



UNIVERSIDAD NACIONAL DEL SUR

TESIS DE DOCTOR EN CONTROL DE SISTEMAS

**Desarrollo de Algoritmos para la  
Exploración Submarina mediante  
Vehículos Autónomos con Visión  
Artificial Monocular**

Emanuel TRABES

Bahía Blanca

Argentina

Año 2018



## Prefacio

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Control de Sistemas, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ingeniería Eléctrica y de Computadoras, durante el período comprendido entre el 3 de Octubre de 2013 y el 12 de diciembre de 2017 , bajo la dirección del Doctor Mario Alberto Jordán, Profesor de Fundamentos de Control Realimentado y de Postgrado en “Control de Sistemas” del Departamento de Ingeniería Eléctrica de la Universidad Nacional del Sur.

Emanuel Trabes



**UNIVERSIDAD NACIONAL DEL SUR**  
Secretaria General de Posgrado y Educación Continua  
La presente tesis ha sido aprobada el ...../...../....., mereciendo la  
calificación de ..... (.....)



## *Agradecimientos*

Quiero agradecer a mi director Mario Jordán por guiarme pacientemente durante todos estos años. A mis compañeros del Grupo de Control Adaptivo en Ingeniería Oceánica, Jorge Bustamante, Carlos Berger y José Bonitatibus por sus recomendaciones y amistad. A Alejandra Villegas, por su incondicional apoyo y estoica compañía. Por último a mi Madre, mi Padre y mis Hermanos.



## *Resumen*

La motivación de esta tesis consistió en investigar el empleo de sistemas visuales monoculares en ambientes subacuáticos e implementar soluciones novedosas a los problemas complejos que en tales ambientes se presentan. Algunos de los principales aportes de este trabajo son: el desarrollo de algoritmos para la navegación en ambientes altamente auto-símiles, la implementación de un sistema monocular SLAM (Simultaneous localization and Mapping) denso orientado a su uso en ambientes subacuáticos, el desarrollo de una metodología robusta para la exploración retorno y revisita basada en SLAM, la implementación de un filtro de onda cáustica para ambientes tridimensionales. La validez de los resultados obtenidos teóricamente se verificó experimentalmente en ambientes subacuáticos controlados.



## *Abstract*

The motivation of this thesis consisted in researching the use of visual-monocular systems in underwater environments and to implement novel solutions to the complex problems that arise in such scenes. Some of the main contributions of this work are: the development of algorithms for navigation in highly self-similar scenes, the implementation of a dense monocular SLAM system (Simultaneous localization and Mapping) developed to be used in underwater scenes, the development of a robust methodology for exploration, return and revisiting based on SLAM, the implementation of a caustic wave filter for tridimensional environments. The validity of the obtained results were verified experimentally in controlled underwater environments.



# Índice general

<b>Prefacio</b>	<b>III</b>
<b>Resumen</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Hipótesis de Trabajo	2
1.2. Análisis del Estado del Arte	3
1.2.1. Metodología	6
1.2.2. Principales Aportes	6
<b>2. Exploración Subacuática Autónoma</b>	<b>9</b>
2.1. Introducción	9
2.2. Sistema Propuesto	11
2.2.1. Heurística	12
2.2.2. Fase de Exploración	13
2.2.3. Fase de Consolidación	15
2.2.4. Implementación del Algoritmo	15
2.3. Casos de Estudio	16
2.3.1. Caso de Estudio I: Ambiente Simulado y Sistema SLAM Indirecto No-denso	16
2.3.2. Caso de Estudio II: Ambiente Natural y Sistema SLAM Directo Denso	16
<b>3. Odometría Visual</b>	<b>21</b>
3.1. Introducción	21
3.1.1. Métodos Indirectos-No-Densos	21
3.1.2. Métodos Directos-Densos	22
3.2. Nuevo Método de Odometría Visual	23
3.2.1. Estimación de la Postura	23
3.2.2. Cálculo y Acumulación del Costo	23
3.2.3. Estimación de Profundidad y Varianza	26
3.2.4. Reposicionamiento del Volumen de Costo	26
3.2.5. Remoción de Oclusiones	28
3.2.6. Muestreo de la Línea Epipolar	29
3.2.7. Regularización	32
3.2.8. Adaptación Dinámica de Rango	32
3.2.9. Integración en un Sistema SLAM Completo	32
3.3. Evaluación del Método Propuesto	33
<b>4. Filtro de Onda Cáustica</b>	<b>41</b>
4.1. Introducción	41
4.2. Algoritmo Propuesto	42
4.2.1. Adaptación del Umbral de Binarización	45

Optimización . . . . .	46
4.3. Evaluación . . . . .	46
<b>5. Control Adaptivo con Visión Artificial</b>	<b>49</b>
5.1. Introducción . . . . .	49
5.1.1. Sistema de Control Adaptivo . . . . .	51
5.2. Ecuación de Movimiento de Vehículos Subacuáticos . . . . .	51
5.2.1. Control Adaptivo de Gradiente de Velocidad . . . . .	53
Funcional de costo para seguimiento de trayectorias . . . . .	53
5.2.2. Leyes Adaptivas . . . . .	54
5.2.3. Leyes Adaptivas Modificadas . . . . .	55
5.2.4. Simplificación de los Modos a Controlar . . . . .	56
5.2.5. Identificación de Parámetros . . . . .	56
Ambigüedad en Escala . . . . .	57
5.3. Evaluación Comparativa del Sistema de Control . . . . .	57
5.4. Simulación . . . . .	58
<b>6. Experimentación</b>	<b>61</b>
6.1. Vehículo Subacuático Utilizado . . . . .	61
6.1.1. Hardware . . . . .	61
Computadoras a bordo . . . . .	62
Computadora Externa . . . . .	62
6.1.2. Software . . . . .	62
6.1.3. Configuración de Cámaras . . . . .	63
Cámara Principal . . . . .	63
Cámara Secundaria . . . . .	65
6.2. Escena Submarina . . . . .	66
6.3. Navegación en Suelo Autosimil . . . . .	67
6.3.1. Trayectorias . . . . .	68
6.3.2. Evaluación del Éxito Durante las Revisitas . . . . .	68
6.3.3. Velocidad de Navegación . . . . .	70
6.3.4. Mapa Denso . . . . .	70
6.4. Filtrado de Onda Cáustica . . . . .	71
6.4.1. Sistema Indirecto ORB-SLAM . . . . .	73
6.4.2. Sistemas Directos . . . . .	74
<b>7. Conclusiones</b>	<b>79</b>
<b>A. Implementación en Paralelo</b>	<b>83</b>
A.1. Introducción . . . . .	83
A.2. Sistema Propuesto . . . . .	84
A.2.1. Análisis Previo . . . . .	84
Funcionamiento General . . . . .	84
Definiciones . . . . .	84
Módulo Localizador . . . . .	85
Módulo Mapeador . . . . .	85
Módulo de Acumulación del Volumen de Costo . . . . .	85
Módulo de Optimización de la Función de Costo . . . . .	86
Módulo Acumulación Incremental de los Keyframes . . . . .	87
Conclusiones del Análisis . . . . .	87
A.2.2. Implementación FPGA . . . . .	87

Proyección al Plano Focal . . . . .	88
Estructura de Memoria . . . . .	89
Representación Numérica . . . . .	90
Conexión a PC . . . . .	90
A.2.3. Resultados de la Implementación . . . . .	90
A.3. Pruebas . . . . .	91
A.3.1. Dataset Utilizado . . . . .	91
A.4. Conclusiones . . . . .	91
<b>B. Escala Métrica</b> . . . . .	<b>93</b>
B.1. Introducción . . . . .	93
B.2. Sistema Propuesto . . . . .	94
B.3. Validación Experimental . . . . .	95
<b>C. Segmentación Por Color</b> . . . . .	<b>99</b>
C.1. Introducción . . . . .	99
C.2. Sistema Propuesto . . . . .	100
C.2.1. Algoritmo General . . . . .	100
C.2.2. Algoritmo Adaptativo . . . . .	100
C.3. Evaluación Experimental . . . . .	101
<b>Bibliografía</b> . . . . .	<b>105</b>



# Índice de figuras

1.1. Ejemplos de imágenes subacuáticas . . . . .	4
2.1. Heurística para generar el corredor . . . . .	14
2.2. Escena simulada, vista de la cámara . . . . .	16
2.3. Escena simulada, trayectoria . . . . .	17
2.4. Resultados de la simulación . . . . .	17
2.5. Escenario subacuático . . . . .	18
2.6. Construcción del corredor . . . . .	19
2.7. Visualización del corredor . . . . .	19
3.1. Diagrama en bloques . . . . .	24
3.2. Volumen de costo cúbico . . . . .	27
3.3. Volumen de costo frustum . . . . .	28
3.4. Error reproyección volumen de costo frustum . . . . .	29
3.5. Oclusiones . . . . .	30
3.6. Línea epipolar con poco paralaje . . . . .	31
3.7. Línea epipolar con mucho paralaje . . . . .	31
3.8. Primer secuencia, comparación . . . . .	34
3.9. Primer secuencia, trayectoria . . . . .	34
3.10. Segunda secuencia, comparación . . . . .	35
3.11. Tercer secuencia, comparación . . . . .	37
3.12. Tercer secuencia, trayectoria . . . . .	38
3.13. Cuarta secuencia, comparación . . . . .	38
3.14. Cuarta secuencia, trayectoria . . . . .	38
3.15. Quinta secuencia, comparación . . . . .	39
3.16. Quinta secuencia, trayectoria . . . . .	39
4.1. Ondas cáusticas . . . . .	41
4.2. Algoritmo de filtrado básico basado en una estructura realimentada. . . . .	43
4.3. Predicción de posiciones de los píxeles y su brillo . . . . .	44
4.4. Escena de evaluación . . . . .	47
5.1. Propuesta de control adaptivo . . . . .	49
5.2. Resultados controlador PID . . . . .	59
5.3. Resultados controlador modo deslizante . . . . .	59
5.4. Resultados controlador adaptivo . . . . .	60
6.1. OpenROV . . . . .	62
6.2. Distorsión del domo . . . . .	64
6.3. Distorsión en la cámara principal . . . . .	64
6.4. Efecto de rolling shutter . . . . .	65
6.5. OpenROV cámara secundaria . . . . .	66
6.6. Distorsión en la cámara secundaria . . . . .	67
6.7. Escena submarina de experimentación . . . . .	67

6.8. Trayectorias x-y estimadas . . . . .	69
6.9. Imágenes en la posición inicial . . . . .	70
6.10. Imágenes en la posición final . . . . .	71
6.11. Imágenes en la posición media . . . . .	71
6.12. Velocidad del AUV . . . . .	72
6.13. Distintas visualizaciones del mapa denso estimado para la realización 2. . . . .	72
6.14. ORB-SLAM bajo perturbaciones cáusticas . . . . .	73
6.15. Sistemas directos bajo perturbación cáustica . . . . .	74
6.16. LSD-SLAM con filtro . . . . .	75
6.17. LSD-SLAM mapa . . . . .	75
6.18. Sistema directo propuesto con filtro . . . . .	76
6.19. Mapa estimado mediante el sistema directo propuesto . . . . .	77
A.1. Diagrama en bloques DTAM . . . . .	84
A.2. Diagrama en bloques del módulo Mapeador . . . . .	85
A.3. Volumen de costo . . . . .	86
A.4. Implementación FPGA . . . . .	88
A.5. Arquitectura del módulo de re-proyección . . . . .	89
A.6. Recorrido de la simulación . . . . .	92
A.7. Resultados . . . . .	92
B.1. Validación experimental . . . . .	95
B.2. Imagen del sistema cámara-láser utilizado para la validación experi- mental . . . . .	96
B.3. Resultados de la experimentación . . . . .	96
B.4. Ilustración de un instante en la experimentación . . . . .	97
B.5. Mapa estimado . . . . .	97
C.1. Ducto . . . . .	99
C.2. Diagrama en bloques del sistema propuesto . . . . .	100
C.3. Proceso de optimización . . . . .	101
C.4. Algoritmo adaptativo . . . . .	102
C.5. Experimento . . . . .	102
C.7. Resultados experimentales . . . . .	103
C.6. Segmentación por apariencia . . . . .	103

# Lista de Acrónimos

<b>SLAM</b>	<b>S</b> imultaneous <b>L</b> ocalization <b>A</b> nd <b>M</b> apping
<b>AUV</b>	<b>A</b> utonomous <b>U</b> nderwater <b>V</b> ehicle
<b>ROV</b>	<b>R</b> emotely <b>O</b> perated <b>V</b> ehicle
<b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning <b>S</b> ystem
<b>MSIS</b>	<b>M</b> echanically <b>S</b> canned <b>I</b> maging <b>S</b> onar
<b>TSDF</b>	<b>T</b> runcated <b>S</b> igned <b>D</b> istance <b>F</b> unction
<b>FPGA</b>	<b>F</b> ield <b>P</b> rogrammable <b>G</b> ate <b>A</b> rray
<b>FOV</b>	<b>F</b> ield <b>O</b> f <b>V</b> iew
<b>C-SLAM</b>	<b>SLAM</b> basado en <b>C</b> orredor



## Capítulo 1

# Introducción

Los océanos presentan amplias extensiones de terreno mayormente inexplorados, de los cuales se supone que contienen grandes reservas de recursos útiles para la industria. Investigar la forma de acceder a estos recursos es de suma importancia, principalmente para países con grandes plataformas marítimas. La falta de exploración y explotación de estos recursos se debe a varios factores, principalmente porque los ambientes subacuáticos son sumamente hostiles para la vida humana. La profundidad puede llegar a los once Kilómetros, y las enormes presiones hacen imposible su exploración directa, aun utilizando submarinos tripulados. Por estas razones, para avanzar con su exploración y la consecuente obtención de los recursos valiosos, es necesaria la utilización de métodos indirectos, que suponen también cruzar barreras tecnológicas aún no alcanzadas en la robótica subacuática.

Un método de exploración sumamente interesante, aunque visionario todavía, supone el empleo de vehículos subacuáticos autónomos (AUV). Se podrían utilizar estos vehículos para realizar misiones como la exploración de grandes extensiones de terreno, realizar un mapa marcando los recursos de interés encontrados, extraerlos con herramientas a bordo del vehículo, para luego concluir llevando los recursos a la superficie, todo de manera autónoma y continua en el tiempo sin que corra peligro ningún operador.

La propiedad más importante de los sistemas autónomos es la toma de decisiones basada en interpretaciones sobre situaciones seleccionadas de la escena circundante. Este comportamiento es independiente de cualquier orden o sugerencia humana, más allá de una supervisión y monitoreo ocasional o periódico y de una eventual intervención humana necesaria.

Desde una condición inicial, el sistema autónomo construye una base de conocimiento y la expande según comprende e interpreta el ambiente de acuerdo a una estrategia planificada. Desde su puesta en marcha, el sistema autónomo evoluciona en su comportamiento, aumentando en forma gradual su habilidad y precisión en tareas repetitivas y su capacidad de abordar desafíos del ambiente real acorde a su tarea específica.

Otras propiedades destacables de un sistema autónomo son la "conciencia" de su posición en todo instante, el reconocimiento de zonas ya exploradas y la incorporación de éstas en un mapa.

Los sistemas autónomos robóticos subacuáticos se sirven en general de la Inteligencia Artificial para la interpretación de conocimiento difuso y además de la Cibernética y la Automatización mediante Sistemas de Control para realizar sus misiones.

En particular la percepción del mundo subacuático real con visión artificial representa un camino promisorio de la robótica submarina actual, sobre todo a baja altitud en la navegación. La visión artificial abarca una gran cantidad de métodos mediante los cuales pueden analizarse, identificarse e interpretarse circunstancias

del mundo submarino. Los métodos operan con el procesamiento de imágenes y videos por medio de sistemas de cómputo con capacidad para generar en tiempo real, información numérica o simbólica con diversos grados de semántica.

Teniendo en consideración la marcada inaccesibilidad del océano, por su profundidad, su baja visibilidad, por la propiedad auto-símil del suelo y en ocasiones debido a una significativa perturbación lumínica del fondo causada por ondas cáusticas de los rayos del sol atravesando la superficie, es que tanto el estudio y desarrollo de sistemas autónomos subacuáticos para la navegación autónoma como así también las aplicaciones en este ambiente, han quedado muy atrás en comparación con el estado del arte en la robótica móvil terrestre o aérea.

Ni siquiera hay evidencia sólida de la validez de las técnicas consolidadas en aplicaciones subacuáticas, es decir, no existen suficientes estudios que demuestren que el estado actual de la navegación autónoma pueda extenderse sin más ni más al ambiente submarino. Por ello, sería desacertado aseverar que la eficiencia, prestación y robustez de técnicas usuales de la robótica móvil y aérea como así también la tecnología necesaria podría ser similar en el agua.

En áreas afines como las aplicaciones submarinas desde un barco o plataforma semi-sumergible, la tecnología estándar para investigar el fondo marino la constituye la acústica provista por sonares y sondas ultrasónicas, las cuales operan con frecuencias bajas de muestreo, insuficientes para la navegación autónoma en tiempo real. Otras aplicaciones con plataformas sumergibles no-tripuladas pero teleoperadas por cable como los ROV (Remotely Operated Vehicles), emplean cámaras de visión para operar en distancias cercanas al objeto con la finalidad de transmitir una imagen nítida al operador en el barco para su interpretación. La visión en esta aplicación no tiene una proyección que encuadre en un navegador autónomo basado en visión que pueda describir por sí mismo el ambiente en tiempo real.

El estudio realizado en esta tesis doctoral pretende contribuir en el área de la robótica subacuática, enfocando la navegación autónoma basada en visión artificial como único medio de percepción, contemplando el ambiente del fondo oceánico con todas las dificultades asociadas a escenas submarinas complejas que dificultan notablemente la visibilidad y el discernimiento de patrones en el fondo provocados por la auto-similitud de su textura.

## 1.1. Hipótesis de Trabajo

Se postula como hipótesis de trabajo a la viabilidad que ofrecen los sistemas de visión artificial para la navegación autónoma en el medio subacuático por zonas totalmente desconocidas. La hipótesis supone tanto soluciones estándar del estado del arte de ambientes terrestres/aéreos, como así también modificadas y novedosas, recurriendo a técnicas de la Inteligencia Artificial y de los Sistemas de Control Adaptivo. El medio submarino se entiende con todas sus particularidades importantes, estacionarias y cambiantes en el tiempo, que afectan la navegación, en particular la baja visibilidad y reconocimiento incierto de rasgos del suelo. La validez de la hipótesis de trabajo sobre la base de soluciones propuestas deberá confirmarse con una justificación teórica apropiada y metodología experimental, tanto en plataformas de pruebas (benchmark) y resultados experimentales que marcarán las conclusiones definitivas.

## 1.2. Análisis del Estado del Arte

En escenarios terrestres el problema de la auto-localización puede ser fácilmente solucionado mediante la utilización de sistemas de posicionamiento global (GPS), tecnología que es ampliamente utilizada en la literatura [1]. Sin embargo, en ambientes submarinos no es sencillo utilizar estos sistemas directamente, porque las ondas electromagnéticas sufren una gran atenuación al ingresar en la columna de agua, imposibilitando la comunicación con el vehículo. Por lo que se requiere investigar otras metodologías de auto-localización.

Recientemente la investigación de los sistemas de auto-localización y mapeo (SLAM) ha logrado obtener resultados excepcionales. Estos sistemas realizan un mapa del terreno atravesado y, simultáneamente, estiman la posición vehículo relativa a este mapa.

Reseñas de estas técnicas, orientadas a su utilización en vehículos submarinos autónomos, se pueden encontrar en [2], [3], [4] [5].

La elección del sensor utilizado a bordo es sumamente importante y estos determinarán la calidad de la auto-localización y, naturalmente, la fidelidad del mapa reconstruido. En la literatura existen varias alternativas, como sonares, cámaras, sistemas inerciales, etc. Históricamente el sensor utilizado por excelencia en sistemas autónomos subacuáticos es el sonar. Las ondas sonoras penetran fácilmente en la columna de agua y es posible así estimar el relieve del fondo a grandes distancias. Sin embargo, este tipo de sensor resulta costoso, lento y de baja resolución, y por su naturaleza activa consume considerablemente más energía que un sensor pasivo. Como ejemplo de estos tipos de sistemas, existen los trabajos de [6], [7], que utilizan un sistema denominado Mechanically Scanned Imaging Sonar (MSIS) para realizar SLAM en ambiente subacuáticos. En [8] también se emplea un sensor MSIS, pero con un algoritmo orientado a ambientes estructurados, como puertos y represas, utilizando la detección de líneas (en relieves y contornos) basada en la transformada de Hough [9]. En [10], [11], [12]; se emplean sonares de barrido lateral como sensor principal.

Las cámaras tienen las excelentes características de bajo costo, una gran densidad de información, altas tasas de refresco y un bajo consumo de energía. Existe en la literatura excelentes resultados en SLAM en ambientes terrestres utilizando cámaras como único sensor a bordo [13] [14] [15].

Los ambientes subacuáticos tienen características visuales particulares que los hacen considerablemente más complejos que los ambientes terrestres. La dispersión de la luz (fenómeno también conocido como scattering) en el agua causa una iluminación ambiental difusa, que genera imágenes con bajo contraste. A su vez, la luz sufre una atenuación selectiva en frecuencia, dependiente de la profundidad, elevando el color rojo en muy bajas profundidades y virando al verde y azul en profundidades bajas e intermedias hasta las últimas longitudes de onda del espectro visible para profundidades altas de miles de metros. Este fenómeno causa que las imágenes capturadas presenten una crominancia dominada por los tonos verde-azulados. Las ondas cáusticas, causadas por la refracción que sufre la luz al atravesar la superficie del agua, generan un cambio dinámico y errático de brillo en el fondo marino, particularmente en escenas de baja profundidad. Al ser un ambiente no estructurado, es decir con objetos y relieve que tienen poca o ninguna connotación semántica, se pueden encontrar grandes extensiones de terreno que presentan pocas características descriptibles y gran auto-similitud, es decir con una alta probabilidad de repetición. Ejemplo de estas zonas son bancos de arena o grandes extensiones de bentos. Estas características dificultan la auto-localización basada en visión, ya que

se repiten en formas parecidas en múltiples lugares, obstaculizando la asociación a una única coordenada. Así, las imágenes capturadas en ambientes subacuáticos son mayormente monocromáticas, presentan bajo contraste, poseen grandes perturbaciones en su brillo, son bajas en características prominentes y muy auto-símiles. La figura 1.1 ejemplifica dos de estos fenómenos con apariencia autosímil. También se puede apreciar que el fondo marino carece en general de elementos geométricos reconocibles semánticamente, y su baja estructura es muy amorfa.

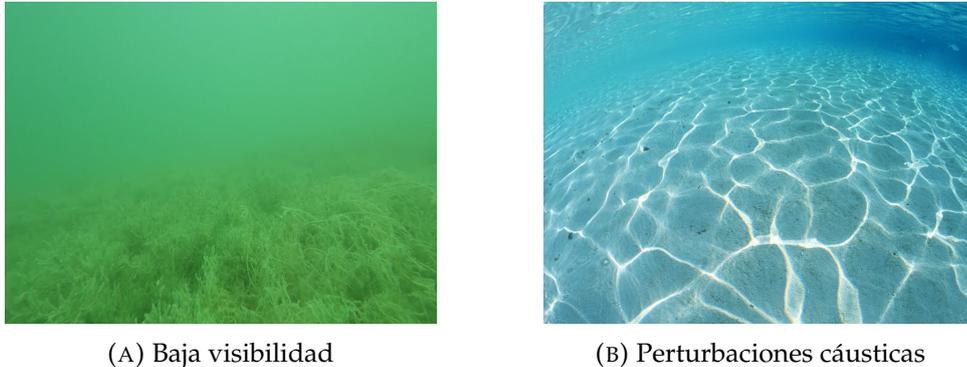


FIGURA 1.1: Ejemplos de imágenes subacuáticas

Estas características encausaron los esfuerzos en preprocesar las imágenes de manera de filtrar las perturbaciones existentes, como el trabajo [16], que implementa una corrección de color a imágenes subacuáticas usando técnicas de Aprendizaje de Máquinas (Machine Learning). Asimismo en [17] se desarrolla un modelo de refracción de la luz, para mejorar la estimación densa de la profundidad en sistemas binoculares. No es claro si estas técnicas simplemente mejoran la apreciación humana de las imágenes subacuáticas, o si también son útiles para la navegación autónoma.

A su vez también se destinaron esfuerzos en robustecer los mecanismos de reconocimiento del recorrido, conocido técnicamente como cierres de lazo, el cual se explicará más adelante en el capítulo 2. En [18], los autores desarrollan un sistema de cerrado de lazo robusto, implementando racimos (clusters) de puntos relevantes (keypoints) e imágenes de referencia (keyframes). En [19], se desarrolla un método novedoso basado en técnicas LSH (Locality Sensitive Hashing) para acelerar la detección de cerrados de lazo, probándose el sistema en ambientes subacuáticos. Los autores en [20] desarrollan técnicas de cerrado de lazo diseñadas particularmente para ambientes submarinos. En [21] se propone un sistema de cerrado de lazo para que funcione en circunstancias de cambio de apariencia, probándose en ambientes marinos en inspección de cascos de barcos. Los autores en [22] presentan un algoritmo de navegación subacuática autónoma, basada en características sobresalientes, para la inspección de cascos de barcos. Estos trabajos particulares no atacan la problemática de suelos extremadamente autosímiles, como los bancos de arena, donde los cerrados de lazo son proclives a fallar. En contextos terrestres, existen los trabajos como [23] [24] [25], donde se realiza una navegación sobre una misma trayectoria, facilitando considerablemente la tarea de cerrado de lazo. Pero estos trabajos no formulan una manera de aprender el primer recorrido de navegación, ni si la selección de la representación del mapa es la adecuada, de manera de permitir la navegación en sentido inverso. También queda en duda si estos sistemas funcionarían en contextos subacuáticos. Todos los aportes mencionados, ayudan a entender el problema-solución particular vinculado a la visión subacuática, pero no está claro

si pueden proyectarse directamente o con algún grado de utilidad en la navegación autónoma.

En cuanto a la estimación de movimiento mediante visión, en años recientes varios sistemas fueron desarrollados para ambientes subacuáticos, como los sistemas binoculares de [26], [27] [28] y monoculares [29], [30], [31], [32], [33], [34] [35], [36]. Todos estos trabajos tienen en común que no utilizan toda la información de la imagen para estimar el movimiento del vehículo y el mapa del ambiente. Por el contrario, utilizan el concepto de keypoint [37] [38], para detectar en la imagen las características sobresalientes robustas. A primera vista esta estrategia tiene dos desventajas principales. En primer lugar, al estimar la profundidad solamente en algunos píxeles seleccionados, se obtienen mapas no densos, o bien ralo. Si bien estos mapas son útiles para la localización del vehículo, tienen una utilidad limitada en cuanto a las tareas que requieran un mapa detallado del ambiente. Por ejemplo una situación muy común en navegación subacuática es tener que disminuir la altitud para ganar visibilidad sobre el fondo, esto conlleva a aumentar la posibilidad de colisiones, cuya detección a tiempo sería dificultosa con este tipo de mapa ralo. Por otro lado, los ambientes subacuáticos son en general altamente borrosos y auto-símiles, por lo no siempre se puede asegurar que existan características sobresalientes robustas.

Para solucionar estos problemas, existen en la literatura los métodos llamados "densos", que estiman profundidad en todos los píxeles observados por la imagen, y obtienen así un mapa completo de la superficie marina circundante. La bibliografía en cuanto a la utilización de estas técnicas en ambientes subacuáticos es mucho más limitada, donde se destacan los trabajos de [39], donde se desarrolla un sistema SLAM visual denso usando cámaras binoculares, para comparar el rendimiento de dos técnicas de SLAM comúnmente utilizadas. Los autores no atacan los problemas mencionados del entorno, sino que muestran resultados en ambientes donde hay buena visibilidad, características sobresalientes y poca perturbación lumínica. En [40], se implementa un filtro de turbidez (haze) para eliminar la zona poco texturada captada por la cámara en un ambiente marino y así utilizar un sistema denso para estimar la profundidad. El trabajo presenta resultados de estimación de la profundidad, pero no presenta resultados con respecto a la localización. En [17], se implementa un pre-procesamiento de la imagen, para ser utilizada en la estimación de la profundidad densa en un sistema binocular. Se muestran resultados de estimación de profundidad a partir de dos imágenes tomadas simultáneamente, pero no se deja entrever como se comportaría el sistema en un experimento largo con imágenes en movimiento.

Nuevamente, todos estos resultados son afines pero particularizados a un problema visual del entorno y de la estimación de profundidad. Sin embargo no alcanzan a responder claramente a la hipótesis de trabajo sobre la aptitud de ellos, o de parte de ellos, para la navegación submarina autónoma.

El auge de los Sistemas de Control Adaptivo aplicado a sistemas complejos como los que se dan en la robótica en entornos subacuáticos es posterior a las contrapartes de robótica terrestre y aérea. Por ello se considera un área científica abierta y de sumo interés. Los vehículos subacuáticos poseen no-linealidades inherentes en sus ecuaciones de movimiento, deben desempeñarse en ambientes altamente perturbados y las misiones que deben llevar a cabo son altamente complejas. Así es como se reconoce que estos vehículos requieren ser guiados mediante controladores complejos, que involucran usualmente control de velocidad automática, posicionamiento y seguimiento dinámico, y sistemas de autopiloto para control de profundidad y altitud. A su vez, es posible encontrar en la literatura verificaciones experimentales que muestran cómo la utilización de técnicas adaptivas pueden proveer una mejora

en el rendimiento de tareas tales como el seguimiento trayectoria, comparándolos con controladores de modelo fijo [41], [42]. En los últimos 15 años en la bibliografía se han discutidos diferentes enfoques para el diseño de controladores adaptivos y controladores adaptivos-robustos para vehículos subacuáticos, de manera de tener buen rendimiento bajo incertidumbres en la dinámica, hidrodinámica y perturbaciones externas, véase por ejemplo [43], [44], [45], [46], [47]. Sin embargo, el empleo metodologías de diseño de controladores novedosos de alto rendimiento no-lineales como backstepping [48], enfoques basados en pasividad [49] o modo deslizante [44] no aparecen en la literatura excepto como aplicaciones incipientes, véase por ejemplo [50], [51], [52], [53], [54]. Partiendo de resultados teóricos previos y algunas corroboraciones experimentales, parece que técnicas adaptivas novedosas pueden dar lugar a un mejoramiento en el rendimiento global en seguimiento de trayectoria, sobre todo cuando es necesaria una maniobrabilidad precisa con alta velocidad en el movimiento en un ambiente subacuático cambiante e incierto [55], [56], [57].

En resumen, del estudio del estado del arte se desprende que no existen todavía resultados donde se presente un sistema visual monocular para ambientes submarinos que permita la exploración autónoma y el retorno a la posición inicial de manera robusta, bajo perturbaciones como autosimilitud, ondas cáusticas y alta borrosidad.

### 1.2.1. Metodología

La metodología empleada para el desarrollo de esta tesis consistió, en primer lugar, de un estudio exhaustivo del estado del arte con análisis y prueba comparativa y sistemática de métodos consolidados. Primeramente se persiguió el estudio del océano y sus características visuales. Luego se investigaron las distintas metodologías SLAM existentes en la literatura, en vista de su inherente capacidad de superar las dificultades que el ambiente marino presenta. Este estudio dio como resultado la propuesta de un sistema diferenciado en dos etapas: la exploración de un camino gestado de forma autónoma, y el regreso seguro del vehículo, conjuntamente con la consolidación y refinamiento del mapa obtenido. Se prosiguió con el estudio del vehículo a utilizar, que dio como resultado tanto un modelo dinámico del mismo y un controlador basado en técnicas adaptivas. Los distintos desarrollos luego fueron evaluadas exhaustivamente tanto en simulación como en “datasets” reales, para culminar con el diseño y la creación de experimentos necesarios para verificar la hipótesis de trabajo.

### 1.2.2. Principales Aportes

La motivación de esta tesis es explorar la utilización de sistemas visuales monoculares para la auto-localización y mapeo densos en ambientes subacuáticos, y en ese contexto, intentar contribuir con una solución original en la problemática descrita de navegación autónoma en un medio acuático.

Cada capítulo de esta tesis desarrolla una problemática particular, mencionando detalladamente la solución propuesta, la justificación y las publicaciones propias donde se presentaron resultados relevantes. A su vez, cada sección destaca los principales aportes realizados en cada temática.

A modo de resumen, se destacan aquí algunos de los principales aportes de esta tesis: el desarrollo de algoritmos para la navegación en ambientes altamente auto-símiles, la implementación de un sistema monocular SLAM denso orientado para la exploración en ambientes subacuáticos, la implementación de un filtro de onda cáustica para ambientes tridimensionales y los primeros resultados de la literatura

---

sobre la implementación de un sistema monocular denso en FPGA, entre otros. Resultados parciales de estas investigaciones fueron publicados en [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73].



## Capítulo 2

# Exploración Subacuática Autónoma

### 2.1. Introducción

Gran parte de la navegación autónoma sin posicionamiento global realizada por un vehículo no-tripulado está basada en técnicas de SLAM (Auto-Localización y Mapeo), de cuya filosofía se desprenden innumerables métodos. Esta filosofía consiste en construir incrementalmente un mapa consistente de un entorno desconocido en el que el vehículo se encuentra, a la vez que se vale de dicho mapa para estimar su trayectoria al desplazarse dentro de este entorno. Adicionalmente el mapa puede ser consolidado y extendido con estrategias planificadas, o más complejamente, sujetas a decisiones propias del sistema autónomo que guía al vehículo.

La auto-localización y el mapeo pueden considerarse como dos operaciones interactivas, en donde una se apoya en la otra y viceversa. Queda claro que el ruido presente en los sistemas de navegación (sensores) junto a los inevitables errores cometidos en la representación del entorno y su creciente complejidad a lo largo del tiempo, provocan imprecisiones crecientes en el mapa y la localización. Prácticamente, la auto-localización produce un alejamiento continuo y errático del vehículo respecto a su verdadera posición, y este alejamiento se acrecienta en el tiempo a menos que el vehículo vuelva a un punto de su antigua trayectoria y pueda asociar la escena con la escena vista anteriormente. Sólo en estos casos la precisión global tanto de mapas como de auto-posición puede acotarse y mejorarse.

Es posible dividir conceptualmente a los sistemas SLAM en dos subsistemas. Primeramente está el llamado "front-end", se encarga principalmente de las tareas de bajo nivel, como la estimación del movimiento local de la cámara (llamado odometría visual) y la estimación local del mapa (como se abordará en el capítulo 3). El segundo subsistema es el denominado "back-end", el cual en cambio utiliza la información suministrada por el front-end para generar un mapa global coherente de todo el terreno explorado, como así también estimar la posición global del vehículo.

Esta división de tareas está basada en la observación de que la estimación odométrica de la posición presenta ineludiblemente un error. Dicho error tiene la particularidad de ser integrado a lo largo de la trayectoria del vehículo.

Así, luego de un cierto tiempo, el vehículo está en peligro de perderse puesto que cada vez se aleja de la supuesta posición, lo que en un caso real de vehículo submarino puede representar la pérdida definitiva del mismo en un ambiente extenso, desconocido e inaccesible.

Una de las principales tareas del back-end es disminuir en lo posible la deriva odométrica. Si ocurre que el vehículo atraviesa nuevamente una zona previamente explorada, se pueden obtener dos fuentes de información para estimar la localización del vehículo. Por un lado la información proveniente de la medición odométrica, y por otro lado el conocimiento de una revisita en una misma zona física. Estas dos fuentes de información pueden combinarse para generar una estimación global

mejorada, con menor error de localización. A este proceso se lo denomina cierre de lazo [33], [18], y es la técnica principal utilizada en el estado del arte para disminuir el error odométrico. La tarea más dificultosa es la detección de un cierre de lazo, es decir, detectar cuándo efectivamente el vehículo transcurre por una zona previamente explorada. La gran cantidad de información contenida en las imágenes [33], [18], hace que los sensores visuales sean uno de los más utilizados para la detección de los cierre de lazos. De ahora en más se tratará la navegación autónoma por visión y el SLAM visual.

En este capítulo se argumenta que, dadas las características visuales particulares de los ambientes subacuáticos, las técnicas comúnmente utilizadas en ambientes terrestres para disminuir la deriva odométrica no funcionarían correctamente. Para justificar esta aseveración, se ahondará en el funcionamiento particular de este tipo de algoritmos.

Estas técnicas están basadas comúnmente en algoritmos denominados bag of words [74], originarios del procesamiento del lenguaje natural. Un resumen de su funcionamiento es el siguiente. En primer lugar, se genera a-priori un “vocabulario”, compuesto por las características sobresalientes más comunes para el ambiente particular (llamadas “palabra”). Luego se computa para cada imagen proveniente de la cámara, un histograma de las características sobresalientes (palabra) que ésta contenga. Al mismo tiempo, se confecciona una estructura de árbol con los histogramas de las imágenes procesadas, teniendo como estructura de ordenamiento para dicho árbol el “vocabulario”. A medida que el tiempo de navegación transcurre, el sistema autónomo debe almacenar una enorme cantidad de información visual a lo largo de su trayectoria. Por ello, el empleo de este método permite buscar rápidamente en el árbol un determinado grupo de palabras, y de esta manera se puede determinar si una zona particular fue previamente observada o no.

Como se puede entrever, estas técnicas requieren que los terrenos contengan dos propiedades fundamentales. En primer lugar, deben tener suficientes zonas con características sobresalientes, comúnmente llamadas “keypoints”. En segundo lugar, necesitan que las zonas sean diferentes entre sí, de manera que al compararlas, haya pocas probabilidades de cometer un error en la asociación de ésta con una posición geométrica. Nótese que un error en el cierre de lazo podría ser catastrófico para el vehículo en sí mismo, ya que una distorsión significativa del mapa podría provocar la pérdida irrecuperable de éste.

Es común encontrar vastas regiones del suelo subacuático caracterizadas por una baja densidad de keypoints, escenas autosímiles y difuminadas como así también perturbadas lumínicamente con cambios rápidos de intensidad en el tiempo y en el espacio sobre el fondo. Dado que esto es una particularidad de los ambientes submarinos en general, la elección de técnicas de SLAM en una estrategia de navegación general deberá adaptarse a estas características ambientales. Asimismo la denominada estrategia de navegación se refiere a la autonomía o independencia para definir el guiado del vehículo.

Teniendo en cuenta que un sistema SLAM adaptado a un submarino autónomo es propenso a la falla a causa de errores inducidos por el medio subacuático durante la exploración por terreno desconocido, es difícil pensar que el mismo no quede expuesto a la pérdida a corto plazo. Tratándose de equipos submarinos de altísimo valor extrínseco y sin nexo físico con el barco de operaciones (por ejemplo a través de un cable de seguridad), es que la posibilidad de su eventual pérdida debe ser prácticamente descartada dentro del diseño.

Una forma de resolver estos problemas es utilizar las técnicas llamadas “teach and repeat”. En éstas, se enseña al vehículo un camino determinado, para que luego

éste lo pueda seguir con un grado de autonomía. Como el camino es único, para realizar el cierre de lazo el vehículo sólo tiene que poder diferenciar la zona en la que se encuentra de las inmediatamente circundantes, simplificando el proceso de seguimiento considerablemente. Este tipo de cierre de lazo no logra disminuir los errores globales de localización. Pero sí es altamente robusto ante la influencia de errores locales. Así, es posible navegar con éxito por el camino, aún cuando esté presente poca descriptibilidad por medio de características sobresalientes y/o auto-similitud del terreno.

En [75] los autores desarrollan un algoritmo basado en “teach and repeat” para lograr que un vehículo terrestre dotado con un sistema de cámaras binoculares navegue por un terreno dificultoso. El mapa se implementa mediante keypoints, y los autores muestran navegaciones exitosas de kilómetros bajo un suelo auto-símil. A su vez los autores en [76] implementan técnicas de “teach and repeat” para lograr la navegación autónoma de un MAV (Micro Air Vehicle) pequeño, utilizando una cámara monocular y un sensor de altura. El trabajo [77] muestra resultados de estas técnicas utilizando un cuadricóptero, desarrollando un algoritmo que requiere bajos recursos computacionales. Los autores en [78] robustecen técnicas de “teach and repeat” para que funcionen correctamente aún bajo importantes cambios en la iluminación de la escena. En [79] se utiliza un sistema consistente de varias cámaras binoculares para sobrellevar grandes cambios en la apariencia de la escena, tales como aquellos producidos por cambios de estaciones. En [23] se utiliza las subsiguientes visitas del camino original para mejorarlo incrementalmente. Estas mejoras tienen en cuenta las zonas que presenten grandes diferencias de apariencia durante las visitas, que serán marcadas como inseguras, y no se utilizarán para realizar la localización del vehículo sobre el camino. En [24] muestran resultados de “teach and repeat” sobre extensiones largas de terrenos, sobre zonas sometidas a grandes cambios de iluminación.

En todos los trabajos mencionados anteriormente, no se presenta una técnica para generar la primera navegación de aprendizaje. Tampoco es claro si la utilización de mapas de características sobresalientes (keypoints) es la opción más acertada para este tipo de aplicación, ya que desde diferentes perspectivas serían visibles diferentes keypoints, y así para la correcta navegación debería utilizarse el mismo ángulo de visión, lo cual no siempre es posible cumplir. Por ejemplo, navegar en reversa por el camino aprendido sería dificultoso.

En este capítulo se desarrollan algoritmos para la navegación autónoma en ambientes subacuáticos altamente autosímiles. Las principales innovaciones son, entre otras, el desarrollo de un corredor óptimo en la exploración que sea robusto para la visita, la utilización de mapas densos, sobre los que se argumenta que mejoran sustancialmente la resolución en estos tipos de problemas, en conjunto con un sistema autónomo para generar una primera navegación de aprendizaje. Los resultados parciales que dieron lugar a este trabajo fueron presentados en [58] [63].

## 2.2. Sistema Propuesto

En vista de las particularidades del problema real que representa la navegación subacuática con visión, de las soluciones aportadas en el estado del arte y de sus limitaciones es que se emprende un camino diferente para abordar la navegación subacuática autónoma.

La tesis se centra en un objetivo particular e importante para las aplicaciones, el cual consiste en la exploración autónoma y el retorno seguro de un vehículo submarino a su punto de partida. Más específicamente se construye un corredor sobre el suelo oceánico basado en rasgos del mismo para facilitar la navegación visual en ambos sentidos.

La navegación segura por un corredor afirma la condición necesaria de robustez de la metodología SLAM, la cual no es una propiedad destacada en las aplicaciones del estado del arte, sobre todo en la exploración por zonas desconocidas sin posicionamiento global. A su vez la condición de obtener un sistema de visión completamente autónomo en sus decisiones para guiar a un vehículo por un corredor que se construye apropiadamente y se consolida en sus visitas, es un tema abierto, apenas tratado, en la comunidad científica en esta área. Primeramente se presenta una heurística para el desarrollo de una metodología para desarrollar la solución. A esta metodología se la denominó C-SLAM debido a su conexión con una estrategia que emplea SLAM para la formación de un corredor.

El sistema visual debe posibilitar una guía autónoma tanto para escudriñar el terreno y encontrar las direcciones más apropiadas en la construcción del corredor, como así también para su consolidación paulatina con eventuales visitas del mismo.

### 2.2.1. Heurística

Para el desarrollo de la heurística se concibe un vehículo subacuático con una cámara monocular ubicada en la parte frontal del submarino, con una leve inclinación hacia adelante permitiendo en general visualizar la línea del horizonte en la parte superior de la imagen y el suelo en las zonas restantes. De esta manera el sistema autónomo podrá recabar información visual para un desplazamiento hacia adelante seguro del submarino con la posibilidad de tomar medidas de cambio de rumbo rápidamente y acelerar o frenar ante posibles colisiones. La altitud se auto-regula de acuerdo a un sistema de control independiente de los movimientos de avance o giro. Para los modos fundamentales de movimiento, a saber traslación tangencial a la trayectoria y rotación de rumbo y sus velocidades respectivas, se empleará un control adaptativo de gradiente de velocidad que se encargará de cumplir las decisiones de movimiento impuestas por el sistema de guía de SLAM visual.

Para la elaboración del sistema de navegación, se dividirá la tarea en dos fases. En primer lugar se encuentra la fase de exploración, en la que se debe navegar sin poseer un mapa previo. Aquí se utiliza la odometría visual para estimar el movimiento y la posición de la cámara, como se describirá más detalladamente en el capítulo 3. Como la escena es altamente auto-símil, se corre el peligro de no lograr acotar los errores odométricos. Esto es así a causa de la dificultad para detectar los cierres de lazos, debido a una confusión en la asociación de la postura estimada de la cámara con la coordenada real de los rasgos captados del suelo. Por esta razón, se utilizará como estrategia la exploración en zonas que presenten la mayor descriptividad posible. De esta manera el algoritmo ubica la zona en la imagen que presentan mayor cantidad de keypoints, donde más fácilmente se puede realizar la localización en una visita (mediante el cierre de lazo) y finalmente orienta el vehículo en dicha dirección. La etapa de exploración termina cuando se llega al objetivo, como puede ser un mapa identificado con anterioridad o si se encuentra una zona de interés. Durante esta etapa se realiza un mapa de la zona explorada, cuya composición global representa la geometría del entorno captado del fondo marino a lo largo del corredor. El mismo constituye una estructura de puntos de coordenadas tridimensionales que

representan una superficie del entorno estimada (mapa de profundidades de cámara). Claramente esta información geométrica está disponible en las escenas captadas por la cámara y no contiene información de las partes ocluidas en las líneas de visión. A su vez, la fase de exploración construye paso a paso un corredor a través de una sucesión de segmentos direccionados (nodos), conformando una trayectoria de puntos de vistas de cámara asociados a una alta densidad de keypoints.

De esta forma, la idea clave detrás de la presente heurística es que el vehículo subacuático pueda construir una cadena de nodos entre dos regiones distantes e inconexas, mediante la exploración del fondo marino. Aún cuando el terreno sea altamente borroso y autosímil, donde el cierre de lazo es difícil, se puede utilizar esta cadena de nodos para repetir la ruta aprendida, de manera robusta y sin cometer errores.

Se describe a continuación la generación de trayectoria concerniente al par de nodos  $n$  and  $n + 1$  (véase la figura 2.1).

Se hará distinción entre dos clases de trayectos particulares que se pueden encontrar durante la navegación. Se consideran aquellos tramos que conectan un nodo localizado en un punto de referencia  $\mathbf{p}_n$  con algún punto a la distancia  $\mathbf{o}_{n+1}$ , mediante un suelo carente de características sobresalientes. Por otra parte se distinguen los tramos que conectan  $\mathbf{o}_{n+1}$  con el próximo nodo  $\mathbf{p}_{n+1}$  sobre un suelo rico en características. El punto  $\mathbf{o}_{n+1}$  es el primer punto de vista en el camino entre nodos desde el cual es posible visualizar gran cantidad de características en el lecho marino. El nodo de referencia  $\mathbf{p}_{n+1}$  representa el último punto de vista en el tramo, desde el cual se visualizan gran cantidad de características sobresalientes. Vale la pena mencionar aquí que un punto de vista es una postura de la cámara en seis grados de libertad.

En ausencia de características confiables, el primer tramo involucra una navegación guiada mediante odómetro, siguiendo una trayectoria rectilínea dada por  $\vec{\mathbf{p}}_n$ . A su vez, el segundo tramo utiliza una gran cantidad de características sobresalientes observables para determinar una serie de puntos de vistas desde  $\mathbf{v}_{n+1}$  hasta  $\mathbf{p}_{n+1}$ , puntos de vista que son proclives a ser detectados en fase de revisita (mediante la detección del cierre de lazo). Nótese que la postura del vehículo obtenida mediante odometría  $\mathbf{o}_{n+1}$  es solo aparente, mientras que la postura asociada  $\mathbf{v}_{n+1}$  es la postura real.

El tramo entre los extremos  $\mathbf{o}_{n+1}$  y  $\mathbf{p}_{n+1}$  está compuesto por varios puntos de vista, desde los cuales es posible observar abundante cantidad de características sobresalientes en el suelo. Esta abundancia permitirá al sistema SLAM detectar correctamente un cierre de lazo durante la etapa de consolidación, pudiendo así eliminar el error odométrico incurrido durante la navegación en la zona carente de keypoints (donde no se puede cerrar el lazo), error representado mediante la diferencia entre  $\mathbf{o}_{n+1}$  y la pose real del vehículo  $\mathbf{v}_{n+1}$ .

### 2.2.2. Fase de Exploración

De acuerdo con la heurística previamente descripta, la selección de nodos durante la exploración es la tarea más delicada e importante a realizarse en dicha fase. Los parámetros de un nodo son: su postura  $\mathbf{p}_n$  y la dirección de partida desde la misma  $\vec{\mathbf{p}}_n$ .

En primer lugar, los keypoints deben ser detectados para luego, con la ayuda de un criterio de robustez acorde, seleccionados o descartados. Mediante la utilización de un método eficiente de detectores/descriptores como SURF (Speeded-Up Robust Features), se puede detectar un conjunto de características en el terreno marino tal que éstas se puedan observar bajo diferentes puntos de vista.

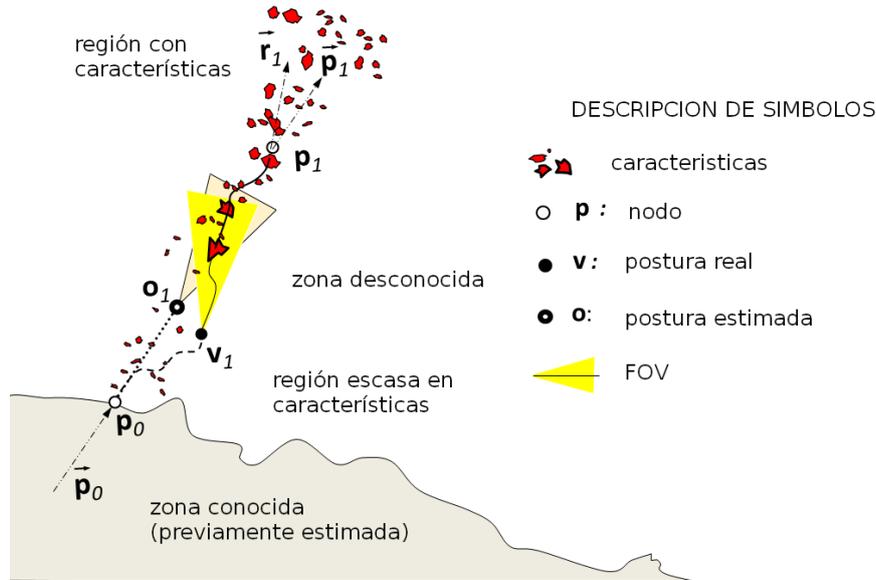


FIGURA 2.1: Heurística para generar el corredor empleando un sistema de guía visual basado en características sobresalientes, compuesto de dos etapas. Descripción del ejemplo. Etapa 1: El sistema comienza desde la pose  $p_0$ . En presencia de un suelo carente de características sobresalientes, el sistema selecciona una dirección  $\vec{p}_0$ , navegando de manera rectilínea. Luego de un determinado tiempo, en  $v_1$ , se logra obtener una perspectiva desde donde se puede observar suficientes características sobresalientes.  $o_1$  es la postura obtenida mediante odometría. Etapa 2: El sistema navega desde  $v_1$  y define una trayectoria que apunta hacia la mayor cantidad de características sobresalientes. Luego de un tiempo, se llega a una región con pocas características y se fija un nodo de referencia  $p_1$ , con orientación  $\vec{r}_1$ . El sistema define una dirección de preferencia  $\vec{p}_1$  para continuar la navegación. Luego se comienza nuevamente en la etapa 1.

De esta manera se define la postura de un nodo de referencia  $p_n$  como la postura desde donde el vehículo observó características sobresalientes robustas durante la fase de exploración en el instante  $n$ .

Otra variable importante del nodo es la orientación de partida  $\vec{p}_n$ , descrita por un vector unidad apuntando a una dirección conveniente en el horizonte. Para garantizar la construcción de un corredor sólido, la orientación debe ser definida tal que maximice la probabilidad de éxito del cierre de lazo durante la fase de consolidación.

Como se convino anteriormente, se asume una cámara frontal montada en el vehículo con una posición tal que le permita visualizar tanto el fondo marino como el horizonte. Así, la mitad superior de la imagen observa objetivos distantes.

Por consiguiente, un algoritmo efectivo para estimar la orientación de partida de un nodo consiste en analizar la sección superior en la imagen capturada por la cámara, para luego seleccionar las zonas con la mayor densidad de keypoints. Específicamente, el algoritmo clasificará los keypoints en racimos (clusters) de acuerdo con el principio de Nearest Neighbor [80]. Subsiguientemente se calcula el punto medio del cluster con mayor cantidad de keypoints.

El sistema de guía mantendrá el vehículo siguiendo la orientación con mayor densidad de keypoint. La navegación sobre estas zonas maximiza la probabilidad de cierre de lazo en fases de revisita. Claramente, puede darse la contingencia de

que en el horizonte no se encontrara ninguna zona con características sobresalientes. Para este caso particular se realizará una navegación rectilínea desde dicho punto, hasta nuevamente alcanzar una zona rica en características robustas.

La trayectoria de nodos resultantes (el corredor generado), será empleado como camino de referencia en visitas futuras. La estrategia de exploración termina eventualmente cuando el vehículo entra en una zona previamente conocida, o en el caso que la exploración deba ser interrumpida con el objetivo de inicializar el retorno del vehículo hacia el punto de partida.

### 2.2.3. Fase de Consolidación

La fase de consolidación de la navegación ocurre cuando el corredor construido es empleado para cruzar nuevamente la zona explorada. Así se realiza una visita de todos los nodos de manera ordenada, con la misión de alcanzar la zona de destino o el punto de partida. Los cierres de lazo que ocurrirán al visitar un determinado nodo harán que el mapa construido durante la fase de exploración se afiance utilizando esta nueva información y de esa manera se consolide.

Para lograr la navegación sobre el corredor, el sistema de guía debe encontrar cada nodo dentro del corredor en orden estricto. En los nodos internos, el vehículo empieza desde un nodo  $p_n$ , y es guiado de forma rectilínea de acuerdo con  $\vec{p}_n$  en dirección al nodo  $o_n$ . Durante el recorrido, el sistema de cierre de lazo está activo para detectar el momento cuando se puede observar el nodo  $o_n$ . En ese instante, el error entre la posición del vehículo estimada (que se encuentra en las inmediaciones de  $o_n$ ) y la posición real ( $v_n$ ) se corrige. Luego el sistema reanuda su navegación hasta el siguiente nodo  $p_n$ .

Tanto el sistema de odometría visual como el sistema de cierre de lazo dependerán a su vez del sistema SLAM que se esté utilizando como base, ya sea un sistema indirecto como [13] o uno directo como [14]. Sea como fuere la implementación particular del cierre de lazo, la elección de recorrer los nodos en forma ordenada da como resultado que se tenga una buena idea del próximo nodo a encontrarse. Luego utilizando dicha información se mejora significativamente la probabilidad de éxito del cierre de lazo.

Por otra parte, la utilización de mapas densos (como [14]) tiene una ventaja significativa en la etapa de visita. Como se estima todo el perfil de profundidades del suelo, es posible hacer una suposición de la apariencia del suelo desde una perspectiva completamente diferente y así tener mayor éxito en la instancia de cerrar lazo. De esta manera, la visita puede ser en la dirección contraria a la que se realizó en la etapa de exploración. Esto no es posible con un mapa de keypoints, porque la apariencia de estos cambia con la perspectiva. Para esta clase de mapas una solución podría ser la generación de dos mapas, una para la dirección de exploración y otra para la dirección la contraria.

### 2.2.4. Implementación del Algoritmo

El sistema de navegación C-SLAM desarrollado en secciones anteriores es completamente independiente del sistema SLAM subyacente. Dicho sistema SLAM debe, en primer lugar, proveer a C-SLAM de la medición odométrica de su posición y, en segundo lugar, debe encargarse de realizar los cierres de lazo, ambas características que por definición debe poseer cualquier sistema SLAM.

Por estas razones, se probó el C-SLAM en conjunto con dos sistemas SLAM visuales monoculares completamente diferentes en su construcción. Ellos son los sistemas indirecto-no-denso ORB-SLAM [13] y por otro lado el sistema directo-denso de confección propia, que combina el sistema de odometría visual desarrollado en el capítulo 3, y el back-end desarrollado en [14].

## 2.3. Casos de Estudio

En esta sección, la adecuación de enfoque propuesto para la navegación sobre suelo autosímil se valida cuantitativamente en escenas tanto simuladas como naturales. Los parámetros de operación principales se configuraron de manera de que el vehículo mantenga una altitud adecuada (para poder tener buena visibilidad) y la velocidad del vehículo sea conforme a las velocidades máximas permitidas por el sistema SLAM subyacente. En este escenario podrá apreciarse la borrosidad, autosimilitud y ondas cáusticas.

### 2.3.1. Caso de Estudio I: Ambiente Simulado y Sistema SLAM Indirecto No-denso

Para el primer caso de estudio se utilizó una escena simulada. La misma fue realizada mediante el paquete de software GAZEBO [81]. Se simuló un suelo arenoso, con numerosas lomadas, depresiones y piedras colocadas aleatoriamente. La escena esta perturbada por borrosidad de tono azulado, la cual se incrementa junto con la distancia. El sistema SLAM subyacente utilizado es el descrito en el trabajo [13].

En las figuras 2.3 y 2.2 se muestra la trayectoria de exploración del vehículo y la imagen cruda capturada por la cámara.



FIGURA 2.2: Escena simulada. Se observa la visualización realizada por la cámara en tres instantes a lo largo de la trayectoria

Desde la trayectoria realizada por el vehículo se puede observar la tendencia a la navegación sobre zonas con alta cantidad de características sobresalientes. Por el contrario, las zonas con baja cantidad de características (como los bancos de arena) son evitadas por el sistema C-SLAM.

En la figura 2.4 se puede apreciar la evolución de la cantidad de características sobresalientes en el tiempo. En la misma figura se observa la trayectoria real, obtenida extrayendo datos de la simulación. Se puede observar cómo las trayectorias reales coinciden físicamente durante la exploración, cómo también durante las revisitas.

### 2.3.2. Caso de Estudio II: Ambiente Natural y Sistema SLAM Directo Denso

Para este caso de estudio se utilizó un escenario natural, descrito en detalle en el capítulo 6. El escenario contiene auto-similitud resultado de la abundancia de arena, un horizonte difuso y está afectado por ondas cáusticas. El vehículo utilizado

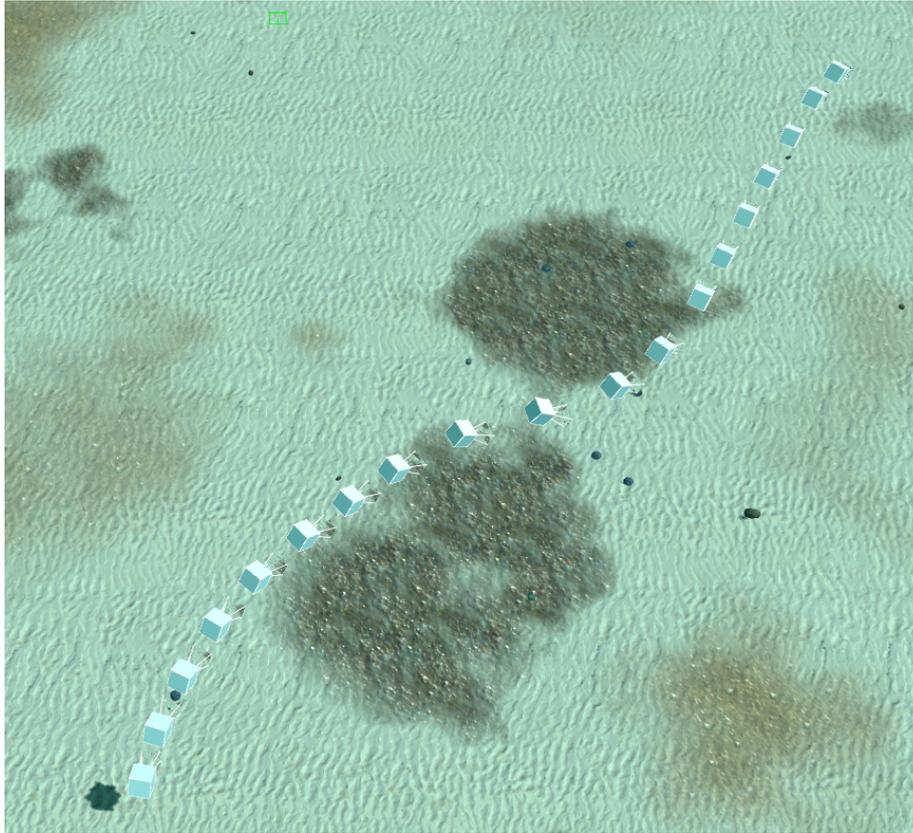


FIGURA 2.3: Escena simulada. Se observa la escena en conjunto con la trayectoria de exploración realizada por el vehículo.

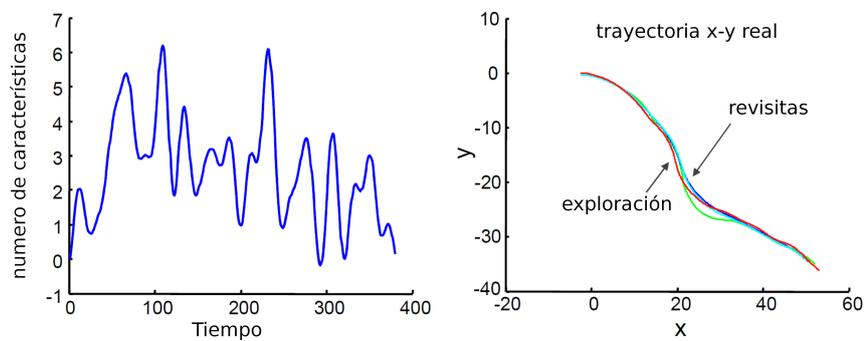


FIGURA 2.4: Izquierda: evolución de las características robustas detectadas, derecha: trayectorias reales durante las fases de exploración y consolidación. En rojo se representa la trayectoria de exploración, mientras que en verde, azul y celeste las revisitas.

contiene una cámara cuya inclinación es de 60 grados con respecto a la horizontal. En este caso, se utilizó como SLAM en base a un sistema directo-denso de confección propia con un sistema de odometría visual descrito en el capítulo 3 y un back-end basado en el trabajo [14].

Para evaluar el rendimiento, se emplea un indicador de la densidad de cierres de lazo realizados durante la navegación. Idealmente, durante la fase de consolidación, en cada nodo de la secuencia se debería poder realizar el cierre de lazo exitosamente.

Para aumentar la robustez, se relajó la condición de ordinalidad estricta en el

seguimiento de nodos, permitiendo el reconocimiento de un nodo dentro de un intervalo pequeño de nodos subyacentes. De esta manera la probabilidad de identificación del próximo nodo se aumenta sacrificando cierto grado de justeza en favor de sostener el seguimiento de la cadena de nodos.

Otro indicador de rendimiento es la consolidación del corredor, donde se mejora progresivamente el mapa obtenido con cada revisita.

En la figura 2.5 se presentan dos imágenes del ambiente subacuático. La imagen a la izquierda ilustra la escena con un horizonte difuso, piedras autosímiles y partículas en suspensión. La imagen a la derecha, por el contrario, muestra una escena perturbada mediante ondas cáusticas. Ambos tipos de escenas son parte del caso de estudio presente, en diferentes momentos durante la navegación. El éxito del sistema visual en la presencia de ambos tipos de perturbaciones requiere cierto nivel de robustez. En escenas borrosas, el bajo gradiente en la imagen hace dificultosa la estimación de la profundidad. A su vez, las escenas perturbadas con ondas cáusticas no conservan la condición de constancia de brillo (véase los capítulos 3 y 4).



FIGURA 2.5: Escenario subacuático. Izquierda: escena capturada con horizonte difuso, rocas autosímiles y partículas en suspensión. Derecha: escena perturbada mediante ondas cáusticas.

El número de cierres de lazo exitosos se ilustra en la figura 2.6, en la columna izquierda. Se puede apreciar un incremento permanente de cierres de lazo. Este hecho atestigua la robustez del sistema.

La figura 2.6 en la columna de la derecha, muestra el proceso de mejora del mapa mientras se realizaban las revisitas. El mapa está descrito a través de una nube de puntos, cada uno de los cuales representa un punto físico en el espacio, cuyas coordenadas fueron determinadas con la estimación del mapa denso. En la figura se aprecia cómo esta nube de puntos se densifica, mientras que las trayectorias se relocalizan ligeramente. En la figura 2.7 se ilustra el corredor final desde distintos puntos de vistas en direcciones principales.

Se concluye categóricamente que si bien las trayectorias estimadas no son coincidentes, e incluso algunos tramos difieren entre sí más que otros, el retorno en numerosas revisitas fue exitoso. Esta aparente dispersión de trayectorias no es determinante para nada en el uso del corredor ya que las trayectorias reales son completamente desconocidas. También se desprende del conteo de nodos identificados que el sistema visual guía el movimiento de la cámara en forma correcta reconociendo casi en forma ordenada la cadena de nodos en un sentido y en otro. El sistema visual C-SLAM propuesto manifiesta claramente la habilidad de construir un corredor robusto en ambientes complejos y poder retornar en cada cruce.

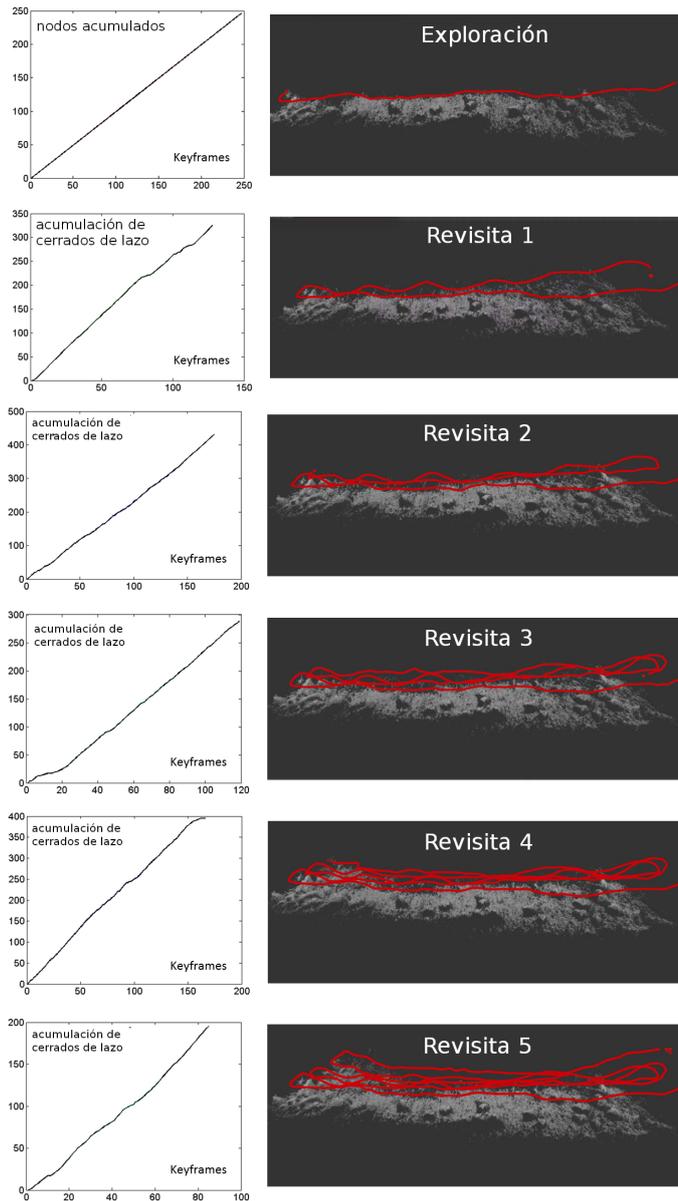


FIGURA 2.6: Construcción del corredor. En la columna izquierda: la acumulación de cierres de lazo en el tiempo. En la columna derecha: la mejora progresiva del mapa del corredor con cada revisita.

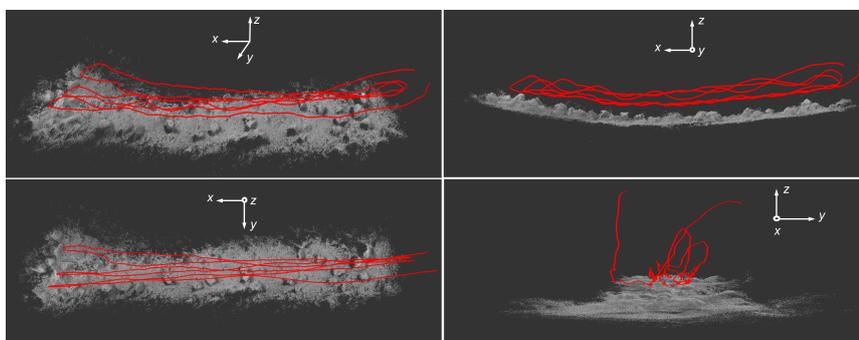


FIGURA 2.7: Visualización del corredor construido bajo diferentes perspectivas.



## Capítulo 3

# Odometría Visual

### 3.1. Introducción

Interpretamos el concepto de Odometría Visual como el mecanismo que provee la autolocalización de la cámara en base a un mapeo simultáneo. Usualmente se incluye en el bloque front-end del SLAM. Este concepto no incluye el cierre de lazo comúnmente incluido en el otro bloque, el back-end del SLAM. El C-SLAM elaborado y descrito en el capítulo anterior incluye ambos bloques para la navegación por corredor.

Es posible clasificar los métodos de odometría visual en cuatro categorías, en relación con la clase de información utilizada (directos e indirectos) y la cantidad de estimaciones que realizan (densos y no-densos). [82] [83].

En este capítulo, se realiza un breve análisis de los métodos existentes con vistas de su aplicación en un contexto subacuático. Una de las conclusiones que se extraen de este análisis es que los métodos directos-densos tienen ventajas con respecto al resto de los métodos, particularmente en su aplicación en ambientes subacuáticos. Estos métodos son, simultáneamente, uno de los métodos menos explorados en la literatura. Subsecuentemente, en la tesis, se realizan necesarios avances en el desarrollo de un método de odometría visual directo-denso, y se muestran sus ventajas con respecto a métodos en el estado del arte en ambientes subacuáticos reales.

#### 3.1.1. Métodos Indirectos-No-Densos

Este tipo de métodos son los más estudiados en la literatura [84], [13]. Los principios fundamentales de este tipo de métodos se describen a continuación.

En primera instancia, la imagen capturada por la cámara es pre-procesada de manera de obtener una representación de alto nivel de la escena, es decir, con un grado conceptual y/o semántico más acabado que las simples características fotométricas y cromáticas de los píxeles de la imagen. En general, se procura extraer zonas de la imagen que sean altamente diferenciables, y proclives a ser descritas inequívocamente, facilitando de esa manera la búsqueda de la zona analizada a lo largo de la secuencia de imágenes. Es posible detectar líneas rectas [85], zonas con textura o tonalidad similar, o más generalmente se pueden extraer los llamados “keypoints” [37], [38].

La condición de detección de una misma zona en la secuencia de imágenes, conjuntamente con algoritmos robustos como RANSAC y FLANN [86], [87], se utilizan con resultados de geometría epipolar [88] y optimización [89], para obtener conjuntamente la representación de la posición tridimensional de las zonas detectadas, así como también la postura de la cámara a lo largo de la secuencia de imágenes.

La naturaleza indirecta-no-densa de estos métodos conlleva ciertas ventajas y, paralelamente, ciertas desventajas. El hecho de utilizar una representación de alto

nivel de la escena permite que la cantidad de información a procesar se reduzca considerablemente, lo que da como resultado algoritmos muy eficientes computacionalmente. Esta reducción en la cantidad de información también permite realizar una optimización global, en toda la secuencia de imágenes, que da como resultado una gran precisión en las estimaciones.

Pero, al mismo tiempo, un mapa así obtenido es un mapa de no-denso. Si bien excelente para la localización relativa del vehículo, este mapa tiene muchas limitaciones en las situaciones en las que se necesite tener un mapa detallado del terreno. Como ejemplo, con mapas así construidos, no es posible estar seguro si hay una colisión inminente en una zona carente de una descripción de alto nivel. Al mismo tiempo, utilizar estas descripciones restringe los ambientes en los que se los puede utilizar. Estos ambientes necesariamente tienen que presentar suficientes zonas diferenciables para generar las descripciones. Las estructuras hechas por el hombre están caracterizadas por su gran cantidad de bordes y esquinas, que son proclives a ser descritas en alto nivel. Pero los terrenos no estructurados pueden poseer baja diferenciabilidad, incidiendo así en el desempeño de estas técnicas. Cabe esperar que los ambientes subacuáticos, por sus características visuales particulares, sean especialmente difíciles para estas técnicas [82], [83].

### 3.1.2. Métodos Directos-Densos

La literatura en este caso es más reducida, y las técnicas utilizadas son de naturalezas diversas. Una implementación es la detallada en [90]. Se basa, en general, en la información de profundidad de cámara que aporta cada uno de los píxeles de la imagen, es decir, en la información fotométrica de la escena frente a la cámara. En dicho trabajo se utiliza un volumen de costo, en el cual se acumula evidencia de que determinado píxel posea una profundidad específica. La naturaleza de esta información es la diferencia entre la muestra del píxel obtenida en una posición de la cámara, frente a una muestra obtenida en otra posición (error fotométrico). De esta manera, para encontrar la profundidad de un píxel, simplemente se busca entre todas las profundidades posibles presentes en el volumen de costo, y se elige aquella que posea un menor error fotométrico medio. Para disminuir el ruido en la estimación se utiliza un método de optimización, para encontrar las profundidades para cada píxel que posean tanto un costo pequeño como también continuidad con las profundidades de píxeles circundantes, logrando así una mayor regularidad de la profundidad estimada. La optimización emplea el gradiente de intensidad en la imagen para separar zonas de la escena angulosas (cantos, bordes), sobre las cuales la regularización se torna menos intensa.

Otra técnica es la descrita en [91]. Los autores en este caso procuran triangular cada píxel de la imagen para encontrar su profundidad. De manera de encontrar el píxel correspondiente en las imágenes subsiguientes, se realiza una búsqueda exhaustiva del mismo a lo largo de la línea epipolar. Para cada triangulación obtiene la media y varianza de la profundidad posible. Se utiliza un filtro de Kalman para fusionar las medidas realizadas desde diferentes perspectivas.

Una técnica similar a la anterior es la descrita en [14], con la principal variación de no estimar profundidad para los píxeles con gradiente pequeño. La motivación es que los píxeles con poco gradiente son más difíciles de encontrar mediante una búsqueda en la línea epipolar y, naturalmente, la estimación de su profundidad tiene un error considerable. De esta manera no se obtiene un mapa denso sino semi-denso. Pero a cambio se logra una gran robustez, que el autor muestra en experimentos que se extienden por varios kilómetros.

En todos los casos, se estima la posición relativa de la cámara utilizando técnicas de optimización [92], las cuales obtienen la postura  $T \in \mathfrak{R}^{4 \times 4}$  que minimice el error entre la imagen actual y una imagen virtual generada mediante una estimación de  $T$  actual, formalmente:

$$F(T) = \sum_{u \in \Omega} (|I_{kf}(K\pi(Tp_u)) - I_a(u)|) \quad (3.1)$$

Donde  $\Omega$  es el espacio de las posibles coordenadas de los píxeles pertenecientes a la imagen capturada por la cámara,  $u$  es un píxel particular,  $I_a$  es la imagen más reciente capturada por la cámara,  $p_u \in \mathfrak{R}^3$  es el punto estimado perteneciente al píxel  $u$ .  $\pi$  es la función de proyección,  $K \in \mathfrak{R}^{3 \times 3}$  es la matriz intrínseca de la cámara, que contiene los parámetros de la geometría de la lente.

Es posible analizar las diferentes características de estos métodos observando las suposiciones que cada uno realiza en cuanto a la distribución de probabilidad de la profundidad. En [91] y [14], la estimación de la profundidad es paramétrica, suponiendo una distribución gaussiana. En [90], por otro lado, la suposición es no-paramétrica.

Una distribución paramétrica funciona correctamente en ambientes con buena visibilidad y buen gradiente, como las estructuras hechas por el hombre. Pero cuando se utilizan los métodos en ambientes difíciles, donde grandes ambigüedades son comunes, las ventajas de una distribución no-paramétrica deberían ser evidentes.

## 3.2. Nuevo Método de Odometría Visual

En vista de las limitaciones ofrecidas por métodos densos del estado del arte en ambientes submarinos, es que se propone una mejora en base a un método novedoso para la estimación directa-densa de la profundidad. Junto a los detalles del desarrollo del método propuesto, se mencionan los puntos de contacto con el otros métodos del estado del arte.

El sistema propuesto esta basan en métodos de estimación de profundidad basado en volumen de costo, específicamente en el trabajo [90]. Un diagrama en bloques del sistema puede apreciarse en la figura 3.1.

### 3.2.1. Estimación de la Postura

Para estimar la postura de la cámara actual, como se mencionó anteriormente, se utiliza la información del mapa actual (compuesta por profundidad y la apariencia relativa a una postura de cámara referencia) y la imagen actual, buscando la postura  $T$  que minimice la función descrita en la ecuación 3.1.

Para esta tarea existen muy buenas implementaciones en la literatura. En este trabajo se utiliza específicamente la desarrollada en [14].

### 3.2.2. Cálculo y Acumulación del Costo

Para la estimación de la profundidad, se utilizan técnicas basadas en volumen de costo, pudiéndose citar como ejemplo de este tipo de sistemas el trabajo [90]. Este volumen almacena una medida de la penalización (el costo) que se incurre al elegir como correcta una determinada profundidad  $\eta$  para los píxeles  $u$  pertenecientes a una imagen de referencia  $I_{kf}$ , localizada en una postura de referencia  $T_{kf}$ . El volumen está compuesto por pequeños volúmenes indivisibles, llamados voxels, con

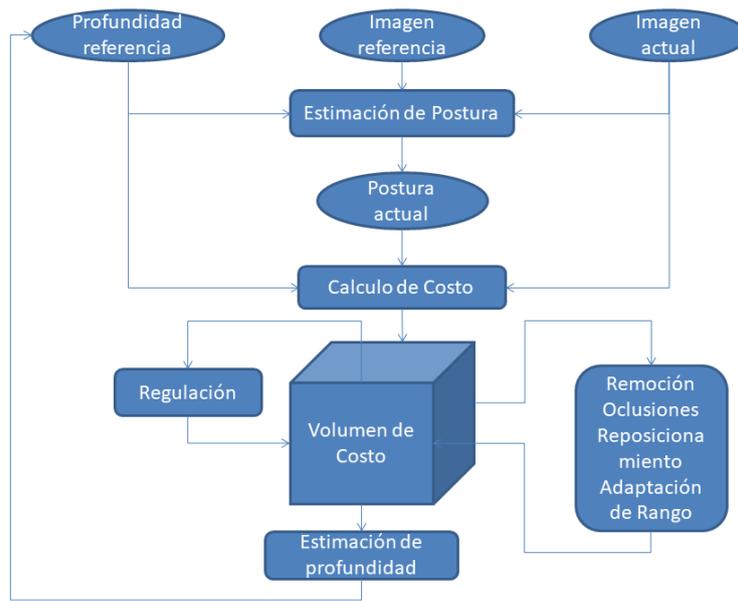


FIGURA 3.1: Estimación de profundidad, diagrama en bloques

una dimensión de  $u_{dim} \times d_{dim}$ , donde  $u_{dim} \in \mathbb{R}^2$ . Normalmente  $u_{dim}$  coincide con las dimensiones de las imágenes utilizadas, y  $d_{dim}$  puede elegirse libremente.

En trabajos tales como [93] [94], se calcula el costo utilizando la función TSDF (truncated signed distance function) [95]. Esta función codifica elegantemente tanto la incertidumbre en la medición, como las zonas de espacio libre y las ocluidas. La utilización de la TSDF surge en la literatura en aplicaciones que utilizan sensores de profundidad de muy bajo ruido, como ser el kinect [96]. Sistemas utilizando este tipo de sensores fueron presentados en trabajos como [95] [97], donde se obtuvieron excelentes resultados. A continuación se mostrará que, en contextos monoculares, para utilizar la función TSDF es necesario utilizar estimaciones a priori de la profundidad. Estas estimaciones previas son generalmente ruidosas. Por este motivo es que se elige utilizar una implementación alternativa. Para comenzar el análisis, se ahondará en la aplicación de la TSDF en sistemas monoculares.

La función TSDF  $\Psi$  es definida tal que:

$$\Psi(u, d, \eta, \mu) = \begin{cases} \min(1, \frac{|\eta(u) - d|}{\mu(u)}) \text{sign}(\eta(u) - d) & \text{si } \eta(u) - d \geq \eta(u) - \mu(u) \\ \text{nulo} & \text{si no} \end{cases} \quad (3.2)$$

donde  $u$  es el píxel analizado en la imagen de referencia  $I_{kf}$ ,  $d$  es la profundidad específica a la cual se desea calcular el costo (dentro de la profundidad representables dentro del volumen de costo),  $\eta(u)$  es la medida de profundidad relacionada con el píxel  $u$  y  $\mu(u)$  su incertidumbre. Como se ve, la representación del costo mediante TSDF requiere una estimación de  $\eta$  y  $\mu$ .

De esta manera, en trabajos que utilizan cámaras monoculares, como [93] [94], estiman la profundidad en cada muestra  $k$ , de manera similar a como se realiza en [14] [91], es decir:

$$\begin{aligned}
\eta_o(u) &= \arg \min_{d_{ep}} (|I_{kf}(u) - I_k(p(KT_k(d_{ep}K^{-1}u)))|) \\
d_{ep} &\in [\eta_{k-1}(u) - \mu_{k-1}(u), \eta_{k-1}(u) + \mu_{k-1}(u)] \\
\mu_o(u) &= \mu_{1 \text{ pixel}}(\eta_o(u)) \\
\eta_k(u) &= \eta_{k-1}(u) + (\eta_o(u) - \eta_{k-1}(u)) \frac{\mu_o(u)}{\alpha \mu_{k-1}(u) + \mu_o(u)} \\
\mu_k(u) &= \frac{\alpha \mu_{k-1}(u) \mu_o(u)}{\alpha \mu_{k-1}(u) + \mu_o(u)}
\end{aligned} \tag{3.3}$$

donde  $I_{kf}$  es la imagen en la posición de referencia,  $I_k$  es la imagen capturada en el instante  $k$ ,  $K$  y  $T_k$  son las matrices intrínsecas y extrínsecas de la cámara, respectivamente,  $\pi()$  es la función de proyección y  $\mu_{1 \text{ pixel}}$  es la incertidumbre en la profundidad relacionada a un corrimiento de un píxel en la línea epipolar. Nótese que  $\mu$  se estima simplemente utilizando un filtro de Kalman donde  $\alpha$  modela la incertidumbre en la predicción de  $\eta_k(u)$  y  $\mu_k(u)$ .  $p(KT_k(d_{ep}K^{-1}u))$  corresponde a la línea epipolar en  $I_k$  relacionada a  $u$  en  $I_{kf}$ .

Las estimas previas  $\eta_{k-1}(u)$  y  $\mu_{k-1}(u)$  utilizadas para estimar  $\eta_k(u)$  y  $\mu_k(u)$  son considerablemente ruidosas, como consecuencia de errores en la estimación de la postura  $T$  y, principalmente, cuando la calidad de la medición  $I_k$  es baja. Esto puede ocurrir cuando la escena presenta autosimilitud, poca textura o borrosidad considerable, situación que hace difícil encontrar el mínimo descrito en la ecuación 3.3.

Para evitar la utilización de estimaciones previas ruidosas, en este trabajo se implementa el costo similar al presentado en [98], dado por una combinación afín de error fotométrico y error de gradiente fotométrico con valores de saturación para cada error. Es decir:

$$\begin{aligned}
\Psi(u, d) &= (1 - \gamma)(\min(|I_{kf}(u) - I_k(p(KT_k(dK^{-1}u)))|, \alpha)) \\
&\quad + \gamma(\min(|G_{kf}(u) - G_k(p(KT_k(dK^{-1}u)))|, \beta))
\end{aligned} \tag{3.4}$$

donde  $\gamma \in [0, 1]$ ,  $G_k$  es el gradiente de la imagen  $I_k$ ,  $\alpha$  y  $\beta$  constantes de saturación. Utilizando cámaras monoculares (o binoculares), esta última formulación del costo  $\Psi$  no requiere la utilización de mediciones previas, lo que permite una estimación no-paramétrica robusta de  $\eta$  y  $\mu$ , como se verá en secciones siguientes. La utilización de un  $\gamma \sim 1$  aumenta la robustez frente a cambios de iluminación, es decir cuando el costo sopesa mucho más los cambios fotométricos que los errores fotométricos, ya que  $G_k$  no se ve tan severamente afectado a cambios de iluminación como  $I_k$ .

Asociada a la medida del costo  $\Psi_k$ , se utiliza una medida de la información contenida en la muestra  $k$ ,  $\Phi_k$ . Para acumular el volumen de costo los  $\Psi_k$  y  $\Phi_k$  obtenidos en diferentes  $k$ , se elige utilizar una implementación similar a [95], [97], que responde a la ecuación:

$$\begin{aligned}
F_k(u, d) &= \frac{W_{k-1}(u, d)F_{k-1}(u, d) + \Phi_k(u, d)\Psi_k(u, d)}{W_{k-1}(u, d) + \Phi_k(u, d)} \\
W_k(u, d) &= W_{k-1}(u, d) + \Phi_k(u, d)
\end{aligned} \tag{3.5}$$

A diferencia de los trabajos mencionados, no utiliza un  $\Phi_k = 1$ , que significaría realizar una promediación. Este valor es ajustado como se detalla en secciones siguientes.

### 3.2.3. Estimación de Profundidad y Varianza

La profundidad  $\eta$  correspondiente al píxel  $u$  en la imagen de referencia  $I_{kf}$  se calcula entonces con una búsqueda exhaustiva en el volumen de costo.

$$\eta_e(u) = \arg \min_{d_u} (F_k(u, d_u)) \quad (3.6)$$

$$d_u \in d_{min}, d_{max}$$

donde  $d_{min}$  y  $d_{max}$  corresponden a los valores mínimos y máximos de profundidad representables dentro el volumen de costo.

Para estimar  $\mu$ , se utiliza una técnica similar a [94]. Se determina la distancia  $d_\mu$  tal que el costo  $F(u, d_\mu) = \alpha F(u, \eta)$ , donde  $\alpha$  es un factor de escala. De esta forma, píxeles  $u$  que posean  $F(u, d)$  para  $d_u \in d_{min}, d_{max}$  con un mínimo marcadamente agudo tendrán una incertidumbre asociada pequeña. De manera recíproca, si no tienen un mínimo agudo, el valor  $\alpha F(u, \eta)$  estará alejado de  $F(u, \eta)$  y así  $u$  tendrá una incertidumbre elevada. Se calcula tanto  $d_\mu^{sup}$   $d_\mu^{inf}$ , correspondientes al valor inmediato superior e inferior a  $\eta$ . Finalmente se estima  $\mu$  como:

$$\mu_e(u) = \sqrt{\max(|d_\mu^{sup} - \eta_e(u)|, |d_\mu^{inf} - \eta_e(u)|)} \quad (3.7)$$

A diferencia de [94], se distingue entre diferentes casos particulares en la estimación de  $\mu$ . Los costos  $F(u, d)$  asociados a un  $u$  particular, no siempre superan el valor  $\alpha F(u, \eta)$  para distancias altas y bajas. Esto puede indicar que la incertidumbre puede ser alta, más allá de la distancia de incertidumbre representable mediante un volumen de costo definido dentro de  $d_{min}$  y  $d_{max}$ . Por esta razón, la incertidumbre para un determinado  $u$  puede no estar bien delimitada y no se puede asegurar que la medición asociada sea correcta. De esta manera se logra diferenciar píxeles  $u$  que no poseen una estimación de profundidad confiable.

### 3.2.4. Reposicionamiento del Volumen de Costo

En [90] [93] [94], al momento de requerirse una nueva imagen de referencia  $I_{kf}$ , el costo acumulado  $F_k$  y  $W_k$  se re-inicializan en 0, perdiendo así toda información previamente acumulada. Claramente, sería una mejor estrategia utilizar los costos observados con anterioridad. Esto permitiría obtener mayores paralajes entre las imágenes, dando así mayor robustez y menores niveles de ruido en las estimaciones  $\eta$  y  $\mu$ .

En [97], se realiza un sistema de SLAM utilizando en el sensor Kinect [96]. Se utiliza un costo basado en TSDF, de manera análoga al realizado en [95]. La innovación principal del trabajo en cuestión radica en que, a medida que la cámara se mueve, el volumen de costo es trasladado en conjunto. De esta manera se logra acumular mediciones a lo largo de toda la trayectoria de la cámara de manera continua. El volumen de costo se lo construye con una topología cúbica, y cuando se necesita alinearlos con la cámara simplemente se lo traslada de manera de que la cámara esté posicionada en el centro del volumen de costo. La utilización de un volumen de costo rectangular da la posibilidad de que al trasladarlo no se incurre en ningún error de reposicionamiento (véase [97]).

Utilizar un volumen de costo con tal topología tiene varias desventajas, como se ilustra en la figura 3.2. El frustum de la cámara no coincide con la forma del volumen de costo cúbico, por lo que muchos lugares de memoria no son utilizados. También,

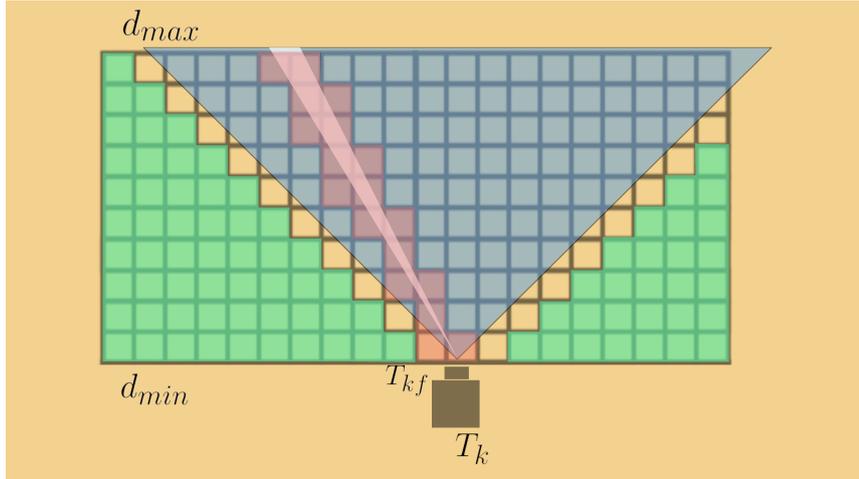


FIGURA 3.2: Volumen de costo cúbico. Se observa un corte transversal del volumen de costo con dimensión  $20 \times 10$ .  $T_{kf}$  representa la postura del volumen de costo,  $T_k$  es la postura de la cámara en el instante  $k$ ,  $d_{min}$  y  $d_{max}$  representan las distancias en  $z$  (relativo a la postura de la cámara) mínimas y máximas representables en el volumen de costo. En Azul se ve el frustum de la cámara en  $k$ , en blanco se ve el frustum de un píxel particular observado por la cámara. En rojo se representan los vóxeles dentro del volumen de costo afectados por el píxel analizado. Se observa una resolución baja, particularmente cerca del origen de la cámara. En verde se observan los vóxeles no afectados por ningún píxel de la cámara. Se observa que gran cantidad de los vóxeles dentro del volumen de costo no son utilizados.

la forma cúbica conlleva una cuantización espacial constante en las tres dimensiones. Dada la naturaleza proyectiva de las cámaras tipo pinhole, sería más provechoso una topología que permita tener una resolución alta a distancias cercanas, y una resolución baja a distancias altas.

Por el contrario, en este trabajo se elige utilizar un volumen de costo con una topología conforme al frustum de la cámara, como se ilustra en la figura 3.3. Se logra así una mejor utilización del espacio de memoria, y una mejora en resolución espacial. Al mismo tiempo se utiliza una parametrización en distancia inversa (no ilustrada en las figuras para simplificar el análisis).

Como contrapartida de estas mejoras, el algoritmo de reposicionamiento del volumen de costo es considerablemente más complejo, en comparación al utilizable con un volumen de costo cúbico. Véase la figura 3.4. El movimiento del volumen de costo no corresponde a un simple traslado como en el caso de topologías cúbicas, sino que responde a una reproyección de un volumen de costo sobre el nuevo volumen de costo. Acá se supone que la escena es compartida al menos en una parte significativa por ambos volúmenes de costo.

El algoritmo se resume en las siguientes ecuaciones:

$$\begin{aligned}
 F_r(u, d) &= F_a(u_p, d_p) \\
 \gamma &= T_k K^{-1} d \cdot u \\
 u_p &= p(K\gamma) \\
 d_p &= \gamma_z
 \end{aligned} \tag{3.8}$$

donde  $F_a$  y  $F_r$  son los volúmenes de costo anterior y el reposicionado con coordenadas

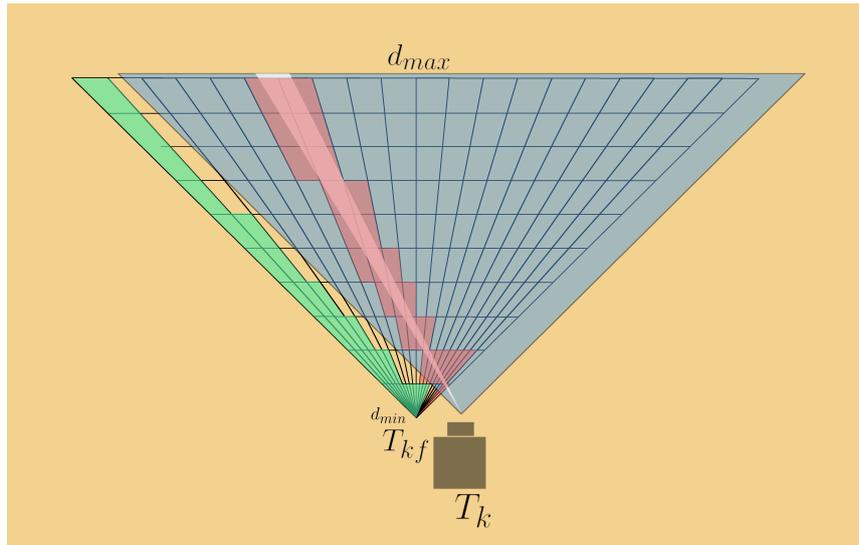


FIGURA 3.3: Volumen de costo frustum. Se observa un corte transversal del volumen de costo con dimensión  $20 \times 10$ .  $T_{kf}$  representa la postura del volumen de costo,  $T_k$  es la postura de la cámara en el instante  $k$ ,  $d_{min}$  y  $d_{max}$  representan las distancias en  $z$  mínimas y máximas representables en el volumen de costo. En azul se representa el frustum de la cámara actual. En blanco se analiza un píxel particular observado por la cámara. En rojo se representan los vóxeles dentro del volumen de costo afectados por el píxel analizado. En este caso se observa una buena resolución espacial (numerosos vóxeles rojos son afectados por el mismo píxel), especialmente cerca del origen de la cámara. En verde se representan los vóxeles dentro del volumen de costo no afectados por ningún píxel de la cámara actual. Se observa que en este caso el número de los mismos es considerablemente menor

$u_p, d_p$  y  $u, d$ , respectivamente. Se ve que esta ecuación corresponde a una reproyección del volumen de costo.

A su vez, para cada posición en el volumen de costo destino, existen varios puntos en el volumen de costo origen, como se ilustra en la figura 3.4. Como la implementación de la 3.8 es costosa computacionalmente, realizar una interpolación precisa no es factible. Por el contrario, se elige realizar una aproximación del tipo Nearest Neighbor [99], y modelar el error en el traslado en la ecuación para el traslado de  $W_n$ , siendo:

$$W_{T_k}(u, d) = \delta \cdot W_{T_{k-1}}(u, d) \quad (3.9)$$

Se encuentra empíricamente que un valor de  $\delta$  de 0,5 modela correctamente dicho error.

### 3.2.5. Remoción de Oclusiones

Utilizando un costo basado en TSDF, tanto las oclusiones como la incertidumbre queda elegantemente representada en la información almacenada en el volumen de costo. En cambio, utilizando un costo como el descrito en la ecuación 3.4, no se encuentra una forma directa de representar las oclusiones. Este inconveniente se acentúa a medida que se reposiciona el volumen de costo, donde zonas ocluidas y

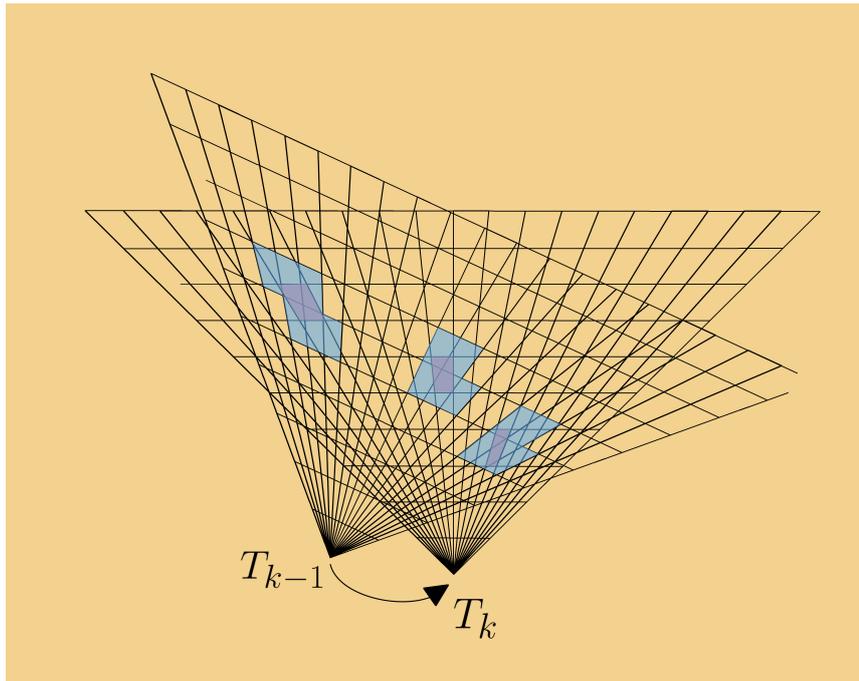


FIGURA 3.4: Error de reposición en volumen de costo con topología frustum.  $T_{k-1}$  es la postura del volumen de costo anterior, mientras que  $T_k$  es la postura actual del volumen de costo. Se puede apreciar en rojo en vóxel destino dentro del volumen de costo posicionado en  $T_k$ , al cual le corresponden en este caso los vóxeles dentro del volumen de costo posicionado en  $T_{k-1}$  marcados en azul.

de-ocuidas normalmente aparecen en la escena. Para resolver este inconveniente, se desarrolló la siguiente metodología de remoción de oclusiones.

Los vóxeles del volumen de costo correspondiente a zonas ocluidas, si bien imprescindible para encontrar  $\eta$  y  $\mu$  en la imagen de referencia correspondiente  $I_{k,f}$ , no contiene ninguna información si se utiliza una nueva imagen de referencia. Esto se puede observar en la figura 3.5. En resumen, los píxeles de objetos ocluidos no son útiles para la estimación de profundidad desde la nueva postura de referencia, y son inicializados apropiadamente como valores inciertos. Su estimación de profundidad será mejorada paulatinamente en las siguientes imágenes donde estos objetos son capturados.

### 3.2.6. Muestreo de la Línea Epipolar

Un problema en las técnicas que utilizan volumen de costo es la ineludible discretización del espacio, según las dimensiones del volumen de costo utilizado  $u_{dim}$   $d_{dim}$  en  $u$  y en  $d$  respectivamente. En [95], la discretización de un volumen de costo impone severas limitaciones en la dimensión del área a reconstruir, así como la resolución de reconstrucción misma. En la literatura se propusieron varias soluciones a este problema, como utilizar representaciones del volumen eficientes en memoria [100], o utilizar un volumen de costo continuo como en [97].

Una observación clave es que en estos trabajos, que utilizan sensores de profundidad, la realización de observaciones y el filtrado final de las mismas son procesos diferentes. Se obtienen mediciones muy precisas, que luego se acumulan en un volumen de costo cuantizado para obtener la reconstrucción final.

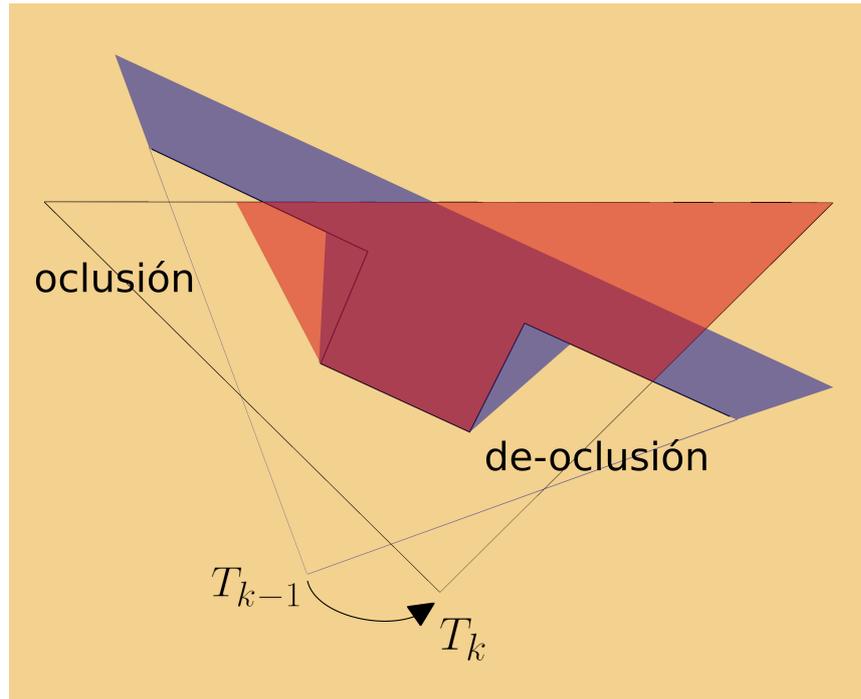


FIGURA 3.5: Oclusiones. Se representa la profundidad real mediante una línea negra. En azul se representan los vóxeles ocluidos en el volumen de costo previo. En rojo se representan los vóxeles ocluidos en el nuevo volumen de costo. La zona de-ocluída no contiene información útil para el volumen de costo situado en  $T_k$ , dado que esa información fue heredada del volumen de costo situado en  $T_{k-1}$ , y desde esa perspectiva esa zona no era visible. Se propone remover toda la zona ocluida antes de re-proyectar la información de un volumen de costo a otro. Esto significa re-inicializar toda la zona azul.

En un contexto monocular (o binocular), el problema es diferente. Como se analizó en secciones anteriores, el volumen de costo no solamente se encarga de acumular las medidas realizadas por el sensor (medidas fotométricas), sino que también se encarga de realizar las mediciones de profundidad mismas. Por lo que la discretización del volumen de costo afecta tanto las mediciones de profundidad, como la calidad de la reconstrucción. Se analizará aquí la consecuencia de obtener mediciones en un volumen de costo cuantizado.

En las figuras 3.6 y 3.7, se observa que bajo un movimiento moderado de la cámara, un determinado vóxel proyecta en varios píxeles de la línea epipolar. Lo que genera una incertidumbre en cuanto al valor de  $d$  que se debería utilizar para actualizar el costo descrito en la ecuación 3.4. En [90], se utiliza implícitamente el valor de  $d$  correspondiente al centro del vóxel analizado. Esta elección funcionará bien mientras el cambio de brillo dentro de la línea epipolar sea suave. Si se intenta recuperar zonas donde el brillo cambia con una frecuencia alta, se puede esperar que el valor de  $d$ , elegido en el centro del vóxel, no siempre corresponda al mismo punto físico observado desde el píxel  $u$  de la imagen  $I_{kf}$ . Este fenómeno viola la hipótesis de constancia de brillo, en la que se fundamenta la validez de la ecuación 3.4. Esta imprecisión se refleja como ruido en la estimación de profundidad final.

Se propone entonces la siguiente solución. Se actualizan los vóxeles en el volumen de costo relacionados a un píxel  $u$  solo si la longitud de la línea epipolar correspondiente  $e_{long}$  sea menor a  $d_{dim}$ . Así, todos los píxeles dentro de la línea epipolar

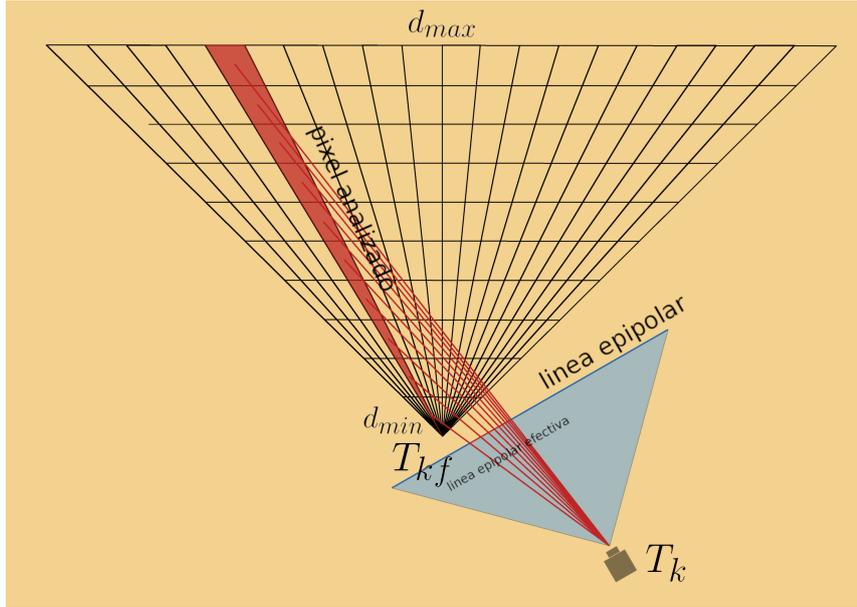


FIGURA 3.6: Línea epipolar obtenida mediante un pequeño desplazamiento de la cámara. Las posibles profundidades de  $I_{k_f}$  proyectan en  $I_k$  en una sección pequeña de la línea epipolar, marcada en rojo. Así, un determinado vóxel proyecta en una pequeña sección de la línea epipolar.

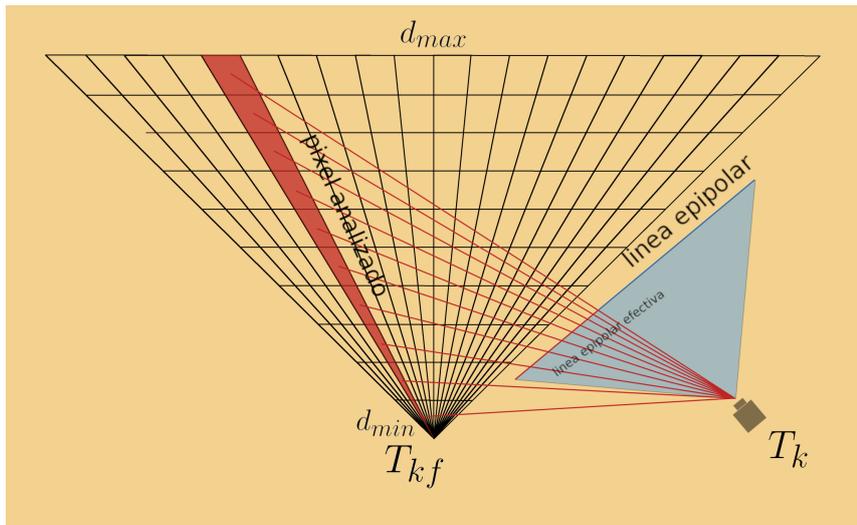


FIGURA 3.7: Línea epipolar obtenida mediante un movimiento de la cámara moderado. En este caso, las posibles profundidades dentro de  $I_{k_f}$  proyectan en una sección amplia de la línea epipolar. Por lo que un determinado vóxel dentro del volumen de costo toma varios pixeles dentro de la línea epipolar.

serán utilizados para actualizar el volumen de costo y no ocurrirá en un submuestreo de la línea epipolar. A su vez, líneas epipolares de baja longitud se relacionan con poco paraleje y baja información útil para estimar profundidades. Por esta razón, se ajusta  $\Phi_k$  para reflejar la información contenida en la línea epipolar, como:

$$\Phi_k(u, d) = \begin{cases} \frac{e_{long}}{d_{dim}} & \text{si } e_{long} < d_{dim} \\ \text{nulo} & \text{si no} \end{cases} \quad (3.10)$$

### 3.2.7. Regularización

La regularización propuesta en [90] requiere una gran cantidad de cómputo, como se indica en [98]. Así, [90] requiere movimientos de cámara suaves, de manera de dar tiempo al regularizador terminar su procesamiento. Para permitir movimientos de cámara más rápidos, se eligió utilizar el filtro llamado “Guided Filter” desarrollado en [98]. Este tipo de filtros utilizan una imagen para guiar la acción del filtrado. En el contexto de estimación de profundidad mediante volumen de costo, se filtra cada conjunto  $F_{u,d}$  perteneciente a un mismo  $d$ . Se utiliza como guía el gradiente de la imagen de referencia  $I_{kf}$ , de manera de suavizar las zonas en el volumen de costo donde la información es menos confiable. Se obtiene una regularización de calidad comparable con la obtenida en regularizadores globales, pero con una fracción del poder de cómputo requerido.

### 3.2.8. Adaptación Dinámica de Rango

Las profundidades mínimas y máximas  $d_{min}$   $d_{max}$  representables dentro del volumen de costo deben tener la magnitud necesaria para abarcar las dimensiones reales de la escena visible. Si, por ejemplo, se encuentran en la escena objetos a distancias menores a  $d_{min}$ , sus profundidades no son representables dentro del volumen de costo y sus estimaciones asociadas serán erróneas. En [90] se eligen valores  $d_{min}$   $d_{max}$  fijos a-priori, de manera de poder representar las profundidades que comúnmente se encontrarán en la mayoría de las escenas. Sin embargo, sería ventajoso poder adaptar los valores  $d_{min}$   $d_{max}$  de manera de ajustarse a la escena actual particular.

Por esa razón, se realizó un algoritmo de adaptación de  $d_{min}$   $d_{max}$  para ajustar dinámicamente las profundidades representables dentro del volumen de costo. Una vez completa la estimación de profundidad  $\eta_e(u)$  estimada se computa:

$$n_{max} = \text{num } \eta_e(u) \text{ con } \eta_e(u) \in [d_{max}, d_{max-5}] \quad (3.11)$$

donde  $[d_{max}, d_{max-5}]$  son las últimas 5 profundidades  $d$  representables en el volumen de costo.

De esta manera, el algoritmo de ajuste desarrollado responde a:

$$d_{max} = d_{max} + \beta(n_{max} - \alpha) \quad (3.12)$$

donde se ajusta  $\alpha$  y  $\beta$  para permitir un determinado número de estimaciones fuera de rango y ajustar la velocidad de adaptación, respectivamente. Un algoritmo recíproco se utiliza para ajustar  $d_{min}$ .

### 3.2.9. Integración en un Sistema SLAM Completo

Para realizar las pertinentes evaluaciones al algoritmo de estimación de profundidad presentado en este capítulo, se completó su integración dentro de un sistema SLAM completo, de manera de obtener funcionalidad referida al cierre de lazo y el manejo del mapa global. Se eligió para este menester utilizar el sistema de SLAM visual monocular LSD-SLAM [14], por ser un sistema denso-directo cuyo código fuente es distribuido libremente. Se reemplazó entonces el sistema de estimación de profundidad presente en LSD-SLAM por el propuesto en este capítulo.

### 3.3. Evaluación del Método Propuesto

Se realizará la comparación del desempeño tomando como referencia dos sistemas de localización y mapeo directos-densos, reconocidos dentro del estado del arte. En primer lugar, se utilizará el sistema LSD-SLAM [14]. En segundo lugar, se utilizará un sistema basado en DTAM [90]. El sistema DTAM, dado su implementación de la optimización de la función de costo, requiere un tiempo considerable para generar la estimación de profundidad regularizada. Así, en el momento que se requiere una nueva imagen de referencia, se debe esperar sin realizar movimiento de cámara bruscos hasta que la profundidad de la imagen de referencia actual se termina de procesar. Por ese motivo, no está preparado para experimentaciones en datasets con movimientos arbitrarios. Para poder realizar una comparación lo más justa posible, se eliminó el sistema original de optimización de función de costo, y se lo reemplazó por el propuesto en la sección 3.2.7, con el cual se puede obtener una profundidad regularizada rápidamente. A su vez, como el dataset utilizado presenta muchas zonas borrosas, donde no es posible realizar estimaciones de profundidad correctamente, se implementó un sistema de rechazo de profundidades incorrectas como el descrito en la sección 3.2.3.

Para esta experimentación, el sistema LSD-SLAM utiliza sus parámetros por defecto, con la resolución de imagen recomendada por los autores de  $640 \times 480$ . Los sistemas DTAM y el desarrollado en este capítulo, por el contrario, estiman la profundidad en imágenes reducidas en tamaño, de  $320 \times 240$ , y el volumen de costo se configura de manera de obtener  $d_{dim} = 32$

Las experimentaciones se realizaron en base a un conjunto de datos de campo reales (dataset), que muestra secuencias de imágenes capturadas en las costas de Japón. Las secuencias de este dataset son cortas y no contienen revisitas, lo que imposibilita la realización de cierres de lazo, permitiendo así realizar comparaciones odométricas exclusivamente. Las imágenes capturadas por la cámara muestran zonas rocosas, tanto planas como con relieves angulosos y suaves, en conjunto con cuevas y túneles. De esta manera el dataset presenta distintas secuencias, con movimientos de cámara arbitrarios en un ambiente desafiante.

La primera secuencia consiste en un movimiento de cámara hacia adelante, la cual luego pasa por una zona rocosa en forma de cueva, para seguidamente encontrarse en un espacio abierto nuevamente.

La figura 3.8 ilustra la estimación de profundidad en esta secuencia para cinco cuadros a lo largo de la experimentación. En esta figura y las siguientes, las columnas representan instantes diferentes dentro de la experimentación, ordenadas temporalmente. La primera fila se destina para ilustrar las imágenes crudas capturadas por la cámara. La segunda fila muestra resultados de la estimación de profundidad obtenidos del sistema LSD-SLAM. La tercera fila ilustra los resultados obtenidos mediante el sistema basado en DTAM. Por último, la cuarta fila muestra resultados obtenidos mediante el sistema propuesto en este capítulo. En todas las estimaciones, la profundidad se codifica mediante colores, correspondiendo el rojo a los píxeles cercanos a la cámara, los verdes a distancias medias, y los tonos azulados a distancias lejanas con respecto a la cámara. Aquellos píxeles que no posean estimación de profundidad se los colorea en escala de grises utilizando el brillo que fue observado por la cámara.

Se puede apreciar en la figura 3.8 como LSD-SLAM no logra estimar correctamente las profundidades al momento de salir de la cueva. Una vez afuera, esas estimaciones erróneas persisten, debido a la naturaleza paramétrica de dichas estimaciones. DTAM y el método propuesto logran estimar las profundidades dentro

y fuera de la cueva correctamente. Sin embargo DTAM contiene significativamente más ruido en las mediciones de profundidad, muchas de las cuales son marcadas como erróneas.

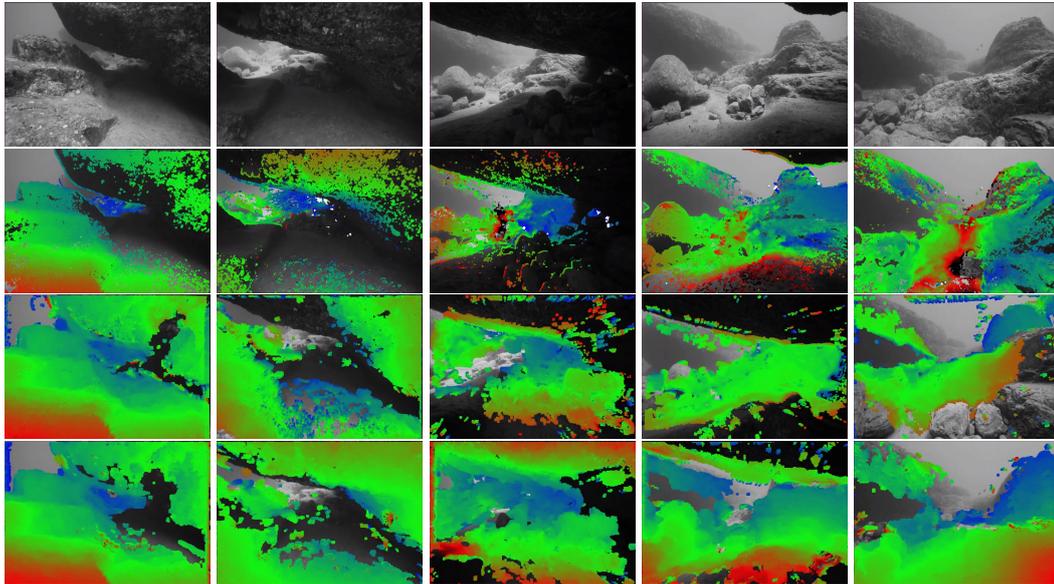


FIGURA 3.8: Primer secuencia. En la primera fila: imágenes crudas. En la segunda fila: estimación de profundidad mediante LSD-SLAM. Tercer fila: estimación de profundidad mediante DTAM. Cuarta fila: estimación mediante el sistema propuesto

El movimiento de la cámara estimado por el sistema propuesto se puede observar en la figura 3.9.

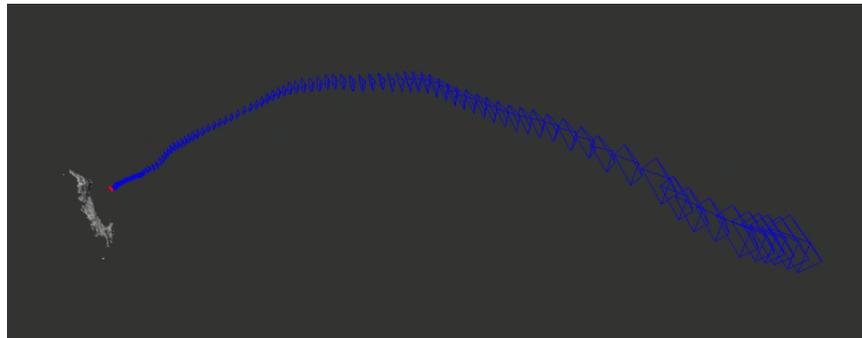


FIGURA 3.9: Trayectoria de la cámara realizada durante la primer secuencia, obtenida utilizando el método propuesto.

La segunda secuencia 3.10 muestra en primer lugar un movimiento de la cámara descendente, para luego estabilizarse a nivel al lado de una formación rocosa. Nuevamente, se puede ver como LSD-SLAM contiene numerosos píxeles con profundidades incorrectamente estimadas. DTAM, en este caso, no logra estimar la profundidad de numerosos píxeles. Esto se debe principalmente al rango de escalas encontrado en la secuencia de imágenes, donde se encuentran grandes profundidades en el margen izquierdo, conjunto con una formación rocosa cercana en el margen derecho. El método propuesto, en este caso, puede estimar las profundidades tanto cercanas como lejanas correctamente. Sin embargo, se puede observar estimaciones

erróneas en el borde superior de la imagen inferior derecha de la figura 3.10. Esto se debe a un efecto borde producto del reposicionamiento del volumen de costo.

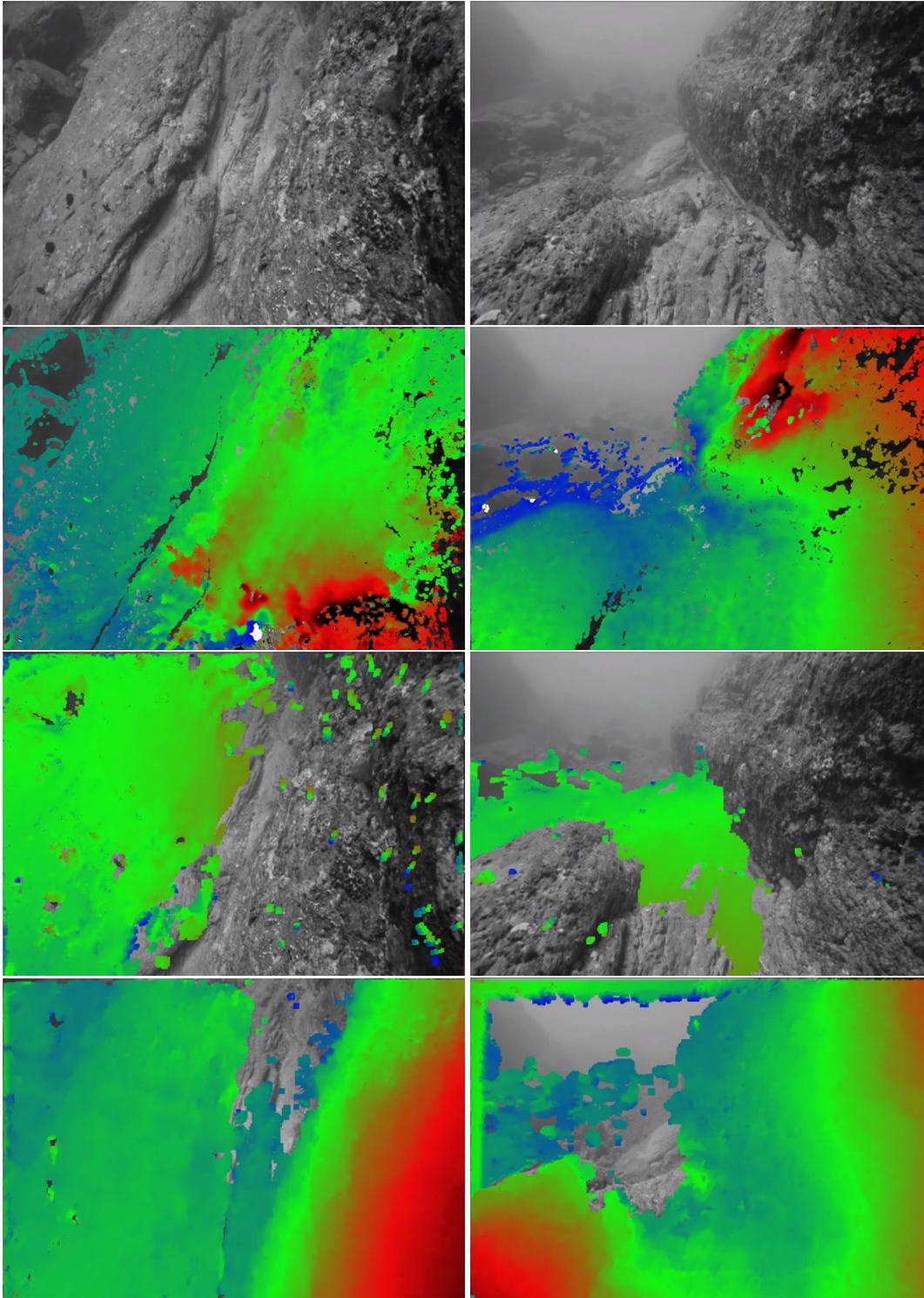


FIGURA 3.10: Segunda secuencia. En la primera fila: imágenes crudas. En la segunda fila: estimación de profundidad mediante LSD-SLAM. Tercer fila: estimación de profundidad mediante DTAM. Cuarta fila: estimación mediante el sistema propuesto

La tercer secuencia 3.11 ilustra una cámara mirando hacia abajo, mientras realiza un movimiento hacia adelante, para luego rotar de manera de culminar mirando

hacia la dirección del movimiento. En esta secuencia, como la cámara mira hacia abajo, no se observan zonas borrosas en el horizonte. De esta manera LSD-SLAM puede estimar correctamente la profundidad en la mayoría de los píxeles, con un alto grado de precisión. DTAM estima correctamente la profundidad de la escena, si bien con mayor ruido y con menor precisión. Esto último se debe a la cuantización del volumen de costo. El sistema propuesto estima la profundidad con un bajo nivel de ruido, aunque con menos precisión que LSD-SLAM, debido también a la baja cantidad de profundidades diferentes representables en el volumen de costo.

Un mapa denso estimado por el sistema propuesto, en conjunto con la trayectoria realizada por la cámara se puede observar en la figura 3.12.

La cuarta secuencia 3.13 ilustra una cámara con movimiento hacia adelante, observando un ambiente espacioso al principio, para luego terminar en un espacio muy angosto. Se puede ver que LSD-SLAM estima las profundidades correctamente, hasta el momento de adentrarse en la cueva, donde las estimaciones se tornan claramente incorrectas. DTAM también logra estimar correctamente la profundidad antes de acercarse a la cueva. El sistema propuesto logra tanto estimar profundidad en el ambiente espacioso como en el angosto, con un bajo nivel de ruido y una buena densidad de estimaciones dentro de la cueva. Nótese en la imagen inferior derecha como los límites de la cueva son correctamente estimados, lo que permitiría a un vehículo autónomo navegar sin colisiones.

Un mapa denso de la zona conjunto con la trayectoria de la cámara se observan en la figura 3.14.

La última secuencia muestra un movimiento hacia adelante de la cámara, realizado sobre una formación rocosa mayormente planar. La figura 3.15 ilustra los resultados. En la zona de la imagen que contiene poco paralaje, es decir la zona superior, LSD-SLAM estima incorrectamente la profundidad para numerosos píxeles. Dada su naturaleza paramétrica, estas estimaciones erróneas se arrastran, culminando la secuencia con profundidades claramente incorrectas, como las que se ilustran en la segunda fila última columna. DTAM, por el contrario, logra estimar las profundidades correctamente, aún con estimaciones altamente ruidosas. El sistema propuesto logra estimar correctamente las profundidades, en conjunto de presentar un nivel de ruido pequeño.

La profundidad y trayectoria de la cámara estimada por el sistema propuesto para esta secuencia se ilustran en la figura 3.16.

A través de estas experimentaciones se pudo comprobar el excelente desempeño del sistema propuesto ante diversas escenas subacuáticas desafiantes, obteniendo estimaciones densas con bajos nivel de ruido, superando a otros sistemas directos presentes en la literatura. En el capítulo 6 se realizan comprobaciones del presente sistema en un ambiente subacuático controlado perturbado por ondas cáusticas.

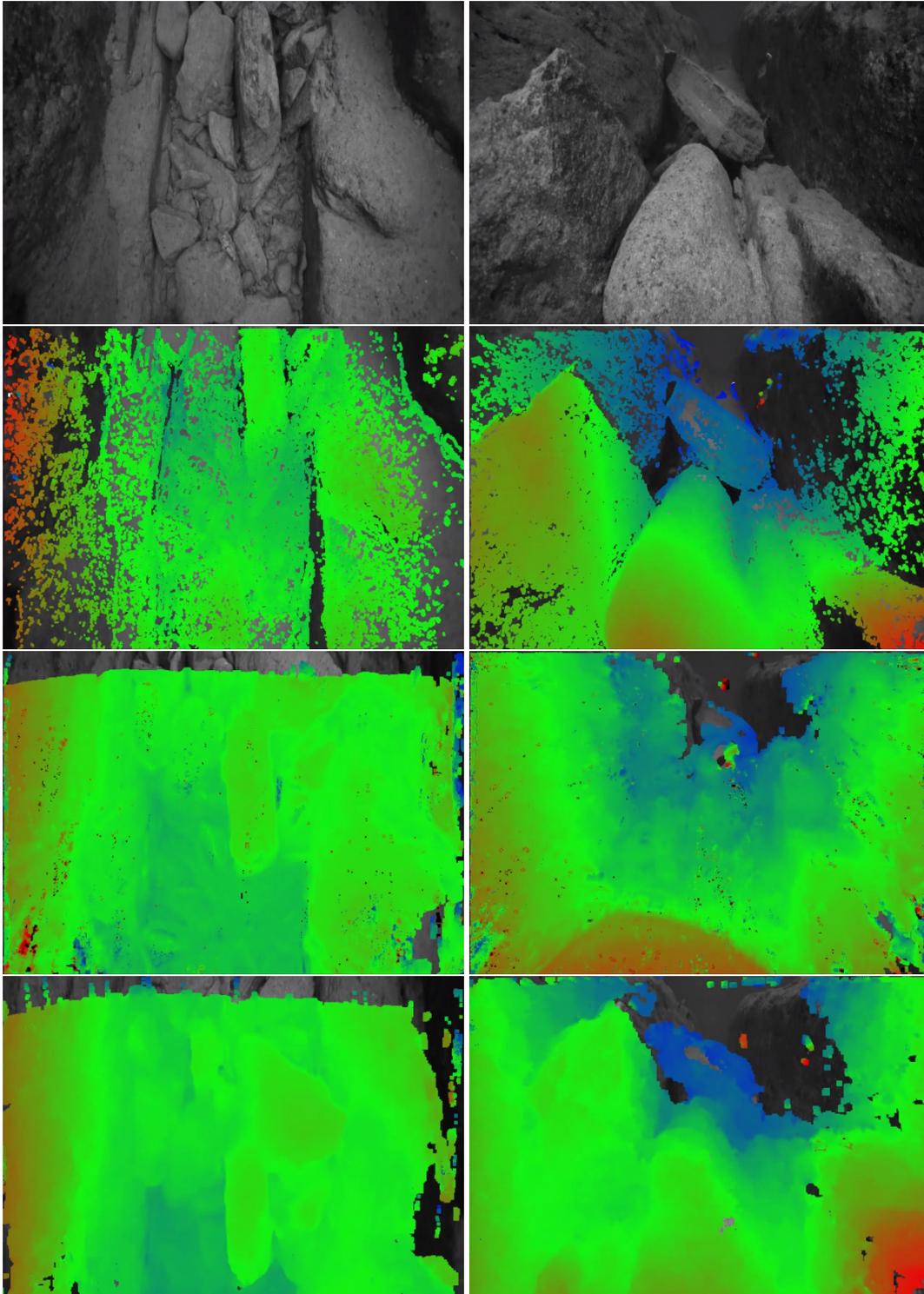


FIGURA 3.11: Tercer secuencia. En la primera fila: imágenes crudas. En la segunda fila: estimación de profundidad mediante LSD-SLAM. Tercer fila: estimación de profundidad mediante DTAM. Cuarta fila: estimación mediante el sistema propuesto

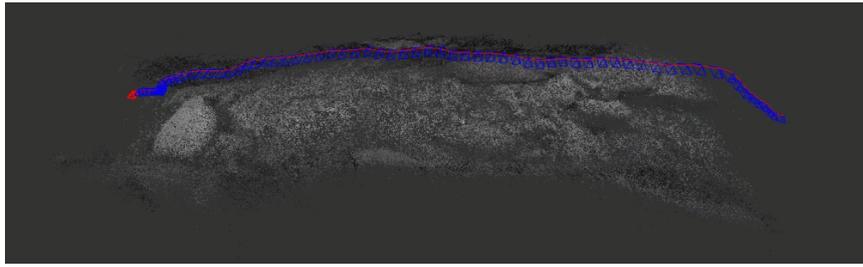


FIGURA 3.12: Trayectoria de la cámara realizada durante la tercer secuencia, obtenida utilizando el método propuesto.

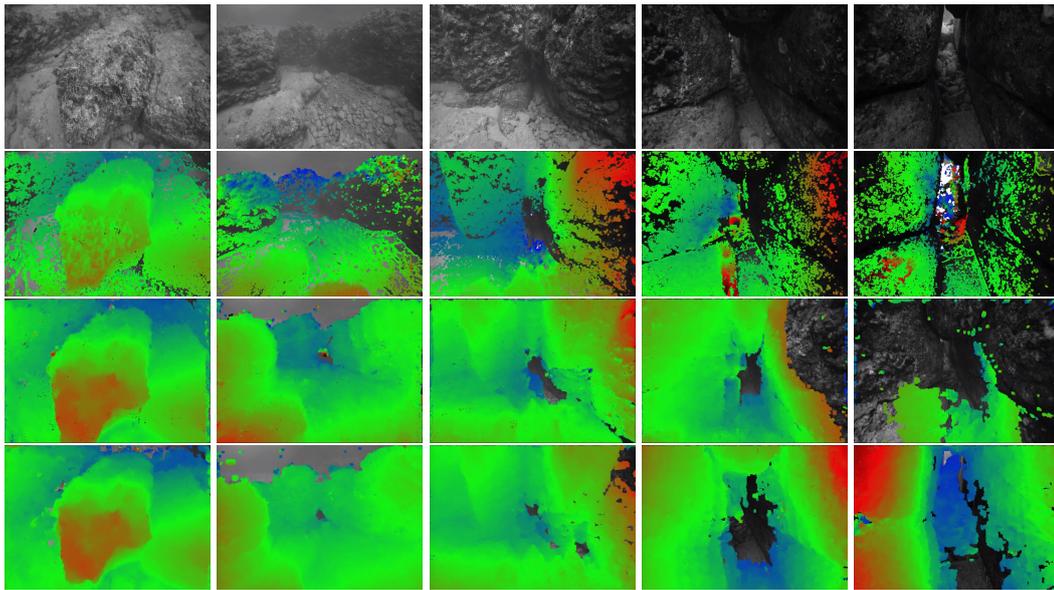


FIGURA 3.13: Cuarta secuencia. En la primera fila: imágenes crudas. En la segunda fila: estimación de profundidad mediante LSD-SLAM. Tercer fila: estimación de profundidad mediante DTAM. Cuarta fila: estimación mediante el sistema propuesto

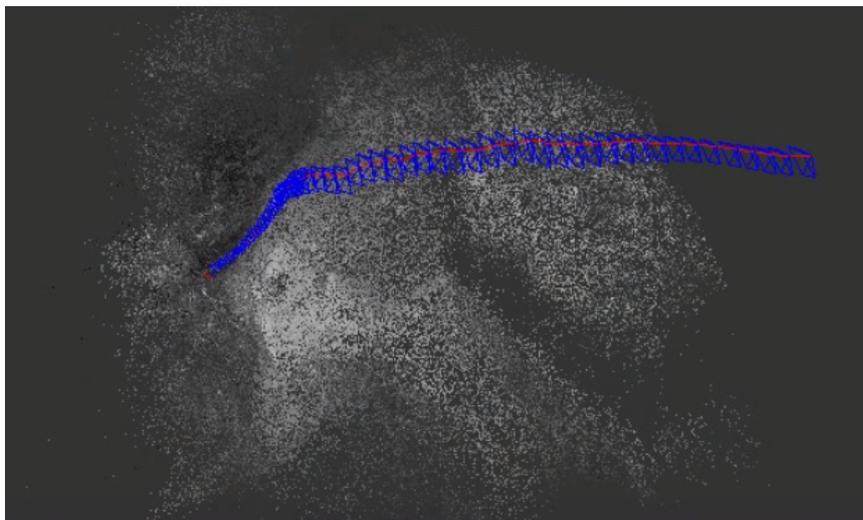


FIGURA 3.14: Trayectoria de la cámara realizada durante la cuarta secuencia, obtenida utilizando el método propuesto.

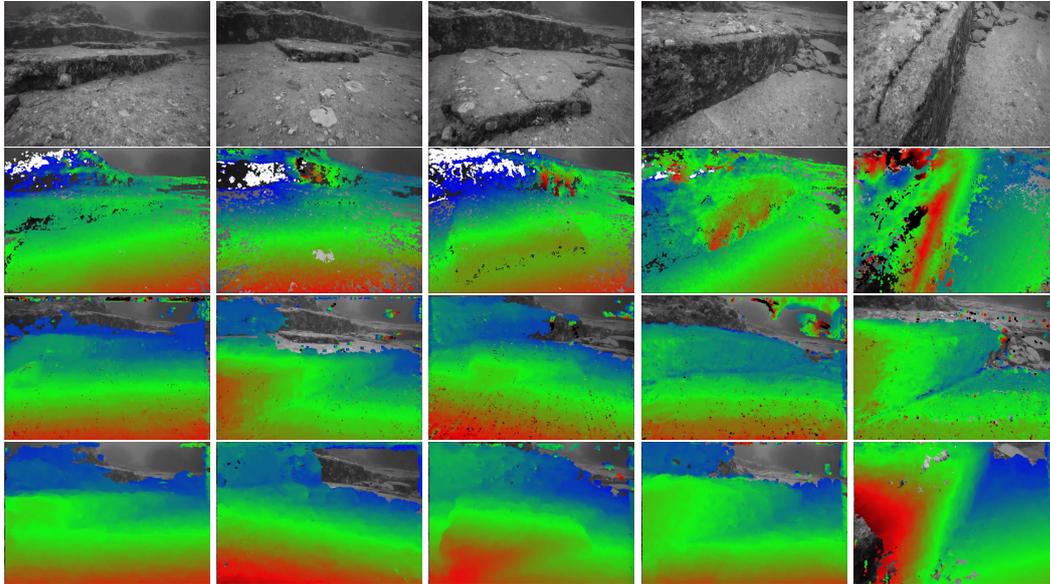


FIGURA 3.15: Quinta secuencia. En la primera fila: imágenes crudas. En la segunda fila: estimación de profundidad mediante LSD-SLAM. Tercer fila: estimación de profundidad mediante DTAM. Cuarta fila: estimación mediante el sistema propuesto

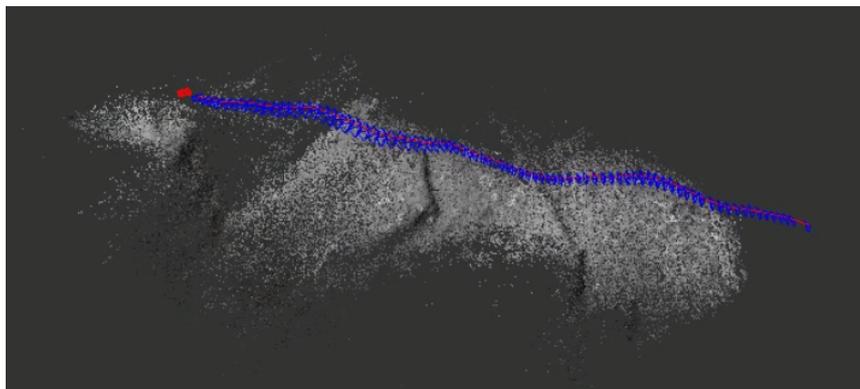


FIGURA 3.16: Trayectoria de la cámara realizada durante la quinta secuencia, obtenida utilizando el método propuesto.



## Capítulo 4

# Filtro de Onda Cáustica

### 4.1. Introducción

Los sistemas de odometría visual (véase el capítulo 3) se fundamentan en la hipótesis de constancia de brillo. Esta hipótesis postula que un mismo punto físico será registrado en las imágenes con un brillo aproximadamente constante, aún cuando cambiase la posición desde donde se visualiza la escena y el momento en el que se captura la imagen. Así, esta afirmación básica fundamenta gran parte de los algoritmos utilizados para realizar odometría visual (particularmente en la formulación de los métodos directos). Como ejemplo se puede mencionar el algoritmo de estimación de profundidad utilizando la búsqueda dentro de la línea epipolar del mínimo error fotométrico.

Las características particulares encontradas en las distintas escenas pueden deteriorar la veracidad de dicha hipótesis. Reflexión especular o partículas en suspensión con reflexiones hacia adelante y atrás (scattering y backscattering) producen brillos dependientes de la postura de la cámara. Los ambientes subacuáticos de baja profundidad contienen las llamadas ondas cáusticas sobre el fondo, una importante violación de la constancia de brillo. El efecto de sombras de nubes en movimiento sobre la superficie terrestre se asemeja en su mecanismo de perturbación visual, sin embargo el cambio de iluminación producido por las sombras es más uniforme y mucho más lento que las perturbaciones luminosas sobre el fondo submarino. Una ilustración de este fenómeno se aprecia en la figura 4.1.

La luz solar, al atravesar el aire y penetrar en la columna de agua, sufre una difracción en su trayectoria, que depende del oleaje presente en la superficie del agua y de la diferencia de densidades entre los dos medios. Dado que la superficie del agua está constantemente sometida a movimientos aleatorios causados principalmente por el viento, la difracción de la luz produce un patrón de iluminación en el fondo marino complejo y variante



FIGURA 4.1: Ondas cáusticas sobre el fondo marino

FIGURA 4.1: Ondas cáusticas sobre el fondo marino complejo y variante

tanto en espacio como en el tiempo, dando como resultado que la luz se concentra en alguna zonas y sea menos densa en otras.

Dada la magnitud que puede alcanzar este fenómeno en determinadas circunstancias (como son baja profundidad y viento fuertes), cabe esperar que los sistemas de odometría visual vean su desempeño altamente deteriorado bajo estas condiciones. En consecuencia, se requiere encontrar soluciones de manera de poder utilizar los sistemas de odometría visual en estos ambientes submarinos particulares.

En la literatura, es posible encontrar trabajos como [101], donde se utiliza un filtro promediador sencillo para atenuar el cambio de brillo en el fondo de una pileta. Este filtro tiene un desempeño modesto y no está claro cómo extenderlo sobre superficies no planares. En [102], se realiza un atenuador más sofisticado, combinando trabajos en filtros de sombra de nubes [103] en una secuencia de imágenes registrable de forma planar. Los resultados presentados son valiosos, sin embargo, en ambientes subacuáticos la baja visibilidad puede obligar al submarino guiado por visión a navegar cerca del suelo, adquiriendo así la escena una apariencia marcadamente tridimensional. En tales circunstancias, el desempeño de dicho filtro puede deteriorarse considerablemente.

Por otro lado, los autores en [104] utilizan un sistema de odometría visual binocular, y en ese contexto, la onda cústica deja de ser una perturbación y es posible utilizar la aleatoriedad de la misma para mejorar la precisión de la estimación densa de la profundidad. Una vez estimado correctamente el mapa de la escena en conjunto con la postura de la cámara, se puede proceder a eliminar el efecto de la onda cústica fácilmente. Claramente este proceso no es aplicable en el caso de cámaras monoculares.

En este capítulo se presentan algoritmos novedosos de filtrado de onda cústica para ser utilizados en sistemas de odometría visual monoculares. Los principales aportes constituyen la generación de algoritmos robustos que presentan buen funcionamiento en ambientes con estructura tridimensional y en el filtrado de hasta ondas muy intensas. Resultados parciales del filtro propuesto fueron presentados en [61], [65], [66], [68].

## 4.2. Algoritmo Propuesto

La heurística fundamental del filtrado propuesto se basa en separar zonas afectadas en la imagen que tienen brillo alterado debido a la perturbación, de las zonas que no fueron afectadas. Acto seguido, se pueden reparar las intensidades de los píxeles afectados y así la imagen quedaría libre de los característicos flecos brillantes de las ondas.

Para realizar esta separación solo hay un camino que consiste en aislar la onda cústica en una imagen estimada de la misma. Dado que la corrección de las intensidades afectadas debe ser en tiempo real, la forma más adecuada de estimar la onda cústica es por comparación instantánea entre la imagen perturbada con una imagen filtrada resultante de una predicción de un paso en adelante.

Otro aspecto de la heurística es que el resultado de esta comparación de imágenes puede interpretarse como una imagen de error que puede ser transformada y realimentada en un lazo de control.

De acuerdo a esta heurística, se diseña el filtro de onda cústica basándose en una estructura realimentada (véase la figura 4.2), en donde la entrada es una secuencia de imágenes capturadas sobre una escena subacuática perturbada por ondas cústicas.

La escena se observa mediante una cámara monocular, la cual se mueve sobre una escenario tridimensional acorde a las órdenes impartidas por un sistema de guía.

En la parte superior de la figura 4.2 se observan dos bloques interconectados en donde se decide si el filtro de onda cáustica debe ser aplicado a la secuencia de imagen o no. En caso de que la perturbación cáustica sea insignificante, un histograma de brillo de la imagen de entrada revelará un porcentaje muy bajo de píxeles de color claro. Por ello, umbralizando el histograma, el filtro podrá desconectarse o conectarse según la intensidad de la perturbación relativa a este umbral.

Se comienza con la descripción del filtro en su trayecto directo (véase los bloques en la fila superior en la figura 4.2) y trayecto realimentado (los bloques de la fila inferior) independientemente.

Un cuadro con brillo  $I(x, y, t)$  capturado por la cámara es comparado con un estimación *a-priori* de la imagen filtrada  $\hat{I}(x, y, t)$ . La diferencia resulta en la función  $\Delta I(x, y, t)$  que contiene (con cierta precisión) una primera estimación de la onda cáustica observada en el instante  $t$ . También  $\Delta I$  contiene usualmente pequeñas perturbaciones, algunas relacionadas con el efecto de borrosidad sobre el suelo marino y otras ocasionadas por ruido de alta frecuencia originado por imprecisiones en el procesamiento realizado por los distintos bloques del filtro.

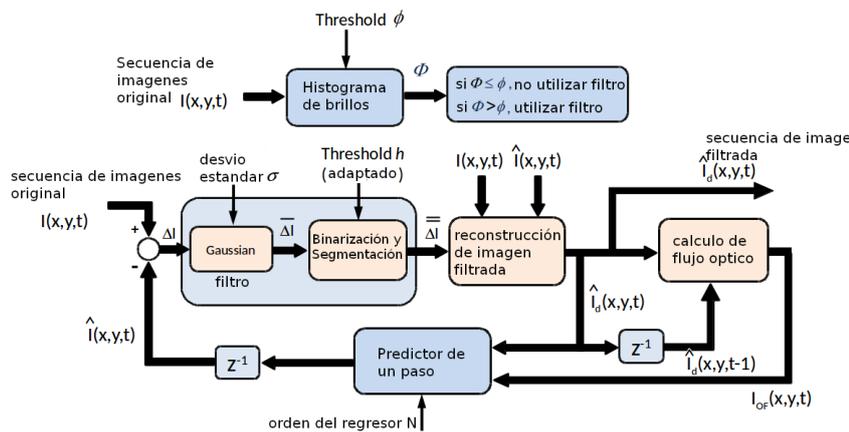


FIGURA 4.2: Algoritmo de filtrado básico basado en una estructura realimentada.

Mientras que el ruido ocasionado por la borrosidad es posible en general reducirlo por una promediación temporal, que se lleva a cabo en el lazo de realimentación, la otra clase de ruido espacial procedente de operaciones previas sobre  $I$  es filtrado empleando un filtro Gaussiano espacial. La desviación estándar del filtro gaussiano, denominada  $\sigma$ , es responsable de la nitidez de la imagen filtrada final.

Luego, la imagen filtrada  $\overline{\Delta I}(x, y, t)$  es segmentada por medio de una binarización, eligiendo para este propósito un umbral de segmentación  $h$ . De esta manera, las áreas que contengan los puntos más brillantes se supone pertenecientes a áreas afectadas por ondas cáusticas y se verán representadas en la imagen segmentada  $\overline{\Delta I}(x, y, t)$ . La unión de estas áreas es referida como  $S_1$ . Los píxeles no afectados en  $\overline{\Delta I}$  representan regiones en el suelo marino que están mucho menos perturbadas por ondas cáusticas. La unión de este conjunto de píxeles es el complemento de  $S_1$ , referido como  $\overline{S}_1$ . De hecho, los contornos de las ondas cáusticas no serán nítidos, sino suaves. Así, la selección de  $h$  es de suma importancia para la correcta segmentación de las zonas con y sin onda cáustica. Encontrar su valor correcto no es una tarea trivial.

Esta estimación a-posteriori de la imagen filtrada  $\hat{I}_d(x, y, t)$  resulta de la siguiente ecuación de reconstrucción:

$$\begin{aligned}\hat{I}_d(x, y, t) &= I_{S_1}(x, y, t) + I_{\bar{S}_1}(x, y, t), \text{ donde} \\ I_{S_1}(x, y, t) &= \hat{I}(x, y, t) \text{ para } (x, y) \in S_1 \text{ y} \\ I_{S_1}(x, y, t) &= 0 \text{ para } (x, y) \in \bar{S}_1 \\ I_{\bar{S}_1}(x, y, t) &= I(x, y, t) \text{ para } (x, y) \in \bar{S}_1 \text{ y} \\ I_{\bar{S}_1}(x, y, t) &= 0 \text{ para } (x, y) \in S_1\end{aligned}$$

conservando así la imagen reconstruida  $\hat{I}_d$  la nitidez de la imagen original en los píxeles correspondientes a  $I_{\bar{S}_1}$  donde no se detecto onda cústica. Por el contrario, los lugares donde se estimaron ondas cústicas  $I_{S_1}$  son completados con estimaciones a-priori  $\hat{I}(x, y, t)$ .

Con ayuda de la figura 4.2, los pasos y operaciones aplicados sobre el lazo de realimentación pueden ser detallados más apropiadamente.

Los bloques en este lazo operan sobre las últimas  $N$  estimaciones a-posteriori  $\hat{I}_d$  y sus correspondientes flujos ópticos  $I_{OFF}$ . Con la imagen reconstruida  $\hat{I}_d(x, y, t)$  y la anterior  $\hat{I}_d(x, y, t - 1)$ , se calcula el flujo óptico en el instante  $t$ , referido como  $I_{OFF}$ . En este trabajo la estimación del flujo óptico se realiza mediante el algoritmo desarrollado por Farnebäck [105].

La tarea principal en el lazo de realimentación es la estimación del trazo de cada píxel sobre los  $N$  pasos, cada uno correspondiendo a su respectivo punto físico en el ambiente. Este enlazado de puntos  $[x, y]$  empieza en cada imagen en el instante  $(t - N + 1)$  y se concatena cada punto  $[x, y]$  en los  $(N - 1)$  pasos en las imágenes previas hasta el tiempo  $t$ . La base de este enlazado de píxeles son los  $N$  flujos ópticos  $I_{OFF}(x, y, t - i)$ , así cada píxel es desplazado de imagen en imagen siguiendo la dirección del flujo óptico sobre los  $(N - 1)$  eslabones.

Una vez que todos los lazos son computados hasta el instante  $t$ , se procese a predecir un eslabón más por cada píxel  $[x, y]$ . Este es el predictor de un paso  $[\hat{x}(t + 1), \hat{y}(t + 1)]$

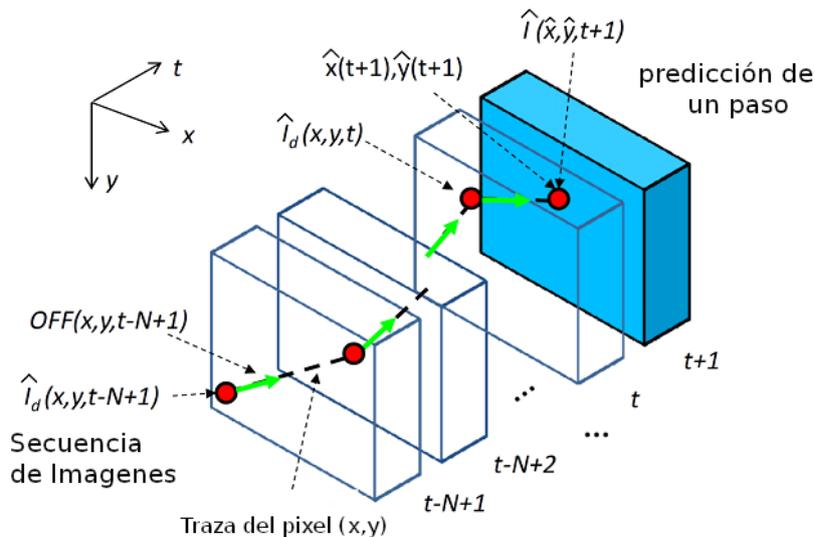


FIGURA 4.3: Predicción de posiciones de los píxeles y su brillo

Se obtiene luego una estimación *a-priori* del brillo de cada píxel  $\hat{I}$  para el próximo instante  $t + 1$ , basándose en las estimaciones previas *a-posteriori*  $\hat{I}(x, y, t - i)$  con  $i = 0, \dots, N - 1$ . Para este propósito se emplea un simple proceso de promediación a lo largo del trazo de cada píxel en los  $N$  pasos. Cabe aclarar que es posible utilizar métodos más sofisticados, como promediación ponderada. La imagen  $\hat{I}$  estimada *a-priori* sera utilizada solo en el próximo paso como se ilustra en el bloque de retardo de un paso al final del lazo de realimentación.

Finalmente la salida del filtro es la estimación *a-posteriori*  $\hat{I}_d(x, y, t)$ .

#### 4.2.1. Adaptación del Umbral de Binarización

La estructura del filtro propuesto tiene propiedades de estabilidad que son las mismas otorgadas por el sistema de control que lo contiene. Sin embargo, dados los componentes complejos que constituyen el filtro, en particular de flujo óptico, binarización y predicción, solo es posible sacar conclusiones cualitativas sustentadas por la simulación de numerosos casos de estudio.

En términos generales la inestabilidad sobreviene cuando un valor muy bajo del umbral de segmentación produce un flujo óptico incorrecto en las áreas de flecos no filtrados. Este error se realimenta potenciando su incremento en el lazo. Dada las características diferenciales espaciales del flujo óptico y del predictor, es de esperar que el error se propague como una mancha de nivel de gris indefinido, que aumenta paulatinamente su área dentro de la imagen. Si la elección del umbral  $h$  es muy desacertada, es posible que la imagen *a-posteriori*  $I_d$  se deteriore completamente con el correr del tiempo. De lo contrario, con un umbral  $h$  muy alto los errores en el lazo son más acotados porque la binarización abarca zonas no afectadas de la imagen que solo incorporan errores de predicción pequeños sin afectar mayormente el flujo óptico en dichas zonas.

El valor apropiado del umbral corresponde a un valor óptimo. Con este umbral, el filtro mantiene un buen desempeño de manera estable. Dado que el valor óptimo es desconocido, se intenta en esta tesis diseñar un complemento del algoritmo para la auto-sintonización de  $h$  y adaptación permanente de éste a la intensidad de la perturbación.

Hay un síntoma claro en la secuencia *a-posteriori*  $\hat{I}_d$  cuando el valor del parámetro  $h$  es sobrevaluado, observándose que las áreas circundantes a la onda cáustica serán subestimadas. Así, aparecerán unos "halos" alrededor de  $\hat{I}_d$ , aproximadamente en los bordes de la onda cáustica. Por el contrario, si  $h$  es subvalorado, los contornos estimados de la onda cáustica serán más grandes que su tamaño real, y como consecuencia se reducirá el área de  $\hat{I}$  donde el brillo original de  $I$  es heredado.

De acuerdo con el síntoma descrito anteriormente, debe existir un valor de  $h$  tal que el plano de corte de la función de intensidad  $I(x, y, t)$  produce una segmentación precisa de las zonas alrededor de la onda cáustica. En relación con la dinámica compleja de la superficie del agua y su efecto de lente que causa en la luz que la atraviesa, los bordes de la onda cáustica son típicamente borrosos. Esto significa que la transición de brillo desde la onda cáustica a sus inmediaciones cambia suavemente. Así, se debe realizar un compromiso en relación a la identificación de los bordes suaves.

De forma de construir una función error y de costo acorde al problema, se comenzará con la imagen  $\hat{I}_d(x, y, t; h)$ , la cual esta parametrizada con respecto a  $h$ .

Aplicando operaciones morfológicas de dilatación, erosión y substracción en  $\overline{\Delta I}$  con un elemento estructurante pequeño (por ejemplo  $3 \times 3$ ), se definirá un conjunto

de contornos  $S_i$  para un valor particular de segmentación  $h_0$ . El ancho de  $S_i$  es un píxel. Comúnmente ocurrirá que los contornos estarán anidados.

En primer lugar, se definen bordes de un píxel de ancho para cada  $S_i$ , referido como  $B_{i\_S_i}$ , y un segundo borde fuera de  $S_i$ , denominado  $B_{o\_S_i}$ . Luego, se cumple que  $S_i \cap B_{i\_S_i} \cap B_{o\_S_i} = \phi$  (el conjunto nulo).

De esta manera, el error para  $S_i$  se puede formular, en conjunto con una función de costo en la norma Euclidiana:

$$e_{S_i}(h) = \sum_{i=1}^{n_i} \hat{I}_{d_i}(x_i, y_i, t) - \sum_{i=1}^{n_o} \hat{I}_{d_i}(x_o, y_o, t) \quad (4.1)$$

con los píxeles  $(x_i, y_i) \in B_{i\_S_i}$  y  $(x_o, y_o) \in B_{o\_S_i}$

$$V(h) = \frac{1}{n_s} \sqrt{\sum_{i=1}^{n_s} n_s e_{S_i}^2(h)} \quad (4.2)$$

donde  $n_i$  y  $n_o$  son la cantidad de píxeles de los bordes dentro y fuera de cada  $S_i$ ,  $\hat{I}_{d_i}$  es el brillo de cada píxel perteneciente al borde y  $n_s$  el numero de contornos.

### Optimización

Se diseñó el bloque de optimización para  $h$  de acuerdo al siguiente razonamiento. Cuando se comienza con valores altos de  $h$ , es posible observar halos brillantes en  $\hat{I}_d$ , los cuales gradualmente son eliminados a medida que el valor  $h$  es disminuido iterativamente. En una determinada iteración durante el proceso, los halos desaparecerán. En cuanto esto ocurra, se considera que se encontró el valor óptimo de  $h$ .

Más detalladamente, se comenzará el proceso mediante un valor de  $h_0$  alto, pero que permita detectar por lo menos un halo. Con este valor inicial se opera morfológicamente sobre  $\overline{\Delta I}$  para identificar todos sus contornos. Luego, se evalúa cada  $e_{S_i}(h_0)$  y  $V(h_0)$ . Claramente, el valor alto de contraste entre las zonas interiores y exteriores de cada  $S_i$  origina un valor relativamente grande de  $V(h_0)$ . Se prosigue iterativamente a disminuir  $h$  y evaluar  $V(h)$ . De esta manera se generará una secuencia de valores  $V(h)$  para cada  $h$ . Para una determinada iteración ocurrirá que la velocidad de disminución de  $V(h)$  bajara abruptamente, y desde este valor no se observan cambios significativos en  $V$ . Cuando se alcanza esta situación, el algoritmo finaliza y se provee el valor más reciente de  $h$ .

### 4.3. Evaluación

En primera instancia, se evalúa el desempeño del algoritmo propuesto inspeccionando visualmente la calidad de la imagen filtrada en el vídeo. A continuación se presentan resultados obtenidos con esta metodología. En cuanto a la adecuación del filtro propuesto para mejorar la calidad de las estimaciones odométricas, tarea que se postuló como la verdadera finalidad del filtro diseñado, pueden verse resultados en el capítulo 6.

La figura 4.4 concentra resultados del filtro propuesto aplicado a diferentes secuencias de imágenes reales. A la izquierda de la figura se aprecia la imagen cruda, mientras que a la derecha se presenta la imagen final filtrada. La primera y segunda fila ilustran imágenes capturadas por un vehículo subacuático mientras realizaba una navegación dentro de una piletta. En la imagen capturada se logra ver que en el

fondo de dicha piletta se hallaba un tubo, marcado con un patrón de franjas negras y blancas intercaladas. Se puede observar en la imagen izquierda cómo la onda cáustica se mimetiza con las franjas blancas pertenecientes al tubo, haciendo dificultosa la identificación visual. La imagen filtrada a la derecha presenta una onda cáustica filtrada casi por completo, haciendo la tarea de la detección del tubo mucho más sencilla, mientras que al mismo tiempo se mantuvo la nitidez y los detalles contenidos en el tubo.

En la tercer y cuarta fila se visualizan resultados sobre una secuencia de imágenes capturadas en un ambiente submarino de baja profundidad. La escena presenta fuertes perturbaciones cáusticas. Las rocas en el suelo hicieron al ambiente marcadamente tridimensional. Un detalle que no se puede apreciar en esta descripción particular es el movimiento de la cámara, que para estas experimentaciones particulares es de alta velocidad, con numerosos y abruptos cambios de dirección. En ambas imágenes de la derecha se puede apreciar la notable atenuación de la onda cáustica.

Para la quinta y sexta fila, las imágenes fueron capturadas en aguas más profundas, por lo que la onda cáustica que perturba la escena no es de la magnitud de los ejemplos anteriores. Sin embargo, se puede apreciar la atenuación de la onda cáustica, y cómo la imagen final mantiene los detalles nítidamente.

Mediante los resultados obtenidos en estos casos de estudio, es posible aseverar que utilizando el filtro propuesto es posible reducir la incidencia de la onda cáustica en la imagen filtrada considerablemente, obteniendo una imagen sin perturbaciones pero manteniendo la nitidez original. Como se explica anteriormente, la motivación principal para el desarrollo del filtro propuesto es aplicarlo para evitar las posibles desmejoras en el desempeño de sistemas SLAM visual en ambiente con perturbaciones cáustica. Resultados de la utilización del filtro sobre dicha aplicación se concentran en el capítulo 6.

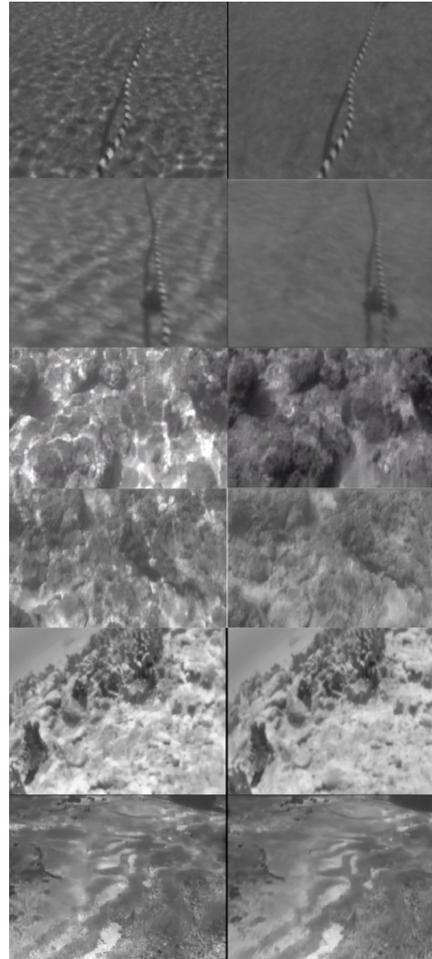


FIGURA 4.4: Escenas de evaluación del filtro de onda cáustica propuesto. La primer columna muestra las imágenes crudas capturadas por la cámara. La columna de la derecha muestra la correspondiente imagen filtrada mediante el sistema propuesto



## Capítulo 5

# Control Adaptivo con Visión Artificial

### 5.1. Introducción

En este capítulo se presenta el conjunto del sistema de control con todos los elementos del sistema de visión que posibilitan la exploración y las revisitas en un ambiente subacuático totalmente desconocido.

La estructura del conjunto se ilustra en diagrama en bloques en la figura 5.1. En la misma se pueden apreciar tres bloques fundamentales en torno a un vehículo submarino autónomo (AUV), a saber:

1. el sistema de navegación constituido por una cámara monocular y un bloque de filtrado de imagen perturbada por ondas cáusticas,
2. el sistema de guía para construcción de un corredor basado en C-SLAM, odometría visual, ajuste de referencias en cierres de lazo y mapeo denso, y
3. el sistema de control adaptivo basado en gradiente de velocidad.

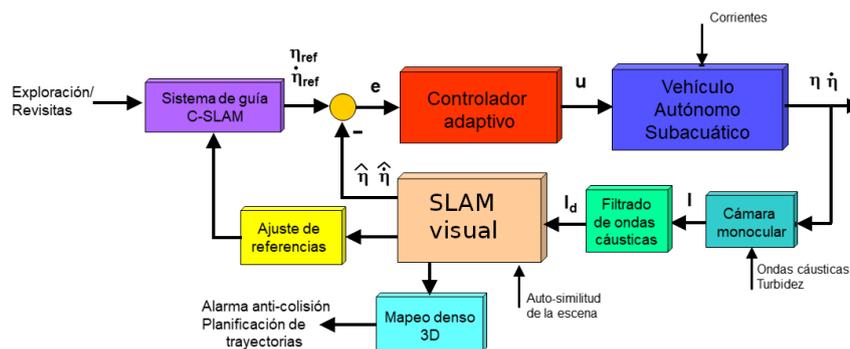


FIGURA 5.1: Propuesta de un sistema de control adaptivo con visión artificial para la exploración subacuática y retorno al punto de origen por un corredor del fondo marino.

En primer lugar se encuentran las entradas del sistema de visión constituido por las instancias de exploración o revisita del corredor. El tiempo de exploración es limitado por la autonomía energética, la cual debe ser de conocimiento antes de la exploración, lo que prácticamente significa no sobrepasar el punto de retorno. Luego están las perturbaciones que se describen junto a los bloques que determinan su ámbito de afectación, por ejemplo, las corrientes marinas dificultan la gobernabilidad del vehículo submarino. Por otro lado, las perturbaciones lumínicas producidas por

la refracción de la luz solar sobre el fondo afectan la fotometría natural de la imagen. Similarmente las partículas en suspensión en el medio acuoso perjudican la nitidez de la imagen. Finalmente, las características de la escena del suelo tales como la baja textura y relieve como la auto-similitud de los patrones de características sobre el fondo, dificultan los cierres de lazo del sistema odométrico visual sobre todo en el retorno del vehículo.

Finalmente las salidas del sistema las componen la posición y cinemática  $\eta$  y  $d\eta_{ref}/dt$  en seis grados de libertad del vehículo, las alarmas de anticollisión y la planificación de sub-trayectorias de exploración transversales al corredor (bifurcaciones).

Existen también salidas internas del sistema de visión, que son el mapa denso global 3D del corredor, también el mapa ralo de keypoints del mismo corredor, y la trayectoria odométrica  $\hat{\eta}$  y la de referencia  $\eta_{ref}$  la cual es ajustada en cada nodo en el retorno por el sistema de odometría visual.

Tanto los mapas como la trayectoria troncal del corredor son ajustados en cada ida y vuelta completa entre los puntos extremos del corredor troncal. Se debe recordar que la primera trayectoria de referencia  $\eta_{ref}$  representa la trayectoria en la exploración obtenida óptimamente siguiendo las direcciones en la escena que apuntaban a la máxima densidad de keypoints robustos. Luego en la primera vuelta ésta es reajustada a medida que el odómetro visual va encontrando los nodos en un orden estricto.

Cabe destacar que no existen coordenadas globales para describir la trayectoria del corredor, sólo una secuencia de nodos y pequeños tramos internodales que constituyen la trayectoria, los cuales son reconocidos paso a paso, en un orden prácticamente inmutable, por el sistema C-SLAM, sin saber su posición real.

La desviación de la trayectoria odométrica respecto a la real desconocida, es permanente debido al error estocástico de estimación de posición que se acumula paulatinamente. Más aún, la verdadera trayectoria troncal del corredor puede lucir errática o incluso cruzarse a sí misma en un punto cuando la descripción odométrica desconoce este hecho y marca una trayectoria distinta.

Dado que la estrategia del sistema de control es formar un corredor para la navegación segura entre dos puntos extremos, el objetivo estaría cumplido con la capacidad de unir estos puntos sin perderse y sin necesidad de conocer la posición correcta. Tampoco los problemas de escala que comúnmente aparecen en métodos de SLAM, afectarán el objetivo de navegar por el corredor, ya que el éxito depende solamente del cierre de lazo entre nodos, es decir del reconocimiento del próximo nodo desde otro nodo próximo a él.

Los seguimientos de la trayectoria geométrica de referencia del corredor  $\eta_{ref}$  a un ritmo dado por la cinemática de referencia  $d\eta_{ref}/dt$ , alcanzan una calidad máxima cuando el sistema de control puede llevar el vehículo en tiempo y forma al sitio de cada nodo con la orientación correcta de la cámara para captar la próxima escena del siguiente nodo. En resumen, si el sistema C-SLAM con su odómetro visual logra reconocer al siguiente nodo de la trayectoria, entonces llegar a él depende de la performance y robustez del sistema de control. Hay que resaltar las perturbaciones específicas de cada sistema por separado, es decir las perturbaciones lumínicas o repeticiones de escenas similares que merman la capacidad de sistema C-SLAM y por otro lado las corrientes e incertidumbres dinámicas del sistema de control que afectan su comportamiento.

En resumen, el éxito de la exploración con retorno seguro radica en la robustez del sistema C-SLAM propuesto conjuntamente con la alta performance del sistema de control. Esto es debido a que pequeños errores de postura de la cámara al arribo

del submarino a un nodo, perturban la línea de visión y el punto de vista con que la imagen de la escena del nodo es captada, exigiendo por ello al sistema de visión una capacidad significativa para reconocer la misma escena.

### 5.1.1. Sistema de Control Adaptivo

En esta sección se presenta el diseño de un sistema de control adaptivo con la justificación sólida del porqué de esta elección, y finalmente se comparará su performance en casos de estudio con otros sistemas de control robustos conocidos.

Al igual que el sistema C-SLAM que actualiza continuamente los mapas (denso y ralo) para refinar la estimación de la trayectoria del vehículo dentro del corredor, el control adaptivo ofrece la ventaja del auto-ajuste del controlador a la dinámica cambiante del AUV y la disminución del efecto de inevitables incertidumbres en su dinámica. Con ello el controlador manifiesta la constancia de su performance ante estos cambios.

La propiedad adaptiva del control es muy conveniente si se analiza la masa del mismo durante las operaciones de una misión. Para una clase particular de AUVs con regulación de altitud por cambios de flotabilidad, la variación de masa como su distribución dentro del mismo es cambiante. Al subir o bajar o en una inclinación permanente los volúmenes de lastre cambian desplazando el baricentro en altura como a lo largo del vehículo. Si bien estos cambios no son muy grandes relativamente a la masa total, afectan la justeza del seguimiento de trayectorias con cinemáticas significativas.

Existe otra razón para la elección de un control adaptivo y está vinculado a la complejidad de la dinámica de submarinos subacuáticos. En particular la resistencia hidrodinámica de carácter no-lineal y la masa aditiva que representa a la inercia del vehículo por arrastrar una masa de agua consigo mismo, representan matrices de parámetros desconocidos muy difíciles de modelar. Existe la posibilidad de estimar el modelo dinámico con costosos ensayos en laboratorio, en particular con canal de corrientes, bancos de prueba e instrumental sensible para medición de fuerzas [106]. O bien, se pueden identificar sus parámetros en tiempo real con estimadores basados en regresiones lineales y método de mínimos cuadrados [107]. También este método resulta poco práctico para un AUV dada la compleja sensorica de posición y velocidad exigida, la cual puede ser práctica para submarinos teleoperados (ROV) pero no para AUVs.

Por esta razón, resulta muy apropiado adaptar el controlador a los parámetros de la dinámica desconocidos, los cuales también pueden variar lenta y persistentemente o cambiar súbitamente pero con suficiente espaciamento en el tiempo.

## 5.2. Ecuación de Movimiento de Vehículos Subacuáticos

A lo largo de este capítulo, se adopta la nomenclatura utilizada en [108]. Se define la posición del vehículo  $\eta \in \mathbb{R}^6$  con respecto a un sistema de coordenadas fijo en la tierra  $r_I$  como:

$$\begin{aligned}\eta &= [\eta_1, \eta_2]^T \\ \eta_1 &= [x, y, z]^T \\ \eta_2 &= [\phi, \theta, \psi]^T\end{aligned}\tag{5.1}$$

La velocidad del vehículo  $\nu \in \mathbb{R}^6$  con respecto a un sistema de coordenadas fijo al vehículo  $r_0$ :

$$\begin{aligned}
\boldsymbol{\nu} &= [\boldsymbol{\nu}_1, \boldsymbol{\nu}_2]^T \\
\boldsymbol{\nu}_1 &= [u, v, w]^T \\
\boldsymbol{\nu}_2 &= [p, q, r]^T
\end{aligned} \tag{5.2}$$

Las fuerzas aplicados al vehículo  $\boldsymbol{\tau} \in \mathfrak{R}^6$  representadas con respecto a  $\boldsymbol{r}_0$ :

$$\begin{aligned}
\boldsymbol{\tau} &= [\boldsymbol{\tau}_1, \boldsymbol{\tau}_2]^T \\
\boldsymbol{\tau}_1 &= [X, Y, Z]^T \\
\boldsymbol{\tau}_2 &= [K, M, N]^T
\end{aligned} \tag{5.3}$$

El centro de masa del vehículo  $\boldsymbol{r}_g \in \mathfrak{R}^6$  representado con respecto a  $\boldsymbol{r}_0$ :

$$\boldsymbol{r}_g = [x_g, y_g, z_g]^T \tag{5.4}$$

Utilizando estas definiciones, se puede obtener la siguiente ecuación dinámica de movimiento con 6 grados de libertad:

$$\begin{aligned}
M\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) &= \boldsymbol{\tau} \\
\dot{\boldsymbol{\eta}} &= \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}
\end{aligned} \tag{5.5}$$

Donde la matriz  $\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A \in \mathfrak{R}^{6 \times 6}$  corresponde a la masa del vehículo teniendo en cuenta los efectos de la masa adherida. La matriz  $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}) \in \mathfrak{R}^{6 \times 6}$  describe los efectos de fuerzas de coriolis y centrípetas, donde también se tienen en cuenta efectos por masa adherida. La matriz  $\boldsymbol{D}(\boldsymbol{\nu}) \in \mathfrak{R}^{6 \times 6}$  modela los efectos causados por fricción y viscosidad del medio acuoso, y el vector  $\boldsymbol{g}(\boldsymbol{\eta}) \in \mathfrak{R}^6$  modela las fuerzas gravitatorias y de flotación actuando sobre el centro de masa del vehículo.  $\boldsymbol{J}(\boldsymbol{\eta}) \in \mathfrak{R}^{6 \times 6}$  es una matriz de cambio de sistemas de coordenadas, desde  $\boldsymbol{r}_o$  hacia  $\boldsymbol{r}_I$ .

Las expresiones para las matrices  $\boldsymbol{M}_{RB}$  y  $\boldsymbol{C}_{RB}$  están dadas por:

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_3 & -m\boldsymbol{S}(\boldsymbol{r}_g) \\ m\boldsymbol{S}(\boldsymbol{r}_g) & \boldsymbol{I}_0 \end{bmatrix} \tag{5.6}$$

$$\boldsymbol{C}_{RB} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -m\boldsymbol{S}(\boldsymbol{\nu}_1) - m\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{S}(\boldsymbol{r}_g) \\ -m\boldsymbol{S}(\boldsymbol{\nu}_1) + m\boldsymbol{S}(\boldsymbol{r}_g)\boldsymbol{S}(\boldsymbol{\nu}_2) & -\boldsymbol{S}(\boldsymbol{I}_0\boldsymbol{\nu}_2) \end{bmatrix} \tag{5.7}$$

Donde  $m$  es la masa total del vehículo,  $\boldsymbol{I}_0 \in \mathfrak{R}^{3 \times 3}$  es la matriz de inercia expresada en  $\boldsymbol{r}_0$  y  $\boldsymbol{S}(x) \in \mathfrak{R}^{3 \times 3}$  es una matriz operador que realiza el producto vectorial:

$$\boldsymbol{S}_x = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{5.8}$$

Para un vehículo totalmente sumergido en el agua, con 3 planos de simetría, se pueden derivar las siguientes expresiones  $\boldsymbol{M}_A$ ,  $\boldsymbol{C}_A(\boldsymbol{\nu})$  y  $\boldsymbol{D}(\boldsymbol{\nu})$

$$\boldsymbol{M}_A = -diag \{m_{A_{11}}, m_{A_{22}}, m_{A_{33}}, m_{A_{44}}, m_{A_{55}}, m_{A_{66}}\} \tag{5.9}$$

$$\mathbf{C}_A = \begin{bmatrix} 0,0 & 0,0 & 0,0 & 0,0 & -m_{A_{33}} |w| & m_{A_{22}} |v| \\ 0,0 & 0,0 & 0,0 & m_{A_{33}} |w| & 0,0 & -m_{A_{11}} |u| \\ 0,0 & 0,0 & 0,0 & -m_{A_{22}} |v| & m_{A_{11}} |u| & 0,0 \\ 0,0 & -m_{A_{33}} |w| & m_{A_{22}} |v| & 0,0 & -m_{A_{66}} |r| & -m_{A_{55}} |q| \\ m_{A_{33}} |w| & 0,0 & -m_{A_{11}} |u| & m_{A_{66}} |r| & 0,0 & -m_{A_{44}} |p| \\ -m_{A_{22}} |v| & m_{A_{11}} |u| & 0,0 & -m_{A_{55}} |q| & m_{A_{44}} |p| & 0,0 \end{bmatrix} \quad (5.10)$$

$$\mathbf{D}(\boldsymbol{\nu}) = -diag \{d_{l_{11}}, d_{l_{22}}, d_{l_{33}}, d_{l_{44}}, d_{l_{55}}, d_{l_{66}}\} + \\ -diag \{d_{c_{11}} |u|, d_{c_{22}} |v|, d_{c_{33}} |w|, d_{c_{44}} |p|, d_{c_{55}} |q|, d_{c_{66}} |r|\} \quad (5.11)$$

Las fuerzas gravitatorias y de flotación responden a:

$$\mathbf{g}(\boldsymbol{\eta}) = \begin{bmatrix} \mathbf{g}_1(\boldsymbol{\eta}) \\ \mathbf{g}_2(\boldsymbol{\eta}) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_g(\boldsymbol{\eta}) + \mathbf{f}_b(\boldsymbol{\eta}) \\ \mathbf{r}_g \times \mathbf{f}_g(\boldsymbol{\eta}) + \mathbf{r}_b \times \mathbf{f}_b(\boldsymbol{\eta}) \end{bmatrix} \quad (5.12)$$

donde  $\mathbf{f}_g$  modela las fuerzas gravitatorias actuando sobre el centro de masa  $\mathbf{r}_g$  y  $\mathbf{f}_b$  modela las fuerzas de flotación actuando sobre el centro de flotación  $\mathbf{r}_b$ .

### 5.2.1. Control Adaptivo de Gradiente de Velocidad

Siguiendo la referencia [109], en donde un control adaptivo de gradiente de velocidad prueba su adecuación y performance en vehículos subacuáticos teleoperados (ROVs), en esta tesis se adaptará este tipo de controlador a un vehículo subacuático genérico con el propósito de lograr la guía del vehículo en el contexto de la figura 5.1. A continuación se hará un breve resumen del control adaptivo de gradiente de velocidad desarrollado en [109].

#### Funcional de costo para seguimiento de trayectorias

El objetivo de seguimiento se define como:

$$\lim_{t \rightarrow \infty} (\boldsymbol{\eta}(t) - \boldsymbol{\eta}_r(t)) = 0 \quad (5.13)$$

$$\lim_{t \rightarrow \infty} (\boldsymbol{\nu}(t) - \boldsymbol{\nu}_r(t)) = 0 \quad (5.14)$$

Para diseñar una ley de control de manera de obtener condiciones descritas en las ecuaciones 5.13 y 5.14, se realiza el cambio de coordenadas:

$$\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_r \quad (5.15)$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \mathbf{J}^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}_r + \mathbf{J}^{-1}(\boldsymbol{\eta})\mathbf{K}_p\tilde{\boldsymbol{\eta}} \quad (5.16)$$

donde  $\mathbf{K}_p = \mathbf{K}_p^T \geq 0$  es una matriz de diseño de ganancia para el vector de error de posición. Claramente de las Ec. (5.15)-(5.16), si  $\tilde{\boldsymbol{\eta}}$  es cero con  $\boldsymbol{\nu} = \mathbf{J}^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}$ , es válido que  $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \boldsymbol{\nu}_r$ .

La función de costo energético responde a:

$$\mathbf{Q}(t, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{\nu}}) = \frac{1}{2}\tilde{\boldsymbol{\eta}}^T \tilde{\boldsymbol{\eta}} + \frac{1}{2}\tilde{\boldsymbol{\nu}}^T \mathbf{M}(t)\tilde{\boldsymbol{\nu}} \quad (5.17)$$

Realizando la derivada temporal de  $Q(t)$  en los tramos continuos de la trayectoria, es decir en  $\mathcal{S}_t \setminus \mathcal{S}_{t_k}$ , se obtiene combinando las Ec. (5.16)-(5.17) con las Ec. (5.5):

$$\dot{Q}(t, t_0, \eta_0, \nu_0) = -\tilde{\eta}^T K_p \tilde{\eta} + \tilde{\eta}^T J(\eta) \tilde{\nu} - \quad (5.18)$$

$$\tilde{\nu}^T C_c(t, \nu) \nu - \tilde{\nu}^T D_l(t) \nu - \tilde{\nu}^T D_q(t, |\nu|) \nu - \tilde{\nu}^T g(t, \eta) - \quad (5.19)$$

$$\tilde{\nu}^T M(t) d(t, \tilde{\eta}, \tilde{\nu}) - \tilde{\nu}^T \frac{\dot{M}_c(t)}{2} \nu + \tilde{\nu}^T \frac{\dot{M}_c(t)}{2} \tilde{\nu} + \quad (5.20)$$

$$\tilde{\nu}^T \tau_c + \tilde{\nu}^T \tau_t \quad (5.21)$$

donde

$$d(t, \tilde{\eta}, \tilde{\nu}) = \frac{d}{dt}(J^{-1}(\eta) \tilde{\eta}_r) - \frac{dJ^{-1}(\eta)}{dt} K_p \tilde{\eta} + \quad (5.22)$$

$$J^{-1}(\eta) K_p^2 \tilde{\eta} - J^{-1}(\eta) K_p J(\eta) \tilde{\nu} \quad (5.23)$$

La fuerza de acción de control propuesta en [109] responde a:

$$\tau_t(t) = \sum_{i=1}^6 U_i \times C_{\nu_i}(\nu_i) \nu + U_7 \nu + \sum_{i=1}^6 U_{i+7} |\nu_i| \nu + \quad (5.24)$$

$$U_{14} g_1 + U_{15} g_2 + U_{16} d + U_{17} \tilde{\nu} - K_\nu \tilde{\nu} - J^T \tilde{\eta} - \tau_c \quad (5.25)$$

donde  $U_i \in \mathbb{R}^{6 \times 6}$  son matrices de control de estados realimentados,  $C_{\nu_i}$  es la matriz indicada en la Ec. (5.5),  $g_1$  y  $g_2$  son los vectores definidos en la Ec. (5.12),  $d$  está definido por la Ec. (5.23) y, finalmente,  $K_\nu$  es una nueva matriz de diseño de ganancia para los errores cinemáticos tal que  $K_\nu = K_\nu^T \geq 0$ .

### 5.2.2. Leyes Adaptivas

Las matrices  $U_i$ 's se definieron por medio de  $n_u$  leyes adaptivas de control basadas en funciones SG:

$$\dot{U}_i = -\Gamma_i \frac{\partial \dot{Q}(U_i)}{\partial U_i}, \text{ para } t > t_0, \quad (5.26)$$

con  $\Gamma_i = \Gamma_i^T > 0$  una matriz de ganancia definida positiva.

Particularmente, se cumple para las  $U_i$ 's con  $i = 1, \dots, 6$ :

$$\dot{U}_i = -\Gamma_i (\tilde{\nu} \nu^T) \cdot C_{\nu_i}(\nu_i) \quad (5.27)$$

Luego, para la componente lineal de la fuerza de arrastre  $D_l$ , sumado al término de Coriolis  $\frac{1}{2} \dot{M}_c(t)$  le corresponde la ley:

$$\dot{U}_7 = -\Gamma_7 \tilde{\nu} \nu^T \quad (5.28)$$

Análogamente, para la componente cuadrática de la fuerza de arrastre  $D_c$  en (5.11) e  $i = 8, \dots, 13$ , se obtiene:

$$\dot{U}_i = -\Gamma_i \tilde{\nu} \nu^T |\nu_{i-7}| \quad (5.29)$$

donde  $\nu_i$  es el elemento  $i$ -ésimo de  $\nu$ .

Similarmente, se definen las siguientes leyes para las matrices de parámetros correspondientes al vector de flotabilidad  $g$  considerando la descomposición realizada en la Ec. (5.12):

$$\dot{U}_{14} = -\Gamma_{14}\tilde{\nu}g_1^T \quad (5.30)$$

$$\dot{U}_{15} = -\Gamma_{15}\tilde{\nu}g_2^T \quad (5.31)$$

A continuación, para la matriz de inercia  $M$  se asigna la siguiente ley:

$$\dot{U}_{16} = -\Gamma_{16}\tilde{\nu}d^T \quad (5.32)$$

donde  $d$  es el vector auxiliar definido en la Ec. (5.23).

Finalmente, la velocidad de cambio de la componente de Coriolis variante en el tiempo  $\frac{1}{2}\dot{M}_c$  también debe compensarse a través de:

$$\dot{U}_{17} = -\Gamma_{17}\tilde{\nu}\tilde{\nu}^T. \quad (5.33)$$

La integración de las leyes adaptivas con  $U_i(t_0) \in \mathcal{S}_U \subset \mathbb{R}^{6 \times 6}$ , para  $i = 1, \dots, 17$ ,  $\mathcal{S}_U$  siendo un conjunto compacto para las condiciones iniciales de los integradores de las Ec. (5.27)-(5.33), proveen de manera directa las matrices adaptivas del controlador  $U_i(t)$  para el problema de seguimiento de trayectoria.

### 5.2.3. Leyes Adaptivas Modificadas

A fin de obtener *a-priori* matrices  $U_i(t)$  acotadas y suaves, en [109] se modifica la ley adaptiva de la Ec. (5.26) con una proyección dinámica suavizada sobre cada vector columna  $j$  de  $U_i$  denominado  $u_{i,j}$ :

$$\begin{aligned} \dot{u}_{i,j} &= \text{Proy} \left( - \left( \Gamma_i \frac{\partial \dot{Q}}{\partial U_i} \right)_j \right) = \\ &= \begin{cases} \left( -\Gamma_i \frac{\partial \dot{Q}}{\partial U_i} \right)_j, & \text{para } u_{i,j} \in \overline{M}_u \text{ ó } \left( \nabla_{u_{i,j}} \mathcal{P} \right)^T \left( -\Gamma_i \frac{\partial \dot{Q}}{\partial U_i} \right)_j \leq 0 \\ \left( \mathbf{I} - c(u_{i,j}) \Lambda \frac{\nabla_{u_{i,j}} \mathcal{P} \nabla_{u_{i,j}} \mathcal{P}^T}{\nabla_{u_{i,j}} \mathcal{P} \Lambda \nabla_{u_{i,j}} \mathcal{P}^T} \right) \left( -\Gamma_i \frac{\partial \dot{Q}}{\partial U_i} \right)_j \\ & \text{para } u_{i,j} \in M_{u+\varepsilon} \setminus \overline{M}_u \text{ ó } \left( \nabla_{u_{i,j}} \mathcal{P} \right)^T \left( -\Gamma_i \frac{\partial \dot{Q}}{\partial U_i} \right)_j > 0 \end{cases} \end{aligned} \quad (5.34)$$

con  $c(u_{i,j})$  una función vectorial definida como:

$$c(u_{i,j}) = \min \left\{ 1, \frac{\mathcal{P}(u_{i,j})}{\varepsilon} \right\} \quad (5.35)$$

donde  $\text{Proy}(\cdot)$  denota la modificación propiamente dicha de la proyección mencionada, además  $(\cdot)_j$  se refiere a la columna  $j$  de una matriz,  $\mathcal{P}$  es una función convexa en el espacio paramétrico de  $u_{i,j}$  y se define sobre un conjunto convexo  $M_{u+\varepsilon} = \{u_{i,j} \in \mathcal{R}^6 / \mathcal{P}(u_{i,j}) \leq \varepsilon\}$ , que se compone de la unión de  $M_u = \{u_{i,j} \in \mathcal{R}^6 / \mathcal{P}(u_{i,j}) \leq 0\}$  y la capa circundante  $M_{u+\varepsilon} \setminus M_u$  de espesor  $\varepsilon > 0$  arbitrariamente pequeña. Así  $\overline{M}_u$  es el interior de  $M_u$ ,  $\partial M_u$  es su contorno y  $\partial M_{u+\varepsilon}$  es el contorno externo de  $M_{u+\varepsilon}$ , ambos asumidos suaves. Finalmente,  $\Lambda$  es una matriz real que cumple  $\Lambda = \Lambda^T > 0$ .

### 5.2.4. Simplificación de los Modos a Controlar

Dado que las dinámicas parciales de rolido y cabeceo frontal son auto-estabilizantes y los movimientos relativamente pequeños para la navegación de un AUV con forma de torpedo, es que dichos modos pueden despreciarse y su interacción con los modos principales de altitud y navegación planar es muy pequeño. Si  $r_g$  coincide con  $r_0$  y si la altitud  $z$  es mantenida constante con un controlador independiente a los modos  $x$  y  $y$  y  $\psi$ , se obtiene el siguiente modelo simplificado similar a un sistema no-holonomo:

$$\begin{aligned}\eta &= [x, y, \psi]^T \in \mathbb{R}^3 \\ \nu &= [u, v, r]^T \in \mathbb{R}^3 \\ \tau &= [X, Y, N]^T \in \mathbb{R}^3\end{aligned}\quad (5.36)$$

$$M = \begin{bmatrix} m - m_{A_{11}} & 0,0 & 0,0 \\ 0,0 & m - m_{A_{22}} & 0,0 \\ 0,0 & 0,0 & I_{33} - m_{A_{33}} \end{bmatrix}\quad (5.37)$$

$$C(\nu) = \begin{bmatrix} 0,0 & 0,0 & m_{A_{22}}v - m \cdot v \\ 0,0 & 0,0 & -m_{A_{11}}u + m \cdot u \\ -m_{A_{22}}v + m \cdot v & m_{A_{11}}u - m \cdot u & 0,0 \end{bmatrix}\quad (5.38)$$

$$D(\nu) = \begin{bmatrix} -d_{l_{11}} - d_{c_{11}} \cdot |u| & 0,0 & 0,0 \\ 0,0 & -d_{l_{22}} - d_{c_{22}} \cdot |v| & 0,0 \\ 0,0 & 0,0 & -d_{l_{33}} - d_{c_{33}} \cdot |r| \end{bmatrix}\quad (5.39)$$

el cual se expresa en espacio de estados como:

$$\begin{aligned}\dot{\nu} &= \Phi \nu + \tau \\ \dot{\eta} &= J(\psi) \nu\end{aligned}\quad (5.40)$$

$$\Phi = \begin{bmatrix} \frac{d_{l_{11}} + d_{c_{11}} \cdot |u|}{m - m_{A_{11}}} & 0,0 & \frac{m_{A_{22}}v - m \cdot v}{m - m_{A_{11}}} \\ 0,0 & \frac{d_{l_{22}} + d_{c_{22}} \cdot |v|}{m - m_{A_{22}}} & \frac{-m_{A_{11}}u + m \cdot u}{m - m_{A_{22}}} \\ \frac{-m_{A_{22}}v + m \cdot v}{I_{33} - m_{A_{33}}} & \frac{-m_{A_{11}}u + m \cdot u}{I_{33} - m_{A_{33}}} & \frac{d_{l_{33}} + d_{c_{33}} \cdot |r|}{I_{33} - m_{A_{33}}} \end{bmatrix}\quad (5.41)$$

$$J(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}\quad (5.42)$$

Finalmente los propulsores son comandados por una acción de control  $u = [avance, giro]$  que responde a:

$$\tau = \begin{bmatrix} \frac{(\gamma_1(\text{sign}(u_1)+1) - \gamma_2(\text{sign}(u_1)-1))}{2} & \frac{(\gamma_3(\text{sign}(u_2)+1) - \gamma_4(\text{sign}(u_2)-1))}{2} \\ \frac{(\gamma_5(\text{sign}(u_1)+1) - \gamma_6(\text{sign}(u_1)-1))}{2} & \frac{(\gamma_7(\text{sign}(u_2)+1) - \gamma_8(\text{sign}(u_2)-1))}{2} \\ \frac{(\gamma_9(\text{sign}(u_1)+1) - \gamma_{10}(\text{sign}(u_1)-1))}{2} & \frac{(\gamma_{11}(\text{sign}(u_2)+1) - \gamma_{12}(\text{sign}(u_2)-1))}{2} \end{bmatrix} \cdot u\quad (5.43)$$

### 5.2.5. Identificación de Parámetros

A los fines de probar el desempeño del diseño propuesto, se decide primeramente emplear un simulador de un escenario submarino y un vehículo existente

de marca OpenROV. El objetivo es lograr la formación de un corredor con control adaptivo de gradiente de velocidad con visión artificial.

Para identificar los parámetros  $P$  desconocidos del vehículo OpenROV, constituidos por  $m_{A_{11,33}}$ ,  $d_{l_{11,33}}$ ,  $d_{c_{11,33}}$  y  $\gamma_{1,12}$ , se realizó una experimentación en pileta. Se excitó  $u$  con una señal persistente existente y se obtuvo  $\eta_{ref}$  utilizando un sensor visual basado en ORB-SLAM [13]. Luego, se utilizó un método de identificación por gradiente, para obtener los parámetros  $P$  que minimicen el costo:

$$F(P) = \nu_{ref} \cdot \Lambda \cdot \nu(P) \quad (5.44)$$

con:

$$\begin{aligned} \nu_{ref} &= J(\psi)^{-1} \dot{\eta}_{ref} \\ \dot{\eta}_{ref_k} &= \alpha \dot{\eta}_{ref_{k-1}} + (1 - \alpha) \frac{\eta_{ref_k} - \eta_{ref_{k-2}}}{2\Delta T} \\ \nu(P) &= reg \cdot P \end{aligned} \quad (5.45)$$

donde  $\Delta T$  es el tiempo de muestreo y  $reg$  es el regresor correspondiente a la ecuación 5.40.

### Ambigüedad en Escala

Mediante métodos de SLAM monocular, como [13], se obtiene una estimación de  $\eta$  afectada por un factor de escala desconocido  $\lambda\eta$ . De esta manera los parámetros obtenidos  $P$  sufren de una ambigüedad de escala. Para solucionar este problema, se escalaron los parámetros  $P$  con respecto a las variables físicas conocidas mediante experimentación de laboratorio, como son  $m$  y  $I_{33}$ . Véase el trabajo [110], donde estos parámetros son obtenidos por simulaciones hidrodinámicas detalladas para el submarino OpenROV descrito en el capítulo 6.

## 5.3. Evaluación Comparativa del Sistema de Control

En esta sección se compara el desempeño de tres controladores de distintos principios de diseño, a saber: un PID, un controlador de modo deslizante descrito en [106], y el controlador adaptivo diseñado anteriormente. A continuación se detallan las ecuaciones que rigen el comportamiento de cada controlador.

Controlador Proporcional-Derivativo-Integral (PID):

$$\begin{aligned} \tilde{\eta} &= \eta_{ref} - \eta \\ \tilde{\nu} &= J^{-1} \tilde{\eta} \\ u_1 &= K_{p_1} (\tilde{\nu}(1) + K_{i_1} \int \tilde{\nu}(1)) \\ u_2 &= K_{p_2} (\tilde{\nu}(2) + K_{d_2} \dot{\tilde{\nu}}(2)) \end{aligned} \quad (5.46)$$

Controlador de Modo Deslizante:

$$\begin{aligned} u &= B^\dagger [K_D s + K_s \text{sign}(s)] \\ \text{sign}(x_i) &= \begin{cases} 1 & \text{si } x_i \geq 0 \\ -1 & \text{si } x_i < 0 \end{cases} \\ s &= \Lambda [R_I^B \tilde{\eta}] - [\nu] \end{aligned} \quad (5.47)$$

donde  $B^\dagger$  es la matriz pseudo inversa de  $B$ .

Controlador Adaptivo de gradiente de velocidad:

$$\begin{aligned}\tilde{\eta} &= \eta - \eta_r \\ \tilde{\nu} &= \nu - J^{-1}\dot{\eta}_r + J^{-1}K_p\tilde{\eta}\end{aligned}\quad (5.48)$$

$$Q(\tilde{\eta}, \tilde{\nu}) = \tilde{\eta}^T \tilde{\eta} + \tilde{\nu}^T \Phi \tilde{\nu} \quad (5.49)$$

$$\begin{aligned}\tau &= -J^{-1}\tilde{\nu} + U_1 \times \Lambda_r \nu + U_2 d - K_v \tilde{\nu} \\ d &= J^{-1}(\ddot{\eta}_r - K_p \dot{\eta}_r) + \dot{J}^{-1}(\dot{\eta}_r - K_p \tilde{\eta}) \\ \Lambda(r) &= \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix} \\ \dot{U}_1 &= -\Gamma_1 (\tilde{\nu} \nu^T) \times \Lambda(r) \\ \dot{U}_2 &= -\Gamma_2 \tilde{\nu} d^T\end{aligned}\quad (5.50)$$

## 5.4. Simulación

Para la evaluación se realizó una simulación utilizando el modelo simplificado del sistema obtenido en la sección 5.2.4. Se evaluó el desempeño del controlador en una trayectoria de referencia sinusoidal, sobre una superficie con dimensiones similares a las de la piletta empleada para obtener los resultados experimentales (véase el capítulo 6). Se postula un objetivo de control de manera que el vehículo recorra una trayectoria de referencia en determinada dirección, para luego volver utilizando la dirección contraria, de idéntica forma a como se define en el capítulo 2 para el método C-SLAM propuesto. La velocidad crucero de referencia se estableció en  $0,1m/s$ , la cual es deseable según las dimensiones de la piletta de experimentación y las capacidades del sensor visual utilizado (véase los capítulos 2 y 6).

### Controlador Proporcional-Derivativo-Integral (PID)

Se propone un controlador PID cuyos parámetros son ajustados manualmente buscando una buena performance en cada modo controlado. En las figuras siguientes, en (A) se marca en rojo la trayectoria de referencia a seguir y en azul las trayectorias de ida y vuelta, esta última moviéndose el vehículo hacia atrás. En (B) la acción de control empuja el vehículo minimizando los errores de seguimiento. En (C), la velocidad  $\nu$  del vehículo, ilustrando en color rojo  $u$ , en azul  $r$  y en verde  $v$ . Finalmente en (D) la performance del control dada por el error de seguimiento.

De los resultados obtenidos mediante el controlador PID ilustrados en la figura 5.2, se puede observar que se obtiene un error de seguimiento considerablemente alto, principalmente en los periodos de giro.

### Controlador de Modo Deslizante

Se puede notar en la figura 5.3, que si bien hay poco error en las zonas rectas de la trayectoria, el error en los giros es aún importante. También se aprecia una acción de control con un contenido de alta frecuencia elevado, normal en este tipo de controladores. Es posible que los propulsores del ROV sufran deterioro prematuro utilizando este controlador debido a las altas corrientes de armadura en sus motores y a la rotura de sus rodamientos causado por cambios de velocidad abruptos

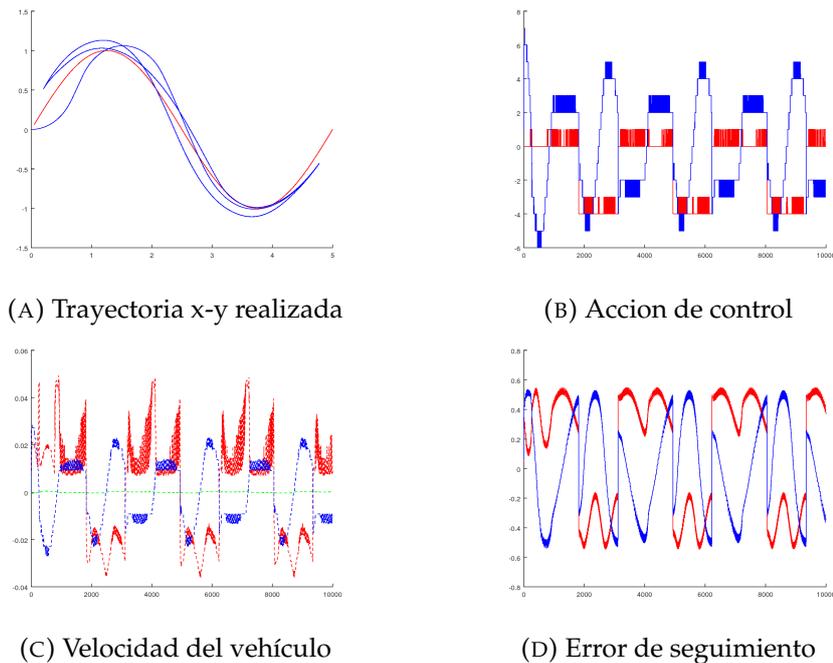


FIGURA 5.2: Resultados de simulación del controlador PID

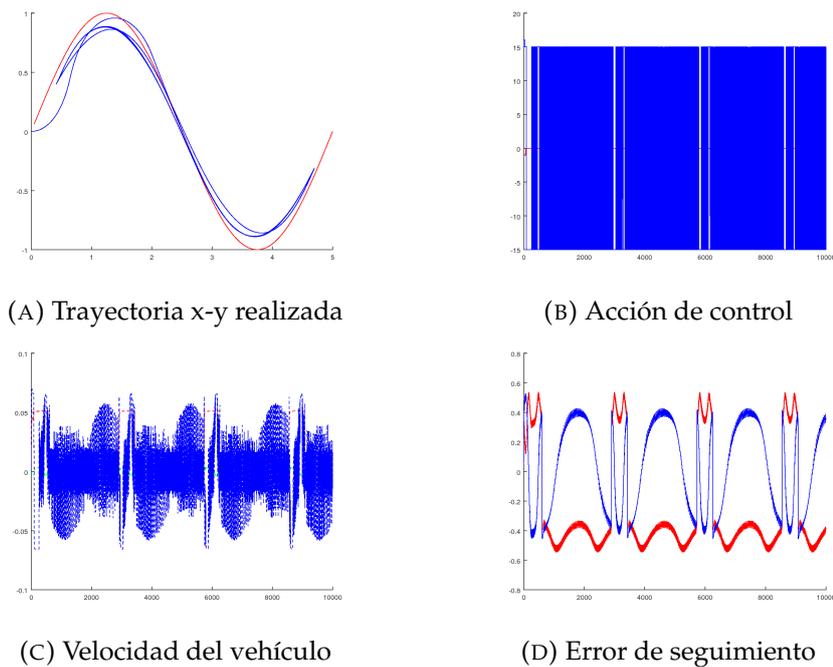
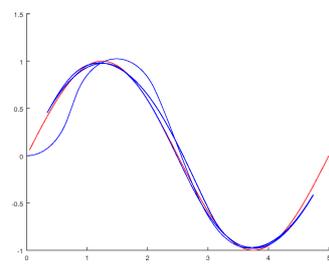


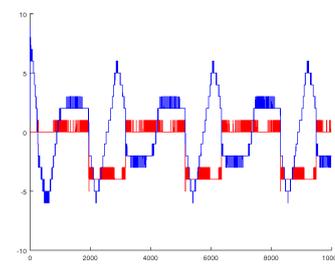
FIGURA 5.3: Resultados de simulación del controlador de modo deslizante

### Control Adaptivo de Gradiente de Velocidad

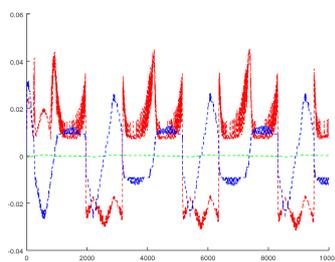
Finalmente se puede concluir que la trayectoria del vehículo utilizando el controlador adaptivo tiene un buen desempeño, tanto en las zonas rectas como en los giros (véase la figura 5.4). Esto afianza la decisión de emplear el método de control adaptivo de gradiente de velocidad propuesto.



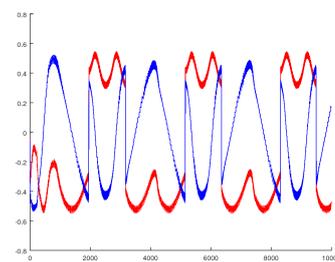
(A) Trayectoria x-y realizada



(B) Acción de control



(C) Velocidad del vehículo



(D) Error de seguimiento

FIGURA 5.4: Resultados de simulación del controlador adaptivo de gradiente de velocidad

## Capítulo 6

# Experimentación

Para la verificación de resultados se prefirió usar una plataforma estandarizada, con cierta capacidad de cálculo a bordo y posibilidades de transferir imágenes digitales en tiempo real. Se seleccionó un submarino teleoperable, cuyo tamaño y versatilidad se adaptan a una pileta de pruebas de dimensiones acotadas pero adecuadas para la ambientación subacuática y la navegación experimental. Se usaron cámaras subacuáticas adicionales para apoyo de la verificación de resultados, a la filmación de alta resolución y al seguimiento independiente del submarino.

Los experimentos fueron concebidos a cielo abierto abarcando las más variadas situaciones de iluminación sobre el fondo y viento sobre la superficie del agua. Así mismo, la turbidez del agua pudo regularse a niveles uniformes desde aguas claras hasta borrosidad muy alta con visibilidad muy dificultosa, obligando al vehículo a mantenerse en niveles muy cercanos al sobrerrelieve del fondo. Adicionalmente, se presentan también resultados de simulación en banco de pruebas ambientado (benchmark GAZEBO). Todos los algoritmos programados funcionaron bajo el esquema de visión artificial y autonomía completa en las decisiones de guiado, tal como fue descrito en los capítulos 2, 3, 4 y 5.

### 6.1. Vehículo Subacuático Utilizado

#### 6.1.1. Hardware

Como vehículo se utilizó un ROV (Remotely Operated Vehicle) desarrollado por la empresa OpenROV Underwater Drones [111]. Este ROV fue construido con dimensiones de 30 cm x 20 cm x 15 cm. Esta constituido principalmente por un domo cilíndrico estanco, dispuesto transversalmente, dentro del cual reposa toda el equipamiento a bordo. Contiene dos cilindros angostos, los cuales alojan las baterías. Una estructura plástica/metálica da rigidez al conjunto. Se puede apreciar el ROV en la figura 6.1.

El mismo está equipado con tres propulsores, uno para movimiento vertical, y dos para avance/retroceso y giro. Las baterías a bordo permiten operar el ROV durante dos horas aproximadamente, dependiendo de la frecuencia de uso de los propulsores y de los dispositivos a bordo. Asimismo, dispone de una cámara, que será detallada más adelante, cuatro luces, dos láseres de baja potencia e instrumental inercial (IMU).

El vehículo posee un cable umbilical de par trenzado, de una longitud de 100 metros, que permite la comunicación entre las computadoras a bordo y dispositivos externos.

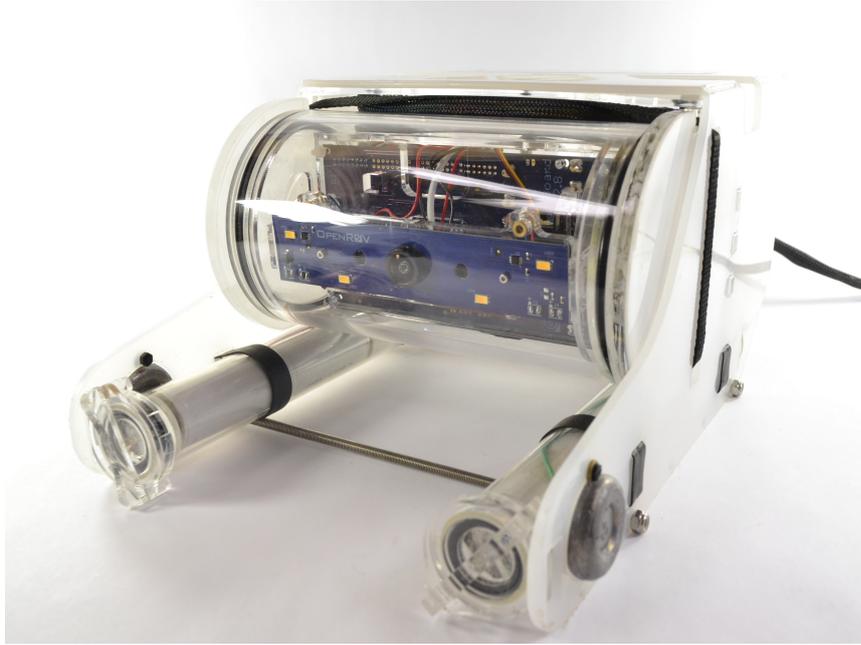


FIGURA 6.1: OpenROV

### Computadoras a bordo

El vehículo está provisto con dos computadoras independientes. La primera está diseñada a partir de un microcontrolador atmega [112]. Tiene como funcionalidad principal hacer las veces de interface de bajo nivel entre los distintos dispositivos instalados en el vehículo, como son los motores, las luces, etc.

La segunda computadora, es una mini computadora BeagleBone Black [113]. Esta se encarga principalmente de tareas de alto nivel, como el control de la cámara y la comunicación con el usuario.

### Computadora Externa

El ROV fue diseñado para que la capacidad computacional a bordo del mismo sea baja, de manera de utilizar poco espacio dentro del cilindro estanco y requerir poca energía. La idea principal es que el ROV fuera controlado por un humano utilizando una computadora externa, mediante el cable umbilical. Por esta razón se utilizó como computadora externa una notebook, que ejecuta los programas de guía navegación y control. La notebook cuenta con un procesador Intel Core i7 3610QM y 8 gb de RAM.

#### 6.1.2. Software

El fabricante provee el ROV con un sistema operativo pre-instalado basado en Debian [114], en el cual se ejecuta un servidor HTTP. Este servidor tiene como finalidad hacer las veces de interfaz de usuario, creando una página web, a la cual el usuario puede acceder, mediante el cable umbilical, desde la computadora externa. En esta página el usuario tiene acceso a la imagen capturada por la cámara, así como al completo estado del vehículo y las funciones necesarias para comandar el ROV.

En un contexto autónomo, no es necesario poseer una interfaz de usuario. Lo que es más, su ejecución significa una carga extra para el procesador a bordo, que en ciertas ocasiones se pudo observar dan como consecuencia retardos considerables en

la comunicación con la computadora externa. Por otro lado, la tendencia moderna es utilizar el sistema operativo ROS [115], especialmente diseñado para aplicaciones de robótica. Si bien el fabricante del ROV provee una interfaz ROS para el sistema operativo pre-instalado, se encuentra que en su estado actual esta interfaz se halla altamente incompleta y no posee la funcionalidad necesaria para las experimentaciones complejas que se deben realizar en esta tesis.

Por estas razones, se eliminó el sistema operativo pre-instalado y se lo reemplazó por otro basado en Ubuntu 16.04 [116]. Este sistema solamente ejecuta una aplicación, de diseño propio. La aplicación en cuestión es un nodo de ROS, el cual se encarga de capturar la imagen de la cámara, aplicar un pre-procesamiento, para luego publicar la misma como un mensaje ROS. El pre-procesamiento actualmente toma la forma de una disminución de la resolución por defecto, a 640x480, y un cambio del color a escala de grises, lo que disminuye la carga de la computadora a bordo, principalmente en lo que respecta a la latencia en la comunicación. También el nodo ROS tiene la tarea de esperar los comandos provenientes desde la computadora externa, para luego ejecutarlos o bien re-direccionarlos al controlador según el caso lo requiera. Particularmente, estos mensajes instruyen al vehículo a mantener su profundidad o controlar la velocidad de giro de los propulsores.

### 6.1.3. Configuración de Cámaras

#### Cámara Principal

La cámara principal instalada dentro del cilindro estanco del ROV es una web-cam de bajo costo modelo genius f100 . Esta tiene la capacidad de resoluciones de hasta 1270x720 píxeles, con tasas de refresco de hasta 30 cuadros por segundo. Posee un FOV (Field of view) de 120 grados (medida en ambientes terrestres). La tecnología de captura de las imágenes es denominada "Rolling Shutter" , común en este tipo de cámaras de bajo costo. La tecnología de rolling shutter se basa en una captura secuencial de los píxeles de la imagen. Por ello existe un retardo relativo en la disposición de los píxeles, cuyo efecto se deja percibir sobre todo cuando el movimiento de la cámara es rápido. Este fenómeno interferirá en las estimaciones fotométricas por lo que de ahora en más será considerado como una perturbación desconocida pero acotada en el procesamiento de imagen.

En ambientes subacuáticos, la distorsión percibida por la cámara no depende solamente de las características constructivas de la lente. La luz, para llegar al sensor de la cámara, atraviesa la columna de agua, el acrílico del domo, para luego penetrar en el aire dentro del domo y llegar finalmente a la lente. Este recorrido por medios de distintas densidades da como resultado que la luz difracte varias veces durante su trayectoria. Véase la figura 6.2. El domo hace entonces las veces de segunda lente, cuya topología también incide en las características intrínsecas de la cámara.

En la dirección vertical del domo, por su forma circular, la difracción es pequeña. Pero, por otro lado, en la dirección horizontal el domo tiene forma planar, lo que causa una gran difracción en los rayos de luz. Al mismo tiempo, dado que la cámara no está perfectamente alineada con el domo, la difracción vertical y horizontal se entremezclan en la imagen, dando como resultado un patrón de distorsión complejo. En la figura 6.3 se puede apreciar los resultados de calibrar la cámara para un modelo pinhole siguiendo el algoritmo presentado en [117]. El error medio de reproyección es alto, teniendo un valor de 0.481808, causado por la baja calidad de la cámara, la distorsión compleja del domo y por el efecto causado por el "rolling shutter", que se describe a continuación. Se puede apreciar también una disminución considerable

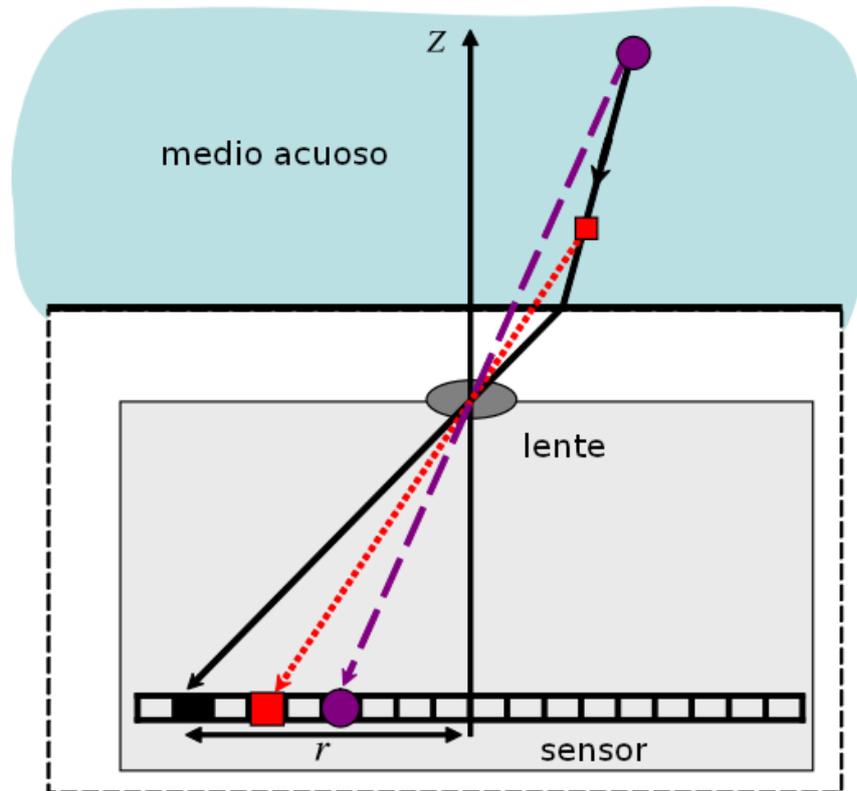
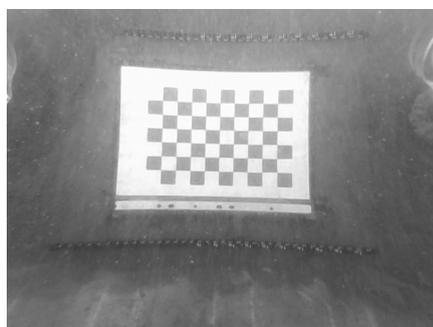
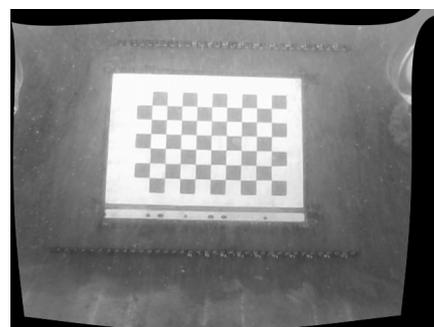


FIGURA 6.2: Ilustración de la distorsión causada por el domo en vehículos subacuáticos. La línea violeta ilustra la trayectoria realizada por la luz, partiendo desde un punto en el ambiente, atravesar la lente de la cámara e impactar en un píxel dentro del sensor. La línea negra ilustra, en cambio, la trayectoria de la luz teniendo en cuenta la difracción causada por el domo. Se observa que la luz impacta en píxeles más alejados del centro del sensor, lo que causa una reducción en el ángulo de visión efectivo

del FOV horizontal, resultando de 80.5 grados. Similarmente, los errores de calibración por efectos de distorsión y captura serán interpretados como una perturbación no-medible acotada.



(A) Imagen original sin corrección



(B) Imagen corregida

FIGURA 6.3: Distorsión en la cámara principal

Con la tecnología "rolling shutter", la lectura de los píxeles de la imagen se realiza de a una fila por vez, en lugar de leer todos los píxeles simultáneamente. Así, si

la velocidad de lectura de cada fila es pequeña (relativa a la velocidad de movimiento), se tiene que cada fila fue capturada en un instante de tiempo diferente y en una posición de la cámara diferente.

Esta forma de lectura de los píxeles resulta en distorsiones en la imagen capturada. Ejemplo de estos fenómenos se aprecian en la figura 6.4. Para evitar este problema, generalmente se utilizan cámaras “global shutter”, que efectivamente capturan todos los píxeles observados simultáneamente. Estas cámaras son más costosas y requieren computadoras a bordo potentes para su correcta utilización.



FIGURA 6.4: Efecto de rolling shutter. La imagen superior fue capturada por una cámara global shutter. La imagen inferior, por el contrario, fue obtenida utilizando una cámara rolling shutter. Se puede observar la distorsión causada por la captura de píxeles en instantes diferentes, particularmente en los objetos con movimientos rápidos como son las aspas del helicóptero

En años recientes se han dado los primeros pasos para incorporar las particularidades de las cámaras rolling shutter directamente en la formulación de sistemas de odometría visual. En [118] se desarrolla una solución precisa donde tanto el algoritmo de posicionamiento como el de estimación de la profundidad son sustancialmente modificados para adaptarse a cámaras “rolling shutter”. Con ello se obtienen excelentes resultados, con la gran desventaja de que el sistema no se ejecuta en tiempo real, tardando varios segundos en procesar cada imagen.

La última consideración con respecto a la cámara que se quiere hacer notar, la cual surgió durante el desarrollo de las experiencias, es la velocidad de refresco efectiva. Mediante la computadora a bordo Beaglebone Black, que se encarga de procesar y transmitir los datos obtenidos desde la cámara, no es posible transmitir las imágenes a 30hz. Esto se debe a su pequeño poder de cómputo y la velocidad de los periféricos instalados. Efectivamente se tiene una tasa de refresco de 10 cuadros por segundo aproximadamente. Esta consideración marcará el tiempo de muestreo del sistema de control y definirá también la velocidad máxima crucero del submarino autónomo.

### Cámara Secundaria

Resultados encontrados dentro de la bibliografía indican que las distorsiones y retardos descritos en esta sección afectan de una manera particularmente severa a

los sistemas de odometría visual directos [83] [82]. Por este motivo se instaló una segunda cámara a bordo, con las características necesarias para evaluar correctamente el sistema de odometría visual descrito en el capítulo 3.

La cámara secundaria utilizada es fabricada por la empresa GoPro, modelo Hero 4 session. Esta cámara está diseñada específicamente para ser utilizada en ambientes subacuáticas, siendo posible sumergirla hasta 10 metros de profundidad sin necesidad de una carcasa especial. Si bien se puede configurar para obtener una gran variedad de resoluciones de imagen y ángulos de visión, se configuró la cámara para obtener imágenes con resolución de 848x640 píxeles y un ángulo de visión de 170 grados. La tecnología de captura de imagen es también de rolling shutter, pero el sensor de imagen instalado es muy veloz, capaz de tasas de refresco de 120 cuadros por segundo. Por este motivo la distorsión causada por el rolling shutter es considerablemente baja.

Se montó la cámara en el ROV como indica la figura 6.5, y se agregó trozos de material boyante de manera de mantener la flotación neutral del vehículo.

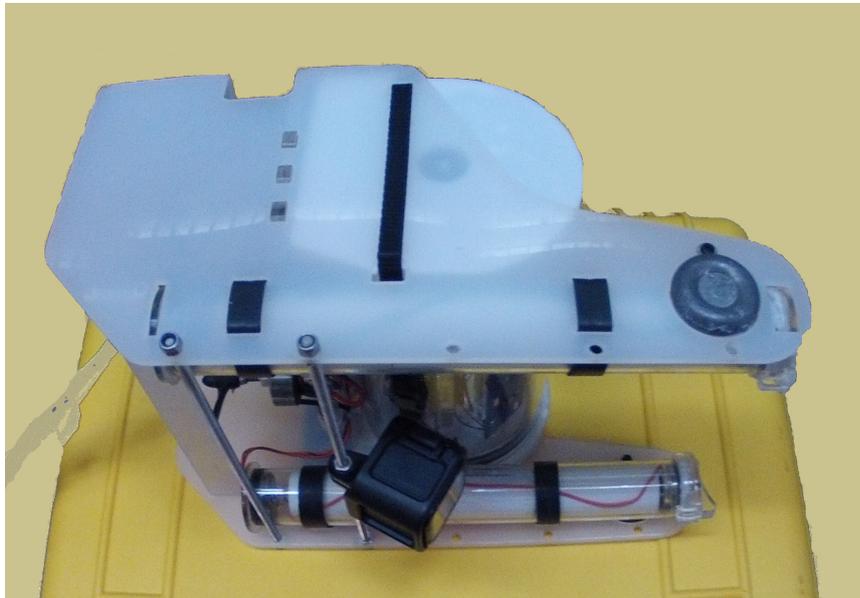


FIGURA 6.5: OpenROV con la cámara secundaria montada

Los resultados de la calibración de la cámara secundaria se aprecian en la figura 6.6. Se puede ver una distorsión mucho menor que la encontrada en la cámara principal. El error de reproyección es de solo 0.277336, sensiblemente menor al encontrado durante la calibración de la cámara principal. El FOV horizontal se mantuvo en 120 grados tanto en aire como debajo del agua.

## 6.2. Escena Submarina

Con el objeto de realizar experimentos en un ambiente controlado y que sean reproducibles bajo las mismas condiciones, se empleó la pileta de experimentaciones que posee el grupo, instalada a la intemperie dentro del predio perteneciente al CONICET Bahía Blanca. La misma posee unas dimensiones de 6.15m x 2.15m, con una profundidad de 0.8 m. Se preparó la escena cubriendo el fondo con arena y colocando rocas de distintos tamaños. Las rocas fueron colocadas de manera aleatoria, pero procurando que el centro de la pileta contenga una mayor cantidad de ellas, de

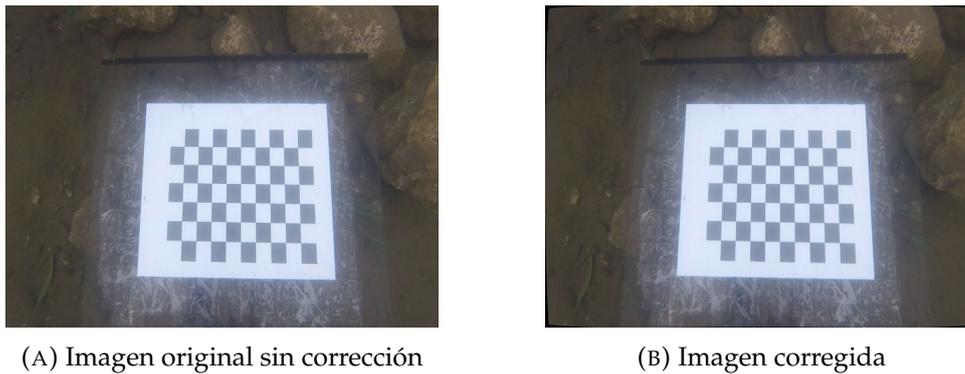


FIGURA 6.6: Distorsión en la cámara secundaria

manera que el sistema de navegación autónomo encuentre una mayor cantidad de características en esta zona y así se evite que el vehículo pueda tocar los bordes de la misma o colisione con ellos. Asimismo, el escenario montado muestra un alto grado de monotonía al repetirse las rocas con cierto patrón, emulando de esta manera un suelo autosímil sin puntos preferenciales.

Al encontrarse a la intemperie, el agua tomó una apariencia naturalmente turbia, aunque este nivel de sedimentos en suspensión podía regularse con tiempo con el empleo de precipitantes. Además, en días soleados y ventosos se pudieron realizar experimentaciones con ondas cáusticas naturales. Una ilustración de apariencias típicas de la escena se observa en la figura 6.7.



FIGURA 6.7: Escena submarina de experimentación

### 6.3. Navegación en Suelo Autosimil

Para evaluar el sistema de guía, navegación y control se configuró el controlador de profundidad de tal manera para que el AUV (autonomous underwater vehicle) mantenga una profundidad media, unos 40 cm debajo de la superficie. Se estableció un ángulo de visualización (tilt) de la cámara primaria y secundaria de 60 grados con respecto a la línea de agua. Para cada serie de experimentos, se comenzó la navegación desde posiciones distintas dentro de la pileta, permitiendo que el AUV realizara la exploración hasta recorrer 3/4 de la extensión de la pileta aproximadamente. También se destaca que el AUV elige por si solo la línea de visión para empezar a explorar dentro de un sector cónico posible. Esta línea de visión coincide con la dirección de mayor densidad de rasgos. Luego de la fase de exploración, se continuó la experiencia hasta obtener 8 revisitas. En 4 de estas revisitas (enumeradas 2, 4, 6 y 8) la dirección de desplazamiento del AUV fue idéntica a la realizada durante la exploración. Por el contrario, en las 4 revisitas restantes (enumeradas 1, 3, 5 y

7) el AUV se desplazó en la dirección contraria, realizando movimientos en reversa. De las distintas experimentaciones realizadas se ilustran aquí cuatro realizaciones representativas.

Se puntualiza que las trayectorias verdaderas son desconocidas ya que como se demostró en el capítulo 2, el sistema de guía conduce al submarino por un camino favorable para el retorno al punto inicial, y ese objetivo lo consigue sin saber a ciencia cierta su ubicación real. Asimismo, las trayectorias estimadas pueden dispersar unas con otras sin poder conocerse tampoco el tenor de esta dispersión que está afectada del error odométrico de navegación por visión y naturalmente por el sistema de control.

La manera más práctica y segura de juzgar el éxito de la navegación por el corredor formado, consiste por un lado en la comparación de todas las trayectorias del corredor bajo un mismo sistema de referencia, y por otro lado, en el enfoque de algunos puntos del corredor incluyendo los extremos revisitados, es decir nodos intermedios y nodos extremos. Se sabe que una pérdida del camino supondría un apartamiento progresivo del submarino respecto al corredor troncal como así también la imposibilidad de visitar el siguiente extremo.

### 6.3.1. Trayectorias

En la figura 6.8 se ilustran las trayectorias estimadas por el sistema de navegación. En azul se ilustra la trayectoria de referencia generada durante la fase de exploración. En rojo se ilustran las distintas revisitas. Todas las trayectorias de cada grupo conforman el corredor, el cual se consolida revisita a revisita con la actualización de mapa estimado del odómetro visual.

Se puede notar que en todas las realizaciones las trayectorias de exploración no tuvieron gran variabilidad, sino que se mantuvieron dentro de una franja angosta. Esto es debido a que el sistema de guía eligió exitosamente para la exploración las zonas con mayor cantidad de características sobresalientes, las cuales conformaron una cadena de nodos que pudo seguirse exitosamente en sentido ascendente y descendente.

### 6.3.2. Evaluación del Éxito Durante las Revisitas

Para poder aseverar el éxito de la navegación, se requiere verificar que el AUV efectivamente haya navegado circundando las mismas posiciones reales durante la fase de exploración. Dado que para esta experimentación no se tienen medidas de la posición real, por la dificultad de obtener esta información en ambientes subacuáticos, se verifico el éxito de la navegación mediante métodos alternativos. Este método consistió en la confirmación visual de la posición real del AUV dentro de la pileta en tres puntos diferentes a lo largo de la trayectoria.

Se eligieron tres posiciones a lo largo de la trayectoria para realizar tal verificación. Precisamente se eligió la posición inicial, una posición en el medio de la trayectoria y la posición final. De esta manera, si las posiciones reales en los tres puntos dentro de la trayectoria mencionados son idénticas, se debería poder confirmar visualmente que las imágenes capturadas en dichas posiciones perteneces efectivamente a la misma escena. En otras palabras, se puede aseverar el éxito de la revisita, es decir, descartar un cruce fortuito, si es posible confirmar que las imágenes capturadas en la misma posición en diferentes revisitas muestran perspectivas similares de la misma posición dentro de la pileta.

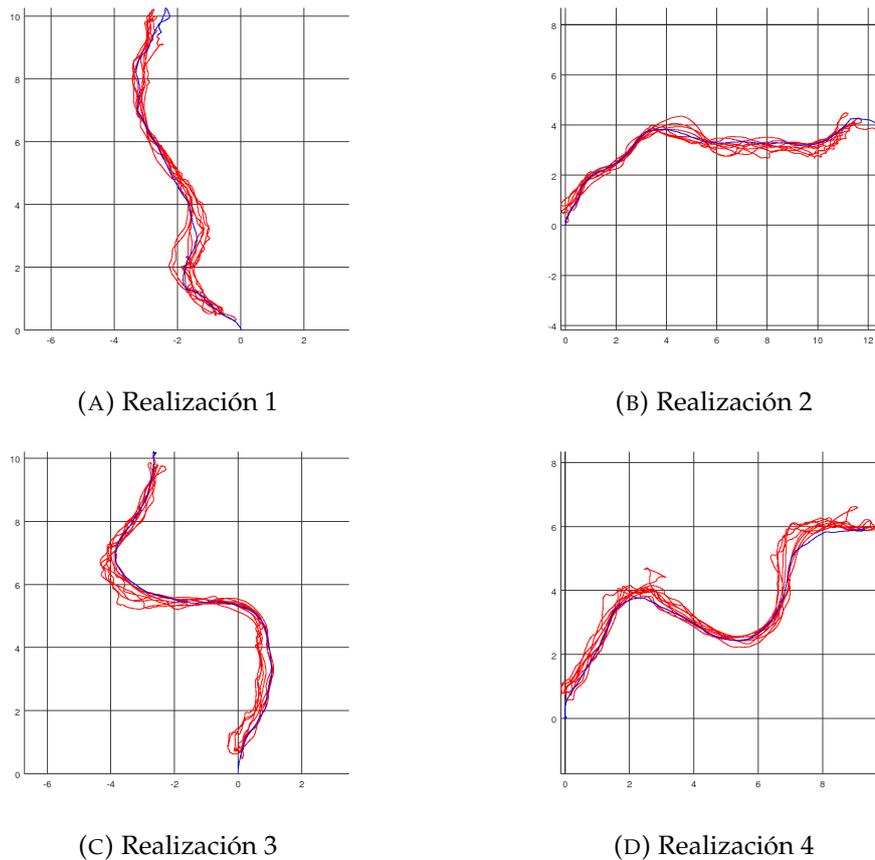
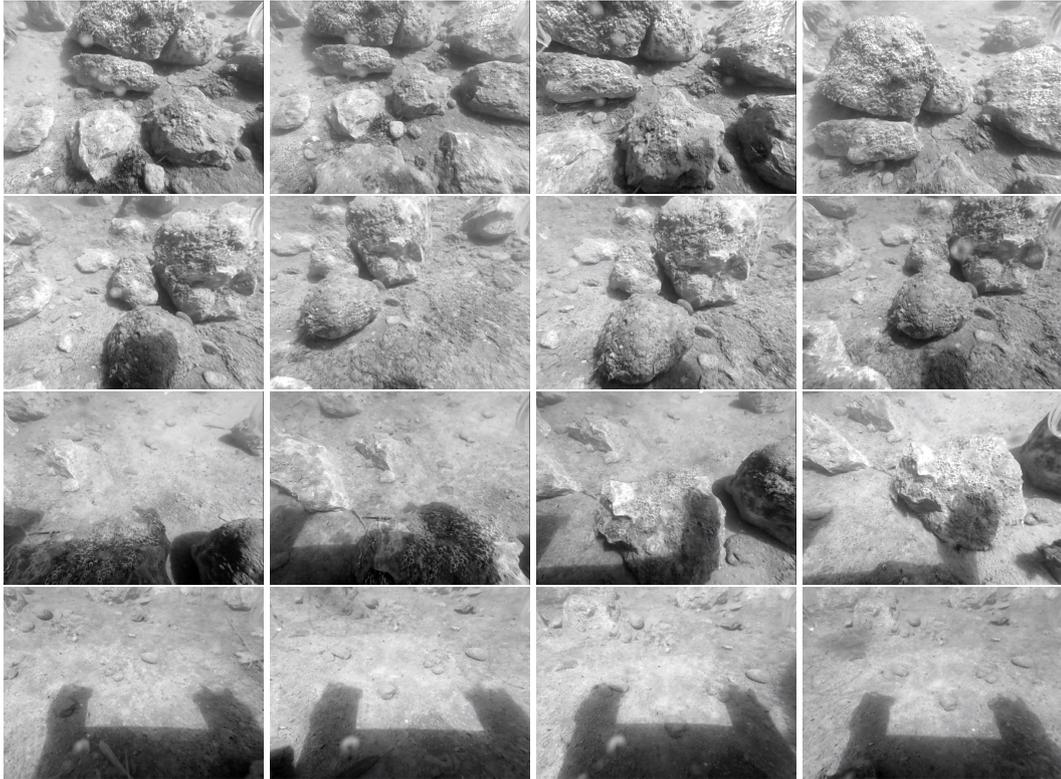


FIGURA 6.8: Trayectorias x-y estimadas para las 4 realizaciones distintas. En azul se representa la trayectoria del submarino autónomo estimada durante la fase de exploración. En rojo se ilustran las distintas revisitas.

En la figura 6.9, se aprecian las imágenes capturadas en la posición inicial. Para las realizaciones 1, 2 y 4, se puede ver claramente que la escena coincide en gran medida, si bien con ligeros cambios en el ángulo de visión. Por ejemplo, en la realización 1 el inicio de la revisita 8 se tiene una posición ligeramente diferente, pero igualmente es posible reconocer visualmente que pertenecen a la misma escena. La realización 3 presenta mayor diferencia en el punto de captura de la imagen que las otras 3 realizaciones. Pero, al igual que en el ejemplo anterior, se puede reconocer claramente que las imágenes pertenecen a la misma escena, por lo que la posición de la cámara desde donde se capturaron las imágenes es muy similar en cada caso.

En la figura 6.10, se ilustran las imágenes capturadas desde la posición final. Al igual que en el caso anterior, se observa que en todas las realizaciones se capturaron imágenes pertenecientes a la misma escena en todas las revisitas, si bien con pequeños cambios de perspectiva, como en la revisita 1 para la realización 1.

La figura 6.11 ilustra las imágenes capturadas en una posición en el medio de la trayectoria. Como antes, en todos los casos las escenas fueron las mismas. Nótese particularmente el caso de la realización 2, donde las imágenes muestran un gran grado de similitud, evidenciando que la posición de captura de dichas imágenes fue prácticamente idéntica.



(A) Inicio revisita 2, (B) Inicio revisita 4, (C) Inicio revisita 6, (D) Inicio revisita 8,  
fin revisita 1                      fin revisita 3                      fin revisita 5                      fin revisita 7

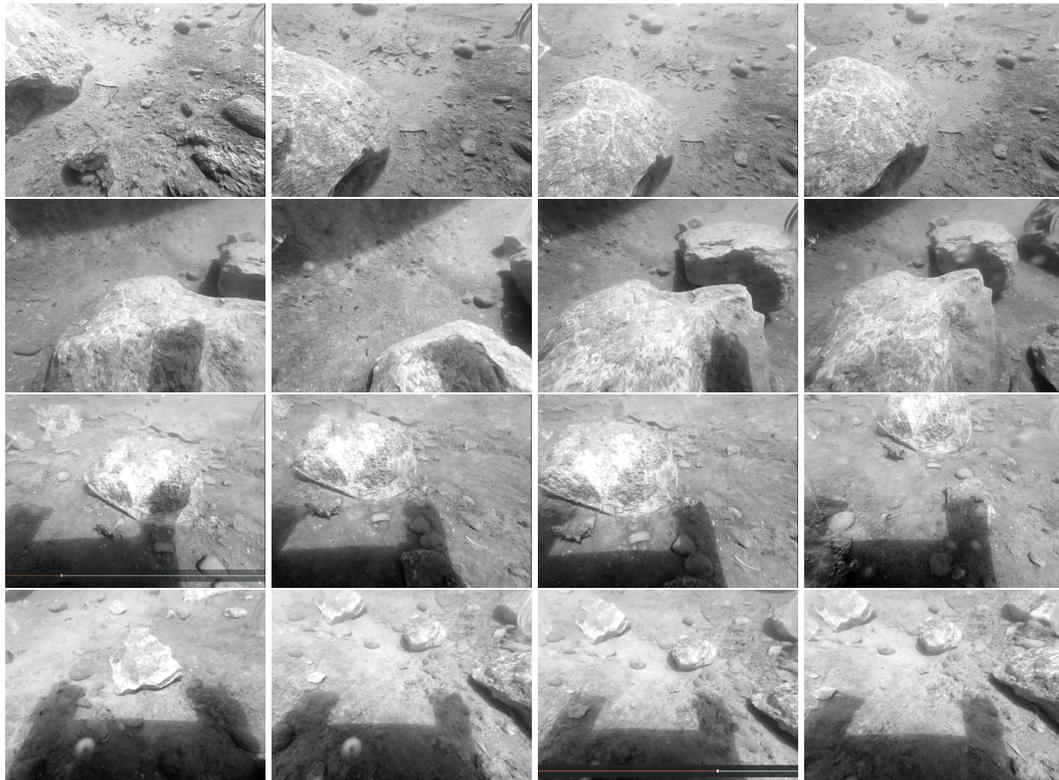
FIGURA 6.9: Imágenes capturadas en la posición inicial. Las filas 1-4 ilustran imágenes capturadas en las realizaciones 1-4, respectivamente. Las columnas ilustran distintos instantes durante la navegación

### 6.3.3. Velocidad de Navegación

Es interesante evaluar la velocidad  $v$  del vehículo. La figura 6.12 ilustra  $u$ ,  $v$  y  $r$  a lo largo de la trayectoria. Se puede observar que  $u$  tiene una magnitud similar tanto al avanzar como durante el retroceso, acusando que parte de la dinámica antisimétrica para avance y retroceso en el ROV utilizado pudo ser compensada.

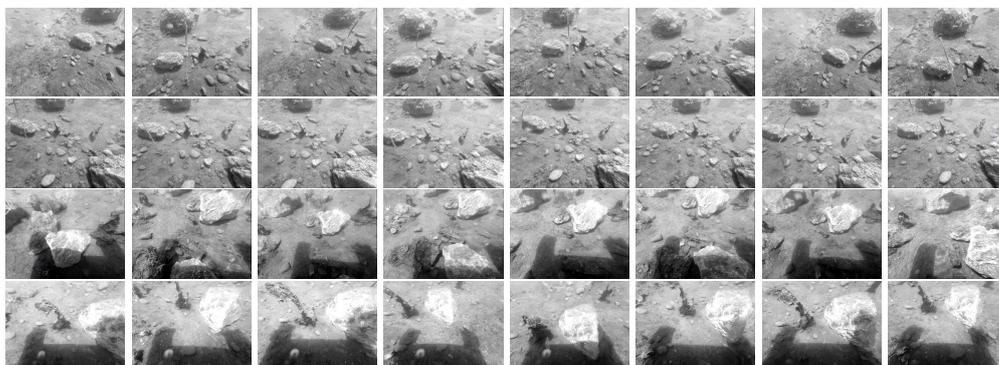
### 6.3.4. Mapa Denso

Las imágenes capturadas durante la navegación mediante la cámara secundaria fueron utilizadas para estimar un mapa denso de la escena por medio del sistema propuesto en el capítulo 3. La figura 6.13 muestra distintas visualizaciones para el mapa obtenido utilizando las imágenes capturadas durante la realización 2. Se puede observar que el mapa representa un corredor, cuyas curvas y contracurvas coinciden con la trayectoria del vehículo estimada durante la navegación, como se desprende de la figura 6.8. La visualización de las distintas perspectivas del mapa denso permite apreciar un gran detalle del suelo subacuático, como son las distintas rocas encontradas a lo largo de la trayectoria. Así este mapa sería de gran utilidad para estimar una trayectoria libre de peligros de colisiones.



(A) Inicio revisita 1, (B) Inicio revisita 3, (C) Inicio revisita 5, (D) Inicio revisita 7,  
fin exploración fin revisita 2 fin revisita 4 fin revisita 6

FIGURA 6.10: Imágenes capturadas en la posición final. Las filas 1-4 ilustran imágenes capturadas en las realizaciones 1-4, respectivamente. Las columnas ilustran distintos instantes durante la navegación



(A) Revi- (B) Revi- (C) Revi- (D) Revi- (E) Revi- (F) Revi- (G) Revi- (H) Revi-  
sita 1 sita 2 sita 3 sita 4 sita 5 sita 6 sita 7 sita 8

FIGURA 6.11: Imágenes capturadas en una posición en el medio de la trayectoria. Las filas 1-4 ilustran imágenes capturadas en las realizaciones 1-4, respectivamente. Las columnas ilustran distintos instantes durante la navegación

## 6.4. Filtrado de Onda Cáustica

En esta sección se evaluará la utilidad del filtro de onda cáustica desarrollado en el capítulo 4 para evitar el deterioro del rendimiento de sistemas SLAM visual

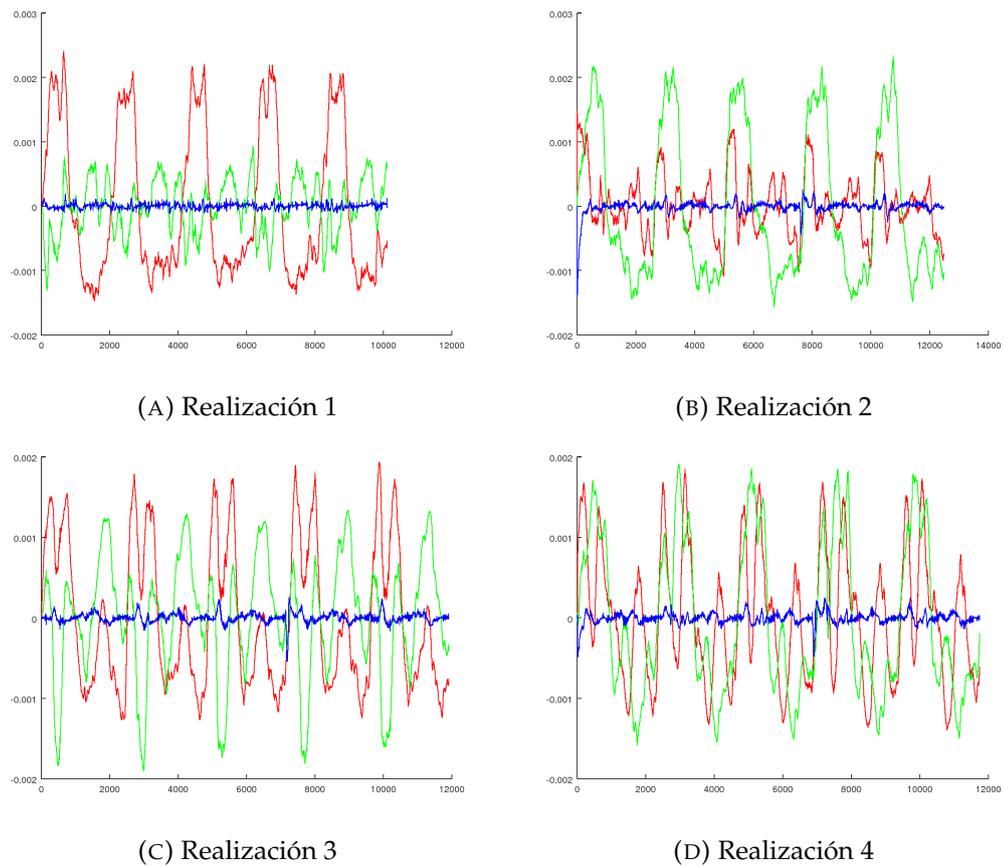


FIGURA 6.12: Velocidad del AUV a lo largo de la experimentación.  
En rojo  $u$ , en verde  $r$  y en azul  $v$

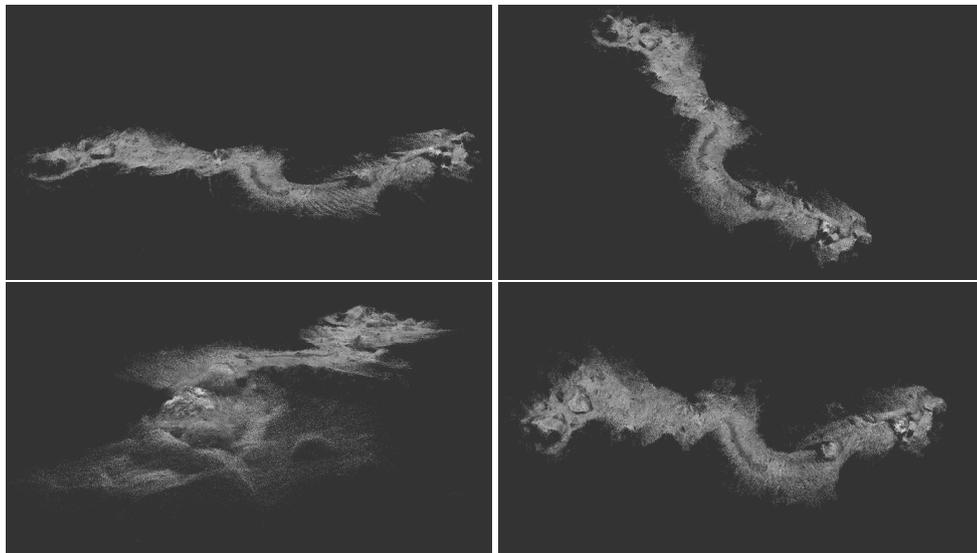


FIGURA 6.13: Distintas visualizaciones del mapa denso estimado para la realización 2.

bajo dicho tipo de perturbaciones. Se propuso evaluar la funcionalidad bajo tres sistemas SLAM visual diferentes. Se probó en dos sistemas ampliamente citados en el estado de arte, ORB-SLAM [13] como representante de los sistemas indirectos, y

LSD-SLAM [14] como representante de los sistemas directos y finalmente se probó la funcionalidad del filtro bajo el sistema desarrollado en el capítulo 3. Con respecto a la experimentación, se controló el AUV de forma de obtener un movimiento rectilíneo controlado hacia adelante, a lo largo de la piletta, de manera de que cualquier error en la estimación odométrica de la posición se reflejara como un corrimiento de la trayectoria rectilínea preestablecida. Se utilizó la cámara secundaria, configurada para obtener un ángulo de visualización de 60 grados con respecto al suelo. La experimentación fue realizada en un día soleado con fuertes vientos, por lo se capturaron fuertes perturbaciones lumínicas en las imágenes.

#### 6.4.1. Sistema Indirecto ORB-SLAM

En primer lugar, se evaluó el desempeño de ORB-SLAM sin la utilización del filtro de ondas cáusticas. De esas experimentaciones se notó que este sistema es altamente robusto hasta aún las perturbaciones lumínicas más fuertes que se pudo encontrar en la experimentación particular presente. Es posible especular que esta robustez sea debida en parte a la buena elección del detector y descriptor de características sobresalientes, que permite reconocer el mismo punto físico aún ante cambios de iluminación tan drásticos como los presentados en esta experimentación. En la figura 6.14 se pueden apreciar dos imágenes capturadas por la cámara en instantes diferentes, los keypoints detectados por el sistema ORB-SLAM, y se observa también una ilustración de la trayectoria estimada, que en todo momento fue rectilínea reflejando el movimiento real mantenido por el ROV.

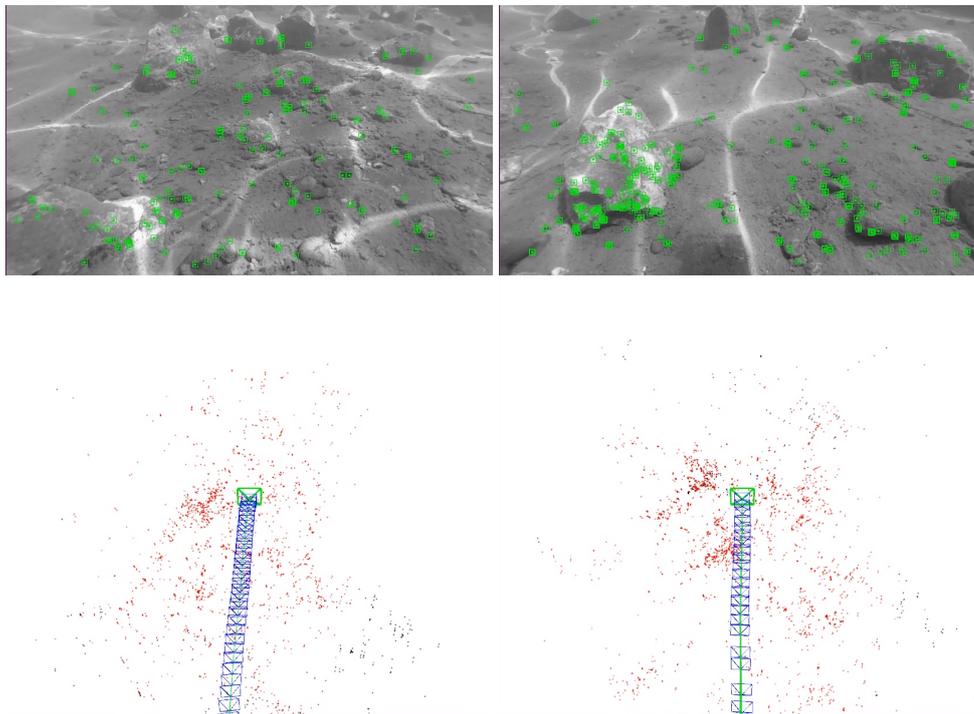


FIGURA 6.14: Desempeño de ORB-SLAM bajo perturbaciones cáusticas. Se puede observar en las figuras de la fila superior, la escena perturbada mediante ondas cáusticas en conjunto con los keypoints detectados por el método. En la fila inferior se observa la trayectoria estimada. Nótese que la trayectoria es rectilínea, lo que indica que la estimación odométrica fue correcta, aún en presencia de la perturbación lumínica.

De esta manera, la utilización del filtro de onda cáustica en este sistema no es necesaria, por lo que se prosiguió en la evaluación de los dos sistemas de SLAM visual restantes.

#### 6.4.2. Sistemas Directos

Por otro lado, el rendimiento de los sistemas directos evaluados, a saber LSD-SLAM y el desarrollado en el capítulo 3 fue severamente deteriorado por la presencia de ondas cáusticas. La figura 6.15 ilustra la severidad de la desmejora en la estimación de la profundidad de la escena. En rojo se codifican profundidades cercanas al punto focal de la cámara, mientras que en azul se representan los puntos alejados. Justamente las ondas cáusticas son una perturbación lumínica, que afecta espacial y temporalmente el brillo de la escena detectado por la cámara. Ya que estos métodos utilizan la información fotométrica hallada en las imágenes, de manera directa y sin realizar descripciones abstractas de alto nivel, resultan ser justamente estos métodos los más afectados por este tipo de perturbación. De manera similar aunque menos dañina para la estimación, son las sombras esporádicas del mismo submarino captadas en la imagen que falsean el brillo a la velocidad del vehículo.

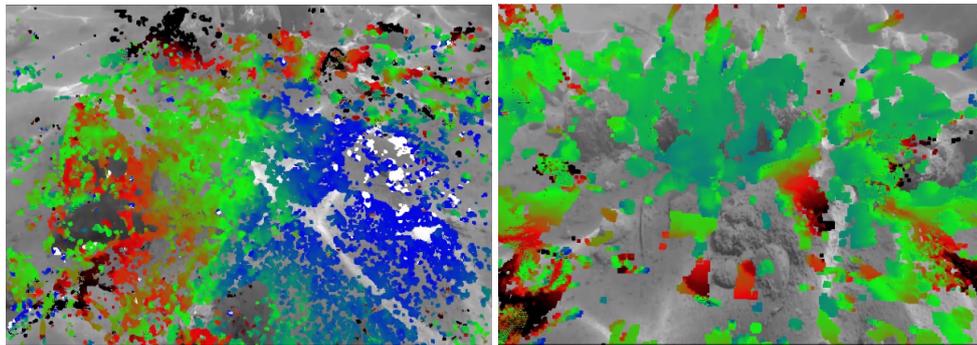


FIGURA 6.15: Desempeño de LSD-SLAM y del sistema propuesto bajo perturbaciones cáusticas. A la izquierda se muestra la estimación de profundidad realizada por LSD-SLAM, y a la derecha se muestra la estimación del sistema propuesto en el capítulo 3. Se codifica en rojo las profundidades cercanas y en azul las lejanas. Como la cámara se encuentra a 60 grados con respecto al suelo, los píxeles en la zona superior deberían tener mayor profundidad que los de la zona inferior. Ninguno de los dos métodos logra una estimación correcta.

Se procedió entonces a evaluar el desempeño de las estimaciones obtenidas por LSD-SLAM y el sistema desarrollado en 3. La imagen cruda capturada por la cámara fue pre-procesada utilizando el filtro de onda cáustica, para luego utilizar la imagen filtrada como entrada para los sistemas de SLAM mencionados.

En la figura 6.16, se ilustran resultados del filtro de onda cáustica en conjunto con la estimación de profundidad realizada por el sistema LSD-SLAM. Se puede observar una mejora sustancial en la estimación de la profundidad de la escena cuando se utiliza la imagen filtrada, al punto que no se reconocen visualmente píxeles con profundidades incorrectamente estimadas. También se puede observar el gran nivel de atenuación que sufrió la onda cáustica en la imagen filtrada, dejando intacta la nitidez de la imagen.

En la figura 6.17 se aprecia la trayectoria y el mapa denso estimado durante el recorrido realizado por el AUV. Se puede observar el movimiento rectilíneo del AUV correctamente estimado, en conjunto con el mapa denso de la escena, en el cual se

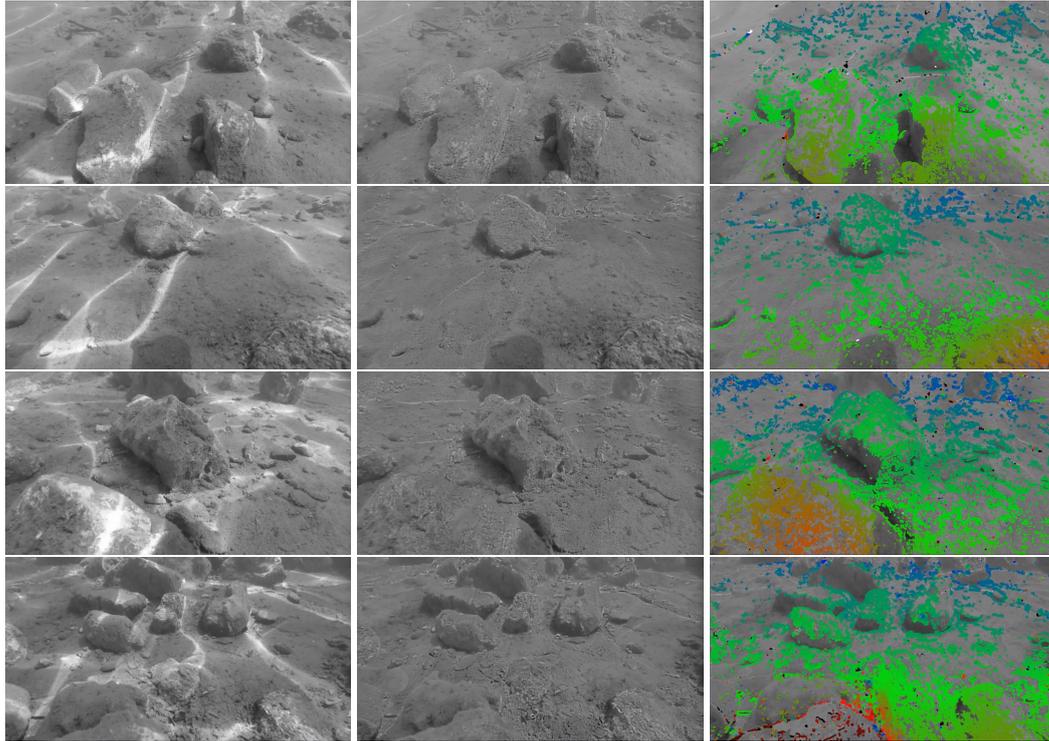


FIGURA 6.16: Desempeño del sistema LSD-SLAM bajo perturbaciones cáusticas, utilizando el filtro propuesto. Las filas muestran 4 instantes diferentes durante la experimentación. La primer columna muestra la imagen cruda capturada por la cámara, La segunda columna ilustra la imagen procesada utilizando el filtro de onda cáustica, y por ultimo la tercer columna muestra la estimación de profundidad realizada por el sistema SLAM.

puede observar detalles como las distintas rocas encontradas durante la trayectoria del vehículo.

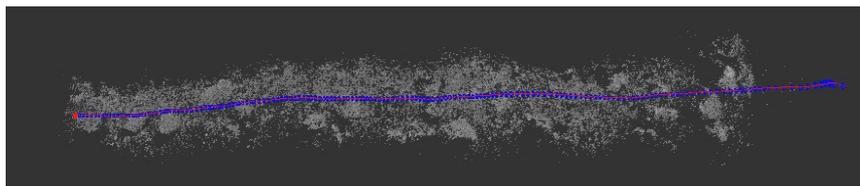


FIGURA 6.17: Mapa del suelo perturbado con onda cáustica, estimado por el sistema LSD-SLAM utilizando el filtro de onda cáustica propuesto

A continuación, se probó el rendimiento del sistema propuesto en el capítulo 3 utilizando como entrada la imagen pre-procesada proveniente del filtro de onda cáustica. Se puede ver nuevamente que la estimación de profundidad recupera su apariencia correcta cuando se utiliza el filtro. En este caso se puede observar que la estimación de la profundidad de la escena no fue completamente densa. Varios estimaciones realizadas en determinados píxeles fueron rechazadas automáticamente por el sistema, considerándolas potencialmente incorrectas. Claramente esto se debe a que el filtro no pudo eliminar completamente la onda caustica. Así, varias estimaciones contenían todavía ligeras perturbaciones lumínicas que afectaron la habilidad

de estimación de profundidad del sistema propuesto en 3. Aún así, se logró estimación de profundidad en la mayoría de los píxeles encontrados en la escena.

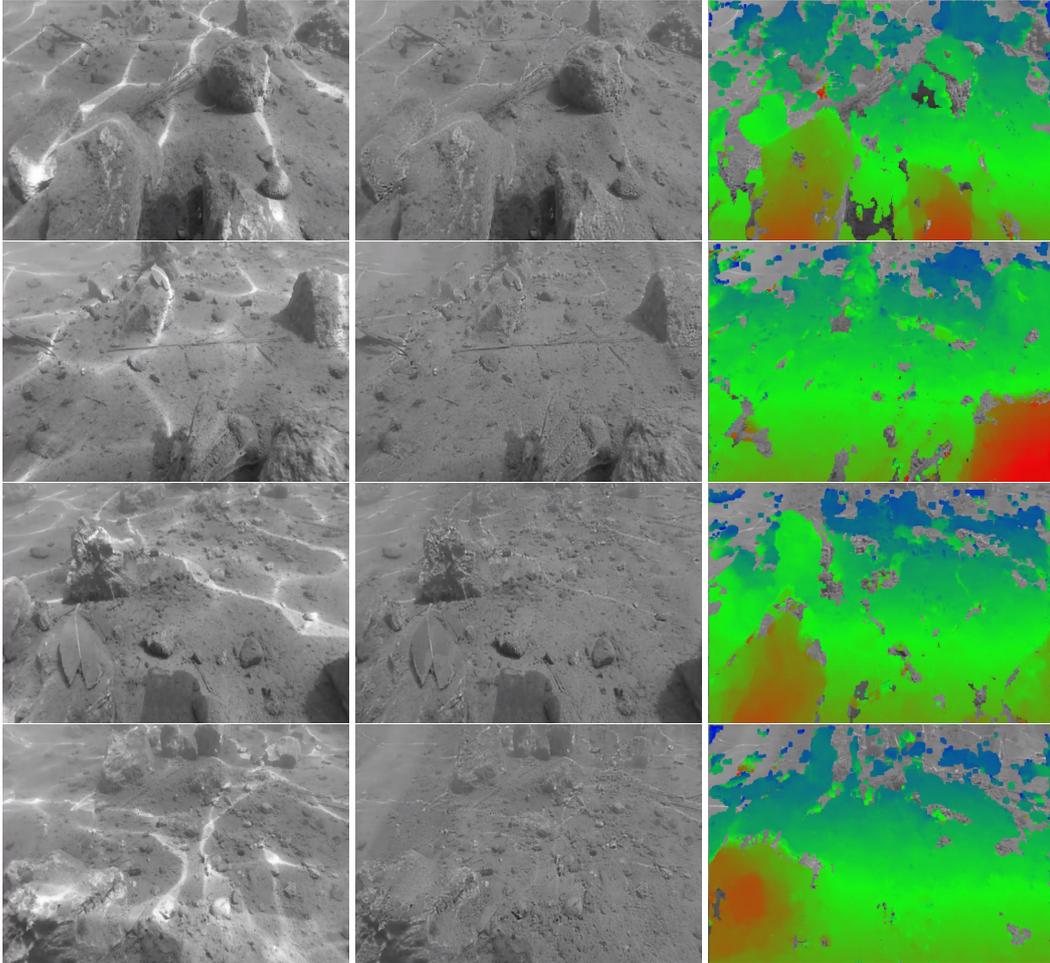


FIGURA 6.18: Desempeño del sistema desarrollado en el capítulo 3 bajo perturbaciones cáusticas, utilizando el filtro propuesto. Las filas muestran 4 instantes diferentes durante la experimentación. La primera columna muestra la imagen cruda capturada por la cámara, la segunda columna ilustra la imagen procesada utilizando el filtro de onda cáustica, y por último la tercer columna muestra la estimación de profundidad realizada por el sistema SLAM.

La figura 6.19 ilustra el mapa estimado. Nuevamente, se observa una apariencia correcta, pudiéndose observar detalles en el suelo como las rocas. No obstante se puede visualizar que el este mapa contiene más información que el obtenido utilizando el sistema LSD-SLAM, ya que las estimaciones de profundidad abarcaron más píxeles.

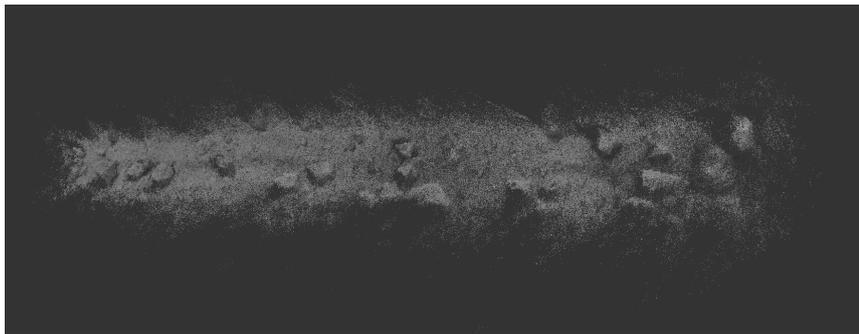


FIGURA 6.19: Mapa del suelo perturbado con onda cáustica, estimado por el sistema desarrollado en el capítulo 3 utilizando el filtro de onda cáustica propuesto



## Capítulo 7

# Conclusiones

La navegación autónoma con visión artificial en ambientes subacuáticos presenta dificultades propias que no se encuentran en su contraparte de la robótica terrestre. La baja visibilidad, escenas no estructuradas repetidas y altas perturbaciones lumínicas representan un desafío para los algoritmos del estado del arte actual.

Esta tesis enfocó particularmente el objetivo de exploración del fondo oceánico con una estrategia basada en la construcción de corredores para la navegación segura en ambas direcciones con la posibilidad potencial de bifurcar en algún punto del corredor troncal formado una extensa red para exploración del suelo. El objetivo se logra con una estrategia que conjuga tres elementos esenciales, a saber: un sistema de navegación, un sistema de guía y un sistema de control adaptivo. En los dos primeros elementos es donde se despliega la mayor parte de soluciones originales, siendo el conjunto completo también una propuesta elaborada en esta tesis.

Dado que una parte significativa del sistema visual propuesto se basa en fotometría, una escena con altas perturbaciones espacio-temporales de ondas cáusticas de luz solar cambia la constancia de brillo enormemente. Por ello, el primer eslabón del sistema consistió en el desflecado de ondas en las imágenes sin afectar la calidad de las mismas. La heurística seguida se fundamentó en estimar la onda cáustica pura como comparación entre la imagen perturbada y una predicción de un paso de la imagen nítida, en el contexto de un lazo de control realimentado. La idea fundamental consistió en la separación de áreas nítidas y áreas de píxeles con brillo incierto mediante binarización. El rastreo de cada píxel alterado en una secuencia de imágenes consecutivas permitió estimar el brillo más aproximado para él en media cuadrática. La calidad sin embargo, dependía de un umbral de binarización adecuado. La solución propuesta se basó en la observación de que los cambios del gradiente en los contornos de los flecos de la imagen filtrada alcanzaban su mínimo para un valor óptimo de umbral de binarización. Así basado en un funcional de costo de una norma sobre la segunda derivada espacial de la imagen, el umbral se autoajustaba al valor mínimo del costo, ofreciendo un excelente filtrado en tiempo real aún en la escenas con profundidades muy disímiles en ambientes 3D.

Teniendo en cuenta las particularidades de una exploración subacuática autónoma, se optó por la descripción de la apariencia del suelo más que por su relieve para delinear el rumbo del corredor. No obstante la baja visibilidad del medio (emulada en las ambientaciones experimentales) obligaba a una navegación a baja altura y con ello el relieve adquiriría un aspecto tridimensional exigiendo al sistema de control suficiente rapidez y maniobrabilidad para evitar eventuales colisiones. Por consiguiente se concluyó que, en este contexto, el mapeo más apropiado para la odometría visual propuesta debía ser doble, a saber: de modelo denso para captar la geometría volumétrica del suelo y ralo para describir a éste con rasgos robustos de apariencia. Aún cuando en la experimentación se evidenció un buen mapeo y seguimiento con ambos modelos, la sensibilidad de SLAM denso a perturbaciones de brillo con onda

cáustica intensa fue mayor que en el SLAM basado en rasgos, aunque con el filtrado previo de la perturbación cáustica las diferencias se allanaban. En las pruebas se contrastaron métodos consolidados de reconocida superioridad en desempeño como el ORB-SLAM de modelo no-denso, el LSD-SLAM de modelo semi-denso junto a un método novedoso surgido en la tesis de SLAM denso basado en volumen de costo de errores fotométricos. También estas tres metodologías fueron probadas en ambientes simulados en su fase de reconocimiento de escenas anteriores (cierres de lazo o revisitas), de los cuales sólo ORB-SLAM y el método denso propio mostraron robustez. Por ello se incorporan ambos a la odometría visual de doble modelo descripta para realizar la navegación autónoma propuesta por corredor.

El método denso propuesto se caracteriza por usar eficientemente toda la información fotométrica disponible en las transiciones de un keyframe a otro durante el mapeo. La idea fundamental se basa en re proyectar información fotométrica de zonas físicas capturadas anteriormente que aún permanecen en la siguiente escena, ganando en general mayor paralaje y por ende mayor exactitud en la estimación de profundidad. También el método logra eliminar los típicos errores de cuantificación de los modelos densos, siendo el relieve estimado más regular.

El problema de la escala en modelos densos obtenido de imágenes monoculares se trató de resolver con algoritmos capaces de recuperar la escala métrica de la escena, con solamente adicionar al sistema de cámara monocular dos láser de bajo costo y bajo consumo y medir indirectamente en la imagen la distancia en algunos puntos claves.

El sistema de guía propuesto para la navegación autónoma constituye uno de los pilares más importantes de la tesis. El mismo sigue una heurística lógica desarrollada y mejorada a lo largo de la tesis. La formación de un corredor es la solución más razonable en ambientes desconocidos con alto riesgo de pérdida del vehículo. Por ello la heurística propone crear un camino óptimo basado en la línea de máxima densidad de rasgos robustos (keypoints) con la idea de poder retornar siguiendo el mismo camino. Por consiguiente, la referencia de trayectoria se crea nodo a nodo con segmentos inter-nodales marcando ese rumbo. A su vez ese camino notorio en su textura es favorable para permitir un buen mapeo denso simultáneo. La cinemática propuesta por el sistema de guía sobre la trayectoria de referencia dependerá de la cautela sugerida mediante el mapa denso para evitar una colisión con objetos del fondo en distancias próximas.

Teniendo en cuenta la autonomía energética limitada del submarino, es que el radio de exploración es consecuentemente limitado y con ello la longitud del corredor máxima debe definirse en el punto de no-retorno. Esto implica la supervisión continua de carga de baterías y su reserva para identificar prematuramente este punto.

Se recalca que la exploración unidimensional propuesta (es decir, a lo largo de una curva en el espacio) tiene asegurada los cierres de lazo en cada nodo de manera robusta pues entre ellos el error odométrico es mínimo. De esta manera la navegación por corredor se basa en el seguimiento ordenado (y estricto) de una secuencia de nodos en sentido creciente y decreciente (retorno). Esta ventaja contrasta con la exploración bidimensional según grillas u otro método de búsqueda extensiva, donde los cierres de lazos pueden convertirse en poco menos que improbables por el gran error odométrico acumulado, aumentando de esta manera la pérdida del vehículo.

La ampliación potencial del mapa a otra dimensión pero conservando el método C-SLAM propuesto, sugiere la posibilidad de bifurcar del corredor troncal en algún punto sugerido por el mapa denso con un relieve promisorio para su reconocimiento. En este sentido el sistema de guía comienza la construcción de una red sobre el

terreno. Cabe destacar que C-SLAM no provee coordenadas globales, por ello la ubicación precisa del corredor como su forma son desconocidas debido a la distorsión de escala métrica y los errores odométricos acumulativos de carácter estocástico. El único objetivo de C-SLAM es posibilitar la ida y la vuelta en forma segura con fines de exploración.

El último eslabón en la estrategia es el sistema de control. La ventaja de un control adaptivo para ejecutar el C-SLAM, emerge de manera natural si se tiene en cuenta la complejidad en el modelado de un sistema hidrodinámico, la dificultad de modelar incertidumbres dinámicas y la eventual variación temporal de algunos parámetros físicos del sistema submarino autónomo. Todo esto se resuelve de manera simple con la aplicación de un control adaptivo que disminuye el efecto de las incertidumbres y se adapta a la nueva dinámica manteniendo su desempeño. El método de diseño empleado corresponde al principio de gradiente de velocidad, el cual se adaptó apropiadamente a las referencias geométricas y cinemáticas generadas por el sistema de visión. Se pone énfasis que el requisito de obtener una alta maniobrabilidad con el control es indispensable en el concepto de C-SLAM. De lo contrario, grandes errores de posicionamiento del vehículo respecto a las referencias marcadas por el sistema de guía, colocarán a la cámara en otra línea de visión con la posibilidad de enfocar otra escena y sacar al vehículo del corredor. Según los estudios previos con otros controladores robustos como PID y de modo deslizante, el control adaptivo de gradiente de velocidad ofreció una marcada superioridad en desempeño y flexibilidad de diseño.

El volumen de cálculo que exige la odometría visual de doble modelo y el filtro de onda cáustica es usualmente elevado y crece geoméricamente con el tamaño de la imagen. Los elementos de cálculo exigen paralelismo para lograr tiempos de muestreo en el control al menos de 10 imágenes procesadas por segundo. Esto impone hardware de CPU con núcleos cuádruples, GPGPU o FPGA. En esta tesis se propuso la implementación de FPGA para el módulo más demandante que es el estimador de profundidad denso. Las pruebas se hicieron fuera de línea logrando tiempos comparativamente bajos en relación a estructuras de CPU.

Todos los sistemas fueron extensamente evaluados primeramente con simulaciones en una plataforma GAZEBO especialmente empleada en robótica, con la programación de un escenario submarino de fondo. También se emplearon datases reales de la comunidad SLAM. Por último, lo más importante, las pruebas experimentales en piletta escenificada apropiadamente que emula las situaciones más difíciles que pueden darse en misiones reales respecto a la baja visibilidad por turbidez, el relieve auto-símil y las ondas cáusticas. Se empleó un vehículo submarino OpenROV con cámara monocular y un hardware de multi-núcleo ubicado en tierra que procesa la información de imágenes recibidas de submarino y envía los comandos del control adaptivo por cable. De esta manera se logró la autonomía completa del vehículo en el esquema de control adaptivo de visión artificial con estrategia C-SLAM.

A la luz de los resultados, podemos concluir que la hipótesis de trabajo de esta tesis queda probada. Los sistemas basados en visión en ambientes subacuáticos no solamente son posibles de implementar, sino que presentan una excelente alternativa a los sistemas más utilizados históricamente, como aquellos basados en sonares, DVLS e IMUS.

Durante el transcurso de esta tesis surgieron conclusiones sobre mejoras futuras del C-SLAM. Parte de ellas se empezaron a investigar actualmente, por ejemplo el empleo de multicámaras en estéreo orientadas hacia el suelo en ambas direcciones de la trayectoria y transversales a ella, resuelven el problema de escalamiento del

mapa y proveen de más alternativas al sistema de guía. El aumento del procesamiento sugiere el empleo de procesadores orientados a imágenes como tecnología GPGPU, la cual además es más fácil de programar que las estructuras FPGA. En los últimos años aparecieron módulos de pequeño tamaño y bajo consumo de energía, como las placas de desarrollo jetson TX1 [119], contando con un procesador de cuatro núcleos ARM de altas prestaciones y una GPGPU de Nvidia basada en la arquitectura MAXWELL. Por otro lado se conjetura que un sistema de C-SLAM completamente denso puede tener ventajas sobre los modelos basados en keypoints, sobre todo al retornar por el corredor. Esto no fue probado definitivamente en la tesis por problemas tecnológicos en el tipo de captura de cámara. Sin embargo, existen evidencias de que estas ventajas podrían cristalizarse con cámaras con propiedad de captura instantánea conocida como *global shutter*. También se estima la posibilidad de enriquecer estas técnicas particulares con una solución para la no-constancia de brillo, utilizando técnicas de estimación de BRDF [120], cuyo logro representaría una mejora significativa en el estado del arte de la odometría visual.

Estos temas aún abiertos seguramente potenciarán el concepto C-SLAM de visión para la exploración subacuática.

## Apéndice A

# Implementación en Paralelo

### A.1. Introducción

Un verdadero sistema autónomo requiere poder ejecutarse en tiempo real. En la literatura, normalmente se entiende tiempo real como una tasa de actualización de 30Hz. Los algoritmos desarrollados en los capítulos 2, 3 y 4 son especialmente complejos, y esfuerzos son necesarios para lograr la tasa de refresco requerida.

Para acelerar este tipo de sistemas, se encuentran en la literatura trabajos como [121], donde se reduce el requerimiento de cómputo en un sistema SLAM visual diseñado para funcionar bajo el agua, seleccionando cuidadosamente el momento para agregar un nuevo keyframe, y estimando el momento en que se puede producir un cerrado de lazo. De esa manera logran reducir la utilización del procesador a bordo en lo que respecta al back-end (véase el capítulo 2). Por otro lado, en trabajos como [84] [14] se hace notar que no es necesario ejecutar el back-end en tiempo real para obtener buenos resultados de localización.

En cuanto al front-end, las soluciones que se encuentran generalmente en la literatura son, o bien utilizar atajos para limitar la cantidad de información que se procesa [14], obteniendo así un sistema capaz que ejecutarse en tiempo real, o utilizar coprocesadores para acelerar el procesamiento, basados en GPGPU [90]. Si bien en años recientes hubo grandes avances hacia la disminución del tamaño y el consumo de energía de los GPGPU, estos sistemas no están diseñados para resolver específicamente este tipo de problemas, por lo que es valioso explorar soluciones diferentes.

Una posible alternativa es la utilización de circuitos reconfigurables, como FPGA. Al sintetizar un procesador hecho a medida de la aplicación, las implementaciones basadas en FPGA podrían ser rápidas, con un bajo consumo de energía y con un tamaño más apropiado para incluirse dentro de un vehículo autónomo.

Las implementaciones en FPGA son relativamente novedosas en la bibliografía. En [122], se acelera un sistema odométrico directo, basado en el trabajado de [14]. No se logra llegar a velocidades de ejecución de tiempo real, y la estimación de la profundidad es semi-densa, por lo que no estima la profundidad para todos los píxeles, sino solo algunos selectos. En [123], por otro lado, se diseña un sistema en FPGA para acelerar no el front-end, sino el back-end de un sistema SLAM, basado en [124]. Estos sistemas poseen un buen rendimiento en procesadores clásicos, por lo que la utilidad de una implementación en procesadores a medida es discutible.

En este capítulo se explora la utilización de FPGA para la aceleración de un algoritmo de odometría visual similar al desarrollado en el capítulo 3. El estudio se limitará a las posibles mejoras en velocidad de cálculo, dejando un análisis del consumo de energía para trabajos futuros. En primer lugar se analizó detalladamente el funcionamiento de un sistema de odometría visual denso [90]. Se identificaron los

subsistemas que son computacionalmente más costosos, y se realizó una implementación en FPGA de los mismos. Se probó la implementación con datos obtenidos de simulación. Se muestran resultados comparativos con implementaciones en CPU que muestran las mejoras posibles que otorga la implementación en FPGA.

## A.2. Sistema Propuesto

### A.2.1. Análisis Previo

En primer lugar, se hará un análisis del algoritmo de odometría visual denso seleccionado, para identificar los subsistemas que necesitan ser acelerados.

#### Funcionamiento General

En general, los sistemas de odometría visual están compuestos por dos subsistemas. El módulo de localización se encarga de utilizar los últimos datos provistos por el sensor para encontrar la posición de la cámara relativa al mapa estimado hasta el momento. Por otro lado, el módulo mapeador utiliza la información provista por el localizador para estimar el mapa de la zona explorada. La figura A.1 esquematiza un sistema de odometría visual típico. Se observa en la figura que entre los dos subsistemas existe un lazo de realimentación. Esto indica

que tanto la localización como el mapeo se deben realizar simultáneamente. Para más detalles del funcionamiento de los sistemas de odometría visual, véase el capítulo 3.

#### Definiciones

Sea  $K \in \mathbb{R}^{3 \times 3}$  la matriz intrínseca y  $T \in \mathbb{R}^{4 \times 4}$  la matriz extrínseca de la cámara en relación a una postura de referencia  $T_{kf}$ . Sea  $\pi$  una función de proyección de un punto  $p \in \mathbb{R}^3$  a un punto  $u \in \mathbb{R}^2$ , respondiendo a:  $\pi(x, y, z) = (x/z, y/z)$ . Sea  $I$  una imagen compuesta por píxeles con coordenadas  $u \in \Omega$ . Sea  $E$  un conjunto de puntos  $p \in \mathbb{R}^3$  que conforman la estimación del mapa hasta la muestra actual. Por último  $d$  será la distancia de un píxel  $u$  al punto focal de la cámara.

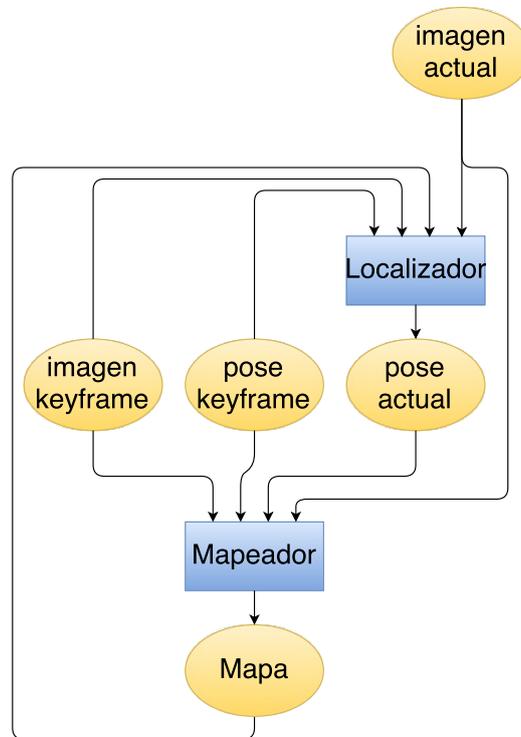


FIGURA A.1: Diagrama en bloques DTAM

### Módulo Localizador

El funcionamiento del módulo localizador consiste en encontrar la matriz  $T \in \mathbb{R}^{4 \times 4}$  tal que minimice la función [92]:

$$\begin{aligned} F(T) &= K\pi(Tp) - I_a(u) \\ \forall u \in \Omega, \forall p \in E \end{aligned} \quad (\text{A.1})$$

donde  $I_a$  es la imagen capturada por la cámara en la postura  $T$  a determinar. La resolución a estos problemas generalmente toma la forma de una solución clásica a problemas de mínimos cuadrados con mejoras iterativas [125].

### Módulo Mapeador

Para el módulo mapeador, se utilizará como referencia la implementación de DTAM [90]. El funcionamiento de la misma está esquematizado en la figura A.2.

En primer lugar se toma una imagen de referencia llamada “key-frame”, la cual servirá de punto base y a la cual se le estimará la profundidad de todos sus píxeles. Luego se genera incrementalmente un volumen de costo que concentra la información de similitud entre los píxeles del key-frame y todas las imágenes capturadas por la cámara (en un lapso de tiempo). Este módulo se explicará en detalle en secciones subsiguientes. A continuación, la información

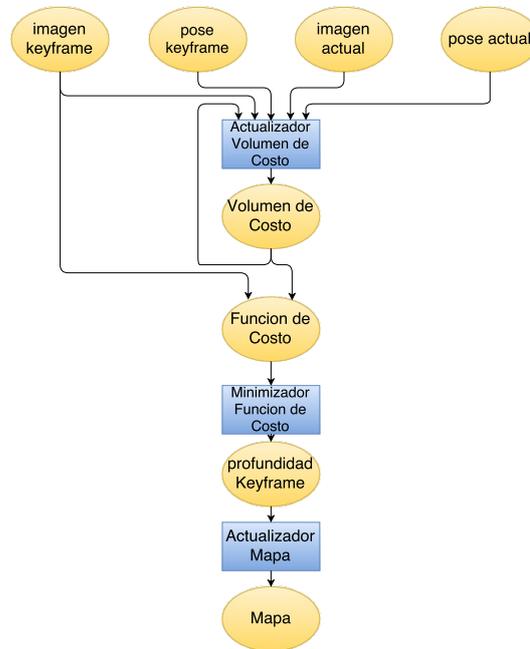


FIGURA A.2: Diagrama en bloques del módulo Mapeador

del volumen de costo se complementa con información de la imagen del key-frame para generar una función de costo, cuyo mínimo se espera corresponda con la profundidad de los píxeles del key-frame. Este mínimo se mejora incrementalmente con cada nueva imagen capturada por la cámara, hasta que se necesite capturar un nuevo keyframe. En ese momento, la información de profundidad se guarda en el mapa de la zona, se toma una nueva imagen de referencia y el proceso comienza nuevamente.

### Módulo de Acumulación del Volumen de Costo

El volumen de costo es una función  $F(u, d)$ , la cual acumula información sobre las posibles profundidades que pueden poseer los píxeles pertenecientes al keyframe. Siguiendo resultados de geometría estereoscópica [126], se puede deducir que

los puntos focales en un sistema de cámaras estéreo (en este caso, estéreo temporal) y un punto en la escena  $p$  (observable desde ambas cámaras) generan un plano epipolar. Este plano corta la imagen de la cámara actual (proyectada al plano focal) en la llamada línea epipolar. Obsérvese la figura A.3. Así, el punto  $p$  en la escena, observado por el píxel  $u$  perteneciente al keyframe, debe proyectar en el plano focal de la cámara actual en algún lugar a lo largo de línea epipolar. En general, se puede concluir que el píxel dentro de la línea epipolar que corresponde al punto  $p$  observado por  $u$ , debe tener una apariencia similar a la observada en el keyframe. En otras palabras, debe minimizar el error fotométrico entre ambos píxeles.

Así, el error fotométrico  $\Psi$  responde a la ecuación:

$$\Psi(u, d) = |I_{kf}(u) - I_k(p(KT_k(dK^{-1}u)))| \quad (\text{A.2})$$

De esta manera, se calcula el error fotométrico entre los píxeles en el keyframe y sus correspondientes píxeles en la línea epipolar de la cámara actual, y esta información se acumula dentro del volumen de costo  $F(u, d)$ , realizando la promediación de todas las observaciones de error fotométrico  $\Psi$  realizadas para un mismo  $u$  y  $d$ .

Un punto a destacar es la cuantización del volumen de costo. Este tiene dimensiones  $x - y$  y  $d$ . La dimensión  $x - y$  coincide generalmente con la resolución de la cámara, por lo que es equivalente a  $u$ . La elección de la resolución en la dirección  $d$  impactará la calidad de la estimación de profundidad, ya que mayor resolución en  $d$  significa un barrido en la línea epipolar más fino. Al mismo tiempo, mientras mayor resolución en  $d$  mayor se requiere realizar una mayor cantidad de cómputos, ya que por cada píxel  $u$  perteneciente al keyframe, se debe computar la ecuación A.2 para cada  $d$  representable dentro del volumen de costo.

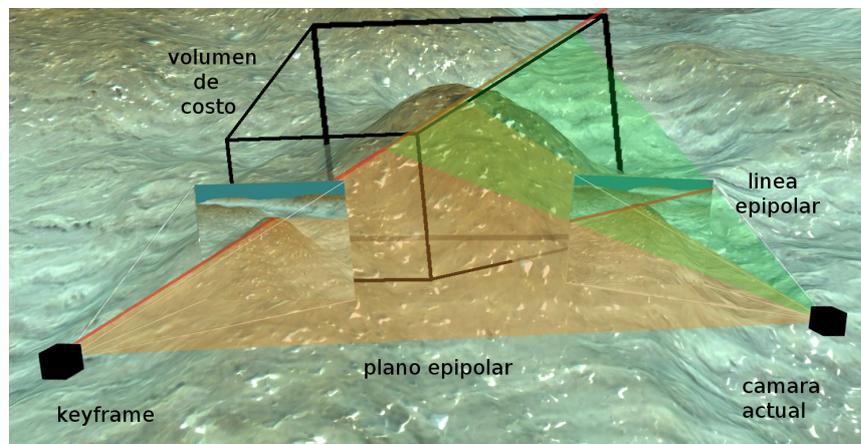


FIGURA A.3: Volumen de costo

### Módulo de Optimización de la Función de Costo

Para finalmente estimar la profundidad de los píxeles pertenecientes al keyframe, se genera una función de costo que responde a la ecuación:

$$E(\eta) = \int_{\Omega} \{g(u) \|\nabla\eta(u)\| + \lambda F(u, \eta(u))\} du \quad (\text{A.3})$$

donde  $\eta$  es la estimación de la profundidad de los píxeles  $u \in \Omega$ ,  $g(u)$  es una función de ponderación y  $F(u, d)$  representa el costo acumulado para el píxel  $u$  en

la profundidad  $d$ . Así, se busca la profundidad óptima  $\eta_{op}$  que minimice la función de costo  $E$  (para más detalles véase [90]).

### Módulo Acumulación Incremental de los Keyframes

Finalmente, se acumula incrementalmente las profundidades estimadas de los píxeles del keyframe en un mapa global, que representa la totalidad de la zona explorada.

### Conclusiones del Análisis

La resolución común para sistemas de SLAM visual es de 640x480, con una tasa de refresco de 30 Hz [14] [90] [84]. De esta manera, en este trabajo se utilizará 640x480 como resolución base, y se analizará la posibilidad de utilizar resoluciones superiores e inferiores. La tasa de refresco de 30 Hz es normal en cámaras comunes de bajo costo, por lo que ésta será la tasa de refresco analizada a lo largo de este trabajo.

Del análisis del algoritmo realizado en secciones previas, se puede observar que el módulo de localización se debe ejecutar a la misma velocidad de la tasa de refresco de la cámara. En la literatura existen ejemplos de módulos localizadores densos que se ejecutan en tiempo real en un CPU de bajo rendimiento, de manera que este módulo no se acelerará en este trabajo.

Los módulos de optimización de la función de costo y acumulación incremental de los keyframes se ejecutan solamente una vez por cada keyframe completo. Dependiendo de la velocidad del vehículo, es posible que sea necesario un nuevo keyframe luego de varios segundos. Por esto se concluye que la velocidad de estos módulos no es crítica, y pueden ser implementados correctamente en CPU.

Por otro lado, el módulo de acumulación del error fotométrico en el volumen de costo sí es necesario acelerarlo. La cantidad de operaciones a realizar en la ecuación A.2 es considerable, y debe realizarse para cada  $u$  y  $d$ . Nótese que la resolución de la cámara normalmente utilizada es 640x480 píxeles, mientras que  $d$  en trabajos como [90] puede llegar a tener 64 valores diferentes. Con estos valores, debe realizar el computo expresando en la ecuación A.2 19 millones de veces por cada nueva imagen capturada por la cámara.

Para apoyar estas conclusiones, se realizó una implementación en CPU de todo el sistema, ejecutándose en un procesador FX6100, en sus 6 núcleos. Para la realización del paralelismo, se utilizó la librería boost [127], con la cual simplemente se paralelizó los ciclos "for" que se encuentran a lo largo del programa. Con esta implementación, se logró, a una resolución de imagen de 320x240 y una resolución de profundidad del volumen de costo de 64, una tasa de refresco del volumen de costo de solamente 20 Hz. En otras palabras, cada tres cuadros capturados por la cámara, solamente se utilizan dos para la estimación de la profundidad.

Consecuentemente, los esfuerzos en la implementación de FPGA se concentraron en acelerar la ejecución de la acumulación del volumen de costo.

#### A.2.2. Implementación FPGA

En la figura A.4 se representa un diagrama en bloques de la implementación en FPGA. El sistema completo se ejecuta parte en una CPU convencional, parte en la FPGA. La comunicación se realiza entre ambos mediante el puerto PCIe, por su gran ancho de banda.

Un mensaje enviado desde el CPU indica a la unidad de control implementada en la FPGA que se requiere almacenar un nuevo keyframe, o le indica que se tiene disponible una imagen actual nueva. Terminada la transferencia de datos desde el CPU, el sistema comienza a realizar la reproyección de los píxeles pertenecientes en la imagen actual, y realizar el error fotométrico como indica la ecuación A.2. Estos datos son enviados en orden directamente al CPU a través del puerto PCIe, por lo que no se requiere almacenamiento adicional.

Uno de los objetivos que se tuvo a lo largo del diseño es que el sistema sea lo más rápido posible. Por lo que se utilizaron block-ram, y se utilizó técnicas de “pipelining” en todo el sistema de manera que sea capaz de procesar un píxel por cada ciclo de reloj.

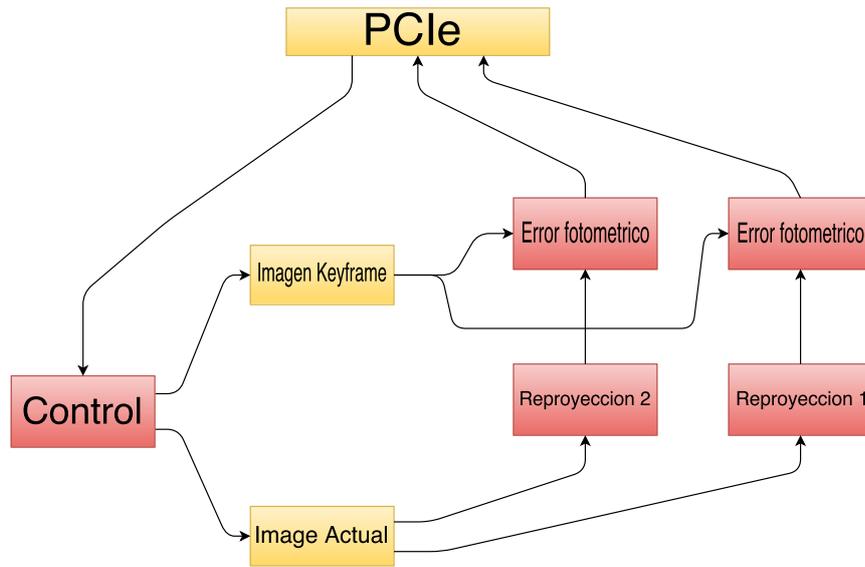


FIGURA A.4: Implementación FPGA

### Proyección al Plano Focal

Para calcular el error fotométrico entre la imagen keyframe y la imagen actual, se debe calcular para cada píxel  $u$  perteneciente a la imagen keyframe, la diferencia entre el píxel de proyección  $u_p$  dentro de imagen actual, correspondiente a la distancia  $d$ . Esta proyección responde a la ecuación:

$$u_p = K\pi(T^{-1}K^{-1}ud) \quad (\text{A.4})$$

Teniendo en cuenta que la matriz

$$M = K\pi(T^{-1}K^{-1}d) \quad (\text{A.5})$$

es constante para todo  $u$  sobre un  $d$  particular, es posible precalcular  $M$  para cada  $d$  perteneciente al volumen de costo, y realizar la proyección a la profundidad  $d$  como

$$x_p = \frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}} \quad (\text{A.6})$$

$$y_p = \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}} \quad (\text{A.7})$$

Lo cual requiere 9 multiplicaciones, 6 sumas y 2 divisiones.

Este módulo se lo realizó en pipeline, de manera de procesar un píxel por ciclo de reloj.

Un diagrama de este módulo se muestra en la figura A.5.

Si la reproyección se realiza en forma directa, esto es, se barre todos los píxeles en la imagen actual y se guardan los píxeles re-proyectados, se tienen dos problemas. En primer lugar, la imagen reproyectada presentaría huecos, lugares donde ningún píxel en la imagen actual proyectó para la imagen reproyectada. Por lo que se necesita implementar un módulo de interpolación para llenar los huecos. En segundo lugar, se necesita una memoria para ir guardando los píxeles reproyectados y así generar la imagen reproyectada.

Por el contrario, en este trabajo se eligió realizar la proyección inversa. Se barren todos los píxeles de la imagen proyectada, y utilizando la proyección inversa  $M^{-1}$ , se determinan los píxeles en la imagen actual que le corresponden. Así se evita la necesidad de realizar una interpolación, ya que esta está implícita en el tipo de cuantificación que se realiza a la salida del módulo de reproyección. Es necesario que los valores de  $u_p$  sean enteros, para que representen coordenadas de píxeles (y de memoria). En este trabajo se optó por utilizar un simple truncamiento, por ser la implementación menos costosa computacionalmente. Este truncamiento corresponde en la imagen a una interpolación "nearest-neighbor" [87].

Además de la ventaja computacional mencionada anteriormente, utilizar la proyección inversa  $M^{-1}$  conlleva una menor utilización de memoria, en comparación a la arquitectura resultante de utilizar la proyección directa  $M$ . Esto se detalla en la siguiente sección.

### Estructura de Memoria

La memoria se realizó utilizando block-ram disponibles en la FPGA. Las primitivas de utilización de dichas memorias, provistas por el fabricante de la FPGA utilizada, poseen doble puerto. De esta manera, es posible leer dos datos de la memoria simultáneamente, por lo que se pueden reproyectar dos píxeles pertenecientes a la imagen actual en paralelo. Para poder realizar más de dos reproyecciones simultáneamente, se eligió almacenar la imagen actual en varias memorias paralelamente, de manera de multiplicar así el número de puertos y como consecuencia la capacidad de lectura de píxeles. A primera vista, esta arquitectura pareciera poco eficiente en la utilización de memoria. Pero realizando un análisis más profundo, se puede llegar a la conclusión de que es la opción que menos memoria utiliza. Si se implementara una sola memoria para almacenar la imagen actual, para luego

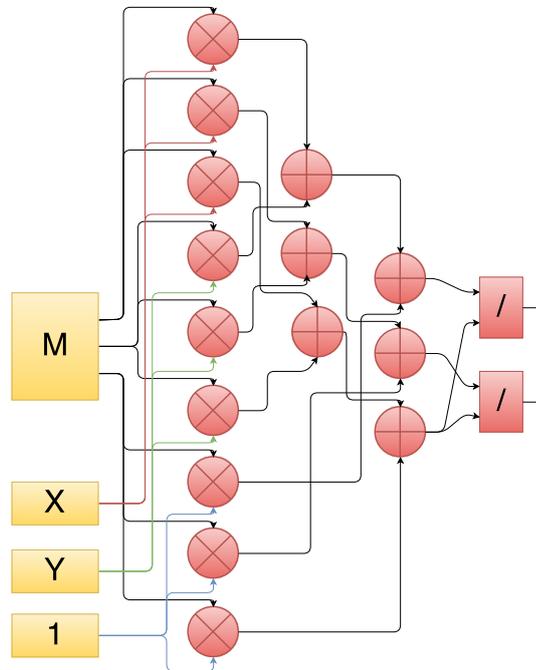


FIGURA A.5: Arquitectura del módulo de reproyección

leer sus píxeles uno a uno y realizar así distintas reproyecciones en paralelo a cada píxel, se tendría la situación descrita en la sección anterior, donde se tiene la proyección directa  $M$ . Como consecuencia, para cada proyección se debería almacenar su resultado, para luego realizar la interpolación en la imagen final. Así, utilizar la proyección directa  $M$  requiere disponer de memoria suficiente para almacenar una imagen completa por cada reproyección. Al contrario, utilizar  $M_{-1}$  requiere solamente de una memoria de imagen por cada dos proyecciones (por las memorias de doble puerto utilizadas), disminuyendo así el espacio requerido a la mitad.

### Representación Numérica

Como representación numérica se eligió el punto fijo, con una longitud de 32 bits. Se eligió utilizar 20 bits para representar la fracción. Esta elección es consecuencia de observaciones experimentales. En trabajos futuros se evaluará la utilización de otras representaciones. De esta manera, con los 12 bits de la parte entera se pueden representar valores de hasta 4096, dando una resolución máxima posible de 4096x4096.

Se implementaron las operaciones necesarias, suma, resta, multiplicación y división, todas en pipeline de forma de que procesen un valor nuevo por ciclo de reloj.

La suma y resta se implementaron sencillamente con las facilidades que da el lenguaje HDL utilizado, permitiendo que el sintetizador elija la mejor implementación.

Por otro lado, la multiplicación y división se realizaron cuidadosamente, de forma de utilizar los recursos de la FPGA disponible.

La FPGA utilizada [128] contiene módulos DSP para la realización de multiplicaciones rápidas. Estas pueden realizar multiplicaciones de 25x18 bits. Para mantener compatibilidad con FPGAs más antiguas se dividió la multiplicación en valores de 18 bits. En consecuencia, para la multiplicación se realizan cuatro multiplicaciones simultáneas en la primera etapa del pipe, y en la segunda, tercera y cuarta etapa se suman los resultados parciales. Dando así un multiplicador con pipeline con cuatro etapas.

La operación de división es la menos utilizada. Por lo que se optó implementarla sencillamente como restas sucesivas.

### Conexión a PC

Como se mencionó anteriormente, la conexión con la PC se realiza usando el puerto PCIe. Mediante el módulo de PCIe provisto por el fabricante y el módulo Xillybus [129] la comunicación es implementada mediante FIFOs, para la interfase con la FPGA, y la interface de la PC mediante "named-pipes".

Se logró obtener los datos de los errores fotométricos desde la FPGA, y guardarlos en las estructuras de datos "Mat" de la librería OpenCV. De esta manera el algoritmo sigue naturalmente en la PC.

### A.2.3. Resultados de la Implementación

Para la implementación se utilizó la placa de evaluación KC705 de Xilinx [128]. Se probó la implementación en tres configuraciones. Una configuración con tamaño de imagen de 640x480 con 64 niveles de profundidad, como la utilizada en el trabajo original [90]. La siguiente configuración probada fue un tamaño de imagen de 320x240, con 128 profundidades. Por último, se probó la implementación también

CUADRO A.1: Resultados de la Implementación FPGA

	640x480x64	320x240x128	320x240x1302
módulos en paralelo	2	1	10
Frecuencia máxima	212.5 MHz	210.2 MHz	170.0 MHz
utilización LUT	24 %	15 %	90 %
utilización Block-Ram	57 %	7 %	71 %
utilización DSP48	17 %	8 %	85 %

con 320x240 como tamaño de imagen, pero con la máxima cantidad de profundidades que fue posible en esta FPGA, 1302. La utilización de Block-Ram como memoria hizo imposible realizar pruebas con resoluciones de imágenes más altas. Esto se discute con más detalle en las conclusiones.

Los resultados se suman en la tabla [A.1](#).

### A.3. Pruebas

#### A.3.1. Dataset Utilizado

Finalmente se realizó una simulación utilizando el paquete de software para simulación GAZEBO Gazebo. Se simuló un ambiente subacuático complejo. El terreno está constituido principalmente de arena y piedras, y la superficie tiene lomadas y valles. En la figura [A.6](#) se muestra una vista de pájaro del terreno. Se observa, en conjunto con el terreno, la posición que el vehículo simulado fue tomando en diferentes etapas de la simulación. A la imagen obtenida por el vehículo se le agregó una difuminación, que depende de la profundidad, y se tomó la imagen monocromática. La figura [A.7a](#) muestra la visión de la cámara en un cuadro de la simulación.

### A.4. Conclusiones

Los resultados de este trabajo sugieren que la utilización de FPGA en sistemas de odometría visual densos puede dar como resultado una aceleración considerable. Es particularmente destacable el resultado a baja resolución, que puede realizar miles de re-proyecciones en tiempo real. Aún así, las restricciones de memoria imposibilitan usar imágenes de resoluciones mayores. Esto indica que si bien la implementación es buena, la arquitectura de memoria utilizada no es la preferible. En trabajos futuros se centrarán esfuerzos para utilizar una memoria externa (DDR3), de forma de poder utilizar resoluciones mayores.

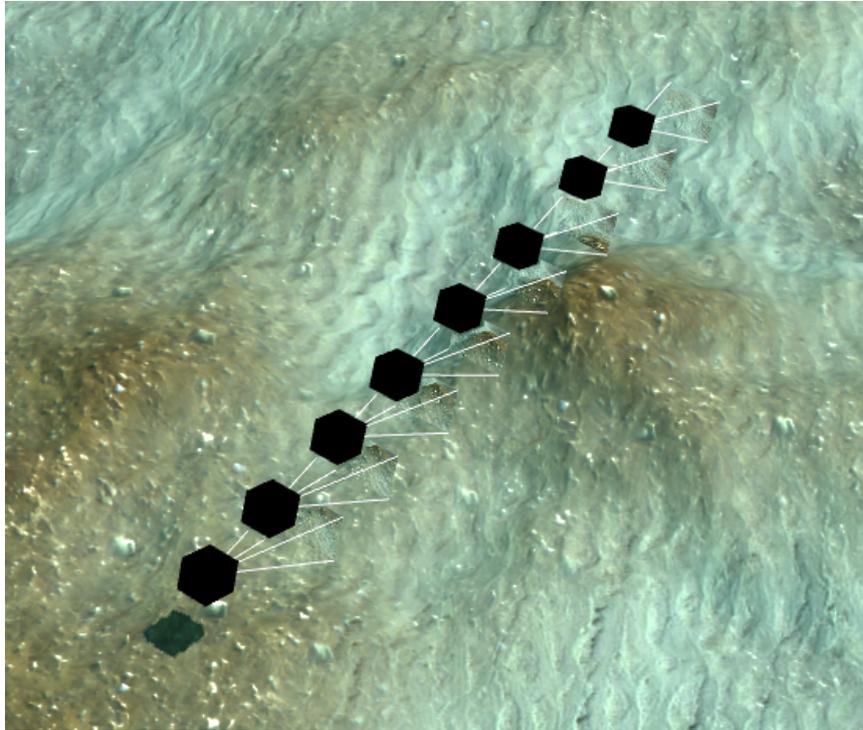
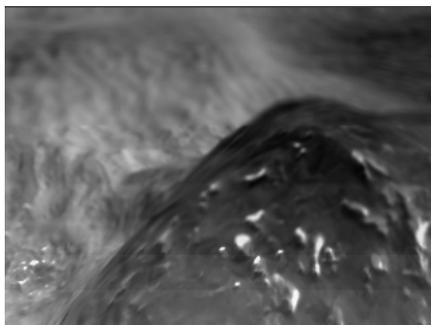
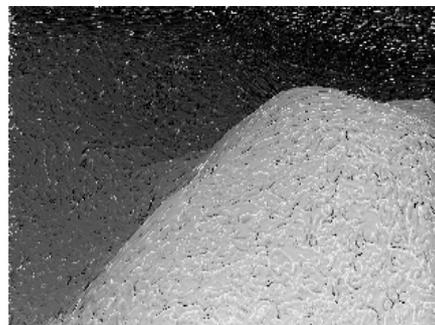


FIGURA A.6: Recorrido de la simulación



(A) Vista desde la cámara



(B) Profundidad estimada

FIGURA A.7: Resultados

## Apéndice B

# Escala Métrica

### B.1. Introducción

Las cámaras son sensores comúnmente llamados “bearing only”, significando que solamente permiten realizar, de manera directa, mediciones angulares. La escala de la escena no puede ser obtenida inmediatamente de la información contenida en los píxeles. Por lo cual los métodos de odometría visual solamente pueden estimar la escala de manera local y relativa. Así, esta estimación de la escala sufre de una diferencia con respecto a la escala real, y esta diferencia varía a lo largo de la trayectoria. Por ende, la utilidad de los mapas obtenidos mediante estas técnicas es limitada. Sin la escala métrica, sería dificultoso evitar colisiones con el ambiente inmediatamente circundante, por ejemplo. Los sistemas de SLAM monocular encontrados en la literatura, como ser [14], [13], no hacen esfuerzo alguno en obtener la escala métrica real del movimiento realizado ni del mapa tridimensional reconstruido.

En lo que respecta a métodos que permitan reconstruir escenas métricamente correctas, en la literatura abundan sistemas basados en cámaras binoculares, por ejemplo [39] [15]. En estos, la escala de la escena se obtiene a partir de las posiciones relativas de las cámaras en el sistema binocular. Como esta posición está dada por las características intrínsecas del sistema multi-cámara, que son constantes y conocidas a priori con precisión, estos métodos utilizan dicha información para obtener escenas métricamente correctas. Por supuesto, si se utiliza una cámara monocular, no es posible hacer uso de tales técnicas.

Por otro lado, existen aquellos métodos que utilizan luz estructurada [130]. Este tipo de sistemas funcionan utilizando un dispositivo que proyecta un patrón de luz conocido en la escena. El patrón es visualizado por una cámara, para luego basarse en dicha información para triangular los puntos proyectados. En general estos sistemas utilizan luz infrarroja, lo que no los hace aptos para ser utilizados en ambientes exteriores, donde el sol hace imposible la detección del patrón de luz proyectado. Al mismo tiempo estos sistemas no son utilizables en ambientes subacuáticos, medio en el cual la absorción en el espectro infrarrojo es alta. Nótese también que la proyección del patrón de luz tiene una distancia máxima, la cual en sistemas desarrollados actualmente es del orden de los 4 metros, y que este tipo de soluciones incurren en un costo energético considerable, en comparación con métodos de detección pasivos.

También se hayan en la literatura trabajos como [131], donde los autores estiman la escala métrica mediante la fusión de datos proveniente de un sensor IMU. Si bien los autores obtienen buenos resultados, uno de los objetivos del presente trabajo es la implementación de un sistema que no utilice sensores adicionales.

En este capítulo se desarrolla un método de estimación de escala métrica híbrido, utilizando estimación de profundidad monocular conjunto con una corrección utilizando luz estructurada. De esta manera se logra un sistema con la simpleza de los métodos monoculares y al mismo tiempo se logra obtener medidas métricamente

correctas. Resultados parciales que dieron lugar a este trabajo fueron presentados en [60].

## B.2. Sistema Propuesto

Se desarrolló un sistema con dos láser, que se agrega a la cámara monocular mediante una estructura rígida, ilustrada en la figura B.2.

La luz de los láseres impactará en dos puntos en la escena, que luego se observan en la imagen capturada por la cámara. El ángulo entre los láser y la cámara  $\beta$ , y la distancia entre la cámara y los láser  $a$  son conocidos con precisión, obtenidos durante el proceso calibración. De esta manera, midiendo el ángulo del píxel en el cual se observa el láser  $\gamma$ , es posible triangular la distancia entre el punto focal de la cámara y el punto impactado por cada láser.

A partir de estos parámetros, la distancia  $d_r$  desde el punto focal de la cámara hasta la posición en la escena donde impacta el láser responde a la ecuación:

$$d_r = \sin(\delta)a/\sin(\pi/2 - \delta - \gamma) \quad (\text{B.1})$$

$$\delta = \pi/2 - \beta \quad (\text{B.2})$$

$$\gamma = \text{acos}(\text{norm}(K^{-1}u)v_z)) \quad (\text{B.3})$$

donde  $K$  es la matriz intrínseca de la cámara,  $u$  es el vector homogeneizado que corresponde a las coordenadas del píxel donde se detecta el láser, y  $v_z$  el versor alineado con la dirección de visión de la cámara.

Simultáneamente se utilizan las imágenes capturadas como entrada de un sistema SLAM monocular denso, como el descrito en el capítulo 3. Este tipo de sistemas tienen la característica de no poder estimar la escala métrica de la escena. Por lo que su estimación de la distancia del punto focal hasta el punto donde impacta el láser  $d_e$ , sufre de una discrepancia en magnitud con respecto a  $d_r$ . Esta discrepancia es variable en el tiempo, pero sin embargo es idéntica para todas las estimaciones de profundidad  $d_e$  correspondientes a diferentes  $u$  (realizadas en el mismo instante). Esto significa que, si es posible obtener un factor de corrección de escala para un determinado  $u$ , las estimaciones  $d_e$  para todo  $u$  realizadas en ese mismo instante pueden ser corregidas utilizando el mismo factor de corrección.

Así, se estima el factor de corrección de escala  $s$  como:

$$s = d_r/d_e \quad (\text{B.4})$$

Claramente, la medición de  $d_r$  tendrá poca incertidumbre, ya que su estimación depende de  $a$ ,  $\beta$  y  $\gamma$ , valores conocidos con precisión. Por otro lado, la estimación de  $d_e$  provista por el sistema de odometría visual puede ser altamente ruidosa. En aras de disminuir dicho error, se implementó la siguiente heurística. La profundidad estimada no puede pertenecer a rangos de distancia muy alejada de la cámara, ya que estas estimaciones presentan gran incertidumbre. Simultáneamente, los puntos muy cercanos a la cámara son en general visualizados por poco tiempo, resultando así que también sea incierta su estimación de profundidad, por lo que también son descartados. Los que se utilizarán son entonces aquellos píxeles que correspondan a distancias medias. En los casos habituales, que corresponde a ambientes abiertos y cámaras a ángulos de 45 grados con respecto al suelo, los píxeles a distancias medias corresponden a los píxeles que se encuentran en la zona media de la imagen.

La detección de los láseres en la imagen capturada por la cámara no es trivial, por lo que se desarrolló el siguiente algoritmo. En primer lugar, se binarizó la imagen teniendo en cuenta como umbral el color y brillo del láser utilizado, cuyos valores se miden durante un proceso de calibración. Simultáneamente, se detectó en la imagen las zonas que presenten gradiente alto, utilizando un detector como el presentado en [37]. Finalmente, se buscaron coincidencias entre estas dos detecciones. Así, se obtienen las zonas observadas que presenten el color y brillo que concuerden con las características del láser, y que simultáneamente contengan diferencias sustanciales con respecto al resto de la escena circundante. Con este algoritmo se logró tener una tasa nula de falsos positivos (la detección de láseres donde realmente no estaban presentes en la escena) al costo de tener una tasa alta de falsos negativos (la no detección de láseres que sí se encontraban en la escena).

Estando provisto el sistema con dos láseres independientes, si se diera el caso de que fue posible determinar  $d_r$  para ambos láseres, y al mismo tiempo las mediciones  $d_e$  para ambos láseres cumplen con las condiciones detalladas anteriormente, se puede promediar ambas estimaciones de  $s$  para obtener una medición mejorada.

### B.3. Validación Experimental

Para evaluar la solución propuesta, se utilizó el sistema SLAM descrito en el capítulo 3. En cada cuadro capturado por la cámara se realizó la detección de los láseres. Si por lo menos uno de ellos fue detectado, se prosiguió a estimar  $s$ . Por último, cada estimación de profundidad  $d_e$  realizada por el sistema de SLAM utilizado fue corregida en escala utilizando el factor  $s$  más recientemente calculado.



FIGURA B.1: Validación experimental. A la izquierda se muestra una imagen del ambiente boscoso utilizado para realizar la experimentación. A la derecha se ilustra el patrón de calibración utilizado para ratificar la precisión en la estimación de la escala de la escena

Se probó el sistema en un ambiente real boscoso, ilustrado en la figura B.1. En dicho ambiente se adicionó un patrón de calibración, constituido por una tabla que expone dos puntos fácilmente detectables, distanciados precisamente a 48 centímetros. Se adicionó al sistema un algoritmo de detección del patrón de calibración. Este sistema, utilizando la estimación del mapa del ambiente realizada por el sistema SLAM empleado, realiza una medición de la distancia estimada entre los dos puntos contenidos dentro del patrón de calibración. De esta manera se puede contrastar la distancia estimada con la distancia real conocida a-priori.

Se utilizó un sistema compuesto por una cámara GOPRO HERO 2, utilizando una resolución de  $848 \times 480$ , con tasa de refresco de 120 Hz. Los láser utilizados tienen una potencia de 50 mW, están separados horizontalmente 17 cm del punto focal de

la cámara, y tienen unos ángulos con respecto a la línea de unión entre ambos de 45.5 y 46.7 grados, respectivamente.



FIGURA B.2: Imagen del sistema cámara-láser utilizado para la validación experimental

Los experimentos consistieron en realizar recorridos variados a través del ambiente boscoso. Cada recorrido comienza y termina visualizando el patrón de calibración. Se obtienen así dos instantes dentro de la experimentación (al principio y al final), donde es posible contrastar la escala estimada con la escala real, obtenida a partir del patrón de calibración.

En la figura B.3 se ilustran resultados para una experimentación representativa. En la figura de la izquierda se ilustra el error entre la medida real y la estimada, cuando se utiliza la corrección de escala y cuando no. Se puede observar cómo, para el sistema sin corrección, el error en escala aumenta a medida que la experimentación avanza. Por otro lado, el sistema con corrección pudo mantener la escala correcta. La imagen de la izquierda muestra la evolución del factor de escala a lo largo de la experimentación. Este muestra una gran variación, producto de la deriva en escala presente en los sistemas de odometría visual monoculares. El error presente en la escala es del orden de  $\pm 10$  cm, que en relación a la escala total de la escena (1.5 metros), representa un error de 10%. Si bien este error no es pequeño, como consecuencia de la sencillez del sistema de estimación de escala, es suficientemente precisa como para detectar con antelación posibles colisiones con el ambiente circundante.

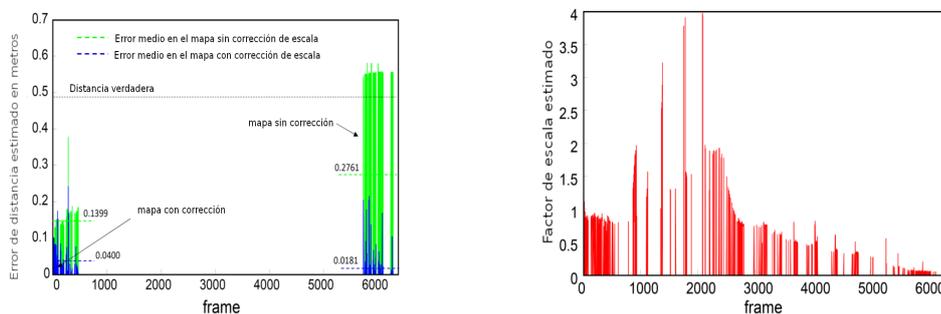


FIGURA B.3: Resultados de la experimentación. En la imagen izquierda se ilustra el error en escala, para un sistema con y sin corrección. En la imagen derecha se muestra la evolución del factor de escala estimado a lo largo de la experimentación.

A la vista de los resultados expuestos anteriormente, se puede concluir que el sistema propuesto en este capítulo puede producir estimaciones métricamente correctas, partiendo desde estimaciones provenientes de un sistema SLAM monocular.

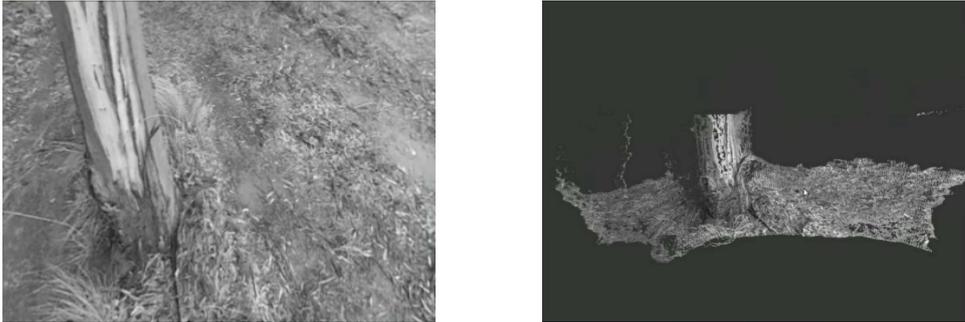


FIGURA B.4: Ilustración de un instante en la experimentación. En la imagen izquierda se puede observar la imagen capturada por la cámara. En la imagen derecha el correspondiente mapa estimado

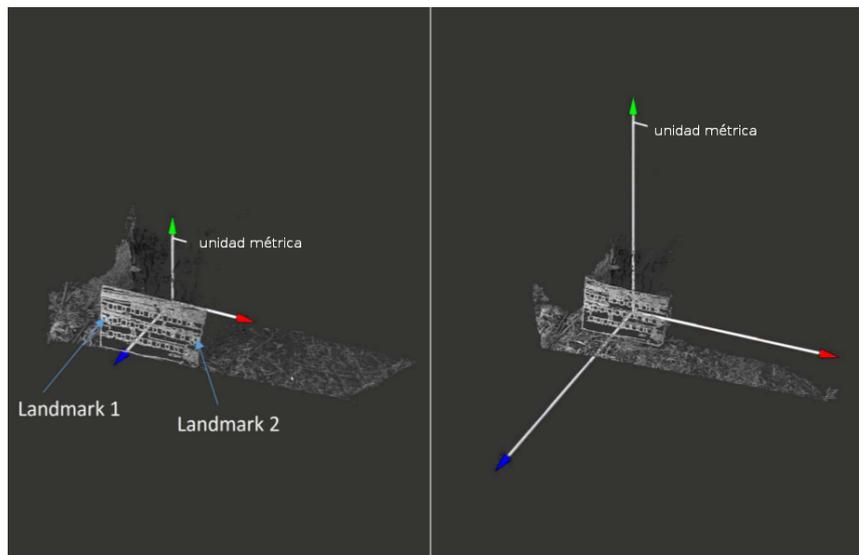


FIGURA B.5: Mapa estimado. En la imagen izquierda se observa la escala estimada para un mapa con corrección de escala. En la imagen derecha, la escala estimada para un mapa sin corrección. Se puede observar que, en el mapa corregido, la escala coincide con las dimensiones del patrón de calibración.

Dada la sencillez del sistema propuesto las estimas de escala presentan cierta incertidumbre. Sin embargo las mediciones serían lo suficientemente precisas como para permitir a un sistema autónomo evitar colisiones con el ambiente circundante.



## Apéndice C

# Segmentación Por Color

### C.1. Introducción

Una tarea realizada frecuentemente en la industria petrolífera es la inspección de ductos. La rotura de los mismos puede causar pérdidas monetarias significativas, así como graves daños a la flora y fauna submarina. Por lo que es necesario realizar inspecciones frecuentes. Dada la profundidad en que se encuentran tales ductos, así como también su longitud, es difícil para un operario hacer una inspección directa. Entonces, los métodos indirectos son frecuentemente utilizados, y entre estos los vehículos autónomos son una alternativa conveniente.

Para realizar una inspección autónoma, el vehículo debe contar con algoritmos que le permitan encontrar y seguir el ducto a lo largo de toda su extensión, y al mismo tiempo identificar las zonas donde hay una posible rotura, para posteriormente dar aviso al operario de la situación.

Entonces, estos algoritmos requieren tanto poder realizar auto-localización y mapeo, como también las tareas de identificación de ducto y seguimiento. Véase la figura C.1.

La identificación del ducto entre el lecho marino no es una tarea trivial. Partes de mismo pueden estar cubierto con arena, oxido o también con la flora existente en la zona, haciendo su identificación difícil. Entonces es importante utilizar toda la información disponible en la imagen para poder realizar identificaciones lo más precisas posibles.

Claramente, para identificar un ducto, la característica más sobresaliente es su morfología. En la literatura, se utiliza el hecho de que en general los caños son rectilíneos utilizando técnicas paramétricas [9]. Aun así, esta información puede no ser suficiente para la detección robusta. Si bien las imágenes subacuáticas carecen de información considerable de color, la poca distintibilidad que quede puede ser de ayuda para su detección.

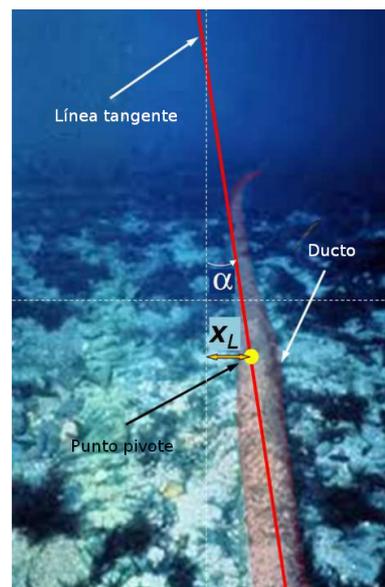


FIGURA C.1: Ilustración de un ducto submarino, en conjunto las estimaciones necesarias para su seguimiento, como su inclinación y un punto pivote.

En este capítulo se desarrolla un sistema de identificación de ductos, utilizando tanto información morfológica, como también información de su apariencia visual, dada principalmente por su color y brillo. Desarrollos parciales del trabajo presentado en este capítulo fueron presentados en [64], [69], [70], [71].

## C.2. Sistema Propuesto

### C.2.1. Algoritmo General

En la figura C.2, se ilustra el diagrama en bloques del sistema propuesto. En primer lugar, se toma la imagen en sus componentes de colores rojo, verde y azul (RGB). Se convierte esta imagen al espacio de colores color, saturación y brillo (HSV) [132]. Utilizando una muestra obtenida a-priori de la apariencia del ducto a seguir, se aplica un segmentador en los 3 canales resultantes. Paralelamente, se utiliza un algoritmo de detección de bordes basado en [133], el cual da como resultado una imagen binaria, conteniendo los bordes encontrados en la imagen. Todas las imágenes binarias provenientes de cada módulo son unidas en una única imagen binaria. Las zonas de esta manera detectadas, que serán las que contengan simultáneamente zonas con bordes y tengan la apariencia adecuada, son utilizados como entrada para un módulo identificador morfológico de líneas, que utiliza como base la transformada de Hough [9].

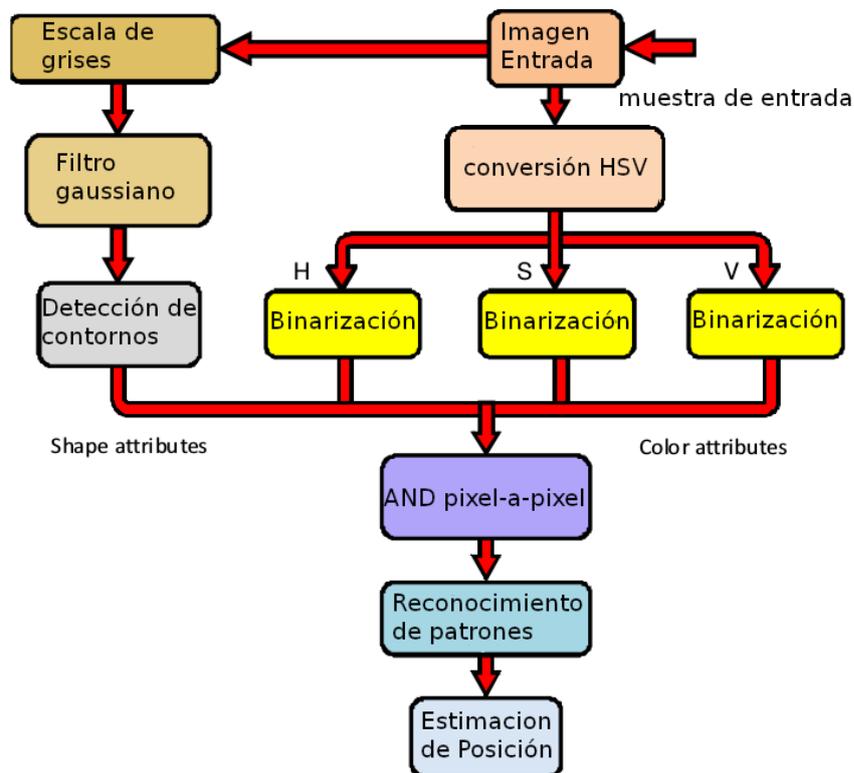


FIGURA C.2: Diagrama en bloques del sistema propuesto

### C.2.2. Algoritmo Adaptativo

El algoritmo descrito anteriormente funcionará siempre y cuando la apariencia del ducto se mantenga constante. La muestra a-priori que se tomó de su apariencia es

la que gobierna la segmentación, y si a lo largo de la extensión del ducto esta cambia significativamente, (por ejemplo, si se modifican las características de iluminación del ambiente, daría como resultado que el brillo percibido en el ducto aumentará) esa muestra ya no será relevante para identificar el ducto.

Para solucionar este problema, se implementó una mejora al algoritmo anterior. Se aplicó una modificación adaptativa de los niveles de color, saturación y brillo. Partiendo de la muestra a-priori que se toma de la apariencia del ducto, los valores del segmentador se ajustan a medida que el tiempo transcurre.

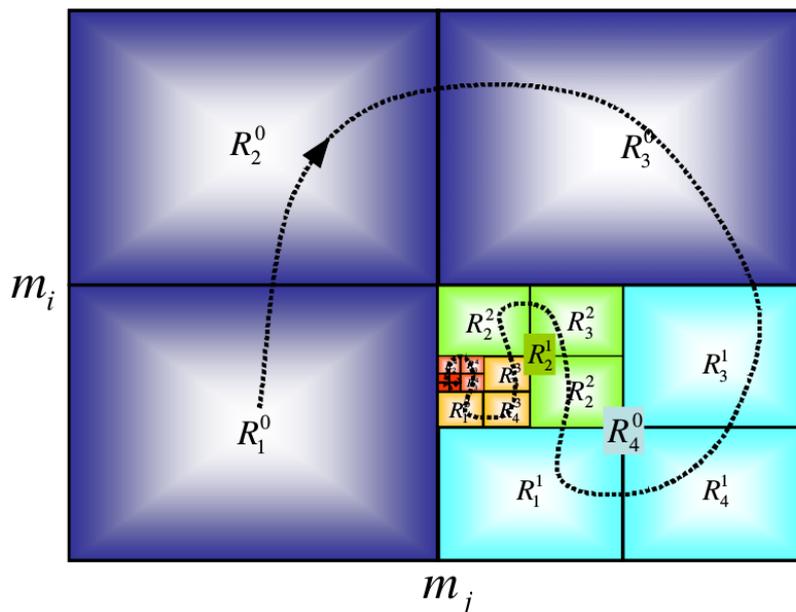


FIGURA C.3: Proceso de optimización

La figura C.3 ilustra el proceso de optimización. El principio detrás de su funcionamiento es el hecho de que la locación del ducto relativa a la imagen no puede cambiar abruptamente de cuadro a cuadro a lo largo de la secuencia de video. Así, se realiza una búsqueda exhaustiva entre los posibles valores del binarizado. Los valores correctos serán aquellos que minimicen la cantidad de píxeles identificados como pertenecientes al ducto, y al mismo tiempo maximice la elección como correctos de píxeles pertenecientes a la recta donde se encuentra el ducto, identificada en el cuadro anterior. La búsqueda exhaustiva se realiza siguiendo el principio de “coarse to fine” [99], buscando en las primeras iteraciones la zona óptima general, para luego, en iteraciones subsiguientes, realizar una búsqueda más precisa. De esta manera, la optimización es computacionalmente rápida.

La figura C.4 ilustra en diagrama en bloques el sistema con la modificación adaptativa.

### C.3. Evaluación Experimental

Para la evaluación del sistema, se desarrolló una experimentación de laboratorio. Se confeccionó una escena subacuática constituida por algas, rocas, un sustrato con alta auto-similitud simulando arena, y varios ductos entrecruzados. La iluminación de la escena fue variada a lo largo de la experimentación, simulando los cambios propios que pueden surgir en escenas subacuáticas.

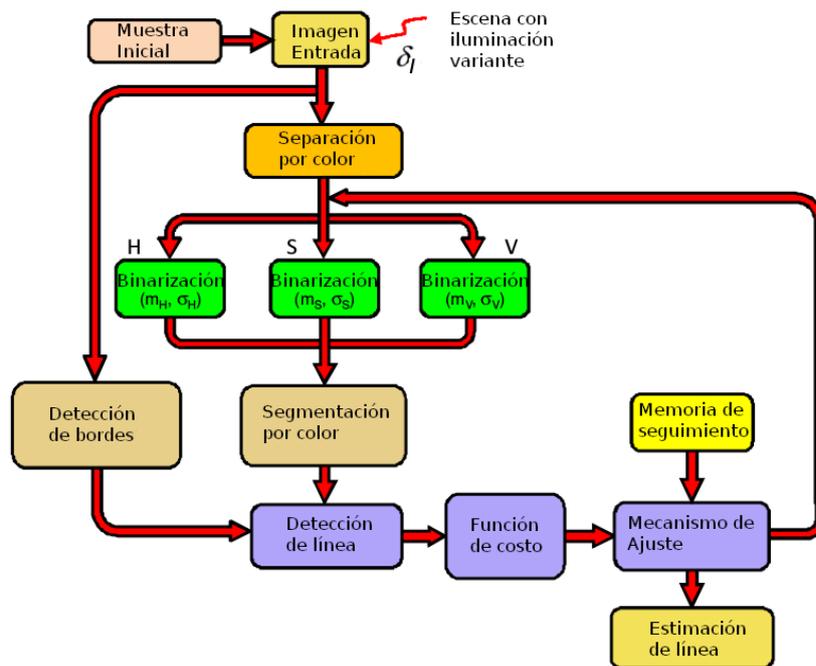


FIGURA C.4: Algoritmo adaptativo

En la figura C.5, se ilustra tres imágenes experimentales tomadas en diferentes instantes durante la experimentación. Particularmente, una imagen temprano en la experimentación (arriba), luego de un tiempo (medio), y cerca del final (abajo). A la izquierda se muestra las imágenes crudas, y a la derecha se muestra la realidad aumentada, con la identificación del ducto superpuesta a la imagen. Se muestra la identificación realizada por el algoritmo original, así como la modificación adaptativa y una línea ilustrativa de la verdadera locación del ducto.

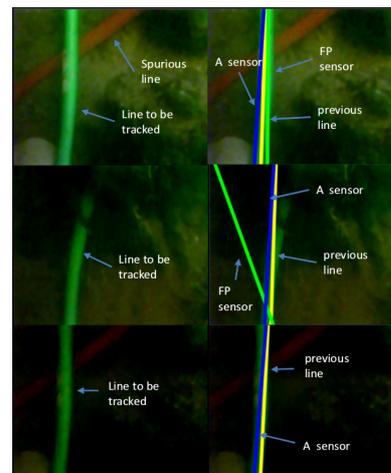


FIGURA C.5: Experimento

Nótese cómo cambió la iluminación global, haciendo la escena más oscura. Claro, bajo condiciones cambiantes el algoritmo original falló al identificar el ducto, mientras el adaptativo logró la identificación correctamente.

En la figura C.6, se muestra la segmentación originada por el algoritmo original y el adaptativo, para los mismos cuadros de la figura anterior. Se ve claramente que el origen del fallo en el algoritmo original es la incapacidad de detectar correctamente el ducto cuando las condiciones iniciales de apariencia cambian.

En la figura C.7 en la imagen superior se ilustra la tasa de fallas a lo largo de toda la experimentación. Las grandes zonas donde el algoritmo original falla coinciden con cambios en la iluminación global. Esto se ve evidenciado en la imagen inferior, donde se observa cómo cambia en el tiempo el valor de segmentación para el canal

de brillo. En azul se ilustra el valor elegido por el algoritmo adaptivo, mientras en rojo se muestra el valor fijo seleccionado para el sistema original.

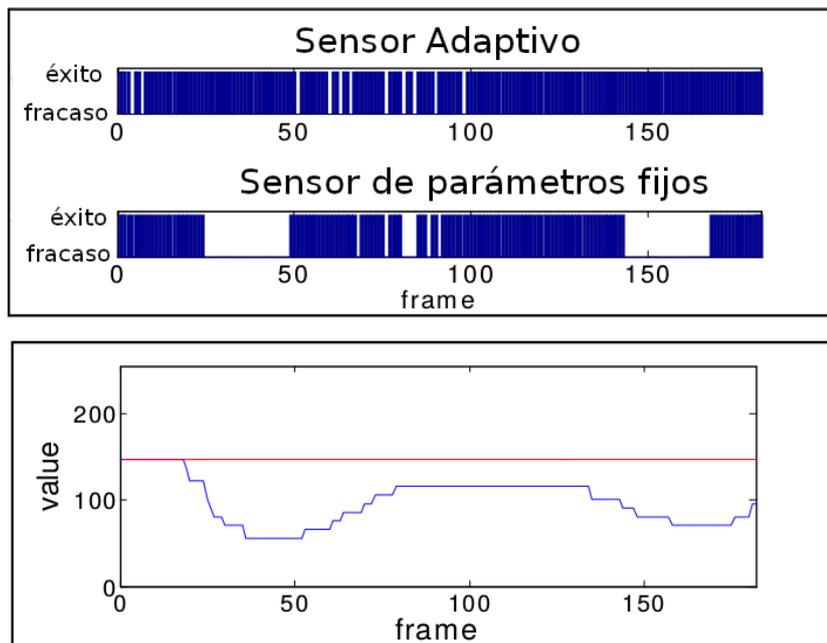


FIGURA C.7: Resultados experimentales

Dado los resultados anteriormente expuestos, se puede concluir que el sistema propuesto permite detectar ductos bajo condiciones de visibilidad y iluminación desafiantes. La modificación adaptiva propuesta permite al sistema adaptarse a cambios en la iluminación en el ambiente, que pueden darse lugar en la transición entre día y noche, proveyendo así al sistema de robustez y autonomía ante misiones prolongadas en el tiempo.

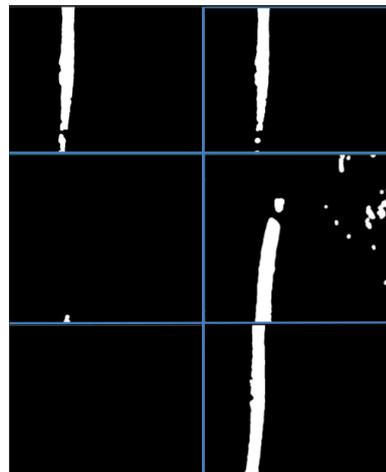


FIGURA C.6: Ilustración de la segmentación por apariencia. La columna izquierda y derecha muestra la binarización realizada mediante el algoritmo original y el adaptivo, respectivamente. Verticalmente se ilustran tres instantes durante la experimentación. Se puede observar cómo el algoritmo adaptivo pudo binarizar correctamente el ducto durante los tres instantes.



# Bibliografía

- [1] D. P. Shepard y T. E. Humphreys. «High-precision globally-referenced position and attitude via a fusion of visual SLAM, carrier-phase-based GPS, and inertial measurements». En: *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*. 2014, págs. 1309-1328. DOI: [10.1109/PLANS.2014.6851506](https://doi.org/10.1109/PLANS.2014.6851506).
- [2] L. Paull y col. «AUV Navigation and Localization: A Review». En: *IEEE Journal of Oceanic Engineering* 39.1 (2014), págs. 131-149. ISSN: 0364-9059. DOI: [10.1109/JOE.2013.2278891](https://doi.org/10.1109/JOE.2013.2278891).
- [3] F. Guth y col. «Challenges and State-of-the-Art Solutions to Underwater Slam». En: *2014 Symposium on Automation and Computation for Naval, Offshore and Subsea (NAVCOMP)*. 2014, págs. 10-13. DOI: [10.1109/NAVCOMP.2014.10](https://doi.org/10.1109/NAVCOMP.2014.10).
- [4] F. Guth y col. «Underwater SLAM: Challenges, state of the art, algorithms and a new biologically-inspired approach». En: *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. 2014, págs. 981-986. DOI: [10.1109/BIOROB.2014.6913908](https://doi.org/10.1109/BIOROB.2014.6913908).
- [5] F. Hidalgo y T. Bräunl. «Review of underwater SLAM techniques». En: *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. 2015, págs. 306-311. DOI: [10.1109/ICARA.2015.7081165](https://doi.org/10.1109/ICARA.2015.7081165).
- [6] A. Burguera, Y. González y G. Oliver. «Underwater SLAM with robocentric trajectory using a mechanically scanned imaging sonar». En: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, págs. 3577-3582. DOI: [10.1109/IROS.2011.6094850](https://doi.org/10.1109/IROS.2011.6094850).
- [7] A. Burguera, G. Oliver e Y. González. «A trajectory based framework to perform underwater SLAM using imaging sonar scans». En: *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*. 2010, págs. 1-6. DOI: [10.1109/ETFA.2010.5641227](https://doi.org/10.1109/ETFA.2010.5641227).
- [8] D. Ribas y col. «Underwater SLAM in a marina environment». En: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007, págs. 1455-1460. DOI: [10.1109/IROS.2007.4399222](https://doi.org/10.1109/IROS.2007.4399222).
- [9] Richard O. Duda y Peter E. Hart. «Use of the Hough Transformation to Detect Lines and Curves in Pictures». En: *Commun. ACM* 15.1 (ene. de 1972), págs. 11-15. ISSN: 0001-0782. DOI: [10.1145/361237.361242](https://doi.org/10.1145/361237.361242). URL: <http://doi.acm.org/10.1145/361237.361242>.
- [10] K. Siantidis. «Side scan sonar based onboard SLAM system for autonomous underwater vehicles». En: *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. 2016, págs. 195-200. DOI: [10.1109/AUV.2016.7778671](https://doi.org/10.1109/AUV.2016.7778671).
- [11] M. Hammond y S. M. Rock. «A SLAM-based approach for underwater mapping using AUVs with poor inertial information». En: *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*. 2014, págs. 1-8. DOI: [10.1109/AUV.2014.7054419](https://doi.org/10.1109/AUV.2014.7054419).

- [12] M. F. Fallon y col. «Relocating Underwater Features Autonomously Using Sonar-Based SLAM». En: *IEEE Journal of Oceanic Engineering* 38.3 (2013), págs. 500-513. ISSN: 0364-9059. DOI: [10.1109/JOE.2012.2235664](https://doi.org/10.1109/JOE.2012.2235664).
- [13] R. Mur-Artal, J. M. M. Montiel y J. D. Tardós. «ORB-SLAM: A Versatile and Accurate Monocular SLAM System». En: *IEEE Transactions on Robotics* 31.5 (2015), págs. 1147-1163. ISSN: 1552-3098. DOI: [10.1109/TRO.2015.2463671](https://doi.org/10.1109/TRO.2015.2463671).
- [14] J. Engel, T. Schöps y D. Cremers. «LSD-SLAM: Large-Scale Direct Monocular SLAM». En: *European Conference on Computer Vision (ECCV)*. 2014.
- [15] J. Engel, J. Stückler y D. Cremers. «Large-scale direct SLAM with stereo cameras». En: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, págs. 1935-1942. DOI: [10.1109/IROS.2015.7353631](https://doi.org/10.1109/IROS.2015.7353631).
- [16] J. Hoth y W. Kowalczyk. «Colour correction of underwater images». En: *OCEANS 2015 - Genova*. 2015, págs. 1-5. DOI: [10.1109/OCEANS-Genova.2015.7271356](https://doi.org/10.1109/OCEANS-Genova.2015.7271356).
- [17] J. Servos, M. Smart y S. L. Waslander. «Underwater stereo SLAM with refraction correction». En: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, págs. 3350-3355. DOI: [10.1109/IROS.2013.6696833](https://doi.org/10.1109/IROS.2013.6696833).
- [18] P. L. Negre, F. Bonin-Font y G. Oliver. «Cluster-based loop closing detection for underwater slam in feature-poor regions». En: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, págs. 2589-2595. DOI: [10.1109/ICRA.2016.7487416](https://doi.org/10.1109/ICRA.2016.7487416).
- [19] F. Bonin-Font y col. «LSH for loop closing detection in underwater visual SLAM». En: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 2014, págs. 1-4. DOI: [10.1109/ETFA.2014.7005245](https://doi.org/10.1109/ETFA.2014.7005245).
- [20] Shujing Zhang y col. «SLAM and a novel loop closure detection for autonomous underwater vehicles». En: *2013 OCEANS - San Diego*. 2013, págs. 1-4. DOI: [10.23919/OCEANS.2013.6741257](https://doi.org/10.23919/OCEANS.2013.6741257).
- [21] J. Li, R. M. Eustice y M. Johnson-Roberson. «Underwater robot visual place recognition in the presence of dramatic appearance change». En: *OCEANS 2015 - MTS/IEEE Washington*. 2015, págs. 1-6. DOI: [10.23919/OCEANS.2015.7404369](https://doi.org/10.23919/OCEANS.2015.7404369).
- [22] A. Kim y R. M. Eustice. «Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency». En: *IEEE Transactions on Robotics* 29.3 (2013), págs. 719-733. ISSN: 1552-3098. DOI: [10.1109/TRO.2012.2235699](https://doi.org/10.1109/TRO.2012.2235699).
- [23] L. P. Berczi y T. D. Barfoot. «It's like De javu all over again: Learning place-dependent terrain assessment for visual teach and repeat». En: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, págs. 3973-3980. DOI: [10.1109/IROS.2016.7759585](https://doi.org/10.1109/IROS.2016.7759585).
- [24] M. Paton y col. «Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat». En: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, págs. 1918-1925. DOI: [10.1109/IROS.2016.7759303](https://doi.org/10.1109/IROS.2016.7759303).

- [25] J. Dequaire y col. «Off the beaten track: Predicting localisation performance in visual teach and repeat». En: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, págs. 795-800. DOI: [10.1109/ICRA.2016.7487209](https://doi.org/10.1109/ICRA.2016.7487209).
- [26] J. Salvi, Y. Petillot y E. Batlle. «Visual SLAM for 3D large-scale seabed acquisition employing underwater vehicles». En: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, págs. 1011-1016. DOI: [10.1109/IROS.2008.4650627](https://doi.org/10.1109/IROS.2008.4650627).
- [27] S. Nagappa y col. «Single cluster PHD SLAM: Application to autonomous underwater vehicles using stereo vision». En: *2013 MTS/IEEE OCEANS - Bergen*. 2013, págs. 1-9. DOI: [10.1109/OCEANS-Bergen.2013.6608107](https://doi.org/10.1109/OCEANS-Bergen.2013.6608107).
- [28] S. Pi y col. «Stereo visual SLAM system in underwater environment». En: *OCEANS 2014 - TAIPEI*. 2014, págs. 1-5. DOI: [10.1109/OCEANS-TAIPEI.2014.6964369](https://doi.org/10.1109/OCEANS-TAIPEI.2014.6964369).
- [29] V. Bobkov y col. «Vision-based navigation method for a local maneuvering of the autonomous underwater vehicle». En: *2017 IEEE Underwater Technology (UT)*. 2017, págs. 1-5. DOI: [10.1109/UT.2017.7890304](https://doi.org/10.1109/UT.2017.7890304).
- [30] J. Aulinas y col. «Feature extraction for underwater visual SLAM». En: *OCEANS 2011 IEEE - Spain*. 2011, págs. 1-7. DOI: [10.1109/Oceans-Spain.2011.6003474](https://doi.org/10.1109/Oceans-Spain.2011.6003474).
- [31] M. Meireles y col. «Real time visual SLAM for underwater robotic inspection». En: *2014 Oceans - St. John's*. 2014, págs. 1-5. DOI: [10.1109/OCEANS.2014.7003097](https://doi.org/10.1109/OCEANS.2014.7003097).
- [32] A. Burguera, F. Bonin-Font y G. Oliver. «Towards robust image registration for underwater visual SLAM». En: *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*. Vol. 3. 2014, págs. 539-544.
- [33] S. Hong, T. Kim y J. Kim. «Underwater visual SLAM with loop-closure using image-to-image link recovery». En: *OCEANS 2015 - Genova*. 2015, págs. 1-6. DOI: [10.1109/OCEANS-Genova.2015.7271448](https://doi.org/10.1109/OCEANS-Genova.2015.7271448).
- [34] M. Wu y Jian Yao. «Adaptive UKF-SLAM based on magnetic gradient inversion method for underwater navigation». En: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2015, págs. 839-843. DOI: [10.1109/ICUAS.2015.7152369](https://doi.org/10.1109/ICUAS.2015.7152369).
- [35] J. Jung y col. «AUV localization using depth perception of underwater structures from a monocular camera». En: *OCEANS 2016 MTS/IEEE Monterey*. 2016, págs. 1-4. DOI: [10.1109/OCEANS.2016.7761113](https://doi.org/10.1109/OCEANS.2016.7761113).
- [36] S. Hong y J. Kim. «Efficient visual SLAM using selective image registration for autonomous inspection of underwater structures». En: *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. 2016, págs. 189-194. DOI: [10.1109/AUV.2016.7778670](https://doi.org/10.1109/AUV.2016.7778670).
- [37] Herbert Bay y col. «SURF: Speeded Up Robust Features». En: *Computer Vision and Image Understanding (CVIU)* 110.3 (2008), págs. 346-359.
- [38] David G. Lowe. «Distinctive Image Features from Scale-Invariant Keypoints». En: *Int. J. Comput. Vision* 60.2 (2004), págs. 91-110. ISSN: 0920-5691. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94). URL: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.

- [39] F. Bonin-Font y col. «Stereo SLAM for robust dense 3D reconstruction of underwater environments». En: *OCEANS 2015 - Genova*. 2015, págs. 1-6. DOI: [10.1109/OCEANS-Genova.2015.7271333](https://doi.org/10.1109/OCEANS-Genova.2015.7271333).
- [40] Mario Campos Alejo Concha Paulo Drews-Jr y Javier Civera. «Real-Time Localization and Dense Mapping in Underwater Environments from a Monocular Sequence». En: *OCEANS 2015 - Genova*. 2015, págs. 1-5.
- [41] D. A. Smallwood y L. L. Whitcomb. «Adaptive identification of dynamically positioned underwater robotic vehicles». En: *IEEE Transactions on Control Systems Technology* 11.4 (2003), págs. 505-515. ISSN: 1063-6536. DOI: [10.1109/TCST.2003.813377](https://doi.org/10.1109/TCST.2003.813377).
- [42] D. A. Smallwood y L. L. Whitcomb. «Model-based dynamic positioning of underwater robotic vehicles: theory and experiment». En: *IEEE Journal of Oceanic Engineering* 29.1 (2004), págs. 169-186. ISSN: 0364-9059. DOI: [10.1109/JOE.2003.823312](https://doi.org/10.1109/JOE.2003.823312).
- [43] Thor I. Fossen y Ola-Erik Fjellstad. «Robust Adaptive Control of Underwater Vehicles: A Comparative Study». En: *IFAC Proceedings Volumes* 28.2 (1995). 3rd IFAC Workshop on Control Applications in Marine Systems, Trondheim, Norway, 10-12 May, págs. 66 -74. ISSN: 1474-6670. DOI: [https://doi.org/10.1016/S1474-6670\(17\)51653-5](https://doi.org/10.1016/S1474-6670(17)51653-5). URL: <http://www.sciencedirect.com/science/article/pii/S1474667017516535>.
- [44] Liu Hsu y col. «Dynamic positioning of remotely operated underwater vehicles». En: *IEEE Robotics Automation Magazine* 7.3 (2000), págs. 21-31. ISSN: 1070-9932. DOI: [10.1109/100.876908](https://doi.org/10.1109/100.876908).
- [45] G. Antonelli, F. Caccavale y S. Chiaverini. «Adaptive tracking control of underwater vehicle-manipulator systems based on the virtual decomposition approach». En: *IEEE Transactions on Robotics and Automation* 20.3 (2004), págs. 594-602. ISSN: 1042-296X. DOI: [10.1109/TRA.2004.825521](https://doi.org/10.1109/TRA.2004.825521).
- [46] Jeen-Shing Wang y C. S. G. Lee. «Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle». En: *IEEE Transactions on Robotics and Automation* 19.2 (2003), págs. 283-295. ISSN: 1042-296X. DOI: [10.1109/TRA.2003.808865](https://doi.org/10.1109/TRA.2003.808865).
- [47] K.D. Do, J. Pan y Z.P. Jiang. «Robust and adaptive path following for underactuated autonomous underwater vehicles». En: *Ocean Engineering* 31.16 (2004), págs. 1967 -1997. ISSN: 0029-8018. DOI: <https://doi.org/10.1016/j.oceaneng.2004.04.006>. URL: <http://www.sciencedirect.com/science/article/pii/S0029801804000940>.
- [48] Miroslav Krstic, Petar V. Kokotovic y Ioannis Kanellakopoulos. *Nonlinear and Adaptive Control Design*. 1st. New York, NY, USA: John Wiley & Sons, Inc., 1995. ISBN: 0471127329.
- [49] A.L. Fradkov, I.V. Miroshnik y V.O. Nikiforov. *Nonlinear and Adaptive Control of Complex Systems*. Mathematics and Its Applications. Springer Netherlands, 1999. ISBN: 9780792358923. URL: <https://books.google.com.ar/books?id=jNomjo-on0IC>.
- [50] K. D. Do y J. Pan. «Robust and adaptive path following for underactuated autonomous underwater vehicles». En: *Proceedings of the 2003 American Control Conference, 2003*. Vol. 3. 2003, 1994-1999 vol.3. DOI: [10.1109/ACC.2003.1243367](https://doi.org/10.1109/ACC.2003.1243367).

- [51] Ji-Hong Li, Pan-Mook Lee y Bong-Huan Jun. «An adaptive nonlinear controller for diving motion of an AUV». En: *OCEANS '04. MTTT/IEEE TECHNO-OCEAN '04*. Vol. 1. 2004, 282-287 Vol.1. DOI: [10.1109/OCEANS.2004.1402930](https://doi.org/10.1109/OCEANS.2004.1402930).
- [52] M. A. Jordan y J. L. Bustamante. «A Speed-Gradient Adaptive Control with State/Disturbance Observer for Autonomous Subaquatic Vehicles». En: *Proceedings of the 45th IEEE Conference on Decision and Control*. 2006, págs. 2008-2013. DOI: [10.1109/CDC.2006.377169](https://doi.org/10.1109/CDC.2006.377169).
- [53] M. A. Jordan y J. L. Bustamante. «An adaptive control system for perturbed ROVs in discrete sampling missions with optimal-time characteristics». En: *2007 46th IEEE Conference on Decision and Control*. 2007, págs. 1300-1305. DOI: [10.1109/CDC.2007.4434786](https://doi.org/10.1109/CDC.2007.4434786).
- [54] A. Serrani y G. Conte. «Robust nonlinear motion control for AUVs». En: *IEEE Robotics Automation Magazine* 6.2 (1999), págs. 33-38, 62. ISSN: 1070-9932. DOI: [10.1109/100.774926](https://doi.org/10.1109/100.774926).
- [55] M. A. Jordan y J. L. Bustamante. *Adaptive Control for Guidance of Underwater Vehicles*. Underwater Vehicles. INTECH, 2009. URL: [https://www.intechopen.com/books/underwater\\_vehicles/adaptive\\_control\\_for\\_guidance\\_of\\_underwater\\_vehiclesC](https://www.intechopen.com/books/underwater_vehicles/adaptive_control_for_guidance_of_underwater_vehiclesC).
- [56] M. A. Jordan y J. L. Bustamante. «Robust and adaptive path following for underactuated autonomous underwater vehicles». En: *Proceedings of the In American Control Conference*. 2011.
- [57] M. A. Jordan y J. L. Bustamante. *A General Approach to Discrete-Time Adaptive Control Systems with Perturbed for Complex Dynamics. Case Study : Underwater Operated Vehicles*. INTECH, 2011.
- [58] Emanuel Trabes y Mario Alberto Jordán. «A Node-Based Method for SLAM Navigation in Self-Similar Underwater Environments: A Case Study». En: *Robotics* 6.4 (2017). ISSN: 2218-6581. DOI: [10.3390/robotics6040029](https://doi.org/10.3390/robotics6040029). URL: <http://www.mdpi.com/2218-6581/6/4/29>.
- [59] E. Trabes y M. Jordán. «Depth Range Adaptation to Variable Scale in 3D-Scenarios for Dense SLAM». En: *8th International Conference on Physics and Control (Physcon 2017)*. 2017.
- [60] E. Trabes y M. Jordán. «Automatic Scaling in 3D Map Building for SLAM». En: *8th International Conference on Physics and Control (Physcon 2017)*. 2017.
- [61] E. Trabes y M. Jordán. «A Solution to the Problem of Caustic Waves Perturbations in Visual Sensors Underwater». En: *Cybernetics And Physics Open Access Journal (IN PRESS)* (2016).
- [62] E. Trabes y M. Jordán. «Aceleración de un sistema SLAM visual denso utilizando FPGA». En: *VII Congreso de Microelectrónica Aplicada*. 2016.
- [63] E. Trabes y M. Jordán. «A chained-nodes-based method to complement slam navigation in self-similar underwater environments». En: *Congreso Argentino de Control Automatico 2016 (AADECA 16)*. 2016.
- [64] E. Trabes y M. Jordán. «An adaptive vision-based sensor for underwater line detection employing shape and color image segmentation». En: *Journal of Automation and Control Engineering* (2015).

- [65] E. Trabes y M. Jordán. «Employing Feedback to Filter Caustic Waves in Underwater Scenes in Motion». En: *44JAIIO Jornadas Argentinas de Informática*. 2015.
- [66] E. Trabes y M. Jordán. «On-line filtering of sunlight caustic waves in underwater scenes in motion». En: *7th Int. Scientific Conference on Physics and Control*. 2015.
- [67] E. Trabes y M. Jordán. «A multi-stage visual odometry system and failure detection mechanism for a monocular visual SLAM implementation in monotonously textured underwater scenes». En: *2015 XVI Workshop on Information Processing and Control (RPIC)*. 2015, págs. 1-6. DOI: [10.1109/RPIC.2015.7497133](https://doi.org/10.1109/RPIC.2015.7497133).
- [68] E. Trabes y M. A. Jordán. «Self-tuning of a sunlight-deflickering filter for moving scenes underwater». En: *2015 XVI Workshop on Information Processing and Control (RPIC)*. 2015, págs. 1-6. DOI: [10.1109/RPIC.2015.7497107](https://doi.org/10.1109/RPIC.2015.7497107).
- [69] Mario A. Jordán y Emanuel Trabes. «An adaptive vision-based sensor for underwater line detection employing shape and color image segmentation». En: *3rd International Conference on Control, Robotics and Informatics (ICCRI 2014)*. 2014.
- [70] Mario A. Jordán y Emanuel Trabes. «An adaptive vision-based sensor for underwater line detection in changing and low illumination levels». En: *Congreso Argentino de Control Automatico 2014*. 2014.
- [71] Mario A. Jordán y col. «A Sensor for Vision-based Navigation in Underwater Path Tracking with Color and Edge Segmentation». En: *15º Argentine Symposium on Technology*. 2014.
- [72] Carlos E. Berger y col. «A camera-guided controller for path tracking of autonomous underwater vehicles». En: *Congreso Argentino de Control Automatico 2014*. 2014.
- [73] Carlos E. Berger y col. «A Vision-based Controller for Path Tracking of Autonomous Underwater Vehicles». En: *15º Argentine Symposium on Technology*. 2014.
- [74] A. Angeli y col. «Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words». En: *IEEE Transactions on Robotics* 24.5 (2008), págs. 1027-1037. ISSN: 1552-3098. DOI: [10.1109/TRO.2008.2004514](https://doi.org/10.1109/TRO.2008.2004514).
- [75] C. J. Ostafew, A. P. Schoellig y T. D. Barfoot. «Visual teach and repeat, repeat, repeat: Iterative Learning Control to improve mobile robot path tracking in challenging outdoor environments». En: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, págs. 176-181. DOI: [10.1109/IRROS.2013.6696350](https://doi.org/10.1109/IRROS.2013.6696350).
- [76] A. Pfrunder, A. P. Schoellig y T. D. Barfoot. «A Proof-of-Concept Demonstration of Visual Teach and Repeat on a Quadcopter Using an Altitude Sensor and a Monocular Camera». En: *2014 Canadian Conference on Computer and Robot Vision*. 2014, págs. 238-245. DOI: [10.1109/CRV.2014.40](https://doi.org/10.1109/CRV.2014.40).
- [77] T. Nguyen, G. K. I. Mann y R. G. Gosine. «Vision-based qualitative path-following control of quadrotor aerial vehicle». En: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2014, págs. 412-417. DOI: [10.1109/ICUAS.2014.6842281](https://doi.org/10.1109/ICUAS.2014.6842281).

- [78] M. Paton y col. «It's not easy seeing green: Lighting-resistant stereo Visual Teach and Repeat using color-constant images». En: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, págs. 1519-1526. DOI: [10.1109/ICRA.2015.7139391](https://doi.org/10.1109/ICRA.2015.7139391).
- [79] M. Paton, F. Pomerleau y T. D. Barfoot. «Eyes in the Back of Your Head: Robust Visual Teach and Repeat Using Multiple Stereo Cameras». En: *2015 12th Conference on Computer and Robot Vision*. 2015, págs. 46-53. DOI: [10.1109/CRV.2015.16](https://doi.org/10.1109/CRV.2015.16).
- [80] K. L. Clarkson. «Fast algorithms for the all nearest neighbors problem». En: *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*. 1983, págs. 226-232. DOI: [10.1109/SFCS.1983.16](https://doi.org/10.1109/SFCS.1983.16).
- [81] Nathan Koenig y Andrew Howard. «Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator». En: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan, 2004, págs. 2149-2154.
- [82] N. Yang y col. «Challenges in Monocular Visual Odometry: Photometric Calibration, Motion Bias and Rolling Shutter Effect». En: *ArXiv e-prints* (mayo de 2017). arXiv: [1705.04300 \[cs.CV\]](https://arxiv.org/abs/1705.04300).
- [83] J. Engel, V. Koltun y D. Cremers. «Direct Sparse Odometry». En: *arXiv:1607.02565*. 2016.
- [84] Georg Klein y David Murray. «Parallel Tracking and Mapping on a Camera Phone». En: *Proc. Eighth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*. Orlando, 2009.
- [85] H. Zhou y col. «StructSLAM: Visual SLAM With Building Structure Lines». En: *IEEE Transactions on Vehicular Technology* 64.4 (2015), págs. 1364-1375. ISSN: 0018-9545. DOI: [10.1109/TVT.2015.2388780](https://doi.org/10.1109/TVT.2015.2388780).
- [86] Martin A. Fischler y Robert C. Bolles. «Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography». En: *Commun. ACM* 24.6 (jun. de 1981), págs. 381-395. ISSN: 0001-0782. DOI: [10.1145/358669.358692](https://doi.org/10.1145/358669.358692). URL: <http://doi.acm.org/10.1145/358669.358692>.
- [87] T. M. Lehmann, C. Gonner y K. Spitzer. «Survey: interpolation methods in medical image processing». En: *IEEE Transactions on Medical Imaging* 18.11 (1999), págs. 1049-1075. ISSN: 0278-0062. DOI: [10.1109/42.816070](https://doi.org/10.1109/42.816070).
- [88] David Nistér. «An Efficient Solution to the Five-Point Relative Pose Problem». En: *IEEE Trans. Pattern Anal. Mach. Intell.* 26.6 (jun. de 2004), págs. 756-777. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2004.17](https://doi.org/10.1109/TPAMI.2004.17). URL: <http://dx.doi.org/10.1109/TPAMI.2004.17>.
- [89] Bill Triggs y col. «Bundle Adjustment - A Modern Synthesis». En: *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice. ICCV '99*. London, UK, UK: Springer-Verlag, 2000, págs. 298-372. ISBN: 3-540-67973-1. URL: <http://dl.acm.org/citation.cfm?id=646271.685629>.
- [90] R. A. Newcombe, S. J. Lovegrove y A. J. Davison. «DTAM: Dense tracking and mapping in real-time». En: *2011 International Conference on Computer Vision*. 2011, págs. 2320-2327. DOI: [10.1109/ICCV.2011.6126513](https://doi.org/10.1109/ICCV.2011.6126513).
- [91] Matia Pizzoli, Christian Forster y Davide Scaramuzza. «REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time». En: *IEEE International Conference on Robotics and Automation (ICRA)*. 2014.

- [92] P. Ondrůška, P. Kohli y S. Izadi. «MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones». En: *IEEE Transactions on Visualization and Computer Graphics* 21.11 (2015), págs. 1251-1258. ISSN: 1077-2626. DOI: [10.1109/TVCG.2015.2459902](https://doi.org/10.1109/TVCG.2015.2459902).
- [93] Z. Yang, F. Gao y S. Shen. «Real-time monocular dense mapping on aerial robots using visual-inertial fusion». En: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, págs. 4552-4559. DOI: [10.1109/ICRA.2017.7989529](https://doi.org/10.1109/ICRA.2017.7989529).
- [94] Thomas Schöps y col. «3D Modeling on the Go: Interactive 3D Reconstruction of Large-Scale Scenes on Mobile Devices». En: *3DV*. 2015.
- [95] R. A. Newcombe y col. «KinectFusion: Real-time dense surface mapping and tracking». En: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 2011, págs. 127-136. DOI: [10.1109/ISMAR.2011.6092378](https://doi.org/10.1109/ISMAR.2011.6092378).
- [96] *Kinect*. <https://en.wikipedia.org/wiki/Kinect>. Accessed: 2017-07-22.
- [97] Thomas Whelan y col. «Real-time Large-scale Dense RGB-D SLAM with Volumetric Fusion». En: *Int. J. Rob. Res.* 34.4-5 (abr. de 2015), págs. 598-626. ISSN: 0278-3649. DOI: [10.1177/0278364914551008](https://doi.org/10.1177/0278364914551008). URL: <http://dx.doi.org/10.1177/0278364914551008>.
- [98] Christoph Rhemann y Carsten Rother. «Fast Cost-Volume Filtering for Visual Correspondence and Beyond». En: *CVPR*. 2012. URL: <https://www.microsoft.com/en-us/research/publication/fast-cost-volume-filtering-for-visual-correspondence-and-beyond/>.
- [99] Hossein Mobahi y John W. Fisher. «Coarse-to-Fine Minimization of Some Common Nonconvexities». En: *EMMCVPR*. 2014.
- [100] F. Steinbruecker, J. Sturm y D. Cremers. «Volumetric 3D Mapping in Real-Time on a CPU». En: *Int. Conf. on Robotics and Automation*. Hongkong, China, 2014.
- [101] Y. Y. Schechner y N. Karpel. «Attenuating natural flicker patterns». En: *OCEANS '04. MTT/IEEE TECHNO-OCEAN '04*. Vol. 3. 2004, 1262-1268 Vol.3. DOI: [10.1109/OCEANS.2004.1405760](https://doi.org/10.1109/OCEANS.2004.1405760).
- [102] N. Gracias y col. «A motion compensated filtering approach to remove sunlight flicker in shallow water images». En: *OCEANS 2008*. 2008, págs. 1-7. DOI: [10.1109/OCEANS.2008.5152111](https://doi.org/10.1109/OCEANS.2008.5152111).
- [103] Y. Matsushita y col. «Shadow elimination for robust video surveillance». En: *Workshop on Motion and Video Computing, 2002. Proceedings*. 2002, págs. 15-21. DOI: [10.1109/MOTION.2002.1182208](https://doi.org/10.1109/MOTION.2002.1182208).
- [104] Y. Swirski y col. «Underwater stereo using natural flickering illumination». En: *OCEANS 2010 MTS/IEEE SEATTLE*. 2010, págs. 1-7. DOI: [10.1109/OCEANS.2010.5664275](https://doi.org/10.1109/OCEANS.2010.5664275).
- [105] Gunnar Farneback. «Two-Frame Motion Estimation Based on Polynomial Expansion». En: *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29 – July 2, 2003 Proceedings*. Ed. por Josef Bigun y Tomas Gustavsson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, págs. 363-370. ISBN: 978-3-540-45103-7. DOI: [10.1007/3-540-45103-X\\_50](https://doi.org/10.1007/3-540-45103-X_50). URL: [https://doi.org/10.1007/3-540-45103-X\\_50](https://doi.org/10.1007/3-540-45103-X_50).

- [106] G. Antonelli. *Underwater Robots*. Springer Tracts in Advanced Robotics. Springer International Publishing, 2013. ISBN: 9783319028774. URL: <https://books.google.com.ar/books?id=cu05BQAAQBAJ>.
- [107] M. A. Jordán y col. «Two Approaches to Adaptive Control Applied to a Wheeled Mobile Robot». En: 2011.
- [108] T. Fossen. *Guidance and Control of Ocean Vehicles*. New York, NY: Wiley, 1995.
- [109] Jorge Luis Bustamante. «Control Adaptivo de Vehículos subacuáticos Autónomos y Teleoperados con Perturbaciones». Tesis doct. Universidad Nacional del Sur, 2009.
- [110] Javier García Martín. *Adapting a Low Cost Robot to Environmental Underwater Monitoring*. 2014.
- [111] *OpenROV*. <https://www.openrov.com/>. Accessed: 2017-07-22.
- [112] *Atmel*. <http://www.atmel.com>. Accessed: 2017-07-22.
- [113] *BeagleBoneBlack*. <http://beagleboard.org/>. Accessed: 2017-07-22.
- [114] *OpenROV Source Code*. <https://github.com/OpenROV>. Accessed: 2017-07-22.
- [115] *ROS*. <http://www.ros.org>. Accessed: 2017-07-22.
- [116] *Ubuntu*. <https://www.ubuntu.com/>. Accessed: 2017-07-22.
- [117] *OpenCV Camera Calibration*. [https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html). Accessed: 2017-07-22.
- [118] J. H. Kim, C. Cadena e I. Reid. «Direct semi-dense SLAM for rolling shutter cameras». En: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, págs. 1308-1315. DOI: 10.1109/ICRA.2016.7487263.
- [119] *Nvidia Jetson*. <http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html>. Accessed: 2017-07-22.
- [120] P. Kühnreiter, M. Knecht y C. Traxler. «BRDF approximation and estimation for Augmented Reality». En: *15th International Conference on System Theory, Control and Computing*. 2011, págs. 1-6.
- [121] A. Burguera, F. Bonin-Font y G. Oliver. «Reducing the computational cost of underwater visual SLAM using dynamic adjustment of overlap detection». En: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 2014, págs. 1-8. DOI: 10.1109/ETFA.2014.7005083.
- [122] K. Boikos y C. S. Bouganis. «Semi-dense SLAM on an FPGA SoC». En: *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. 2016, págs. 1-4. DOI: 10.1109/FPL.2016.7577365.
- [123] M. Abouzahir y col. «High-level synthesis for FPGA design based-SLAM application». En: *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. 2016, págs. 1-8. DOI: 10.1109/AICCSA.2016.7945638.
- [124] Michael Montemerlo y col. «FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping That Provably Converges». En: *Proceedings of the 18th International Joint Conference on Artificial Intelligence. IJCAI'03*. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003, págs. 1151-1156. URL: <http://dl.acm.org/citation.cfm?id=1630659.1630824>.

- [125] Rick Szeliski. *Image Alignment and Stitching: A Tutorial*. Inf. téc. 2004, pág. 89. URL: <https://www.microsoft.com/en-us/research/publication/image-alignment-and-stitching-a-tutorial/>.
- [126] K. Hata y M. Etoh. «Epipolar geometry estimation and its application to image coding». En: *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*. Vol. 2. 1999, 472-476 vol.2. DOI: [10.1109/ICIP.1999.822941](https://doi.org/10.1109/ICIP.1999.822941).
- [127] Boris Schling. *The Boost C++ Libraries*. XML Press, 2011. ISBN: 0982219199, 9780982219195.
- [128] *Xilinx Kintex-7 FPGA KC705 Evaluation Kit*. <http://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html>. Accessed: 2017-07-22.
- [129] *Xillibus*. <http://xillybus.com/>. Accessed: 2017-07-22.
- [130] T. Igelbrink, T. Wiemann y J. Hertzberg. «Generating topologically consistent triangle meshes from large scale Kinect Fusion». En: *2015 European Conference on Mobile Robots (ECMR)*. 2015, págs. 1-6. DOI: [10.1109/ECMR.2015.7324205](https://doi.org/10.1109/ECMR.2015.7324205).
- [131] E. Kobzili, C. Larbes y A. Allam. «Multi-rate robust scale estimation of monocular SLAM». En: *2017 6th International Conference on Systems and Control (ICSC)*. 2017, págs. 1-5. DOI: [10.1109/ICoSC.2017.7958644](https://doi.org/10.1109/ICoSC.2017.7958644).
- [132] Noor A. Ibraheem y col. «Understanding Color Models: A Review». En: *ARPN Journal of Science and Technology* (2012), págs. 265-275.
- [133] J Canny. «A Computational Approach to Edge Detection». En: *IEEE Trans. Pattern Anal. Mach. Intell.* 8.6 (jun. de 1986), págs. 679-698. ISSN: 0162-8828. DOI: [10.1109/TPAMI.1986.4767851](https://doi.org/10.1109/TPAMI.1986.4767851). URL: <http://dx.doi.org/10.1109/TPAMI.1986.4767851>.