



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERIA ELECTRÓNICA

**SISTEMA DE GENERACIÓN DE MOSAICOS 2D  
PARA ROBOTS MÓVILES A PARTIR DE VIDEO  
MONOCULAR**

Por:

Victor Yovanni Garcia Carmona

Realizado con la asesoría de:

José de la Cruz Cappelletto Fuentes

**PROYECTO DE GRADO**

Presentado ante la Ilustre Universidad Simón Bolívar  
como requisito parcial para optar al título de  
Ingeniero Electrónico

Sartenejas, Marzo de 2018



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA ELECTRÓNICA

ACTA FINAL PROYECTO DE GRADO

**SISTEMA DE GENERACIÓN DE MOSAICOS 2D PARA ROBOTS  
MÓVILES A PARTIR DE VIDEO MONOCULAR**

Presentado por:

**Victor Yovanni Garcia Carmona**

Este Proyecto de Grado ha sido aprobado por el siguiente jurado examinador:

---

José de la Cruz Cappelletto Fuentes

---

Novel Antonio Certad H.

---

Gerardo Fernandez López

**Sartenejas, @día de Mayo de 2018**



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA ELECTRÓNICA  
**SISTEMA DE GENERACIÓN DE MOSAICOS 2D PARA ROBOTS  
MÓVILES A PARTIR DE VIDEO MONOCULAR**  
PROYECTO DE GRADO  
PRESENTADO POR:  
Victor Yovanni Garcia Carmona, Carnet: 12-10738

## RESUMEN

Al realizar tareas de exploración para el análisis del suelo desde espacios aéreos o del fondo marino, es muy común emplear sistemas de adquisición basados en captura de videos para su posterior análisis. En la actualidad, el incremento de la tecnología sobre el procesamiento de datos, ha permitido que los algoritmos de visión por computadora coloquen a la cámara como principal sensor para la reconstrucción de entornos recorridos por vehículos móviles. El presente trabajo se encuentra enfocado al análisis y la implementación de distintos algoritmos para la reconstrucción de un mosaico 2D (dos dimensiones), a partir de la información proveniente de una cámara monocular ubicada en la parte inferior de un robot. El robot en cuestión puede realizar recorridos aéreos para realizar la adquisición del video, o incluso trayectorias más desafiantes como serían las aplicaciones subacuáticas. Para esto, se implementarán distintos algoritmos usando técnicas de procesamiento de imágenes y visión por computadora, para la elaboración de un sistema automatizado que permita generar un mapa en dos dimensiones de la trayectoria recorrida, con la menor distorsión posible, mejorando la detección de puntos clave, y optimizando el cálculo de las matrices de transformación para la alineación de imágenes en el mosaico, además de realizar análisis sobre el error de reproyección de dichas imágenes en el mapa del suelo generado.

**Palabras clave:** mosaico, video monocular, puntos clave, matriz de transformación.

# Agradecimientos

# Índice general

Resumen	I
Agradecimientos	I
Índice de Figuras	IV
Lista de Tablas	V
Acrónimos y Símbolos	VI
<b>1. Introduccion</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Justificacion y planteamiento del problema . . . . .	1
1.3. Objetivos . . . . .	2
1.3.1. Objetivo General . . . . .	2
1.3.2. Objetivos Específicos . . . . .	2
1.4. Estructura del trabajo . . . . .	2
<b>2. Estado del Arte</b>	<b>4</b>
2.1. Detección de puntos de interés . . . . .	4
2.2. Proyección de imagenes . . . . .	11
2.3. Fusión de imagenes . . . . .	11
<b>3. Plataformas Experimentales</b>	<b>12</b>
3.1. Seccion 1 . . . . .	12
3.2. Sección 2 . . . . .	12
3.2.1. sub-sección 2 . . . . .	12
<b>4. Algoritmos para la generacion de mosaico</b>	<b>13</b>
4.1. Seccion 1 . . . . .	13
4.2. Sección 2 . . . . .	13
4.2.1. sub-sección 2 . . . . .	13

---

<b>5. Resultados Experimentales</b>	<b>14</b>
5.1. Seccion 1 . . . . .	14
5.2. Sección 2 . . . . .	14
5.2.1. sub-sección 2 . . . . .	14
<b>6. Conclusiones</b>	<b>15</b>
<b>A. @nombreApendice</b>	<b>17</b>
A.1. @sección . . . . .	17
A.1.1. @subsección . . . . .	17
<b>B. @nombreApendice</b>	<b>18</b>

# Índice de figuras

2.1. Puntos de interés . . . . .	5
2.2. Deteccion de esquinas . . . . .	6
2.3. Deteccion de esquinas . . . . .	6
2.4. SIFT - Espacio de escalas . . . . .	7
2.5. SURF - Espacio de escalas . . . . .	8
2.6. ORB - Espacio de escalas . . . . .	9
2.7. KAZE - Espacio de escalas . . . . .	10

# Índice de Tablas



# Acrónimos y Símbolos

<b>SIGLAS</b>	<b>S</b> iglas <b>I</b> sla <b>G</b> rafo <b>L</b> aos <b>A</b> ve <b>S</b> erpiente
<b>ACM</b>	<b>A</b> ssociation for <b>C</b> omputing <b>M</b> achinery

---

$\iff$	doble implicación, si y sólo si
$\Rightarrow$	implicación lógica
$[u := v]$	sustitución textual de $u$ por $v$

## ***Dedicatoria***

*A @personasImportantes, por @razonesDedicatoria.*

# Capítulo 1

## Introduccion

### 1.1. Antecedentes

Antecedentes

### 1.2. Justificacion y planteamiento del problema

Para esta tarea, es común el uso de herramientas que requieren de una gran intervencion por medio del usuario para la elaboración de un mapa.

Atendiendo a esta necesidad, es necesario contar con un sistema que permita realizar la reconstruccion del suelo que recorre un vehiculo con la menor interaccion posible del ser humano.

## 1.3. Objetivos

### 1.3.1. Objetivo General

Analizar e implementar un sistema automatizado que permita la reconstrucción de un mapa en dos dimensiones, del entorno recorrido por robot, aereo o submarino.

### 1.3.2. Objetivos Específicos

- Análisis comparativo de metodos vigentes en la reconstruccion de mosaicos 2D, a partir de imagenes y videos de entrada.
- Implementacion de modulo de preprocesamiento y correccion de entrada.
- Analisis comparativo de metodos de deteccion y description de puntos clave.
- Implementación de módulo de alineación de imagenes mediante la detección de puntos clave.
- Cuantificar el error de reproyeccion y distorsion en los modelos 2D generados.

## 1.4. Estructura del trabajo

Luego de presentar el planteamiento del problema y la descripción del proyecto, la presente investigacion se encuentra dividida en X capitulos, organizados de la siguiente manera:

En el **Capítulo 2** se presenta una revisión del estado del arte, en el cual se exponen los distintos metodos para la detección e identificacion de puntos de interés en una imagen. Ademas, se presentan diversos algoritmos utilizados en la actualidad para la reduccion de errores en el calculo de matrices de transformación. Para terminal, se muestran las tecnicas que se utilizan para la fusion entre las imagenes, que logran reducir errores en el color, y la calidad del mapa final.

---

Los resultados experimentales son mostrados en el ***Capitulo X***, en este se contemplan los modelos generados de forma automatica del suelo, tanto terrestre como submarino. Además se presenta un analisis comparativo de estos resultados sobre los que proveen herramientas no automatizads.

# Capítulo 2

## Estado del Arte

En este capítulo se presenta una revisión teórica del estado actual de las aplicaciones e investigaciones que se han desarrollado en el área de procesamiento de imágenes, aplicado a la construcción de mosaicos. Además se presenta una reseña histórica de la evolución de dichos métodos. Con esto se pretende recuperar y trascender el conocimiento acumulado en esta área de estudio, además de familiarizar al lector con los conceptos básicos, necesarios para la comprensión del presente trabajo.

En primer lugar, se reseña el progreso de los algoritmos de extracción de puntos de interés, seguidamente se presentan avances sobre los módulos de alineación de imágenes. Finalmente, en la última sección se describen los métodos más utilizados en la actualidad para la etapa final del mapeo, como lo son la búsqueda de la mejor línea de corte y corrección final de color.

### 2.1. Detección de puntos de interés

Antes de mencionar la evolución de los algoritmos de detección de puntos de interés, primero es necesario definir que son. Los puntos de interés, puntos claves, o "*features*" (en español: características) como son comúnmente llamados, son puntos en una imagen los cuales tienen patrones específicos, lo que hace que puedan ser fácilmente seguidos o ubicados en otra imagen.



FIGURA 2.1: Ejemplo de caracterización de puntos en una imagen

Atendiendo a la imagen 2.1, se puede observar que se caracterizadas seis áreas de interés. Analizando estos segmentos, vemos que **A** y **B** corresponden con superficies planas, lo que hace que sean muy difícil de identificar la ubicación exacta de esta superficie en la imagen completa. Por otro lado, tenemos las regiones **C** y **D**, las cuales corresponden con bordes en la imagen, si bien, se puede limitar en gran medida el área de búsqueda pero de igual forma es difícil encontrar la ubicación exacta. Finalmente, analizando las regiones **E** y **F** tenemos que corresponden a esquinas de la imagen original, en este caso se puede identificar fácilmente la ubicación exacta de la región en la imagen.

A partir de esta idea, en la cual se consideran las esquinas como regiones fácilmente identificables en una imagen, en 1988 nace el primer algoritmo de detección de puntos de interés [1] llamado Detector de esquinas de Harris.

Tomando la idea planteada previamente, este detector busca la diferencia de intensidad para cada desplazamiento de la ventana de búsqueda, en todas las direcciones. Es decir se detectará una esquina, para aquellas regiones que presenten una alta variación de intensidad al desplazar la ventana estudiada en todas las direcciones. En la figura 2.2 se puede apreciar visualmente como funciona esta ventana de búsqueda.

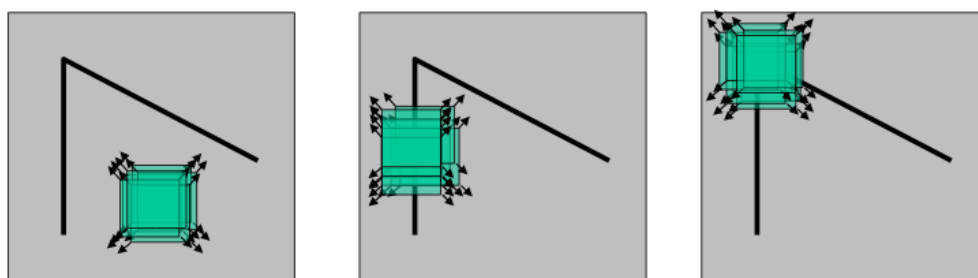


FIGURA 2.2: Desplazamiento de la ventana de búsqueda, para de deteccion de esquinas

Cuando se trabajan con detectores de características, se desea que estos sean invariantes ante la mayor cantidad de variables posibles, es decir, que sean capaces de detectar las mismas características en una imagen a pesar de cambios en la traslación, rotación, escala, variaciones de iluminación, variaciones del punto de vista, entre otras. Si bien el detector presentado anteriormente es invariante ante la traslación y la escala, ya que las esquinas se mantienen como esquinas si son rotadas o movidas, no funciona de la misma forma ante cambios de escala. Como se observa en 2.3, una región considerada como esquina, puede ser considerada plana si es ampliada.

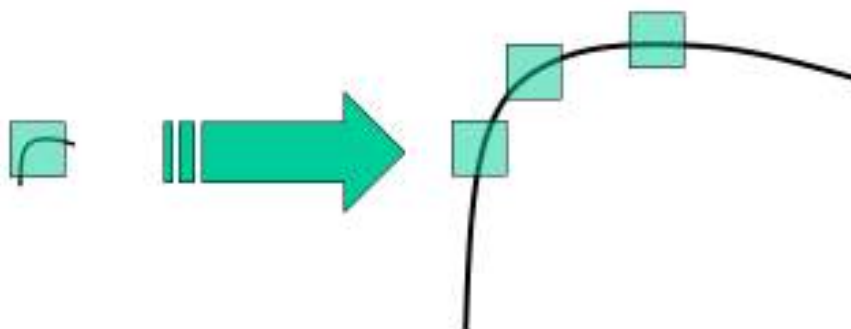


FIGURA 2.3: Desplazamiento de la ventana de búsqueda, para de deteccion de esquinas

Otra característica que se busca en estos algoritmos para lograr ubicar estos puntos claves en una imagen distinta, es la capacidad de describir estos puntos una vez se hayan detectado. Para esto los algoritmos descriptores, una vez tengan la ubicación de los puntos característicos se encargan de convertir la información de su alrededor en una serie de números, o un vector que permita diferenciar un punto clave de otro. Esta información también es necesaria que sea invariante ante las



variablen antes mencionadas, para lograr una identificación eficiente del mismo punto en distintas imágenes.

Partiendo de estos problemas, en 2004 D.Lowe crea el detector *SIFT* (del inglés: Scale Invariant Feature Transform), el cual presenta invarianza ante cambios en la escala. El proceso para la detección y descripción de puntos de interés de este algoritmo, consta de cuatro pasos principales:

En primer lugar, realiza una detección de máximos en el espacio de la escala aplicando la diferencia gaussiana. Para esto, se aplica el filtro gaussiano con distintos tamaños de media (se tienen distintas escalas), luego restando estas imágenes para distintos pares de escalas se logra la diferencia de gaussianas. Posteriormente se buscan los máximos locales a lo largo del espacio (coordenadas X,Y) para cada correspondiente escala. Este proceso de detección se puede visualizar en 2.4.

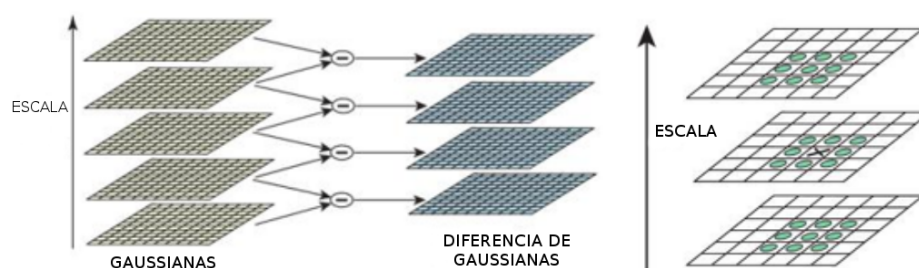


FIGURA 2.4: Detección de máximos en el espacio de la escala

En segundo lugar para la localización de puntos de interés, se descartan los puntos encontrados en el paso anterior que no superen cierto valor de umbral, es decir, que no estén lo suficientemente contrastados con su entorno. Con esta etapa el algoritmo solo toma en cuenta los puntos claves más fuertes por cada escala. Además, con el objetivo de eliminar los bordes suficientemente contrastados que no correspondan con esquinas, el algoritmo usa una matriz hessiana para calcular las curvaturas principales, y así quedarse solo con esquinas.

Para garantizar la invarianza con respecto a la rotación, se toman los píxeles vecinos al punto clave y se calcula la magnitud y dirección del gradiente en esa región. Con esto se hace un histograma de la magnitud del gradiente en cada dirección, donde el pico mayor del histograma indica la orientación. En el caso que exista un pico mayor al 80 % del pico principal, este se utiliza para crear otro punto de interés en la misma posición pero con la distinta rotación. Finalmente

para la descripción por cada punto clave se crea una matriz de  $16 \times 16$  alrededor de este, dividido en 4 subregiones de  $4 \times 4$  pixeles con un histograma de orientaciones para cada uno. Finalmente, el descriptor del punto será el vector con los valores de los histogramas de las regiones  $4 \times 4$  concatenados.

En el año 2006, David G. Lowe desarrolla un nuevo detector [2], el cual está basado en SIFT, pero con modificaciones que aumentan su velocidad de detección. Si bien, sacrifica un poco de rendimiento y precisión, lo hace más provechoso para aplicaciones que demanden mayor velocidad de cómputo, como por ejemplo *SLAM* (del inglés: Simultaneous Localization and Mapping). Las etapas para la extracción de puntos se componen de la siguiente manera:

Como primer paso, en lugar de aproximar el laplaciano de Gauss (LoG) con la diferencia de Gaussianas (DoG) como lo hace SIFT, este algoritmo aproxima LoG con cuadrados para promediar la imagen. La ventaja de aplicar filtros con cuadrados es que con la ayuda de imágenes integrales el cálculo computacional se reduce en gran medida.

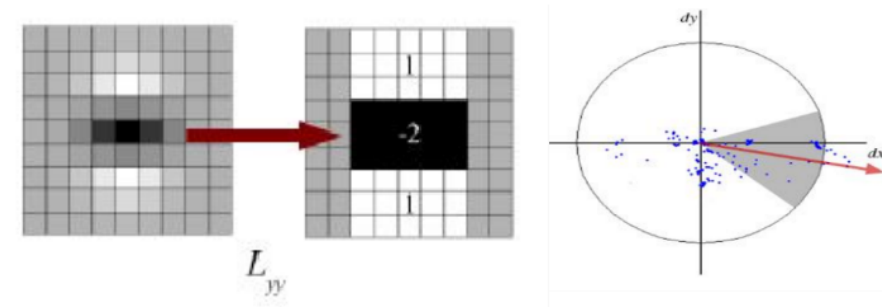


FIGURA 2.5: Detección de máximos en el espacio de la escala

En función de identificar la orientación, el algoritmo utiliza la respuesta wavelet Haar en horizontal y vertical en un vecindario de  $6s$  ( $s$  es la escala del punto clave) píxeles al rededor del punto de interés. Luego estas respuestas son representadas como puntos en el espacio para luego calcular la orientación dominante con la suma de todos los resultados dentro de una ventana deslizante de apertura  $60^\circ$ . En 2.5 se puede visualizar la forma del filtro que aplica el detector, y el vector de orientación en función a la distribución de puntos estudiados.

El siguiente avance importante en los algoritmos de detección aparece en el año 2001 con *ORB* (del inglés: Oriented FAST and Rotated BRIEF) [3], este utiliza una

combinación del detector FAST (del inglés: Features from Accelerated Segment Test) y del descriptor BRIEF (del inglés: Binary Robust Independent Elementary Features), caracterizado por su velocidad, gracias al uso de un descriptor binario y manteniendo un buen rendimiento.

Como se mencionó utiliza el algoritmo FAST el cual consiste en encontrar esquinas evaluando los píxeles en un perímetro circular, de esta forma, un punto será detectado como si la cantidad de píxeles de color opuesto al evaluado supera cierto valor de umbral (ver izquierda en 2.6), posteriormente con el fin de incrementar la cantidad de puntos, el algoritmo de esquinas de Harris es aplicado. También se hace de forma piramidal evaluando varias escalas (al igual que SIFT).

Como el algoritmo FAST no toma en cuenta la orientación, en el ORB se modificó para que calculara la orientación de la siguiente forma: se calcula a partir de la figura de una esquina (correspondiente al punto de interés) el centroide que le corresponde, luego se selecciona la orientación del vector que va desde este centroide hasta el centro de la esquina.

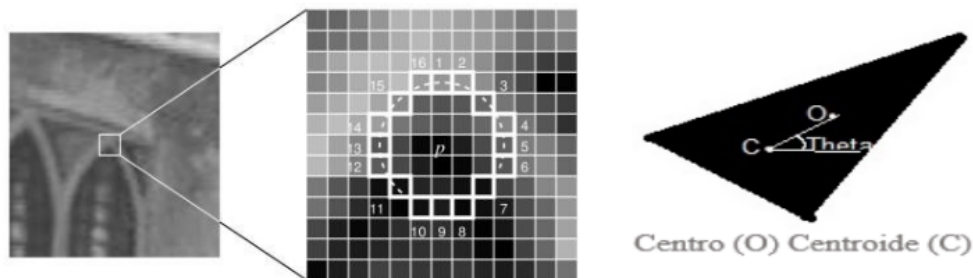


FIGURA 2.6: Detección de máximos en el espacio de la escala

Para el descriptor utiliza BRIEF, a diferencia de los anteriores (SIFT y SURF) este es un descriptor binario y no vectorial. El descriptor BRIEF produce una palabra de  $n$ -bits usando el algoritmo Local Binary Tests (LBT), el problema de esta representación es que no es muy robusta ante cambios en la rotación. Para resolver esto ORB utiliza la información de la orientación previamente calculada para aplicar LBT en esa orientación y obtener la información del descriptor.

AKAZE: Detector y Descriptor - Detector: Este algoritmo está basado en el descriptor KAZE, en cual tiene varias mejoras destinadas a aumentar la velocidad. A diferencia de los anteriores, el algoritmo AKAZE opera en espacios de escala

no lineales, los algoritmos anteriores utilizan filtros gaussianos para detectar las características, mientras que AKAZE realiza una difusión no lineal en la imagen, lo que permite mantener ciertos detalles y eliminar ruido mientras se cambia de escala la imagen, a diferencia del filtro gaussiano que difumina los bordes y detalles de la imagen al cambiar de escala.

- Orientación: Este algoritmo utiliza un descriptor para la orientación similar al que emplea SURF, encuentra la orientación dominante en un área circular de radio  $6s$  ( $s$  es la escala). Para cada muestra del círculo del descriptor se deriva en  $x$  e  $y$ , estos puntos se ubican en un espacio para calcular la orientación dominante en la orientación con más puntos.

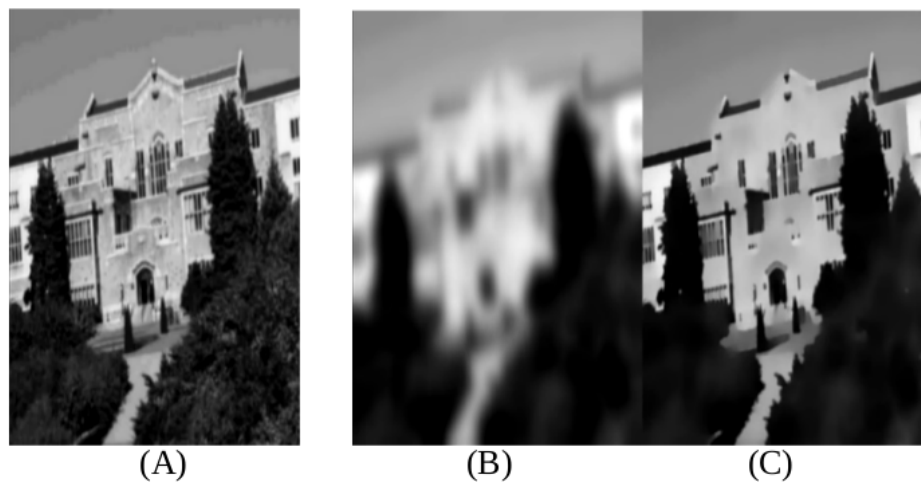


FIGURA 2.7: Detección de máximos en el espacio de la escala

- Descriptor: El descriptor de este algoritmo está basado en diferencia local binaria (LDB Local Difference Binary), el cual sigue el mismo principio que el BRIEF. Al añadirle una modificación para este algoritmo se le llamó M-LBD para que aproveche al máximo la información del espacio de escala no lineal. La modificación consiste en hacer un submuestreo de cada subregión que divide la zona del descriptor, en vez de sacar el promedio de todos los píxeles de la subdivisión, es decir, se tienen muestras de cada subdivisión para distintas escalas.

## **2.2. Proyección de imagenes**

## **2.3. Fusión de imagenes**

mensaje de prueba subsección 2

# Capítulo 3

## Plataformas Experimentales

Resumen del capitulo 3

### 3.1. Seccion 1

mensaje de prueba

### 3.2. Sección 2

#### 3.2.1. sub-sección 2

mensaje de prueba subsección 2;

# Capítulo 4

## Algoritmos para la generacion de mosaico

Resumen del capitulo 4

### 4.1. Seccion 1

mensaje de prueba

### 4.2. Sección 2

#### 4.2.1. sub-sección 2

mensaje de prueba subsección 2

# Capítulo 5

## Resultados Experimentales

Resumen del capitulo 5

### 5.1. Seccion 1

mensaje de prueba

### 5.2. Sección 2

#### 5.2.1. sub-sección 2

mensaje de prueba subsección 2



# Capítulo 6

## Conclusiones

Mensaje de prueba para conclusiones

# Bibliografía

- [1] Chris Harris y Mike Stephens. A combined corner and edge detector. *The Plessey Company pic.*, 1988.
- [2] Herbert Bay Tinne Tuytelaars Luc y Gool. Surf: Speeded up robust features. *European Conference on Computer Vision*, 2006.
- [3] Kurt Konolige y Gary Bradski Ethan Rublee, Vincent Rabaud. Orb: an efficient alternative to sift or surf. *International Conference on Computer Vision*, 2011.

# Apéndice A

## @nombreApendice

### A.1. @sección

#### A.1.1. @subsección

“Saludo”.

**Apéndice B**

**@nombreApendice**