



industriales
etsii

**Escuela Técnica
Superior
de Ingeniería
Industrial**

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Sistema de detección de objetos mediante cámaras de tiempo de vuelo (ToF). Aplicación en conducción autónoma

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES



**Universidad
Politécnica
de Cartagena**

Autor: Antonio Ros García
Director: Pedro Javier Navarro Lorente
Codirector: Carlos Fernández Andrés

Cartagena, diciembre de 2017

Resumen

El objetivo de este trabajo es el diseño, implementación y evaluación de un sistema de detección robusta de objetos, empleando únicamente la información adquirida con una cámara de tiempo de vuelo (ToF). Para la detección, y su discriminación frente a otros elementos que puedan aparecer en la escena, se han estudiado diferentes descriptores y clasificadores. Para la evaluación del algoritmo se ha empleado imágenes obtenidas con la cámara de tiempo de vuelo (Sentis3D-M420) del grupo de investigación DSIE de la universidad Politécnica de Cartagena .El sistema ha sido diseñado para colocarse en la zona de ángulo muerto de un vehículo pudiendo utilizarse como un asistente avanzado de ayuda al conductor (ADAS) o integrándolo como parte un sistema de conducción autónoma.

Abstract

The objective of this project is the design, implementation and evaluation of a robust object detection system, using only the information acquired with a time-of-flight camera (ToF). For detection, and its discrimination against other elements that may appear in the scene, different descriptors and classifiers have been studied. To evaluate the algorithm, images obtained with the time-of-flight camera (Sentis3D-M420) of the DSIE research group of the Polytechnic University of Cartagena have been used. The system has been designed to be placed in the blind spot area of a vehicle. It can be used as an advanced driver assistance system (ADAS) or integrated as part of an autonomous driving system.

AGRADECIMIENTOS

Quisiera agradecer a varias personas la ayuda que me han prestado en la realización de este Trabajo de Fin de Grado.

En primer lugar, a mi director, Pedro Javier Navarro por su inestimable ayuda y enseñanzas, sin las cuales hubiera sido muy difícil la realización de este trabajo. Ante todas las dificultades que se han presentado, siempre ha estado dispuesto a dedicarme su tiempo y su conocimiento. Se lo agradezco de corazón.

También me gustaría agradecer a mis padres por sus sabios consejos y su comprensión. Siempre habéis estado ahí para ayudarme y sin ellos nada de esto hubiera sido posible.

A Marielo, por todo su apoyo durante estos los últimos meses del proyecto que ha hecho que a pesar de los duros días de trabajo, siempre me queden fuerzas para continuar.

Muchas gracias a todos.

Antonio Ros Garcia

Cartagena, 22 de Diciembre de 2017.

Índice de contenidos

Resumen	iii
Abstract	v
Capítulo 1	1
<i>Introducción y objetivos</i>	1
1.1 <i>Introducción</i>	2
1.2 <i>Objetivos del proyecto</i>	2
1.3 <i>Objetivos</i>	3
1.4 <i>Descripción de los capítulos del trabajo</i>	4
Capítulo 2	7
<i>Estado del arte</i>	7
2.1 <i>Introducción</i>	8
2.3 <i>Un breve recorrido por la historia</i>	8
2.4 <i>Sensores de profundidad</i>	9
2.4.1 <i>Sensores de profundidad basados en luz estructurada</i>	10
2.4.2 <i>Sensores de profundidad basados en tiempo de vuelo</i>	11
2.5 <i>Ruido en las imágenes de profundidad</i>	14
2.6 <i>Filtrado de imágenes</i>	16
2.7 <i>Descriptores</i>	17
2.7.1 <i>Patrón binario local</i>	18
2.7.2 <i>Histograma de gradientes orientados</i>	19
2.7.3 <i>Filtros Haar</i>	20
2.8 <i>Clasificadores</i>	21
2.8.1 <i>Tipos de clasificadores</i>	21
2.8.2 <i>Precisión del clasificador</i>	24
Capítulo 3	29
<i>Descripción y calibración del sensor ToF</i>	29
3.1 <i>Introducción</i>	30
3.2 <i>Selección de la cámara ToF</i>	30
3.3 <i>Información técnica Sentis 3D-M420</i>	31
3.4 <i>Parámetros: Tiempo de integración y frecuencia de modulación</i>	33
3.5 <i>Pruebas de calibración</i>	34

3.5.1	Pruebas con un campo de visión de 90º.....	35
3.5.2	Pruebas con un campo de visión de 110º.....	39
3.6	<i>Calibración</i>	41
Capítulo 4	45
	<i>Desarrollo Software/Hardware</i>	45
4.1	<i>Introducción</i>	46
4.2	<i>Software de captura de datos</i>	46
4.2	<i>Hardware de captura de datos</i>	49
4.3	<i>Extracción de características</i>	51
4.4	<i>Entrenamiento del clasificador</i>	54
4.6	<i>Proceso de medición de distancia al vehículo</i>	58
4.7	<i>Pruebas del clasificador</i>	60
Capítulo 5	63
	<i>Conclusión y trabajos futuros</i>	63
5.1	<i>Introducción</i>	64
5.2	<i>Conclusión</i>	64
5.3	<i>Trabajos futuros</i>	65
Bibliografía	67
Anexo I	69
	<i>Programa de captura de datos SENTIS3D-M420</i>	70
	<i>Programa de reproducción de datos SENTIS3D-M420</i>	72
Anexo II	75
	<i>Pruebas del clasificador</i>	75

CAPÍTULO 1

INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

Este capítulo tiene como finalidad el planteamiento del presente Trabajo Fin de Grado, los objetivos que con él se persiguen y una breve descripción de los diferentes capítulos que conforman esta memoria.

Dicen que el sentido de la vista fue una de las mayores dudas de Darwin al apostar por su teoría de la evolución. No fue hasta que encontró especies marinas con ojos menos evolucionados cuando realmente se convenció de su propia hipótesis. Para el homosapiens, la vista es el principal mecanismo de obtención de información del entorno. Como humanos percibimos la estructura tridimensional del mundo que nos rodea con facilidad. Prueba de ello lo constatamos al mirar a nuestro alrededor. Somos capaces de identificar los objetos que nos rodean y distinguir los puntos que pertenecen a un objeto de los que corresponden al entorno. Psicólogos han dedicado décadas a intentar entender cómo funciona nuestro sistema visual para poder aplicarlo a las máquinas. La solución al puzle sigue incompleta.

¿Por qué la visión es algo tan complicado? Debemos distinguir entre visión y percepción. Cualquier cámara del mundo es capaz de ver el mundo exactamente como nosotros, pero no es capaz de interpretarlo. En parte, es debido a que la visión es un problema inverso. Intentamos encontrar las incógnitas que nos den la información para solucionar el problema por completo y por ello recurrimos a modelos físicos y de probabilidad para diferenciar entre las posibles soluciones.

La motivación detrás de este trabajo es llegar a entender parte de la tecnología que consigue que un coche sea autónomo, esta tecnología me resulta fascinante, concretamente la parte de visión artificial. En los últimos años se ha producido un avance espectacular en el campo de conducción autónoma teniendo ya a la venta vehículos que son capaces de mantenerse en el carril, tomar el control en un atasco, adaptar la velocidad a la señalada o al vehículo que va delante como es el caso del autopilot de Tesla, no cabe ninguna duda que de aquí a unos pocos años tendremos circulando en nuestras calles vehículos totalmente autónomos que no necesiten la intervención humana

Uno de los procesos esenciales para conseguirlo es poder describir el entorno. Llegar al nivel de visión de los humanos es muy complicado, pero con el paso de los años se harán sistemas muy robustos y quien sabe incluso mejores que la visión humana.

1.2 Objetivos del proyecto

Para el ser humano, la vista es el principal mecanismo de obtención de información del entorno, hasta el punto que al conducir extendemos el campo de visión mediante espejos a modo de retrovisores. Pero no llegan a proporcionar una visión omnidireccional, siempre queda un ángulo muerto.

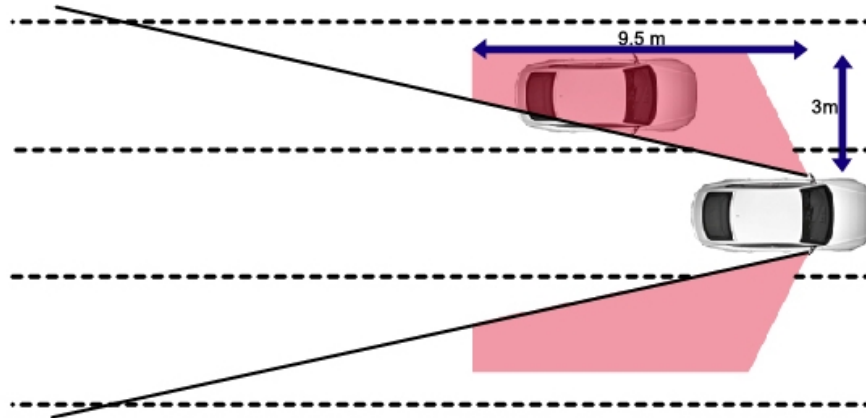


Figura 1.1 Ángulo muerto

Como se puede ver en la Figura 1.1, la zona en color rosa es la zona ciega. Cuando un vehículo o peatón se encuentre en esta zona no será posible visualizarlo y si tenemos en cuenta que circulamos con una cierta velocidad es muy posible que cuando veamos el vehículo ya no podremos hacer nada para evitar la colisión de ahí la importancia de saber en todo momento que es lo que se encuentra cerca de nuestro vehículo aunque no lo podamos ver con nuestros propios ojos

Según un estudio de la Unión Europea, más del 10% de los accidentes que se producen en nuestras carreteras son por cambios de carril erráticos debido al ángulo muerto. En el proyecto se pretende crear un sistema de detección de objetos mediante una cámara de tiempo de vuelo en el ángulo muerto de un vehículo.

La principal ventaja de utilizar este tipo de cámaras es poder conocer a que distancia se encuentran los objetos del vehículo, aunque tiene una gran desventaja respecto a las cámaras convencionales pues su resolución es mucho menor que estas últimas, lo que puede dificultar bastante la tarea de clasificar un objeto debido a la disminución de información que obtenemos con cada imagen.

El objetivo de este proyecto es crear un algoritmo que pueda utilizarse como un sistema avanzado de asistencia al conductor (ADAS) para la zona del ángulo muerto, así cuando un objeto se encuentre en la zona de peligro dar un aviso al conductor o ser integrado como parte de un sistema de conducción autónoma.

1.3 Objetivos

El objetivo del trabajo es diseñar e implementar un algoritmo con funcionamiento autónomo capaz de identificar objetos y conocer la distancia a la que se encuentran del vehículo. Para lograr este fin se han establecido una serie de objetivos:

- 1) Estudio del estado de la técnica sobre sistemas de detección de objetos en sistemas vehiculares.

La elección del hardware y software empleado implica la tarea más importante. Una vez seleccionados se procede a su análisis y estudio.

2) Calibración de la cámara ToF.

Se realizará la calibración para conseguir la medición más precisa posible en los diferentes escenarios.

3) Estudio y análisis de los algoritmos de detección de objetos basados en visión artificial.

Se estudiarán los diferentes algoritmos capaces de detectar un objeto con la cámara viendo cual resulta más adecuado para el objetivo del proyecto.

4) Estudio de algoritmos de aprendizaje supervisado.

Se analizan los principales algoritmos y se elegirá el que mejor se adapte a las necesidades del proyecto.

5) Desarrollar el software de procesamiento de imágenes.

Desde la web del fabricante se puede descargar la información relativa al modelo en concreto de la cámara que se va a usar, así como el software necesario y el kit de desarrollador con librerías, programas y ejemplos necesarios para su integración.

6) Realizar pruebas del software y comprobar que se han cumplido los objetivos.

Con objeto de comprobar los resultados del proyecto se realizarán pruebas en un ambiente real de uso.

1.4 Descripción de los capítulos del trabajo

El Trabajo Fin de Grado se estructura en los siguientes capítulos:

1^{er} Capítulo: Introducción y objetivos.

En este capítulo se describe la motivación que ha llevado a la realización de este trabajo, los objetivos que se persiguen y su estructuración.

2^o Capítulo: Estado del arte.

El objetivo de este capítulo es la presentación de la técnica que conforma el trabajo. Se mostrarán los diferentes dispositivos con los que se podría llevar a cabo el proyecto, las diferentes técnicas de detección de objetos y los diferentes clasificadores.

3^{er} Capítulo: Descripción y calibración del sensor ToF.

En este capítulo se verán diferentes dispositivos con los que medir la distancia de profundidad, se estudiara en profundidad la cámara ToF y se llevara a cabo la calibración.

4º Capítulo: Desarrollo del Software/Hardware.

Se explicara cómo se han obtenido los datos necesarios para la realización del proyecto, como se realiza la extracción de características de las imágenes, como se ha entrenado el clasificador para obtener un resultado óptimo y por ultimo como se realiza el proceso de medida de distancia al objeto clasificado.

5º Capítulo: Pruebas, conclusiones y trabajos futuros.

Se presentaran las pruebas realizadas sobre imágenes en condiciones reales como las conclusiones del presente trabajo fin de grado y se propondrán mejoras al proceso que se pueden abordar mediante otros trabajos y trabajos que puedan continuar al presente trabajo.

6º Capítulo: Bibliografía.

Conjunto de manuales, libros y sitios web consultados para la elaboración tanto de la parte práctica como de la escritura de la memoria.

CAPÍTULO 2

ESTADO DEL ARTE

2.1 Introducción

Este apartado del TFG se explica en profundidad los fundamentos teóricos necesarios para su elaboración. Así, en los siguientes apartados se va a detallar los tipos de sensores de profundidad existentes y su tecnología de medida, viendo en mayor profundidad el que vamos a utilizar en este trabajo. A continuación se detallaran los descriptores y clasificadores más relevantes que permitan distinguir un objeto en la imagen.

2.3 Un breve recorrido por la historia

La visión artificial, también conocida como “visión por computador” o “visión técnica” es un subcampo de la inteligencia artificial que tiene como finalidad la extracción de información del mundo físico a partir de imágenes, utilizando para ello un computador. La historia de la visión por ordenador comienza a principios de los años 60. Por aquel entonces se pensaba que resolver el problema de la visión sería mucho más sencillo.

Se dice que el padre de la visión artificial es Larry Roberts[1], el cual, en 1961 creó un programa que podía "ver" una estructura de bloques, analizar su contenido y reproducirla desde otra perspectiva, demostrando así a los espectadores que esa información visual que había sido mandada al ordenador por una cámara, había sido procesada adecuadamente por él.

Lo primero que se debía hacer para reconocer una escena era encontrar los bordes de los objetos que la componían, para así poder encontrar la estructura del objeto. En los 60 se empezaron a desarrollar algunos algoritmos en este aspecto[2][3].

En 1982, David Marr en su libro: *A Computational Investigation into the Human Representation and Processing of Visual Information* [4]. Trabaja a la visión como un sistema de procesamiento de información que se dividía en tres niveles.

- Nivel computacional: describe lo que hace el sistema (qué problemas tiene que resolver y superar) y por qué lo hace.
- Nivel de representación o de algoritmo: describe cómo se puede implementar esta teoría computacional, cual es la representación del input/output y cuál es el algoritmo para la transformada.
- Nivel físico: describe cómo puede realizarse físicamente el algoritmo y la representación.

Este acercamiento de Marr sigue siendo válido hoy en día.

En los años 80 se centraron en crear técnicas matemáticas más sofisticadas para así conseguir mejorar el análisis de escenas e imágenes. Se empezó a extender el uso de las pirámides de imágenes para extender la búsqueda de contorno de las imágenes procesadas. Paralelamente también se introdujo otro concepto para la detección del contorno y margen de la imagen [5].

En los noventa gracias a los avances de la tecnología se empezaron programas de reconocimiento de objetos y los problemas relacionados con el movimiento. Pero fue a partir

del año 2000 cuando se tomó más importancia la creación de algoritmos para la detección de objetos y la interpretación de escenas[6]. Este es el camino que se ha seguido en los últimos años pues es el principal objetivo de la visión artificial. La máquina tiene que ser capaz de analizar lo que la rodea para poder determinar las acciones correctas.

2.4 Sensores de profundidad

El objetivo principal del proyecto es obtener un dispositivo que sea capaz de “ver” y —

Actualmente las cámaras más conocidas son las cámaras de color, o Red, Green, Blue (RGB). Como se puede ver en la Figura 2.1, estas cámaras permiten obtener una imagen de color de la escena que visualizan, ello aporta gran cantidad de información de cara a la caracterización y búsqueda de objetos. Sin embargo, con este tipo de cámaras es complicado conocer la distancia a la que se encuentra un objeto.

A continuación se exponen algunos de los problemas que presentan:

- Una única cámara de color o RGB no permite obtener información espacial del entorno. En conjunto, dos cámaras RGB pueden constituir un par de visión estéreo que permite obtener información espacial a partir de cámaras de color pero implica una mayor complejidad en el sistema y además de ello ofrece datos menos fiables que una cámara de profundidad.
- Las cámaras de color no son invariantes a la iluminación, requieren una cantidad de iluminación mínima y concreta para ofrecer buenos resultados. Esto limita en gran medida las condiciones de utilización, no pudiendo emplearse entornos poco iluminados o nocturnos.

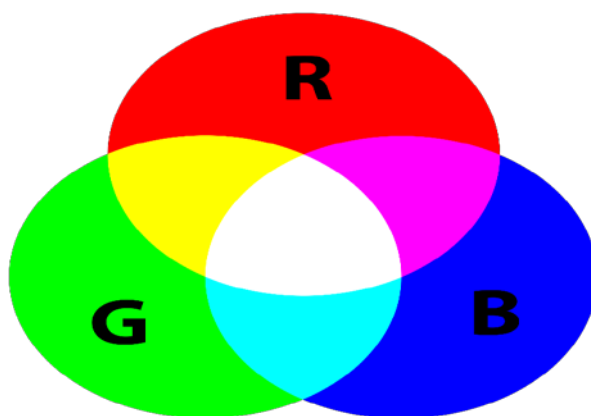


Figura 2.1 Canales RGB de cámara de color

Los inconvenientes mencionados implican la imposibilidad de trabajar con cámaras de color para la aplicación planteada, ya que es necesario saber con exactitud a la distancia que se encuentra el objeto.

Existen múltiples alternativas para la obtención de profundidad, dentro de los métodos más conocidos encontramos los sensores de profundidad basados en luz estructurada, y los

basados en tiempo de vuelo (ToF), estos últimos están actualmente en auge, debido a su gran demanda para diferentes aplicaciones, tales como la robótica móvil, la recuperación de la estructura a partir del movimiento, la detección y seguimiento de personas, o los videojuegos interactivos.

2.4.1 Sensores de profundidad basados en luz estructurada

Los sensores de luz estructurada se basan en la proyección de un patrón de luz, generalmente infrarroja, conocido, habitualmente en forma de red o puntos, como en el caso de la Kinect v1. Tras proyectar este patrón, la distancia se obtiene en función de la deformación del mismo al alcanzar los objetos y cuerpos de la imagen. La luz estructurada puede ser visible o invisible al ojo humano, la primera mencionada se emplea habitualmente para aplicaciones tales como virtualización de espacios, escaneo en tres dimensiones, etc.

La luz estructurada imperceptible al ojo humano se utiliza mediante patrones de luz infrarroja o patrones de luz a alta frecuencia. Este es el caso de la Kinect v1, de Microsoft o la Xtion Pro de Asus. En este caso la cámara utilizada posee un proyector del patrón, y un sensor sensible al infrarrojo.

En general, las cámaras de profundidad basadas en luz estructurada proporcionan, además de la profundidad, una imagen de intensidad o de color de la escena. Sin embargo, presentan algunas limitaciones, tales como la dependencia de la iluminación, o el rango de distancias a medir. Esto es debido a que para poder triangular el sensor de infrarrojos requiere ver el patrón proyectado, si esto no se diera entregaría una imagen de profundidad nula o errónea, así como una baja exactitud de las medidas, ya que la triangulación ofrece resultados con una baja calidad de medida. Por último, es muy destacable el ruido que se produce en estas cámaras debidas principalmente a oclusiones, dado que el receptor infrarrojo al estar situado a cierta distancia del emisor de patrón infrarrojo tiene un campo de visión diferente.

Como ya se ha comentado, entre las cámaras de luz infrarroja estructurada más conocidas, destaca la Kinect v1, que puede verse en la Figura 2.2, aunque hay que mencionar alternativas a la misma tales como la Astra Pro de Orbbec o la Xtion Pro de Asus. En la Figura 2.2 se observa como la cámara Kinect v1 dispone de un proyector de infrarrojos, un sensor receptor de infrarrojos y una cámara de alta calidad RGB. Tal como se ha mencionado anteriormente se servirá del sensor de infrarrojos y del proyector de los mismos para obtener la medida de profundidad.



Figura 2.2 Cámara Kinect

En la Figura 1.1 Figura 2.3, se puede observar un ejemplo del tipo de imágenes que entrega una cámara de luz estructurada. En la misma se observa a la izquierda la imagen en nivel de gris, cabe destacar que no es una imagen usual de infrarrojo debido a que se puede apreciar un cierto punteado que representa el patrón proyectado. En la imagen de la derecha puede verse la imagen de profundidad, a la que se ha aplicado un mapa de color para facilitar su interpretación. En este mapa el color blanco representa los puntos más cercanos mientras que el verde los puntos más alejados. Tal como se ha comentado, es posible observar una gran cantidad de puntos perdidos a causa de errores en la triangulación.

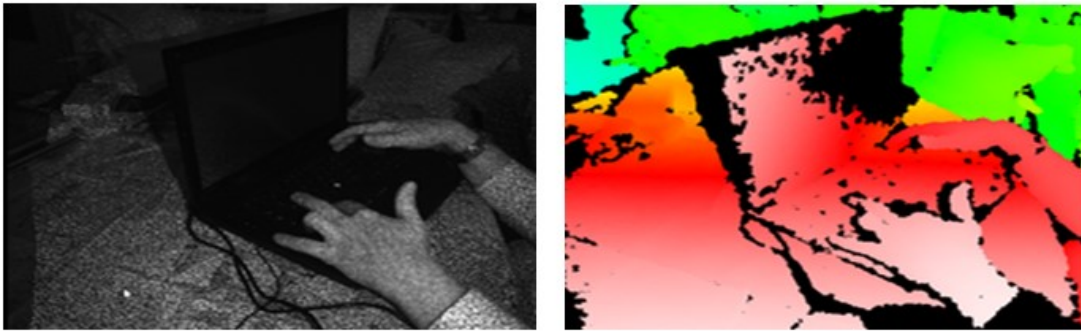


Figura 2.3 Ejemplo de imágenes de intensidad y profundidad proporcionadas por la Kinect

2.4.2 Sensores de profundidad basados en tiempo de vuelo

La tecnología 3D Time-of-Flight (ToF) está revolucionando la industria de la visión artificial debido a que son capaces de proporcionar imágenes en 3D utilizando un CMOS de bajo coste. Con una construcción compacta, fácil de usar junto con una alta precisión y velocidad de fotogramas hace de las cámaras ToF una solución atractiva en un amplio rango de aplicaciones como pueden ser: robótica, conteo de objetos, áreas de la salud, navegación, industria, automóviles, aviación[7].

Hay muchos tipos de ToF en la Figura 2.4 se puede ver una imagen de la cámara ToF SENTIS3D-M420.

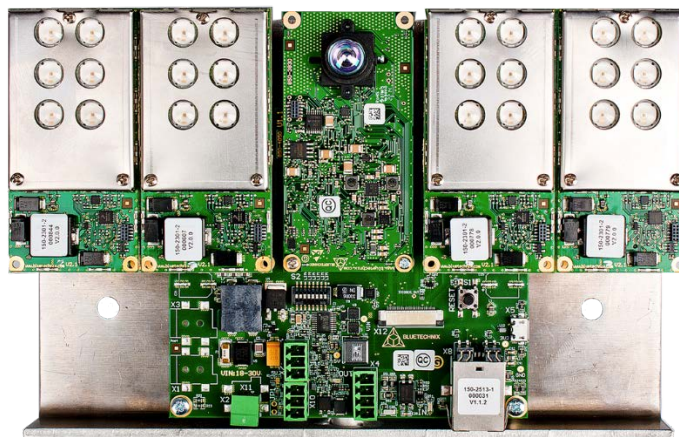


Figura 2.4 Cámara de tiempo de vuelo

Las cámaras de tiempo de vuelo realizan la medida de distancia sin contacto mediante un sensor activo óptico reflexivo. Básicamente, una cámara de tiempo de vuelo produce una imagen de profundidad en la cual cada píxel codifica la distancia al correspondiente punto de la escena, en lugar de indicar su color, como realizan las cámaras RGB. En la Figura 2.5 (derecha) se muestra un ejemplo de una imagen obtenida por una de la cámara de tiempo de vuelo utilizadas en el trabajo actual, en la que la escala de color se corresponde con la distancia, con respecto a su imagen en pseudo escala de grises (izquierda).

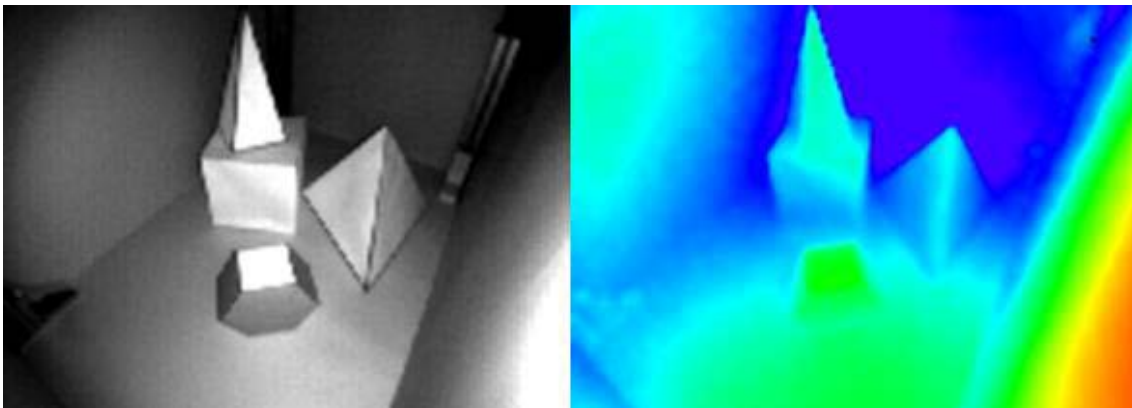


Figura 2.5 Izquierda: imagen en escala de grises. Derecha: imagen de distancias de una cámara ToF

Para comprender cómo se obtiene esta distancia, hay que estudiar el principio de medida de distancia mediante tiempo de vuelo, que aparece de forma gráfica en la Figura 2.6.

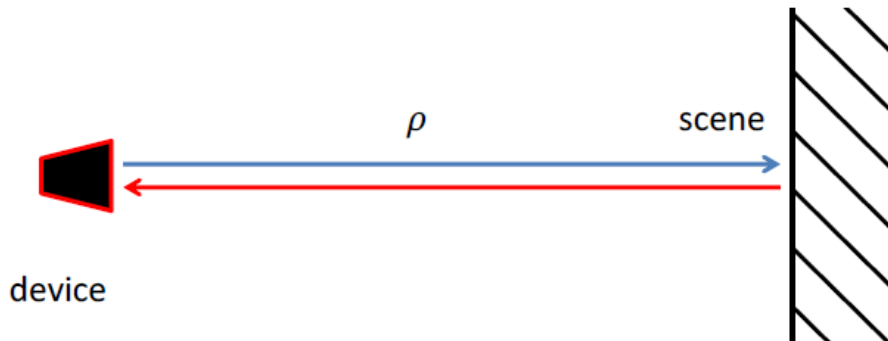


Figura 2.6 Principio de medidas ToF. Medida de tiempo

Básicamente, consiste en calcular la distancia que hay entre el dispositivo o sensor hasta los elementos de la escena emitiendo luz y midiendo el tiempo que tarda en volver, ya que se conoce la velocidad c a la que viaja la luz ($c = 3 \cdot 10^8$ m/s). De esta manera, calculando el tiempo τ que tarda en volver la luz desde que se emite, se puede calcular la distancia d o profundidad mediante la ecuación.

$$d = \frac{c\tau}{2} \quad (1)$$

Sin embargo, este método de medida de tiempo tiene un problema, y dada la velocidad de la señal electromagnética, se necesitaría un sistema que midiera el tiempo con muchísima precisión, porque de lo contrario, se produciría un error elevado en distancia. Por ejemplo, si hubiera un error de 1 ns en la medida temporal, sustituyendo en la ecuación (1), obtendríamos

un error de distancia de 3 m. Para asegurar esta precisión, han de diseñarse sistemas muy precisos en los que se pueda asegurar un error muy bajo, lo que implica un mayor coste económico. Por lo tanto, en las cámaras ToF se suele por medir la diferencia de fase $\Delta\varphi$ existente entre la señal infrarroja modulada emitida y la recibida, como se observa esquemáticamente en la Figura 7.

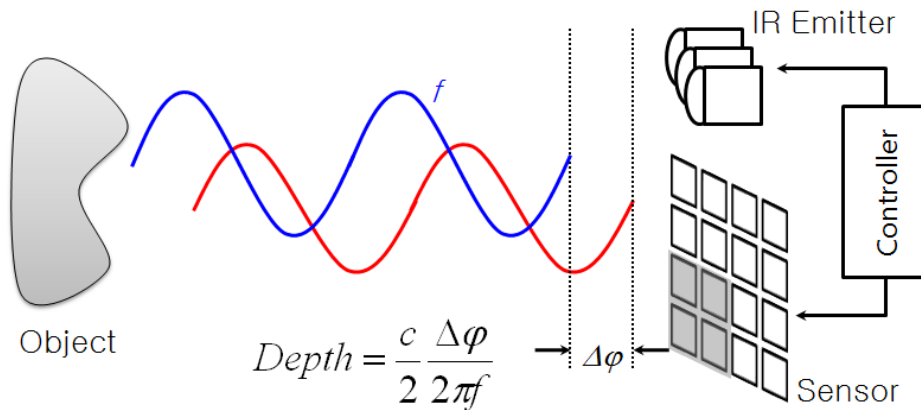


Figura 2.7 Principio de medidas ToF. Medida de desfase.

De esta manera, es más fácil y barato obtener una medida de distancia fiable. En la Figura 8 se observa una posible señal sinusoidal emitida (en azul), y la señal reflejada en la escena que se recibe (en rojo), observando la diferencia de fase entre ambas. Según este método, la profundidad o distancia d se calcula según la expresión (2), donde $\Delta\varphi$ es el desfase medido entre ambas señales, c la velocidad de la luz y f_{mod} la frecuencia de la señal modulada que se emite.

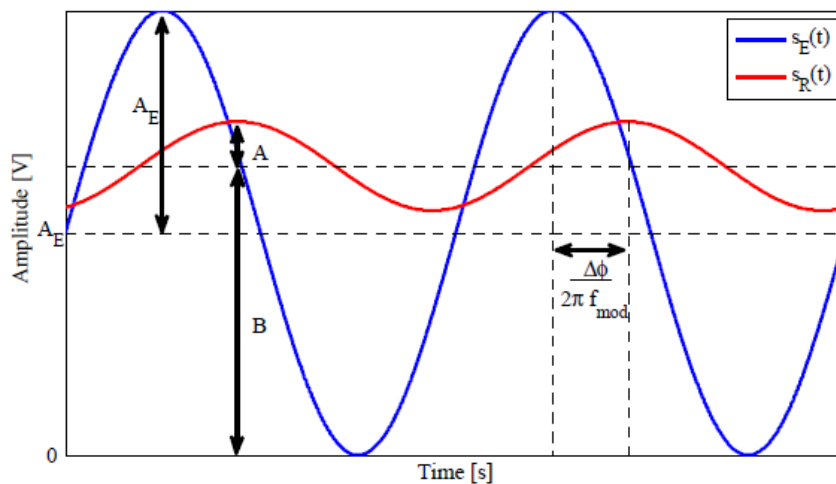


Figura 2.8 Señal emitida (azul) y señal recibida (rojo)

$$d = \frac{c}{2} * \frac{\Delta\varphi}{2\pi f_{mod}} \tag{2}$$

Por tanto, se ha de medir la diferencia de fase. Sin embargo, la fase no puede ser medida directamente, sino que se debe medir correlando la señal óptica recibida en el sensor con la señal de referencia eléctrica, que está en fase con la luz modulada emitida, proceso que se lleva

a cabo en la propia cámara ToF. Para medir la luz recibida, se utilizan unos elementos llamados 'smartpixels' basados en la tecnología CMOS.

Por otro lado, la pseudo-imagen en escala de grises que proporciona la cámara es obtenida a partir de la amplitud de la señal recibida, que correspondería al valor de la amplitud A de la señal $s_R(t)$ de la Figura 2.8, y que como va a depender de la c de la superficie en la que se refleje la señal emitida $s_E(t)$.

Es importante tener en cuenta que las cámaras de tiempo de vuelo proporcionan una medida de distancias, no coordenadas tridimensionales, las cuales se pueden obtener matemáticamente conociendo los parámetros intrínsecos de la cámara mediante un proceso de calibración.

2.5 Ruido en las imágenes de profundidad

El ruido digital o cuando se habla de imágenes digitales simplemente ruido, es la variación aleatoria de brillo y color por lo general no deseada y producida en el dispositivo de entrada. Las cámaras de tiempo de vuelo no están exentas este tipo de problemas a continuación se exponen algunos de los más importantes.

- **Interferencia por multicamino:** las reflexiones de las ondas infrarrojas en situaciones como esquinas o bordes pueden provocar que, además de la onda directa (con una sola reflexión), lleguen al sensor ondas tras dos o más reflexiones. Esto provoca que las medidas puedan ser erróneas ocasionando ruido, y una sobre estimación de la distancia, así como medidas sin sentido. En la Figura 2.9 puede apreciarse el efecto y errores que produce el multicamino. En la misma se muestran cuatro situaciones diferentes de las cuales tres sufrirán multicamino, la situación a) es la deseada en la cual no se produce ninguna clase de error por multicamino. Esto cambia en las situaciones b),c) y d) en las cuales podemos ver como ya sea por obstáculos o esquinas se producen reflexiones secundarias que alcanzan el sensor pudiendo producir en éste errores de medida. Dentro de estas situaciones destaca la d) que a día de hoy es la situación más conflictiva del multicamino cuyo error es similar al de la Figura 2.10, donde se produce una sobre estimación de la distancia, y un redondeo de la esquina.

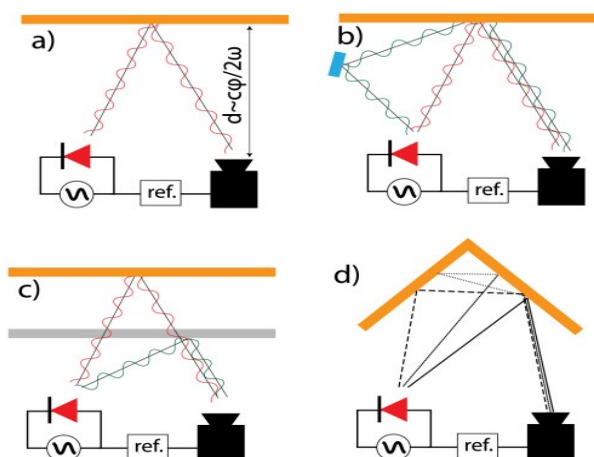


Figura 2.9 Ejemplos de error de multicamino

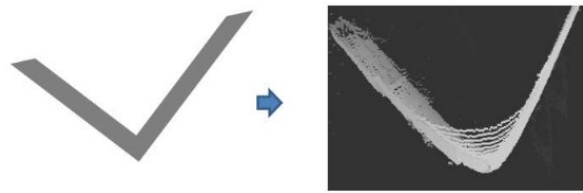


Figura 2.10 Efecto del multicamino en las esquinas

- **Artefactos de movimiento:** dado que estas cámaras requieren de un tiempo de integración para poder obtener las medidas de profundidad, si en el intervalo de dicho tiempo se producen grandes cambios en la escena como movimientos, se registrarán medidas erróneas o ruidosas tal y como puede verse en la Figura 2.11. En este caso se observa la transición del punto rojo en las tres imágenes, pudiendo ver que entre la primera y segunda imagen, la silla ha sufrido un movimiento durante el tiempo de integración. Esto resulta en unas medidas erróneas de profundidad como se puede ver en la tercera imagen en la cual estos píxeles erróneos se manifiestan como un suavizado alrededor del objeto con cambios de profundidad[8].



Figura 2.11 Ejemplo de artefactos de movimiento

- **Error por iluminación no uniforme:** en la gran mayoría de las cámaras de tiempo de vuelo se utilizan grandes y potentes módulos de iluminación infrarroja[9], el gran problema de estos módulos es que están excesivamente focalizados ofreciendo una iluminación muy centrada y no una iluminación difusa. Esto produce un doble problema debido a que por una parte puede producirse una saturación en la zona focalizada y una falta de iluminación especialmente en la zona de las esquinas. Este efecto puede observarse en la Figura 2.12 en la cual se enfoca una pared con una cámara de tiempo de vuelo y a medida que se está más cerca de las esquinas de la imagen se tiene una mayor falta de iluminación y en consecuencia un error mayor. Otro ejemplo de esta distorsión más cercano a la realidad podría ser el de la Figura 2.13.

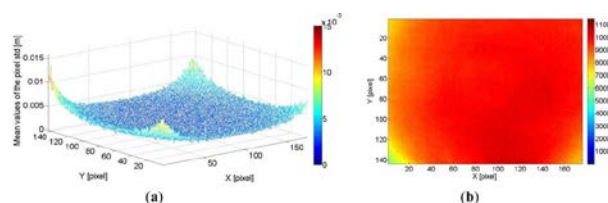


Figura 2.12 Ejemplo de distorsión por la dispersión de la luz en una imagen de profundidad de una pared

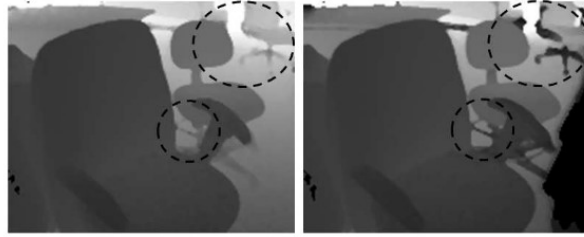


Figura 2.13 Ejemplo de distorsión por la dispersión de la luz en una imagen de profundidad

2.6 Filtrado de imágenes

Para suavizar los efectos del ruido que se obtiene al capturar una imagen se puede aplicar una serie de filtros. El filtrado [10],[11] es el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen original, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella.

Los filtros pueden dividirse en aquellos que trabajan en el dominio de la frecuencia y los que trabajan en el dominio del espacio. Los primeros trabajan con la transformada de Fourier de la imagen, el proceso general es:

- Se aplica la transformada de Fourier.
- Se multiplica por la función de filtro que se escoge.
- Se vuelve al dominio espacial aplicando la transformada inversa de Fourier.

Existen tres tipos de filtrado en el dominio de la frecuencia:

- Paso bajo: atenúa las frecuencias altas y mantiene sin variaciones las bajas. El resultado en el dominio espacial es equivalente al de un filtro de suavizado, donde las altas frecuencias que son filtradas se corresponden con los cambios fuertes de intensidad. Consigue reducir el ruido suavizando las transiciones existentes.
- Paso alto: hace lo contrario que el anterior, esto es, atenúa las frecuencias bajas manteniendo invariables las frecuencias altas. Puesto que las altas frecuencias corresponden en las imágenes a cambios bruscos de densidad, este tipo de filtros es usado, porque entre otras ventajas, ofrece mejoras en la detección de bordes en el dominio espacial, ya que estos contienen gran cantidad de dichas frecuencias. Refuerza los contrastes que se encuentran en la imagen.
- Paso banda: atenúa los extremos (frecuencias muy altas o muy bajas) manteniendo una banda de rango medio.

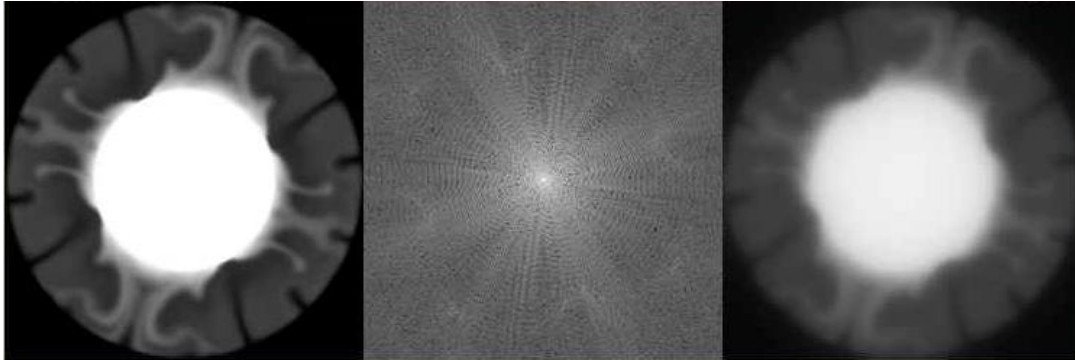


Figura 2.14 Filtrado de una imagen en el dominio de las frecuencias

En cuanto a los filtros en el dominio del espacio las operaciones de filtrado pasan a llevarse a cabo directamente sobre los píxeles de la imagen. En este proceso se relaciona, para todos y cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objetivo con la finalidad de obtener una información útil, dependiente del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado para obtener mejoras sobre la imagen y/o datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella. Este conjunto de píxeles próximos es la máscara de convolución, una matriz de píxeles como la que se usaba en el caso de los gradientes, de hecho algunos de estos gradientes (Roberts, Prewitt, Sobel, Frei-Chen y demás métodos de detección y realce de bordes).

Algunos de los procesos más empleados son:

- Filtro de la media: Dada una imagen $f(i,j)$, el procedimiento consiste en generar una nueva imagen $g(i,j)$ cuya intensidad para cada píxel se obtiene promediando los valores de intensidad de los píxeles $f(i,j)$ incluidos en un entorno de vecindad predefinido.
- Filtro gaussiano: El valor de cada punto es el resultado de promediar con distintos pesos los valores vecinos a ambos lados de dicho punto. Este tipo del filtro también tiene el problema del difuminado de los bordes, pero no es tan acusado como el caso de la media simple. Este tipo de filtro reduce especialmente el ruido tipo gaussiano. Además este filtro emborrona menos los bordes.
- Filtro de la mediana: Los píxeles de la nueva imagen se generan calculando la mediana del conjunto de píxeles del entorno de vecindad del píxel correspondiente a la imagen origen. De esta forma se homogeneizan los píxeles de intensidad muy diferente con respecto a la de los vecinos. Este tipo de filtro es bastante indicado cuando se tiene ruido aleatorio.

2.7 Descriptores

La extracción de características es el primer paso para la clasificación de objetos y la construcción de los algoritmos de reconocimiento de patrones. A continuación se exponen dos descriptores que han demostrado dar muy buenos resultados

2.7.1 Patrón binario local

El patrón binario local(Local Binary Pattern)[12] es un operador de textura simple y eficiente que etiqueta cada píxel de la imagen analizando su vecindario, estudiando si el nivel de gris de cada píxel supera un determinado umbral y codificando dicha comparación mediante un número binario. Debido a su bajo coste computacional y al gran poder discriminativo que ha demostrado tener, LBP se ha convertido en los últimos años en una de las soluciones más utilizadas en numerosas aplicaciones relacionadas con textura. La característica más importante de LBP en aplicaciones reales es la robustez que ofrece frente a variaciones en la intensidad del nivel de gris, causado, entre otras muchas cosas, por diferencias en la iluminación. Para calcular LBP sobre una imagen en escala de grises se utiliza la siguiente ecuación:

$$LBP_{P,R} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p, \quad s(x) = f(x) = \begin{cases} 1 & \text{si } x < 0 \\ 0 & \text{si } x \geq 0 \end{cases} \quad (3)$$

Donde P es el número de vecinos que se van a considerar, R es el tamaño del vecindario y, g_c y g_p son los valores de gris del píxel central y cada uno de los p píxeles del vecindario respectivamente.

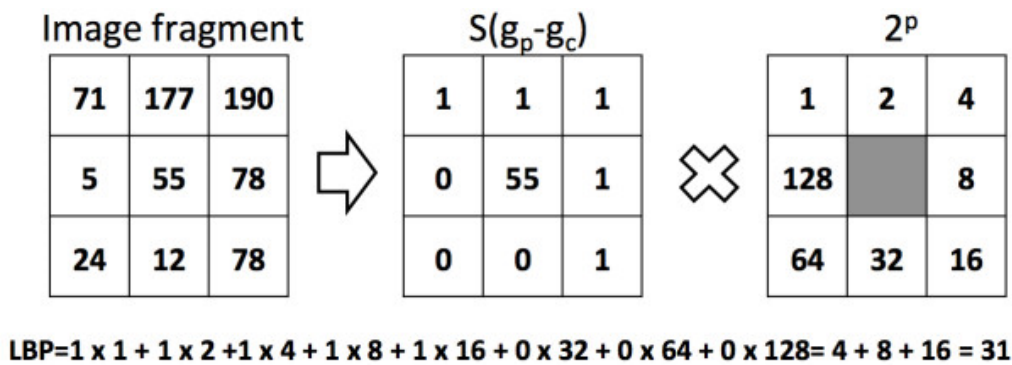


Figura 2.15 Ejemplo aplicación LBP

El operador LBP ha ido desarrollando a lo largo de los años, utilizándolo con diferentes tamaños, no solo con ocho puntos, sino a muestreos circulares y, la bilinealidad se consigue con la interpolación de los valores de los píxeles, lo que permite utilizar cualquier radio y por lo tanto cualquier número de píxeles vecinos. Para los entornos circulares se usa la notación (P, R), donde P indica el número de puntos de muestreo y R el radio del círculo, en la Figura 2.16 se pueden observar ejemplos para diferentes tamaños.

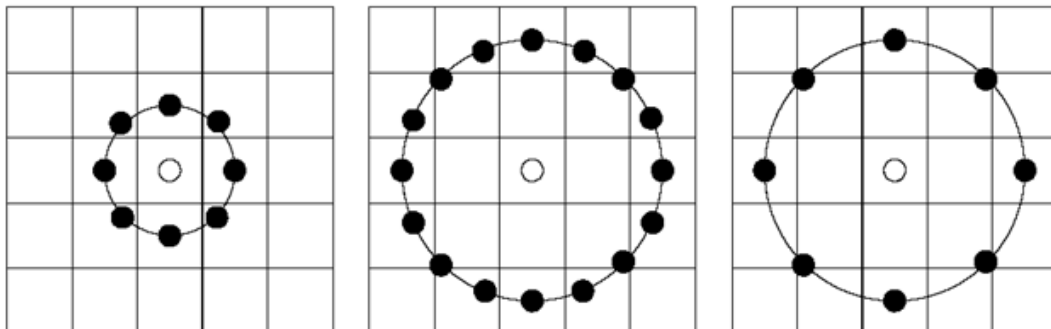


Figura 2.16 Simetría circular para conjuntos de vecinos

2.7.2 Histograma de gradientes orientados

El Histograma de gradientes orientados (Histogram of oriented gradients) fue presentado por Navneet Dalal y Bill Triggs [13].

Este descriptor es una transformación del espacio 2D (imagen) al espacio 1D (histograma), al dividir la imagen en varios bloques para calcular el histograma de su intensidad basados en las orientaciones del gradiente

Los pasos a seguir son los siguientes:

1. Calcular el gradiente de las imágenes, utilizando la ecuación (4) y (5) sobre la imagen I:

$$G_x(x, y) = I(x + 1, y) - I(x - 1, y) \quad (4)$$

$$G_y(x, y) = I(x, y + 1) - I(x, y - 1) \quad (5)$$

Donde (x, y) es la intensidad en el pixel (x, y) .

2. Obtener la norma M_G y la orientación del gradiente θ_G utilizando (6) y (7)

$$M_G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \quad (6)$$

$$\theta_G(x, y) = \tan^{-1} \left(\frac{G_x(x, y)}{G_y(x, y)} \right) \quad (7)$$

Calcular el histograma mediante la discretización de θ_G en N_b intervalos sobre cada celda de tamaño $m \times m$ píxeles. La discretización de θ_G se realiza en el intervalo $[\pi, -\pi]$. El tamaño de m dependerá del número de características que sean necesarias cuando menor sea el valor más valores se podrán obtener de una imagen.

3. Último paso se normaliza el vector de características para hacerlos más robusto antes cambios de iluminación

$$v = \frac{v}{\sqrt{\epsilon + \|v\|_2^2}} \quad (8)$$

Donde $\epsilon=0,001$ sirve para evitar divisiones por cero

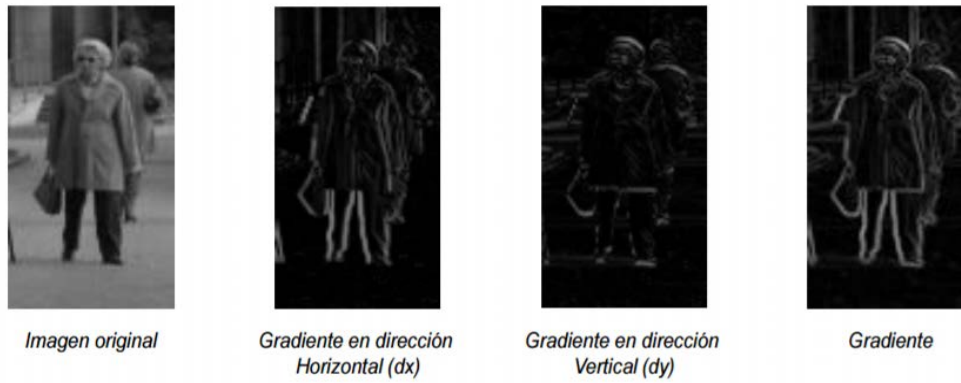


Figura 2.17 Gradiente en dirección horizontal y vertical de una imagen

2.7.3 Filtros Haar

La extracción de características es realizada aplicando a la imagen filtros con bases Haar [14]. Estos filtros pueden ser calculados eficientemente sobre la imagen, son selectivos en la orientación espacial y frecuencia, y permiten ser modificados en escala y orientación. En la Figura 2.18 se muestran algunos de los filtros usados para la extracción de características. Los filtros con bases Haar, realizan una codificación de diferencia de intensidades en la imagen, generando características de contornos, puntos y líneas, mediante la captura de contraste entre regiones.

Filtros de Haar básicos

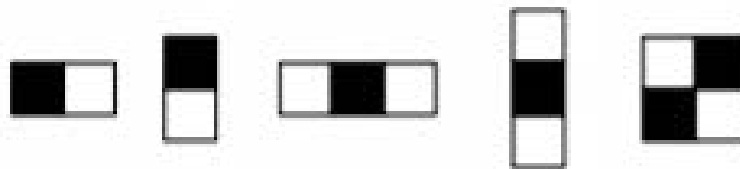


Figura 2.18 Filtros Haar

Para computar rápidamente cada uno de los rectángulos se usa una representación de la imagen llamada "Integral de la imagen" (Integral Image). La integral de una imagen respecto un punto x,y consiste en la suma de los píxeles por arriba y a la izquierda de dicho puntos, x,y incluidos.

$$ii(x, y) = \sum_{x' < x, y' < y} i(x', y') \quad (9)$$

La integral de la imagen representa en computación una manera elaborada de obtener valores de forma eficiente

2.8 Clasificadores

Clasificar un objeto consiste en asignarlo a una de las clases disponibles. Los objetos se pueden definir por una serie de características, como pueden ser el color de sus píxeles, su textura, su tamaño o su forma.

Un clasificador es aquel algoritmo que mediante el análisis de un conjunto de características, previamente extraídas, permite distinguir si la zona u objeto analizado pertenecen a una clase o conjunto de clases. Existe un amplio abanico de clasificadores de diferentes tipos, precisiones y entrenamientos, dentro de los más conocidos podemos encontrar las máquinas de soporte vectorial (SVM), *k-nearest neighbors (KNN)*, los Random Forest o las Redes neuronales. Los clasificadores se pueden dividir en dos grupos en función de si pueden discriminar entre múltiples clases (clasificadores multiclase), o únicamente entre dos (clasificadores binarios). Por otro lado, también pueden clasificarse en función de si se basan en aprendizaje supervisado o no supervisado. El aprendizaje supervisado es una forma de aprendizaje artificial que se basa en la creación de una hipótesis mediante unos conjuntos (o sets) de datos de entrenamiento con etiquetado manual, creado específicamente para ello. El clasificador emplea estos datos de entrenamiento para inferir con respecto a unos datos de entrada cual es la clase a la cual pertenecen. Por otro lado los algoritmos con un aprendizaje no supervisado carecen de un conjunto de entrenamiento, por lo cual recurre a las llamadas técnicas de agrupamiento que tratan de crear estas etiquetas y catalogar en conjuntos cuyos datos tengan un gran parecido entre ellos y se diferencien en gran medida de los demás grupos.

En este proyecto se van a usar clasificadores con aprendizaje supervisado, a continuación vamos a ver alguno tipos.

2.8.1 Tipos de clasificadores

- La máquina de vectores de soporte, o máquina de soporte vectorial (SVM) es un tipo de clasificador que fue desarrollado por Vladimir Vapnik y su equipo en los laboratorios AT [15]. Se trata de un algoritmo de aprendizaje supervisado, por lo que requiere de un entrenamiento previo, así como de la creación de las clases necesarias.

Originalmente, las SVM se diseñaron para ser empleadas como un clasificador binario, por lo que a continuación se presenta el fundamento matemático para este caso. Sin embargo, existen alternativas para emplear las SVM para clasificación multiclase mediante aproximaciones uno-contra-todos (*one-against-all*), uno-contra-uno (*one-against-one*) o DAGSVM.

El objetivo de un clasificador basado en SVM es la búsqueda de un hiperplano que separe de forma óptima a las muestras pertenecientes a una clase de las que corresponden a otras. Dichas muestras pueden haber sido previamente proyectadas a un espacio de dimensionalidad superior.

El plano óptimo buscado es aquel que ofrece el mayor margen de distancia entre el mismo y las muestras más próximas de cada una de las clases, lo que se trata de

conseguir es definir una frontera de separación entre las clases definidas, de forma que al proyectar el nuevo candidato se pueda decidir a qué clase pertenece sabiendo en qué lado del hiperplano reside. La Figura 2.19 muestra un ejemplo de la frontera de separación para la que se obtiene el máximo margen en un caso simple en dos dimensiones.

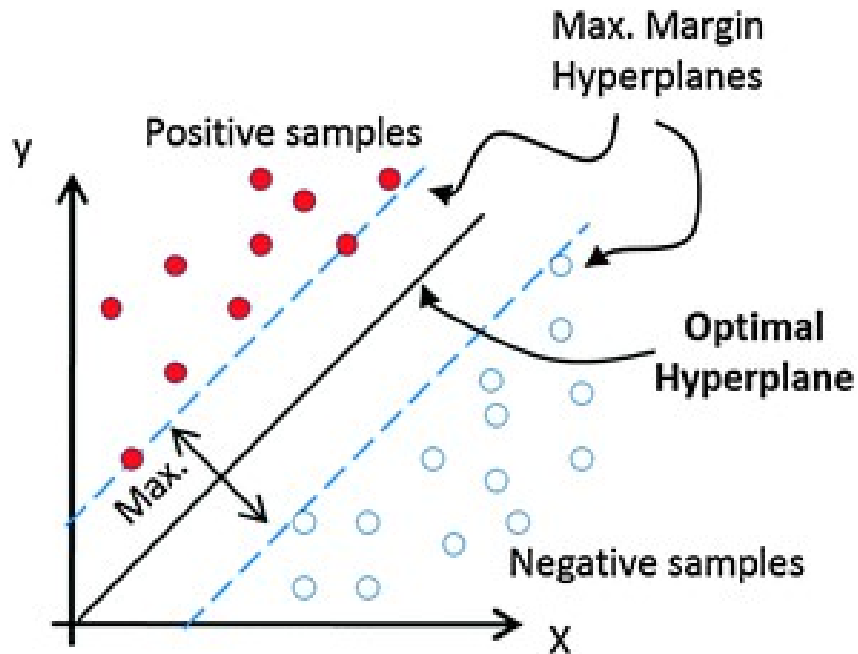


Figura 2.19 Ejemplo de SVM lineal y su plano de separación óptimo en dos dimensiones.

- El algoritmo KNN o K-NN (*k-nearest neighbors*, literalmente los k vecinos más cercanos, Figura 2.20) es un método no paramétrico usado para la clasificación y la regresión. En ambos casos la principal entrada del método es el número entero positivo k que hace referencia a los k elementos más cercanos a aquel que se estudia. Este elemento pertenecerá a uno u otro grupo según lo sean la mayoría de estos elementos más cercanos. El elemento k será por lo general pequeño, la experiencia dicta que no llegará por lo general a 10. Este método es posiblemente el más simple de los algoritmos de clasificación.

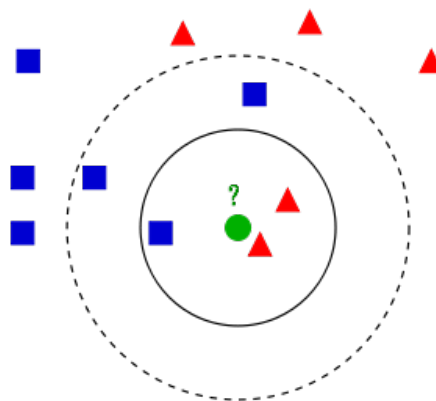


Figura 2.20 En la clasificación KNN el elemento central pertenecerá al grupo rojo si $k=3$ y al azul si $k=5$.

Este método depende de la métrica, es decir, si se mide una distancia euclídea, de Hamming, Mahalanobis, etc. Otro parámetro que define este método es la normalización, esto es, en varias ocasiones se darán medidas que serán muy grandes frente a otras y en las que cambios relativamente grandes repercutirán poco o nada en la solución, como puede ser el caso de estudiar el precio del suelo en una moneda de poco valor relativo en función del área o en el caso que ocupa este TFG la cantidad de píxeles de las regiones seleccionadas en la imagen (que puede ser del orden de millones) frente al valor RGB (entre 0 y 255) y que por tanto deberán ser por lo general debidamente normalizadas para que atributos con mucho rango no tengan más peso que los demás, centrando y escalando cada columna según su media y desviación típica.

- El Adaboost fue presentado en 1995 por Y. Freund y Schapire [16] y debe su nombre a que es un tipo de Boosting que se ajusta adaptativamente a los errores de las reglas de clasificación débiles que genera el WeakLearn (algoritmo genérico de aprendizaje débil) evaluadas sobre la distribución de muestras o ejemplos positivos (imágenes ejemplo del objeto). El método de Boosting accede al algoritmo de aprendizaje débil, WeakLearn (el nombrado anteriormente cuyo objetivo es generar reglas de decisión de baja fiabilidad) mediante numerosas llamadas, en cada una de las cuales se evalúa una regla débil en concreto sobre todas las imágenes del conjunto muestra y por tanto, se las clasifica en función de las etiquetas obtenidas para esa regla débil, así que están distribuidas de determinada forma porque unas habrán resultado más complicadas que otras. El objetivo del WeakLearn es encontrar la hipótesis que consigan minimizar el error entre la distribución de las muestras que se le proporcionó.

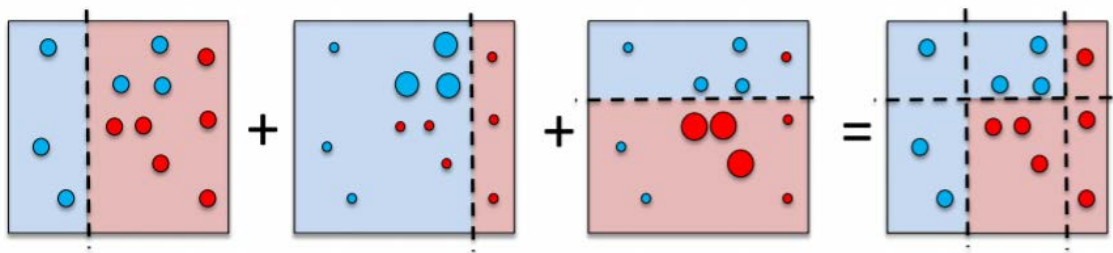


Figura 2.21 Suma clasificadores débiles para obtener un clasificador fuerte

El algoritmo se quedará por tanto, en cada vuelta del boosting, con la hipótesis que mejor clasifique la distribución de muestras, la que menor error presente. Además, la distribución muestral en la que se basa el algoritmo de WeakLearn para calcular el error, se va recalculando a cada vuelta, para ir dándole mayor peso a aquellas muestras mal clasificadas por la hipótesis de la vuelta anterior. Este mismo proceso continuará durante un total de T llamadas al WeakLearn.

2.8.2 Precisión del clasificador

Una vez realizados los cálculos el objetivo será seleccionar el mejor clasificador, esto es, el que tenga una mayor tasa de acierto. Existen distintas maneras de obtener la precisión de un clasificador, entre las que destacan [17]:

- Validación cruzada: La validación cruzada es una técnica de validación de modelos para evaluar cómo los resultados de un análisis estadístico generalizarán a un conjunto de datos independientes. Se utiliza principalmente en entornos donde la meta es la predicción, y se quiere estimar la precisión con que un modelo funcionará en la práctica. En un problema de predicción, al modelo generalmente se le administra un conjunto de datos conocidos a partir de los cuales se crea el clasificador (datos de entrenamiento), y un conjunto de datos desconocidos (o datos vistos por primera vez) contra el que se prueba el modelo de prueba (conjuntos de datos). El objetivo de la validación cruzada es definir un conjunto de datos para "probar" el modelo en la fase de entrenamiento, con el fin de limitar los problemas como sobreajuste (*overfitting*, figura 42), dando una idea de cómo el modelo se generalizará a un conjunto de datos independiente es decir, un conjunto de datos desconocido, un problema real, etc. Una ronda de validación cruzada implica la partición de una muestra de datos en subconjuntos complementarios, realizar el análisis de un subconjunto (conjunto de entrenamiento), y validar el análisis en el otro subconjunto (llamado el conjunto de validación o conjunto de pruebas). Para reducir la variabilidad, se realizan múltiples rondas de validación cruzada utilizando diferentes particiones, y los resultados de la validación se promedian para todas las rondas. Una de las principales razones para el uso de validación cruzada en lugar de utilizar la validación convencional (por ejemplo partir el conjunto de datos en dos conjuntos de 70% para la formación y el 30% para la prueba) es que no hay suficientes datos disponibles para dividir en un conjunto de entrenamiento y uno de pruebas sin perder capacidad. En estos casos, una buena manera para estimar adecuadamente el rendimiento predicción del modelo es el uso de validación cruzada. En resumen, la validación cruzada combina (promedia) medidas de ajuste para obtener una estimación más precisa del rendimiento del modelo.
- Existen distintos tipos de validación cruzada que ofrecen distintos resultados. Los principales son:
 - K-fold cross-validation: En la validación cruzada de K iteraciones o *K-fold cross-validation* (Figura 2.22) los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto (K-1) como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional. En la práctica, la elección del número de iteraciones

depende de la medida del conjunto de datos. Lo más común es utilizar la validación cruzada de 10 iteraciones (10-fold cross-validation).

- Validación cruzada aleatoria: Este método consiste al dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final se corresponde a la media aritmética de los valores obtenidos para las diferentes divisiones. La ventaja de este método es que la división de datos entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar.
- Leave one out: La validación cruzada dejando uno fuera o Leave-one-out cross-validation (LOOCV) implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.
- Leave one out: La validación cruzada dejando uno fuera o Leave-one-out cross-validation (LOOCV) implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

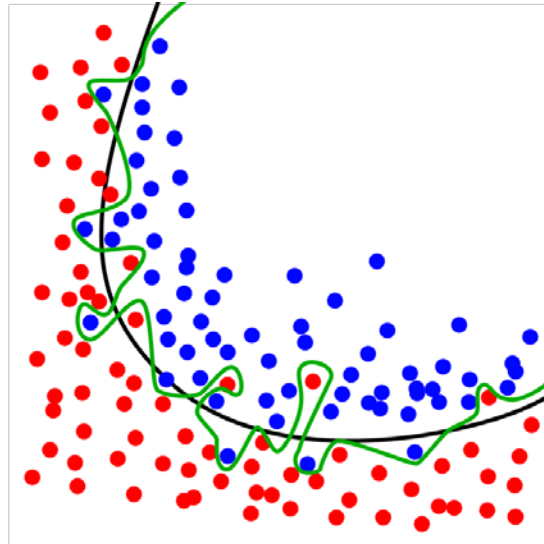


Figura 2.22 Overfitting; aunque la curva verde ofrecería mejores resultados para el conjunto de datos actual la curva negra ofrecerá un rendimiento mejor para datos nuevos

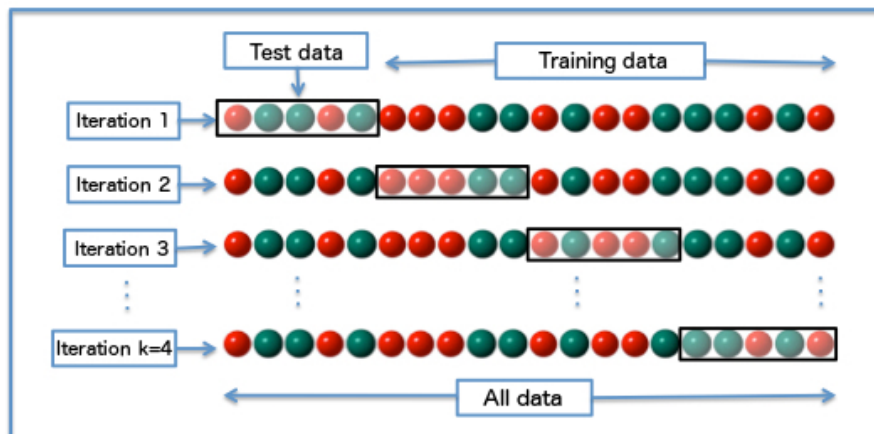


Figura 2.23 Ejemplo de validación cruzada tipo k-fold para $k=4$

- Matriz de confusión: También se conoce como matriz de errores. En el campo de la clasificación estadística una matriz de confusión es una matriz o tabla que permite la visualización de la ejecución de un algoritmo (Figura 2.24), típicamente un aprendizaje supervisado. Cada columna de la matriz representa las clases predichas mientras que cada fila representa las clases reales (o viceversa). El nombre se deriva del hecho de que hace que sea fácil ver si el sistema está confundiendo dos clases.

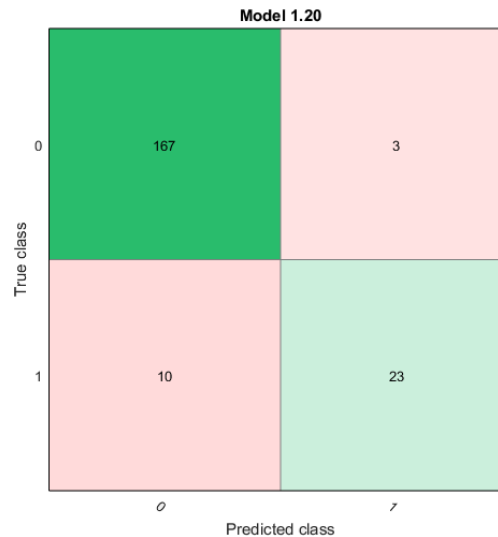


Figura 2.24: Matriz de confusión obtenida en Matlab para las regiones de estudio de este TFG.

- Curva ROC: ROC es el acrónimo de *Receiver Operating Characteristic*, literalmente Característica Operativa del Receptor. Son una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación, como se ve en la Figura 2.25. Otra interpretación de este gráfico es la representación del ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente al ratio de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo).

Los valores que se cruzan son también conocidos como:

- La sensibilidad, que indica la capacidad del clasificador para dar como casos positivos los casos que realmente lo son o proporción de positivos correctamente identificados.
- La especificidad, que indica la capacidad del clasificador para dar como casos negativos los casos que lo sean, es decir proporción de negativos correctamente identificados. Es decir, la especificidad caracteriza la capacidad de la prueba para detectar la ausencia de datos que indiquen que se trata de un caso positivo.

Se calculan como sigue:

$$\text{Sensibilidad} = \frac{VP}{VP + FN}; VP = \text{Falsos Negativos} \tag{9}$$

$$\text{Especificidad} = \frac{VN}{VN + FP}; VN = \text{Falsos Positivos} \tag{10}$$

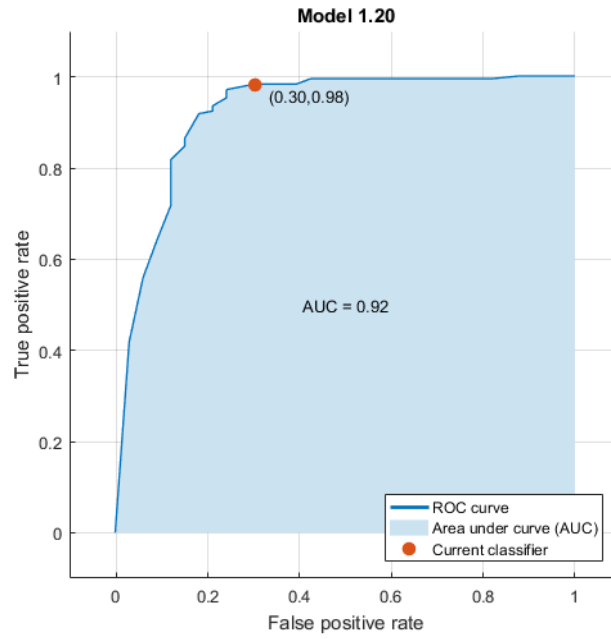


Figura 2.25: Curva ROC para un clasificador ficticio, con una sensibilidad óptima del 98 % y un valor 1-especificidad de 0,30.

CAPÍTULO 3

DESCRIPCIÓN Y CALIBRACIÓN DEL SENSOR ToF

3.1 Introducción

Una vez planteadas las bases del sistema se procede a centrarse en los elementos que se van a utilizar para llevarlo a cabo, que en este caso es básicamente el elemento de hardware (la cámara de tiempo de vuelo). En este capítulo se procederá a la descripción del sensor ToF y a la descripción del procedimiento necesario para la calibración de la cámara.

3.2 Selección de la cámara ToF

Este tipo de cámaras son dispositivos con un elevado precio debido a que es tecnología relativamente nueva, en los últimos años se está produciendo un aumento del número de dispositivos en el mercado, así como una disminución del precio. Las cámaras ToF puede competir con la tecnología LIDAR 3D en un futuro.

A continuación se exponen algunos sensores

El nuevo sensor sacado por Microsoft para su consola Xbox One, la cámara Kinect v2 [18], Figura 3.1., la cual incorpora una cámara de color de alta resolución, un sensor de profundidad de tiempo de vuelo y un sensor infrarrojo. Es una cámara bastante interesante para temas de investigación debido a su bajo coste aunque esta algo limitada.

Kinect for Windows v2 Sensor

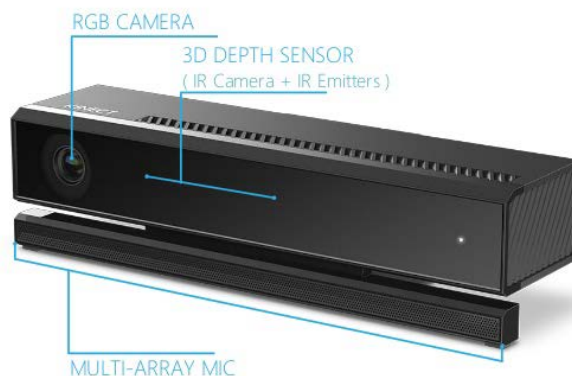


Figura 3.1 Kinect V2

- La cámara SR4000 es una cámara de tiempo de vuelo del fabricante MESA [19], que incluyen características de altas prestaciones en supresión de iluminación de fondo y mejora de rango dinámico. Es una cámara de grado industrial, la cual proporciona medidas de alta calidad en entornos donde se requiere un alto rendimiento. Utilizando un diseño óptico innovador, la cámara se autocalibra continuamente durante su funcionamiento. Además están disponibles modelos con campo de visión (FOV) de 70x55°, que permiten cubrir áreas más grandes o reducir la distancia de trabajo entre la cámara y la escena. La características más

interesante de este modelo es su capacidad de autocalibración lo que facilita mucho su uso en diferentes condiciones lumínicas.



Figura 3.2 SR4000

- La empresa Bluetechnix proporciona la cámara Sentis3D-M420 [20], esta cámara de tiempo de vuelo es capaz de dar información de profundidad y escala de grises de cada pixel, por lo que es perfecta para analizar escenas basadas en datos 3D, usa iluminación IR, con un rango de visión de hasta 4 metros en exteriores y 7 metros en interiores y con un campo de visión de hasta 110 grados.



Figura 3.3 Sentis3D-M420

El grupo de investigación División de Sistemas e Ingeniería Eléctrica de la Universidad Politécnica de Cartagena cuenta con dos cámaras Sentis3D-M420 adquiridas mediante el proyecto TIN2012-39279, las cuales han sido elegidas para realizar este proyecto.

3.3 Información técnica Sentis 3D-M420

Es una cámara de tiempo de vuelo modular, es decir se basa en un sistema de bloques de construcción que fue desarrollado para satisfacer una amplia variedad de demandas y aplicaciones.

El módulo elegido ha sido el TIM-UP-19k-S3 USB 2.0 cuyas principales características son:

- Tamaño: 85 x 125 x 205 mm
- Conector del módulo 6x 20-Pin LIM
- Interfaz RS485
- Conexión USB Micro
- 1 entrada (con aislamiento galvánico, con voltaje de entrada de hasta 50V)
- 1 salida (con aislamiento galvánico)
- Activación / desactivación del disparador
- Conector de 10/100 Mbit
- Configuración DIP Switch
- Botón de reset
- Voltaje necesario 18V-30V
- Puede funcionar un rango de temperaturas -20 to 55 °



Figura 3.4 Partes Sentis3D-M420

El rango de la cámara

- 7 metros en interior
- 4 metros en exterior

Información óptica:

- 24 LEDs de alto poder de 850nm
- Tecnología de detección de profundidad: Time of Flight (ToF)
- Puede capturar hasta 160 imágenes por segundo
- Angulo de visión de hasta 110°
- Tiene una resolución de 120x160 píxeles
- Sensor: PMD PhotonICs® 19k-S3

Para realizar este proyecto se ha elegido la lente de 110° que tiene el mayor ángulo de visión dado por la cámara ToF, para captar la mayor información posible en cada escena.

3.4 Parámetros: Tiempo de integración y frecuencia de modulación

La distancia que calculan las cámaras de tiempo de vuelo es fuertemente dependiente del tiempo de integración que se configura en el sensor y de la frecuencia de modulación empleada por el sistema de iluminación que integran. A continuación se exponen los diferentes parámetros que se pueden ajustar en el sensor[21]:

- Tiempo de integración (τ): define el tiempo que requiere el sensor de la cámara para recibir la señal de luz reflejada por la superficie de los objetos que se han establecido como objetos en la escena. Esto significa que si el tiempo de integración es pequeño, la distancia a los objetos muy alejados no podrá ser medida correctamente, puesto que la señal de luz no habrá tenido tiempo de retornar y excitar el sensor de la cámara. Además, por otro lado, si el tiempo de integración es grande, la cámara reduce su velocidad de adquisición y los mapas de profundidades pueden reflejar valores de distancia incorrectos, generalmente sobresaturados.

Frecuencia de modulación (f_{mod}): La variación de este parámetro cambia la apariencia física de la forma coseno que modela matemáticamente esa señal de luz (Figura 3.5).

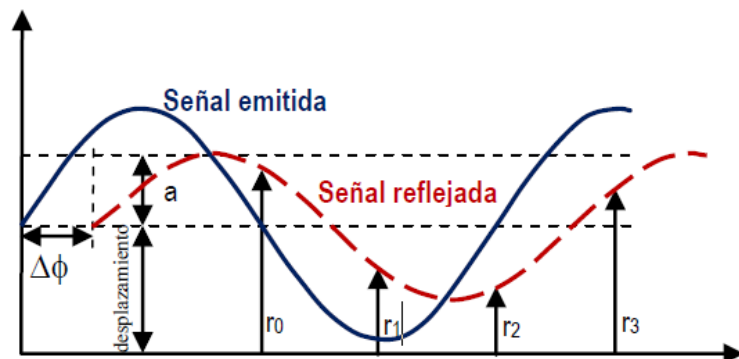


Figura 3.5 Señal emitida y señal reflejada

Este parámetro determina la máxima distancia al objetivo que la cámara es capaz de estimar. Este valor se conoce con el sobrenombre de 'rango de no ambigüedad'. La cámara usada en este proyecto tiene dos frecuencias de modulación 20MHz y 10MHz [22], ambas frecuencia tienen resultados muy parecidos, por lo que se usará la frecuencia de modulación preestablecida por el fabricante 20MHz.

- Offset: simplemente se usa para sumar cierta cantidad a la medida para así conseguir mayor exactitud, es una compensación de la media.

Hay otros parámetros que podemos cambiar como el número de imágenes por segundo, pero este valor, no será tan importante a la hora de conseguir un dato robusto de distancia, solo interesará un mayor número de imágenes por segundo en el caso de que tengamos videos con objetos en movimiento.

3.5 Pruebas de calibración

Por un lado, la principal ventaja de una cámara ToF es la extracción de profundidad para determinar características geométricas y planos de la escena de trabajo, además una correcta calibración de ésta, permite transformar la imagen de profundidad en un conjunto de puntos 3D del entorno, proporcionando así información espacial tridimensional. Por otro lado, el principal inconveniente son las fluctuaciones en la precisión con la que se mide la distancia al objetivo, principalmente debido a las interferencias causadas por factores externos, como por ejemplo la luz solar, la orientación, la reflectividad de la superficie de los objetos de la escena y la distancia de trabajo entre cámara y objetivo.

Para conseguir que estas interferencias afecten a la información de profundidad lo menos posible, se han cogido los grupos de píxeles que pertenecen al patrón y se ha realizado la media de 30 imágenes para obtener el valor de profundidad.



Figura 3.6 Calibración cámara ToF

3.5.1 Pruebas con un campo de visión de 90º

La cámara viene de fábrica con calibrada para un campo de visión de 90º. En primer lugar se realiza una prueba con esta configuración, cuyos parámetros son:

Tiempo de integración	500 μ s
Fotogramas por segundo	5
Offset	-2837
Frecuencia de modulación	20MHz

Tabla 3.1 Parámetros por defecto

La prueba consiste en realizar una medida de profundidad a la distancia de 50cm, 100cm, 150cm y 200cm, con los parámetros por defecto.

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(μ s)	Frecuencia de Modulación(MHz)	Error (%)
500	506,7	500	20	1,34
1000	1013,7	500	20	1,37
1500	1412,1	500	20	5,86
2000	1806	500	20	9,70

Tabla 3.2 Lente 90º-Prueba con los parámetros por defecto

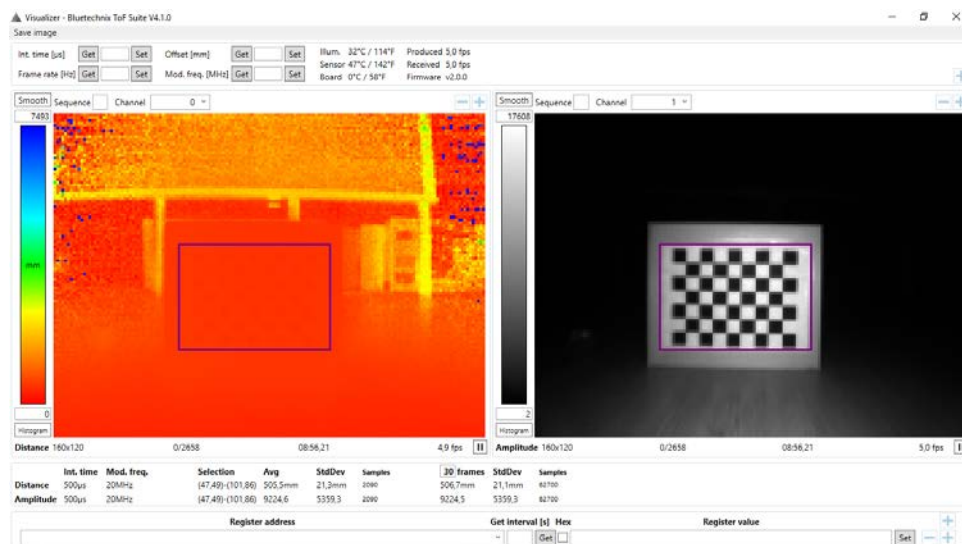


Figura 3.7 Lente 90º- Prueba 50cm

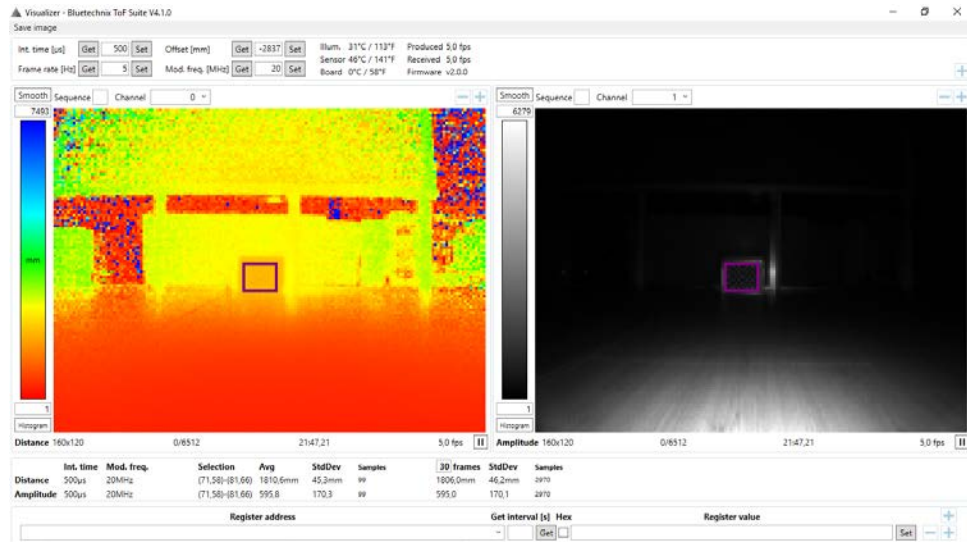


Figura 3.8 Lente 90°- Prueba de distancia 200cm

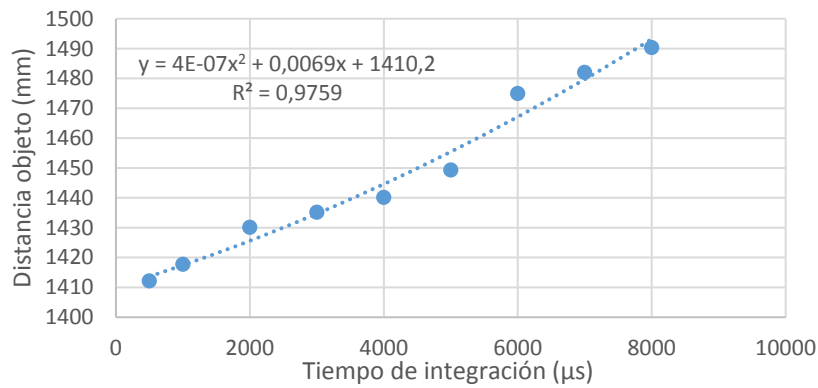
En la Tabla 3.2 se puede observar que hasta un metro el error es de un 1,37%, mientras que partir esa medida aumenta considerable hasta casi un 10% en la prueba realizada a 200cm. En la Figura 3.7 y en la Figura 3.8 se muestran la imagen de distancia (izquierda) y la de amplitud (derecha) obtenida por el sensor.

La siguiente prueba consiste en aumentar el tiempo de integración para una misma distancia (150cm) hasta obtener un resultado con un error menor al 1%.

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(µs)	Frecuencia de Modulación(MHz)	Error (%)
1500	1412,1	500	20	5,86
1500	1417,7	1000	20	5,49
1500	1430,1	2000	20	4,66
1500	1435,1	3000	20	4,33
1500	1440,1	4000	20	3,99
1500	1449,3	5000	20	3,38
1500	1474,9	6000	20	1,67
1500	1482	7000	20	1,20
1500	1490,3	8000	20	0,65

Tabla 3.3 Lente 90°- Prueba aumento tiempos de calibración

Si se muestran los datos en la Grafica 3.1:



Grafica 3.1 Lente 90º-Aumento de tiempo de integración a 150cm

Imágenes tomadas por el sensor con diferentes tiempos de integración:

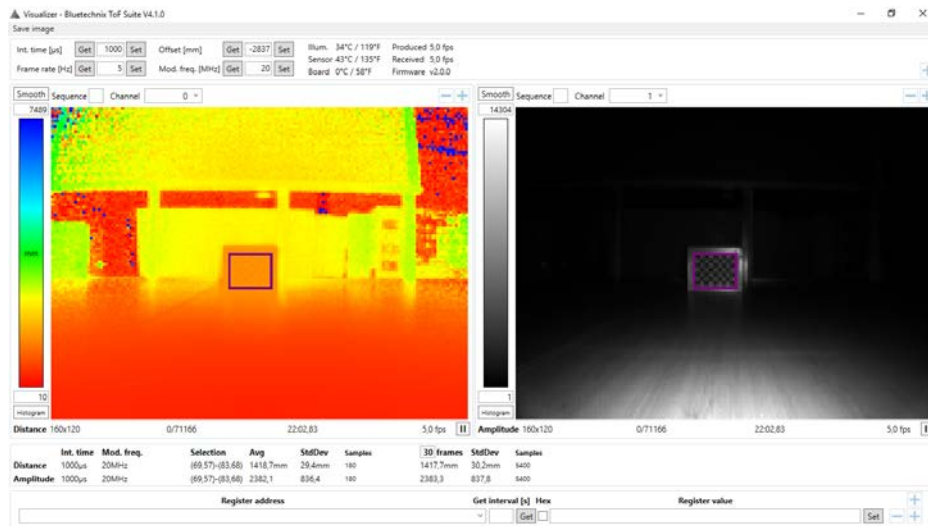


Figura 3.9 Lente 90º-Tiempo de integración 1000µs



Figura 3.10 Lente 90º-Tiempo de integración 8000µs

De los datos anteriores se observa que para llegar a un error menor del 1% se ha tenido que aumentar el tiempo de integración hasta 8000 μ s. Entre la Figura 3.9 y la Figura 3.10 se puede ver una gran diferencia, debido a que en la Figura 3.10 la imagen de amplitud está muy saturada lo que provoca un error en el cálculo de la medida, y en la imagen de distancia se observa menos ruido en comparación con la primera imagen.

En una nueva prueba se ha colocado el patrón a una distancia de 200cm y se ha aumentado el tiempo de integración hasta conseguir que el error en la media sea pequeño, teniendo que aumentarlo hasta 60000 μ s.

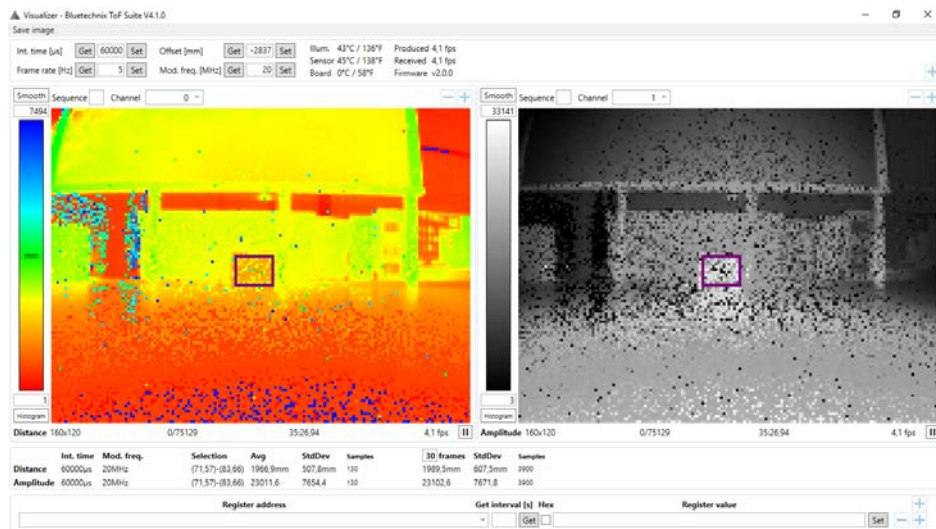


Figura 3.11 Lente 90 $^{\circ}$ -Tiempo de integración 60000 μ s

Como se observa en la Figura 3.11, la imagen de amplitud está muy saturada. Y la imagen de distancia con mucho, además se observó en la cámara un aumento de temperatura bastante considerable de más de 10 $^{\circ}$, el cual puede hacer variar la medida.

En conclusión se puede decir que para una buena calibración hay que conseguir un equilibrio entre la imagen de distancia y la imagen de amplitud, es decir, una imagen de distancia que no contenga demasiado ruido y una imagen de amplitud que no esté saturada.

Otro de los errores cometidos en esta primera prueba es colocar la cámara cerca del suelo, ya que esto provoca que los objetos (con una alta reflectividad) que se encuentren cerca de la cámara produzcan un efecto de luz dispersa, produciendo un error en la medida y que el fondo aparezca más cerca de la cámara.

En las siguientes pruebas realizadas se ha tenido en cuenta ese efecto, por lo que se ha colocado la cámara en alto para evitarlo.

3.5.2 Pruebas con un campo de visión de 110º

El campo de visión de la cámara ToF que se va a usar en este proyecto es de 110º, es decir, tiene una apertura 20º mayor que la que la lente de fábrica, con ella se podrá obtener más información en cada imagen pero también se obtendrá más ruido, por lo que una buena calibración es muy importante, teniendo en cuenta las pruebas realizadas en el estudio de la lente de 90º se va a proceder a la calibración de esta lente.

Los resultados de la prueba con los parámetros que vienen por defecto son:

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(µs)	Frecuencia de Modulación(MHz)	Error (%)
500	512,3	500	20	2,46
1000	996,7	500	20	0,33
1500	1402,4	500	20	6,51
2000	1748,3	500	20	12,59

Tabla 3.4 Lente 110º con parámetros por defecto

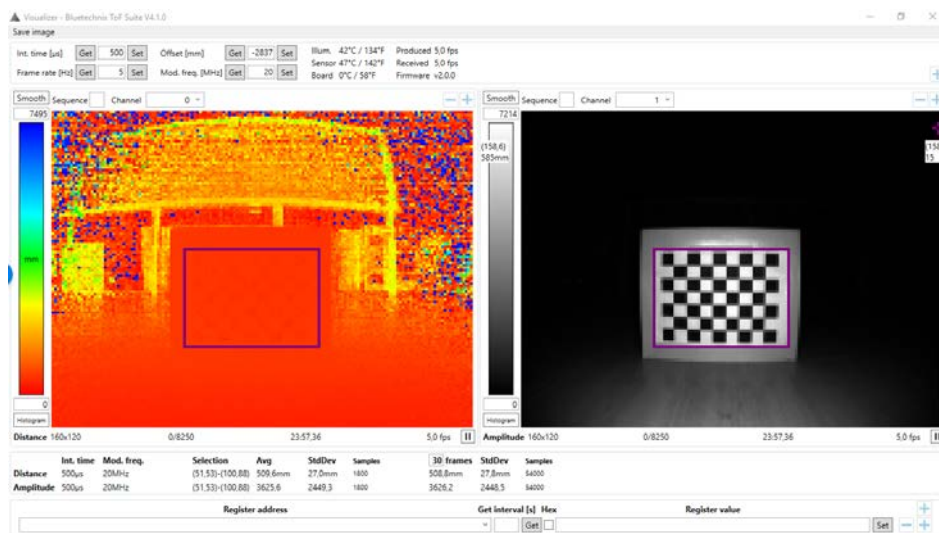


Figura 3.12 Lente 110º-Prueba de distancia 50cm

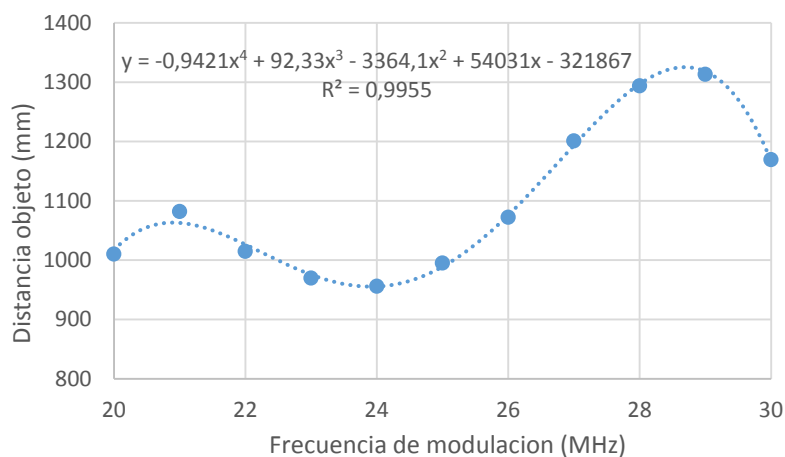
Se puede observar que en general los errores son grandes excepto en la medida de 1m. El objetivo de la calibración es conseguir una estabilización de la medida y de esa manera poder tomar todas las medidas necesarias con los mismos parámetros.

En la Figura 3.12 se puede ver que la imagen de distancia contiene mucho ruido, esto es fácil de corregir aumentando el tiempo de integración.

A continuación se ha realizado una prueba de como varia la distancia obtenida por la cámara aumentando la frecuencia de modulación.

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(μ s)	Frecuencia de Modulación(MHz)	Error (%)
1000	1010,1	500	20	1,01
1000	1082,1	500	21	8,21
1000	1014,7	500	22	1,47
1000	969,8	500	23	3,02
1000	955,9	500	24	4,41
1000	995	500	25	0,50
1000	1072,4	500	26	7,24
1000	1201	500	27	20,10
1000	1293,7	500	28	29,37
1000	1313,3	500	29	31,33
1000	1169,4	500	30	16,94

Tabla 3.5 Lente 110^o-Prueba con diferente Modulación de Frecuencia



Grafica 3.2 Variación de la distancia cámara variando la frecuencia de modulación

En la Tabla 3.5 se muestran los resultados variando la frecuencia entre 20 y 30MHz, para un valor menor o mayor los resultados eran mucho peores.

Se puede observar que tiene una tendencia a aumentar el error según aumenta la frecuencia, pero no ocurre así en todos los casos por lo que no es un parámetro estable, así que para la calibración lo vamos a dejar como viene por defecto además que el fabricante recomienda no cambiar este parámetro.

3.6 Calibración

Una vez realizadas todas las pruebas anteriores se ha creado un método para calibrar la cámara, a continuación se describen los pasos a seguir:

1. Se ajusta el enfoque de la lente de 110° en la posición correcta, aquella en la que se pueda distinguir con claridad un patrón (tablón de ajedrez) colocado a un metro.

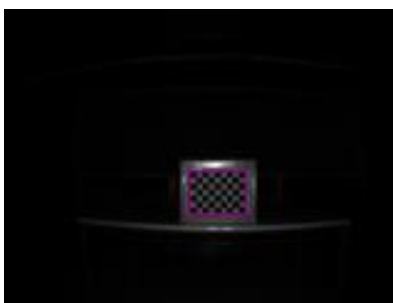


Figura 3.13 Patrón

2. Ajuste del tiempo de integración, para ello hay que conseguir obtener un equilibrio entre una imagen de amplitud que no esté saturada y una imagen de distancia con el menor ruido posible.
3. Una vez que ajustado el tiempo de integración, se ajustará el offset (compensación de la medida) a una distancia de referencia.

En el proyecto se busca que la medida de distancia sea lo más realista posible cuando más cerca se encuentre un objeto, es decir al final si un objeto se encuentra a 2 metros no será significativo que este se encuentre a 2,05m o a 1.95m, pero en cambio un objeto que se encuentre a 50 cm es muy importante saber con precisión a que distancia se encuentra del vehículo. Es por ello que para la realización de la calibración vamos a tomar como referencia que las medidas cercanas sean los más realistas posibles obteniendo así los errores más pequeños cuando el objeto se encuentre cerca.

A continuación se muestran los resultados tomando diferentes distancias como referencia:

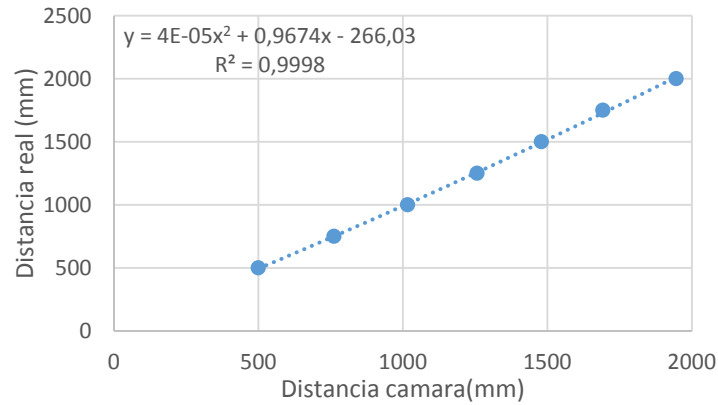
- Distancia de referencia 50cm:

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(μ s)	Frecuencia de Modulación(MHz)	Offset (mm)	Error (%)
300	244,6*	1000	20	-2874	18,47
500	500,2	1000	20	-2874	0,04
750	762,1	1000	20	-2874	1,61
1000	1016,8	1000	20	-2874	1,68
1250	1256,8	1000	20	-2874	0,54
1500	1479,5	1000	20	-2874	1,37
1750	1692,3	1000	20	-2874	3,30

2000	1945,7	1000	20	-2874	2,72
------	--------	------	----	-------	------

Tabla 3.6 Datos tomando como referencia 50cm

(*) Imagen saturada



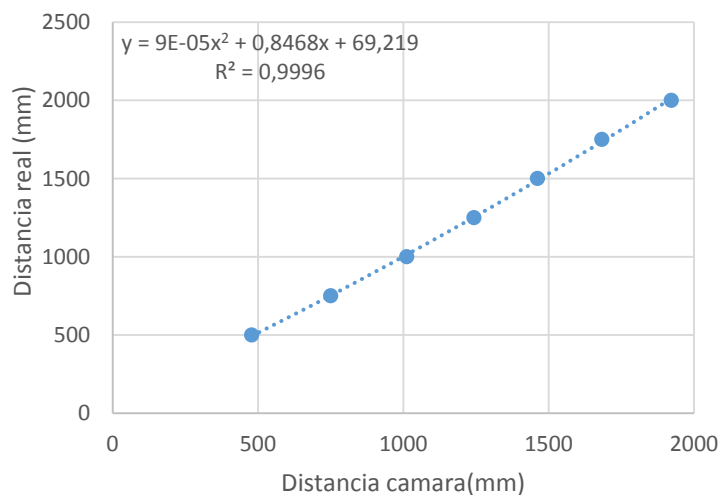
Gráfica 3.3 Datos tomando como referencia 50cm

- Distancia de referencia 75cm:

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(μ s)	Frecuencia de Modulación(MHz)	Offset (mm)	Error (%)
300	209,7*	1000	20	-2890,00	30,10
500	478,4	1000	20	-2890,00	4,32
750	750,1	1000	20	-2890,00	0,01
1000	1011,9	1000	20	-2890,00	1,19
1250	1242,8	1000	20	-2890,00	0,58
1500	1461,5	1000	20	-2890,00	2,57
1750	1682,8	1000	20	-2890,00	3,84
2000	1922,1	1000	20	-2890,00	3,90

Tabla 3.7 Datos tomando como referencia 75cm

(*) Imagen saturada



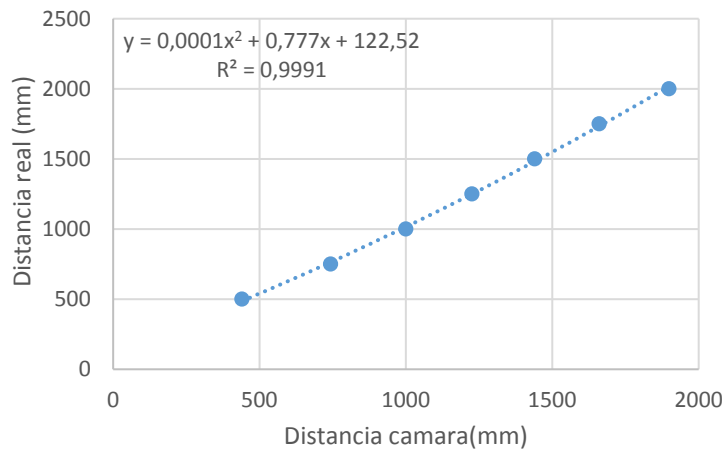
Grafica 3.4 Datos tomando como referencia 75cm

- Distancia de referencia 100cm:

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(μs)	Frecuencia de Modulación(MHz)	Offset (mm)	Error (%)
300	258,5*	1000	20	-2915,00	13,83
500	460,4	1000	20	-2915,00	7,92
750	743,2	1000	20	-2915,00	0,91
1000	1000,1	1000	20	-2915,00	0,01
1250	1225,7	1000	20	-2915,00	1,94
1500	1440,2	1000	20	-2915,00	3,99
1750	1660,2	1000	20	-2915,00	5,13
2000	1898,6	1000	20	-2915,00	5,07

Tabla 3.8 Datos tomando como referencia 100cm

(*) Imagen saturada



Grafica 3.5 Datos tomando como referencia 100cm

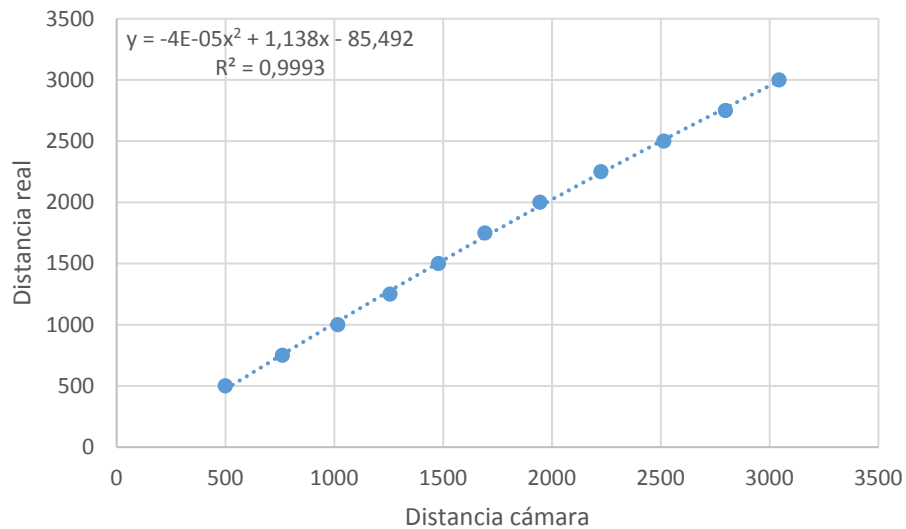
Al observar la Tabla 3.6, lo primero que se puede ver es que en la medida de 30cm se obtiene una imagen saturada lo que provoca un aumento bastante considerable del error, por consejo del fabricante y como se muestra en las tablas no se debe usar la cámara para medir distancias menores a 50cm ya que puede proporcionar un error muy alto, esto ocurre en todas las tablas.

La Tabla 3.8 se descarta ya que como se ha explicado anteriormente se le dará prioridad al resultado que muestre el menor error posible cuando menor sea la distancia al objeto. Si se observan la Grafica 3.3 y la Grafica 3.4, se puede ver que se obtiene el mayor coeficiente de regresión lineal en la primera, además la medida de 50cm obtiene un error de 0,04 que es prácticamente insignificante, y cumple la condición requerida, luego se tomara la distancia de 50cm como referencia.

La calibración queda:

Distancia real(mm)	Distancia cámara(mm)	Tiempo Integración(μ s)	Frecuencia de Modulación(MHz)	Offset (mm)	Error (%)
500	500,2	1000	20	-2874	0,04
750	762,1	1000	20	-2874	1,61
1000	1016,8	1000	20	-2874	1,68
1250	1256,8	1000	20	-2874	0,54
1500	1479,5	1000	20	-2874	1,37
1750	1692,3	1000	20	-2874	3,30
2000	1945,7	1000	20	-2874	2,72
2250	2225,6	1000	20	-2874	1,08
2500	2515	1000	20	-2874	0,60
2750	2797,8	1000	20	-2874	1,74
3000	3043,8	1000	20	-2874	1,46

Tabla 3.9 Tabla de medidas obtenidas por la ToF



Grafica 3.6 Calibración

Aplicando a la Grafica 3.6 el método de mínimos cuadrados, para hallar la recta que mejor se ajusta, queda como resultado una ecuación polinómica de grado 2

$$y = -4 * e^{-5} * x^2 + 1,138 * x - 85,492 \quad (11)$$

Con un coeficiente de correlación lineal muy bueno $R^2=0,9993$, por lo que se usará la ecuación para ajustar los datos obtenidos de distancia.

CAPÍTULO 4

DESARROLLO SOFTWARE/HARDWARE

4.1 Introducción

En este capítulo se expondrá como se obtienen los datos de profundidad y amplitud de la cámara ToF, que características se usaran para la clasificación de las imágenes, como se ha seleccionado el mejor calificador, como es el proceso para obtener la medida de distancia a la que se encuentra el objeto y las pruebas realizadas.

4.2 Software de captura de datos

Para obtener los datos necesarios el primer paso es el estudio del software de la cámara de tiempo de vuelo. El fabricante proporciona un kit de desarrollo de software (SDK) el cual permite el desarrollo de una aplicación informática en MATLAB.

MATLAB (*MATrix LABoratory*, cuya traducción literal sería "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M)[23].

Para llevar a cabo la captura de datos se ha creado un programa en Matlab que se puede ver en el Anexo I, el cual es capaz de configurar los parámetros elegidos en la calibración del capítulo anterior y obtener la información de profundidad y amplitud obtenida por la cámara, como muestra la Figura 4.1.

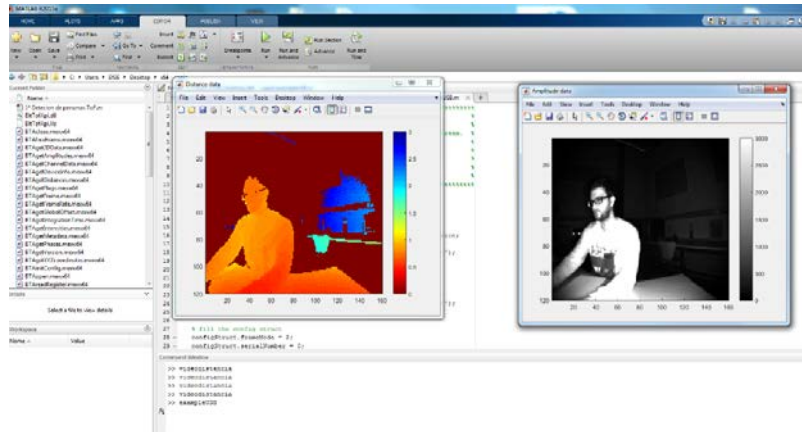


Figura 4.1 Imagen de profundidad y amplitud captada por Matlab

En la Figura 4.2 se pueden ver las dos matrices obtenidas, la primera con los datos de amplitud y la segunda con los datos de profundidad.

Workspace	
Name	Value
FrameBufferAmp	120x160x30 double
FrameBufferDist	120x160x30 double

Figura 4.2 Matrices de distancia y amplitud

El tamaño de las matrices corresponde con la resolución de la cámara 120 filas por 160 columnas en total son 19200 pixeles de información en cada matriz y la dimensión de la matriz corresponderá con el número de imágenes tomadas en este caso 30 imágenes.

En la imagen de distancia la información que captura cada pixel se corresponde con la distancia a la que se encuentra el objeto captado.

Variables - P										
P										
120x160 double										
	102	103	104	105	106	107	108	109	110	
103	1.9242	1.8466	1.9717	1.8449	1.8522	1.8517	1.7890	1.8453	1.7681	
104	1.7469	1.8281	1.7926	1.9053	1.7860	1.8026	1.8497	1.8652	1.7452	
105	1.7752	1.8222	1.8394	1.8155	1.8272	1.8928	1.8488	1.8300	1.8137	
106	1.8112	1.7913	1.8263	1.8554	1.9531	1.7825	1.8329	1.7707	1.8327	
107	1.7808	1.7803	1.8526	1.8539	1.8084	1.8669	1.7722	1.8327	1.8533	
108	1.8495	1.8165	1.8806	1.9372	1.8641	1.8832	1.8926	1.8459	1.7994	
109	1.8175	1.8693	1.8273	1.9369	1.7772	1.9009	1.8446	1.8367	1.9134	
110	1.7896	1.9317	1.8838	1.8312	1.9120	1.9099	1.8376	1.8666	1.7722	
111	1.9340	1.8982	1.9024	1.8945	1.9919	1.8692	1.8506	1.7794	1.8280	
112	1.9402	1.9748	2.0234	1.9480	1.8597	1.8920	1.8623	1.7915	1.8322	
113	1.9315	1.8615	2.0089	1.9158	1.9661	1.7989	1.7992	1.7343	1.8549	
114	1.8678	1.9779	2.0089	1.9373	1.9467	1.7888	1.9104	1.8292	1.8278	
115	1.9182	2.0042	1.9030	1.8928	1.8704	1.8653	1.7733	1.8628	1.7814	
116	1.9006	1.9757	1.9223	2.0178	2.0169	1.8801	1.8744	1.8470	1.8654	
117	1.9757	1.8489	1.8848	2.0104	2.0074	1.9382	1.9206	1.8367	1.7891	
118	1.8214	1.9377	1.8650	1.8072	1.9763	2.0510	1.8522	1.9082	1.8451	
119	1.8034	1.8246	1.9497	1.8360	1.9530	1.8885	1.8840	1.8193	1.8880	
120	1.8241	1.9125	1.8677	1.9403	1.7193	1.9030	1.8928	1.7180	1.7772	

Figura 4.3 Parte de una matriz de distancias

En la Figura 4.3 se puede ver una parte de la matriz de profundidad, en ella se observa el valor que tiene cada pixel, el cual corresponde con la distancia en metros a la que se encuentra los objetos de la imagen. Cuando se obtiene una medida de distancia con la matriz de profundidad, no se debe coger un solo pixel para saber a qué distancia se encuentra la cámara del objeto, ya que un solo pixel puede tener un error muy alto por el ruido que sufren estas imágenes como se vio en el capítulo dos, por lo que para obtener la medida a un objeto es aconsejable coger un grupo de pixeles.

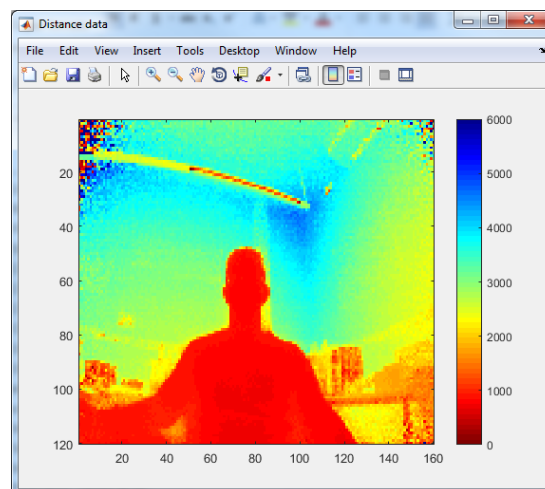


Figura 4.4 Imagen de profundidad

En la Figura 4.4 se puede ver la matriz de profundidad como si fuera una imagen a color, para ello utilizamos una función de Matlab capaz de mostrar una imagen con colores escalados, de esta manera si utilizamos una escala de 0 a 6000 se mostrará una imagen con una escala de colores capaz de mostrar de un color diferentes todos los elementos que se encuentren en este rango.

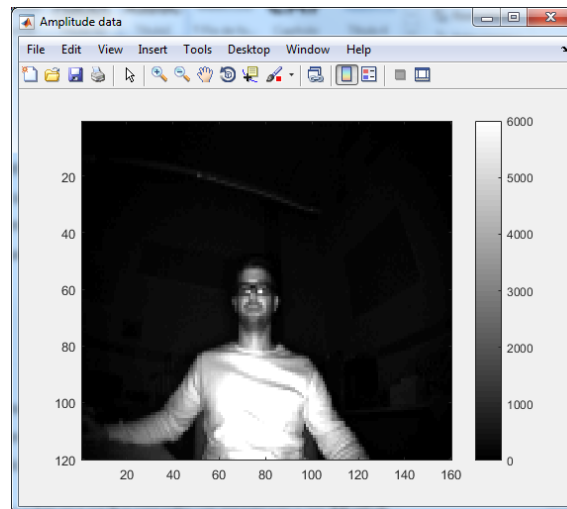


Figura 4.5 Imagen de amplitud

Lo mismo ocurre la Figura 4.5, con los valores de amplitud de la onda que proporciona la cámara se mostrara en este caso una imagen en escala de grises de la escena.

En la Figura 4.6 se puede ver una parte de la matriz de amplitud.

Variables - P										
P										
120x160 double										
	89	90	91	92	93	94	95	96	97	
14	426	421	416	416	347	388	359	393	451	
15	460	441	356	393	344	393	387	364	378	
16	378	383	404	412	387	480	369	371	427	
17	376	439	443	439	414	421	414	449	391	
18	495	431	423	429	470	395	379	430	426	
19	454	443	410	386	481	422	387	416	409	
20	352	401	462	458	471	471	455	434	437	
21	466	380	451	418	421	408	395	390	421	
22	404	425	416	395	493	440	460	414	374	
23	526	462	462	425	522	477	469	500	444	
24	452	441	499	452	522	478	465	438	423	
25	403	477	437	432	512	423	502	423	447	
26	451	457	435	420	470	421	401	409	412	
27	517	383	449	364	407	343	364	406	410	
28	353	360	340	327	353	318	344	333	356	
29	362	318	322	345	344	324	354	349	428	
30	353	325	478	306	318	348	333	333	322	
31	362	325	322	339	331	322	320	324	339	

Figura 4.6 Parte de una matriz de amplitud

4.2 Hardware de captura de datos

Una vez que se ha desarrollado el programa (Anexo I) necesario para la obtención de datos y entendido la forma en la que se obtienen, se procede a la fabricación de un soporte para colocar la cámara en el techo de un vehículo.

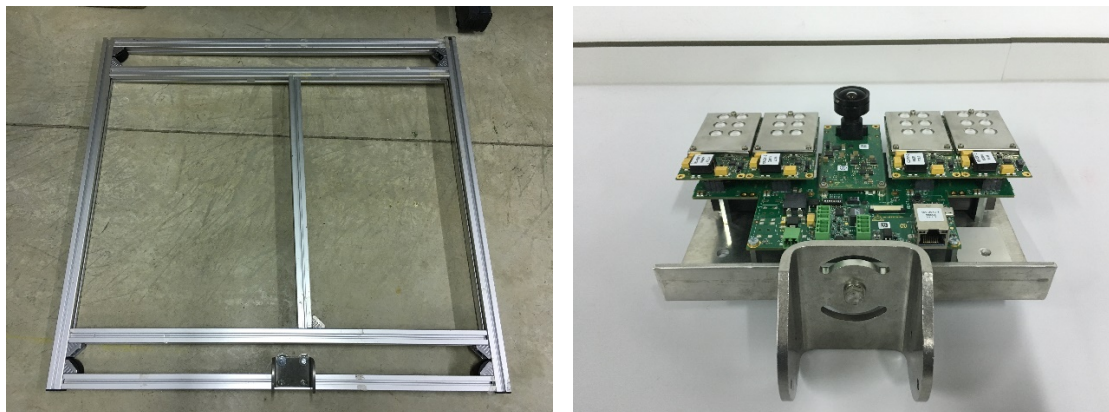


Figura 4.7 Soporte de la cámara ToF para el coche

La estructura de la Figura 4.7, ira montada sobre la vaca de un vehículo, montando la cámara encima de esta. La cámara se posiciona inclinada hacia abajo, de manera que sea visible en la imagen el lateral del coche y que cubra como mínimo una distancia aproximada de tres metros que es el rango que alcanza la cámara sin cometer demasiado error.

En la Figura 4.8 y la Figura 4.9 se puede ver cómo queda la cámara montada en el coche.



Figura 4.8 Cámara ToF montada en el vehículo

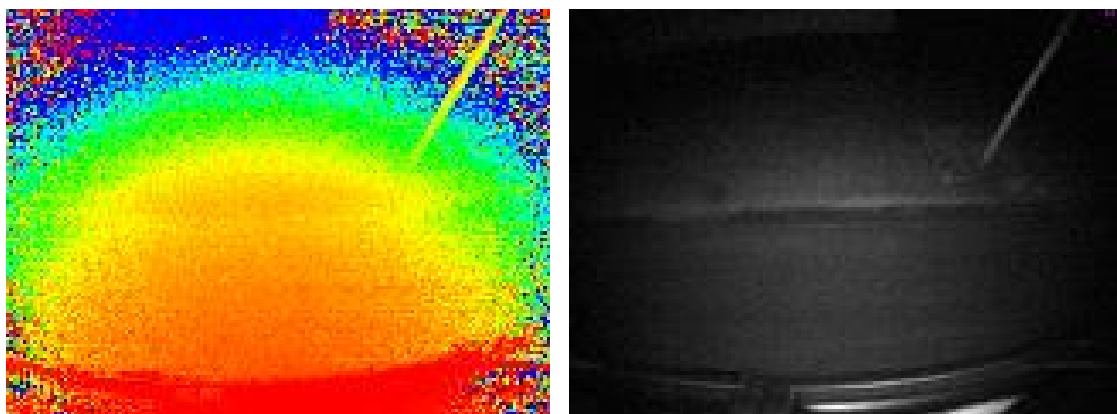


Figura 4.9 Cámara ToF montada en el vehículo vista perpendicular

Para realizar la prueba es necesario una conexión de corriente alterna, que tenga una potencia mínima de 40W para alimentar la cámara, para ello se utilizó un inversor de corriente capaz de transformar la señal continua de la batería del coche en una señal alterna con 300W de potencia [24].

Una vez concluido todo el proceso anterior se realizó una salida con la cámara montada en el vehículo para tomar los datos necesarios para este proyecto.

A continuación se muestran algunas de las imágenes que fueron tomadas durante la prueba.



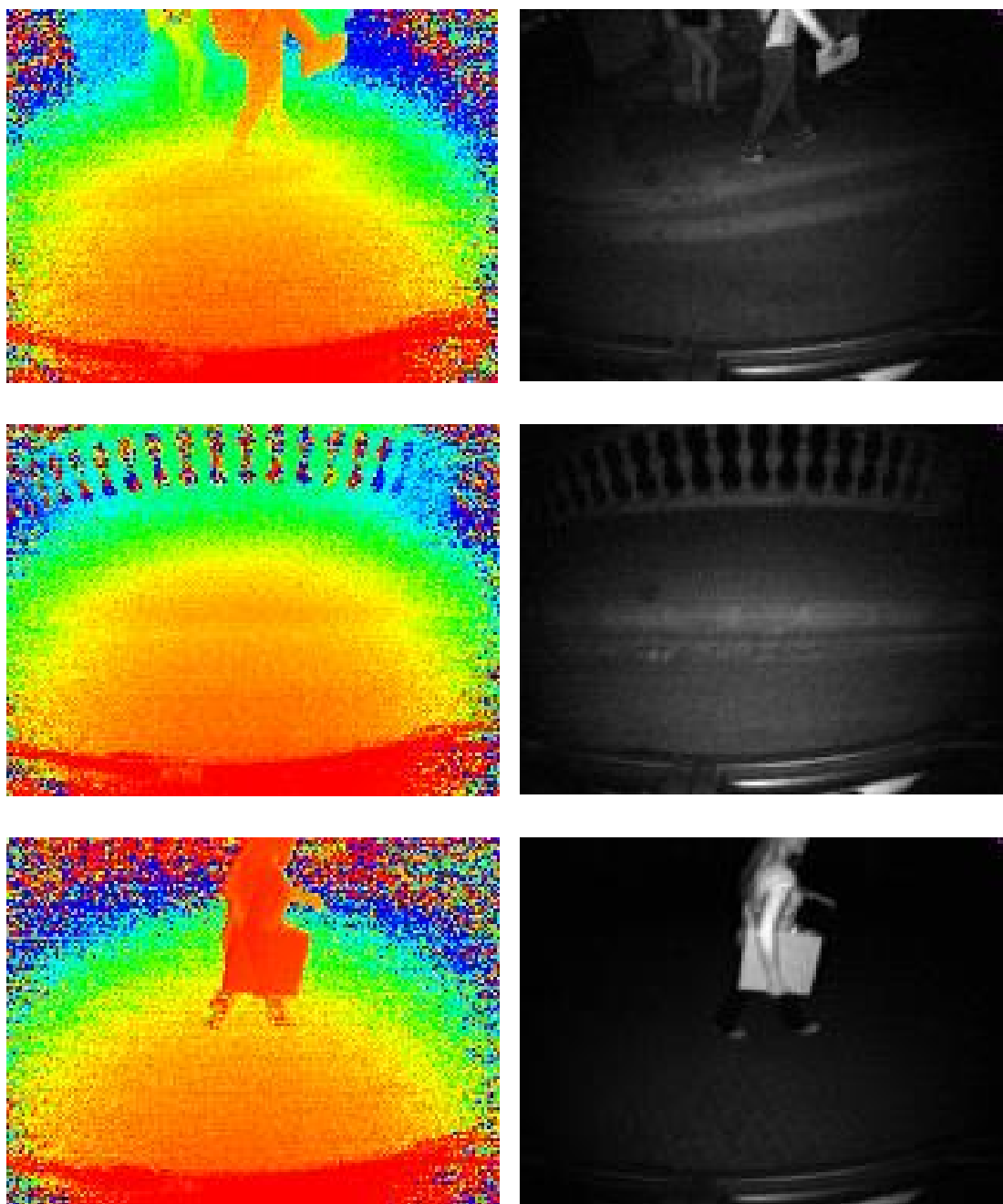


Figura 4.10 Algunas imágenes de profundidad y amplitud captadas en la prueba

4.3 Extracción de características

En este apartado se pretende extraer las características necesarias de las imágenes de profundidad y amplitud para entrenar un clasificador capaz de detectar un borde. En la Figura 4.11 se puede ver el proceso a seguir.

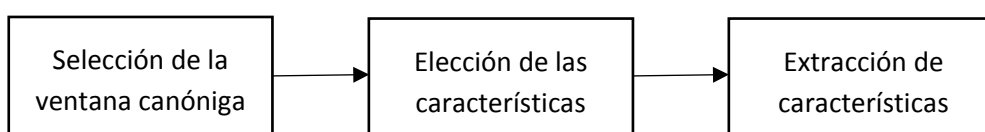


Figura 4.11 Proceso de extracción de características

Para la extracción de características se define una ventana canónica, a la cual se le da una forma rectangular debido a que normalmente un borde es estrecho y alargado, los tamaños de la ventana canónica que han sido elegidos para el entrenamiento son de 20x40 y de 10x40 píxeles.



Figura 4.12 Ejemplo de imagen de amplitud que contiene un borde

En la Figura 4.12 se puede ver un ejemplo de una imagen que contiene un borde, como se vio anteriormente la resolución de la imagen es de 120x160 píxeles, por lo que no tenemos demasiados píxeles para extraer características, por ello se ha elegido una ventana canónica que contenga suficientes características como para entrenar un clasificador con buenos resultados. El análisis métrico obtenido sobre el conjunto de entrenamiento determinará cuál de estas dos ventanas ofrece un resultado óptimo.

Así se procederá a extraer muestras de diferentes imágenes en las que se encuentre un borde y muestras en las que no se encuentre un borde para el posterior entrenamiento del clasificador.

Algunos ejemplos de las muestras extraídas

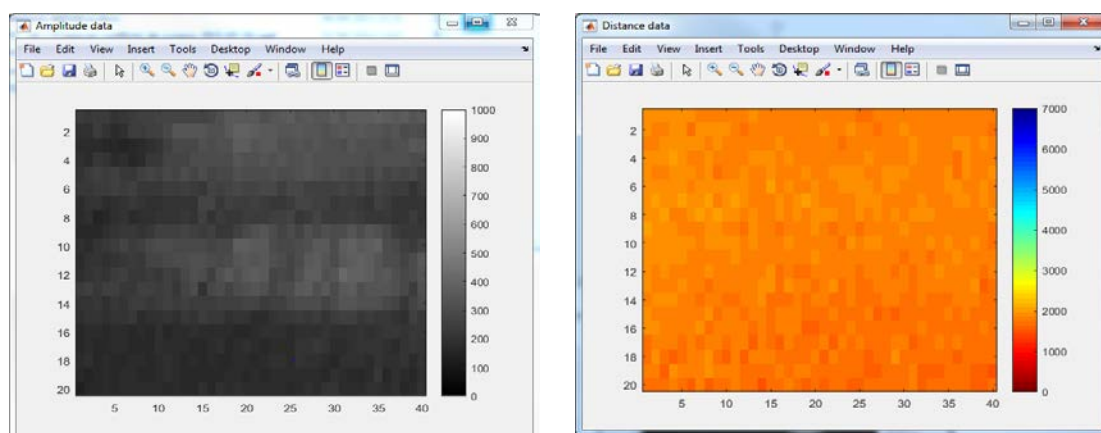


Figura 4.13 Ejemplo de ventanas canónicas con borde

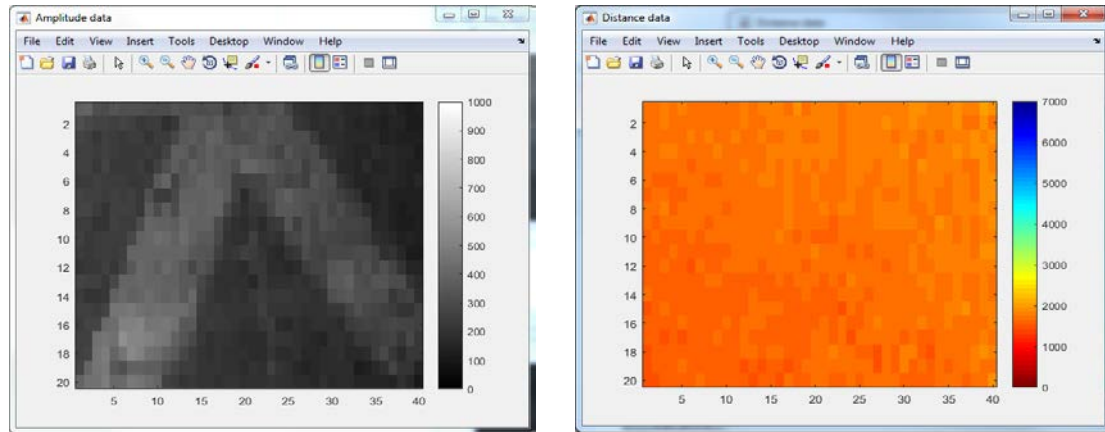


Figura 4.14 Ejemplo ventanas canónicas sin borde

Se han extraído 150 muestras que contienen un borde y 300 muestras que no contienen un borde.

Una vez extraídas las muestras de las imágenes de profundidad y amplitud se ha procedido al cálculo de las características. Si se utiliza el método HOG las cuales fueron explicadas en el capítulo 2, se obtiene un vector de características por lo tanto es importante asegurarse de que este vector contenga la cantidad correcta de información para ello se analizarán los diferentes rendimientos en el próximo apartado. Al variar el tamaño de celda de HOG, se puede ver el efecto que tiene sobre la cantidad de información obtenida.

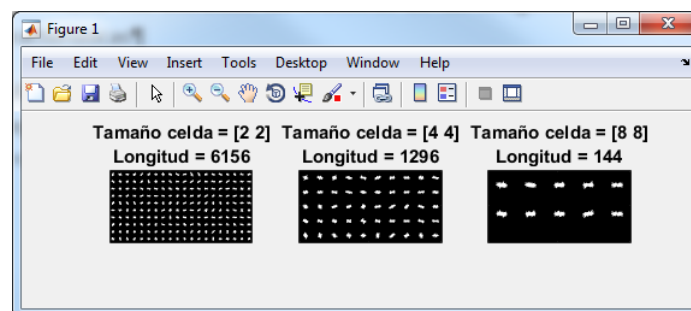


Figura 4.15 Diferentes tamaños de HOG aplicados a una ventana canónica

En la Figura 4.15 se muestran las longitudes del vector de características para diferentes tamaños de celda. Se observa que para un tamaño de celda de 8x8 con 144 valores codifica muy poca información mientras que para un tamaño de celda de 2x2 codifica 6156 valores, esto es muchísima información pero aumenta significativamente la dimensión del vector de características, mientras que el tamaño de celda 4x4 tiene un número de características más limitado con 1296 valores, hay que encontrar un equilibrio en el cual se obtenga un buen resultado del clasificador sin que haya demasiadas características que puedan ser redundantes.

Lo mismo ocurre en la extracción de características LBP que fueron explicadas en el capítulo 2, se divide la ventana en celdas, si el tamaño de la celda es demasiado grande se pierden detalles locales, por lo que hay que elegir un tamaño adecuado para ello como en el caso anterior se deben analizar los resultados del clasificador.

4.4 Entrenamiento del clasificador

En este apartado se procede al entrenamiento del clasificador, para obtener el mejor rendimiento posible se seguirá el procedimiento de la Figura 4.16.

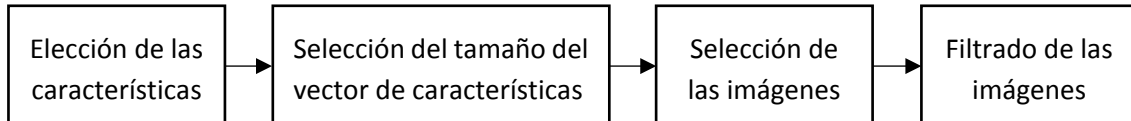


Figura 4.16 Proceso de entrenamiento del clasificador

Para el entrenamiento de los clasificadores se utilizan 450 muestras, 150 en las que se encuentra un borde y 300 en las que no. Los clasificadores que se van a emplear son dos de los explicados en el capítulo 2, SVM y KNN y para comprobar su rendimiento se usará el método de validación cruzada, leave-one-out. Los clasificadores van a ser entrenados con la siguiente configuración

KNN		SVM	
K	4	'Standardize'	true
'Standardize'	true	'KernelFunction'	'linear'

Tabla 4.1 Configuración de los clasificadores

En la tabla 4.1, se puede ver que el clasificador KNN se va a entrenar con una vecindad de 4, se estandarizarán los datos del predictor utilizando la media y desviación típica, estos también se estandarizarán en el SVM, en el que se usará la KernelFuncion lineal que es la que viene por defecto cuando existen dos clases ya que la polinómica o la gaussiana complicarían los cálculos

En primer lugar se elegirán las características que den un mejor rendimiento, para ello se comprueba que con que descriptor se obtiene un mayor rendimiento.

Si se utiliza el vector de características LBP, con tamaño de celda de 5x5, una vecindad de 8 y un vector de características de 7776 valores el resultado es:

KNN		SVM	
Tamaño del vector de características:7776			
Rendimiento	33.333	Rendimiento	66.667
Error	66.667	Error	33.333

Tabla 4.2 Rendimiento del clasificador con características LBP 5x5

Utilizando un vector de características HOG, con un tamaño de celda de 2x2 con vector de características de 6156:

KNN		SVM	
Tamaño del vector de características:6156			
Rendimiento	95.778	Rendimiento	93.111
Error	6.889	Error	4.222

Tabla 4.3 Rendimiento del clasificador con características HOG 2x2

En la Tabla 4.2 se puede ver que el resultado con las características LBP el clasificador KNN tiene un resultado de acierto del 33.333% siendo el porcentaje de error más alto con un 66.667% y en el caso del SVM tiene un porcentaje de acierto del 66.667%, mientras que en la Tabla 4.3 se obtiene un rendimiento muy superior en el caso del clasificador KNN con un rendimiento del 95.778%, por lo que se usaran estas las características HOG para la clasificación.

El siguiente paso es la elección del tamaño de la ventana canónica con el que se obtenga un mayor rendimiento, en la Tabla 4.3 se observa el resultado de la ventana canónica para un tamaño de 20x40, si se usa el tamaño de 10x40 el resultado es:

KNN		SVM	
Tamaño del vector de carasteristicas:6156			
Rendimiento	92.000	Rendimiento	79.333
Error	8.000	Error	20.667

Tabla 4.4 Rendimiento del clasificador con una ventana canónica de 10x40

En la Tabla 4.4, se observa un rendimiento del 92% en el clasificador KNN, esto es 3.778% menor comparado con el mejor resultado de la Tabla 4.3 , por lo que se usará la ventana canónica de 20x40 al tener mayor rendimiento y resultar más fácil encontrar un buen resultado en las imágenes debido a su mayor tamaño.

Una vez elegido el tamaño de la ventana canónica, se elegirá el tamaño del vector de características para esta ventana, en la Tabla 4.3 se vio el resultado con un vector de características de 6156 valores.

Si se utiliza un tamaño de celda de 4x4, con un vector de características de 1296 valores se obtiene un resultado de:

KNN		SVM	
Tamaño del vector de carasteristicas:1296			
Rendimiento	96.889	Rendimiento	92.889
Error	3.778	Error	7.111

Tabla 4.5 Rendimiento del clasificador con características HOG 4x4

Utilizando un tamaño de celda de 8x8, con un vector de características de 144 valores se obtiene un resultado de:

KNN		SVM	
Tamaño del vector de carasteristicas:144			
Rendimiento	93.778	Rendimiento	87.111
Error	6.222	Error	12.889

Tabla 4.6 Rendimiento del clasificador con características HOG 8x8

Si se observan la Tabla 4.3, la Tabla 4.5 y la Tabla 4.6, se puede ver que el mejor resultado se obtiene en el clasificador KNN con un rendimiento del 96.889%, utilizando un tamaño de celda de 4x4 y un vector de 1296 valores. En el caso de un tamaño de celda 8x8 con un vector de 144 valores, que es bastante menor si se compara con el vector de la celda 4x4, lo cual hace disminuir su rendimiento hasta el 93.778%, en cambio cuando se usa el tamaño de la celda de

[2 2] con un vector de 6156 valores se obtiene un rendimiento SVM del 95.778%, aunque la diferencia con el tamaño de celda 4x4 es de tan solo un 1%, se ha optado por la elección del tamaño de celda 4x4, además de por su rendimiento superior por contar con 4860 valores menos, lo que simplifica bastante los cálculos a realizar.

Para realizar todos estos entrenamientos se ha usado solamente la imagen de amplitud, a continuación se muestra el resultado de entrenar al clasificador usando un vector que contenga las características de la imagen de amplitud y las de la imagen de profundidad.

KNN		SVM	
Tamaño del vector de carasteristicas:2592			
Rendimiento	92.667	Rendimiento	91.333
Error	7.333	Error	7.333

Tabla 4.7 Entrenamiento del clasificador utilizando la imagen de amplitud y distancia

En la Tabla 4.7 se obtiene el mejor resultado en el clasificador KNN con un 92.667%, esto es un 4% inferior al obtenido en la Tabla 4.5. Hay que tener en cuenta que el vector de características pasa a tener el doble de tamaño, de 1296 valores pasa a 2592 datos, aunque en un primer momento se pueda pensar que al introducir más datos pueda mejorar el rendimiento del clasificador también se introduce mucho más ruido sobretodo en la imagen de profundidad que contiene normalmente más ruido que la de amplitud, como consecuencia se reduce su rendimiento.

Para reducir el ruido de las muestras se han probado con los diferentes filtros comentados en el capítulo 2, dando como mejor resultado el filtro de la mediana.

En la Tabla 4.8 se obtiene el rendimiento del clasificador usando el filtro de la mediana únicamente en la imagen de amplitud.

KNN		SVM	
Tamaño del vector de carasteristicas:1296			
Rendimiento	97.111	Rendimiento	93.111
Error	2.889	Error	6.889

Tabla 4.8 Entrenamiento del clasificador utilizando la imagen de amplitud y filtro de la mediana

Si se obtiene su curva ROC para comprobar su precisión en la Figura 4.17, con una área bajo la curva ROC de 0.9981, lo que puede interpretarse como la probabilidad de que ante un par de muestras, una que se trate de borde y otra que no se, el clasificador los clasifique correctamente.

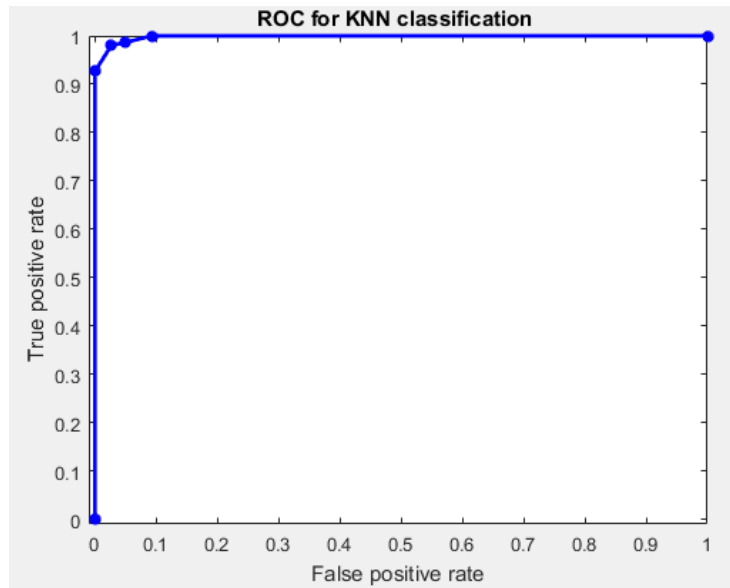


Figura 4.17 Curva ROC para el clasificador KNN con las características HOG de la imagen de amplitud

Y si se utiliza el filtro de la mediana en la imagen de amplitud y distancia:

KNN		SVM	
Tamaño del vector de carasteristicas:2592			
Rendimiento	97.778	Rendimiento	95.333
Error	2.222	Error	4.667

Tabla 4.9 Entrenamiento del clasificador utilizando la imagen de amplitud y de distancia y filtro de la mediana

Obteniendo una curva ROC mostrada en la Figura4.18 en la cual se obtiene área bajo la curva ROC (AUC) de 0.9987.

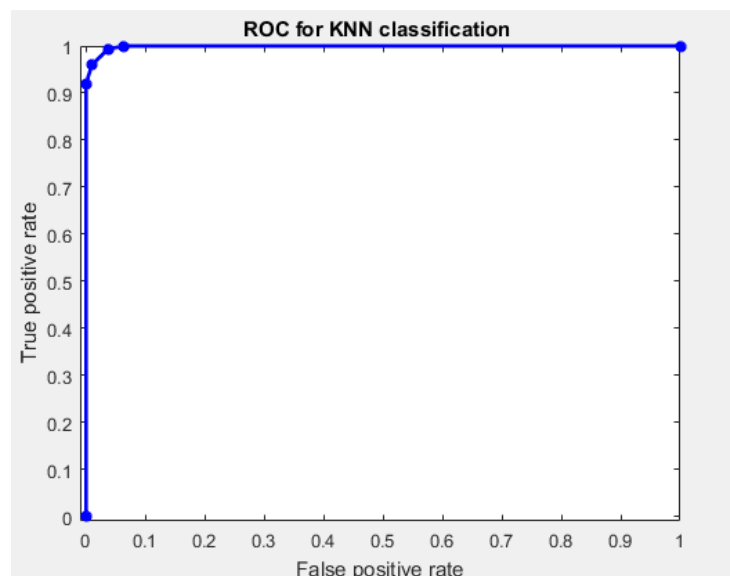


Figura 4.18 Curva ROC para el clasificador KNN con las características HOG de la imagen de amplitud y profundidad

Si se compararan las dos curvas ROC en la figura 4.19 se puede observar que los dos clasificadores dan un resultado muy semejante.

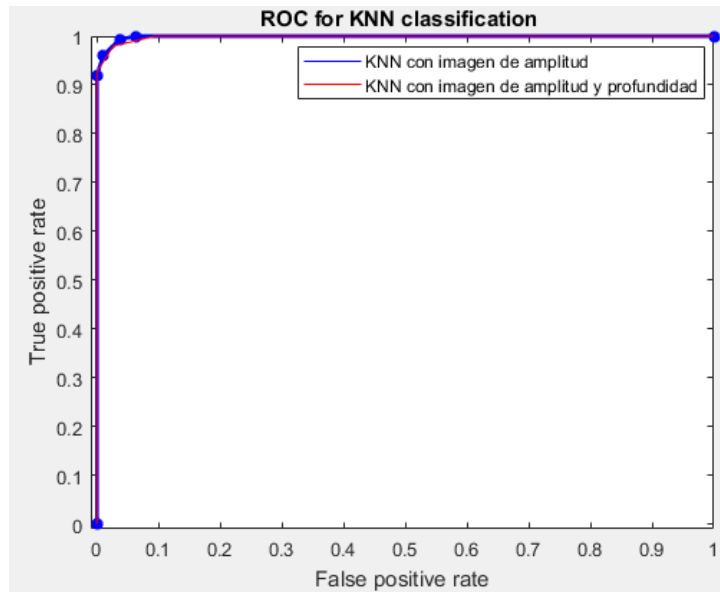


Figura 4.19 Comparación de curvas ROC

Como se esperaba, la utilización de filtros reduce el ruido de las imágenes y en consecuencia aumenta el rendimiento del clasificador. En la Tabla 4.9 se obtiene un rendimiento del clasificador KNN del 97.778% AUC ROC de 0,9987, siendo el mayor rendimiento obtenido en todo el proceso, por lo cual estas serán las características usadas para el entrenamiento del clasificador, en resumen para entrenar el clasificador se le aplicaran el filtro de la mediana a las muestras y se extraerán las características HOG , en una ventana canónica de 20x40 de las imágenes de amplitud y distancia con un tamaño de celda 4x4 que produce un vector de características de 2592 elementos.

4.6 Proceso de medición de distancia al vehículo

Para saber a qué distancia se encuentra el borde del vehículo se seguirán los siguientes pasos:

1. Se extraen los candidatos de la parte central de la imagen que es la que corresponde con la perpendicular de la cámara.
2. Una vez extraídos los candidatos se le aplica el filtro de la mediana y se el vector de características HOG para su posterior clasificación.
3. Una vez encontrado el borde en la imagen, se calcula la distancia del borde a la cámara para posteriormente con el teorema de Pitágoras obtener la distancia del borde al vehículo.

En el primer paso se recorre la imagen por su parte central con un tamaño de 20x40 píxeles que es el que el correspondiente a la ventana canónica de la que se extraen las características para su clasificación, como esta ventana tiene 40 columnas y el tamaño la imagen obtenida por la cámara de tiempo de vuelo es de 120x160 píxeles, la cual tiene 160 columnas como se observa en la , para que el centro de la venta canónica se corresponda con el centro de la imagen se tendrá que situar entre la columna 60 y 100, de esta manera se recorrerá la imagen por su centro el cual corresponde con la perpendicular de la cámara ToF

En el segundo paso una vez obtenidos los candidatos del centro de la imagen se filtrarán y extraerán sus características para su clasificación.

En el último paso una vez obtenida la ventana canónica de profundidad donde se encuentra el bordillo, primero se halla la distancia del bordillo a la cámara para ello se halla la mediana de la matriz de la ventana canónica de profundidad que como se comentó anteriormente cada dato esa matriz se corresponde con la distancia del objeto a la cámara en metros. Se usa la mediana en lugar de la media para evitar que si hay algún error muy grande en un dato no se vea afectada la medida.

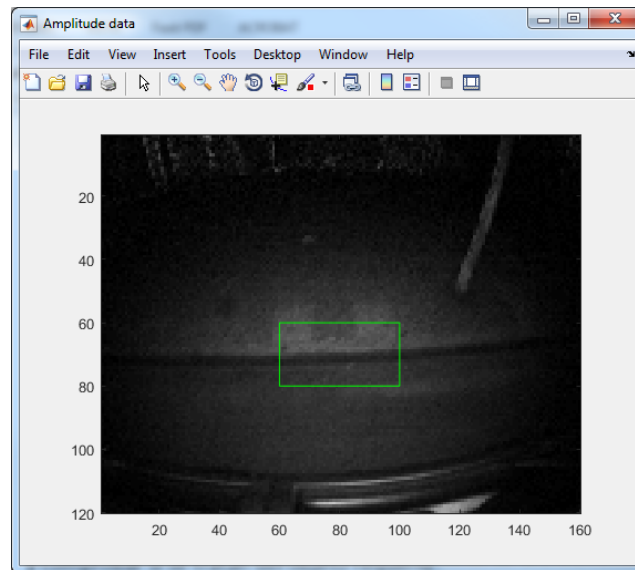


Figura 4.20 Bordillo usado para la medida de distancia

En la se puede observar el resultado de este proceso en el que la distancia del bordillo a la cámara es:

$$Distancia\ camara = 1784\ mm \quad (12)$$

Para obtener una medida de distancia más precisa, se usa la ecuación calculada en el apartado de calibración para tener en cuenta el error que de la cámara ToF.

$$Distancia\ camara\ sin\ error = -4 * e^{-5} * 1784^2 + 1,138 * 1784 - 85,492 \quad (13)$$

$$Distancia\ camara\ sin\ error = 1817.4\ mm$$

Una vez conocido el dato de distancia de la cámara al bordillo y conocida la distancia a la que se encuentra la cámara del suelo que son 1460 mm, la cual es siempre constante, se puede calcular finalmente mediante Pitágoras la distancia que hay entre el bordillo y el vehículo. En la se puede ver cómo queda el triángulo de medida.



Figura 4.21 Triangulo de medida

La distancia que hay del bordillo al vehículo en la es:

$$\text{Distancia del bordillo al vehiculo} = \sqrt{1817.4^2 - 1460^2} = 1082.3 \text{ mm} \quad (14)$$

4.7 Pruebas del clasificador

Para comprobar el rendimiento del clasificador en situaciones reales se han utilizado 30 imágenes que no han sido utilizadas para el entrenamiento del clasificador.

Para ello se seguirá el siguiente procedimiento: en primer lugar se aplicaran los filtros correspondientes a las imágenes de amplitud y distancia para eliminar el ruido, una vez eliminado se procede a generar los candidatos para clasificarlos, de esta manera poder saber si en el candidato se encuentra un bordillo, para ello se recorrerán las imágenes de amplitud y distancia con una ventana que tenga el tamaño de la ventana canónica usada anteriormente, de 20x40, a cada ventana se le extraerán las características y por último se clasificarán. A continuación se muestra un flujograma del proceso en la Figura 4.22.

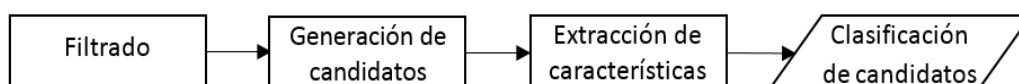
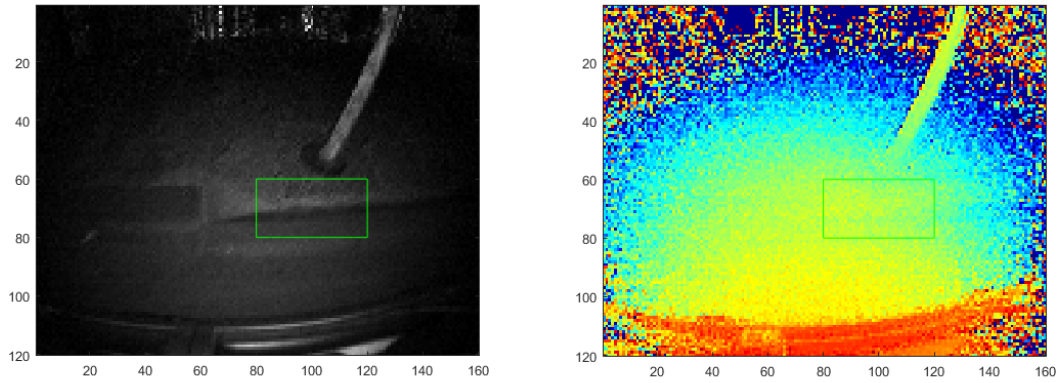


Figura 4.22 Proceso de clasificación

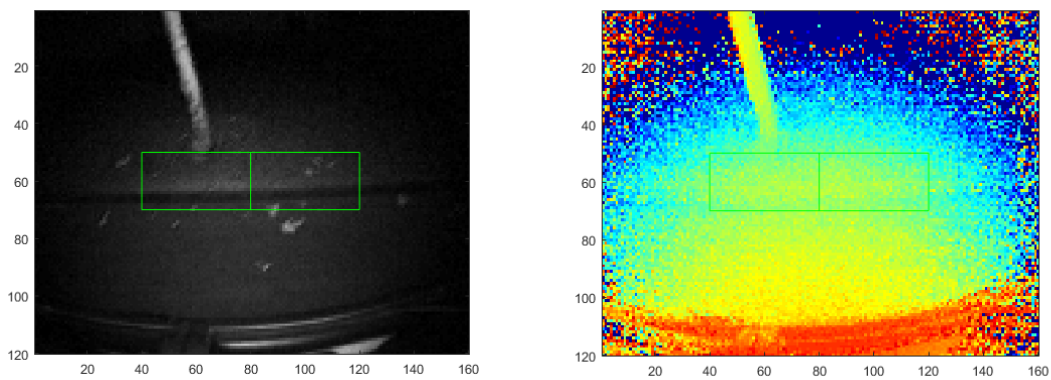
Para hacer el proceso más eficiente se descartaran algunos zonas de la imagen, estas zonas son: la zona inferior de la imagen ya que en esta zona siempre se encontrara el lateral del

coche y nunca se podrá clasificar como un bordillo y los extremos laterales de la imagen por contener demasiado ruido.

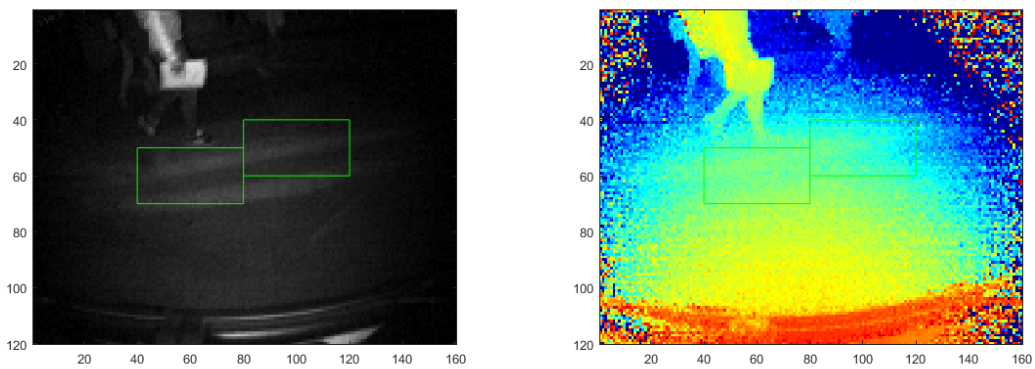
En la Figura 4.23 se exponen algunas pruebas , el resto se pueden ver en el Anexo II



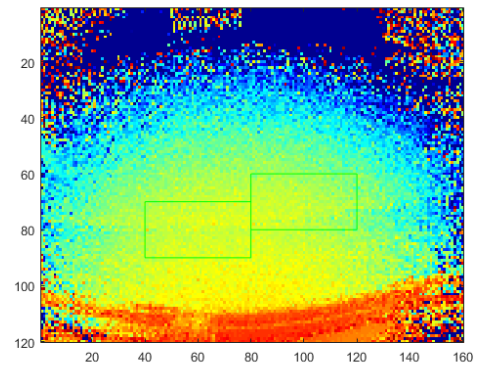
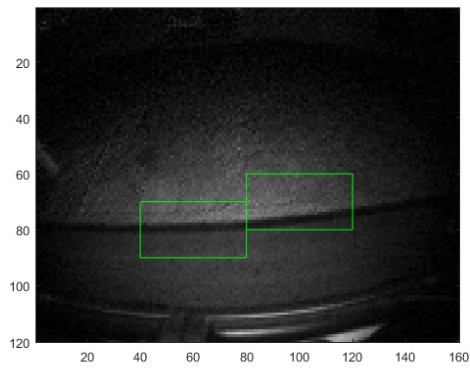
Prueba 1- El bordillo se encuentra a 1099.3 mm



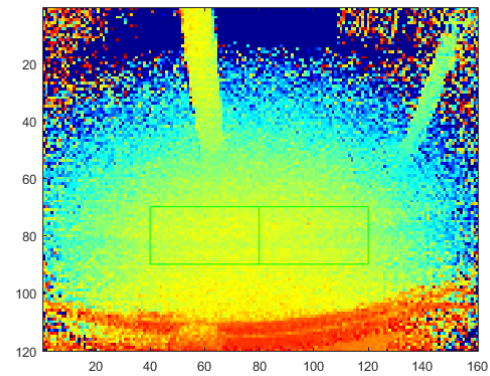
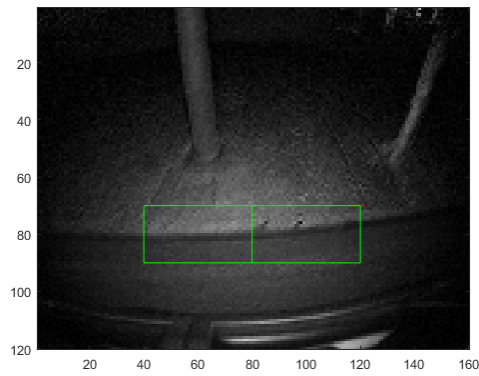
Prueba 4-El bordillo se encuentra a 1276.6mm



Prueba 23-El bordillo se encuentra a 1922.9mm



Prueba 25-El bordillo se encuentra a 1022.81 mm



Prueba 30-El bordillo se encuentra a 928.11 mm

Figura 4.23 Pruebas del clasificador

En las 30 pruebas realizadas se ha obtenido 24 imágenes bien clasificadas y 6 con errores de clasificación, dando un rendimiento del 80%.

CAPÍTULO 5

CONCLUSIÓN Y TRABAJOS FUTUROS

5.1 Introducción

En este capítulo se presentarán los resultados obtenidos durante la ejecución del proyecto, así como los trabajos futuros derivados del mismo.

5.2 Conclusión

Las conclusiones que se han obtenido en este trabajo son:

- ✓ Se ha realizado un estado arte sobre la tecnología de visión por computador basada en la medición ToF. El estudio demuestra un aumento del número de dispositivos en el mercado, así como una disminución del precio. Las cámaras ToF puede competir con la tecnología LIDAR 3D en un futuro.
- ✓ En este trabajo se ha implementado un sistema que permite la detección y medición de bordillos en condiciones reales para su aplicación en un vehículo. Para ello, se ha empleado únicamente la información adquirida por una cámara ToF ubicada en el techo del mismo.
- ✓ Se ha llevado acabo un modelo de calibración de la cámara de tiempo de vuelo (Sentis3D-M420). Para ello se ha analizado como varia la distancia que obtiene la cámara con el tiempo de integración y la frecuencia de modulación y como le afectan a la imagen de profundidad y amplitud. El modelo ha permitido obtener un error menor de del 2% en distancias menores o iguales a un metro y medio.
- ✓ Se ha creado un software capaz de calibrar la cámara, capturar y reproducir datos en Matlab y un hardware para poder utilizar la cámara en un vehículo.
- ✓ Para el entrenamiento del clasificador se han utilizado 100 imágenes de las cuales se han obtenido 450 muestras, 150 que contienen un bordillo y 300 muestras en las que no se encuentra un bordillo. Para obtener el mayor rendimiento del clasificador mediante el método de validación cruzada Leave-one-out, se han probado diferentes tamaños de ventana canónica, diferentes vectores de características, diferentes filtros, diferentes clasificadores y las diferentes imágenes de amplitud y profundidad proporcionadas por la cámara obteniendo el mejor resultado en el clasificador KNN, con un rendimiento del 97.778% y con un AUC ROC de 0,9987. Para obtener este resultado se ha elegido una venta canónica de 20x40píxeles, se han filtrado las muestras mediante el filtro de la mediana para reducir su ruido y se han extraído las características HOG de la ventana canónica de amplitud y distancia con un tamaño de celda de 4x4 obteniendo un vector de características de 2592 valores.
- ✓ Se ha realizado una prueba del software con las 30 imágenes que no se han utilizado para el entrenamiento del clasificador, en la que se ha obtenido un una correcta clasificación en 24 imágenes y en las 6 restantes se han producidos errores de clasificación, por lo que en esta prueba se ha obtenido un rendimiento del 80%.
- ✓ Se ha desarrollado un método para medir la distancia entre el vehículo y el bordillo.

Para acabar se puede concluir que se han cumplido todos los objetivos propuestos para este TFG.

5.3 Trabajos futuros

A continuación se presentan una serie de líneas o proyectos que podrían realizarse a partir del trabajo realizado:

- Incrementar el número de muestras. No existe una cantidad óptima, pero el clasificador será mejor cuantas más imágenes se dispongan para su entrenamiento.
- Implementar el clasificador en un sistema que trabaje sobre el vehículo. Para ello habría que mejorar el tiempo de ejecución que en este momento es elevado como para poder usarse en tiempo real y así obtener información del entorno.
- Incrementar el número de elementos a clasificar, siguiendo el mismo método utilizado en este proyecto se puede llevar a cabo el entrenamiento del clasificador, para detectar otros objetos como peatones, coches, señales o cualquier objeto que se pueda encontrar en la carretera.
- Aplicar este sistema en un vehículo como un asistente de ayuda a la conducción (ADAS), para que de esta manera avise al conductor de que tiene un objeto en la zona de peligro indicándole de que objeto se trata peatón, motocicleta, coche, etc. y a la distancia a la que se encuentra del vehículo.

BIBLIOGRAFÍA

- [1] T. Huang, "Computer Vision: Evolution And Promise," *19th Cern Sch. Comput.*, pp. 21–25, 1996.
- [2] H. D. A, "'Impossible objects as nonsense sentences' in Machine Intelligence," 1971, pp. 295–323.
- [3] M. B. Clowes, "On seeing things," *Artif. Intell.*, vol. 2, no. 1, pp. 79–116, 1971.
- [4] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. 1982.
- [5] M. Kass, A. Witkin, and D. Terzopoulos, "Active Contour Models," *Int. J. Comput. Vis.*, vol. 331, pp. 321–331, 1988.
- [6] B. Leibe, A. Leonardis, and B. Schiele, "Toward Category-Level Object Recognition," *Towar. Categ. Object Recognit.*, vol. 4170, pp. 508–524, 2006.
- [7] L. Li, "Time-of-Flight Camera – An Introduction," no. January, 2014.
- [8] D. Jimenez, D. Pizarro, and M. Mazo, "Single frame correction of motion artifacts in PMD-based time of flight cameras," *Image Vis. Comput.*, vol. 32, no. 12, pp. 1127–1143, 2014.
- [9] D. Piatti and F. Rinaudo, "SR-4000 and CamCube3.0 Time of Flight (ToF) cameras: Tests and comparison," *Remote Sens.*, vol. 4, no. 4, pp. 1069–1089, 2012.
- [10] R. Gonzalez and R. Woods, *Digital image processing*. 2002.
- [11] R. C. González and P. Wintz, "Filtrado de imágenes," vol. 34, pp. 89–269, 1996.
- [12] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.
- [13] N. Dalai, B. Triggs, I. Rhone-Alps, and F. Montbonnot, "Histograms of oriented gradients for human detection," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2005. CVPR 2005*, vol. 1, pp. 886–893, 2005.
- [14] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," *Proceedings. Int. Conf. Image Process.*, vol. 1, p. I-900-I-903.
- [15] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [16] S. Ma and T. Du, "Improved adaboost face detection," in *2010 International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2010*, 2010, vol. 2, pp. 434–437.
- [17] I. Moreno, "SISTEMA AUTOMATIZADO BASADO EN VISIÓN POR COMPUTADOR DE DETECCIÓN DE DEFECTOS EN SISTEMAS MARINOS," 2017.
- [18] "Kinect: V2." [Online]. Available: <https://developer.microsoft.com/es-es/windows/kinect>. [Accessed: 11-Oct-2017].
- [19] "SR4000 | Cámaras 3D | Cámaras | Sistemas de visión Artificial para INDUSTRIA |

- INFAIMON.” [Online]. Available: <http://www.infaimon.com/es/sr4000>. [Accessed: 11-Oct-2017].
- [20] “ToF 3D Cameras - Bluetechnix.” [Online]. Available: <https://www.bluetechnix.com/en/products/depthsensing/product/sentis3d-m420/>. [Accessed: 11-Oct-2017].
- [21] P. Gil, T. Kisler, G. J. García, C. A. Jara, and J. A. Corrales, “Calibración de cámaras de tiempo de vuelo: Ajuste adaptativo del tiempo de integración y análisis de la frecuencia de modulación,” *RIAI - Rev. Iberoam. Autom. e Inform. Ind.*, vol. 10, no. 4, pp. 453–464, 2013.
- [22] “Changing Modulation Frequency.” [Online]. Available: http://datasheets.bluetechnix.at/goto/TIM/TIM-UP-19k-S3-Spartan6/TIM-uP-19k-S3-Spartan6_SUM_V9.pdf. [Accessed: 13-Oct-2017].
- [23] “MathWorks - Makers of MATLAB and Simulink.” [Online]. Available: <https://es.mathworks.com/>. [Accessed: 11-Dec-2017].
- [24] “Suaoki - 300W Inversor de Corriente, DC 12V, 220V-240V AC salida, Dual Puertos USB 5V/2.1A (Diseño de Aluminio, con Pinzas de Coche Batería y Mechero de Coche) Negro: Amazon.es: Coche y moto.” [Online]. Available: https://www.amazon.es/gp/product/B01LYGF7WV/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1. [Accessed: 11-Dec-2017].

ANEXO I



Programa de captura de datos SENTIS3D-M420

```
close all;
clear all;

%Se define el numero de images que se captura
N=300;

% Se obtiene la estructura de configuracion
[status, configStruct] = BTAinitConfig;
if status ~= 0
    error('BTAinitConfig failed. See "status" for
error description');
end

% Se configura la estructura
configStruct.frameMode = 2;
configStruct.serialNumber = 0;

% Inicia el dispositivo
[status, deviceHandle] = BTAopen(configStruct);
if status ~= 0
    error('BTAopen failed. See "status" for error
description');
end

% Se establece el modo DistAmp
status = BTAsetFrameMode(deviceHandle, 2);
if status ~= 0
    error('BTAsetFrameMode failed. See "status" for
error description');
end

% Se ajustan los parametros
%Tiempo de integración
integrationTime=1000;
status = BTAsetIntegrationTime (deviceHandle,
integrationTime);

%offset
globalOffset=-2874;
status = BTAsetGlobalOffset (deviceHandle,
globalOffset);

% FrameRate
status = BTAsetFrameRate(deviceHandle, 20);

%Se espera un tiempo para que los cambios hagan efecto
```

```
pause(2);

for i = 1 : N
    [status, frameHandle, frameCounter, timeStamp] =
    BTAGetFrame(deviceHandle, 500);
    if status ~= 0
        statusClose = BTAClose(deviceHandle);
        error('BTAGetFrame failed. See "status" for
error description');
    end
    %Se obtiene la distancia
    [status, distData, integrationTime,
modulationFrequency, unit] =
    BTAGetDistances(frameHandle);
    if status == 0
        FrameBufferDist12(:, :, i) = distData;
    end
    %Se obtiene la amplitud
    [status, ampData, integrationTime,
modulationFrequency, unit] =
    BTAGetAmplitudes(frameHandle);
    if status == 0
        FrameBufferAmp12(:, :, i) = ampData;
    end

    %free frame
    [status] = BTAfreeFrame(frameHandle);
    if status ~= 0
        error('BTAfreeFrame failed. See "status" for
error description');
    end
end

%Se cierra el dispositivo
status = BTAClose(deviceHandle);
if status ~= 0
    error('BTAClose failed. See "status" for error
description');
end

% Se guardan las matrices de amplitud y distancia
save('FrameBufferDist', 'FrameBufferDist');
save('FrameBufferAmp', 'FrameBufferAmp');
clear all;
```

Programa de reproducción de datos SENTIS3D-M420

```

clc
clear all
%Numero de imagenes que se reproducen
N=600;
%Tiempo de reproduccion de cada imagen
frameTime = 0.05;

%Se cargan los datos
load('FrameBufferDist');
load('FrameBufferAmp');
%%
FrameBufferDist=FrameBufferDist10;
FrameBufferAmp=FrameBufferAmp10;

%create window for amplitudes
%Se crea una ventana para los datos de amplitud
figure('name','Amplitude data');
set(gcf,'NumberTitle','off')
%Con un rango de 0 a 6000
clims = [0 6000];
hAmp=imagesc(FrameBufferAmp(:,:,1), clims);
colorbar;
colormap('GRAY');

%Se crea una ventana para los datos de distancia
figure('name','Distance data');
set(gcf,'NumberTitle','off')
%Con un rango de 0 to 3
clims = [0 3];
hDist = imagesc(squeeze(FrameBufferDist(:,:,1)),
clims);
colorbar;
cm=flipud(jet);
colormap(cm);

for i = 1 : N
    %Reproduce la matriz de distancias

FrameBufferDist(:,:,i)=imrotate(FrameBufferDist(:,:,i)
,180);

set(hDist,'CData',squeeze(FrameBufferDist(:,:,i)));
refreshdata(hDist);
pause(frameTime);

```

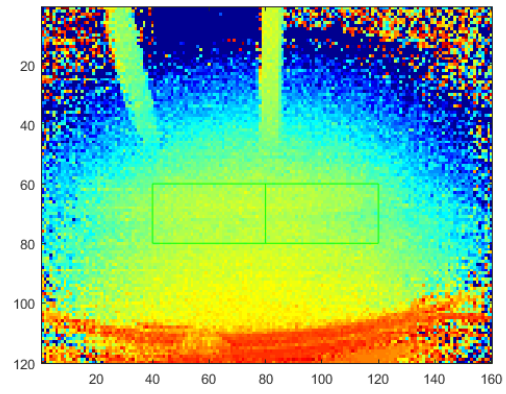
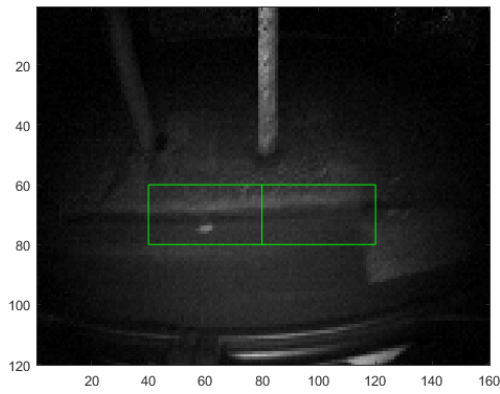
```
%Reproduce la matriz de amplitud

FrameBufferAmp(:, :, i) = imrotate(FrameBufferAmp(:, :, i), 180);
set(hAmp, 'CData', squeeze(FrameBufferAmp(:, :, i)));
refreshdata(hAmp);
pause(frameTime);
end

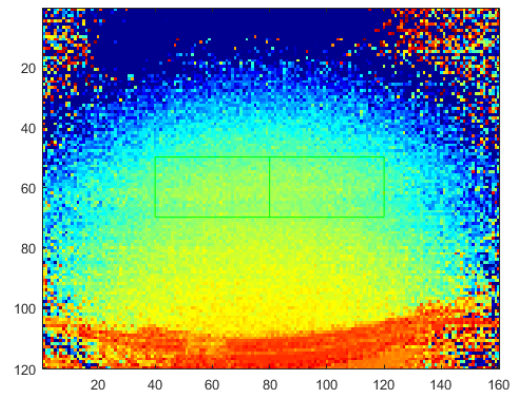
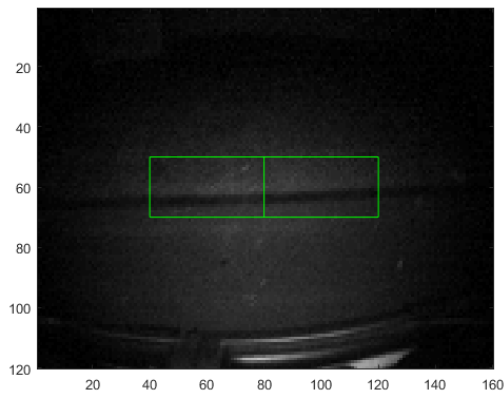
clear all;
close all;
```


ANEXO II

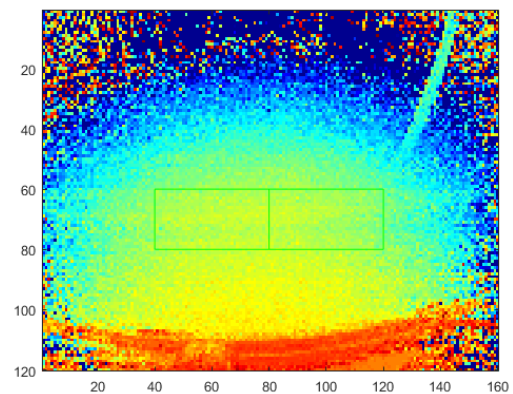
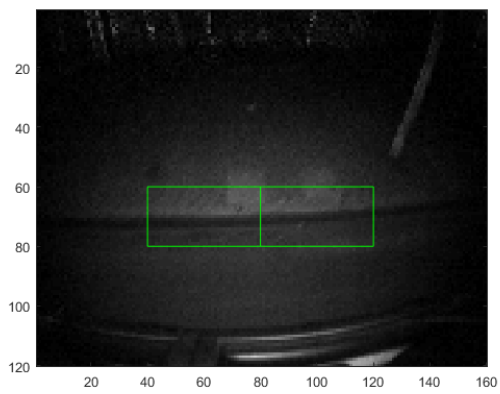
PRUEBAS DEL CLASIFICADOR



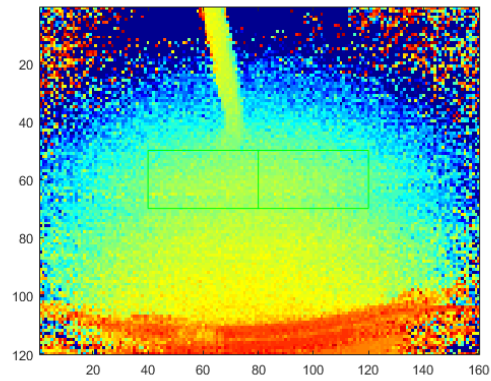
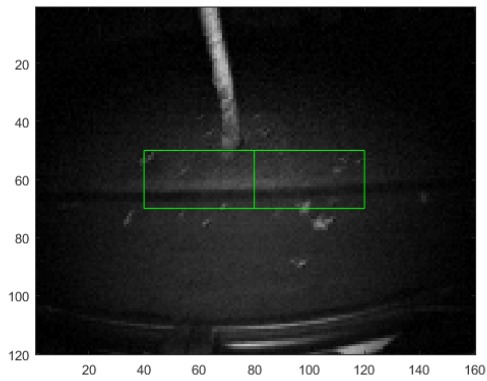
Prueba 5-El bordillo se encuentra a 1212.3mm



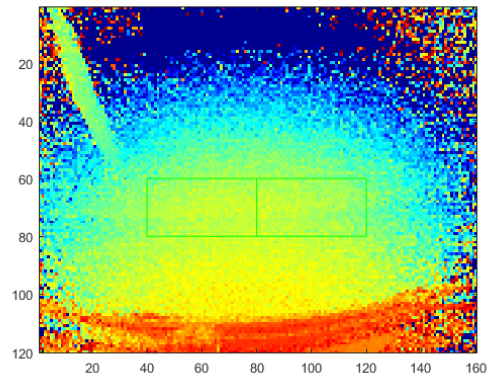
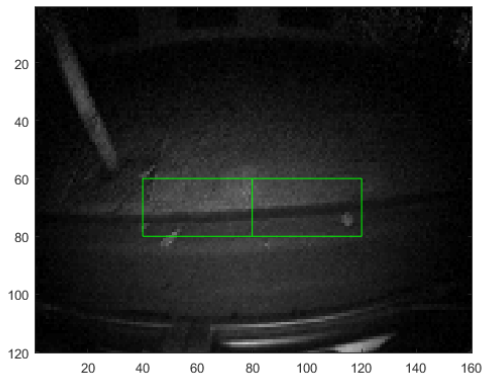
Prueba 9-El bordillo se encuentra a 1322.9mm



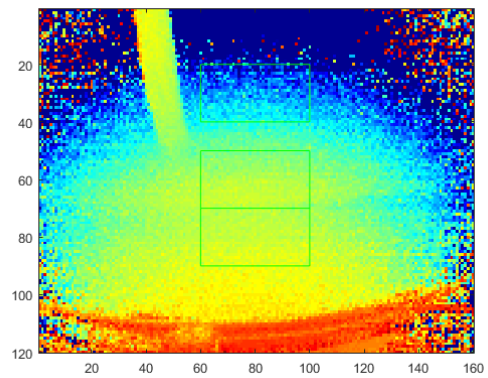
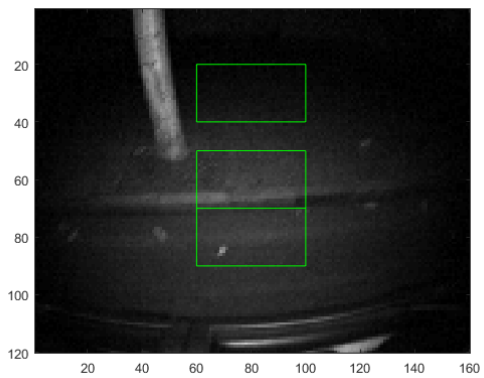
Prueba 12-El bordillo se encuentra a 1079.3 mm



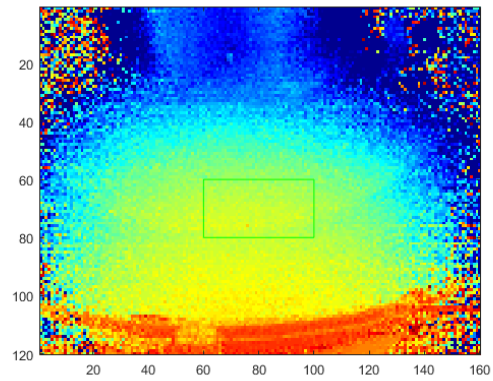
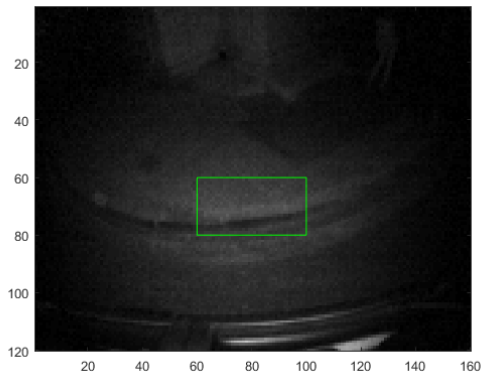
Prueba 13-El bordillo se encuentra a 1277.8 mm



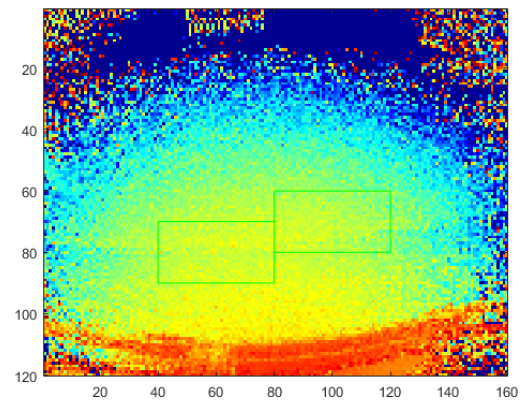
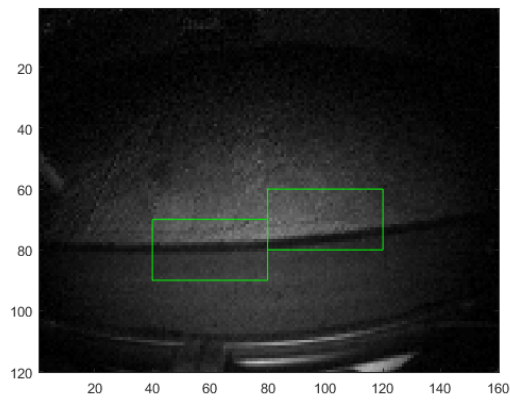
Prueba 15-El bordillo se encuentra a 1085.7 mm



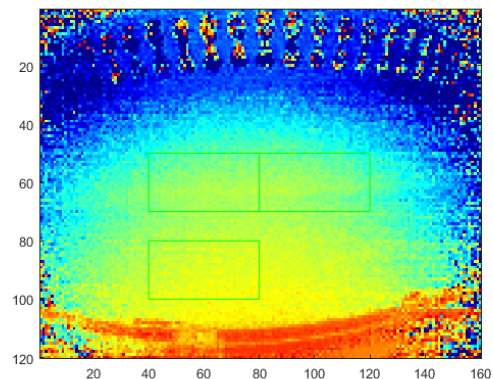
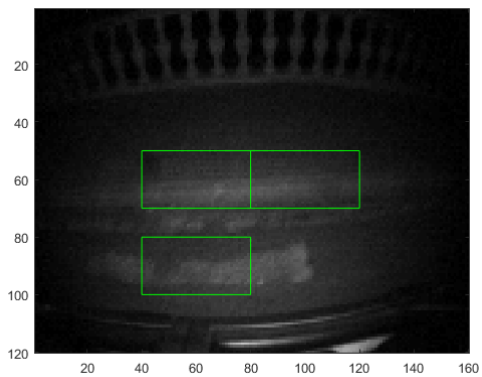
Prueba 18-Error en la clasificación



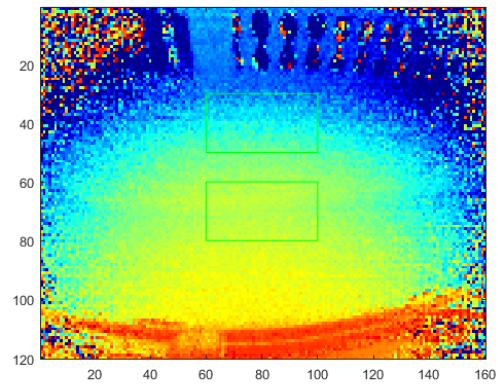
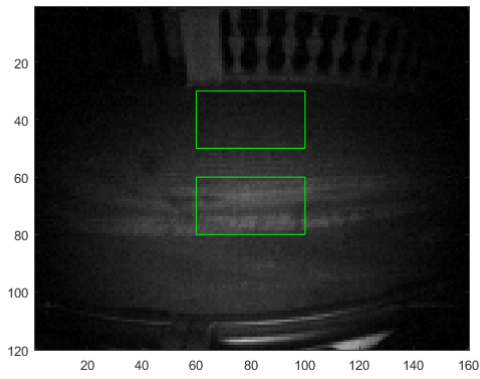
Prueba 22-El bordillo se encuentra a 1060.4 mm



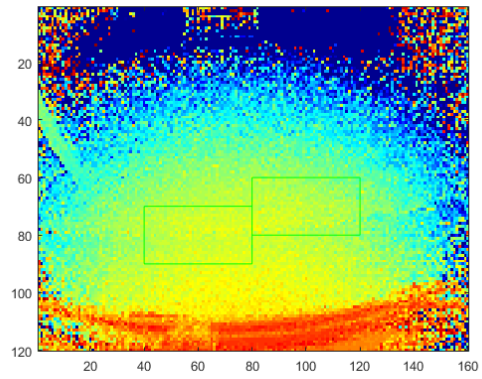
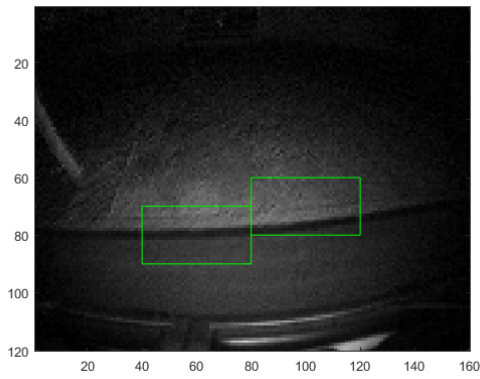
Prueba 25-El bordillo se encuentra a 1022.3 mm



Prueba 27- Error en la clasificación



Prueba 28- Error en la clasificación



Prueba 29- El bordillo se encuentra a 928.11