al efector final del robot. La propia herramienta resulta un desafío mecatrónico fuera del alcance de esta tesis y no ha sido posible realizar una intervención real con ella. En cambio, la generación de la trayectoria que debe seguir la herramienta resulta interesante como aplicación de la metodología presentada.

El algoritmo utilizado para obtener la trayectoria que debe seguir el manipulador consiste en una serie de pasos análogos a los utilizados en la manipulación:

1. Reconstrucción 3D de la zona de interés, indicada por el arqueólogo, mediante visión estéreo.
2. Segmentación del objeto. Para procesar la escena se utiliza la segmentación del plano y del objeto basadas en RANSAC, descritas previamente. De esta forma, los puntos que representan el fondo son eliminados y los correspondientes al objeto son extraídos (ver Figura 6.4). En este caso, el objeto es una ánfora cubierta parcialmente de arena.
3. Proyección de la nube de puntos correspondiente al objeto sobre el plano que caracteriza el fondo marino.
4. Cálculo del *Concave Hull* o envolvente convexa de la proyección de puntos. Así, se genera de forma automática la trayectoria a seguir por el efector final

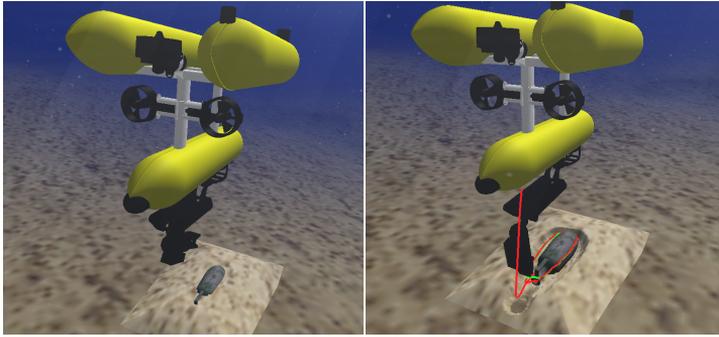


Figura 6.6: Estado final e inicial de la intervención de desenterrado.

del manipulador del H-ROV, donde está instalada la herramienta de dragado. Esta trayectoria está formada por una serie de puntos de ruta obtenidos con respecto a la cámara $Traj_c = (cMp_1, cMp_2, \dots, cMp_n)$. En la práctica, se ha optado por distanciar la trayectoria ligeramente del fondo para no entrar en contacto con el objeto.

5. Cálculo de la orientación de los puntos. Para obtener una trayectoria completa, se define la orientación adecuada en cada punto de la trayectoria. Para ello, se dirige el eje z de cada punto de ruta hacia el centroide de la nube de puntos del objeto proyectada. De esta manera, la herramienta siempre está en dirección al objeto.
6. Transformación de los puntos de ruta al sistema de coordenadas del mundo (wMp_n).

El resultado de este proceso es la trayectoria a seguir por la herramienta en forma de puntos de ruta. Cada uno de ellos se envía secuencialmente al controlador del robot, en ese caso el controlador basado en prioridad de tareas de [Peñalver et al., 2018]. Se ha calculado la ruta tanto en simulación (ver Figura 6.4) como con reconstrucciones reales (ver Figura 6.5).

Siguiendo la línea del trabajo realizado recientemente, la trayectoria a seguir por el efector final debe ser supervisada por un usuario utilizando la reconstrucción 3D del entorno. De esta manera, se puede asegurar que el objeto, que puede ser valioso, no sufre daños. Alternativamente, se ha descrito previamente cómo la interfaz permite al usuario definir de forma manual una serie puntos de ruta que el robot debe seguir en el proceso de desenterrado del ánfora. De esta forma, se provee al usuario una herramienta de alto nivel para definir mejor el camino para retirar la arena que cubre el objeto, en caso de que no esté satisfecho con el que propone el sistema ¹.

Para la simulación de la trayectoria, fuera del contexto de esta tesis, se ha utilizado Simurv [Antonelli, 2014], una librería de cinemática y dinámica para el control de UVMS disponible en Simulink (MATLAB). Simula el movimiento de

¹Intervención de limpieza de una vasija en simulación: <https://www.youtube.com/watch?v=sCoYF5WSI8Q>

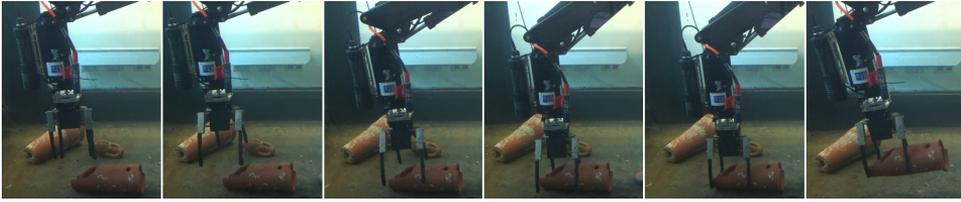


Figura 6.7: Secuencia de ejecución de la manipulación para el objeto C en el laboratorio.

vehículos con un manipulador en entorno submarino. Se utiliza para complementar a UWSim simulando la dinámica de un robot compuesto por un vehículo y un manipulador. El vehículo GIRONA500 [Ribas et al., 2012] proporciona 4 GDL al robot mientras que el ARM5E [Fernández et al., 2013] acoplado aporta 4 GDL. El vehículo se mueve utilizando una estrategia *free-floating* que utiliza todos los GDL del vehículo y el brazo para controlar el movimiento de la herramienta de dragado para seguir la trayectoria deseada. En la Figura 6.6 se muestra la trayectoria seguida por el efector final y el estado final de la escena tras realizar la limpieza siguiendo el camino propuesto [Fernández et al., 2015].

6.6. Resultados de Ejecución

En esta sección, se presentan los resultados correspondientes a los distintos experimentos realizados. El robot y los algoritmos desarrollados en esta tesis han sido probados satisfactoriamente en el tanque de agua (Figure 2.3), en la piscina del CIRS y en el mar. Cada experimento detalla los módulos involucrados.

Se han realizado experimentos de manipulación con múltiples tipos de vasijas, ánforas, objetos aproximadamente esféricos y cajas en el laboratorio del IRS Lab. Este entorno es el más controlado y permite realizar experimentos de forma continua. Por el contrario, en la piscina y en el mar sólo ha sido posible realizar los experimentos programados en el contexto de un proyecto mayor, el proyecto MERBOTS, por lo que se han realizado únicamente de experimentos de manipulación de una vasija.

En cuanto a ejecución del agarre se refiere, el simulador sirve para obtener la posición del agarre, visualizarla y ver dónde se posiciona el manipulador. Sin embargo, es poco útil para evaluar la posición de agarre resultante puesto que, como se ha explicado, la capacidad de simulación física de la manipulación no es adecuada. Así, permite probar las técnicas de segmentación, agrupamiento, simetrías, estimación y seguimiento con un conjunto de datos sintético.

6.6.1. Experimentos de manipulación en el laboratorio IRS Lab

Los experimentos de Ejecución se han utilizado para demostrar el funcionamiento del sistema completo de forma coordinada. La realización de estos experimentos muestra el funcionamiento conjunto de los diferentes algoritmos, así como los mecanismos desarrollados para la coordinación entre módulos. La mecatrónica

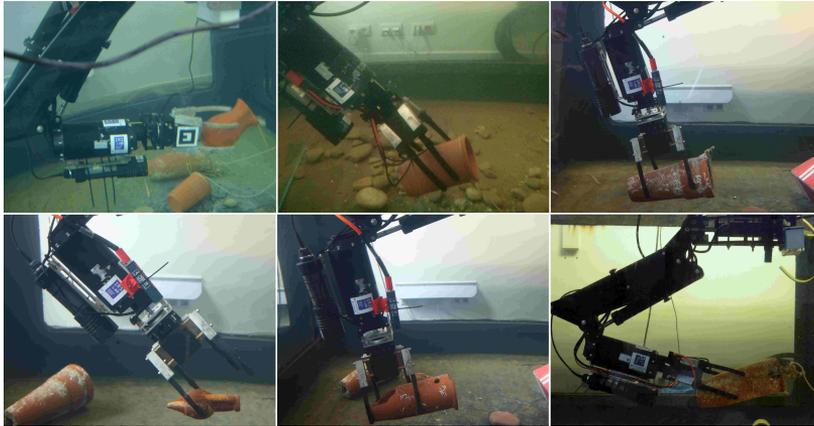


Figura 6.8: Experimentos de manipulación en el IRS Lab. Agarre de diferentes objetos.

utilizada en los experimentos en el laboratorio, detallada en el Capítulo 2, consiste en el brazo ARM5E anclado a una base fija, equipado con el efector final en forma de garra, la cámara estéreo Videre (puesto que ha demostrado ser superior a la DUO M) y un ordenador externo para realizar los cálculos más intensivos. El escenario es la cuba del IRS Lab con un fondo negro con arena y piedras para proporcionar un fondo que no sea totalmente plano. La Figura 6.7 muestra las diferentes etapas del proceso, que se pueden ver en movimiento en el siguiente vídeo: <https://youtu.be/8wSd33c4j3g>.

Los resultados de diferentes agarres realizados con el sistema propuesto se muestran en la Figura 6.8. En ella se observa la posición de agarre final para diferentes objetos. La posición de agarre resultó adecuada para la mayoría de objetos. Sin embargo, en el caso del objeto más pequeño (objeto *G*) la especificación autónoma no es suficiente para obtener un agarre satisfactorio. Por este motivo, el usuario introduce su conocimiento para ubicar el brazo de forma precisa y coger el objeto. El objeto *H* no se puede manipular por sus dimensiones, aunque ha sido de gran utilidad a la hora de evaluar los algoritmos. Los objetos *K* y *L*, por estar deteriorados, tampoco pueden ser manipulados. No obstante, suponen un *benchmark* muy interesante ya que se asemejan a un objeto enterrado.

En la Figura 6.7 se muestra la secuencia de agarre del objeto *C* en el laboratorio. En estas imágenes, extraídas del vídeo de la ejecución del agarre, se puede ver como el robot pasa por las diferentes etapas definidas al inicio de este capítulo, como son: *Inicio*, *Apertura*, *Aproximación*, *Agarre*, *Cierre* y *Transporte*. En la Figura 6.9 se muestra la secuencia de agarre del objeto *D* en el laboratorio, en la que la fase de *Apertura* se omite, ya que el efector final parte con la apertura deseada desde la posición de inicio.

La trayectoria seguida por el efector final durante la manipulación del objeto *D* se muestra en la Figura 6.10. En ella se puede ver dicha trayectoria con respecto a la posición de agarre especificada. La posición de agarre se actualiza continuamente en función de cada nueva detección del objeto. El proceso tiene una duración de 75 segundos. Se pueden observar tres fases, que corresponden algunas de las indicadas

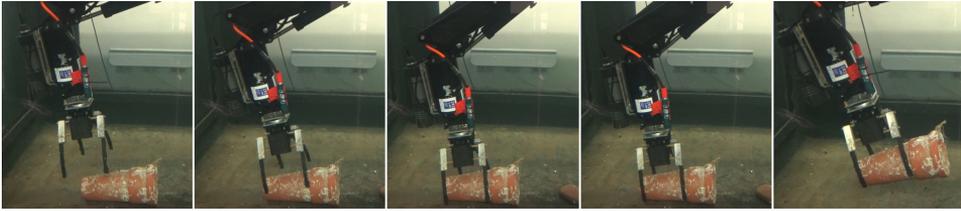


Figura 6.9: Secuencia de ejecución de la manipulación para el objeto D en el laboratorio.

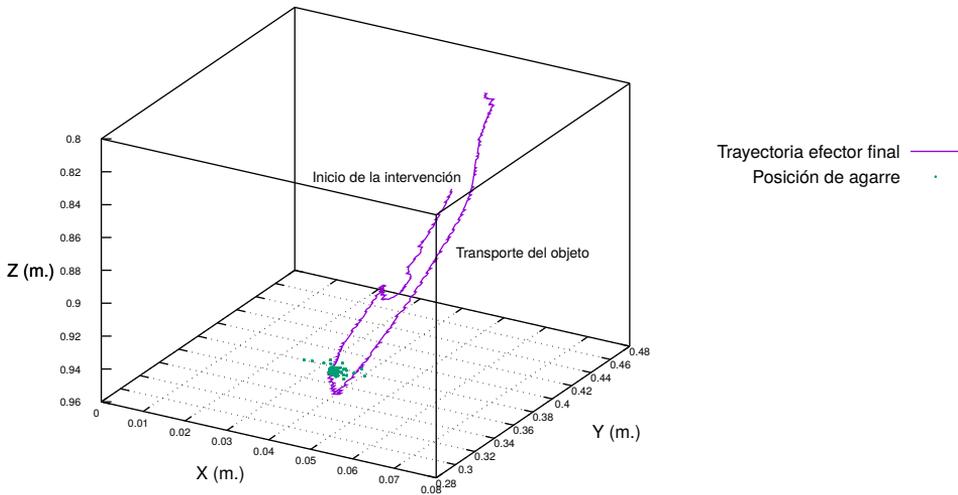


Figura 6.10: Trayectoria seguida por el manipulador respecto a la posición de agarre deseada en el laboratorio para el objeto D.

previamente. Las fases de *Apertura* y *Cierre* del manipulador no son observables en esta representación, mientras que la fase de *Inicio* se omite. En primer lugar, el manipulador se aproxima al objeto y se detiene. Después se dirige a la posición de agarre comandada para cerrar la garra. Finalmente se transporta el objeto a la posición final.

6.6.2. Experimentos de manipulación en el CIRS

En esta sección se detallan los experimentos de manipulación de una vasija en la piscina del CIRS de Gerona en el contexto del proyecto MERBOTS. En concreto, se enmarcan en el subproyecto MERMANIP dedicado a la manipulación. La mecatrónica utilizada en estos experimentos, detallada en el Capítulo 2 (ver Figura 6.11), consiste en el brazo ARM5E acoplado a la base del AUV GIRONA500, equipado con el efector final en forma de garra, la cámara estéreo BumbleBee 2,

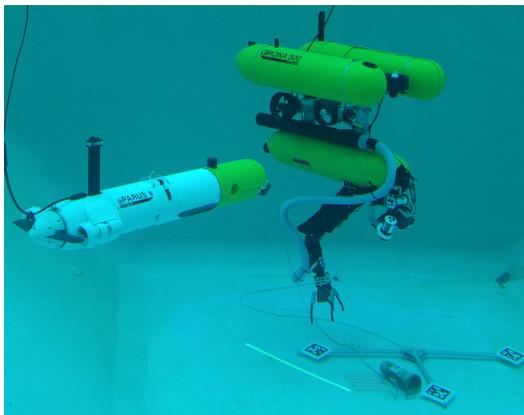


Figura 6.11: Experimentos en el CIRS. El GIRONA500 dispone del ARM5E acoplado, cámara estéreo y sensor láser 3D. El SPARUS II se utiliza en el MERBOTS como apoyo.

un sensor 3D basado en láser y un ordenador dedicado para realizar los cálculos más intensivos en el interior del vehículo. El escenario es la piscina del CIRS donde se sitúa una vasija cilíndrica y un grupo de marcadores que ayudan a los sistemas de posicionamiento del vehículo, que no pueden funcionar correctamente en aguas cerradas. Así mismo, el objeto dispone de múltiples marcadores para su ubicación, sirviendo de referencia para los realizar comparativas. Respecto a los experimentos en el laboratorio, estas pruebas hacen uso del I-AUV al completo por lo que no se tienen las mismas restricciones de GDL y el control del robot es más complejo. Los experimentos en la piscina tuvieron lugar durante febrero de 2017. El AUV SPARUS II también fue parte de la misión realizando tareas de supervisión. No obstante, su uso no interactúa con el sistema propuesto y está fuera del contexto de este trabajo.

En primer lugar, se hizo uso únicamente del método de segmentación de cilindros mediante RANSAC para establecer el objetivo de la manipulación, ya que el resto de métodos se desarrollaron con posterioridad y han sido probados en el laboratorio. A continuación, se utilizó la interfaz mostrada en el Capítulo 5 para visualizar los resultados de la estimación, realizar la especificación heurística del agarre y poner en marcha la manipulación (ver Figura 6.13). Durante la manipulación la posición de agarre se actualiza de forma constante realizando el seguimiento continuo del objeto. Por otro lado, el control de la tarea a nivel de manipulación de Peñalver y Fernández [Peñalver et al., 2018], los algoritmos para la navegación, el sistema láser utilizado [Palomer et al., 2018], y otros métodos de detección [Joan et al., 2017] alternativos, están fuera del contexto de esta tesis. En esta figura se muestra la secuencia desde la captura de la imagen, la reconstrucción 3D, la especificación al agarre resultante (ver Figura 6.13).

La posición de agarre especificada para manipular el objeto y la trayectoria seguida por el efector final durante los experimentos en la piscina se muestran en la Figura 6.12. En esta figura se muestra la trayectoria seguida durante una intervención de 70 segundos de duración. Se pueden observar tres etapas. En la

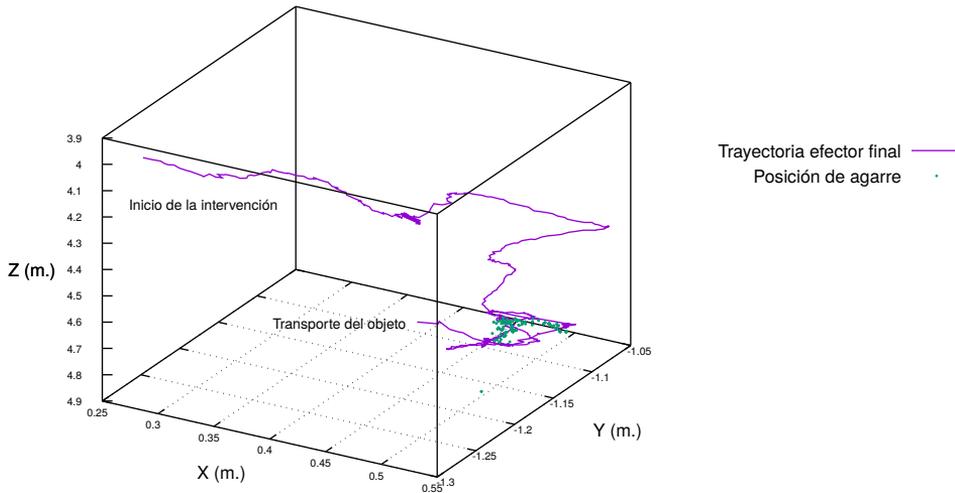


Figura 6.12: Trayectoria del manipulador respecto a la posición de agarre deseada en la piscina.

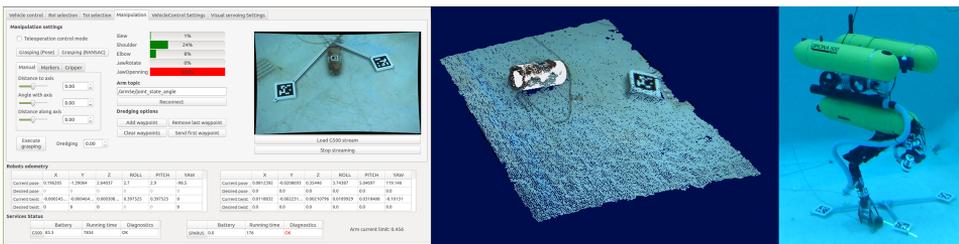


Figura 6.13: Manipulación en el CIRS. Estimación, Especificación, Supervisión y Ejecución del agarre.

primera, el robot se aproxima rápidamente a la posición de agarre deseada desde un punto alejado del fondo. Después, el I-AUV trata de estabilizarse en la posición deseada para cerrar el efector final y coger el objeto. Dicha fase de estabilización, como es lógico, no es necesaria en los experimentos en el laboratorio. Una vez ha sido cogido, se ve la fase final en la que se transporta el objeto. La posición de agarre deseada se actualiza en función de las detecciones del objeto mediante RANSAC. Este escenario es ideal para evaluar el sistema final antes de realizar pruebas en el entorno más complejo como es el mar.

Estos experimentos se han realizado utilizando tanto visión estéreo como el sensor 3D desarrollado en la UdG. Este doble experimento, junto con los experimentos en la laboratorio y simulación, demuestran la posibilidad de utilizar el sistema con información procedente de distintas fuentes de datos 3D, con diferentes resoluciones, precisión y naturaleza.



Figura 6.14: Experimentos en el mar. El AUV GIRONA500 dispone del ARM5E acoplado y cámara estéreo. El AUV SPARUSII se utiliza como apoyo.

6.6.3. Experimentos de manipulación en el mar

Los experimentos propuestos más desafiantes se realizaron en el interior del puerto de Sant Feliu de Guíxols, como parte de los experimentos finales del proyecto MERBOTS. De forma muy similar a los experimentos en piscina, la mecatrónica utilizada en estas pruebas consiste en el brazo ARM5E acoplado en la base del GIRONA500, equipado con el efector final en forma de garra, la cámara estéreo BumbleBee2 y el ordenador dedicado para realizar los cálculos más intensivos en el interior del vehículo. A 10 m. de profundidad se dispuso el escenario, donde se situó una vasija cilíndrica y un grupo de marcadores. En esta ocasión, la estructura con marcadores no es estrictamente necesaria para la odometría del vehículo, pero se introdujo para demostrar distintos modos de navegación. Igualmente, el objeto dispone de los marcadores para determinar su ubicación, sirviendo de referencia para los experimentos. En la Figura 6.14 se muestra el escenario y el robot utilizado. De nuevo, se puede observar el uso del AUV SPARUS II. Los experimentos en el mar tuvieron lugar en marzo de 2017.

De forma análoga a los experimentos en el CIRS, en el experimento que se muestra en la Figura 6.15 se utilizó la interfaz de control para la realización de la estimación basada en RANSAC, la especificación heurística del agarre y la supervisión de la tarea. De nuevo, el control de la tarea a nivel de manipulación y navegación están fuera del contexto de esta tesis. En esta figura se muestra la secuencia desde la captura de la imagen, la reconstrucción 3D, la especificación al agarre resultante. En esta ocasión, sin embargo, únicamente se dispone de la visión estéreo.

La trayectoria del manipulador durante los experimentos en el mar se muestra en la Figura 6.16. En dicha figura se representa el movimiento del efector final durante una intervención de 75 segundos de duración. Además, se puede ver la posición de agarre deseada en función de las detecciones del objeto. Se aprecian las tres fases citadas previamente. Primero el robot se aproxima a la posición de agarre. En la fase siguiente, el I-AUV trata de estabilizarse sobre la posición deseada para cerrar el efector final y coger el objeto. Una vez ha sido cogido con éxito, se aprecia la fase final de transporte del objeto. La posición de agarre deseada varía debido a que la estimación de la posición detectada del objeto no es siempre en el mismo

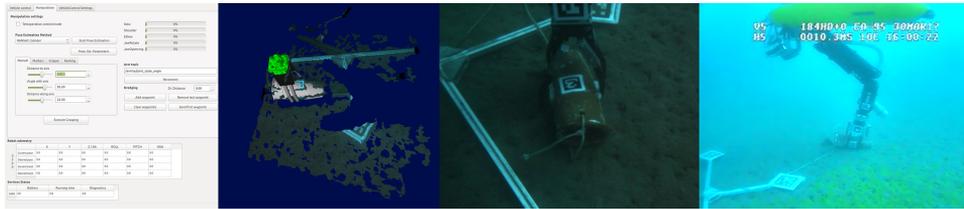


Figura 6.15: Manipulación en el mar. Estimación, especificación, supervisión y ejecución del agarre.

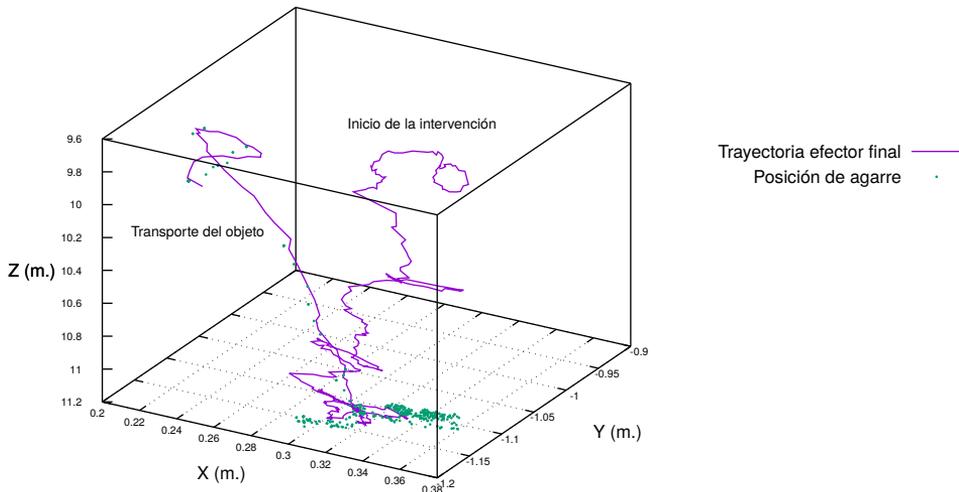


Figura 6.16: Trayectoria del manipulador respecto a la posición de agarre deseada en el mar.

lugar preciso. Además, la odometría del vehículo presenta una pequeña deriva con el tiempo. Por estos motivos es muy importante realizar la detección continua del objeto.

En resumen, los experimentos han sido realizados con éxito en los escenarios del laboratorio del IRS Lab, la piscina del CIRS y el mar en Sant Feliu de Guíxols. Así, se ha seguido la metodología iterativa propuesta en el Capítulo 2 y demostrado el funcionamiento del sistema con la retroalimentación que provee el simulador. Esta metodología ha permitido realizar mejoras al sistema progresivamente y conocer las limitaciones de cada uno de los módulos en los diferentes entornos. Como trabajo futuro, sería muy interesante evaluar el sistema de manipulación integral desarrollado en experimentos de integración con un mayor número de objetos con diferentes formas, evaluando el funcionamiento de los algoritmos de reconstrucción, estimación y especificación, disponiendo de un UVMS completo.

Conclusiones y Líneas de futuro

El presente trabajo ha mostrado un sistema integral capaz de manipular objetos cuyo modelo es desconocido en el contexto de una intervención supervisada. En un medio en el que los trabajos disponibles se centran en las complejidades de control de los robots, se ha aportado una visión que se centra en aprovechar la percepción 3D del entorno y la capacidad del usuario para supervisar el sistema. Esta propuesta toma como caso de uso las misiones en la arqueología submarina. En este capítulo se remarcan los aspectos más importantes de este trabajo.

En primer lugar se introducen las conclusiones extraídas de este trabajo en el contexto de la visión artificial y la manipulación en entornos subacuáticos. En segundo lugar, se discuten las posibles aplicaciones del sistema presentado. A continuación, se sintetizan las contribuciones de esta tesis y se muestran las publicaciones que ha dado como resultado el trabajo presentado. Finalmente, se realiza una discusión de las líneas de futuro que abre este trabajo. En este sentido, el proyecto nacional TWINBOT resulta una línea de trabajo natural, ya que profundiza en problemas que han sido comentados en capítulos previos, así como en las conclusiones.

7.1. Conclusiones

El objetivo de esta tesis ha sido el de ofrecer un **sistema integral de manipulación guiado por visión**. Este *framework* permite al usuario la manipulación de objetos con diferentes grados de **autonomía**. Como se muestra en los resultados de los capítulos 4 a 6, el sistema consigue su objetivo, si bien no está exento de limitaciones y se ha utilizado en escenarios restringidos. En esta sección se detallan las conclusiones de este trabajo.

Los algoritmos de reconstrucción 3D en el medio subacuático cuentan cada vez con mejores posibilidades. Por ejemplo, al inicio de esta tesis se disponía de un sistema de reconstrucción láser lento e impreciso. Sin embargo, durante el transcurso de este tiempo han aparecido sistemas cada vez mejores capaces de reconstruir una escena en décimas de segundo con precisión [Palomer et al., 2018]. No obstante, son todavía dispositivos en la frontera de la tecnología. En este trabajo se ha mostrado la importancia de disponer de una reconstrucción de calidad y un sistema de

filtrado que la mejore.

En cuanto a la estimación de la geometría de los objetos, una herramienta muy importante ha resultado ser la búsqueda de simetrías. Los algoritmos de **búsqueda de simetrías** han sido utilizados en el entorno de la robótica doméstica con éxito. En este trabajo, han sido aplicados en el ámbito submarino para facilitar la posterior reconstrucción de los objetos, que sería mucho más compleja sin esta técnica.

En este trabajo se ha optado por utilizar técnicas cuya robustez ha sido demostrada como **RANSAC**. Este algoritmo presenta limitaciones, como que es capaz de detectar únicamente un tipo de objeto al mismo tiempo. Sin embargo, se trata de un algoritmo muy eficiente y capaz de funcionar en presencia de ruido notable. Además, no necesita disponer del objeto aislado del resto de la escena. Este método ha sido utilizado con éxito en algoritmos diferentes para la detección de cilindros, esferas y cajas.

Por otro lado, la estimación utilizando **SQ** proviene del ámbito de los gráficos por computador y ha sido menos utilizada en la robótica. Se trata de un método lento pero muy flexible, ya que es capaz de representar infinidad de formas geométricas de forma muy compacta y utilizando un único método de estimación. Por otro lado, se trata de un método que necesita disponer de los puntos pertenecientes a un objeto separados del resto de la escena y es sensible al ruido. Se ha utilizado la estimación de momentos de inercia basada en **PCA** para obtener una estimación del objeto muy rápida. Este método es de gran utilidad para comparar los diferentes métodos. Además, en un contexto en el que se dispone de la ayuda del usuario, este método resulta adecuado para tener una primera aproximación. En proyectos como RAUVI, por ejemplo, se descarga toda la responsabilidad de la especificación en el usuario.

Los métodos de estimación propuestos incluyen un sistema de **seguimiento** capaz de obtener la posición y orientación del objeto de forma continua. Esta capacidad ha demostrado ser de utilidad a la hora de superar las dificultades en los escenarios reales, como es la baja visibilidad en el mar.

El sistema de percepción desarrollado ha conseguido el objetivo de **estimar la geometría de una escena** correctamente, ya sea con uno o **múltiples objetos**. Se ha adoptado la condición de que el fondo de la escena sea relativamente plano, lo cual es una restricción del sistema importante. Sin embargo, esta limitación es necesaria y ha demostrado no ser un impedimento para llevar a cabo los experimentos propuestos.

Se ha propuesto un método de **especificación** del agarre heurístico basado en la geometría del objeto estimada. Además, el sistema es capaz de obtener la caracterización del agarre con un algoritmo de **análisis**, que provee un sistema para clasificar las hipótesis de agarre obtenidas. Estos métodos permiten ofrecer al usuario diversas maneras de obtener una posición de agarre adecuada.

El sistema **HRI** desarrollado en esta tesis ha demostrado ser de utilidad en un proyecto real como es MERBOTS. Además, ha sido probado en multitud de experimentos en el laboratorio de IRS Lab. La **GUI** desarrollada ofrece una variedad de utilidades para llevar a cabo tareas como manipulación de objetos, teleoperación del robot, guiado del robot a través de puntos de ruta o supervisión de la misión.

La **ejecución** de la manipulación ha sido llevada a cabo en el laboratorio con el sistema de control en cascada desarrollado, capaz de ofrecer diferentes modos de

interacción con el robot. En los experimentos de integración, el sistema ha utilizado un controlador basado en prioridad de tareas que no forma parte de esta tesis.

Se han propuesto experimentos con niveles de complejidad incrementales, demostrando el uso de los **sistemas de percepción y manipulación**. En particular, la percepción y especificación ha sido probada en simulación. El sistema completo ha sido capaz de determinar el agarre para una vasija en la piscina del CIRS y en el mar. Finalmente, en el laboratorio del IRS Lab se ha probado el sistema de forma integral con múltiples objetos, demostrando su idoneidad para la tarea propuesta. Así, este trabajo ha cumplido uno de los objetivos subyacentes como es el de **aumentar los niveles de autonomía** de los sistemas de intervención submarinos.

7.2. Aplicaciones potenciales

En base a estas conclusiones, se puede ver que este trabajo tiene potencial para su aplicación en diversos campos. A continuación se muestran posibles aplicaciones del sistema y los algoritmos desarrollados, así como las mejoras introducidas pueden ayudar en estos ámbitos:

- **Arqueología submarina:** es el caso de uso propuesto en este trabajo y una vez superadas con éxito las pruebas de campo, resultaría de gran interés colaborar con arqueólogos para aplicar a sus problemas cotidianos la metodología presentada. En este sentido se han realizado esfuerzos en poner en conocimiento de esta comunidad los avances realizados en este área y se ha obtenido retroalimentación con la que mejorar los robots para hacerlos más útiles en escenarios reales. Sin embargo, los requisitos de precisión y madurez de los métodos propuestos hacen que sea necesario avanzar más en el estado del arte de la mano de expertos en este contexto.
- **Recuperación de objetos:** es un campo de aplicación más general que la propia arqueología. El escenario original de RAUVI y TRIDENT es la recuperación de la caja negra de un avión. A parte de este escenario excepcional, el sistema puede ser adaptado para recuperar otro tipo de objetos de interés en otros campos. Por ejemplo, es posible su uso para recuperar un equipo de medición de datos del fondo marino, artefactos en el fondo de un puerto o muestras de material en el contexto la geología marina.
- **Operación de observatorios marinos:** el escenario de manipulación de un panel submarino no ha sido tratado en detalle en este trabajo, sino que es el escenario principal del TRITON. Las técnicas de estimación propuestas no parecen muy útiles en este escenario. No obstante, el tipo de formas geométricas que el sistema es capaz de reconocer pueden resultar adecuadas a la hora de detectar este tipo de estructuras hechas por el hombre. Conectores, válvulas y tuberías son estructuras que pueden ser reconocidas adaptando el sistema.
- **Operación en la industria petrolífera:** Este escenario está orientado típicamente a ROV pesados que pueden levantar grandes cargas, utilizar potentes

herramientas y sumergirse miles de metros. Sin embargo, algunas de las tareas de mantenimiento llevadas a cabo en plataformas petrolíferas se realizan a baja profundidad, por lo que también resultaría una potencial aplicación. En concreto, es un entorno en el que sería interesante la introducción de funciones autónomas en los vehículos utilizados. De esta manera, sería posible aliviar la fatiga cognitiva que sufren los pilotos.

7.3. Contribuciones

En esta sección se sintetizan los aspectos que, en opinión del autor, resultan las aportaciones más interesantes de este trabajo:

- Se ha realizado un amplio estado del arte en robótica y en su vertiente subacuática. En este sentido, se ha hecho hincapié en el acercamiento a la manipulación de objetos y en las técnicas de estimación de la geometría de los objetos y la especificación del agarre. También se ha repasado el estado del arte en arqueología submarina, así como en el uso de HRI en el medio submarino.
- Estudio sobre reconstrucción y registro de nubes de puntos en 3D en el medio subacuático.
- Métodos de estimación de la geometría de la escena y los objetos multimodal en la que se proponen estrategias complementarias para la detección. Esta metodología incluye la búsqueda de simetrías de los objetos, la estimación de formas primitivas mediante RANSAC y la estimación mediante SQ como elementos más interesantes aplicados al reconocimiento de objetos en el entorno submarino.
- Métodos de especificación autónoma del agarre de forma heurística y analítica, resultando una metodología novedosa en el entorno subacuático.
- Sistema de HRI basado en una GUI intuitiva en la que se ofrece al usuario la capacidad de supervisar e intervenir en el sistema a un alto nivel, sin tener que tomar el control del vehículo necesariamente. Además, se ha propuesto el uso de la herramienta UWSim tanto para la toma de decisiones en la manipulación como para la visualización del sistema en tiempo de ejecución. Esta visualización 3D mejora la inmersión del usuario y su percepción del escenario.
- Sistema de manipulación completo desde la adquisición de la información de los sensores hasta el transporte del objeto. Este enfoque integral no ha sido abordado frecuentemente en la robótica submarina. Además el enfoque propuesto hace especial énfasis en un diseño modular y adaptable para que pueda ser utilizado en diversos entornos con diferentes configuraciones de robots.

7.4. Publicaciones científicas

Las metodologías presentadas en esta tesis han sido aceptadas y publicadas en diferentes revistas científicas y congresos de carácter nacional e internacional. Algunas de estas publicaciones detallan el funcionamiento completo del sistema, mientras que en otras se muestran partes de este trabajo en el contexto de otro proyecto. Los trabajos publicados en revistas de impacto son los siguientes:

1. **Fornas, D.**, Sales, J., Peñalver, A., Pérez, J., Fernández, J. J., Marín, R., Sanz, P. J., *Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects*. Journal of Experimental & Theoretical Artificial Intelligence, pp. 369 - 384. Reino Unido. 2016.
Factor de impacto: 1,384
2. Peñalver, A., Pérez, J., Fernández, J. J., Sales, J., Sanz, P. J., García, J. C., **Fornas, D.**, Marín, R., *Visually-guided manipulation techniques for robotic autonomous underwater panel interventions*. Annual Reviews in Control. 40, pp. 201-211. Reino Unido. 2015.
Factor de impacto: 2,042
3. Pérez, J., Sales, J., Peñalver, A., **Fornas, D.**, Fernández, J. J., García, J. C., Sanz, P. J., Marín, R., Prats, M., *Exploring 3-d reconstruction techniques: A benchmarking tool for underwater robotics*. IEEE Robotics & Automation Magazine, 22(3):85-95. 2015.
Factor de impacto: 1,822
4. Perez, J., **Fornas, D.**, Marín, R., Sanz, P. J., *UWSim, un simulador submarino conectado a la nube como herramienta educativa*. Revista Iberoamericana de Automática e Informática Industrial RIAI. Elsevier. vol. 15, núm. 1, 2018.
Factor de impacto: 0,471

Además, se han publicado artículos de investigación en congresos internacionales siguientes:

1. **Fornas, D.**, Sanz P. J., Vincze, M., *Increasing underwater manipulation autonomy using segmentation and visual tracking*. IEEE/MTS Oceanic Engineering Society Conference (OCEANS). Aberdeen, Reino Unido. 2017.
2. Fernández, J. J., Pérez, J., Peñalver, A., Sales, J., **Fornas, D.**, Sanz, P. J., *Benchmarking using UWSim, Simurv and ROS: An autonomous free floating dredging intervention case study*. IEEE/MTS Oceanic Engineering Society Conference (OCEANS). Génova, Italia. 2015.
3. **Fornas, D.**, Sales, J., Peñalver, A., Pérez, J., Fernández, J. J., Sanz, P. J., *Improving autonomous underwater grasp specification using primitive shape fitting in point clouds*, en 17th International Conference of the Catalan Association for Artificial Intelligence (CCIA) Barcelona, España. 2014.

4. Perez, J., Sales, J., Peñalver, A., Fernández, J.J., **Fornas, D.**, García, J. C., Marín, R., Sanz, P.J., *Benchmarking water turbidity effect on tracking algorithms*. 20th IFAC World Congress, Toulouse, Francia. 2017.
5. Sanz P.J., Peñalver A., Sales, J., **Fornas, D.**, Fernández, J. J., Pérez, J., Bernabé, J.A., *GRASPER: A Multisensory Based Manipulation System for Underwater Operations*. IEEE International Conference on Systems, Man, and Cybernetics, Manchester, Reino Unido. 2013.
6. Perez, J., Sales, J., Peñalver, A., Fernández, J. J., Sanz, P. J., García, J. C., Martí, J. V., Marín, R., **Fornas, D.**, *Robotic manipulation within the underwater mission planning context*. Motion and Operation Planning of Robotic Systems, pp. 495-522. Springer. 2015. Capítulo de libro.

En el ámbito nacional, se han publicado los siguientes artículos:

1. **Fornas, D.**, Fernández, J. J., Soriano, A., Centelles, D., Sanz, P. J., *Avances en manipulación robótica submarina mediante segmentación y seguimiento visual*. XXXIX Jornadas de Automática, Badajoz, España. Comité Español de Automática. 2018.
2. **Fornas, D.**, Sanz, P. J., Porta, J. M., Thomas, F., *Improving local symmetry estimations in RGB-D images by fitting superquadrics*. XXXVII Jornadas de Automática, Madrid, España. Comité Español de Automática. 2016.
3. Sanz, P. J., Marín, R., Peñalver, A., **Fornas, D.**, Centelles, D., *Merbots Project: Overall description Multisensory autonomous perception and grasping for underwater robotics interventions*. XXXVIII Jornadas de Automática. Gijón, España pp. 232 - 237. Universidad de Oviedo. 2017.
4. Perez, J, **Fornas, D.**, Marín, R., Sanz, P. J., (2016). UWSim, un simulador submarino como herramienta educativa. En *XXXVII Jornadas de Automática, Madrid*.
5. Sanz, P., Perez, J., Sales, J., Fernández, J., Peñalver, A., **Fornas, D.**, García, J. C., Marín, R., *Meromanip: Avances recientes en la manipulación autónoma cooperativa submarina*. XXXVI Jornadas de Automática, Bilbao, España. 2015.

Aparte de estas publicaciones, se está preparando una publicación de revista con los resultados más recientes de la investigación.

7.5. Líneas de futuro

Este trabajo ha supuesto un pequeño avance para la robótica submarina. No obstante, la diferencia entre las posibilidades de manipulación entre la robótica general y la subacuática es todavía abismal. Algunas técnicas no son directamente aplicables como, por ejemplo, la visión basada en luz infrarroja. Otras, sin embargo, pueden ser adaptadas a este medio con un esfuerzo de desarrollo y en muchas ocasiones un esfuerzo económico. En esta sección se exponen líneas de trabajo interesantes, en opinión del autor, en diferentes frentes.

Contexto del proyecto TWINBOT

El proyecto nacional TWINBOT, introducido en el Capítulo 1, trata la manipulación cooperativa entre dos I-AUV para realizar el transporte de una tubería y un ensamblaje estilo *peg-in-hole*. En primer lugar, resulta un desafío ya que el IRS Lab asume la construcción de un brazo robótico de 6 GDL. Disponer de un brazo robótico solucionaría algunos de los problemas. Se mejoraría la manipulabilidad general del sistema y no sería necesario recurrir a los grados de libertad del vehículo. No existen todavía alternativas comerciales que ofrezcan manipuladores diseñados para ser usados de forma autónoma con un coste que permita utilizarlos en el ámbito científico.

El desafío que plantea cooperación entre dos vehículos se encuentra fuera del alcance de esta tesis. Sin embargo, la percepción del entorno sí está muy relacionada. El escenario propuesto contiene estructuras industriales que deben ser reconstruidas y algunos de los objetos a manipular son tuberías. Las metodologías mostradas en el Capítulo 4 podrían ser mejoradas y expandidas en este contexto.

Un aspecto fundamental, no considerado en esta tesis, sería la forma de compartir información sobre el entorno entre los dos vehículos. Es decir, la construcción de un modelo del mundo. Este es un desafío recurrente en la robótica cooperativa [Roth et al., 2003]. Debido a las limitaciones en las comunicaciones submarinas, la posibilidad de compartir información geométrica entre dos robots parece muy adecuada. Utilizando los estimadores propuestos en este trabajo es posible reducir una nube de puntos a un número muy reducido de parámetros que representan la escena de forma muy compacta. En contraste a la transmisión de imágenes completas o nubes de puntos, incluso aplicando técnicas de compresión, el coste es mucho menor. Por otro lado, entre ambos vehículos podrían construir un mapa de ocupación del espacio dónde los objetos se representen por sus propiedades geométricas en vez de con modelos totalmente 3D o sus nubes de puntos correspondientes. Además, esta información geométrica podría ser acompañada de información semántica para ayudar en la toma de decisiones entre ambos vehículos [Li et al., 2017].

Aplicación a la arqueología submarina

Así mismo, es de gran interés presentar este sistema a arqueólogos, potenciales usuarios del sistema, para determinar su utilidad real y las necesidades concretas de este campo de investigación. El proyecto ARROWS es un ejemplo de cooperación entre usuarios potenciales e investigadores [Allotta et al., 2015]. De esta manera, es posible conocer las mejoras que necesitan los sistemas dedicados a la intervención en este contexto. En este sentido, se han realizado debates en torno a las capacidades técnicas de los robots. Por ejemplo, en el ámbito de la fotografía, un buceador con una cámara de alta calidad es capaz de tomar imágenes de un objeto de interés con unos resultados mucho mejores que un robot en la actualidad. Por lo tanto, sin tener en cuenta el factor de la profundidad, los robots submarinos tienen que mejorar sus prestaciones progresivamente para ser cada vez más útiles y facilitar su adopción.

Aplicación de tecnologías en el estado del arte

Como se ha explicado en las conclusiones, los sistemas de visión 3D están mejorando progresivamente y actualmente han aparecido tecnologías basadas en láser que mejoran en gran medida el estado del arte previo. Además, la tecnología que utilizan las cámaras mejora continuamente. Como consecuencia de la reducción en tamaño que sufren, es más sencillo integrarlas en un robot submarino. En este sentido, también ha aumentado la capacidad de proceso de los sistemas empujados y micro-controladores como Raspberry Pi o NVIDIA Jetson. Esta ganancia puede permitir que se puedan ejecutar los algoritmos de visión en equipos de este tipo. En la actualidad, como referencia, no es temporalmente factible ejecutar los algoritmos de visión en el ordenador que controla el ARM5E.

Al disponer de una representación 3D del entorno mejorada sería interesante implementar métodos de planificación de trayectorias que tuvieran en cuenta la presencia de obstáculos [Youakim et al., 2017]. Como se ha mencionado previamente el control dinámico de la cadena vehículo-manipulador es un gran desafío, y existe un gran interés en mejorar el estado del arte en este sentido con diferentes implementaciones de controladores adaptativos y de prioridad de tareas.

Por otro lado, es posible implementar métodos de control de la manipulación más avanzados haciendo uso de sensores de fuerza y táctiles. Estos sensores son ampliamente utilizados en la robótica móvil. No obstante, no son muy utilizados en la robótica submarina debido a que no se pueden utilizar por las propiedades físicas del medio (sensores basados en presión) o si se pueden utilizar su coste de integración es elevado. Han aparecido en la actualidad modelos de sensores que pueden ser utilizados. Su uso permitiría complementar la visión y la retroalimentación del brazo a la hora de determinar si un agarre ha sido satisfactorio o erróneo de forma rápida. Además, disponer de esta información permitiría implementar estrategias de control reactivas, que sería de gran utilidad también en el contexto del proyecto TWINBOT.

Mejoras en el sistema propuesto

En cuanto a la metodología de estimación de la geometría de la escena, la principal área de mejora es la extracción del fondo y el agrupamiento de objetos. En la actualidad, como en muchos proyectos que se llevan a cabo en el ámbito doméstico, se asume que el fondo es aproximadamente plano y fácil de segmentar. En la actualidad se están utilizando diferentes técnicas en el estado del arte para la extracción del fondo, así como técnicas de agrupamiento basadas en aprendizaje automático.

En este trabajo el cálculo de SQ se ha realizado en dos etapas: segmentación y extracción de las primitivas geométricas. Algunos autores [Biegelbauer y Vincze, 2007] [Pascoal et al., 2015] proponen realizar estas dos operaciones de forma simultánea. Con la metodología propuesta, por ejemplo, dos caras de una caja en ocasiones pueden ser detectadas como grupos de puntos diferentes y ser aproximadas con dos modelos. Utilizando una metodología integrada, se podría generar un único modelo. Además, existe la posibilidad de utilizar los parámetros ϵ_1 y ϵ_2 para determinar de qué tipo de objeto se trata. De esta forma, sería posible seleccionar entre diferentes algoritmos RANSAC implementados, para obtener una estimación

inicial flexible basada en SQ y capaz de realizar el seguimiento del objeto de forma rápida con RANSAC.

Por otro lado, este método podría ser de mayor utilidad si la estimación fuese más rápida. Para ello, es interesante considerar la posibilidad de implementar estos algoritmos de forma paralela, utilizando tarjetas gráficas o *hardware* especializado como una FPGA (Field Programmable Gate Arrays). Además, existen otros algoritmos la frontera del estado del arte que proponen nuevos métodos para la estimación de objetos desconocidos [Richtsfield et al., 2012]. El uso de estos algoritmos se hace posible gracias a la aparición de sensores 3D cada vez mejores.

En definitiva, la robótica actual en general y la manipulación en particular está avanzando a gran velocidad. Los desarrollos en robótica submarina toman cada vez más ideas de este campo, al tiempo que resuelven los problemas propios de medio subacuático. En los próximos años se resolverán muchos de los desafíos presentes en las intervenciones submarinas.

Bibliografía

- [Aldoma et al., 2012] Aldoma, A., Marton, Z. C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S., y Vincze, M. (2012). Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robotics Automation Magazine*, 19(3):80–91. (Citado en la páginas 19 and 20.)
- [Allotta et al., 2015] Allotta, B., Costanzi, R., Ridolfi, A., Colombo, C., Bellavia, F., Fanfani, M., Pazzaglia, F., Salvetti, O., Moroni, D., Pascali, M. A., Reggianini, M., Kruusmaa, M., SalumÃd'e, T., Frost, G., Tsiogkas, N., y Lane, D. M. (2015). The arrows project: adapting and developing robotics technologies for underwater archaeology. *IFAC-PapersOnLine*, 48(2):194 – 199. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV 2015. (Citado en la páginas 12 and 143.)
- [Angeles, 1985] Angeles, J. (1985). On the numerical solution of the inverse kinematic problem. *The International Journal of Robotics Research*, 4(2):21–37. (Citado en la páginas 32 and 126.)
- [Antonelli, 2014] Antonelli, G. (2014). Simurv 4.0. En *Underwater Robots*, páginas 257–265. Springer. (Citado en la página 129.)
- [Antonelli et al., 2008] Antonelli, G., Fossen, T. I., y Yoerger, D. R. (2008). Underwater robotics. En *Springer handbook of robotics*, páginas 987–1008. Springer. (Citado en la páginas 1, 21, 103, and 127.)
- [Antonelli y Leonessa, 2008] Antonelli, G. y Leonessa, A. (2008). Underwater robots: motion and force control of vehicle-manipulator systems. *Springer tracts in advanced robotics*, 2. (Citado en la página 127.)
- [Artac et al., 2002] Artac, M., Jogan, M., y Leonardis, A. (2002). Incremental pca for on-line visual learning and recognition. En *Pattern Recognition, 2002. Proceedings. 16th international conference on*, volumen 3, páginas 781–784. IEEE. (Citado en la página 71.)
- [Bale, 2012] Bale, K. (2012). osgocean. (Citado en la página 39.)
- [Barr, 1981] Barr, A. H. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23. (Citado en la página 72.)

- [Benhimane y Malis, 2004] Benhimane, S. y Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. En *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volumen 1, páginas 943–948. IEEE. (Citado en la página 74.)
- [Besl y McKay, 1992] Besl, P. J. y McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256. (Citado en la página 56.)
- [Bicchi y Kumar, 2000] Bicchi, A. y Kumar, V. (2000). Robotic grasping and contact: A review. En *ICRA*, volumen 348, página 353. Citeseer. (Citado en la página 103.)
- [Biegelbauer y Vincze, 2007] Biegelbauer, G. y Vincze, M. (2007). Efficient 3d object detection by fitting superquadrics to range image data for robot object manipulation. En *Robotics and Automation, 2007 IEEE International Conference on*, páginas 1086–1091. IEEE. (Citado en la páginas 20, 73, and 144.)
- [Bigg et al., 2003] Bigg, G., Jickells, T., Liss, P., y Osborn, T. (2003). The role of the oceans in climate. *International Journal of Climatology*, 23(10):1127–1159. (Citado en la página 1.)
- [Bingham et al., 2010] Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C., y Sakellariou, D. (2010). Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):702–717. (Citado en la páginas 3 and 12.)
- [Bohg et al., 2011] Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., Kragic, D., y Morales, A. (2011). Mind the gap - robotic grasping under incomplete observation. En *2011 IEEE International Conference on Robotics and Automation*, páginas 686–693. (Citado en la páginas 20, 60, 66, and 69.)
- [Bohg et al., 2014] Bohg, J., Morales, A., Asfour, T., y Kragic, D. (2014). Data-driven grasp synthesis - a survey. *IEEE Transactions on Robotics*, 30(2):289–309. (Citado en la páginas 18, 19, 20, 59, and 103.)
- [Bowen et al., 2013] Bowen, A. D., Jakuba, M. V., Farr, N. E., Ware, J., Taylor, C., Gomez-Ibanez, D., Machado, C. R., y Pontbriand, C. (2013). An un-tethered roV for routine access and intervention in the deep sea. En *2013 OCEANS - San Diego*, páginas 1–7. (Citado en la página 4.)
- [Bradski, 2000] Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal of Software Tools*. (Citado en la página 49.)
- [Brignone et al., 2015] Brignone, L., Raugel, E., Opderbecke, J., Rigaud, V., Piasco, R., y Ragot, S. (2015). First sea trials of hrov the new hybrid vehicle developed by ifremer. En *OCEANS 2015 - Genova*, páginas 1–7. (Citado en la página 4.)

- [Campos et al., 2016] Campos, R., Gracias, N., y Ridao, P. (2016). Underwater multi-vehicle trajectory alignment and mapping using acoustic and optical constraints. *Sensors*, 16(3). (Citado en la página 3.)
- [Carrera et al., 2015] Carrera, A., Palomeras, N., Hurtós, N., Kormushev, P., y Carreras, M. (2015). Cognitive system for autonomous underwater intervention. *Pattern Recognition Letters*, 67:91–99. (Citado en la páginas 7 and 19.)
- [Carreras et al., 2013] Carreras, M., Candela, C., Ribas, D., Mallios, A., Magí, L., Vidal, E., Palomeras, N., y Ridao, P. (2013). Sparus ii, design of a lightweight hovering auv. En *Martech 2013 5th International Workshop on Marine Technology*. SARTI. (Citado en la páginas 10 and 108.)
- [Casalino et al., 2012] Casalino, G., Zereik, E., Simetti, E., Torelli, S., Sperindé, A., y Turetta, A. (2012). Agility for underwater floating manipulation: Task & subsystem priority based control strategy. En *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, páginas 1772–1779. IEEE. (Citado en la página 127.)
- [Centelles et al., 2015] Centelles, D., Rubino, E., Soler, M., Martí, J., Sales, J., Marin, R., y Sanz, P. (2015). Underwater radio frequency based localization and image transmission system, including specific compression techniques, for autonomous manipulation. En *OCEANS 2015-Genova*, páginas 1–5. IEEE. (Citado en la página 7.)
- [Chavez et al., 2017] Chavez, A. G., Mueller, C. A., Birk, A., Babic, A., y Miskovic, N. (2017). Stereo-vision based diver pose estimation using lstm recurrent neural networks for auv navigation guidance. En *OCEANS 2017-Aberdeen*, páginas 1–7. IEEE. (Citado en la páginas 6, 46, and 107.)
- [Choi et al., 1994] Choi, S., Takashige, G., y Yuh, J. (1994). Experimental study on an underwater robotic vehicle: Odin. En *Autonomous Underwater Vehicle Technology, 1994. AUV'94., Proceedings of the 1994 Symposium on*, páginas 79–84. IEEE. (Citado en la página 3.)
- [Choi y Easterday, 2001] Choi, S. K. y Easterday, O. (2001). An underwater vehicle monitoring system and its sensors. En *Experimental Robotics VII*, páginas 551–560. Springer. (Citado en la página 107.)
- [Cieslak et al., 2015] Cieslak, P., Ridao, P., y Giergiel, M. (2015). Autonomous underwater panel operation by girona500 uvms: A practical approach to autonomous underwater manipulation. En *2015 IEEE International Conference on Robotics and Automation (ICRA)*, páginas 529–536. (Citado en la páginas 7 and 9.)
- [Company-Corcoles et al., 2018] Company-Corcoles, J. P., Ortiz, A., y Negre-Carrasco, P. L. (2018). *Target Detection and Tracking for Grasping Sea Floor Objects During Underwater Intervention Operations*, páginas 245–250. IOS Press. (Citado en la páginas 11, 20, and 75.)
- [Coumans, 2012] Coumans, E. (2012). Bullet physics engine. (Citado en la página 39.)

- [Denavit y Hartenberg, 1955] Denavit, J. y Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, 22:215–221. (Citado en la página 30.)
- [Diankov, 2010] Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute. (Citado en la páginas 104 and 105.)
- [Ding et al., 2001] Ding, D., Lee, Y.-H., y Wang, S. (2001). Computation of 3-d form-closure grasps. *IEEE Transactions on Robotics and Automation*, 17(4):515–522. (Citado en la página 105.)
- [Drap et al., 2008] Drap, P., Seinturier, J., Conte, G., Caiti, A., Scaradozzi, D., Zanolini, S. M., y Gambogi, P. (2008). Underwater cartography for archaeology in the venus project. *Geomatica*, 62(4):419–427. (Citado en la páginas 13 and 107.)
- [Duntley, 1963] Duntley, S. Q. (1963). Light in the sea. *JOSA*, 53(2):214–233. (Citado en la páginas 19 and 44.)
- [Escartín et al., 2008] Escartín, J., García, R., Delaunoy, O., Ferrer, J., Gracias, N., Elibol, A., Cufi, X., Neumann, L., Fornari, D. J., Humphris, S. E., y Renard, J. (2008). Globally aligned photomosaic of the lucky strike hydrothermal vent field (mid-atlantic ridge): Release of georeferenced data, mosaic construction, and viewing software. *Geochemistry, Geophysics, Geosystems*, 9(12):n/a–n/a. Q12009. (Citado en la página 3.)
- [Eustice et al., 2006] Eustice, R. M., Singh, H., y Leonard, J. J. (2006). Exactly sparse delayed-state filters for view-based slam. *IEEE Transactions on Robotics*, 22(6):1100–1114. (Citado en la página 3.)
- [Evans et al., 2001] Evans, J., Keller, K., Smith, J., Marty, P., y Rigaud, O. (2001). Docking techniques and evaluation trials of the swimmer auv: an autonomous deployment auv for work-class rovs. En *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volumen 1, páginas 520–528. IEEE. (Citado en la página 3.)
- [Evans et al., 2003] Evans, J., Redmond, P., Plakas, C., Hamilton, K., y Lane, D. (2003). Autonomous docking for intervention-auvs using sonar and video-based real-time 3d pose estimation. En *Oceans 2003. Proceedings*, volumen 4, páginas 2201–2210. IEEE. (Citado en la página 4.)
- [Farr et al., 2010] Farr, N., Bowen, A., Ware, J., Pontbriand, C., y Tivey, M. (2010). An integrated, underwater optical /acoustic communications system. En *OCEANS 2010 IEEE - Sydney*, páginas 1–6. (Citado en la páginas 4, 6, and 13.)
- [Fernández et al., 2015] Fernández, J. J., Pérez, J., Peñalver, A., Sales, J., Fornas, D., y Sanz, P. J. (2015). Benchmarking using uwsim, simurv and ros: An autonomous free floating dredging intervention case study. En *Oceans’15 MTS/IEEE Genova*, Génova (Italia). IEEE, IEEE. (Citado en la páginas 11, 128, and 130.)

- [Fernández et al., 2013] Fernández, J. J., Prats, M., Sanz, P. J., García, J. C., Marín, R., Robinson, M., Ribas, D., y Ridao, P. (2013). Grasping for the seabed: Developing a new underwater robot arm for shallow-water intervention. *IEEE Robotics Automation Magazine*, 20(4):121–130. (Citado en la páginas 8, 29, 33, and 130.)
- [Ferrer et al., 2007] Ferrer, J., Elibol, A., Delaunoy, O., Gracias, N., y Garcia, R. (2007). Large-area photo-mosaics using global alignment and navigation data. En *OCEANS 2007*, páginas 1–9. (Citado en la página 3.)
- [Fischinger et al., 2015] Fischinger, D., Weiss, A., y Vincze, M. (2015). Learning grasps with topographic features. *The International Journal of Robotics Research*, 3(234):12–35. (Citado en la páginas 20 and 21.)
- [Fischler y Bolles, 1987] Fischler, M. A. y Bolles, R. C. (1987). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. En *Readings in computer vision*, páginas 726–740. Elsevier. (Citado en la página 61.)
- [Flacco y De Luca, 2014] Flacco, F. y De Luca, A. (2014). A reverse priority approach to multi-task control of redundant robots. En *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, páginas 2421–2427. IEEE. (Citado en la página 127.)
- [Flacco y De Luca, 2015] Flacco, F. y De Luca, A. (2015). Unilateral constraints in the reverse priority redundancy resolution method. En *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, páginas 2564–2571. IEEE. (Citado en la página 127.)
- [Fletcher et al., 2008] Fletcher, B., Young, C., Buescher, J., Whitcomb, L. L., Bowen, A., McCabe, R., y Yoerger, D. R. (2008). Proof of concept demonstration of the hybrid remotely operated vehicle (hrov) light fiber tether system. En *OCEANS 2008*, páginas 1–6. (Citado en la página 4.)
- [Foley et al., 2009] Foley, B. P., Dellaporta, K., Sakellariou, D., Bingham, B. S., Camilli, R., Eustice, R. M., Evagelistis, D., Ferrini, V. L., Katsaros, K., Kourkoumelis, D., Mallios, A., Micha, P., Mindell, D. A., Roman, C., Singh, H., Switzer, D. S., y Theodoulou, T. (2009). The 2005 chios ancient shipwreck survey: New methods for underwater archaeology. *Hesperia: The Journal of the American School of Classical Studies at Athens*, 78(2):269–305. (Citado en la página 12.)
- [Forest et al., 2004] Forest, J., Salvi, J., Cabruja, E., y Pous, C. (2004). Laser stripe peak detector for 3d scanners. a fir filter approach. En *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volumen 3, páginas 646–649 Vol.3. (Citado en la página 50.)
- [Fornas et al., 2015] Fornas, D., Sales, J., Peñalver, A., Pérez, J., Fernández, J. J., Marín, R., y Sanz, P. J. (2015). Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects. *Journal of Experimental & Theoretical Artificial Intelligence*. (Citado en la página 94.)

- [Fornas et al., 2016] Fornas, D., Sanz, P. J., Porta, J. M., y Thomas, F. (2016). Improving local symmetry estimations in rgb-d images by fitting superquadrics. En *XXXVII Jornadas de Automática*, Madrid (España). (Citado en la páginas 20 and 72.)
- [Fornas et al., 2017] Fornas, D., Sanz, P. J., y Vincze, M. (2017). Increasing underwater manipulation autonomy using segmentation and visual tracking. En *OCEANS 2017 - Aberdeen*, Aberdeen (UK). IEEE, IEEE. (Citado en la página 11.)
- [FP7-PANDORA, 2015] FP7-PANDORA (2015). Persistent Autonomy through learnIng, aDaptation, Observation and Re-plAnning (PANDORA). <http://persistentautonomy.com/>. (Citado en la página 107.)
- [Gancet et al., 2015] Gancet, J., Urbina, D., Letier, P., Ilzkovitz, M., Weiss, P., Gauch, F., Chemisky, B., Antonelli, G., Casalino, G., Indiveri, G., Birk, A., Pflingsthorn, M. F., Calinon, S., Turetta, A., Walen, C., y Guilpain, L. (2015). Dexrov: Enabling effective dexterous rov operations in presence of communication latency. En *OCEANS 2015 - Genova*, páginas 1–6. (Citado en la página 6.)
- [Garcia et al., 2010a] Garcia, J., Fernandez, J., Sanz, P., y Marin, R. (2010a). Increasing autonomy within underwater intervention scenarios: The user interface approach. En *Systems Conference, 2010 4th Annual IEEE*, páginas 71–75. (Citado en la página 107.)
- [Garcia et al., 2010b] Garcia, J., Prats, M., Sanz, P., Marin, R., y Belmonte, O. (2010b). Exploring multimodal interfaces for underwater intervention systems. En *Proceedings of the IEEE ICRA 2010 Workshop on Multimodal Human-Robot Interfaces*. (Citado en la página 7.)
- [García et al., 2017] García, J. C., Patrao, B., Almeida, L., Pérez, J., Menezes, P., Dias, J., y Sanz, P. J. (2017). A natural interface for remote operation of underwater robots. *IEEE Computer Graphics and Applications*, 37(1):34–43. (Citado en la páginas 7 and 107.)
- [Garcia-Fidalgo et al., 2018] Garcia-Fidalgo, E., Ortiz, A., y Massot-Campos, M. (2018). *Vision-Based Control for an AUV in a Multi-robot Undersea Intervention Task*, páginas 36–48. Springer International Publishing, Cham. (Citado en la páginas 10 and 109.)
- [Geiger et al., 2010] Geiger, A., Roser, M., y Urtasun, R. (2010). Efficient large-scale stereo matching. En *Asian Conference on Computer Vision (ACCV)*. (Citado en la página 49.)
- [Gelfand et al., 2003] Gelfand, N., Ikemoto, L., Rusinkiewicz, S., y Levoy, M. (2003). Geometrically stable sampling for the icp algorithm. En *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, páginas 260–267. (Citado en la página 56.)
- [Grudic y Lawrence, 1993] Grudic, G. Z. y Lawrence, P. D. (1993). Iterative inverse kinematics with manipulator configuration control. *IEEE transactions on robotics and Automation*, 9(4):476–483. (Citado en la páginas 32 and 126.)

- [Guennebaud et al., 2010] Guennebaud, G., Jacob, B., et al. (2010). Eigen v3. <http://eigen.tuxfamily.org>. (Citado en la página 39.)
- [Hirschmuller, 2008] Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341. (Citado en la páginas 49 and 55.)
- [Holz et al., 2015] Holz, D., Ichim, A. E., Tombari, F., Rusu, R. B., y Behnke, S. (2015). Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robotics Automation Magazine*, 22(4):110–124. (Citado en la página 56.)
- [Houston Machatronics, 2018] Houston Machatronics (2018). Houston Mechatronics AQUANAUT Robot. <http://www.houstonmechatronics.com/aquanaut>. (Citado en la página 5.)
- [Huebner y Kragic, 2008] Huebner, K. y Kragic, D. (2008). Selection of robot pregrasps using box-based shape approximation. En *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, páginas 1765–1770. IEEE. (Citado en la páginas 21 and 63.)
- [Huebner et al., 2008] Huebner, K., Ruthotto, S., y Kragic, D. (2008). Minimum volume bounding box decomposition for shape approximation in robot grasping. En *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, páginas 1628–1633. IEEE. (Citado en la página 20.)
- [Ichim, 2013] Ichim, A.-E. (2013). Superquadrics. PCL Toyota Code Sprint 2.0. (Citado en la página 73.)
- [Joan et al., 2017] Joan, P., Ortiz, A., y Negre-Carrasco, P. L. (2017). Target detection and tracking for grasping sea floor objects uring underwater intervention operations. En *Recent Advances in Artificial Intelligence Research and Development: Proceedings of the 20th International Conference of the Catalan Association for Artificial Intelligence*, volumen 300, página 245. IOS Press. (Citado en la página 133.)
- [Johnson-Roberson et al., 2010] Johnson-Roberson, M., Pizarro, O., Williams, S. B., y Mahon, I. (2010). Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51. (Citado en la página 3.)
- [Kaiser, 2013] Kaiser, C. (2013). Hybrid auv inspection, monitoring, and intervention of seafloor and sub-seafloor pipelines. URL <http://www.whoi.edu/fileserver.do>. (Citado en la página 6.)
- [Kalwa et al., 2012] Kalwa, J., Pascoal, A., Ridao, P., Birk, A., Eichhorn, M., Brignone, L., Caccia, M., Alvez, J., y Santos, R. (2012). The european r&d-project morph: Marine robotic systems of self-organizing, logically linked physical nodes. *IFAC Proceedings Volumes*, 45(5):349–354. (Citado en la página 40.)

- [Katsoulas et al., 2008] Katsoulas, D., Bastidas, C. C., y Kosmopoulos, D. (2008). Superquadric segmentation in range images via fusion of region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):781–795. (Citado en la página 73.)
- [Kazhdan y Hoppe, 2013] Kazhdan, M. y Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29. (Citado en la página 69.)
- [Khatib et al., 2016] Khatib, O., Yeh, X., Brantner, G., Soe, B., Kim, B., Ganguly, S., Stuart, H., Wang, S., Cutkosky, M., Edsinger, A., Mullins, P., Barham, M., Woolstra, C. R., Salama, K. N., L’Hour, M., y Creuze, V. (2016). Ocean one: A robotic avatar for oceanic discovery. *IEEE Robotics Automation Magazine*, 23(4):20–29. (Citado en la páginas 5, 13, and 107.)
- [Kim et al., 2001] Kim, B.-H., Oh, S.-R., Yi, B.-J., y Suh, I. H. (2001). Optimal grasping based on non-dimensionalized performance indices. En *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volumen 2, páginas 949–956. IEEE. (Citado en la página 105.)
- [Lane et al., 1997] Lane, D. M., Davies, J. B. C., Casalino, G., Bartolini, G., Cannata, G., Veruggio, G., Canals, M., Smith, C., O’Brien, D. J., Pickett, M., et al. (1997). Amadeus: advanced manipulation for deep underwater sampling. *IEEE Robotics & Automation Magazine*, 4(4):34–45. (Citado en la página 3.)
- [Lane et al., 2015] Lane, D. M., Maurelli, F., Kormushev, P., Carreras, M., Fox, M., y Kyriakopoulos, K. (2015). Pandora-persistent autonomy through learning, adaptation, observation and replanning. *IFAC-PapersOnLine*, 48(2):238–243. (Citado en la página 7.)
- [Li et al., 2017] Li, X., Bilbao, S., Martín-Wanton, T., Bastos, J., y Rodriguez, J. (2017). Swarms ontology: a common information model for the cooperation of underwater robots. *Sensors*, 17(3):569. (Citado en la página 143.)
- [Li y Sastry, 1988] Li, Z. y Sastry, S. S. (1988). Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1):32–44. (Citado en la página 105.)
- [Marani et al., 2009] Marani, G., Choi, S. K., y Yuh, J. (2009). Underwater autonomous manipulation for intervention missions auvs. *Ocean Engineering*, 36(1):15–23. Autonomous Underwater Vehicles. (Citado en la página 107.)
- [Marchand et al., 2005] Marchand, É., Spindler, F., y Chaumette, F. (2005). Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics & Automation Magazine*, 12(4):40–52. (Citado en la página 39.)
- [Massot-Campos y Oliver-Codina, 2014] Massot-Campos, M. y Oliver-Codina, G. (2014). Underwater laser-based structured light system for one-shot 3d reconstruction. En *SENSORS, 2014 IEEE*, páginas 1138–1141. (Citado en la páginas 46 and 50.)

- [Massot-Campos y Oliver-Codina, 2015] Massot-Campos, M. y Oliver-Codina, G. (2015). Optical sensors and methods for underwater 3d reconstruction. *Sensors*, 15(12):31525–31557. (Citado en la páginas 19 and 46.)
- [Massot-Campos et al., 2016] Massot-Campos, M., Oliver-Codina, G., Bodenmann, A., y Thornton, B. (2016). Submap bathymetric slam using structured light in underwater environments. En *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, páginas 181–188. (Citado en la página 3.)
- [Miller et al., 2003] Miller, A. T., Knoop, S., Christensen, H. I., y Allen, P. K. (2003). Automatic grasp planning using shape primitives. En *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volumen 2, páginas 1824–1829. IEEE. (Citado en la página 20.)
- [Mindell y Bingham, 2001] Mindell, D. y Bingham, B. (2001). New archaeological uses of autonomous underwater vehicles. En *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No.01CH37295)*, volumen 1, páginas 555–558 vol.1. (Citado en la página 12.)
- [Nguyen y Le, 2013] Nguyen, A. y Le, B. (2013). 3d point cloud segmentation: A survey. En *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, páginas 225–230. (Citado en la página 20.)
- [Osfield et al., 2004] Osfield, R., Burns, D., et al. (2004). "open scene graph". (Citado en la página 39.)
- [Paduan et al., 2009] Paduan, J. B., Caress, D. W., Clague, D. A., Paull, C. K., Thomas, H., Chiocci, F., Ridente, D., Casalbore, D., y Bosman, A. (2009). High-resolution mapping of mass wasting, tectonic, and volcanic hazards using the mbari mapping auv. En *International Conference on seafloor mapping for geo-hazard assessmen*, volumen 7, páginas 181–186. (Citado en la página 3.)
- [Palomer et al., 2018] Palomer, A., Ridao, P., Youakim, D., Ribas, D., Forest, J., y Petillot, Y. (2018). 3d laser scanner for underwater manipulation. *Sensors*, 18(4). (Citado en la páginas 11, 46, 50, 77, 119, 133, and 137.)
- [Palomeras et al., 2016] Palomeras, N., Peñalver, A., Massot-Campos, M., Negre, P. L., Fernández, J. J., Ridao, P., Sanz, P. J., y Oliver-Codina, G. (2016). I-auv docking and panel intervention at sea. *Sensors*, 16(10):1673. (Citado en la página 34.)
- [Pascoal et al., 2015] Pascoal, R., Santos, V., Premebida, C., y Nunes, U. (2015). Simultaneous segmentation and superquadrics fitting in laser-range data. *IEEE Transactions on Vehicular Technology*, 64(2):441–452. (Citado en la página 144.)
- [Peñalver et al., 2018] Peñalver, A., , Fernández, J. J., Soriano, A., y Sanz, P. J. (2018). A multi-task priority framework for redundant robots with multiple kinematic chains under hard joint and cartesian constraints. En *Intelligent Robots and Systems (IROS 2018), 2018 IEEE/RSJ International Conference on*. IEEE. (Citado en la páginas 11, 35, 123, 127, 129, and 133.)

- [Peñalver et al., 2017] Peñalver, A., Fernández, J. J., y Sanz, P. J. (2017). Autonomous underwater grasping using multi-view laser reconstruction. En *OCEANS 2017 - Aberdeen*, Aberdeen (UK). IEEE, IEEE. (Citado en la páginas 50 and 95.)
- [Peñalver et al., 2015] Peñalver, A., Pérez, J., Fernández, J. J., Sales, J., Sanz, P. J., García, J., Fornas, D., y Marín, R. (2015). Visually-guided manipulation techniques for robotic autonomous underwater panel interventions. *Annual Reviews in Control*, 40:201–211. (Citado en la páginas 7, 10, 13, 56, 61, and 107.)
- [Perez et al., 2017] Perez, J., Attanasio, A. C., Nechyporenko, N., y Sanz, P. J. (2017). A deep learning approach for underwater image enhancement. En *International Work-Conference on the Interplay Between Natural and Artificial Computation*, páginas 183–192. Springer. (Citado en la página 7.)
- [Pérez et al., 2017] Pérez, J., Fornas, D., y Marín, R. y P. J. S. (2017). Uwsim, un simulador submarino conectado a la nube como herramienta educacional. *Revista Iberoamericana de Automática e Informática industrial*, 15(1):70–78. (Citado en la páginas 7 and 41.)
- [Pérez et al., 2015] Pérez, J., Sales, J., Peñalver, A., Fornas, D., Fernández, J. J., García, J. C., Sanz, P. J., Marín, R., y Prats, M. (2015). Exploring 3-d reconstruction techniques: A benchmarking tool for underwater robotics. *IEEE Robotics & Automation Magazine*, 22. (Citado en la páginas 7, 41, 51, 54, and 61.)
- [Pizarro et al., 2009] Pizarro, O., Eustice, R. M., y Singh, H. (2009). Large area 3-d reconstructions from underwater optical surveys. *IEEE Journal of Oceanic Engineering*, 34(2):150–169. (Citado en la página 3.)
- [Prankl et al., 2015] Prankl, J., Aldoma, A., Svejda, A., y Vincze, M. (2015). Rgb-d object modelling for object recognition and tracking. En *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, páginas 96–103. (Citado en la página 20.)
- [Prats et al., 2012a] Prats, M., Fernández, J. J., y Sanz, P. J. (2012a). Combining template tracking and laser peak detection for 3d reconstruction and grasping in underwater environments. En *International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal. (Citado en la páginas 46, 50, 51, 74, and 95.)
- [Prats et al., 2012b] Prats, M., García, J. C., Wirth, S., Ribas, D., Sanz, P. J., Ridaio, P., Gracias, N., y Oliver, G. (2012b). Multipurpose autonomous underwater intervention: A systems integration perspective. En *2012 20th Mediterranean Conference on Control Automation (MED)*, páginas 1379–1384. (Citado en la páginas 9, 21, and 29.)
- [Prats et al., 2012c] Prats, M., Pérez, J., Fernández, J., y Sanz, P. (2012c). An open source tool for simulation and supervision of underwater intervention missions. En *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, páginas 2577–2582, Vilamoura, Algarve, Portugal. (Citado en la páginas 7, 22, 39, 107, and 109.)

- [Prats et al., 2012d] Prats, M., Ribas, D., Palomerias, N., García, J. C., Nannen, V., Wirth, S., Fernández, J. J., Beltrán, J. P., Campos, R., Ridaio, P., et al. (2012d). Reconfigurable auv for intervention missions: a case study on underwater object recovery. *Intelligent Service Robotics*, 5(1):19–31. (Citado en la páginas 7, 8, 29, 34, 61, and 107.)
- [Prats et al., 2010] Prats, M., Sanz, P. J., y Pobil, A. P. (2010). A framework for compliant physical interaction. *Autonomous Robots*, 28(1):89–111. (Citado en la páginas 21 and 124.)
- [QT, 2018] QT (2018). QT cross-platform software framework. <https://www.qt.io/>. (Citado en la páginas 39 and 109.)
- [Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., y Ng, A. Y. (2009). Ros: an open-source robot operating system. En *ICRA workshop on open source software*, volumen 3, página 5. Kobe, Japan. (Citado en la páginas 38 and 110.)
- [Ribas et al., 2012] Ribas, D., Palomerias, N., Ridaio, P., Carreras, M., y Mallios, A. (2012). Girona 500 auv: From survey to intervention. *IEEE/ASME Transactions on Mechatronics*, 17(1):46–53. (Citado en la páginas 8, 35, and 130.)
- [Richtsfeld et al., 2012] Richtsfeld, A., Mörwald, T., Prankl, J., Zillich, M., y Vincze, M. (2012). Segmentation of unknown objects in indoor environments. En *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, páginas 4791–4796. IEEE. (Citado en la página 145.)
- [Ridaio et al., 2010] Ridaio, P., Carreras, M., Ribas, D., y Garcia, R. (2010). Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):759–778. (Citado en la página 3.)
- [Ridaio et al., 2015] Ridaio, P., Carreras, M., Ribas, D., Sanz, P. J., y Oliver, G. (2015). Intervention auvs: The next challenge. *Annual Reviews in Control*, 40:227 – 241. (Citado en la página 2.)
- [Rigaud et al., 1998] Rigaud, V., Coste-Maniere, E., Aldon, M.-J., Probert, P., Perrier, M., Rives, P., Simon, D., Lang, D., Kiener, J., Casal, A., et al. (1998). Union: underwater intelligent operation and navigation. *IEEE Robotics & Automation Magazine*, 5(1):25–35. (Citado en la página 3.)
- [Rizzini et al., 2017] Rizzini, D. L., Kallasi, F., Aleotti, J., Oleari, F., y Caselli, S. (2017). Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks. *Computers & Electrical Engineering*, 58:560 – 571. (Citado en la página 46.)
- [Roman y Mather, 2010] Roman, C. y Mather, R. (2010). Autonomous underwater vehicles as tools for deep-submergence archaeology. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 224(4):327–340. (Citado en la página 12.)
- [Romero-Ramirez et al., 2018] Romero-Ramirez, F. J., Muñoz-Salinas, R., y Medina-Carnicer, R. (2018). Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38 – 47. (Citado en la páginas 56 and 79.)

- [Roth et al., 2003] Roth, M., Vail, D., y Veloso, M. (2003). A real-time world model for multi-robot teams with high-latency communication. En *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volumen 3, páginas 2494–2499. (Citado en la página 143.)
- [Rubert et al., 2018] Rubert, C., León, B., Morales, A., y Sancho-Bru, J. (2018). Characterisation of grasp quality metrics. *Journal of Intelligent & Robotic Systems*, 89(3):319–342. (Citado en la páginas 21, 103, 104, and 105.)
- [Rusinkiewicz y Levoy, 2001] Rusinkiewicz, S. y Levoy, M. (2001). Efficient variants of the icp algorithm. En *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, páginas 145–152. (Citado en la página 53.)
- [Rusu, 2009] Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany. (Citado en la página 65.)
- [Rusu et al., 2009a] Rusu, R. B., Blodow, N., y Beetz, M. (2009a). Fast point feature histograms (fpfh) for 3d registration. En *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, páginas 3212–3217. Citeseer. (Citado en la página 56.)
- [Rusu y Cousins, 2011] Rusu, R. B. y Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). En *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China. (Citado en la páginas 39, 51, and 62.)
- [Rusu et al., 2009b] Rusu, R. B., Holzbach, A., Diankov, R., Bradski, G., y Beetz, M. (2009b). Perception for mobile manipulation and grasping using active stereo. En *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, páginas 632–638. IEEE. (Citado en la página 19.)
- [Rusu et al., 2008] Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., y Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927 – 941. Semantic Knowledge in Robotics. (Citado en la página 52.)
- [Sakellariou et al., 2007] Sakellariou, D., Georgiou, P., Mallios, A., Kapsimalis, V., Kourkoumelis, D., Micha, P., Theodoulou, T., y Dellaporta, K. (2007). Searching for ancient shipwrecks in the aegean sea: the discovery of chios and kythnos hellenistic wrecks with the use of marine geological-geophysical methods. *International Journal of Nautical Archaeology*, 36(2):365–381. (Citado en la página 12.)
- [Sanz et al., 2015] Sanz, P., Peñalver, A., Sales, J., Fernández, J. J., Pérez, J., Fomas, D., y García, J. (2015). Multipurpose underwater manipulation for archaeological intervention. *Instrumentation viewpoint*, (18):50–51. (Citado en la página 10.)
- [Sanz et al., 2017a] Sanz, P. J., Marín, R., Peñalver, A., Fornas, D., y Centelles, D. (2017a). Merbots project: overall description, multisensory autonomous perception and grasping for underwater robotics interventions. *Actas de las XXXVIII Jornadas de Automática*. (Citado en la páginas 10, 61, and 108.)

- [Sanz et al., 1998] Sanz, P. J., Pobil, A. P. D., Inesta, J. M., y Recatala, G. (1998). Vision-guided grasping of unknown objects for service robots. En *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volumen 4, páginas 3018–3025 vol.4. (Citado en la página 18.)
- [Sanz et al., 2005] Sanz, P. J., Requena, A., Inesta, J. M., y Pobil, A. P. D. (2005). Grasping the not-so-obvious: vision-based object handling for industrial applications. *IEEE Robotics Automation Magazine*, 12(3):44–52. (Citado en la páginas 7 and 18.)
- [Sanz et al., 2012] Sanz, P. J., Ridao, P., Oliver, G., Casalino, G., Insaurralde, C., Silvestre, C., Melchiorri, C., y Turetta, A. (2012). Trident: Recent improvements about autonomous underwater intervention missions. *IFAC Proceedings Volumes*, 45(5):355 – 360. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles. (Citado en la páginas 7, 9, 61, and 107.)
- [Sanz et al., 2017b] Sanz, P. J., Vincze, M., y Fornas, D. (2017b). Increasing underwater manipulation autonomy using segmentation and visual tracking. En *OCEANS 2017 - Aberdeen*, páginas 1–5. (Citado en la página 62.)
- [Scharstein y Szeliski, 2002] Scharstein, D. y Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42. (Citado en la página 49.)
- [Schiebener et al., 2016] Schiebener, D., Schmidt, A., Vahrenkamp, N., y Asfour, T. (2016). Heuristic 3d object shape completion based on symmetry and scene context. En *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, páginas 74–81. IEEE. (Citado en la páginas 20, 66, and 69.)
- [Schmidt y Rzhanov, 2012] Schmidt, V. E. y Rzhanov, Y. (2012). Measurement of micro-bathymetry with a gopro underwater stereo camera pair. En *2012 Oceans*, páginas 1–6. (Citado en la página 46.)
- [Schwarz et al., 2015] Schwarz, M., Schulz, H., y Behnke, S. (2015). Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. En *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, páginas 1329–1335. IEEE. (Citado en la página 19.)
- [Sheridan, 1992] Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. MIT press. (Citado en la página 106.)
- [Shimoga, 1996] Shimoga, K. B. (1996). Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, 15(3):230–266. (Citado en la página 18.)
- [Siciliano, 1990] Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *Journal of intelligent and robotic systems*, 3(3):201–212. (Citado en la páginas 32 and 127.)
- [Simetti y Casalino, 2017] Simetti, E. y Casalino, G. (2017). Manipulation and transportation with cooperative underwater vehicle manipulator systems. *IEEE Journal of Oceanic Engineering*, 42(4):782–799. (Citado en la página 6.)

- [Simetti et al., 2017] Simetti, E., Wanderlingh, F., Casalino, G., Indiveri, G., y Antonelli, G. (2017). Robust project: Control framework for deep sea mining exploration. En *OCEANS 2017 Anchorage*, páginas 1–5. (Citado en la página 6.)
- [Simetti et al., 2018] Simetti, E., Wanderlingh, F., Torelli, S., Bibuli, M., Odetti, A., Bruzzone, G., Rizzini, D. L., Aleotti, J., Palli, G., Moriello, L., et al. (2018). Autonomous underwater intervention: Experimental results of the maris project. *IEEE Journal of Oceanic Engineering*, 43(3):620–639. (Citado en la página 6.)
- [Smits et al., 2011] Smits, R., Bruyninckx, H., y Aertbeliën, E. (2011). Kdl: Kinematics and dynamics library. Available: <http://www.orocos.org/kdl>. (Citado en la página 39.)
- [Taubin y Cooper, 1991] Taubin, G. y Cooper, D. B. (1991). *Object recognition based on moment (or algebraic) invariants*. IBM TJ Watson Research Center. (Citado en la páginas 20 and 70.)
- [Tomasi y Manduchi, 1998] Tomasi, C. y Manduchi, R. (1998). Bilateral filtering for gray and color images. En *Computer Vision, 1998. Sixth International Conference on*, páginas 839–846. IEEE. (Citado en la página 52.)
- [TRITON, 2014] TRITON (2014). Multisensory Based Underwater Intervention through Cooperative Marine Robots. <http://www.irs.uji.es/triton/>. (Citado en la página 10.)
- [Trower y Gray, 2015] Trower, J. y Gray, J. (2015). Creating new languages in blockly: Two case studies in media computation and robotics. En *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, páginas 677–677. ACM. (Citado en la página 41.)
- [Ulbrich et al., 2011] Ulbrich, S., Kappler, D., Asfour, T., Vahrenkamp, N., Bierbaum, A., Przybylski, M., y Dillmann, R. (2011). The.opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation. En *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, páginas 1761–1767. (Citado en la página 105.)
- [Urabe et al., 2015] Urabe, T., Ura, T., Tsujimoto, T., y Hotta, H. (2015). Next-generation technology for ocean resources exploration (zipangu-in-the-ocean) project in japan. En *OCEANS 2015-Genova*, páginas 1–5. IEEE. (Citado en la página 1.)
- [Wang et al., 1995] Wang, H. H., Rock, S. M., y Lees, M. (1995). Experiments in automatic retrieval of underwater objects with an auv. En *OCEANS'95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings.*, volumen 1, páginas 366–373. IEEE. (Citado en la página 3.)
- [Weichert et al., 2013] Weichert, F., Bachmann, D., Rudak, B., y Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393. (Citado en la página 108.)
- [Williams et al., 2014] Williams, K., Robertis, A. D., Berkowitz, Z., Rooper, C., y Towler, R. (2014). An underwater stereo-camera trap. *Methods in Oceanography*, 11:1 – 12. (Citado en la página 46.)

- [Yoo et al., 2015] Yoo, S. Y., Jun, B. H., Shim, H., y Lee, P. M. (2015). Design and analysis of carbon fiber reinforced plastic body frame for multi-legged subsea walking robot, crabster. *Ocean Engineering*, 102:78 – 86. (Citado en la página 6.)
- [Youakim et al., 2017] Youakim, D., Ridao, P., Palomeras, N., Spadafora, F., Ribas, D., y Muzzupappa, M. (2017). Moveit!: Autonomous underwater free-floating manipulation. *IEEE Robotics Automation Magazine*, 24(3):41–51. (Citado en la páginas 7, 21, and 144.)
- [Yuh et al., 1998] Yuh, J., Choi, S., Ikehara, C., Kim, G., McMurty, G., Ghasemi-Nejhad, M., Sarkar, N., y Sugihara, K. (1998). Design of a semi-autonomous underwater vehicle for intervention missions (sauvim). En *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, páginas 63–68. IEEE. (Citado en la página 4.)
- [Zennaro et al., 2015] Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., y Menegatti, E. (2015). Performance evaluation of the 1st and 2nd generation kinect for multimedia applications. En *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, páginas 1–6. IEEE. (Citado en la páginas 19, 43, 46, and 73.)
- [Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22. (Citado en la página 49.)