UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

VISUAL LIGHT COMMUNICATION (VLC)

ON MOBILE PHONES.

APPLIED TO INDOOR LOCATION

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Marc Combalia

In partial fulfilment

of the requirements for the degree in

CITTEL ENGINEERING

Advisor: Prof Josep Paradells

Barcelona, October 2015

Abstract

This thesis will explore the operation of the tracking system in interior spaces that we propose using the technology of VLC (Visible Light Communication).

Through the flexibility of the new operational system Android 5.0 (with the Camera2 API) to control camera phone and four VLC lights in the field of view of the camera the mobile's position and orientation can be calculated. Using a processing data obtained from the lights we can obtain high location accuracy.

It will not be necessary to alter the lighting conditions of the room in which the system is implemented, so it will be invisible to its users, while superior to other interior location systems now available in the market.

Resum

La present tesis té com a objectiu exposar el estudi de un sistema de localització en espais interiors que s'ha desenvolupat fent servir tecnologia VLC (Visible Light Communication).

Per tal de comprendre millor el procés i els resultats de l'estudi, és necessari saber que es calcula la orientació i posició del dispositiu utilitzant únicament fotometria. Per aconseguir-ho, s'aprofita la flexibilitat que proporciona el sistema 5.0 de Android (que incorpora la API Camera2) al control de la càmera del mòbil i es disposen 4 panells LED transmetent informació mitjançant VLC en el camp de visió de la càmera.

A més, s'ha d'aclarir que no serà necessari alterar les condiciones de il·luminació de la sala en la que s'implementi el sistema. Es mostra superior a la resta de sistemes de localització interior fins ara disponibles en el mercat en termes de precisió.

Resumen

La presente tesis tiene como objetivo exponer el estudio de un sistema de localización en espacios interiores. Se ha desarrollado utilizando tecnología de VLC (Visible Light Communication).

Para comprender mejor el proceso y los resultados del estudio, es necesario saber que calcula la orientación y la posición del dispositivo utilizando únicamente fotometría. Para ello, se aprovecha la flexibilidad que otorga el sistema 5.0 de Android (con la API Camera2) al control de la cámara del móvil y se disponen 4 paneles LED en el techo que transmiten información mediante VLC en el campo de visión de la cámara.

Además, se debe aclarar que no será necesario alterar las condiciones de iluminación de la sala en la que se implemente el sistema. En términos de precisión se muestra superior al resto de sistemas de localización interior ahora disponibles en el mercado.

Agradecimientos

Mi agradecimiento a todo el equipo de telemática, especialmente al Prof. Josep Paradells y a Laura Herrera, por su ayuda a lo largo del proyecto y haberme proporcionado las ideas y el material necesario para poder llevarlo a cabo. Asimismo, me gustaría remarcar la importante colaboración de la Universidad Politécnica de Cataluña permitiéndome hacer uso de su material e instalaciones a lo largo de la realización de este estudio.

Revision history and approval record

Revision	Date	Purpose
0	16/09/2015	Document creation
1	08/10/2015	Document revision
2	14/10/2015	Document modification
3	15/10/2015	Document Final Revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Marc Combalia	marc.gog@gmail.com
Josep Paradells	josep.paradells@entel.upc.edu

Written by:		Reviewed and approved by:	
Date	14/10/2015	Date	14/10/2015
Name	Marc Combalia	Name	Prof Josep Paradells
Position	Project Author	Position	Project Supervisor

Índice

The table of contents must be detailed. Each chapter and main section in the thesis must be listed in the “Table of Contents” and each must be given a page number for the location of a particular text.

Abstract 2
Resum 3
Resumen 4
Agradecimientos 5
Revision history and approval record 6
Índice 7
Lista de figuras 10
Lista de tablas 11
1. Introducción 12
1.1. Formulación del problema 12
1.2. Finalidad 12
1.3. Antecedentes del proyecto 13
1.4. Requisitos y especificaciones del proyecto 13
1.5. Plan del proyecto 14
1.5.1. Paquetes de trabajo 14
1.5.2. Diagrama de Gantt 18
1.5.3. Desviaciones del plan de trabajo original 18
2. Perspectiva de la tecnología usada en este proyecto 19
2.1. Introducción a las tecnologías de localización interior de dispositivos 19
2.2. Funcionamiento de la comunicación por luz visible (VLC) 20
2.3. Localización interior de dispositivos usando la tecnología VLC 21
2.3.1. Epsilon [10] 22
2.3.2. Luxapose [9] 22

3. Metodología y desarrollo del proyecto 24
3.1. Espacio de desarrollo 24
3.2. Diseño del sistema transmisor 24
3.3. Obtención de imágenes 26
3.4. Delimitación de las zonas donde se encuentran las luces moduladas 29
3.4.1. Obtención de zonas de interés mediante técnicas de procesado de imagen 29
3.4.2. Obtención de zonas de interés mediante técnicas de aprendizaje automático 31
3.5. Lectura de la información transmitida 32
3.6. Estimación de la posición 33
3.6.1. Cálculo de los parámetros intrínsecos 34
3.6.2. Obtención de parámetros extrínsecos 39
3.7. Protocolo para la transmisión de datos 41
3.7.1. Velocidad de transmisión 41
3.7.2. Códigos usados para la transmisión de los datos 42
3.7.2.1. Codificación Manchester 42
3.7.2.2. Codificación cíclica 44
3.8. Desarrollo de la aplicación 46
4. Resultados 49
4.1. Precisión del sistema de localización 50
5. Presupuesto 56
5.1. Componentes del prototipo, lista de software y hardware: 56
6. Conclusiones y desarrollo futuro: 57
6.1. Uso de la API Camera2: 57
6.2. Codificación 57
6.3. Precisión 57
6.4. Desarrollo futuro 59

7. Bibliografía .....**Error! Bookmark not defined.**

Glosario 61

Lista de figuras

Figura 1: Representación de transmisión focalizada....21
Figura 2: Laboratorio de telemática....25
Figura 3: Sistema transmisor....26
Figura 4: Barrido horizontal....27
Figura 5: Barrido horizontal en panel LED....27
Figura 6: Imagen tiempo de exposición automático....28
Figura 7: Imagen tiempo exposición reducido....28
Figura 8: Proceso de recorte de la zona de interés....31
Figura 9: Panel LED transmitiendo 101010....32
Figura 10: Recorte de la zona de interés....33
Figura 11: Recorte de la fila central....33
Figura 12: Matriz de parámetros intrínsecos....34
Figura 13: Vector de distorsión....34
Figura 14: Figura usada para la calibración....36
Figura 15: Imágenes de calibración....36
Figura 16: A la izquierda imagen original, a la derecha imagen corregida....37
Figura 17: Imagen tomada con complemento gran angular, original y corregida....38
Figura 18: Imagen tomada desde la misma distancia sin el complemento Gran Angular 38
Figura 19: Sistema P3P....39
Figura 20: Sistema de ecuaciones P3P....39
Figura 21: Representación de roll, yaw y pitch....41
Figura 22: Se puede observar como las líneas blancas son ligeramente más grandes que las negras, a pesar de estar transmitiendo 101010....42
Figura 23: Ejemplo codificación Manchester....43
Figura 24: Ejemplo codificación cíclica....44
Figura 26: Pantalla Menú....47
Figura 27: Pantalla Ajustes....47
Figura 28: Pantalla principal....48
Figura 29: Histograma y cálculo de la probabilidad de error acumulada....51
Figura 30: Posiciones en el plano XY....52
Figura 31: Histograma del error de orientación....53
Figura 25: Representación bits (distancia)....55

Lista de tablas

Each table in the thesis must be listed in the “List of Tables” and each must be given a page number for its easy location.

Tabla 1: Comparación codificación Manchester-Cíclica 46
Tabla 3: Cálculo de la precisión de localización 49
Tabla 4: Cálculo del error de orientación 50
Tabla 5: Histograma de error de posición 51
Tabla 6: Tabla del histograma del error de orientación 53
Tabla 2: Número de bits en función de la distancia (10000 bps) 54

1. Introducción

1.1. Formulación del problema

Los sistemas de localización en espacios interiores intentan encontrar la posición y orientación de objetos o personas dentro de edificios usando señales electromagnéticas. Actualmente no existe ningún sistema que realice este tipo de funciones de forma universal, ya que no se cuenta con la precisión suficiente. Además, la mayoría de ellos no cuentan con la predicción de orientación, parámetro de gran utilidad para muchas de las aplicaciones. Es por ello que se trata de un campo abierto a la investigación a pesar de que son muchas las utilidades que puede tener esta tecnología:

- Sistema de guiado para invidentes
- Sistemas de localización estilo mapa para el interior de edificios
- Sistemas de información focalizada (información específica del lugar dónde se encuentre la persona: carreteras, cines, museos o parques de atracciones se podrían beneficiar) [1]

Los sistemas contemporáneos no permiten una localización suficientemente precisa o requieren sistemas intrusivos para el usuario (cómo la implementación de una nueva tecnología en el edificio en el que se pretenda implementarlos). Por esta razón, se intenta solucionar estos problemas mediante la implementación de un nuevo sistema para la localización interior de dispositivos que se aproveche del uso de VLC (comunicación por luz visible) combinada con fotometría.

1.2. Finalidad

La finalidad de este proyecto es desarrollar un sistema de localización de dispositivos que aproveche la presencia de fuentes de luz para transmitir información usando la tecnología de VLC (Visible Light Communication). Con ello, se podrá obtener una posición más precisa sin alterar de manera evidente las condiciones de iluminación de la sala en la que se pretenda implementarlo (ni en intensidad de luz, ni en distribución de los puntos de luz). El sistema debería poder funcionar a pesar de la presencia de cualquier tipo de iluminación, tanto natural como artificial.

Los objetivos principales del proyecto son:

- Conocer de qué manera el nuevo sistema Android 5.0 (Lollipop) permite la lectura de las fuentes de luz a partir de la cámara
- Desarrollar un algoritmo que permita separar las diferentes fuentes de luz de

manera rápida y eficaz

- Leer la información de cada fuente de luz por separado de manera fiable y establecer un protocolo para la transmisión
- Desarrollar un algoritmo que permita obtener la localización del dispositivo con la información obtenida por la referencia visual de la cámara y por la lectura de las fuentes de luz

Una de las consideraciones a tener en cuenta a la hora de conseguir estos objetivos es que se pretende desarrollar el código de la manera más abierta posible, es decir, que pueda ser utilizado por todos los dispositivos a medida que incorporen la librería Camera2 de Android 5.0. Pero además, es importante tener en cuenta la importancia de no afectar notablemente la duración de la batería del teléfono móvil usado.

1.3. Antecedentes del proyecto

Este estudio utiliza la tecnología de transmisión VLC (Visible Light Communication) desarrollada por el equipo del Profesor Josep Paradells para enviar información a través de paneles LED mediante la modulación de éstos.

1.4. Requisitos y especificaciones del proyecto

Los requisitos del proyecto son los siguientes:

- Capturar imágenes con la cámara del móvil que permitan trabajar con la tecnología VLC
- Localizar las fuentes de luz en la imagen de la cámara
- Leer información de las fuentes de luz
- Procesar la información para localizar el dispositivo desde el cual se obtienen las imágenes

Las especificaciones del proyecto son las siguientes:

- Capturar imágenes modificando los parámetros de la cámara gracias a la librería Camera2 de Android 5.0
- Programar un algoritmo que permita diferenciar entre distintas fuentes de luz
- Programar un algoritmo que permita la lectura de los datos transmitidos por cada una de las fuentes de luz
- Combinar la información 2D obtenida de la cámara, con la información 3D que nos proporcionan las fuentes de luz para encontrar la localización del dispositivo capturador de imágenes

- El software está diseñado para ejecutarse en Android 5.0 con la API Camera2 instalada con soporte completo. Se ha usado la librería OpenCV para el procesado de imagen y la posterior localización del dispositivo

1.5. Plan del proyecto

Este proyecto se ha dividido en 5 paquetes de trabajo principales: el primero consiste en la investigación de la nueva librería Camera2 de Android. En el segundo y tercer paquete de trabajo se procesa la imagen para obtener las localizaciones e información de las luces. Finalmente, en el cuarto paquete de trabajo se localiza la posición del dispositivo capturador. El quinto plasma la implementación del sistema de localización en un escenario real.

1.5.1. Paquetes de trabajo

<table>
<tr><td colspan="2">Project: RESEARCH</td><td>WP ref: (WP1)</td></tr>
<tr><td colspan="2">Major constituent: research</td><td>Sheet 1 of 2</td></tr>
<tr><td colspan="2" rowspan="2">Short description:
Study and get used to Android programming.
Study the predecessors of the technology.</td><td colspan="2">Planned start date:02/03/15
Planned end date:30/03/15</td></tr>
<tr><td colspan="2">Start event:
End event:</td></tr>
<tr><td>Internal task T1: Get used to Android programming.

Internal task T2: Study the predecessors of the technology.</td><td>Deliverables: Project Proposal and Work Plan</td><td>Dates:</td></tr>
</table>

Project: VLC RECEPTOR IMPLEMENTATION	WP ref: (WP2)

Major constituent: software	Sheet 1 of 2	
Short description: Implement the VLC reader only on the source zone.	Planned start date:30/03/15 Planned end date:25/05/15	
	Start event: End event:	
Internal task T1: Read one source Internal task T2: Differentiate and read more than one light source	Deliverables: Critical Review	Dates:

Project: VLC DECODER DESIGN	WP ref: (WP3)	
Major constituent: software	Sheet 1 of 2	
Short description: Design of the technology used to receive information from the VLC transmitters.	Planned start date:25/05/15 Planned end date:29/06/15	
	Start event: End event:	
Internal task T1: VLC decoder design and implementation	Deliverables:	Dates:

Project: INDOOR LOCALIZATOR DESIGN	WP ref: (WP4)	
Major constituent: software	Sheet 2 of 2	
Short description: locate the device using fotometry.	Planned start date:29/06/15 Planned end date:31/08/15	
	Start event: End event:	
Internal task T1: Work with photometry to obtain the location	Deliverables:	Dates:

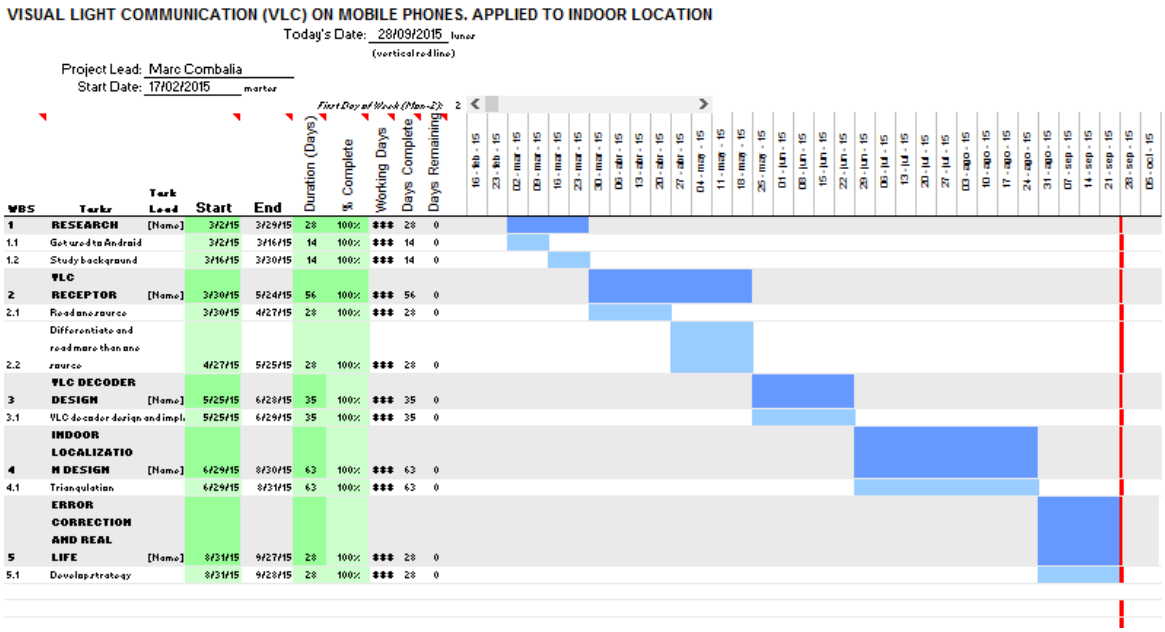
Project: REAL LIFE APPLICATION	WP ref: (WP5)
Major constituent: software	Sheet 2 of 2
Short description: make a real usage app with the technology studied. Locating the device in an indoor map.	Planned start date:31/08/15 Planned end date:28/09/15
	Start event: End event:

Internal task T1: Application development.	Deliverables:	Dates:

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
1	1 2	Android Learning Project background	Project Proposal and Work Plan	1 2-4
2	1 2	Single source receptor Multiple source receptor	Implement 1 source receptor Implement multiple source rx	4-8 8-12
3	1	Decoder design	Design the receptor software	12-17
4	1	Locator design	Calculate p via fotometry	17-26
5	1	Real Life app	Application using the technology developed	26-30

1.5.2. Diagrama de Gantt



1.5.3. Desviaciones del plan de trabajo original

Para poder conseguir los objetivos del proyecto se ha tenido que alargar el periodo de entrega a la segunda convocatoria. También se ha cambiado el enfoque inicial de localización (de triangulación a fotometría).

2. Perspectiva de la tecnología usada en este proyecto

Se presenta un breve repaso de las tecnologías de localización más importantes en espacios interiores. A continuación se explica cómo funciona la comunicación por luz visible y finalmente se explican más profundamente aquellas tecnologías similares a la investigada en este proyecto, que pretendan localizar el dispositivo usando la tecnología VLC.

2.1. Introducción a las tecnologías de localización interior de dispositivos

Un sistema de posicionamiento en espacios interiores es una tecnología que permite la localización de objetos o personas dentro de un edificio mediante el uso de ondas de radio, campos magnéticos, señales acústicas o alguna otra información obtenida por los sensores del dispositivo móvil. [2] [3] [4] [5] [6] [7] [8] [9] [10]

- Posicionamiento magnético: se basa en el estudio de las variaciones de los campos magnéticos causadas por los metales del interior de los edificios. Permite una precisión de entre 1 y 2 metros. [3] [5]
- Medidas de inercia: analizan los movimientos de la escena para calcular una predicción de la posición del dispositivo. Una de las técnicas utilizadas consiste en el análisis de los pasos que hace la persona que lleva el dispositivo. Se puede intuir que la localización no gozará de una precisión demasiado elevada. [4] [5]
- RFID tags: mediante el etiquetaje de objetos cuya posición es conocida, podemos saber qué posición ocupa el sujeto entre ellos. El problema principal de este sistema es que se necesita tener contacto con estos objetos etiquetados continuamente, de lo contrario no se podrá localizar con precisión el dispositivo. Dependiendo del algoritmo que se use, se necesitará una gran densidad de RFID (K-nearest neighbours). [6]
- Si se cuenta con la presencia de señales electromagnéticas en la sala dónde se pretenda realizar la localización se pueden realizar diferentes estrategias para aproximar la posición del dispositivo a localizar: [6] [7] [8] [9] [10]
    - Ángulo de llegada: si se conoce el ángulo al que están cada una de las fuentes respecto al objeto a localizar, se puede, mediante **Triangulación**, encontrar la posición del dispositivo receptor. Se necesita un mínimo de 3 fuentes para que el algoritmo pueda obtener la posición con precisión. A pesar de parecer una buena opción, el ángulo suele ser una medida difícilmente calculable con precisión, de manera que tampoco se podrá obtener una localización demasiado precisa.
    - Distancia de las fuentes: si se conoce la distancia a la que están cada una de las fuentes respecto al objeto a localizar, se puede, mediante **Trilateración** encontrar

la posición del dispositivo receptor. Se necesita un mínimo de 3 fuentes para que el algoritmo pueda obtener la posición con precisión. Suele ser una técnica mucho más utilizada que la anterior por qué es más fácil obtener la distancia de las fuentes que el ángulo de la señal recibida. Se listan a continuación las técnicas más usadas para llevar a cabo dicho cálculo:

- Tiempo de llegada: si se conoce la diferencia entre el tiempo en el que se ha emitido la señal y el tiempo en el que se ha recibido se puede calcular la distancia entre el emisor y el receptor (x = v*t – dónde v = c = 3*10^8 en la mayoría de los casos). Se debe tener en cuenta que el tiempo de llegada puede ser modificado por reflexiones, cosa que puede dificultar el cálculo. Asimismo, para poder calcular la distancia con precisión es necesario tener todos los dispositivos sincronizados. Es por ello que no es una técnica recomendable para utilizar en espacios interiores, dónde el tiempo de vuelo de la señal es muy pequeño.
- Potencia de señal: sabiendo de qué manera disminuye la potencia de señal a medida que se propaga, se puede calcular la distancia a la que se encuentra el dispositivo receptor de la fuente. Esta técnica suele ser la empleada por la mayoría de sistemas para, mediante Trilateración, calcular la posición del dispositivo. [6] [7] [8] [10]

2.2. Funcionamiento de la comunicación por luz visible (VLC)

La comunicación por luz visible es aquella que usa las ondas de luz visible del espectro electromagnético (entre 780 y 375 nm de longitud de onda) para transmitir información. Dicha transmisión se puede hacer de múltiples maneras:

- Mediante modulación frecuencial: cambiando la frecuencia de transmisión de la luz se pueden obtener diferentes longitudes de onda. Cada una de las cuales puede llevar asociada una información determinada (actuando de identificador). Mediante pequeñas variaciones en la iluminación de la sala (imperceptibles al ojo humano) se puede saber en qué lugar de la sala está el sujeto.
- Mediante modulación pulsada: la variación de la alimentación a un sistema led permite modular la señal que éste emite, pudiendo transmitir bits de información. [9] [10]

La tecnología de comunicación por luz visible presenta varias ventajas:

- La presencia de luz artificial es algo común en la mayoría de los lugares. Es por ello que tiene una gran utilidad poder darle una funcionalidad extra a la fuentes de

iluminación.

- La presencia de fuentes de iluminación no presenta ningún problema de salud para el ser humano:
  - Las personas conviven de manera habitual con fuentes de luz artificiales, y hasta ahora no se ha conocido ningún caso en el que la luz artificial haya supuesto algún inconveniente.
  - Dado que es una tecnología que el ser humano puede percibir mediante el sentido de la vista, se puede proteger de ella si lo considera necesario.
- No supone interferencias para otros sistemas electromagnéticos.
- Permite realizar una transmisión de información focalizada. [1] Puede transmitir información a dispositivos de forma mucho más direccional que otras tecnologías (como el WIFI). Este fenómeno limita mucho el número de interferencias que se puede encontrar cada fuente, permitiendo una velocidad de transmisión mucho más elevada.

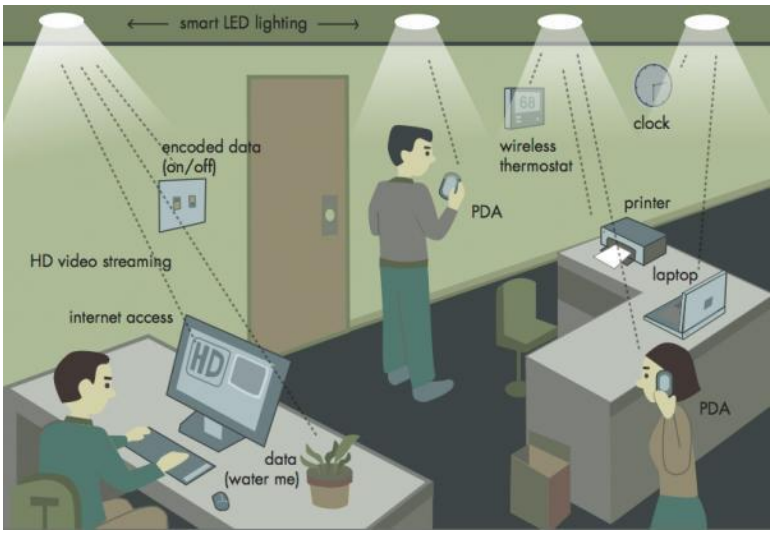


Figura 1: Representación de transmisión focalizada

2.3. Localización interior de dispositivos usando la tecnología VLC

Existen varios investigadores que han propuesto soluciones de localización en espacios interiores usando tecnologías de comunicación por luz visible. En este apartado se exponen aquellos que se han considerado más importantes y de mayor similitud con el sistema propuesto en la tesis.

2.3.1. Epsilon [10]

El sistema propuesto en esta tesis consiste en la transmisión de información a través de unos paneles LED al dispositivo, y de esta manera poder encontrar su posición.

El sistema transmisor combina una modulación frecuencial y una modulación pulsada. Para dar un nivel más de redundancia a la diferenciación entre fuentes de luz se crean canales de distinta frecuencia.

El sistema de localización consiste en la técnica de trilateración explicada en el apartado 2.1. Para el cálculo de la distancia del emisor al receptor realiza el algoritmo de RSS (Received Signal Strenght). Mediante el conocimiento de la función de atenuación de la intensidad de la luz, puede saber la distancia a la que encuentra el foco. Cabe comentar que la intensidad de la luz puede estar expuesta a interferencias de luces vecinas, o a interferencias por luz natural (también puede ocurrir que el nivel de luz generado por cada foco sea ligeramente diferente).

Para poder realizar el cálculo de intensidad lumínica, el dispositivo que se quiera localizar debe incorporar un sensor de luz externo que se acopla al puerto de audio.

Este sistema logra un error medio de 0.4 metros.

2.3.2. Luxapose [9]

El sistema Luxapose es el que más similitudes comparte con el propuesto en la tesis. El sistema de transmisión de información propuesto consiste en la modulación de intensidad de paneles LED. Se utiliza un dispositivo Lumia (Windows Phone) que les permite el control de la cámara del dispositivo suficiente como para poder capturar las modulaciones de la intensidad de iluminación.

Se trabaja con un identificador para cada panel LED. El protocolo que usan consiste en modular el panel LED a diferentes frecuencias, y asociar cada una de las frecuencias a un identificador (en la publicación se comenta que también han probado la transmisión de datos mediante codificación Manchester).

El sistema no necesita ningún hardware adicional (a diferencia de Epsilon) para la captura de la información de los paneles LED, usa la misma cámara del dispositivo. Lo que sí necesita es conectividad con un ordenador para detectar y realizar la lectura de los paneles LED (el procesado de imagen no se hace en el propio dispositivo).

Mediante el uso de codificación mediante modulación frecuencial o codificación Manchester sólo puede realizar la transmisión de aproximadamente 30 identificadores.

La precisión de la localización es de aproximadamente 10 cm.

3. Metodología y desarrollo del proyecto

3.1. Espacio de desarrollo

El software ha sido desarrollado en Android Studio, ejecutado en Ubuntu 14.04. Está pensado para ser ejecutado en un dispositivo Android 5.0 (Lollipop) o superior que sea compatible al máximo nivel con la API Camera2.

Para las funciones de procesado se ha utilizado la librería OpenCV 3.0.0, que viene instalada junto a la aplicación. Más concretamente para el procesado de imagen, técnicas de aprendizaje automático y fotometría. OpenCV es una librería de funciones programables que está principalmente enfocada a la visión por computador en tiempo real. La librería está escrita en C++ pero cuenta con una versión en Android.

Además, se ha usado un dispositivo Nexus 5 para la ejecución del software y el testeo. Se ha desarrollado tanto para la cámara frontal como para la trasera del dispositivo, siendo la primera opción la de preferencia debido a la mayor facilidad de utilización para el usuario final.

3.2. Diseño del entorno

Para que el dispositivo pueda calcular su posición y orientación es necesario disponer de un espacio donde pueda detectar como mínimo 4 luces transmitiendo información VLC en el campo de visión de la cámara. Para ello, se han dispuesto 4 luces en el techo del laboratorio para poder testear el software, como se puede ver en la siguiente figura:

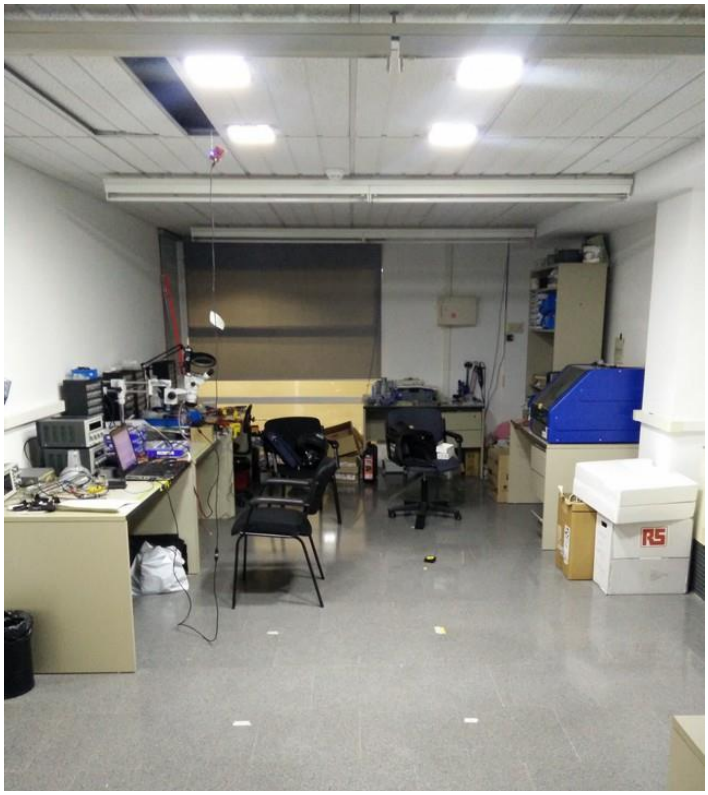


Figura 2: Laboratorio de telemática

Las luces están dispuestas en forma cuadrangular de 0.90m de costado a una altura de 2.40 metros del suelo. Se sitúa el móvil en el suelo, consiguiéndose 2.40 metros de separación entre el dispositivo y los paneles. Teniendo en cuenta que en una situación normal el móvil estará a una altura aproximada de 1.20 metros del suelo (sostenido por el usuario), se conseguirían unas condiciones similares a las de un espacio interior con un techo de aproximadamente 3.60 metros de altura.

Para poder calcular la precisión de localización se ha dispuesto una cuadrícula en el suelo con referencias a las posiciones X y Y de cada panel.

Los paneles LED que se han usado para el desarrollo del proyecto son “alverlamp 20 W 1480 lm” (17x17 cm). Se podría usar cualquier otro panel que permitiera la modulación. Para poderla efectuar se han acoplado a una controladora y a un driver. La controladora viene programada con un software al que se le puede introducir la velocidad de modulación a la vez que el mensaje que se quiera transmitir. Junto con el driver, proporciona los niveles de alimentación necesarios para poder modular el panel LED.

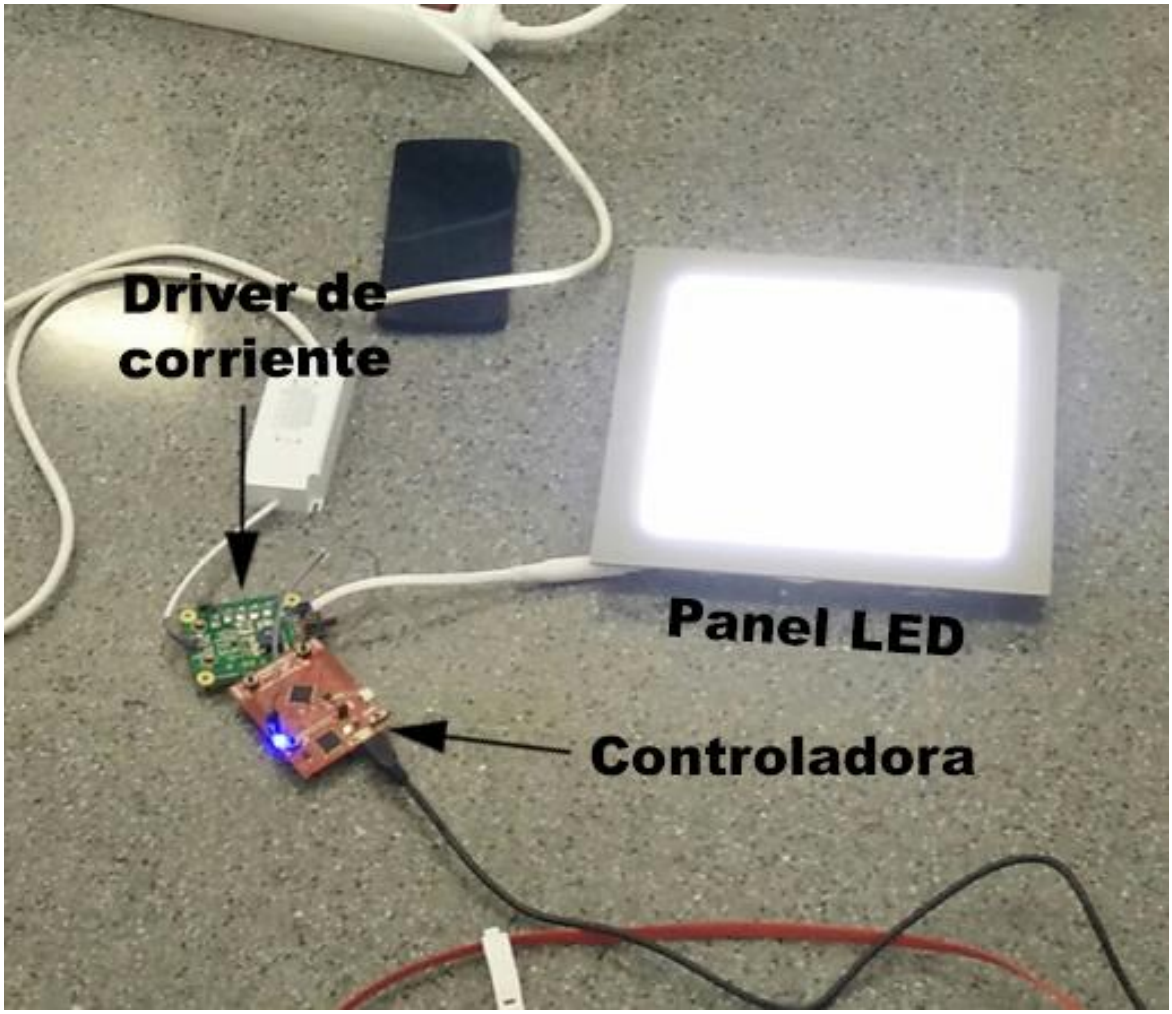


Figura 3: Sistema transmisor

La controladora usada es “Texas Instruments TM4C123g Eval Board” y el driver de corriente usado es “Texas Instruments LM3409HV Eval Board”.

3.3. Obtención de imágenes

En este apartado se describe el funcionamiento de las cámaras digitales que vienen incorporadas en la mayoría de dispositivos móviles. Consiste en un barrido horizontal en el que se escanea la imagen por columnas hasta completar el tamaño final.

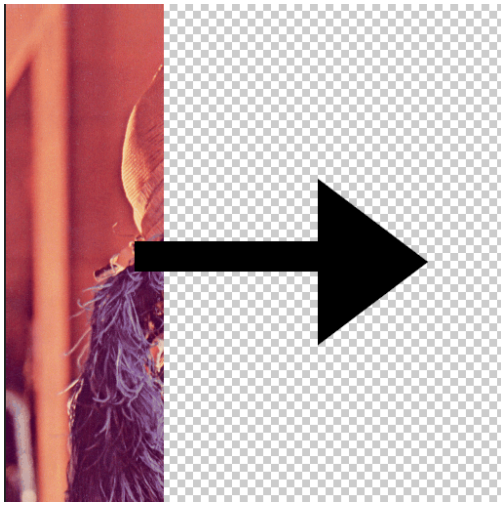


Figura 4: Barrido horizontal

Este efecto es conocido como “Rolling Shutter”, y en el caso de este proyecto, puede ser aprovechado. A medida que el sensor barra el entorno se irán capturando los cambios de iluminación a lo largo de la imagen. De manera que si se modula un panel LED a alta velocidad, podremos ver representados estos cambios como barras verticales. Véase en la siguiente figura:

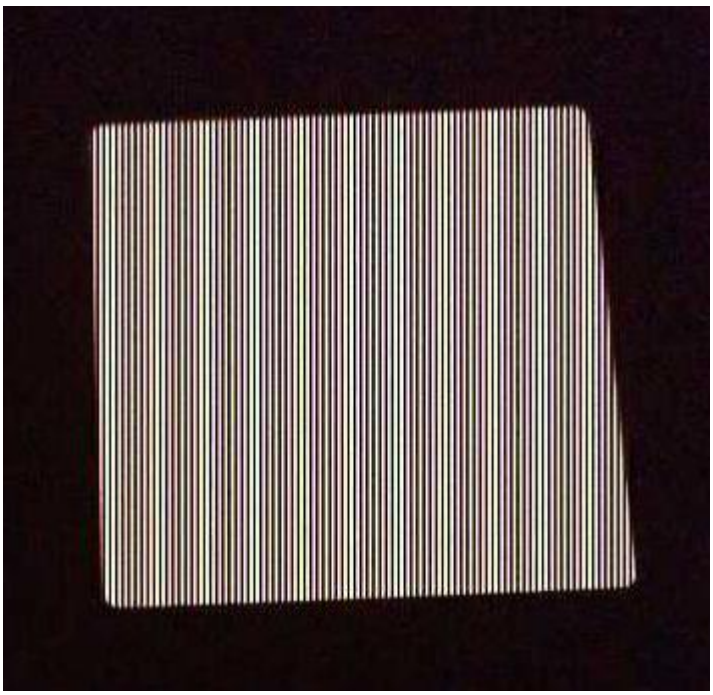


Figura 5: Barrido horizontal en panel LED

Para que se puedan mostrar esos cambios de iluminación se debe modificar el tiempo de exposición de la cámara. Si éste es demasiado alto, el sensor captará demasiada luz y no se podrá diferenciar si las luces están siendo moduladas o no.

Figura 6: Imagen tiempo de exposición automático

Sin embargo, al reducir el tiempo de exposición, el sensor captura mucha menos luz y de manera mucho más concentrada, pudiendo visualizar las zonas de luz donde se transmite información:

Figura 7: Imagen tiempo exposición reducido

Como se puede ver en la figura anterior, sólo uno de los paneles está siendo modulado (se observan los bits de información representados en forma de columnas a lo ancho de la imagen en el panel).

A la vez que se reduce el tiempo de exposición, se aumenta la sensibilidad del sensor con el objetivo de poder capturar suficiente luz como para visualizar los paneles en la imagen.

Hasta ahora el sistema Android no permitía la modificación de parámetros como los ya descritos, pero desde la versión 5.0, en aquellos dispositivos que implementan al máximo

nivel la librería Camera2, se puede acceder a un control de la cámara mucho más exhaustivo.

El código que permite consultar las características de la cámara, en concreto el tiempo de exposición y la sensibilidad del sensor es el siguiente:

```
String cameraId = manager.getCameraIdList()[1];
CameraCharacteristics characteristics = manager.getCameraCharacteristics(cameraId);
Log.e("Camera support", "" + characteristics.get(CameraCharacteristics.SENSOR_INFO_EXPOSURE_TIME_RANGE).toString());
Log.e("Camera support", "" + characteristics.get(CameraCharacteristics.SENSOR_INFO_SENSITIVITY_RANGE).toString());
```

Y el de modificación de los parámetros de interés:

```
captureBuilder.set(CaptureRequest.CONTROL_AF_MODE,
       CaptureRequest.CONTROL_AF_MODE_AUTO);
captureBuilder.set(CaptureRequest.CONTROL_AE_MODE,
       CaptureRequest.CONTROL_AE_MODE_OFF);
captureBuilder.set(CaptureRequest.CONTROL_AWB_MODE,
       CaptureRequest.CONTROL_AWB_MODE_AUTO);
captureBuilder.set(CaptureRequest.SENSOR_SENSITIVITY, DataHolder.getInstance().getISO());
captureBuilder.set(CaptureRequest.SENSOR_EXPOSURE_TIME, DataHolder.getInstance().getExposureTime());
```

3.4. Delimitación de las zonas donde se encuentran las luces moduladas

Una vez se ha desarrollado la forma de capturar la información transmitida por los paneles, se debe diferenciar del resto de la imagen para poder leerlos posteriormente. Para ello se han explorado dos técnicas distintas de detección de objetos dentro de imágenes: mediante procesado de imagen tradicional y mediante técnicas de machine learning.

3.4.1. Obtención de zonas de interés mediante técnicas de procesado de imagen

Al observar las zonas de interés, donde existe la presencia de luz modulada, se puede ver que tienen unas características determinadas. Están formadas por franjas verticales siempre orientadas en la misma dirección a lo ancho de la imagen. Se puede entonces suponer que la derivada horizontal encima de dichas zonas particulares será de un valor absoluto bastante elevado. Esta información es suficiente para poder diferenciarlas, pero además, el hecho de que la derivada vertical debe ser casi nula (ya que no existe variación en el eje Y) permite diferenciarlas aún mejor. Aprovechando esta peculiaridad, se pueden restar en valor absoluto ambas derivadas y destacar las zonas de interés dentro de la imagen.

Al restar las derivadas horizontal y vertical las zonas donde se presentan las luces moduladas quedan destacadas del resto de la imagen. Después, mediante métodos de *erode* y *dilate*, se consigue resaltar mejor esas zonas, a la vez que eliminar las partes aisladas.

Una vez se tienen las zonas de interés bien diferenciadas se buscan los contornos, que serán los que se consideren zonas de luz modulada. Se descartarán aquellos con área demasiado pequeña (considerados ruido). Véase el código:

```
public Mat locateBarcode3(Mat mat){
		mat.copyTo(image);

		//Paso a blanc i negre per poder fer el processat
		Imgproc.cvtColor(image, grey, Imgproc.COLOR_BGR2GRAY);
		//Agafo les zones més brillants de la imatge perque seran les que contindran les llums
		Imgproc.threshold(grey, grey, 200, 255, Imgproc.THRESH_BINARY);

		//Faig les derivades en el eix vertical i l'horitzontal, la derivada horitzontal serà gran a la zona de interes
		// i la vertical sera petita
		Imgproc.Sobel(grey, sobX, CvType.CV_32F, 1, 0);
		Imgproc.Sobel(grey, sobY, CvType.CV_32F, 0, 1);

		//Resto les dues derivades per el que he dit abans, em permetrà trobar la zona on son les llums
		Core.absdiff(sobX, sobY, gradient);
		//grey.copyTo(gradient); // per si vull detectar llums no modulades
		//sobX.copyTo(gradient);
		Core.convertScaleAbs(gradient, gradient);
		System.out.println(Core.mean(gradient).toString());

		//Imgcodecs.imwrite("../Camera Calibration/gradient.jpg", gradient);

		//Calculo els coeficients que faig servir per erode i dilate a partir de la resolució de la imatge (obtinguts de 		forma experimental))
		int aux1 = image.cols() / 50;
		int aux2 = image.cols() / 100;
		int aux3 = image.cols() / 100;

		//Aquesta funció permet eliminar el soroll que es podria produir
		kernel = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size(aux1, aux1));
		closed = new Mat();
		Imgproc.morphologyEx(gradient, closed, Imgproc.MORPH_CLOSE, kernel);

		//Dilate i erode em permeten unificar tota la zona de les derivades, que normalment no son continues,
		// i ferles com si ho fossin per poderles detectar millor
		kernel = Imgproc.getStructuringElement(Imgproc.MORPH_DILATE, new Size(aux2, aux2));
		Imgproc.dilate(closed, closed, kernel, new Point(-1, -1), 4);
		kernel = Imgproc.getStructuringElement(Imgproc.MORPH_ERODE, new Size(aux3,
								aux3));
		Imgproc.erode(closed, closed, kernel, new Point(-1, -1), 4);

		//Imgcodecs.imwrite("../Camera Calibration/closed.jpg", closed);

		//Detecto els contorns que s'han poduit a la imatge
		List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
			Imgproc.findContours(closed.clone(), contours, hireachy, Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
		ArrayList<RotatedRect> rectContours = new ArrayList<RotatedRect>();
		MatOfPoint2f aux = new MatOfPoint2f();
		double maxArea = 0;
		RotatedRect rectAux = null;
		for (MatOfPoint contour : contours) {
			//Agafo els rectangles de minima area al rectAuxvoltant dels contorns
			aux = new MatOfPoint2f(contour.toArray());
			rectAux = Imgproc.minAreaRect(aux);
			//System.out.println(rectAux.size.area());
			//rectContours.add(rectAux);
			if(rectAux.size.area() > image.size().area()/3000){
			rectContours.add(rectAux);
			}
		}

		/*for(RotatedRect rRect : rectContours){
			Point points[] = new Point[4];
			rRect.points(points);
			for(int i=0; i<4; ++i){
				Imgproc.line(image, points[i], points[(i + 1) % 4], new Scalar(0, 255, 0), 5);
			}

		}*/

		//Imgcodecs.imwrite("../Camera Calibration/final.jpg", image);

		cReader.readCode(mat, rectContours);

		return image;
}
```

Podemos ver el funcionamiento en la siguiente cadena de figuras:

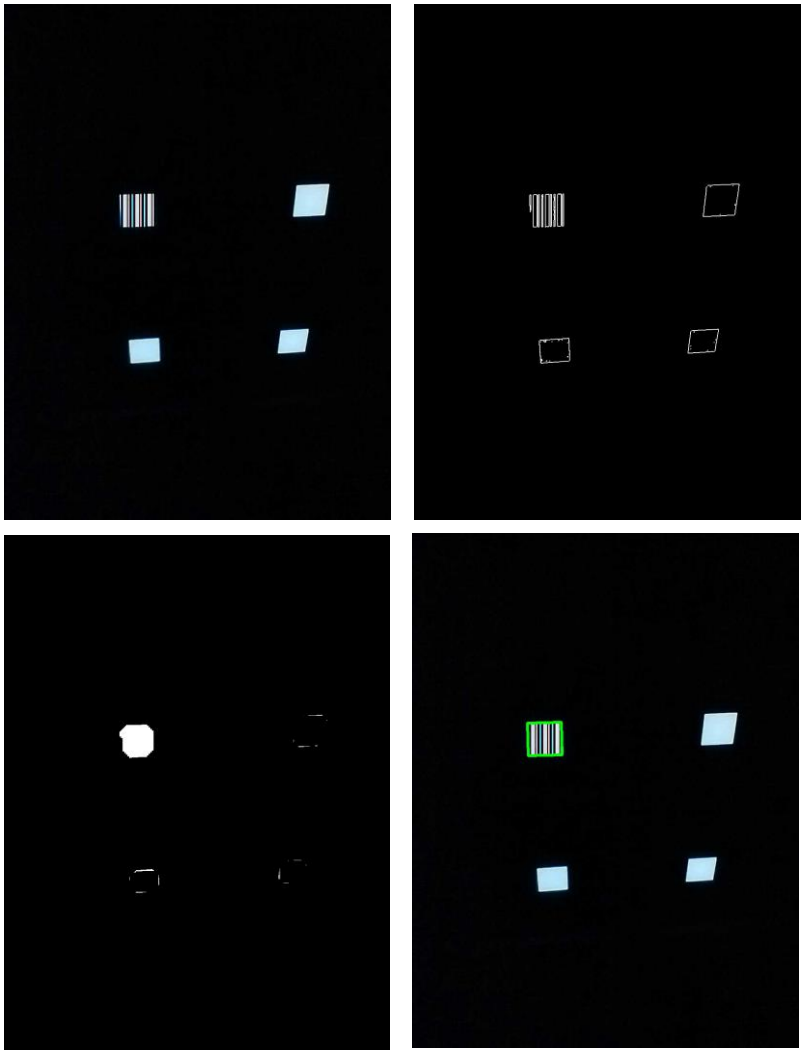


Figura 8: Proceso de recorte de la zona de interés

3.4.2. Obtención de zonas de interés mediante técnicas de aprendizaje automático

Dado la naturaleza del problema (detección de zonas particulares dentro de una imagen), parece lógico el querer incorporar un algoritmo de machine learning para detectar las zonas de interés.

Resultaron útiles aquellas técnicas de obtención de características que hacían operaciones con el gradiente de la imagen. Una de las técnicas más usadas para detectar objetos es la de HOGDescriptors. Consiste en hacer pequeñas ventanas y calcular la orientación del gradiente dentro de ella. Serán las direcciones de los gradientes calculados las que se usaran como características del clasificador.

Una vez obtenidos los descriptores de cada porción de la imagen, se puede entrenar un algoritmo SVM para decidir si el fragmento de imagen que estamos observando contiene una detección positiva o negativa.

La fiabilidad de detección obtenida con el método SVM + HOGDescriptors se muestra elevada, pero para cada imagen debemos pasar una ventana de medida variable a lo largo y ancho hasta encontrar las zonas de interés. Esto hace que sea un proceso especialmente pesado para imágenes de alta resolución, haciéndolo no adecuado para ejecuciones en dispositivos portátiles que no cuentan con la capacidad de procesado suficiente.

Dado que las áreas de detección son fácilmente identificables mediante técnicas de procesado de imagen, no es necesario aplicar técnicas de machine learning para la detección. Por lo tanto, queriendo conseguir el menor tiempo de ejecución, optamos por la primera solución descrita, que permite ejecutar el algoritmo más veces por minuto. El tiempo de ejecución del algoritmo es de aproximadamente 600 milisegundos, comparado con los casi 5 segundos del algoritmo de machine learning. Esto permite realizar la ejecución 100 veces por minuto, obteniendo una tasa de actualización de localización muy alta.

3.5. Lectura de la información transmitida

La información transmitida se presenta en la imagen como una serie de columnas dentro de la zona de interés que representan los intervalos en los que la luz está apagada y encendida.

Figura 9: Panel LED transmitiendo 101010

Para realizar la lectura de la imagen, se pueden contar la cantidad de pixeles dentro de cada intervalo, para después hacer una transformación de numPix → bits.

Con el objetivo de conseguir la información más nítida posible, se decide trabajar solamente con una fila. Se escoge la fila central de cada región detectada (debido a que

es la que contiene más información) y se barre de manera horizontal, guardando el número de pixeles de cada intervalo.

Para conocer el número de pixeles que se corresponden a la transmisión de un bit, se transmite “101010”. De esta manera y mediante el cálculo del ancho de los rectángulos, se obtendrá la información necesaria para conocer el número de pixeles que corresponde a cada bit. Para después transformar a número de bits mediante la relación obtenida:

numBits = numPix*bps/relation

A continuación, se muestra un ejemplo de lectura correspondiente a un panel LED:

Figura 10: Recorte de la zona de interés

Primero se escoge la fila del centro de la zona de interés, obteniendo la siguiente imagen:

Figura 11: Recorte de la fila central

La primera parte del algoritmo cuenta los pixeles realizando un escaneo horizontal (alternando a cada cambio de nivel de píxel), lo cual proporciona la siguiente información:

[3, 9, 6, 3, 6, 3, 3, 9, 6, 3, 6, 3, 4, 8, 6]

Como se puede ver, permite conocer el ancho de cada región en pixeles. Cabe observar que no se tiene en cuenta ni la primera ni la última columna blanca, ya que podrían estar recortadas por el ancho del panel LED. Mediante la segunda parte del algoritmo, se transforma la información de pixeles a información de bits. Véase:

[0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0]

La operación que se ha realizado es la que anteriormente se ha descrito, con bps = 10000 y relation = 30000. Finalmente se ponen el número de unos o ceros en función del resultado de dicha operación.

3.6. Estimación de la posición

Normalmente, estos tipos de sistema usan algún tipo de estimación de la distancia ya sea mediante el cálculo del tiempo entre la emisión y recepción, o mediante el nivel de

intensidad de la señal recibida. Este cálculo permite obtener la posición del receptor conociendo únicamente 3 emisores.

Dado que el sistema estudiado no usa ningún método para calcular de la distancia entre el receptor y las fuentes de luz, se necesitan un mínimo de 4 puntos de referencia para poder localizar la posición del dispositivo mediante fotometría. Dicha posición se obtiene calculando los parámetros intrínsecos y extrínsecos de la cámara.

3.6.1. Cálculo de los parámetros intrínsecos

Los parámetros intrínsecos son aquellos propios de la cámara, que no cambian en función de la imagen capturada. Los podemos ver en la siguiente matriz:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 12: Matriz de parámetros intrínsecos

fx, fy: distancias focales en dirección vertical y horizontal (expresadas en píxeles)

cx, cy: punto central de la cámara (expresado en píxeles)

También se calcula la distorsión de la lente, que se guarda en el siguiente vector:

$$\text{Distortion}_{\text{coefficients}} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

Figura 13: Vector de distorsión

a) Obtención de los parámetros intrínsecos de forma manual

Basta con una consulta a las características de la cámara del dispositivo en estudio para obtener las distancias focales. Se puede hacer mediante el siguiente fragmento de código:

```
SizeF lengts =  characteristics.get(CameraCharacteristics.SENSOR_INFO_PHYSICAL_SIZE);
float[] focal =  characteristics.get(CameraCharacteristics.LENS_INFO_AVAILABLE_FOCAL_LENGTHS);
```

De ésta manera se obtendrán las medidas del sensor y la distancia focal (expresadas en mm) de la cámara. Para poder trabajar con la matriz de parámetros intrínsecos, debemos transformar estos valores a valores de píxel lo cual se consigue mediante la siguiente operación:

focal_pixel = (focal_mm / sensor_width_mm) * image_width_in_pixels

Se suele considerar los centros de la cámara, cx y cy, como el centro de la imagen, por lo tanto se pueden obtener mediante el siguiente cálculo:

cx = anchoImagen/2

cy = altoImagen/2

Los valores de distorsión se suponen nulos, ya que con la utilización de éste método, no hay manera de calcularlos.

Para la cámara frontal del dispositivo Nexus 5 se obtiene una matriz de parámetros intrínsecos y un vector de distorsión con los siguientes valores:

$$Matriz_{intrínseca} = \begin{bmatrix} 1216.57 & 0 & 480 \\ 0 & 1221.57 & 640 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Coeficientes_{distorsión} = [0, 0, 0, 0, 0]$$

b) Calibración de la cámara: obtención de parámetros intrínsecos

Para obtener unos valores más exactos del sensor de cámara con el que se esté trabajando, y por lo tanto poder obtener posteriormente unos cálculos de posición más precisos, se puede realizar una calibración de la cámara para la obtención de los parámetros intrínsecos.

Consiste en realizar fotos de un objeto del cual se conozcan las medidas, y ver de qué manera dichas medidas quedan distorsionadas y escaladas dentro de la imagen de la cámara.

OpenCV proporciona unos métodos que facilitan la calibración de la cámara. El procedimiento para calcular los parámetros intrínsecos de una cámara en concreto consiste en capturar imágenes de este objeto en particular:

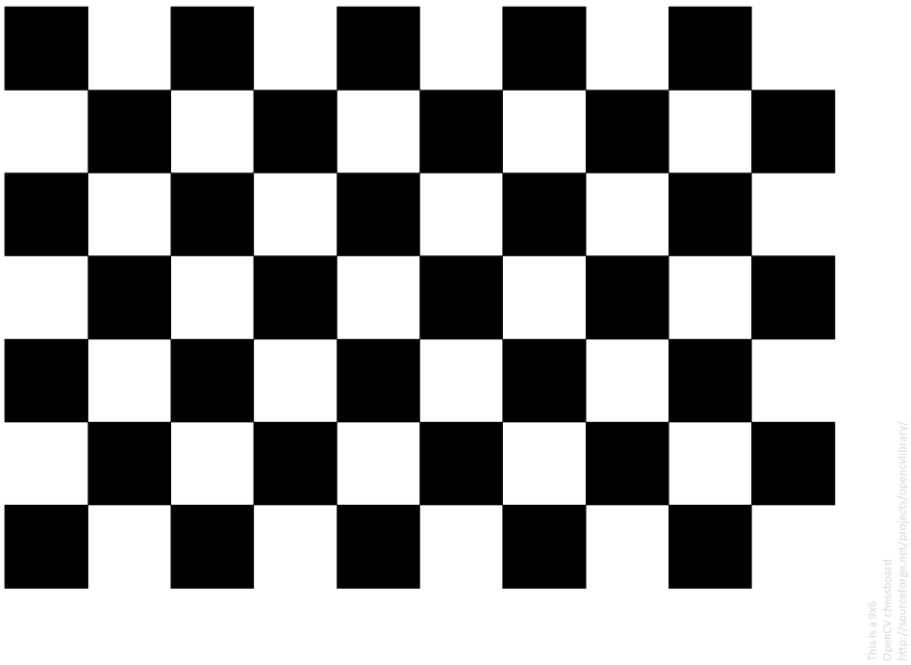


Figura 14: Figura usada para la calibración

Se pueden ver algunas de las fotografías usadas para la calibración de la cámara frontal del Nexus 5 en la siguiente figura:

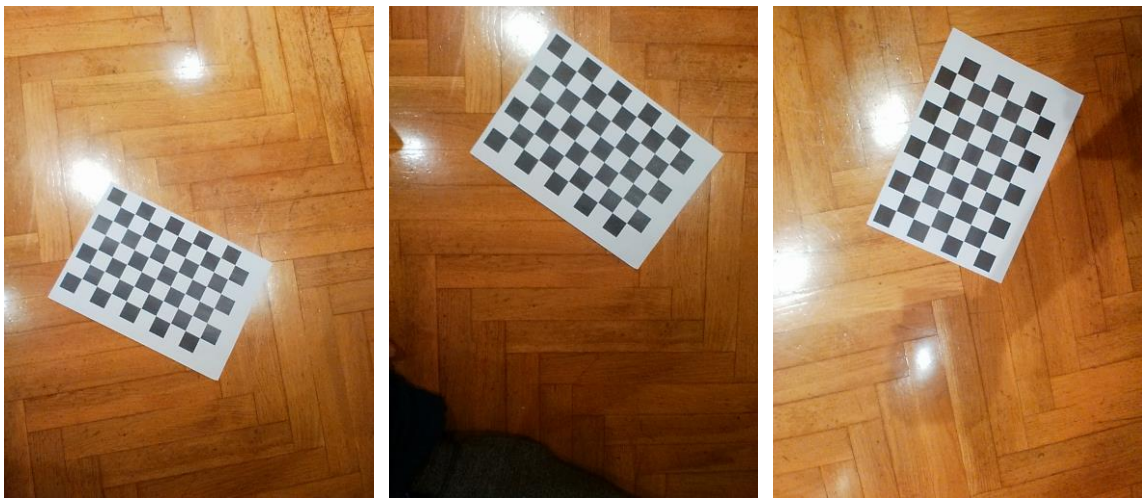


Figura 15: Imágenes de calibración

Se debe procurar capturar imágenes en todos los puntos de la cámara con el objetivo de poder detectar la distorsión a lo largo y ancho de la imagen. OpenCV facilita la detección del “tablero de ajedrez” devolviendo los puntos en los que se encuentra cada cuadrado. De esta manera ya se habrán calculado las posiciones 2D en la imagen. Para facilitar el

cálculo de las posiciones 3D, se supone que el tablero de ajedrez se sitúa en el punto (0, 0, 0), y lo que varía es la posición de la cámara, de manera que si cada cuadrado mide 3 cm, las posiciones serán (0, 0.3, 0), (0.3, 0.3, 0), etc.

Una vez se tienen las correspondencias en 2D y 3D del tablero, se puede efectuar la calibración de la cámara mediante el siguiente código:

```
double errReproj = Calib3d.calibrateCamera(objectPoints, imagePoints,
((Mat) vImg.get(0)).size(), cameraMatrix, distCoeffs, rvecs,
tvecs, flagsCalib);
```

Para el caso de la cámara frontal del Nexus 5 se obtienen los siguientes parámetros intrínsecos:

$$Matriz_{intrínseca} = \begin{bmatrix} 1231.08 & 0 & 479.5 \\ 0 & 1232.37 & 639.5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Coeficientes_{distorsión} = [0{,}09160743985879177, -0{,}2704692452595001, 0, 0, 0]$$

Se puede observar que los resultados son similares a los obtenidos consultando las características de la cámara, pero son estas pequeñas variaciones las que permitirán obtener unos cálculos más precisos. Se puede ver como los parámetros de distorsión afectan a la imagen comparando una imagen obtenida con la cámara y la corregida:

Figura 16: A la izquierda imagen original, a la derecha imagen corregida

Este proceso es importante para obtener buenas medidas en el cálculo de la posición, pero lo es aún más en el caso de que se quiera usar una lente no convencional (por ejemplo una de gran angular).

Figura 17: Imagen tomada con complemento gran angular, original y corregida

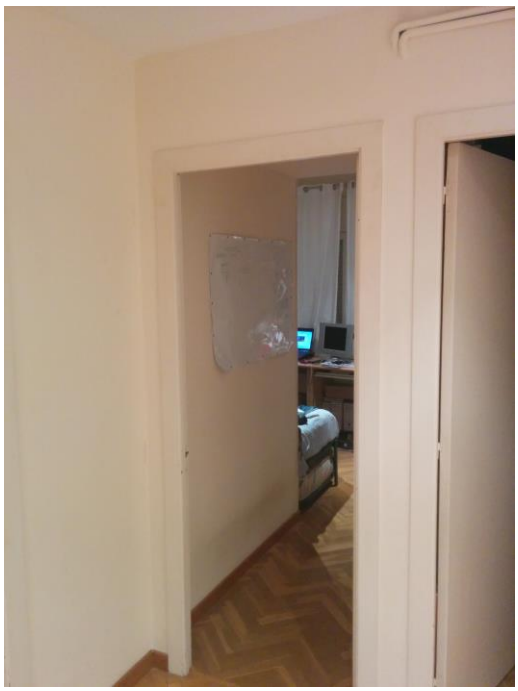


Figura 18: Imagen tomada desde la misma distancia sin el complemento Gran Angular

La calibración de la cámara permite trabajar con los métodos de cálculo de posición utilizando lentes que efectúen una deformación de la imagen (o lentes de teléfonos móviles

de mala calidad). Fíjese que el uso de una lente gran angular facilita la inclusión de más paneles LED en el campo de visión.

3.6.2. Obtención de parámetros extrínsecos

Los parámetros extrínsecos de la cámara son aquellos que sí varían en función de la imagen tomada en cada preciso momento, son la orientación y la posición de la cámara. Para su cálculo se debe encontrar un algoritmo que nos permita, mediante pares de puntos 2D – 3D, encontrar la posición de ésta. La forma de obtener estos parámetros extrínsecos es mediante el algoritmo Perspective-n-Point.

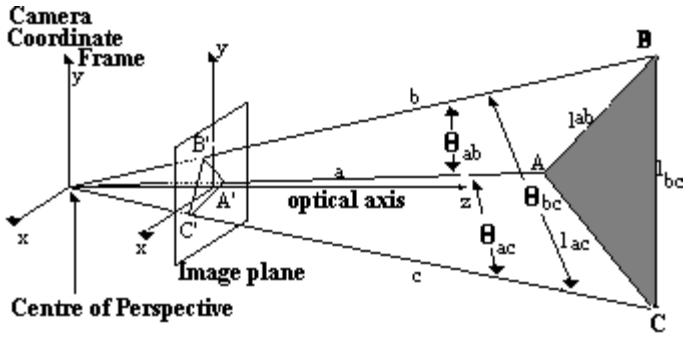


Figura 19: Sistema P3P

$$b = k_1 a \qquad c = k_2 a$$

$$l_{ab}^2 = a^2 + b^2 - 2ab\cos\theta_{ab}$$

$$l_{ac}^2 = a^2 + c^2 - 2ac\cos\theta_{ac}$$

$$l_{bc}^2 = b^2 + c^2 - 2bc\cos\theta_{bc}$$

$$K_1 = \frac{l_{bc}^2}{l_{ac}^2} \qquad K_2 = \frac{l_{bc}^2}{l_{ab}^2}$$

$$q_4 {k_1}^4 + q_3 {k_1}^3 + q_2 {k_1}^2 + q_1 k_1 + q_0 = 0$$

Figura 20: Sistema de ecuaciones P3P

La misma librería que se usa para el tratado de imagen, OpenCV, contiene la resolución de este sistema. De esta manera si se cargan los pares de puntos 2D – 3D juntamente con la matriz de parámetros intrínseca obtenida se puede conseguir la matriz de rotación y el vector de translación que caracterizan la cámara.

```
Calib3d.solvePnP(objectPoints, imagePoints, cameraInstrinsics, distorsionCoef, rvec, tvec);
```

Esta función devuelve una matriz de rotación y un vector de translación que permiten transformar puntos 3D en el espacio a puntos 3D expresados en las coordinadas de la cámara (suponiendo la cámara como punto central del mundo (0, 0, 0)). Lo que se quiere es la transformación inversa para encontrar la posición y rotación de la cámara en función al mundo. Para hacerlo se deben realizar unas operaciones de matrices que se demuestran en la siguiente figura:

```
Mat Rt = new Mat();
Calib3d.Rodrigues(rvec, Rt);
Mat R = new Mat();
Rt.t().copyTo(R);
Core.multiply(R, new Scalar(-1,-1,-1), R);
Mat pos = new Mat();
Core.gemm(R, tvec, 1, new Mat(), 0, pos);
```

Donde *rvec* y *tvec* son los vectores obtenidos mediante el método solvePnP() de OpenCV. Una vez finalizado este proceso, la posición ya puede ser mostrada al usuario.

Los parámetros extrínsecos de la cámara también incluyen información sobre la rotación del dispositivo. La información queda contenida en la matriz R del código superior. Sin embargo, para poder usarla se tiene que cambiar de formato a un vector que contenga la información de “yaw, roll i pitch”:

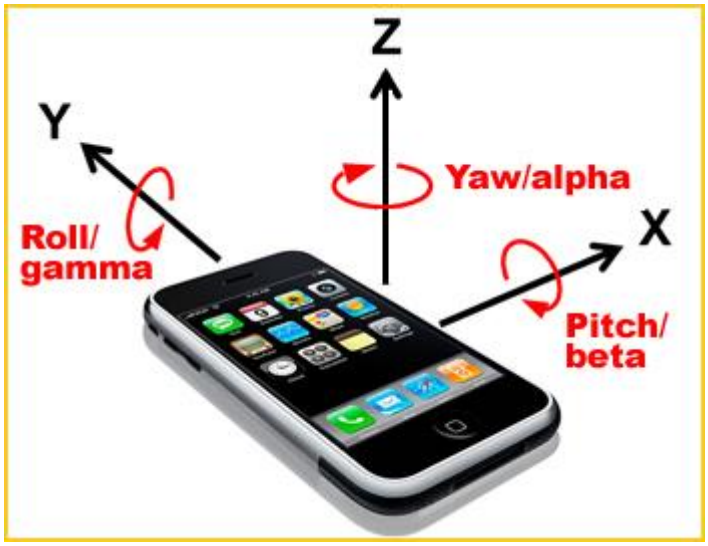


Figura 21: Representación de roll, yaw y pitch

La transformación se puede realizar mediante el siguiente fragmento de código:

```
Mat projMatrix = new Mat();
Mat col = Mat.zeros(new Size(1, 3), CvType.CV_64F);
ArrayList<Mat> matList = new ArrayList<Mat>();
matList.add(R);
matList.add(col);
Core.hconcat(matList, projMatrix);
Calib3d.decomposeProjectionMatrix(projMatrix, cameraMatrix,
                rotMatrix, transVect, rotMatrixX, rotMatrixY, rotMatrixZ,
                eulerAngles);
System.out.println(eulerAngles.dump());
```

3.7. Protocolo para la transmisión de datos

Se debe encontrar la manera fiable de efectuar la transmisión de datos de las luces al dispositivo móvil. Para ello, se debe establecer un protocolo bajo el cual se efectúa la transmisión, por lo tanto se ajustaran la velocidad de transmisión y la codificación.

3.7.1. Velocidad de transmisión

En el sistema, la velocidad de transmisión puede venir limitada por dos factores principalmente:

- **Resolución de la cámara:** Para poder capturar los bits se debe disponer como mínimo un píxel por cada bit transmitido (preferiblemente de 3 a 4 pixeles es lo ideal). Entonces, la velocidad de transmisión puede verse limitada por la resolución de la cámara. Para el dispositivo Nexus 5, se dispone de una resolución en la cámara frontal de 960*1280 px.
- **Mínimo tiempo de exposición:** el mínimo tiempo de exposición puede que no sea suficientemente bajo para establecer una separación precisa entre las distintas columnas de 0 y 1 de la luz dificultando la transmisión a alta velocidad. Este

problema puede verse mejor ilustrado en la siguiente imagen:

Figura 22: Se puede observar como las líneas blancas son ligeramente más grandes que las negras, a pesar de estar transmitiendo 101010

Tras hacer varias pruebas en el laboratorio se ha encontrado que una buena opción de velocidad de transmisión de los datos es a 10000 bps. Se debe tener en cuenta que, cuanto más lejos esté el panel LED de la cámara, menos bits de información se podrán transmitir hacia el móvil, por eso es adecuado situarse en el límite superior de velocidad de transmisión en el que se pueda realizar una lectura fiable.

Cabe destacar que el número de bits que se puedan transmitir es de vital importancia a la hora de escoger una codificación con la que enviar los datos.

3.7.2. Códigos usados para la transmisión de los datos

Para transmitir el identificador o la posición de cada luz, se deben tener en cuenta lo siguiente:

- Dado que dentro del panel LED los bits se pueden presentar en cualquier orden, el código debe poder ser descodificado aunque no se sepa su posición de inicio
- El código debe tener un número similar de 1 y 0 para todos los identificadores con el objetivo de no afectar a la intensidad de iluminación

Para conseguir tales propósitos, se sugieren dos tipos de códigos: codificación Manchester y codificación cíclica.

3.7.2.1. Codificación Manchester

La codificación Manchester consiste en sobreponer el mensaje que se quiere enviar a una señal de clock al doble de frecuencia. De esta manera se consigue substituir cada bit de información por 2 bits de codificación.

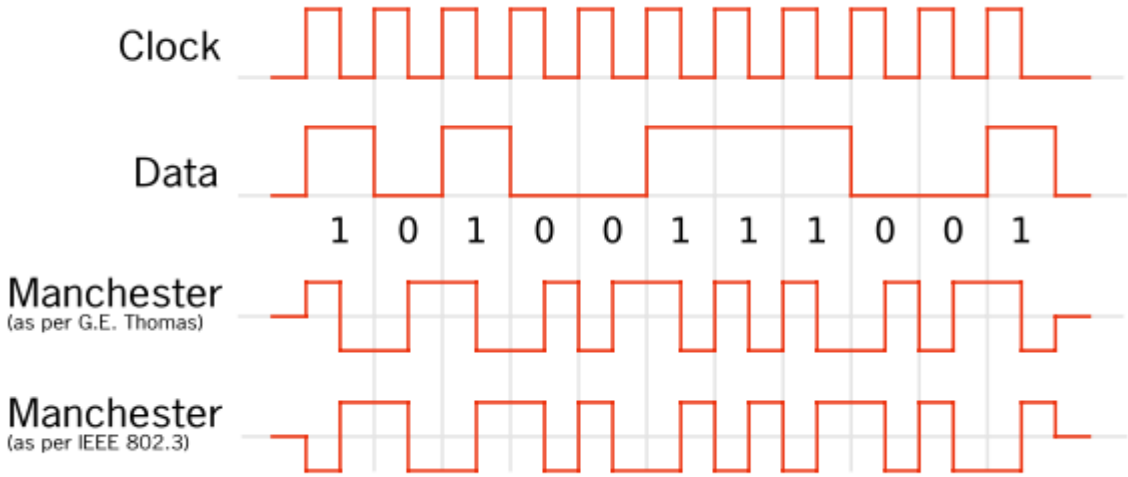


Figura 23: Ejemplo codificación Manchester

Como bien se observa en la figura superior, en el mensaje codificado nunca se obtienen más de tres bits iguales de manera consecutiva. De esta manera, se podría usar este conocimiento para establecer una secuencia de inicialización para nuestro mensaje (por ejemplo, poner 3 ceros o 3 unos seguidos).

Este sistema presenta tanto ventajas como desventajas:

- Ventajas:
  - Presenta un número igual de unos y ceros (exceptuando la secuencia de inicialización), por lo tanto, la cantidad de luminosidad no depende de la palabra que se transmita en ese preciso momento.
  - No es necesario que el receptor conozca la velocidad a la que se transmiten los datos para poder leer la codificación.
- Desventajas
  - La eficiencia es del 50%: la palabra codificada ocupa el doble que la palabra sin codificar.
  - Para poder realizar la lectura, se debe poder leer cómo mínimo dos secuencias de inicialización, por lo tanto se tendrían que transmitir dos mensajes en el lugar del que se dispone para realizar la transmisión.

Así pues, para poder enviar X bits de información necesitaremos 4*X bits codificados (más las cabeceras de inicialización):

X * 2 (propios de la codificación Manchester) * 2 (para poder captar dos mensajes de inicialización) + mensajes de inicialización

3.7.2.2. Codificación cíclica

Este tipo de codificación puede ser muy útil al trabajar con identificadores transmitidos de manera periódica. Los códigos no tienen ni inicio ni final. Pertenecen todos a una circunferencia de longitud L(C). Véase en la siguiente figura:

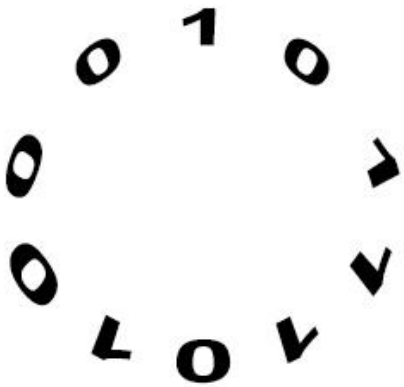


Figura 24: Ejemplo codificación cíclica

Como se puede ver, en este caso todas las siguientes palabras pertenecerían al mismo mensaje codificado:

0010111010, 0101110100, 1011101000, 0111010001, 110100101, 101001011, 0100010111, 100010110, 0001011101

Para darles uso, se guardaría la misma base de datos tanto en el emisor como en el receptor, conteniendo una de las posibles codificaciones asociada a su mensaje (que se llamará palabra codificada madre). Para realizar la lectura se haría el siguiente proceso (siempre cortando el mensaje por un múltiplo de la longitud del código):

```
mensaje = 10100

p_baseDatos = 00101

mensaje += mensaje //ahora mensaje seria 1010010100
```

Se probaría para todas las palabras guardadas en la base de datos. Se puede ver el código usado para la lectura:

```
int codeLenght = (int)bits.size()/codeSize;
for(int i = 0; i < codeSize*codeLenght; i++){
data += bits.get(i);
}
data+=data;
Iterator<Entry<String, Point3>> it =
database.entrySet().iterator();
while(it.hasNext()){
Entry<String, Point3> pair = (Entry<String,
Point3>)it.next();
if(data.contains(pair.getKey())){
return pair.getValue();
}
}
```

Este código, como el anterior, presenta ventajas y desventajas:

- Ventajas:
  - No necesita cabecera de inicialización para poder ser leído
  - Se puede aprovechar todo el ancho del panel LED para la transmisión de datos (al no tener punto de inicialización, para poder leerse no se necesita nada más que un mensaje igual de largo que la longitud del código)
- Desventajas
  - No se pueden usar todas las palabras de la longitud seleccionada cómo palabras código diferentes
  - No tiene por qué contener un número uniforme de 0 y 1, haciendo que la cantidad de iluminación pueda variar según el identificador de cada luz.

Como se puede ver, los dos tipos de codificación tienen ventajas e inconvenientes el uno respecto al otro.

Se muestra a continuación una tabla dónde se pueden observar cuántos códigos se disponen para cada tipo en función del número de bits que se visualicen en el panel LED:

Bits panel LED	Bits Manchester	Bits Cíclica	Id Manchester	Id Cíclica
1	-	1	-	2
2	-	2	-	3
3	-	3	-	4
4	-	4	-	6
5	-	5	-	8
6	-	6	-	14

7	-	7	-	20
8	-	8	-	36
9	-	9	-	60
10	2	10	2	108
11	2	11	2	188
12	3	12	2	352
13	3	13	2	632
14	4	14	4	1182
15	4	15	4	2192
16	5	16	4	4116
17	5	17	4	7712
18	6	18	8	14602

Tabla 1: Comparación codificación Manchester-Cíclica

Cabe destacar que no todos los identificadores de codificación cíclica podrán utilizarse ya que algunas contendrán demasiados ceros (haciendo que la intensidad lumínica del panel quede afectada).

En el caso de transmitir la posición será más útil emplear un código Manchester ya que existe una transformación directa entre la palabra a transmitir y la palabra código (siempre que se dispongan de suficientes bits para realizar la transmisión), la decodificación es directa sin necesidad del uso de un diccionario.

Para la transmisión de identificadores es más conveniente usar codificación cíclica, ya que permite transmitir un número mucho mayor de palabras para el mismo número de bits.

3.8. Desarrollo de la aplicación

Con el objetivo de presentar al usuario final las funciones programadas, a la vez que permitir el testeo, se ha escrito una aplicación que calcula la posición del dispositivo a partir de una fotografía tomada en el momento, en la que aparezcan 4 o más paneles LED trasmitiendo información.

La aplicación consiste de diversas pantallas las cuales se describen a continuación:

- Pantalla Menú:

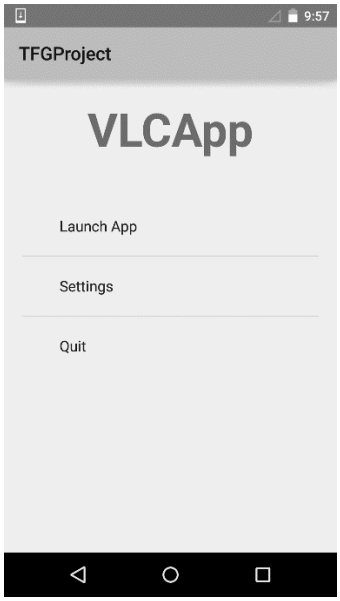


Figura 25: Pantalla Menú

Nos permite navegar por las funciones de la aplicación. Desde esta pantalla podremos acceder a las pantallas de la aplicación principal, ajustes a la vez que cerrar la aplicación de manera segura.

En segundo plano se inicializan las variables compartidas por la aplicación.

- Pantalla Ajustes:

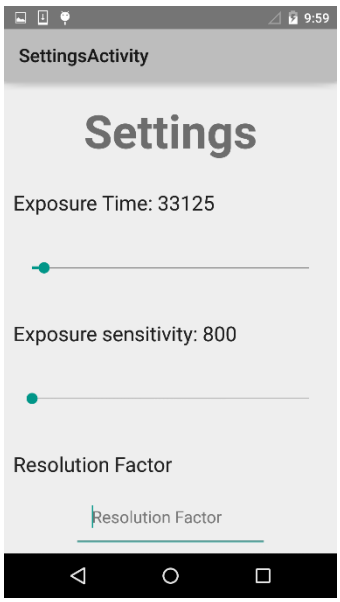


Figura 26: Pantalla Ajustes

Nos permite ajustar los diferentes parámetros para determinar cuáles son los más adecuados: tiempo de exposición, sensibilidad, factor de resolución (permite trabajar con imágenes con menos resolución para acelerar la ejecución del algoritmo), velocidad de transmisión y tamaño del código usado (los dos últimos para el uso con codificación cíclica).

- Pantalla principal de la aplicación:

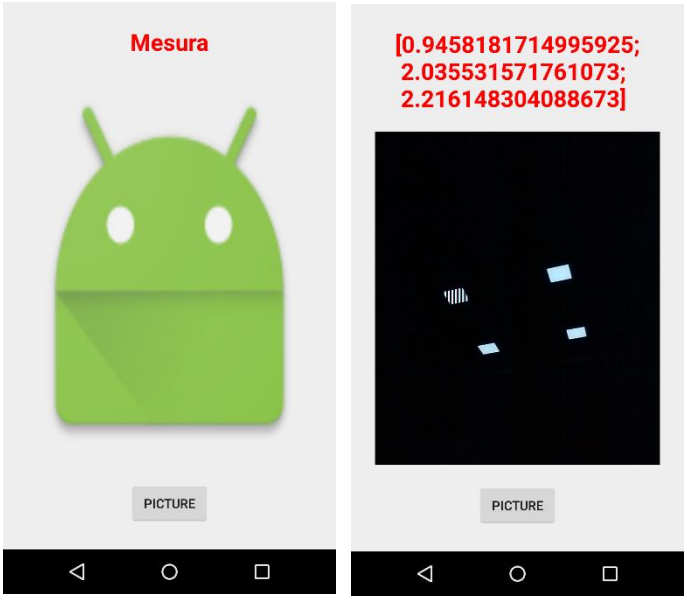


Figura 27: Pantalla principal

Esta pantalla permite ejecutar la aplicación: cuando se pulsa en el botón “Picture”, el móvil toma una imagen. Es a partir de esta imagen se hacen los cálculos pertinentes para la localización del dispositivo. Una vez se ha obtenido la posición y la orientación, estos parámetros se muestran por pantalla, juntamente con la imagen tomada (para poder realizar ajustes si no han aparecido 4 luces en la imagen).

4. Resultados

Cómo se ha explicado en el apartado 3.2, se ha utilizado el laboratorio de Telemática como centro para el testeo del software. Para hacer el cálculo de precisión se han instalado 4 paneles LED en el techo a una altura de 2.40 metros, situando el móvil en el suelo. De esta manera se simula una distancia entre el móvil y los paneles similar a la que nos encontraríamos en la mayoría de edificios. En el suelo se han marcado las posiciones de los paneles LED, a la vez que una cuadrícula para medir la precisión de la localización.

Para testear el software, se coloca el dispositivo en los puntos marcados en el suelo, con diferentes orientaciones y se guardan la posición real y la calculada por el dispositivo en una tabla, junto con las orientaciones. Se adjunta un fragmento de la tabla a continuación:

Posición real				**Posición calculada**		
0	0	2,39		0,02	0,06	2,34
0	0	2,39		0,03	0,04	2,36
0	0	2,39		0,03	0,05	2,34
0	0	2,39		0,02	0,03	2,35
0	-0,3	2,39		0,01	-0,27	2,36
0	-0,3	2,39		0,01	-0,25	2,39
0	-0,3	2,39		0,03	-0,256	2,36
0	-0,3	2,39		0,04	-0,27	2,368
0	-0,6	2,39		0	-0,6	2,36
0	-0,6	2,39		0	-0,6	2,36
0	-0,6	2,39		0	-0,61	2,36
0	-0,6	2,39		0,01	-0,61	2,358
0	-0,9	2,39		0	-0,93	2,34
0	-0,9	2,39		0	-0,94	2,35
0	-0,9	2,39		0,02	-0,88	2,352
0	-0,9	2,39		0	-0,92	2,35
0,3	-0,9	2,39		0,27	-0,96	2,34
0,3	-0,9	2,39		0,32	-0,96	2,351
0,3	-0,9	2,39		0,32	-0,96	2,34
0,3	-0,9	2,39		0,25	-0,94	2,34

Tabla 2: Cálculo de la precisión de localización

Ángulo real				Ángulo calculado		
180	180	0		178	178	0,38
180	180	-90		178	178	-90
180	180	180		179	179	179
180	180	90		177	177	91
180	180	0		181	178	-0,82
180	180	-90		177	177	-89
180	180	180		177	179	179
180	180	90		178	179	90
180	180	0		182	177	0,94
180	180	-90		177	177	-90
180	180	180		177	181	179
180	180	90		182	180	90
180	180	0		179	177	1,64
180	180	-90		177	178	-87
180	180	180		179	178	182
180	180	90		182	178	90

Tabla 3: Cálculo del error de orientación

Se han realizado 176 medidas de posición (96 con el sensor normal y 80 con el complemento gran angular) y 96 de orientación (16 con el sensor normal y 80 con el complemento gran angular) para poder obtener un cálculo fiable de error. Como se puede observar, se realizan diversas medidas en cada punto para poder valorar cuál es la estabilidad del sistema de localización.

4.1. Precisión del sistema de localización

Se ha obtenido un error medio de posición de 0,07943172 metros. Para caracterizar mejor el sistema se han calculado las gráficas de densidad de probabilidad del error de posición:

Divisiones histograma	*Frecuencia*	*Cumulative %*
0	0	0,00%
0,01	0	0,00%
0,02	0	0,00%
0,03	0	0,00%

0,04	4	4,17%
0,05	10	14,58%
0,06	11	26,04%
0,07	9	35,42%
0,08	17	53,13%
0,09	15	68,75%
0,1	10	79,17%
0,11	6	85,42%
0,12	4	89,58%
0,13	4	93,75%
0,14	4	97,92%
0,15	1	98,96%
0,16	0	98,96%
0,17	1	100,00%

Tabla 4: Histograma de error de posición

Se puede observar que en el 90 % de los casos el error es menor a 0.12 metros y que no hay ningún caso en el que supere los 17 cm.

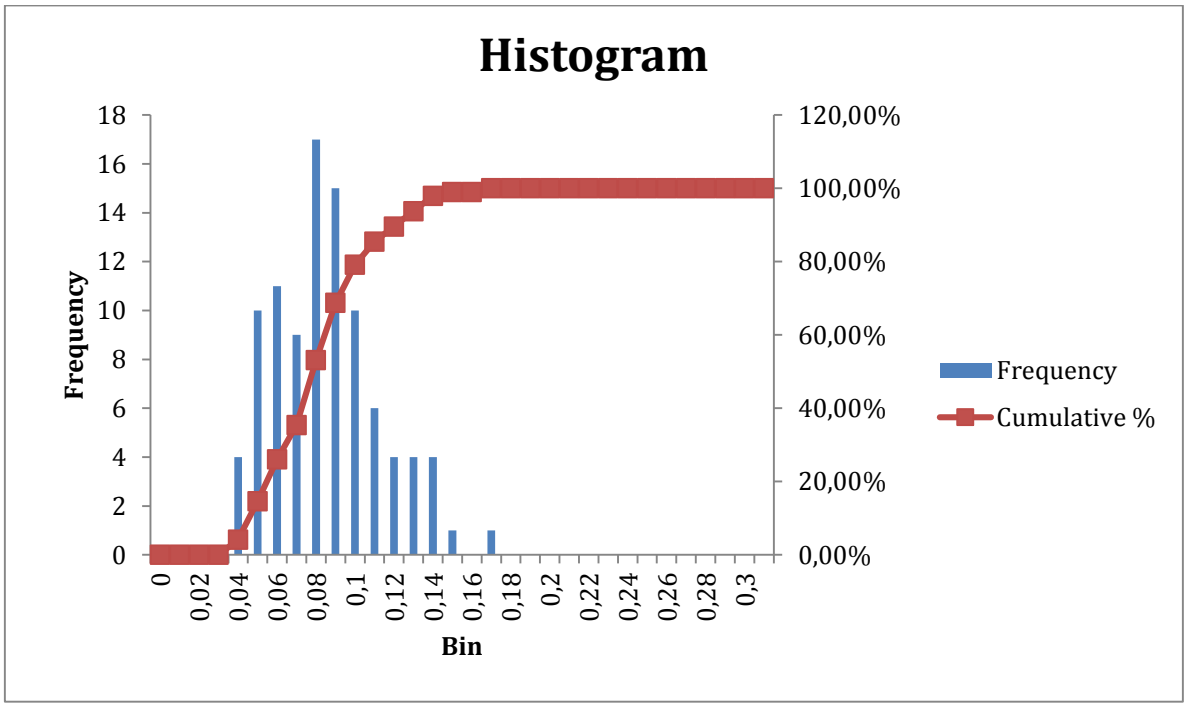


Figura 28: Histograma y cálculo de la probabilidad de error acumulada

También se quiere valorar la variación en el error en función de la distancia a la que el dispositivo se sitúe de los paneles LED, para ver si la precisión se conserva o disminuye a medida que nos alejamos a acercamos a los paneles (de esta manera podemos determinar la densidad de paneles LED que necesitamos en el espacio en el que queramos aplicar la localización):

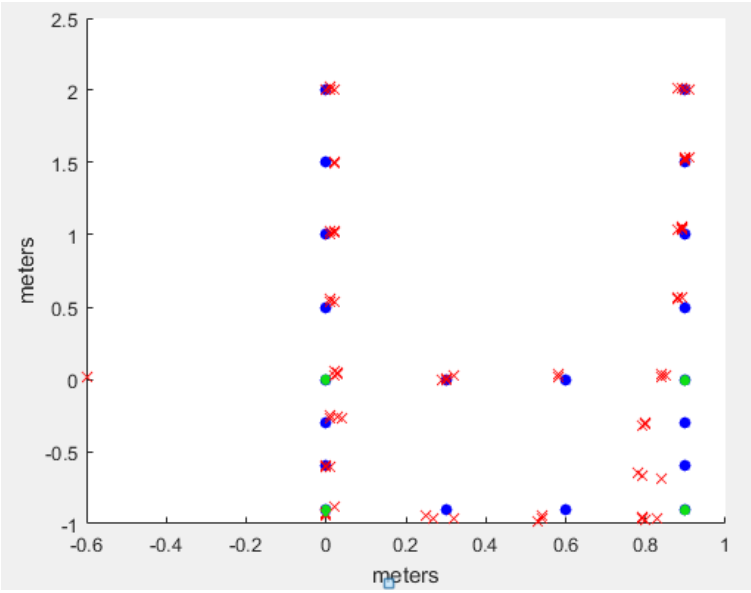


Figura 29: Posiciones en el plano XY

En la figura superior los círculos verdes representan los lugares dónde se sitúan los focos LED en el techo, y los círculos azules las posiciones dónde se ha medido la precisión del software. Se puede ver cómo el sistema funciona incluso si el dispositivo no se sitúa debajo de los puntos de iluminación. De hecho se obtiene mayor precisión de localización cuando la dirección de la normal del móvil es distinta a la del techo.

Se ha medido también la precisión del cálculo de la orientación, obteniendo el siguiente histograma:

Divisiones histograma	***Frecuencia***	***Cumulative %***
0	0	0,00%
0,2	0	0,00%
0,4	0	0,00%
0,6	0	0,00%
0,8	0	0,00%
1	0	0,00%
1,2	0	0,00%

1,4	0	0,00%
1,6	0	0,00%
1,8	1	6,25%
2	1	12,50%
2,2	0	12,50%
2,4	2	25,00%
2,6	0	25,00%
2,8	0	25,00%
3	4	50,00%
3,2	0	50,00%
3,4	2	62,50%
3,6	1	68,75%
3,8	1	75,00%
4	0	75,00%
4,2	0	75,00%
4,4	3	93,75%
4,6	0	93,75%
4,8	1	100,00%
5	0	100,00%
More	0	100,00%

Tabla 5: Tabla del histograma del error de orientación

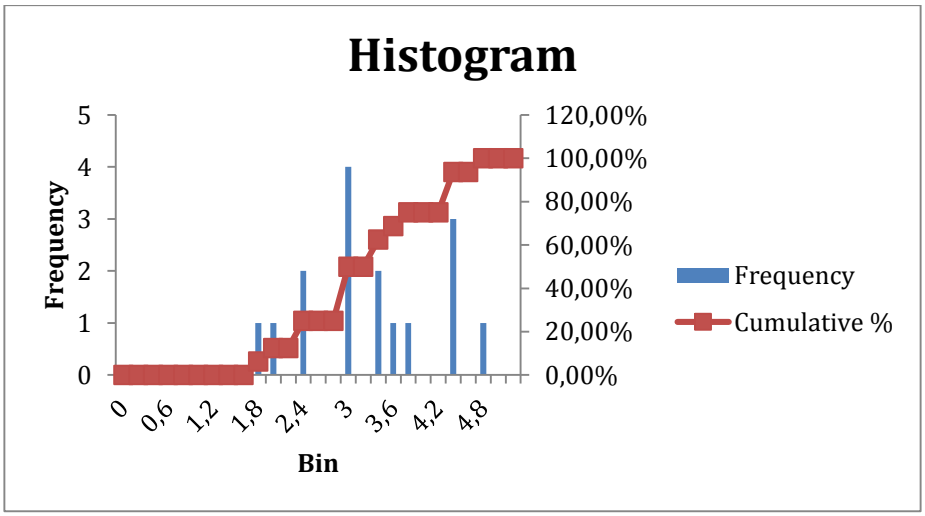


Figura 30: Histograma del error de orientación

El error medio de orientación es de 3,21455708 grados, y la totalidad de las medidas se concentran por debajo de los 4.8 grados.

Se ha calculado también la precisión del sistema cuándo se emplea un accesorio de gran angular para la lente de la cámara, permitiendo aumentar el campo de visión. Se ha obtenido un error medio de posición de 0.088 metros y de orientación de 3.83 grados.

4.2. Velocidad de ejecución del sistema

Se ha medido el tiempo de ejecución del programa y se ha promediado entre 21 ejecuciones obteniendo un valor de 600 milisegundos por ejecución. Esto permitiría poder ejecutar el algoritmo hasta 100 veces por minuto.

4.3. Codificación

Se ha medido de qué manera disminuyen el número de bits por panel en función de la distancia a una velocidad de 10000 bps:

Distancia	Numero Bits
1,82	38
1,87	38
2,34	30
2,36	30
2,37	26
2,42	28
2,43	26
2,43	26
2,45	26
2,52	26
2,53	26
3,18	18
3,46	14
3,83	12

Tabla 6: Número de bits en función de la distancia (10000 bps)

Se obtiene la siguiente gráfica:

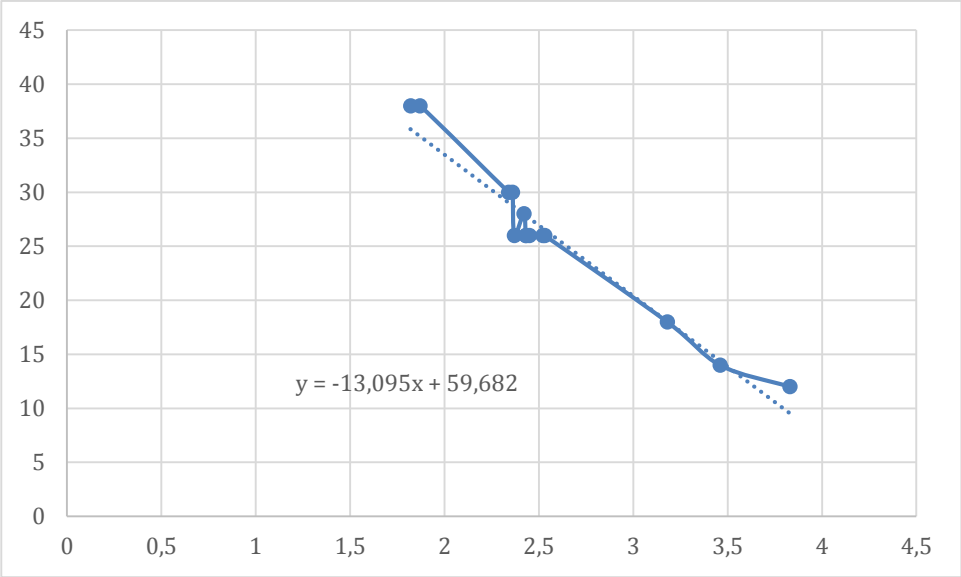


Figura 31: Representación bits (distancia)

La cantidad de bits disponibles en función de la distancia disminuye de manera lineal.

Como se puede observar en la figura superior, a una distancia de aproximadamente 2.50 metros, se pueden visualizar hasta 26 bits por panel a una velocidad de 10000 bps (el valor de bits es inversamente proporcional a la velocidad de transmisión).

Suponiendo que no se visualizaran fuentes a una distancia mayor de 3.50 metros, dispondríamos de 13 bits con los que realizar la transmisión. Esto supondría la posibilidad de transmitir hasta 682 identificadores.

La cantidad de identificadores depende directamente del tamaño del panel LED que se use (en el caso explorado el panel es de 17 x 17 cm).

5. Presupuesto

5.1. Componentes del prototipo, lista de software y hardware:

El coste del proyecto viene representado en las siguientes tablas:

TIPO DE PRODUCTO	PRODUCTO	CANTIDAD	PRECIO (€)	TOTAL	VIDA ÚTIL (AÑOS)	PERIODO USADO (MESES)	VALOR DEPRECIATIVO (€)
PANELES LED	BUSCAR NOMBRE	4	23,5	94	10	5	2,115
CONTROLADORA	BUSCAR NOMBRE	4	11,5	46	5	5	2,07
DRIVER	BUSCAR NOMBRE	4	70	280	5	5	12,6
DISPOSITIVO MÓVIL	Nexus 5	1	345	345	5	9	62,1
SO	Ubuntu 14.04	1	0	0	-	9	-
LIBRERÍA	OpenCV 3.0.0	1	0	0	-	9	-
ORDENADOR	HP Pavilion 2080es	1	800	800	5	9	144

FUNCIÓN	Horas estimadas	Coste/hora	Total (€)
INGENIERO JUNIOR	700	8	5600

Entonces el coste total del proyecto es de 5792,855 €.

5.2. Análisis de viabilidad económica

El coste de un panel LED convencional es de 23,5 €. Si se le quieren incorporar las funciones descritas en la tesis, se debe incrementar su coste hasta 105 €. Esto supone un incremento del 346,80 %.

Teniendo la mejora de precisión que supone el sistema por encima de todos los demás disponibles en el mercado [5] [10] [4] [2] [6] [7] [3] [9] y que la mayoría de estos sistemas suponen también un coste de implementación elevado, se entiende que es viable económicamente proporcionar el sistema de localización estudiado en un espacio interno.

6. Conclusiones y desarrollo futuro:

Tras realizar el desarrollo del proyecto, se pueden extraer una serie de conclusiones que se exponen continuación, separándolas en función del apartado al que pertenecen:

6.1. Uso de la API Camera2:

El uso de la interfaz Camera2 ha sido de vital importancia en el momento de la obtención de imágenes de la cámara. Sin el uso de esta API no se hubieran podido capturar las imágenes necesarias para el posterior desarrollo.

Se ha podido desarrollar la aplicación en dispositivos Android (hasta el momento, la flexibilidad necesaria sólo estaba disponible para dispositivos Windows Phone [9]). A medida que esta librería sea implementada en los dispositivos, el software de detección será compatible automáticamente con ellos, haciendo que la tecnología pueda ser explotable a gran escala.

Tanto para el uso de la cámara frontal como la posterior del dispositivo Nexus 5, se han podido obtener velocidades de transmisión relativamente altas sin errores. Para el caso de la cámara frontal de 10000 bps.

6.2. Codificación

La utilización de la codificación cíclica ha facilitado mucho el envío de datos entre los paneles LED y el dispositivo móvil. Elimina la limitación en el momento de distribuir identificadores para los paneles LED en edificios grandes. Se soluciona el problema de la secuencia de inicialización (necesaria para la mayoría de codificaciones), pudiendo transmitir muchos más identificadores que en resto de sistemas explorados [9].

6.3. Precisión

La precisión obtenida es muy elevada, del orden de centímetros. Este hecho permite que el sistema propuesto pueda ser utilizado en aplicaciones que requieran el mínimo error posible, como sistemas de guiado de invidentes. También cabe destacar la obtención de orientación, que proporciona más información al desarrollador de la aplicación, pudiendo proporcionar información mucho más focalizada.

Se confirma que la localización mediante el uso de técnicas de VLC combinadas con fotometría es un método realista y que garantiza buenas medidas de precisión y de estabilidad. El análisis por fotometría se ha mostrado superior a todos los sistemas explorados en el estudio de antecedentes, sea cual sea la tecnología usada.

El uso de un complemento de gran angular no afecta en gran medida el cálculo de localización y permite obtener un campo de visión muy superior, permitiendo más facilidad para el usuario.

6.4. Tiempo de ejecución

El tiempo de ejecución es suficientemente bajo como para poder ejecutar el algoritmo en tiempo real. Este hecho es de gran utilidad a la hora de realizar sistemas de guiado. También es adecuado porqué proporciona una mejor experiencia de usuario.

6.5. Comparación con principales sistemas de localización con VLC

Se ha realizado una tabla comparativa entre los principales aspectos que diferencian el sistema propuesto en esta tesis y los sistemas de localización explorados en el apartado 2.3.

Sistema	Precisión	Cantidad de identificadores	Velocidad de ejecución
Epsilon [10]	~ 0.4 m - º	No usa base de datos	-
Luxapose [9]	~ 0.1m 3 º	25 – 30	9 s
Sistema propuesto	~0.08 m 3.2 º	682 a 3.5 metros	0.6 s

Tabla 7: Comparación sistemas localización VLC

6.5.1. Epsilon [10]

El sistema propuesto en esta tesis logra una precisión de localización muy superior a Epsilon. Además, no necesita un hardware adicional acoplado al dispositivo para poder capturar la intensidad de señal ya que no trabaja con ese valor. Epsilon tampoco permite el cálculo de la orientación.

6.5.2. Luxapose [9]

El sistema propuesto logra una precisión similar (aunque ligeramente superior) que Luxapose tanto en posición como en orientación. Además, dado que realiza la ejecución de los algoritmos en el propio dispositivo, consigue una velocidad de ejecución mucho mayor, a la vez que permitiendo ejecutar el sistema sin el soporte de un ordenador.

El sistema Luxapose no permite transmitir bits a una velocidad tan elevada como el propuesto en la tesis, de manera que dispone de menos bits por panel. Además,

disponiendo del mismo número de bits, la codificación propuesta en esta tesis permite un número muy superior de identificadores.

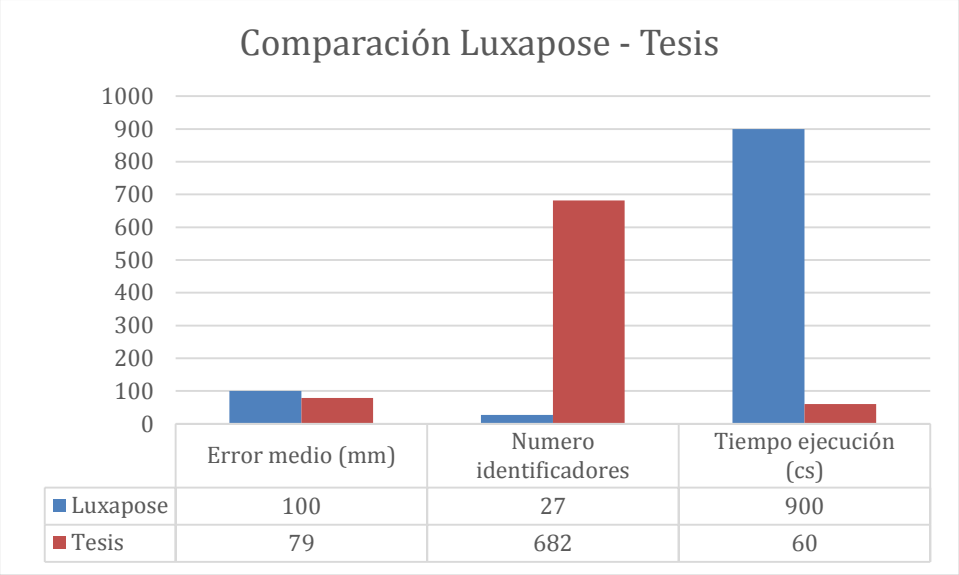


Figura 32: Comparación Luxapose - Tesis

6.6. Desarrollo futuro

Las técnicas desarrolladas en este proyecto proporcionan una precisión suficiente como para desarrollar sistemas de guiado de edificios. Para estos sistemas, es de gran utilidad poder extraer información de la orientación. Además, para sistemas que muestren información localizada, este parámetro también cobra gran importancia.

7. Bibliografía

[1] I. T. H. O. T. F. T. Y. S. A. M. A. T. H. K. Y. K. K. a. S. K. Takaya Yamazato, “Image-Sensor-Based Visible Light Communication for Automotive Applications,” *IEEE Communications Magazine,* 2014.

[2] D. R. Mautz, “Indoor Positioning,” Institute of Geodesy and Photogrammetry, Zurich, 2012.

[3] W. F. Storms, “Magnetic Field Aided Indoor Navigation,” DEPARTMENT OF THE AIR FORCE, Ohio, 2009.

[4] J. D. H. a. T. B. Manon Kok, “Indoor positioning using ultrawideband and inertial measurements,” 2015.

[5] J. W. Konstantin Klipp, “Low cost high precision indoor localization system fusing inertial- and magnetic field sensor data with radio beacons,” Berlin, 2015.

[6] S. K. A. H. T. a. G. T. H. S.L. Ting, “The Study on Using Passive RFID Tags for Indoor Positioning,” *International Journal of Engineering Business Management,* vol. 3, 2011.

[7] G. S. Solomon Chan, “INDOOR LOCALIZATION USING WI-FI BASED FINGERPRINTING AND TRILATERATION TECHIQUES FOR LBS APPLICATIONS,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 38, 2012.

[8] G. B. I. S. J. M. T. A. B Cook, “Indoor Location Using Trilateration Characteristics,” University College London, Siemens Communications, Nottingham Trent University.

[9] P. P. K.-J. H. a. P. D. Ye-Sheng Kuo, “Luxapose: Indoor Positioning with Mobile Phones and Visible Light,” *MobiCom,* 7-11 2014.

[10] P. H. C. P. G. S. F. Z. Liqun Li, “Epsilon: A Visible Light Based Positioning System,” in *Proc. of the 11th USENIX*, 2014.

Glosario

VLC: Visual Light Communication

SVM: Support Vector Machine