Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Visual Place Recognition

Autor: Leandro Candau Sánchez de Ybargüen
Tutor: Begoña Chiquinquira Arrue Ulles

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Carrera
Grado de Ingeniería en Tecnologías Industriales

Visual Place Recognition

Autor:
Leandro Candau Sánchez de Ybargüen

Tutor:
Begoña Chiquinquira Arrue Ulles

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2019

Trabajo de Fin de Grado: Visual Place Recognition

Autor: Leandro Candau Sánchez de Ybargüen
Tutor: Begoña Chiquinquira Arrue Ulles

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

AGRADECIMIENTOS

A Paola y mi familia, por su apoyo incondicional a lo largo de los años.

A mis maestros por sus enseñanzas y dedicación.

RESUMEN

El objetivo de este trabajo de fin de grado consiste en estudiar y probar las tecnologías actuales que permiten el reconocimiento visual de un lugar previamente visitado. Este proceso se conoce como visual place recognition y es un problema bien definido en el ámbito de la visión artificial que se lleva estudiando desde hace varios años.

Visual place recognition es una técnica utilizada para conocer con precisión la posición de la cámara que captura las imágenes, usualmente integrada como un sensor de un robot móvil. Consiste en reconocer un lugar previamente visto, y triangular la distancia a este lugar para obtener la posición relativa a la del observador.

Visual place recognition permite, junto con otras técnicas de mapeado y estimación de posición, construir mapas sin tener conocimiento previo del terreno y dar soporte a sistemas de navegación autónoma. Existen diversas técnicas para realizar el reconocimiento visual de un lugar, y para definir este lugar. En este proyecto se hará uso de algunos de estos algoritmos, analizando los resultados obtenidos.

ABSTRACT

The objective of this end of grade project is to study and test the current technologies that allow the visual recognition of a previously visited place. This process is known as visual place recognition and is a well defined problem in the area of artificial vision that has been subject of study for many years.

Visual place recognition is a technique employed to obtain a precise location and position of the camera that captures images, commonly integrated as a sensor in a mobile robot. This is done by visualizing a known place, one that has been seen before from a certain position, and triangulate its position relative to that of the viewer.

Visual place recognition helps to, in conjunction with other mapping and positioning techniques, build maps while not having any previous knowledge of the terrain, and give support to autonomous navigation systems. There are diverse techniques that implement a visual place recognition system, and define a place. The present project makes use of some of these algorithms, analyzing the results obtained.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Antecedentes

El presente proyecto se enmarca bajo la normativa de los Trabajos de Fin de Grado de la Escuela Técnica Superior de Ingeniería de Sevilla para la obtención de los créditos correspondientes a la asignatura "Trabajo Fin de Grado" de la titulación de grado en ingeniería en tecnologías industriales.

El departamento adjudicador del proyecto es el Departamento de Ingeniería de Sistemas y Automática, siendo la tutora Dña. Begoña Chiquinquira Arrue Ulles.

1.2 Objetivos

El problema de la detección, por medio de la visión artificial, de una escena o lugar es uno de los retos por resolver más estudiados, debido a la creciente necesidad del mercado y la industria de plataformas con capacidad de navegar de forma segura y autónoma. Algunas de las aplicaciones más demandadas son robots móviles, vehículos autónomos o sistemas de navegación por ciudades. Es un problema fácil de definir, pero muy difícil de resolver debido a las condiciones cambiantes del propio ambiente.

Esta problemática es tratada en numerosos artículos académicos y **proyectos,** y con el desarrollo de las recientes tecnologías de machine learning y deep learning, está teniendo de nuevo un auge en su **popularidad**.

La visión por medio de cámara es uno de los sistemas más sencillos y baratos de aplicar en un sistema, y con los años los sistemas de captura de imagen han mejorado tanto en calidad como en coste y tamaño, siendo dispositivos cada vez más sofisticados. Esto causa que las cámaras sean uno de los sensores más empleados en la industria, siendo algunos de estos dispositivos cámaras multiespectrales, cámaras RGB o cámaras infrarrojas. En la Figura 1.1 vemos una aplicación sencilla de visión artificial en el ámbito de la robótica, en este caso aplicando una cámara y un sensor láser para detectar objetos y su posición en una cinta transportadora, para poder extraerlos de esta. Sistemas similares son ampliamente empleados en la industria dada su rentabilidad y sencillez, fomentando el desarrollo de mejores dispositivos.

Figura 1.1 – Robots industriales empleando un sistema de reconocimiento de objetos con visión artificial.

El problema que se trata de resolver es; dada una imagen de un lugar, ver si nuestro sistema es capaz de

determinar si ese lugar ha sido visto anteriormente.

En este proyecto se definirá en detalle el concepto de percepción visual de un lugar, reconocimiento de este y se analizan algunos de los algoritmos clásicos empleados para la resolución del problema. Se evaluarán dichos conceptos teóricos mediante un sistema sencillo, empleando una webcam que permite a un PC tomar imágenes y procesarlas para tratar de determinar si corresponden al mismo lugar. Se escoge este sistema dado que el sensor empleado, la webcam, es fácil de obtener y emplear con un PC, y se puede calibrar para obtener resultados más fiables. Se contemplan las problemáticas relacionadas con el proceso como son los cambios de iluminación y posición para observar la respuesta de nuestro modelo frente a las mismas y se sacarán conclusiones de dicho estudio.

El objetivo del proyecto es estudiar los sistemas de procesado de imágenes mediante software en una de las aplicaciones más demandadas por el mercado, como es el reconocimiento de lugares, empleando para su resolución librerías de software libres ampliamente utilizadas, las librerías de OpenCV [1] con el lenguaje de programación C++. Dichas librearías permiten procesamientos complejos de imágenes que resulta de gran utilidad en aplicaciones como el entrenamiento de una red neuronal de machine learning, estando además optimizadas para garantizar la eficiencia computacional y permitir su uso para aplicaciones de tiempo real. Otro objetivo perseguido con la resolución de este proyecto es estudiar el problema del reconocimiento de lugar, y cómo se ha tratado de resolver a lo largo de los años.

1.3 Metodología

En este proyecto se ha llevado a cabo un estudio de los algoritmos clásicos de descripción de imagen extrayendo puntos de interés, aplicados en un sistema de reconocimiento de lugar. Para ello se ha desarrollado una aplicación software, cuyo esquema básico vemos en la Figura 1.2. Se ha desarrollado en el lenguaje de programación C++, empleando unas librerías populares de procesamiento de imagen *open source*, OpenCV, y desarrollando el software con la IDE Visual Studio de Microsoft.

Como se explicará en detalle más adelante, la aplicación tiene dos modos de funcionamiento: emplear una cámara, previamente calibrada, para capturar imágenes o cargar varias imágenes guardadas en memoria. Las imágenes almacenadas forman un dataset; es decir un conjunto de datos conocidos que se puede emplear para evaluar el rendimiento del sistema.

En el transcurso del se han estudiado los algoritmos SIFT y SURF de extracción de puntos característicos de imágenes y se han implementado en una aplicación con motivo de obtener un resultado visual del proceso.

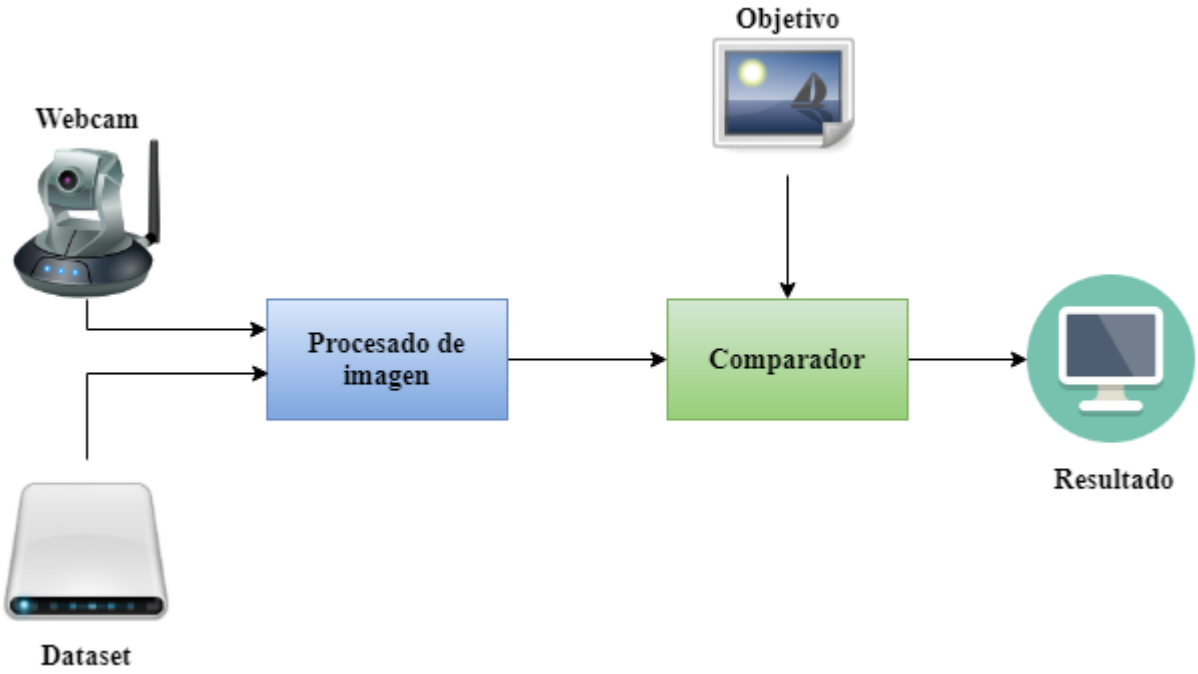

Figura 1.2 – Esquema de funcionamiento del sistema.

1.4 Estructura de la memoria

Para una mejor exposición del contenido de este proyecto, se define brevemente el contenido que abarcan los capítulos posteriores.

Se da una introducción teórica al problema de *visual place recognition* en el capítulo 2. Se define el problema del reconocimiento de lugar y se expone el modelo teórico de un sistema de reconocimiento de lugar empleando visión artificial. Se explican los módulos que componen dicho modelo en los apartados posteriores de dicho capítulo.

La descripción de la arquitectura empleada en la aplicación se define el en capítulo 3, desde una perspectiva general, mostrando los componentes en los que se divide el sistema. Se dará una introducción de las librerías empleadas, justificando su utilización en este proyecto. El dataset empleado es presentado aquí, describiendo cómo se ha generado y las razones para su uso.

Con la arquitectura general del sistema definida se pasa a definir el problema a resolver en el capítulo 4. Se definen los requisitos impuestos a las diversas partes del sistema. Posteriormente se explica la resolución del problema planteado para los dos modos de funcionamiento de la aplicación.

Los resultados obtenidos se analizan en el capítulo 5, estudiando los problemas descritos en la definición teórica del capítulo 2, y cómo responde el sistema ante ellos.

Por último, las conclusiones sacadas y las lecciones aprendidas se enumeran en el capítulo 6, así como los problemas que han surgido en el transcurso del proyecto y las mejoras propuestas.

2 VISUAL PLACE RECOGNITION

2.1 Introducción

En el presente apartado se dará un contexto teórico del problema de reconocimiento de lugar. En primera instancia, definiremos el concepto de lugar viendo algunos ejemplos de estudios sobre la manera de describir el espacio que emplean los animales y los humanos. Se detallará cómo los componentes del sistema de reconocimiento de lugar por medio de la visión artificial heredan del modelo neuronal para definir computacionalmente la información que procesan. Por último, se pasará a explicar la relevancia del problema tratado y qué aplicaciones puede tener.

Más adelante explicamos el modelo tradicional de un sistema de visual place recognition, los módulos que lo componen, así como las ventajas e inconvenientes de usar este tipo de sistemas, y los problemas que se tratan de resolver a la hora de aplicar su uso. Se explicarán en detalle los tres módulos que componen el sistema y las categorías que hay dentro de estos, así como algunos ejemplos del módulo de procesamiento de imagen que se va a tratar en el presente documento.

2.2 Definición del problema

El reconocimiento de un lugar visualmente es un problema bien definido y muy tratado en el ámbito de la visión por computador, pero muy complejo de resolver. Consiste en dada una imagen de un lugar ser capaz de distinguir qué lugar es. Para poder llevar a cabo dicha distinción se necesita un conocimiento previo de dicho lugar, una definición de él con la que comparar la imagen que se visualiza. Por tanto, en primera instancia se debe definir el concepto de lugar. Según la Real Academia de lengua Española se define como: "Espacio ocupado o que puede ser ocupado por un cuerpo cualquiera.", por tanto, se tiene que un lugar es una región del espacio. A la hora de definir un lugar se está definiendo una región del espacio, asignándole unas características distintivas y es esto lo que hace interesante la resolución del problema de su reconocimiento, pues conociendo el lugar y su definición se conoce una región del espacio a partir de la cual es posible obtener la región del espacio ocupada por el observador.

En la naturaleza, los animales y los humanos son capaces de generar mapas en su memoria gracias a la distinción de lugares que son capaces de reconocer y gracias a ello pueden reconocer su posición, de hecho, el conocimiento que tienen del espacio se almacena como una serie de lugares conocidos y la relación espacial entre ellos. El estudio del proceso de mapeado y navegación, llevado a cabo en el cerebro de forma natural, tiene una larga tradición en los campos de psicología y neurociencia. En 1948, la investigación de Tolman [2] con ratas navegando por laberintos le llevaron a proponer el concepto de un mapa cognitivo, una representación mental del mundo con información de las relaciones entre los lugares que los animales aprenden. Este mapa cognitivo sirvió como base conceptual al concepto moderno del mapa interno en un sistema de visión por computador, donde el mapa almacena información disponible de los puntos en el espacio, de manera similar a como los animales almacenan diversos lugares de interés y las relaciones espaciales entre ellos.

Investigaciones posteriores aplicando técnicas para obtener la actividad neuronal del cerebro de animales [3] llevó a la identificación de las células encargadas de reconocer un lugar previamente visitado, denominadas células de lugar, del inglés *place cells*, halladas en el hipocampo de las ratas por O'Keefe y Dostrovsky [4]. El reconocimiento de un lugar, observado por la activación de las células de lugar, es causado por la percepción visual y la noción de moción propia [5], es decir la idea aproximada de cuánto se ha movido y en qué dirección el cuerpo. Estudios posteriores comprobaron que, incluso si el ambiente cambia, alterando por ejemplo las distancias entre el punto de partida y el destino final, las células de lugar se actualizarán con la correcta localización corrigiendo el error en la estimación de movimiento.

Los conceptos extraídos de los estudios anteriormente mencionados sirvieron de base en el desarrollo de los sistemas de reconocimiento visual por computador, como en el trabajo de C.Siagian y L. Itti de un sistema de localización inspirado en el modelo del mapa biológico [6]. Dichos sistemas incluyen componentes para la extracción de información mediante observación, sensores para capturar datos de moción y un mapa interno almacenado, y esta información se emplea en la tarea de reconocer un lugar previamente conocido. El lugar

por reconocer no necesariamente ha debido ser visitado por el sistema dado que el sistema de computación permite cargar información previamente obtenida. Esto nos permite distinguir entre dos procesos, uno en el cual el sistema tiene una preconcepción del lugar y otro en el que se parte de no tener ninguna información previa. Mientras que este conocimiento previo ayuda a mejorar la precisión del proceso, entendiendo por precisión el número de lugares reconocidos correctamente partido por el total de lugares reconocidos tanto de forma acertada como falsos positivos, también provoca que el sistema se comporte peor frente a variaciones en el aspecto de dichos lugares.

La importancia del estudio del reconocimiento visual por computador viene de la mano de la creciente necesidad de mayor autonomía en las plataformas móviles, como robots o vehículos autónomos, y la necesidad, por seguridad, de conocer correctamente su entorno. La información visual es sencilla y barata de obtener y procesar, siendo ampliamente empleada en algoritmos de visión artificial. Algunos de éstos incluyen detección de lugares, objetos, formas y personas, en aplicaciones de realidad aumentada y mixta, y en el desarrollo de sistemas de navegación autónoma, entre otros.

2.3 *Visual Place Recognition*

Visual place recognition, en adelante VPR, es un problema definido en el ámbito de la visión artificial. Consiste en, dada una imagen de un lugar, tomar la decisión de si ha sido previamente visitado o no. Existen una serie de requisitos que todo sistema de reconocimiento de lugar debe tener y debe hacer. Dado que el objetivo es reconocer un lugar, éstos se han de definir obteniendo, a partir de la información que se tiene de ellos, un modelo del lugar con el cual comparar. Un sistema de VPR debe mantener un mapa interno donde se posiciona el modelo que define el lugar, pues el objetivo final del sistema es determinar la posición propia en dicho mapa, comparando la información visual obtenida con los modelos almacenados en el mapa. Por último, se debe generar una estimación de si la información visual obtenida corresponde a un lugar existente en el mapa, y determinar qué lugar.

Al determinar el lugar y la relación entre la ubicación propia y éste, el sistema de VPR permite obtener la posición propia dentro del mapa a partir de un proceso computacionalmente poco costoso y unos sensores económicos. Sin embargo, resolver el problema de VPR es una tarea difícil debido a la naturaleza inconsistente del proceso de captura de imágenes, y la variabilidad de aspecto en un lugar. Un buen sistema de reconocimiento de lugar debe ser capaz de distinguir entre imágenes con lugares aparentemente idénticos, conocido como fenómeno de *aliasing* perceptual, y ser inmune a las variaciones de apariencia debidas a tomar la captura de imagen desde un ángulo o posición diferente, con diferentes condiciones de iluminación y a cambios o movimientos de los objetos contenidos en el espacio de dicho lugar, Figura 2.1. Esto último se conoce como condiciones cambiantes, y suponen el mayor reto al que enfrentarse al desarrollar un sistema de percepción basado en visión artificial.

Figura 2.1 – Condiciones cambiantes, izquierda. Fenómeno de aliasing perceptual, derecha.

Los sistemas de VPR se componen de tres módulos, cada uno de ellos encargado de resolver un problema independiente. De manera similar a la naturaleza la principal fuente de información para resolver el problema es la visión artificial y la odometría, es decir la percepción visual y la noción de moción propia mencionadas en los trabajos de O'Keefe y Dostrovsky. Esta información se contrasta con el mapa interno que contiene los modelos de los lugares conocidos y la relación entre ellos, y si se encuentra una o más coincidencias se decide cuál es la más probable de ser correcta. Los tres módulos del sistema de VPR son el módulo de captura y procesado de imagen, el mapa interno y el generador de confianza y se relacionan entre sí como se indica en la Figura 2.2.

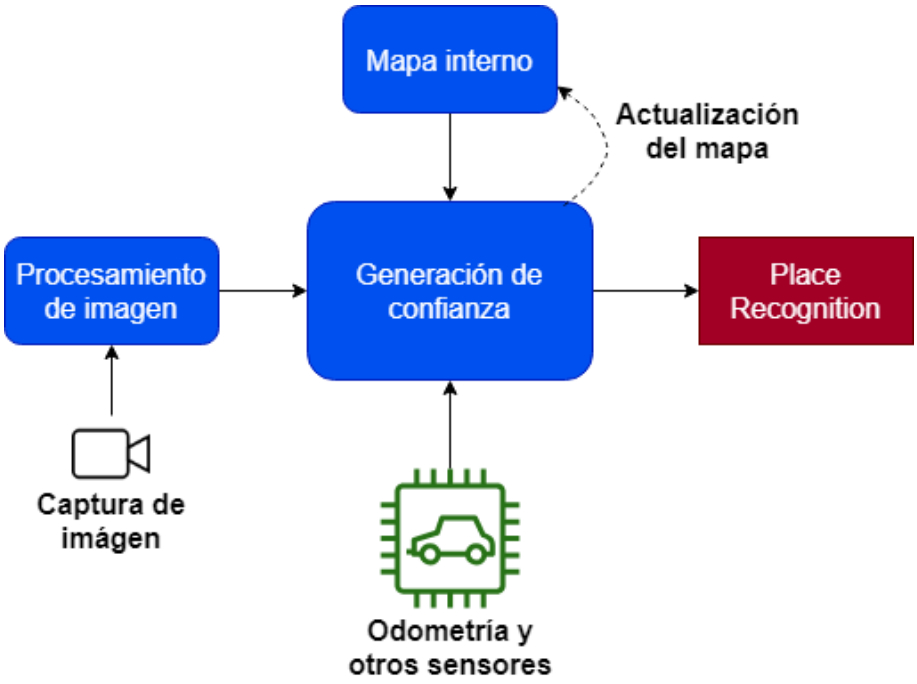

Figura 2.2 - Esquema de un sistema de VPR

Como se ha mencionado anteriormente, el primer problema a resolver utilizando un sistema VPR es la definición del lugar que queremos reconocer. Cuando se define el lugar se crea un modelo con información obtenida a través de los sensores empleados, siendo el principal la visión artificial mediante el uso de cámaras y procesamiento de imágenes. Por tanto, se debe crear un modelo a partir de la información visual del lugar observado. Para ello se obtiene de la imagen una serie de puntos o zonas de interés que describen la imagen y se forma un modelo del lugar visualizado. Este primer paso se conoce como el procesamiento de imagen.

Los modelos obtenidos a partir de las imágenes, combinados con otros sensores, se procesan para generar una representación interna del conocimiento que se tiene del espacio, conocida como el mapa interno del sistema. El mapa puede ser creado y cargado previamente, dotando al sistema de un conocimiento previo del ambiente por el que va a trabajar. Según el tipo de información almacenada en el mapa una clasificación de los tipos de mapas de VPR la encontramos en la Tabla 2.1, donde la principal distinción se da entre modelos con y sin información métrica del lugar.

Nivel de abstracción del modelo	Tipo de modelado del lugar	Comentario
Puramente recuperación de imagen	Modelado por apariencia	Sin información sobre posición.
Topológico	Modelado por apariencia	Incluye información de la moción del observador.
Topológico-métrico	Modelado por apariencia	Incluye información métrica entre los lugares, pero no de los propios lugares.
Topológico-métrico	Información métrica esparcida	Sistema de SLAM. Incluye información métrica entre los lugares y de los propios lugares.
Topológico-métrico	Información métrica densa	Sistema de SLAM. Incluye información métrica entre los lugares y de los propios lugares.

Tabla 2.1 – Clasificación de mapas internos en sistemas de VPR

Finalmente, el último módulo tiene como fin determinar si un lugar ha sido previamente visualizado. El módulo se conoce como el módulo de generación de confianza, donde se compara la información visual entrante, ya tratada en el módulo de procesamiento, con la representación interna y, opcionalmente, la información captada por otros sensores, para generar una estimación con un cierto grado de confianza, sobre si el lugar visualizado coincide con un lugar previamente representado y cuál. Por lo general se entiende que, si dos modelos que representan a un lugar son aparentemente idénticos, o guardan mucha similitud, se trata del mismo lugar. Pero la veracidad de esta afirmación depende del ambiente en particular. Por ejemplo, si se trata de un ambiente repetitivo cabe la posibilidad de caer en el fenómeno de *aliasing* mencionado anteriormente.

2.4 Módulo de procesamiento de imagen

Como se ha mencionado anteriormente, un aspecto fundamental de VPR es el modelo descriptivo de la imagen. Las técnicas de descripción de imagen se dividen en dos categorías: por un lado, las que extraen selectivamente fragmentos concretos de la imagen que son de alguna forma relevantes o notorios, y por otro lado las que describen la escena de forma global. Por tanto, las categorías en las que se divide el módulo de procesamiento de imagen son los que emplean descriptores locales y los que emplean descriptores globales. Los descriptores locales requieren de una fase previa de detección de regiones con características de interés en la imagen. Los descriptores globales no requieren de ninguna fase previa de detección y describen la imagen entera sin importar su contenido. En el presente documento se trata el módulo de procesamiento de imagen con descriptores locales, empleado distintos tipos de detectores de características.

Los descriptores locales permiten combinar los puntos característicos con información geométrica, y son más robustos a cambios de orientación. Además, permiten definir lugares nuevos no definidos previamente y combinarlos para crear un mapa interno, incluso sin tener ningún conocimiento previo del entorno. La principal desventaja encontrada empleando descriptores locales es que presentan un rendimiento más pobre frente a los globales cuando existe una variación de las condiciones de iluminación.

En cambio, los descriptores globales no se pueden combinar con información métrica, siendo más susceptibles de cambiar con la perspectiva que los locales, por lo que presentan problemas con cambios de orientación o posición del observador. La principal ventaja de su uso es que, mientras que los descriptores locales presentan un rendimiento pobre en condiciones ambientales y lumínicas variables, los globales son más robustos ante éstos.

2.4.1 Descriptores de características locales

Los descriptores locales dividen la imagen, seleccionando regiones o puntos que se consideran de interés para definir el modelo de ésta. Tienen una primera etapa donde se detectan dichos puntos de interés que después se emplean para generar un modelo que describe la imagen. No existe una definición global de lo que se considera un punto característico en una imagen y su definición varía según el problema o el tipo de aplicación. En VPR se considera que un punto característico es un punto o región de interés para la definición del modelo de la imagen.

Se emplean diversos algoritmos para identificar estas áreas de interés y describir la imagen. Los descriptores tempranos se centraban en zonas fácilmente reconocibles en la escena como esquinas, bordes y manchas, por ejemplo, el detector de bordes de Canny [7] de la Figura 2.3, o el detector de esquinas de Harris *et al.* [8] de la Figura 2.4. Estos algoritmos presentan un buen rendimiento frente a rotaciones en la imagen, sin embargo, son poco robustos frente a cambios de escala y perspectiva.

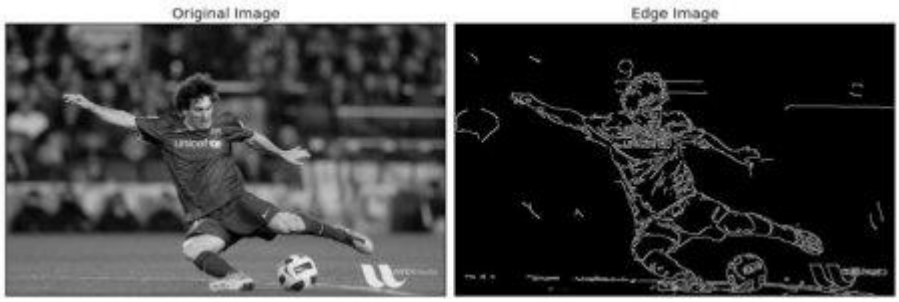

Figura 2.3 – Detector de bordes de Canny.

Algunos ejemplos de algoritmos más populares son el SIFT [9] (Scale-Invariant Feature Transforms) propuesto por David Lowe en 1999, que corrige los defectos mencionados previamente siendo robusto frente a cambios de escala, orientación y posición, y SURF [10] (Speeded-Up Robust Features) propuesto en 2006, heredando varios conceptos de SIFT y que demuestra un tiempo de procesamiento menor comparado a éste.

Figura 2.4 – Detector de esquinas de Harris.

Las imágenes pueden contener cientos de puntos característicos, por lo que una comparación directa de éstos puede ser computacionalmente costosa e ineficiente. Existen modelos más eficientes con gran número de puntos, siendo uno de los más empleados el modelo de bolsa de palabras, del ingles *bag-of-words*, que agrupa estos puntos en palabras formando un vocabulario finito, y en lugar de comparar los puntos comparan las palabras con técnicas de procesado de texto. Todas las palabras posibles forman un vocabulario, donde cada punto característico de una imagen pertenece a una palabra, sin considerar restricciones espaciales o

geométricas. Esto permite definir las imágenes con histogramas binarios de longitud n, siendo n el número de palabras en un vocabulario.

La definición de un modelo empleando puntos característicos locales consta de dos partes; la detección y la descripción del modelo. En algunas apliaciones se han aplicado distintos algoritmos para cada proceso; por ejemplo, Mei *et al.* [11] usaba como detector el algoritmo FAST y los puntos característicos encontrados eran descritos con SIFT.

Los descriptores de caracteristicas SIFT y SURF son muy empleados en problemas de localización visual, reconocimiento visual de lugares u objetos, modelado 3D, tracking, entre otros. La ventaja de estos algortimos es que presenta un buen rendimiento ante cambios de perspectiva, ya sea orientación, escala o una combinación de ambos. En el presente proyecto haremos uso de ambos.

2.4.1.1 Scale-Invariant Feature Transforms (SIFT)

En la Figura 2.5 se ve una escena donde se ha aplicado el agoritmo SIFT para la detección de puntos de interés. El algoritmo SIFT fue propuesto por primera vez por David Lowe en una publicación en 1999, donde lo describió como un algortimo para detección de puntos característicos, o *features* en inglés, invariable frente a cambios en la perspectiva, es decir en orientación o escala.

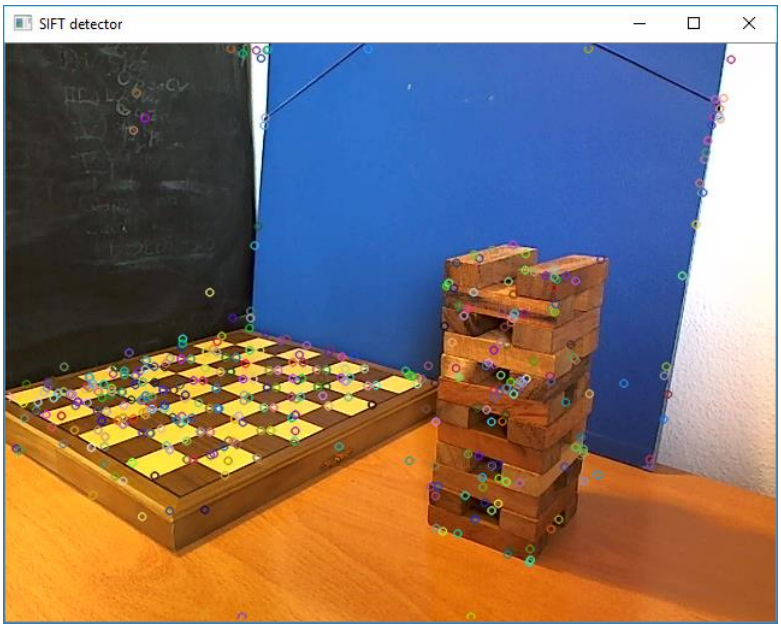

Figura 2.5 – Puntos característicos detectados en una escena aplicando SIFT

Muchos detectores se basan en la detección de esquinas, como el detector de Harris mencionado anteriormente. Sin embargo la detección de esquina presenta un problema frente al escalado pues una esquina puede dejar de serlo o ser no poder ser definida por un solo punto al cambiar la escala en la imagen. Este fenómeno está graficamente ilustrado en la Figura 2.6.

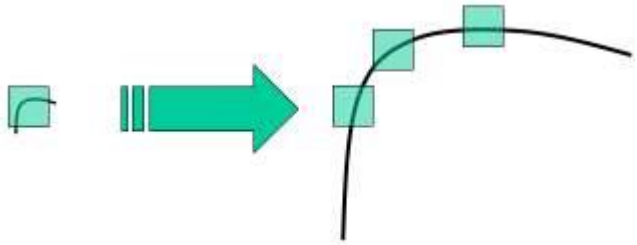

Figura 2.6 – Problema frente a cambio de escala observado en los detectores de esquinas.

El algoritmo de detección SIFT se realiza en 5 pasos. En primer lugar, está la detección de extremos en la escala-espacio, que consiste en aplicar la diferencia gaussiana, que actúa a modo de detector de esquinas y manchas, y buscar máximos y mínimos tanto en la escala, determinado por un factor de escalado t, como en el espacio x, y. Por ejemplo, un píxel es comparado con sus 8 vecinos, así como los 9 píxeles en la proxima escala y los 9 de la anterior escala. En la Figura 2.7 vemos un esquema de este concepto.

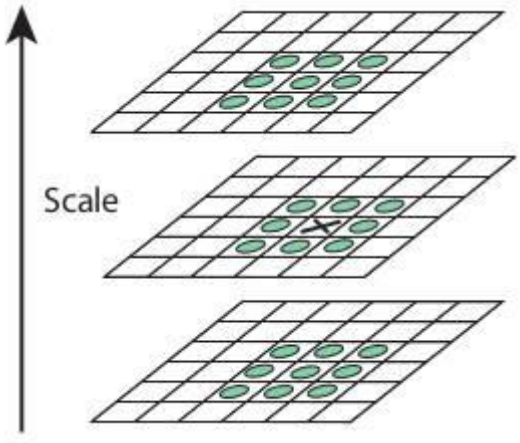

Figura 2.7 – Fase de detección de extremos en la escala-espacio del algoritmo SIFT.

El segundo paso en el algoritmo SIFT es la localización de los puntos de interés. Los extremos anteriormente hallados son candidatos, pero deben pasar este filtro. Usando una expansión de la serie de Taylor de la escala-espacio se comprueba si el valor en el punto es menor que un cierto umbral, y en caso de serlo se descarta. Aplicando la diferencia gaussiana tambien se incluyen bordes que deben ser eliminados. Para ello, se aplica un concepto similar al detector de esquinas de Harris. Aplicando una matriz Hessiana de dimensión 2x2 se computan las curvaturas principales, y se descartan los puntos para los cuales los autovalores de la matriz hessiana difieren en un orden de magnitud o más, dado que probablemente corresponden a bordes en la imagen y no esquinas.

A continuación sigue el asignado de orientación a cada punto característico para garantizar la propiedad de invarianza frente a cambios de perspectiva. Con la información ya obtenida de los puntos vecinos en torno a los puntos aún no descartados se calcula el máximo gradiente y se forma un histograma de orientación. El máximo valor marca la orientación, pero si existen direcciones donde el valor es mayor del 80% del máximo se usan para crear otros puntos de interés con dicha orientación.

El cuarto paso consiste en crear los descriptores de los puntos de interés. Se toma un vecindario de 16x16 que a su vez se divide en sub-bloques de 4x4. Para cada uno se define el histogrma de orientación, y se concatenan en un vector para formar un descriptor del punto.

Por último, se realiza el matching de los puntos entre dos imágenes para compararlas. La técnica empleada es identificar los puntos vecinos más cercanos en el espacio de los descriptores, vectores de dimensión 128. Sin embargo el segundo vecino puede estar muy cerca del primero por culpa de ruido. En este caso, se toma la razón entre la distancia al punto más cercano y al segundo más cercano, y si se encuentra por encima de un cierto umbral los puntos se descartan.

2.4.1.2 Speeded-Up Rebust Features (SURF)

En la Figura 2.8 se observa la misma escena anterior, pero esta vez el algorítimo empleado para la detección de puntos característicos es SURF. El algoritmo SURF se publicó por Herbert Bay and al. en 2006. Su funcionamiento está basado en su predecesor, SIFT, donde se trató de mejorar la velocidad de detección de los puntos característicos.

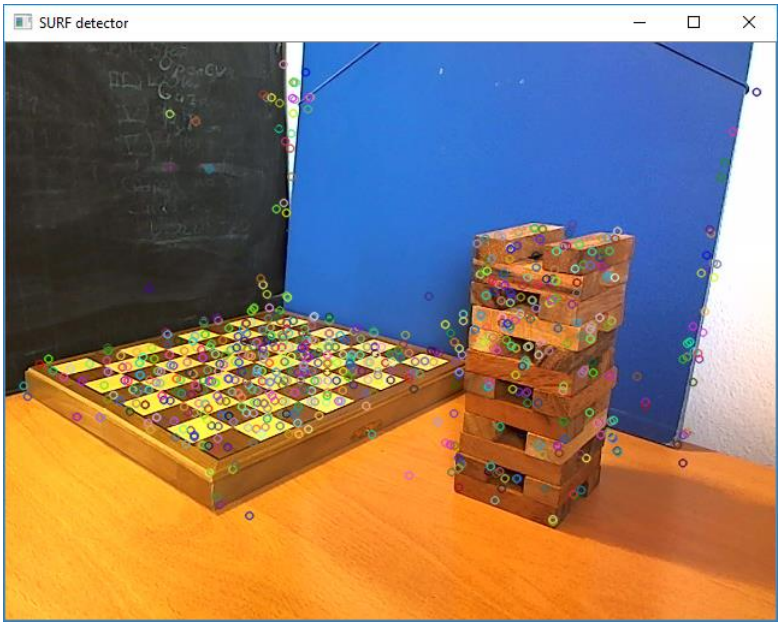

Figura 2.8 – Puntos característicos detectados en una escena aplicando SURF

El algoritmo SURF se realiza en tres pasos: detección, asignación de una orientación y extracción de los descriptores, de forma similar al algoritmo SIFT explicado anteriormente. En cada uno de los pasos se emplean técnicas para agilizar el proceso de manera que se obtiene un tiempo de cómputo menor que aplicando el algoritmo SIFT. En la Tabla 2.2 vemos una comparativa del tiempo de respuesta utilizando ambos algortimos. Los datos de dicha tabla son extraídos de una análisis hecho por A.M. Rimero **y** M. Cazorla [12] (Comparativa de detectores de caracteristicas visuales y su aplicación a SLAM). SLAM, *Simultaneous Location And Mapping*, es una algortimo empleado para la navegación atónoma que permite construir un mapa de un entorno desconocido y emplearlo para estimar la trayectoria seguida dentro de este.

109 imágenes	SIFT	SURF
Puntos detectados	1292	42
Media de tiempo	1646.53 ms	485.77 ms

Tabla 2.2 – Comparativa de resultados empleando algoritmos SIFT y SURF

Como se describió anteriormente en SIFT, se emplea la diferencia gaussiana para aproximar el laplaciano del gaussiano. En SURF sin embargo éste se aproxima con filtros cuadrados, dado que el filtrado de la imagen con cuadrados es mucho más rápido si se emplea una imagen integral, que consiste en aproximar la imagen por el promediado del valor de los píxeles dentro del cuadrado incluído en el filtro. En la Figura 2.9 se observa un esquema del filtro cuadrado con imagen integral.

Se emplea un detector de manchas basado en la matriz hessiana para encontrar los puntos de interés, donde se toman como puntos de interés aquellos para el cual el determinante presenta un máximo local, y también empleando la matriz hessiana para iterar por la escala.

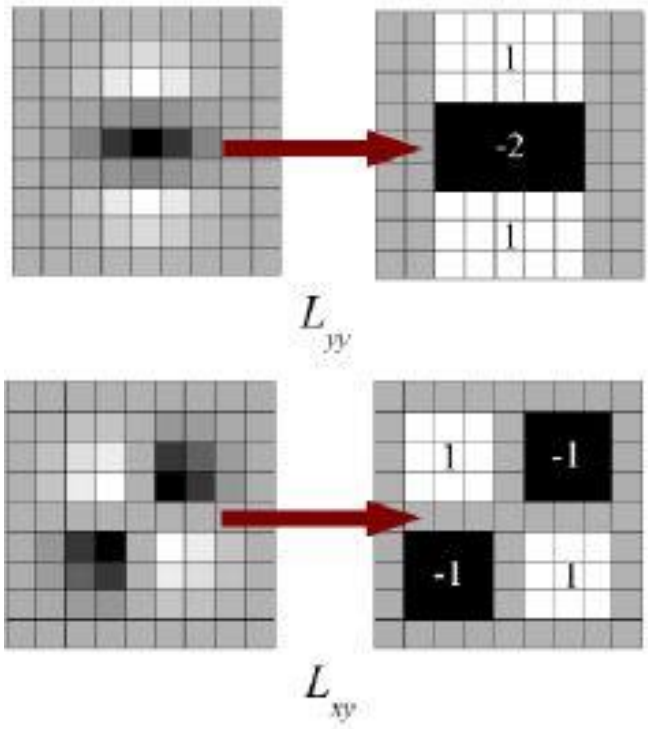

Figura 2.9 - Esquema de la aproximación con filtros de caja e imágenes integrales.

Para recorrer la escala-espacio de forma similar a como se hizo en SIFT, en SURF se aplican filtros cuadrados de diversos tamaños, siendo el menor de los empleados 9x9. El resultado del filtro 9x9 se considera como una aproximación de un gaussiano con σ igual a 1.2. Las siguientes capas se obtienen aplicando filtros de dimensión gradualmente mayor, resultando en una sucesión de filtros de 9x9, 15x15, 21x21, 27x27…

2.4.2 Comparación con descriptores globales

En contraste con los descriptores locales que procesan la imagen tratando de encontrar puntos característicos para definir un vector representativo de esta, los descriptores globales definen la escena completa, sin seleccionar o extraer partes de ésta. Algunas de las aproximaciones iniciales a la resolución de este problema consistían en emplear histogramas de color con descriptores basados en análisis principal, para describir la imagen. Los descriptores globales incluyen características como representación de contornos, descripción de figuras o texturas, y descripción de colores entre otros. Uno de los descriptores globales más popular es GIST, presentado por Oliva y Torralva en 2001 [13].

2.5 Módulo de mapeado

La representación interna del conocimiento que el sistema posee del entorno que lo rodea se define como mapa interno. En este mapa se guarda toda la información que se posee, distinguiéndose entre los diversos tipos según a qué clase de información tiene acceso. Los mapas internos en VPR poseen como mínimo información proveniente de la captura de visión artificial, ya tratada por el módulo de procesamiento de imagen. La principal diferencia es si, adicionalmente, se conoce información espacial o no, y qué nivel de información espacial se conoce, pudiendo conocerse datos de moción entre las capturas de las distintas imágenes, formando mapas topológicos, o conociendo distancias entre las imágenes del mapa o dentro de las propias imágenes. Para lograr esta información adicional se hace uso de sensores adicionales, como GPS, IMU y lidar, entre otros.

Los mapas que solo poseen información visual se conocen como mapas de recuperación de imagen pura, del inglés *pure image retrieval maps*. En este tipo de mapas se comparan los lugares basándose únicamente en la similitud en la apariencia. Computacionalmente son muy eficientes al ser los mapas que menos información almacenan y, por tanto, menos comparan.

Los mapas topológicos puros contienen información de las distancias entre los lugares, pero no poseen información métrica de los propios lugares. Esta información se puede emplear para aumentar el número de correspondencias encontradas, y a la vez disminuir los falsos positivos, que son casos en los cuales el sistema de reconocimiento considera erróneamente que un lugar del mapa se está visualizando. Computacionalmente resultan más costosos que los mapas anteriores, sin embargo, al aumentar la información del mapa se puede filtrar la comparación por distancia, no comparando la visualización actual con lugares que se encuentren a mucha distancia. En cambio, en los mapas de recuperación de imagen, al aumentar el número de imágenes del mapa, aumenta de manera proporcional el tiempo de cómputo al necesitar comparar con más imágenes.

Por su propia definición, en el mapa se almacena toda la información que se posee del entorno resultando una cantidad muy extensa, sobre todo cuando el área de funcionamiento va aumentando y el mapa aumenta de tamaño. Este problema se conoce como escalabilidad del sistema; a medida que se conocen más lugares, la necesidad de almacenamiento aumenta, y la eficiencia en la extracción de información del mapa cae. Como se ha mencionado previamente, en mapas topológicos se puede filtrar la información por distancias para mejorar la eficiencia del cómputo. Sin embargo, al escalar el sistema el propio tiempo de acceso a la memoria aumenta también. Una solución a esto puede ser organizar la información del mapa en el modelo de bolsa de palabras mencionado previamente, discretizando la información visual almacenada. Discretizar la información del mapa permite emplear técnicas de búsqueda jerárquicas, resultando más eficiente que la comparación directa.

Los mapas también se ven afectados cuando se dan cambios en las condiciones de trabajo del sistema, como variaciones climáticas o lumínicas. El sistema debe decidir qué información del mapa debe mantenerse y cuál se debe eliminar, dado que al cambiar las condiciones de trabajo esta información queda obsoleta y penaliza el funcionamiento del sistema. Por otra parte, si el tiempo previsto de funcionamiento del sistema es muy largo es posible que las condiciones del sistema cambien de forma cíclica, por ejemplo, la iluminación entre el día y la noche. Para estos casos interesa mantener la información, pudiendo formarse varios niveles de mapa, que representan las distintas condiciones de funcionamiento y van rotando su uso.

El primer caso se conoce como mapa dinámico, donde la información del mapa debe actualizarse periódicamente. En estos casos debe encontrarse un equilibrio entre la información recordada y la olvidada, que permita mantener una representación actualizada del entorno, pero que no permita que cambios pasajeros se almacenen en el mapa, por ejemplo, el movimiento de una persona por delante de la cámara. Una solución adaptada del concepto del mapa biológico consiste en dividir la información del mapa entre memoria a largo plazo y memoria a corto plazo. La información sensorial se almacena en la memoria a corto plazo, y solo se pasan a la memoria a largo plazo elementos con suficiente relevancia o repetitividad. Adicionalmente, los elementos obsoletos se van filtrando progresivamente de la memoria a largo plazo. Este tipo de mapas funcionan bien en casos donde se dan suficientes visualizaciones del mismo elemento.

El segundo caso se conoce como mapas de múltiples representaciones. En estos mapas no solo se actualiza la información con el tiempo, sino que cíclicamente esta información varía de forma que no se puede representar de forma eficiente en un mapa único. Ranganathan *et al.* [14] propuso que en un ciclo de 24 horas se necesitan entre tres y cuatro imágenes para cada localización. Las representaciones múltiples pueden darse para el mapa completo o en cada lugar, y se pueden combinar con el mapa dinámico.

2.6 Módulo de generación de confianza

El objetivo de un sistema de VPR es tomar una decisión de si un lugar visualizado ha sido previamente visitado. Para ello se hace uso de la descripción de la imagen, generado un vector con características representativas de esta, y de un mapa interno, almacenando la información obtenida de las visualizaciones previas junto a la información proporcionada por otros sensores. La imagen capturada se compara con la información del mapa, donde para cada pareja de lugares se analizan los vectores de puntos de características, emparejando los puntos de ambos vectores en un proceso conocido como matching. Tras realizar dicho matching se obtiene una matriz de dos columnas que contiene los puntos emparejados. Empleando esta información, junto con los valores conocidos de los sensores adicionales, se genera una estimación de confianza de que se está visualizando el mismo lugar. Esta generación de confianza es el objetivo último del sistema, proporcionando una medida de la certeza que se tiene de estar en el mismo lugar.

El último módulo es el encargado de generar esta información de confianza, comparando la información contenida en el mapa con el input de los sensores y la cámara, aplicando post procesados si se considera

necesario. Como se observa en la Figura 2.2 es el módulo central, que recoge toda la información conocida y presenta el resultado del proceso. Dicho resultado puede ser utilizado en procesos como SLAM con reconocimiento de lugar proponiendo candidatos para el cierre del bucle, o para generar un bordado, *stitching*, en imágenes panorámicas, entre otras aplicaciones. Un ejemplo de bordado se muestra en la Figura 2.10, donde se emplea el algoritmo SIFT para describir dos imágenes del mismo lugar y se comparan para generar unos *inliners*, puntos emparejados entre los dos vectores de características. Por último, se aplica homografía, el cálculo de la variación de la posición de la cámara entre ambas tomas, para obtener la superposición de una imagen sobre otra.

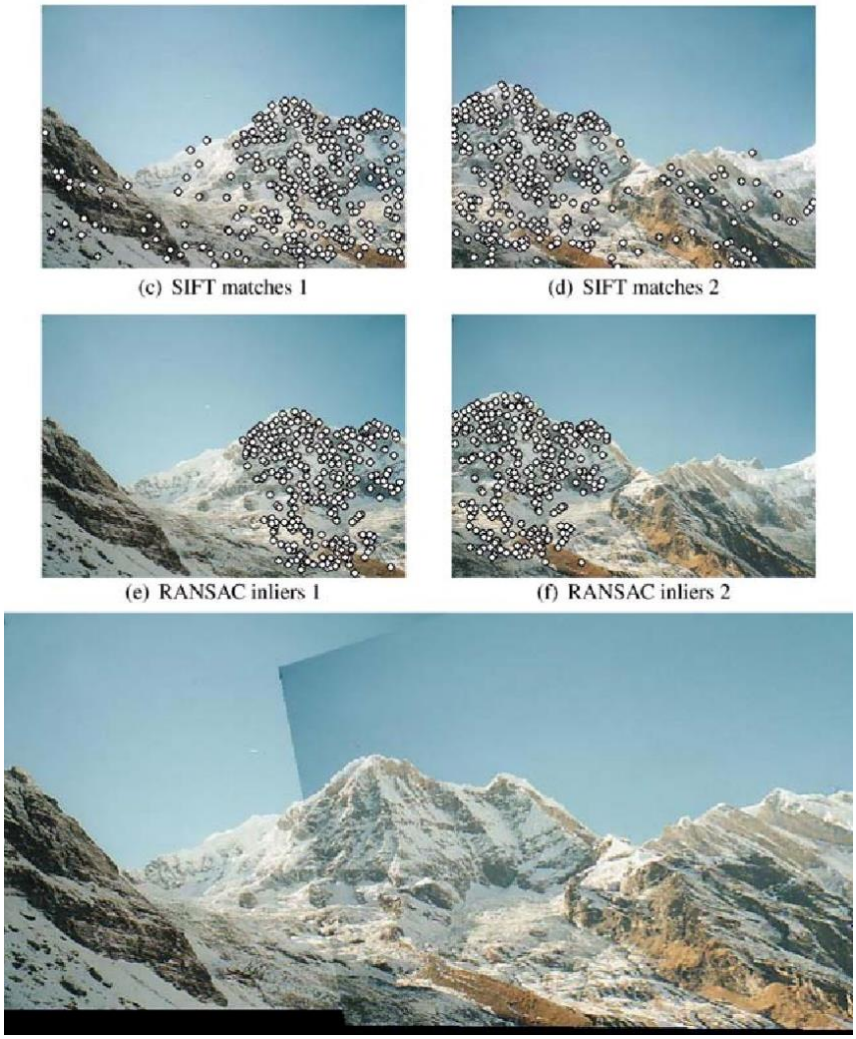

Figura 2.10 – Stitching de imágenes empleando puntos característicos SIFT.

En entonos repetitivos se pueden dar casos de aliasing visual, el determinar erróneamente que dos lugares distintos, pero visualmente idénticos, corresponden a la misma localización. Para evitar este problema la solución más robusta se basa en incluir información métrica en el proceso. Otro problema que se puede presentar en este módulo es la variación del ambiente entre la imagen capturada y la almacenada en el mapa, provocando que se descarte un *matching* erróneamente. Estos problemas son ejemplos de lo que se conoce como falsos positivos y falsos negativos, respectivamente. Una medida usual del rendimiento de un sistema VPR se expresa en la Ecuación 1.

$$Precisión = \frac{Positivos\ correctos}{Positivos\ correctos + Falsos\ positivos} \qquad Evocación = \frac{Posotivos\ correctos}{Positivos\ correctos + Falsos\ negativos}$$

Ecuación 1 - Precisión y evocación de un sistema VPR

3 ARQUITECTURA SOFTWARE DE LA APLICACIÓN

3.1 Introducción

Para la resolución de este proyecto, desde la extracción de imágenes para su procesado hasta la presentación de un resultado, se ha hecho uso de una arquitectura modular software que se explicará en el presente apartado. Los módulos corresponden a objetos de C++ encargados de proporcionar el servicio requerido, de forma independiente al modo de funcionamiento del sistema. El motivo de realizar esta estructura modular es que resulta sencillo adaptar los módulos para modos de funcionamiento distintos, o incorporarlos en aplicaciones independientes. La funcionalidad de cada uno de los módulos se implementa mediante una clase de C++ utilizando las librerías de OpenCV.

Uno de los modos de funcionamiento del sistema hace uso de un dataset como entrada de datos al sistema. Se explicará el proceso de selección de dicho dataset y la motivación de su uso.

3.2 Arquitectura global

La aplicación desarrollada se estructura según una arquitectura modular esquematizada en la Figura 3.1. El sistema se compone de una clase que permite crear objetos con las funcionales de ambos módulos para que trabajen de forma independiente, donde cada módulo es una instancia de dicha clase. La clase permite inicializar un objeto como descriptor con cualquier algoritmo implementado en la librería *opencv/features.h*, incluyendo tanto SIFT como SURF, e implementa métodos que reciben de entrada una imagen como un objeto Mat de OpenCV y devuelven un vector de puntos de interés, dándole la funcionalidad del módulo de procesamiento de imagen. Estos vectores, junto con las imágenes originales, son los que emplea el segundo objeto para realizar la comparación y tomar la decisión de si corresponde al mismo lugar.

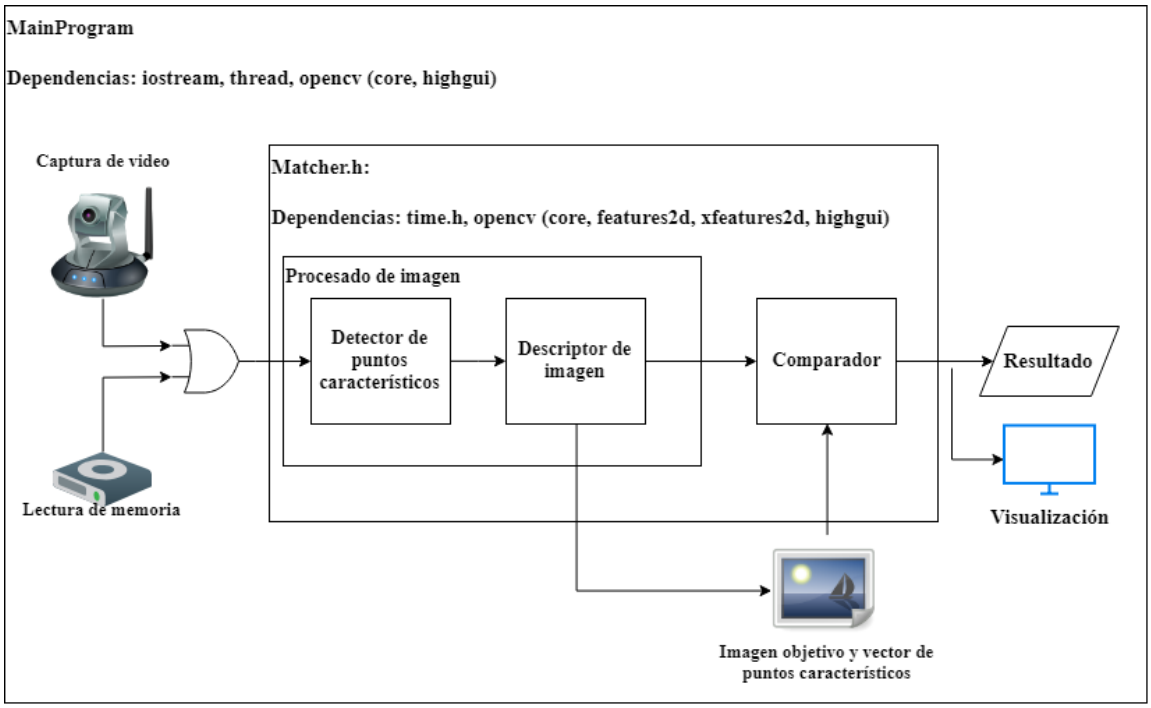

Figura 3.1 – Arquitectura global del sistema.

La función *main* de la aplicación se encarga de instanciar las clases, inicializar un hilo independiente que ejecuta la cámara o lee las imágenes de la ruta donde se almacena el dataset, pasada como argumento a *main*, y hace uso de los objetos para procesar dichas imágenes. También se encarga de mostrar las imágenes en una ventana tras los diferentes procesados y lee comandos de usuario que permiten abrir o cerrar la aplicación,

detener o continuar la captura de la cámara, seleccionar distintas imágenes del dataset y mostrar los resultados al usuario; tanto gráficamente como numéricamente. En la Figura 3.2 vemos la ventana principal de la aplicación, a la derecha para su uso con la cámara y a la izquierda para su uso con el dataset.

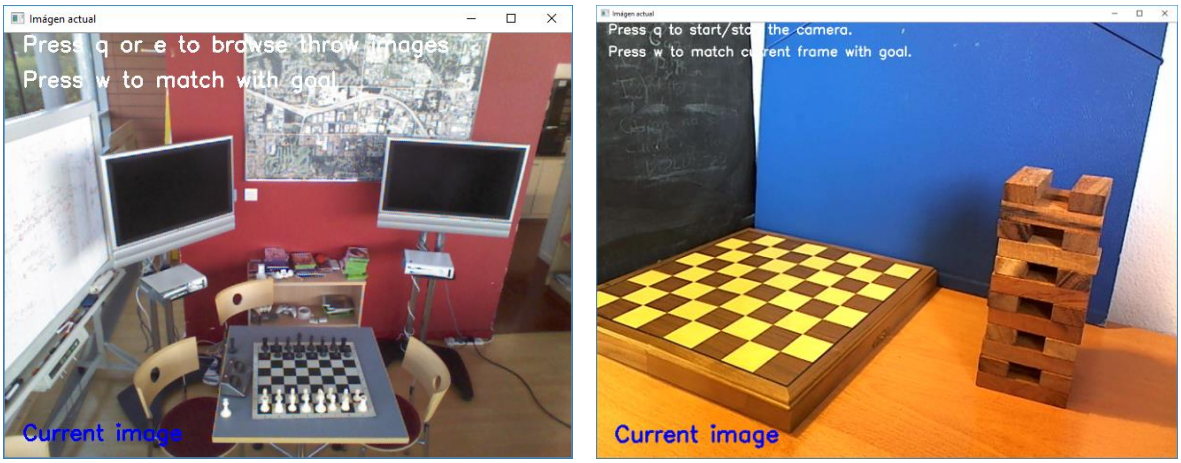

Figura 3.2 – Ventana principal de la aplicación.

3.3 Visión

Para la gestión de la visión artificial se hace uso de OpenCV, Figura 3.3, unas librerías *open source* desarrolladas originalmente por Intel [15] para visión artificial. Algunas aplicaciones incluyen el reconocimiento de objetos, la edición de imágenes y vídeos, calibración de cámaras, realidad aumentada y detección de puntos de interés de imágenes. Están desarrolladas en los lenguajes de programación C++, Java y Python, pero existen muchos *wrappers*, envoltorios, que permiten su uso en lenguajes no nativos, como EmguCV para C#. Emplearemos las librerías en su versión 3.4.3, con los módulos extras para emplear las funciones de detección de puntos de interés.

Estas librerías son ampliamente utilizadas y poseen una extensa comunidad y buena documentación, facilitando su extensión. Entre las funciones que posee se incluyen detección y descripción de puntos de interés en imágenes aplicando varios algoritmos para ello, entre ellos los algoritmos SIFT y SURF empleados en este proyecto. También permite configurar varios parámetros de dichos algoritmos, como el nivel del umbral hessiano o el radio aplicado en el filtro de Lowe.

Figura 3.3 – Logo OpenCV.

Otras utilidades de las librerías empleadas son la lectura de imágenes a partir de su ruta en el sistema, empleado para cargar las imágenes del dataset, o la inicialización de un objeto de captura de video que permite inicializar una cámara conectada al PC, calibrarla y obtener los fotogramas para su posterior procesamiento. También permite mostrar imágenes, editarlas, dibujar o escribir sobre ellas. Esto permite crear la interfaz de la aplicación combinándolo con la lectura por teclado de los comandos del usuario, para generar una aplicación interactiva.

3.4 Dataset

Para el desarrollo de este proyecto se ha empleado un dataset de imágenes para su procesamiento y análisis. Los requisitos fundamentales que debía cumplir dicho set de imágenes era que existieran imágenes que cubrieran la casuística descrita en el apartado 2 de visual place recognition: cambios en la posición, orientación, condiciones climáticas e iluminación. Con esto se pretende observar los problemas descritos en el procesamiento de imagen.

El dataset contiene 72 imágenes, de 5 escenas diferentes. Para su elaboración se han tomado imágenes del dataset RGB-D 7-Scenes de Microsoft [16], concretamente de las escenas *chess*, *pumpking* y *fire*, donde se observan los mismos lugares desde perspectivas distintas. Se han tomado 8 imágenes de cada escena con cambios progresivos, resultando en un total de 24 de las 72 imágenes, y correspondiente a imágenes de interiores. Las 2 escenas restantes se han obtenido con la cámara de un móvil, viendo las mismas 2 escenas de exteriores en diferentes condiciones, para observar los fenómenos de cambio de iluminación y climáticos y de estos junto a cambios de perspectiva. Se han capturado 8 imágenes de cada escena en diferentes perspectivas para un día soleado y un día lluvioso, tratando de tomar las mismas 8 perspectivas. También se han tomado otras 4 fotos de noche y al amanecer de ambas escenas, para un total de 48 imágenes. Podemos ver un extracto representativo de las escenas en la Figura 3.4. para el dataset de Microsoft, y en la Figura 3.5. para las imágenes capturadas.

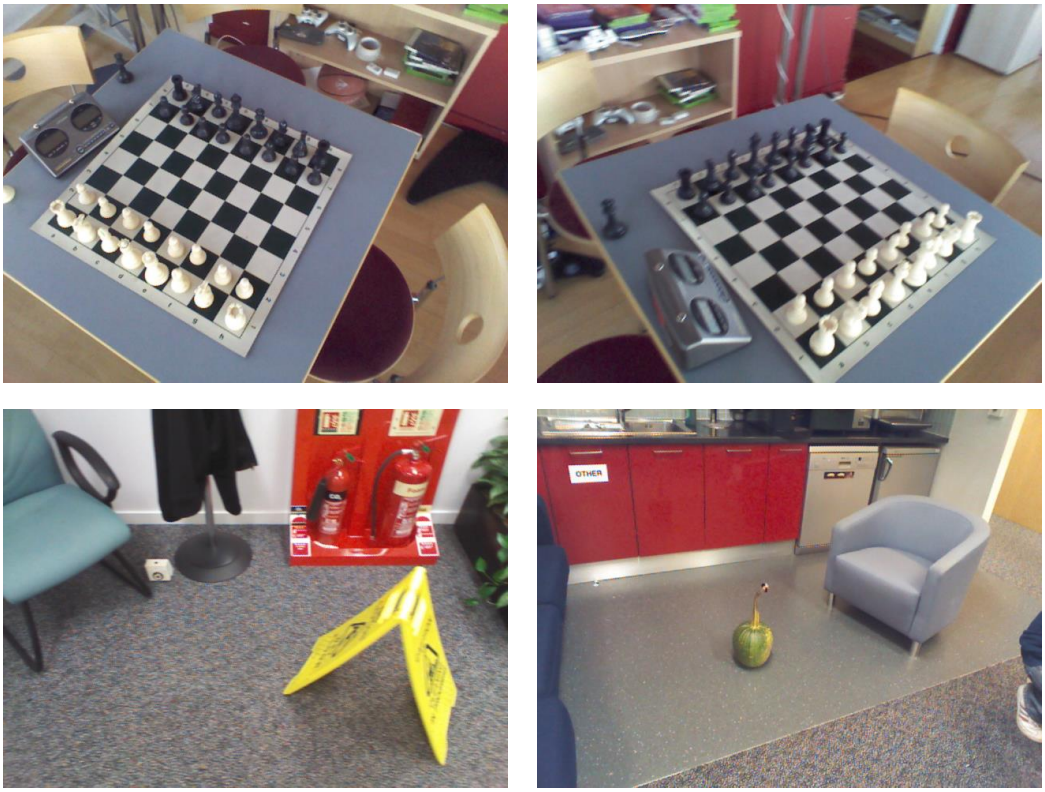

Figura 3.4 – Imágenes extraídas de RGB-D Dataset 7-Scenes de Microsoft.

Se ha optado por este dataset, pequeño en comparación con otros encontrados, para poder seleccionar una muestra aleatoria de este y compararla con todas las demás imágenes del conjunto. Dado que el tiempo de procesamiento en cada iteración era largo, debido a la limitada capacidad de computación del hardware empleado, se ha escogido un dataset poco extenso, pero con suficientes casos representativos de los problemas encontrados en un sistema de VPR más complejo.

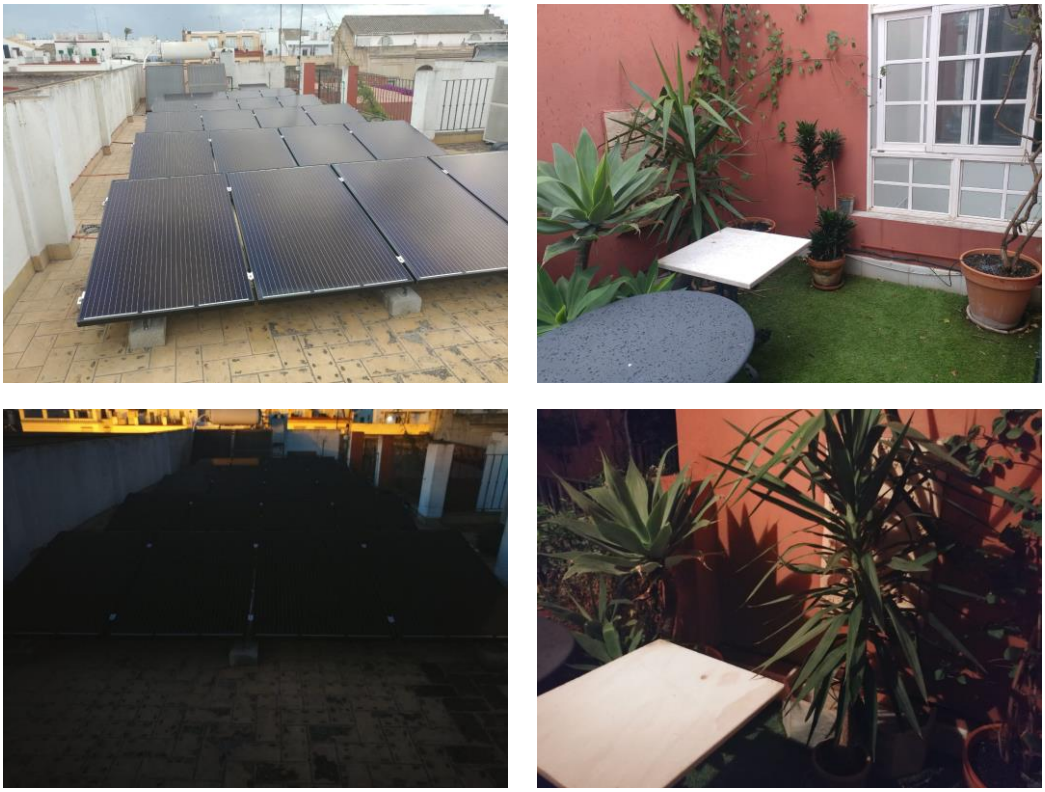

Figura 3.5 – Imágenes propias tomadas con diferentes condiciones climáticas y lumínicas.

4 PROCESO DE RESOLUCIÓN

4.1 Introducción

Una vez establecida la arquitectura general del proyecto, se procede ahora a explicar el proceso seguido para implementar dicha arquitectura. En el presente apartado se definirá el problema a resolver, y en particular se propondrán una serie de exigencias cuyo objetivo será evaluar el funcionamiento de la aplicación. Mediante dicha propuesta se espera justificar el empleo de la arquitectura en módulos, empleando para ello programación orientada a objeto.

Una vez delimitado los objetivos particulares que deben cumplirse se procederá a su resolución para ambos modos de funcionamiento del sistema, capturando imágenes de forma continua mediante una cámara o procesando imágenes preestablecidas ordenadas en un dataset.

4.2 Definición del problema

Para explicar el procedimiento seguido para la resolución del problema, en primer lugar, se definirá el problema que se espera resolver. Los objetivos de la aplicación se definen a continuación, según el proceso.

En cuanto a la adquisición de imágenes con una webcam conectada:

- Inicializar correctamente la cámara.
- Calibrar la cámara para compensar los efectos de distorsión propios de cámaras estenopeicas, que se dan en mayor o menor medida en todas las cámaras comerciales.
- Capturar la imagen de forma continua, independientemente del procesado de imagen que ocurra de fondo, de forma que el usuario pueda ver un video continuo.
- Pausar, continuar y detener la adquisición de imágenes con la cámara.

En la lectura de imágenes del dataset:

- Leer la ruta pasada como argumento correctamente.
- Detectar la existencia del directorio, y si contiene alguna imagen en formato válido.
- Almacenar la dirección de todas las imágenes dentro de la ruta.
- Abrir las imágenes, introduciendo su ruta y almacenarlas en un objeto Mat de OpenCV.

Para la descripción de imágenes:

- Detectar los puntos de interés empleando el algoritmo SIFT o SURF.
- Describir las imágenes según el algoritmo SIFT o SURF, independientemente del empleado para la detección.
- Funcionar de forma paralela a la adquisición de imagen, de forma que no penalice la latencia del video mostrado al usuario.

Por último, al comparar las imágenes:

- Realizar el emparejado, o *matching*, de los vectores de puntos de interés de ambas imágenes.
- Filtrar los emparejamientos erróneos.
- Tomar la decisión de si corresponden al mismo lugar.

Adicionalmente, para el caso en el que se emplea el dataset se comprobarán las tomas de decisiones del módulo de comparación, midiendo el rendimiento del sistema.

Los objetivos perseguidos se adaptan a la estructura modular presentada en la arquitectura del sistema, justificando el empleo de bloques independientes para la resolución del problema.

4.3 Imágenes de la webcam

Para la resolución del problema definido previamente se hace uso de una webcam comercial, una cámara Logitech modelo Carl Zeiss Tessar HD 1080p. Las cámaras comerciales requieren ser calibradas previamente, un proceso que consiste en tomar varias capturas de un objeto de dimensiones conocidas para obtener las distorsiones propias causadas por la cámara. Las causas de estas distorsiones son dos; la distorsión óptica, causada por la lente de la cámara, y la distorsión de perspectiva.

Para realizar la calibración se emplea una función de ejemplo de OpenCV con un tablero de ajedrez de dimensiones conocidas. Los resultados de la calibración se muestran en la Figura 4.1.

$$Camera\ Matrix: \begin{bmatrix} 5.335912589e+02 & 0 & 320 \\ 0 & 5.335912589\text{e}+02 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Distortion\ Matrix: \begin{pmatrix} 2.862152\text{e}-02 \\ 3.735802\text{e}-01 \\ 0 \\ 0 \\ -1.20631999 \end{pmatrix}$$

Figura 4.1 – Cámara Logitech Carl Zeiss Tessar 1080p.

Se inicializa la cámara con un objeto *video capture* de OpenCV, y los objetos de los módulos independientes. Dicho objeto se inicia y desde ese momento graba un vídeo continuo. El algoritmo muestra esta captura de vídeo leyendo el fotograma actual cada 30 milisegundos y mostrándoselo al usuario en una ventana. Se pide al usuario que pulse una tecla para realizar una captura, que será la imagen objetivo con la cual se compararán posteriormente las imágenes. El objetivo de la aplicación será comprobar si la imagen visualizada corresponde al mismo lugar que la imagen elegida. Dicha imagen pasa por el módulo de procesado de imagen, y se almacena junto con sus puntos de interés.

Tras esto se activa un hilo de ejecución concurrente encargado de mostrar en todo momento al usuario la captura de la cámara frente a la imagen objetivo, mientras que la aplicación entra en el bucle principal. La ventana mostrada se ve en la Figura 4.2. Este hilo se podrá pausar y reanudar con una señal enviada desde el bucle en el proceso principal. Al cerrar dicho bucle la captura se detiene y se libera la memoria reservada por el hilo antes de detener la ejecución. El hilo mantiene actualizada una variable compartida de imagen, el fotograma de la cámara, y es el único que lo puede modificar. El bucle principal podrá copiar en un momento dado la imagen, pero no podrá modificar su valor, para evitar problemas de sincronización.

El bucle principal de la aplicación espera en todo momento un comando de parte del usuario, en forma de una pulsación de tecla. Las opciones que ofrece son la pausa/reanudación del hilo de la cámara, el procesado del fotograma actual y la finalización de la aplicación. Si el usuario elige el procesado del fotograma actual se realiza una copia de la variable compartida por el hilo y se envía al módulo de procesado. El módulo de procesado aplica el algoritmo de detección y descripción escogido y devuelve la imagen y un vector de puntos de interés, que se envían al módulo de comparación y generación de confianza.

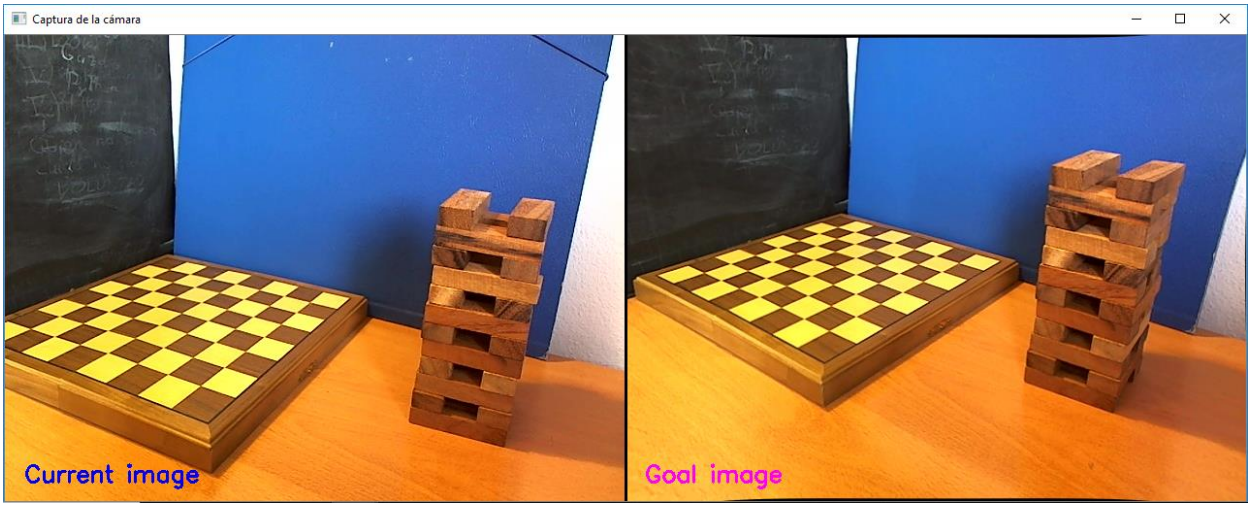

Figura 4.2 – Captura de imagen con webcam.

En el módulo de comparación se emparejan los vectores de puntos de interés que recibe de entrada y devuelve una matriz de 2xn, siendo n el número de emparejados realizado. En dicho vector se almacenan los puntos de interés de ambas imágenes que se han emparejado y se procede a filtrarlos, empleando para ello el filtro de radios de Lowe. Tras el filtrado la matriz se reduce, dejando únicamente los elementos que pasan el filtro de radios. Con esta información el módulo toma la decisión de si la imagen corresponde al mismo lugar o no.

4.4 Imágenes del dataset

Cuando se hace uso del dataset se pasa como argumento a la aplicación una ruta del sistema operativo hacia el directorio que contiene el conjunto de imágenes. En primer lugar, el código tratará de acceder a dicho directorio, avisando al usuario si no es capaz. Almacenará el nombre de todos los archivos en una lista de *strings*, que se filtra para quedarse únicamente con aquellos que sean imágenes, quedando finalmente una lista con los nombres de todas las imágenes, y la dirección de la carpeta donde están almacenadas.

Tras inicializar los objetos de los distintos módulos, se mostrará al usuario una primera ventana, Figura 4.3, donde se permite visualizar las imágenes y escoger entre ellas la imagen objetivo, y, al seleccionar la imagen, ésta se enviará al módulo de procesamiento para obtener el vector de puntos de interés. La imagen y su vector de descripción se almacenan dónde puede acceder a ellas el módulo de comparación.

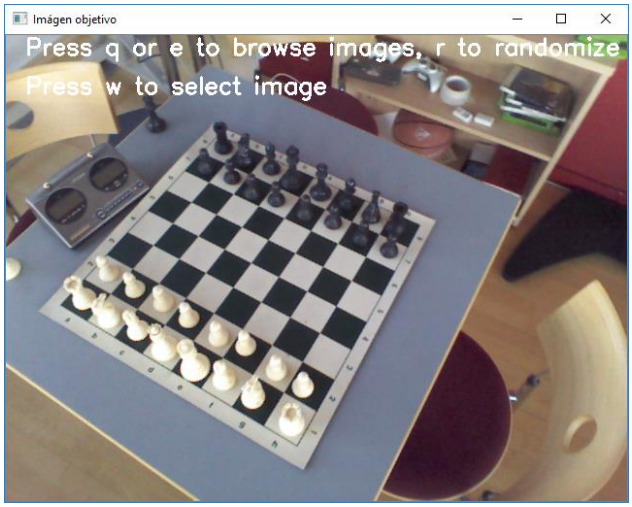

Figura 4.3 – Ventana inicial de selección de imagen.

Una vez seleccionada la imagen objetivo se inicia un bucle que itera por la lista de imágenes, abriéndolas con OpenCV, y mostrando una ventana interactiva al usuario, la ventana principal de la aplicación mostrada en la Figura 4.4 a la izquierda. Se leen los comandos del usuario como pulsaciones del teclado permitiendo mostrar la imagen siguiente o anterior, y enviar la imagen actual al módulo de procesado para comparar con el objetivo, a la derecha en la Figura 4.4.

El tratamiento de la imagen elegida es idéntico al caso de la webcam, se procesa en el módulo de procesamiento y se envía al comparador junto con el objetivo para tomar la decisión de si corresponden al mismo lugar. Como en el caso de dataset los nombres de las imágenes corresponden a las escenas seguidas de un número se puede comprobar con una simple comparación de nombres si el reconocimiento de lugar se ha realizado correctamente. Esto nos permite obtener una medida del rendimiento del sistema.

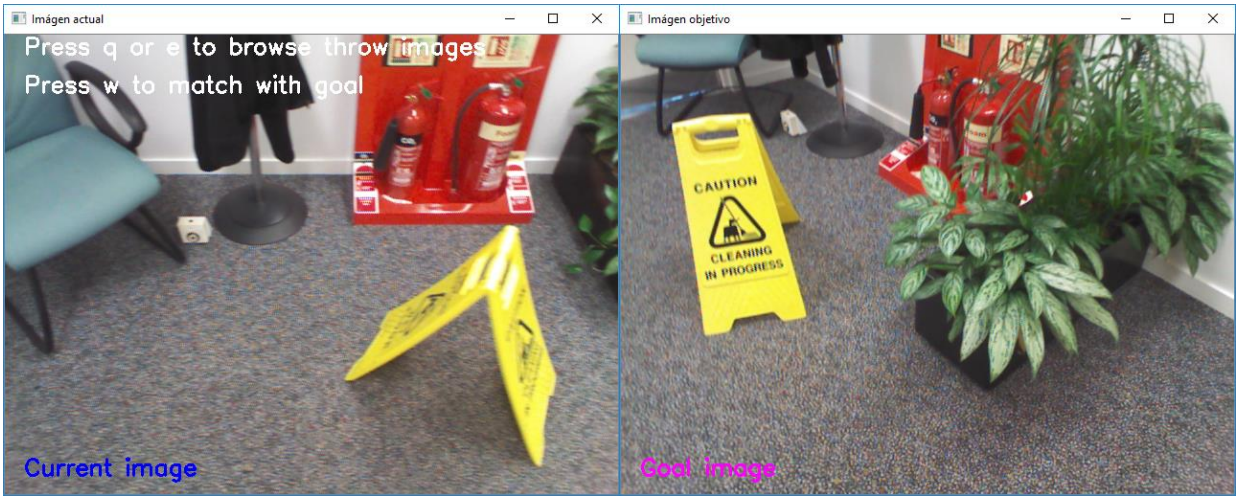

Figura 4.4 – Lectura de imágenes del dataset.

5 ANÁLISIS DE RESULTADOS

5.1 Introducción

En este capítulo se mostrarán los resultados de ejecución del proceso de resolución explicado anteriormente, para ambos casos de funcionamiento del sistema. Se comentarán dichos resultados, comparándose con los problemas descritos para sistemas de VPR en el capítulo 2.

5.2 Resultados obtenidos con la webcam

Se procede ahora a presentar los resultados obtenidos ejecutando la aplicación con la webcam. Se crea un escenario formado por un tablero de ajedrez y una torre formada por piezas de madera, sobre una mesa de madera y con un fondo formado por dos planos de color liso. Se ha escogido este escenario debido a que el tablero y la torre presentan abundantes puntos característicos, y el tablero presenta una fuerte simetría, siendo propenso a generar falsos positivos. Para evitar falsos positivos se aplica el filtro de radios de Lowe tras el proceso de matching. Se puede apreciar el efecto del filtro entre las figuras 5.1 y 5.2.

En la Figura 5.1 se muestra el matching entre dos fotogramas con cambio de perspectiva despreciable, pero sin filtrado. Las líneas unen los puntos característicos emparejados, y dado que no se produce movimiento entre los fotogramas deberían ser horizontales. Sin embargo, se aprecia que varios de las líneas enlazan puntos distintos, cruzando de forma diagonal entre las imágenes.

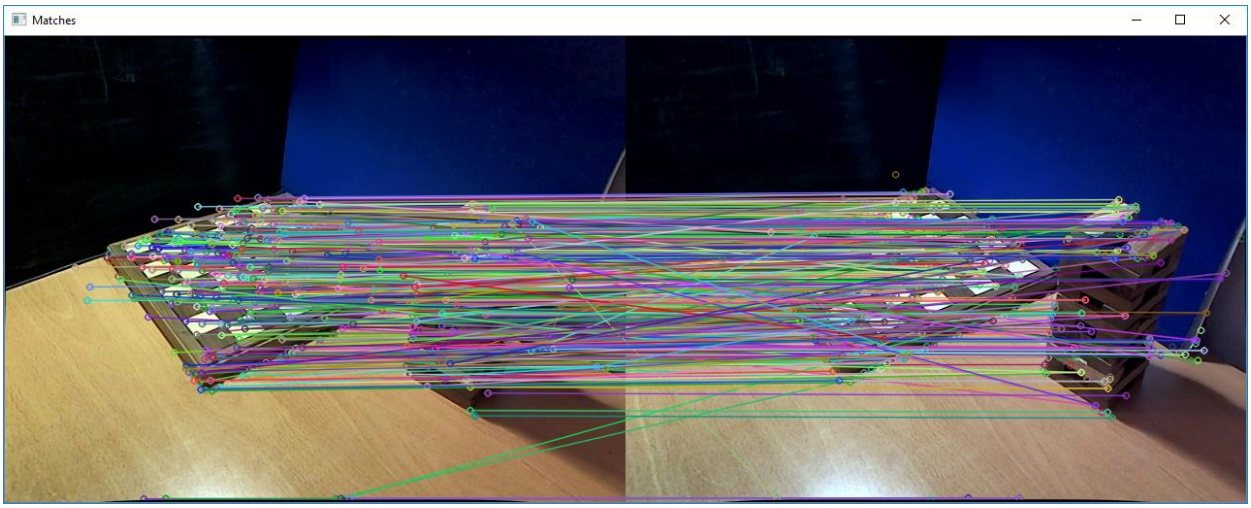

Figura 5.1 – Matching sin filtrado de imágenes capturadas por webcam.

En la Figura 5.2 se muestra dos capturas consecutivas, con un movimiento de traslación leve entre ellas. Se aprecia que las líneas que representan los *matches* presentan un cierto paralelismo, dado que la rotación entre ambas escenas es despreciable, y que no aparecen líneas cruzadas como en el caso anterior, siendo eliminadas por el filtro de Lowe.

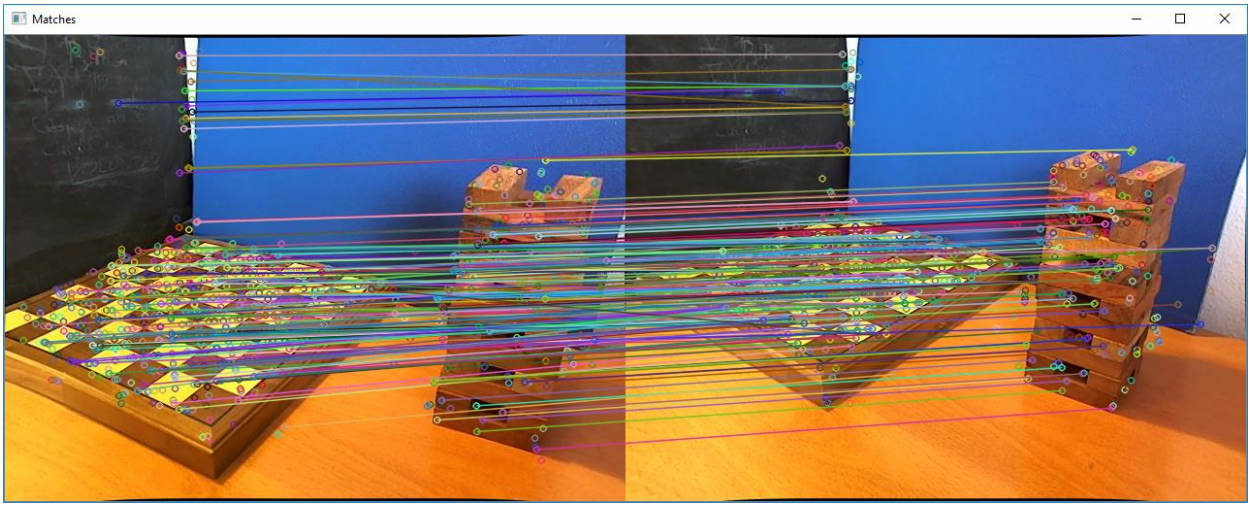

Figura 5.2 – Matching tras el filtrado de imágenes capturadas por webcam.

Como se ha comentado previamente, en la Figura 5.2 se muestran dos fotogramas donde la variación de orientación es leve mientras que si se da una traslación apreciable. Esto resulta en un número de *matches* alto, en proporción al número de puntos característicos, por lo que se tiene buena certeza de que se está visualizando el mismo lugar.

Como se explicó en el capítulo 2 los problemas de reconocimiento de lugar presentan mayores dificultades cuando la perspectiva varía fuertemente, como se observa en la Figura 5.3, donde se da un *matching* muy bajo, causando mucha incertidumbre en el reconocimiento de lugar.

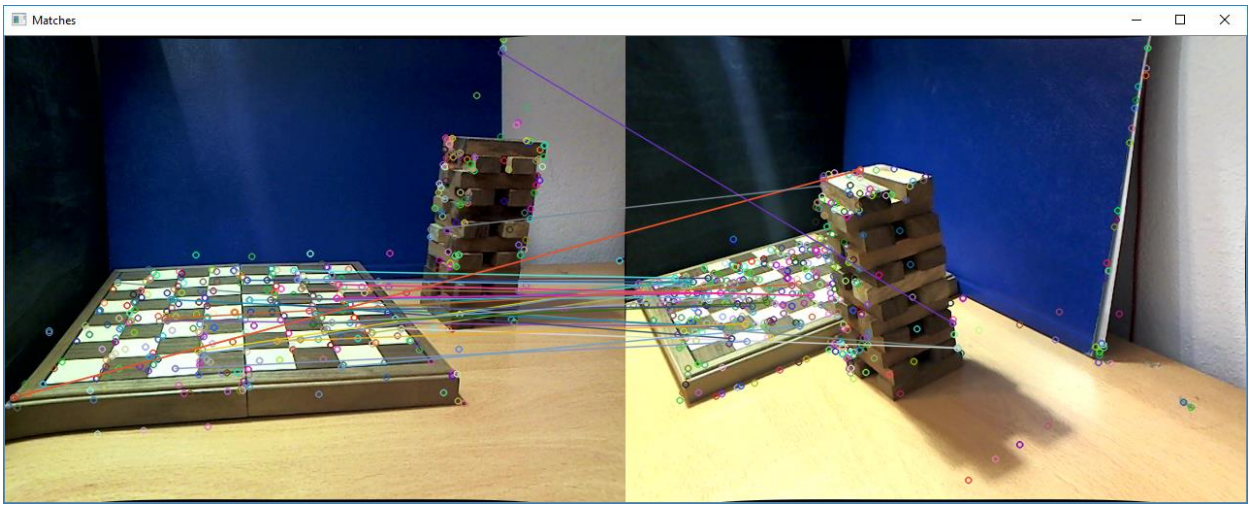

Figura 5.3 – Matching de imágenes capturadas por webcam con cambio brusco de perspectiva.

Los cambios de iluminación también causan que el rendimiento de la aplicación baje, debido a que en la fase de detección los gradientes de intensidad de píxel para cada canal de color varían, detectándose puntos característicos que no guardan relación con otras condiciones lumínicas. Este efecto se muestra en la Figura 5.4, donde, aunque no existe movimiento entre ambas imágenes, el número de puntos característicos emparejados es bajo. Se observa que en imágenes oscuras el tamaño del vector de puntos característicos es menor que con mucha iluminación.

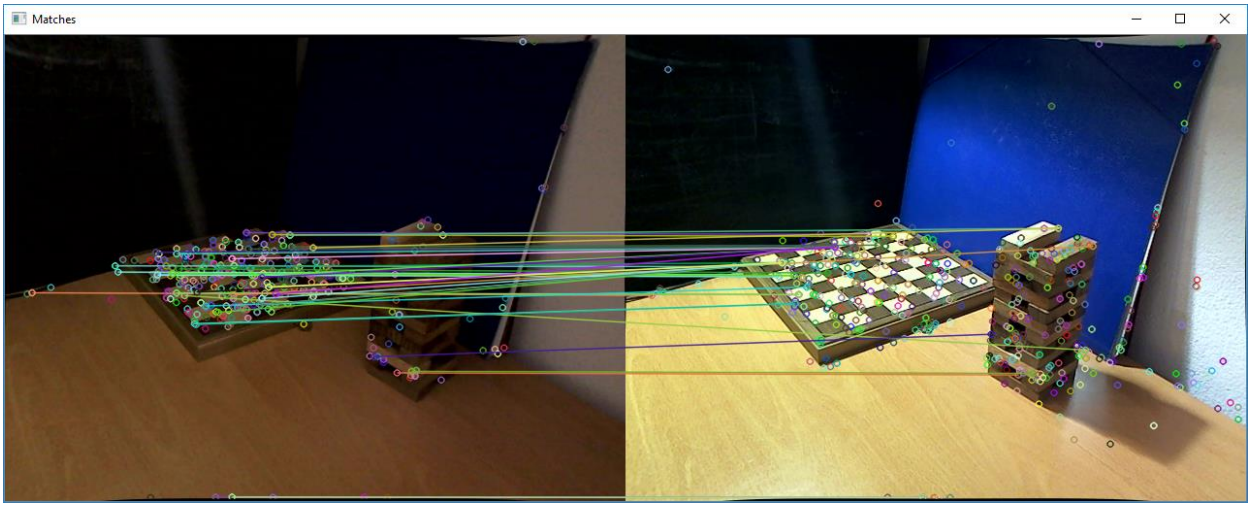

Figura 5.4 – Matching de imágenes capturadas por webcam con cambio de iluminación.

5.3 Resultados obtenidos con el dataset

Los resultados obtenidos al emplear el sistema con el dataset se muestran a continuación. Se realizará un análisis de las distintas situaciones casos contemplados en la elaboración del dataset; cambios de perspectiva y de iluminación.

En la Figura 5.5 se muestran dos imágenes del dataset donde se ve la misma escena en condiciones de iluminación idénticas, con una variación leve de perspectiva. Se observa que los puntos característicos se concentran en las baldosas del suelo de la imagen, mientras que los paneles muestran muchos menos puntos. Esto se debe a que las zonas con cambios bruscos de intensidad de píxel son mejores candidatas de puntos, mientras que superficies homogéneas que no presentan manchas o esquinas tienen un gradiente menor. Los bordes entre los objetos de la imagen también son zonas donde se observan muchos puntos.

Como se explicó en el apartado de SIFT y SURF, los candidatos de puntos de interés se obtienen aplicando una aproximación del gradiente en las imágenes, por lo que se justifica la aparición de puntos de interés concentrados en bordes y esquinas. En la Figura 5.5 se observa que se concentran muchas líneas en la zona de baldosas, con más contrastes, y en los tejados de la zona superior, mientras que los paneles no muestran prácticamente líneas de emparejado. Fue por este motivo que se incorporó esta escena en el dataset del proyecto, pues las superficies homogéneas son problemáticas de definir con técnicas de detección de *features*.

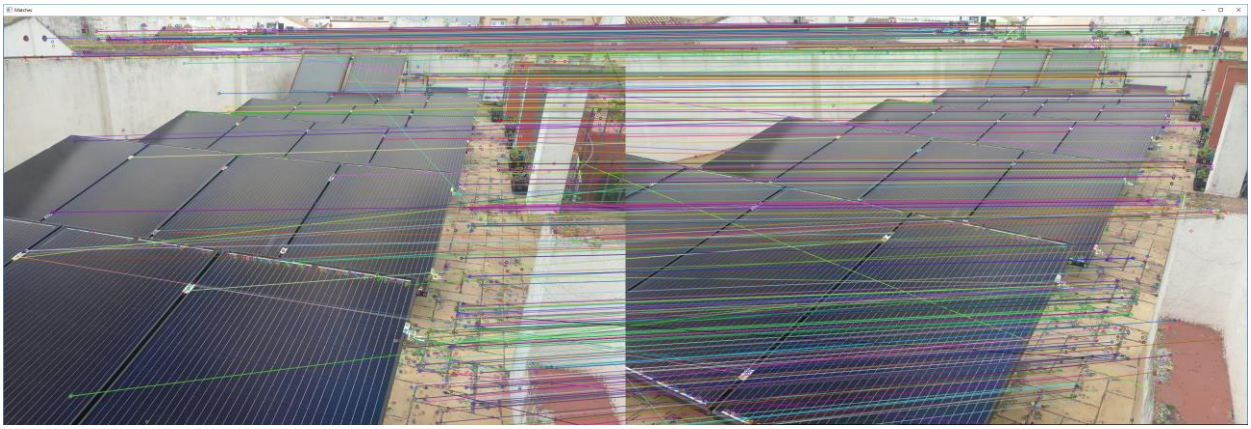

Figura 5.5 – Emparejado entre imágenes del dataset con cambio leve de perspectiva.

En la Figura 5.6 se muestran dos imágenes del dataset de la misma escena que presenta un cambio de perspectiva más acentuado que en la figura anterior. En esta escena los puntos de alto contraste se concentran principalmente en el centro, en el tablero de ajedrez, donde se observa una mayor concentración de líneas de

emparejado. Sin embargo, dado que el tablero es altamente simétrico también se observan varios *matches* incorrectos entre esquinas opuestas del tablero. Tanto en esta figura como en la anterior la proporción de puntos emparejados frente a puntos característicos totales es alta, más del 10%, siendo mayor la certeza de que se visualiza el mismo lugar.

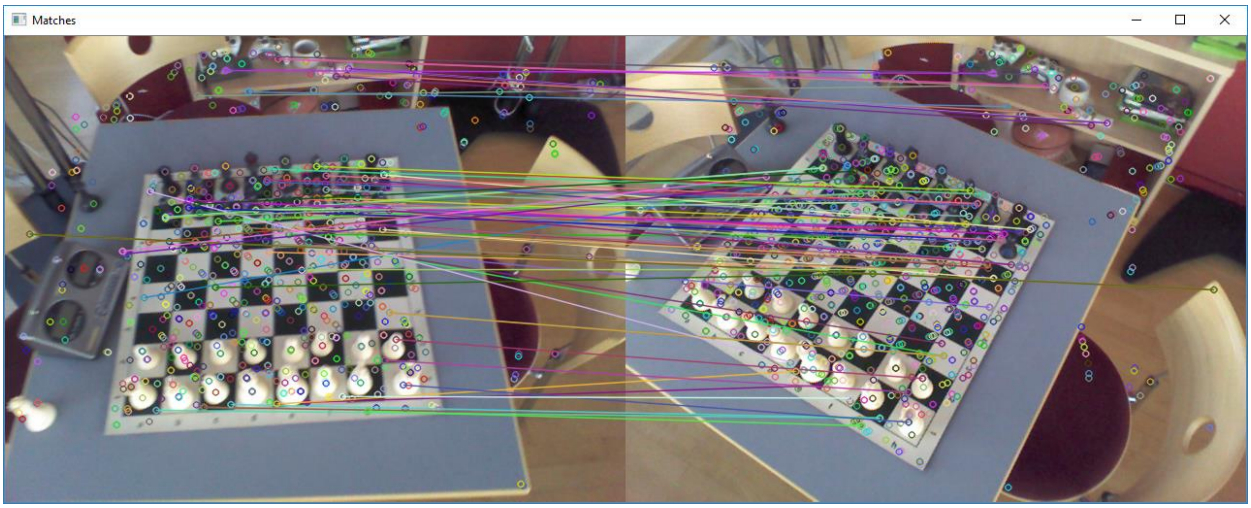

Figura 5.6 – Emparejado entre imágenes del dataset con cambio de perspectiva.

En la Figura 5.7 se observa el problema causado por cambios de iluminación, teniéndose que dos imágenes del mismo lugar muestren un número de *inliners*, puntos emparejados, pequeño frente al tamaño del vector de puntos característicos. La causa de esto es que los cambios de iluminación varían bruscamente la intensidad lumínica de los píxeles de la imagen, provocando que los gradientes en ambas sean distintos. Este problema es característico de los descriptores locales, como ya se introdujo en el capítulo 2.

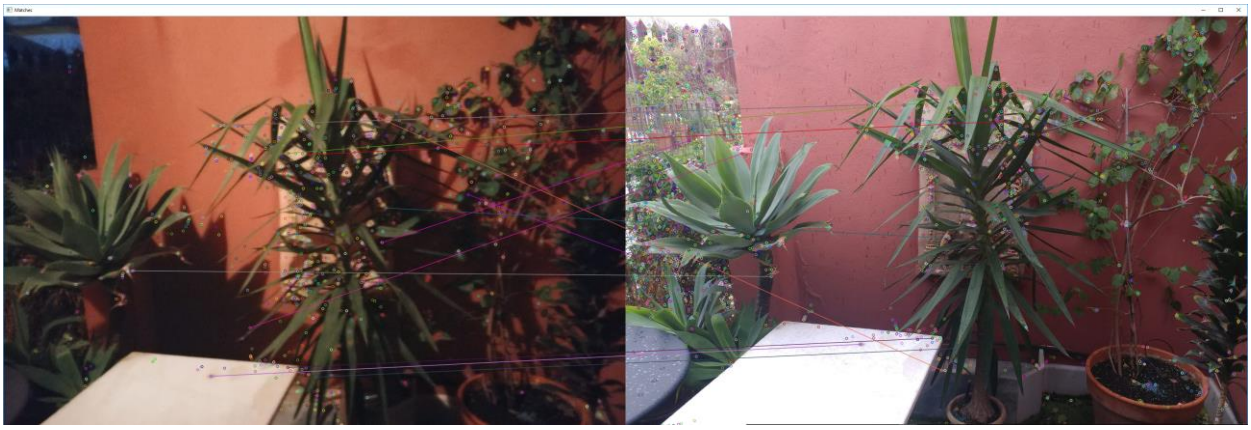

Figura 5.7 – Emparejado entre imágenes del dataset con cambio en la iluminación.

Por último, en la Figura 5.8 observamos la misma escena del tablero de ajedrez que en la Figura 5.6, salvo que esta vez la variación de la perspectiva es más brusca. Se observa que el sistema no responde bien ante estos cambios, pues el número de *inliners* es reducido. Sin embargo, pese a que en ambas escenas existen abundantes puntos característicos no se aprecian puntos emparejados incorrectamente.

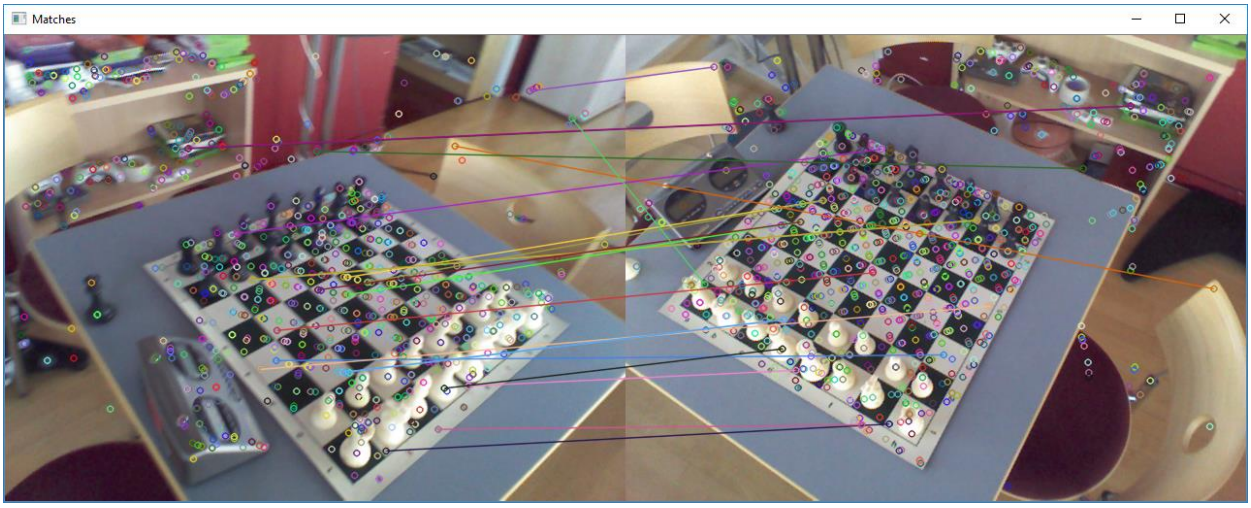

Figura 5.8 – Emparejado entre imágenes del dataset con cambio brusco de perspectiva.

5.4 Análisis del rendimiento del sistema

Se procede a realizar un análisis del rendimiento, calculando la evocación y la precisión del sistema según la formula expuesta en la Ecuación 1. Para ello se toma una imagen aleatoria del dataset y se realiza compara con todas las demás imágenes. Como las escenas son conocidas y están organizadas por nombre podemos comprobar los falsos positivos y falsos negativos mediante la comparación del nombre, y con esto calcular ambos indicadores. En la Tabla 5.1 se muestran los resultados del proceso, siendo la imagen escogida para comparar la mostrada en la Figura 5.9.

Los falsos positivos son cero, los falsos negativos son 6 y los positivos correctos son 2, resultando una precisión de un 100% y una evocación de un 25%, para este dataset.

Figura 5.9 – Imagen objetivo para el análisis del rendimiento.

Nombre de la imagen	Número de puntos de interés	Numero de coincidencias	Mismo lugar
chess-1	851	20	No
chess-2	752	58	No
chess-3	965	77	No
chess-4	535	10	No
chess-5	808	14	No
chess-6	321	8	No
chess-7	913	46	No
chess-8	1020	35	No
fire-1	337	337	Si
fire-2	400	109	Si
fire-3	627	41	No
fire-4	510	6	No
fire-5	416	6	No
fire-6	522	4	No
fire-7	697	11	No
fire-8	354	8	No
panels-1	2083	41	No
panels-2	1754	20	No
panels-3	2405	76	No
panels-4	1985	39	No
panels-5	1798	75	No
panels-6	1918	67	No
panels-7	2285	41	No
panels-8	1960	42	No
panels-day-1	3675	69	No
panels-day-2	2565	54	No
panels-day-3	2819	55	No
panels-day-4	1991	24	No
panels-day-5	3066	72	No
panels-day-6	3180	43	No
panels-day-7	3849	78	No
panels-day-8	3677	94	No
panels-night-1	854	9	No
panels-night-2	383	11	No
panels-night-3	625	11	No
panels-night-4	497	13	No
panels-night-5	635	17	No
panels-night-6	297	6	No
panels-night-7	536	9	No
panels-night-8	725	7	No
pumpkin-1	144	8	No
pumpkin-2	253	7	No
pumpkin-3	214	11	No
pumpkin-4	337	14	No
pumpkin-5	255	9	No
pumpkin-6	366	8	No

pumpkin-7	368	7	No
pumpkin-8	198	1	No
terrace-1	2374	22	No
terrace-2	1940	34	No
terrace-3	2122	34	No
terrace-4	1808	41	No
terrace-5	1970	43	No
terrace-6	1604	39	No
terrace-7	2069	47	No
terrace-8	1966	75	No
terrace-day-1	1101	23	No
terrace-day-2	2289	48	No
terrace-day-3	1472	45	No
terrace-day-4	3293	41	No
terrace-day-5	3015	65	No
terrace-day-6	2721	36	No
terrace-day-7	2205	31	No
terrace-day-8	2342	32	No
terrace-night-1	1591	43	No
terrace-night-2	2441	81	No
terrace-night-3	953	22	No
terrace-night-4	1627	58	No
terrace-night-5	1420	26	No
terrace-night-6	1464	35	No
terrace-night-7	1820	47	No
terrace-night-8	1611	27	No

Tabla 5.1 – Resultado de comparación del dataset.

6 CONCLUSIONES Y DESARROLLOS FUTUROS

6.1 Conclusiones

Se han estudiado y probado los algoritmos clásicos de detección de puntos característicos en imágenes más populares SIFT y SURF, comprobándose experimentalmente las limitaciones en su uso al tratar cambios abruptos de perspectiva o iluminación. Los algoritmos empleados se han implementado haciendo uso de las librerías de OpenCV en el lenguaje C++, creándose una aplicación interactiva para experimentar con estos algoritmos, visualizando los resultados en imágenes.

Los resultados obtenidos demuestran que el sistema presenta un resultado pobre al lidiar con los problemas mencionados, pero da resultados muy robustos en condiciones apropiadas. Como se mencionó en la introducción el problema de visual place recognition ha tenido un resurgimiento con la aparición de las técnicas de deep learning, que permiten desarrollar algoritmos más avanzados basados en el uso de imágenes descritas mediante estos algoritmos, pero mejoran la eficiencia del sistema de reconocimiento permitiendo por ejemplo realizar tareas de clasificación y etiquetado de lugar según los objetos detectados, entre otros. Deep learning consiste en aplicar modelos computacionales de aprendizaje automático, inspirados en el comportamiento de las redes neuronales biológicas, entrenadas para realizar una tarea específica, como en [17], donde se emplea una red neuronal convolucional para la detección de objetos en tiempo real.

El problema de VPR es de gran interés, y se siguen desarrollando soluciones en busca de un sistema que permita navegar de forma completamente autónoma y segura a robots en entornos con humanos presentes, un problema complejo de resolver.

Durante el capítulo 2 se ha introducido el concepto de VPR, así como los problemas que se deben resolver para su uso viable en sistemas que se puedan considerar completamente autónomos. Se concluye que estos sistemas deben afrontar aun varios problemas que causan que no se viable su uso en sistemas de navegación autónomos de forma segura. Dichos problemas surgen de las limitaciones tecnológicas de estos sistemas y constituyen un campo de estudio que aún está madurando.

Para la resolución del problema se ha preparado un conjunto de imágenes, y se ha desarrollado una aplicación que, mediante el uso de las librerías de OpenCV, permite implementar una solución presentada en el capítulo 4. Esta definición se propone tras la consideración de la necesidad de una estructura modular para el desarrollo de la aplicación, propuesta en el capítulo 3. Dada la dificultad de obtener un set de imágenes que cumpla las necesidades con el que probar el sistema, se toma la decisión de generar un dataset propio para este proyecto para evaluar el rendimiento.

En el capítulo 5 se analizan los resultados obtenidos procediendo como se indica en proceso de resolución. Tras el análisis del rendimiento se obtiene una alta precisión a costa de un nivel de evocación muy bajo, sugiriendo que nuestro sistema no es capaz de identificar correctamente los lugares presentados al ser sometido a los problemas descritos en el capítulo 2. Sin embargo, en el caso más favorable, la respuesta del sistema es la esperada siendo capaz de reconocer la localización.

Durante el desarrollo del proyecto la mayor dificultad ha consistido en el desarrollo de dicha aplicación, dada la falta de familiaridad con lenguajes de programación orientada a objeto como C++. Otras dificultades encontradas han sido el estudio del problema de reconocimiento visual de un lugar, dado que es un campo aun en desarrollo con múltiples publicaciones, estudios y artículos que tratan este problema de muy diversas formas.

Se concluye este proyecto con los objetivos definidos en el capítulo 4 cumplidos, teniendo una aplicación final y un set de imágenes válidos y probados.

6.2 Mejoras y desarrollos futuros

En el ámbito de la visión artificial existen muchos problemas en constante evolución, obteniendo resultados cada vez más eficientes y precisos mediante técnicas avanzadas como machine learning y deep learning. En este proyecto se han empleado algunos de los algoritmos más empelados para la descripción de puntos

característicos de imágenes y se han aplicado para la resolución de un problema de reconocimiento de lugar. Existen otros problemas con funcionamientos similares como el reconocimiento de objetos, de caras o la realidad aumentada que hacen uso de estos algoritmos y también resultan de interés.

En cuanto a la aplicación desarrollada, sería deseable mejorar la eficiencia de esta, planteando estrategias más avanzadas para la resolución del problema de VPR. Además, computacionalmente la aplicación resulta costosa, no siendo aprovechable en aplicaciones de tiempo real. Por otro lado, el conjunto de imágenes empleado debe ser ampliado y mejorado para contemplar todas las situaciones que potencialmente puedan encontrarse al utilizar un sistema de navegación autónomo, de forma que se pueda experimentar y que puedan afrontarse de forma segura soluciones para la navegación autónoma basadas en VPR.

7 REFERENCIAS

[1] O. S. C. V. L. (OpenCV). [En línea]. Available: https://opencv.org/.

[2] E. C. Tolan, «Cognitive maps in rats and men,» *Psychol. Rev.,* vol. 55, nº 4, pp. 189-208, 1948.

[3] F. Strumwasser, «Long-term recording from single neurons in brain of unrestrained mammals,» *Science,* vol. 127, nº 3296, pp. 469-470, 1958.

[4] J. O. a. J. Dostrovsky, «The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat,» *Brain Res.,* vol. 34, nº 1, pp. 171-175, 1971.

[5] J. O'Keefe, «Place units in the hippocampus of the freely moving rat,» *Exp. Neurol.,* vol. 51, nº 1, pp. 78-109, 1976.

[6] L. I. C. Siagian, «Biologically inspired mobile robot vision localization,» *IEEE Trans. Robot,* vol. 25, nº 4, p. 861–873, 2009.

[7] J. Canny, «A computational approach to edge detection,» *IEEE Trans. Pattern Anal. Mach. Intell.,* Vols. %1 de %2PAMI-8, nº 6, pp. 679-698, 1986.

[8] C. H. a. M. Stephens, «A combined corner and edge detector,» de *Proc. 4th Alvey Vis. Conf*, 1988.

[9] D. Lowe, «Object recognition from local scale-invariant features,» 1999.

[10] T. T. a. L. V. G. H. Bay, «SURF: Speeded up robust features,» de *Proc. Eur. Conf. Comput. Vis*, 2006.

[11] G. S. M. C. P. N. a. I. R. C. Mei, «A constanttime efficient stereo SLAM system,» de *Brit. Mach. Vis. Conference*, Londres, U.K., 2009.

[12] A. M. R. a. M. Cazorla, «Comparativa de detectores de características visuales y su aplicación al SLAM,» de *X Workshop de agentes físicos*, Cáceres, 2009.

[13] A. O. a. A. Torralba, «Visual Perception – Fundamentals of Building the gist of a scene: The role of global image features in recognition,» de *Awareness: Multi-Sensory Integration and High-Order Perception*, New York, NY, USA, Elsevier, 2006, pp. 23-36.

[14] S. M. a. D. I. A. Ranganathan, «Towards illumination invariance for visual localization,» de *Proc. IEEE Int. Conf. Robot. Autom.*, 2013.

[15] Intel. [En línea]. Available: https://www.intel.com/content/www/us/en/homepage.html.

[16] Microsoft, «RGB-D Dataset 7-Scenes,» 2013. [En línea]. Available: https://www.microsoft.com/en-us/research/project/rgb-d-dataset-7-scenes/.

[17] K. H. R. G. J. S. Shaoqing Ren, «Faster R-CNN: Towards Real-Time Object Detection,» 2015.

[18] B. G. C. Z. S. I. A. C. A. F. Jamie Shotton, «Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images | Proc. Computer Vision and Pattern Recognition (CVPR),» *IEEE,* Junio 2013.

8 GLOSARIO

VPR: *Visual Place Recognition*

SLAM: *Simultaneous Location And Mapping*

SIFT: *Scale-Invariant Feature Transforms*

SURF: *Speeded-Up Robust Features*