

Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE
TELECOMUNICACIÓN



Trabajo Fin de Grado

Estudio del sensor OPT8241 de tiempo de vuelo para la
localización de robots móviles en espacios inteligentes

ESCUELA POLITECNICA
SUPERIOR

Autor: Alberto Mata Caldera

Tutor: José Sáez Landete

2018

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Estudio del sensor OPT8241 de tiempo de vuelo para la localización
de robots móviles en espacios inteligentes

Autor: Alberto Mata Caldera

Tutor: José Sáez Landete

TRIBUNAL:

Presidente: Blanco Velasco, Manuel

Vocal 1º: Álvarez Pérez, José Luis

Vocal 2º: Sáez Landete, José

FECHA:

Agradecimientos

“Antes de convencer al intelecto, es imprescindible tocar y predisponer el corazón”

Blaise Pascal

Este proyecto cierra una etapa en mi vida, llena de obstáculos, pero también de buenos momentos que recordar.

En primer lugar, me gustaría dar las gracias a mi padre Rafael, mi madre Toñi y mi hermana Laura, por todo su esfuerzo y por creer en mí cuando nadie lo hacía, todo lo que soy es por vosotros.

También quiero dar las gracias a mi pareja Sabrina, por apoyarme y entenderme siempre, incluso en los momentos más difíciles dónde no se lo he puesto fácil.

Me llevo a unos amigos para toda la vida, la Piñita, los cuales han hecho el camino mucho más sencillo a su lado, ayudándonos en los momentos difíciles y disfrutando en los buenos. Entre ellos, quiero dar las gracias en especial a Raúl, quien me ha acompañado desde el primer día de carrera hasta el último, creciendo juntos y aprendiendo el uno del otro, entramos como desconocidos, pero salimos siendo como hermanos.

No puedo olvidar a mis amigos de siempre, la Chavalería, los que siempre me han apoyado y por muy agobiado que estuviese con la Universidad, siempre era buen momento para quedar, reírse y olvidarse de todo por unos momentos.

Por último, quiero dar las gracias a mis cuatro abuelos, en especial a mi abuelo Antonio por enseñarme a pensar y luchar por cumplir mis objetivos por mí mismo.

Índice general

Índice de figuras.....	9
Índice de tablas.....	11
Resumen.....	13
Resumen extendido	17
1. Capítulo 1: Introducción.....	21
1.1. Objetivos	21
1.2. Estructura del documento	22
2. Capítulo 2: Fundamentos teóricos	25
2.1. Trilateración	25
2.2. Sensores de profundidad.....	31
2.2.1. Cámaras de color o RGB – Red Green Blue	31
2.2.2. Sensores de profundidad basados en luz estructurada.....	31
2.2.3. Sensores de profundidad basados en tiempo de vuelo	33
2.2.3.1. Funcionamiento mediante modulación continua (CWM)	34
2.2.3.2. Funcionamiento mediante modulación pulsada (PM)	36
2.2.3.3. Fuentes de error en cámaras TOF	38
3. Capítulo 3: Desarrollo práctico.....	41
3.1. Sensor OPT8241 Texas Instrument	41
3.2. Evaluación de los perfiles de calibración y calibraciones	48
3.2.1. Ensayo con los perfiles del fabricante.....	49
3.2.2. Calibración de la lente	51
3.2.3. Calibración de fase común.....	54
3.3. Caracterización del perfil de medidas FaseCal	56
3.3.1. Distance PCD y Algoritmo de la ventana	58

3.3.2.	Medidas en rayo directo	62
3.3.3.	Mediciones en ángulo	64
3.4.	Espejos Corner Cube	69
3.5.	Conclusiones	73
4.	Capítulo 4: Posicionamiento	75
4.1.	Detección de tres marcas	75
4.2.	Celda de posicionamiento	78
5.	Conclusiones y líneas futuras	83
5.1.	Conclusiones	83
5.2.	Líneas futuras de desarrollo	85
5.2.1.	Obtención de medidas en tiempo real	85
5.2.2.	Detección de marcas	85
5.2.3.	Modificación de la lente	86
	Bibliografía	89
	Anexo – Código programa Distance PCD	93

Índice de figuras

Figura 1 Lugar geométrico 1	26
Figura 2 Lugar geométrico 2	26
Figura 3 Lugar geométrico 3	27
Figura 4 Lugar geométrico 4	27
Figura 5 Trilateración en 3D [1].....	28
Figura 6 Representación bidimensional de la trilateración esférica [1]	29
Figura 7 Proyección puntual [6]	31
Figura 8 Proyección planar [6]	32
Figura 9 Proyección de un patrón [6]	32
Figura 10 Proyección de un patrón codificado [6].....	32
Figura 11 Principio de modulación continua (CWM) [24]	34
Figura 12 Distribución de cargas CWM [19]	35
Figura 13 Principio de modulación pulsada (PM) [24].....	36
Figura 14 Distribución de cargas PM [19].....	37
Figura 15 Efecto multicamino sobre equinas [22]	39
Figura 16 Diagrama de bloques OPT8241 + OPT9221 [17]	42
Figura 17 Componentes Placa de Evaluación [9]	42
Figura 18 Diagrama de bloques completo [12].....	44
Figura 19 Pantalla principal Voxel Viewer [12]	47
Figura 20 Escenario para medidas en rayo directo	49
Figura 21 Medidas en rayo directo con perfiles del fabricante	49
Figura 22 Modelo de cámara estenopeica [13]	51
Figura 23 Tablero de calibración [13]	51
Figura 24 Imagen de calibración de lente correcta [13].....	52
Figura 25 Ejemplo de una mala imagen de calibración	52
Figura 26 Comparación del perfil "Long Lente"	53
Figura 27 Configuración de la calibración de fase común [13]	54
Figura 28 Menú de configuración de la fase común [13]	55
Figura 29 Comparación del perfil "FaseCal"	55
Figura 30 Análisis de precisión del perfil FaseCal	56
Figura 31 Evolución del error en función de la distancia	57
Figura 32 Interfaz Distance PCD	58

Figura 33 Vista lateral del escenario de medida para estudiar la variación de distancia entre puntos cercanos de la marca y la cámara de tiempo de vuelo	60
Figura 34 Caracterización en rayo directo	62
Figura 35 Error medio en rayo directo	63
Figura 36 Desviación media en función de la distancia.....	63
Figura 37 Escenario de medidas angulares.....	64
Figura 38 Caracterización de medidas angulares.....	65
Figura 39 Caracterización angular temporal.....	66
Figura 40 Evolución del error en caracterización temporal	67
Figura 41 Evolución de la desviación típica en caracterización temporal.....	67
Figura 42 Funcionamiento espejo Corner Cube [14].....	69
Figura 43 Corner Cube	70
Figura 44 Diagrama de fase con dos Corner Cube	70
Figura 45 Espejo Corner Cube con polarizador a 90°	71
Figura 46 Diagrama de fase con Corner Cube con polarizador	72
Figura 47 Semi-rango de visión con sensor rotado.....	75
Figura 48 Escenario con tres marcas	76
Figura 49 Localización gráfica del robot.....	77
Figura 50 Definición de área mínima	79
Figura 51 Estructura teselar.....	79
Figura 52 Cálculo gráfico del triángulo de posicionamiento	80
Figura 53 Celda de posicionamiento	80

Índice de tablas

Tabla 1 Principales especificaciones [11].....	43
Tabla 2 Especificaciones del láser [11]	43
Tabla 3 Leyenda de colores de los LEDs [11].....	43
Tabla 4 Especificación de componentes de la placa de evaluación [12].....	44
Tabla 5 Características de perfiles por defecto [12]	45
Tabla 6 Distancias reales de prueba de medias angulares.....	65
Tabla 7 Medidas para caso a 220 cm en riel secundario	67
Tabla 8 Distancias de escenario con tres marcas	77

Resumen

Este proyecto aspira a detallar el funcionamiento de las cámaras de tiempo de vuelo (ToF), concretamente un modelo específico de ellas, el OPT8241 del fabricante Texas Instruments.

Se pretende caracterizar este sensor en los aspectos más influyentes para su implementación en un sistema que permita localizar un robot dotado de este sensor en un espacio inteligente.

Se estudiará el número de marcas que debe poseer dicho espacio para poder localizar el robot en todo momento, las características de las marcas y la correcta situación de ellas, definiendo la estructura de las celdas de posicionamiento y la escalabilidad del sistema.

Palabras clave: Cámaras de tiempo de vuelo (ToF), OPT8241, Point Data Cloud (PCD), localización, celda de posicionamiento.

Abstract

This project aims to detail the operation of the time of flight (ToF) cameras, specifically a specific model of them, the OPT8241 from the manufacturer Texas Instruments.

The aim is to characterize this sensor in the most influential aspects for its implementation in a system that allows locating a robot equipped with this sensor in an intelligent space.

We will determinate the number of marks that this space must have in order to place the robot at all times, the characteristics of the marks and their correct location, defining the structure of the positioning cells and the scalability of the system.

Keywords: Time-of-flight (ToF), OPT8241, Point Data Cloud (PCD), location, positioning cell.

Resumen extendido

Este proyecto pretende abordar el reto la localización de móviles en interiores a partir de la tecnología infrarroja, concretamente a partir de cámaras de tiempo de vuelo (ToF).

Se parte de un análisis del principio teórico sobre localización que se utilizará para obtener la posición del robot a partir de la distancia a la que se encuentran unas marcas posicionadas en la escena, la trilateración.

Se estudian las diferentes alternativas de cámaras para fines de posicionamiento existentes en el mercado, profundizando en los distintos tipos de cámaras ToF según su funcionamiento, diferenciando entre las cámaras de modulación continua (CWM) y la de modulación pulsada (PM).

Se mencionan las diferentes fuentes de error que afectan a estos sistemas, profundizando en aquellas que más repercuten en nuestro sistema: los errores debidos a “flying píxels” y errores debidos a “multipath” o “multicamino”.

Una vez estudiados los fundamentos teóricos que soportan este proyecto, se procede a entrar en detalle acerca la cámara concreta con el que se trabajará, el modelo OPT8241 del fabricante Texas Instruments. Se estudian sus características y su funcionamiento básico, así como el entorno de desarrollo que el fabricante proporciona, tratando las ventajas y desventajas del sistema.

Se realiza un estudio de los diferentes perfiles de calibración que el sistema ofrece, determinando cuál de ellos proporciona unas mejores prestaciones para nuestra aplicación concreta, a partir de una serie de ensayos para determinar la precisión cada una de ellas.

Sobre el perfil de calibración del fabricante escogido, se realiza la calibración de algunos factores con la intención de mejorar las prestaciones de este. Para ello, se calibra la lente y la fase común del sensor, a partir de indicaciones proporcionadas por el fabricante. Se obtienen los resultados de estas calibraciones y se determina el perfil final que mejores resultados proporciona en nuestro sistema.

Con el perfil definitivo, el objetivo ahora es caracterizar la precisión que la cámara ofrece bajo cualquier circunstancia. Se realizan ensayos de medidas en rayo directo y angular para ver la potencialidad del sistema antes sus dos mayores limitaciones: la distancia máxima de medida con precisión y el rango de visión.

Se determina la manera óptima de obtener las distancias con la cámara, para lo que fue necesario la realización de un software de apoyo que complementa las medidas que aporta el software del fabricante. Este programa obtiene la distancia de un punto de referencia o “marca” a partir de ficheros con el formato estandarizado PCD (Point Data Cloud), los cuales se obtienen mediante el software del fabricante. El programa desarrollado tiene en cuenta la inestabilidad temporal y espacial de los píxeles ya que toma la distancia de varios puntos sobre la marca deseada en varios instantes de tiempo adyacentes.

También se caracteriza la forma de las marcas que el sensor debe detectar en el escenario, donde se estudia el comportamiento de un tipo especial de espejos como marcas, los Corner Cube.

En la última parte del proyecto, se pretende simular un escenario de medida real para validar el sensor como elemento principal de la localización. Para ello se realiza una detección a tres marcas situadas dentro del rango de visión del sensor y se calcula de manera gráfica la posición en la cual el sensor se encuentra a partir de las tres distancias a las marcas obtenidas.

Por último, se caracteriza las dimensiones de la celda de posicionamiento del sistema. Se determina su tamaño y la posición de las marcas en su interior, así como la escalabilidad del problema. Es decir el modo de diseñar un espacio inteligente donde se puede localizar un robot, mediante la agregación de diferentes celdas de posicionamiento.

1. Capítulo 1: Introducción

La localización precisa en interiores es un reto que se aborda desde diferentes aproximaciones y mediante diferentes tecnologías, los sistemas sensoriales AV (audio + visión), los sistemas sensoriales US&RF (ultrasonidos y radiofrecuencia) y los sistemas sensoriales IR (infrarrojos).

En este proyecto pretendemos abordar la problemática desde el ámbito de la tecnología infrarroja, concretamente realizando un estudio de los sensores de tiempo de vuelo o ToF (Time-of-Flight).

Dicho tipo de sensores nos proporcionan información acerca de la distancia a la que se encuentran los objetos del sensor, proporcionando una “imagen” en la cual podremos ver la profundidad punto a punto de la escena en cada uno de sus píxeles, pudiendo así obtener una medida real de la distancia a la que se encuentran cada uno de los objetos que nos rodean.

Se pretende estudiar la implementación del modelo concreto OPT8241 del fabricante Texas Instruments en un sistema que permita localizar un robot dentro de un espacio inteligente. Para ello, se realizará una caracterización exhaustiva del sensor comprobando fundamentalmente la precisión que se obtiene en las medidas en diversos escenarios.

Este proyecto pretende ser el punto de partida del sistema completo de localización, aquí nos centraremos en validar este sensor para dicho propósito, estudiar la precisión que proporciona y diseñar un procedimiento para obtener dichas distancias con el sensor. Asimismo, se estudiará las características que deben tener las marcas del sistema de localización, las características de la celda de posicionamiento y su escalabilidad.

1.1. Objetivos

El objetivo principal de este estudio consiste en conocer en detalle las prestaciones que puede tener la implementación de los sensores ToF en lugar de otros mecanismos para obtener la posición de móviles en interiores, así como las ventajas y las limitaciones de estos dispositivos.

Se pretende poder, a partir de una “imagen” captada por este sensor, reconocer en ella ciertas marcas o puntos de referencia cuya posición es conocida y así, poder geolocalizar el móvil en este espacio inteligente a partir de las distancias a la que se encuentran dichas marcas.

Se estudiará el proceso de obtención de las distancias con el sensor y la forma de automatizar el proceso, pretendiendo conseguir el sistema más eficiente posible.

Se pretende caracterizar las marcas que se utilizarán en dicho espacio, así como el número de ellas necesario para implementarlo.

Por último, se definirá las dimensiones de la celda de posicionamiento, su forma y la manera de interconectar varias celdas para poder cubrir el máximo área posible con el menor número de marcas, de manera que el robot se encuentre siempre localizado en el entorno.

1.2. Estructura del documento

- Desarrollo de los fundamentos teóricos que soportan los objetivos que se desean conseguir en el proyecto. Este capítulo pretende introducir los principios teóricos de las cámaras de tiempo de vuelo en el contexto de la localización, ya que será el tipo de dispositivo que emplearemos. Se plantea el concepto de trilateración, mediante el cual una vez obtenidas las distancias con la cámara ToF se hallará la posición del móvil.
- Desarrollo práctico donde se estudiará en concreto del modelo específico de cámara que se utilizará en el proyecto y sus especificaciones. Se estudia los diferentes perfiles que posee el sensor con la configuración de fábrica y cómo se pueden calibrar ciertos aspectos para conseguir mejorar la precisión. En este capítulo se trata también de optimizar el proceso de obtención de las distancias con el sensor, para ello se realizó un programa capaz de obtener de manera rápida y eficaz diversas medidas del entorno reconocido por el sensor. Una vez configurado el sensor y las herramientas necesarias, se caracteriza la precisión que se obtiene teniendo en cuenta la configuración específica de la aplicación. Por último, se estudia una alternativa a las marcas pasivas utilizadas en las mediciones anteriores, los Corner Cube, extrayendo diferentes conclusiones sobre las ventajas e inconvenientes que aportan.
- En el último capítulo se realiza un ensayo de posicionamiento real a partir de la medición de las distancias a tres marcas predefinidas, se define las características de la celda mínima de posicionamiento del sistema, así como la escalabilidad de estas.

2. Capítulo 2: Fundamentos teóricos

El objetivo de este proyecto consiste en la localización de robots móviles en espacios inteligentes a partir de la información de profundidad que nos proporciona el sensor de tiempo de vuelo (ToF) OPT8241 de Texas Instruments. En primer lugar, planteamos una caracterización del sensor, y para ello es necesario establecer algunos conceptos teóricos que forman las bases de este proyecto.

En este capítulo hablaremos de los principios teóricos en cuanto a localización, tratando el concepto de trilateración, ya que este será el principal soporte teórico para la localización.

Estudiaremos las diferentes alternativas en cuanto a los tipos de sensores o cámaras orientadas a localización, centrándonos en las cámaras ToF. Se explicarán cómo operan estos dispositivos, su principio de funcionamiento y los diferentes tipos de modulación que pueden implementar, así como las ventajas e inconvenientes que conlleva su uso frente a otras alternativas en el mercado.

2.1. Trilateración

La técnica de trilateración es la utilizada en la tecnología GPS, se basa en la medición de distancias entre el dispositivo y un conjunto de satélites, a diferencia de la triangulación, tecnología con la que frecuentemente se confunde y en la cual la posición se obtiene a partir de ángulos

El tipo de señal y las características de la tecnología GPS hacen que su uso quede restringido a exteriores. Por esta razón, se han desarrollado multitud de técnicas de posicionamiento en interiores, sin que predomine ninguna tecnología sobre las demás. En este caso, se plantea el principio de funcionamiento del GPS aplicado a las medidas de distancia obtenidas mediante una cámara de tiempo de vuelo.

La obtención de buenos resultados en la localización depende en gran medida de la correcta ubicación de las balizas o marcas de posicionamiento, así como del número adecuado de estas. El objetivo consiste en establecer un buen compromiso entre una buena cobertura en el entorno de trabajo, unos costes reducidos y la máxima precisión posible en la estimación de la localización del objeto o móvil.

En nuestra aplicación disponemos de ciertas marcas en el techo de cada celda de localización, donde conoceremos perfectamente las coordenadas en las que se encuentran, pero se desconoce la posición del robot con la cámara ToF que deseamos posicionar. El robot se desliza sobre un plano paralelo al plano del techo, dando lugar a un problema de trilateración tridimensional.

Podemos encontrar dos tipos de trilateración, trilateración esférica (TE) y trilateración hiperbólica (TH). En este estudio nos vamos a centrar en la TE debido a que los principios básicos de funcionamiento son similares, pero más sencillos de comprender gráficamente. Una vez realizado el estudio de TE se podría extrapolar a TH sustituyendo las esferas y ecuaciones de las esferas por hipérbolas con sus respectivas ecuaciones.

En nuestro sistema real se utilizará TE debido a que estas soluciones utilizan alrededor del 40% menos de marcas para posicionamiento que la TH.

En primera instancia, la cámara detectará una marca en el entorno y hallará la distancia a la que se encuentra. En este punto podríamos saber que el robot estará en el lugar geométrico de los puntos de la circunferencia de radio igual a la distancia medida y el centro la marca conocida (partiendo de un escenario inicial bidimensional) donde gráficamente obtendríamos un esquema como el de la Figura 1:

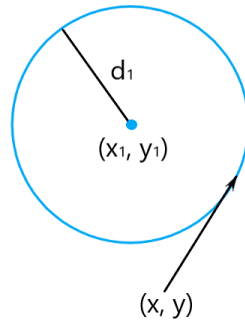


Figura 1 Lugar geométrico 1

En segundo lugar, el sensor detectará una segunda marca en el entorno y la distancia a la que se encuentra de manera similar a la primera marca. Conocida la distancia a la que se encuentra la segunda marca, se genera un segundo lugar geométrico de los puntos donde según la nueva marca podría situarse el robot. Cruzando ambos lugares geométricos podemos saber que ahora el robot debe encontrarse en la intersección de ambas circunferencias, lo cual puede ser un único punto, dos o bien ninguno si las circunferencias no se cortan.

En la situación en la cual las circunferencias se corten en dos puntos, gráficamente nos encontraríamos en la situación que podemos ver en la Figura 2:

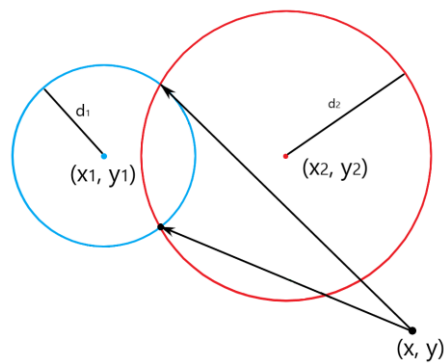


Figura 2 Lugar geométrico 2

En este caso, los puntos de corte de las circunferencias serían dos puntos candidatos a ser la ubicación del robot.

En el caso de tener circunferencias cuya intersección fuese un único punto, el problema estaría resuelto y ese punto sería directamente la posición buscada, como podemos ver en la Figura 3.

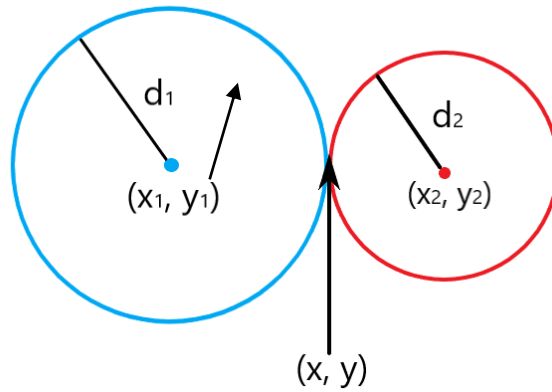


Figura 3 Lugar geométrico 3

Por el contrario, si las circunferencias no tuvieran ningún punto en común sería imposible determinar la posición del robot, al menos, con estas marcas.

Volviendo al caso general donde las circunferencias se cortan en dos puntos, localizar y medir la distancia a una tercera marca en el entorno solucionaría el problema en la mayoría de los casos.

Esto se debe a que ahora la intersección de las tres circunferencias revelará cuál de los dos candidatos anteriores contiene la verdadera localización del robot, gráficamente lo podemos ver en la Figura 4:

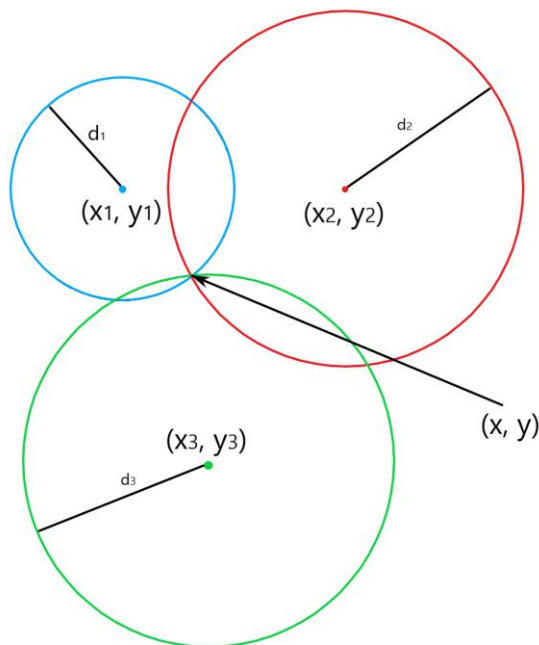


Figura 4 Lugar geométrico 4

Existen algunas limitaciones, como pudiera ser el caso donde disponemos de las tres marcas alineadas y el robot se sitúa en la línea que une los centros de las circunferencias, en este caso, el problema tendría dos posibles soluciones independientemente del número de marcas medidas.

Como conclusión, de este modo, podríamos posicionar nuestro robot en el espacio inteligente de forma similar a como los receptores GPS localizan la posición a partir de las señales que reciben de los satélites.

De forma analítica se podría calcular la posición deseada como: [3]

$$d_1 = \sqrt{(x - x_1)^2 - (y - y_1)^2}$$

$$d_2 = \sqrt{(x - x_2)^2 - (y - y_2)^2}$$

$$d_3 = \sqrt{(x - x_3)^2 - (y - y_3)^2}$$

Siendo d_1 , d_2 y d_3 las distancias medidas a cada una de las marcas del entorno, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , los puntos donde se encuentran cada una de las respectivas circunferencias y punto (x, y) el punto de corte de las tres circunferencias, la solución de nuestro problema y el lugar donde se encuentra el robot.

Trasladar esto a un escenario tridimensional conlleva una serie de problemáticas añadidas, aunque el procedimiento de resolución es el mismo.

En el caso tridimensional la distancia a la que se encuentra cada una de las marcas producirá un lugar geométrico de puntos posibles las cuales serían esferas en lugar de circunferencias.

En el caso general serán necesarias al menos tres distancias que proporcionen la intersección de tres esferas, gráficamente se puede ver la Figura 5:

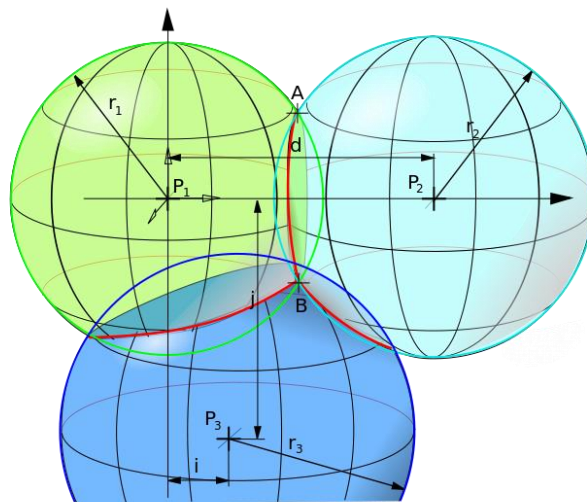


Figura 5 Trilateración en 3D [1]

Analíticamente se podría afrontar tomando las ecuaciones de las tres esferas e igualándolas. Esto conlleva asumir ciertas limitaciones en cuanto a la situación de sus centros: [1]

- Todas las esferas deben estar en el plano $z=0$
- Una esfera debe situarse en el origen
- Una esfera debe situarse en el eje X

Si no se cumplen estos requisitos siempre es posible trasladar cualquier conjunto de tres puntos para cumplir con estos límites, encontrar el punto solución, y luego invertir la traslación para encontrar el punto solución en el sistema de coordenadas original. [1]

El análisis partiría de las ecuaciones de las tres esferas: [1]

$$r_1^2 = x^2 + y^2 + z^2$$

$$r_2^2 = (x - d)^2 + y^2 + z^2$$

$$r_3^2 = (x - i)^2 + (y - j)^2 + z^2$$

Las cuales se representan bidimensionalmente en la Figura 6:

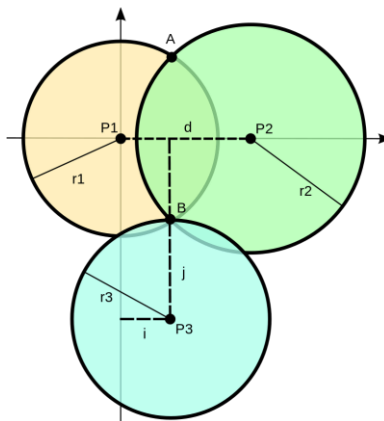


Figura 6 Representación bidimensional de la trilateración esférica [1]

En la Figura 6 se define de manera gráfica los parámetros d, j, i que se utilizan en las ecuaciones de cada una de las esferas y en las próximas ecuaciones.

Si restamos la segunda fórmula a la primera y despejamos x : [1]

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

Si se sustituye ' x ' en la fórmula de la primera esfera se hallamos la ecuación de un círculo, la cual es solución la intersección de las dos primeras esferas: [1]

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x - i)^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x$$

Una vez hallamos despejado las componentes ' x ' e ' y ' del punto solución buscado, se puede despejar la coordenada ' z ' de la ecuación de la primera esfera: [1]

$$z = \sqrt{r_1^2 - x^2 - y^2}$$

Conocidas las tres coordenadas del punto quedaría resuelto el problema, aunque debido a que la coordenada z se calcula de forma de raíz cuadrática, podría dar lugar a ninguna, una o dos soluciones. [1]

Visualmente la intersección de la primera y segunda esfera forma un círculo. Este círculo puede no tener puntos en común con la tercera esfera, lo cual quiere decir que el resultado de la raíz cuadrada del cálculo de la coordenada z es un número negativo, por lo que no existe solución real. Si, por el contrario, dicho círculo corta con la tercera circunferencia en un único punto, quiere decir que la solución a la raíz cuadrada es 0, por lo que $z=0$. Por último, si el círculo corta en dos puntos con la tercera esfera, esto quiere decir que z es igual a la raíz cuadrada (positiva y negativa) de un número positivo y por lo tanto existen dos puntos candidatos a ser la localización del robot. [1]

Cuando los datos presentan una componente de ruido, con frecuencia no se puede encontrar solución en el cálculo de la última coordenada. En este caso, bajo ciertas condiciones que permiten controlar el error, se puede aproximar a $z=0$. [1]

En el caso de obtener dos soluciones posibles suelen utilizarse algunas técnicas para descartar una de las dos posibles soluciones, la técnica que emplea la localización GPS es obviar el punto que se obtiene más alejado de la superficie terrestre, en nuestro caso sería tomar como solución válida el punto que se obtuviese dentro de nuestra celda de posicionamiento, pues tomando las separaciones adecuadas entre las marcas del entorno, se podría eliminar la posibilidad de obtener dos soluciones dentro de ella.

Un escenario real conlleva un estudio más detallado del error, ya que la introducción de cierta incertidumbre en las medidas de la distancia se traduce en intersecciones de circunferencias o esferas que no son puntuales, sino regiones geométricas donde se encuentra la solución. En un análisis más exhaustivo de localización mediante trilateración sería necesario asumir un modelo de error dentro de estas regiones para determinar una solución compromiso, esto podría ser un cálculo mediante mínimos cuadrados o simplemente asumir el centro de esta región de posibles localizaciones como solución de compromiso.

2.2. Sensores de profundidad

En primera instancia se pueden diferenciar tres tipos de tecnologías para la medida de la distancia a la que se encuentra un objeto, sean: sistemas basados en cámaras de color o RGB, sistemas basados en luz estructurada y los sensores basados en tiempo de vuelo o ToF.

2.2.1. Cámaras de color o RGB – Red Green Blue

Este tipo de cámaras son las más utilizadas actualmente para fotografía, proporcionan una información muy detallada del entorno, permitiendo múltiples funcionalidades como la detección de colores o el reconocimiento facial, pero su uso para medición de distancias y localización no está muy desarrollado ya que presentan ciertos inconvenientes.

Estas cámaras se basan en el modelo RGB donde se obtiene el color a partir de la síntesis aditiva de los tres colores de luz primarios.

Los principales problemas que presentan este tipo de cámaras son: [23]

- Necesidad de disponer de dos cámaras RGB para poder obtener la información espacial del entorno (imágenes 3D) mediante la construcción de una imagen a partir de la visión en estéreo.
- Complejidad en la algoritmia necesaria para el procesado de imagen y la necesidad de equipos potentes para que el sistema sea eficiente y rápido.
- Limitaciones lumínicas debido a que en situaciones de poca luminosidad los resultados obtenidos pueden ser deficientes o llegar a imposibilitar la medición.

2.2.2. Sensores de profundidad basados en luz estructurada

Los sensores de luz estructurada son similares a los sensores estereoscópicos clásicos, reemplazando una cámara por una fuente de luz, comúnmente un láser.

La forma más simple de estos sistemas se basa en la proyección de un solo punto sobre la escena, como se puede ver en la Figura 7, donde la correspondencia es directa, aunque es necesario escanear ambos ejes en busca de este punto. [6]

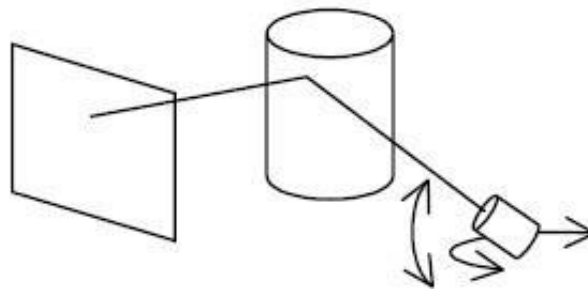


Figura 7 Proyección puntual [6]

En un segundo nivel, se proyecta un plano de luz que aparece como una recta sobre la superficie del dibujo, como se ve en la Figura 8, aunque continúa siendo necesario realizar una exploración a lo largo de un eje. [6]

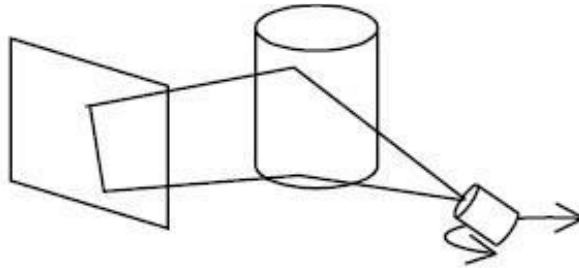


Figura 8 Proyección planar [6]

Por último, para evitar un escaneo mecánico que consume mucho tiempo, una solución consiste en proyectar un patrón de varias franjas, cuadrículas o varios puntos, es decir, un patrón bidimensional como el de la Figura 9. [6]

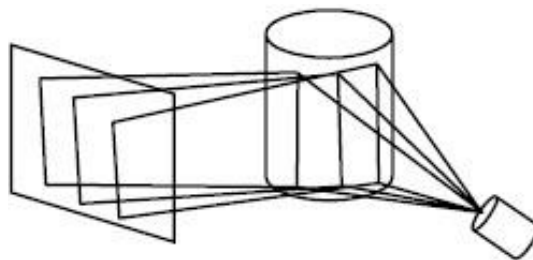


Figura 9 Proyección de un patrón [6]

Para este tipo de patrones se produce un problema de correspondencia que puede resolverse codificando el patrón, de forma que cada señal de luz se pueda distinguir fácilmente de las demás. Esto se consigue mediante proyecciones con formas irregulares como la de la Figura 10. [6]

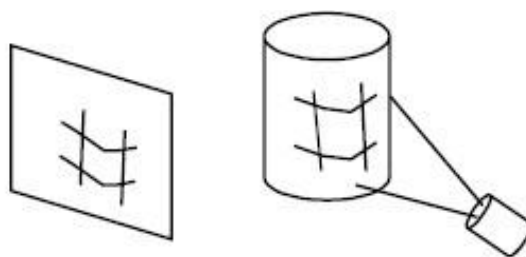


Figura 10 Proyección de un patrón codificado [6]

Comúnmente se utiliza luz estructurada invisible o imperceptible, cuya finalidad es evitar interferir con otras tareas de visión computacional donde visualizar el patrón emitido podría llevar a errores. [6]

Suele realizarse mediante un rayo infrarrojo para generar dichos patrones invisibles, los cuales pueden ser puntuales, de una sola línea o bidimensionales. La luz se proyecta normalmente en el infrarrojo cercano, es decir, de 800 nm a 2500 nm. La escena se observa con una cámara CCD debido a la sensibilidad espectral del CCD, de 300 nm a 1100 nm por lo que generalmente no es necesario emplear una cámara infrarroja. [6]

2.2.3. Sensores de profundidad basados en tiempo de vuelo

Los sensores ToF son relativamente novedosos, en los años noventa se propuso como un concepto académico, sin embargo, los componentes necesarios y las capacidades de procesamiento que lo hacen práctico comenzaron a estar disponibles en la última década.

En general, según el tipo de señal utilizada, existen dos modos de funcionamiento de los sensores TOF. El primero de ellos utiliza un haz de luz pulsado repetidamente con un ciclo de trabajo bajo y en el segundo se modula a través de una función sinusoidal continua o una señal cuadrada.

En el modo de modulación pulsada, la energía reflejada se muestra mediante la utilización de dos señales desfasadas, cuyo desfase se utilizará para computar la distancia al objeto. En el modo de modulación continua, el sensor realiza cuatro medidas con un desfase relativo de 90° para cada muestra adquirida y a partir de ellas se calcula el desfase entre la señal incidente y reflejada, obteniendo así la distancia al objeto. [18]

El resultado del procedimiento es un conjunto de puntos que muestra las áreas representadas en la imagen donde cada punto nos da una información de profundidad, de ahí el término “nube de puntos”, la cual se puede tratar mediante herramientas de análisis numérico como Matlab. [18]

Estos sensores disponen de ciertas ventajas y desventajas frente a los demás. En primera instancia sería necesario elegir entre la utilización de cámaras de imágenes tradicionales o ToF en función de los detalles de la aplicación. Si se desea la inspección de una imagen donde el objeto que se busca se conoce de antemano y el objetivo es su comparación en un entorno iluminado, la mejor opción sería la utilización de cámaras 2D. Por el contrario, si el escenario presenta una iluminación variante, escasa o bien movimientos rápidos donde se precisen algoritmos lo suficientemente rápidos para poder procesar las imágenes en tiempo real, la mejor opción será la utilización de cámaras TOF. [18]

Esto se debe a que las cámaras ToF no requieren iluminación externa y su funcionamiento es prácticamente invariante al nivel de iluminación ambiente. Asimismo, estas cámaras no requieren de procesamiento de las imágenes ya que obtienen directamente las distancias a las que se encuentran los objetos, lo que las hace idóneas para sistemas rápidos o de trabajo en tiempo real. En cambio, una de las principales desventajas es su baja resolución, lo cual avanza rápidamente en los últimos años donde se han llegado a obtener cámaras ToF con resoluciones cercanas a las cámaras tradicionales.

En cuanto a las imágenes en 3D a partir de dos cámaras convencionales, existen diversos problemas añadidos como cuestiones mecánicas y de montaje, además del complejo sistema de procesamiento de la imagen el cual debe contar con robustos algoritmos para obtener la correspondencia entre las imágenes tomadas por cada una de las cámaras. Esto requiere una gran variación en el color o en la escala de grises, lo que frecuentemente repercute en la precisión de profundidad obtenida. Por el contrario, los sistemas TOF tienen limitaciones muy inferiores respecto a cuestiones mecánicas, de iluminación y de contraste, además de obtener una medida directa, que no requiere de búsqueda de correspondencias. [18]

Las cámaras ToF presentan características idóneas para traducir gestos o movimientos en comandos, lo que está generando multitud de aplicaciones, como pueden ser su implementación en vehículos autónomos. [18]

Los sensores basados en TOF disponen principalmente de cinco bloques: [18]

- Fuente de luz
- Difusor de luz infrarroja emitida
- Sensor de imagen
- Gestión electrónica
- Unidad informática e interfaz

Para detectar cambios de fase entre la iluminación y la reflexión, la fuente de luz es pulsada (PM) o modulada por una fuente de onda continua (CWM), típicamente una onda sinusoidal o cuadrada. [18]

La modulación de onda cuadrada es más común porque puede realizarse fácilmente utilizando circuitos digitales. [18]

2.2.3.1. Funcionamiento mediante modulación continua (CWM)

Este tipo de sensores calculan la profundidad a la que se encuentra cada objeto a partir del cálculo de la diferencia de fase entre la señal emitida y recibida en el sensor.

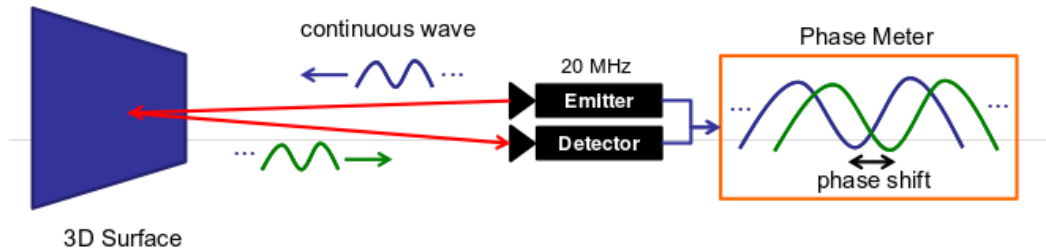


Figura 11 Principio de modulación continua (CWM) [24]

Típicamente, la iluminación se basa en un láser de estado sólido o un LED operando en el rango de infrarrojo cercano (~850 nm) el cual es el encargado de emitir la señal sobre la que se realizará la diferencia de fase, comúnmente se trata de una señal sinusoidal:

$$e(t) = A_0 \times \text{sen}(\omega t) = A_0 \times \text{sen}(2\pi f t)$$

Donde A_0 es la amplitud de la potencia emitida y f la frecuencia de modulación de la señal emitida. [19]

La señal que recibe el sensor es similar a la emitida, pero con un desplazamiento de fase debido al tiempo transcurrido desde que se emitió hasta que se refleja y llega de nuevo al sensor. Además, esta señal se ve influida por la reflectividad del objeto en el cual se refleja y la atenuación en el medio: [19]

$$r(t) = O + a_0 \times K \text{sen}(\omega(t + t_d)) = O + a_0 \times \text{sen}(\omega t + \beta)$$

donde:

- O es el offset añadido a la señal recibida debido a la iluminación de fondo y a la componente continua de la luz IR.
- a_0 es la amplitud de la señal recibida.
- K es la atenuación de la potencia de la señal recibida causada por la reflexión y los efectos de la óptica.
- t_a es el tiempo de ida y vuelta de la señal luminosa a un punto de la escena.
- $\beta = \omega t_d$ es el desplazamiento de fase a través del cual es posible obtener la información de distancia.
- ω es la pulsación de modulación, $\omega = 2\pi f$.

Esta función si se define $B_0 = a_0 \times k$ se puede simplificar como: [19]

$$r(t) = O + B_0 \times \text{sen}(\omega t + \beta)$$

Donde B_0 es la amplitud de la potencia recibida y β se puede definir como: [19]

$$\beta = \omega t_d = 2\pi f \frac{2d}{c} \Rightarrow d = \beta \frac{c}{4\pi f}$$

Una vez establecidos estos parámetros, si fijamos un desplazamiento de fase $\beta = 2\pi$ podremos calcular la **distancia máxima** posible a la que podemos medir profundidad sin ambigüedad: [19]

$$d_{max} = 2\pi \frac{c}{4\pi f} = \frac{c}{2f}$$

Por otro lado, podemos entender el procedimiento estudiando los pulsos y la distribución de cargas en el sensor, como podemos ver en la Figura 12:

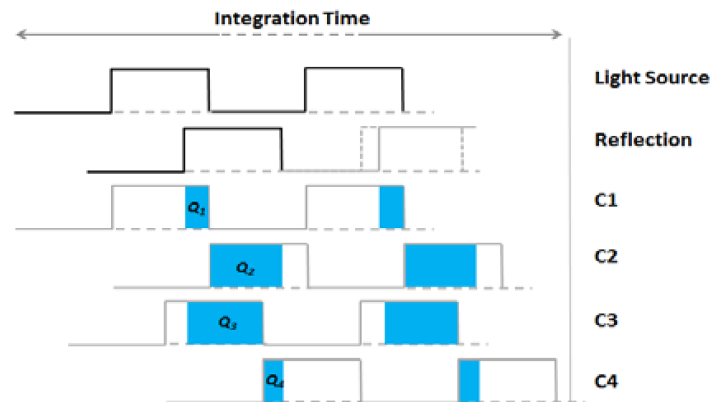


Figura 12 Distribución de cargas CWM [19]

El método de pulsos es relativamente sencillo, la fuente de luz se ilumina durante un breve período de tiempo (Δt) y la energía reflejada se muestra en cada píxel, en paralelo, utilizando cuatro ventanas desfasadas, C1, C2, C3 y C4 escalonadas en 90° . Durante el tiempo de integración (Δt) las cargas eléctricas se acumulan en cada píxel, dando lugar a las muestras Q_1 , Q_2 , Q_3 y Q_4 . [19]

A partir de esta distribución de cargas, la intensidad del píxel medido (A) y el offset (B) se pueden calcular como: [19]

$$A = \frac{\sqrt{(Q_1 - Q_2)^2 + (Q_3 - Q_4)^2}}{2}$$

$$B = \frac{Q_1 + Q_2 + Q_3 + Q_4}{4}$$

Utilizando esta técnica, podemos obtener el desfase entre la iluminación y la reflexión (φ) así como la distancia (d), las cuales pueden calcularse mediante las siguientes funciones: [19]

$$\varphi = \arctan\left(\frac{Q_3 - Q_4}{Q_1 - Q_2}\right)$$

$$d = \frac{c}{4\pi f} \varphi$$

2.2.3.2. Funcionamiento mediante modulación pulsada (PM)

Este tipo de sensores calculan la profundidad a la que se encuentra cada objeto a partir del cálculo de la diferencia de fase entre los pulsos de la señal emitida y los pulsos recibidos por en el sensor.

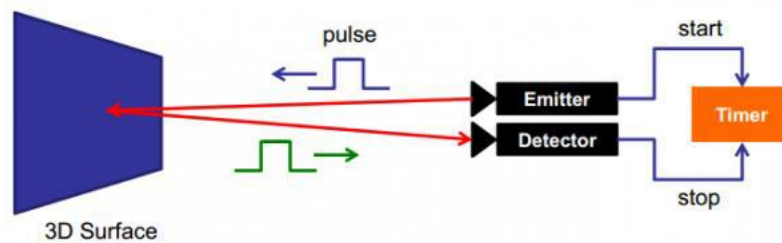


Figura 13 Principio de modulación pulsada (PM) [24]

En este caso, la diferencia de fase se calcula únicamente entre dos cargas eléctricas Q_1 y Q_2 mediante las cuales se determina la cantidad de electrones detectados por el sensor. [19]

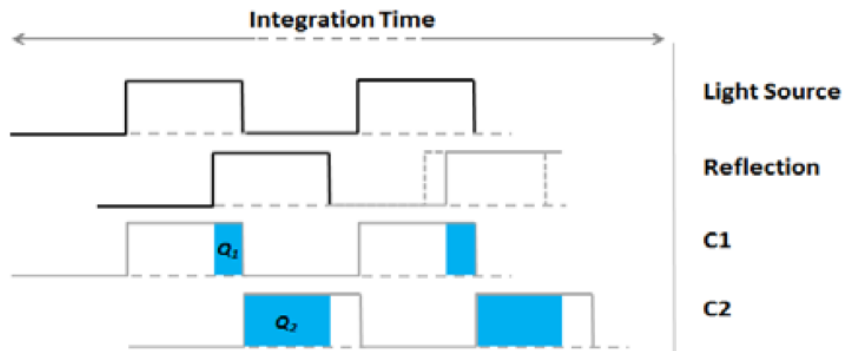


Figura 14 Distribución de cargas PM [19]

La distancia podría calcularse entonces a partir de dichas cargas y parámetros conocidos, como la velocidad de la luz (c) y el tiempo durante el cual el pulso de luz está activo (Δt), según la siguiente función: [19]

$$d = \frac{1}{2} c \Delta t \left(\frac{Q_2}{Q_1 + Q_2} \right)$$

El cálculo de la distancia máxima es directo a partir de la expresión anterior, la cual se define como: [19]

$$d_{max} = \frac{c \times t_p}{2}$$

Debido a que este tipo de modulaciones permiten transmitir una gran cantidad de energía en muy poco tiempo, permite que los sensores que las implementan sean adecuados para exteriores ya que se reduce la influencia de la iluminación de fondo y se puede alcanzar una relación señal ruido (SR) alta sin la necesidad de incrementar en demasía la potencia de emisión. [19]

Debido a esto, las cámaras que implementan modulación pulsada son más apropiadas para medir largas distancias o uso en exteriores, mientras que las que implementan una modulación continua son más apropiadas para distancias moderadas (0.5-6 m.) o uso en interiores. [19]

A primera vista, la complejidad del método de modulación continua, en comparación con el método de pulsos, puede parecer injustificada, pero un análisis más detallado de las ecuaciones de modulación continua revela que los términos $(Q_3 - Q_4)$ y $(Q_1 - Q_2)$ reducen ciertos errores sistemáticos en la medida. [19]

Además, el cociente en la ecuación de fase reduce los errores provocados por los factores de ganancia en las medidas de distancia, como la amplificación y atenuación del sistema, o la intensidad reflejada, lo cual hace que este sistema sea más robusto y preciso en las medidas. [19]

2.2.3.3. Fuentes de error en cámaras TOF

Los sensores de tiempo de vuelo están afectados por diferentes fuentes de error. En esta aplicación concreta, la influencia de error en la medida afecta considerablemente en el sistema global, debido a que como se comentó en el apartado acerca de la trilateración, un error en el conocimiento de las distancias provoca unas regiones de incertidumbre de la posición, que son cruciales a la hora de obtener la posición exacta a la que se encontraría el móvil en el escenario.

Como en cualquier sistema de medida encontraremos dos tipos de errores: sistemáticos y aleatorios.

Los errores sistemáticos aparecen debido a errores en la medida, en la recopilación de información o análisis de los datos. Este tipo de errores frecuentemente se pueden eliminar mediante procesos de calibración.

Los **errores sistemáticos** más comunes que encontramos en las cámaras TOF son: [21]

- "Wiggling error". Error debido a señales sinusoidales no perfectas.
- "Fixed Pattern Noise". Ruidos de patrón fijo en determinados píxeles.
- "Reflectivity and amplitude variations". Error debido a variaciones en amplitudes debido a iluminación y reflectividad no constantes.
- "Shotnoise" ruido de disparo que produce variaciones en el número de electrones de los MOSFET variaciones en la temperatura que producen desajustes en el material del semiconductor.
- "Temperature related". Errores debido a las variaciones de temperatura que producen variaciones en el material semiconductor del emisor infrarrojo.

Por otro lado, nos encontraremos con errores aleatorios, los cuales se producen por eventos imposibles de controlar durante el proceso de medición.

Los **errores aleatorios** más comunes que encontramos en las cámaras TOF son: [21]

- Ambigüedad en la distancia cuando se supera la máxima distancia fiable d_{max} .
- "Scattering light". Error debido a las reflexiones que se producen dentro de la lente de la cámara al emitir el haz infrarrojo.
- "Motion artifacts". Ambigüedad debida al movimiento de los objetos a medir.
- "Flying píxeles". Los píxeles voladores son estimaciones erróneas de profundidad que se producen cerca de las discontinuidades de profundidad y también en los límites de la imagen. [10] Esto se aprecia como píxel cuya estimación de medida cambia continuamente, sin aportar una información concreta y válida.

- Interferencias debidas al error de "Multicamino". Esto se produce cuando un píxel recibe la luz procedente del haz de luz directa que se refleja en el objeto junto con otros haces de luz que sufren múltiples reflexiones en el entorno hasta llegar al pixel, emporando la precisión de la medida.

Este efecto tiene mayor influencia sobre superficies cóncavas, como pueden las esquinas de una habitación. Estas reflexiones afectan al tiempo que tarda la señal desde que rebota en el objeto hasta llegar al receptor, lo cual deriva en la imagen que toma el sensor como una representación errónea de dichas esquinas, las cuales adoptan una representación redondeada como se puede ver en la Figura 15: [20]

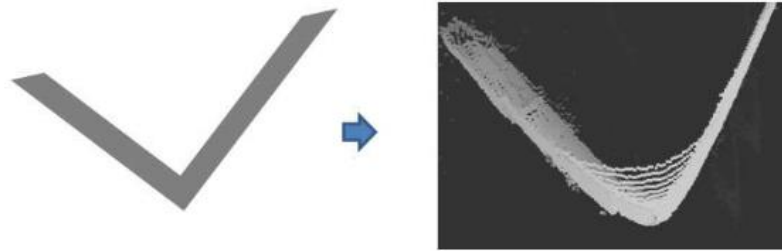


Figura 15 Efecto multicamino sobre equinas [22]

3. Capítulo 3: Desarrollo práctico

En este apartado se va a tratar en detalle los aspectos que caracterizan el sensor con el que vamos a trabajar, concretamente el modelo OPT8241 del fabricante Texas Instrument.

Posteriormente trataremos los diferentes aspectos configurables del sensor, estudiando el comportamiento de los diferentes perfiles de calibración que proporciona el fabricante y analizando cuál de ellos es el óptimo para nuestra aplicación.

Una vez conocido el perfil de calibración óptimo, se realizará un estudio cuya finalidad será la de mejorar mediante diversos tipos de modificaciones los resultados que ofrece el perfil de fábrica seleccionado. Para ello se analizan los resultados y se estudian las posibilidades de mejorar las prestaciones, o si por el contrario, la mejor decisión sería no modificar estos parámetros de calibración y mantener los del fabricante.

Una vez elegido, se realizará una batería exhaustiva de medidas cuyo fin es poner a prueba el sensor en un escenario lo más realista posible en cuanto a entorno de trabajo y a las diferentes situaciones que podrían darse en una situación real, estudiando cómo se comportaría el sensor en cada una de ellas y la precisión en la medida que se obtendría en cada caso.

Por último, se estudiará el funcionamiento del sistema si se cambian las marcas pasivas comunes por un tipo especial de espejo, con el objetivo de mejorar la precisión y el alcance del sensor.

El objetivo de este capítulo es determinar la configuración óptima del sensor y tener un conocimiento preciso de su comportamiento en todas las situaciones que podrían darse en el entorno real de medida, lo que influirá de manera determinante en la precisión de la localización del móvil en el espacio inteligente.

3.1. Sensor OPT8241 Texas Instrument

El fabricante Texas Instrument ofrece una familia de cámaras TOF, los cuales se presentan como placas de evaluación compuestas por 1, 2 o 3 chips. La primera generación de estos sensores, denominados “Haddock” se trata de una solución de 3 chips (OPT8140+VSP5324+OPT9220). La segunda generación, denominados “TinTin” se trata de una solución de 2 chips (OPT8241+OPT9221). Nosotros estudiaremos esta segunda generación de dispositivos en nuestro proyecto, los cuales proporcionan una resolución de 320x240 píxeles. Las generaciones futuras incluirán soluciones de 1 o 2 chips en diferentes resoluciones. [13]

El diagrama de bloques de la configuración “TinTin” se puede ver en la Figura 16, pueden observarse tres elementos diferenciados: una parte dedicada a la iluminación, la cual se controla con una señal de modulación. Otra parte dedicada al sensor de profundidad junto con la electrónica de control y sincronía temporal entre señales. El tercer elemento es el formado por el controlador OPT9221, responsable de la etapa de computación y gestión de la memoria del sistema. También se puede observar en la figura el principio de funcionamiento de la cámara, mediante la reflexión en la escena de luz modulada.

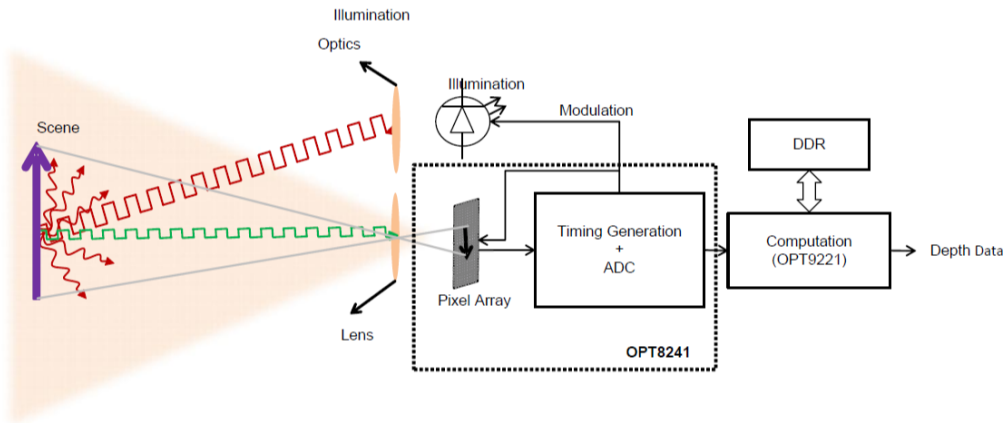


Figura 16 Diagrama de bloques OPT8241 + OPT9221 [17]

En la Figura 17 podemos encontrar los elementos más importantes de la placa de evaluación:

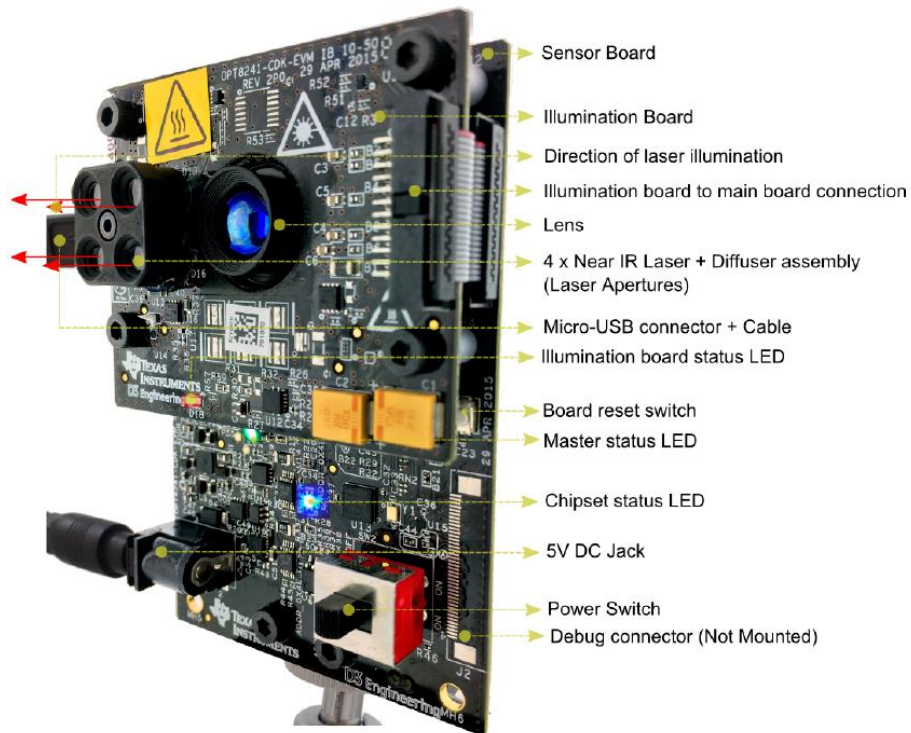


Figura 17 Componentes Placa de Evaluación [9]

Las principales especificaciones del dispositivo se muestran en la Tabla 1 donde cabe destacar la resolución del sensor y el campo de visión, dos aspectos fundamentales a tener en cuenta en nuestra aplicación.

Item	Value (Typical)
External power supply requirements	3 A at 5 V
Pixel resolution	320 x 240
Field of View	87° Diagonal. 74 (H) x 59 (V)
Maximum average optical output power	1.5 W
Connectivity	USB 2.0
Operating Conditions	0°C to 40°C (Ambient)

Tabla 1 Principales especificaciones [11]

Con este sensor se obtiene una información de distancia en una cuadrícula formada por 320x240 píxeles y disponemos de un rango de visión de 87° en diagonal, datos a partir de los cuales se obtendrá el tamaño máximo de la celda de posicionamiento admisible.

En cuanto a las características del láser del sensor, podemos ver los aspectos más relevantes en la Tabla 2, donde cabe destacar el ancho del pulso y la frecuencia de emisión.

Item	Value (Typical)
Wavelength	850 nm
Peak Optical Output	10 W
Pulse train width	18 μs–28 ms
Pulse repetition frequency	12-MHz to 80-MHz square wave at 50% duty cycle.
Laser Beam Divergence	80° at 90% power relative to the center
Laser Beam Shape	Circular
Transverse Beam Mode	Diffused

Tabla 2 Especificaciones del láser [11]

La placa de evaluación dispone de dos leds de control que aportan información acerca del estado de funcionamiento en que se encuentra en cada momento. El denominado Led Maestro muestra el estado general de la tarjeta, mientras que el Led del Chipset indica el estado del ToF. En la Tabla 3 podemos observar todos los posibles estados y cómo se identifican mediante dichos leds.

State	Master LED	Chipset LED	Status Type
All OK, Streaming OFF	Bright Green	Blue	Info
All OK, Streaming On	Cyan	X	Info
TFC Test Mode Enabled	X	Off	Info
Firmware Upgrade in progress	Blink (Magenta)	X	Info
Firmware Upgrade done	Magenta	X	Info
DC Jack unplugged	Dimmed colors	X	Info
Over Temperature	X	Magenta	Warning
TFC booted, but status failed	X	Red	Error
TFC did not boot	Blink (Red)	Off	Error

Tabla 3 Leyenda de colores de los LEDs [11]

En condiciones de funcionamiento normales, el LED maestro debe ser verde/cian y el LED del chipset debe ser azul. [11]

Los datos de correlación de profundidad obtenidos por el sensor OPT8241 se digitalizan y se suministran al OPT9221 ToF Controller (TFC). El TFC procesa los datos y proporciona la información de distancia para cada píxel. [12]

En la Figura 18 se muestra un diagrama de bloques del sistema con información más detallada sobre la dependencia entre los diferentes elementos.

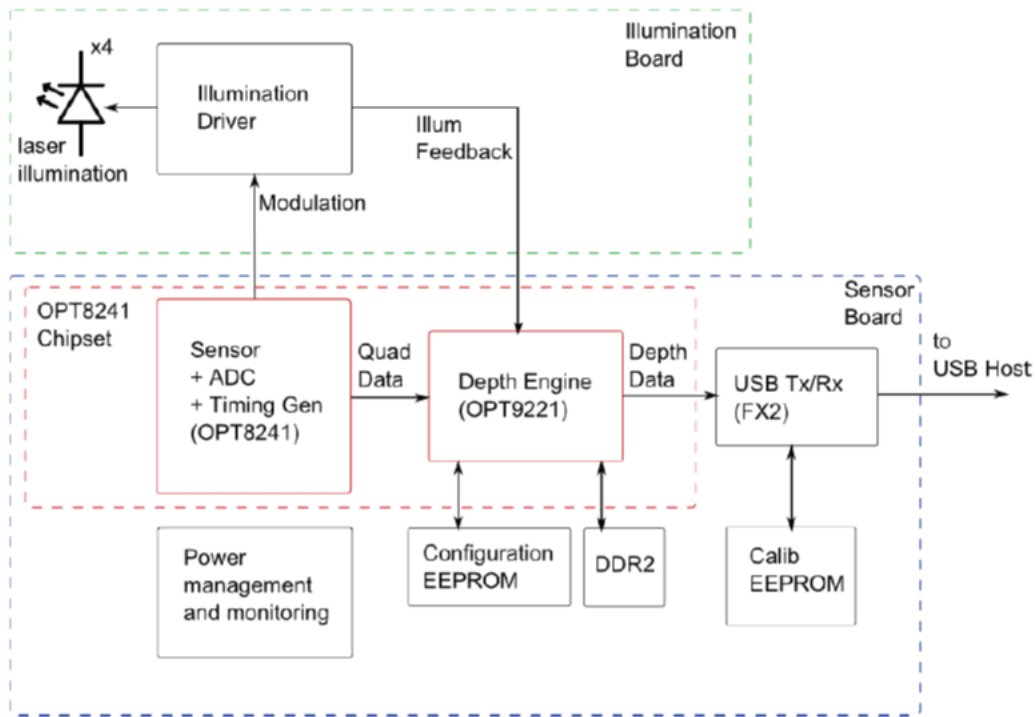


Figura 18 Diagrama de bloques completo [12]

Esta placa de evaluación se compone de los elementos que podemos ver en la Tabla 4:

Item	Specification
Time of Flight Sensor	OPT8241
Time of Flight Controller	OPT9221
Sensor resolution	320 x 240 (QVGA)
Field of view	74.4 (H) x 59.3 (V)
Frame-rate	12 ... 60 fps
Illumination source wavelength	850 nm
Operating range	Up to 4 m
Connectivity	USB 2.0, micro connector
Cable	1.8 m, Micro USB B-type and standard Male A type
Operating Conditions	0°C to 40°C (Ambient)
CDK Power supply max power	15 Watt
Size	88.8 mm (L) x 60 mm (W) x 24.3 mm (H)

Tabla 4 Especificación de componentes de la placa de evaluación [12]

La ventaja que proporciona el empleo de una placa de evaluación frente otras alternativas comerciales es la posible modificación del hardware, el software y el firmware de fábrica. En una fase inicial esto no es recomendable ya que requiere un estudio previo muy exhaustivo, pero en siguientes etapas del desarrollo será necesario realizar algunos cambios como remplazar la óptica de la cámara para poder abrir o cerrar el campo de visión en función de la aplicación concreta.

Otra de las ventajas que proporciona el uso una placa de evaluación consiste en la posibilidad de tener acceso a señales intermedias el sensor, las cuales sería imposible observar en un modelo integrado. Esta información podría ser útil para captar las señales que reciben cada uno de los píxeles para realizar un procesamiento de señal de más bajo nivel, así como modificar los esquemas de modulación de la etapa de iluminación

El kit de desarrollo que ofrece el fabricante proporciona varios perfiles de calibración, los cuales según informa el fabricante son con fines de demostración, no están pensados con la finalidad de ser la calibración más precisa posible en una aplicación determinada. El dispositivo permite realizar una configuración propia mediante el software específico proporcionado para trabajar con este sensor, denominado “Voxel Viewer”.

En el propio hardware vienen instalados tres perfiles, los cuales pueden ser utilizados con la configuración de fábrica o bien utilizarlos como plantilla para realizar una calibración propia, aspecto que se tratará más adelante.

En la Tabla 5 se muestra un resumen de las características de cada uno de estos tres perfiles de calibración:

Parameter	Long Range	Short Range	High Ambient
Parent Profile	Lens Only	Long Range	Long Range
Frame rate (fps)	30	60	30
Sub-Frames	4	2	4
Quads	6	6	6
Integration Duty Cycle (%)	30	10	5
De-aliasing	Enabled	Enabled	Enabled
Illumination Power (%)	100	40	100
Unambiguous Range (m)	10	5	5
Base Frequency (MHz)	60	60	60
De-aliasing Frequency (MHz)	40	40	40
Usable Range	4 m	1.5 m	1.5 m
Calibrations performed	Temperature, Common phase offset, Pixel to Pixel, Pixel cross-talk ⁽¹⁾	Common phase offset ⁽²⁾	Common phase offset ⁽²⁾

Tabla 5 Características de perfiles por defecto [12]

El factor principal en la elección de un perfil u otro se trata del rango de distancias en el cual vaya a trabajar el sensor, para mediciones de hasta 4 metros y no muy cercanas es más apropiado utilizar le perfil “Long Range”, mientras que para medidas hasta 1.5 metros será más apropiado utilizar el perfil “Short Range” o “High Ambient” donde en interiores, experimentalmente se han obtenido mejores resultados con el primero de ellos.

En función de la versión del modelo adquirido, este puede incluir más perfiles de calibración por defecto además de estos tres fundamentales. En nuestro caso el sensor incluía 7 perfiles de calibración, los tres comentados anteriormente, además de los siguientes: “Metriculous Long Range”, “Metriculous Short Range”, “Lens Only” y “No calibration”. [12]

De estos perfiles añadidos son de interés los denominados “Metriculous” pues son una versión mejorada de los perfiles primarios, “Long Range” y “Short Range”. En concreto, el nuevo perfil de largo alcance proporciona medias correctas hasta aproximadamente los 6 metros mientras que con el perfil primario el máximo son 4 metros. Posteriormente se realizará un estudio concreto de estos perfiles, realizando diversas pruebas para decidir cuál es la calibración óptima en nuestro sistema. [12]

El software proporcionado en el kit de desarrollo permite interactuar con la placa de evaluación de una manera muy visual y relativamente sencilla. Este SW soporta las siguientes funcionalidades: [12]

- Ver los siguientes datos en tiempo real:
 - Fase
 - Amplitud
 - Ambiente
 - Distancia
 - Profundidad
 - Nube de puntos
- Configurar los ajustes de la cámara
- Estadísticas básicas:
 - Promedio temporal y espacial
 - Desviación estándar temporal y espacial
 - Histograma
- Aplicación de filtros (Espaciales y temporales):
 - Adición/eliminación/inserción de filtros
 - Configuración de los coeficientes de filtrado
- Calibrar la cámara
- Actualizar del firmware del OPT9221

En la Figura 19 podemos ver el aspecto de la pantalla principal del programa Voxel Viewer, donde nos encontramos tres ventanas para la visualización, en tiempo real, de parámetros como: fase, distancia, nube de puntos, amplitud, ambiente o profundidad. Siendo la representación de la fase, amplitud y nube de puntos los escenarios de interés en nuestro estudio.

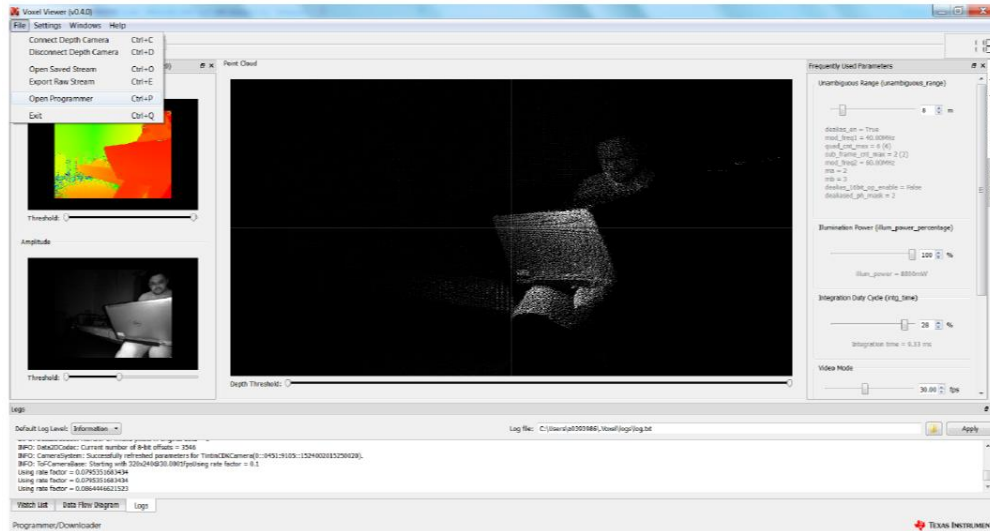


Figura 19 Pantalla principal Voxel Viewer [12]

En el documento “Quick Start Guide” podemos encontrar información muy detallada acerca de la calibración de todos los elementos susceptibles de calibración del sensor, sean:

- Lente
- Frecuencia
- Interferencia
- No linealidad
- Temperatura
- Fase común
- Pixel-Wise

En el siguiente punto se explicarán más en detalle aquellas calibraciones que se han realizado.

El programa Voxel Viewer nos permite extraer la información de la distancia a la que se encuentra un objeto de dos formas posibles.

La primera de ellas consiste en seleccionar mediante una pulsación del botón “Ctrl” y el botón izquierdo del ratón sobre el píxel que se desea obtener la distancia, esto se puede hacer para tantos puntos como se desee. La información de estos píxeles se podrá ver en tiempo real en la pestaña Watch List del programa, donde nos muestra información acerca de las coordenadas del píxel seleccionado, la intensidad de este y la distancia respecto al eje Z a la que se encuentra del sensor. Una vez recopilados los píxeles deseados, pulsando con el botón derecho del ratón sobre la pestaña Watch List podemos guardar estos datos como un fichero CSV (fichero de datos delimitado por comas) donde guardaremos en un fichero los datos que se estén muestreando en la Watch Windows en el instante en el cual se guarda el fichero.

La segunda manera de obtener los datos de distancia de los píxeles consiste en la extracción de un fichero PCD (Point Data Cloud). Este fichero se extrae pulsando con el botón derecho sobre

la imagen de nube de puntos, guardando la información de las distancias que recoge cada píxel en el instante concreto en el que se guarda el fichero. Este fichero se trata de un standard en el área de visión tridimensional, donde se recoge la información de todos los píxeles que componen la imagen, guardando información de la distancia a la que se encuentra el sensor para cada una de las coordenadas (x, y, z), así como información de la intensidad que recoge cada píxel. La información en este fichero se recoge ordenada de forma que cada fila corresponde a un píxel, representando primero los 320 de la primera fila de la imagen, después los 320 de la segunda fila y así sucesivamente para las 240 filas de la imagen. Sabiendo esto, si se quiere acceder a la información del píxel de la coordenada (10, 5) deberemos dirigirnos a la fila $320 \cdot 5 + 10 + 11$. Se suma 11 filas al final debido a que el fichero tiene una serie de cabeceras que aportan información acerca del archivo, podemos ver información detallada de cada uno de estos campos en la referencia [15]

Para tratar este tipo de ficheros mediante Matlab, existe una función denominada `pcread()` mediante la cual podemos cargar este tipo de ficheros. Esta función distribuye la información del fichero en una serie de campos, incluyendo en el campo “.Location” la información de las distancias respecto a cada eje en una matriz de mismo número de filas como píxeles y tres columnas, es decir, elimina las cabeceras y configura una cuarta columna que indica la intensidad que recibe cada píxel. El resto de los campos nos aportan otra información como los límites en los ejes x, y, z. [16]

3.2. Evaluación de los perfiles de calibración y calibraciones

En este apartado vamos a realizar una serie de pruebas para caracterizar cada uno de los perfiles que proporciona el fabricante y poder determinar cuál de ellos es el más apropiado para nuestra aplicación.

Para ello, vamos a realizar una batería de pruebas para analizar la fiabilidad de la medida que nos proporciona el sensor a diferentes distancias, contrastándolas con la distancia real a la que se coloca dicho objeto.

Para realizar las medidas experimentales, se ha construido un banco de pruebas basado en un carril, en el cual se puede situar la cámara y se puede situar un objeto o marca en diferentes distancias de forma precisa. El carril consta de dos perfiles tipo Bosch de 270cm que han sido unidos de forma longitudinal, dando lugar a un rango de desplazamientos de 0 a 540cm. Sobre el carril se ha unido una cinta métrica para poder cambiar la posición de forma sencilla, precisa y estable.

El proceso de medida consiste en situar la cámara en el origen del carril y medir la distancia hasta una marca pasiva colocada a una distancia conocida.

En primer lugar, se ha medido la posición de una marca situada en una distancia conocida comparando los resultados con los diferentes perfiles de calibración. De esta comparativa se han seleccionado los perfiles “Long Range”, “Metriculous Long Range” y “Metriculous Short Range” como los perfiles más adecuados para el margen de distancias de interés. El resto de perfiles han sido descartados, ya que las medidas son de escasa precisión.

Con los perfiles de calibración seleccionados se realiza una batería de medidas, en la cual se mide la posición en todas las distancias de 0 a 540cm a intervalos de 20 cm, tomando en cada situación un mínimo de 50 medidas con la cámara en los píxeles centrales de la marca, de las cuales se realizará posteriormente la media aritmética y se comparará el resultado con la distancia real a la

que se colocó la marca. La extracción de estos datos se realizará mediante el primero de los procedimientos comentados anteriormente, es decir, extrayendo un fichero CSV para poder observar la distancia respecto al eje Z. La marca está alineada con el sensor, por lo que son los píxeles correspondientes a la marca los que nos aportan la medida de interés. El escenario se puede observar en la Figura 20.



Figura 20 Escenario para medidas en rayo directo

3.2.1. Ensayo con los perfiles del fabricante

En primer lugar, como se comentó anteriormente se va a realizar diversas medidas enfrentando sobre el carril la marca y la cámara.

Se realizaron más de 50 medidas para cada distancia en cada uno de los perfiles de calibración, las cuales fueron promediadas y representadas conjuntamente la Figura 21 para su mejor comparación:

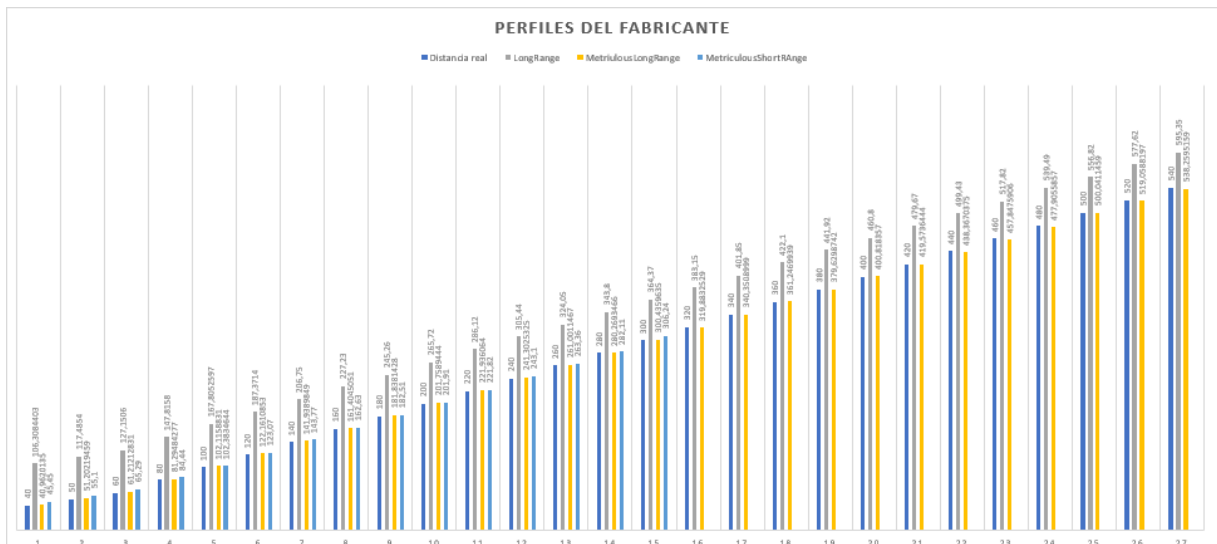


Figura 21 Medidas en rayo directo con perfiles del fabricante

En el gráfico de barras superior podemos ver una representación donde las barras de color azul representan la medida real a que se colocó la marca pasiva en el experimento y el resto de las barras nos indican la medida que se obtuvo con la cámara en los diferentes perfiles que el fabricante ofrece.

El experimento se realizó en pasos de 20 cm sobre el carril para todos los perfiles de calibración, alcanzando las 27 medidas de la gráfica superior en los perfiles de largo alcance y 15 medidas para el perfil de corto alcance, para el cual también se realizaron otras medidas adicionales en el primer tramo del carril en pasos de 10 cm, donde se apreció que su funcionamiento es bueno en distancias cortas, pero no tan eficiente en distancias medias.

Haciendo un primer análisis visual del gráfico, podemos ver como los perfiles “Metriculous Long Range” y “Metriculous Short Range” consiguen unas medidas muy cercanas a la real, mientras que el perfil “Long Range” comete un error sistemático en todas sus medidas que las aleja una distancia en torno a los 60cm de la distancia real.

De una manera más analítica, si calculamos el error medio que se obtiene entre todas las medidas de cada uno de los perfiles obtenemos los siguientes resultados:

- Error medio (Long Range): 63.286cm.
- Error medio (Metriculous Long Range): 1.214 cm.
- Error medio (Metriculous Short Range): 3.546 cm.

De esta manera podemos ver cómo el perfil del fabricante que mejores resultados nos ofrece es el denominado Metriculous Long Range.

Una vez conocido cual es el perfil que nos proporciona el fabricante que mejor se adapta nuestras necesidades, vamos a procurar mejorar su precisión mediante una serie de calibraciones.

Para ello, vamos a calibrar dos aspectos. En primer lugar, se realizará una calibración de la lente y en segundo lugar se calibrará la Fase común del sensor, siguiendo para cada caso el procedimiento de calibración que proporciona el fabricante en el documento “Quick Start Guide – Voxel Viewer”. [13]

3.2.2. Calibración de la lente

El objetivo de la calibración de la lente es superar las distorsiones ópticas introducidas por el objetivo. En ausencia de distorsiones ópticas, cada píxel (u_i, v_i) puede ser convertido a las coordenadas mundiales equivalentes (x_i, y_i, z_i) aplicando el modelo de cámara estenopeica mostrado en la Figura 22. El objetivo de la calibración de la lente es transformar las coordenadas distorsionadas de los píxeles en coordenadas equivalentes que se pueden utilizar bajo el modelo de cámara estenopeica. [13]

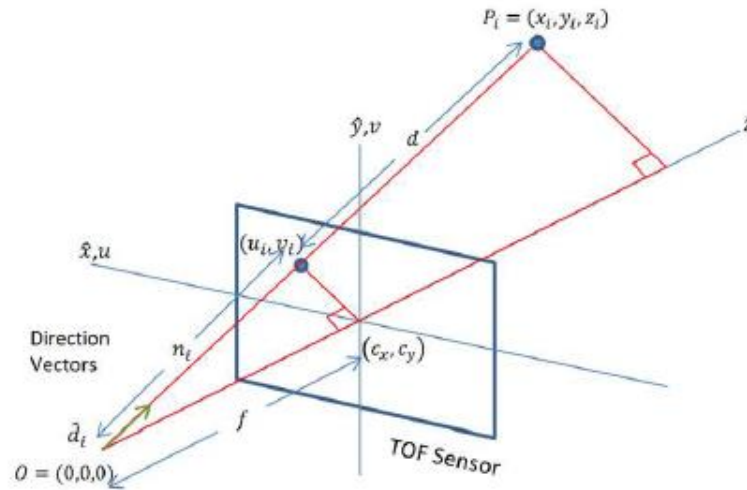


Figura 22 Modelo de cámara estenopeica [13]

El proceso extrae un conjunto de parámetros comúnmente conocidos como *intrínsecos* que incluyen nueve valores $[k_1, k_2, k_3, p_1, p_2, c_x, c_y, f_x, f_y]$, donde $[k_1, k_2, k_3]$ son los coeficientes de distorsión radial, $[p_1, p_2]$ son los coeficientes de distorsión tangencial y $[c_x, c_y]$ son las coordenadas del centro de la imagen y por último $[f_x, f_y]$ son las distancias focales x e y. [13]

La calibración del lente requerirá un patrón de tablero de ajedrez, plano y rígido como el que se muestra en la Figura 23.

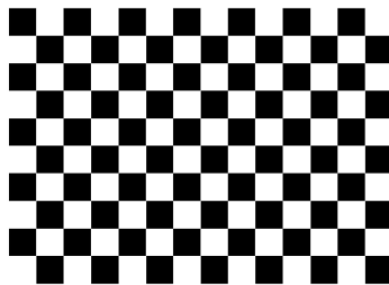


Figura 23 Tablero de calibración [13]

En nuestro caso, vamos a realizar una calibración de la lente heredando el resto de los parámetros del perfil de “Metriculous Long Range” debido a que fue es el perfil que como hemos visto ofrece mejores resultados.

El proceso a seguir es el que se explica en el documento comentado anteriormente, entrando en el menú configuración seleccionaremos que únicamente se desea calibrar la lente y heredar los demás parámetros del perfil correcto. [13]

Una vez hecho esto, procedemos a tomar varias capturas del tablero de calibración de la lente donde observaremos resultados como el de la Figura 24.

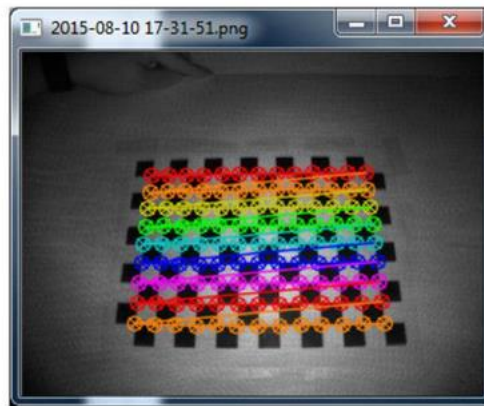


Figura 24 Imagen de calibración de lente correcta [13]

Sabremos que una captura de calibración ha sido tomada correctamente cuando una vez realizada, sobre la imagen aparece una estructura de colores con forma de Zig-Zag como podemos ver en la figura 21. Por el contrario, si no se obtiene una imagen correctamente representada, únicamente observaremos los puntos en color rojo, donde se puede apreciar la carencia de algún cuadrado del tablero, lo que imposibilita la representación de la estructura en Zig-Zag y la consecuente captura correcta para la calibración, como podemos ver en la Figura 25:

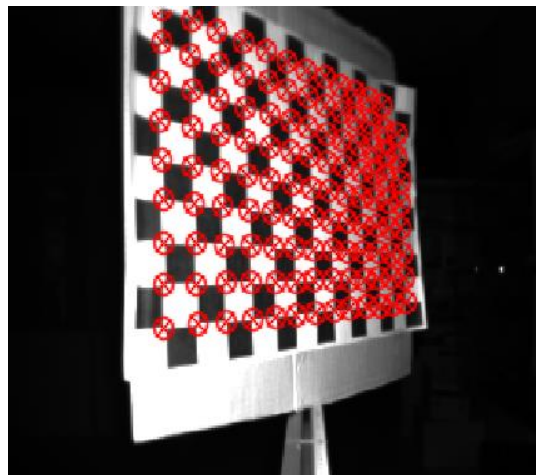


Figura 25 Ejemplo de una mala imagen de calibración

Para obtener unos buenos resultados en las capturas de calibración tomadas, es recomendable seguir las siguientes indicaciones: [13]

- Ocupar el rango de visión tanto como sea posible con el patrón del tablero de ajedrez.
- Presentar el patrón en diferentes orientaciones en cada captura de calibración.
- Mantener el patrón del tablero de ajedrez quieto mientras se realiza la captura, ya que múltiples muestras (el valor predeterminado es 200) serán capturadas y promediadas para eliminar el ruido, antes de que se extraigan los puntos de las esquinas para calibración.

Una vez realizada una captura orientando el tablero en todas las posiciones posibles (al menos 8) el programa podrá realizar la calibración de lente a partir de estas imágenes, con esto crearemos un perfil nuevo, que denominaremos “Long Lente”.

Ahora repetiremos el ensayo anterior para estudiar si hemos conseguido nuestro propósito de mejorar la precisión de la medida con respecto al perfil proporcionado por el fabricante, o si por el contrario hemos empeorado la precisión, lo que veremos en el gráfico de la Figura 26.

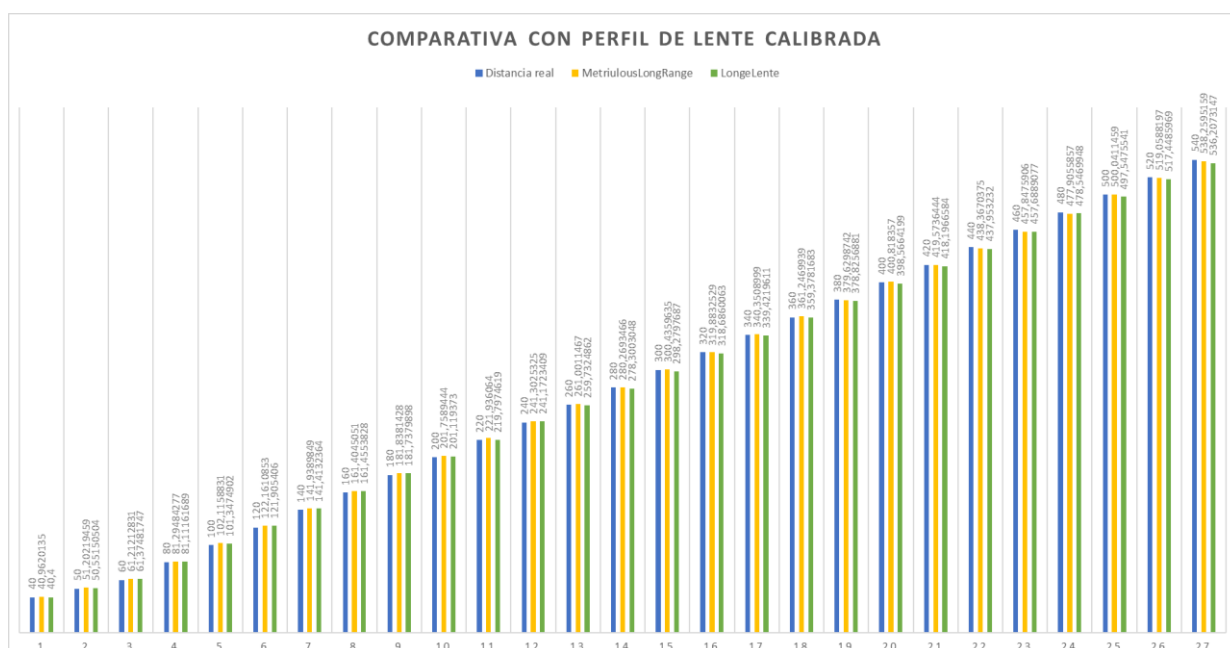


Figura 26 Comparación del perfil "Long Lente"

Los errores medios obtenidos son:

- Error medio (Metriulous Long Range): 1.214 cm.
- Error medio (Long Lente): 1,445 cm.

Como podemos ver en la gráfica superior y en el error medio cometido, pese a realizar la calibración de la lente los resultados obtenidos son peores que los que se obtuvieron con el perfil que proporcionaba el fabricante.

Esto se debe a que las calibraciones de fábrica se realizan en escenarios ideales realizados por el fabricante, por lo que frecuentemente es complicado que el usuario consiga unas mejores prestaciones con su calibración.

3.2.3. Calibración de fase común

Teniendo en cuenta la verificación de la calibración de la lente aportada por el fabricante, vamos a calibrar la fase común de la cámara con el mismo objetivo.

La calibración de fase común asume que la relación fase-profundidad es linealizada por otra calibración en el momento en que se ejecuta. La conversión de fase a profundidad puede entonces resumirse mediante la ecuación lineal: [13]

$$d_i = \frac{c}{2f\rho} (\varphi_i - \varphi_{0i}) + d_{0i}$$

La calibración de fase común correlaciona la fase medida con la profundidad real en un solo punto (φ_{0i}, d_{0i}) . [13]

Esto se hace para todos los píxeles indexados por i . Sólo el píxel central está correlacionado y se utiliza el promedio de 200 muestras (por defecto), como se puede ver en la Figura 27. [13]

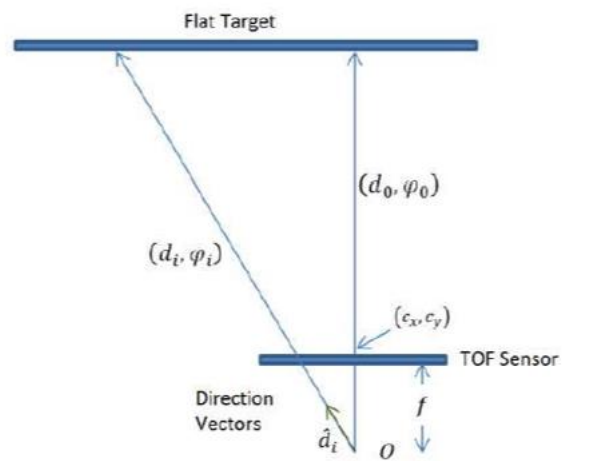


Figura 27 Configuración de la calibración de fase común [13]

Para realizar esta calibración, como se explica en el documento [13], entrando en el menú de configuración seleccionaremos realizar ahora únicamente la calibración de la fase común y heredar el resto de los parámetros de nuevo del perfil “Metriculous Long Range”. Este nuevo perfil que se creará lo denominaremos “**FaseCal**”.

Una vez nos encontramos en el menú de configuración de la fase, nos aparecerá una pantalla como la que podemos ver en la Figura 28. Debemos ahora situar en frente de la cámara una superficie plana lo suficientemente grande para que ocupe todo el campo de visión de la cámara, de tal manera que, en la medida de lo posible, todos los píxeles de la cámara obtengan la misma profundidad. Debemos conocer la distancia a la que se encuentra dicho plano e introducirlo en el menú de calibración, la exactitud de esta medida será decisiva en la precisión que obtendremos posteriormente. [13]

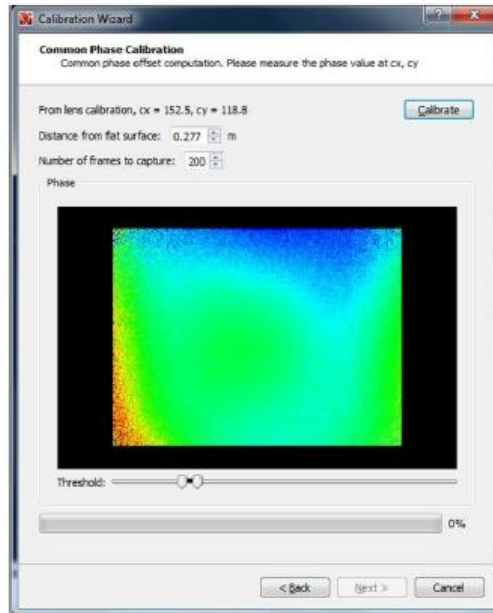


Figura 28 Menú de configuración de la fase común [13]

Ahora realizaremos las mismas medias que hicimos en los experimentos anteriores para ver los resultados obtenidos, los cuales se representan en la Figura 29.

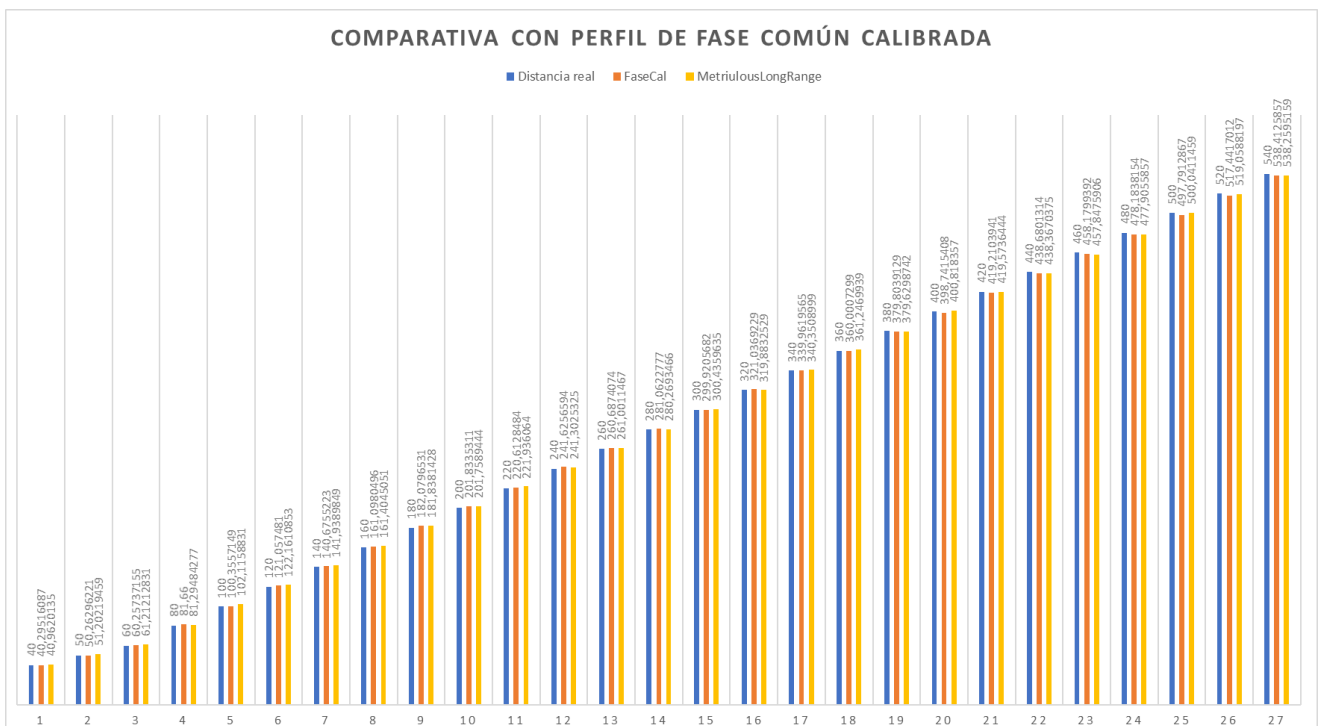


Figura 29 Comparación del perfil "FaseCal"

Los errores medios obtenidos son:

- Error medio (Metriculous Long Range): 1.214 cm.
- Error medio (FaseCal): 1,047 cm.

En este caso, a diferencia de la calibración anterior, sí que podemos ver cómo hemos conseguido reducir el error medio 1.67 mm respecto al perfil del fabricante.

Como conclusión de estos experimentos, podemos decir que el perfil que mejores resultados ofrece se trata del perfil heredado de “Metriculous Long Lente” con una calibración de la fase común, el denominado “**FaseCal**”, con el cual conseguimos un error medio de tan solo ~1,05 cm aproximadamente.

3.3. Caracterización del perfil de medidas FaseCal

Una vez calibrado y obtenido el perfil de calibración que se utilizará en todas las siguientes medidas, se va a proceder a realizar una caracterización exhaustiva del sensor, determinando la precisión que se obtiene en las medidas en diferentes escenarios y distancias.

En primer lugar, para ver la precisión del sensor en medidas de la posición cuando las marcas están alineadas con el sensor se utilizarán las mismas medidas que se utilizaron para determinar la validez de la calibración de fase común. Lo resultados que se obtuvieron fueron los que podemos ver en la Figura 30:

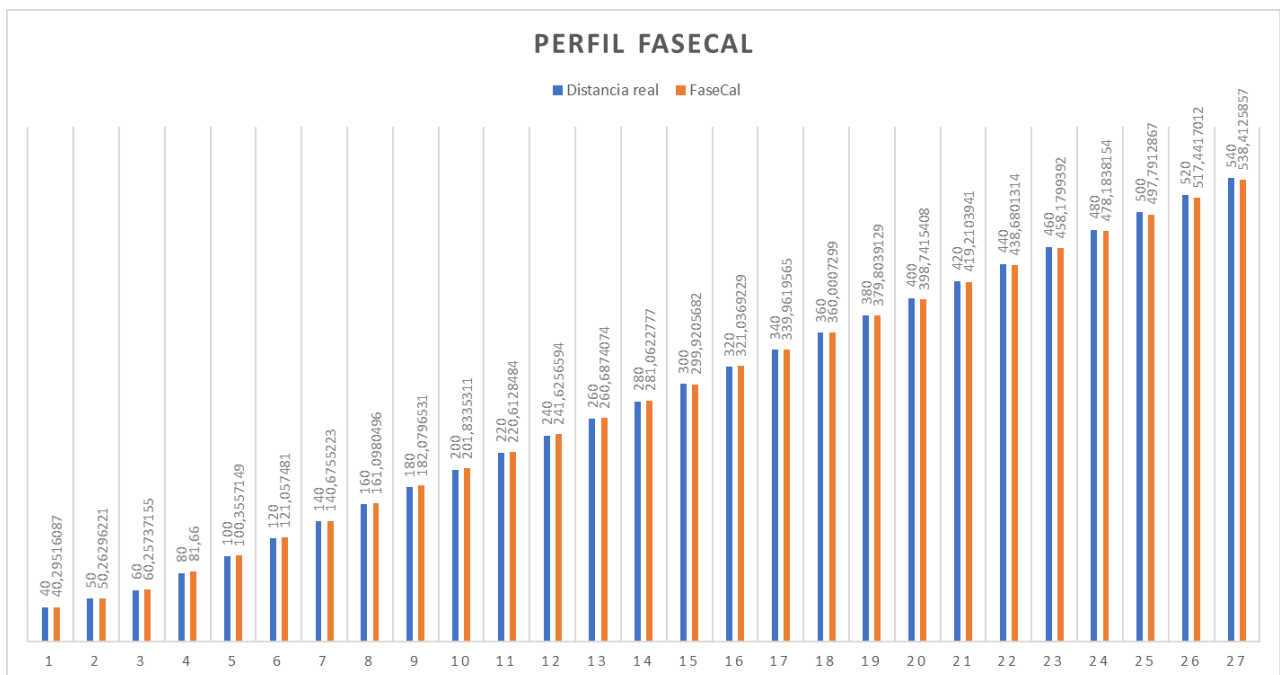


Figura 30 Análisis de precisión del perfil FaseCal

En la Figura 30 podemos ver cómo la distancia obtenida en todo el abanico de distancias se asemeja considerablemente a la medida real, obteniendo un error medio, como se comentó anteriormente de ~1,05 cm.

Este error, aunque es en gran medida “aleatorio” debido a que se debe principalmente a la inestabilidad temporal de la medida que ofrecen los píxeles (Flying pixels) se puede ver una evolución en el aumento del error de la medida en función del aumento de la distancia a la que se encuentre el objeto, esto lo podemos ver en la Figura 31.

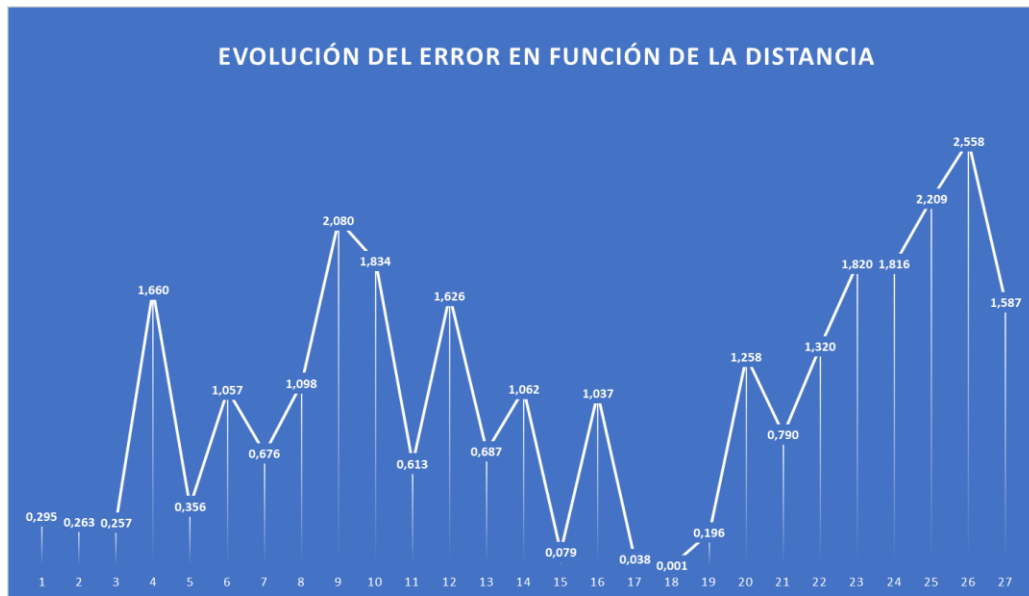


Figura 31 Evolución del error en función de la distancia

En la figura superior podemos ver que, aunque no sigue un aumento lineal definido, a medida que la distancia aumenta se produce también un aumento del error cometido.

Es muy importante definir que estas medidas son el resultado de la media aritmética de las distancias medidas en más de 50 puntos sobre la marca, pero en un solo instante temporal. El hecho de aumentar los píxeles de medida sobre la marca reduce notablemente la inestabilidad de la medida, pero para una medición realmente precisa lo más apropiado sería realizar dichas medidas en diversos instantes temporales y realizar el promedio de los resultados obtenidos.

Para ver la influencia de la inestabilidad temporal se ha colocado la marca pasiva a una distancia de 415cm y se ha realizado la medida en una ventana de 25 píxeles sobre ella. Este procedimiento se ha repetido 10 veces para comparar la variación en los resultados obtenidos, donde hemos apreciado en el caso que más difiere la medida, la variación es de 2.23cm. Esta es una situación bastante pesimista debido a que 4.15m es una distancia donde la precisión del sensor empieza a decaer, pero nos hace ver que la influencia de la variación temporal de las medidas es de gran relevancia y que debe tenerse en cuenta en la caracterización del sensor.

A la hora de realizar las medidas de la marca en posiciones no enfrentadas al sensor, se presenta el problema de la imposibilidad de obtener la medida directamente a partir de la interfaz gráfica del programa Voxel Viewer, debido a que éste, de manera directa, únicamente aporta información acerca de la distancia a la que se encuentra el objeto respecto del eje Z, por lo que para la obtención de medidas en otras posiciones será necesario calcular el módulo del vector de distancia a la que

se encuentra dicho objeto, teniendo en cuenta la distancia a la que se encuentra en las tres coordenadas (x, y, z).

Para solucionar estos problemas se ha diseñado un programa capaz de obtener la medida a la que se encuentra el objeto, realizando la medida sobre una ventana de píxeles sobre el objeto a medir durante varios instantes temporales, sabiendo el píxel central del objeto y obteniendo varios ficheros PCD a partir del programa Voxel Viewer donde se encuentre dicho objeto en el píxel seleccionado. El funcionamiento de dicho programa se explicará más en detalle en el siguiente punto.

3.3.1. Distance PCD y Algoritmo de la ventana

Como se comentó anteriormente, el programa proporcionado por el fabricante, Voxel Viewer, permite seleccionar una serie de píxeles en la imagen de fase o amplitud y recopilar algunos datos de estos puntos como, por ejemplo, el valor de la coordenada z del vector posición. Guardándolos en un fichero CSV, el problema recae en que estos datos toman la información de un único instante temporal.

El segundo problema de este método de obtención de información es la limitación en cuanto a la información de los otros ejes x, y. Por ello, es necesario optar por la otra vía de información que proporciona el programa, los ficheros PCD, en ellos, como se comentó anteriormente, podemos encontrar información mucho más detallada para cada uno de los puntos que componen la matriz de 320x240 píxeles.

El problema de este método es la lentitud del proceso manual para extraer un fichero PCD de la imagen, buscar el píxel concreto en el fichero y calcular el módulo del vector posición a través de las componentes (x, y, z). Este proceso es aún más tedioso cuando no se trata de un único píxel si no de varios puntos en el objeto.

Para automatizar este tedioso proceso se ha creado un programa denominado “Distance PCD”, el cual se encarga de calcular la distancia a la que se encuentra un cierto píxel a partir del fichero PCD de la escena.

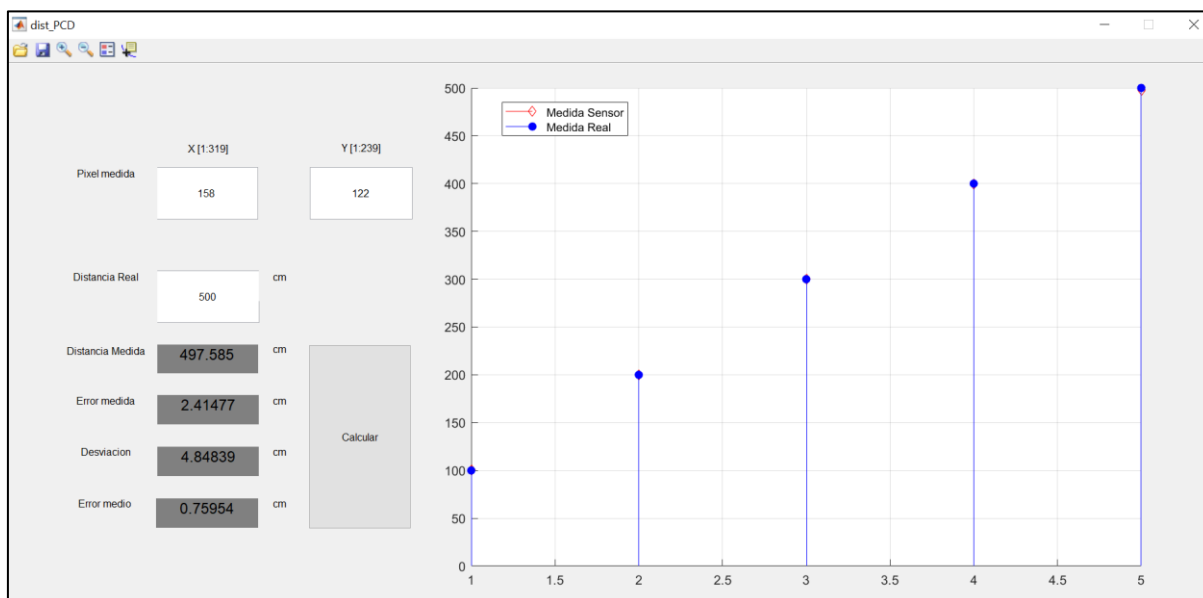


Figura 32 Interfaz Distance PCD

El programa se ha realizado a partir de una GUI de Matlab, cuya interfaz gráfica podemos ver en la Figura 32.

Este programa nos permite de una manera muy rápida conocer con precisión la distancia a la que se encuentra el objeto deseado. Para ello, es necesario introducir en la interfaz superior las coordenadas (x, y) del píxel central del objeto, y si queremos, la distancia real a la cual se encuentra situado para poder ver una comparación rápida entre las dos distancias. Esto se realiza mediante el cálculo de la diferencia de ambas, lo que en la interfaz se muestra como “error medida” y mediante una representación gráfica a la derecha de la interfaz donde se muestra tanto las distancias reales como las medidas.

Una vez introducidos estos valores, el programa pedirá por pantalla abrir el fichero PCD donde se encuentran los datos de la nube de puntos con el objeto en cuestión, por lo que este programa trabaja necesariamente en paralelo con Voxel Viewer ya que necesitamos este último para conocer cuál es el píxel central del objeto, pulsando con el ratón sobre él mientras se presiona la tecla Ctrl y observando su coordenada en la pestaña “Command window”. Además, para obtener el fichero PCD también es necesario el uso de este programa, lo que se realiza pulsando con el botón derecho del ratón sobre la imagen de la nube de puntos y eligiendo la opción de generar fichero PCD.

Este programa no solo halla en el fichero PCD la información de distancia del píxel solicitado, sino que lo hace para una ventana de píxeles sobre el objeto. Si se introduce el píxel (5, 5) el programa tomará este punto como centro de una ventana de 25 píxeles sobre la marca, de los cuales hallará las distancias a la que se encuentran cada uno de ellos y realizará una media aritmética de cada una de las medidas, resultado que consideraremos como la distancia final a la que se encuentra el objeto.

Del mismo modo se calculará la desviación típica de las distancias de estos 25 píxeles para observar la validez de las medidas, pues el objetivo de tomar un conjunto de píxeles en lugar de uno solo es obtener una medida con más precisión y paliar los efectos de los “flying píxels”. Por lo que las medidas en cada uno de estos píxeles no deberían variar en más de 1 mm entre sí debido a dicha inestabilidad, por lo que la desviación típica del conjunto debería ser baja, de no ser así se estarían produciendo errores en la medida.

Para decidir el tamaño idóneo de la ventana de píxeles a tomar, se han realizado una serie de pruebas. Sabiendo que las marcas en una aplicación real se encontrarán orientadas hacia el suelo, sujetas sobre unos postes verticales desde el techo y a una altura a 3.5 metros respecto al suelo. Mediante el teorema de Pitágoras, sobre el escenario que se puede ver en la Figura 33, vamos a estudiar la variación en la medida comparando la distancia obtenida a un cierto punto (A) y la distancia a uno cercano (B) para comprobar cómo afecta en la medida final seleccionar un píxel concreto o uno muy próximo.

Partiendo de la base de que se mide sobre una superficie plana, tomando como primer cateto (c1) el rayo directo perpendicular tanto al sensor como a la marca sobre el punto A y el segundo cateto (c2) como la distancia desde el punto en el cual el rayo directo impacta sobre la marca (A) hasta otro punto cercano al anterior (B). Vamos a calcular la distancia máxima a la que puede situarse el píxel B para que la distancia de la medida (h) no varíe más de 1 mm respecto a la medida al punto inicial, A.

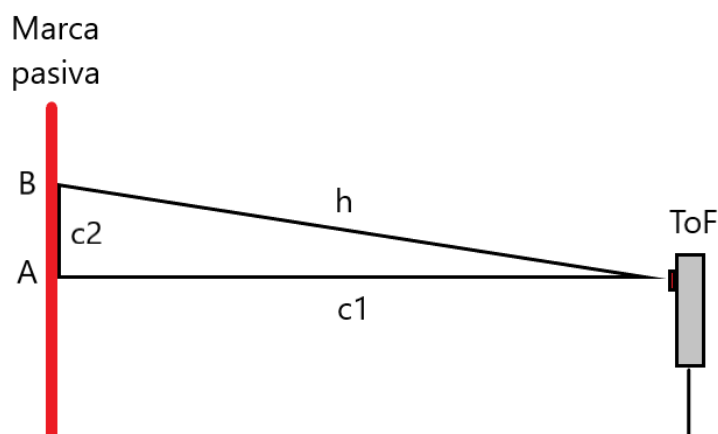


Figura 33 Vista lateral del escenario de medida para estudiar la variación de distancia entre puntos cercanos de la marca y la cámara de tiempo de vuelo

Situando la cámara a una distancia de la marca de 5 metros, podemos ver cómo una marca de 10x10 cm en el sensor se traduce a 5x5 píxeles. Siendo el primer cateto (c_1) igual a 5 metros hasta el píxel central, fijando la hipotenusa (h) en 5.001 m. A partir del teorema de Pitágoras podemos calcular la distancia (c_2) a la que se puede colocar el píxel respecto B de A para la diferencia entre medir al punto A y al punto B no varíe en más de 1 mm.

$$c_2 = \sqrt{h^2 - c_1^2} = \sqrt{500.1^2 - 500^2} \approx 10 \text{ cm}$$

El punto máximo al que puede situarse B resultante es a 10 cm de A, por lo que la medida a cualquier píxel dentro de la marca de medidas 10x10 cm tendrá menos de 1 mm de diferencia respecto a las demás de la marca, pues estará situado a menos de 10 cm del píxel central, A.

Estos cálculos se han realizado sobre una situación límite en la cual la marca está situada a 5 m del sensor, si la marca se aproxima al objetivo, la confianza en la medida aumentará aún más ya que al acercarse al sensor la variación entre píxeles próximos será menor.

Si se realizan medidas angulares esta aproximación realizada mediante del teorema de Pitágoras deja de ser válida, ya que el ángulo que se forma entre la marca y el haz procedente del sensor dejaría de ser recto. Debido a que se ha tomado un margen de error muy pequeño (1 mm) en primera instancia se puede considerar este tamaño como válido. Asimismo, a partir de la batería de ensayos que se explicará posteriormente, podemos deducir de manera empírica que este valor de ventana es correcto, ya que conseguimos unos buenos resultados con este tamaño de ventana.

A priori, el incremento del tamaño de la ventana es positivo, pues se realiza la media en más puntos de medida y se reducen los errores producidos por los “flying píxeles”, pero este beneficio se consigue únicamente cuando se mide sobre una superficie plana enfrentada al sensor. Si la superficie sobre la que se mide tiene una forma irregular o bien se trata de un plano con mucha inclinación respecto al sensor, si se tomase una ventana con un tamaño muy elevado no podríamos asumir que todos los píxeles que la componen se encuentren a la misma distancia del sensor. Por ello, es necesario encontrar una solución de compromiso entre el tamaño para asegurar que la ventana cubra una superficie lo suficientemente pequeña para que siempre se pueda suponer como plana, así como tomar el mayor número de píxeles posibles para reducir lo máximo el error.

El programa Distance PCD proporciona una gráfica en la cual se representan los valores medidos enfrentados a los valores de las distancias reales a la que se encuentra el objeto en cada medición,

por lo que resulta de gran utilidad a la hora de caracterizar el sensor y se utilizará en las pruebas que se realizarán a continuación.

Cuando se completa la primera medición, el programa pregunta si se desea guardar y dónde un fichero Excel con los resultados obtenidos, así como una imagen que contiene el gráfico que se representa a la derecha del programa, lo cual se actualizará en cada medición.

Si entramos más en detalle acerca de la programación de esta interfaz, se realizó creando una GUI de Matlab con la interfaz de la Figura 32 e incluyendo el grueso del programa en la “callback” del botón “calcular”.

Cada vez que se presiona este botón el programa pregunta qué fichero PCD es el correspondiente a la medida que se quiere realizar y una vez conocido, el script lee dicho fichero con la función específica para este tipo de archivos “pcread” donde se procesan las coordenadas como una matriz de tres columnas (x, y, z) y 320x240 filas, una para cada píxel.

Sabiendo que en el fichero PCD se ordenan los píxeles de tal manera que se sitúan los 320 píxeles de cada fila a continuación de los de la fila anterior, si quisiéramos conocer la información del píxel (5, 10) deberíamos buscar en el la fila $(10 \times 320) + 5$ del fichero PCD.

Esto es lo que hace el programa para cada uno de los 25 píxeles que componen la ventana alrededor del píxel central, guardando para cada uno de ellos el valor de las tres componentes del vector y calculando el módulo de ellas como:

$$r = \sqrt{x^2 + y^2 + z^2}$$

Una vez almacenado el valor de cada una de las distancias se calcula la media aritmética y la desviación típica de ellas.

Además, el programa representa en una misma gráfica la distancia medida y la distancia real superpuestas, mediante la función para representaciones discretas “stem”.

Asimismo, al finalizar la representación de cada medida, el programa guardará dicha figura mediante la función “saveas” y actualizará un Excel con los datos de la nueva medida mediante la función “xlswrite”

De esta manera se pueden obtener gráficos similares a lo que se mostraron en el capítulo anterior cuando se validaban los diferentes perfiles de calibración, pero realizando medidas en cualquier dirección, no necesariamente en medidas enfrentadas al sensor.

Este programa también permite de forma muy sencilla calcular la distancia exacta a un punto teniendo en cuenta la inestabilidad temporal de los píxeles. Sería suficiente con una vez conocido el pixel central en el programa Voxel Viewer, tomar en dicho programa varios ficheros PCD del mismo escenario y una vez hecho esto, con el programa Distance PCD introducir el mismo píxel en todas las mediciones abriendo los distintos ficheros PCD anteriormente obtenidos. Así conseguiríamos la máxima precisión posible en la medida.

El código fuente de este programa se encuentra adjunto en el Anexo.

3.3.2. Medidas en rayo directo

Ahora vamos a estudiar de manera exhaustiva el funcionamiento del sensor para el perfil definitivo “**FaseCal**”, teniendo en cuenta la inestabilidad temporal de los flying píxeles y tomando una ventana de píxeles en torno al punto central del objetivo a medir. Para ello se empleará el programa “Distance PCD” para la obtención de una medida más precisa, ya que tiene en cuenta, aunque idealmente serían nulas, las componentes x e y del vector de posición real de la marca (x, y, z).

Se van a realizar una serie de medidas en el mismo escenario que se utilizó para caracterizar los perfiles del fabricante del sensor (Figura 20), tomando varios ficheros PCD de cada medición, para tener en cuenta la inestabilidad temporal y obtener la mayor precisión posible en la caracterización.

Para ello, se ha decidido extraer a las distancias 1, 2, 3, 4 y 5 metros con programa Voxel Viewer la posición del píxel central y 10 ficheros PCD para cada una de esas posiciones. Con estos datos y el programa Distance PCD podemos obtener la medida exacta de cada una de las 5 mediciones.

El resultado de estas mediciones lo podemos ver en la Figura 34:

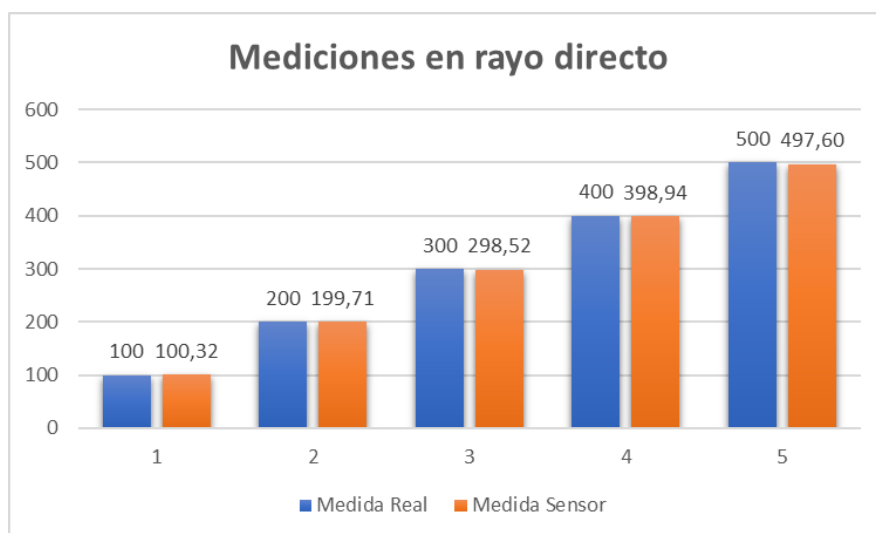


Figura 34 Caracterización en rayo directo

En la gráfica superior podemos ver cómo la medida ahora es más precisa y fiable que la obtenida anteriormente en la caracterización sin tener en cuenta la inestabilidad temporal.

Si se representa el error medio de las 10 mediciones temporales que se realizaron para la obtención de cada una de las 5 medidas finales representadas en la gráfica superior, obtenemos la gráfica que podemos ver en la Figura 35.

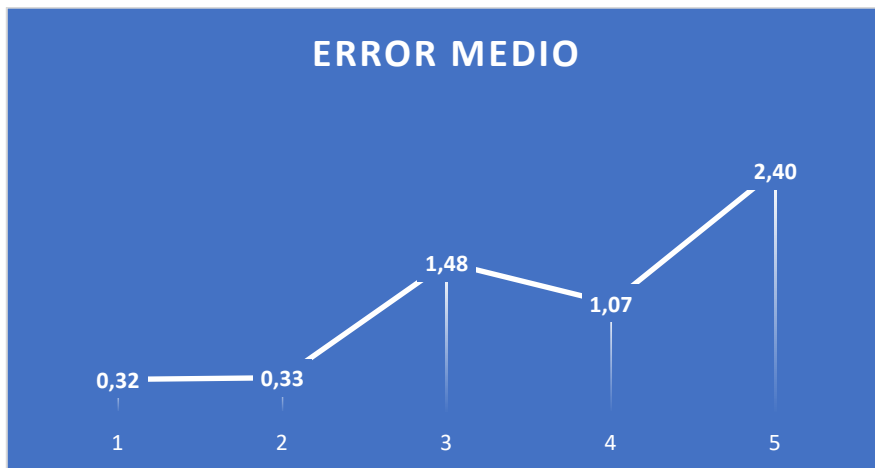


Figura 35 Error medio en rayo directo

En esta figura podemos ver cómo el error generalmente aumenta a medida que se realizan mediciones a objetos situados a más distancia, por ello, viendo estos resultados se estima que la distancia adecuada para situar las marcas que debe detectar el sensor, se situarán entre 3 y 4 metros, llegando a una situación de compromiso entre distancias lo suficientemente lejanas para que el sistema sea viable y la obtención de una precisión aceptable en las medidas y posterior localización.

Del mismo modo que se aprecia una evolución en el error de la medida en función de la distancia a la que se encuentra el objetivo, se puede ver como la desviación entre los píxeles es mayor a medida que la distancia a la que se realiza la medida aumenta. Esto se observa en la desviación típica entre las medidas de los 25 píxeles que forman la ventana del programa Distance PCD en cada medición. En la Figura 36 podemos ver dicha evolución.

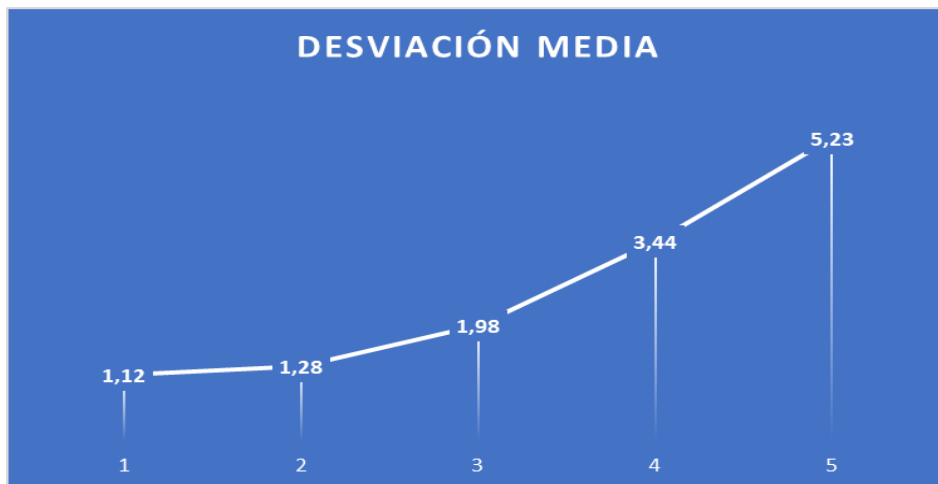


Figura 36 Desviación media en función de la distancia

3.3.3. Mediciones en ángulo

En este apartado vamos a estudiar la precisión del sensor cuando realizamos medidas a objetos que no se encuentran enfrentados al sensor. Para ello, se ha modificado el escenario de medidas anterior añadiendo un nuevo carril perpendicularmente a los dos carriles enfrentados que teníamos previamente, como se puede ver en la Figura 37:



Figura 37 Escenario de medidas angulares

Vamos a caracterizar el funcionamiento del sensor en función del ángulo en el que se encuentra la marca, para ello vamos a situar inicialmente la marca enfrentada con el sensor (0°) y se va a desplazar en pasos de 20 cm sobre el carril perpendicular, obteniendo la medida a la que se encuentra en cada posición.

El carril secundario se situará perpendicularmente en la posición de 350 cm del carril primario. Se ha considerado ésta la distancia máxima a la que se estima debería colocar las marcas en la celda de localización, situadas sobre unos soportes verticales desde el techo.

La obtención de la medida real a la que se encuentran situadas las marcas respecto del sensor se realizará mediante el teorema de Pitágoras, sabiendo que los dos carriles forman un ángulo de 90° , que el valor cateto que forma el carril primario estará fijo a 350 cm y que se aumentará el segundo cateto en pasos de 20 cm, el cálculo de la hipotenusa es inmediato.

En la Tabla 6 podemos ver el cálculo de las distancias reales de la prueba que vamos a realizar:

CATETOS PRIMARIO	CATETOS SECUNDARIO	HIPOTENUSA
350	0	350,00
350	20	350,57
350	40	352,28
350	60	355,11
350	80	359,03
350	100	364,01
350	120	370,00
350	140	376,96
350	160	384,84
350	180	393,57
350	200	403,11
350	220	413,40
350	240	424,38
350	260	436,00

Tabla 6 Distancias reales de prueba de medias angulares

Una vez conocidas estas distancias se irá desplazando el objeto por el carril secundario, extrayendo en cada posición la ubicación del píxel central y el PCD correspondiente. Una vez realizado esto, mediante el programa Distance PCD obtenemos la imagen comparativa (Figura 38) y los datos de las medidas.

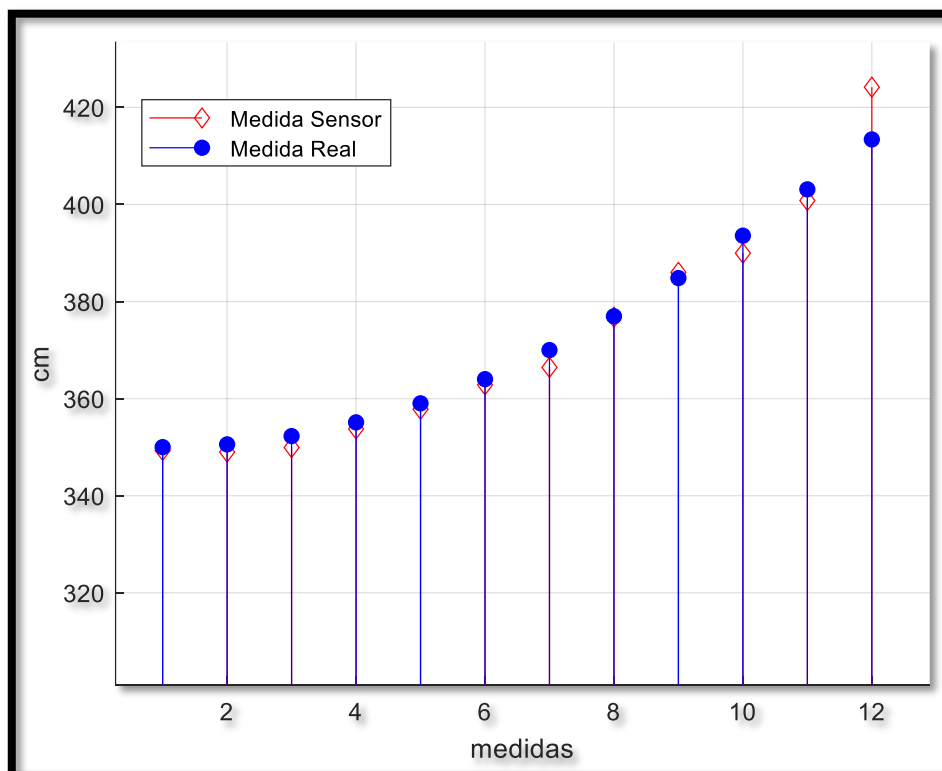


Figura 38 Caracterización de medidas angulares

A partir de la interfaz gráfica del programa Distance PCD, la cual podemos ver en la Figura 32, obtenemos una representación gráfica como la de la figura superior, así como el error medio y la desviación típica media de las medidas realizadas.

En estas medidas hemos obtenido un **error de 1.65 cm**, el cual como podemos ver en la figura superior se ve agravado en las últimas mediciones, las cuales son más exigentes en ángulo y distancia, pues nos acercamos a los límites de las especificaciones del sensor.

La desviación típica media obtenida entre las 25 medidas que componen cada medición mediante el programa Distance PCD, son todas inferiores a 10 cm, por lo que podemos considerar estas mediciones como válidas. Esto se debe a que si el objeto estuviese en una posición en la cual por algún motivo el pixel central de la ventana estuviese situado sobre una arista del objeto, las distancias medidas entre todos los píxeles que componen la medida no podrían considerarse equivalentes y por lo tanto ante una desviación típica elevada, la medición debería descartarse.

Este efecto se ve con claridad si se realiza una caracterización del sensor teniendo en cuenta la inestabilidad temporal de los píxeles. Para ello, de forma similar al procedimiento seguido para el caso de las mediciones en rayo directo, se va a colocar la marca en las posiciones del carril secundario a 0 cm, 80 cm, 140 cm, 220 cm, estando siempre éste perpendicular al primario en 350 cm.

Los resultados obtenidos los podemos ver en la Figura 39:

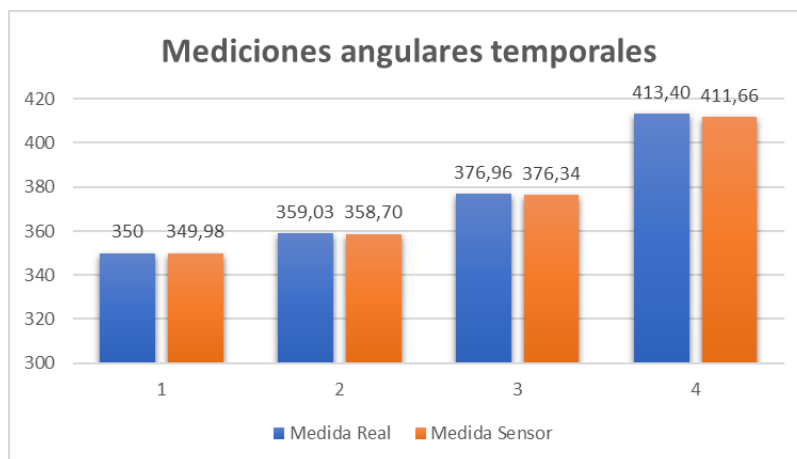


Figura 39 Caracterización angular temporal

En este gráfico podemos ver representado el resultado de la media de cada una de las 10 medidas temporales para las 4 distancias seleccionadas. **Se ha obtenido un error medio total de 0.79 cm y una desviación media de las medidas que se han realizado en las ventanas de medida de 6.81 cm.** Los resultados obtenidos son peores de los esperados, para encontrar la justificación observaremos la evolución del error y desviación en función de la distancia de medida en la Figura 40 y Figura 41.

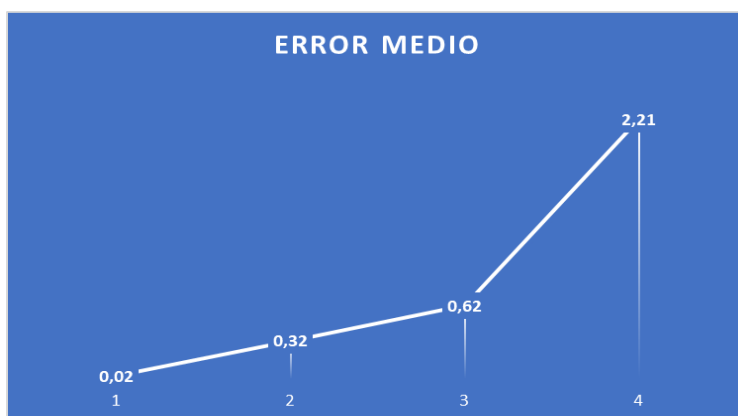


Figura 40 Evolución del error en caracterización temporal



Figura 41 Evolución de la desviación típica en caracterización temporal

En estas gráficas podemos ver cómo el error y la desviación típica obtenidos en la cuarta medida son excesivamente elevados, lo que afecta considerablemente de manera negativa en el error medio final del sensor.

Mdedida sensor	Medida real	Error	Desviacion
406,0373	413,4005	7,3633	6,4486
412,6148	413,4005	0,7858	8,0138
410,6825	413,4005	2,7180	8,4119
408,3501	413,4005	5,0504	7,8537
411,2875	413,4005	2,1130	8,3201
411,6877	413,4005	1,7129	7,8981
409,4626	413,4005	3,9379	5,9142
423,0958	413,4005	9,6952	70,4472
408,6624	413,4005	4,7381	7,1877
410,0579	413,4005	3,3426	9,1723
PROMEDIOS			
411,1939	413,4005	4,1457	13,9668
Error medio	2,2067		

Tabla 7 Medidas para caso a 220 cm en riel secundario

Si observamos la Tabla 7 podemos ver como en la octava medida se ha producido una medida totalmente errónea. Esto se debe a que el algoritmo de la ventana de 25 píxeles que realiza el programa Distance PCD en ocasiones puede tomar medidas erróneas si dado el caso, como se comentó anteriormente el píxel central se toma en una arista o mide sobre flying píxels.

Por lo tanto, en situaciones como ésta el programa debe de ser capaz de detectar una desviación típica demasiado elevada y considerar dicha medida como un outlier y ser descartada.

En vistas de estos resultados, se perfeccionó el programa de tal manera que descarta toda medida con una desviación típica ≥ 10 cm.

Si ahora se excluye dicha medida de la medición total, en esa posición se obtiene un nuevo error de 3,53 cm, frente a los 2.2 cm (413.4 - 411.2 cm). El error final ahora empeora, pero es debido a que la medida que hemos eliminado era la única cuyo error era por exceso en lugar de por déficit como el resto de las medidas y compensaba el error cometido por las demás, pero pese a que el resultado final, aparentemente era mejor, con una desviación típica de 70 cm no es correcto considerar la medida como válida.

Como conclusión, cabe destacar la alta precisión que se obtiene solapando las mediciones espaciales mediante el algoritmo de la ventana y las mediciones temporales extrayendo varios ficheros PCD.

Se puede observar como en la posición más crítica, la que se realizó en la posición de 220 cm en el riel secundario, lo que supone realizar la medida a $\sim 31.15^\circ$, es decir, muy cerca del límite del rango de visión del sensor (37° según el fabricante), pese a obtener algunas medidas temporales con errores muy elevados, como puede verse en la Tabla 7, donde se ha llegado a cometer un error de 7.3 cm con una desviación de 6.4 cm, se ha conseguido una medición final, promediando las 10 medidas temporales anteriores con un error de tan solo 2.2 cm.

Para mediciones sin tanta exigencia angular, la precisión que se obtiene es muy elevada, consiguiendo como se ve en la Figura 39 un error de 0.62 cm a $\sim 23.6^\circ$.

Experimentalmente se ha comprobado que el límite angular del sensor horizontalmente se encuentra en $\sim 33^\circ$, siendo inviable alcanzar los 37° que el fabricante especifica.

Por último, se realizó una prueba para observar el resultado del sensor realizando medidas en el plano vertical de la cámara, ya que todas las anteriores se realizaron para el plano horizontal, en el cual el rango angular es mayor.

Para ello, se realizó un ensayo compuesto por la medición de la distancia colocando la marca a una distancia real de 400 cm, sobre el riel secundario con la cámara rotada 90° . El resultado que se obtuvo fue una medición con un error medio de 2.26 cm, es decir, una precisión muy similar a la obtenida con la cámara en su posición natural, por lo que, aunque en esta posición el sensor se encuentra más limitado en rango angular, a 59° frente a los 74° que tiene en la posición horizontal, la precisión en las mediciones no se ve afectada.

3.4. Espejos Corner Cube

En este apartado vamos a estudiar el comportamiento de la cámara ante un tipo de marcas diferentes, los espejos Corner Cube, cuya peculiaridad principal es la capacidad de relejar el haz de luz incidente en él con la misma dirección (idealmente) que la dirección del rayo incidente, como podemos apreciar en la Figura 43. [14]

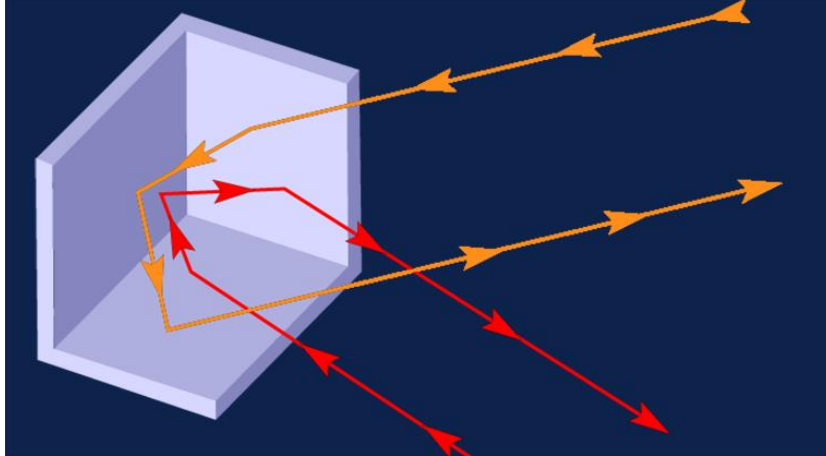


Figura 42 Funcionamiento espejo Corner Cube [14]

El objetivo de utilizar este tipo de marcas es aprovechar las características del sensor para intentar potenciar la señal que recibe. Como se vio en el primer capítulo, los sensores ToF emiten una señal, la cual se refleja en el objeto sobre el que se quiere medir su distancia, se recibe en el sensor y a partir de la diferencia de fase entre ambas se estima la distancia a la que se encuentra dicha marca.

Como se comentó en el apartado sobre los errores de este tipo de sensores, al reflejar la señal sobre la marca se produce una pérdida de energía debida a la absorción y dispersión de una parte de ésta sobre ella. Por eso, se plantea como método de mejora de prestaciones la implementación de los espejos Corner Cube como marcas, ya que, al aprovechar su característica de reflejar toda señal incidente sobre él en la misma dirección, idealmente se reducirán las pérdidas y se podrá detectar a mayor distancia y con mayor precisión.

Para realizar estas pruebas se han adquirido dos Corner Cube de 64 mm de diámetro, como el modelo de la Figura 43.

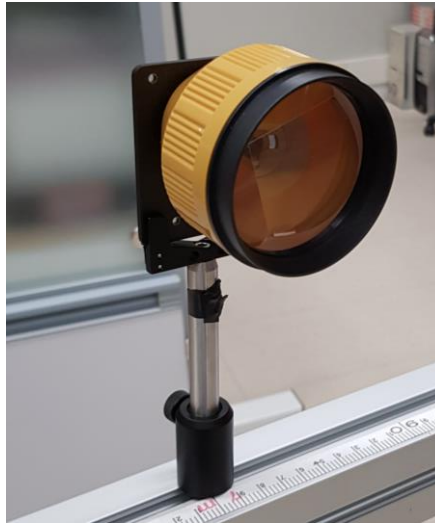


Figura 43 Corner Cube

Se han realizado diferentes pruebas preliminares, apreciándose un comportamiento muy singular. En la Figura 44 podemos ver como en la imagen de fase se aprecia claramente dos óvalos azules, los cuales se tratan de los dos Corner Cube.

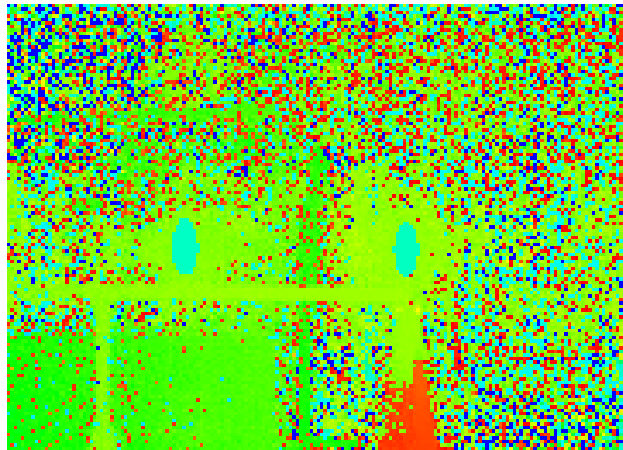


Figura 44 Diagrama de fase con dos Corner Cube

En el diagrama de fase podemos ver mediante colores una representación de la escena con información de la distancia a la que se encuentran los objetos en ella. Los objetos que se encuentran más próximos se representan con color rojo y a medida que se alejan con colores amarillos, verdes y azules.

En la Figura 44 podemos ver como los espejos dan una información falsa de distancia, pues cambia su color a uno más lejano respecto a los objetos que se encuentran en su proximidad. Si con el programa queremos saber la distancia a la que se encuentran, introduciendo en él alguno de sus píxeles, veremos cómo para cualquier localización de los espejos en la escena la medida que nos indica el programa es de ~ 6.8 m.

Esto se debe a que el espejo devuelve tanta potencia que se aproxima al máximo de fase que puede medir, el cual es una vuelta completa de fase 2π .

A priori, esto imposibilita realizar mediciones sobre el espejo, ya que por mucha distancia a la que se coloque siempre devuelve la misma medida.

En la Figura 44 podemos ver cómo existe gran cantidad de píxeles con ruido en la imagen, en el fondo se aprecia multitud de píxeles con colores diversos muy próximos unos a otros, evitar este ruido es fundamental para obtener una medición exenta de ruido y es uno de los objetivos que se busca con el uso de este tipo de espejos. Como se puede ver en dicha figura, los óvalos azules que representan los espejos, aunque dan una medida irreal, la estabilidad es absoluta y aunque en una sola imagen no se puede apreciar esto, en tiempo real se ve como el fondo multicolor está cambiando continuamente, siendo una nube de puntos sin un color definido, mientras que en el espejo se tiene un óvalo azul donde no varía el color.

Para afrontar el problema de la excesiva potencia que devuelve el espejo e imposibilita la medida, se ha optado por colocar dos polarizadores lineales situados a 90° sobre el espejo, para restringir al mínimo la luz que entra y sale de él.

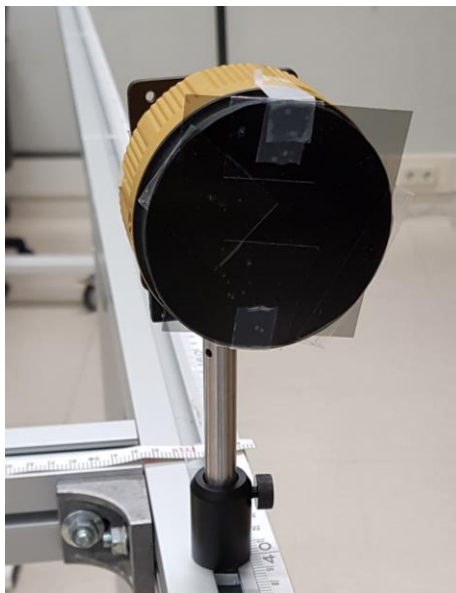


Figura 45 Espejo Corner Cube con polarizador a 90°

Una vez colocado el polarizador como puede verse en la Figura 45, se procede a conectar el sensor y estudiar los resultados obtenidos. Para ello compararemos la imagen de fase que se obtuvo sin el polarizador y con el mismo.

En la Figura 46 podemos ver la nueva imagen de fase donde se aprecia cómo el óvalo azul anterior se ha visto reducido y ahora vemos dos partes diferenciadas en él, un círculo con un color aproximadamente naranja y un pequeño óvalo azul dentro de este círculo.

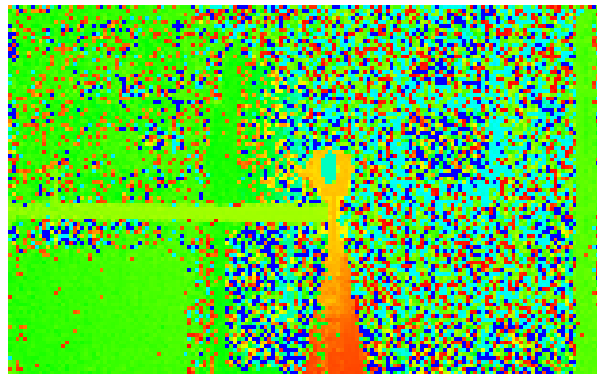


Figura 46 Diagrama de fase con Corner Cube con polarizador

Si se mide la distancia a la cual, según el programa se encuentra alguno de los píxeles del óvalo azul, de la misma forma que en el caso anterior el programa siempre devuelve una medida de ~6.8m.

En cambio, si se mide en alguno de los píxeles que forman el círculo naranja se obtendrá una medida relativamente cierta, pero con una precisión muy similar a la que se obtenía con una marca normal.

Los píxeles correspondientes al círculo naranja corresponden a trayectorias donde el rayo atraviesa el polarizador al igual que el resto, pero al estar situados en la periferia del espejo no sufren una concentración de luz tan grande como en el centro, por esta razón la potencia luminosa de los rayos con esta trayectoria se encuentra dentro del rango dinámico de la cámara, en una situación próxima a la saturación, pero sin llegar a ella.

A nivel de estabilidad temporal de estos píxeles consiguen una mejor resolución que en un único píxel sobre una marca normal, cercano al mismo orden de magnitud. Pero el hecho de que la selección de estos píxeles “no saturados” es muy compleja e inexacta, además de no ser muy numerosos, impide realizar algún tipo de algoritmo similar al algoritmo de la ventana que se realizó para el caso de marcas pasivas corrientes.

Por lo que, a nivel de marca para mejorar la precisión del sensor, no se ha conseguido unos mejores resultados que los conseguidos sobre marcas normales y la implementación del algoritmo de la ventana.

Debido a la gran estabilidad que presentan sus píxeles y el hecho de que ofrezcan una medida constante para cualquier situación de ellos en el entorno, se plantea su uso como método para agilizar la detección de las marcas en el escenario. Situando dicho espejo junto a la marca pasiva clásica. El propósito es primero detectar el espejo, el cual es fácilmente reconocible en la escena y posteriormente realizar la medida de distancia a la marca.

Para estudiar el comportamiento de este sistema, se colocaron el espejo y la marca anterior sobre el techo, simulando el sistema real, y con la cámara desde el suelo se hicieron una serie de medidas obteniendo de igual manera el píxel central de la cámara.

Los resultados obtenidos no fueron muy positivos, dado que la gran potencia que devuelve el espejo cuando se proyecta la luz del sensor sobre él, no solo afecta en ese óvalo de luz azul saturada que hemos comentado anteriormente, sino que falsea todas las medidas que se realicen

en su entorno cercano. Esto se comprobó, realizando mediciones a la marca con y sin el espejo situado junto a ella y se comprobó cómo la medida empeoraba al situarlo en su proximidad.

El espejo únicamente agiliza la detección de la marca cuando hacemos un reconocimiento de la escena de forma visual. Cuando se optimice el sistema para que detecte la marca en el entorno de forma automática, el programa deberá recorrer todo el mapa de píxeles y el tiempo de procesamiento será el mismo para detectar la marca que para detectar el espejo, lo que conllevaría posteriormente tener que detectar de nuevo la marca junto a él y haría que el sistema no fuese eficiente.

Durante el proceso de ensayos, se realizaron diversas pruebas modificando, en la medida que el programa permite, el hardware del sensor. Se modificaron principalmente los parámetros de “Integration time” y “Illumination Power” consiguiendo una escena en la cual todo el escenario se podría considerar ruido, excepto los espejos, que se reconocían con claridad y de forma estable. Incluso, reduciendo al máximo estos parámetros hardware se consiguió en algunos casos realizar medidas directamente sobre los espejos, pero nunca con una precisión lo suficientemente buena como para hacer frente a los resultados obtenidos a una marca convencional.

Por estas razones, se descartó temporalmente el uso de los espejos Corner Cube como método de localización mejorada de marcas, al menos en esta etapa del desarrollo.

3.5. Conclusiones

En primer lugar, en este capítulo se han extraído los parámetros más importantes del sensor para nuestra aplicación concreta.

Se ha estudiado las diferencias entre los distintos perfiles de calibración que el fabricante ofrece, concluyendo cuál de ellos es el más favorable, siendo el denominado “Metriculous Long Range”.

Una vez determinado cuál de los perfiles de fábrica es el mejor para nuestra aplicación, se han realizado varias calibraciones con el objetivo de mejorar las prestaciones de este. Primero, se realizó una calibración de la lente heredando el resto de los parámetros de dicho perfil y se observó que el resultado obtenido era peor que el de fábrica, por lo que se descartó este perfil. Posteriormente se realizó una calibración de la fase común del sensor, heredando de nuevo el resto de los parámetros del perfil que calibración del fabricante que mejores resultados ofrece, y en este caso se vio como se aumentó la precisión de este, por lo que se trabajará con este perfil, que se ha denominado “FaseCal”.

Con el sensor configurado con el perfil de calibración que mejores resultados ofrece, se procedió a realizar una caracterización más profunda del mismo, para estudiar la precisión que ofrece en diversos escenarios, obteniendo un resultado muy favorable de un error medio de 8 mm sobre 1000 medidas, realizando mediciones hasta 4 metros de distancia en cualquier dirección, pese a obtener una desviación típica media de 6.8 cm.

Se comprobó como el rango de medida horizontal del sensor es de 33° frente a los 37° que el fabricante indica, así como la influencia del ángulo de medida en la precisión de esta.

Por último, se ha estudiado el comportamiento del sistema sustituyendo marcas pasivas corrientes por un tipo peculiar de espejos, los Corner Cube, donde a pesar de teóricamente poder incrementar la precisión de las medidas y la longitud hasta la cual se pueden realizar, no se consiguió mejorar los resultados obtenidos previamente con marcas normales, por lo que descartó su uso.

4. Capítulo 4: Posicionamiento

En este capítulo se va a tratar la integración del sistema anterior en un escenario real de medida.

Se va a caracterizar la celda de posicionamiento y observar la precisión media que se obtiene en ella, primero la precisión hasta las marcas de posicionamiento y posteriormente ver cómo estos errores influyen en la posición final obtenida por el sistema de localización basado en el principio de trilateración esférica que se trató en el apartado 2.1.

Se va a detallar cual es el proceso por seguir hasta obtener posición del robot, partiendo desde cómo se obtienen las distancias a las que se encuentran las marcas del sensor, hasta la obtención de la posición final.

4.1. Detección de tres marcas

Las principales limitaciones del sistema son las limitaciones del propio sensor, el rango de visión y la precisión. Pues para que el sistema sea viable, el sensor tiene que ser capaz de reconocer en todo momento tres marcas de posicionamiento que nos permita obtener la posición final con la precisión adecuada.

El rango de visión del sensor es de 87° en diagonal (74° en horizontal y 59° en vertical). Debido a que el rango vertical es el más restrictivo y el robot con la cámara puede rotar en cualquier dirección. Las marcas deben poder reconocerse siempre dentro de los 59° de visión. Este rango de medida se ha comprobado que es levemente inferior al indicado por el fabricante, pues situando una marca sobre el riel secundario del escenario que se utilizó para la caracterización de las medidas en ángulo, situando el sensor rotado 90° y la cámara en el extremos del campo de visión, se observó que la distancia a la que se encontraba en el riel secundario (cateto 2) era de 193 cm, por lo que como se puede ver gráficamente en la Figura 47 el semi-rango de visión es de $\alpha = \arctg\left(\frac{\text{cateto } 2}{\text{cateto } 1}\right) = 29^\circ$. Por lo que el rango de visión será de $29 \times 2 = 58^\circ$

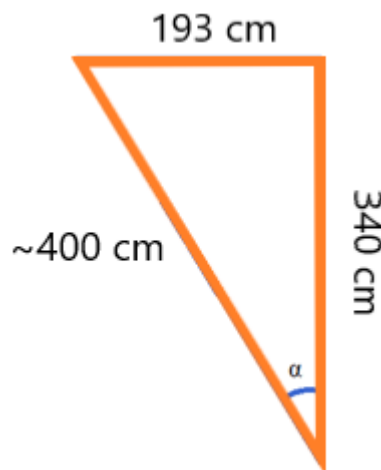


Figura 47 Semi-rango de visión con sensor rotado

Debido a esto, el sensor deberá ser capaz reconocer las tres marcas estando situado en los límites de la celda de posicionamiento, bajo su semi-rango de 29° .

Para realizar un ensayo del sistema se colocaron tres marcas simulando el escenario más crítico que el escenario de mediciones angulares permite. Una marca enfrentada al sensor, otra cercana al extremo del rango de visión con el sensor rotado 90° y la tercera situada en el punto medio de las anteriores, como podemos ver en la Figura 48.

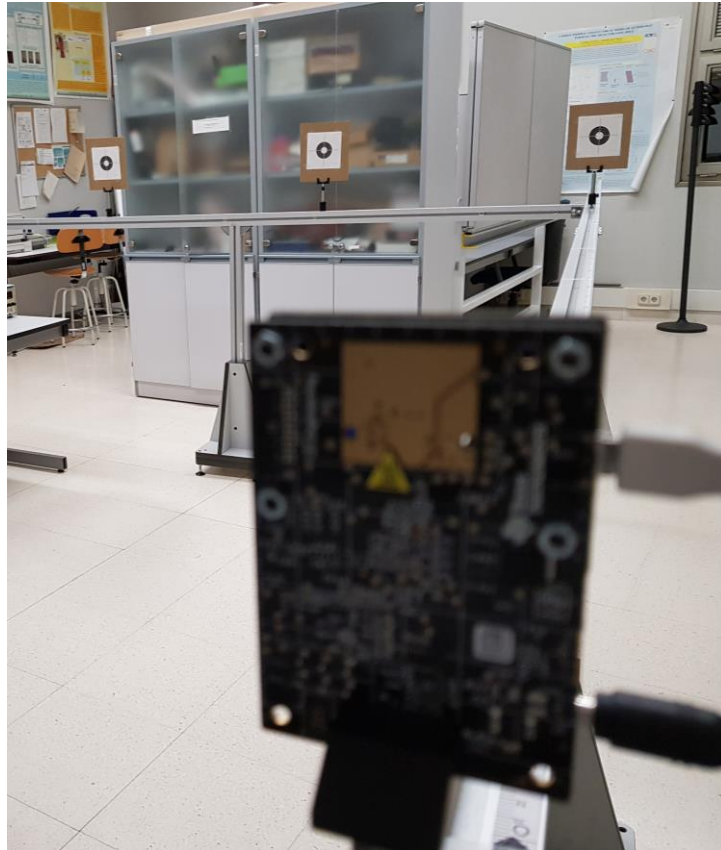


Figura 48 Escenario con tres marcas

Una vez situado el escenario, se conecta el sensor y mediante el programa Voxel Viewer se obtienen las coordenadas centrales de cada una de las tres marcas, así como 10 ficheros PCD que contienen la nube de puntos del entorno. Con estos datos, a través del programa Distance PCD se obtiene la distancia media a la que se encuentra cada una de estas marcas, teniendo en cuenta tanto la inestabilidad temporal como espacial de los píxeles.

Los resultados obtenidos se mostrarán en las siguientes tablas, una por medición.

Marca 1 - 00 cm				Marca 2 - 100 cm			Marca 3 - 200 cm		
No. Medida	Mdedida sensor	Error	Desviacion	Mdedida sensor	Error	Desviacion	Mdedida sensor	Error	Desviacion
1	340,55	0,55	4,57	355,88	1,48	7,90	392,09	2,37	14,07
2	341,87	1,87	4,56	355,60	1,20	7,36	395,13	0,67	10,18
3	341,31	1,31	4,62	355,09	0,69	7,58	379,86	14,60	69,68
4	338,82	1,18	4,44	355,25	0,85	7,45	395,18	0,71	9,88
5	341,19	1,19	3,71	358,19	3,79	8,38	406,37	11,91	68,47
6	340,81	0,81	5,12	359,78	5,38	9,15	389,25	5,21	8,09
7	340,44	0,44	5,33	357,57	3,17	7,44	405,32	10,85	68,02
8	342,24	2,24	4,72	357,27	2,87	8,12	400,00	5,53	9,03
9	340,51	0,51	4,03	355,49	1,09	6,01	430,37	35,90	113,85
10	341,23	1,23	3,56	360,34	5,94	7,56	404,18	9,72	142,74
Medida real	340			Medida real	354,40		Medida real	394,46	
Medida media	340,90			Medida media	357,05		Medida media	394,89	
Error medio	0,90			Error medio	2,64		Error medio	0,43	

Tabla 8 Distancias de escenario con tres marcas

En la Tabla 8 podemos ver las distancias que se han obtenido en cada una de las tres mediciones y el error medio en cada una de ellas. En la medición con más inclinación se puede ver cómo el algoritmo de la ventana más frecuentemente toma valores erróneos y es necesario descartar la medida por tener una desviación típica mayor de los 10 cm.

Estos valores medios serán los que tomemos para calcular la posición del robot. La intersección de las dos primeras esferas formará una circunferencia y la intersección de esta con la otra esfera tendrá como resultado dos puntos, de los cuales únicamente uno se encontrará en el plano de la cámara (dato conocido).

De manera gráfica podemos ver el cálculo de la posición a la que se encuentra el robot en la Figura 49:

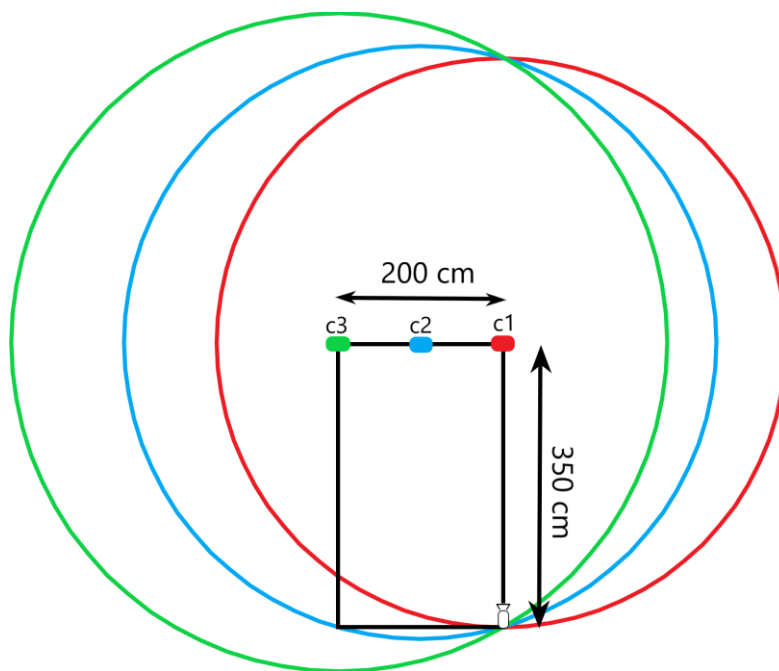


Figura 49 Localización gráfica del robot

En la Figura 49 se muestra una representación en la cual la cámara se encuentra situada en el suelo de la escena y las tres marcas sobre el techo a 350 cm del suelo y 340 cm del sensor. En dicha figura se aprecia cómo existen dos posibles soluciones del sistema, pero únicamente una de ellas se encuentra a la altura del suelo, la otra solución del sistema se encuentra fuera de celda de posicionamiento, con una altura teórica superior al techo.

En un caso bidimensional, dos circunferencias serían suficientes para obtener la posición deseada como se observa en la figura superior, pero dado que esta se trata de una vista frontal del sistema, en el caso real tridimensional serían necesarias tres marcas para poder obtener la solución al problema, partiendo de la premisa de la situación del robot sobre el suelo, en caso contrario serían necesaria una cuarta marca.

4.2. Celda de posicionamiento

Para poder conocer la posición en la que se encuentra el robot en la celda de posicionamiento, es imprescindible definir sus dimensiones y la localización de las marcas en ella.

Debido a que el rango de visión de la cámara a una distancia de 340 cm es en torno a 385x410 cm, si nos encontramos en una situación límite en los extremos de la celda de posicionamiento, como se trató anteriormente, el rango de visión de esta se verá reducido a la mitad, siendo idealmente 29°.

Debido a esto, podemos asumir que el sensor tiene que ver las marcas bajo el rectángulo de 193x205 cm, (Figura 50) siendo la parte más restrictiva la apotema menor, ya que la distancia desde el centro hasta los vértices es mayor que la apotema, esto hace que no se puedan colocar las marcas cercanas a dichos vértices y a una distancia mayor desde el centro del cuadrado a la de la apotema, ya que en el caso de la cámara rotara sobre sí misma podría no ser capaz de “ver dichas marcas”. Por ello, es la circunferencia inscrita en este cuadrado de 193x193 cm la que define el área de posicionamiento donde siempre el robot estará localizado y donde deberán estar situadas las marcas en nuestro sistema.

Para el caso de situar las marcas a una altura diferente, el cálculo de este cuadrado se hallaría de forma similar a el caso de la Figura 47 donde la altura eran 340 cm y el lado 193.

$$\operatorname{tg} 28.95^\circ = \frac{\text{lado}}{\text{altura}} \rightarrow \text{lado} = \operatorname{tg}(28.95^\circ) \times \text{altura}$$

Esto lo podemos ver de manera gráfica en la Figura 50 donde se representa el rectángulo de 320x240 píxeles de la cámara y el círculo cuyo radio es igual a la mitad del lado definido anteriormente (en el caso anterior 193 cm), calculado bajo el ángulo de visión de la cámara en las condiciones más restrictivas posibles y la cámara situada a 340 cm del suelo.

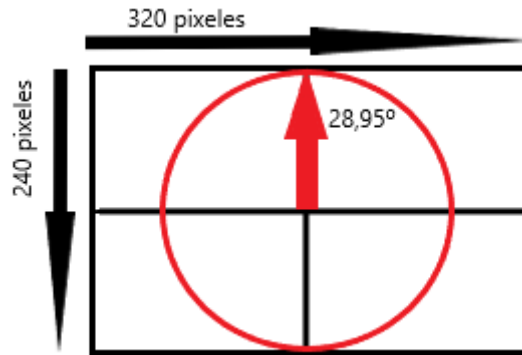


Figura 50 Definición de área mínima

Por lo tanto, el espacio de visión mínimo garantizado del robot será el círculo representado en la figura superior. Debido a que en este espacio la cámara debe ser capaz de detectar tres marcas, se situaran de tal manera que definan un triángulo equilátero en dicho círculo, siendo una marca cada uno de los tres vértices del triángulo.

La agregación de la Figura 51 sería una buena implementación de celda de posicionamiento en vista aportar una buena escalabilidad, ya que con la inclusión de una única marca más en el sistema se crearía una celda de posicionamiento adicional, de tal manera que de forma similar a un panel de abeja (Figura 51) podría cubrirse cualquier espacio para localizar al robot en su interior.

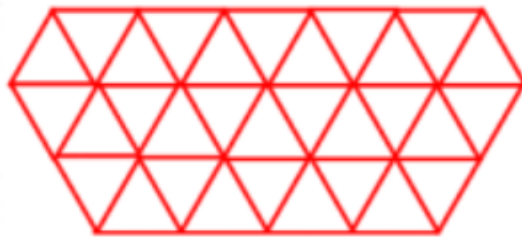


Figura 51 Estructura teselar

De esta manera, garantizamos que aun encontrándose el robot en las zonas límite del escenario total de posicionamiento, siempre será capaz de reconocer las tres marcas mínimas que necesita para posicionarse. En los casos no límite, el robot detectará varias marcas en el entorno, pero siempre tomará como referencia las tres más cercanas, puesto que serán las que ubiquen al robot en la celda que estas delimitan.

El lado del triángulo inscrito en el círculo de la Figura 52 se puede calcular como:

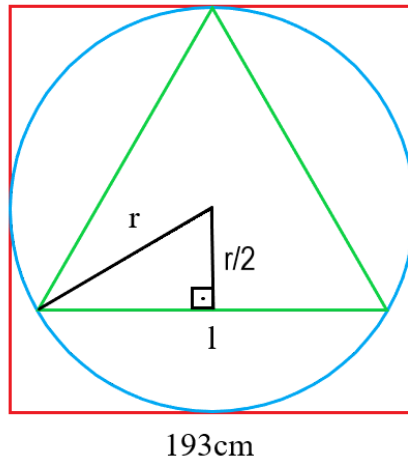


Figura 52 Cálculo gráfico del triángulo de posicionamiento

Sea el radio de la circunferencia (r) igual $1/2$ del cuadrado, es decir, $193 / 2 = 96.5$ cm. El lado del triángulo se puede calcular como:

$$l = 2 \times \sqrt{96.5^2 - \frac{96.5^2}{2}} = 167.143 \text{ cm}$$

Por lo tanto, la celda de posicionamiento quedaría definida por estos triángulos equiláteros de lado 167.143 cm, como se puede ver en la Figura 53.

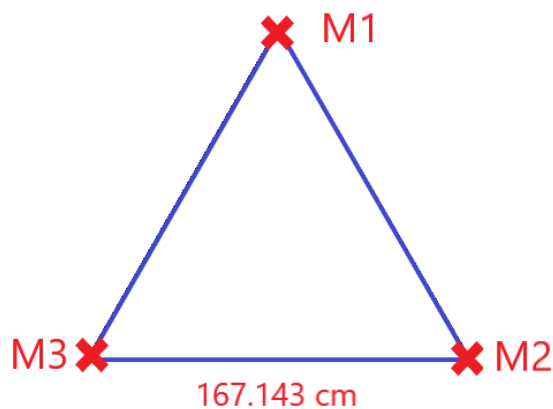


Figura 53 Celda de posicionamiento

5. Conclusiones y líneas futuras

En este último capítulo se pretende resumir las conclusiones que se han extraído durante cada una de las etapas de este proyecto, así como dar una orientación de los próximos pasos que se deberían seguir para continuar avanzando con este proyecto y aportar algunas posibles ideas para ello.

5.1. Conclusiones

Este trabajo se ha centrado en el estudio de la utilización de cámaras ToF con fines de localización en espacios inteligentes.

Tras un estudio general de este tipo de dispositivos, los tipos que existen y las alternativas del mercado, se estudió detalladamente el modelo concreto con el que se iba a trabajar, un modelo de la compañía Texas Instrument. En este proceso se observaron las limitaciones del sistema y cómo algunas de ellas se pueden paliar mediante procesos relativamente sencillos de calibración y configuración del software. Otras limitaciones, como los ángulos de visión de la cámara, se podrían aumentar mediante la modificación de la lente del sensor, lo cual, al tratarse de un modelo para desarrollo, podría ser una mejora interesante que realizar en futuras etapas del proyecto.

En las primeras etapas del desarrollo se caracterizaron los distintos perfiles de calibración que el fabricante ofrece con el dispositivo. Para ello se realizó una serie de mediciones comparando la medida real a la que se encontraba la marca en cada caso con la medida que indicaba el sensor. De estas pruebas se extrajo cuál de estos perfiles era el idóneo para nuestra aplicación.

Una vez obtenido el perfil de fábrica que ofrece mejores resultados, se procedió a calibrar ciertos aspectos de este para intentar aumentar la precisión obtenida. Se realizó una calibración de la lente siguiendo los pasos que el fabricante indica para ello y se compararon los resultados obtenidos, donde se observó una precisión inferior que con la calibración de la lente de fábrica. Por otro lado, se calibró la fase común del sensor obteniendo en este caso una mejora en la precisión de las medidas del dispositivo, por lo que el perfil con estas modificaciones será la configuración definitiva usada por el sensor en nuestra aplicación.

Una vez conseguida la configuración final del dispositivo se realizó una batería de pruebas exhaustivas para caracterizar el sensor, tanto en mediciones en ángulo como en rayo directo. En estas pruebas se ha obtenido una precisión final en torno a los 8 mm de error de media sobre 1000 medidas realizadas entre 1 y 4 metros de distancia y recorriendo todo el rango de visión del sensor.

Para poder realizar de manera eficiente estas medidas, se desarrolló un programa denominado Distance PCD, capaz de obtener a partir del fichero PCD con la nube de puntos de la escena y el píxel central del objeto, la media y la desviación típica de la distancia a la que se encuentra dicho píxel y los 25 que le rodean. Este algoritmo ha sido denominado método de la ventana, demostrando una mejora en la precisión de la medida respecto a la medida puntual. Teniendo en cuenta que la inestabilidad de la medida de los píxeles influye notoriamente en el sistema, de esta forma se consigue paliar en cierta medida los efectos de la fluctuación en la precisión del sistema. Del mismo modo que con este algoritmo se reducen los errores debidos a la inestabilidad espacial de los píxeles, mediante la obtención y promediado de la medida de 10 ficheros PCD se consigue reducir enormemente los errores debidos a la inestabilidad temporal de los píxeles voladores, respecto de la medida basada en un único PCD.

Con este algoritmo se consigue mejorar la precisión de las medidas drásticamente, esto se puede ver si se compara la distancia final que se obtiene como media de las 25 medidas con la desviación típica de ellas. A modo de ejemplo, en un caso típico se obtiene una desviación típica de cada medición en torno a 6.8 cm mientras que la precisión final puede situarse en torno a 8 mm.

Es importante tener en cuenta que este algoritmo puede llevarnos a error cuando el píxel central de la medición se encuentra sobre una arista o con una inclinación demasiado alta como para poder suponer que todos los píxeles de la ventana se encuentran a la misma distancia. Estos errores se corrigen tomando como erróneas aquellas mediciones cuya desviación típica sea mayor de 10 cm.

Las mediciones anteriores se realizaron sobre marcas pasivas corrientes, es decir, un objeto diseñado para reflejar la luz en todas las direcciones. Se probó el funcionamiento de un tipo especial de objeto, los denominados Corner Cube, un tipo de espejo cuya peculiaridad principal es la capacidad de reflejar el haz de luz incidente en él con la misma dirección (idealmente) que la dirección del rayo incidente. Esta característica se suponía que podía mejorar las características del sistema debido a que reduciría las pérdidas debidas a la reflexión Lambertiana del rayo infrarrojo que emite el sensor cuando incide sobre las marcas pasivas convencionales. Desafortunadamente el uso de este tipo de espejos no ha sido tan beneficioso como se esperaba, el espejo devuelve demasiada potencia al sensor, llegando a saturar los receptores e imposibilitando la obtención de una medición válida. Este problema se abordó por diferentes vías, instalando un sistema de polarizadores lineales cruzados a 90° en la base del espejo para reducir la energía que este devuelve y reduciendo parámetros del sensor vía software, con estas modificaciones en algunos casos se consiguió llegar a realizar medidas sobre el espejo, pero siempre muy lejos de la precisión obtenida con las marcas pasivas corrientes, por lo que su uso en esta aplicación fue descartado, al menos, en esta etapa del desarrollo.

Una vez definida la configuración del sensor, el tipo de marcas que se va a utilizar y la precisión que se obtiene en el sistema, el reto fue validar la posibilidad de realizar localización del sensor a partir de tres marcas. Para ello se situaron las marcas en la escena, se obtuvieron las distancias al a que se encuentra con el sensor y geoméricamente se obtuvo el punto final de localización. Esto es una primera etapa del desarrollo, donde no se pretende llegar a un posicionamiento dinámico real, sino validar el sensor para dicho uso.

Por último se ha definido un modelo de celda de posicionamiento con tres marcas, el cual cumple con todas las limitaciones del sensor de forma que el robot siempre pueda estar posicionado dentro de ella. La geometría de esta está diseñada para un funcionamiento óptimo en un uso escalado con multitud de dichas celdas, no como celda única de posicionamiento. Aunque un sistema real donde se implementase este escenario lo óptimo sería realizar un estudio concreto del posicionamiento de las marcas para llegar a el mejor compromiso entre número de marcas y cobertura conseguida.

5.2. Líneas futuras de desarrollo

Este proyecto se considera un punto de partida del sistema completo de localización en interiores basado en cámaras ToF, en el cual se han realizado grandes avances, pero lejos de alcanzar el sistema final.

Durante la elaboración de este trabajo se han planteado diversas vías de desarrollo futuras que no se han podido abarcar en este proyecto, las cuales se pretenden resumir en este punto con la finalidad de que puedan servir de guía para continuar esta investigación.

5.2.1. Obtención de medidas en tiempo real

Como se ha estudiado en este proyecto, la inestabilidad de la medida que producen los flying píxels repercute notoriamente en la precisión de la medida que obtiene el sistema. En este trabajo este problema se ha paliado tomando 10 ficheros PCD para la obtención de cada medida, realizando la media de la distancia ofrecida por el sensor en cada uno de estos 10 instantes temporales.

Al tomar estos 10 instantes temporales se ha demostrado que el error cometido en las medidas se reduce enormemente, pero para diseñar un sistema lo más eficientemente posible lo óptimo sería poder tomar no 10 instantes, sino realizar un cálculo continuo de la distancia a la que se encuentran las marcas en la escena.

Esto además de reducir los errores cometidos dichos píxeles será necesario para que el robot pueda moverse por el escenario sin la necesidad de detenerse para hallar las distancias a la que se encuentran las marcas y realizar el cálculo de su posición.

El primer paso para trabajar con el sensor ha sido la obtención de distancias y posiciones sobre “imágenes estáticas” de la escena, pero el siguiente paso para el sistema final es poder medir y las distancias y localizar el robot sobre “imágenes dinámicas” es decir, en tiempo real.

Este es uno de los motivos principales por los que se ha utilizado cámaras de tiempo de vuelo en lugar de cámaras convencionales, la obtención de las distancias por el sensor es mucho más rápida y los programas de procesamiento de las “imágenes” trabajan con algoritmos de procesamiento de datos mucho más rápidos que los algoritmos de procesado de visión.

5.2.2. Detección de marcas

En este proyecto el proceso de detección de las marcas ha sido de manera puramente manual, detectando en la imagen de fase la marca y obteniendo con el programa que proporciona el fabricante las coordenadas del píxel central de la marca.

En un proceso de tiempo real como el comentado anteriormente es inviable la idea de que el usuario tenga que detectar de forma manual la marca, por ello, en siguientes pasos de desarrollo será necesario automatizar este proceso para realizar la detección de las marcas de forma automatizada.

Existen multitud de formas de abordar este reto. La forma que a priori parece más directa y la cual sería la siguiente línea de desarrollo, sería mejora del programa creado para la obtención de la distancia de los ficheros PCD, para que mediante la realización de un barrido por todo el mapa

de 320x240 píxeles, detecte la una zona de píxeles cuya medida se encuentra en la región esperada de la marca y con una variación de distancia muy leve entre ellos.

Esto, aunque pueda parecer muy complejo, en realidad no lo es tanto si se tienen en cuenta ciertas características del sistema. La cámara siempre estará situada sobre el robot orientada hacia el techo, donde estarán situadas las marcas orientadas hacia el suelo, por lo que el sensor y las marcas se encuentran enfrentados entre sí. Esto hace que la visión que tenga la cámara de las marcas sea la de una forma muy aproximada a la forma real del objeto, con las leves deformaciones que se produzcan al ver la marca con una cierta proyección cuando no se encuentre situada perpendicularmente respecto del eje óptico de la lente.

La situación de las marcas no se realizará exactamente en el techo, sino que situadas a 3.50 m del suelo sujetas desde el techo mediante unos postes verticales. Esto hace que la detección de la marca sea mucho más sencilla, ya que el contraste de la distancia a la que se encuentran con la distancia del techo hace posible detectar las mismas de forma muy directa.

Por lo tanto, el algoritmo deberá recorrer el mapa de píxeles comparando la distancia a la que se encuentra cada uno de ellos con los que le rodean y detectar una región de píxeles donde la desviación típica de sus medidas sea baja en relación con el resto del escenario.

De esta forma podrían detectarse marcas en el techo, pero existiría el problema de poder diferenciarlas entre sí. Una posible solución a este problema sería dotar a las tres marcas de formas conocidas simples, de manera que fuese relativamente sencillo buscar una región circular, cuadrada o triangular de distancias comunes en la imagen.

5.2.3. Modificación de la lente

Una de las limitaciones principales del sistema es el rango de visión del sensor, esta limita la zona en la cual se deben situar las marcas para que la localización sea posible, por lo tanto, define directamente las dimensiones de la celda de posicionamiento.

Una mejora en el sistema sería la sustitución de la lente del dispositivo de manera que se amplíe el rango de visión, penalizando lo mínimo posible la precisión de las medias.

Bibliografía

- [1] "Trilateración", *Es.wikipedia.org*, 2018. [Online]. Available: <https://es.wikipedia.org/wiki/Trilateraci%C3%B3n>. [Accessed: 27- Dec- 2018].
- [2] "¿Cómo funcionan los dispositivos GPS? Trilateración vs Triangulación", El blog de franz, 2018. [Online]. Available: <https://acolita.com/como-funcionan-los-dispositivos-gps-trilateracion-vs-triangulacion/>. [Accessed: 27- Dec- 2018].
- [3] Ocw.upm.es, 2018. [Online]. Available: http://ocw.upm.es/ingenieria-cartografica-geodesica-y-fotogrametria/topografia-ii/contenidos/Mis_documentos/Tema-9-Triangulacion-y-Trilateracion/Teoria_Triang_Tema_9.pdf. [Accessed: 27- Dec- 2018].
- [4] Ugm.org.mx, 2018. [Online]. Available: <https://www.ugm.org.mx/publicaciones/geos/pdf/geos14-2/trilateracion-34-2.pdf>. [Accessed: 27- Dec- 2018].
- [5] "Luz estructurada", *Es.wikipedia.org*, 2018. [Online]. Available: https://es.wikipedia.org/wiki/Luz_estructurada. [Accessed: 27- Dec- 2018].
- [6] "Tutorial_Coded_Light_Projection_Techniques 1", *Eia.udg.es*, 2018. [Online]. Available: http://eia.udg.es/~qsalvi/Tutorial_Coded_Light_Projection_Techniques_archivos/v3_documento.html. [Accessed: 27- Dec- 2018].
- [7] *Fofi.pagesperso-orange.fr*, 2018. [Online]. Available: http://fofi.pagesperso-orange.fr/Downloads/Fofi_EI2004.pdf. [Accessed: 27- Dec- 2018].
- [8] 2018. [Online]. Available: https://www.researchgate.net/figure/Time-of-Flight-camera-principle_fig4_260432866. [Accessed: 27- Dec- 2018].
- [9] D. Jiménez-Cabello, Correction of errors in time of flight cameras, Tesis Doctoral. Universidad de Alcalá de Henares, 2015.
- [10] *Dfki.de*, 2018. [Online]. Available: https://www.dfki.de/fileadmin/user_upload/import/8767_wasenmuller2016comparison.pdf. [Accessed: 27- Dec- 2018].
- [11] Texas Instrument, "OPT8241 CDK Evaluation Module Quick Start Guide", 2015. [Online]. Available: <http://www.ti.com/lit/ug/sbou156b/sbou156b.pdf>. [Accessed: 27- Dec- 2018].
- [12] Texas Instrument, "OPT8241 Evaluation Module User's Guide", 2015. [Online]. Available: <http://www.ti.com/lit/ug/sbou155b/sbou155b.pdf>. [Accessed: 27- Dec- 2018].

- [13] Texas Instrument, "Voxel Viewer User's Guide", 2015. [Online]. Available: <http://www.ti.com/lit/ug/sbou157/sbou157.pdf>. [Accessed: 27- Dec- 2018].
- [14] Tech Blog: Novel Cube-Corner Retro-Reflector Increases Accuracy in Long Distance Measuring | Tech Launch Arizona", Techlaunch.arizona.edu, 2018. [Online]. Available: <https://techlaunch.arizona.edu/news/tech-blog-novel-cube-corner-retro-reflector-increases-accuracy-long-distance-measuring>. [Accessed: 27- Dec- 2018].
- [15] "Documentation - Point Cloud Library (PCL)", Pointclouds.org, 2018. [Online]. Available: http://pointclouds.org/documentation/tutorials/pcd_file_format.php. [Accessed: 27- Dec- 2018].
- [16] "Read 3-D point cloud from PLY or PCD file - MATLAB pcread- MathWorks España", Es.mathworks.com, 2018. [Online]. Available: <https://es.mathworks.com/help/vision/ref/pcread.html>. [Accessed: 27- Dec- 2018].
- [17] Texas Instrument, "OPT8241 3D Time-of-Flight Sensor datasheet", 2015. [Online]. Available: <http://www.ti.com/lit/ds/symlink/opt8241.pdf>. [Accessed: 27- Dec- 2018].
- [18] "Utilice las imágenes para mejorar las capacidades robóticas | DigiKey", Digikey.es, 2018. [Online]. Available: <https://www.digikey.es/es/articles/techzone/2016/may/use-infrared-time-of-flight-imaging-to-enhance-robotic-capabilities>. [Accessed: 27- Dec- 2018].
- [19] Texas Instrument, "Introduction to Time-of-Flight Camera", 2014. [Online]. Available: <http://www.ti.com/lit/wp/sloa190b/sloa190b.pdf>. [Accessed: 27- Dec- 2018].
- [20] Daniel Montagud Cortés, "Estudio de cámaras de tiempo de vuelo (ToF) para aplicaciones en exteriores", Trabajo Fin de Grado, Universidad de Alcalá, 2018.
- [21] Raquel García Jiménez, "Detección y conteo de personas, a partir de mapas de profundidad cenitales capturados con cámaras ToF", Trabajo Fin de Grado, Universidad de Alcalá, 2015.
- [22] M. Hansard, S. Lee, O. Choi and R. Horaud, *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.
- [23] María Higuera Pinillos, "Desarrollo de una interfaz común para la configuración y adquisición de datos de cámaras de tiempo de vuelo", Trabajo Fin de Grado, Universidad de Alcalá, 2018.
- [24] David Fuentes Jiménez, "Diseño, implementación y evaluación de un sistema de conteo de personas basado en cámaras de tiempo de vuelo", Trabajo Fin de Grado, Universidad de Alcalá, 2016.
- [25] "Tiempo de vuelo", Bashny.net, 2018. [Online]. Available: <https://bashny.net/t/es/271646#>. [Accessed: 27- Dec- 2018].

Anexo – Código programa Distance PCD

```
function varargout = dist_PCD(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @dist_PCD_OpeningFcn, ...
                  'gui_OutputFcn',   @dist_PCD_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function dist_PCD_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = dist_PCD_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function entrada2_Callback(hObject, eventdata, handles)

function entrada2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function entrada1_Callback(hObject, eventdata, handles)

function entrada1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function entrada3_Callback(hObject, eventdata, handles)

function entrada3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```

function entrada4_Callback(hObject, eventdata, handles)

function entrada4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dist_Callback(hObject, eventdata, handles)

function dist_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton1_Callback(hObject, eventdata, handles)

pixel_X=str2double(get(handles.entrada1,'String'));
pixel_Y=str2double(get(handles.entrada2,'String'));
real=str2double(get(handles.dist,'String'));
ejec=pwd;
[file,path] = uigetfile('*.pcd');
if isequal(file,0)
    disp('User selected Cancel');
else
    disp(['User selected ', fullfile(path,file)]);
end

cd(path);
ptCloud = pcread(file);

%Definición de variables a usar
distancia=0;
persistent error_acumulado;
persistent medida;
persistent reales;
persistent n;
persistent aux;
persistent aux2;
persistent inicio;
persistent Nombre;
persistent ruta;

medida();
reales();
aux();
aux2();

% 'n' es un contador utilizado para las inicializaciones
if(n<20)
    n=n+1;
else
    n=1;
    inicio=1;
end

if(n==1)
    error_acumulado=0;
end

```

```

%Ventana de 5x5 pixeles: Al pixel (x, y) se le suma y
%resta 2 para crear los límites de la ventana
mayor_i=pixel_Y+2;
menor_i=pixel_Y-2;
mayor_j=pixel_X+2;
menor_j=pixel_X-2;

%Índices del bucle para obtener la distancia a cada pixel de la ventana
ind_i=menor_i:mayor_i;
ind_j=menor_j:mayor_j;

for i=1:length(ind_i)%Bucle del eje vertical
    for j=1:length(ind_j)%Bucle del eje horizontal
        %Fórmula que obtiene la línea del fichero PCD en la que está cada
        pixel
        coordenadas=(320*ind_i(i)+ind_j(j));
        %Obtenemos las medidas de los ejes (X,Y,Z) de cada pixel
        pixelx= ptCloud.Location(coordenadas,1);
        pixely= ptCloud.Location(coordenadas,2);
        pixelz= ptCloud.Location(coordenadas,3);
        %Realizamos el módulo para obtener la distancia
        potencias = pixelx*pixelx+pixely*pixely+pixelz*pixelz;
        %Almacenamos la distancia en una matriz del tamaño de la ventana (5x5)
        distancia(i,j)=sqrt(potencias);
    end
end

%Obtenemos la distancia (en cm) como la media de las 25 medidas que tiene la
matriz
dist_media=100*mean(mean(distancia));

%Calculamos la desviación típica (en cm) entre esas 25 medidas
des=100*std(distancia(:));

%Calculamos el error medio (en cm y en valor absoluto) de todas las medias que
se realicen
error_acumulado=error_acumulado+abs(real-dist_media);

%Guardamos los datos obtenidos de las 25 muestras de la ventana
aux(n)=abs(real-dist_media);
aux2(n)=des;
medida(n)=dist_media;
reales(n)=real;

%Mostramos resultados por pantalla
set(handles.medida,'String',dist_media);
set(handles.error_medio,'String',error_acumulado/n);
set(handles.error_medida,'String',abs(real-dist_media));
set(handles.desviacion,'String',abs(des));

%Representación gráfica
hold on
stem(medida,'diamondr')
hold on
stem(reales,'b','filled');
grid on
legend('Medida Sensor','Medida Real')

fig=figure;
hold on
stem(medida,'diamondr')
hold on
stem(reales,'b','filled');
grid on

```

```

legend('Medida Sensor','Medida Real')
xlabel('medidas');
ylabel('cm');

saveas(fig,'imagen.fig')%Guardado automático de la figura
close(figure(1));

%Conversión de vector fila a vector columna
medida_transpuesta=medida';
reales_transpuesta=reales';
error=aux';
desv=aux2';

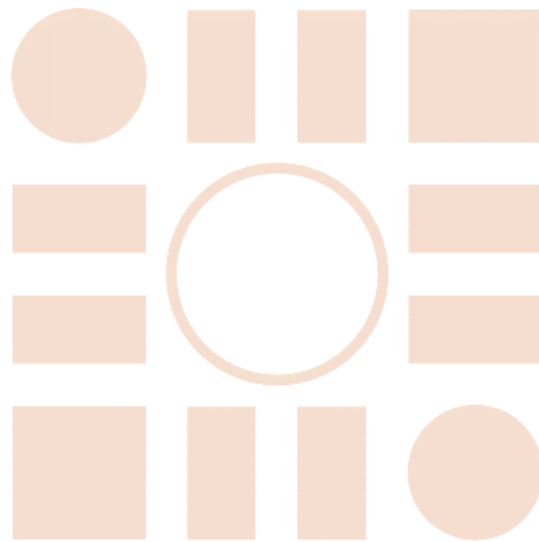
%Almacenamiento de datos en un fichero Excel
cab=["Mdedida sensor","Medida real","Error","desviacion"];
if(inicio==1)
    [Nombre, ruta] = uiputfile ({'*.xlsx'},'Guardar como');
    inicio=0;
end
xlswrite(fullfile(ruta, Nombre), cab);
matriz=[medida_transpuesta, reales_transpuesta, error, desv];
xlswrite(fullfile(ruta, Nombre), matriz, 'Hoja1', 'A2');
cd(ejec);

handles.n=n;
guidata(hObject,handles);
function figure1_CloseRequestFcn(hObject, eventdata, handles)
opc=questdlg('Desea salir del programa?','SALIR','Si','No','No');
if strcmp(opc,'No')
    return;
end
handles.n=1; %Inicializamos la variable n a 1
guidata(hObject,handles);
delete(hObject);

function uipushtool1_ClickedCallback(hObject, eventdata, handles)

```


Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá