



Universidad
Zaragoza

Trabajo Fin de Grado

Interfaz gestual de visión infrarroja para control de electrodomésticos.

Infrared vision gesture recognition for domestic appliances control.

Autor

Iñaki Asensio Burgaleta

Director

Don Gonzalo López Nicolás

Departamento de Informática e Ingeniería de Sistemas

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

Noviembre de 2017

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Iñaki Asensio Burgaleta,

con nº de DNI 78758071-E en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado _____, (Título del Trabajo)

Interfaz gestual de visión infrarroja para control de electrodomésticos.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 22 de Noviembre del 2017



Fdo: _____

Resumen

En la actualidad, son muchas las aplicaciones que emplean el reconocimiento gestual como interfaz para la comunicación humano-máquina. Esta modalidad de interacción se ha convertido en una alternativa fiable en campos como el de los videojuegos de realidad virtual, aplicaciones de domótica o incluso en la interfaz de coches de alta gama.

Los electrodomésticos en general presentan una amplia variedad de opciones de uso que pueden conllevar que su control o programación sea tedioso mediante botonera o interfaz táctil. En el caso de los fogones o cocinas de inducción sucede que su interfaz táctil capacitiva no responde de forma adecuada si las manos del usuario no se encuentran limpias. Siendo habitual que el usuario se manche las manos mientras prepara la comida, el uso de la botonera táctil requiere múltiples pulsaciones y re-intentos dificultando por ejemplo la elección del fogón a utilizar o la programación del nivel de potencia deseado. Motivado por lo anterior, se propone el estudio e implementación de una interfaz de usuario basada en gestos aplicada al contexto de las cocinas de inducción. El sistema será capaz de detectar e interpretar los gestos del usuario de forma que se pueda interactuar sin contacto con la placa vitrocerámica.

Como sistema de percepción se va utilizar visión infrarroja. El infrarrojo está en un rango dentro de las diferentes longitudes de onda de la luz, no perceptible para el ojo humano, pero que con las lentes apropiadas puede ser detectada y utilizada, sin resultar molesta para el usuario. La configuración del sistema a desarrollar para la iluminación y adquisición de la imagen infrarroja serán: una fuente de luz infrarroja regulable, constituida por diodos emisores de luz infrarroja colocados de forma que la iluminación se reparta de forma homogénea y una cámara monocromática, a la que incorporaremos un filtro de luz IR para la detección de la longitud de onda deseada. Gracias al circuito de iluminación desarrollado, el sistema será robusto ante cambios de iluminación ambiental.

El desarrollo del código para la adquisición y procesamiento de las imágenes requiere el estudio de algoritmos de segmentación de objetos, así como la identificación de gestos. Se implementará una solución precisa y válida para el microprocesador utilizado, una Raspberry Pi 3 con un sistema operativo Linux y lenguaje de programación Python.

Una vez construido el sistema de visión se comparará la adquisición de datos con respecto a la del sistema comercial de detección gestual "Leap Motion". De esta forma se evaluará la eficiencia del sistema desarrollado a lo largo de este Trabajo Fin de Grado con respecto a las imágenes de un sistema comercial ampliamente reconocido.

El producto final de este trabajo consiste en un dispositivo que implica el desarrollo del "hardware" y del código implementado, así como la evaluación del sistema mediante una demostración experimental en varias cocinas de inducción simuladas con distintos tipos de fogones en las que se probarán diferentes opciones de encendido y variación de potencia de los fogones con la interfaz creada.

Índice General

Resumen.....	3
1. Introducción	5
1.1 Marco de trabajo.....	5
1.2 Motivación	5
1.3 Objetivos	6
2. Estado de la materia.....	7
2.1 Sensores de visión	7
2.2 Leds infrarrojos.....	8
2.3 Detección de gestos	10
3. Diseño y configuración hardware	12
3.1 Requerimientos y componentes del sistema de visión.....	12
3.2 Elección del sensor de visión.....	15
3.3 Elección de la fuente de luz IR.....	16
3.4 Montaje del hardware y guía de usuario	18
4. Detección de gestos	21
4.1 Imágenes adquiridas por el sensor	21
4.2 Extracción del contorno	22
4.3 Detección de manos.....	24
5. Resultados experimentales	27
5.1 Detección de manos en tiempo real	27
5.2 Análisis y discusión	30
5.3 Comparación con dispositivo comercial Leap Motion	32
6. Detección de gestos en cocinas de inducción.....	34
7. Conclusiones y líneas futuras de trabajo	38
7.1 Conclusiones y valoración del trabajo realizado.....	38
7.2 Líneas futuras de trabajo.....	38
8. Bibliografía	40
Anexo 1: Características de los leds infrarrojos y el dispositivo regulador.....	41
Anexo 2: Código del descriptor de contorno e identificación gestual	50
Anexo 3: Código completo del descriptor y el simulador de cocina de inducción	52
Anexo 4: Resultados experimentales.....	57

1. Introducción

1.1 Marco de trabajo

El presente Trabajo de Fin de Grado se ha llevado a cabo en el Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, con la supervisión de Don Gonzalo López Nicolás, profesor en dicho departamento, y dentro del marco de colaboración con la empresa BSH electrodomésticos S.A.

Este trabajo se realiza como continuación de un Proyecto Fin de Master realizado en la misma área de trabajo, presentado en septiembre de 2015 y realizado por Alejandro Inogés Blasco [1] cuyo tema también es el reconocimiento gestual y sus aplicaciones en el diseño de interfaces. En aquel trabajo se utilizó la plataforma comercial “Leap Motion” para la detección y análisis de gestos de la mano. Este dispositivo comercial incluye librerías y software propietario que permiten implementar de forma sencilla el análisis e interpretación de imágenes. En el presente proyecto se pretende profundizar en los fundamentos del sistema comercial mediante el diseño e implementación de un sistema análogo realizado pieza a pieza por el estudiante desde su diseño, montaje, programación y evaluación experimental.

1.2 Motivación

La interacción a través de expresiones corporales se está convirtiendo en una prestación muy valorada en la tecnología de diversos campos, por la ventaja que conlleva realizar cambios en tu entorno sin contacto. El usuario queda sorprendido y se aporta “valor tecnológico” a los productos que la incorporan.

Las nuevas generaciones de aparatos electrodomésticos son cada vez más versátiles, cuentan con más opciones y modos de funcionamiento que hace unos años. Este avance requiere nuevas técnicas de control que permitan un manejo más sencillo y eficiente para cualquier usuario, independientemente de su edad, nivel de conocimientos, etc.

La tecnología actual busca la sustitución de algunas funcionalidades que actualmente se gestionan mediante las interfaces táctiles, por funciones equivalentes que puedan ser controladas por el sistema de visión gestual haciendo posible la reducción del espacio destinado al interfaz táctil. La aplicación de este espacio sobrante permite mejorar las capacidades del electrodoméstico, y con respecto al campo de las cocinas de inducción no solo aumentaría la flexibilidad en la comunicación usuario-máquina, sino que también aumentaría la superficie libre en la vitrocerámica que podría ser destinada a ampliar los fogones.

Por último, es interesante destacar la posibilidad de extender esta tecnología a otras áreas de la casa, no solo a electrodomésticos de la cocina. Se denomina domótica a la tecnología que busca la automatización de la vivienda para aumentar la comodidad e

independencia de los sistemas de seguridad, comunicaciones, gestión energética, etc. Algunos ejemplos de aplicación pueden ser apagar el despertador solo con señalarlo, o hacerse un café sin tener que pasar por la cocina, son estos algunos de los detalles que la tecnología busca desarrollar a costes lo suficientemente asequibles para que sean incorporados a los hogares de todo ciudadano.

1.3 Objetivos

El objetivo principal es la implementación de un sistema de visión infrarrojo que detecte el contorno y dedos de la mano para poder convertir esta información en órdenes para el interfaz de un electrodoméstico (Ilustración 1). En concreto, una placa de inducción. No solo debe registrarse la forma de la mano, sino que debe hacerse un seguimiento de ésta para saber su evolución en el tiempo (video) y su posición en el espacio que ocupa. Además, se simulará el interfaz de una cocina de inducción para poner a prueba la robustez y fiabilidad del sistema, de ese modo se podrán establecer cuáles son las áreas a mejorar en futuras investigaciones.

Para la realización del trabajo se han desarrollado los aspectos que se enumeran a continuación.

1. Estudio de la tecnología de características similares en el mercado tanto en reconocimiento gestual como en iluminación infrarroja.
2. Selección de un sistema de visión y elección de las piezas de la fuente de iluminación regulable, para su posterior diseño y montaje.
3. Desarrollo del algoritmo de detección de manos en la plataforma Raspberry Pi que permita adquirir video en tiempo real de los gestos de la mano de la forma más eficiente posible.
4. Realización de las pruebas pertinentes y comparación del sistema propio con los sistemas comerciales.
5. Implementación de un programa que permita evaluar la aceptación y sencillez del sistema en un simulador de una cocina de inducción.
6. Conclusiones, áreas de mejora y futuras líneas de trabajo.

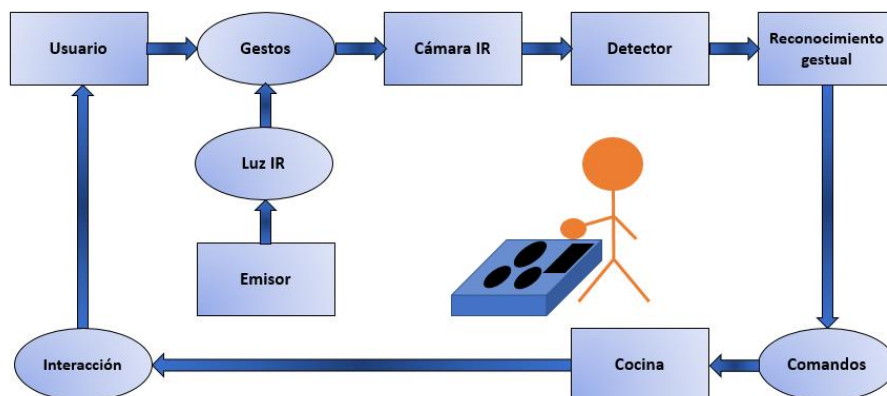


Ilustración 1. Esquema espacial de los elementos que interactúan en el proyecto.

2. Estado de la materia

En este apartado se ha llevado a cabo un estudio de las principales técnicas basadas en materia de reconocimiento gestual y técnicas de percepción. Es importante este análisis de cara a la selección de componentes, ya que el algoritmo final que se desarrolle depende en gran medida del sensor de visión utilizado y sus características, y también de las prestaciones de los diodos LED que oferta el mercado.

2.1 Sensores de visión

Los sensores de visión son dispositivos que utilizan imágenes capturadas por una o varias cámaras para determinar la presencia, orientación y forma de las piezas, pueden crear mapas 2D que detectan la evolución de los movimientos realizados o incluso detectar profundidad mediante técnicas de triangulación con dos cámaras, creando mapas tridimensionales, como es el caso del dispositivo comercial Leap Motion [7].

En la actualidad pueden encontrarse una gran cantidad de estos sensores en el mercado, tanto para el seguimiento/reconocimiento de manos, como para otras aplicaciones como el seguimiento de personas en videovigilancia, o en la industria médica que está desarrollando nano-cámaras para el diagnóstico y tratamiento de cirugías internas.

En la industria de los videojuegos también se están implementando técnicas similares a la que se desea desarrollar en el presente proyecto. La consola Nintendo Wii [2] incorpora un algoritmo que permite la detección y localización de la posición de las manos en tiempo real y la aplica en el control de sus juegos, haciendo de éstos una experiencia dinámica para el usuario (Ver Ilustración 2). Otro dispositivo para aplicaciones de realidad virtual es “Oculus Rift” [3], un casco de realidad virtual desarrollado por la empresa Oculus VR que permite visionar películas y videojuegos en primera persona. La versión comercial de este dispositivo es reciente y los videojuegos que se pueden reproducir deben ser especialmente diseñados para funcionar en Oculus Rift.



Ilustración 2. Mando de la Nintendo Wii y distintos accesorios para hacer del videojuego una experiencia más real.

Dentro de las cámaras que utilizan estos dispositivos de interfaz, las más comunes son las cámaras RGB. Una cámara RGB [4] entrega los tres componentes de colores básicos (rojo, verde y azul) en tres cables diferentes. Este tipo de cámara a menudo utiliza tres sensores CCD independientes para adquirir las tres señales de color. Las cámaras RGB se utilizan para la captura de imagen de color muy precisas. Su fácil adquisición y bajo coste hacen de éstas la alternativa más utilizada en sus diversas formas, existen diferentes tamaños y modelos, desde las que se utilizan en aplicaciones industriales de vigilancia, hasta las que podemos encontrar en los teléfonos móviles actuales.

Por último, en el área de desarrollo del 3D han surgido cámaras capaces de combinar el RGB y la detección de profundidad en las imágenes, cuya combinación es capaz de segmentar y localizar diferentes planos en una misma imagen y clasificar los objetos en función de la distancia al objetivo. Estas cámaras son denominadas RGB-D, el ejemplo más reconocible de este tipo es la Microsoft Kinect que utiliza la consola Xbox (Ilustración 3), la obtención del mapa 3D se realiza mediante un emisor de luz infrarroja que proyecta un patrón de puntos y un receptor que permite triangular y establecer distancias entre objetos mediante la comparación con un patrón calibrado.

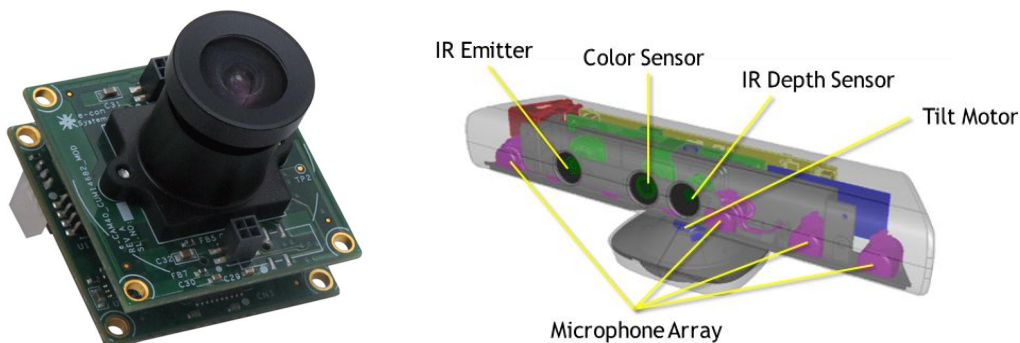


Ilustración 3. (Izq.) cámara RGB con objetivo desmontable utilizada en aplicaciones industriales. (Dcha.) cámara RGB-D Microsoft Kinect utilizada en el mundo de los videojuegos.

Como se puede observar el marco tecnológico en el que se encuadra el proyecto es amplio. Existen una gran cantidad de dispositivos y entornos en los que llevar a cabo un proyecto de estas características, la elección de nuestro sistema de visión tendrá en cuenta parámetros como la sencillez de uso, el coste, y la disponibilidad inmediata para su compra.

2.2 Leds infrarrojos

A continuación, se describen los aspectos del sistema de iluminación para entender mejor el papel de la luz infrarroja en el proyecto. La idea es utilizar algún tipo de iluminación extra sobre el objeto o escena que deseamos enfocar, pero que no se encuentre en el rango de espectro visible del ser humano, ni tampoco en el de los sensores de vídeo convencionales.

Todo lo que somos capaces de ver los humanos, todas las formas y colores, se encuentran en una pequeña porción de espectro electromagnético comprendida entre los rayos infrarrojos y los ultravioleta. Iluminando solamente el objeto de estudio (la mano) con este tipo de luz, vamos a obtener una imagen en la que ésta destaque por encima del resto de elementos del entorno cuando se analice la imagen bajo un filtro que solo deje pasar la franja de luz IR.

El componente que permitirá llevar a cabo esto es el led IR (Light Emissor Diode). Una configuración en la que controlemos varios de estos LEDs es lo que debemos implementar para iluminar la imagen que posteriormente será procesada. Conseguir un nivel adecuado de iluminación será de vital importancia en el resultado final del proyecto.

Existen una gran cantidad de configuraciones y formas de diodos IR, pero analizando las principales distribuidoras de electrónica (Farnell, Mouser, Digikey...) se puede llegar a la conclusión que la gran mayoría de los que existen en el mercado tienen como finalidad ser vistos, en vez de iluminar otros objetos por sí mismos. Deben buscarse LEDs utilizados en aplicaciones de video-vigilancia, con una potencia mayor, denominados LEDs de potencia. Dentro de los distintos tipos de geometrías de estos LEDs de iluminación, pueden encontrarse tres grandes grupos [5] [6]. Los diodos de ojo de buey, diodos cuadrados con patillas y diodos en estrella (Ilustración 4).



Ilustración 4. De izquierda a derecha: diodo en ojo de buey, diodo con patillas y diodo en estrella.

Las geometrías seleccionadas y de las cuales se compraron varias unidades de LEDs son las dos primeras. El criterio utilizado, además de sus características técnicas es la facilidad de soldadura con la placa a la que irán fijados, debe tenerse en cuenta que este tipo de LEDs son muy sensibles a impactos y su pequeño tamaño hace difícil su manipulación (alrededor de 10mm de lado), cuanto mayor sea la superficie de conexión más fácil será trabajar con ellos. En cuanto a las prestaciones eléctricas y de características existe una gran variedad de cada una de las configuraciones, por lo que una vez determinada la potencia necesaria, la variedad de cada familia de diodos es tan grande para cada uno de los elegidos, que no será un problema su localización y encargo en páginas webs antes mencionadas. En el Anexo 1 se describirán en mayor detalle los diodos escogidos en cuanto a potencia, ángulo de iluminación, curvas características, etc.

Para aumentar la vida útil de los LEDs se implementará un sistema de regulación de intensidad para que estos trabajen siempre a potencia nominal, pudiendo regularse y

adaptarse a las condiciones lumínicas del entorno donde esté ubicado el dispositivo de visión. Como se ha dicho anteriormente este tipo de diodos son muy sensibles tanto a impactos como a condiciones eléctricas diferentes a las nominales, por lo que un pequeño error en su alimentación podría degradarlos rápidamente o estropearlos. Es por esto que el regulador de intensidad cobra una gran relevancia al trabajar con este tipo de electrónica tan frágil.

2.3 Detección de gestos

El "Reconocimiento de gestos" es un tema de gran importancia en el área de la computación y la tecnología del lenguaje. Se persigue el objetivo de interpretar gestos humanos a través de algoritmos informáticos. Los gestos pueden ser entendidos como cualquier movimiento corporal que transmite información o expresión no verbal, los más comunes son los que se producen en la cara o en las manos. El reconocimiento de gestos puede ser visto como una manera de que los sistemas informáticos empiecen a interpretar el lenguaje corporal humano, dejando atrás sistemas como las interfaces de usuario de texto o con teclas. El reconocimiento de gestos puede llevarse a cabo con técnicas de visión por medio de cámaras industriales, webcams o dispositivos móviles y procesamiento de imágenes. Esta tecnología no sólo va a reducir el impacto del hardware en el sistema, también aumentará el rango de usos aplicables desde un objeto en el mundo físico a un objeto en el mundo digital, como teclados, ratones, etc. El uso de esta tecnología puede llevar a la creación de una nueva generación de hardware sin necesidad de monitores, como la creación de la pantalla holográfica o la interacción con máquinas a través de pensamientos.

El enfoque de los modelos en 3D se puede utilizar para la creación de modelos volumétricos o esqueléticos (Ilustración 5) que representan el objeto real con una gran fiabilidad, estos modelos han evolucionado en gran medida en la industria de la animación por ordenador. Los modelos 2D no utilizan una representación espacial del cuerpo, porque se obtienen los parámetros directamente de las imágenes o vídeos con una base en los datos de la plantilla.

Hay muchos retos asociados con la precisión y utilidad del software de reconocimiento de gestos. Debido a que está basado en imágenes hay limitaciones derivadas del equipo utilizado y la información no útil que pueda aparecer en el fotograma. Las imágenes o vídeos pueden no estar en condiciones de iluminación homogénea, o desenfocadas. Los antecedentes o rasgos distintivos de los usuarios pueden hacer más difícil el reconocimiento.

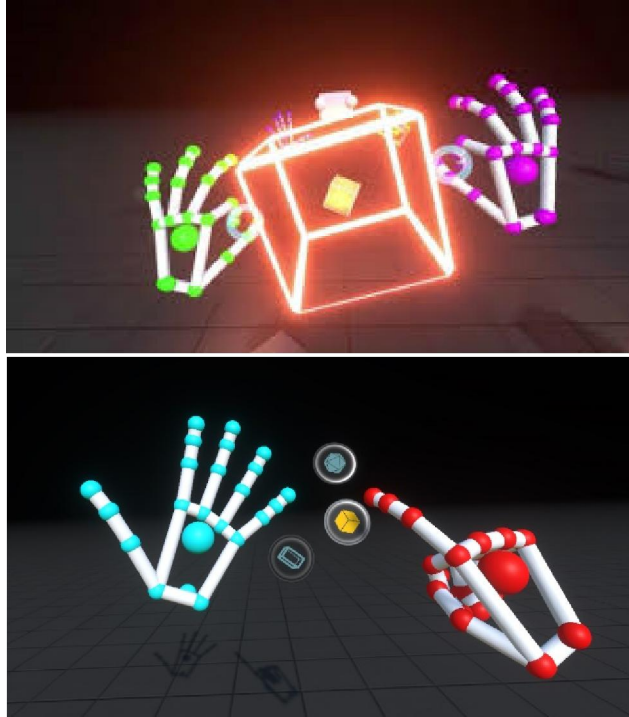


Ilustración 5. Imágenes de un programa de reconocimiento de gestos 3D realizado con el sensor Leap Motion [7] en el que se muestra un modelo esquelético.

3. Diseño y configuración hardware

En este capítulo se describen los componentes que van a formar parte de nuestro sistema de visión, dentro del cual se pueden diferenciar el sistema de iluminación, el sistema de regulación y el dispositivo de captura de video e imágenes.

3.1 Requerimientos y componentes del sistema de visión

Como se ha comentado anteriormente es necesario un sistema que capture imágenes en tiempo real y a partir de ahí sea capaz de su procesamiento mediante un algoritmo. Además, se busca la realización de un sistema compacto y de tamaño suficientemente pequeño para que su colocación y adaptabilidad a los electrodomésticos sea factible.

Por lo anterior, se pensó desde el primer momento en un dispositivo Raspberry pi. Se trata de un microprocesador de bajo coste muy utilizado en el ámbito de la investigación, la enseñanza y con una gran comunidad que desarrolla programas de uso libre (Ilustración 6).

Su uso extendido hace que la Fundación Raspberry pi [8] haya sacado al mercado una gran cantidad de complementos y dispositivos que se pueden alimentar desde la placa y hacer de ella un dispositivo muy polivalente. Existen sensores de temperatura, iluminación, fundas protectoras, displays... y el que más relevancia tiene para el proyecto, cámaras RGB y cámaras RGB infrarrojas.



Ilustración 6. Raspberry pi 3.

Para el desarrollo del regulador de intensidad de los LEDs es necesario que se implemente un circuito que permita de alguna forma controlar desde el micro la intensidad que circula por los diodos, para ello se ha pensado en un dispositivo reductor (Ilustración 7) que permite variar la intensidad del circuito mediante una señal PWM, en función del “duty cycle” de la señal PWM de entrada, la intensidad variará proporcionalmente. De esta manera con duty cycle del 0% los LEDs están apagados y con 100% los LEDs están trabajando casi a potencia nominal (existen unas pequeñas pérdidas producidas por el consumo del “driver” regulador).

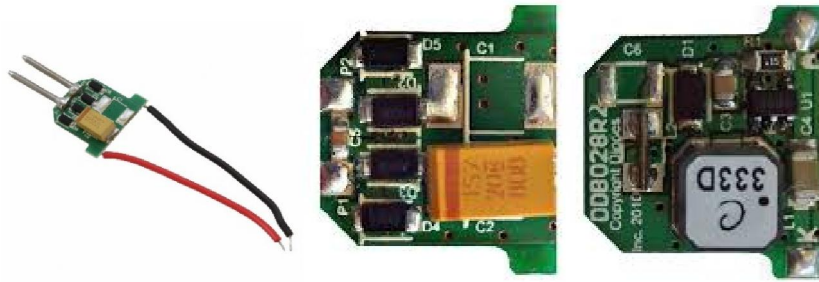


Ilustración 7. Driver AL8807EV2 regulador de intensidad mediante señal PWM [6].

Para ocupar el menor espacio posible se desea diseñar una PCB (“Printed Circuit Board”) donde irán montados los diferentes componentes del circuito. Esta PCB será diseñada y probada en un programa destinado al diseño de componentes electrónicos y simulación de circuitos.

El programa utilizado para diseñar la PCB se llama EAGLE [9], (siglas de Easily Applicable Graphical Layout Editor) muy popular en el entorno de los proyectos electrónicos *DiY* (do it yourself), Existen diferentes versiones de este programa, pero la versión gratuita cuenta con una licencia temporal y gran cantidad de bibliotecas de componentes, que permiten a los usuarios realizar proyectos avanzados a bajo coste.

El diseño de la placa comienza estableciendo los componentes que la van a integrar, como puede apreciarse en la Ilustración 8. Estos son: cuatro tornillos donde colocar unas pequeñas patas a la PCB, otros cuatro tornillos donde se situará la cámara IR, los tres LEDs (cuyas huellas pueden verse en rojo), el resto son el conjunto de fichas donde se conectarán los cables que unen los componentes exteriores que se utilizan (salidas de la Raspberry pi) con el circuito interno de la placa. En la parte superior de ésta pueden verse dos pistas en paralelo una de ellas se trata de Vcc, voltaje nominal de la placa definido a 12V y el otro es la línea de tierra. La placa tiene unas dimensiones de 126x98 mm, las medidas corresponden a las dimensiones necesarias para que los elementos que la formen se puedan disponer sobre ella de forma cómoda.

EAGLE incluye una gran cantidad de bibliotecas donde se pueden encontrar LEDs IR con huellas similares a los usados, no obstante, los modelos escogidos no se encontraban disponibles en la versión gratuita, por lo que fue necesario tomar la huella de un led con características similares y variar las dimensiones para adaptarlas a las necesidades geométricas del requerido.

Posteriormente se envía a fabricar a la empresa “Eurotronics” que comprueba que el circuito es eléctricamente correcto y lo implementa en una placa. Como se compraron dos LEDs distintos, y sus huellas y forma de soldar era distintas se diseñaron dos PCB, una para cada tipo de LEDs.

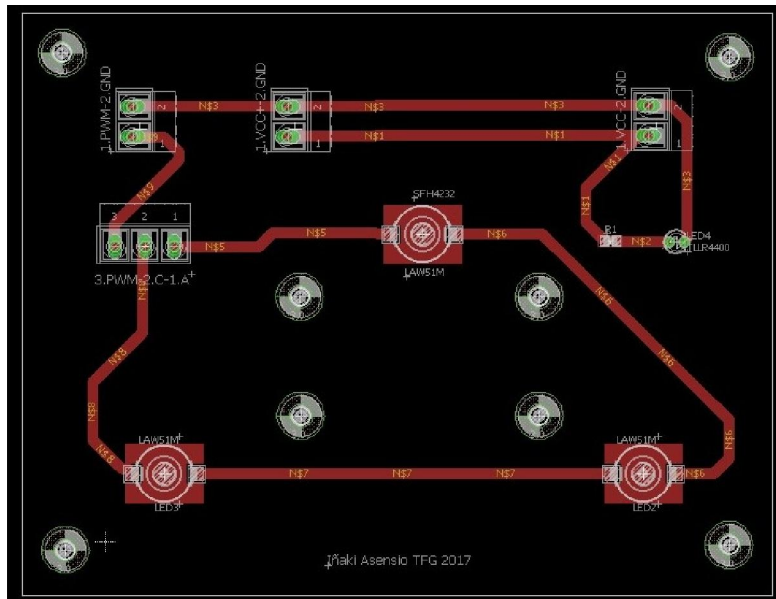


Ilustración 8. Imagen del programa Eagle del diseño realizado de las placas del circuito para los leds de patillas.

A continuación, se muestra el resultado final de una de las placas, la correspondiente al led de patillas (Ilustración 9). Mediante el uso de fichas de plástico es posible la colocación del cableado necesario para su regulación sobre la PCB. También se puede ver como del dispositivo regulador salen tres cables que van a parar al circuito de iluminación, como la luz infrarroja no es visible se coloca un circuito auxiliar con un led piloto que se enciende para indicar al usuario si los diodos están operativos.

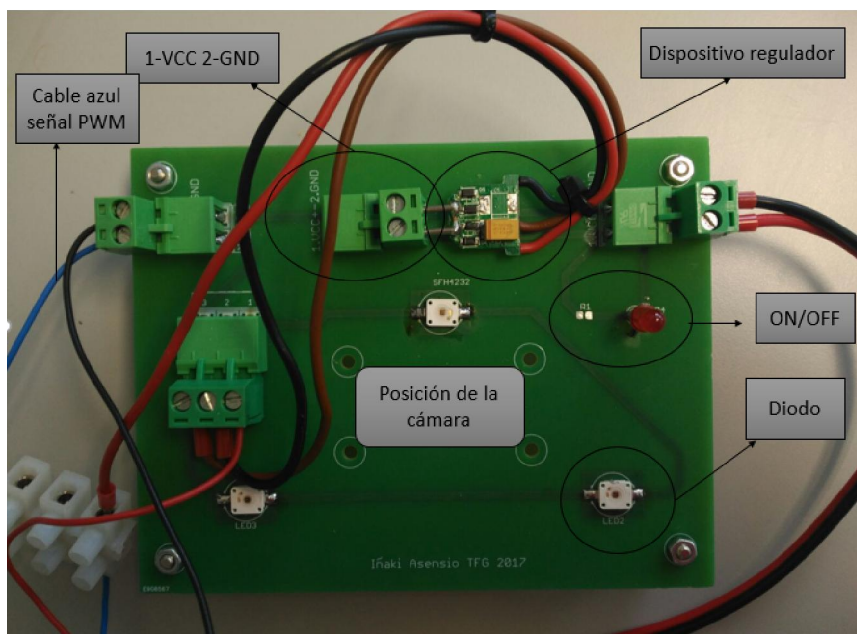


Ilustración 9. PCB con el regulador y los LEDs de patillas ya montados, el cable rojo es el ánodo, el negro el cátodo y por el marrón irá la señal PWM de regulador que se puede ver en la parte superior.

3.2 Elección del sensor de visión

El módulo infrarrojo de la cámara v2 Pi NoIR (Ilustración 10) [8] conectable a Raspberry pi es la selección más adecuada por las justificaciones que se han dado anteriormente. Tiene un sensor Sony IMX219 de 8 megapíxeles, pero no cuenta con filtro de infrarrojos. Esto supondrá que, al realizar fotos de imágenes con luz natural, el efecto de los infrarrojos dará a las fotografías un tono rosáceo, similar al filtro “pastel” disponible en editores de imágenes.

La cámara se controla mediante código Python en el terminal de la Raspberry pi. Existen una gran cantidad de bibliotecas destinadas al control de la cámara. No obstante, la propia Raspberry incluye una como predeterminada, llamada “picamera”, la cual destaca por su sencillez y la gran cantidad de manuales disponibles en la página oficial de la compañía [8]. En caso de necesitar cambiar alguna característica como la resolución, el sentido de la imagen, rotación de la imagen, efectos visuales de iluminación o tiempo de disparo, se puede acceder a las opciones de configuración de la cámara a través del terminal, escribiendo “sudo raspi-config”.

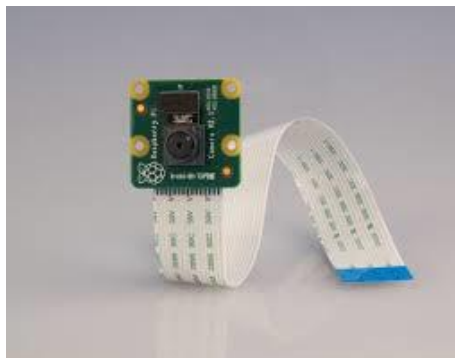


Ilustración 10. Cámara PI NoIR.

En la Ilustración 11 puede apreciarse la diferencia de hacer fotos con la cámara PI NoIR que además de detectar el rango de luz visible, detecta también el rango de longitud de onda infrarroja.



Ilustración 11. Efecto de la luz infrarroja en las fotos de la cámara Pi Noir.

La cámara se une a la Raspberry mediante una entrada CSI o puerto cámara y se puede montar y desmontar con facilidad, lo que aporta flexibilidad al sistema.

Para que la cámara reciba únicamente el reflejo de la luz emitida por los emisores LED implementados, es necesario filtrar el resto de fuentes de luz. Para bloquear todo el rango de luz no infrarrojo o de distinta longitud de onda que la producida por los LEDs (situada en torno a los 850 nm de longitud de onda) se colocará encima del objetivo de la cámara un filtro IR que solo deje pasar el rango deseado. Dicho filtro se asemeja a una lente de forma circular y tonalidad azul oscura (Ilustración 12).

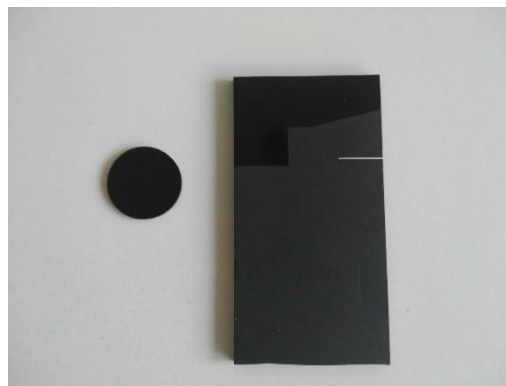


Ilustración 12. Filtro infrarrojo al lado de porción de placa vitrocerámica cedida por BSH.

El sensor de visión ya está bien definido, el coste total de la cámara y el filtro se encuentra entorno a los 25 euros, incluyendo el microprocesador cuyo coste es de 35 euros aproximadamente alcanzamos un precio de mercado total del sistema de visión de 60 euros, por lo que, por un precio económico se esperan conseguir resultados satisfactorios en el proyecto.

3.3 Elección de la fuente de luz IR

Como se ha descrito anteriormente se han escogido dos tipos de LEDs infrarrojos con características eléctricas similares, pero geometrías y formas de soldar distintas. Se han escogido dos tipos para evaluarlos de forma práctica y por si alguno de ellos finalmente no era compatible con las huellas que facilita el programa de diseño de las PCB o si existían unas diferencias de iluminación suficientes como para decantarse claramente por uno de los dos.

El primero de ellos tiene forma cuadrada y patillas a los lados, que facilita el soldado por aire caliente pues al sobresalir las patillas, no hay posibilidad de dañar el led o derretir el plástico protector. Su nombre técnico es "SFH 4232" (Anexo 1) de la empresa OSRAM semiconductores, véase (Ilustración 4. De izquierda a derecha: diodo en ojo de buey, diodo con patillas y diodo en estrella.

El segundo es del tipo ojo de buey, el modelo es “SFH 4715” (Anexo 1), cuyas conexiones se encuentran por debajo del led, lo que dificulta su unión y no se tenía seguridad de que la precisión fuese suficiente para su correcta colocación. Finalmente se colocaron sin daño alguno.

En la Tabla 1 se pueden apreciar sus principales características eléctricas. Son dos LEDs bastante similares en condiciones nominales, salvo por el ángulo de iluminación y una pequeña variación en la potencia de radiación.

(IF=1A, tp=10ms)		
modelo	SFH 4715	SFH 4232
forward current	1000 mA	1000mA
power consumption	1,8 W	1,8W
peak wavelength	860 nm	860 nm
centroid wavelength	850 nm	850 nm
forward voltage	1,5 (<1,8)	1,5 (<1,8)
total radiant flux	630mw	530mw
half angle	45º	60º

Tabla 1. Características eléctricas de los diodos seleccionados para el proyecto.

Los LEDs, como se ha podido observar en ilustraciones de la placa PCB están colocados formando un triángulo isósceles para repartir la iluminación de una forma homogénea imitándose la disposición alargada que se utiliza en otros sistemas similares, pero sin llegar a colocar los 3 diodos en línea, para evitar que ángulo de visión de la cámara capture zonas a las que el “cono” de luz infrarroja no llega. La cámara se colocará en el centro de dicha configuración, no obstante, en caso de no ser iluminación suficiente se podrían conectar ambas placas en serie para tener un total de 6 LEDs infrarrojos trabajando y colocar la cámara entre ambas placas.

Los LEDs trabajarán a potencia nominal cuando la señal PWM proporcionada por la Raspberry tenga un duty cycle del 100%, es decir, sea una señal continua de 3V, valor máximo capaz de suministrar el microprocesador. Para mostrar las diferencias de trabajar a distintos porcentajes se han realizado dos fotos con 25% y 100%. Como puede verse en la foto de la derecha (Ilustración 13), la mano está más iluminada que en la de la izquierda. Existe mayor contraste entre la propia mano y el fondo de la habitación, lo que permite que el algoritmo filtre mejor todas aquellas perturbaciones derivadas del entorno de trabajo.



Ilustración 13. Comparación de imagen con distintas potencias de luz IR (Izq.) foto del visor con duty cycle 25%. (Dcha.) foto con duty cycle 100%.

Aunque la diferencia puede parecer poco apreciable la contaminación e intensidad lumínica tiene una gran relevancia en la correcta interpretación del algoritmo, como veremos posteriormente.

3.4 Montaje del hardware y guía de usuario

Una vez se han definido todos los componentes, para la realización de las pruebas derivadas de este proyecto son necesarios una serie de elementos que no se han detallado hasta ahora. En este apartado no solo se detallará todo lo necesario para la utilización del microprocesador y la PCB, sino que se llevará a cabo un pequeño manual de usuario a nivel de montaje y utilización.

La Raspberry pi no incorpora memoria flash en el momento de su compra, es necesario adquirir también una tarjeta SD para poder almacenar información, una tarjeta de 16GB es más que suficiente para la realización del trabajo. Tampoco viene instalado por defecto ningún sistema operativo en ella, para poder utilizarla se descarga de la página web de la compañía uno de los sistemas operativos compatibles que se ofrecen, el más utilizado, y por tanto la elección escogida, es Raspbian Debian. Raspbian [10] es una distribución del sistema operativo GNU/Linux de uso libre, basado en Debian Jessie (Debian 8.0) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática.

Una vez el sistema operativo está correctamente instalado, éste cuenta con el lenguaje de programación Python que se va a utilizar, instalado por defecto. Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible y limpio.

La conexión del micro se lleva a cabo a la red eléctrica común, y para su manipulación es necesario conectar un ratón, un teclado y un monitor/display para su visualización. En el correspondiente puerto de la Raspberry se conectará la cámara cuya instalación es inmediata. Además, la señal PWM se obtendrá de uno de los pines GPIO por lo que se debe colocar un cable desde este lugar hacia la PCB para su regulación, esta conexión

por comodidad se llevará a cabo en un entrenador con un cabezal adaptado a los pines de la Raspberry.

Por otro lado, la PCB debe alimentarse teniendo en cuenta las especificaciones de las hojas de características proporcionadas por los fabricantes, tanto de los LEDs como del regulador de intensidad. De acuerdo a ellas cada led consume un poco menos de 2V (1,8V) a intensidad nominal de 1A. Para llevar a cabo la regulación se conecta la PCB a una fuente de 12V y 1A de continua, los 6 voltios restantes que no caen en los LEDs se consumirán en un pequeño diodo Zener a modo de pequeño disipador. La elección de este valor de 12V no es un valor escogido al azar, sino que atiende a dos especificaciones. La primera es que el regulador AL8807EV2 debe trabajar por encima de los 7V y por otro es que 12V y 1A es un valor estándar del cual es fácil encontrar fuentes de intensidad en el mercado, es por esto que los valores quedan definidos de esta forma y será necesaria la disipación de la potencia sobrante en un diodo Zener.

A modo de resumen se va a recapitular (Ilustración 15) de qué consta nuestro hardware completo de visión. Por un lado, la Raspberry alimentada a la red eléctrica, con la cámara PI NoIR conectada en su correspondiente puerto y por otro la PCB alimentada a 12V y 1A. Ambas están unidas por un cable en el cual viaja la señal PWM que determina la intensidad que circulara por los diodos LEDs (Ilustración 14).

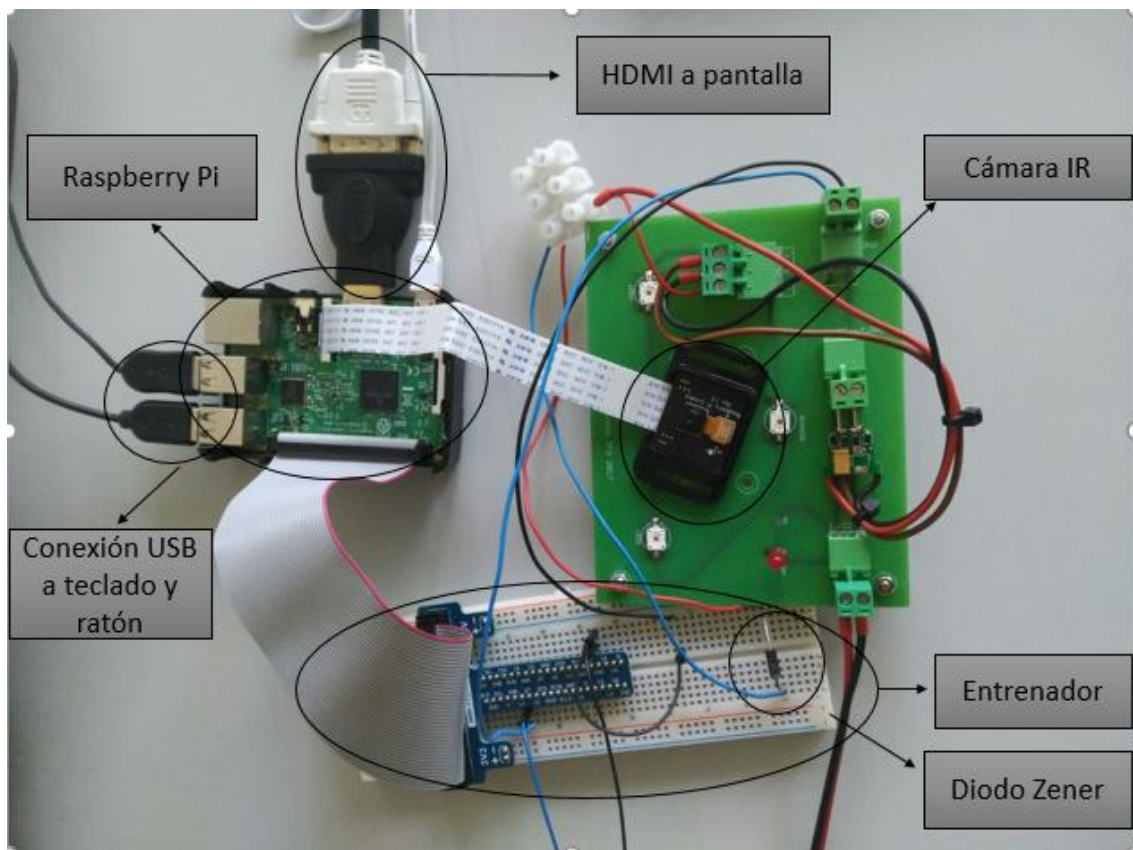


Ilustración 14. Conjunto Hardware final desarrollando. Los principales componentes son las Raspberry, la cámara, el circuito LED y el conexionado entre ellos.

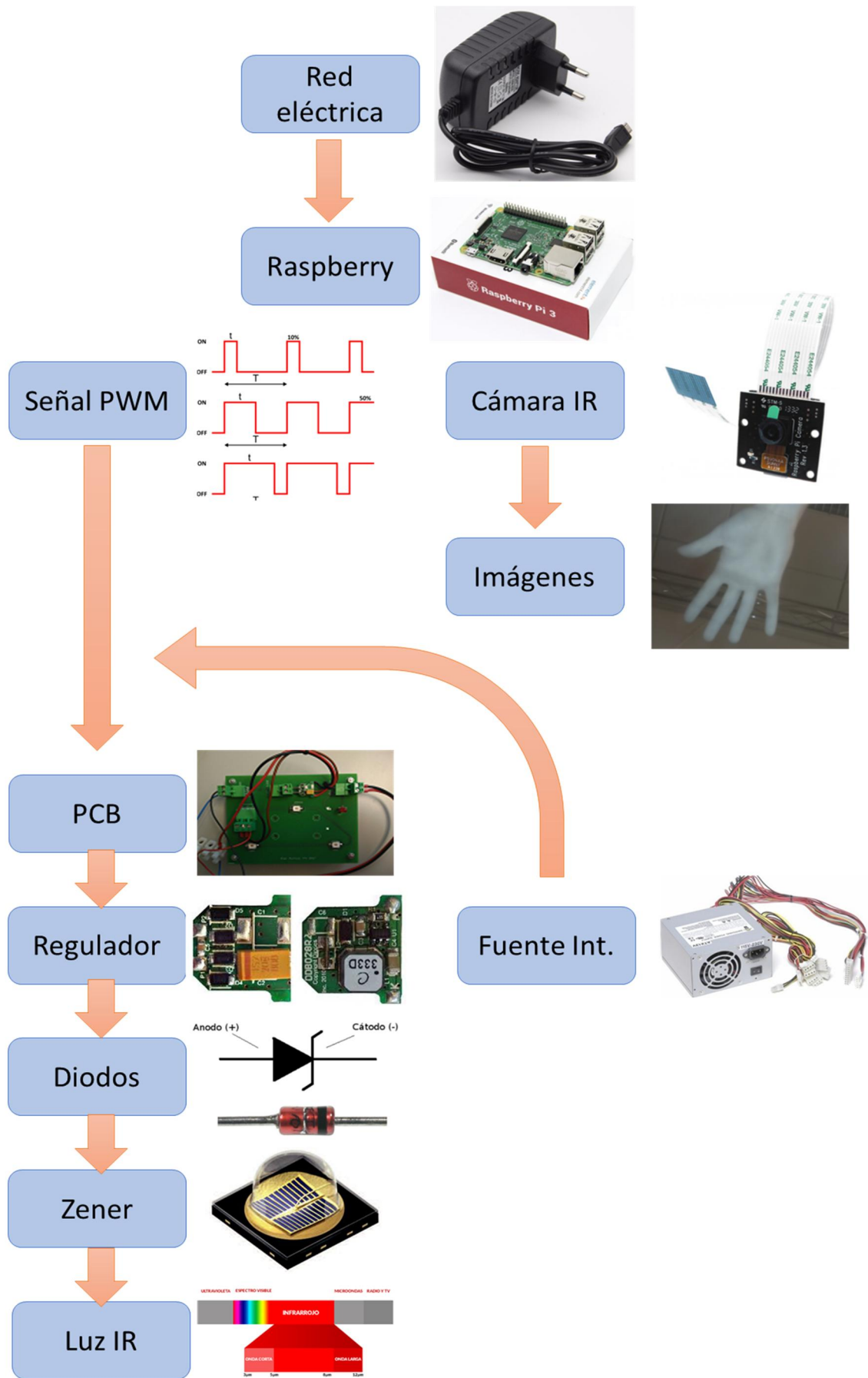


Ilustración 15. Esquema de conexionado de los diferentes elementos que componen el sistema de visión.

4. Detección de gestos

Una vez determinado y desarrollado el hardware, en este capítulo se presentan los pasos a seguir para llevar a cabo la toma de imágenes en tiempo real y el análisis de éstas.

4.1 Imágenes adquiridas por el sensor

Como paso previo a la implementación de algoritmos y secuencias que permitan determinar la ubicación y el contorno de la mano del usuario, es preciso estudiar la tipología de las imágenes que nos vamos a poder encontrar. En la Ilustración 16 se muestran algunos ejemplos de imágenes capturadas por el sensor de visión. Existe una gran casuística en este tipo de imágenes, puede suceder que la mano aparezca de forma nítida, manteniendo sus proporciones y con una iluminación homogénea o, por el contrario, puede suceder que aparezcan deformaciones producidas por la lente del visor o que la iluminación sea irregular, lo que distorsionará la imagen y exigirá a nuestro algoritmo mayor eficacia y fiabilidad.



Ilustración 16. (Arriba izq.) foto de la mano deformada por estar demasiado cerca del objetivo. (Arriba dcha.) la mano mantiene sus proporciones y vemos como la persona de detrás esta menos iluminada y de forma homogénea. (Abajo Izq.) la imagen tiene brillos producidos por los elementos de la habitación. (Abajo Dcha.) imagen con condiciones óptimas.

De las fotos en la Ilustración 16 mostradas tan solo la de abajo a la derecha se adecúa a lo que necesitamos para que la fiabilidad de los resultados sea la deseada. A la vista de las imágenes estudiadas queda demostrada la necesidad de definir un algoritmo de detección de manos que sea lo suficientemente robusto como para permanecer invariante incluso ante situaciones como las anteriormente comentadas. En los siguientes apartados se explica en detalle el método implementado.

4.2 Extracción del contorno

Para el desarrollo del algoritmo de ubicación y extracción del contorno de la mano vamos a apoyarnos en la biblioteca OpenCV. OpenCV es una biblioteca libre de visión por ordenador originalmente desarrollada por Intel que contiene una serie de funciones de digitalización y tratamiento de imágenes que harán el desarrollo del código más sencillo.

Para la utilización de esta biblioteca y muchas otras que están disponibles en los paquetes de instalación del sistema operativo es necesario importarlas antes de trabajar, algunas están incluidas dentro del propio lenguaje de programación y solo es necesario importarlas mediante el código “import” seguido del nombre de la biblioteca, por ejemplo “import cv2”, mientras que para utilizar otras, primero se deberá descargar sus ficheros e instalarlas mediante la orden “sudo apt-get -f install” seguido del nombre del fichero descargado.

Las imágenes adquiridas por el sensor monocromático, debido a que solo vislumbra el rango de luz infrarroja que el filtro deja pasar, tienen el aspecto de imágenes en blanco y negro en las que los objetos más cercanos o en contacto directo con la fuente de luz tienen un color más claro que los objetos situados en segundo plano. Dicho de otra forma, se tratan de imágenes en escala de grises. El valor de esta escala viene definido por la luminosidad que refleja en ellos la luz infrarroja. Se puede establecer entonces una relación entre la intensidad lumínica y la distancia al foco, sin embargo, estos valores de intensidad no solo van a depender de la distancia de cada elemento al sensor sino también de la posición de los mismos.

Como paso previo para la implementación del algoritmo de extracción del contorno se utiliza la función `cv2.threshold` [11]. Si el valor del píxel es mayor que un valor umbral, se le asigna una constante correspondiente a un color (puede ser blanco), de lo contrario se le asigna otra (puede ser negro). El primer argumento de la función es la imagen fuente, que debería ser una imagen en escala de grises y el segundo es el valor umbral que se utiliza para clasificar los valores de los píxeles. Tras la utilización de esta función conseguimos una imagen binaria, con solo dos colores, blanco y negro. El valor del umbral que determina que píxeles serán blancos o negros puede elegirse manualmente, sin embargo, resulta más eficaz dejar que sea el propio algoritmo el que elija este umbral mediante “`cv2.THRESH_OTSU`”. Véase el resultado de utilizar este filtro en la Ilustración 17.

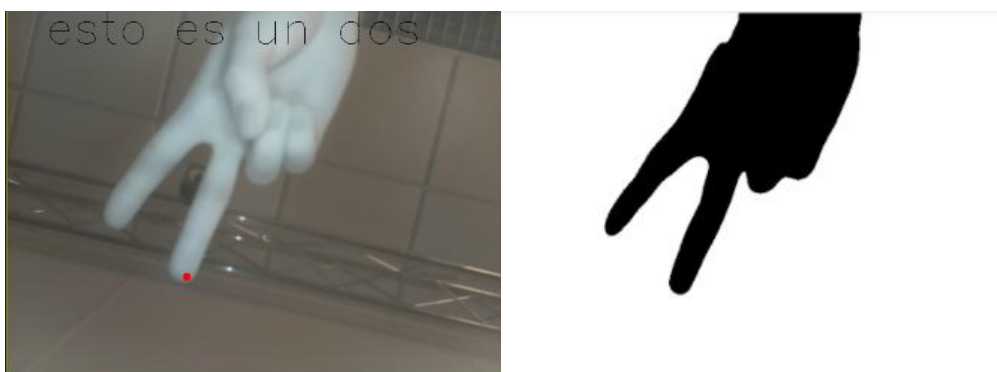


Ilustración 17. Resultado de utilizar la función `cv2.threshold` sobre una imagen en escala de grises.

La segunda función que resulta de gran utilidad es `cv2.findcontours` [11] a la cual debe darse como entrada una imagen binaria, esta función detecta todos aquellos puntos exteriores con propiedades similares y dibuja una línea exterior que los une, de tal forma que el resultado es un contorno de puntos con el mismo valor en la escala de grises. De esta forma la función extrae el contorno de la imagen en blanco y negro que se define como entrada en la función. Para entender este paso de forma sencilla véase Ilustración 18.



Ilustración 18. Ejemplo de imagen antes y tras la utilización de la función `cv2.findcontours`. Se puede ver el resultado de detección de los contornos en la imagen de la derecha.

Por último, cabe destacar una función que nos permitirá con posterioridad obtener el número de dedos que está detectando el sensor de visión en la mano. Ésta es `cv2.convexhull` [11], una función que permite detectar zonas convexas en un conjunto de puntos 2D. Tiene como entrada el conjunto de puntos que forman el contorno de la imagen y a partir de ella podemos obtener el número de “convexity defects” que en nuestro caso corresponderán con el hueco entre dedos. Realizando una discriminación de estos defectos bajo un ángulo máximo podemos obtener el número de dedos que hay en la imagen. Primero se debe definir un polígono con los puntos exteriores de la mano, o polígono de ajuste (Ilustración 19 en rojo), una vez está definido el polígono esta función compara el contorno de la mano con el polígono, detecta aquellos lugares en los que existe una zona convexa y los identifica con un punto en la zona más profunda de la convexidad.



Ilustración 19. Ejemplo que muestra el contorno convexo generado y un defecto de convexidad que permite detectar el espacio entre dedos.

Estas son algunas de las funciones más importantes del código desarrollado, existen otras de menor relevancia que complementan el proceso. En el Anexo 2 se adjunta la parte del código utilizado en la implementación del análisis estructural y descriptor de forma de la mano.

4.3 Detección de manos

Tras el estudio del software existente en el mercado y las distintas formas de extracción de gestos de la mano, se ha determinado que las órdenes básicas para el control de electrodomésticos (concretamente placas vitrocerámicas) son el conteo del número de dedos y la posición de la mano. Debido a que nuestro sistema de visión solo crea mapas e imágenes 2D estamos lejos de realizar programas tan creativos como los que se implementan en el sensor Leap Motion, capaz de detectar la distancia de los objetos y realizar programas en los que el usuario puede jugar con la profundidad.

El conjunto de gestos que puedan realizarse con la mano y que no sean mostrar un número de dedos levantados no serán recogidos como órdenes en nuestro algoritmo y no se detectarán como una acción determinada. El algoritmo detectará los números del 0 al 5 y para aquellos gestos que se consideren más cómodos y sencillos de realizar para el usuario serán escogidos como las órdenes de nuestra placa vitrocerámica.

El algoritmo de detección puede trabajar tanto con imágenes infrarrojas como con imágenes a color tomadas sin colocar el filtro IR, imágenes en color como las que tomaría cualquier cámara. La gran diferencia entre ambas opciones es que si las imágenes son a color pueden existir elementos de la habitación o situados a una profundidad mayor en la imagen (lámparas del techo, elementos decorativos, el propio cuerpo...) que sean detectados como formas lo suficientemente nítidas para que el sensor de visión realice su trabajo, y esto conducirá a resultados erróneos y desdibujados respecto al número de dedos que se muestran en ese momento. Si la imagen está iluminada y filtrada con IR hemos comprobado que los elementos de más profundidad están mucho menos iluminados que aquellos que se encuentran en primer

plano, pasando por alto todos los elementos superfluos y centrando el algoritmo en la forma de la mano.

Un ejemplo de esto se puede observar en la Ilustración 20, en la que un elemento metálico colocado en el techo de la habitación donde se realiza la prueba, crea destellos de luz contaminando la imagen. Aunque en este caso el algoritmo consigue detectar el correcto número de dedos, puede producirse que la perturbación en la imagen binarizada sea suficiente como para inducir a error al programa.

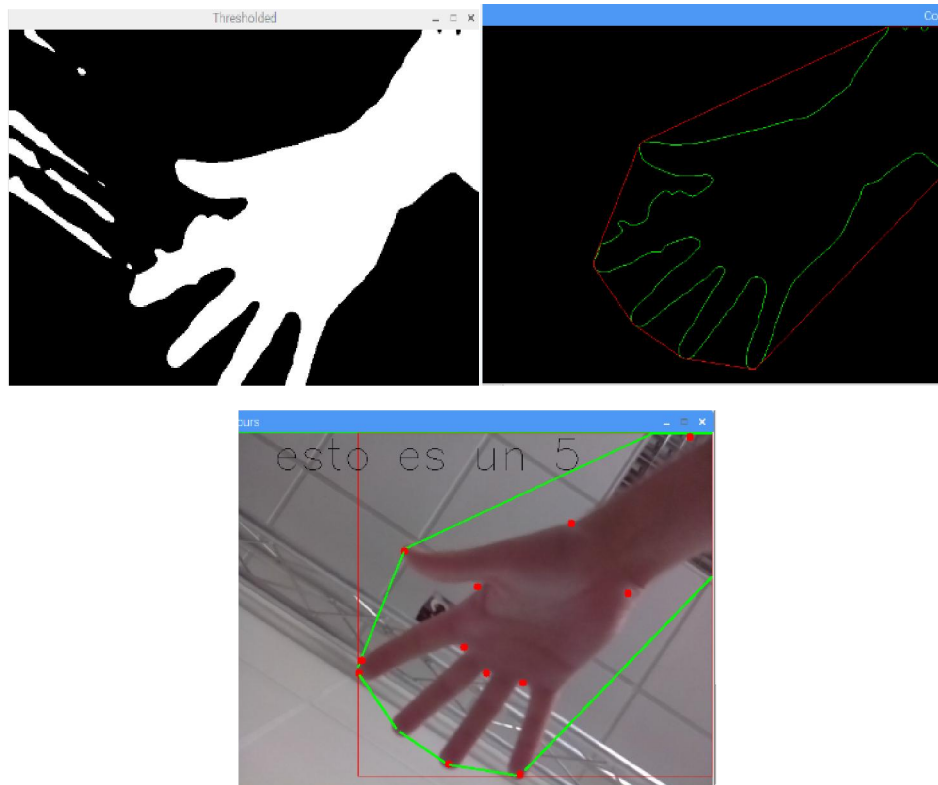


Ilustración 20. Contaminación de la imagen del visor por elementos de la habitación durante la realización de pruebas sin la conexión de la luz infrarroja. (Arriba Izq.) imagen binarizada. (Arriba dcha.) extracción del contorno y el polígono de referencia. (Abajo) contorno detectado.

Un punto a destacar acerca del algoritmo es que a partir de la imagen tomada por el visor el primer paso del código es la conversión a una imagen binaria (blanco y negro), para obtener la imagen umbralizada. El resto de funciones que se encargan de extraer el contorno, y obtener el número de elementos cóncavos para obtener el número de dedos trabajan a partir de la imagen binaria y no sobre la original. Debido a esto, si en la imagen binaria la mano está perfectamente delimitada en blanco, el algoritmo trabajará de forma correcta, pero si en esta se aprecian elementos exteriores pueden falsear la salida del algoritmo.

La frecuencia de muestreo de la Raspberry pi es lo suficientemente alta para poder realizar fotos y videos en tiempo real de forma fluida (alrededor de 1,2GHz de procesador y 30 fps en la cámara), sin que se noten tirones en el movimiento de la mano. Éste es un aspecto importante a la hora de detectar la posición de la mano sin que el

video esté continuamente dando saltos debidos a un bajo número de imágenes por segundo.

Uno de los objetivos que se derivan de la detección de la posición de la mano en el video en tiempo real, es la de poder controlar el puntero del microcontrolador colocando una referencia en algún punto de la mano, para conseguir que la posición de esta referencia en la pantalla video coincida con la del ratón en la pantalla del monitor. En otras palabras, que el movimiento de la mano mueva el ratón en la misma dirección, acoplándose ambas de forma que puedas utilizar la mano como ratón.

Para realizar lo descrito en el párrafo anterior es fundamental una biblioteca de nombre Pygame y de uso gratuito. Pygame [12] es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos (a nivel básico) en dos dimensiones de una manera sencilla. También puede utilizarse para crear otros programas multimedia o interfaces gráficas de usuario. Dentro de esta biblioteca se encuentran una serie de funciones las cuales permiten el control y localización del ratón. A partir de ellas, puedes obtener la posición del ratón en todo momento, simular cualquier tipo de clic, y posicionarlo en las coordenadas de la pantalla deseadas (Ilustración 21).

pygame.mouse

Pygame módulo para trabajar con el ratón

<u>Pygame.mouse.get_pressed</u>	- Obtener el estado de los botones del ratón
<u>Pygame.mouse.get_pos</u>	- Obtener la posición del cursor del ratón
<u>Pygame.mouse.get_rel</u>	- Obtener la cantidad de movimiento del ratón
<u>Pygame.mouse.set_pos</u>	- Establecer la posición del cursor del ratón
<u>Pygame.mouse.set_visible</u>	- Ocultar o mostrar el cursor del ratón
<u>Pygame.mouse.get_focused</u>	- Compruebe si la pantalla está recibiendo la entrada del ratón
<u>Pygame.mouse.set_cursor</u>	- Establecer la imagen para el cursor del ratón del sistema
<u>Pygame.mouse.get_cursor</u>	- Obtener la imagen para el cursor del ratón del sistema

Ilustración 21. Módulo Pygame mouse con las funciones que permiten el control del ratón desde el terminal.

5. Resultados experimentales

En este capítulo se describen los resultados obtenidos para los posibles gestos reproducibles con la mano y se analizan los posibles errores o situaciones que se pueden dar en la utilización del software, comentando áreas de mejora y técnicas para la correcta utilización del dispositivo. En el Anexo 3 se pueden ver fotografías del resto de pruebas realizadas para diversos gestos, con la finalidad de ver cómo funciona el algoritmo ante la gran casuística que puede darse. Los ensayos experimentales han sido realizados en el departamento de Informática del edificio Ada Byron, Véase Ilustración 22.

5.1 Detección de manos en tiempo real

Como puede verse en la Ilustración 23 el algoritmo responde de forma correcta a las variaciones en la forma. Una vez conectada la luz infrarroja, se aprecia que en el funcionamiento del algoritmo es mucho más eficiente. Con una imagen en escala de grises trabaja de forma más precisa que con una imagen en color. Esto se debe a que el filtro “threshold” aplicado a una imagen en escala de grises solo debe comparar el valor de cada pixel con el valor umbral de la función. Si la imagen es en color el paso previo es pasar la imagen a escala de grises y ya a partir de ahí binarizarla.

En la Ilustración 23 se puede apreciar el resultado de aplicación del programa para los gestos “número 1” y “número 2”, el programa puede configurarse para mostrar las imágenes antes y después de todas las funciones y pasos intermedios que se dan hasta el reconocimiento gestual. De las 3 imágenes que se muestran por gesto, la primera (situada en la parte superior a la izquierda), muestra la captura de la cámara sin ningún filtro en la que se ha destacado el contorno exterior y los puntos utilizados en la construcción del polígono de ajuste. A su derecha se ha extraído el contorno (en verde) y el polígono (en rojo) para poder visualizarlos de manera más clara. Por último en la parte inferior se muestra la imagen binarizada, en la que una buena lectura del contorno es fundamental para asegurar el correcto funcionamiento del programa.

En el gesto número 2 se pueden apreciar 2 puntos que no forman parte de la construcción del polígono y además están colocados en las zonas cóncavas entre los dedos. Estos puntos son los que el algoritmo cuantifica para determinar el número de dedos que se están detectando. Pueden darse dos situaciones que el punto sea descartado por no cumplir con los requerimientos (Anexo 2) o que éste se considere como un punto representativo, de forma que si existen n puntos representativos el número de dedos que se están mostrando a cámara es de $n+1$.

Para entender esta idea de forma más clara imagínesse que se colocan los dedos formando un 4 en el objetivo de la cámara, los puntos representativos que coinciden con los huecos de unión de los dedos a la mano son $n=3$, por tanto $n+1=4$ que coincide con el valor correcto que se detecta.

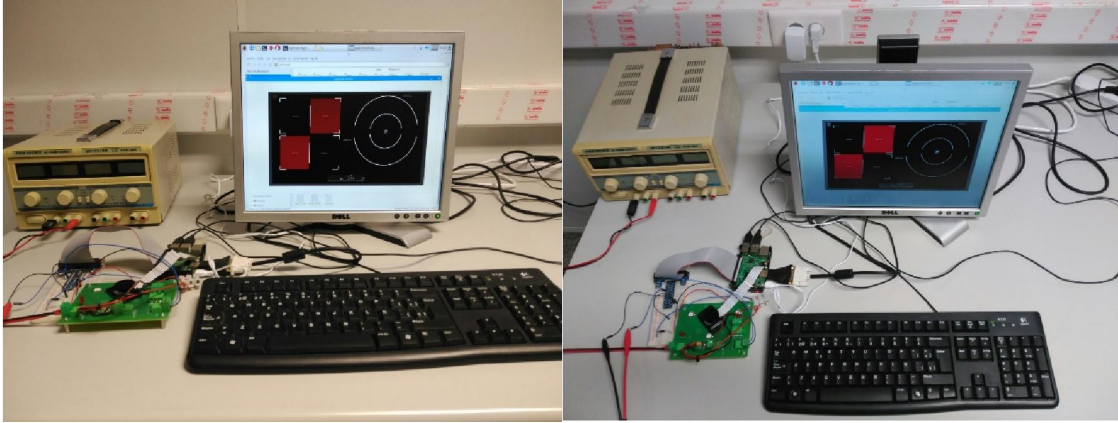


Ilustración 22. Puesto de trabajo departamental donde se lleva a cabo el proyecto.

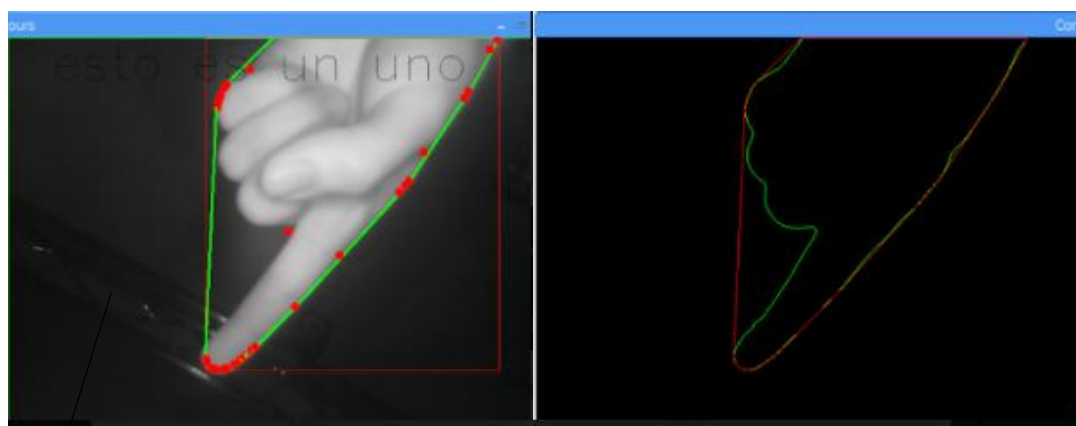


Imagen sin filtros. En verde el polígono y en rojo los puntos de interés.

En verde el contorno y en rojo de nuevo el polígono.

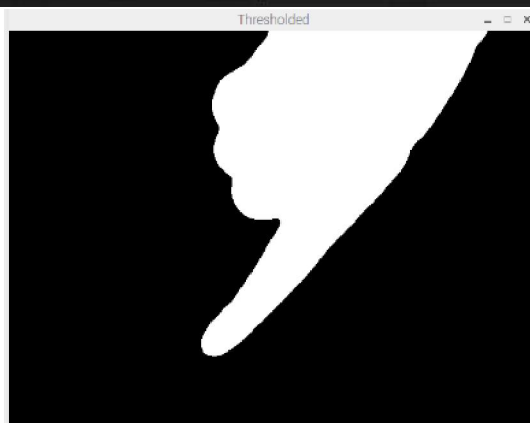
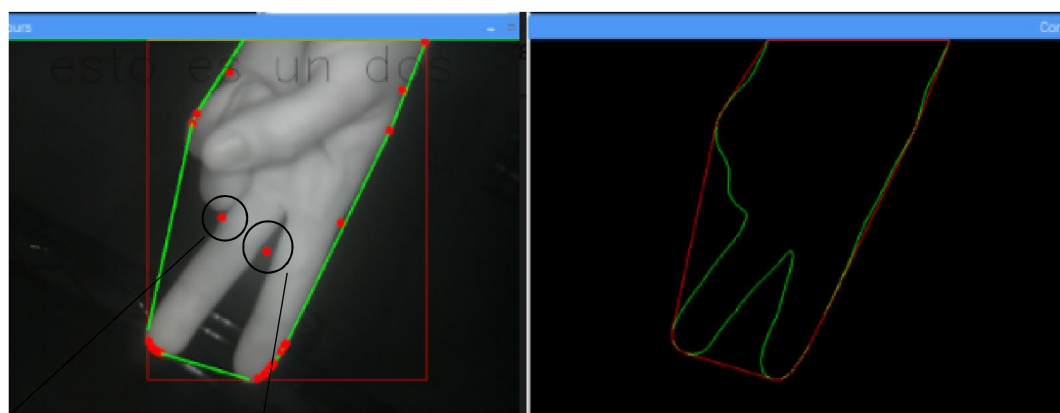


Imagen binarizada.



Punto no representativo.

Punto representativo.



Ilustración 23. Resultados tras la conexión de la luz infrarroja para los gestos 1 y 2, con resultados satisfactorios en el reconocimiento del contorno y la detección del número de dedos.

5.2 Análisis y discusión

Aunque como hemos podido ver el objetivo del proyecto se cumple satisfactoriamente, existen una gran cantidad de puntos en los que se puede profundizar con la finalidad de mejorar algunos aspectos o añadirle nuevas prestaciones.

Se vislumbra que se han quedado sin explorar un gran número de alternativas hardware y software. A nivel de hardware, las posibilidades eran muy grandes y fue necesario centrar el objetivo en conseguir una fuente de iluminación con una potencia determinada, la cual se ha podido comprobar que era suficiente. A nivel de precisión, la funcionalidad del sistema desarrollado no destaca errores relevantes, se trata de un sistema de iluminación con una implementación de prototipo funcional, sin prestar atención a componente estético, y cuyas dimensiones podrían compactarse de forma importante, pero los resultados son satisfactorios.

Una de las posibles áreas de mejora hardware es la implementación de un sensor de detección de luminosidad, el cual regule la señal PWM que alimenta con una determinada intensidad los LEDs. En nuestro sistema esta señal se elige de forma manual, con el sensor se podría realizar un sencillo programa que disminuyese la potencia de los LEDs si existe mucha luz infrarroja o aumentarla en caso de detectarse que es insuficiente.

Otra área de mejora es el número y disposición de los diodos sobre la PCB (Ilustración 24), en ángulo de iluminación de los LEDs es más grande que el ángulo que captura el objetivo de la cámara, es por esto que no se puede determinar qué porcentaje de la luz infrarroja emitida se está visionando en la imagen y cual se está perdiendo en ángulos en los que el objetivo del sensor de visión no es capaz de alcanzar. Por ello un estudio de distintas configuraciones de los diodos sobre la placa para observar la homogeneidad de la iluminación sería de gran relevancia. Con los resultados de dicho estudio se puede determinar el número y disposición óptima que hacen nuestra fuente IR más eficiente.

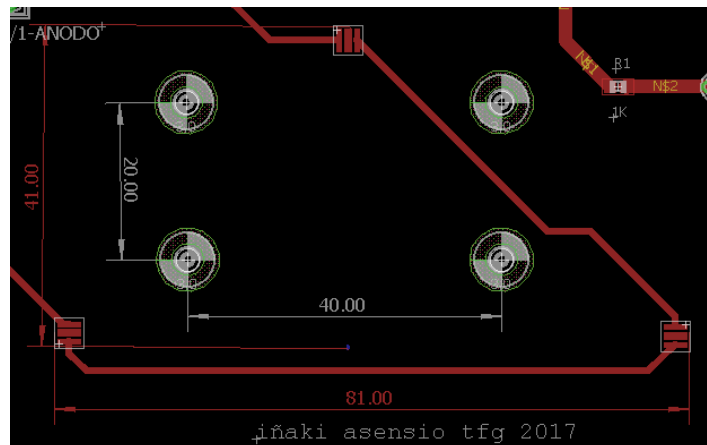


Ilustración 24. Configuración en triángulo utilizada en la PCB (en rojo la posición de los tres LEDs).

A nivel software sí que se han detectado componentes con margen de mejora, que podrían dar lugar a complicaciones en el control del electrodoméstico. El aspecto más significativo es la posible falta de robustez en el conteo del número de dedos. Son varias las causas que pueden llevar al algoritmo a confundir y extraer mal el contorno de la mano, el principal es la detección de objetos o formas externas que pueden introducirse en el campo de visión de la cámara. Los elementos que están alejados de la fuente de IR (es decir, del sensor de visión) tienen una menor influencia que aquellos que se encuentran en posiciones cercanas, desde una pulsera que cuelga de la muñeca más de lo debido, una segunda mano que sin querer sale en la imagen, o incluso un elemento decorativo del techo o la habitación. Si detecta que este elemento refleja iluminación infrarroja suficiente será tratado por el algoritmo como una “forma a la que analizar sus apéndices y contarlos”. En caso de aparecer tanto la forma de la mano como el de esa entidad no relevante en el mismo fotograma, el algoritmo procesará solo el contorno más grande (que generalmente es la mano).

Las posibles mejoras del algoritmo pasan por la utilización de otras librerías y la optimización/depurado del programa. OpenCV es una biblioteca con una gran versatilidad y la comunidad de programadores dedicados al tratamiento de imágenes la ha convertido en una herramienta con gran potencial, no obstante, la limitación de tiempo y el uso de software de uso libre y gratuito han llevado a la creación de un algoritmo eficaz, pero con errores en situaciones poco comunes, en las que no se profundiza, como son las que se exponen en el párrafo anterior.

Se plantea como posible mejora, el estudio de las diferentes bibliotecas, programas de tratamiento de imágenes y el estudio de los diferentes lenguajes de programación con la finalidad de encontrar alternativas computacionales más eficientes y que tengan una mejor respuesta en tiempo real al reconocimiento de imágenes. Las compatibilidades entre lenguajes y el uso de determinados paquetes de funciones es complejo, y el usuario se debe asegurar que se trata de software libre antes de descargarlo, de lo contrario cuando la librería esta descargada pueden aparecer alertas que indican que el programa descargado no es reconocido por el compilador.

5.3 Comparación con dispositivo comercial Leap Motion

En este capítulo se va a tratar de establecer los parecidos y principales diferencias de nuestro sensor de visión desarrollado, con los desarrollados por empresas de renombre mundial, cuyo nivel económico y tecnológico las sitúan en la cima del tratamiento computacional de imágenes.

Una vez más no queda más remedio que prestar atención al reconocido Leap Motion [7] que desarrolla la empresa americana Leap Motion INC. El controlador Leap Motion es un pequeño dispositivo periférico USB que está diseñado para ser colocado en un escritorio físico, mirando hacia arriba. Utilizando dos cámaras IR monocromáticas y tres LEDs infrarrojos, el dispositivo observa un área aproximadamente semiesférica, a una distancia de aproximadamente 1 metro. Los LEDs generan luz infrarroja sin patrón y las cámaras generan casi 200 fotogramas por segundo de datos reflejados. Esto se envía a través de un cable USB al ordenador central, donde es analizado por el software Leap Motion usando "matemáticas complejas" de una manera que no ha sido revelada por la compañía, de alguna manera sintetiza datos de posición 3D por comparación con los fotogramas 2D generados por las dos cámaras.

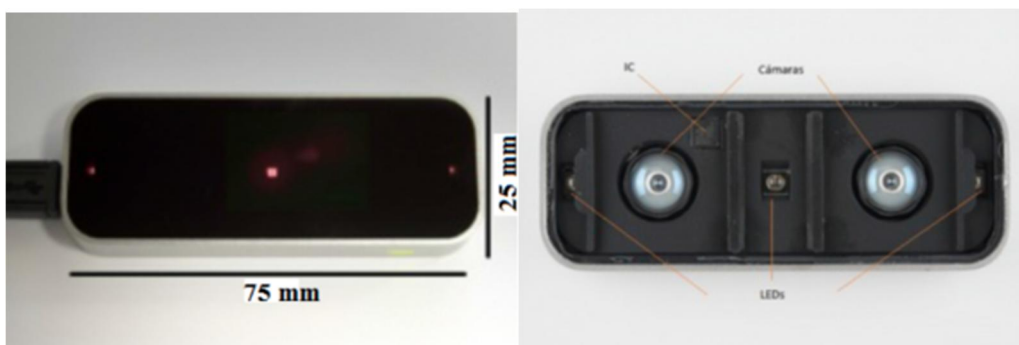


Ilustración 25. (Izquierda) Leap plenamente funcional donde se distinguen los 3 diodos IR. (Derecha) Leap con la carcasa superior retirada en él se pueden ver los componentes internos.

Este dispositivo tiene unas dimensiones muy reducidas en comparación con otros interfaces gestuales que hay actualmente en el mercado, tan solo mide 75 mm de largo, 25 mm de ancho y 11 mm de alto. En comparación con el nuestro el cual tiene bien diferenciados la parte de los diodos y la cámara conectada al microprocesador, la Leap es un diseño más compacto y robusto, en la que los componentes no se encuentran expuestos a golpes, líquidos, polvo...

Al comienzo del proyecto se estudió la configuración y componentes de este dispositivo, con la intención de reproducirlo de la forma más fiel posible, se buscó información acerca del modelo y potencia de los diodos, ángulo de visión de las cámaras, etc. Existe una gran información acerca de la electrónica interna que contiene, sin embargo, los

modelos de los diodos es algo muy protegido, se puede ver que son de ojo de buey, pero el modelo no ha sido identificado.

El dispositivo Leap Motion tiene el mismo número de diodos que el nuestro, aunque la distribución es diferente, en la Ilustración 25 puede apreciarse como el led central parece lucir más que los laterales, por lo que puede deducirse que tiene mayor potencia. La superficie superior hace las veces de filtro, este material funciona de manera similar al filtro IR que se coloca en el objetivo de la cámara de nuestro sensor.

Existe una diferencia sustancial entre ambos dispositivos, el que tenga dos cámaras no es algo casual, la disposición de éstas permite tomar dos fotos simultáneas que mediante algoritmos matemáticos permiten comparar ambos fotogramas para triangular la distancia de los objetos a las cámaras monocromáticas. Con ello la Leap puede crear mapas 3D y crear programas de control gestual mucho más complejos.

Para ver si existen grandes diferencias entre las imágenes tomadas por la Leap Motion a una mano y las que se pueden realizar con nuestro hardware véase la Ilustración 26, la iluminación de la Leap es ligeramente más potente que la del proyecto porque puede apreciarse la gran variación de contraste entre el primer plano y los posteriores. Pero a nivel de calidad, resolución y versatilidad se puede afirmar que estamos en condiciones de igualar las prestaciones de la cámara comercial en el campo estudiado.



Ilustración 26. Imágenes tomadas con Leap Motion (arriba) comparada con imagen del dispositivo propio (abajo).

6. Detección de gestos en cocinas de inducción

Como se pueden observar en las diversas fotografías que se han adjuntado en el proyecto, los objetivos se han alcanzado. A nivel teórico, se tiene un programa que obtiene las coordenadas de la mano grabada por una cámara y con ellas se es capaz de controlar el cursor, entre otras funciones. Además, el algoritmo reconoce el número de dedos que se están mostrando, pudiéndose asociar eventos a cada una de las posiciones.

La aplicación más inmediata es obvia, podemos controlar el microprocesador con la palma de la mano, clic izquierdo, clic derecho, desplazarnos por la pantalla... pero las posibilidades de éste son muy amplias. Como el proyecto se ha enfocado desde el principio en la utilización del programa para control de electrodomésticos, se va a continuar por ese camino, sin cerrar la mente a las innumerables oportunidades que éste ofrece.

Para evaluar nuestro sistema de detección de gestos, se va a desarrollar un programa que simule que el display del microprocesador es una cocina de inducción sobre el cual se podrán realizar acciones como elegir uno de los fuegos, subir la temperatura, bajarla y encender y apagar el sistema. Con esto se conseguirá no solo que el usuario se familiarice con las aplicaciones del programa desarrollado, sino que también se evaluará la aceptación y comodidad de los distintos gestos, para asociarlos a las acciones del simulador.

La empresa de electrodomésticos BSH ha facilitado imágenes de diferentes modelos de sus placas vitrocerámicas de inducción con mayor éxito en el mercado internacional. Serán dos de estas placas las que introduciremos en la demo, para su posterior simulación.

Dentro de las imágenes cedidas por la empresa de electrodomésticos podemos ver que existen diversas configuraciones y formas de fuegos. En España no son tan comunes de encontrar las placas vitrocerámicas con forma cuadrada o rectangular, pero en otros lugares como Alemania estas tienen una gran popularidad (Ilustración 27).

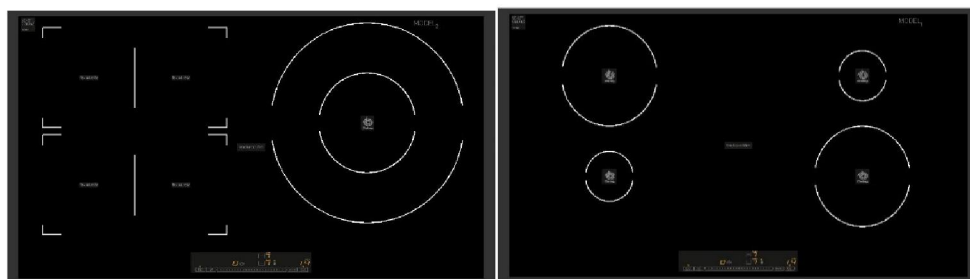


Ilustración 27. Configuración de las placas de inducción elegidas para llevar a cabo el programa de prueba.

Como puede apreciarse en la Ilustración 28 se debe ser capaz de manipular fuegos circulares de distintos tamaños, cuadrados y configuraciones dobles en las que pueden encenderse simultáneamente dos fuegos concéntricos para ajustarse al diámetro de la olla/sartén.

Existen muchas formas de llevar a cabo el “User interface” de la placa, diferentes posibilidades en la elección de los gestos y las formas de selección de cada fuego. Para comprobar cuál se adapta mejor a las necesidades y comodidad de los usuarios se va a llevar a cabo un programa en la que se requieran diferentes instrucciones para el control de cada una de las placas.

Para el desarrollo del programa se ha utilizado el mismo lenguaje de programación, con la intención de mantener una coherencia entre el algoritmo de detección y el programa de simulación. El módulo Pygame el cual permite realizar el control de los eventos del ratón nos va a permitir también la realización de la demo de prueba, el módulo incluye una serie de paquetes con funciones destinadas a la creación de videojuegos a nivel amateur con animaciones muy simples. Haciendo uso de estas funciones podemos hacer sencillas operaciones como crear círculos rojos para simular que un fuego está encendido en lugares determinados y colocar a su lado un indicador numérico que indique la potencia.

A priori se puede ver que la forma más sencilla de seleccionar un fuego es colocar la posición de la mano sobre él o sobre una representación de él, y una vez está situada encima se seleccionará con la detección de un determinado número de dedos, el cual esté asociado a la orden clic (por ejemplo, el número de dedos 1, puesto que lo habitual es “navegar” con la mano totalmente extendida o formando un 2 con los dedos índice y corazón).

Una vez el fuego que se desea manipular está seleccionado, lo que sigue es poder subir el nivel de potencia, para ello, de forma predeterminada el ciclo de la placa está configurado como ciclo ascendente (como no hay ninguno encendido, lo normal es subir la intensidad) esto significa que con un sencillo clic sobre el fuego seleccionado pasará de estar apagado a tener un nivel 1, así sucesivamente podemos llegar hasta el nivel más alto, el 9. En caso de querer bajarlo debemos cambiar el ciclo a descendente, para ello existirán dos zonas con unas flechas hacia arriba y hacia abajo, son las encargadas de cambiar el ciclo de la placa. Haciendo clic en la flecha que señala hacia abajo ya podremos reducir la intensidad del fuego cuya intensidad se desea reducir.

Como alternativa existe la posibilidad de prescindir de este sistema de ciclos y utilizar otras formas de discernir entre el aumento y la disminución del nivel de un fuego. Por ejemplo, en las placas donde los fogones están destinados para grandes ollas, paelleras... y por tanto tienen mayor superficie se puede hacer una separación ficticia entre la mitad superior e inferior del fuego, para aumentar la intensidad bastará con situarte en la mitad superior y para disminuirlo en la parte inferior de éste. Este sistema es más sencillo a nivel de interacción, pero menos visual por lo que podría resultar

menos obvio y más complicado de utilizar para personas mayores o personas que lo utilizan por primera vez.

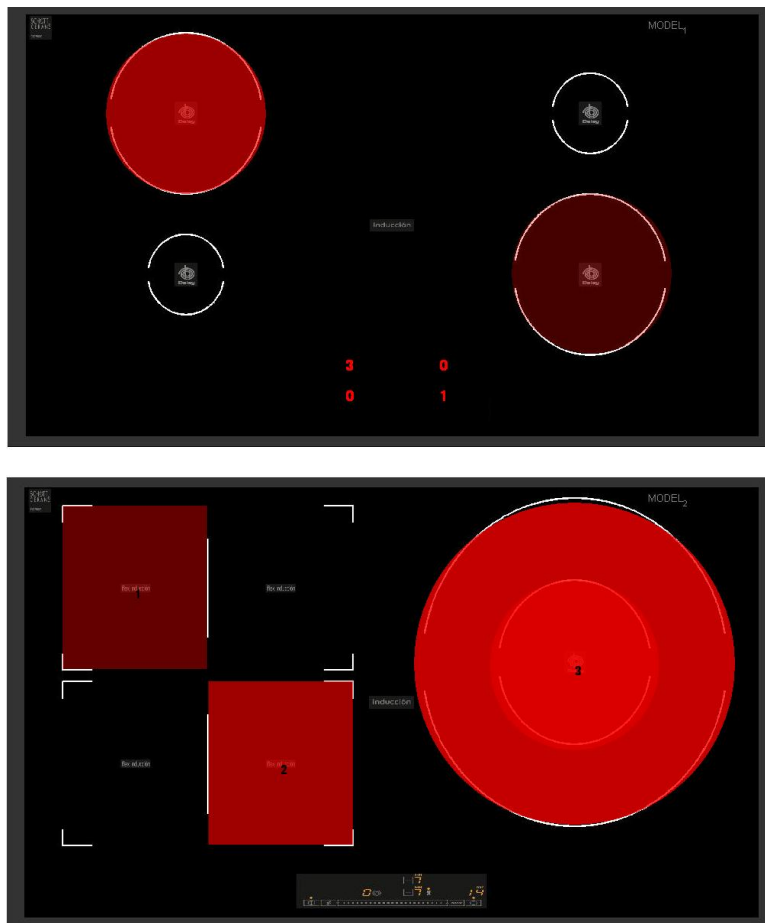


Ilustración 28. Resultado de las pruebas del simulador para las dos placas vitrocerámicas seleccionadas.

Puede verse que en función de la potencia seleccionada el nivel de rojo es más o menos traslucido, como ocurre en las cocinas de verdad. Es una forma didáctica y visual de ver el nivel de cada fuego sin necesidad de tener que mirar el indicador numérico. No obstante, los temas visuales del simulador solo tienen una función estética y didáctica puesto que si el sistema se implementa en una placa de inducción real el color rojo del fuego no es apreciable. Lo realmente importante es la implementación del control gestual en el control de las vitrocerámicas.

Cuando se implemente el dispositivo desarrollado en la cocina de inducción, es el propio cristal vitrocerámico el que actúa como filtro infrarrojo, prescindiendo del filtro apreciable en la Ilustración 12. Por esto se solicita de nuevo a BSH una porción de vitrocerámica con propiedades de filtro infrarrojo para realizar las pertinentes pruebas y comprobar si existen diferencias apreciables entre las imágenes tomadas con el filtro en forma de lente o con la placa vitrocerámica, con un espesor tres veces más grande.

Tras analizar las fotos con ambos filtros infrarrojos, se llega a la conclusión de que no existe diferencia apreciable entre las imágenes tomadas con uno u otro. El espesor no tiene una influencia directa en la apariencia de los fotogramas. Además, el rango de longitud de onda de ambos filtros es similar y cercano a los 850 nm. Se puede apreciar en la Ilustración 29 una misma escena colocando en el objetivo de la cámara los dos tipos de filtro que se tienen, la imagen tomada con el filtro de lente es más uniforme y nítida, se puede apreciar como la luz del fondo no crea tanto brillo que con la porción de vitrocerámica. También se puede ver como influyen los dos tipos de filtro en la nitidez de captura (se ha intentado enfocar correctamente en ambas tomas, la distorsión es causa del filtro), con la lente es más nítida que con la vitrocerámica.

A pesar de estas pequeñas variaciones, la utilización del filtro vitrocerámico no supone un impedimento en la realización del proyecto, y al realizar ensayos prácticos se confirma esta idea.

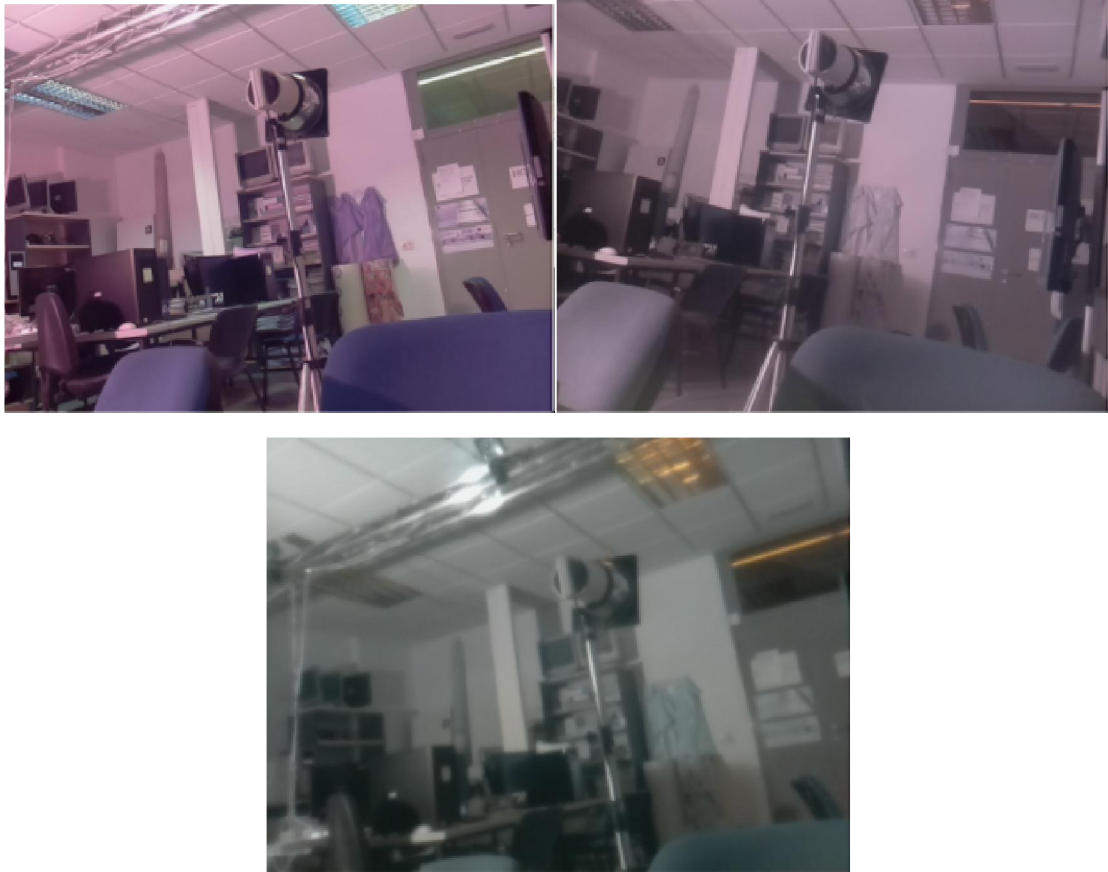


Ilustración 29. (Arriba Dcha.) foto de la habitación sin colocar el filtro IR. (Arriba Izq.) Foto tomada con el filtro de lente colocado. (Abajo) Foto tomada con el fragmento de placa vitrocerámica.

7. Conclusiones y líneas futuras de trabajo

7.1 Conclusiones y valoración del trabajo realizado

En este Capítulo se van a recuperar los objetivos propuestos al inicio del trabajo y a analizar los resultados obtenidos. El objetivo principal que se perseguía mediante la realización del trabajo era el desarrollo de un dispositivo que permitiese detectar la posición de la mano del usuario mediante un sensor de visión. Para ello en primer lugar ha sido preciso realizar una revisión de las distintas tecnologías de visión existentes en la actualidad. Para cada una de ellas se ha estudiado el principio de funcionamiento, de forma que se esté en condiciones de determinar de forma acertada las ventajas y limitaciones que el empleo de cada una de ellas supone para el objetivo concreto de este trabajo, que es el desarrollo de una interfaz gestual para aparatos electrodomésticos.

Por otro lado, se ha llevado a cabo una revisión del estado de la materia en las distintas técnicas de adquisición basada en visión por ordenador y a partir de ella se han elegido una serie de componentes para crear un sistema de visión propio adecuado a las especificaciones del proyecto.

A partir de los conocimientos adquiridos y las posibilidades que ofrecen los algoritmos estudiados se ha desarrollado un código de detección de manos propio. Este algoritmo combina funciones utilizadas en el tratamiento de imágenes de la biblioteca libre OpenCV y el resultado es un algoritmo versátil y con una respuesta muy rápida pero que obviamente podría ser desarrollado y utilizado como base para la implementación de algoritmos más complejos.

Otro de los objetivos planteados al inicio del trabajo era el desarrollo de un simulador que permitiese evaluar distintos gestos a la hora de implementar un sistema de reconocimiento gestual para diferentes aparatos electrodomésticos. En este caso el estudio se ha centrado en dos cocinas de inducción comercializadas por BSH.

Atendiendo a los objetivos planteados al inicio del Trabajo Fin de Grado y evaluando el trabajo realizado se puede concluir que no solo se han cumplido todos los objetivos propuestos, sino que además se ha desarrollado un sistema de detección de manos que supone una contribución tecnológica interesante en el campo del reconocimiento de formas y detección de gestos, con el consiguiente beneficio para BSH en materia de investigación e innovación en sus productos.

7.2 Líneas futuras de trabajo

Aunque se puede concluir que las áreas planteadas en el proyecto han sido abordadas con resultados satisfactorios, las limitaciones de tiempo que se tienen en este tipo de proyectos académicos hacen que una gran cantidad de líneas de trabajo no puedan

plantearse y realizarse. Es por esto que se van a proponer una serie de alternativas que pueden resultar interesantes como líneas futuras de trabajo.

- El algoritmo desarrollado obtiene el contorno del objeto de mayor área de la imagen y obtiene el número de apéndices que este tiene, no obstante, en ningún momento el algoritmo detecta que lo que está viendo por pantalla es una mano. Cualquier objeto con salientes bien diferenciadas podría ser reconocido por el algoritmo, se propone, por tanto, realizar el desarrollo de un código que compare la imagen obtenida por el visor con una biblioteca de imágenes donde existan una gran cantidad de imágenes de las distintas posiciones y formas que puede producir una mano al colocarse delante de una cámara. Con esto se podría conseguir que el algoritmo solo entendiese como órdenes aquellas imágenes las cuales ha contrastado con la biblioteca y se trata de una mano verdadera. En este contexto se puede hacer referencia a una biblioteca de imágenes ya existente y con imágenes almacenadas para esta finalidad el “dataset” de imágenes MPEG-7 [13] podría utilizarse para realizar una comparativa con el descriptor de forma.
- Una alternativa a la opción anterior es la implementación de un sistema que en el momento de la colocación de la placa vitrocerámica realice una serie de imágenes para detectar aquellos elementos de la habitación que van a permanecer inalterados, es decir, que corresponden con el mobiliario de la cocina. De esta forma se pueden identificar objetos fijos que siempre van a salir en el sensor de visión, con la intención de que el algoritmo los ignore y no los tenga en cuenta en el reconocimiento de la mano que va a interactuar con el electrodoméstico.
- Una vez se ha explorado el anterior punto, se puede proceder a desarrollar gestos más complejos que no solo supongan el conteo del número de dedos, realizándose éstos por el mismo procedimiento de comparación con bibliotecas, de esta forma las posibilidades del algoritmo aumentarían exponencialmente.
- Por último, este trabajo se ha centrado en el control gestual a la aplicación de la realización de un “User Interface” para cocinas de inducción, pero el siguiente paso es el desarrollo de técnicas similares para cualquier otro electrodoméstico o aplicación de la domótica, con la función de hacer la vida más cómoda a los usuarios. Dentro de los electrodomésticos que podrían incorporar esta tecnología se encuentran algunos como el horno, el frigorífico, las persianas, la iluminación, el control remoto del televisor, etc.

8. Bibliografía

- [1] *“Sistema de interfaz gestual basado en visión para electrodomésticos”*. Trabajo Fin de Master. Departamento de Informática e Ingeniería de Sistemas. Escuela de Ingeniería y Arquitectura Universidad de Zaragoza, Septiembre de 2015. Autor: Alejandro Inogés Blasco.
- [2] <https://www.nintendo.es/Wii-U/La-consola/Especificaciones> (27 de 09 de 2017).
- [3] <https://www3.oculus.com/en-us/dk2> (27 de 09 de 2017).
- [4] <https://es.wikipedia.org/wiki/RGB/> (27 de 09 de 2017).
- [5] <https://www.digikey.es/> Empresa de componente electrónicos a nivel internacional (27 de 09 de 2017).
- [6] <http://es.farnell.com/> (27 de 09 de 2017).
- [7] <https://www.leapmotion.com/> (27 de 09 de 2017).
- [8] <https://www.raspberrypi.org/> (27 de 09 de 2017).
- [9] <https://www.autodesk.com/products/eagle/overview/> PCB Design & Schematic Software (EAGLE) Autodesk (27 de 09 de 2017).
- [10] <https://www.raspbian.org/> (27 de 09 de 2017).
- [11] *“OpenCV Computer Vision Projects With Python”*. Autor: Joseph Howse, Prateek Joshi, Michael Beyeler. Editorial: Packt. Octubre de 2016.
- [12] <https://www.pygame.org/docs/> (27 de 09 de 2017).
- [13] F. Mokhtarian and M. Bober, *“Curvature Scale Space Representation: Theory, Applications & MPEG-7 Standardization”*. Dordrecht: Kluwer Academic Publishers, 2003.

Anexo 1: Características de los leds infrarrojos y el dispositivo regulador

High Power Infrared Emitter (850 nm) Version 1.4

SFH 4232



Features:

- IR lightsource with high efficiency
- Low thermal resistance (Max. 9 K/W)
- Centroid wavelength 850 nm
- Superior Corrosion Robustness (see chapter package outlines)
- The product qualification test plan is based on the guidelines of AEC-Q101-REV-C, Stress Test Qualification for Automotive Grade Discrete Semiconductors.
- **Package:** SMT package

Applications

- Infrared Illumination for cameras
- Surveillance systems
- Machine vision systems

Notes

Depending on the mode of operation, these devices emit highly concentrated non visible infrared light which can be hazardous to the human eye. Products which incorporate these devices have to follow the safety precautions given in IEC 60825-1 and IEC 62471.

Ordering Information

Type:	Total Radiant Flux Φ_e [mW] $I_F = 1A, t_p = 10ms$	Ordering Code
SFH 4232	530 (≥ 320)	Q65110A8754

Note: Measured with integrating sphere.

Maximum Ratings ($T_A = 25\text{ °C}$)

Parameter	Symbol	Values	Unit
Operation and storage temperature range	$T_{op}; T_{stg}$	-40 ... 125	°C
Junction temperature	T_j	145	°C
Reverse voltage	V_R	1	V
Forward current	I_F	1000	mA
Surge current ($t_p \leq 200\ \mu\text{s}$, $D = 0$)	I_{FSM}	5	A
Power consumption	P_{tot}	1800	mW
ESD withstand voltage (acc. to ANSI/ ESDA/ JEDEC JS-001 - HBM)	V_{ESD}	2	kV
Thermal resistance junction - soldering point	R_{thJS}	9	K / W

Characteristics ($T_A = 25\text{ °C}$)

Parameter	Symbol	Values	Unit
Peak wavelength ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	λ_{peak}	860	nm
Centroid wavelength ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	$\lambda_{centroid}$	850	nm
Spectral bandwidth at 50% of I_{max} ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	$\Delta\lambda$	30	nm
Half angle	ϕ	± 60	°
Dimensions of active chip area	L x W	1 x 1	mm x mm
Rise and fall times of I_e (10% and 90% of $I_{e,max}$) ($I_F = 5\text{ A}$, $R_L = 50\ \Omega$)	t_r / t_f	7 / 14	ns
Forward voltage ($I_F = 1\text{ A}$, $t_p = 100\ \mu\text{s}$)	V_F	1.5 (≤ 1.8)	V
Forward voltage ($I_F = 5\text{ A}$, $t_p = 100\ \mu\text{s}$)	V_F	2 (≤ 2.9)	V
Radiant intensity ($I_F = 1\text{ A}$, $t_p = 100\ \mu\text{s}$)	$I_{e, typ}$	180	mW/sr
Temperature coefficient of I_e or Φ_e ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	TC_I	-0.3	% / K
Temperature coefficient of V_F ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	TC_V	-1	mV / K
Temperature coefficient of wavelength ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	$TC_{\lambda_{centroid}}$	0.3	nm / K

Grouping ($T_A = 25\text{ °C}$)

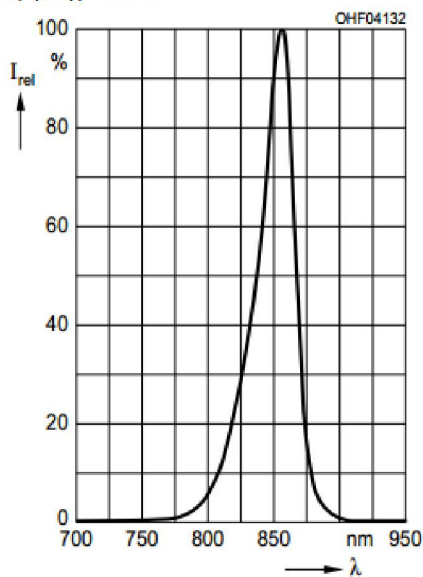
Group	Min Total Radiant Flux	Max Total Radiant Flux
	$I_F = 1\text{ A}, t_p = 10\text{ ms}$ $\Phi_{e\text{ min}}$ [mW]	$I_F = 1\text{ A}, t_p = 10\text{ ms}$ $\Phi_{e\text{ max}}$ [mW]
SFH4232 - CB	320	500
SFH4232 - DA	400	630
SFH4232 - DB	500	800

Note: Measured with integrating sphere.

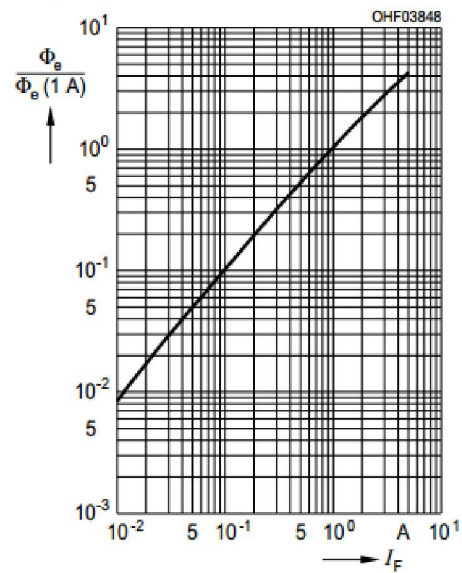
Only one group in one package unit (variation lower 1.6:1)

Relative Spectral Emission ^{1) page 12}

$$I_{\text{rel}} = f(\lambda), T_A = 25\text{ °C}$$

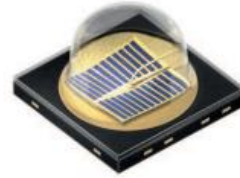
**Relative Total Radiant Flux** ^{1) page 12}

$$\Phi_e / \Phi_e(1\text{ A}) = f(I_F), T_A = 25\text{ °C}, \text{ Single pulse, } t_p = 100\text{ }\mu\text{s}$$



OSLON Black Series (850 nm) - 90° Version 1.5

SFH 4715



Features:

- IR lightsource with high efficiency
- Low thermal resistance (Max. 11 K/W)
- Centroid wavelength 850 nm
- Superior Corrosion Robustness (see chapter package outlines)
- The product qualification test plan is based on the guidelines of AEC-Q101-REV-C, Stress Test Qualification for Automotive Grade Discrete Semiconductors.

Applications

- Infrared Illumination for cameras
- Surveillance systems
- Machine vision systems
- Eye tracking systems

Notes

Depending on the mode of operation, these devices emit highly concentrated non visible infrared light which can be hazardous to the human eye. Products which incorporate these devices have to follow the safety precautions given in IEC 60825-1 and IEC 62471.

Ordering Information

Type:	Radiant Intensity I_e [mW/sr] $I_F=1\text{ A}, t_p=10\text{ ms}$	Ordering Code
SFH 4715	270 (≥ 160)	Q65111A2364

Note: measured at a solid angle of $\Omega = 0.01\text{ sr}$

Maximum Ratings ($T_A = 25\text{ °C}$)

Parameter	Symbol	Values	Unit
Operation and storage temperature range	$T_{op}; T_{stg}$	-40 ... 125	°C
Junction temperature	T_j	145	°C
Reverse voltage	V_R	1	V
Forward current	I_F	1000	mA
Surge current ($t_p = 500\text{ }\mu\text{s}$, $D = 0$)	I_{FSM}	5	A
Power consumption	P_{tot}	1.8	W
ESD withstand voltage (acc. to ANSI/ ESDA/ JEDEC JS-001 - HBM)	V_{ESD}	2	kV
Thermal resistance junction - solder point	R_{thJS}	11	K / W

Note: For the forward current and power consumption please see "maximum permissible forward current" diagram

Characteristics ($T_A = 25\text{ °C}$)

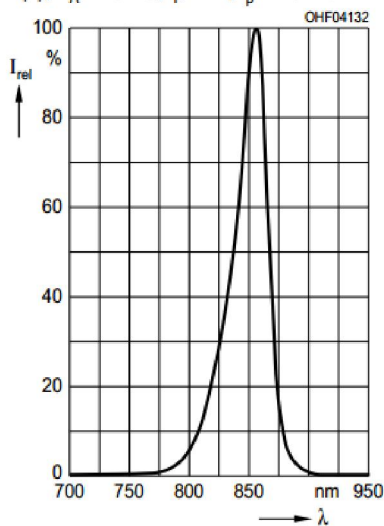
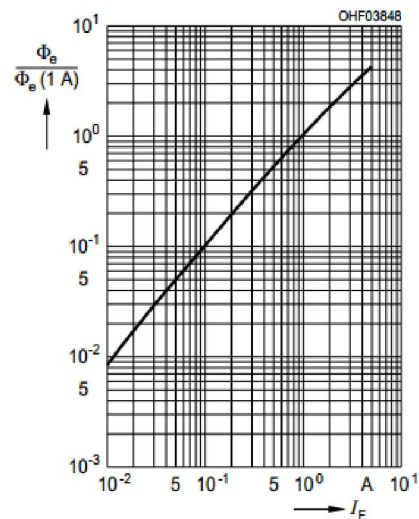
Parameter	Symbol	Values	Unit
Peak wavelength ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	λ_{peak}	860	nm
Centroid wavelength ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	$\lambda_{centroid}$	850	nm
Spectral bandwidth at 50% of I_{max} ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	$\Delta\lambda$	30	nm
Half angle	φ	± 45	°
Dimensions of active chip area	L x W	1 x 1	mm x mm
Rise and fall times of I_e (10% and 90% of $I_{e,max}$) ($I_F = 5\text{ A}$, $R_L = 50\text{ }\Omega$)	t_r / t_f	7/14	ns
Forward voltage ($I_F = 1\text{ A}$, $t_p = 100\text{ }\mu\text{s}$)	V_F	1.5 (≤ 1.8)	V
Forward voltage ($I_F = 5\text{ A}$, $t_p = 100\text{ }\mu\text{s}$)	V_F	2 (≤ 2.9)	V
Total radiant flux ($I_F = 1\text{ A}$, $t_p = 100\text{ }\mu\text{s}$)	Φ_e	630	mW

Parameter	Symbol	Values	Unit
Temperature coefficient of I_e or Φ_e ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	TC_I	-0.3	% / K
Temperature coefficient of V_F ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	TC_V	-1	mV / K
Temperature coefficient of wavelength ($I_F = 1\text{ A}$, $t_p = 10\text{ ms}$)	TC_λ	0.3	nm / K

Grouping ($T_A = 25\text{ °C}$)

Group	Min Radiant Intensity	Max Radiant Intensity
	$I_F=1\text{ A}$, $t_p=10\text{ ms}$ $I_{e, \min}$ [mW / sr]	$I_F=1\text{ A}$, $t_p=10\text{ ms}$ $I_{e, \max}$ [mW / sr]
SFH4715 - BA	160	250
SFH4715 - BB	200	320
SFH4715 - CA	250	400
SFH4715 - CB	320	500

Note: measured at a solid angle of $\Omega = 0.01\text{ sr}$
Only one group in one packing unit (variation lower 1.6:1).

Relative Spectral Emission ^{1) page 13}
 $I_{rel} = f(\lambda)$, $T_A = 25\text{ °C}$, $I_F = 1\text{ A}$, $t_p = 10\text{ ms}$
Relative Total Radiant Flux ^{1) page 13}
 $\Phi_e / \Phi_e(1\text{ A}) = f(I_F)$, $T_A = 25\text{ °C}$, Single pulse, $t_p = 100\text{ }\mu\text{s}$


Description

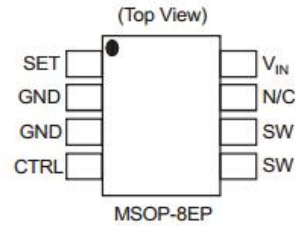
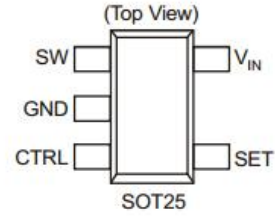
The AL8807 is a step-down DC/DC converter designed to drive LEDs with a constant current. The device can drive up to 9 LEDs, depending on the forward voltage of the LEDs, in series from a voltage source of 6V to 36V. Series connection of the LEDs provides identical LED currents resulting in uniform brightness and eliminating the need for ballast resistors. The AL8807 switches at frequency up to 1MHz with controlled rise and fall times to reduce EMI. This allows the use of small size external components, hence minimizing the PCB area needed.

Maximum output current of AL8807 is set via an external resistor connected between the V_{IN} and SET input pins. Dimming is achieved by applying either a DC voltage or a PWM signal at the CTRL input pin. An input voltage of 0.4V or lower at CTRL switches off the output MOSFET simplifying PWM dimming.

Features

- LED Driving Current up to 1.3A (MSOP-8EP)
- Better Than 5% Accuracy
- High Efficiency up to 96%
- Optimally Controlled Switching Speeds
- Operating Input Voltage from 6V to 36V
- PWM/DC Input for Dimming Control
- Built-In Output Open-Circuit Protection
- SOT25, MSOP-8EP: Available in "Green" Molding Compound (No Br, Sb)
- **Totally Lead-Free & Fully RoHS Compliant (Notes 1 & 2)**
- **Halogen and Antimony Free. "Green" Device (Note 3)**

Pin Assignments

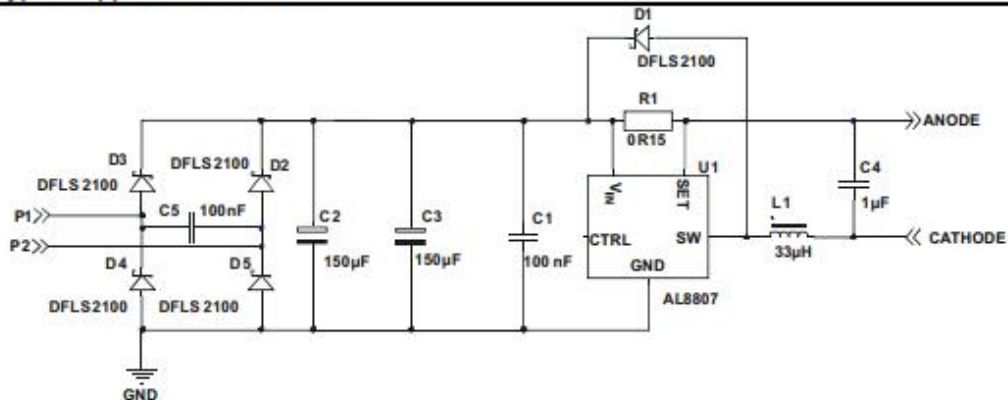


Applications

- MR16 Lamps
- General Illumination Lamps
- 12V Powered LED Lamps
- 24V Powered LED Lamps

Notes: 1. No purposely added lead. Fully EU Directive 2002/95/EC (RoHS) & 2011/65/EU (RoHS 2) compliant.
2. See http://www.diodes.com/quality/lead_free.html for more information about Diodes Incorporated's definitions of Halogen- and Antimony-free, "Green" and Lead-free.
3. Halogen- and Antimony-free "Green" products are defined as those which contain <900ppm bromine, <900ppm chlorine (<1500ppm total Br + Cl) and <1000ppm antimony compounds.

Typical Applications Circuit



Electrical Characteristics (@ $V_{IN} = 12V$, $T_A = +25^\circ C$, unless otherwise specified.)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{INSU}	Internal Regulator Start-Up Threshold	V_{IN} rising			5.9	V
V_{INSH}	Internal Regulator Hysteresis Threshold	V_{IN} falling	100		300	mV
I_Q	Quiescent Current	Output not switching (Note 4)			350	μA
I_S	Input Supply Current	CTRL pin floating $f = 250kHz$		1.8	5	mA
V_{TH}	Set current Threshold Voltage		95	100	105	mV
V_{THH}	Set Threshold Hysteresis			± 20		mV
I_{SET}	SET Pin Input Current	$V_{SET} = V_{IN} - 0.1$		16	22	μA
R_{CTRL}	CTRL Pin Input Resistance	Referred to internal reference		50		k Ω
V_{REF}	Internal Reference Voltage			2.5		V
$R_{DS(on)}$	On Resistance of SW MOSFET	$I_{SW} = 1A$		0.25	0.4	Ω
t_R	SW Rise Time	$V_{SENSE} = 100 \pm 20mV$, $f_{SW} = 250kHz$		12		ns
t_F	SW Fall Time	$V_{SW} = 0.1V$ to $12V$, $C_L = 15pF$		20		ns
$I_{SW_Leakage}$	Switch Leakage Current	$V_{IN} = 30V$			0.5	μA
θ_{JA}	Thermal Resistance Junction-to-Ambient (Note 5)	SOT25 (Note 6)		250		°C/W
		MSOP-8EP (Note 7)		69		
Ψ_{JL}	Thermal Resistance Junction-to-Lead (Note 8)	SOT25 (Note 6)		50		
θ_{JC}	Thermal Resistance Junction-to-case (Note 9)	MSOP-8EP (Note 7)		4.3		

- Notes:
- AL8807 does not have a low power standby mode but current consumption is reduced when output switch is inhibited: $V_{SENSE} = 0V$. Parameter is tested with $V_{CTRL} \leq 2.5V$
 - Refer to figure 35 for the device derating curve.
 - Test condition for SOT25: Device mounted on FR-4 PCB (25mm x 25mm 1oz copper, minimum recommended pad layout on top layer and thermal vias to bottom layer ground plane. For better thermal performance, larger copper pad for heat-sink is needed.
 - Test condition for MSOP-8EP: Device mounted on FR-4 PCB (51mm x 51mm 2oz copper, minimum recommended pad layout on top layer and thermal vias to bottom layer with maximum area ground plane. For better thermal performance, larger copper pad for heat-sink is needed
 - Dominant conduction path via Gnd pin (pin 2).
 - Dominant conduction path via exposed pad.

Typical Performance Characteristics

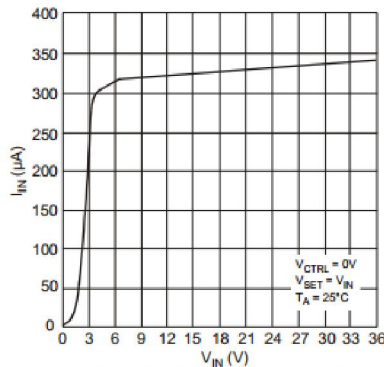


Figure 1. Supply Current (not switching) vs. Input Voltage

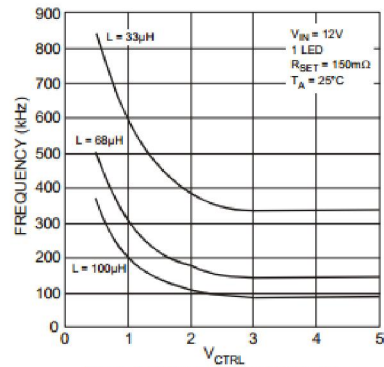


Figure 2. Switching Frequency vs. V_{CTRL}

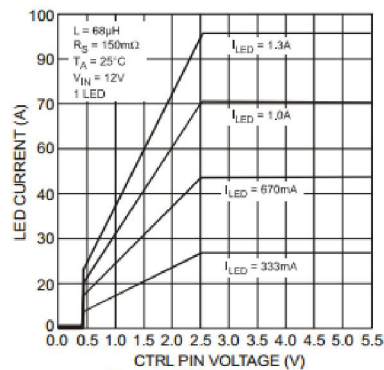


Figure 3. LED Current vs. V_{CTRL}

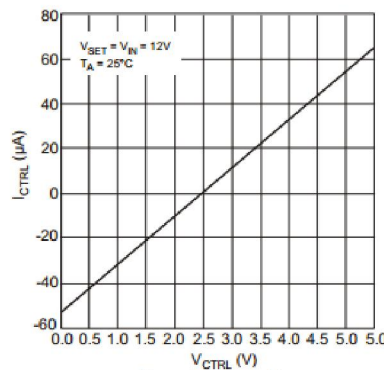


Figure 4. I_{CTRL} vs. V_{CTRL}

Application Information (cont.)

Recommendations for minimising radiated EMI and other transients and thermal considerations are:

1. The decoupling capacitor (C1) has to be placed as close as possible to the V_{IN} pin and D1 Cathode
2. The freewheeling diode's (D1) anode, the SW pin and the inductor have to be placed as close as possible to each other to avoid ringing.
3. The Ground return path from C1 must be a low impedance path with the ground plane as large as possible
4. The LED current sense resistor (R1) has to be placed as close as possible to the V_{IN} and SET pins.
5. The majority of the conducted heat from the AL8807 is through the GND pin 2. A maximum earth plane with thermal vias into a second earth plane will minimise self-heating
6. To reduce emissions via long leads on the supply input and LEDs low RF impedance capacitors (C2 and C5) should be used at the point the wires are joined to the PCB

A typical application for the AL8807 is an LED MR16 lamp (schematic shown in Figure 38).

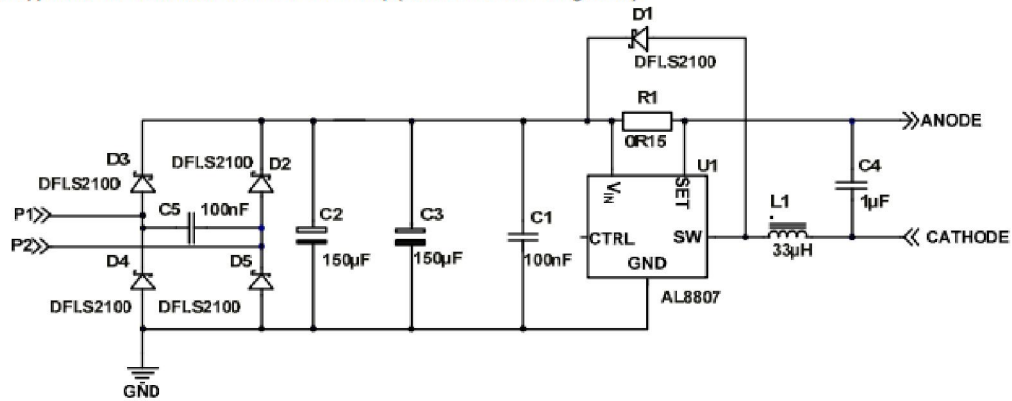


Figure 38 MR16 Circuit Schematic

An evaluation board for the AL8807 (named the AL8807EV2) for MR16 is available on request from your local Diodes' sales representative. This board follows Diodes' recommendations for low EMI. Images of the top layer and bottom layers are shown in Figure 39.

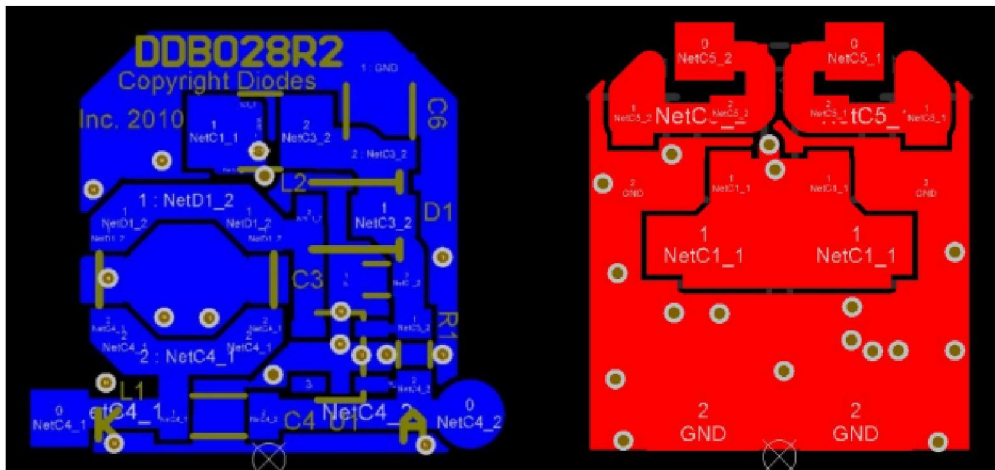


Figure 39 Recommended MR16 PCB Layout

Anexo 2: Código del descriptor de contorno e identificación gestual

En este anexo se presenta un fragmento código principal de Python cuya finalidad es detección de la mano en imágenes con luz IR. Este fragmento no incluye el algoritmo para realización del interfaz



```
## cv2.imshow('drawing', drawing)
```

```
    if count_defects == 1:
        cv2.putText(img, "esto es un dos", (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
    elif count_defects == 2:
        str = "esto es un tres"
        cv2.putText(img, str, (5,50), cv2.FONT_HERSHEY_SIMPLEX, 1,
2)
    elif count_defects == 3:
        cv2.putText(img, "esto es un 4", (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
    elif count_defects == 4:
        cv2.putText(img, "esto es un 5", (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
    else:
        cv2.putText(img, "esto es un uno", (50,50), \
cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
```

Detección del número de dedos.

```
## cv2.imshow('end', crop_img)
## cv2.imshow('Gesture', img)
all_img = np.hstack((drawing, crop_img))
cv2.imshow('Contours', all_img)
k = cv2.waitKey(5) & 0xFF
if k == 27:
    break
```

Función: mostrar en pantalla.

```
cv2.destroyAllWindows()
```

Anexo 3: Código completo del descriptor y el simulador de cocina de inducción

Este es el código completo utilizado en las simulaciones de las cocinas de inducción. Se puede ver como en este caso se realiza clic realizando el gesto “uno” con la mano. En la realización de los experimentos se optó por navegar formando un dos con la mano y juntar los dedos para realizar clic, no obstante, las modificaciones de los gestos asignados a cada orden se pueden modificar de forma sencilla. Las funciones correspondientes a la biblioteca. Las funciones correspondientes a la biblioteca Pygame son fácilmente reconocibles y permiten mostrar en pantalla la interfaz de usuario.

```
import cv2
import numpy as np
import math
import pygame
import pygame, sys
from pygame.locals import *
from pygame import Color

pygame.init()
(sx,sy)=(1200,700)
pantalla = pygame.display.set_mode([1200,700],pygame.SRCALPHA,32)
raton=False
Salir = False
color=(255,0,0,100)
radiol=55
radio2=105
i=0
j=0
k=0
l=0
mx=0
my=0
cx=0
cy=0
ancho=240
largo=240
arriba=True
abajo=False
tecla=False
centroid=tuple()

rojo_surface = pygame.Surface((ancho, largo), pygame.SRCALPHA, 32)
rojo_surface.fill((0,0,0,0))

cnt1=i
cnt2=j
cnt3=k
cnt4=l
imagen = pygame.image.load("cocina_1.jpg")
pantalla.fill((255,255,255))
pantalla.blit(imagen, (0,0))
```

```

while Salir == False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            Salir = True
            pygame.quit()
            Sys.exit()
##     pygame.display.update()

cap = cv2.VideoCapture(0)
while(cap.isOpened()):
    ret, img = cap.read()
    (camx,camy)=(320,240)
    img=cv2.resize(img,(camx,camy))
    cv2.rectangle(img,(900,900),(000,000),(0,255,0),0)
    crop_img = img[000:900, 000:900]
    grey = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
    value = (35, 35)
    blurred = cv2.GaussianBlur(grey, value, 0)
    _, thresh1 = cv2.threshold(blurred, 127, 255,

cv2.THRESH_BINARY+cv2.THRESH_OTSU)
## cv2.imshow('Thresholded', thresh1)

    contours, hierarchy =
cv2.findContours(thresh1.copy(),cv2.RETR_TREE, \
cv2.CHAIN_APPROX_NONE)

    cnt = max(contours, key = lambda x: cv2.contourArea(x))

    x,y,w,h = cv2.boundingRect(cnt)
    cv2.rectangle(crop_img,(x,y),(x+w,y+h),(0,0,255),0)

    hull = cv2.convexHull(cnt)
    drawing = np.zeros(crop_img.shape,np.uint8)
    cv2.drawContours(drawing,[cnt],0,(0,255,0),0)
    cv2.drawContours(drawing,[hull],0,(0,0,255),0)
    hull = cv2.convexHull(cnt,returnPoints = False)
    defects = cv2.convexityDefects(cnt,hull)
    count_defects = 0
    cv2.drawContours(thresh1, contours, -1, (0,255,0), 3)

    for i in range(defects.shape[0]):

        if raton == True:
##             print("raton activo")
                pygame.mouse.set_pos(sx-(cx*sx/camx),cy*sy/camy)

        if cv2.waitKey(1) & 0xFF == ord('a'):
            raton = True
        elif cv2.waitKey(1) & 0xFF == ord('s'):
            raton = False
##             print("raton inactivo")

```

```

        s,e,f,d = defects[i,0]
        start = tuple(cnt[s][0])
        end = tuple(cnt[e][0])
        far = tuple(cnt[f][0])
        a = math.sqrt((end[0] - start[0])**2 + (end[1] -
start[1])**2)
        b = math.sqrt((far[0] - start[0])**2 + (far[1] -
start[1])**2)
        c = math.sqrt((end[0] - far[0])**2 + (end[1] -
far[1])**2)
        angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57
        if angle <= 90:
            count_defects += 1
            cv2.circle(crop_img,far,1,[0,140,0],30)
##            print(far)
##            print(far[0])
##            print(far[1])
            cx=far[0]
            cy=far[1]
            centroid=(cx,cy)
            cv2.circle(crop_img,centroid,5,[255,0,255],2)

            dist = cv2.pointPolygonTest(cnt,far,True)
            linea = cv2.line(crop_img,start,end,[0,255,0],2)
            cv2.circle(crop_img,far,5,[0,0,255],-1)

if cv2.waitKey(1) & 0xFF == ord('p'):
    tecla=True
    print("tecla habilitada")

if count_defects == 1:
    cv2.putText(img,"esto es un dos", (20,20),
cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
elif count_defects == 2:
    str = "esto es un tres"
    cv2.putText(img, str, (20,20),
cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
elif count_defects == 3:
    cv2.putText(img,"esto es un 4", (20,20),
cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
elif count_defects == 4:
    cv2.putText(img,"esto es un 5", (20,20),
cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
else:
    cv2.putText(img,"esto es un uno", (20,20),\
cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
if tecla==True:
    mx, my = pygame.mouse.get_pos()
    print("click")
    print(mx, my)

```

```

cv2.imshow('silueta', drawing)
cv2.imshow('mano', crop_img)
## k = cv2.waitKey(5) & 0xFF
## if k == 27:
##     break
## while Salir == False:

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            Salir = True
##         pygame.quit()
##         sys.exit()

    if (mx<640) and (my>570) and (my<618):
        print("has cambiado a ciclo descendente")
        arriba=False
        abajo=True
    if (mx>640) and (my>570) and (my<618):
        print("has cambiado a ciclo ascendente")
        arriba=True
        abajo=False

    if (mx>294) and (mx<442) and (my>127) and (my<266) and
tecla:
        tecla=False
        px1=368
        py1=202
        position=(px1,py1)
        print("circulo grande en ",position)
        pygame.draw.circle(rojo_surface, color, (ancho/2,
ancho/2),radio2)
        pantalla.blit(rojo_surface, (px1-ancho/2, py1-ancho/2))
        if arriba==True:
            cnt1=cnt1+1
        if abajo==True:
            cnt1=cnt1-1
##         font = pygame.font.SysFont("Showcard Gothic",24)
##         text = font.render(str(cnt1), True, (0,0,0))
##         pantalla.blit(text, (px1,py1))

    if (mx>830) and (mx<970) and (my>355) and (my<487) and
tecla:
        tecla=False
        px2=900
        py2=411
        position=(px2,py2)
        print("circulo grande en ",position)
        pygame.draw.circle(rojo_surface, color, (ancho/2,
ancho/2),radio2)
        pantalla.blit(rojo_surface, (px2-ancho/2, py2-ancho/2))
        if arriba==True:
            cnt2=cnt2+1

```

```

        if abajo==True:
            cnt2=cnt2-1

##         font = pygame.font.SysFont("Showcard Gothic",24)
##         text = font.render(str(cnt2), True, (0,0,0))
##         pantalla.blit(text, (px2,py2))

        if (mx>336) and (mx<408) and (my>385) and (my<447) and
tecla:
            tecla=False
            px3=368
            py3=411
            position=(px3,py3)
            print("circulo pequeno en ",position)
            pygame.draw.circle(rojo_surface, color, (ancho/2,
ancho/2),radiol)
            pantalla.blit(rojo_surface, (px3-ancho/2, py3-ancho/2))

            if arriba==True:
                cnt3=cnt3+1
            if abajo==True:
                cnt3=cnt3-1

##         font = pygame.font.SysFont("Showcard Gothic",24)
##         text = font.render(str(cnt3), True, (0,0,0))
##         pantalla.blit(text, (px3,py3))

        if (mx>854) and (mx<940) and (my>165) and (my<239) and
tecla:
            tecla=False
            px4=902
            py4=203
            position=(px4,py4)
            print("circulo pequen en ",position)
            pygame.draw.circle(rojo_surface, color, (ancho/2,
ancho/2),radiol)
            pantalla.blit(rojo_surface, (px4-ancho/2, py4-ancho/2))

            if arriba==True:
                cnt4=cnt4+1
            if abajo==True:
                cnt4=cnt4-1

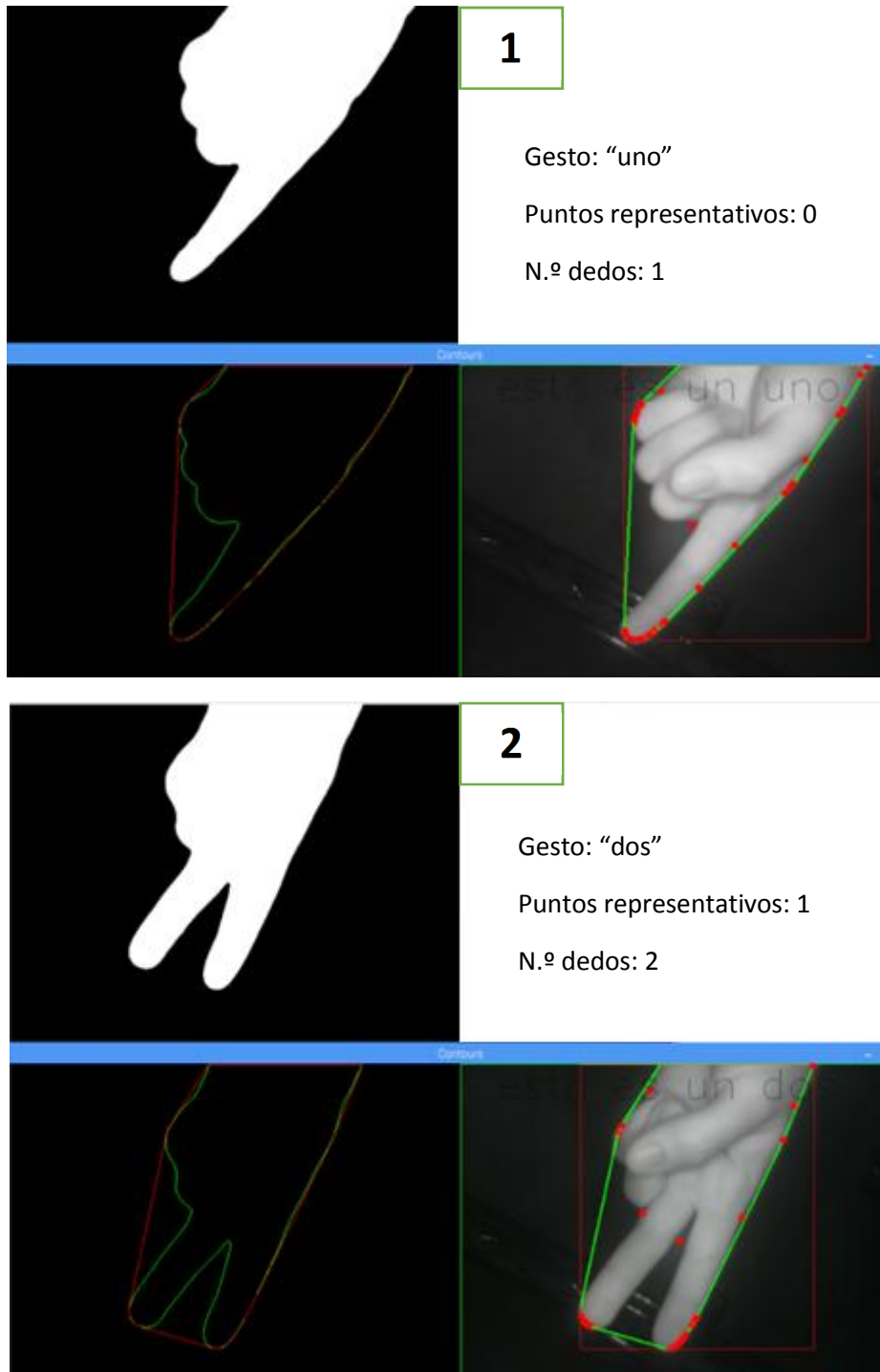
##         font = pygame.font.SysFont("Showcard Gothic",24)
##         text = font.render(str(cnt4), True, (0,0,0))
##         pantalla.blit(text, (px4,py4))

    mx=0
    my=0
    pygame.display.update()
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break
cv2.destroyAllWindows()

```


Anexo 4: Resultados experimentales

A continuación se muestran ejemplos de detección de diferentes gestos. Las figuras del 1-5 corresponden con el gesto indicando diferente número de dedos, mientras que las figuras 6, 7 y 8 corresponden a gestos que no representan números, para analizar como trabaja el algoritmo en diferentes escenarios.



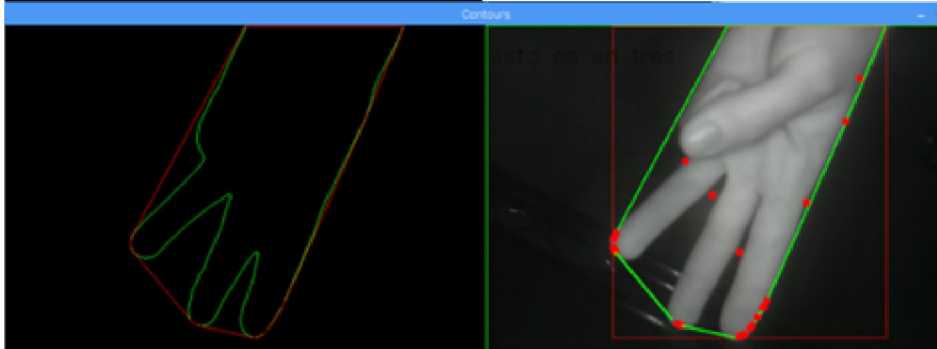


3

Gesto: "tres"

Puntos representativos: 2

N.º dedos: 3

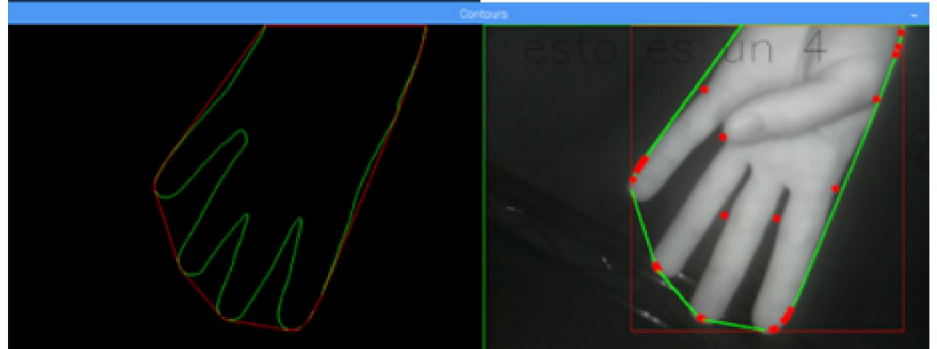


4

Gesto: "cuatro"

Puntos representativos: 3

N.º dedos: 4



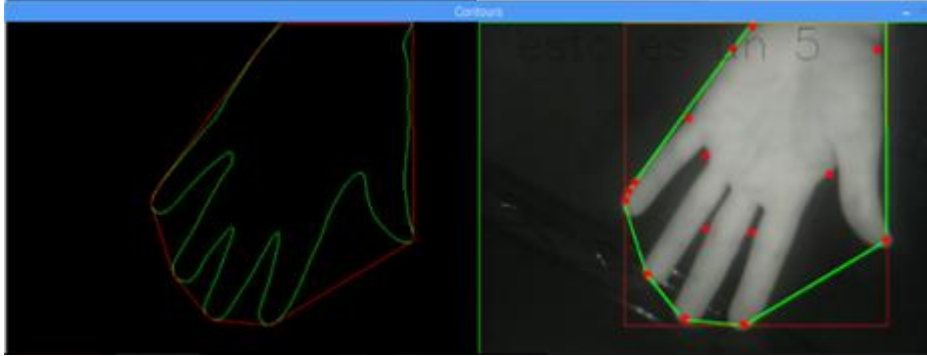


5

Gesto: "cinco"

Puntos representativos: 4

N.º dedos: 5

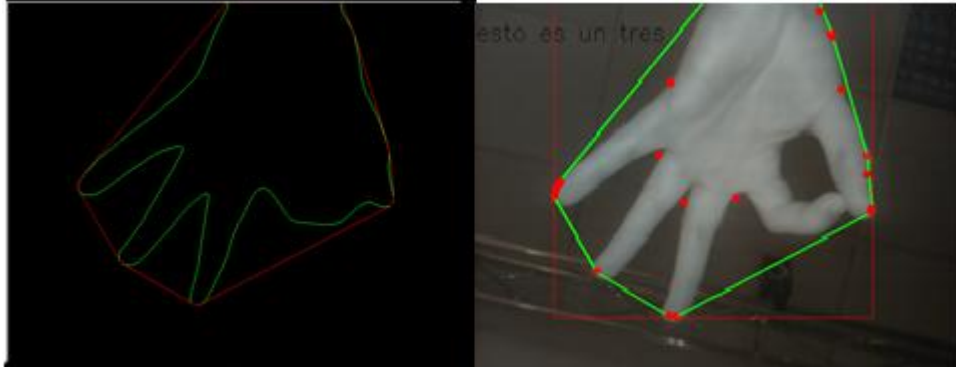


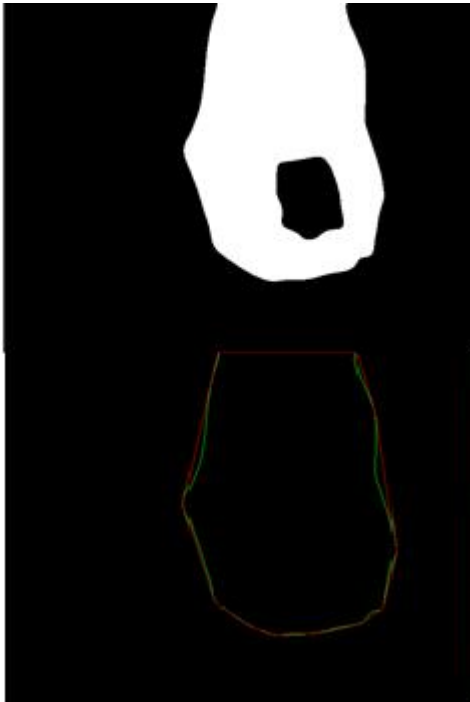
6

Gesto: "OK"

Puntos representativos: 2

N.º dedos: 3





7

Gesto: "nada"

Puntos representativos: 1 (entre el 0 y el uno pueden existir errores).

N.º dedos: 1



8

Gesto: "mano de perfil"

Puntos representativos: 1

N.º dedos: 2



