

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA
DEPARTAMENTO DE INGENIERÍA DEL SOFTWARE E INTELIGENCIA ARTIFICIAL



Reconocimiento de una plataforma para aterrizaje de UAVs mediante procesamiento de imágenes

LUIS ARROYO LARA
DAVID CABALLERO RUIZ
EDUARDO PUERRO CEPA

TRABAJO DE FIN DE GRADO EN INGENIERÍA DEL SOFTWARE
DIRIGIDO POR: GONZALO PAJARES MARTINSANZ

MADRID, ESPAÑA

JUNIO, 2017

“Live together, die alone.”
Jack Shephard. Lost.

Agradecimientos

“Wanderlust.”

Quiero comenzar agradeciendo a Gonzalo Pajares Martinsanz por brindarnos la oportunidad de realizar este proyecto y habernos guiado durante el transcurso de este. A mis compañeros de proyecto, David y Eduardo, porque una vez más hemos demostrado que somos un equipo y podemos con todo. Al resto de compañeros que junto a David y Eduardo hemos ido avanzando peldaño a peldaño en la carrera hasta llegar al punto en el que hoy nos encontramos. A mi amigo “Maru” con el cual empecé en este maravilloso mundo de la informática y el cual se ha vuelto imprescindible en mi vida, gracias por tanto. A mi prima Ana, que más que una prima para mí es mi hermana mayor, has sido siempre un referente para mí. A mi “sister” de ti he intentado copiar tu capacidad de trabajo, estoy tremendamente orgulloso de ti, la vida es más fácil con una hermana como tú. A mi madre, a la cual la debo todo lo que soy, por no rendirte nunca conmigo, sin tu tesón a lo largo de los años nunca habría llegado a estar donde estoy hoy, este proyecto y la carrera también son tuyas. A mi abuela, mi otra madre, por enseñarme lo que es el amor puro, te llevo muy dentro. Y a ti Papá, por haberme dejado volar alto, por tu lucha y valentía, me has dejado la mejor lección de vida que un padre puede dejarle a un hijo, este proyecto te lo dedico a ti. Y a todos aquellos que de un modo u otro habéis formado parte de mi vida, gracias por enriquecerme.

Luis Arroyo Lara.

“Winter is coming.”
House Stark.

Llegó el momento de concluir esta gran aventura y, me gustaría hacerlo dando las gracias a todas las personas cómplices de ello. En primer lugar, a Gonzalo Pajares Martinsanz por la confianza depositada en nosotros desde el principio y durante el desarrollo del proyecto. A la Universidad Complutense de Madrid por haberme hecho partícipe de esta experiencia y por haberme brindado cada uno de los momentos vividos en estos últimos años, así como las personas que han ido ligadas a ellos, tantas son, que únicamente ellos sabrán de quienes se trata. Más concretamente, agradezco de corazón al grupo de trabajo con el que comenzó y terminó todo, ya que sin vuestras risas, horas de estudio e incluso enfados, me habría perdido en esta aventura cuya meta me habéis impulsado a alcanzar. En especial doy las gracias a Eduardo y Luis por haber cumplido de nuevo con creces nuestras expectativas y haber sido capaces de conseguir resultados muy gratificantes en este último proyecto juntos. Y para terminar con la Universidad Complutense y enlazar con mi vida personal, agradezco a Sergio Sánchez su continuo apoyo, siempre en forma de sonrisa y humor, lo que me ha aportado mucha confianza en mí mismo, y a Luis Arroyo, del que no olvidaré la infinita paciencia demostrada durante estos años y el haber sido capaz de descubrir en mí algo que realmente me apasiona, la programación. Gracias a los dos. Agradezco también a todas las personas que aún estando lejos, en Tenerife, han conseguido transmitirme fuerza y cariño constantemente, entre los que destaco a mis “Pokémon” y a mis segundos padres. No me olvido de mi primo “Dani” que sin darse cuenta y gracias a su empeño me ha servido de apoyo y ejemplo para no caer durante este largo año. Dar las gracias a mis padres y mi hermana, por su incesante confianza en mí y por no haber dejado de motivarme en ningún momento hasta que he conseguido alcanzar tan anhelada meta. En particular a mi padre, que nos aportó la idea definitiva sobre el material a utilizar, sin escatimar tiempo ni esfuerzo alguno. Y por último, y en esta ocasión sí más importante, gracias a ti, Fabi Pérez. Durante estos años has sido mi mayor fuente de felicidad y motivación, sin ti nunca me habría embarcado en esta aventura. De ti he aprendido que no hay que rendirse independientemente de la situación, que es primordial alejarse de la soledad y, ‘come what may’, no debemos dejar de confiar en el Destino porque puede que un día te sorprenda y cambie tu vida para siempre.

David Caballero Ruiz.

“Todo fluye, nada permanece quieto.”
Heráclito.

Durante cuatro años nos hemos postrado ante un duro destino sin más armas que lo puesto. Hemos sido esclavos de lo que intentábamos ser mientras que sólo podíamos contentarnos con lo que éramos. A pesar de todo no íbamos solos y la compañía de la gente fue la que en muchos momentos nos dio el ánimo de seguir y no tirar la toalla. Doy las gracias a Gonzalo Pajares Martinsanz que nos propuso este proyecto y gracias a su apoyo este año hemos descubierto una rama de la informática que, en cierto modo, tan en las sombras teníamos. De la misma manera a la Facultad de Informática y a su gente que aportó su granito de arena en contribuciones pequeñas, pero grandes en conjunto... No sé qué sería de mi sin ella. Marisol. Tu apoyo compuesto de comprensión y paciencia han sido necesarios en cada paso. Es la diferencia entre lo que era y lo que soy. Sin ella probablemente no hubiera llegado tan lejos y ni siquiera hubiera aprendido tanto como lo he hecho. Aquellos compañeros de estudio que, en tantas ocasiones, te prohibían quedarte atrás y gracias a ello nos encontramos aquí. Mi familia, constante eslabón de la cadena que, muchas veces, me ataba a la realidad. Siempre habéis estado ahí y siempre habéis sido cómplices de lo que yo quería ser, ayudándome a no desviarme demasiado ni andar caminos que nunca quise. En especial mi abuela, que tanto ha sufrido y llorado por sus nietos y que, ahora, sin enterarse demasiado, se pone contenta cuando le comentas que estás terminando la carrera. Tus sacrificios fueron grandes y por desgracia mis agradecimientos siempre insuficientes.

Eduardo Puerro Cepa.

Resumen

A lo largo del tiempo y gracias a los grandes avances tecnológicos, el ser humano ha intentado automatizar cada uno de los procesos que intervienen en su día a día con la intención de facilitar la vida de las personas. Uno de estos grandes avances es la posibilidad de enviar vehículos aéreos no tripulados (Unmanned Aerial Vehicle (UAV)) a zonas de difícil acceso para el ser humano, ya sea por razones ambientales, de seguridad o porque suponga un coste económico insostenible. Sin embargo, existen varios problemas que acotan las situaciones en las que este tipo de avances pueden ser aplicados, entre ellos la poca precisión transmitida por la geolocalización actual.

Es por esto que en el presente proyecto se propone una solución al problema de ubicar un UAV en un punto exacto, de manera que realmente se convierta en un vehículo no tripulado y no dependa de forma directa del ser humano. El proyecto donde se circunscribe el desarrollo que se presenta tiene como título “Sistema Autónomo para la Localización y Actuación ante Contaminantes en el Mar (SALACOM, 2013) (DPI2013-46665-C1)” subvencionado por el Ministerio de Economía, Industria y Competitividad donde se plantea la colaboración entre barcos autónomos y drones. Estos últimos teniendo la necesidad de aterrizar sobre una plataforma situada a bordo de un barco. Por tanto, en este proyecto se plantea el reconocimiento de una plataforma de aterrizaje mediante un sistema de visión por computador basado en el análisis de dos tipos de imágenes, térmicas y de color, captadas por sendos dispositivos, en este caso instalados en un teléfono móvil (smartphone).

La plataforma está específicamente diseñada con tal propósito. En efecto, la misma consta de una base de madera sobre la que se han situado cinco figuras geométricas metálicas de distintos colores. Las figuras geométricas, estratégicamente posicionadas entre sí, configuran una disposición geométrica con nula posibilidad de aparición en los entornos de operación del UAV. Los diferentes colores permiten discernir elementos geométricos con distinta forma espectral. Finalmente, su naturaleza metálica permite distinguir distintas reflectancias térmicas en el espectro visible. Para el reconocimiento se dispone siempre de una imagen en color y otra infrarroja. De esta forma se combinan dos procedimientos que aumentan el grado de precisión en la identificación.

Esto es lo ideal y este es el marco del proyecto, al margen de que estas mismas técnicas se utilicen en otros contextos, no necesariamente en colaboración.

Palabras clave: Visión artificial, Procesamiento de imágenes, Reconocimiento, Plataforma de aterrizaje, Vehículos aéreos no tripulados (UAVs).

Abstract

Over time and thanks to the great technological advances, mankind has tried to automate each of the processes that intervene in daily life with the intention of making life easier for people. One of these is the possibility of sending Unmanned Aerial Vehicles (UAV) to areas that are difficult for human beings to access, either for environmental reasons, safety reasons or because they entail an unsustainable economic cost. However, there are several problems that limit the situations in which this type of advances can be applied, such as the little precision transmitted by the current geolocation.

For this reason, the present project proposes a solution to the problem of locating a UAV at an exact point, so that it really becomes an unmanned vehicle and doesn't depend directly on the human. The project where the development is presented is entitled "Autonomous System for Localization and Action against Pollutants in the Sea (SALACOM, 2013) (DPI2013-46665-C1)" funded by the Ministry of Economy, Industry and Competitiveness where it raises the collaboration between autonomous ships and drones. The latter needing to land on a platform aboard a ship. Therefore, this project involves the recognition of a landing platform through a computer vision system based on the analysis of two types of images, thermal and color, captured by two devices, in this case installed in a Smartphone.

The platform is specifically designed for this purpose. In fact, it consists of a wooden base on which five metallic figures of different colors have been placed. The geometric figures, strategically positioned between them, configure a geometric arrangement with no possibility of appearance in the operating environments of the UAV. The different colors allow geometrical elements with different spectral shapes to be discerned. Finally, its metallic nature allows different thermal reflectances in the visible spectrum to be distinguished. A color and an infrared image are always available for recognition. In this way two procedures are combined that increase the degree of accuracy in identification.

This is the ideal scenario and the framework of the project, regardless of whether these same techniques are used in other contexts, not necessarily in collaboration.

Keywords: Artificial vision, Image processing, Recognition, Landing platform, Unmanned Aerial Vehicles (UAVs).

Índice

Agradecimientos	IV
Resumen/Abstract	VIII
Abreviaturas	XIV
Índice de figuras y tablas	XV
1. Introducción/Introduction	1
1.1. Consideraciones generales	1
1.2. General considerations	3
1.3. Motivación y objetivos	4
1.4. Motivation and goals	4
1.5. Plan de trabajo	5
1.6. Workplan	5
1.7. Estructura de la memoria	6
2. Estado del arte	7
3. Métodos de vision artificial	12
3.1. Procesamiento de imágenes	12
3.1.1. Viola y Jones	12
3.1.2. Etapas del procesamiento	14
4. Desarrollo del proyecto	21
4.1. Trabajo previo de investigación	21
4.2. Proceso de trabajo	22
4.3. Estructura de la aplicación	23
4.4. Descripción del equipamiento	24
4.4.1. Equipamiento hardware	24
4.4.2. Equipamiento software	26
4.5. Casos de uso	27
4.5.1. Limpieza imagen espectro infrarrojo	30
4.5.2. Limpieza imagen espectro visible	31
4.5.3. Procesamiento de la imagen	32
4.5.4. Decisión de la validez de la imagen	33
4.6. Análisis y desarrollo de la solución propuesta	35
4.6.1. Tratamiento de imágenes térmicas	35
4.6.2. Tratamiento de imágenes en color	37
4.6.3. Procesamiento de las regiones de interés	40
4.6.4. Algoritmo de decisión	42
4.7. Plan de gestión de riesgos	43
4.7.1. Identificación de los riesgos	44
4.7.2. Análisis de los riesgos	44
4.7.3. Priorización de los riesgos	48
4.7.4. Estudio y control de los riesgos	48

5. Implementación	49
5.1. Matlab	49
5.2. Técnicas de procesamiento aplicadas	50
5.2.1. Funciones básicas	50
5.2.2. Funciones aplicadas al procesamiento de imágenes	50
5.3. Arquitectura software	53
5.4. Metodologías utilizadas	54
5.4.1. Modelo-Vista-Controlador (MVC)	54
5.4.2. Singleton	55
6. Análisis de resultados	55
6.1. Combinación de imágenes térmica-visible	56
6.2. Fallos detectados por tipo de imagen	59
7. Conclusiones y trabajo futuro	62
7.1. Conclusiones	62
7.2. Conclusions	64
7.3. Trabajo futuro	66
7.4. Contribuciones personales	67
7.4.1. Luis Arroyo Lara	67
7.4.2. David Caballero Ruiz	68
7.4.3. Eduardo Puerro Cepa	70
Referencias	75
Anexo I : Pruebas de campo	76
Anexo II : Manual de uso de la aplicación	78

Abreviaturas

- **GPS:** Global Positioning System.
- **HMI:** Human Machine Interface.
- **HSV:** Hue, Saturation, Value.
- **IHM:** Interfaz Hombre Máquina.
- **LoG:** Laplacian of Gaussian.
- **OTG:** On-The-Go.
- **RGB:** Red, Green, Blue.
- **SALACOM:** Sistema Autónomo para la Localización y Actuación ante Contaminantes en el Mar.
- **UAV:** Unmanned Aerial Vehicle.
- **UC:** Use Case.
- **UML:** Unified Modeling Language.
- **VGA:** Video Graphics Array.

Índice de figuras y tablas

Índice de figuras

1.	Plataforma de aterrizaje patentada con estructuras geométricas características.	2
2.	Plataforma de aterrizaje diseñada con figuras geométricas coloreadas y metálicas.	2
3.	Plataforma de aterrizaje de drones para entregas de Amazon.	9
4.	Plataforma de aterrizaje de Lange.	10
5.	Plataforma de aterrizaje de Lange.	10
6.	Plataforma de aterrizaje de Saripalli.	10
7.	Plataforma de aterrizaje de Sharp.	11
8.	Imagen analizada por el algoritmo de Casetti.	11
9.	Ejemplo de formas rectangulares básicas.	13
10.	Árbol de decisión.	14
11.	Principales etapas del procesamiento de imágenes.	14
12.	Binarización de una imagen con un valor umbral por defecto.	16
13.	Binarización de una imagen con el método de Otsu.	16
14.	Extracción de bordes mediante Canny.	18
15.	Rellenado de regiones y huecos.	18
16.	Rellenado de regiones y huecos con otros elementos.	18
17.	Aplicación de operaciones morfológicas de apertura y cierre.	19
18.	Aplicación de exclusión de objetos.	19
19.	Smartphone Moto 4G Plus junto al dispositivo FLIR ONE.	24
20.	Imagen infrarroja captada con la cámara del dispositivo FLIR ONE de esta naturaleza.	25
21.	Imagen de color captada con un iPhone 6.	25
22.	Diagrama de funcionamiento de la aplicación.	29
23.	Diagrama de actividad del caso de uso número 4.	34
24.	Imagen infrarroja de la plataforma.	35
25.	Imagen en color de la plataforma.	35
26.	Binarización de una imagen en un día nublado.	36
27.	Binarización de una imagen en un día soleado.	36
28.	Aplicaciones morfológicas cuyo elemento estructural es un disco de radio doce.	37
29.	Aplicaciones morfológicas cuyo elemento estructural es un disco de radio cuatro.	37
30.	Imagen de color con sombras sobre la plataforma.	38
31.	Imagen una vez se ha realizado la conversión entre el modelo RGB y HSV.	38
32.	Aplicación del método de Otsu en combinación con el operador Canny.	39
33.	Rellenado de regiones obtenidas por el método anterior.	39
34.	Aplicación de operaciones morfológicas para suavizar la imagen.	39
35.	Eliminación de elementos en función de su tamaño.	40
36.	Obtención del color de las figuras identificadas.	42
37.	Cuadro de figuras en una imagen térmica.	43
38.	Cuadro de figuras en una imagen visible.	43
39.	Diagrama de clases de la aplicación.	53

40.	Combinación imagen térmica - visible con un grado de acierto del 100 %.	56
41.	Combinación imagen térmica - visible con un grado de acierto del 100 %.	56
42.	Combinación imagen térmica (40 %) - visible (100 %) con un grado de acierto del 76 %.	57
43.	Combinación imagen térmica (100 %) - visible (60 %) con un grado de acierto del 76 %.	57
44.	Combinación imagen térmica (60 %) - visible (80 %) con un grado de acierto del 72 %.	58
45.	Combinación imagen térmica (80 %) - visible (60 %) con un grado de acierto del 68 %.	58
46.	Combinación imagen térmica (100 %) - visible (40 %) con un grado de acierto del 64 %.	59
47.	Imagen térmica con un grado de acierto del 60 %.	59
48.	Imagen térmica con un grado de acierto del 20 %.	60
49.	Imagen visible con un grado de acierto del 80 %.	60
50.	Imagen visible con un grado de acierto del 60 %.	61
51.	Imagen visible con sombras con un grado de acierto del 100 %.	61
52.	Imagen visible con sombras con un grado de acierto del 80 %.	62
53.	Prueba de campo interior uno.	76
54.	Prueba de campo exterior uno.	76
55.	Prueba de campo exterior dos blanca.	77
56.	Prueba de campo exterior dos negra.	77
57.	Prueba de campo exterior tres.	77
58.	Captura de pantalla inicio-reset.	78
59.	Captura de pantalla selección de imagen térmica.	79
60.	Captura de pantalla carga de imagen térmica.	79
61.	Captura de pantalla procesado de imagen térmica.	80
62.	Captura de pantalla procesado de imagen visible.	81
63.	Captura de pantalla carga de imagen térmica y visible.	82
64.	Captura de pantalla procesado de imagen térmica y visible.	82
65.	Captura de pantalla simulación.	83
66.	Captura de pantalla procesado de imágenes simuladas.	84
67.	Captura de pantalla fin simulación.	84

Índice de tablas

1.	Plan de trabajo.	5
2.	Workplan.	6
3.	Especificaciones técnicas del dispositivo FLIR ONE.	25
4.	Especificaciones técnicas de la cámara del Smartphone iPhone 6.	25
5.	Especificaciones técnicas del procesador iMac 2011.	26
6.	Especificaciones técnicas del procesador Mac Air 2013.	26
7.	Especificaciones técnicas del procesador G551JK.	26
8.	Caso de uso número uno. Limpieza de imagen en espectro infrarrojo.	30
9.	Caso de uso número dos. Limpieza de imagen en espectro visible.	31
10.	Caso de uso número tres. Procesamiento de la imagen.	32

11.	Caso de uso número cuatro. Decisión de la validez de la imagen.	33
12.	Identificación de figuras en función del valor de su redondez en imágenes térmicas.	41
13.	Identificación de figuras en función del valor de su redondez en imágenes visibles.	42
14.	Riesgo número uno.	44
15.	Riesgo número dos.	45
16.	Riesgo número tres.	45
17.	Riesgo número cuatro.	45
18.	Riesgo número cinco.	45
19.	Riesgo número seis.	46
20.	Riesgo número siete.	46
21.	Riesgo número ocho.	46
22.	Riesgo número nueve.	47
23.	Riesgo número diez.	47
24.	Riesgo número once.	47
25.	Riesgo número doce.	47
26.	Priorización de los riesgos	48
27.	Plan de actuación para el riesgo número dos.	48
28.	Plan de actuación para el riesgo número ocho.	49

1. Introducción/Introduction

En esta primera sección se plantea la introducción al presente proyecto y se exponen los objetivos y el plan de trabajo seguido durante el desarrollo del mismo.

1.1. Consideraciones generales

Es bien sabido el problema que supone cualquier vertido en el mar, particularmente si éste es petróleo. Buena prueba de ello fue la crisis que supuso el naufragio del Prestige en 2002 que afectó tanto a las aguas españolas como a la costa. Este desastre supuso graves perjuicios a la fauna y flora marina. En este caso el vertido estaba localizado, si bien a veces esta situación no es lo habitual siendo necesaria la búsqueda de vertidos contaminantes, que por otra parte pueden ser de cualquier naturaleza, no necesariamente restringidos a petróleo. En consecuencia, uno de los principales problemas consiste en la detección de la mancha de vertido en el mar.

Con el propósito de dar solución a esta ardua tarea nace el proyecto [SALACOM \(2013\)](#), el cual propone el uso de un barco y un Unmanned Aerial Vehicle (UAV) y por tanto autónomo, para la detección de material contaminante. Estos vehículos se conocen también como drones. En este trabajo se usan indistintamente ambos términos (UAV y dron). La función del UAV consiste en explorar la escena por delante del barco localizando zonas con residuos contaminantes. Posteriormente, localizado el mismo, envía la información y posicionamiento correspondiente al centro de control para que el barco acuda a la localización marcada para actuar en consecuencia. Sin embargo, para realizar esta tarea es necesario otorgar la suficiente autonomía al UAV con el objetivo de que éste pueda cubrir zonas de exploración lo más amplias posible.

Con tal propósito, es necesario que el UAV tenga la capacidad de recargar sus baterías por sí solo en una plataforma situada en la superficie del barco. Para ello, tendrá que ser capaz de reconocerla, posarse sobre ella y proceder a la correspondiente recarga hasta estar preparado para despegar de nuevo y proseguir con la exploración, bien realizando nuevas búsquedas o informando sobre la situación ya encontrada.

El presente proyecto de Fin de Grado, aborda una parte del proyecto de investigación [SALACOM \(2013\)](#). Se basa en desarrollar una aplicación informática cuyo objetivo es detectar la plataforma ya mencionada. El procedimiento se ha desarrollado mediante técnicas de procesamiento de imágenes correspondientes al espectro de luz visible e infrarrojo con el fin de que el UAV localice la plataforma con unos patrones preestablecidos y consiga aterrizar en ella con éxito situándose con la orientación adecuada. De esta forma quedaría resuelto el problema de otorgar mayor autonomía al UAV.

Para el desarrollo de este proyecto se ha partido de la investigación previa descrita en [Álvarez-Piñeiro \(2016\)](#), que dio lugar a una aplicación que permitía a un UAV aterrizar sobre una plataforma patentada con formas geométricas características ([Cruz et al., 2013](#)). En la figura 1 se muestra la estructura de esta plataforma. Sobre esta plataforma se han realizado trabajos previos para su identificación ([García-Pulido, 2014](#)); ([García-Pulido et al., 2017](#)).



Figura 1: Plataforma de aterrizaje patentada con estructuras geométricas características.

Sin embargo, gracias a este trabajo se pudo observar el gran problema que supone el procesamiento de imágenes en un entorno real, debido a los reflejos de la luz, las sombras y la aparición de elementos espurios que interfieren en el reconocimiento de la plataforma desde el punto de vista del reconocimiento de imágenes.

Por tanto, el objetivo del trabajo que aquí se presenta consiste en mejorar el índice de resultados obtenidos tras el procesamiento de las imágenes que se reciben, para lo cual se ha diseñado una nueva plataforma con estructuras apropiadas para discernir entre los elementos de color en el espectro visible que caracterizan las figuras geométricas y las distintas reflectancias térmicas que emiten dichos elementos por su naturaleza metálica. La figura 2 muestra una imagen con la plataforma diseñada, sobre la que aparecen las figuras geométricas coloreadas con su característica firma espectral visible sobre sustratos metálicos para su discriminación por diferencias de reflectancia térmica en el espectro infrarrojo.

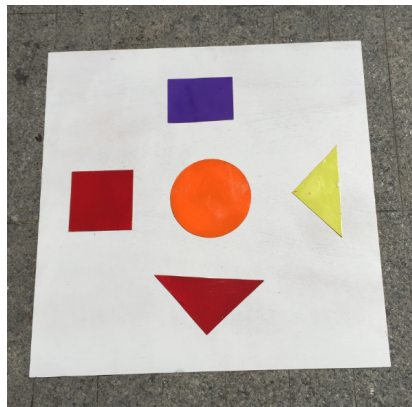


Figura 2: Plataforma de aterrizaje diseñada con figuras geométricas coloreadas y metálicas.

1.2. General considerations

The problem of any dumping in the sea is well known, particularly if it is oil. Proof of this was the crisis caused by the sinking of the *Prestige* in 2002 that affected both Spanish waters and the coast. This disaster caused serious damage to the marine fauna and flora. In this case the discharge was localized, although this is not usually the case with it normally being necessary to search for pollutant discharges, which can also be of any nature, not necessarily oil. Consequently, one of the main problems is the detection of the spill in the sea.

With the purpose of solving this arduous task, the [SALACOM \(2013\)](#) project is born, which proposes the use of a boat and an Unmanned Aerial Vehicle (UAV), and therefore autonomous, for the detection of polluting material. These vehicles are also known as drones. In this work both terms (UAV and drone) are used interchangeably. The function of the UAV is to explore the scene in front of the boat locating areas with polluting residues. Subsequently, once residue is located, it sends the information and positioning corresponding to the control center so that the ship goes to the marked location to act accordingly. However, in order to perform this task, it is necessary to grant the UAV enough autonomy in order to allow it to cover exploration areas as wide as possible.

For this purpose, it is necessary for the UAV to be able to recharge its batteries by itself on a platform located on the surface of the boat. To do this, the UAV will have to be able to recognize it, stand on it and proceed to the corresponding recharge until ready to take off again and continue with the exploration, either doing new searches or reporting on the situation already found.

The present End of Degree project, addresses a part of the research project [SALACOM \(2013\)](#). It is based on developing a computer application whose objective is to detect the platform already mentioned. The procedure has been developed using image processing techniques corresponding to the visible and infrared light spectrum in order for the UAV to locate the platform with pre-set patterns and successfully land on it with proper orientation. This would solve the problem of granting greater autonomy to the UAV.

The development of this project was based on previous research described in [Álvarez-Piñeiro \(2016\)](#), which gave rise to an application that allowed a UAV to land on a patented platform with characteristic geometric forms ([Cruz et al., 2013](#)). Figure 1 shows the structure of this platform. Previous work has been carried out on this platform for its identification ([García-Pulido, 2014](#)); ([García-Pulido et al., 2017](#)).

However, thanks to this work, it was possible to observe the great problem of image processing in a real environment, due to light reflections, shadows and the appearance of spurious elements that interfere in the recognition of the platform from the point of view of image recognition.

Therefore, the objective of the work presented here is to improve the index of results obtained after the processing of the images that are received, for which a new platform has been designed with appropriate structures to discern between the elements of color in the visible spectrum that characterize the geometric figures and the different thermal reflectances that emit these elements by their metallic nature. Figure 2 shows an image with the designed platform, on which the colored geometric figures appear with their characteristic spectral

signature visible on metallic substrates for their discrimination by differences of thermal reflectance in the infrared spectrum.

1.3. Motivación y objetivos

Con el fin de solucionar la falta de precisión de los Global Positioning Systems (GPSs) para dotar de una mayor autonomía a los UAVs, se ha planteado el desarrollo de una aplicación informática basada en el tratamiento de imágenes, que permita a estos vehículos aterrizar en una ubicación específica. Como se ha mencionado previamente, el aterrizaje tiene lugar en una plataforma diseñada con unos patrones predeterminados, cuya principal característica es que es prácticamente imposible encontrar estos patrones fácilmente en la naturaleza, por lo que la hace única en entornos de exterior.

Por lo tanto, el principal objetivo del proyecto es desarrollar una aplicación informática para el reconocimiento de una plataforma mediante el procesamiento de imágenes en el espectro visible e infrarrojo-térmico. Como objetivos específicos se pueden mencionar:

- Desarrollar métodos de tratamiento de imágenes para el reconocimiento de figuras geométricas metálicas basados en el color y la forma en el espectro visible mediante una cámara de color.
- Desarrollar métodos de tratamiento de imágenes para esas mismas figuras metálicas por diferencia de reflectancia térmica, en el espectro infrarrojo-térmico, mediante una cámara de esta naturaleza.
- Desarrollar un método para la toma de decisiones en base a la combinación de la información proporcionada por cada uno de los métodos anteriores.
- Diseñar una interfaz Interfaz Hombre Máquina (IHM) para la gestión de las imágenes y su procesamiento.
- Diseñar una aplicación modular que integre las funcionalidades mencionadas, con posibilidad de incluir nuevas funcionalidades sin necesidad de rediseñar la aplicación.
- Realizar pruebas y validar los avances realizados.

1.4. Motivation and goals

In order to solve the lack of accuracy of the GPSs and so give a greater autonomy to the UAVs, the development of a computer application based on the treatment of images has been proposed, which allows these vehicles to land in a specific location. As previously mentioned, the landing takes place on a platform designed with predetermined patterns whose main feature is that it is practically impossible to find these patterns easily in nature, making it unique in outdoor environments.

Therefore, the main objective of the project is to develop a computer application for the recognition of a platform by processing images in the visible and infrared-thermal spectrum. Specific objectives include:

- Developing methods of treatment of images for the recognition of metallic geometric figures based on the color and the form in the visible spectrum by means of a camera of color.
- Developing methods of treatment of images for those same metallic figures by difference of thermal reflectance, in the infrared-thermal spectrum, by means of a camera of this nature.
- Developing a method for decision making based on the combination of information provided by each of the above methods.
- Designing an Human Machine Interface (HMI) interface for image management and processing.
- Designing a modular application that integrates the mentioned functionalities, with possibility of including new functionalities without needing to redesign the application.
- Performing tests and validating the progress made.

1.5. Plan de trabajo

Para cumplir con los objetivos expuestos, se propone el siguiente plan de trabajo, desglosado por objetivos y duración de los mismos teniendo en cuenta la duración aproximada de ocho meses del proyecto, que hace un total de 32 semanas.

Tabla 1: Plan de trabajo.

Objetivos	Duración (semanas)
Desarrollar métodos de tratamiento de imágenes en color	8
Desarrollar métodos de tratamiento de imágenes infrarrojas	8
Desarrollar un método para la toma de decisiones en base a ambas imágenes	4
Diseñar una interfaz IHM	3
Implementar los métodos de tratamiento y decisión en una aplicación final	2
Realizar pruebas y validar resultados	2 (una vez finalizado el desarrollo de la aplicación)
Redactar la memoria	5 (distribuidas a lo largo del desarrollo del proyecto)

1.6. Workplan

To meet the stated objectives, the following work plan is proposed which is broken down into objectives and duration, taking into account the approximate 8-month duration of the project, which makes a total of 32 weeks.

Table 2: Workplan.

Objectives	Duration (weeks)
Developing color image processing methods	8
Developing methods of treatment of infrared images	8
Developing a method for making decisions based on both images	4
Designing an HMI interface	3
Designing an application that includes the features already mentioned	2
Performing tests and validating results	2 (once the development of the application is finished)
Writing the memory	5 (distributed throughout the development of the project)

1.7. Estructura de la memoria

La presente memoria se ha estructurado en siete secciones y dos anexos, en los que se detallan las diferentes partes del proyecto. Además, se ha incluido al inicio un listado de palabras clave, una lista de abreviaturas y han sido recogidas todas las imágenes y tablas que aparecen en la memoria.

En esta primera sección se ofrece la idea general sobre el proyecto planteado. También se incluye la motivación que ha servido de base para su desarrollo, los principales objetivos que se pretenden alcanzar y se ha detallado el plan de trabajo que se ha seguido. Todos ellos tanto en castellano como en inglés.

En la segunda sección se realiza una revisión sobre el estado del arte en el que se analizan las aplicaciones existentes y que han existido para resolver el problema de aterrizaje de vehículos no tripulados sobre estructuras o plataformas reconocibles. En resumen, se describe una breve historia de la evolución tecnológica en busca de la solución en función de las propuestas existentes y en su caso identificando los problemas que proponen para plantear la propuesta que aquí se formula.

En la tercera sección se detalla el concepto en el que se ha basado el proyecto, la visión artificial. Además, se exponen de forma genérica las técnicas y métodos empleados en el procesamiento de imágenes con el fin de reconocer de la plataforma.

En la sección cuatro se plantea el diseño del proyecto haciendo uso de las diferentes metodologías que proporciona la ingeniería del software. Se ha detallado la planificación seguida por los miembros del proyecto, la descripción del equipamiento hardware y software

que ha sido necesario, los casos de uso basados en la metodología Unified Modeling Language (UML) que describe la estructura del proyecto, el análisis y desarrollo de la solución que se ha planteado y por último el plan de gestión de riesgos.

En la quinta sección se describe el lenguaje elegido para la implementación del proyecto, recopilando las principales funciones proporcionadas por dicho lenguaje. También se detalla la arquitectura de la aplicación y las metodologías software aplicadas.

En la sexta sección se analizan los resultados obtenidos mediante los desarrollos realizados, tanto en la combinación de imágenes térmicas y a color, como en cada una de éstas por separado.

En la séptima y última sección, se describen las conclusiones obtenidas del análisis de los resultados mencionados anteriormente, tanto en castellano como en inglés. A continuación, se incluyen las futuras líneas de investigación que se podrían llevar a cabo en base a estos resultados y las contribuciones personales de cada componente.

Una vez terminadas las secciones, se detallan las fuentes bibliográficas consultadas durante el desarrollo del proyecto.

Finalmente, se han incluido dos anexos. En el primero de ellos se han descrito las principales pruebas de campo que se han realizado y en el segundo se ha detallado un manual de uso de la aplicación.

2. Estado del arte

En esta sección se presentan las líneas actuales de investigación en relación al tema a tratar, detección de plataformas para el aterrizaje de un UAV.

El uso de UAVs en espacios abiertos requiere de una interacción en tiempo real con el propio entorno. Los UAVs, a pesar de llevar ya tiempo en uso, comparten las mismas debilidades que otros dispositivos electrónicos. Las dos más importantes en relación a lo que al planteamiento de este estudio compete son la fragilidad y la baja autonomía, por lo que es necesario proporcionarle la capacidad para solventar esta situación.

El funcionamiento de un UAV comprende un desplazamiento desde un punto de inicio a otro final. El control del vuelo, no exento de sus correspondientes dificultades, no constituye el objetivo del presente proyecto. En cualquier caso, existe un punto clave en que el UAV requiere de una asistencia especial. Durante la realización de una misión, el UAV necesitará descender en un punto concreto de su recorrido, garantizando que este descenso se realice de forma que no afecte a su fragilidad y para proceder a la recarga de la batería por su limitada duración que afecta principalmente a su autonomía. Para ello es necesario que reconozca la zona donde debe realizar la operación de aterrizaje orientándose convenientemente de alguna manera. A pesar de que los sistemas de geolocalización son efectivos, será imprescindible una orientación visual en distancias cortas, ya que la precisión de los mismos a veces no es suficiente para garantizar un aterrizaje seguro en el momento requerido antes de agotar la capacidad de autonomía.

Tras el estudio de las técnicas existentes se ha comprobado que la mayor parte de las aplicaciones se realizan mediante el uso de cámaras operando en el espectro visible. Sin embargo, y como aporte novedoso en el planteamiento de la propuesta que se formula, se pensó que quizás se podría mejorar la capacidad de reconocimiento de plataformas con la combinación de cámaras en el espectro visible e infrarrojo, convenientemente combinadas para garantizar el mayor éxito posible.

Por este motivo, en la realización de este proyecto se ha trabajado con la capacidad de reconocimiento visual a través de las imágenes proporcionadas por ambos tipos de cámaras. Se ha simulado la toma de fotografías con un UAV mediante el uso de una cámara propia de un smartphone y de un dispositivo conectado, compuesto por una cámara térmica instalada en el smartphone. Para reconocer la plataforma se han utilizado figuras geométricas con colores difíciles de encontrar en la naturaleza. Además, y para potenciar el uso de las cámaras térmicas, las figuras son estructuras metálicas. De esa manera la diferencia de intensidad en la escala de grises de las piezas metálicas y del fondo serán lo suficientemente importantes como para ayudar en la detección de éstas por diferencias de reflectividad térmica en el espectro infrarrojo. Se ha utilizado como objetivo de reconocimiento una batería de imágenes tomadas por las cámaras comentadas. De esta forma se simula la captura de imágenes por parte de un UAV en tiempo real y se realiza su posterior estudio. Todo ello haciendo uso de la tecnología de procesamiento de imágenes, como se ha mencionado previamente.

El procesamiento de imágenes busca mejorar o resaltar aspectos de la imagen de tal modo que se pueda conseguir una imagen óptima, ya sea bien para su análisis o por el simple hecho de obtener una imagen nítida. En el caso de este proyecto, se ha comenzado por un tratamiento inicial de la imagen para que, mediante las imágenes modificadas y mejoradas en calidad, se pueda reconocer el contenido de las mismas de una forma lo más eficazmente posible.

En la actualidad existen dos formas claramente identificadas de obtener una imagen, de forma analógica o digital. De igual modo, también existen dos formas de procesarlas, aunque en este proyecto se busca centrar la atención en el procesamiento digital de imágenes y sus múltiples aplicaciones.

Se ha contemplado el gran abanico de posibilidades que proporciona esta tecnología en el reconocimiento de plataformas de aterrizaje. Durante el estudio se ha podido constatar el gran número de proyectos que existen en la actualidad relacionados con este campo.

Actualmente resulta bien conocido el programa Amazon Prime Air ([Amazon, 2015](#)). Según los datos que proporciona la compañía [Amazon \(2015\)](#), empresa líder en logística, gracias a este programa es posible realizar envíos mediante dron a clientes en cualquier parte del mundo en un tiempo récord, ahorrando así costes de mensajería y permitiendo entregar más paquetes. Para ello es necesario disponer de una plataforma de aterrizaje que identifica la zona de entrega sobre la que debe posarse el dron, tal y como muestra la figura 3.



Figura 3: Plataforma de aterrizaje de drones para entregas de Amazon.

Para el aterrizaje del UAV, [Amazon \(2015\)](#) solicita al usuario que imprima una hoja o una cartulina conteniendo el carácter “a” con el fin de que constituya el elemento a identificar por el UAV desde el aire, ubicado en una zona de aterrizaje para realizar la entrega. Estas plataformas, se sitúan sobre entornos cuya identificación resulta fácil, algo que no ocurre en el caso del proyecto que sustenta este trabajo.

En [Álvarez-Piñero \(2016\)](#) se plantea un método de identificación de la plataforma descrita en la figura 1 mediante la utilización del método de [Viola and Jones \(2004\)](#) mediante su aplicación con la tecnología [OpenCV \(2017\)](#). Se fundamenta en la famosa técnica desarrollada para reconocimiento facial. Por un lado, la plataforma es la indicada, que presenta problemas de identificación en exteriores cuando se producen reflejos debidos a la incidencia de los rayos solares. Además, la mencionada técnica requiere de procesos de aprendizaje automático para el reconocimiento de la plataforma.

Por otro lado, el Ministerio de Defensa español trabaja en colaboración con la empresa [Siport21 \(2016\)](#) en un programa para tareas de búsqueda en condiciones climatológicas complicadas. La función de reconocimiento de plataformas proporcionaría al UAV una autonomía mayor en la realización de sus tareas. Se trata de conseguir el aterrizaje de drones en barcos, y por tanto es un planteamiento similar al que se plantea en el proyecto que se expone, si bien cuenta con un planteamiento de control avanzado para hacer frente a determinadas situaciones geográficas y atmosféricas, en ocasiones adversas, algo que excede el ámbito del presente proyecto.

La plataforma que se muestra en la figura 4, diseñada por [Lange et al. \(2008\)](#) está formada por círculos de color blanco y negro de forma alterna. La idea de esta plataforma es que cada círculo concéntrico posee un ancho distinto, creciente conforme la cámara del UAV enfoca a los círculos más externos. Esto permite conocer varias características de la imagen y de la localización del UAV en el momento de la captura de la imagen.



Figura 4: Plataforma de aterrizaje de Lange.

El trabajo propuesto por [Gil Fernández \(2016\)](#) plantea el reconocimiento utilizando una plataforma tipo formada por un fondo blanco con tres cuadrados azules, tal y como se muestra en la figura 5. Mediante una interfaz web y un entorno de desarrollo compuesto por la herramienta [Matlab \(2015\)](#), el objetivo del proyecto es permitir un descenso con la localización y orientación correcta. En este caso las figuras son del mismo color y forma aunque de distinto tamaño.



Figura 5: Plataforma de aterrizaje de Lange.

En [Saripalli et al. \(2002\)](#) se realizó un estudio para el reconocimiento de una plataforma con una figura geométrica impresa en forma de H en su superficie, como se puede observar en la figura 6. El reconocimiento de esta plataforma se basa en la búsqueda del centro geométrico de la figura. Además, por la forma de la figura geométrica, el UAV es capaz de reconocer qué rotación sería la adecuada para aterrizar. Para el reconocimiento de esta plataforma se utiliza la técnica de etiquetado de componentes conexas en la imagen ([Pajares and Cruz, 2007a](#)); ([Pajares and Cruz, 2007b](#)).



Figura 6: Plataforma de aterrizaje de Saripalli.

En [Sharp et al. \(2001\)](#) se realizó un estudio para una aplicación basada en reconocimiento de imágenes cuyo objetivo era el reconocimiento de la plataforma mostrada en la figura 7. Ésta está compuesta por una base blanca con un cuadrado negro en su interior y dentro de él seis cuadrados blancos, uno de ellos de mayor tamaño que los otros. Además de técnicas basadas en la mejora de la calidad de la imagen se utilizó la búsqueda de puntos de interés ([Pajares and Cruz, 2007a](#)); ([Pajares and Cruz, 2007b](#)). En este caso, la problemática que plantea esta plataforma es similar al de la plataforma de la figura 1, esto es, la aparición o desaparición de determinadas figuras por la existencia de reflejos producidos por el sol.

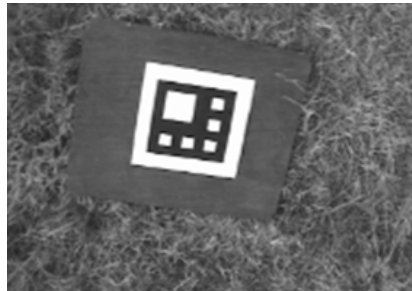


Figura 7: Plataforma de aterrizaje de Sharp.

Por otro lado, existen algunos casos de trabajos realizados para el reconocimiento de plataformas que no se basan en el reconocimiento de figuras geométricas. A pesar de ser una idea con cierta semejanza a las comentadas, ya que se busca comparar una imagen de lo que se busca con una imagen de lo que se visualiza, es diferente ya que no se busca que el contenido de una figura sea igual que el contenido de la otra. Lo que se pretende es identificar qué puntos del terreno de la imagen guardada, que tengan características reconocibles, son similares a otros puntos encontrados en la imagen que se acaba de tomar para el aterrizaje.

El algoritmo desarrollado por [Cesetti et al. \(2010\)](#) se basa en el análisis de una imagen tomada por el UAV, como la mostrada en la figura 8. Este algoritmo pretende buscar puntos de referencia dentro de la imagen. De esa manera y estableciendo semejanzas entre lo que se distingue en la imagen y una serie de valores previamente reconocidos, trata de determinar el lugar exacto de aterrizaje.

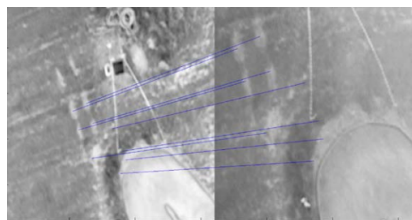


Figura 8: Imagen analizada por el algoritmo de Cesetti.

Como conclusión, se puede comprobar que en su inmensa mayoría los proyectos realizados basados en el reconocimiento visual de plataformas de aterrizaje de UAV han sido implementados mediante imágenes tomadas por cámaras de espectro visible. Este es un aspecto preocupante en el reconocimiento de plataformas debido a las condiciones adversas en

entornos de exterior, donde factores adversos tales como la incidencia de la luz del sol o las sombras, pueden provocar que en ocasiones no se consiga reconocer alguna figura, ya que el brillo puede distorsionar los niveles de intensidad de la imagen. Por otra parte, las técnicas descritas previamente no tienen en cuenta el color de las figuras integradas en las plataformas.

Por ello es necesario contemplar la necesidad de mejora de los resultados añadiendo el uso de imágenes térmicas, que complementan a las imágenes visibles, siendo éstas además en color, lo que permite incrementar el grado de fiabilidad en el reconocimiento de este tipo de plataformas. De esta forma, siempre que se necesite realizar un reconocimiento de la plataforma se dispone de las dos imágenes simultáneamente, que se procesan por separado y cuyos resultados, en términos de aciertos y errores se combinan convenientemente para determinar si realmente se trata o no de la plataforma.

3. Métodos de vision artificial

En esta sección se describen los conceptos básicos, métodos y principales técnicas de visión artificial que se han utilizado durante el desarrollo del proyecto.

La visión artificial, también conocida como visión por computador, es una disciplina científica, perteneciente al campo de la inteligencia artificial, enfocada en el procesamiento, tratamiento y análisis de imágenes mediante la aplicación de diferentes métodos. Su principal objetivo es interpretar de forma digital y automática el contenido de dichas imágenes en conveniencia con la situación a la que estén expuestas para, finalmente, resolver un determinado problema utilizando la información obtenida de su análisis. Dichas imágenes pueden ser capturadas desde una o varias cámaras móviles o fijas, tanto monocromáticas como en color, lo que proporciona multitud de opciones a la hora de interpretar una situación determinada.

En los últimos años, el uso de la visión por computador ha ido en aumento, expandiéndose por sectores tan diferenciados como la agricultura (automatización de la maquinaria), la industria (desarrollo de la calidad y automatización industrial), la medicina (como ayuda al diagnóstico), la detección facial y de objetos o la restauración de imágenes, entre muchas otras.

3.1. Procesamiento de imágenes

Un sistema de procesamiento de imágenes está compuesto por varios subsistemas cuya finalidad es interpretar alguna característica subyacente en la imagen bajo análisis. El presente proyecto se centra en el procesamiento a bajo nivel, es decir, desde el suavizado, eliminación del ruido y definición de bordes (preprocesamiento) hasta la extracción de características, todo ello con el objetivo de detectar los diferentes objetos que aparecen en la imagen para su posterior identificación atendiendo a su forma y color.

3.1.1. Viola y Jones

Viola and Jones (2004) es uno de los principales métodos utilizados en el reconocimiento facial y de objetos en imágenes. Este algoritmo permite realizar un reconocimiento de objetos en tiempo real. El algoritmo reúne píxeles dentro de zonas rectangulares para definir los posibles objetos dentro de la imagen haciendo uso de tres fases que se describen a continuación.

La primera es la utilización de máscaras Haar. En esta fase se trabaja con cuatro tipos de formas rectangulares básicas como las mostradas en la figura 9, gracias a las cuales puede distinguir la diferencia de los valores en escala de grises de los píxeles comprendidos en zonas rectangulares, así como qué tipos de objetos son cada uno de los que hay en la imagen. Estas zonas o figuras se llaman Características Haar o Haar Features.

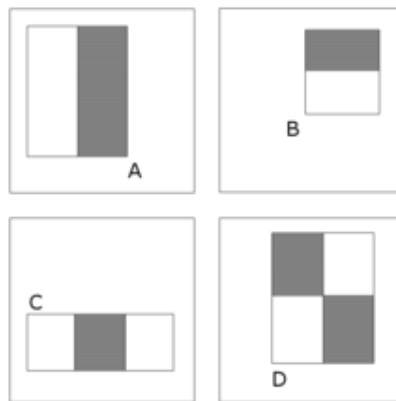


Figura 9: Ejemplo de formas rectangulares básicas.

Por ejemplo en la figura 9, en reconocimiento facial la figura A es utilizada para el reconocimiento de los ojos y la figura C para el reconocimiento de la nariz. En esta fase se elige el tipo de máscara que se va a usar.

La segunda es la generación de una imagen integral, que es la matriz de valores calculada mediante la suma de los valores de los puntos adyacentes a cada uno. El cálculo de esta matriz de valores ha sido incluido en la presente memoria como ecuación (1).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

La tercera es la utilización del algoritmo de aprendizaje automático (Schapire, 2013). Dentro de este algoritmo hay dos fases principales: la primera se basa en el aprendizaje del algoritmo a través de una muestra y del clasificador de Haar elegido. Esto irá generando una sucesión de casos negativos y positivos. Mientras que en la segunda se utiliza un árbol de decisión, figura 10, que recorre la imagen actual utilizando la lógica aprendida en el paso anterior y, de esa manera, detectar si las regiones de la imagen pertenecen a un tipo de figura u otro. El árbol de decisión permite partir de una solución más grande hasta soluciones más pequeñas y de esa manera se optimiza la resolución del problema.

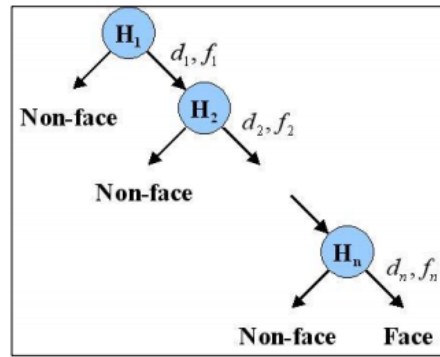


Figura 10: Árbol de decisión.

3.1.2. Etapas del procesamiento

Además del método descrito en el apartado anterior, existen infinidad de métodos para el procesamiento de imágenes. La mayoría de ellos, entre los cuales se encuentra el presente proyecto, siguen una serie de etapas, que se pueden visualizar en la figura 11 y que se han descrito brevemente a continuación. Además, desde la perspectiva que engloba el proyecto a desarrollar, se detallan de forma teórica los métodos y técnicas de procesamiento de imágenes que han sido empleados en cada una de estas fases.

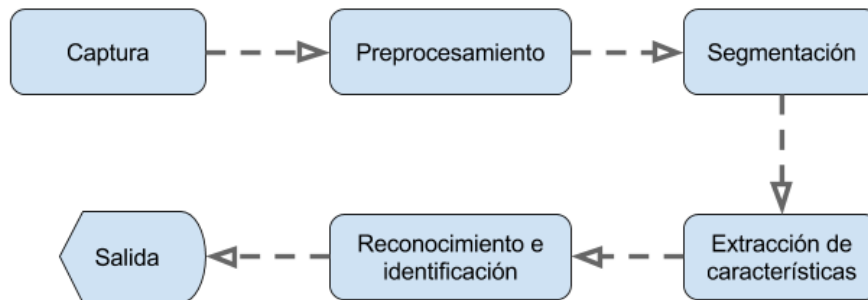


Figura 11: Principales etapas del procesamiento de imágenes.

(a) Captura o adquisición de la imagen

Es una etapa determinante del procesamiento de imágenes debido a que es necesario obtener las imágenes teniendo en cuenta factores tales como la iluminación, el tipo de cámara utilizada o las propiedades de los objetos para no dificultar su posterior procesamiento. En este caso se han utilizado en la etapa de captura de imágenes la cámara perteneciente a un smartphone, iPhone 6 (2014), y una cámara térmica acoplada a otro smartphone, Motorola Moto 4G (2016). Las especificaciones de ambas cámaras han sido detalladas en la sección 4.4.1 de la presente memoria.

(b) Preprocesamiento y segmentación

Una vez capturada la imagen, es necesario digitalizarla para proceder a su posterior

procesamiento. Esto se consigue aplicando un proceso de muestreo, cuantificación y codificación de la imagen capturada, que proporciona una matriz bidimensional de dimensiones M, N .

En el presente proyecto se ha hecho uso de cuatro tipos de imágenes digitales, obtenidas mediante el tratamiento de cada uno de sus píxeles obteniendo imágenes transformadas según las necesidades.

- Red, Green, Blue (RGB): matriz compuesta por tres canales: rojo, verde y azul, cuyo fundamento se basa en los receptores de color existentes en la retina humana.
- Hue, Saturation, Value (HSV): transformación no lineal del modelo RGB en la que cada color viene determinado por su tonalidad, saturación y brillo. Representa la gama cromática más simplificada, de forma similar a la que perciben los humanos.
- Escala de grises: matriz compuesta por valores de intensidad en un rango de valores, donde no existe información de color.
- Binarias: matriz compuesta únicamente por ceros y unos, que corresponden a los dos estados posibles de un píxel (verdadero y falso).

Considerando que la imagen ya ha sido digitalizada es necesario mejorar su calidad por medio de la aplicación de técnicas tales como reducción del ruido, eliminación de sombras, mejora del contraste, eliminación de elementos no deseados o extracción de los bordes de las figuras. De esta forma se pretende identificar en la imagen las zonas de interés en cada momento.

Seguidamente se describen de forma teórica las técnicas que se han aplicado para la implementación de dicha etapa.

Método de Bradley

El método de [Bradley \(1986\)](#) es el método que se ha utilizado para binarizar las imágenes, ya que tiene como objetivo clasificar los píxeles de una imagen como blancos o negros. Esta clasificación se basa en el uso de un valor umbral, que es un valor numérico que permite distinguir de forma eficaz qué píxeles tomarán el valor negro y cuáles el valor blanco al binarizar la imagen, en función de si el valor del píxel es mayor o menor que dicho umbral. Este valor puede ser obtenido de forma automática como resultado de la aplicación del método de [Otsu \(1979\)](#) que se detalla a continuación o manualmente fijando un valor de umbral por defecto, como se muestra en la figura 12. Aunque en la literatura también se ha utilizado el método de [Sauvola and Pietikäinen \(2000\)](#), el método de [Bradley \(1986\)](#) es más rápido y permite realizar el procesado de imágenes en tiempo real. Dicha binarización es muy útil a la hora de separar los objetos del resto de la imagen.

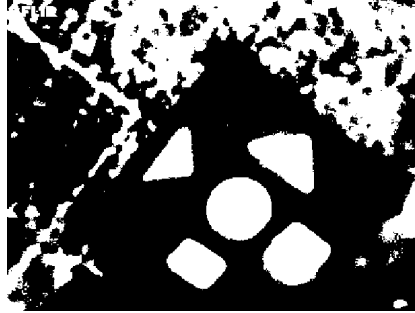


Figura 12: Binarización de una imagen con un valor umbral por defecto.

Método de Otsu

El principal objetivo de este método es obtener un valor de umbral de forma automática cuya finalidad es destacar los diferentes objetos que aparecen dentro de una imagen. Como ya se ha mencionado, es comúnmente utilizado como complemento de la función encargada de la binarización de una imagen (Bradley, 1986), debido a que la aplicación de este valor umbral en el proceso de binarización, como puede verse en la figura 13, permite conseguir mejores resultados en lo que a la sustracción de elementos no deseados se refiere.

Este valor se obtiene analizando el histograma de una imagen en gris de suerte que separa los valores de intensidad en dos clases. El umbral obtenido es tal que maximiza la separación entre las clases. Una vez obtenido el umbral, se determina el valor de cada píxel en función de su intensidad y según que el valor de ésta se sitúe por debajo o por encima de dicho umbral.



Figura 13: Binarización de una imagen con el método de Otsu.

Detección de bordes

Se considera detección de bordes a toda técnica que permita distinguir un píxel de una imagen en función de la variación de intensidad entre él mismo y sus vecinos. De esta forma, los píxeles cuya intensidad difiere del resto serán considerados bordes, siendo descartados los restantes. La mayoría de las técnicas utilizan el cálculo de un operador de la derivada, el gradiente, proporcionado por la ecuación (2), para obtener dichos bordes con mayor o menor nivel de detalle. Dicho gradiente obtendrá un valor distinto de cero en zonas con variaciones de intensidad.

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x) \quad (2)$$

La implementación de los operadores de la primera derivada se realiza, entre otros, mediante los conocidos operadores de Sobel y Prewitt (Pajares and Cruz, 2007a). Éstos utilizan el gradiente de los píxeles de una imagen para calcular de qué manera puede cambiar la intensidad de cada píxel dentro de la imagen que se esté tratando. Mediante la utilización de una máscara sobre las dos aproximaciones, horizontal y vertical, de la derivada sobre la imagen de intensidad. Estas mascarar de Sobel y Prewitt corresponden a las ecuaciones (3) y (4) respectivamente.

$$\frac{1}{4} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \frac{1}{4} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (3)$$

$$\frac{1}{3} \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \quad \frac{1}{3} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (4)$$

Laplacian of Gaussian (LoG)

Se trata de un extractor de borde basado en el concepto segunda derivada (Pajares and Cruz, 2007a), La expresión matemática que lo sustenta es la proporcionada en la ecuación (5).

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (5)$$

Operador de Canny

Este operador utiliza dos fases para mejorar la detección de bordes. Primero realiza una reducción del ruido y seguidamente reconoce la intensidad del gradiente de la imagen. Este algoritmo utiliza un filtrado paso bajo de tipo gaussiano debido a su eficiencia frente al ruido, para obtener una imagen borrosa. Después aplica cuatro filtros en las direcciones horizontal, vertical y diagonal sobre la imagen resultante borrosa.

A diferencia de otros operadores, es este operador el que se ha aplicado en el desarrollo del proyecto, debido a que los resultados obtenidos por su aplicación se adecuaban más a los deseados. La figura 14 muestra el resultado de aplicar el extractor de bordes basado en el operador de Canny (1986).

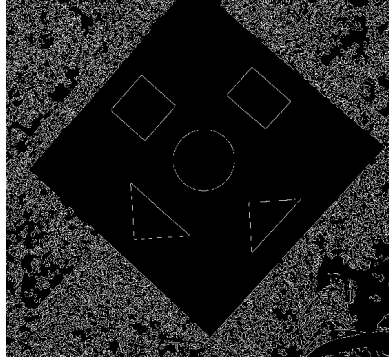


Figura 14: Extracción de bordes mediante Canny.

Rellenado de regiones y huecos

La imagen resultante de la binarización contiene diferentes regiones. Una región se identifica como un conjunto de píxeles conectados, de tal manera que desde un píxel se puede llegar a otro de la región sin abandonar ésta. Estas regiones pueden coincidir con las buscadas o no. En caso de que no coincidan, se trata de elementos inútiles carentes de interés. Las regiones de interés pueden, en cualquier caso, encontrarse con pequeñas zonas o píxeles que necesitan ser eliminados. Las figuras 15 y 16 muestra una imagen binarizada cuyos píxeles en blanco representan regiones. Obsérvese en el caso de la figura 15 cómo las figuras de la plataforma aparecen suficientemente distinguibles y sin zonas que deban ser eliminadas. Sin embargo, en la figura 16 existen diferentes zonas conteniendo pequeñas regiones o píxeles aislados, siendo conveniente su eliminación.

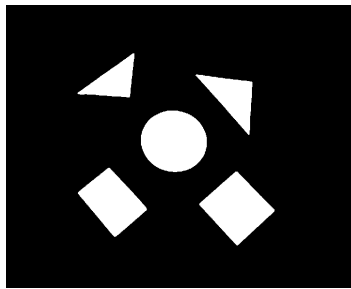


Figura 15: Rellenado de regiones y huecos.

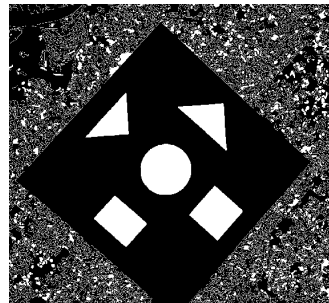


Figura 16: Rellenado de regiones y huecos con otros elementos.

Operaciones morfológicas

Las operaciones morfológicas son operaciones matemáticas basadas en la teoría de conjuntos, que se aplican a nivel de píxel sobre imágenes en escala de grises o binarias, utilizando un elemento estructural. Este elemento estructural es un operador que se proporciona a dichas operaciones para limpiar la imagen. Estas operaciones permiten mejorar la calidad de las regiones obtenidas a la vez que se eliminan partes superfluas sobre las mismas, o como se ha mencionado previamente se eliminan pequeñas regiones o se rellenan huecos, todo ello derivado de la insuficiente calidad durante el proceso de

binarización. En la figura 17 se puede observar la evolución que ha sufrido la imagen después de aplicar operaciones morfológicas a la figura 16.

- **Apertura morfológica**

Dicha operación consiste en la aplicación de un proceso de dilatación a la imagen, que debe haber sido previamente erosionada. El objetivo de este operador es el de suavizar los bordes de las figuras dentro de una imagen.

- **Cierre morfológico**

Consiste en rellenar los huecos de menor tamaño e intentar unir los píxeles de los bordes dentro de una figura con el objetivo de definir una forma concreta.

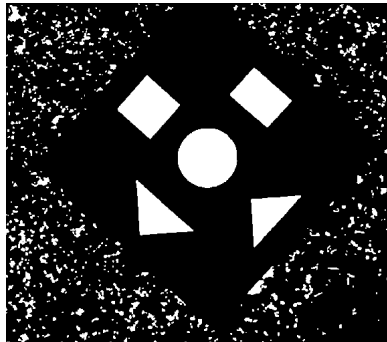


Figura 17: Aplicación de operaciones morfológicas de apertura y cierre.

Exclusión de objetos

Su principal objetivo consiste en descartar y eliminar las componentes conexas no deseadas de una imagen binaria. Estas componentes se identifican por contener un número mínimo de píxeles que en ningún caso representan las regiones de interés. De esta forma es posible acotar aún más las posibilidades de localizar elementos deseados en dicha imagen, en el caso que ocupa las regiones sobre la plataforma. La figura 18 muestra el resultado de aplicar a la figura 17 un procedimiento de eliminación de objetos aplicando el criterio descrito previamente.

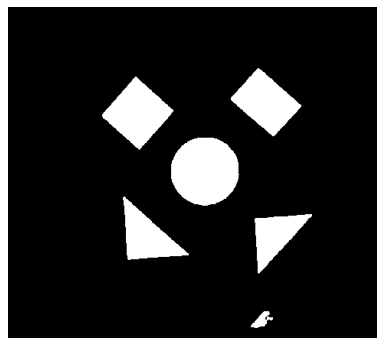


Figura 18: Aplicación de exclusión de objetos.

(c) **Extracción de características**

Una vez se ha obtenido una imagen limpia, en la que solo aparecen las zonas supuestamente deseadas, se procede a la extracción de las propiedades de los objetos que la conforman. Entre estas características destacan el color y algunas propiedades morfológicas descriptivas como son la longitud de los ejes mayor y menor, área o centroide, entre otros. Ha sido necesario aplicar las siguientes técnicas para lograr la extracción de propiedades de las zonas deseadas.

Algoritmo de Moore-Neighbor

El algoritmo de Moore-Neighbor (Moore, 2015) se basa en el hecho simple de que para cada píxel dentro de una imagen bidimensional existen otros ocho píxeles vecinos, conocidos por los ocho vecinos o los vecinos indirectos. Consiste en la obtención del contorno de las figuras que forman una imagen por medio de esos ocho píxeles. El número de celdas vecinas de un píxel puede variar dependiendo de cuál sea el rango, r , lo que se obtiene mediante la ecuación (6).

$$(2r + 1)^2 - 1 \tag{6}$$

Dicho algoritmo realiza una búsqueda de los píxeles vecinos de cada uno de los píxeles en una región determinada. Esto es, dado un píxel binario blanco, supuestamente perteneciente a una determinada región, el método busca píxeles vecinos que son también de la misma naturaleza, es decir blancos.

Existen criterios de parada que permiten que el algoritmo finalice cuando una serie de condiciones se cumplan. El criterio de parada aplicado en el presente proyecto es el criterio definido en Eliosoff (Moore, 2015), que propone que el algoritmo termine en el momento en el que se pase dos veces por el píxel inicial de la misma forma.

Extracción de propiedades

Este método ofrece información acerca de las propiedades de las regiones que se encuentran dentro de la imagen. Suelen ser útiles diversas propiedades. Las utilizadas en el presente trabajo han sido el área, centroide y los ejes mayor y menor.

(d) **Reconocimiento e identificación**

Es en esta etapa en la que se definen los procedimientos encargados de etiquetar cada región en función de sus características. Se basan en la utilización de determinados valores que se obtienen de forma empírica. Estos valores se utilizan para determinar la forma y color de las regiones bajo procesamiento, en este último caso sólo si la imagen es la de color en el espectro visible. En función de las propiedades de la región extraídas, como son la longitud de los lados, el área o centro, se procede a determinar el tipo de figura que se está intentando identificar.

(e) **Salida**

El resultado del proceso es el reconocimiento de la plataforma por la identificación de un número suficiente de las figuras geométricas contenidas en ella. El reconocimiento se aplica tanto sobre las imágenes en color como sobre las térmicas. Para cada imagen, se indica el número de figuras geométricas identificadas sobre la plataforma y mediante una toma de decisión en conjunto se determina si realmente se trata o no de la plataforma.

4. Desarrollo del proyecto

Esta sección presenta las características del proyecto. Primero se detalla el trabajo de investigación previo al desarrollo del mismo, se describe el proceso de trabajo seguido y se proporciona la estructura general de la aplicación. A continuación se realiza una descripción del equipamiento hardware y software necesario, se incluye un diagrama con una visión general de la aplicación, se detalla el análisis, estudio y desarrollo de la solución propuesta y, finalmente, se proporciona un plan de gestión de riesgos.

4.1. Trabajo previo de investigación

Tras las primeras investigaciones sobre la temática acerca de la cual se desarrollaría este proyecto, se eligieron dos potenciales lenguajes de programación posibles con los que trabajar, [Python \(2017\)](#) y [Matlab \(2015\)](#). Finalmente, se tomó la decisión de utilizar [Matlab \(2015\)](#) debido a que se trata de una herramienta que proporciona abundantes funciones para el procesamiento de imágenes.

Una vez decidido el lenguaje de programación, se procedió a la búsqueda de métodos de procesamiento de imágenes aplicables al proyecto, es decir, métodos capaces de reconocer figuras y colores en una imagen. En primera instancia se optó por utilizar el método de [Viola and Jones \(2004\)](#), a menudo utilizado para la detección de patrones (como puede ser el reconocimiento facial), pero tras varias semanas de trabajo con este método y no logrando avances, se decidió su abandono. Fue entonces cuando se tomó la decisión de probar con una combinación de diferentes métodos, entre los que se encuentra el método de Moore-Neighbor ([Moore, 2015](#)) utilizado en la detección de bordes en imágenes, con el fin de lograr los resultados esperados.

Las primeras pruebas realizadas con el objetivo de comprobar la utilidad de los métodos escogidos para realizar el reconocimiento de figuras y colores fueron aplicadas a imágenes digitales generadas mediante la herramienta Paint de Windows. Una vez impresas, se aplicaron diferentes métodos de detección de figuras geométricas en estas imágenes, tomadas con diferentes niveles de luz y brillo. Aunque los resultados de estas pruebas no fueron del todo exitosos, se pudo determinar que éste era el camino correcto.

Tras las conclusiones extraídas de esta investigación previa, el siguiente paso fue pasar a las pruebas de campo, expuestas en el Anexo I del presente documento, con el objetivo de adaptar los métodos utilizados para el procesamiento a las distintas condiciones en las que las imágenes fuesen capturadas.

4.2. Proceso de trabajo

Se describe a continuación el proceso de trabajo que se ha llevado a cabo durante el desarrollo del proyecto, incluyendo la evolución que han seguido las plataformas de reconocimiento de aterrizaje debido a la necesidad de superar las deficiencias de sus precedentes, algunas de las cuales se introdujeron en la sección 2. Todas estas plataformas se han recopilado en el Anexo I Pruebas de campo.

Fase 1: Análisis del proyecto y toma de decisiones

En esta primera fase se determinaron y establecieron los objetivos y el alcance del proyecto, también se tomaron decisiones sobre el lenguaje a utilizar, se obtuvo una idea general sobre un posible primer diseño de la plataforma de aterrizaje, la forma de obtener las imágenes tanto en el espectro visible e infrarrojo, y se esclarecieron las vías de comunicación y organización a seguir durante el proyecto, como metodologías de trabajo en grupo.

Fase 2: Reconocimiento de figuras geométricas (espectro visible)

Esta fase se centró en el reconocimiento de figuras geométricas en el espectro visible y por tanto utilizando imágenes RGB. Para ello, se realizó una primera prueba de campo en un entorno de interior sobre cartón, “Interior”, con la que se identificaron los primeros problemas causados por los reflejos y las sombras. Como consecuencia de estos primeros problemas se decidió que el fondo de la plataforma y las figuras tenían que tener colores muy diferentes con el fin de destacar las figuras sobre ella, por lo que se diseñó la prueba de campo “Exterior-1”. Con ambas plataformas se capturaron imágenes con distintos niveles de luces y sombras con el objetivo de reconocer las figuras geométricas en diferentes situaciones.

Fase 3: Reconocimiento de figuras geométricas (espectro infrarrojo-térmico)

Durante esta fase, y con un avanzado nivel de reconocimiento sobre las imágenes en el espectro visible, se procedió al reconocimiento de figuras geométricas en el espectro de luz infrarroja. Con el fin de mitigar los problemas que aparecían al aplicar el reconocimiento de figuras en el espectro infrarrojo sobre la plataforma anterior, se decidió realizar una nueva prueba de campo, “Exterior-2”, que contaba con una superficie de mayor tamaño compuesta por un material termoaislante y en la que la mayoría de las figuras eran del mismo material y tamaño. De esta forma las figuras no estaban tan próximas entre sí, solucionando así los problemas derivados de diferencias de temperatura entre éstas y la plataforma. Con dicha prueba de campo se capturaron varias imágenes en el espectro infrarrojo que fueron tratadas con la intención de reconocer, de esta otra forma, las figuras geométricas.

Fase 4: Reconocimiento de colores

Una vez completadas las fases anteriores, se prosiguió con el reconocimiento del último filtro para localizar la plataforma, el color. Para finalizar las fases de reconocimiento, se realizó una última prueba de campo, “Exterior-3”, compuesta por una nueva plataforma blanca de madera y cinco figuras metálicas todas del mismo tamaño. La decisión de elegir el fondo blanco fue debido a que este color no absorbe el calor del sol con facilidad, ventaja en las imágenes infrarrojas, y destaca las figuras sobre el fondo, ventaja en las imágenes a color. Se volvieron a capturar imágenes con diferentes grados de exposición al sol, provocando reflejos y sombras, con el propósito de simular un entorno de pruebas real en el que se identificaran tanto formas como colores.

Fase 5: Diseño de la interfaz

En esta fase se diseñó una interfaz gráfica sencilla e intuitiva, con el objetivo de mostrar los resultados obtenidos en las fases anteriores. De esta forma, se puede comprobar en qué casos el reconocimiento era satisfactorio y en cuales desfavorable, con el fin de mejorar el porcentaje de acierto de estos últimos.

Fase 6: Algoritmo de decisión, integración y pruebas

En esta última fase se desarrolló un algoritmo capaz de tomar decisiones en base al grado de reconocimiento de las figuras en el espectro visible e infrarrojo. Además, se procedió a la integración de las diferentes partes del proyecto: reconocimiento de figuras y colores, algoritmo de decisión e interfaz. Para terminar, se seleccionó una batería de imágenes con el objetivo final de simular el reconocimiento de la plataforma en distintas situaciones y calcular los porcentajes de acierto y error de los métodos de reconocimiento propuestos.

4.3. Estructura de la aplicación

La aplicación se ha dividido en dos partes, por un lado, la interfaz gráfica que permite realizar un control sobre la misma en tiempo real y por otro la capa responsable de reconocer las imágenes. Esta disposición de la aplicación permite que el procesamiento de imágenes sea independiente de la interfaz. Cabe destacar dentro de la estructura planteada que, debido a esta división en dos partes bien separadas y a la vez integradas, es posible realizar mantenimiento y desarrollo de las dos partes prácticamente como si fueran aplicaciones completamente distintas.

Mediante la estructura otorgada a la aplicación sería posible, en un futuro, implantar un repositorio central que aloje todas las imágenes y datos capturados durante la ejecución de la aplicación. La aplicación se alimentaría de las imágenes alojadas, en tiempo real, a través de la base de datos y la capa responsable de reconocer las imágenes recogería éstas, las procesaría y enviaría las indicaciones necesarias al UAV. Así, tanto imágenes como decisiones tomadas serían almacenadas en este sistema central.

4.4. Descripción del equipamiento

El proyecto que se describe en la presente memoria ha sido desarrollado mediante equipos portátiles en el entorno de programación ofrecido por [Matlab \(2015\)](#). Se ha implementado una aplicación software, tal como se describe en la figura 22, que recibe como parámetros de entrada dos imágenes, una visible y una infrarroja, siendo posteriormente procesadas, devolviendo una respuesta afirmativa o negativa. Para ello ha sido necesario capturar dos imágenes mediante sendas cámaras operando en el espectro visible e infrarrojo.

4.4.1. Equipamiento hardware

A continuación, se describen los elementos hardware con los que se ha trabajado para el desarrollo del proyecto.

4.4.1.1 Cámara térmica

Se ha utilizado una cámara dual, capaz de tomar imágenes en el espectro de luz visible e infrarrojo, únicamente mediante su versión Android. En concreto, se ha trabajado con el modelo de cámara [FLIRONE \(2017\)](#) cuyas principales especificaciones y características técnicas se describen en la tabla 3. La figura 19 muestra el dispositivo [FLIRONE \(2017\)](#) junto al dispositivo de captura, en este caso un Motorola Moto 4G Plus (2017). Conviene reseñar en este sentido que el dispositivo [FLIRONE \(2017\)](#) está equipado con dos cámaras, una operando en el espectro visible, si bien monocroma y otra infrarroja, tal como se indica a continuación. Ambas imágenes se fusionan para marcar sobre la infrarroja características identificativas de la escena. Si bien, de cara al desarrollo del proyecto, la funcionalidad de la cámara monocroma visible fue anulada con el fin de disponer exclusivamente de la imagen infrarroja de interés. La figura 20 muestra una imagen captada con el dispositivo infrarrojo.



Figura 19: Smartphone Moto 4G Plus junto al dispositivo FLIR ONE.

Tabla 3: Especificaciones técnicas del dispositivo FLIR ONE.

FLIR ONE 2017	
Rango de captación de temperatura	-20°C - 120°C
Capacidad de la batería	350mA-h
Cámara visible	VGA
Precisión de temperatura	Rango de temperatura de $\pm 0,1^\circ\text{C}$

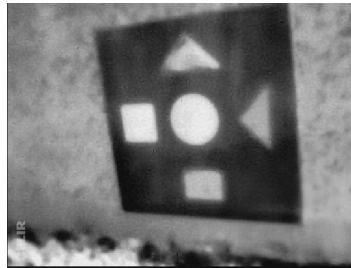


Figura 20: Imagen infrarroja captada con la cámara del dispositivo FLIR ONE de esta naturaleza.

4.4.1.2 Cámaras de color en smartphones

Además de una cámara térmica, también se ha hecho uso de la cámara integrada en uno de los dispositivos móviles (smartphones) de los miembros del proyecto para la captura de imágenes en el espectro visible. La figura 21 muestra una imagen captada con éste dispositivo.

Más específicamente, se ha utilizado la cámara de un iPhone 6 (2014), cuyas especificaciones técnicas se detallan en la tabla 4.

Tabla 4: Especificaciones técnicas de la cámara del Smartphone iPhone 6.

iPhone 6	
Tipo	Cámara iSight de 8 megapíxeles con píxeles de $1,5\mu$
Enfoque	Automático con Focus Pixels
Apertura	$f/2,2$
Estabilización	Automática



Figura 21: Imagen de color captada con un iPhone 6.

4.4.1.3 Computador

Para el desarrollo y ejecución de la aplicación se ha hecho uso de un equipo sobremesa y dos portátiles, a saber: iMac 27-inch (2011), Mac Air (2013) y Asus G551JK (2015). Sus características técnicas se describen en las tablas 5, 6, 7.

Tabla 5: Especificaciones técnicas del procesador iMac 2011.

iMac 2011	
Procesador	Intel Core i5 - 2,7GHz
Memoria RAM	16 GB DDR3 - 1333MHz
Tarjeta gráfica	AMD Radeon HD 6770M 512 MB

Tabla 6: Especificaciones técnicas del procesador Mac Air 2013.

Mac Air 2013	
Procesador	Intel Core i5 - 1,3GHz
Memoria RAM	4 GB DDR3 - 1600MHz
Tarjeta gráfica	Intel HD Graphics 5000 - 1536 MB

Tabla 7: Especificaciones técnicas del procesador G551JK.

Asus G551JK	
Procesador	Intel Core i7 - 2,5/3,5 GHz
Memoria RAM	8GB DD3 - 1600MHz
Tarjeta gráfica	NVIDIA GeForce GTX 850M 4GB

4.4.2. Equipamiento software

A continuación se especifican también los requisitos software necesarios para la realización del proyecto.

Matlab R2015b

Se requiere, al menos, uno de los siguientes compiladores dependiendo del sistema operativo:

- (a) Windows (64 bits)
 - MinGW 4.9.2 C/C++ (Distributor: TDM-GCC)
 - Microsoft Visual C++ 2015 Professional 4
 - Microsoft Visual C++ 2013 Professional 4
 - Microsoft Visual C++ 2012 Professional 4
 - Microsoft Visual C++ 2010 Professional SP1 4
 - Microsoft Windows SDK 7.1 con .NET Framework 4.0.6
 - Microsoft Visual C++ 2008 Professional SP1

- Windows SDK 6.1 1 2 4
- Intel C++ Composer XE 2013 3
- Intel C++ Composer XE 2011 1 3
- Intel Visual Fortran Composer XE 2013 3
- Intel Visual Fortran Composer XE 2011 1 3
- Microsoft .NET Framework SDK 2.0, 3.0, 3.5, 4.0, 4.5

(b) Mac OS

- Xcode 7.x (Review required setup)
- Xcode 5.1+ or 6.x
- Intel Fortran Composer XE (2013)

FLIR ONE App

Aplicación necesaria para la captura de imágenes mediante la cámara térmica mencionada acoplada al Smartphone Motorola Moto 4G Plus (2016).

Android

Requerido sistema operativo Android 4.4.2 o superior. Además, para utilizar la cámara [FLIRONE \(2017\)](#) son necesarias funciones como GPS, localización, On-The-Go (OTG) USB, linterna, micrófono, giroscopio y acelerómetro. Además de permitir a la aplicación el uso de dichas funciones.

4.5. Casos de uso

En la figura 22 se representa el diagrama de funcionamiento de la aplicación. En él aparecen todas las clases, descritas en las secciones 5.3 y 5.4.1 y los pasos que se siguen durante el funcionamiento de la aplicación.

1. Se realiza la carga de forma individual o de ambas imágenes (térmica y de color) pulsando en el botón “Cargar” o “Comenzar Simular” respectivamente.
2. La interfaz (GUI) avisa a la clase Controlador que recoge las imágenes de la ruta establecida y muestra las imágenes correspondientes en la interfaz. Tanto la clase GUI como el Controlador en este caso se refieren a los módulos relativos al diseño software en el que se basa la aplicación, tal y como se describe en la sección 5.4.1.
3. Mediante el botón “Procesar” se avisa al Controlador de qué se va a proceder a un procesamiento individual o en conjunto de las imágenes cargadas.
4. Se procede a la limpieza de la imagen térmica que será devuelta tras su tratamiento para eliminar ruido y elementos irrelevantes.
5. La clase LimpiarT devuelve la imagen térmica tras haber realizado su tratamiento.
6. En este caso se procede a la limpieza de la imagen normal que será devuelta tras su tratamiento para eliminar ruido y elementos irrelevantes.

7. La clase LimpiarC devuelve la imagen visible (color) tras haber realizado su tratamiento.
8. Se envían las dos imágenes tratadas a la clase Procesar.
9. La clase Procesar analiza las imágenes que se han introducido y devuelve un conjunto de figuras y colores en caso de la imagen en el espectro visible y sólo de figuras en caso de la imagen térmica.
10. La clase Controlador avisa a la clase Decisión que determina con este conjunto de figuras y colores el grado de concordancia entre las imágenes introducidas y la plataforma que se está tratando de identificar.
11. El porcentaje final de acierto es devuelto al Controlador.
12. La clase Controlador muestra el porcentaje de acierto de cada imagen y el total en el cuadro de información de la clase GUI.

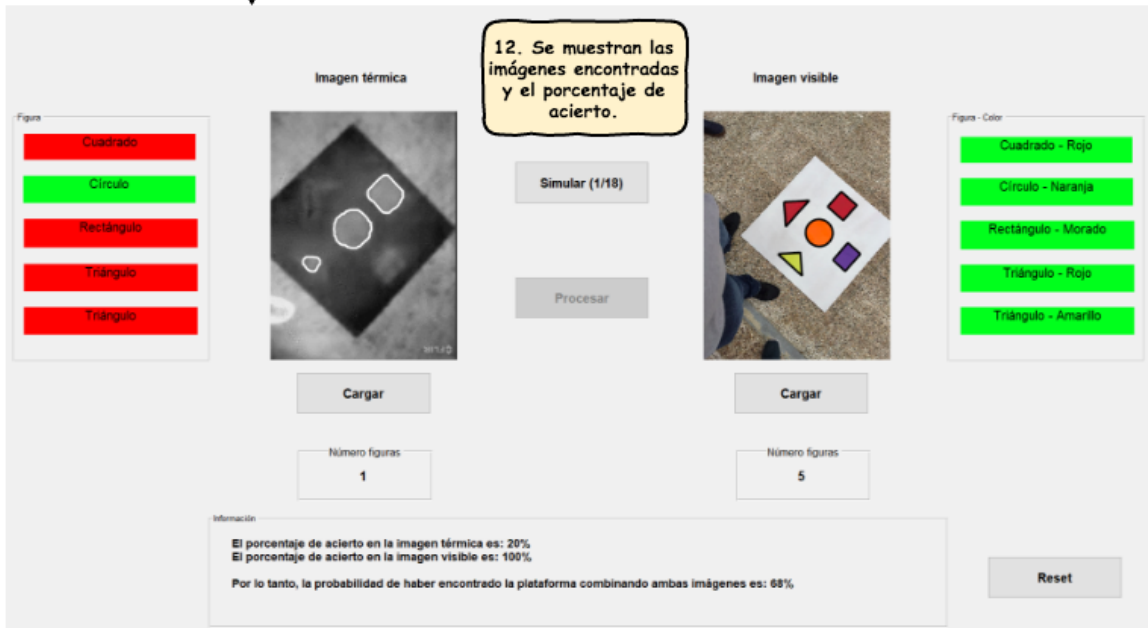
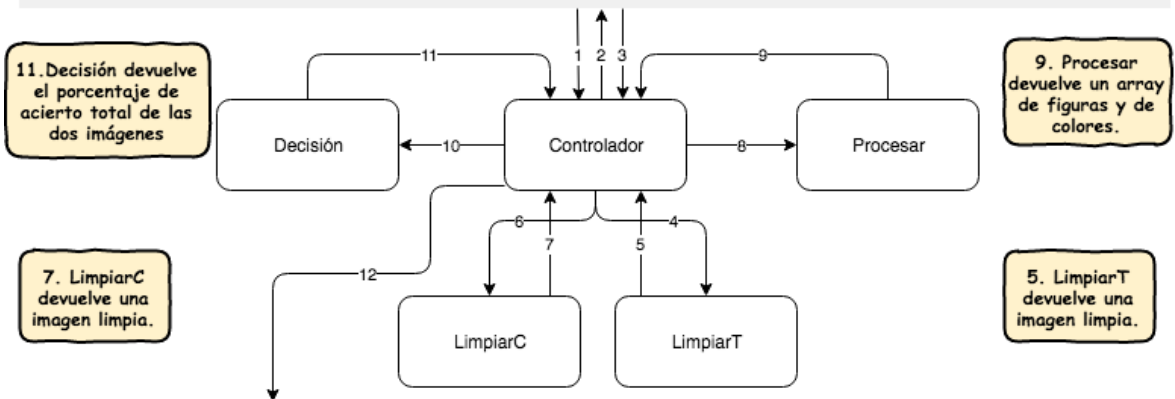
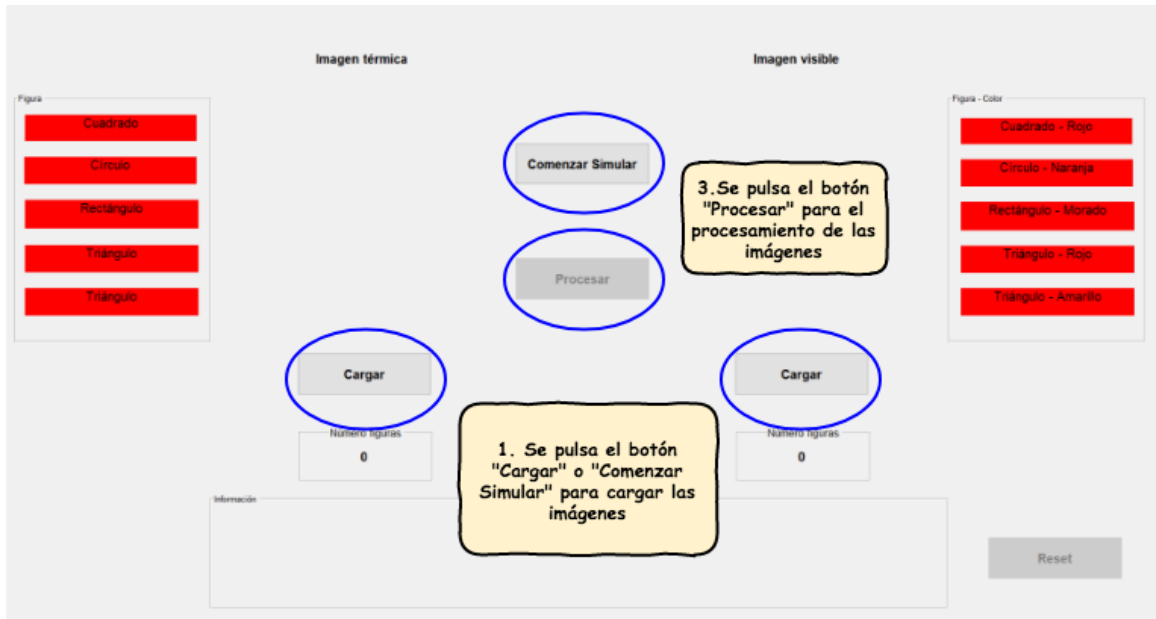


Figura 22: Diagrama de funcionamiento de la aplicación.

4.5.1. Limpieza imagen espectro infrarrojo

La tabla 8 corresponde a los pasos cuatro y cinco del diagrama general de la aplicación descrito previamente según la figura 22. Este caso de uso describe el tratamiento que se aplica a una imagen tomada por una cámara operando en el espectro infrarrojo.

Tabla 8: Caso de uso número uno. Limpieza de imagen en espectro infrarrojo.

ID	UC-1
Título	Limpieza de imagen en espectro infrarrojo.
Descripción	Se realiza el tratamiento de la imagen infrarroja con el fin de obtener otra imagen con las figuras bien diferenciadas en su interior.
Actor principal	Sistema reconocedor de imágenes.
Precondiciones	Existe una imagen infrarroja a reconocer.
Postcondiciones	Imagen limpia y preparada para su análisis.
Escenario principal	<ol style="list-style-type: none">1. Umbralización de la imagen.2. Binarización de la imagen.3. Apertura morfológica.4. Cierre morfológico.
Frecuencia de uso	Alta.
Datos de entrada	Imagen tomada con cámara de luz infrarroja.
Datos de salida	Imagen limpia y lista para analizar.

4.5.2. Limpieza imagen espectro visible

La tabla 9 corresponde a los pasos seis y siete del diagrama de la aplicación descrito previamente según la figura 22. Este caso de uso describe el tratamiento que se aplica a una imagen tomada por una cámara de espectro visible.

Tabla 9: Caso de uso número dos. Limpieza de imagen en espectro visible.

ID	UC-2
Título	Limpieza de imagen en espectro visible.
Descripción	Se realiza el tratamiento de la imagen RGB con el fin de obtener otra imagen con las figuras bien diferenciadas en su interior.
Actor principal	Sistema reconocedor de imágenes.
Precondiciones	Existe una imagen RGB a reconocer.
Postcondiciones	Imagen limpia y preparada para su análisis.
Escenario principal	<ol style="list-style-type: none"> 1. Conversión de la imagen RGB a HSV. 2. Umbralización para aislar las formas. 3. Limpieza de objetos pequeños. 4. Obtención de bordes por Canny (1986) mediante Otsu (1979). 5. Rellenado de huecos. 6. Procesamiento morfológico para limpieza de imperfecciones de la imagen. 7. Limpieza de objetos algo mayores.
Frecuencia de uso	Alta.
Datos de entrada	Imagen tomada con cámara de luz visible.
Datos de salida	Imagen limpia y lista para analizar.

4.5.3. Procesamiento de la imagen

La tabla 10 corresponde a los pasos ocho y nueve del diagrama de la aplicación descrito previamente según la figura 22. Este caso de uso describe la identificación que se realiza de los objetos que pertenecen a la imagen.

Tabla 10: Caso de uso número tres. Procesamiento de la imagen.

ID	UC-3
Título	Procesamiento de la imagen.
Descripción	Se procesa la imagen con el fin de obtener la lista de figuras y colores que aparecen en ella.
Actor principal	Sistema reconocedor de imágenes.
Precondiciones	Existe una imagen limpia.
Postcondiciones	Lista de figuras que se han reconocido.
Escenario principal	<ol style="list-style-type: none">1. Obtención del mapa de píxeles y array de bordes a procesar.2. Obtención de las propiedades de las regiones deseadas.3. Identificación de las figuras.
Frecuencia de uso	Alta.
Datos de entrada	Imagen original, imagen tratada y tipo de imagen.
Datos de salida	Lista de figuras reconocidas y un array de bordes de las figuras.

4.5.4. Decisión de la validez de la imagen

La tabla 11 corresponde a los pasos diez y once del diagrama de la aplicación descrito previamente según la figura 22. Este caso de uso describe el proceso utilizado para determinar de qué manera la imagen coincide con la plataforma deseada.

Tabla 11: Caso de uso número cuatro. Decisión de la validez de la imagen.

ID	UC-4
Título	Procesamiento de la imagen.
Descripción	Se determina si el objeto que existe dentro de la imagen recibida es candidato de ser la plataforma deseada.
Actor principal	Sistema reconocedor de imágenes.
Precondiciones	Existen al menos una imagen a identificar previamente tratada.
Postcondiciones	Lista de figuras encontradas y porcentaje de éxito.
Escenario principal	<ol style="list-style-type: none"> 1. ¿Hay imagen térmica? <ol style="list-style-type: none"> a) Sí. Se va al punto 2. b) No. Se va al punto 4. 2. Coincidencia de tipos de figuras en imagen térmica. 3. Generación de porcentaje de acierto cámara de espectro infrarrojo. 4. ¿Hay imagen de espectro visible? <ol style="list-style-type: none"> a) Sí. Se va al punto 5. b) No. Se va al punto 7. 5. Coincidencia de tipos de figuras en imagen de espectro visible. 6. Generación de porcentaje de acierto en cámara de espectro visible. 7. Generación de porcentaje de acierto con la combinación de los anteriores.
Frecuencia de uso	Alta.
Datos de entrada	<p>Array de figuras y colores contenidas en una imagen tomada por una cámara de luz visible.</p> <p>Array de figuras contenidas en una imagen tomada por una cámara de luz infrarroja.</p>
Datos de salida	Lista de figuras encontradas e información acerca del éxito en la búsqueda de la plataforma.

En la figura 23 se muestra a modo de ejemplo el diagrama de actividad relativo al caso de uso número 4, tabla 11. No se han incluido en el documento los diagramas de actividad de los demás casos de uso debido a que su estructura es lineal, sin contemplar bifurcación alguna y por tanto no aporta información adicional relevante.

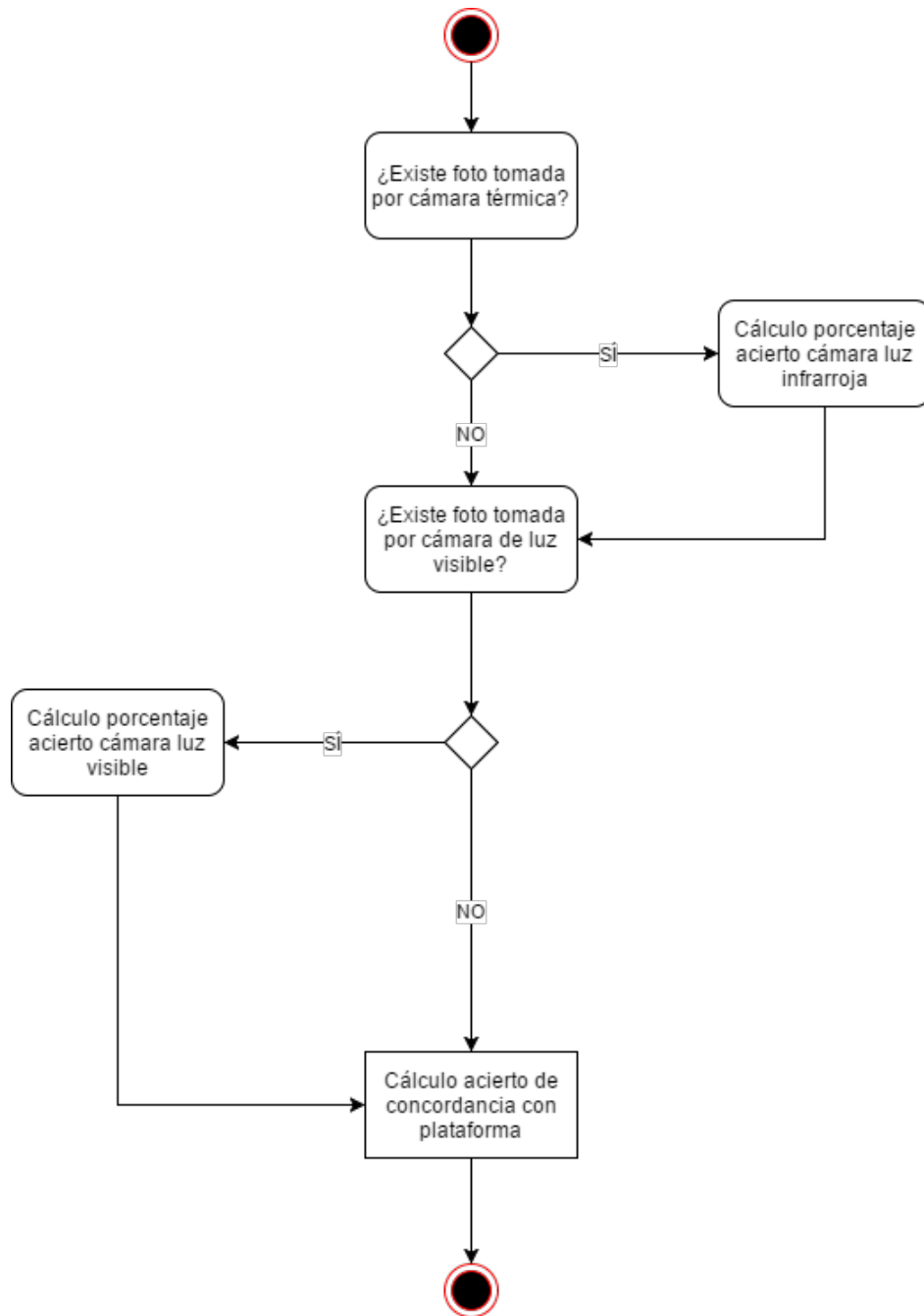


Figura 23: Diagrama de actividad del caso de uso número 4.

4.6. Análisis y desarrollo de la solución propuesta

Partiendo del equipamiento descrito en la sección 4.4.1 y con el objetivo de reconocer la plataforma identificando las figuras que aparecen en ella, ha sido necesario aplicar una secuencia de métodos que constituyen el procedimiento de limpieza de la imagen y un posterior procesamiento de las regiones obtenidas por medio de este tratamiento. Como se ha mencionado previamente, para realizar este reconocimiento, se ha decidido combinar el uso de dos tipos de imágenes, imágenes infrarrojas en el espectro térmico e imágenes de color en el espectro visible. Por último, se ha empleado un algoritmo de decisión que, utilizando los resultados obtenidos de ambas imágenes, se determina si el objeto que se visualiza coincide con la plataforma deseada. Dicha secuencia de métodos aplicados a las imágenes descritas, corresponden a la base del proyecto que se expone.

En las figuras 24 y 25 se muestran dos imágenes de la plataforma a reconocer, la primera capturada con la cámara térmica con las especificaciones descritas en la tabla 3 y la segunda en color capturada con un iPhone 6 (2014), con las especificaciones descritas en la tabla 4. Este par de imágenes sirven de base para comprender y visualizar la evolución del proceso relativo al tratamiento de imágenes hasta llegar al reconocimiento de la plataforma.

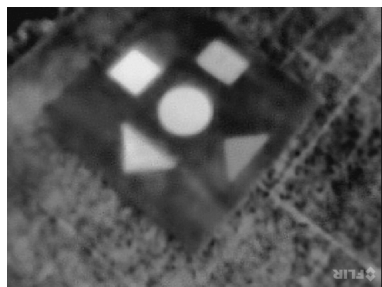


Figura 24: Imagen infrarroja de la plataforma.



Figura 25: Imagen en color de la plataforma.

Una vez las imágenes han sido capturadas con diferentes niveles luminosidad y conteniendo diferentes niveles de sombras, es el momento de proceder a su tratamiento. Este tratamiento es específico en función del tipo de cámara utilizada, debido a la distinta naturaleza entre ambas imágenes. No obstante, la combinación de los resultados del reconocimiento constituye un elemento decisivo para determinar si lo que aparece en la imagen es o no la plataforma. Este hecho ha influido de forma determinante a la hora de aplicar diferentes técnicas de tratamiento en función del tipo de imagen, con el objetivo de intentar obtener los mejores resultados de cada una de ellas. Por este motivo se ha decidido dividir en dos partes tanto el análisis como la descripción de dicho tratamiento.

4.6.1. Tratamiento de imágenes térmicas

Este proceso consiste en la minimización del ruido subyacente en la imagen y la eliminación de aquellas regiones espurias que dificultan el reconocimiento de las figuras geométricas de interés conforman la plataforma.

Las imágenes térmicas son en escala de grises, capturadas mediante la cámara térmica del

dispositivo [FLIRONE \(2017\)](#) descrito previamente. De esta forma, gracias a las diferencias de temperatura existente entre las figuras y el resto de la imagen, sus correspondientes reflectancias son también diferentes, las cuales son captadas por el dispositivo. Este hecho representa el principio de partida para su posterior tratamiento.

El proceso empleado en el tratamiento de la imagen en escala de grises se basa exactamente en la aplicación de técnicas de binarización a partir de un valor umbral seguido de la aplicación de operaciones morfológicas cuyo principal objetivo es eliminar el ruido subyacente para conseguir una imagen más consistente.

En primer lugar, se recibe una imagen, que necesariamente estará en escala de grises. Posteriormente, a esta imagen se le aplica el método de [Otsu \(1979\)](#), descrito en la sección [3.1.2](#) y cuyo objetivo es la separación de los elementos que componen la imagen con el objetivo de encontrar el valor umbral apropiado para realizar la binarización. De esta forma y gracias al valor de umbral obtenido de forma automática, es posible realizar la pretendida binarización de una forma fiable y eficiente. Sin embargo, la validez de esta binarización puede verse comprometida en función de la luminosidad exterior, que influye en la reflectividad de las figuras que forman la plataforma. En efecto, debido a que la variación de intensidad puede no ser suficiente para generar una alta reflectividad, podrían no ser distinguibles frente al resto, lo que podría causar que las figuras afectadas por este fenómeno puedan dejar de percibirse como estructura significativa de la imagen. Las figuras [26](#) y [27](#) muestran claramente los efectos derivados de las situaciones indicadas.

Como se puede observar, existe una notable diferencia entre ambas figuras. La figura [26](#) fue tomada un día nublado, mientras que la figura [27](#) fue tomada un día soleado. Se puede apreciar cómo en la figura [27](#) las figuras se distinguen a la perfección porque el contraste de su temperatura con respecto al resto de la imagen es mucho mayor, incluida la base de la plataforma. Sin embargo, en la figura [26](#) la mayoría de las figuras no se aprecian debido a que no han conducido la temperatura lo suficiente para que la cámara las detecte.

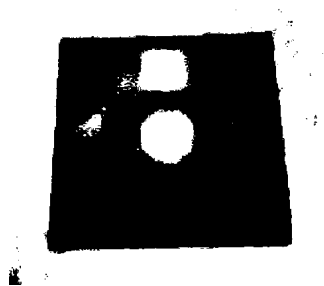


Figura 26: Binarización de una imagen en un día nublado.

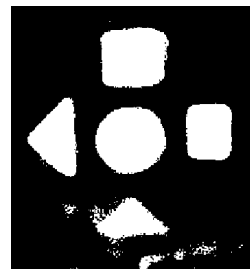


Figura 27: Binarización de una imagen en un día soleado.

Una vez la imagen se ha binarizado, se aplican las siguientes operaciones morfológicas, cuyo objetivo es minimizar el ruido que ha permanecido tras esta binarización. Las operaciones morfológicas aplicadas son la apertura y cierre morfológico. De esta forma, el posterior procesamiento sobre las regiones más relevantes de la imagen resulta más fácil y eficiente. Estas operaciones morfológicas, descritas en la sección [3.1.2](#), requieren un elemento estructural que indica el nivel de precisión con el que se va a modelar cada región. Sin embargo, cuanto mayor sea el radio de dicho elemento más tenderán a asemejarse las figuras a éste, pudiendo

así perderse la forma original de alguna de ellas. Por ello, se ha decidido establecer un elemento estructural en forma de disco y cuyo radio sea el menor posible antes de que la figura pierda su forma, pero consiguiendo a la vez suavizar dichas figuras para facilitar su posterior reconocimiento. Se ha considerado que el radio de este elemento estructural sea 4 píxeles.

Como se ha mencionado previamente, se ha hecho uso de las operaciones de apertura y cierre morfológico debido a que la combinación de ambas, aplicando el elemento estructural indicado, permite suavizar los picos de las figuras y completar los píxeles vacíos que éstas puedan contener. Así, quedarán eliminados los píxeles que no formen parte de la región que se desea reconocer y el contorno e interior de las figuras serán más consistentes, lo que facilitará la posterior obtención de las características de las regiones de interés.

Por ejemplo, se puede observar cómo en la figura 28 el triángulo de la izquierda ha intentado asemejar su forma a la del elemento estructural indicado, disco de radio doce, por lo que la figura ha deteriorado su forma original y será mucho más complicado reconocerla. Por otro lado, en la figura 29 el elemento estructural aplicado es el que ha sido utilizado en la aplicación, disco de radio cuatro, por lo que se puede apreciar como la mayoría de las regiones mantienen su forma original e incluso quedan mejor definidas para su reconocimiento.

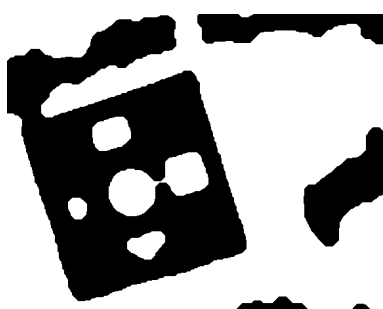


Figura 28: Aplicaciones morfológicas cuyo elemento estructural es un disco de radio doce.



Figura 29: Aplicaciones morfológicas cuyo elemento estructural es un disco de radio cuatro.

4.6.2. Tratamiento de imágenes en color

Si la imagen es capturada por una cámara visible se le aplica un tratamiento distinto al aplicado sobre la imagen térmica. Esto es así, debido a que una imagen RGB puede contener sombras que dificulten el reconocimiento de las zonas de interés. Por ello, si se conoce el tipo de imagen, pueden aplicarse técnicas que minimicen, si fuera necesario, las sombras de la imagen, considerando que éstas afectan considerablemente a las componentes espectrales R, G, B en la imagen de color.

En este proceso de síntesis se han utilizado diferentes técnicas, concretamente las que se enumeran a continuación: conversión de imágenes del modelo de color RGB a HSV para extraer la imagen de intensidad (V), la obtención del valor umbral automático mediante [Otsu \(1979\)](#) aplicado al reconocimiento de bordes, la exclusión de objetos, el relleno de regiones y huecos o el uso de varias operaciones morfológicas. Todas ellas con el objetivo principal de minimizar los errores del posterior procesamiento de las regiones deseadas.

Durante el inicio de esta fase se recibe una imagen capturada por la cámara visible con las explicaciones indicadas en la tabla 4. Seguidamente, se realiza la transformación del modelo de color RGB al modelo HSV, lo que permite la extracción de información acerca del matiz, la saturación y el brillo de la imagen. Sobre la imagen V, se obtiene un mapa de pixeles basado en los valores de la saturación con el objetivo de intentar disminuir el nivel de sombras de la imagen. Una vez realizada la conversión y extracción de la matriz de valores de saturación, se procede a binarizar la imagen, consiguiendo finalmente que la imagen resultante se encuentre sin sombras.

Observando las figuras 30 y 31 es posible comprobar que la conversión realizada entre estos modelos de color permite desprestigiar la totalidad de las sombras de la imagen sin alterar de ninguna manera el contenido principal de la misma.



Figura 30: Imagen de color con sombras sobre la plataforma.

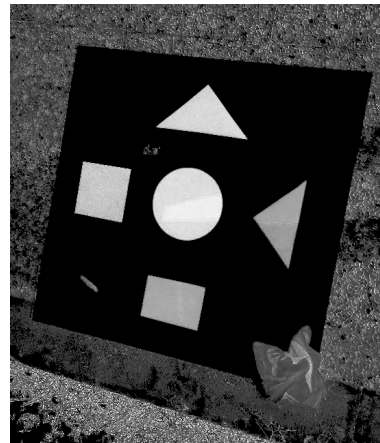


Figura 31: Imagen una vez se ha realizado la conversión entre el modelo RGB y HSV.

Con el problema de las sombras resuelto, se procede a realizar la obtención de los bordes de las regiones de interés dentro de la imagen. Para ello se ha aplicado el método de Otsu (1979) y el operador de Canny (1986), cuya combinación ha ofrecido resultados satisfactorios. Con ello se pretende desechar todo el contenido de la imagen que no sea semejante a una de las figuras que conforman la plataforma, poniendo énfasis en el realzado de los bordes de las mismas. Después se procede a rellenar las zonas comprendidas entre los bordes encontrados para diferenciar de un modo aún más claro las figuras del resto de la imagen.

En la figura 32 se muestra la detección de bordes obtenida al aplicar el método de Otsu (1979) en combinación con el operador de Canny (1986), mientras que en la figura 33 se puede observar el rellenado de las zonas comprendidas entre los bordes previamente detectados.

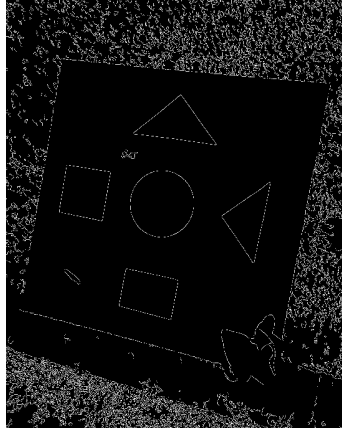


Figura 32: Aplicación del método de Otsu en combinación con el operador Canny.

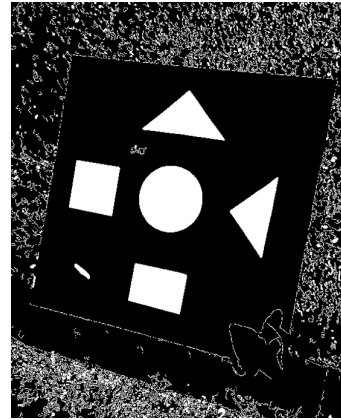


Figura 33: Rellenado de regiones obtenidas por el método anterior.

Debido al ruido que se puede apreciar tras la ejecución de los métodos anteriores, es necesario aplicar una sucesión de operaciones morfológicas que sean capaces de eliminarlo. Estas operaciones, como se ha descrito al inicio de la sección, precisan un elemento estructural que indica la forma que adoptarán las figuras una vez aplicadas dichas operaciones. En primer lugar, se ha utilizado la apertura morfológica, que permite la disminución de los picos de las figuras de la imagen, y seguidamente el cierre morfológico, que resta importancia a los huecos dispersos dentro de la imagen y los descarta. Este procedimiento permite refinar la forma de las figuras, con el propósito de poder obtener de forma mucho más fiable las regiones de interés.

Aun habiendo aplicado operaciones morfológicas, es posible observar, en la figura 34, cómo en la imagen aún permanecen ciertos elementos que podrían impedir el correcto reconocimiento de la plataforma. Dichos elementos poseen un tamaño lo suficientemente grande como para inducir al algoritmo de reconocimiento a creer que puedan tratarse de alguna de las figuras de la plataforma.

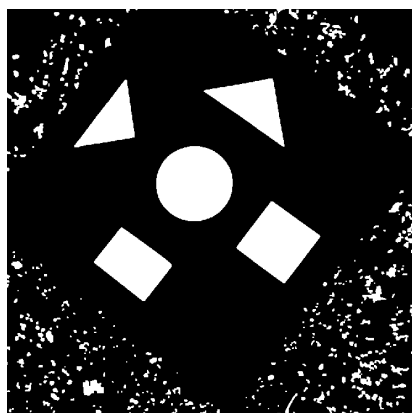


Figura 34: Aplicación de operaciones morfológicas para suavizar la imagen.

Por este motivo y como última técnica aplicada a este tratamiento, se procede a eliminar de forma selectiva aquellos elementos que superen un tamaño determinado en cuanto a número de píxeles que los definen. En el caso de la aplicación se ha tomado como valor mínimo para las regiones de interés 1200 píxeles. La aplicación de esta técnica puede observarse en la figura 35.

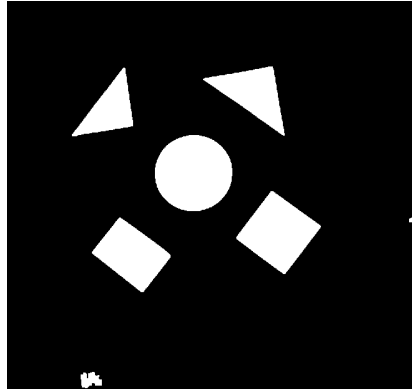


Figura 35: Eliminación de elementos en función de su tamaño.

Como conclusión después de haber descrito el tratamiento aplicado a las imágenes anteriores, utilizados en función del tipo de imagen capturada, cabe destacar que dichos métodos permiten que la imagen obtenida esté preparada para el procesamiento de las regiones de interés que serán, en mayor o menor medida, las figuras necesarias para el reconocimiento de la plataforma.

4.6.3. Procesamiento de las regiones de interés

Como ya se ha expuesto previamente, mediante el tratamiento de imágenes detallado en las secciones anteriores se obtiene una nueva pareja de imágenes que poseen la misma información que las originales, pero en las que destacan las regiones de interés que permitirán reconocer las figuras de la plataforma.

Para comenzar, se elabora el mapa de píxeles en el que se encuentran las zonas a procesar junto con el resto de píxeles de la imagen y una lista en la que se almacenan las regiones de interés que serán procesadas posteriormente.

El siguiente paso consiste en la aplicación de un método que se encarga de extraer las propiedades de las regiones pertenecientes a este mapa de píxeles. Estas características son: el área de la región, su centro, y los ejes mayor y menor. Las cuales se utilizan de forma combinada para el reconocimiento e identificación de las diferentes figuras de interés. Una vez se han calculado las características geométricas de cada región de la imagen, se obtienen las coordenadas de los contornos de cada una de ellas, su área y se realiza una estimación de su perímetro. Con estos datos es posible calcular una característica del objeto, proporcionada en la ecuación (7), que permite definir en qué medida éste coincide con la definición matemática de círculo, debido a que si se tratase de un círculo, este valor se aproxima a la unidad, mientras que si es otra figura este valor estará comprendido entre los intervalos definidos a

continuación en las tablas 12 y 13. Dicha característica quedará definida para el resto del documento como redondez.

$$Redondez = \frac{4 * \pi * area}{perimetro^2} \quad (7)$$

El procesamiento realizado sobre la imagen depende de su naturaleza, esto es: térmica o visible. En cualquier caso se obtienen las longitudes de los lados mayor y menor de la región para calcular un porcentaje de error, necesario para el reconocimiento de figuras con forma de paralelogramo. Este error es el resultado de multiplicar la media de los lados obtenidos, por un valor definido de forma experimental en función de datos empíricos y diversas pruebas de ensayo y error. La expresión matemática que lo sustenta es la proporcionada en la ecuación (8). Este valor, en caso de tratarse de una imagen infrarroja es 0.2, mientras que si es una imagen de color este valor es 0.1. Con estos posibles valores o errores calculados es posible determinar si el paralelogramo encontrado es un cuadrado o un rectángulo, ya que si la diferencia de los lados obtenidos supera el error se trata de un rectángulo (lados desiguales) y en caso contrario de un cuadrado.

$$Media = \frac{(LadoMayor + LadoMenor)}{2} \quad (8)$$

$$ErrorT = Media * 0,1(Térmica)$$

$$ErrorV = Media * 0,2(Visible)$$

La identificación de cada una de las figuras viene determinada por el valor de su redondez. Si la imagen es de tipo infrarrojo los intervalos de medida de la redondez de cada figura están descritos en la tabla 12, mientras que si se trata de una imagen normal se utilizan los intervalos de medida expuestos en la tabla 13. Estos intervalos de medida han sido definidos considerando que un valor cercano a la unidad corresponde a un círculo perfecto y mediante la realización de una gran cantidad de pruebas sobre figuras con diferentes posiciones y tamaños. Como ya se ha mencionado, en caso de que la figura sea un paralelogramo es necesario aplicar el error devuelto tras realizar el cálculo descrito en la ecuación (7) para diferenciar entre cuadrado y rectángulo. Si la diferencia entre la longitud del eje mayor y del eje menor es mayor que dicho error, entendiéndose que los ejes de un rectángulo tienen longitudes distintas por definición, se asume que la figura es un rectángulo, en otro caso y también por definición, se trata de un cuadrado.

Tabla 12: Identificación de figuras en función del valor de su redondez en imágenes térmicas.

Figura	Límite inferior	Límite superior
Triángulo	0.50	0.76
Cuadrado	0.76	0.84
Rectángulo	0.75	0.85
Círculo	0.85	1

Tabla 13: Identificación de figuras en función del valor de su redondez en imágenes visibles.

Figura	Límite inferior	Límite superior
Triángulo	0.40	0.65
Cuadrado	0.66	0.84
Rectángulo	0.66	0.85
Círculo	0.85	1

En caso de que la imagen sea RGB, también es necesario obtener el color de cada una de las figuras procesadas con el fin de añadir un filtro más al reconocimiento de la plataforma. Esto se consigue, como se puede observar en la figura 36, accediendo a un grupo de píxeles del centro de la región analizada y extrayendo su color promedio en valores R, G, B de sus respectivos canales espectrales.



Figura 36: Obtención del color de las figuras identificadas.

Por último, se almacenan las figuras finalmente reconocidas en una lista que contiene el nombre de la figura encontrada y su color asociado en caso de ser una imagen a color, o solamente el nombre de la figura si se trata de una imagen infrarroja. Para considerarse una figura válida, ésta tiene que pasar un filtro previo basado en la comparación del área de la región con unos límites superior e inferior definidos de forma arbitraria. De esta manera se pretende evitar la consideración de elementos demasiado grandes o pequeños que se contemplen como parte de la plataforma y causar confusión y errores a la hora del reconocimiento de la plataforma.

Una vez agrupadas todas las regiones en una lista, se procede a destacar los bordes de las figuras encontradas con el fin de que éstas se aprecien de mejor manera por el usuario en la interfaz gráfica.

4.6.4. Algoritmo de decisión

Una vez han sido tratados y procesados cada uno de los pares de imágenes, se procede a decidir en qué casos estos pares de imágenes capturados son válidos y candidatos para ser la plataforma que se busca.

Las imágenes de tipo infrarrojo únicamente tendrán en cuenta la forma de las figuras que se han encontrado en la imagen, mientras que en las imágenes RGB se debe tener en cuenta

tanto la forma como el color de cada una de las figuras. En las siguientes figuras, 37 y 38, se pueden ver los semáforos de figuras encontradas que aparecen en la interfaz. Si la aplicación detecta una figura, ésta se sombrea de color verde, mientras que si no se ha encontrado mantiene su color rojo original.



Figura 37: Cuadro de figuras en una imagen térmica.



Figura 38: Cuadro de figuras en una imagen visible.

Se tiene en cuenta un porcentaje de acierto diferente para cada una de los dos tipos de imágenes consideradas. Dicho porcentaje se calcula comparando los resultados obtenidos en cada una de las imágenes con una fotografía real de la plataforma que se desea encontrar. Después, se combinan estos dos porcentajes para calcular el porcentaje de acierto final. Además, para el cálculo de este último porcentaje se asigna una importancia diferente a los resultados referentes a los dos tipos de imagen. La expresión matemática que lo sustenta es la proporcionada en la ecuación (9), en la que $PVisible$ hace referencia al porcentaje de acierto en la imagen visible y $PTérmica$ hace referencia al porcentaje de acierto en la imagen térmica.

$$PorcentajeFinal = 0,6 * PVisible + 0,4 * PTérmica \quad (9)$$

4.7. Plan de gestión de riesgos

En esta sección se realiza un análisis de los riesgos más importantes que puedan afectar de manera significativa al proyecto. El análisis se ha desarrollado mediante el modelo Boehm (Alonso et al., 2005), aplicando una estrategia proactiva que gestione los riesgos antes de que afecten al proyecto. Esta decisión ha sido tomada debido a la importancia que supone tener un control de los riesgos adecuado, a la hora de conseguir que el proyecto se desarrolle de forma correcta y en los tiempos establecidos.

La gestión de riesgos se ha dividido en las siguientes fases:

- Identificación de los riesgos.
- Análisis de los riesgos.
- Priorización de los riesgos.
- Estudio y control de los riesgos.

4.7.1. Identificación de los riesgos

En primer lugar, se han enumerado los principales riesgos contemplados en el proyecto.

1. Mala comprensión de los casos de uso.
2. Requisitos no identificados.
3. Alteración de los requisitos.
4. Planificación del trabajo errónea.
5. Baja de uno o varios miembros del grupo.
6. Falta de conocimiento de las tecnologías utilizadas.
7. Mala organización y administración del equipo.
8. Mal funcionamiento de la aplicación.
9. Falta de medios para la realización del proyecto.
10. El tamaño del software ha sido subestimado.
11. Mala gestión de los riesgos.
12. Pérdida de interés y motivación.

4.7.2. Análisis de los riesgos

A continuación, se han analizado cada uno de los riesgos determinando su probabilidad y gravedad. Gracias a la gestión del riesgo Boehm (Alonso et al., 2005) se ha establecido un nivel de riesgo relacionando la probabilidad y consecuencias de los mismos. Estos riesgos han sido brevemente descritos en las tablas 14 a 25.

Tabla 14: Riesgo número uno.

Mala comprensión de los casos de uso	
Descripción	Falta de comprensión de alguno de los casos de uso indicados por el director de proyecto.
Probabilidad	Ocasional
Gravedad	Seria
Nivel del riesgo	Medio

Tabla 15: Riesgo número dos.

Requisitos no identificados	
Descripción	A la hora de identificar los requisitos se pasa por alto algún requisito de vital importancia para el proyecto.
Probabilidad	Ocasional
Gravedad	Crítico
Nivel del riesgo	Alto

Tabla 16: Riesgo número tres.

Alteración de los requisitos	
Descripción	Alguno de los requisitos cambia durante el desarrollo del proyecto.
Probabilidad	Ocasional
Gravedad	Serio
Nivel del riesgo	Medio

Tabla 17: Riesgo número cuatro.

Planificación del trabajo errónea	
Descripción	No se marcan correctamente los tiempos del proyecto, lo que lleva a retrasos en el mismo.
Probabilidad	Ocasional
Gravedad	Serio
Nivel del riesgo	Medio

Tabla 18: Riesgo número cinco.

Baja de uno o varios miembros del grupo	
Descripción	Uno o varios miembros del equipo causan baja durante el desarrollo del proyecto.
Probabilidad	Remoto
Gravedad	Serio
Nivel del riesgo	Bajo

Tabla 19: Riesgo número seis.

Falta de conocimiento de las tecnologías utilizadas	
Descripción	Alguno de los miembros del equipo no domina las tecnologías utilizadas durante el proyecto.
Probabilidad	Probable
Gravedad	Menor
Nivel del riesgo	Medio

Tabla 20: Riesgo número siete.

Mala organización y administración del equipo	
Descripción	Falta de coordinación entre los miembros del equipo, lo que implica duplicidades en el código y fallos a la hora de integrar.
Probabilidad	Ocasional
Gravedad	Menor
Nivel del riesgo	Bajo

Tabla 21: Riesgo número ocho.

Mal funcionamiento de la aplicación	
Descripción	La aplicación no ofrece los resultados esperados, ya sea por tiempos de procesamiento muy amplios o porque los resultados obtenidos son incorrectos.
Probabilidad	Ocasional
Gravedad	Crítico
Nivel del riesgo	Alto

Tabla 22: Riesgo número nueve.

Falta de medios para la realización del proyecto	
Descripción	No disponer de los medios necesarios para realizar el proyecto debidamente. Como puede ser la falta de una cámara térmica o una plataforma adecuada de aterrizaje.
Probabilidad	Ocasional
Gravedad	Serio
Nivel del riesgo	Medio

Tabla 23: Riesgo número diez.

El tamaño del software ha sido subestimado	
Descripción	El tamaño del software estimado al principio del desarrollo es mucho menor que durante el transcurso del proyecto.
Probabilidad	Ocasional
Gravedad	Serio
Nivel del riesgo	Medio

Tabla 24: Riesgo número once.

Mala gestión de los riesgos	
Descripción	No disponer de un plan de riesgos efectivo para prevenir o mitigar los posibles riesgos durante el proyecto.
Probabilidad	Remoto
Gravedad	Crítico
Nivel del riesgo	Medio

Tabla 25: Riesgo número doce.

Pérdida de interés y motivación	
Descripción	Alguno de los miembros del equipo pierde interés por el proyecto.
Probabilidad	Ocasional
Gravedad	Serio
Nivel del riesgo	Medio

4.7.3. Priorización de los riesgos

La técnica de priorización que se ha escogido viene determinada por el previo análisis, ya que los riesgos han sido priorizados según su nivel. Éstos se han agrupado en la tabla 26 ordenados por prioridad.

Tabla 26: Priorización de los riesgos

Prioridad	Riesgo
Alta	Requisitos no identificados
Alta	Mal funcionamiento de la aplicación
Medio	Mala comprensión de los casos de uso
Medio	Alteración de los requisitos
Medio	Planificación del trabajo errónea
Medio	Falta de conocimiento de las tecnologías utilizadas
Medio	Falta de medios para la realización del proyecto
Medio	El tamaño del software ha sido subestimado
Medio	Mala gestión de los riesgos
Medio	Pérdida de interés y motivación
Bajo	Baja de uno o varios miembros del grupo
Bajo	Mala organización y administración del equipo

4.7.4. Estudio y control de los riesgos

En este proyecto se ha decidido estudiar y controlar exclusivamente los riesgos que tengan un impacto alto e intolerable, debido a que son éstos los que realmente podrían influir en el proyecto. El plan de actuación sobre dichos riesgos se ha detallado en las tablas 27 y 28.

Tabla 27: Plan de actuación para el riesgo número dos.

Requisitos no identificados	
Estrategia para evitarlo	<ul style="list-style-type: none">- Poner en conocimiento del director del proyecto todos los requisitos identificados.- Revisar los requisitos por si faltara alguno.
Estrategia para minimizarlo	<ul style="list-style-type: none">- Reuniones periódicas con el director del proyecto para comprobar que no son necesarios nuevos requisitos.
Plan de contingencias	<ul style="list-style-type: none">- Realizar reuniones con el director del proyecto con el fin de obtener nuevas perspectivas del proyecto.

Tabla 28: Plan de actuación para el riesgo número ocho.

Mal funcionamiento de la aplicación	
Estrategia para evitarlo	<ul style="list-style-type: none"> - Realizar un testing riguroso de la aplicación. - Realizar buenas prácticas en la implementación (como tener comentado el código y utilizar patrones). - Contemplar todos los requisitos especificados.
Estrategia para minimizarlo	<ul style="list-style-type: none"> - Contar con un plan de solución de errores, en el caso de que estos se produjeran, para reducir al mínimo el tiempo de resolución.
Plan de contingencias	<ul style="list-style-type: none"> - Analizar los fallos y proceder a solucionarlos, modificando el código en el menor tiempo y de la manera más eficaz posible.

5. Implementación

En dicha sección se presenta el lenguaje de programación con el que se ha implementado el proyecto junto a las técnicas de desarrollo aplicadas, se detalla la arquitectura software de la aplicación incluyendo un diagrama de clases y se describen las metodologías software utilizadas.

5.1. Matlab

Se procede a describir [Matlab \(2015\)](#), la herramienta escogida para desarrollar la aplicación debido a que ofrece un entorno de desarrollo con una curva de fácil aprendizaje y un lenguaje de programación con una gran cantidad de funciones aplicables al procesamiento de imágenes.

[Matlab \(2015\)](#) es una herramienta software orientada a resolver problemas de ingeniería abarcando desde matemática computacional, aprendizaje automático o procesamiento de señales e imágenes hasta robótica o visión artificial. Además, se encuentra en una gran cantidad de sistemas y dispositivos tales como naves espaciales, seguridad de automóviles o monitorización sanitaria. Ofrece un lenguaje de programación propio basado en matrices que permite operar con vectores, funciones y programación orientada a objetos y que además es compatible con Unix, Windows y Mac OS.

Además de este lenguaje de programación, [Matlab \(2015\)](#) ofrece un entorno de desarrollo integrado muy intuitivo con el fin de facilitar la interacción del desarrollador con el mismo. Esta plataforma ofrece la posibilidad de representar datos y funciones mediante gráficos en

2D y 3D que proporcionan una gran cantidad de información, posteriormente utilizada en el análisis y desarrollo del proyecto al que sean aplicados. También es posible diseñar de forma sencilla interfaces con las que mejorar la interacción entre la aplicación y el usuario. Ofrece además grandes librerías, correctamente documentadas, con cajas de herramientas aplicables a cualquiera de los campos mencionados anteriormente.

Para el desarrollo de este proyecto cabe destacar la herramienta “Image Processing”. Se trata de una herramienta enfocada al procesamiento de imágenes, que proporciona funciones y aplicaciones para el análisis (segmentación), mejora (enfoque y reducción de ruido) y visualización de las mismas (ajuste de color y contraste, histogramas). Realiza todas estas operaciones aplicando diferentes operaciones o filtros con el fin de obtener el mejor resultado y de la manera más sencilla posible. Además, soporta un conjunto muy amplio de imágenes para evitar problemas de compatibilidad entre formatos y tamaños. Entre las aplicaciones que proporciona cabe destacar *Color Thresholder* que permite segmentar imágenes en función de sus colores, *Image Viewer* que permite manipular formas en las imágenes y muestra una gran cantidad de información o *Video Viewer* semejante a la anterior pero aplicada a videos. También proporciona funciones capaces de extraer información importante de las imágenes, como contar objetos, localizar bordes y formas o reconocer colores.

5.2. Técnicas de procesamiento aplicadas

Durante el desarrollo de la aplicación se han utilizado diferentes técnicas de procesamiento de imágenes que han sido implementadas en el entorno de desarrollo [Matlab \(2015\)](#). A continuación, se exponen las funciones contenidas en los recursos de [Matlab \(2015\)](#) que han servido para implementar estas técnicas.

5.2.1. Funciones básicas

Lectura de una imagen

La función `imread` se encarga de acceder a la ruta que contiene una imagen y de almacenar dicha imagen en memoria. Los tipos de imágenes soportados son: TIFF, JPEG, GIF, BMP y PNG. La función recibe como argumento la ruta y devuelve la imagen cargada.

```
RGB = imread(RUTA);
```

Visualización de una imagen

La función `imshow` utiliza una ventana vacía para mostrar datos o imágenes contenidos en la variable que se pasa por parámetro.

```
imshow(imagen);
```

5.2.2. Funciones aplicadas al procesamiento de imágenes

Binarización aplicando el método de Bradley

En [Matlab \(2015\)](#) dicha binarización se implementa mediante la función `imbinarize` (versión R2016a) o `im2bw` (versiones anteriores). La función recibe como parámetro de entrada la

imagen a binarizar y, de forma opcional, el umbral a partir del cual se aplica la binarización. Como resultado se obtiene la imagen binarizada que puede observarse en la figura 12.

```
BW = im2bw(GRAY, level);
```

Valor umbral obtenido por el método de Otsu

En [Matlab \(2015\)](#) este valor umbral se obtiene aplicando la función `graythresh`. Esta función recibe como único argumento de entrada la imagen de la que obtener el valor umbral. Devuelve un valor tipo `double`. La aplicación de este valor umbral puede observarse en la figura 13.

```
level = graythresh(GRAY);
```

Detección de bordes

En [Matlab \(2015\)](#) la detección de bordes se implementa mediante varias funciones. En este proyecto se ha hecho uso de la función `edge`, que localiza los bordes de los elementos internos de la imagen utilizando sus intensidades dependiendo del operador que se le indique. La función recibe como parámetros de entrada la imagen a tratar, el operador y, de forma opcional, un valor umbral en función de los resultados requeridos. Devuelve una imagen en la que destacan los bordes, representada en la figura 14.

```
edge(Imagen, 'Canny', level);
```

Rellenar regiones y huecos

En [Matlab \(2015\)](#) se utiliza la función `imfill` para rellenar los huecos de los objetos de la imagen. Dicha función recibe como argumentos la imagen a procesar y, opcionalmente, el parámetro `'holes'`, que indica qué regiones se desean rellenar. Si no se indica este parámetro se toma como hueco a un área de píxeles oscuros rodeados por píxeles más claros. El resultado de este procesamiento se puede observar en las figuras 15 y 16.

```
BW = imfill(Imagen, 'holes');
```

Structuring element decomposition

En [Matlab \(2015\)](#) esta técnica se implementa mediante la función `strel`. Esta función recibe como argumentos el patrón de la figura a obtener y su dimensión. Devuelve el elemento estructural definido.

```
SE = strel('disk', radio);
```

Operaciones morfológicas

En [Matlab \(2015\)](#), existen varios tipos de operaciones morfológicas. En el presente proyecto se ha hecho uso de `imopen` e `imclose`, apertura y cierre morfológico respectivamente. Ambas se realizan indicando la imagen a la que es aplicada la operación y el elemento estructural a utilizar, que determina el nivel de detalle con el que debe realizarse dicha operación. Devuelven la imagen tratada como se puede comprobar en la figura 17.

```
IM2 = imopen(IM, SE);
```

```
IM2 = imclose(IM2, SE);
```

Operaciones morfológicas por parámetro

Además de las descritas anteriormente, en [Matlab \(2015\)](#) también es posible realizar operaciones morfológicas mediante el paso de parámetros. Se implementan mediante la función `bwmorph`, que recibe como entrada la imagen a procesar y la operación a realizar. Devuelve la imagen tratada, expuesta en la figura 17. Algunos tipos son:

- Apertura morfológica: `BW = bwmorph(BW, 'open');`
- Cierre morfológico: `BW = bwmorph(BW, 'close');`
- Eliminación de píxeles del interior de una imagen: `BW = bwmorph(BW, 'remove');`
- Eliminación de píxeles aislados: `BW = bwmorph(BW, 'clean');`

Exclusión de objetos

En [Matlab \(2015\)](#) esta eliminación de objetos no deseados se implementa mediante la función `bwareaopen`, a la que se pasa como argumento la imagen a procesar y el número mínimo de píxeles que debe superar cada elemento para permanecer en ella. El resultado de este procedimiento se puede observar en la figura 18.

```
bwareaopen(BW, numMin);
```

Algoritmo de Moore-Neighbor

En [Matlab \(2015\)](#) dicho algoritmo se aplica con el objetivo de reconocer las regiones destacadas de la imagen. Se implementa mediante la función `bwboundaries`, a la que se introducen como parámetros de entrada la imagen a tratar y, de forma opcional, el parámetro 'noholes' que determina si se desea obtener el borde de los componentes padres e hijos, para una mayor eficacia del algoritmo. Devuelve como salida el mapa de píxeles que representa la imagen y una lista de las regiones encontradas que componen este mapa.

```
[B, L] = bwboundaries(BW, 'noholes');
```

Extracción de propiedades

En [Matlab \(2015\)](#), es posible obtener las propiedades de las zonas deseadas de una imagen mediante la función `regionprops`, en la que se introducen en la misma llamada las características que se desean obtener y la matriz de la cual serán extraídas. Devuelve una lista en la que se almacenan dichas propiedades. Las siguientes propiedades han sido útiles en el desarrollo del proyecto:

```
regionprops(L, 'Area', 'Centroid', 'Perimeter', 'MajorAxisLength',  
'MinorAxisLength');
```

Extracción de color

En [Matlab \(2015\)](#) se pueden conocer los colores RGB de un píxel mediante la función `impixel`, a la que se le pasan como argumentos de entrada la imagen y las coordenadas del píxel en cuestión. Devuelve los valores RGB pertenecientes a dicho píxel. Ésto puede comprobarse en la figura 36, en la que se obtiene el color de cada uno de los píxeles de las regiones del centro de la figura.

```
COLOR = impixel(imagen, x, y);
```

5.3. Arquitectura software

Con respecto a la implementación de la aplicación, se procede a describir la arquitectura software aplicada al proyecto con la ayuda del diagrama de clases de la figura 39.

La aplicación está compuesta por las clases GUI, Controlador, LimpiarT, LimpiarC, Procesar y Decisión. Son, por tanto, seis clases que permiten una fácil comprensión de las diferentes partes que integran la aplicación. Además, como se puede observar en la figura 39, la arquitectura software está basada en el patrón Modelo-Vista-Controlador (Gamma and Fernández-Acebal, 2002). Tanto este patrón como la implementación de las clases se detallan posteriormente. Por último, se ha incluido en el diagrama anterior la unidad de almacenamiento en la que, como su nombre indica, se almacenan las imágenes a procesar.

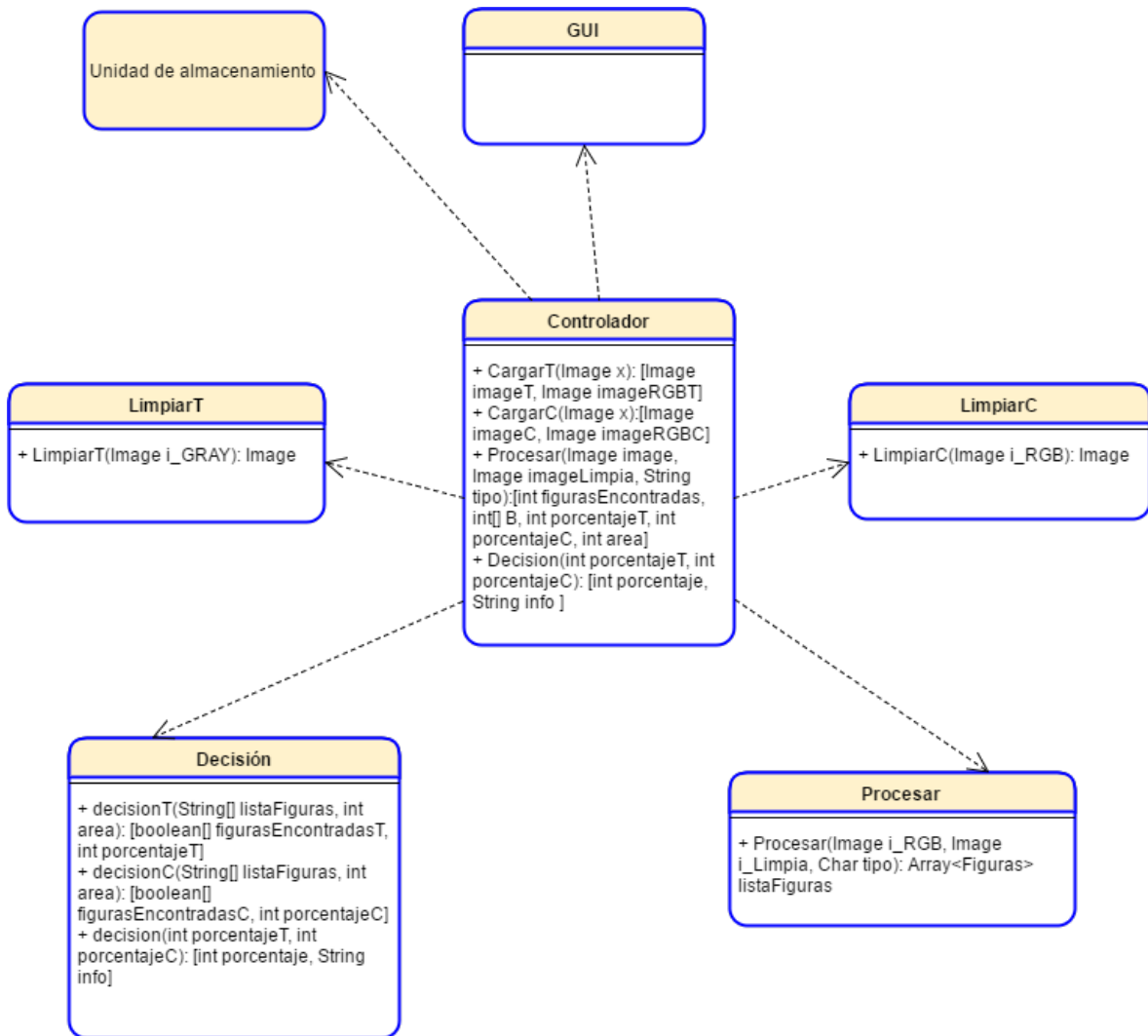


Figura 39: Diagrama de clases de la aplicación.

5.4. Metodologías utilizadas

5.4.1. Modelo-Vista-Controlador (MVC)

El MVC ([Gamma and Fernández-Acebal, 2002](#)) es un patrón de arquitectura software que permite separar la parte visual de la parte lógica de una aplicación. Esto significa que la interfaz gráfica de una aplicación no tiene ningún tipo de contacto directo con la parte del modelo de datos y viceversa. Dentro de esta aplicación el Modelo está compuesto por las imágenes y por la lógica que se utiliza para el tratamiento de éstas. Es en esta capa donde se desarrolla la operatividad de la aplicación. La Vista corresponde a toda la parte gráfica de la aplicación. Esta parte constituye únicamente la interfaz gráfica que permite la interacción del usuario con la aplicación. Por otro lado, el Controlador permite distribuir las peticiones que la vista envía. Éste forma parte de una capa intermedia entre la Vista y el Modelo.

Durante el desarrollo de proyecto se ha realizado una separación entre las capas correspondientes a la Vista, el Controlador y el Modelo de datos utilizado.

Asociación Vista-Controlador

La Vista contiene un conjunto de componentes visuales que permiten realizar un avance de las imágenes y su procesamiento. Cuando se acciona cualquiera de los botones, se realiza una llamada a una de las acciones implementadas dentro del Controlador. El controlador se encuentra permanentemente escuchando las llamadas que la Vista pueda hacer para, en el momento en el que éstas se realicen, estar listo para su ejecución. Posteriormente el Controlador realiza el proceso de actualización de la interfaz. Cuando la línea de ejecución del programa finaliza, los datos resultantes del procesado de las imágenes se muestran en la interfaz. El Controlador es el responsable de recoger estos datos y volcarlos en los componentes de visualización de la interfaz.

Asociación Modelo-Controlador

Durante la ejecución de la aplicación desarrollada, el Controlador cede el control de ésta al Modelo, que implanta la lógica de la aplicación. El Modelo, entendiéndose como la lógica y la representación de los datos a tratar, supone la mayor parte de la carga de procesamiento de la aplicación. En el momento en el que el Modelo termina una fase de su ciclo de vida actualiza el estado de los datos en las diferentes partes de la aplicación. De esto se encarga el Controlador, que actúa como intermediario entre todos los extremos de la aplicación.

Implementación del MVC en el proyecto

- La clase GUI implementa la vista de la aplicación y es la encargada de la interacción con el usuario. Como entrada recibe una imagen térmica y/o una imagen de luz visible en color. Como salida muestra información relativa a los resultados obtenidos al procesar las imágenes como el número de figuras encontradas, la lista de figuras y colores reconocidos así como información acerca de la imagen.
- La clase Controlador es la clase encargada de comunicar la vista con el modelo, responde a los eventos que le envía la vista solicitando al modelo la información necesaria sobre las imágenes.

- El modelo está compuesto por el resto de clases, brevemente descritas a continuación:
 - La clase LimpiarT recibe como parámetro la imagen térmica a procesar. Aplica las técnicas de tratamiento de imágenes sobre ella, descritas en la sección 4.6.1 y devuelve una imagen tratada, en la que los elementos que la componen destacan sobre el resto.
 - La clase LimpiarC recibe como parámetro la imagen visible en el modelo de color RGB a procesar. Igual que la clase anterior, aplica diferentes técnicas de tratamiento de imágenes, descritas en la sección 4.6.2, sobre la imagen RGB para devolver una imagen limpia, cuyos elementos se distinguen del resto de la imagen.
 - La clase Procesar recibe como argumentos la imagen real, la imagen tratada, devuelta por las clases anteriores y una variable que indica el tipo de imagen a procesar. Devuelve una lista de figuras y colores reconocidos mediante técnicas de procesamiento de imágenes, detalladas en la sección 4.6.3 en función del tipo de imagen a procesar.
 - La clase Decisión recibe como entrada la lista de figuras y colores obtenida del procesamiento de las imágenes. Dependiendo de distintos factores, todos ellos descritos en la sección 4.6.4 y de los resultados obtenidos con las dos cámaras, determina si el objeto que aparece en la imagen es la plataforma deseada o no. Devuelve un porcentaje de éxito.

5.4.2. Singleton

El patrón Singleton ([Gamma and Fernández-Acebal, 2002](#)) es utilizado generalmente en diseños de arquitectura software que permite que una determinada clase sea instanciada una única vez. Durante el desarrollo de la aplicación se ha utilizado este patrón en la construcción de la interfaz gráfica. El propio entorno de desarrollo de [Matlab \(2015\)](#) fuerza a que la interfaz sólo se pueda instanciar una vez, por lo que claramente hace gala del uso de este patrón. Durante el desarrollo de una aplicación [Matlab \(2015\)](#) se debe considerar esta limitación ya que esta plataforma obliga a la creación de una clase por interfaz de aplicación deseada, por lo que si se desea contar con más de una interfaz será necesaria la creación de nuevas clases.

6. Análisis de resultados

Esta sección presenta los resultados obtenidos a lo largo del desarrollo del proyecto para reflejar tanto aciertos como errores en el procesamiento.

A continuación, se muestran los resultados obtenidos en el presente proyecto con los que se ha logrado cumplir en gran medida los objetivos marcados al inicio de este proyecto. Estos resultados han sido ordenados de mayor a menor porcentaje de éxito combinando los resultados obtenidos mediante el procesamiento de ambas imágenes. Posteriormente, se han incluido los resultados del tratamiento de algunas imágenes individuales, que exponen los fallos detectados en función del tipo de imagen, junto con su valoración.

Cabe destacar que cada par de imágenes térmica y visible han sido capturadas de forma simultánea, para emular la toma de fotografías por el mismo dispositivo en el mismo instante de tiempo.

Se debe recalcar que cada una de las figuras reconocidas en la imagen térmica se ha bordeado en color blanco, mientras que en la imagen visible las figuras reconocidas han sido remarcadas en color negro. Esta distinción de colores se hace en base al color de la plataforma permitiendo así una fácil visualización de las figuras detectadas.

Durante los experimentos se han tomado del orden de 500 imágenes de ambos tipos. Los resultados que aparecen a continuación corresponden a muestras significativas de imágenes que ilustran en su generalidad los distintos casos observados.

6.1. Combinación de imágenes térmica-visible

Caso 1: Acierto 100 %

Tanto en la figura 40 como en la figura 41, se consigue el 100% de acierto en ambos tipos de imágenes, a pesar de que en la figura 40 uno de los triángulos no es reconocido de forma exacta. Esto es así gracias a que estas imágenes fueron capturadas en un día soleado, lo que permitió que las figuras metálicas se calentaran y se distinguieran con nitidez respecto de la base de la plataforma. Además, las cámaras se situaron de forma perpendicular a la plataforma de forma que no aparecieran reflejos y sombras sobre ella procedentes de otros elementos del entorno.

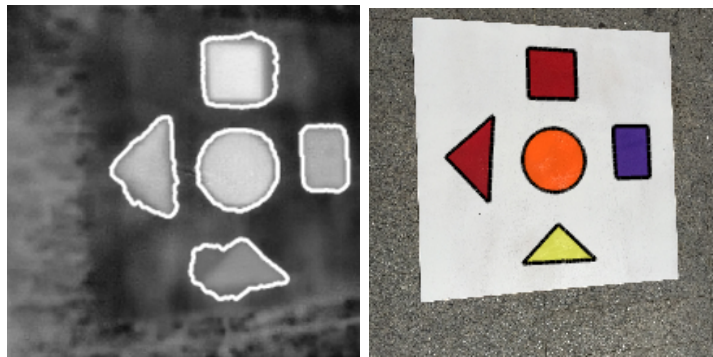


Figura 40: Combinación imagen térmica - visible con un grado de acierto del 100 %.

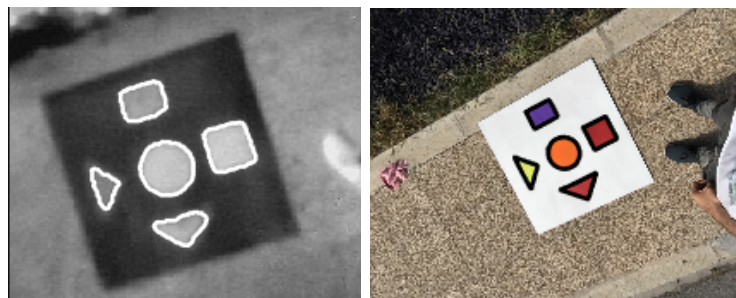


Figura 41: Combinación imagen térmica - visible con un grado de acierto del 100 %.

Caso 2: Acierto 76 %

En la figura 42 se obtiene un porcentaje de acierto del 40 % para la imagen térmica y un 100 % para la imagen visible, lo que supone un 76 % de acierto total. En este caso, las imágenes fueron tomadas en un día nublado, lo que supuso que, como se puede observar en la imagen térmica, las figuras metálicas no se calentaron lo suficiente. Bajo estas condiciones resulta difícil distinguir las figuras de la plataforma con la cámara térmica. No obstante, en la imagen visible se puede observar que la luz es suficiente para identificar la forma y color de cada una de las figuras de forma correcta.



Figura 42: Combinación imagen térmica (40 %) - visible (100 %) con un grado de acierto del 76 %.

Caso 3: Acierto 76 %

En la figura 43 se muestra una situación opuesta a la descrita en el caso 2, aunque el acierto total es el mismo, el acierto en la imagen térmica es del 100 % y en la imagen visible del 60 %. Gracias a que el sol incidió perpendicularmente sobre la plataforma las figuras metálicas se calentaron con mayor facilidad, lo que permitió diferenciarlas claramente de la base de la plataforma. Por el contrario, en la imagen visible se puede observar un reflejo sobre el triángulo amarillo y parte del círculo. Esto impide el reconocimiento correcto de la forma y color en dichas figuras. El triángulo desaparece por completo de la imagen, mientras que el círculo aparece bordeado ya que su forma ha sido reconocida pero su color no, por lo que no se tiene en cuenta en el cálculo del porcentaje de acierto.

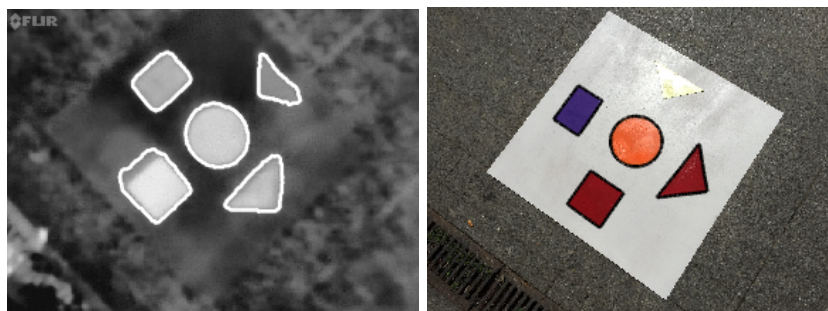


Figura 43: Combinación imagen térmica (100 %) - visible (60 %) con un grado de acierto del 76 %.

Caso 4: Acierto 72 %

En la figura 44 el porcentaje de acierto total es del 72 %. En la imagen térmica se reconocen tres figuras sobre la plataforma y dos fuera de ésta. En este caso, las dos formas que se observan fuera de la plataforma produjeron un falso positivo. La intención es la de ser capaces de contabilizar únicamente las figuras que aparecen dentro de la plataforma, por lo que este caso se ha reservado como trabajo futuro. En la imagen visible el cuadrado se identifica como rectángulo debido a que dicha imagen no se capturó de forma perpendicular a la plataforma. Sin embargo, se ha conseguido reconocer los colores de todas las figuras a pesar de que la sombra afectaba total o parcialmente a algunos de sus colores.

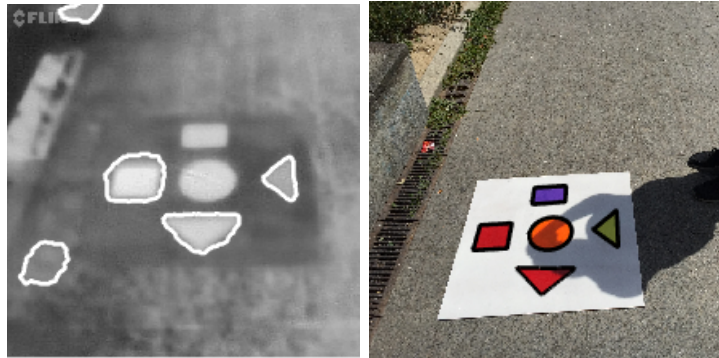


Figura 44: Combinación imagen térmica (60 %) - visible (80 %) con un grado de acierto del 72 %.

Caso 5: Acierto 68 %

En las capturas mostradas en la figura 45, cuyo porcentaje de éxito es del 68 %, aparecen tres objetos que no tienen que ver con la plataforma. Estos elementos han sido colocados de forma intencionada para dificultar el reconocimiento de la plataforma, a la vez que determinar el grado de fiabilidad de la aplicación desarrollada. Por un lado, en la imagen térmica los objetos han sido ignorados debido a su tamaño, si bien el cuadrado no ha podido ser detectado debido a que una gran parte de su estructura ha sido ocultada por otro objeto. Por otro lado, en la imagen visible, al igual que sucede con la imagen térmica, los objetos no pertenecientes a la plataforma han sido ignorados por su tamaño, exceptuando el objeto que está situado encima del cuadrado que impide reconocer con claridad esta figura. También es preciso añadir que debido al reflejo del sol el triángulo amarillo no se detecta de forma correcta.

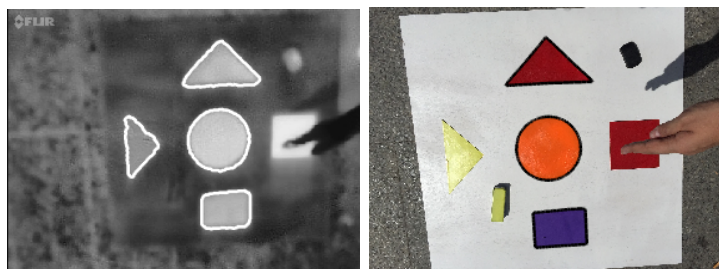


Figura 45: Combinación imagen térmica (80 %) - visible (60 %) con un grado de acierto del 68 %.

Caso 6: Acierto 64 %

En la figura 46 se obtiene un porcentaje de acierto total del 64 %. En la imagen térmica se detectan todas las figuras por las mismas circunstancias descritas en el caso 3. En la imagen visible un reflejo del sol incide en el centro de la plataforma provocando que no se reconozca de forma correcta la forma y el color del círculo y los dos triángulos. Es precisamente en este tipo de situaciones en las que resulta realmente útil la combinación de ambos tipos de imágenes, debido a que en condiciones climáticas adversas una imagen complementa a la otra.

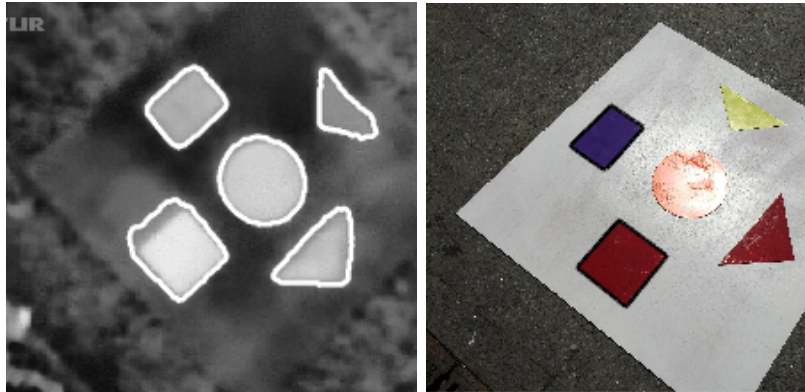


Figura 46: Combinación imagen térmica (100 %) - visible (40 %) con un grado de acierto del 64 %.

6.2. Fallos detectados por tipo de imagen

Caso 7: Acierto 60 %. Imagen térmica

En la figura 47 el porcentaje de acierto obtenido es del 60 %. Sin embargo, observando la figura es posible darse cuenta de que esto es erróneo, debido a que el triángulo que ha sido reconocido en la figura realmente es parte de un rectángulo (se pueden ver parcialmente los otros dos triángulos en la plataforma). Este resultado se debe a que la imagen fue tomada en un día nublado y, por ello, la figura no absorbió el calor de forma homogénea. Por el mismo motivo los triángulos pertenecientes a la plataforma no fueron detectados.

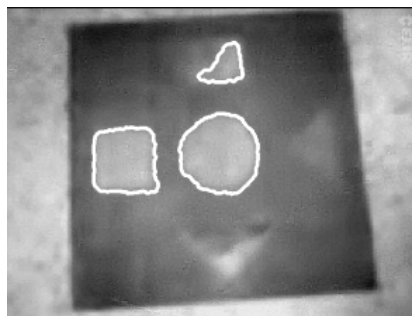


Figura 47: Imagen térmica con un grado de acierto del 60 %.

Caso 8: Acierto 20 %. Imagen térmica

En la figura 48 el porcentaje de acierto obtenido es del 20 %. Este resultado representa uno de los menores porcentaje de acierto obtenido, debido a que la imagen fue tomada en un día nublado y las figuras no absorbieron el calor lo suficiente para ser detectadas de forma correcta. Por este motivo, éste es uno de los resultados en los que más énfasis debería hacerse en cuanto a líneas de investigación futuras se refiere.

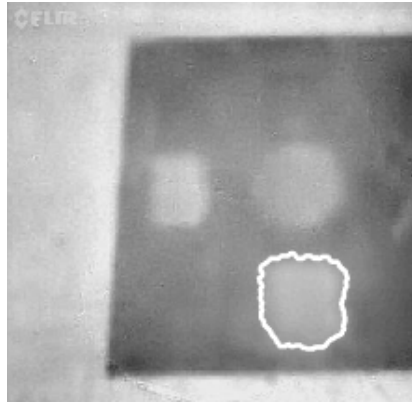


Figura 48: Imagen térmica con un grado de acierto del 20 %.

Caso 9: Acierto 80 %. Imagen visible

En la figura 49 el porcentaje de acierto alcanzado resulta ser del 80 %. Este resultado obtenido es debido a que la imagen fue tomada desde una posición no perpendicular a la plataforma. Esto provocó que el cuadrado se reconociera como rectángulo debido a la proyección de perspectiva del sistema óptico y, por ello, no se reconocieran todas las figuras. No obstante, el color de la figura se reconoció con éxito en todas las figuras.

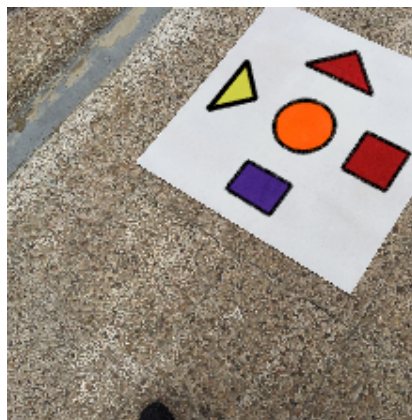


Figura 49: Imagen visible con un grado de acierto del 80 %.

Caso 10: Acierto 60 %. Imagen visible

Se puede observar cómo en la figura 50 se ha obtenido un porcentaje de acierto del 60 %. Este resultado es debido a que la imagen fue tomada desde una posición no perpendicular a la plataforma y con la luz de sol incidiendo directamente sobre una de las figuras metálicas. Esto provocó que la forma y color del cuadrado, así como el color naranja del círculo, fueran distorsionados produciendo que estas dos figuras no fueran detectadas por los procesos desarrollados en la aplicación.



Figura 50: Imagen visible con un grado de acierto del 60 %.

Caso 11: Acierto 100 %. Imagen visible

En la figura 51 se puede observar el caso en el que una sombra se sitúa sobre la plataforma. A pesar de esto, se ha obtenido un porcentaje de acierto del 100 %. Gracias a la capacidad del algoritmo desarrollado para eliminar las sombras, éstas con carácter general no han supuesto ningún problema a la hora de reconocer la forma de las figuras, siendo más complicado el reconocimiento del color de las mismas. Esto es así debido a que la sombra distorsiona su composición RGB oscureciendo estos valores.

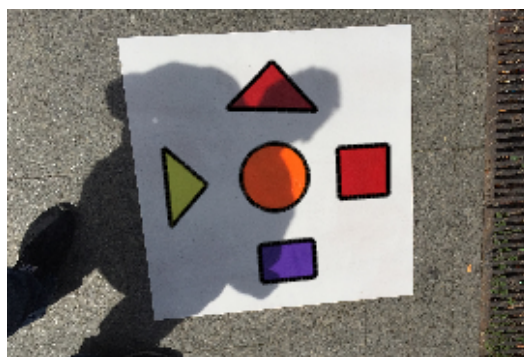


Figura 51: Imagen visible con sombras con un grado de acierto del 100 %.

Caso 12: Acierto 80 %. Imagen visible

Por último, en la figura 52 se observa otro caso en el que una sombra se sitúa sobre la plataforma. En este caso se pueden comprobar de nuevo lo mencionado en el caso anterior. El color del cuadrado rojo no se ha obtenido de la forma correcta debido a la distorsión provocada por la sombra en su composición RGB. Sin embargo, sí ha sido posible reconocer el color y forma del resto de figuras, obteniendo un porcentaje del 80 %.

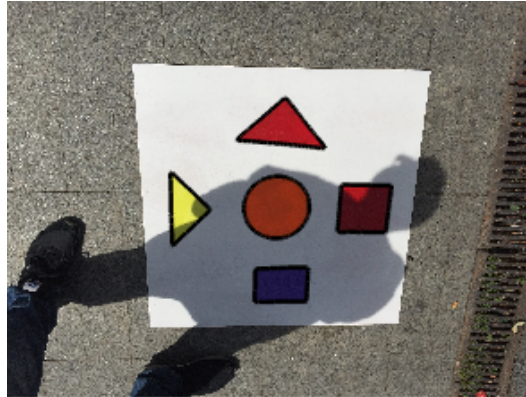


Figura 52: Imagen visible con sombras con un grado de acierto del 80 %.

7. Conclusiones y trabajo futuro

En esta última sección se exponen las conclusiones que han sido extraídas una vez los resultados han sido obtenidos, además de incluir posibles líneas de investigación para los proyectos venideros. También se detallan las contribuciones personales de cada componente del proyecto.

7.1. Conclusiones

El objetivo del presente proyecto consiste en mejorar el índice de resultados positivos obtenidos tras el procesamiento de las imágenes recibidas por los dispositivos que han sido descritos en el apartado 4.4.1. Para ello, se ha diseñado una plataforma compuesta por una base de madera de color blanco y cinco figuras geométricas metálicas de diferentes colores, con el fin de reconocer dichas figuras mediante captura de imágenes tanto en el espectro visible como infrarrojo. Gracias a este diseño de la plataforma y a la combinación de la captura de imágenes de estas naturalezas, se ha conseguido mejorar ostensiblemente la tasa de acierto respecto de otros trabajos precedentes en la misma línea aunque con plataformas diferentes (Álvarez-Piñeiro, 2016); (García-Pulido, 2014); (García-Pulido et al., 2017).

Durante el transcurso del proyecto se ha intentado otorgar a un dispositivo la capacidad de reconocer la plataforma mediante la toma de imágenes y su posterior análisis. Este proyecto,

inspirado en los trabajos mencionados, ha requerido de un estudio del reconocimiento de figuras y colores sobre imágenes tomadas por cámaras operando en los espectros infrarrojo y visible, lo que supone un avance con respecto a la utilización de la plataforma utilizada en los citados trabajos.

Con la aplicación desarrollada se ha conseguido un resultado en la detección de plataformas aceptable. Esto ha sido posible gracias al uso de métodos de tratamiento especializados en el análisis de regiones dentro de una imagen. Así, las imágenes tratadas han sido simplificadas haciendo énfasis en aquellas regiones importantes en la labor de reconocimiento.

Además, se ha podido observar que las condiciones meteorológicas o los materiales usados para la detección de objetos pueden influir en los resultados. Una menor o mayor intensidad luminosa procedente del sol en la plataforma o la aparición de sombras sobre ésta pueden afectar al reconocimiento de las formas o los colores. Igualmente, el material con el que están fabricados los componentes de la plataforma también influye en los resultados. Esto se debe a que la luz de sol afecta de forma negativa al reconocimiento de colores debido a los brillos generados, mientras que afecta de forma positiva en el reconocimiento de imágenes térmicas debido a que el calor se transmite con facilidad a las piezas metálicas de la plataforma y la reflectancia de estos elementos por la cámara visible resulta notable. Por otro lado, estas figuras metálicas permiten generar un contraste de temperaturas en relación a la base de la plataforma, con menor reflectividad, de tal manera que en estos casos, la cámara térmica obtiene resultados igualmente aceptables.

De esta manera, el resultado obtenido en forma de aplicación, otorgaría a futuros estudios una base de investigación simplificando la búsqueda de trabajos previos. El trabajo desarrollado proporciona una cobertura metodológica general desde el punto de vista de reconocimiento de objetos en entornos completamente reales.

Los objetivos marcados al inicio de este proyecto fueron los siguientes:

- Reconocimiento de figuras geométricas en el campo visible, que se ha cumplido en términos generales. Si bien es cierto que se han detectado dos fallos a tener en cuenta en futuras líneas de investigación. El primero es el tratamiento de la figura geométrica del cuadrado dado que al ser una figura perfecta el ángulo de inclinación de la cámara puede confundir dicha figura por un rectángulo debido a la proyección de perspectiva. El segundo ocurre, por otro lado, si un rayo de sol incide directamente sobre una figura, ésta no se detecta apropiadamente en la imagen de color, mientras que en su homóloga térmica se detecta sin ningún problema.
- Reconocimiento de los colores de las figuras geométricas en imágenes de color en el espectro visible. Se puede concluir que el objetivo perseguido al inicio del proyecto ha sido conseguido, ya que ha sido posible identificar dichos colores tanto en días nublados como en días soleados, tan solo fallando en la detección cuando el sol incide directamente sobre alguna de las figuras o en el caso de que la figura sea completamente invadida por sombras.
- Reconocimiento de figuras geométricas en el espectro infrarrojo. El resultado de este objetivo ha sido especialmente satisfactorio. Esto es debido a que en imágenes en las que los contornos de las figuras geométricas estaban difuminadas, haciendo muy difícil su detección, se ha sido logrado adaptar el algoritmo de tal manera que la identificación de

estas figuras fuese satisfactoria. Solo cabe señalar el fallo en la detección de las figuras en días nublados, ya que los objetos metálicos no emiten la suficiente reflectividad debida a la radiación térmica, haciendo difícil su reconocimiento.

- Diseño de una interfaz fácil e intuitiva. El objetivo en este punto era poder mostrar de la forma más clara posible los resultados obtenidos durante el desarrollo de este proyecto, para ello se ha desarrollado una interfaz con un mínimo número de elementos permitiendo que ésta muestre de forma unívoca los resultados obtenidos tras el procesamiento de las imágenes. Esta interfaz se ha probado con diferentes perfiles de usuarios y en todos los casos éstos han sido capaces de utilizar la aplicación y de entender los resultados mostrados por la misma.
- Arquitectura modular. Se ha conseguido realizar una aplicación modular basada en el patrón MVC ([Gamma and Fernández-Acebal, 2002](#)). Todas las clases de la aplicación están claramente diferenciadas, siendo fácil identificar cuál de ellas pertenece a cada capa. Esto también otorga una flexibilidad al diseño, de forma que la inclusión de nuevas funcionalidades se pueden añadir con facilidad, cuando esto sea necesario en el futuro.

Mediante el estudio realizado, se puede concluir que el objetivo general inicial planteado en este trabajo se ha alcanzado con éxito.

7.2. Conclusions

The objective of the present project is to improve the index of positive results obtained after the processing of the images received by the devices that have been described in section 4.5. For this purpose, a platform consisting of a white wooden base and five metallic geometric figures of different colors has been designed, to recognize these figures by capturing images in both the visible and infrared spectrum. Thanks to this design of the platform and the combination of the capture of images of these natures, it has been possible to improve significantly the success rate respect to other previous works in the same line although with different platforms [Álvarez-Piñeiro \(2016\)](#); ([García-Pulido, 2014](#)); [?GarciaPulido\(2017\)](#).

During the course of the project we have tried to grant a device the ability to recognize the platform by taking pictures and carry out their subsequent analysis. This project, inspired by the mentioned works, has required a study of the recognition of figures and colors on images taken by cameras operating in the infrared and visible spectrum, which represents an advance with respect to the use of the platform used in the mentioned works.

With the application developed, an acceptable platform detection result has been achieved. This has been possible thanks to the use of specialized treatment methods in the analysis of regions within an image. Thus, the treated images have been simplified with emphasis on those important regions in the reconnaissance work.

In addition, it has been observed that weather conditions or materials used for the detection of objects can influence the results. A lower or greater luminous intensity coming from the sun on the platform or the appearance of shadows on it can affect the recognition of the forms or the colors. Likewise, the material with which the platform components are manufactured

also influences the results. This is because sunlight affects negatively the recognition of colors due to the generated brightness, while it affects positively in the recognition of thermal images because heat is transmitted easily to the metal parts of the platform and the reflectance of these elements by the visible camera is remarkable. On the other hand, these metallic figures allow to generate a contrast of temperatures in relation to the base of the platform, with less reflectivity, in such a way that in these cases, the thermal camera obtains equally acceptable results.

In this way, the result obtained in the form of application, would grant to future studies a base of investigation, thus simplifying the search of previous works. The work developed provides general methodological coverage from the point of view of object recognition in completely real environments.

The objectives set at the beginning of this project were the following:

- Recognition of geometric figures in the visible field, which has been fulfilled in general terms. While it is true that two failures to be taken into account in future lines of research have been detected. The first is the treatment of the geometric figure of the square because, being a perfect figure, the angle of inclination of the camera can confuse the figure with a rectangle due to the projection of perspective. The second occurs, on the other hand, if a ray of sun directly strikes a figure, it is not detected properly in the color image, while in its thermal counterpart it is detected without any problem.
- Recognition of the colors of the geometric figures in color images in the visible spectrum. It can be concluded that the objective pursued at the beginning of the project has been achieved, since it has been possible to identify these colors both on cloudy days and on sunny days, only failing to detect them when the sun directly affects one of the figures or in the case that the figure is completely invaded by shadows.
- Recognition of geometric figures in the infrared spectrum. The result of this objective has been particularly satisfactory. This is because in images in which the contours of the geometric figures were blurred, and therefore making detection very difficult, it has been possible to adapt the algorithm in such a way that the identification of these figures was satisfactory. It is only possible to point out the fault in the detection of the figures on cloudy days, since the metallic objects do not emit enough reflectivity due to thermal radiation, making it difficult to recognize them.
- Design of an easy and intuitive interface. The objective at this point was to be able to show as clearly as possible the results obtained during the development of this project, for an interface has developed with a minimum number of elements allowing it to show uniquely the results obtained after processing of the images. This interface has been tested with different user profiles and in all cases they have been able to use the application and understand the results displayed by it.
- Modular architecture. A modular application based on the MVC ([Gamma and Fernández-Acebal, 2002](#)) pattern has been achieved. All classes of the application are clearly differentiated, being easy to identify which of them belongs to each layer. This also gives flexibility to the design, so that the inclusion of new functionalities can be added easily, when this is necessary in the future.

Through the study, we can conclude that the initial general objective set out in this work has been successfully achieved.

7.3. Trabajo futuro

A pesar de la buena tasa de acierto conseguida con el desarrollo de los métodos propuestos en este proyecto, también se pueden considerar mejoras que permitirían ampliar el campo de pruebas para tratar una mayor variabilidad en las condiciones meteorológicas.

Durante el desarrollo de este proyecto se han estudiado en profundidad las metodologías existentes dedicadas al procesamiento digital de imágenes. Esto ha permitido que la aplicación resultante contenga un nivel de calidad suficiente. No obstante, es evidente que este trabajo establece la base para continuar con la aplicación de más técnicas, ya sea de procesamiento digital o tecnológico en general, para mejora de los resultados.

Más específicamente, habría que darle más importancia a las mejoras de la aplicación que permitieran mejorar de forma directa los resultados en el reconocimiento de la plataforma. Como tal, sería necesario considerar como un primer paso diseñar un proceso apropiado para distinguir entre los objetos exteriores a la plataforma y los que se encuentran dentro de ella. De esta forma sería más fácil el análisis acotado por el área deseada, minimizando los errores mencionados en el caso cuatro de la sección 6.

Sería recomendable intentar mejorar el aspecto lógico de la aplicación además del alcance de la misma. En la versión entregada la diversidad de casos en los que la aplicación funcionaría es limitada, ya que sólo se utilizan triángulos, cuadrados y círculos perfectos. Convendría añadir figuras con otras formas, formas geoméricamente no perfectas además de generar un juego de colores para las figuras mucho más variado. Esto garantizaría un mejor reconocimiento de las imágenes. Además, esta última modificación permitiría, en un futuro, realizar un análisis de la composición de los colores de las figuras en un rango mayor que el de la versión actual. Así se podría generar una colección de valores RGB otorgando a la aplicación un conjunto de datos de consulta de colores en el espectro visible a modo de paleta de colores.

Por otro lado, otra variable a considerar, además de las formas geométricas y colores, sería la de la altura o inclinación del dispositivo encargado de capturar las imágenes. Esta cuestión se ha mencionado en la sección 6, gracias al análisis de los resultados obtenidos. También sería interesante tener en cuenta las inversiones térmicas que se suceden en el exterior en horas nocturnas con respecto a las diurnas, con el objetivo de aprovechar la capacidad de la cámara térmica para distinguir contrastes de temperatura en cualquier momento del día. La inclusión de estas variables conduciría sin duda a la obtención de mejores resultados.

Además de las mejoras indicadas, sería interesante comentar que, en el aspecto tecnológico, se podría haber utilizado una base de datos, independientemente de ser relacional o no, para poder almacenar las imágenes recogidas así como la información resultante. Este otro servicio de almacenamiento de información sería vital para la creación de un sistema de procesamiento de información dotado de aprendizaje automático. Esto permitiría, en un futuro, generar criterios de reconocimiento en tiempo real basadas en tasas de acierto en función de las imágenes tomadas con anterioridad.

Por otro lado se deja abierta la viabilidad de generar una interfaz web que sustituya a la implementada en esta versión de la aplicación. Esta interfaz web permitiría una mayor compatibilidad con todo tipo de sistemas y así, una mayor adaptación a cualquier tipo de usuario.

Para ello, además de utilizar nuevas tecnologías, sería interesante la aplicación de nuevos patrones de diseño software. Esto supondría un mayor nivel de abstracción de la aplicación. Se podrían aplicar, entre otros, el patrón Comando ([Gamma and Fernández-Acebal, 2002](#)) que permitiría generar una respuesta más modular por parte de la aplicación ante los diferentes eventos definidos, o el patrón Observer ([Gamma and Fernández-Acebal, 2002](#)) que permitiría mejorar la integración del patrón MVC ([Gamma and Fernández-Acebal, 2002](#)), aportando al Controlador una mayor capacidad de respuesta ante los eventos que se producen.

El presente trabajo se ha desarrollado utilizando [Matlab \(2015\)](#). También se pueden considerar otras herramientas para tal propósito, tal como el uso de las librerías [OpenCV \(2017\)](#) que contienen diversas funcionalidades útiles para el tratamiento de imágenes.

En relación al tema de la conectividad, se podría plantear utilizar un servicio web remoto, que permitiera liberar la funcionalidad de la aplicación para multitud de dispositivos que requirieran de la misma funcionalidad. Además, permitiría una ejecución en tiempo real de la aplicación desde un dispositivo con arquitectura móvil. Sería necesario hacer un estudio de viabilidad para la transmisión de imágenes de alta resolución desde este dispositivo a un servidor con la aplicación de reconocimiento en ejecución.

No sería descabellado pensar que con un UAV dotado de cámaras de mayor calidad e incluso dotado de un sistema de iluminación de la escena, el resultado final de la ejecución de la aplicación sería mucho más satisfactorio. Esto da pie para posibles mejoras en futuras versiones.

7.4. Contribuciones personales

Se describe a continuación el trabajo realizado por cada uno de los componentes del proyecto, haciendo referencia tanto al desarrollo de la memoria, organización del trabajo y toma de decisiones, implementación de la aplicación o realización de pruebas.

7.4.1. Luis Arroyo Lara

A principios de octubre concretamos con el director del proyecto, una reunión para conocer de mi primera mano el proyecto a desarrollar durante el curso, en este primer encuentro conocimos el objetivo del proyecto y las principales líneas que tendríamos que seguir durante el proyecto, también tomamos decisiones sobre el desarrollo, como por ejemplo el lenguaje programación sobre el que desarrollaríamos nuestro proyecto, [Matlab \(2015\)](#).

Una vez concluida la primera reunión, y ya con una idea general de lo que sería el proyecto creímos conveniente realizar una mínima investigación sobre el procesamiento de imágenes, para ello nuestro director del proyecto nos facilitó información con la que trabajar. Una vez adquirimos los conocimientos necesarios para empezar a trabajar en el proyecto, comenzamos con pequeños desarrollos en [Matlab \(2015\)](#), yo particularmente comencé detectando diferentes colores en imágenes, lo cual nos sería muy útil en el desarrollo del proyecto.

Tras diversas reuniones con el director se concretó nuestra idea de desarrollo para el proyecto y algunos métodos que nos podrían ser útiles para el desarrollo de éste. Tras lo cual concretamos el hecho de reunirnos periódicamente, como mínimo cada dos semanas, con el tutor para mostrar los nuevos avances y que él a su vez pudiera aconsejarnos sobre las técnicas utilizadas.

David y yo a mediados de noviembre comenzamos con la ardua tarea de la limpieza de imágenes, que consistía en eliminar la mayor parte del ruido existente en las imágenes, para posteriormente poder sacar conclusiones sobre éstas. A este trabajo dedicamos varios meses hasta que por fin conseguimos imágenes con el mínimo ruido y sobre las cuales podíamos obtener las figuras geométricas deseadas.

A principios de febrero comencé a escribir la introducción de nuestra memoria, la cual fue cambiando hasta finales de mayo. Junto con mis compañeros Eduardo y David comenzamos a formalizar la motivación y objetivos previstos al comienzo del proyecto, también comenzamos a estructurar el estado del arte que se recopiló al principio del proyecto, este punto se mantuvo en continua evolución hasta finalizar el proyecto. Planteado el proyecto realice el plan de gestión de riesgos ([Alonso et al., 2005](#)) del mismo, este punto no me resultó especialmente difícil gracias a los conocimientos adquiridos durante la carrera, dichos conocimientos me facilitaron mucho el reconocimiento de estos riesgos y las medidas a tomar antes ellos según fueran apareciendo. Por último, a lo que memoria se refiere, también trabajé junto a mi compañero David en los resultados finales del proyecto, y en las conclusiones sobre estos resultados.

Con el algoritmo de limpieza bien establecido, decidí junto con mi compañero David comenzar a trabajar en la obtención del color de una figura en concreto, este trabajo resultó algo más sencillo gracias a los conocimientos que adquirí al inicio del proyecto, durante el trabajo previo de investigación. Una vez fuimos capaces de reconocer las figuras geométricas en el campo visible e infrarrojo, y el color de las figuras en el campo visible, el siguiente paso requería hacer uso de un algoritmo de decisión planteado por mi compañero Eduardo y el cual yo implementé. Llegados a este punto, sólo quedaba ser capaces de mostrar al usuario los resultados obtenidos, para lo cual diseñé y desarrollé una interfaz fácil e intuitiva con la que usuario pudiese comprobar los resultados de nuestro trabajo en este proyecto. Por último, quedaba integrar todo el trabajo realizado de lo cual me encargue con la ayuda de mi compañero David.

En relación a la elaboración de la memoria, una vez finalizada la parte técnica, sólo nos quedaba finalizar los puntos de anexos, referencias bibliográficas y conclusiones, dichos puntos fueron realizados juntos con mis compañeros David y Eduardo. Por ultimo dimos formato a la memoria en LaTeX.

7.4.2. David Caballero Ruiz

Fue a principios de octubre, con el comienzo del curso, cuando decidimos comenzar con el desarrollo del Trabajo de Fin de Grado que se iba a abordar. Después de la primera reunión con el director, empezamos a plantearnos y comprender el alcance del proyecto y los principales objetivos en los que éste se basaba. Además, el director nos ofreció toda la información de la que disponía, entre ella el proyecto precedente al nuestro [Álvarez-Piñeiro \(2016\)](#), para facilitar lo que para mí resulta ser lo más complicado de un proyecto, su inicio.

También fue en esta primera reunión en la que decidimos el lenguaje de desarrollo a utilizar, [Matlab \(2015\)](#).

Para empezar, gracias a la información proporcionada por nuestro director, realizamos una investigación sobre las tecnologías actuales de tratamiento de imágenes con el fin de adentrarnos más en dicho campo y conocer de primera mano cuán desarrollado está el procesamiento de imágenes en la actualidad. Investigamos de forma individual sobre las distintas opciones con las que resolver nuestro problema para después poner en común la información. Este trabajo de investigación me acercó personalmente al proyecto, ya que empecé a comprender el hilo que tendríamos que seguir.

Una vez conocido el proyecto que íbamos a desarrollar, decidimos reunirnos al menos una vez por semana para contrastar ideas e hipótesis con el objetivo de encontrar la mejor manera de afrontarlo. En este punto, me encargué de organizar la forma de trabajo a seguir, cosa que fue fácil debido a que ya hemos trabajado juntos años atrás y conocemos la forma de trabajar de cada uno. Por otro lado, empezamos a formalizar las primeras decisiones, como la utilización finalmente de ([Matlab, 2015](#)) como lenguaje de desarrollo ya que proporciona una gran multitud de métodos y funciones aplicables al procesamiento de imágenes, algunas de ellas conocidas en cierta medida por Eduardo y por mi tras haber cursado Percepción Computacional el año anterior, o la plataforma desde la que organizaríamos toda la información encontrada y desarrollada, Google Drive.

Desde el principio tanto Eduardo como Luis sabían que yo era el miembro del proyecto con más tiempo, pero esto no fue un problema ya que supimos repartir el trabajo de forma equitativa y teniendo en cuenta las horas disponibles de cada uno. Luis y yo nos encargamos en gran medida de la parte de desarrollo de la aplicación, mientras que Eduardo estaba más enfocado en el progreso de la memoria. En concreto, me encargué de organizar la línea de trabajo a seguir durante el transcurso del año.

Tras unas primeras pruebas con [Matlab \(2015\)](#), empezamos a pensar cuál sería la mejor manera de reconocer la plataforma y qué diseño debería tener la misma. Partiendo de la base de que debía ser una superficie lo suficientemente grande para que un UAV la localizara y debía contener figuras que la identificaran, no siendo capaces de encontrar estas figuras en la naturaleza, concluimos junto con nuestro director que las figuras serían geométricas, metálicas y con colores que no se confundieran en el entorno real. Por lo tanto, se me ocurrió utilizar, para realizar las primeras pruebas, las tapas de latas de conserva. Con estas primeras figuras aplicamos los métodos que habíamos encontrado y combinado para comprobar el efecto que causaban sobre las imágenes. Desde este momento, el procesamiento de identificación de la plataforma no dejó de evolucionar hasta lograr los resultados que han sido expuestos. Me encargué de encontrar cinco figuras metálicas de un tamaño parecido y cuya forma estaba perfectamente definida, lo que nos facilitó mucho la identificación de las mismas. Además fui el encargado de conseguir la plataforma final sobre la que realizaríamos todas las pruebas.

En el mes de noviembre decidimos empezar con la implementación de la aplicación. Luis y yo comenzamos a realizar pruebas sobre las figuras de las que disponíamos, con los métodos de tratamiento de imágenes que habíamos recopilado. Después de varios meses y varias pruebas, encontré una combinación de métodos que funcionaba de una forma aceptable a la hora de limpiar el ruido de la imagen y aislar las figuras geométricas de la base de la plataforma en ambos tipos de imágenes. Conseguí este éxito después de varias horas de esfuerzo y dedica-

ción única y exclusivamente a la aplicación. Una vez hecho esto simplemente era necesario identificar las figuras reconocidas, tarea de la que me encargué, utilizando uno de los métodos que había encontrado Eduardo en las primeras investigaciones. Tras varios días de trabajo analizando propiedades obtenidas de cada una de las figuras, conseguí obtener ciertos valores entre los que en los que las figuras quedaban correctamente identificadas.

En febrero comenzamos a escribir la memoria de una forma más formal. Me encargué personalmente de escribir el resumen y la estructura de la memoria y revisar continuamente el trabajo de mis compañeros. Junto con el director recopilamos la motivación y objetivos en los que se basaba el proyecto. Una vez escrita esta documentación, Eduardo y yo procedimos a recopilar todos los métodos que habíamos aplicado en el desarrollo de la aplicación, para describirlos de forma teórica y obtener una mayor comprensión de todos ellos. Por otra parte, Luis y yo realizamos un análisis de la evolución que habían seguido las plataformas y figuras utilizadas, así como las causas y consecuencias de esta evolución. Por último, añadí a la memoria una sección en la que se describe la forma de implementar los métodos aplicados con [Matlab \(2015\)](#) y las metodologías software.

Una vez la memoria ya había cogido forma, decidimos que lo siguiente sería perfeccionar la aplicación y me adentré con Luis en la tarea de obtener el color de las figuras reconocidas. Fue mucho más fácil de lo que pensábamos debido a las pruebas que realizó Luis al inicio del proyecto. Con otro problema solucionado, únicamente faltaba por aplicar el patrón Modelo-Vista-Controlador ([Gamma and Fernández-Acebal, 2002](#)) en el proyecto e implementar el algoritmo de decisión con el que se calcularían los porcentajes de acierto de cada imagen, tarea que realicé junto a Luis. Por último, mientras Luis desarrollaba una interfaz sencilla a la que se fueron añadiendo nuevas funcionalidades para dar más opciones de prueba al usuario, aproveché para realizar una limpieza de todo el código y añadí comentarios, para facilitar su posterior comprensión si es preciso.

Gracias a este avance en la aplicación y a la información teórica ya recopilada en la memoria, Eduardo y yo pudimos, en el mes de mayo, añadir una de las secciones más importantes de la memoria. En esta sección explicamos el funcionamiento de los procedimientos de limpieza de las imágenes térmica y visible, el posterior procesamiento de las figuras y el algoritmo de decisión del que se extraerían los porcentajes finales.

Por último y una vez la aplicación quedó cerrada, procedimos a recoger todos los resultados obtenidos para sacar conclusiones y posibles futuras líneas de investigación. Además, realicé el anexo en el que se puede comprobar la evolución de las pruebas de campo desde el comienzo. Por último, procedí a traducir al inglés las secciones indicadas y realicé una revisión exhaustiva de la memoria al completo. Concluimos el trabajo dando formato a la memoria en LaTeX.

7.4.3. Eduardo Puerro Cepa

Cuando el director del trabajo nos planteó el tema del TFG a tratar coincidimos en que a principios de octubre sería el mejor momento para empezar a desarrollar la línea de estudio del proyecto.

En octubre, cuando tuvimos la primera reunión con el profesor, vimos el alcance del proyecto en lo que se refiere a objetivo y aplicaciones. Además, entendimos que la mejor opción para

implementar la idea del proyecto sería el uso del entorno [Matlab \(2015\)](#) debido a las múltiples funciones que ofrece para el reconocimiento de imágenes.

Aunque en un principio la labor de desarrollo de la aplicación fue compartida entre los tres componentes del grupo, a la mitad del transcurso del proyecto la carga de trabajo de desarrollo la volcamos más en mis compañeros David y Luis. A pesar de esto, durante el desarrollo de la aplicación, me impliqué de forma activa en el análisis de los resultados que la aplicación devolvía. De esta manera, diseñé un algoritmo que indicaba la decisión y porcentaje final de acierto de la aplicación. Además trabajé en el análisis de los criterios de definición de figuras y formas necesarios para la detección de elementos dentro de las imágenes. A parte de esto, mi esfuerzo estuvo un poco más enfocado en la construcción de la memoria y en el soporte a mis dos compañeros en otras tareas de desarrollo, como pruebas sobre imágenes de nuevas implementaciones.

Las primeras semanas de octubre, después de la primera reunión con el profesor, las dedicamos, entre otras cosas, a la organización del trabajo a realizar, cómo realizaríamos la división del trabajo y cómo gestionaríamos la comunicación entre el grupo y el director de proyecto.

Por otro lado empezamos una fase de investigación, en la que tendríamos que estudiar las técnicas de procesamiento de imagen para, posteriormente, plantearnos qué implementación sería más idónea para la realización del proyecto. Revisamos la documentación existente acerca de este tema. Además, pudimos ver, que una gran parte del temario impartido durante la asignatura Percepción Computacional por el director del proyecto nos serviría directamente para el desarrollo del proyecto.

En el momento en el que observamos la cantidad de materia que nos podría ser de utilidad para el proyecto empezamos a plantearnos de qué manera reconoceríamos la plataforma. La idea principal era que teníamos que reconocer una base con figuras superpuestas en ella. Empecé utilizando los métodos que conocíamos para comprobar qué efecto causaban en las formas que se observaban dentro de las imágenes. A pesar de ello, durante todo el desarrollo de proyecto hubo una evidente evolución en el reconocimiento de imágenes mediante limpieza de ruido.

Además, y ya habiendo comenzado la implementación de la aplicación, se empezó a redactar la memoria. Debido a la leve evolución de la aplicación, la memoria empezó de forma lenta aunque continuada. No fue hasta marzo cuando el desarrollo de la memoria tomó una dimensión considerable.

La investigación acerca de tecnologías y metodologías usadas anteriormente en trabajos del mismo campo me supuso un trabajo considerable ya que existen numerosas aplicaciones de esta tecnología. Este conjunto de técnicas lo plasmé en la sección del estado del arte y, gracias a éstas, pudimos entender mejor el funcionamiento del reconocimiento digital de imágenes y nos dio conocimientos para mejorar nuestra aplicación.

Durante el mes de abril empecé a documentar los requisitos de la aplicación, ya en su vertiente hardware como en su vertiente software, además de generar un diseño de la misma. Este diseño incluye diagramas de clase con las clases más significativas, casos de uso, diagrama de actividad de los casos de uso más significativos y un diagrama explicativo del funcionamiento de la aplicación. En cierta manera esto nos ayudó a darle una visión aún más abstracta y, en ciertos momentos, mejorar la estructura de la aplicación.

Además fue en este mes cuando realicé, junto a mi compañero David, un estudio de los métodos que estábamos utilizando en la ejecución de la aplicación, explicando sobre todo su faceta más teórica. Esto supuso un nivel de inmersión más profundo en el temario, ya que nos obligaba a tener una mayor asimilación de los conceptos que, a priori, no nos parecían sencillos.

Gracias a esta recopilación de información teórica de los métodos de procesamiento digital de imágenes, pudimos, en el mes de mayo, generar una sección que nos permitiera explicar estos conceptos aplicados al software que estábamos desarrollando. De la misma manera que el estudio funcional de la aplicación nos ayudó al desarrollo del programa, el estudio de las bases teóricas que utilizábamos nos impulsó para poder avanzar, cosa que permitió que mis compañeros David y Luis mejoraran el sistema de reconocimiento de imágenes, eliminando obstáculos visuales como las sombras sobre la plataforma.

Durante este mes volví a revisar la documentación funcional y consideré realizar un caso de uso específico para la decisión final de la aplicación tras el tratamiento de las imágenes. Esto me llevó a realizar un boceto en pseudocódigo que se implantó en la aplicación.

Finalmente, a finales del mes de mayo mis compañeros y yo aprovechamos para finalizar la memoria, las partes que más modificamos fueron las de anexos, referencias, agradecimientos y conclusiones, además de formatear la memoria al completo con LaTeX.

Referencias

- F. Alonso, L. Martínez, and J. Segovia. Introducción a la ingeniería del software: Modelos de desarrollo de programas. *Delta Publicaciones*, 2005.
- E. Álvarez-Piñeiro. Reconocimiento de una plataforma para aterrizaje de uav mediante procesamiento de imágenes. *Dpt. Ingeniería del software e inteligencia artificial. Facultad de Informática. Universidad Complutense de Madrid. E-Prints Complutense.*, 2016. Disponible on-line <http://eprints.ucm.es/> (accedido junio 2017).
- Amazon. Programa amazon prime air, 2015. Disponible on-lin <https://www.amazon.com/> (accedido junio 2017).
- Bradley. Bradley local image thresholding., 1986. Disponible on-line <https://es.mathworks.com/matlabcentral/fileexchange/40854-bradley-local-image-thresholding?requestedDomain=www.mathworks.com>. (accedido junio 2017).
- Canny. Canny edge detector., 1986. Disponible on-line https://en.wikipedia.org/wiki/Canny_edge_detector. (accedido junio 2017).
- A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi. A vision-based guidance system for uav navigation and safe landing using natural landmarks. *Journal of intelligent and robotic systems*, 57(1-4):233, 2010. Disponible on-line [http://cmapspublic.ihmc.us/rid=1K7WQVGMRY19F081YKQ/A %20VisionBased %20Guidance %20System %20for %20UAV %20Navigation.pdf](http://cmapspublic.ihmc.us/rid=1K7WQVGMRY19F081YKQ/A%20VisionBased%20Guidance%20System%20for%20UAV%20Navigation.pdf) (accedido junio 2017).
- J.M Cruz, D. Sánchez, and G. Pajares. Sistema de aproximación a una plataforma de un vehículo no tripulado mediante análisis visual. *Universidad Complutense de Madrid. Patente nº 201001592.*, 2013.
- FLIRONE. Flir one (2017)., 2017. Disponible on-line <http://www.flir.com/flirone/content/?id=81730>. (accedido junio 2017).
- E. Gamma and C. Fernández-Acebal. *Patrones de diseño (2002).*, 2002.
- J.A. Garcia-Pulido, G. Pajares, S. Dormido, and J.M. Cruz. Recognition of a landing platform for unmanned aerial vehicles by using computer vision-based techniques. *Expert Systems with Applications*, 76:152–165, 2017.
- J.A. García-Pulido. Reconocimiento de una plataforma para aproximación de vehículos no tripulados mediante técnicas de visión por computador. *Universidad Nacional de Educación a Distancia. Universidad Complutense de Madrid.*, 2014.
- M.E. Gil Fernández. Desarrollo de un sistema de aterrizaje autónomo sobre una plataforma de carga para un uav tipo parrot ar. drone 2. *Trabajo Fin de Grado Dpt. de Automática, Ingeniería Electrónica e Informática Industrial. Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid.*, 2016. Disponible on-line http://oa.upm.es/43909/1/TFG_MARCOS_ENRIQUE_GIL_FERNANDEZ.pdf (accedido junio 2017).

- S. Lange, N. Sünderhauf, and P. Protzel. Autonomous landing for a multirotor uav using vision. In *SIMPAR 2008 Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots*, pages 482–491, 2008. Disponible on-line <https://www.tu-chemnitz.de/etit/proaut/rsrc/simpar08.pdf> (accedido junio 2017).
- Matlab. Matlab-the matworks (2015b)., 2015. Disponible on-line <https://es.mathworks.com/> (accedido junio 2017).
- Moore. Moore-neighbor tracing., 2015. Disponible on-line http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/_contour_tracing_Abeer_George_Ghuneim/moore.html (accedido junio 2017).
- OpenCV. *OpenCV.Open Source Computer Vision.*, 2017. Disponible on-line <http://opencv.org> (accedido junio 2017).
- N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- G. Pajares and J.M. Cruz. *Visión por computador: imágenes digitales y aplicaciones*. Ra-Ma, 2007a.
- G. Pajares and J.M. Cruz. Ejercicios resueltos de visión por computador. Technical report, 2007b.
- Python. Python (2017a)., 2017. Disponible on-line <https://www.python.org/> (accedido junio 2017).
- SALACOM. Sistema autónomo para la localización y actuación ante contaminantes en el mar (dpi2013-46665-c1). *Grupo ISCAR. Ministerio de Economía y Competitividad*, 2013.
- S. Saripalli, J.F. Montgomery, and G.S. Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *Robotics and automation, 2002. Proceedings. ICRA'02. IEEE international conference on*, volume 3, pages 2799–2804. IEEE, 2002. Disponible on-line http://www.researchgate.net/publication/2488105_Visionbased_Autonomous_Landing_of_an_Unmanned_Aerial_Vehicle/file/5046351c7ada50dfe8.pdf (accedido junio 2017).
- J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000.
- R.E. Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013. Disponible on-line <http://rob.schapire.net/papers/explaining-adaboost.pdf>. (accedido junio 2017).
- C.S. Sharp, O. Shakernia, and S.S. Sastry. A vision system for landing an unmanned aerial vehicle. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1720–1727. Ieee, 2001. Disponible on-line <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.7707&rep=rep1&type=pdf> (accedido junio 2017).
- Siport21. Siport21. presentado el “sistema de predicción de operación de vehículos no tripulados en plataformas navales” en desarrollo para defensa., 2016. Disponible

on-line <http://es.siport21.com/news/presentado-el-sistema-de-prediccion-de-operacion-de-vehiculos-no-tripulados-en-plataformas-navales-en-desarrollo-para-defensa/> (accedido junio 2017).

P. Viola and M. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. Disponible on-line <http://www.vision.caltech.edu/htmlfiles/EE148-2005-Spring/pprs/viola04ijcv.pdf> (accedido junio 2017).

Anexo I : Pruebas de campo

En este primer anexo ha sido presentada la evolución que han sufrido las diferentes plataformas durante el transcurso del desarrollo del proyecto. Se describe la composición de cada una de las plataformas y las condiciones en las que se ha capturado la imagen.

■ Interior1

En la figura 53 puede observarse la primera plataforma diseñada. La base de esta plataforma es de cartón y en ella se encuentran las cuatro primeras figuras, dos rectángulos de metal y dos círculos de lata. Esta imagen fue tomada en interior con una lámpara eléctrica simulando la luz del sol.

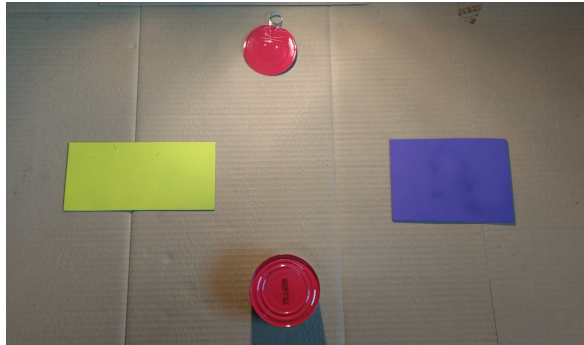


Figura 53: Prueba de campo interior uno.

■ Exterior1

Se puede observar en la figura 54 una segunda plataforma. Dicha plataforma está compuesta por una base de corcho pintado de negro y cinco figuras. En esta prueba se ha descartado el rectángulo amarillo anterior y se han añadido un triángulo y cuadrado metálicos, mientras que los dos círculos de lata continúan formando parte de la plataforma. Esta imagen se capturó en el exterior a primera hora de la tarde para comprobar el efecto del brillo del sol en las nuevas figuras.

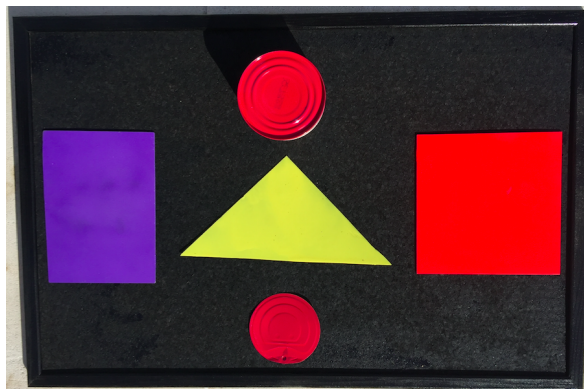


Figura 54: Prueba de campo exterior uno.

■ Exterior2

La base de esta plataforma está compuesta por una lámina de etilvinilacetato, comúnmente conocido como goma eva. Como se puede observar en las figuras 55 y 56, se realizaron dos pruebas, una con el fondo blanco y otra con el fondo negro. Para realizar esta prueba se decidió cambiar uno de los círculos de lata por el rectángulo metálico amarillo, debido a que el sol calentaba considerablemente mejor las figuras metálicas que las de lata. Esta imagen fue tomada en el exterior a primera hora de la tarde, para comprobar el efecto de luces y sombras en distintas horas del día.

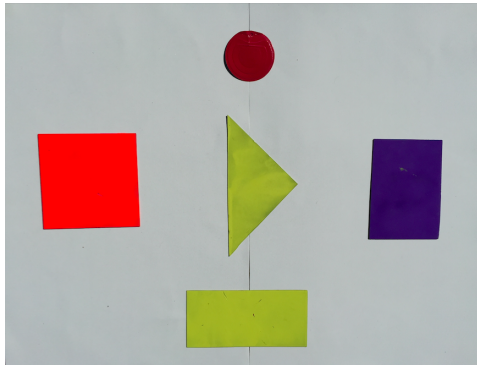


Figura 55: Prueba de campo exterior dos blanca.

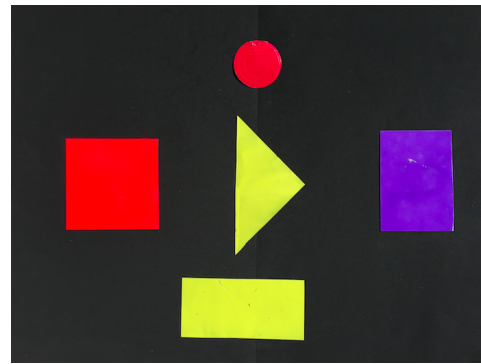


Figura 56: Prueba de campo exterior dos negra.

■ Exterior3

En la figura 57 se presenta la plataforma que finalmente se utilizó para el desarrollo del proyecto. La base de la plataforma consta de una madera de okumen de 80x80cm, pintada de blanco. Se decidió que la base fuese de este material debido a su ligereza y fácil manipulación. En ella, se han colocado cinco figuras metálicas de diferentes colores, eliminando las figuras de lata. Gracias a ello, ha sido mucho más sencillo el reconocimiento de la plataforma tanto en imágenes térmicas como a color. Esta imagen fue capturada en el exterior a primera hora de la tarde.

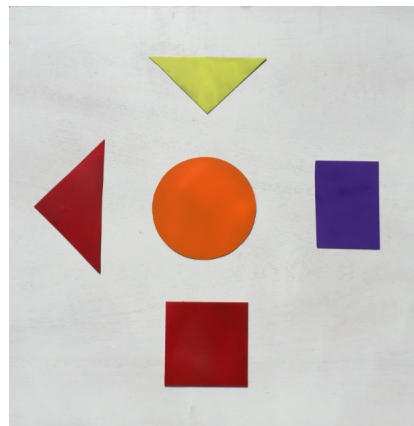


Figura 57: Prueba de campo exterior tres.

Anexo II : Manual de uso de la aplicación

En este anexo se detalla el funcionamiento de la aplicación para comprender el correcto uso de las distintas funcionalidades que se ofrecen.

La aplicación cuenta con una única pantalla en la que se realizan las diferentes interacciones con el usuario.

Pantalla de inicio o reset

Como se puede observar en la figura 58, el usuario puede acceder a las siguientes opciones:

- **Cargar imagen térmica**

Mediante el botón ‘Cargar’ situado debajo del texto “*Imagen térmica*” se puede seleccionar una imagen capturada con la cámara térmica disponible. Esta imagen debe estar en escala de grises para el correcto funcionamiento de la aplicación.

- **Cargar imagen visible**

Mediante el botón “Cargar” situado debajo del texto “*Imagen visible*” es posible seleccionar una imagen RGB tomada con una cámara de esta naturaleza.

- **Comenzar simular**

Mediante el botón “Comenzar Simular” se da comienzo a la simulación de una batería de imágenes, cuyo par de imágenes térmicas-visibles han sido capturadas simultáneamente para simular el vuelo de un UAV.

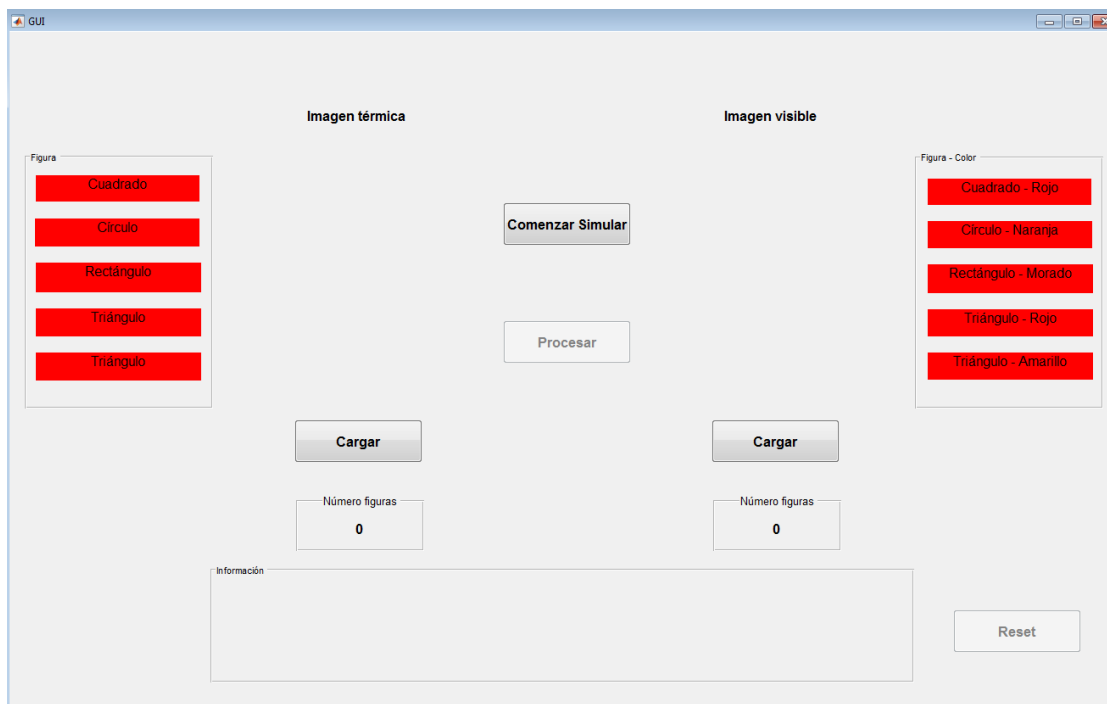


Figura 58: Captura de pantalla inicio-reset.

Carga y procesado de una imagen térmica

Una vez pulsado el botón “Cargar” aparece una pantalla de selección de imagen, como la mostrada en la figura 59.

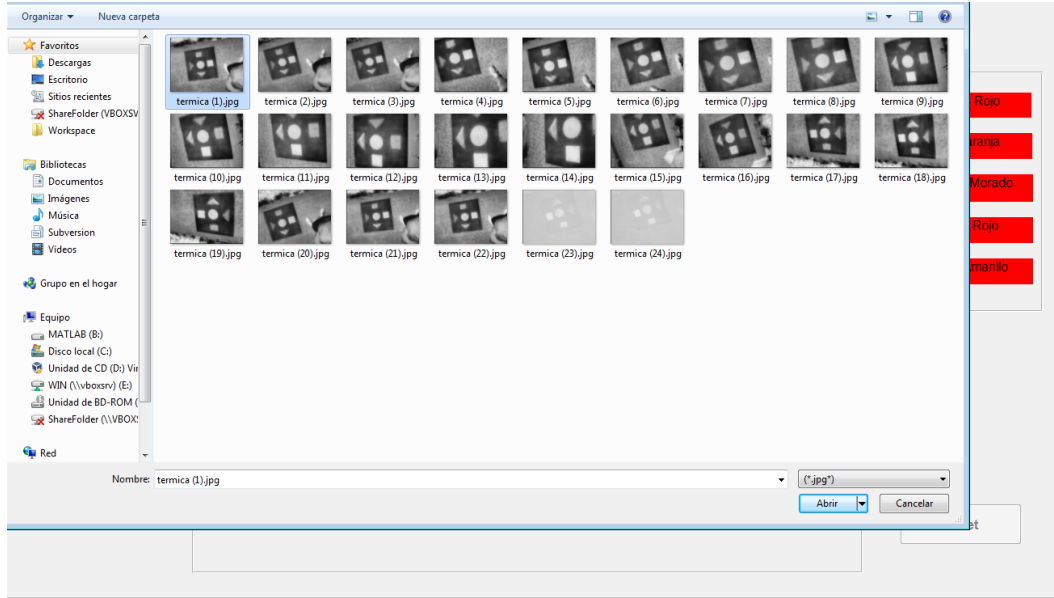


Figura 59: Captura de pantalla selección de imagen térmica.

En la figura 60 se puede observar la imagen térmica una vez ha sido seleccionada y cargada en la interfaz.

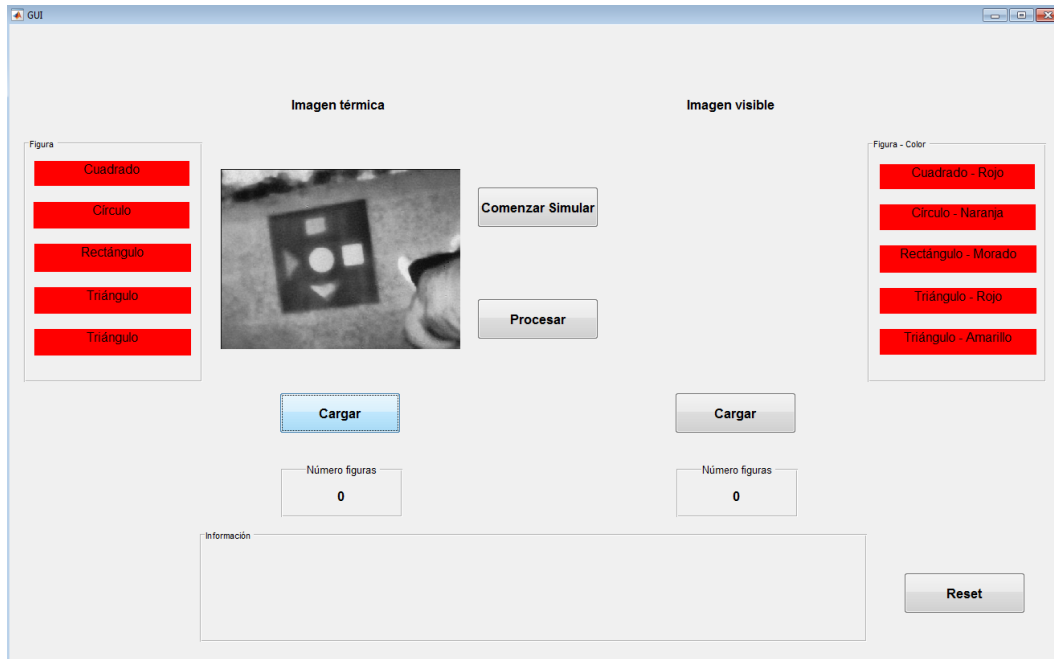


Figura 60: Captura de pantalla carga de imagen térmica.

Al seleccionar el botón “Procesar” la interfaz se actualizará, como se puede ver en la figura 61, y mostrará la siguiente información:

- Las figuras geométricas detectadas en la imagen, que aparecerán bordeadas en color blanco sobre la imagen original.
- El semáforo de figuras en el que se mostraban en rojo todas las figuras antes del procesamiento ahora aparecerán en verde si son detectadas en la imagen.
- El número de figuras detectadas.
- Una breve información en la que se indicará si es la plataforma que se estaba buscando y el porcentaje de acierto en su identificación.



Figura 61: Captura de pantalla procesado de imagen térmica.

Carga y procesamiento de una imagen visible

Dado que la selección y carga de una imagen visible es similar a la selección y carga de una imagen térmica, las figuras 59 y 60 pueden aplicarse a la carga de una imagen visible. En este caso, en la figura 62 se muestra el procesamiento de una imagen visible.

El resultado de procesar la imagen visible respecto al procesamiento de la imagen térmica es mínimo, salvo por dos detalles:

- Las figuras geométricas detectadas en la imagen aparecerán bordeadas en negro sobre la imagen. Decisión detallada en la sección 6.
- El semáforo de figuras ahora aparece acompañado del color de cada una de ellas. De esta forma, si apareciera una figura cuyo color no coincidiese con el color de la figura del semáforo, éste se mantendría de color en rojo.

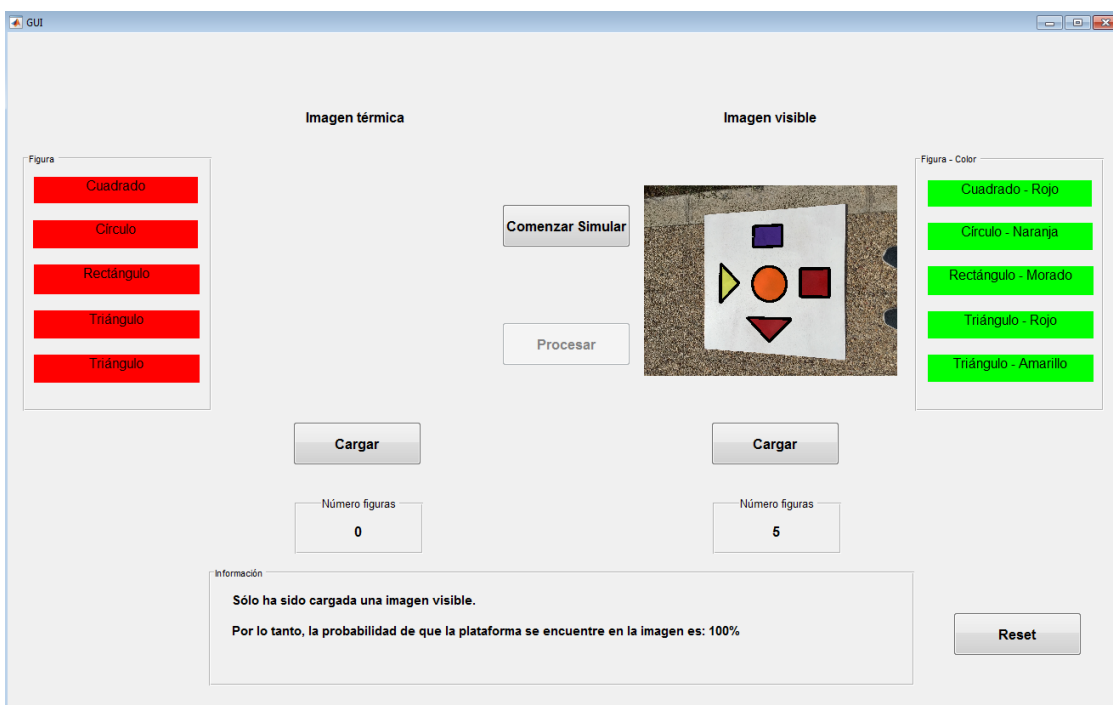


Figura 62: Captura de pantalla procesamiento de imagen visible.

Carga y procesado de una imagen térmica y una imagen visible

La aplicación también permite la carga y procesado simultáneo de la imagen térmica y visible, como puede apreciarse en las figuras 63 y 64 respectivamente. La información que se muestra combina los porcentajes de las imágenes por separado para obtener un porcentaje total de acierto en base a dicha combinación.

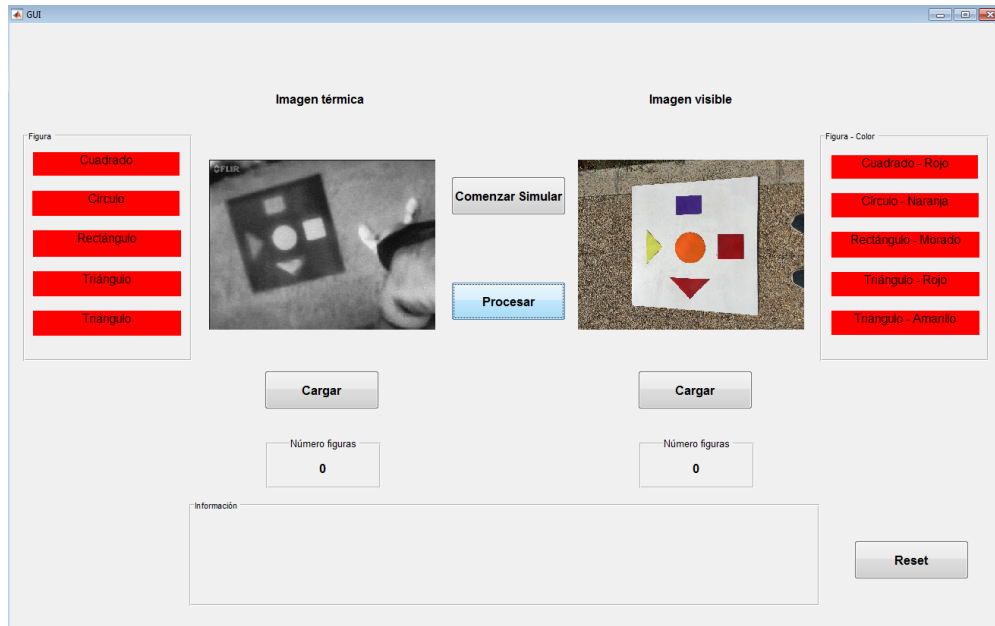


Figura 63: Captura de pantalla carga de imagen térmica y visible.

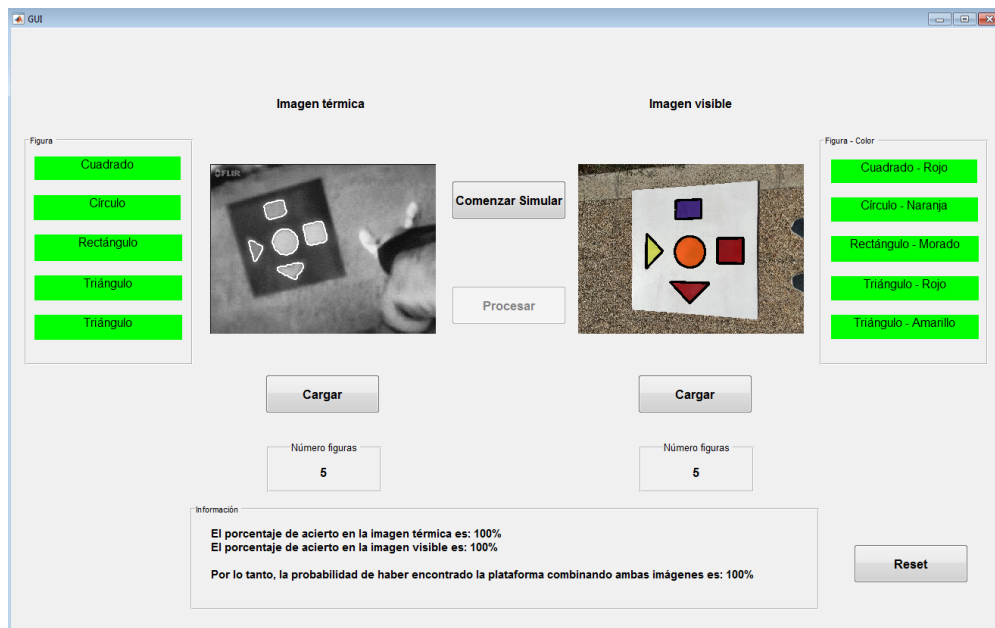


Figura 64: Captura de pantalla procesado de imagen térmica y visible.

Simulación

Al seleccionar el botón “Comenzar simulación”, resultado que puede observarse en la figura 65, aparecerán cargadas una imagen térmica y una imagen visible que corresponden a una fotografía de la plataforma realizada simultáneamente por ambas cámaras. Además, el botón “Comenzar simulación” pasará a indicar un número que identifica la prueba que se está visualizando.

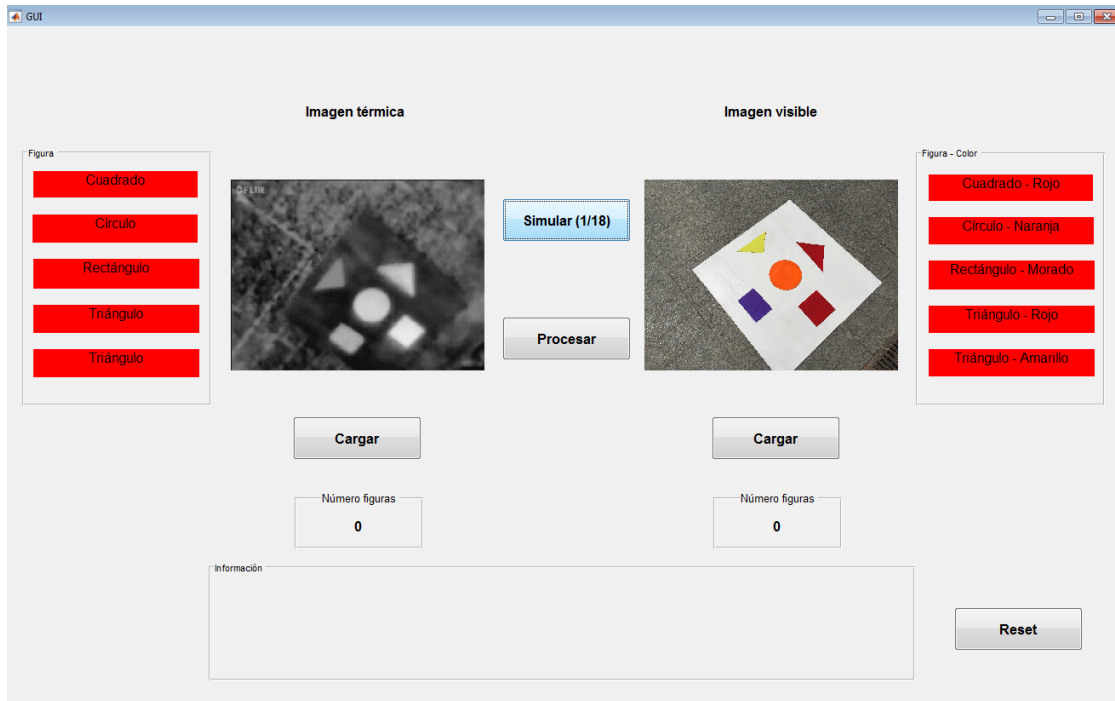


Figura 65: Captura de pantalla simulación.

Al igual que sucede en la figura 64, el procesado que aparece en la figura 66 devuelve el resultado de la combinación de ambas imágenes y en la información se muestra el porcentaje de acierto individual de cada imagen y el resultado final obtenido con la combinación de ambas.

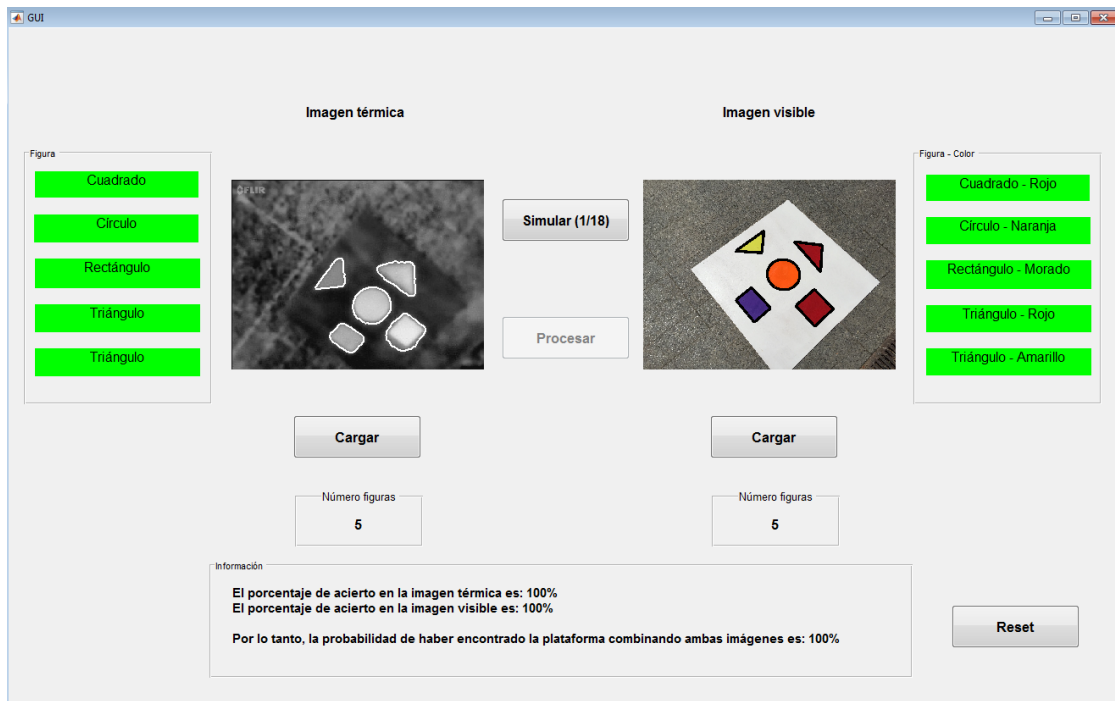


Figura 66: Captura de pantalla procesado de imágenes simuladas.

La aplicación permite simular hasta 18 pares de imágenes. Una vez llegado al máximo, los botones de procesar y simular se deshabilitan, siendo únicamente posible ejecutar las funciones de carga o reset. Esta situación puede verse en la figura 67.

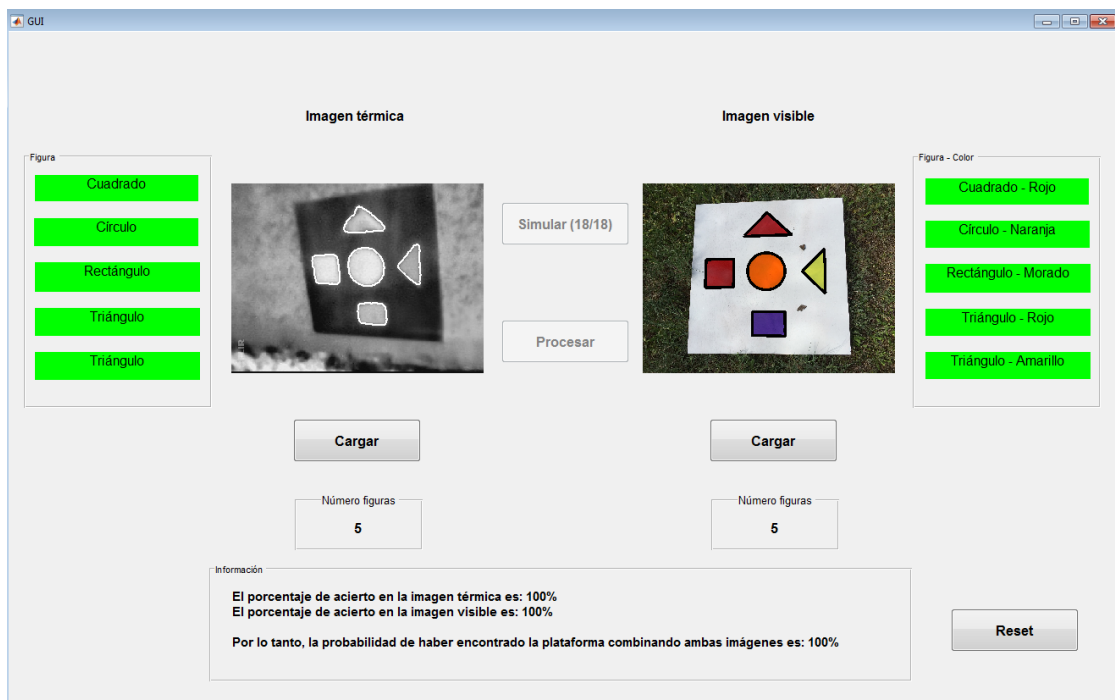


Figura 67: Captura de pantalla fin simulación.

Reinicio de la aplicación

El botón “Reset” reiniciará la aplicación y con ello la batería de simulación. Es posible ver en la figura [58](#) el resultado que se obtendría al pulsar dicho botón.