

Universidad de Córdoba
Grado en Ingeniería Informática
(Doble especialidad computación + computadores)

Sistema de mapeo de interiores mediante mediciones láser

Autor:

Manuel Rafael Navarro Fuentes

Director:

Rafael Muñoz Salinas

Junio de 2020



Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
6. Experimentación
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
6. Experimentación
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

1. Introducción

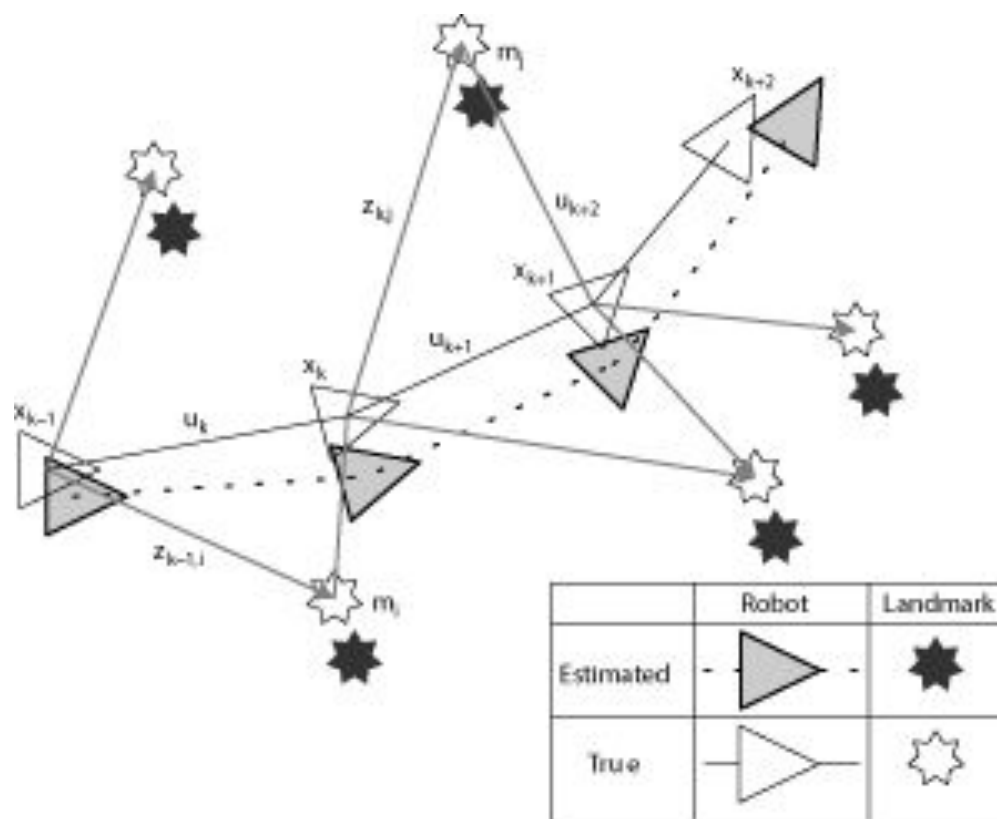
Desafío: Diseñar un sistema de **bajo coste** capaz de realizar **SLAM** (Simultaneous Localization And Mapping).

Utilidad: Entornos industriales, entornos domésticos, etc.



1. Introducción

1.1 SLAM



Índice de contenidos

1. Introducción
2. **Antecedentes**
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
6. Experimentación
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

2. Antecedentes

- Diversas técnicas para resolver **SLAM** (cámaras, gps, LiDAR, otros sensores).

- Robots de limpieza (**conga**).



- Implementaciones en **ROS** (Robot Operating System).

Índice de contenidos

1. Introducción
2. Antecedentes
- 3. Objetivos y restricciones**
4. Recursos
5. Diseño del sistema
6. Experimentación
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

3. Objetivos y restricciones

3.1 - Objetivos principales

- El sistema será capaz de **crear un mapa e identificarse** en este al mismo tiempo.
- El sistema podrá ser **portado por un humano**.
- Se dispondrá de una **interfaz gráfica de usuario**.
- Se podrán **crear y probar simulaciones**.
- El **mapa** podrá ser **exportado**.

3. Objetivos y restricciones

3.2 Restricciones

- Se tratará de usar **librerías de código abierto**.
- El sistema contará con un robusto **control de errores**.
- El **lenguaje de programación** utilizado será **C++**.
- Como **sistema operativo** se usará **Ubuntu**.
- Como **entorno de desarrollo** se usará **Qt Creator**.

Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos, restricciones
- 4. Recursos**
5. Diseño del sistema
6. Experimentación
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

4. Recursos

4.1 Recursos hardware

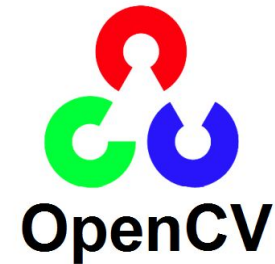
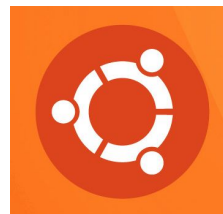
- Ordenador portátil personal:
 - Procesador: APU AMD Quad-Core A8-3530MX (1.9GHz, 4MB L2 Cache)
 - Memoria RAM: 6GB DDR3
 - Sistema de almacenamiento: 750 GB (5400 rpm S-ATA)
- LiDAR: RPLIDAR A1M8
 - Capacidad de muestreo: 8000 muestras/s
 - Rango de medición: 0.15-12m
 - Consumo: 1.7WH



4. Recursos

4.1 Recursos software

- **Librerías y frameworks:** OpenCV, Qt, Qt Creator, LevMarq, RPLIDAR SDK.
- **Sistema operativo:** Ubuntu 18.04.
- **Lenguaje de programación:** C++.



Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos, restricciones
4. Recursos
- 5. Diseño del sistema**
6. Experimentación
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

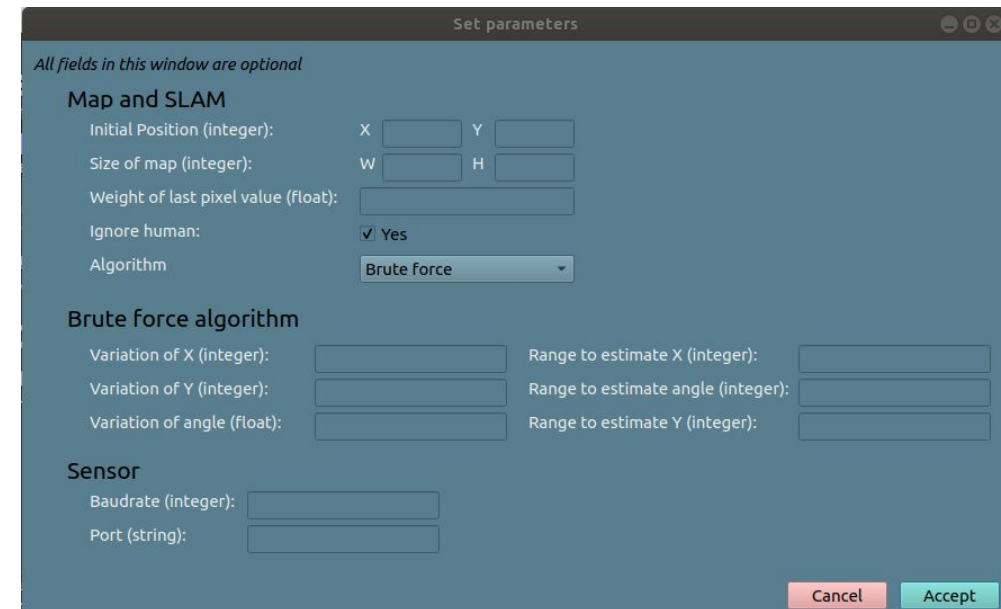
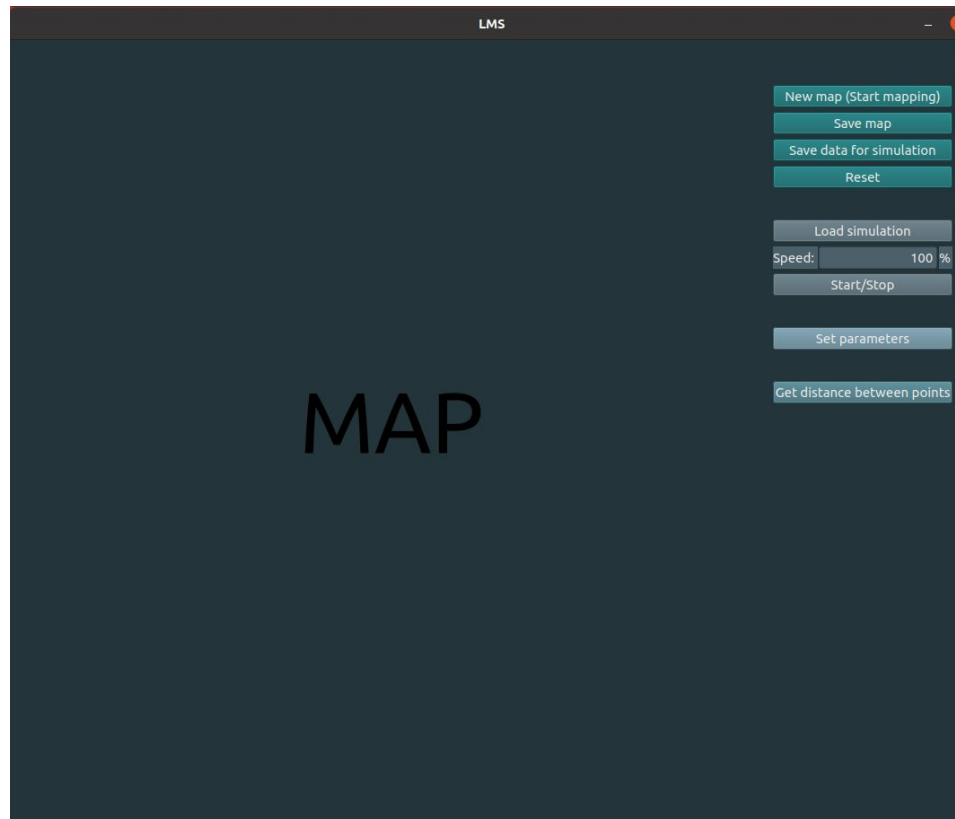
5. Diseño del sistema

Dividido en dos partes:

1. Interfaz gráfica de usuario.
2. Resolución a SLAM.

5. Diseño del sistema

5.1 Diseño de la interfaz gráfica de usuario



5. Diseño del sistema

5.2 Resolución a SLAM

1. Definir el **tipo de mapa**
2. Diseñar una **función de error**
3. Diseñar el **algoritmo de optimización**

5. Diseño del sistema

5.2 Resolución a SLAM

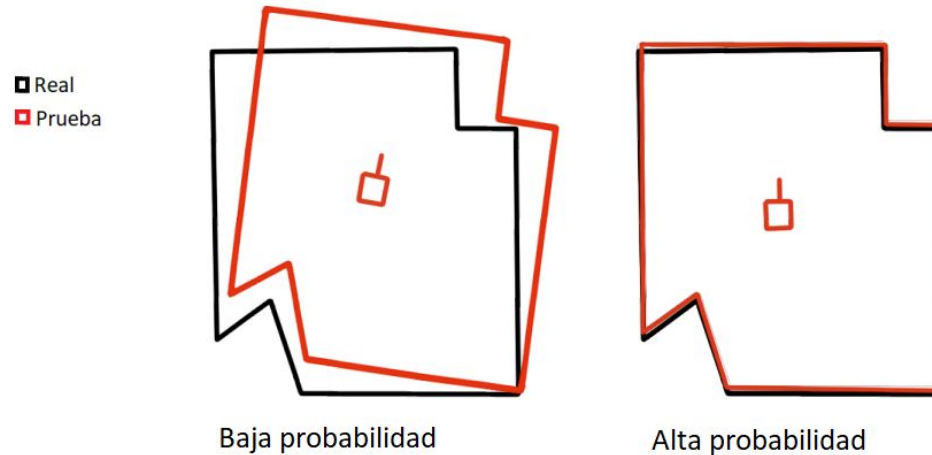
- Definir el **tipo de mapa**
 - **Tipo rejilla:** el espacio se divide en celdas.
 - **Ocupación:** cada celda dispondrá de una probabilidad de ocupación

Esta ocupación se actualizará con las mediciones del LiDAR.

5. Diseño del sistema

5.2 Resolución a SLAM

- Diseñar una **función de error**
 - Dadas las **mediciones del LiDAR** y una **pose** se comprobará la **probabilidad** de ser la **pose real** en el mapa de ocupación.



5. Diseño del sistema

5.2 Resolución a SLAM

- Diseñar el **algoritmo de optimización**
 - **Levenberg Marquardt**
 - Optimización por mínimos cuadrados amortiguados
 - Búsqueda local
 - **Fuerza bruta**
 - Optimización utilizando un gran número de comprobaciones
 - Búsqueda por rangos

Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
- 6. Experimentación**
7. Resultados
8. Demostración
9. Conclusiones y futuras mejoras

6. Experimentación

Parámetros a mejorar:

- **Variación** del ángulo, componente X y componente Y
- **Rango** de búsqueda del ángulo, componente X y componente Y

Métricas de evaluación:

- Tiempo de cómputo
- Veces que el sistema se pierde
- Veces que el sistema se encuentra
- Tolerancia a cambios bruscos
- Precisión

6. Experimentación

Escenario
de
prueba



Índice de contenidos

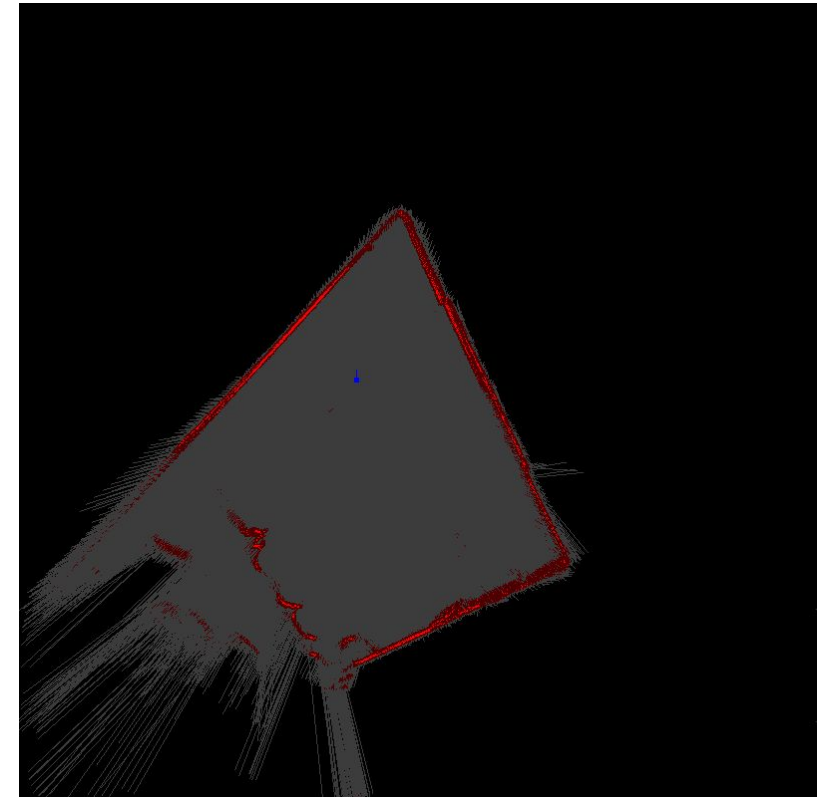
1. Introducción
2. Antecedentes
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
6. Experimentación
- 7. Resultados**
8. Demostración
9. Conclusiones y futuras mejoras

7. Resultados

7.1 Fuerza Bruta

Parameters	Tiempo de cómputo(ms)	Veces perdido	Veces encontrado
1 / 1 / 2 / 40 / 40 / 100	~7100	0	0
1 / 1 / 2 / 30 / 30 / 80	~3400	0	0
1 / 1 / 2 / 20 / 20 / 60	~1100	0	0
2 / 2 / 3 / 40 / 40 / 100	~1250	0	0
2 / 2 / 3 / 30 / 30 / 80	~600	0	0
2 / 2 / 3 / 20 / 20 / 60	~230	0	0
2 / 2 / 4 / 40 / 40 / 100	~900	0	0
2 / 2 / 4 / 30 / 30 / 80	~400	0	0
2 / 2 / 4 / 20 / 20 / 60	~190	0	0
3 / 3 / 4 / 40 / 40 / 100	~460	0	0
3 / 3 / 4 / 30 / 30 / 80	~220	0	0
3 / 3 / 4 / 20 / 20 / 60	~120	0	0
4 / 4 / 5 / 40 / 40 / 100	~260	1	1

2 / 2 / 4 / 20 / 20 / 60

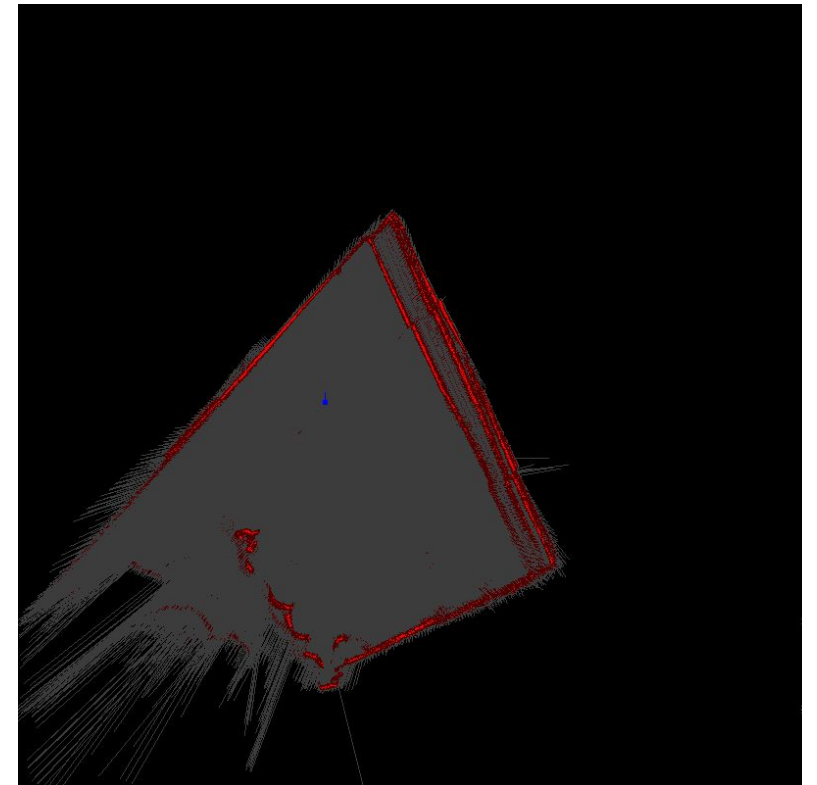


7. Resultados

7.2 Levenberg Marquardt

Parameters	Tiempo de cómputo	Veces perdido	Veces encontrado
1 / 1 / 1	~60	4	1
1 / 1 / 1.5	~60	2	0
1 / 1 / 2	~60	3	1
2 / 2 / 1.5	~60	3	1
2 / 2 / 2	~60	2	1
2 / 2 / 3	~60	1	0
2 / 2 / 4	~60	3	0
3 / 3 / 1.5	~60	2	1
3 / 3 / 2	~60	4	0
3 / 3 / 3	~60	1	0
3 / 3 / 4	~60	1	0
3 / 3 / 5	~60	1	0
4 / 4 / 6	~60	3	0
5 / 5 / 6	~60	2	0
6 / 6 / 6	~60	2	0

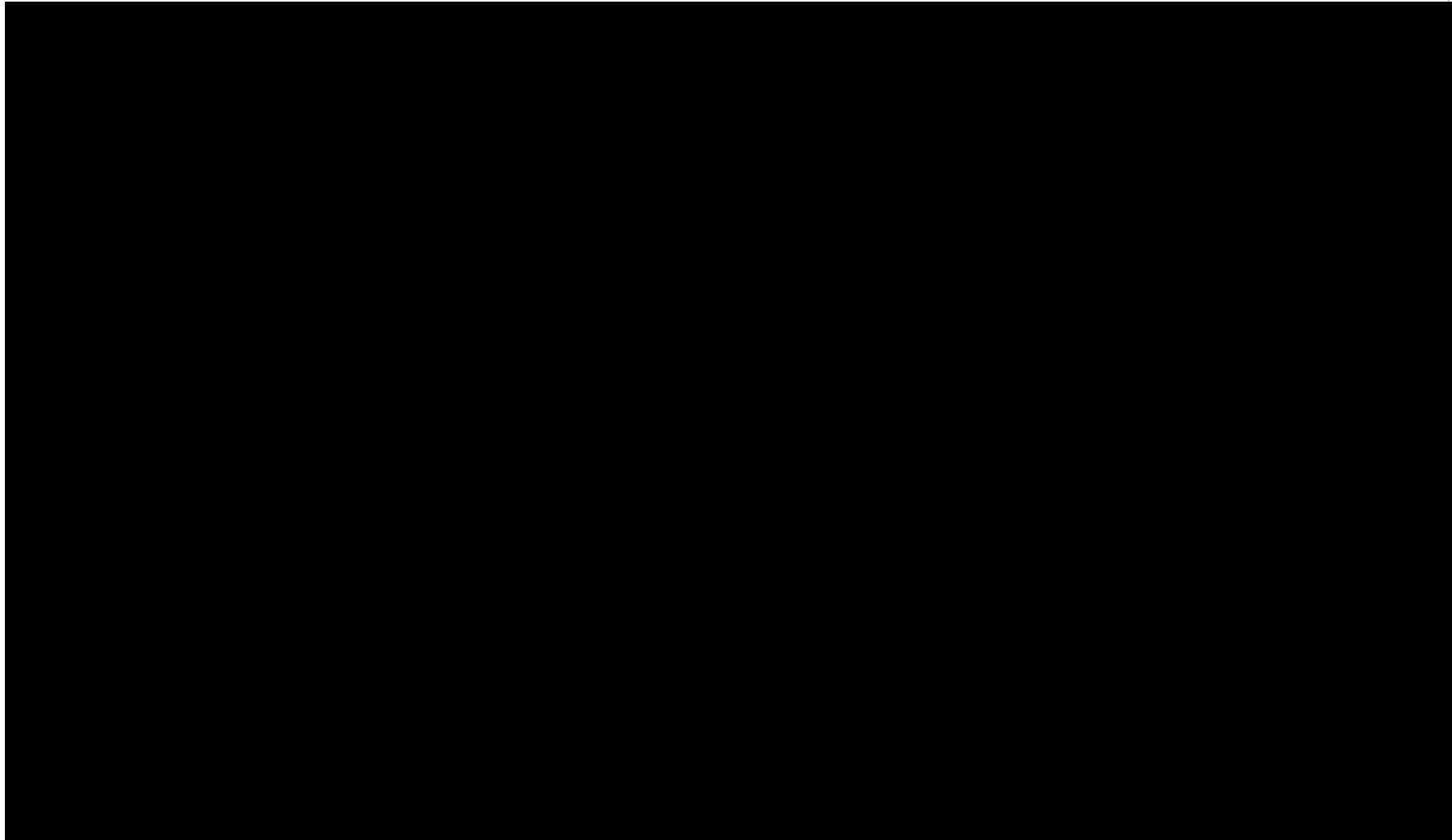
3 / 3 / 5



Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
6. Experimentación
7. Resultados
8. **Demostración**
9. Conclusiones y futuras mejoras

8. Demostración



Índice de contenidos

1. Introducción
2. Antecedentes
3. Objetivos y restricciones
4. Recursos
5. Diseño del sistema
6. Experimentación
7. Resultados
8. Demostración
9. **Conclusiones y futuras mejoras**

9. Conclusiones y futuras mejoras

9.1 Conclusiones

- El sistema desarrollado **cumple con los objetivos**
- **Fuerza bruta** es un método **robusto** pese a sus desventajas
- **Levenberg Marquardt** puede ser aplicado en equipos de **bajas características**
- Fuerza bruta -> **5 frames por segundo**
- Levenberg Marquardt -> **12 frames por segundo**

9. Conclusiones y futuras mejoras

9.2 Futuras mejoras

- Añadir más opciones a la aplicación
 - Crear simulaciones a gusto del usuario
 - Realizar zoom en el mapa
- Añadir algoritmos para la resolución de SLAM
 - Metaheurísticas
 - Redes neuronales
- Añadir un factor de momento para Levenberg Marquardt