

## Hacia la Navegación Visual de un Vehículo Autónomo Submarino en Areas con *Posidonia Oceanica*

Francisco Bonin-Font\*, Carles Coll Gomila, Gabriel Oliver Codina

Grupo Sistemas, Robótica y Visión, Universidad de las Islas Baleares, ctra de Valldemossa km 7.5, 07122 Palma de Mallorca.

### Resumen

Este artículo presenta los resultados de un estudio experimental exhaustivo que determina el tipo de características visuales que presentan una mayor robustez, estabilidad y trazabilidad en imágenes submarinas tomadas en entornos colonizados con *Posidonia Oceanica* (P.O.), sean consecutivas o que cierran bucles (imágenes que muestran una misma área, parcial o totalmente, tomadas en tiempos distintos, desde puntos de vista distintos o incluso en condiciones de iluminación diferentes). El trabajo se ha centrado en dos puntos fundamentales: a) evaluar la capacidad que pueden tener varias técnicas de aumento de contraste en imágenes con P.O. a la hora de aumentar el número y calidad de las características visuales, y b) encontrar la combinación detector/descriptor invariante a rotación y traslación, que maximiza el número de correspondencias *inliers* usadas posteriormente para el cálculo de la odometría visual, o en el registro de imágenes que cierran bucles.

*Palabras Clave:*

Sistemas de navegación, Robot submarino autónomo, Navegación del robot, Visión del robot, Odometría visual.

### Towards Visual Navigation of an Autonomous Underwater Vehicle in Areas with *Posidonia Oceanica*

#### Abstract

This paper presents an exhaustive, extensive and detailed experimental assessment of different types of visual key-points in terms of robustness, stability and traceability, in images taken in marine areas densely colonized with *Posidonia Oceanica* (P.O.). This work has been focused mainly in two issues: a) evaluating the capacity of several image color and contrast enhancing preprocessing techniques to increase the image quality and the number of stable features, and b) finding the pair feature detector/descriptor, from a wide range of different combinations, that maximizes the number of inlier correspondences in consecutive frames or frames that close a loop (images that overlap, taken at distant time instants, from different viewpoints or even with different environmental conditions). Conclusions extracted from both evaluations will affect directly the quality of visual odometers and/or the image registration processes involved in visual SLAM approaches.

*Keywords:* Autonomous Mobile Robots, Robot Navigation, Robot Vision, Visual Motion.

### 1. Introducción

La *Posidonia Oceanica* (P.O.) es una planta acuática endémica del Mediterráneo, que forma grandes praderas y juega un papel importantísimo en el mantenimiento de los ecosistemas marinos, ya que: a) contribuye a la deposición y consolidación de sedimentos, b) atenúa las corrientes marinas, c) es refugio y fuente de comida de numerosas especies de animales

y vegetales, d) absorbe dióxido de carbono y libera oxígeno, aumentando la transparencia y calidad del agua, y mitigando el cambio climático (Díaz-Almela y Duarte, 2008). La *Posidonia* es muy sensible a la contaminación, y se ve muy afectada por la actividad humana, como por ejemplo, los fondos incontrolados y la pesca de arrastre. En general, la extensión de esta planta está en claro declive en todo el Mediterráneo (Short et al., 2012; Jorda et al., 2012), y su deterioro afecta directamente a

\*Autor para correspondencia: francisco.bonin@uib.es

las industrias pesquera y turística.

La Comunidad Europea identifica las praderas de *Posidonia Oceanica* como hábitats de alto interés y exhorta a los diferentes enclaves turísticos Mediterráneos a delimitar áreas de especial protección. Por todo ello, es primordial aplicar estrategias de monitorización y control de estos hábitats.

La cartografía geo-localizada de las colonias de P.O. es crucial para controlar su extensión y estado, y para proteger los ecosistemas simbióticos desarrollados dentro de ellas. Normalmente el control de estas praderas se realiza a través de buzos, quienes instalan marcadores y medidores en el perímetro y dentro de las mismas para delimitar su extensión y altura. En ocasiones, los buzos son seguidos mediante el uso de localizadores acústicos (Scaradozzi et al., 2009). Este proceso es lento, ofrece datos imprecisos y está supeditado a la capacidad de los tanques de oxígeno y a las medidas de seguridad que deben observar los buzos. Actualmente hay algunas propuestas tecnológicas para detectar y delimitar las praderas de Posidonia: a partir de imágenes multi-espectrales vía satélite (Matarrese et al., 2008) o más recientemente, utilizando vehículos submarinos autónomos equipados con cámaras (Bonin-Font et al., 2016, 2017), o un *Side Scan Sonar* (SSS) (Maida et al., 2011; Montefalcone et al., 2013). No obstante, ninguna de estas estrategias ha dado resultados definitivos, dejando abiertas toda una serie de posibilidades incluidas en el proyecto nacional ARSEA (Augmented Reality Subsea Exploration Assistant-TIN2014-58662-R), actualmente en curso.

Este proyecto propone explorar los fondos marinos colonizados con P.O. con un robot submarino autónomo (AUV) modelo SPARUS II (Carreras et al., 2013), equipado con una cámara estéreo enfocada hacia el fondo. El objetivo es construir foto-mosaicos (2D) geo-localizados, que muestren, con gran fiabilidad, la estructura de las praderas, permitiendo calcular automáticamente diversos indicadores biológicos para valorar su estado y evolución. Para conseguir que el vehículo siga la trayectoria programada es necesario contar con un sistema de localización y navegación preciso y efectivo. Además, con el fin de obtener foto-mosaicos ajustados al entorno real, es necesario aplicar unas *características visuales* (*visual features*) suficientemente robustas, que garanticen el registro de imágenes que solapan, tanto consecutivas como si cierran bucles (imágenes no consecutivas, tomadas en instantes claramente separados en el tiempo, desde distintos puntos de vista, y quizás, con diferentes condiciones ambientales, que visualizan la misma zona y que solapan total o parcialmente).

Normalmente los AUVs navegan integrando los datos de movimiento obtenidos por sensores inerciales (giróscopos, acelerómetros), odométricos (*Doppler Velocity Log-DVL*) o de posicionamiento absoluto (GPS, sensores de presión o *Ultra Short Baselines* USBLs). Sin embargo, los sistemas de localización y navegación asistidos por cámaras reducen el precio, aumentan la fiabilidad y la versatilidad y permiten resoluciones espaciales y temporales más altas. Múltiples experimentos con SPARUS II realizados en el contexto del proyecto ARSEA han demostrado que el funcionamiento del DVL navegando relativamente cerca (entre 3 a 5 metros) de fondos cubiertos con Posidonia es suficientemente errático como para no fiar la navegación del vehículo a este sensor, aumentando la necesidad de complementar la localización con odometría visual. La odometría

visual normalmente se basa en el seguimiento de características visuales en imágenes consecutivas que solapan parte de la zona visualizada. Este cálculo será más fiable y robusto cuanto mayor sea el número de correspondencias correctas entre imágenes.

Adicionalmente, con el fin de corregir la típica deriva en el posicionamiento del vehículo cuando ésta se calcula a partir de sensores inerciales u odométricos, se puede ejecutar simultáneamente algún proceso de *Simultaneous Localization and Mapping* (SLAM) visual en el que el registro de imágenes que cierran bucles se convierte en un punto crítico (Eustice et al., 2008). El registro visual de imágenes que solapan se realiza con éxito a partir de un cierto número de correspondencias entre puntos característicos de las dos imágenes involucradas. Si el registro tiene éxito, se obtiene una transformación geométrica entre las dos imágenes. Esta transformación es más robusta cuando el número de correspondencias aumenta, y pierde fiabilidad conforme éste se reduce, de tal manera que existe una correlación directa entre el número de correspondencias y el hecho de poder o no registrar dos imágenes que cierran un bucle (Negre et al., 2016b). Por ello, para que un sistema de SLAM visual funcione correctamente es imprescindible obtener un número considerablemente alto de correspondencias entre imágenes que cierran bucles, y un número muy bajo de correspondencias entre imágenes que no lo cierran.

Sin embargo, los entornos marinos colonizados con P.O. presentan a los sistemas visuales dificultades añadidas a las propias de los entornos acuáticos: a) la absorción y la dispersión de la luz en diferentes direcciones causado por las partículas en suspensión en el agua puede generar diversos efectos indeseados como turbidez, niebla, disminución en el brillo, saturación o viñeteo, los cuales dificultan la extracción de características visuales suficientemente descriptivas y robustas, b) la P.O. genera entornos dinámicos debido al ligero movimiento de las hojas con la corriente marina, aumentando la dificultad de emparejar las mismas características visuales en imágenes que muestran zonas comunes, y c) pueden presentar extensas áreas en las que se combinan texturas extremadamente complejas como en las zonas con alta densidad de P.O., zonas de rocas u otras algas, combinado con áreas sin prácticamente textura (arena) y con muy pocas características visuales. Esto los convierte en entornos muy exigentes donde los sistemas convencionales de navegación, localización y/o SLAM visual pueden fallar o dar resultados poco fiables.

Algún trabajo reciente (Ferreira et al., 2016) explora el comportamiento de distintos tipos de características visuales en entornos marinos. Aunque nuestro trabajo se inspira en (Ferreira et al., 2016), aquí se introducen tres componentes adicionales: a) los experimentos presentados en (Ferreira et al., 2016) están realizados en entornos marinos donde la textura predominante es la arena, muy distintos a los que se exigen en el proyecto ARSEA, donde se incluyen áreas extensas con P.O., abundancia de rocas y otras especies vegetales que cubren el fondo marino, b) esto obliga a aumentar el número de combinaciones de detectores/descriptores, c) se amplía el estudio a imágenes que cierran bucles, y d) aquí además se aplican algoritmos de pre-procesamiento de las imágenes que modifican el color y el contraste, con el fin de poder procesar con éxito áreas con distintos tipos de P.O., de diferentes tonalidades distintas, y bajo

diferentes condiciones ambientales y de iluminación.

Este artículo presenta una evaluación completa del rendimiento de 34 combinaciones de detectores y descriptores de características visuales, potencialmente invariantes a rotación y traslación, aplicadas a secuencias grabadas en áreas con P.O.. El objetivo de esta comparativa es dilucidar qué combinaciones son las más robustas en estos entornos marinos, en términos de cantidad de características y la robustez en su trazabilidad, en dos aspectos esenciales en la localización visual: 1) en imágenes consecutivas (*feature tracking*), lo cual afecta directamente al cálculo de la odometría visual del vehículo, y 2) en imágenes que cierran bucles (*feature matching*), lo cual influye en la corrección de las derivas en posicionamiento mediante técnicas de SLAM visual y también en la construcción de foto-mosaicos.

El porcentaje de *inliers* o de *outliers* es un factor relevante a la hora de evaluar el comportamiento de un proceso de *feature tracking* visual, ya que los *outliers* indican correspondencias incorrectas, descartables para cualquier tipo de cálculo de movimiento, y los *inliers* representan las correspondencias correctas. Cuanto mayor sea el número de *inliers*, más sólida es aquella combinación de detector/descriptor, más frecuente y numerosa en una misma imagen, más fácil de encontrar su correspondencia en imágenes que solapan y en consecuencia, más fiable para el cálculo visual del movimiento del robot. Aunque el buen funcionamiento de muchos de los detectores y descriptores usados en esta comparativa ha sido ampliamente contrastado en la literatura, se ha hecho en la mayoría de los casos en el aire, se han probado en muy pocas ocasiones en entornos marinos y nunca se ha hecho en áreas con alta densidad de vegetación marina y rocas. En general, las características visuales se pueden comportar de forma distinta según el entorno y condiciones ambientales en las que se capturan las imágenes, sea en el aire o en el agua. Por ejemplo, ORB-SLAM (Mur-Artal et al., 2015) es un sistema probadamente robusto en entornos terrestres pero que demuestra fallar considerablemente en entorno marinos con P.O. (Negre et al., 2016a). Es por ello que este estudio es esencial, ya que el uso de un tipo u otro de característica visual, más o menos robusta, en este tipo de entornos condicionará el éxito de los procesos de localización y navegación visual del vehículo. La obtención de la mejor combinación detector/descriptor es un paso previo que permite aumentar al máximo la fiabilidad de la odometría visual calculada después, usando algún método contrastado tipo Viso2 (Geiger et al., 2011) o ORB-Tracker (Mur-Artal et al., 2015). De forma similar a (Ferreira et al., 2016), el objetivo de este trabajo no es el cálculo y optimización de la odometría visual, sino establecer, en un paso previo, cual detector/descriptor debemos usar en este tipo de entornos, que nos garantice una alta fiabilidad y robustez en el cálculo posterior de algún tipo de odometría y/o SLAM visual basados en *feature tracking/matching*.

Además, con el fin de aumentar el número y la calidad de las características visuales, todas las imágenes se pre-procesan con 4 técnicas de mejora de color y aumento de contraste distintas, aumentando el número de combinaciones valoradas hasta 136. El amplio abanico de resultados experimentales extraídos de una serie de secuencias de vídeo grabadas con el robot SPARUS II en la costa norte de Mallorca muestran claramente que características visuales ofrecen mejores prestaciones y las que tienen un comportamiento más errático.

## 2. Registro de imágenes y detección de bucles

En aplicaciones de *photo-mosaicing*, SLAM visual o navegación visual topológica, el registro visual de dos imágenes parte del hecho de que, si dos imágenes solapan (visualizan total o parcialmente la misma área), muchas de las características visuales deberían tener descriptores muy similares, y sus correspondencias se pueden usar para calcular el movimiento relativo entre ambas imágenes. En nuestro caso, dado que la cámara del robot está enfocada hacia abajo, con el eje de la lente perpendicular al plano del fondo marino y el robot se mueve a altura constante girando solo en *yaw* (presenta solo 3 grados de libertad,  $-x, y, \theta$ ), si la altura a la que se mueve el robot es mucho mayor que el relieve del suelo el registro de dos imágenes se puede aproximar por una transformación  $T$  (roto-traslación) afín en el plano horizontal, que mapea cada punto de la imagen objetivo con su correspondiente en la imagen de referencia:

$$T = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

, donde  $\theta$  indica la rotación horizontal entre las dos imágenes, y el par  $(t_x, t_y)$  la traslación de los puntos en las coordenadas horizontales  $(x, y)$ .

El parámetro de escala se obtiene posteriormente a partir de los datos de distancia al fondo del DVL combinado con la altura calculada a partir de las imágenes estéreo. Esta transformación  $T$  se obtiene aplicando un algoritmo iterativo similar a *RANdom SAmple Consensus* (RANSAC) (Fischler y Bolles, 1981) al conjunto de correspondencias entre dos imágenes que solapan. La transformación resultante  $T$  es la que minimiza el error cuadrático  $\varepsilon$  entre una serie sucesiva de subconjuntos  $C$  de dichas correspondencias (Burguera et al. (2015)):  $T = \arg\min_{rt} e(rt)$ , donde  $\varepsilon = e(T)$  (error en la transformada) y  $e(rt) = \sum_{(i,j) \in C} \|p_i - Tq_j\|^2$ , siendo  $p_i$  y  $q_j$  las coordenadas homogéneas de una pareja de *features* correspondientes, uno de cada imagen, y  $rt$  la roto-traslación que transforma  $q_j$  en  $p_i$ .

Para determinar las correspondencias entre puntos de dos imágenes que potencialmente cierran un bucle, se aplica un proceso de *feature matching* por *Brute Force*. Las correspondencias entre características visuales se obtienen a partir de una función *Sum of Square Differences* (SSD) que compara cada descriptor de cada característica de la imagen 1 ( $q_j$ ) con todas las características encontradas en la imagen2 ( $p_i$ ). Este proceso no solo devuelve la transformada que genera un error mínimo en todas las correspondencias encontradas, sino que además elimina las correspondencias erróneas.

Este proceso permite identificar partes del entorno visitados anteriormente por el robot (bucles), imponiendo restricciones de localización en el vehículo, y en consecuencia, aumentando de forma clara la exactitud en su posicionamiento incremental. Se considera que dos imágenes no consecutivas cierran un bucle si el número de *inliers* sobrepasa un cierto umbral. Asimismo se considera que no cierran un bucle si el número de *inliers* está por debajo de otro umbral distinto. Lo deseable, es que el intervalo entre ambos umbrales sea grande, que las imágenes que cierran bucle tengan muchos *inliers*, y aquellas que no lo cierran tengan muy pocos. Si el número de *inliers* tiene el mismo rango tanto para imágenes que cierran bucles como para imágenes

nes que no los cierran, no es posible discriminar unos casos de los otros.

### 3. Odometría visual

El proceso de *feature tracking* entre características visuales de dos imágenes consecutivas que solapan aplica también la función de SSD, pero asumiendo tres condiciones adicionales: 1) un punto en una imagen se proyecta en la siguiente imagen con las mismas características visuales, 2) un punto en una imagen sufre un desplazamiento muy pequeño en la imagen consecutiva; esta asunción lleva implícito suponer que la frecuencia de captura de imágenes es suficientemente alta, 3) coherencia espacial: todos los puntos vecinos se mueven en la misma dirección.

El proceso consiste en la ejecución de dos pasos:

- para cada característica visual en la primera imagen se calcula el SSD entre su descriptor y el descriptor de cada una de las características que se encuentran dentro de una zona de la segunda imagen, restringida a unos cuantos píxeles de superficie y centrada en las coordenadas del punto en la primera imagen;
- aquel punto o característica en la segunda imagen cuyo SSD está por debajo de un cierto umbral  $\gamma$  se considera una correspondencia correcta o *inlier*; si existen varias posibles, se asigna aquella con un SSD mínimo.

Además, se aplica un proceso de *Cross Check Matching* en el que las correspondencias en un sentido (de la imagen 1 a la 2) solo son validadas si también se dan en sentido contrario (de la imagen 2 a la 1).

Una vez calculadas las correspondencias correctas entre puntos característicos de imágenes consecutivas, se puede obtener una primera estimación del movimiento instantáneo de la cámara a través de la Homografía  $H$  que modela la transformación de los puntos que son coplanares de forma análoga a la transformación  $T$  referida en la sección anterior. Cada característica visual en la primera imagen ( $x$ ) y su homóloga en la segunda ( $x'$ ), ambas representadas con sus coordenadas homogéneas

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

, respectivamente, cumplen  $x' = Hx$ .

En general, para el cálculo de  $H$ , son necesarios al menos 4 correspondencias para resolver los 8 grados de libertad sin ambigüedades (Hartley y Zisserman, 2003). No obstante, dado que el número de grados de libertad del vehículo se limita a 3 ( $x, y, \theta$ ), en este caso 2 correspondencias son suficientes. El cálculo de  $H$  también incluye un proceso iterativo convergente basado en *Least MEDian of Square regression* (LMEDS) (Sherzter y Prager, 2002) y la eliminación de las correspondencias con un error de re-proyección superior a un cierto umbral, en nuestro caso configurado a 1 pixel (Wilow-Garage, 2014). Por lo tanto, tanto la matriz de homografía como los *inliers* obtenidos son altamente fiables, si las condiciones de cálculo son óptimas. Cuantas menos correspondencias aparecen entre dos

imágenes, o si estas son de menor calidad (unen puntos que no son exactamente el mismo), menos ajustada será la transformada incluida en la homografía resultante. Y esto dependerá del detector/descriptor utilizado.

En imágenes consecutivas con alto grado de solapamiento (en nuestro caso, baja velocidad del robot, 7 fps), el número de correspondencias será normalmente alto. Aunque el número de puntos necesario para el cálculo de  $H$  es pequeño, si el número de correspondencias entre imágenes consecutivas o imágenes que cierran un bucle es muy bajo y/o no son suficientemente robustas, hay más probabilidades de que RANSAC no converja a una solución óptima aplicable a todas las correspondencias entre las dos imágenes. En este caso, aplicar LMEDS es preferible ya que para garantizar una odometría mínimamente fiable necesitamos más de un 50 % de *inliers*, y si no los hay, hay que desestimar el resultado obtenido. Por otro lado, si el número de características es muy alto, el número de *outliers* puede aumentar considerablemente. Para este trabajo, el resultado final a evaluar no es la transformada resultante, sino la cantidad de *inliers* encontrados para cada detector/descriptor.

En resumen, las características visuales (descriptor/detector) más adecuadas para trabajar en nuestros entornos con Posidonia deberían responder a algunos condicionantes comunes típicos en aplicaciones de cálculo de movimiento o identificación de escenas (Krig, 2014) (Li et al., 2015):

- Robustez, trazabilidad y rapidez en los procesos de *matching* y *tracking*, ya que la localización se debe realizar on-line.
- Invarianza a rotación y traslación para detectar cierres de bucle, y permitir el cálculo de la odometría visual.
- El número de *inliers* entre imágenes consecutivas y imágenes que potencialmente cierran bucles debe ser razonablemente alto.
- El número de *inliers* entre imágenes que no cierran lazos debe ser considerablemente bajo.

En este trabajo se han combinado 7 detectores (STAR, FAST, SIFT, SURF, BRISK, ORB y HARRIS) con 5 descriptores (SIFT, SURF, BRISK, ORB y FREAK), todos ellos ampliamente utilizados y referenciados en la literatura en numerosas aplicaciones de navegación, en un total de 34 combinaciones potencialmente invariantes a rotación y traslación. El objetivo será determinar cuáles de ellos ofrecen un mejor comportamiento a la hora de calcular el movimiento de la cámara y registrar imágenes que solapan.

### 4. Mejora de Contraste en Imágenes Submarinas con P.O.

Con el fin de aumentar la eficiencia del proceso descrito en el anterior apartado, las imágenes se han pre-procesado con distintos filtros de mejora de contraste y de color. En total se han estudiado cuatro filtros: *Local Color Correction* (LCC) (Moore et al., 1991), *Multi Scale Retinex* (MSR) (Jobson et al., 1997), *Multi Scale Retinex - New Kernel* (MSR-NK) (Morel et al., 2014) y un cuarto método de mapeo de tonalidades basado en el algoritmo de Mai et al (Lauga et al., 2014). La figura 1 muestra una imagen extraída de una de las secuencias grabadas por el

SPARUS II en el norte de Mallorca, con presencia de Posidonia y la misma imagen después de haber aplicado los 4 filtros.

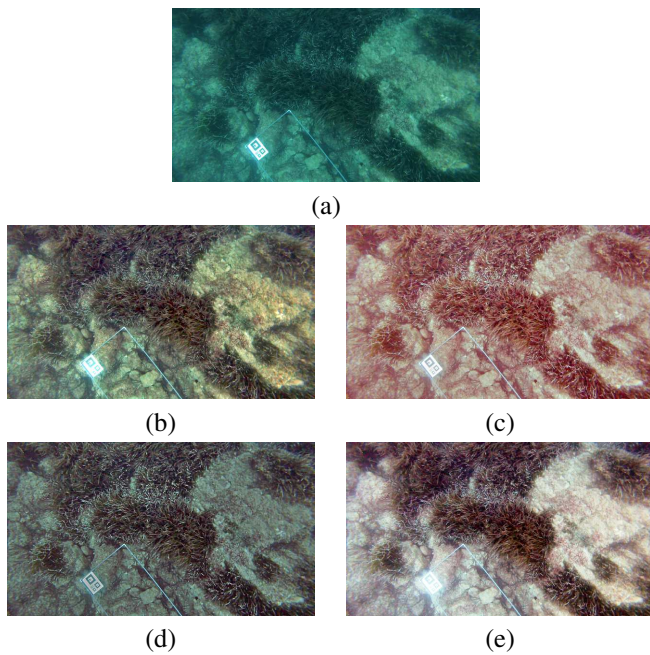


Figura 1: (a) Imagen original. Filtradas con: (b) LCC, (c) MSR, (d) MSR-NK, (e) MAI.

Adicionalmente, se ha aplicado *Contrast Limited Adaptive Histogram Specification* (CLAHS) (Eustice et al., 2002) en alguna secuencia de vídeo, solo para pares de imágenes que potencialmente puede cerrar un bucle. Este método de mejora de contraste está incluido en la librería UWIT (Underwater Image Toolbox for Optical Image Processing and Mosaicking in MATLAB) (Eustice et al., 2002), y está diseñado para tratar la falta de uniformidad en la iluminación, efecto inherente a la toma de imágenes en entornos marinos. La librería UWIT está especialmente enfocada a mejorar diversos aspectos en la construcción de foto-mosaicos, siendo el uso de CLAHS en las imágenes implicadas uno de los puntos claves en el registro de imágenes que cierran bucles.

### 5. Experimentos

El método experimental seguido para evaluar el comportamiento de las 34 diferentes características visuales (detector/descriptor) consta de los siguientes pasos:

A) Se grabaron varias secuencias de vídeo con SPARUS II en diferentes zonas de la costa norte de Mallorca, navegando a una altura y velocidad constantes y garantizando suficiente solapamiento entre imágenes consecutivas. Cada trayectoria del vehículo cubrió no menos de 400m<sup>2</sup>. La distancia al fondo utilizada para el control de altura en la navegación y el cálculo del factor de escala necesario para la conversión pixel-unidades métricas se obtuvo combinando los datos de altura del DVL con datos de profundidad calculados a partir de las imágenes estéreo.

B) Para cada secuencia de vídeo se obtuvo la siguiente información: a) Las características visuales de todas las imágenes de cada secuencia;

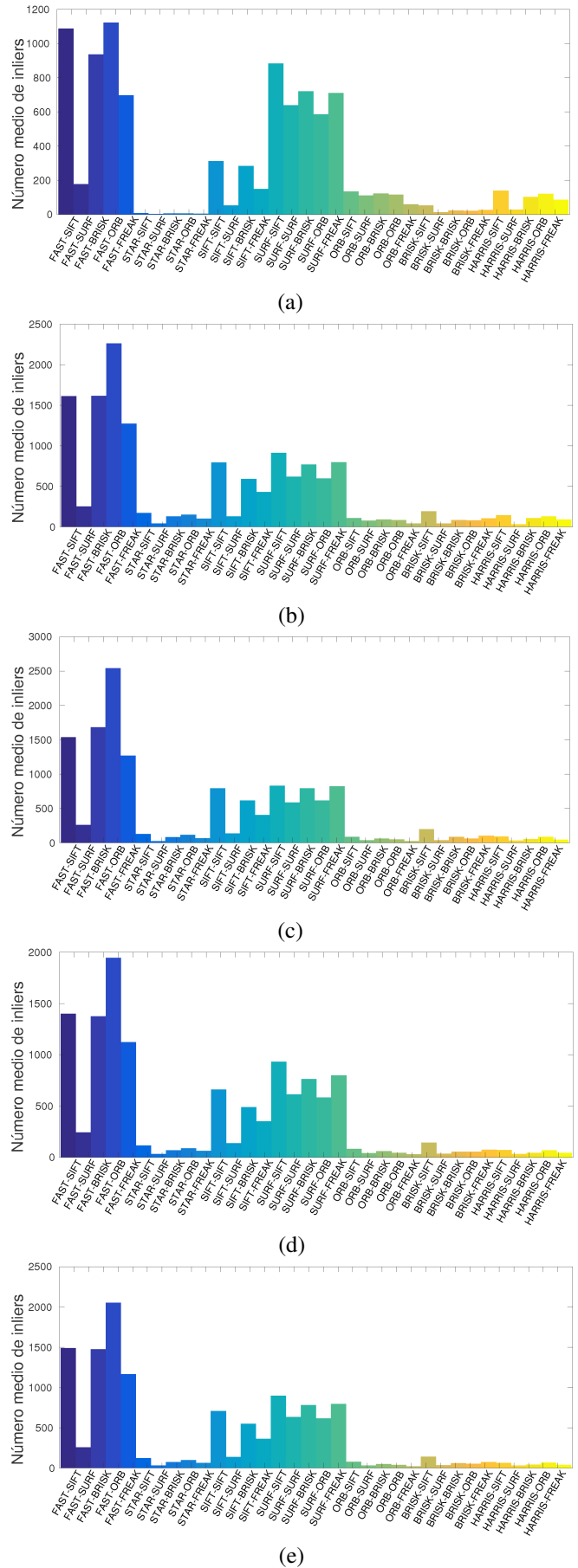


Figura 2: Número medio de *inliers* entre imágenes consecutivas de la secuencia 1. (a) Imágenes originales. (b) LCC. (c) MSR-NK. (d) MAI. (e) MSR.



b) El total y el número medio de *inliers* entre todos los pares de imágenes consecutivas de cada secuencia, desde la primera hasta la última, donde la segunda imagen de un par siempre es la primera imagen del siguiente par, para todas las combinaciones de características enumeradas en la sección 3 y para todas las mejoras de contraste indicadas en la sección 4, excepto CLAHS; c) En cada secuencia se calcularon el total y el número medio de *inliers* entre unas cuantas imágenes de referencia seleccionadas (*queries*) y otras imágenes de la secuencia que cerraban bucle con las de referencia en instantes distintos, obviando las 10 imágenes anteriores y posteriores a la de referencia, para cada combinación detector/descriptor y para cada mejora de contraste; d) Se calculó también el número medio de *inliers* entre las imágenes que no cerraban bucle con la de referencia.

La figura 2 muestra el número medio de *inliers* entre imágenes consecutivas, a lo largo de toda una secuencia de ejemplo (secuencia 1) de la cual se extrajeron 400 imágenes, para cada combinación de detector/descriptor, para las imágenes originales (gráfica 2-(a)) y para las distintas técnicas de aumento de contraste: la gráfica 2-(b) para LCC, la gráfica 2-(c) para MRSNK, la gráfica 2-(d) para MAI y, finalmente, la gráfica 2-(e) para MSR. Se puede observar como las combinaciones FAST\_SIFT, FAST\_BRISK, FAST\_ORB y SURF\_SIFT son las que dan un número medio de *inliers* mayor a lo largo de toda la secuencia, y que las combinaciones con el detector STAR, BRISK, ORB o HARRIS presentan un número medio de *inliers* mucho menor, independientemente de la mejora de contraste aplicada. Los filtros que, en la mayoría de combinaciones, aumentan más el número de *inliers* son LCC y MSR-NK. La combinación SIFT\_SIFT genera unos 300 *inliers* de media en la secuencia original y entre 500 y 600 en la secuencia con el contraste mejorado. Aunque no es la combinación que produce más correspondencias correctas, presenta unos números más que suficientes para garantizar un cálculo fiable tanto de la odometría del vehículo como en el registro de imágenes que cierran bucles. Aunque el cómputo de características SIFT y la búsqueda de correspondencias suele ser mucho más lento que otras como SURF, su robustez y fiabilidad ha sido ampliamente probada en múltiples aplicaciones. Por otro lado, los descriptores binarios como ORB o BRISK, combinados con el detector FAST reducen considerablemente el tiempo de proceso sin disminuir su robustez y fiabilidad en los procesos de *tracking* y *matching* (Ferreira et al., 2016), aspecto importantísimo a la hora de ser aplicados online.

La figura 3 muestra 4 imágenes de la secuencia 1, tomadas al principio, en la mitad y al final de la trayectoria.

La figura 4 muestra el incremento del número medio de *inliers* para la secuencia 1 filtrada con LCC respecto a la original, en función de la combinación detector/descriptor. Existe un aumento del número de *inliers* cercano al 23 % en las combinaciones con el detector STAR, mientras que para el resto de combinaciones el incremento es, como máximo, cercano al 5 % en el caso de BRISK. LCC es el filtro que genera un mayor incremento de *inliers*, en todas las combinaciones. Aunque el incremento para STAR sea de un 23 %, el número de *inliers* resultante sigue siendo bajo comparado con algunas de las combinaciones que usan los detectores SURF, SIFT o FAST.

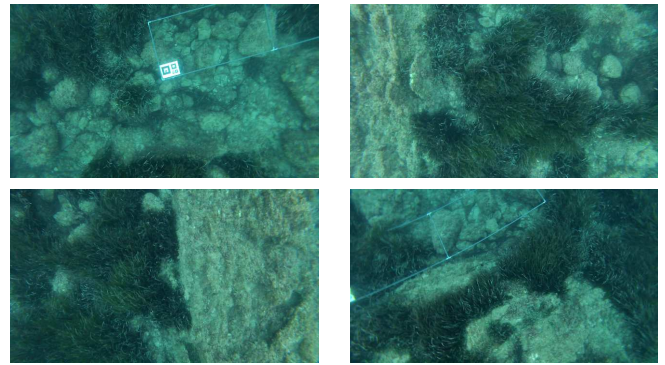


Figura 3: Algunas imágenes originales de la secuencia ejemplo 1.

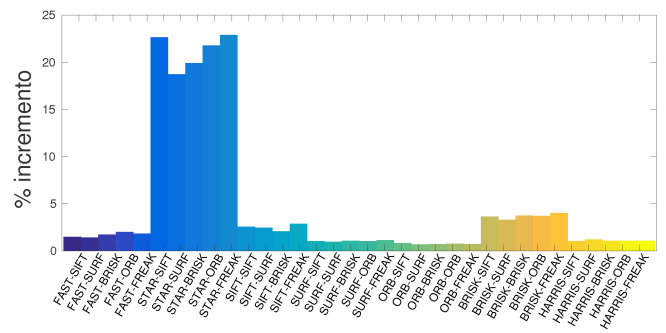


Figura 4: Incremento del número medio de *inliers* (secuencia 1) con filtro LCC.

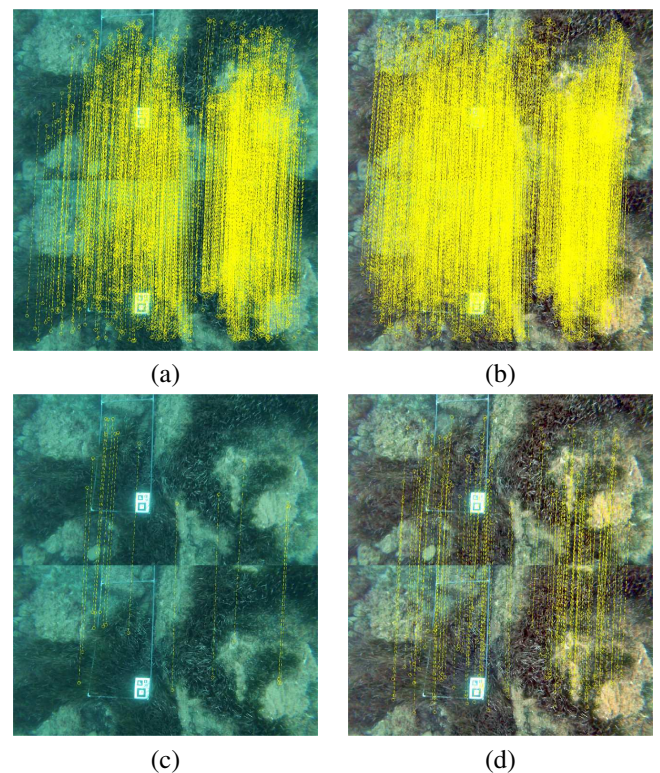


Figura 5: Correspondencias entre características de dos imágenes consecutivas de la secuencia ejemplo 1. (a)-(b) FAST\_ORB. (c)-(d) STAR\_SIFT

Las imágenes 5-(a) y 5-(b) muestran las correspondencias entre características visuales FAST\_ORB, para dos imágenes



consecutivas de la secuencia 1 sin filtrar y filtradas con LCC, respectivamente. La imagen original presenta un total de 2282 *inliers* mientras que la filtrada con LCC 4293. El número de correspondencias aparecidas en las imágenes originales ya es suficientemente alto como para trazar una odometría visual suficientemente estable. Las imágenes 5-(c) y 5-(d) muestran las correspondencias entre características STAR\_SIFT, para las imágenes originales y las filtradas con LCC, respectivamente. El número de puntos STAR en las imágenes originales es de 16, mientras que en las filtradas es de 230. 16 puntos no es suficiente para obtener una matriz  $H$  convergente en los procesos iterativos de eliminación de *outliers*. La consecuencia más inmediata es que estas matrices no representan la transformación de todos los puntos entre las dos imágenes con un error menor a un umbral mínimo establecido, aumentando la deriva en el cálculo visual del desplazamiento del robot. Se puede observar en la figura 5-(c) como, además de haber pocas correspondencias, algunas de ellas presentan un error evidente al intentar unir el mismo punto de la escena. Este error en las correspondencias se genera en el proceso de *matching* de descriptores e influye negativamente a la hora de calcular la matriz  $H$ .

es una cantidad considerable para obtener una aproximación de  $H$  suficientemente fiable, todavía existe alguna correspondencia errónea que puede redundar en un ligero aumento de la deriva en el cálculo visual del movimiento del robot.

La figura 6 muestra el número medio de *inliers* entre imágenes consecutivas, a lo largo de otra secuencia de ejemplo (secuencia 2) de la cual se extrajeron 336 imágenes. Los datos corresponden a cada combinación de detector/descriptor, para las imágenes originales (6-(a)) y para las imágenes filtradas con LCC (6-(b)) y MSR-NK (6-(c)).

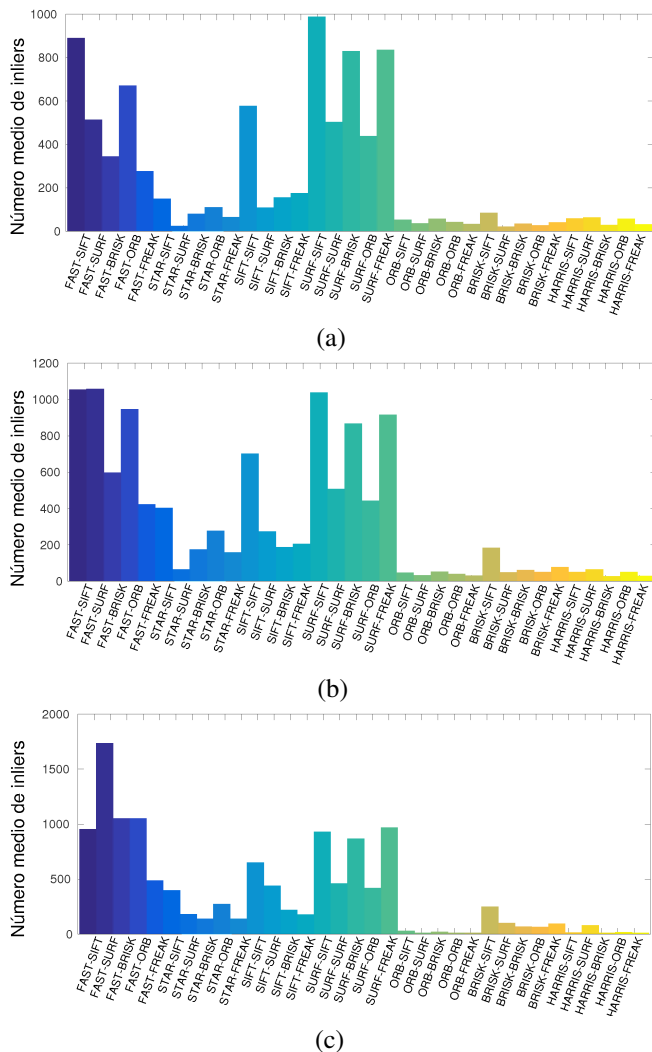


Figura 6: Número medio de *inliers* entre imágenes consecutivas de la secuencia 2. (a) Imágenes originales. (b) LCC. (c) MSR-NK.

Aunque los 230 puntos que aparecen en la figura 5-(d) ya

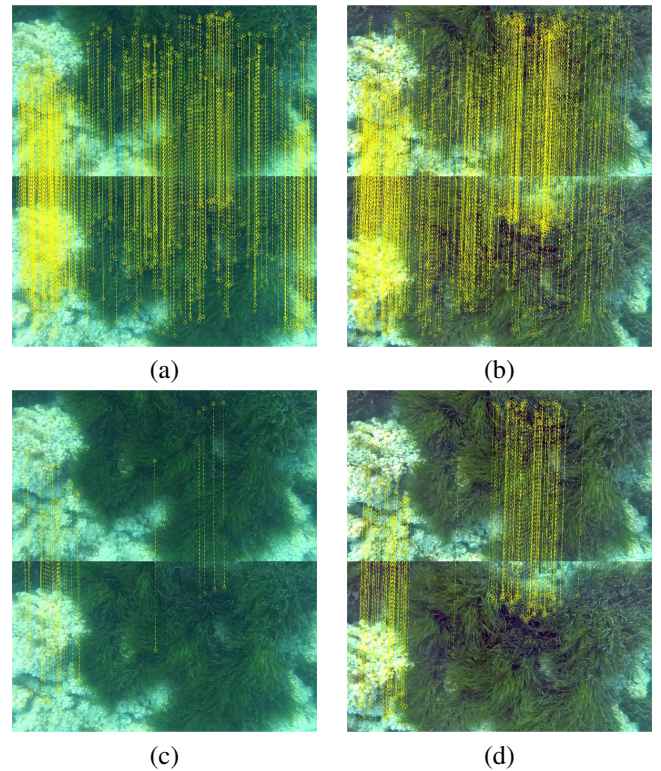


Figura 7: Correspondencias entre características de dos imágenes consecutivas de la secuencia 2. (a)-(b) SURF\_SIFT. (c)-(d) STAR\_SIFT

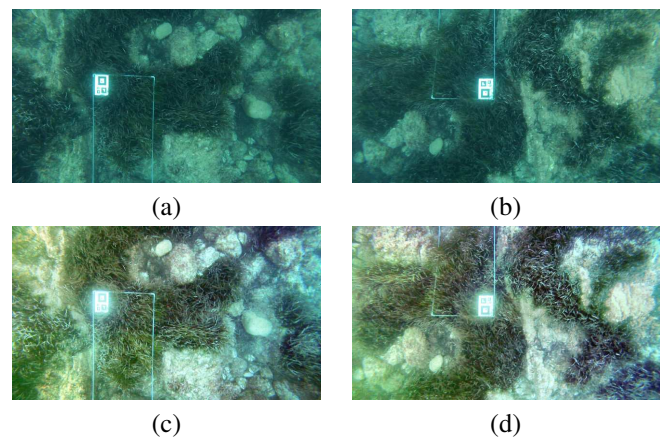


Figura 8: Dos imágenes originales de la secuencia 1 que cierran bucle ((a)-(b)), y las mismas imágenes filtradas con CLAHS ((c)-(d)).

Las mejores combinaciones vuelven a implicar a los detectores FAST y SURF con casi todos los descriptores y la combinación SIFT\_SIFT. La aplicación del filtro LCC también impli-

ca un incremento de *inliers* en todas las combinaciones, especialmente en aquellas con el detector FAST y en SURF\_SIFT, STAR\_ORB y SIFT\_SIFT. Las figuras 7-(a) y 7-(b) muestran los *inliers* entre dos imágenes consecutivas de la secuencia 2, para la combinación SURF\_SIFT aplicada en las imágenes originales (673 puntos) y en las filtradas con LCC (760 puntos), respectivamente. Las figuras 7-(c) y 7-(d) muestran los *inliers* para la combinación STAR\_SIFT aplicada a las imágenes originales (41 puntos) y las filtradas con LCC (171 puntos), respectivamente. En este ejemplo, el error entre las correspondencias es mínimo. El notable aumento de *inliers* respecto los 41 iniciales, al aplicar LCC, sugiere su uso continuado con el fin de aumentar la convergencia en el cálculo de  $H$ .

La figura 8 muestra 2 imágenes que cierran un bucle en la secuencia 1, originales y filtradas con CLAHS.

La figura 9-(a) muestra el número medio de *inliers*, para cada tipo de característica, entre la imagen de referencia 223 de la secuencia 1 sin filtrar y el resto de imágenes de la secuencia que cierran (barras azules), y que no cierran (barras amarillas) bucles con ella.

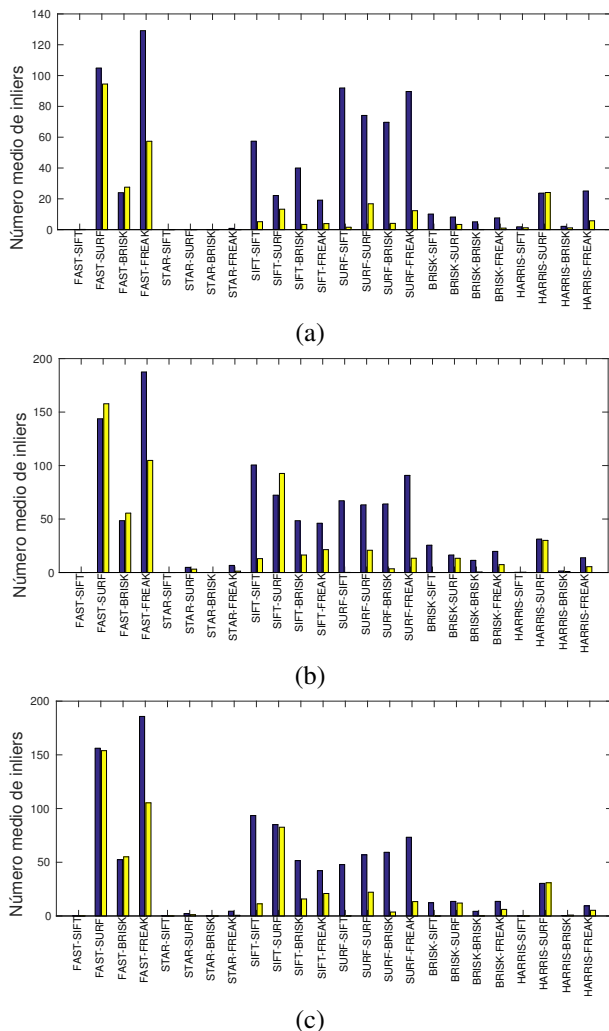


Figura 9: Media de *inliers* entre imágenes que cierran (azul) y no cierran (amarillo) bucle con la 223. (a) Originales, y filtradas con (b) LCC, y (c) CLAHS.

Las figuras 9-(b) y 9-(c) muestran los mismos resultados pero sobre las imágenes filtradas con LCC y CLAHS. Las combinaciones que son interesantes son aquellas que presentan un

número de *inliers* razonablemente alto entre imágenes que cierran bucles, pero muy bajo entre imágenes que no los cierran. Las combinaciones que cumplen mejor estos requisitos son: SIFT\_SIFT, SURF\_SIFT, SURF\_FREAK y SURF\_BRISK. Las combinaciones que peor lo cumplen, y por lo tanto no serían adecuadas para la detección de bucles y el registro de imágenes, serían todas aquellas que incluyen los detectores STAR y HARRIS. El número de *inliers* entre imágenes que cierran bucles con la 223 para la combinación FAST\_FREAK es alto, pero entre imágenes que no cierran bucles también es alto, por lo tanto, no sería una buena combinación para este tipo de aplicaciones.

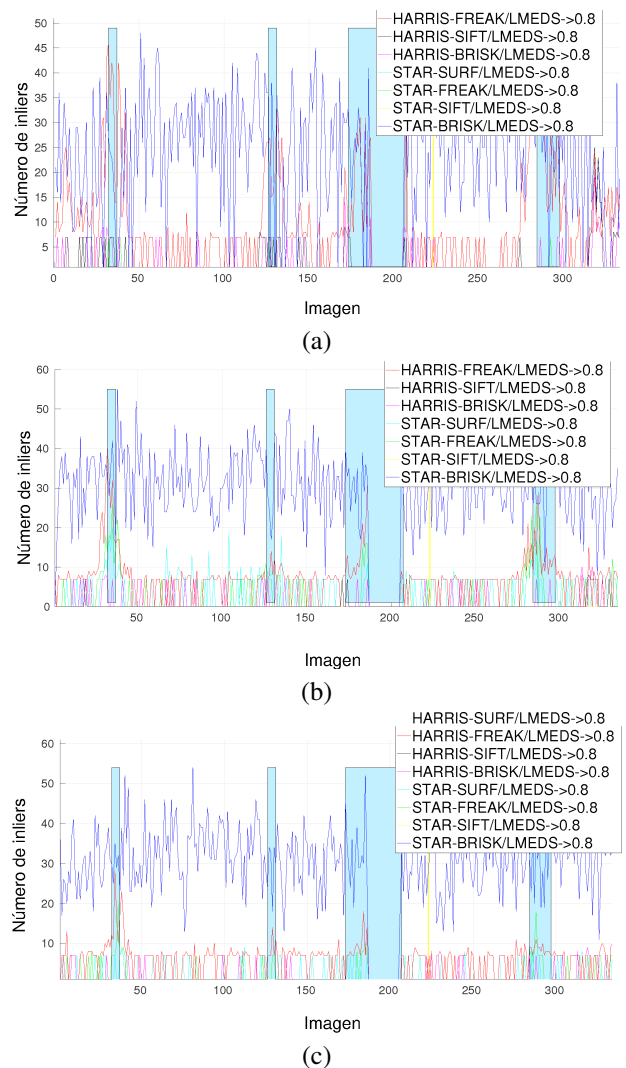


Figura 10: Total de *inliers* entre la 223 y el resto de imágenes (secuencia 1). (a) Originales, y filtradas con (b) LCC y (c) CLAHS. Con STAR y HARRIS.

Aunque la aplicación de mejoras de contraste aumenta el número de *inliers*, este aumento no es significativamente mayor cuando se aplica LCC o CLAHS, y la tendencia entre el número de correspondencias cuando hay cierre de bucle o no lo hay es la misma en las tres gráficas.

La figura 10 muestra el número total de *inliers* entre la imagen de referencia 223 y el resto de imágenes de la secuencia 1, usando los detectores STAR y HARRIS y los descriptores SURF, FREAK, SIFT y BRISK. Las franjas verticales azules muestran el rango de imágenes que cierran bucle con la 223.



Se puede observar como el número de *inliers* oscila alrededor de 30 para la combinación STAR\_BRISK y 5, para todo el resto, sin presentar ningún máximo claramente destacable en las zonas donde se cierran bucles. No es posible discriminar las imágenes que cierran bucle con la 223 de las que no lo hacen a partir del número de *inliers*. Estos dos detectores son claramente inadecuados para el proceso de registro de imágenes. Aunque la aplicación de mejoras de contraste aumenta el número de *inliers*, la tendencia es la misma a la comentada en el párrafo anterior.

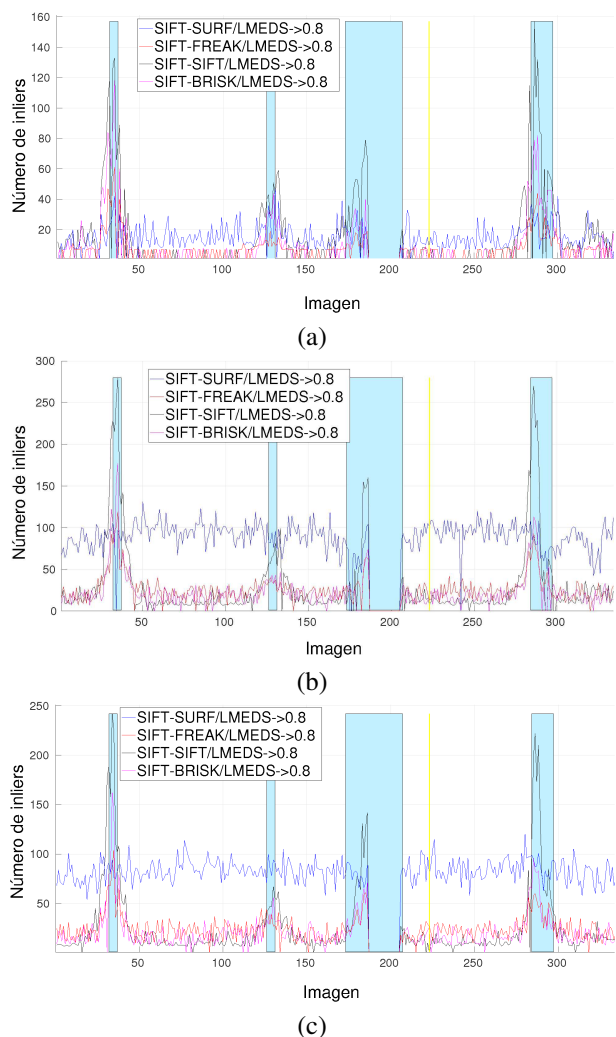


Figura 11: Total de *inliers* entre la 223 y el resto de imágenes (secuencia 1). (a) Originales, y filtradas con (b) LCC y (c) CLAHS. Con SIFT.

La figura 11 muestra el número de *inliers* entre la imagen de referencia 223 y el resto de imágenes de la secuencia 1, usando el detector SIFT con los descriptores SURF, FREAK, SIFT y BRISK. En este caso, la combinación SIFT\_SURF sigue el mismo patrón que STAR y HARRIS. Sin embargo, las combinaciones SIFT\_FREAK y SIFT\_BRISK presentan 2 máximos claramente diferenciados alrededor de las imágenes 30 y 280, y otros dos máximos no demasiado diferenciados alrededor de las imágenes 130 y 180. Finalmente, la combinación SIFT\_SIFT (curva de color negro) muestra 4 picos con un número de *inliers* significativamente mayor en las zonas donde hay cierre de bucles que donde no los hay. Por lo tanto, esta combinación es claramente adecuada para el registro de imágenes que cierran

bucles.

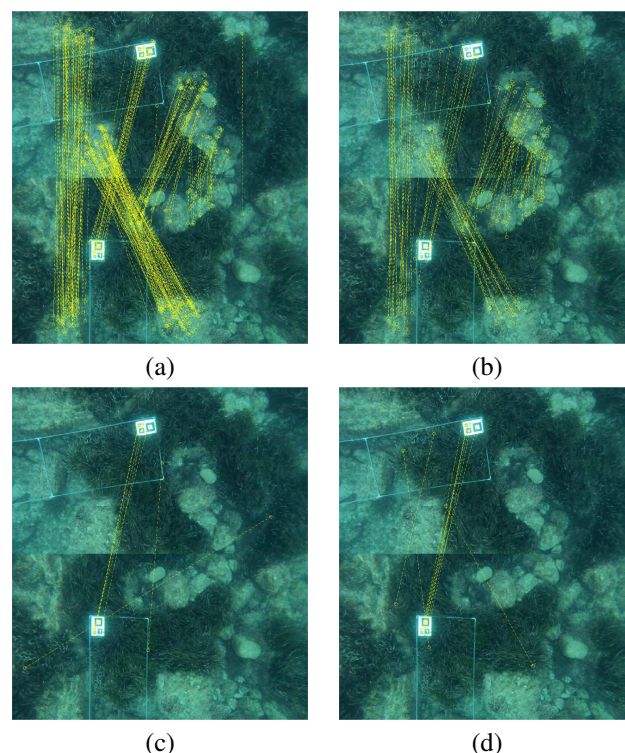


Figura 12: *Feature matchings* entre dos imágenes que cierran bucle. (a) FAST\_FREAK, (b) SURF\_SIFT, (c) HARRIS\_BRISK, (d) HARRIS\_SURF.

La figura 12 muestra los *inliers* entre dos imágenes originales que cierran bucle en la secuencia 1, con características tipo: (a) FAST\_FREAK (314 puntos), (b) SURF\_SIFT (205 puntos), (c) HARRIS\_BRISK (9 puntos) y (d) HARRIS\_SURF (28 puntos). En (c), el número de correspondencias es tan pequeño que solo se pueden calcular dos iteraciones de la matriz  $H$ . Esto complica la obtención de una transformación fiable que modele el movimiento entre todos los puntos de las dos imágenes. En el caso de (d), aunque el número de *inliers* es mayor, hay muchos falsos positivos. Sin embargo, en los casos (a) y (b), el número de *inliers* es considerablemente mayor y los falsos positivos son prácticamente inexistentes. En este último caso, dado que todas las correspondencias coinciden en emparejar los mismos puntos de la escena, la matriz  $H$  ajustará el movimiento de la cámara con mucha más precisión.

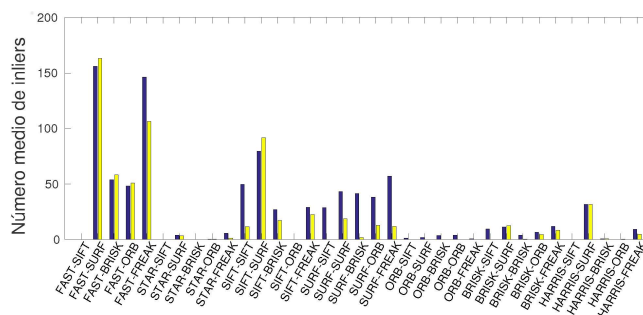


Figura 13: Media de *inliers* entre imágenes que cierran (azul) y no cierran (amarillo) bucle con la image 10 de la secuencia 1, con filtro LCC.

La figura 13 muestra el número medio de *inliers* para ca-

da tipo de característica, entre la imagen de referencia 10 de la secuencia 1 filtrada con LCC y el resto de imágenes que cierran (barras azules) y no cierran (barras amarillas) bucles con ella. Las combinaciones que ofrecen una diferencia mayor entre la cantidad de *inliers* entre imágenes que cierran bucles e imágenes que no cierran bucles son de nuevo: SIFT\_SIFT, SURF\_SIFT, SURF\_FREAK y SURF\_BRISK. Las peores combinaciones incluyen aquellas con los detectores STAR, HARRIS y ORB.

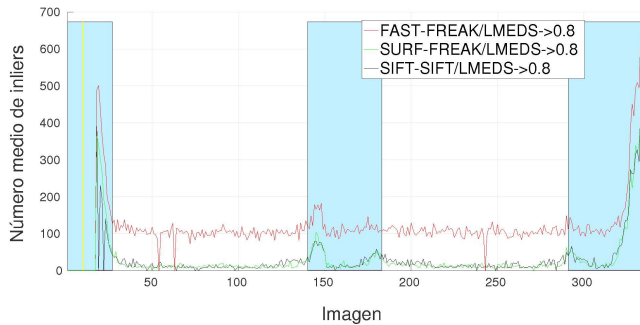


Figura 14: Total de *inliers* entre la 10 y el resto de imágenes (secuencia 1), sin filtrar, y para las combinaciones FAST\_FREAK, SURF\_FREAK y SIFT\_SIFT.

La figura 14 muestra el mismo comportamiento obtenido para la imagen de referencia 223. La combinación FAST\_FREAK muestra picos en las áreas donde se cierran bucles (barras azules), pero el número de *inliers* donde no se cierran bucles también es alto. Sin embargo, el número de *inliers* para las combinaciones SURF\_FREAK y SIFT\_SIFT presenta picos importantes en las zonas donde se cierran bucles, y valores muy pequeños en las zonas donde no se cierran bucles, facilitando su discriminación. Esta gráfica se ha obtenido con las imágenes filtradas con LCC.

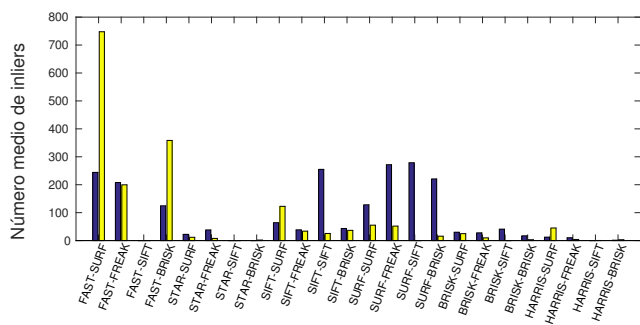


Figura 15: Media de *inliers* entre imágenes que cierran (azul) y no cierran (amarillo) bucle con la imagen 336 de la secuencia 2, para cada tipo de característica.

La figura 15 muestra el número medio de *inliers* entre la imagen de referencia 336 de la secuencia 2 filtrada con LCC y aquellas que cierran (barras azules) y no cierran (barras amarillas) bucle con ella. En este caso, aunque el número medio de *inliers* en la combinación FAST\_FREAK llega a los 200, esta combinación es completamente inútil para discriminar imágenes que cierran bucles de las que no, ya que el número de *inliers* entre imágenes que no cierran bucles es comparable a la cantidad de *inliers* entre imágenes que si cierran bucles. Sin embargo, las combinaciones SURF\_FREAK, SURF\_SIFT, SIFT\_BRISK y SIFT\_SIFT vuelven a presentar gran cantidad

de *inliers* para imágenes que cierran bucles y muy pocos para las que no. Esto las convierte en el tipo de característica más adecuada para la discriminación y registro de imágenes que cierran bucles. Por otro lado, las combinaciones que incluyen al detector HARRIS y STAR son las peores en este aspecto. Los resultados de las combinaciones que incluyen el detector y/o descriptor ORB se han obviado para reducir la cantidad de datos que se muestran en la gráfica, ya que los resultados obtenidos con este detector son irrelevantes.

La figura 16 muestra el número de *inliers* entre la imagen 336 de la secuencia 2, con cada una del resto de imágenes de la secuencia, usando las combinaciones SURF\_SIFT, SURF\_FREAK y SURF\_BRISK. Todas las imágenes se han filtrado usando LCC. Las tres gráficas muestran claramente como el número de *inliers* aumenta hasta 400 en el punto donde se cierra un bucle con la 336, mientras que permanece por debajo de 50 en el resto de imágenes con las que no se cierra bucle. En consecuencia, el uso de estas características permite discriminar con gran fiabilidad las imágenes que cierran lazo de las que no.

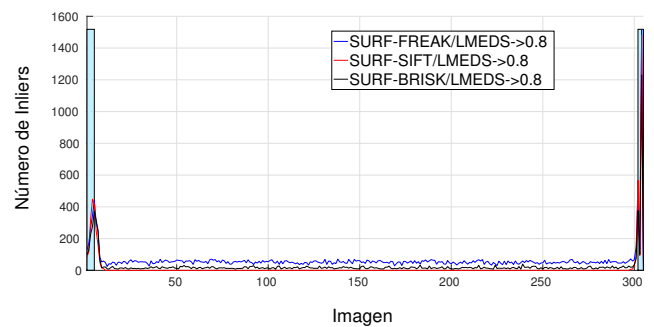


Figura 16: Total de *inliers* entre la 336 y el resto de imágenes (secuencia 2), filtradas con LCC, para SURF\_SIFT, SURF\_FREAK y SURF\_BRISK.

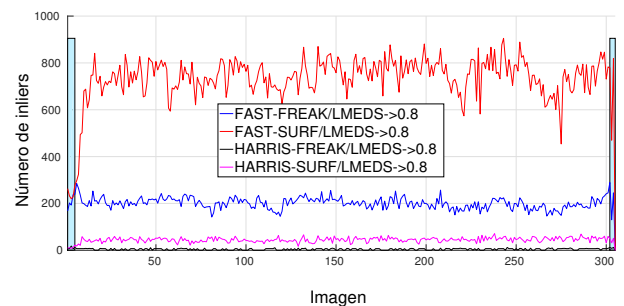


Figura 17: Total de *inliers* entre la 336 y el resto de imágenes (secuencia 2), filtradas con LCC, para FAST\_FREAK, FAST\_SURF, HARRIS\_FREAK y HARRIS\_SURF.

La figura 17 muestra los mismos datos que la figura 16 pero usando las combinaciones FAST\_FREAK, FAST\_SURF, HARRIS\_FREAK y HARRIS\_SURF. Estas combinaciones se vuelven a mostrar inútiles para discriminar imágenes que cierran bucles ya que el número de *inliers* entre la 336 y el resto oscila alrededor de varios valores fijos, sin ningún pico relevante que indique un número de correspondencias significativamente mayor en alguna de las imágenes que cierra un bucle.

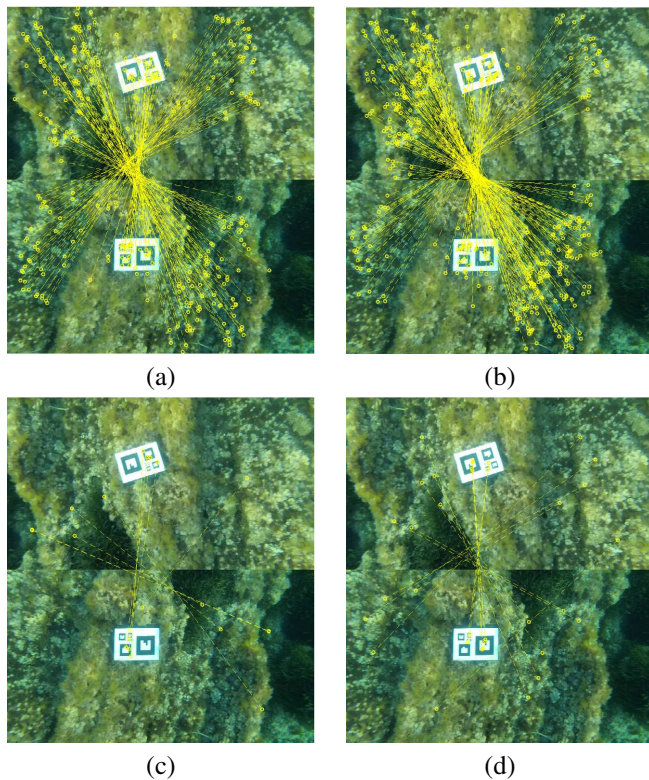


Figura 18: Correspondencias entre características de dos imágenes que cierran bucle. (a) SIFT\_SIFT, (b) SURF\_SIFT, (c) BRISK\_SURF, (d) STAR\_SURF.

La figura 18 muestra las correspondencias entre las imágenes originales 336 y 4, usando 4 tipos de características distintas: (a) SIFT\_SIFT (381 puntos), (b) SURF\_SIFT (545 puntos), (c) BRISK\_SURF (33 puntos), y (d) STAR\_SURF (35 puntos). En los casos (a) y (b) el número de correspondencias es suficientemente alto y éstas son suficientemente precisas como para garantizar una  $H$  que modele la transformación entre todos los puntos de la imagen. Sin embargo, las imágenes mostradas en (c) y (d) presentan pocos *inliers* y alguno de ellos son falsos positivos. La inclusión de falsos positivos en el cálculo de la matriz  $H$  puede dar lugar a transformadas entre las dos imágenes poco fiables o incorrectas, y en consecuencia a un posterior cálculo incorrecto del movimiento del robot, generando una deriva importante.

## 6. Conclusiones y trabajo futuro

El uso de características visuales (*features*) suficientemente estables, robustas y fácilmente identificables en imágenes marinas que solapan, sean consecutivas o que cierran bucles, es esencial para el cálculo visual del movimiento de un vehículo acuático. Los entornos marinos colonizados con Posidonia presentan, además de la problemática inherente al mar (refracción, reflexiones, turbidez, absorción, etc...) texturas visuales profundamente complejas y exigentes. En el contexto del proyecto ARSEA, donde un AUV tiene que navegar específicamente sobre praderas de Posidonia, se requiere decidir que características visuales se deben utilizar para aumentar la fiabilidad en la estimación visual de la trayectoria del robot.

De los extensos resultados experimentales presentados en este artículo, se pueden destacar 6 puntos: a) las mejores com-

binaciones en cuanto a número de correspondencias *inliers* entre imágenes consecutivas son las que incluyen los detectores FAST (excepto FAST\_SURF) y SURF y la combinación SIFT\_SIFT, b) las combinaciones que presentan un incremento mayor del número de *inliers* son las que tienen el detector STAR; aunque el número medio de *inliers* con el detector STAR y filtrado LCC no pasa de 85, hay imágenes con más de 200 *inliers*, lo que fuerza a reconsiderar su uso si se combina con el filtro LCC, c) las combinaciones que incluyen los detectores ORB, HARRIS o BRISK no deben usarse en estos entornos ya que el número de correspondencias es muy pobre, d) las combinaciones que siempre presentan mayor número de correspondencias en imágenes que cierran bucles y menor número en imágenes que no los cierran son SIFT\_SIFT, SURF\_SIFT, SURF\_BRISK y SURF\_FREAK, e) las combinaciones que, por lo tanto, presentan un buen comportamiento, tanto entre imágenes consecutivas como en imágenes que cierran bucles son: SIFT\_SIFT, SURF\_SIFT, SURF\_BRISK y SURF\_FREAK, f) las mejoras de contraste se han demostrado útiles a la hora de aumentar el número de *features*, pero, exceptuando el caso de algunas combinaciones con el detector STAR (STAR\_SIFT o STAR\_ORB), esta mejora no implica cambios en las tendencias argumentadas en los párrafos anteriores.

Como trabajo futuro se plantean dos retos: 1) evaluar la calidad de las transformadas obtenidas a partir de las matrices  $H$  como una primera estimación del movimiento del robot; aunque los algoritmos de odometría visual suelen incluir cálculos adicionales a la simple concatenación de sucesivas homografías, la comparación de estas transformadas obtenidas con las diferentes combinaciones y con alguna muestra de *Ground Truth* puede confirmar o matizar algunas de las conclusiones aquí expuestas, 2) puede ocurrir que alguno de los *inliers* obtenidos en cualquier proceso de *feature matching/tracking* sea un falso positivo, y que dentro de los *outliers* descartados haya algún falso negativo; este diagnóstico se debe hacer por inspección visual, y esto requiere un estudio minucioso de todo un conjunto amplio de pares de imágenes que solapan y sus correspondencias; el ratio de falsos *inliers*, falsos *outliers*, verdaderos *inliers* y verdaderos *outliers*, para cada combinación detector/descriptor, ayudaría a complementar las medidas de calidad de cada característica visual en este tipo de entornos con parámetros como la precisión (definida como el cociente entre el número de verdaderos positivos y el número total de puntos clasificados como positivos), razón de verdaderos positivos respecto el total de correspondencias ciertamente positivas (*recall*) o la razón de falsos positivos respecto el total de correspondencias ciertamente negativas (*fall out*). Con todas estas medidas se pueden construir curvas ROC (Hanley y Neil, 1982) que definen la calidad de las correspondencias, y a partir de estas curvas descartar o aceptar con argumentos adicionales y más contundente las distintas opciones.

Es obvio que esta metodología de estudio puede generalizarse, ejecutándose en otros tipo de entornos predominantemente cubiertos con otro tipo de vegetación, en los que la fisonomía del fondo (sea el hábitat dominante de la Posidonia o de cualquier otra especie) requiera un tratamiento visual seguro y robusto para garantizar una localización y navegación fiables.



## Agradecimientos

Este trabajo está parcialmente financiado por el Ministerio de Economía y Competitividad a través del proyecto TIN2014-58662-R y fondos FEDER.

## Referencias

- Bonin-Font, F., Massot, M., Negre, P. L., Oliver, G., Guerrero, E., Garcia, E., 2017. Towards a new Methodology to Evaluate the Environmental Impact of a Marine Outfall Using a Lightweight AUV. En: In MTS/IEEE Oceans.
- Bonin-Font, F., Massot-Campos, M., Oliver, G., 2016. Towards Visual Detection, Mapping and Quantification of *Posidonia Oceanica* using a Lightweight AUV. En: Proc. of IFAC Conference on Control Applications in Marine Systems. pp. 500–505.
- Burguera, A., Bonin-Font, F., Oliver, G., 2015. Trajectory-Based Visual Localization in Underwater Surveying Missions. *Sensors*, MDPI 15 (1), 1708–1735.
- Carreras, M., Candela, C., Ribas, D., Mallios, A., Magaña, L., Vidal, E., Palomeras, N., Ridaio, P., 2013. SPARUS II, Design of a Lightweight Hovering AUV. Fifth International Workshop in Marine Technology (MARTECH).
- Diaz-Almela, E., Duarte, C., 2008. Management of *Natura 2000* Habitats 1120, (*Posidonia Oceanica*). Tech. rep., European Commission.
- Eustice, R., Pizarro, O., Singh, H., April 2008. Visually Augmented Navigation for Autonomous Underwater Vehicles. *IEEE Journal of Oceanic Engineering* 33 (2), 103–122.
- Eustice, R., Pizarro, O., Singh, H., Howland, J., 2002. UWIT: Underwater Image Toolbox for Optical Image Processing and Mosaicking in MATLAB. En: Proceedings of IEEE International Symposium on Underwater Technology, pp. 141–145.
- Ferreira, F., Veruggio, G., Caccia, M., Bruzzone, G., 2016. A Survey on Real-time Motion Estimation Techniques for Underwater Robots. *Journal of Real Time Image Processing* 11 (4), 693–711.
- Fischler, M., Bolles, R., 1981. Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24 (6), 381–395. DOI: <http://doi.acm.org/10.1145/358669.358692>
- Geiger, A., Ziegler, J., Stiller, C., June 2011. Stereoscan: Dense 3d reconstruction in real-time. En: IEEE Intelligent Vehicles Symposium. Baden-Baden, Germany.
- Hanley, J., Neil, B. M., 1982. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* 143 (1), 521–539.
- Hartley, R., Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- Jobson, D., Rahman, Z., Woodell, G., 1997. A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes. *IEEE Transactions on Image Processing* 6 (7), 965–976.
- Jorda, G., Marba, N., Duarte, C., 2012. Mediterranean Seagrass Vulnerable to Regional Climate Warming. *Nature Climate Change*, 821–824.
- Krig, S., 2014. *Computer Vision Metrics*. Springer, Ch. Interest Point Detector and Feature Descriptor Survey, pp. 217–282.
- Lauga, P., Valenzise, G., Chierchia, G., Dufaux, F., September 2014. Improved Tone Mapping Operator for HDR Coding Optimizing the Distortion/Spatial Complexity Trade-off. En: Proceedings of IEEE European Signal Processing Conference. pp. 1607–1611.
- Li, Y., Wang, S., Tian, Q., Ding, X., 2015. A Survey of Recent Advances in Visual Feature Detection. *Neurocomputing (B)*, 736–751.
- Maida, G. D., Tomasello, A., Luzzu, F., Scannavino, A., Pirrotta, M., Orestano, C., Calvo, S., 2011. Discriminating Between *Posidonia Oceanica* Meadows and Sand Substratum Using Multibeam Sonar. *ICES Journal of Marine Science* 68 (1), 12–19.
- Matarrese, R., Acquaro, M., Morea, A., Tijani, K., Chiaradia, M., 2008. Applications of Remote Sensing Techniques for Mapping *Posidonia Oceanica* Meadows. En: Proceedings of IEEE International Geoscience and Remote Sensing Symposium. pp. 906–909.
- Montefalcone, M., Rovere, A., Parravicini, V., Albertelli, G., Morri, C., Bianchi, C. N., 2013. Evaluating Change in Seagrass Meadows: A time-framed Comparison of Side Scan Sonar Maps. *Aquatic Botany* 104, 204–212.
- Moore, A., Allman, J., Goodman, R. M., 1991. A Real-time Neural System for Color Constancy. *IEEE Transactions on Neural Networks* 2 (2), 237–246.
- Morel, J., Petro, A. B., Sbert, C., 2014. What is the Right Center/Surround for Retinex? En: Proceedings of the International Conference on Image Processing (ICIP).
- Mur-Artal, R., Montiel, J., Tardos, J., October 2015. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* 31 (5), 1147–1163.
- Negre, P. L., Bonin-Font, F., Oliver, G., May 2016a. Cluster-Based Loop Closing Detection for Underwater SLAM in Feature-Poor Regions. En: Proc. of IEEE International Conference on Robotics and Automation (ICRA).
- Negre, P. L., Bonin-Font, F., Oliver, G., 2016b. Global Image Signature for Visual Loop-Closure Detection. *Autonomous Robots* 40 (8), 1403–1417.
- Scaradozzi, D., Conte, G., de Capua, G., Sorbi, L., Luciani, C., de Cecco, P., Sorci, A., 2009. Innovative Technology for Studying Growth Areas of *Posidonia Oceanica*. En: Proceedings of the IEEE WorkShop on Environmental, Energy and Structural Monitoring Systems. pp. 71–75.
- Shertzer, K., Prager, M., 2002. Least Median of Squares: A Suitable Objective Function for Stock Assessment Models. *Canadian Journal of Fisheries and Aquatic Sciences* 59 (9), 1474–1481 Vol.2.
- Short, F., Polidoro, B., Livingstone, S., Carpenter, K., Bandeira, S., Bujang, J., Calumpang, H., Carruthers, T., Coles, R., Dennison, W., Erftemeijer, P., Fortes, M., Freeman, A., Jagtap, T., Kamal, A., Kendrick, G., Kenworthy, W., Nafie, Y. L., Nasution, I., Orth, R., Prathep, A., van Sanchiangco, J., Tusssenbroek, B., Vergara, S., Waycott, M., Zieman, J., 2012. Extinction Risk Assessment of the World Seagrass Species Biological Conservation. *Canadian Journal of Fisheries and Aquatic Sciences* 144 (1961–1971).
- Wilow-Garage, 2014. Open Source Computer Vision (Open Cv). <http://docs.opencv.org/>, function findhomography.