

Localización local y global integrando
información visual y de rango para robots
móviles en ambientes no estacionarios

Josué Sánchez Taxis

24 de febrero de 2009

Contenido

Resumen	1
1. Introducción	2
1.1. Motivación	2
1.2. Antecedentes	3
1.3. Planteamiento del problema	6
1.4. Objetivo	7
1.5. Panorama de la tesis	7
1.6. Organización del documento de tesis	8
2. Construcción de mapas y localización	10
2.1. Tipos de mapas	10
2.2. Localización	13
2.2.1. Localización basada en marcas visuales	14
2.3. Resumen	17
3. Extracción de marcas naturales y representación del ambiente	18
3.1. Tipos de marcas	18
3.2. Marcas de rango	19
3.3. Detector SIFT	21
3.4. Marcas visuales	25
3.5. Representación del ambiente e integración de marcas naturales . .	29
3.5.1. Representación final del ambiente	30
3.6. Resumen	33

4. Localización	34
4.1. Introducción	34
4.2. Localización local	35
4.2.1. Correspondencia entre conjunto de marcas	36
4.2.2. Árboles K-d	39
4.2.3. Algoritmo <i>Best Bin First</i> (BBF)	40
4.2.4. Estimación de la posición	46
4.2.5. Estimación de la orientación	49
4.3. Localización global	50
4.4. Resumen	51
5. Experimentos y resultados	53
5.1. Ambientes de experimentación	53
5.2. Sistema de visión	56
5.2.1. Experimento 1	56
5.2.2. Construcción de mapa de marcas visuales	58
5.3. Seguimiento de la posición	62
5.3.1. Experimento 2	63
5.3.2. Experimento 3	65
5.3.3. Experimento 4	68
5.3.4. Resumen	73
5.4. Localización global	73
5.4.1. Experimento 5	73
5.4.2. Experimento 6	74
5.5. Resumen	76
6. Conclusiones y trabajo futuro	77
6.1. Conclusiones	77
6.2. Trabajo futuro	78
Bibliografía	80

Índice de Figuras

1.1. Problema SLAM. Es necesario una estimación simultanea de la posición del robot y las marcas.	4
1.2. Modelo del ambiente basado en rejillas de ocupación [1].	5
1.3. Identificación y caracterización de discontinuidades.	6
2.1. Mapas basados en rejillas de ocupación.	12
2.2. Mapa topológico: a) conexión entre lugares; b) grafo de conectividad.	12
2.3. Mapa semántico.	13
2.4. Conjunto final de marcas SIFT detectados.	15
2.5. Mapas 3D de marcas SIFT visto de diferentes ángulos[7].	16
3.1. Ejemplo de marcas artificiales, patrones planos.	19
3.2. Sensor láser SICK LMS 200.	20
3.3. Ejemplo de los datos obtenidos por el sensor láser SICK LMS 200 [2].	20
3.4. Identificación y caracterización de discontinuidades.	21
3.5. Imágenes suavizadas a niveles t mediante el filtro Gaussiano.	22
3.6. Pirámide Gaussiana y diferencias Gaussinas en la misma escala [4].	23
3.7. Comparativa de cada píxel con sus 26 vecinos en la misma escala y las escalas vecinas [4].	24
3.8. Histograma de orientación del gradiente y refinación de orientación del punto de interés mediante interpolación.	25
3.9. Segmentación de la vecindad en 4x4 regiones, cada una de 4x4 píxeles.	25
3.10. Sistema de visión estéreo STH-HMCS2-VAR-C.	26
3.11. Sistema de coordenadas del sistema de visión.	26

3.12. Marcas visuales detectadas en una imagen.	27
3.13. Imagen del lente izquierdo y marcas visuales detectadas.	28
3.14. Localización de una marca relativa al robot.	30
3.15. Ejemplo de agregar información de marcas visuales a un mapa construido en [1].	32
3.16. Mapa de celdas probabilista (a) y el correspondiente mapa con marcas visuales (b).	33
4.1. Esquema del proceso de localización local.	35
4.2. Correspondencia entre conjuntos de marcas de rango.	38
4.3. Proceso de construcción de un árbol mediante el algoritmo BBF.	41
4.4. Ejemplos de correspondencias entre conjuntos de marcas SIFT.	43
4.5. Correspondencia entre conjuntos de marcas visuales detectadas.	44
4.6. Ejemplo de aplicación de RANSAC.	46
4.7. Calidad de estimación.	48
4.8. Estimación de la orientación del robot.	49
4.9. Esquema del proceso de localización global.	50
5.1. Mapas construidos a base de sensores de rango [1] de los ambientes utilizados para la experimentación.	55
5.2. Robot móvil PeopleBot dotado de un sistema de visión estéreo.	56
5.3. Recorrido del robot en el ambiente 1, para el experimento 1.	57
5.4. Número de marcas visuales detectadas en el recorrido del experi- mento 1 en el ambiente 1.	58
5.5. Tiempo de procesamiento para obtener las marcas visuales en cada imagen.	58
5.6. Mapa 4 con información visual, construidos en diferentes condi- ciones de iluminación.	59
5.7. Mapa 2 con información visual. a) Trayectoria elegida, b) Mapa resultante.	60
5.8. Proceso de integración de marcas visuales al mapa 2.	61
5.9. Trayectoria recorrida por el robot en el plano x,y para el experi- mento 2	63
5.10. Trayectoria recorrida respecto al mapa del ambiente para el experi- mento 2.	64

5.11. Número de marcas consistentes tomadas como referencia para estimar la posición del robot en cada instante.	64
5.12. Tiempo de procesamiento para estimar la nueva posición del robot en cada instante.	65
5.13. Trayectorias estimadas respecto el mapa del experimento 3. En parte encerra en el semicírculo hubo una colisión según la trayectoria estimada utilizando sólo marcas de rango.	66
5.14. Trayectorias estimadas en el plano x, y del experimento 3.	66
5.15. Orientación del robot durante el recorrido en el experimento 3. Ángulo expresado en radianes.	67
5.16. Marcas naturales consistentes para el experimento 3	68
5.17. Trayectorias estimadas en el plano x, y en el experimento 4	70
5.18. trayectorias estimadas sobre el mapa del ambiente para el experimento 4.	71
5.19. Valores de ángulos durante el recorrido en el experimento 4 según la estimación utilizando el tipo de marca.	72
5.20. Número de marcas naturales consistentes en los diferentes casos de localización para el experimento 4.	72
5.21. Resultado de aplicar el algoritmo de localización global.	74
5.22. Localización global después haber cambios en el ambiente original.	75

Índice de Tablas

4.1. Construcción de un árbol k-2, de acuerdo al algoritmo 4.1.	40
4.2. Notación para el algoritmo 4.4.	47
5.1. Integración de información visual.	62

Índice de Algoritmos

3.1. Agregar información de marcas visuales al mapa construido a base de láser y sonares en [1].	31
4.1. Resumen del algoritmo de construcción de árboles K-d	39
4.2. Algoritmo de búsqueda aproximada de vecino más cercano (BBF).	42
4.3. RANSAC.	45
4.4. Algoritmo para estimar la nueva posición del robot.	47
4.5. Localización global a base de marcas visuales mediante triangulación.	51

Resumen

En este trabajo se abordan dos problemas fundamentales en robótica móvil: la localización local y global para ambientes interiores no estacionarios. En la localización local se conoce la posición inicial y la tarea es ir estimando la nueva posición mientras el robot navega, mientras que en la global se desconoce la posición inicial y el objetivo es localizarse. En tareas en las que el robot tenga la necesidad de navegar en el ambiente, es necesario que el robot cuente con una representación del ambiente. Es decir, un mapa preferiblemente construido con sus propios medios y utilizarlo para determinar su posición en todo momento mientras realiza su trabajo. El tener información de rango y visual proporciona al robot mayor evidencia para que éste pueda localizarse ambas informaciones son complementarias. Los principales objetivos de esta tesis son: *(i)* agregar información visual a los mapas construidos con sensores de rango necesario para la localización global y *(ii)* combinar información visual y de rango para aumentar la robustez en el proceso de localización local o seguimiento de la posición. También se contribuye en la localización global mediante un algoritmo que utiliza sólo las marcas visuales y que es tolerante a cambios en el ambiente. Se realizaron experimentos en cuatro ambientes de interiores diferentes bajo condiciones variables de iluminación. Se obtuvieron resultados prometedores. Particularmente, en la localización local, se demostró que el método es robusto en ambientes dinámicos con un margen de error en promedio de 7cm.

Capítulo 1

Introducción

En este trabajo de tesis se abordan dos problemas fundamentales en el área de robótica móvil, que son la localización local y global. Es decir, el robot primeramente obtiene una representación del ambiente de trabajo utilizando sus sensores y usa el mapa para localizarse dentro de él en cada instante. En particular se aborda el problema de integración de información de diferentes modalidades (láser y visión) para enriquecer el mapa construido en [1] y mejorar el proceso de localización local y global para ambientes interiores no estacionarios. Para realizar el proceso de localización se utilizan marcas naturales las cuales dividiremos en dos tipos, de rango y visuales. A partir de aquí, nos referiremos como marcas de rango a aquellas detectadas mediante el láser [2]; y a las marcas visuales a aquellas detectadas mediante el sistema de visión. En el capítulo 3 se detallan estos tipos de marcas. En este capítulo se describen los antecedentes, la problemática que se aborda y los objetivos del trabajo de tesis.

1.1. Motivación

En robótica móvil uno de los principales requerimientos es que el robot tenga la capacidad de operar de forma independiente en ambientes desconocidos. Para ello una de las necesidades es que el robot construya un mapa con sus propios medios con el objetivo de identificar el espacio libre y ocupado. Un mapa es un modelo del ambiente, existen básicamente tres tipos de mapas: métricos, topológicos y semánticos, en el capítulo 2 se describe cada uno.

En tareas donde es necesario desplazarse, el robot debe conocer su ubicación en todo momento dentro del mapa y así poder tomar decisiones y ejecutar acciones para lograr su objetivo. Existen dos tipos de localización, local y global. En la primera, también conocida como seguimiento de la posición, el robot conoce su posición inicial y va estimando su posición dentro del mapa mientras se mueve. En la localización global, el robot desconoce su posición inicial y su objetivo es determinar su ubicación dentro del mapa sin ninguna información *a priori* a cerca de su posición. Aunque existen muchos desarrollos previos en la construcción de mapas y localización [9, 6, 1, 7, 8], la mayor parte asume ambientes no estacionarios en donde no hay cambios significantes. Si hay cambios en el ambiente el proceso de localización, en particular global, puede fallar y esto ocasiona problemas importantes en la realización de otras tareas. Una alternativa para reducir este problema es la combinación de la información proporcionada por diferentes tipos de sensor. En el presente trabajo se integra los dos tipos de información (rango y visual) y se contempla el problema de localización local y global para ambientes interiores no estacionarios, es decir, se permiten cambios en el ambiente.

1.2. Antecedentes

El proceso por el cual un robot móvil puede construir un mapa de un ambiente y al mismo tiempo usar este mapa para localizarse, es conocido como SLAM (por sus siglas en inglés, *Simultaneous Localization And Mapping*). A nivel teórico, SLAM puede ser considerado ahora como un problema resuelto. Existen dos principales soluciones, una con el Filtro de Kalman Extendido conocido como EKF-SLAM y la otra mediante el filtro de partículas conocido como FastSLAM [9]. Sin embargo, siguen existiendo problemas importantes como el como combinar diferentes tipos de sensores y el localizarse en ambientes dinámicos. En la figura 1.1 se muestra un esquema general del problema SLAM, en donde se requiere estimar la posición del robot en cada instante utilizando odometría y al mismo tiempo las marcas, estas pueden ser artificiales ó naturales. Las artificiales son aquellas que uno diseña y pone intencionalmente en el ambiente, por ejemplo, algún emisor, fotografías, papeles de color intenso o algo distintivo que el robot pueda identificar con sus sensores; para nuestro caso usaremos marcas naturales

que se obtienen mediante el láser y un sistema de visión.

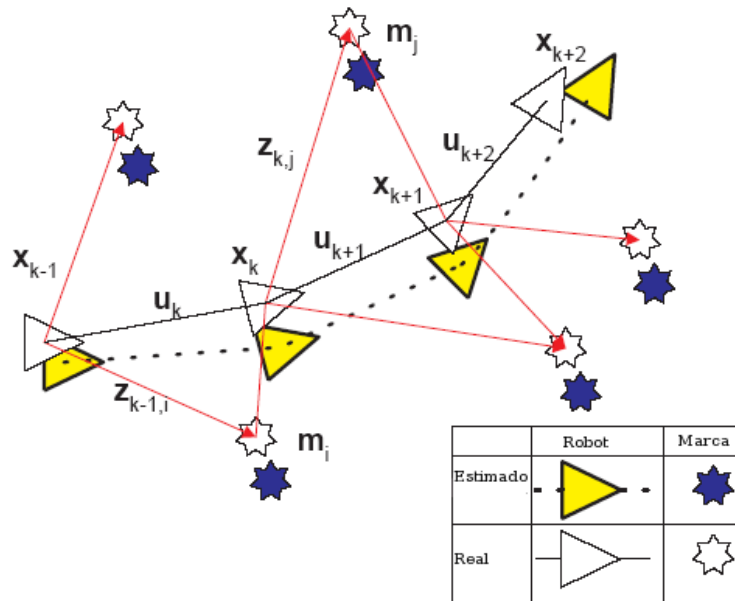


Figura 1.1: Problema SLAM. Es necesario una estimación simultánea de la posición del robot y las marcas [9]. Los símbolos resaltados representan el proceso de la estimación de la posición del robot. El robot (triángulo) ejecuta las acciones, u , y toma como referencia las marcas (estrellas) detectadas, z , para estimar su estado (posición) x .

De acuerdo a la figura 1.1 tenemos:

- x_k : Vector de estado, describe la posición y orientación del robot en el tiempo.
- u_k : Vector de control, aplicado al tiempo $k - 1$ para conducir el robot al estado x_k en el tiempo k .
- m_i : Un vector describiendo la localización de la i marca.
- $z_{i,k}$: Una observación tomada por el robot, de la posición de la marca i en el tiempo k .

En la forma probabilista del problema SLAM se requiere que la distribución de probabilidad sea calculada en todo momento. Es decir, calcular la probabilidad de que el estado del robot sea x_k y se tengan las marcas m dadas las observaciones desde el instante 0 hasta el tiempo k y las acciones que se han ejecutado desde el

tiempo 0 hasta k , y el estado inicial del robot, x_0 . De manera simbólica tenemos que:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (1.1)$$

Para calcular esta distribución se requiere de un modelo de transición de estados y un modelo de observación, que son definidos describiendo los efectos del control y observación, respectivamente:

- Modelo transición de estado o movimiento, $P(x_k | x_{k-1}, u_k)$: probabilidad de pasar al estado x_k dado un estado anterior x_{k-1} y la ejecución de la acción u_k .
- Modelo de observación, $P(z_k | x_k, m)$: probabilidad de observar z_k dado que se encuentra en el estado x_k y se tiene el conjunto de marcas m .

En [1] se hace una análisis más detallado del enfoque probabilista y se hace un estudio de la solución mediante el filtro de partículas. Se construye un modelo del ambiente representado en rejillas de ocupación fusionando láser y sonar, en donde cada celda guarda información de la probabilidad de ocupación. En la figura 1.2 se muestra un ejemplo de este tipo de mapas, cada píxel corresponde a un espacio de 5x5cm del mundo real. La idea en este trabajo es tomar este mapa como mapa inicial y enriquecerlo con información visual invariante.

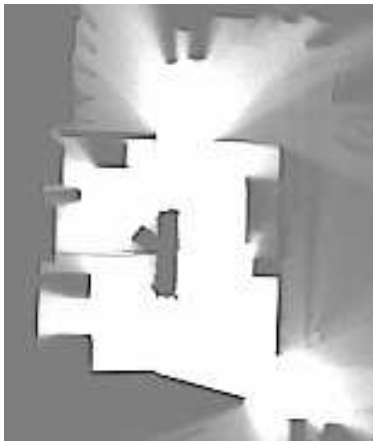


Figura 1.2: Modelo del ambiente basado en rejillas de ocupación [1].

En [2] se aborda el problema de navegación, localización global y localización local o seguimiento de la posición. Se utilizan marcas de rango como referencia

para la localización. Estas marcas son: esquinas, paredes y discontinuidades. Un ejemplo de ello se muestra en la figura 1.3, donde se muestran marcas del tipo discontinuidad que es la diferencia entre medidas consecutivas tomadas con el láser. Se guarda información de la distancia, D , del robot a la discontinuidad, a qué ángulo, Θ , se encuentra respecto al frente del robot, la diferencia, P , entre estas medidas consecutivas y por último, de que lado se encuentra, *izquierdo* o *derecho*, respecto al frente del robot. Para esta tesis tomamos como base estos trabajos, entonces los datos de entrada para nuestro trabajo es el mapa construido en [1] y el conjunto de marcas de rango calculadas en [2].

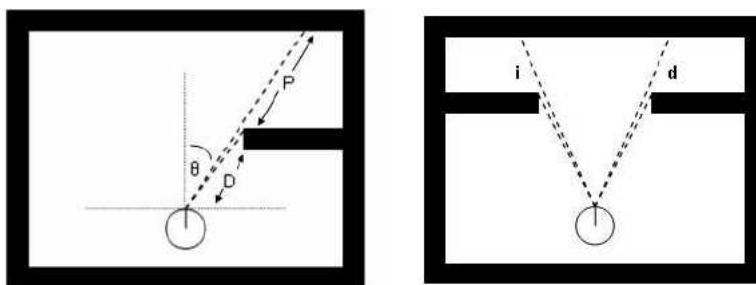


Figura 1.3: Identificación y caracterización de discontinuidades. Imagen izquierda, se muestra la diferencia entre dos lecturas consecutivas del láser que tomamos como marca tipo discontinuidad. En la imagen derecha se muestran los dos tipos de discontinuidad que podemos encontrar, donde i y d se refiere a una discontinuidad izquierda y derecha respectivamente [2].

1.3. Planteamiento del problema

La mayoría de los sistemas de SLAM y de localización, como los enfoques mostrados en los trabajos [5, 2, 1, 8], tienen problemas si:

- Hay pocas marcas naturales en el ambiente.
- Hay cambios respecto al mapa inicial.
- Hay zonas semejantes en cuanto a su forma geométrica.

En este trabajo se parte de una representación del ambiente en rejillas de ocupación y un conjunto de marcas de rango. Se abordan básicamente dos problemas, integración de información y cambios en el ambiente.

1. Un aspecto es cómo integrar la información de los sensores de rango con información visual invariante para obtener una representación del ambiente o mapa que contenga ambos tipos de marcas naturales, de rango y visuales.
2. Otra problemática que se abordará ocurre en la localización global. Una vez que el robot tiene una representación del ambiente e información del ambiente (marcas naturales), sí el ambiente cambia (se mueven objetos en el ambiente y en dichos objetos el robot había identificado marcas), lo más probable es que no se localice adecuadamente. Además si existen zonas que son muy semejantes, entonces puede existir más de una hipótesis en la localización. Al agregar información visual invariante se aumenta la probabilidad de descartar hipótesis incorrectas y mejorar la robustez en la localización global, aún cuando haya cambios en el ambiente.

1.4. Objetivo

El objetivo de este trabajo es desarrollar un método para solucionar el problema de localización global para ambientes interiores no estacionarios, integrando información visual mediante características invariantes con información de sensores de rango. Partiendo de este objetivo general, se definen los siguientes objetivos particulares:

1. Obtener marcas visuales invariantes en cuanto a rotación, escala y parcialmente a la luz.
2. Integrar las marcas visuales con las marcas de rango.
3. Desarrollar algoritmos de localización local y global que combinen información visual y de rango.
4. Probar los métodos anteriores en diferentes ambientes interiores.

1.5. Panorama de la tesis

Este trabajo de tesis fundamentalmente se enfoca al problema de localización para ambientes interiores no estacionarios, particularmente en el problema de integración de información y cambios en el ambiente. Se toman como base los

trabajos [1, 2], mediante una representación inicial del ambiente mediante rejillas de ocupación. Nuestro trabajo primeramente fue dotar al robot de un sistema de visión estéreo y mediante el método SIFT[4] extraer características invariantes que usaremos como marcas visuales. Para enriquecer el mapa con información visual, el robot realiza un recorrido por el ambiente y va asociando las marcas visuales detectadas a su correspondiente celda. Mientras se enriquece el mapa no debe haber cambios en el ambiente. Para la localización local se permiten cambios en el ambiente mientras el robot realiza su tarea. La técnica que se usa para localizarse es mediante triangulación tomando como referencia las marcas naturales, encontrando una correspondencia entre el conjunto de marcas que se detectaron en el estado previo y actual del robot. Se eliminan las correspondencias erróneas entre conjuntos mediante el método RANSAC [12].

El problema de localización global es básicamente un problema de correspondencia entre las marcas visuales del mapa enriquecido y las marcas visuales detectadas. De igual forma que en la localización local, se eliminan las correspondencias erróneas mediante RANSAC [12] y mediante triangulación se estima la posición del robot dentro del mapa.

El ambiente puede cambiar y el robot debe ser capaz de localizarse. Con esto se tiene un método robusto a cambios en el ambiente, de tal forma que un robot pueda construir mapas de ambientes interiores integrando información visual invariante con información de los sensores de rango y usar estos mapas para localizarse. El robot puede localizarse aún existiendo cambios en el ambiente, aunque no se actualiza el mapa. Los métodos desarrollados para construcción de mapas y localización fueron robustos en ambientes interiores con resultados satisfactorios como vemos en detalle en el capítulo 5

1.6. Organización del documento de tesis

El presente documento esta organizado de la siguiente forma. En el capítulo dos se hace una revisión de diferentes trabajos consultados relacionados con el problema de construcción de mapas y localización local y global utilizando marcas visuales, y se describen los diferentes tipos de mapas. En el capítulo tres se describen los diferentes tipos de marcas, así como la forma en que se extraen las marcas de rango y visuales y la integración de estos dos tipos de informa-

ción para obtener un mapa final enriquecido. En el capítulo cuatro se describe el método utilizado para la localización local y global utilizando las marcas naturales, mediante la técnica de triangulación y el método RANSAC. Por último, en el capítulo cinco y seis se describen y discuten los resultados obtenidos de los experimentos hechos en diferentes ambientes y bajo condiciones diferentes, así como conclusiones y trabajo a futuro, respectivamente.

Capítulo 2

Construcción de mapas y localización

La construcción de mapas o elaboración de una representación del ambiente y la localización, local y global, son problemas fundamentales dentro de la robótica móvil. Tener una representación del ambiente y saber en todo momento la ubicación del robot dentro del mismo son indispensables en tareas donde el robot tenga que desplazarse de un sitio a otro. En este capítulo se presenta una revisión de los trabajos relacionados con la construcción de mapas y localización local y global utilizando marcas naturales para robots móviles, particularmente en ambiente interiores.

2.1. Tipos de mapas

Un mapa es un modelo del ambiente, generalmente para representar el espacio libre u ocupado, y en ocasiones también para reconocer regiones u objetos en el ambiente. Que el robot pueda construir su propia representación, es importante, porque estará adecuado a sus capacidades sensoriales. Existen tres tipos de mapas:

1. Métricos o geométricos
2. Topológicos
3. Semánticos

Los mapas métricos o geométricos representan el espacio libre u obstáculos mediante medidas espaciales. Se tienen dos formas básicas, descomposición y geométrico. En la descomposición espacial se representa el espacio libre u ocupado mediante una discretización en un conjunto de celdas básicas, por medio de una rejilla de ocupación espacial. Ésta a su vez puede ser, binaria o probabilista y uniforme o jerárquica. En la figura 2.1(a) se muestra un ejemplo de un mapa binario uniforme, en donde cada celda (píxel) representa un área de 5x5cm del espacio en el ambiente real y solo guarda información de libre/ocupado. En la figura 2.1 (b) se tiene un mapa modelado en rejillas de ocupación probabilista, cada celda guarda información de probabilidad de ocupación, celdas (píxeles) más oscuras indican mayor probabilidad de ocupación. La desventaja de usar rejillas de ocupación es el espacio requerido en memoria para almacenarlo, sobre todo cuando se modelan ambientes grandes y más aún si se desea una representación en 3D. Algunas alternativas para reducir el espacio requerido son usar Quadrees (árboles cuaternarios) y BSP (árboles de partición del espacio) [19, 18]. Los mapas geométricos representan el ambiente mediante formas geométricas básicas como: puntos, líneas, círculos, rectángulos, poliedros, *splines*, etc. Dentro de esta categoría entran también los mapas basados en marcas. Normalmente se utilizan conjuntos de marcas (puntos ó imágenes) que el robot toma como referencia para localizarse. Estas marcas pueden ser características visuales distintivas como esquinas (Harris) o SIFT [13, 4], ó puntos característicos obtenidas mediante sensores de rango (láser y sonar) como: discontinuidades, esquinas y paredes [2].

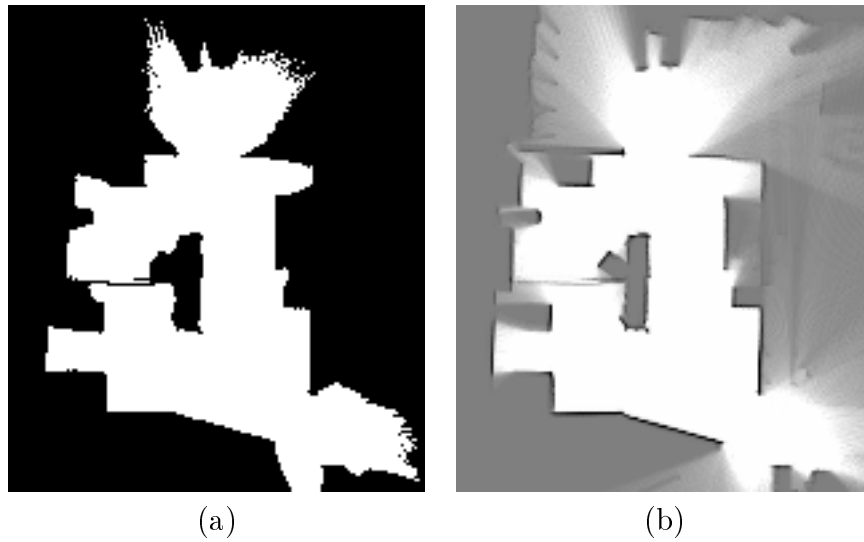


Figura 2.1: Mapas basados en rejillas de ocupación: a) binario, píxel negro indica ocupado, blanco, libre; b) probabilístico, píxeles más oscuros radican mayor probabilidad de ocupación. Cada píxel representa un área de 5×5 cm del espacio en el ambiente real.

En los mapas topológicos, figura 2.2, el ambiente se divide en una serie de lugares y conexiones entre dichos lugares. Esto se puede representar mediante un grafo, en donde los lugares se representan como nodos y las conexiones como arcos. Se le puede agregar información métrica como distancia y orientación de los arcos.

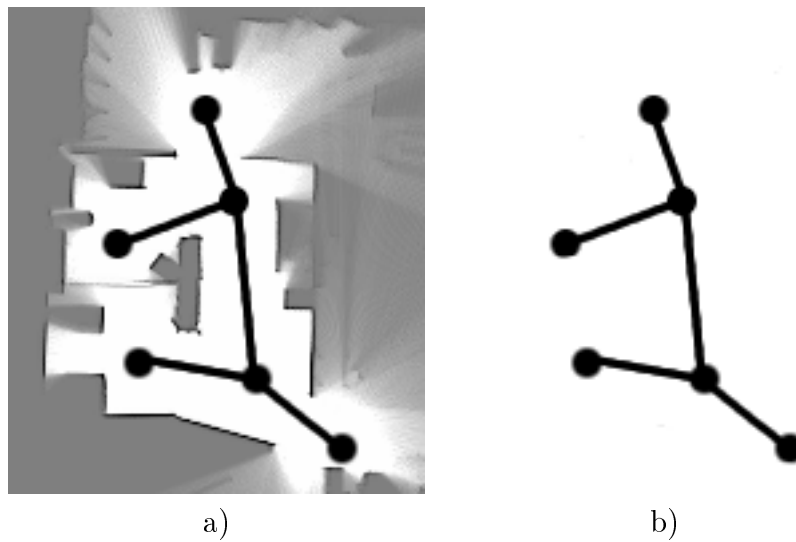


Figura 2.2: Mapa topológico: a) conexión entre lugares; b) grafo de conectividad.

En los mapas semánticos se tiene asociado a un mapa métrico o topológico cierta semántica para los diferentes espacios u objetos en el mapa. La ventaja de utilizar esta representación es que permite una interacción humano-robot más natural, por ejemplo ordenarle al robot: “ve a la cocina” (ver figura 2.3).



Figura 2.3: Mapa semántico.

En este trabajo se usan mapas basados en rejillas de ocupación al que se le integran marcas visuales basadas en características invariantes, SIFT [4]. El objetivo es proporcionar al robot mayor información que pueda tomar como referencia para su localización.

2.2. Localización

Existen dos tipos de localización, local y global. En la primera, la posición inicial es conocida y el objetivo es el seguimiento de la posición mientras el robot se desplaza por el ambiente. La mayor parte de los trabajos reportados en la literatura [2, 5, 6, 7, 8] abordan este problema apoyándose en sensores de rango (láser y sonar) y en la odometría del robot, y recientemente con la ayuda de sistemas de visión. Para ello utilizan como referencia, marcas naturales, como: esquinas, paredes y discontinuidades detectados por el láser [2]; características invariantes en cuanto a la rotación, escala y parcialmente a la iluminación, detectadas por el sistema de visión, utilizando diferentes tipos de marcas destacando el método SIFT [4]. Este último es de nuestro interés, porque sus propiedades proporcionan

estabilidad y aumenta, la robustez en la localización. En cuanto a la localización global el robot desconoce su posición inicial y su tarea es averiguarla, para ello se apoya de información previamente recabada. El tipo de información, como en la localización local, puede ser de diversos tipos integrando marcas artificiales y naturales. Debido a que en este trabajo de tesis se utilizan marcas naturales como referencia, particularmente marcas visuales [4], se analizan trabajos de localización basados en visión.

2.2.1. Localización basada en marcas visuales

En [5] se aborda el problema de seguimiento de la posición o localización local de un robot usando marcas visuales. Se estima el desplazamiento d_k de un robot móvil utilizando un enfoque probabilista, tomando como referencias características visuales invariantes (SIFT). Para la obtención de marcas visuales se utiliza un sistema de visión estéreo. El proceso general es el siguiente, se obtienen dos imágenes en cada instante y se extraen las características SIFT de ambas imágenes. Se halla la relación entre ambos conjuntos de SIFT's y se proyectan al espacio tridimensional, utilizando las matrices de calibración de las cámaras. Después, con la función de densidad de desplazamiento $p(d_k | o_{k-1}, o_k)$, se estima el desplazamiento, que básicamente es la probabilidad de que el robot se haya desplazado una distancia d en el instante k dado el conjunto de SIFT's o que se obtuvieron en el instante $k - 1$ y el conjunto de SIFT's que se detectaron en el instante k .

En [6] se aborda el problema SLAM utilizando una sola cámara. Se enfoca principalmente en la localización de la cámara en el espacio tridimensional, más que en obtener una representación detallada del ambiente. El objetivo principal es determinar la trayectoria 3D de una cámara monocular en tiempo real. Se obtiene un mapa 3D probabilista que representa estimaciones de los estados de la cámara y todas las marcas naturales visuales que usa como referencia, además, contiene información de incertidumbre en la estimación de los estados. Las marcas visuales naturales son subimágenes de 11x11 píxeles, donde se detectaron esquinas mediante el operador de Shi & Tomasi [20]. Estas subimágenes las llamaremos "parches", como no se conoce la orientación del parche se dice que su normal es paralela al vector que va de la marca a la cámara. Matemáticamente el mapa

está representado por un vector de estado \hat{x} y una matriz de covarianza P que modela la incertidumbre. El vector de estado \hat{x} guarda información de posición, orientación y velocidad de la cámara e información de las marcas (esquinas).

Al iniciar el sistema se necesita información previa, típicamente son cuatro marcas conocidas frente a la cámara. El seguimiento se hace mediante el Filtro de Kalman Extendido, el modelo de velocidad lineal y angular se considera constante. Cuando nuevas marcas visuales son detectadas el mapa se extiende o se enriquece con nuevos estados. En este trabajo se asume un ambiente estático. Este método se aplicó en trabajos de realidad aumentada, y en localización de un robot humanoide.

En [7] se aborda el problema SLAM utilizando como marcas características visuales invariantes (SIFT) en ambientes estáticos. Para la obtención de dichas marcas se usa un sistema de visión estéreo Triclops. En cada instante se obtienen 3 imágenes, se halla la relación entre las marcas SIFT's que se detectaron en cada una de las imágenes, mediante la restricción epipolar y de disparidad. El conjunto de SIFT's resultante es proyectado al espacio tridimensional mediante la matriz de calibración de la cámara.



Figura 2.4: Conjunto final de marcas SIFT detectados en [7]. Después de hallar la relación entre conjuntos de SIFTs de cada imagen, las líneas horizontal y vertical representan la disparidad horizontal y vertical [7].

En la figura 2.4 se muestra la relación obtenida de las tres imágenes con sus correspondientes conjuntos de SIFT's. Se obtiene $[r, c, s, o, d, X, Y, Z]$ para cada característica SIFT donde (r, c) son coordenadas en el espacio de la imagen, (s, o, d) representan la escala, orientación y disparidad respectivamente y por último (X, Y, Z) son las coordenadas 3D del espacio relativas a la cámara de referencia (cámara izquierda). Para construir el mapa de marcas 3D de características SIFT es necesario conocer cuánto se ha movido el robot entre cuadros de imágenes. Los datos del odómetro del robot dan una estimación; entonces para encontrar la relación entre marcas visuales - SIFT del instante $t - 1$ con el instante t , la odometría ayuda a predecir la región de búsqueda y mediante mínimos cuadrados se obtiene la nueva posición de la cámara. Los movimientos del robot están restringidos a solo avanzar y girar. Básicamente se hace un seguimiento de marcas para calcular la posición del robot y al mismo tiempo se van guardando las marcas SIFT que representan el modelo del ambiente en 3D. En la figura 2.5 se muestra el resultado de un mapa 3D de marcas SIFT.

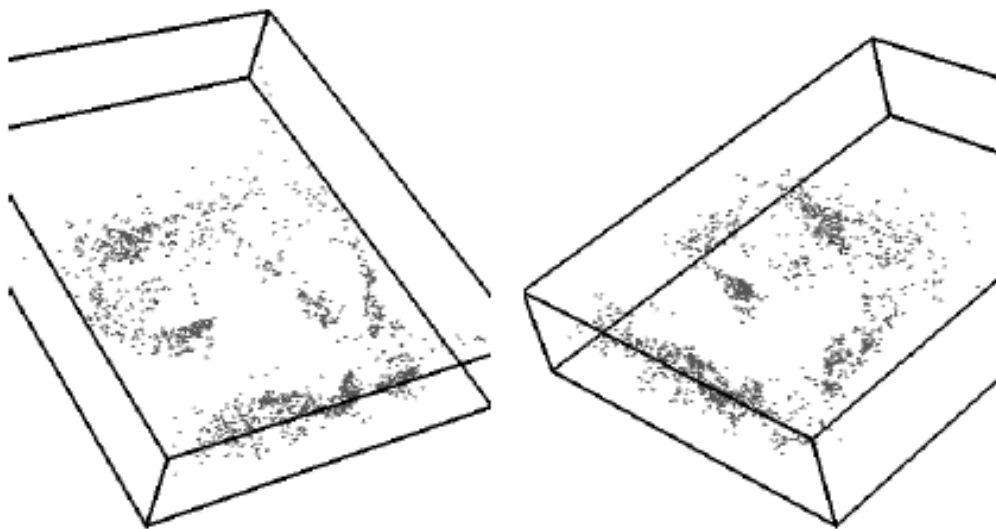


Figura 2.5: Mapas 3D de marcas SIFT visto de diferentes ángulos[7].

En [8], que es una continuación de [7], se considera el problema de localización global como un problema de reconocimiento. Se utilizan como referencias marcas SIFT. Se considera la relación del conjunto de marcas SIFT detectadas en la ubicación actual con un mapa de marcas SIFT previamente construido [7]. Inicialmente se tiene un mapa de marcas visuales en 3D obtenida con un sistema de

visión estéreo Triclops. Se usan dos enfoques para hallar la relación entre dichos conjuntos y así obtener la posición del robot. Estos enfoques son la transformada de Hough y RANSAC [12]. Al final se hace una comparativa entre dichos enfoques. Los mejores resultados se obtuvieron utilizando el enfoque RANSAC.

2.3. Resumen

En esta sección se presentó una descripción de los distintos tipos de mapas o formas de representar un ambiente. En nuestro caso modelamos el ambiente mediante rejillas de ocupación probabilista [1] y construimos un mapa de marcas visuales. La idea es que el robot cuente con mayor información para su localización. Estas marcas visuales ayudan a discriminar zonas estructuralmente semejantes.

Se introdujo, además, una serie de trabajos que abordan el problema de localización local y global, utilizando sólo un tipo de sensor y tomando como referencia marcas visuales. La desventaja de estos trabajos es que se asume que los ambientes son estáticos. Por lo tanto si se modifica el ambiente, en la mayoría de los casos el robot falla a la hora de localizarse globalmente. En este trabajo se integra dos tipos de información, visual y de rango, teniendo así mayor evidencia. Además se contemplan ambientes no estacionarios, es decir, a pesar de que haya cambios en el ambiente el robot es capaz de localizarse globalmente, debido a que se verifica la relación espacial de las marcas, discriminando así las marcas que se han movido.

Capítulo 3

Extracción de marcas naturales y representación del ambiente

En este capítulo se introducen los diferentes tipos de marcas que el robot puede detectar mediante sus sensores. Se describe brevemente la forma en que se obtienen las marcas de rango mediante el láser. Además, se da una descripción detallada de como se extrae el conjunto de marcas visuales que en conjunto el robot utilizará para localizarse. Por último se describe como se integra la información visual y de rango para obtener una representación final del ambiente, modelado mediante rejillas de ocupación probabilista.

3.1. Tipos de marcas

Entenderemos como marcas a elementos únicos con características distintivas dentro del ambiente. Podemos clasificarlas en dos categorías, artificiales y naturales. Las artificiales generalmente son patrones que uno puede diseñar y poner intencionalmente en el ambiente para facilitar su identificación y localización, en la figura 3.1 se tiene un ejemplo de este tipo de patrones. Otros ejemplos de marcas artificiales son: algún tipo de emisor, fotografías, papeles de color intenso o algo distintivo que el robot pueda identificar con sus sensores.

Las marcas naturales son elementos inherentes al ambiente. Existen diversos tipos de marcas naturales, dentro de las más comunes tenemos las discontinuidades, paredes y esquinas [2]. A este tipo de marcas detectadas mediante

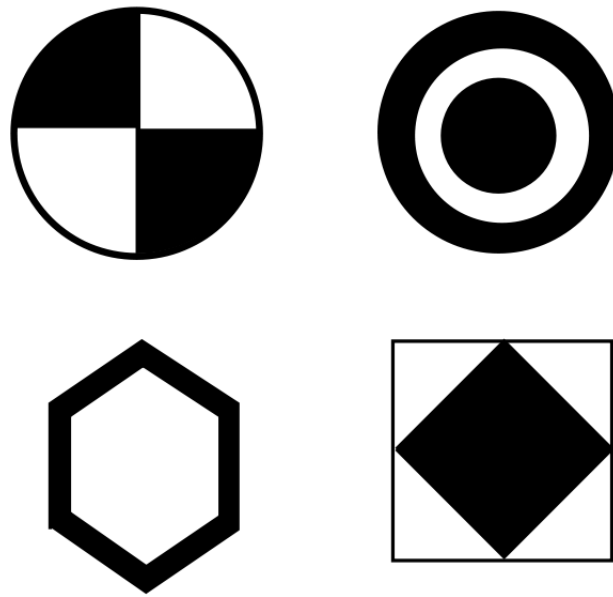


Figura 3.1: Ejemplo de marcas artificiales, patrones planos.

el láser las llamaremos, para este trabajo, marcas de rango. Mientras que las obtenidas con visión como: esquinas, bordes, regiones de color o intensidad y características invariantes en cuanto a escala, orientación y parcialmente a la iluminación (SIFT) [4], nos vamos a referir a ellas como marcas visuales. En este trabajo utilizaremos marcas visuales y de rango, particularmente discontinuidades.

3.2. Marcas de rango

El robot usa un sensor láser de tipo SICK LMS 200 (ver figura 3.2) para detectar marcas de rango. Este sensor devuelve la distancia a los objetos más próximos sobre el plano paralelo al piso, en un rango de 180 grados. con una precisión angular de 0.25 grados por lo que podemos obtener 720 lecturas en un barrido, en donde cada lectura indica la distancia y ángulo del objeto más cercano. En la figura 3.3 se muestra un ejemplo de los datos obtenido por el láser.



Figura 3.2: Sensor láser SICK LMS 200.

En [2] se describe detalladamente la forma en que se detectan las marcas de rango mediante el láser. En este trabajo sólo utilizamos discontinuidades, a continuación se describe como se obtienen a partir del sensor láser.

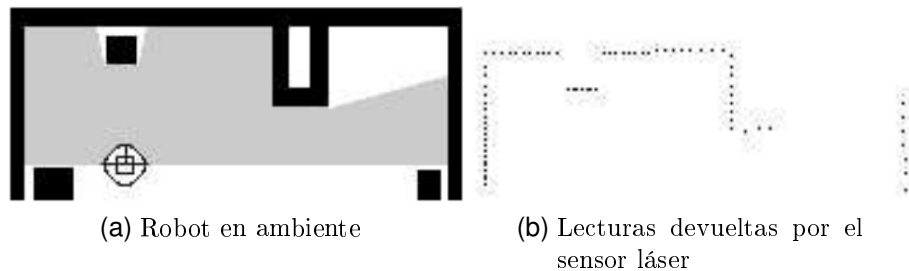


Figura 3.3: Ejemplo de los datos obtenidos por el sensor láser SICK LMS 200 [2].

Se considera una discontinuidad a una variación mayor a cierto umbral en relación a la distancia medida en lecturas consecutivas del láser (ver figura 3.4). Cada discontinuidad tiene cuatro atributos que la caracterizan. D representa la distancia del robot a la discontinuidad en metros. P indica la profundidad, dicho en otras palabras es la diferencia entre las dos lecturas del láser que dieron origen a la discontinuidad. Θ es el ángulo en el que se encuentra la discontinuidad con respecto al frente del robot expresado en radianes y por último i y d indican si se trata de una discontinuidad izquierda o derecha, respectivamente, respecto al frente del robot .

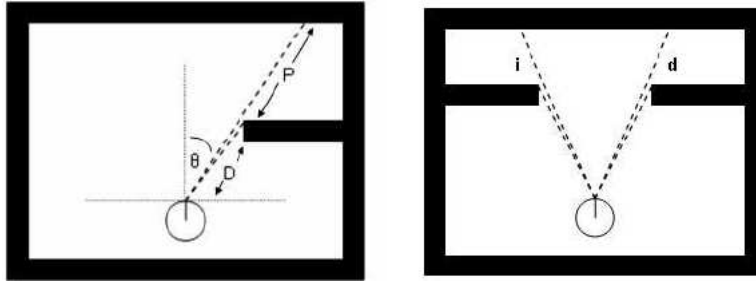


Figura 3.4: Identificación y caracterización de discontinuidades. En la imagen izquierda se muestra la diferencia de dos lecturas consecutivas del láser que tomamos como marca, tipo discontinuidad, en donde, D es la distancia del robot a la discontinuidad, P es la diferencia entre las lecturas y Θ es el ángulo donde se encuentra la marca respecto al frente del robot. En la imagen derecha se muestran los dos tipos de discontinuidad que podemos encontrar, donde i y d indican de qué lado se detectó la discontinuidad respecto al frente del robot izquierdo o derecho, respectivamente [2].

3.3. Detector SIFT

En esta sección describiremos el detector SIFT que es la base de las marcas visuales utilizada en este trabajo. Antes describimos brevemente el espacio-escala de una imagen y el operador LoG utilizada en el método SIFT.

Si una imagen se convoluciona con una función Gaussiana 2D de la forma: $G(x, y) = -e^{-\frac{x^2+y^2}{2\sigma^2}}$, donde σ es la desviación estándar, produce una imagen suavizada, cuyo grado de suavizado depende del valor de σ . La representación espacio-escala de una imagen, consiste en una familia de imágenes suavizadas a diferentes niveles de detalle (definido por un parámetro de escala t). La familia $L(x, y : t)$ es la representación espacio-escala de la imagen $I(x, y)$.

$$L(x, y : t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}} \quad (3.1)$$

La desviación estándar de la función Gaussiana es \sqrt{t} . En la figura 3.5 se presenta una familia de imágenes suavizadas a distintos niveles t .

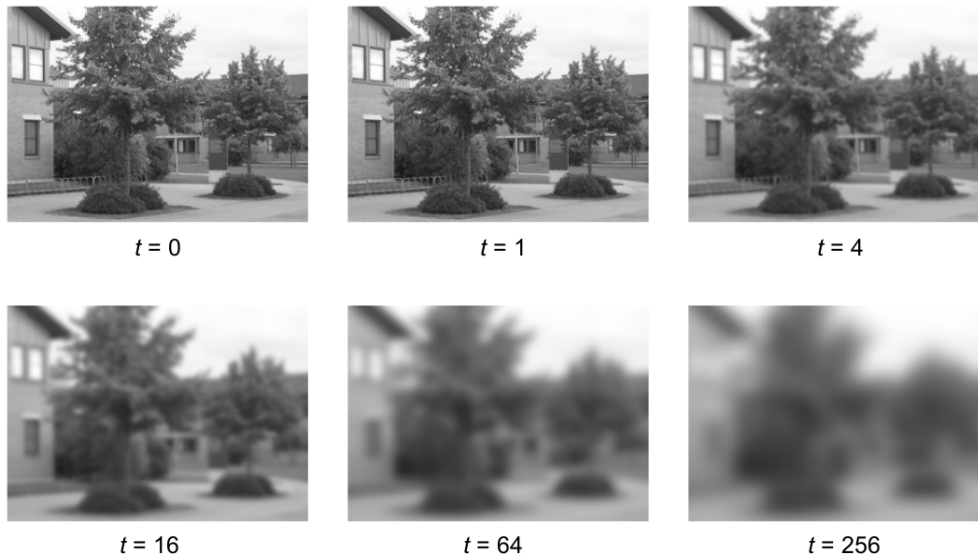


Figura 3.5: Imágenes suavizadas a niveles t mediante el filtro Gaussiano.

El Laplaciano de la función Gaussiana (LoG), es la segunda derivada de G respecto de r :

$$\nabla^2 G = - \left(\frac{r^2 - \sigma^2}{\sigma^4} \right) e^{-\frac{r^2}{2\sigma^2}} \quad (3.2)$$

Donde $r = (x^2 + y^2)$. LoG se puede aproximar mediante la diferencia de Gaussianas (DoG). El propósito de la Gaussiana en la formulación LoG es suavizar la imagen (reducir el ruido), mientras que el propósito del operador Laplaciano es proveer de información sobre la localización de los bordes en la imagen.

En 1999, David Lowe propuso un detector invariante a escala y rotaciones, conocido como *Scale-Invariant Feature Transform* (SIFT). También es parcialmente invariante a cambios de iluminación. Se combinan una representación espacio-escala y una pirámide Gaussiana (figura 3.6):

- Se calculan varias escalas entre t y $2t$ (una octava), todas con la misma resolución.
- A partir de la escala $2t$, se divide a la mitad la resolución y se usa esta resolución reducida para la siguiente octava, etc. El número de octavas influye en la estabilidad en cuanto a escala, en nuestro caso se calculan 3 octavas.

- Una vez obtenidas todas las escalas, se obtienen los LoG por sustracción entre dos escalas consecutivas.

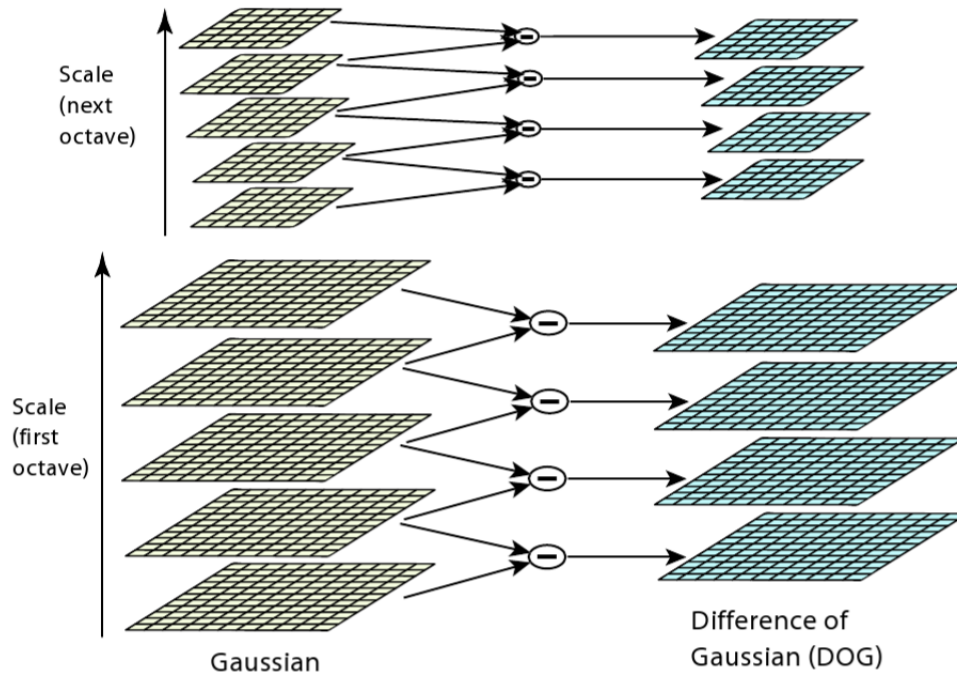


Figura 3.6: Pirámide Gaussiana y diferencias Gaussinas en la misma escala [4].

Se compara cada píxel del LoG con todos sus vecinos, tanto dentro de la misma escala como con las escalas vecinas. Sólo se seleccionan los extremos de espacio-escala (que sean mayores o menores que sus 26 vecinos), figura 3.7.

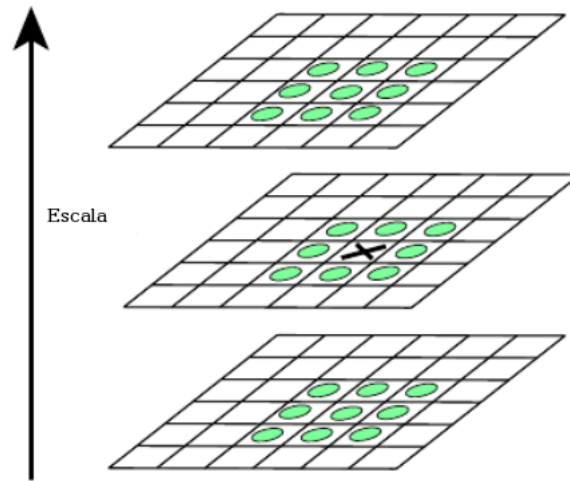


Figura 3.7: Comparativa de cada píxel con sus 26 vecinos en la misma escala y las escalas vecinas [4].

Tener un descriptor para cada punto de interés es útil para poder diferenciarlos. El descriptor SIFT está basado en histogramas de orientación del gradiente.

Ya que conocemos la escala t_0 del punto de interés, entonces ahora con $L(x, y : t_0)$ se calculan la magnitud y la orientación del gradiente en la vecindad del punto:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (3.3)$$

$$\Theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))) \quad (3.4)$$

Obteniendo así un histograma de orientación del gradiente, el máximo determinará la orientación del punto de interés. Se refina interpolando con una parábola entre los tres puntos más cercanos al máximo (ver figura 3.8).

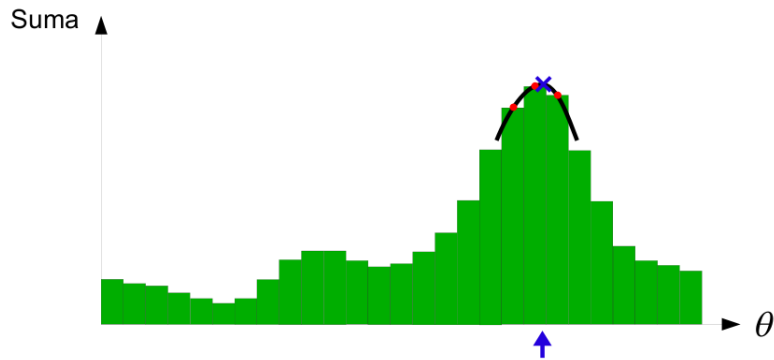


Figura 3.8: Histograma de orientación del gradiente y refinación de orientación del punto de interés mediante interpolación.

Para construir el vector característico de 128 elementos, se segmenta la vecindad en 4×4 regiones cada una de 4×4 píxeles, se genera un histograma de gradiente para cada región (figura 3.9). Se obtiene un histograma tridimensional de $4 \times 4 \times 8 = 128$ casillas, formando así un vector de tamaño 128. Este vector es normalizado para eliminar los cambios de contraste.

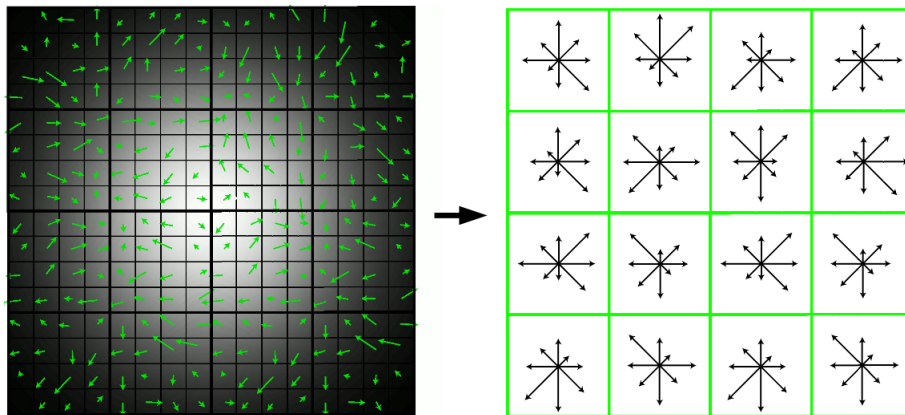


Figura 3.9: Segmentación de la vecindad en 4×4 regiones, cada una de 4×4 píxeles. Imagen izquierda, segmentación en vecindades de 4×4 y a la derecha histograma de gradiente para cada región representada mediante vectores.

3.4. Marcas visuales

Se dotó al robot móvil de un sistema de visión estéreo (figura 3.10), el cual puede obtener imágenes con una resolución de hasta 1288×1032 píxeles y con una velocidad de hasta 30 cuadros por segundo. El trabajar con la máxima resolución

implica un costo computacional mayor al detectar las marcas visuales, por tal motivo se trabaja con una resolución de 320x240. El objetivo de este sistema de visión es obtener imágenes en todo momento y mediante el método SIFT [4] detectar las marcas visuales y su ubicación 3D en el ambiente.



Figura 3.10: Sistema de visión estéreo STH-HMCS2-VAR-C.

El sistema coordenado del sistema de visión estéreo se muestra en la figura 3.11.

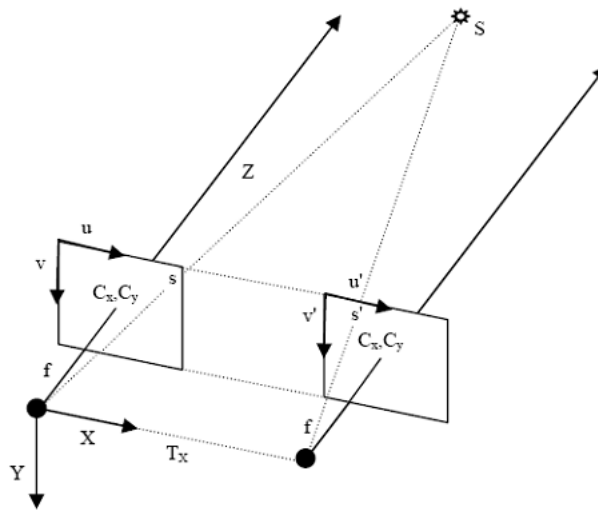


Figura 3.11: Sistema de coordenadas del sistema de visión. El origen del sistema se ubica en la lente izquierda; f representa la distancia focal; C_x, C_y coordenadas en el plano de la imagen por donde atraviesa el eje Z , positivo hacia el frente; X positivo hacia la derecha y Y positivo hacia abajo; u, v coordenadas de cualquier punto de la escena proyectado en las imágenes y T_x es la distancia de separación entre lentes.

Los parámetros del sistema estéreo se obtiene al momento de calibrar la cámara [10]. El propósito de la calibración de la cámara es determinar dos conjuntos de parámetros, internos y externos. Los parámetros internos corrigen la distorsión

de las lentes, y la desigualdad en la distancia focal en ambos lentes; los parámetros externos determinan la relación espacial de ambos lentes. Estos parámetros son usados para realizar transformaciones o rectificar las imágenes y así determinar la correspondencia en 3D.

Con las cámaras calibradas, el proceso para detectar las marcas visuales en cada instante, t , es el siguiente. Se obtiene la imagen de la lente izquierda del sistema de visión estéreo, se extrae un conjunto S_t^n de n marcas visuales mediante el método SIFT [4], $S_t^n = \{S_t^i\}$. Utilizando la matriz de calibración calculamos la posición 3D de cada marca visual, $o_t^i = \langle X_t^i, f_t^i \rangle$, donde X_t^i es la posición 3D y f_t^i es el vector característico de 128 elementos de la marca i . En la figura 3.12(a) se muestra un ejemplo de una imagen capturada y con las marcas visuales detectadas, marcadas con una *cruz* y en (b) la imagen de disparidad, que básicamente es la distancia horizontal entre puntos correspondientes; esto es proporcional a la distancia 3D de cada correspondencia de puntos. Los píxeles en tonos más claros indican objetos más cercanos a la cámara.

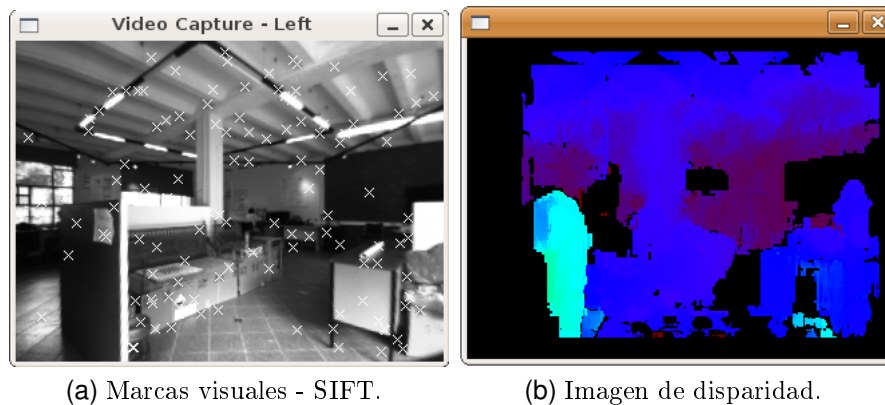
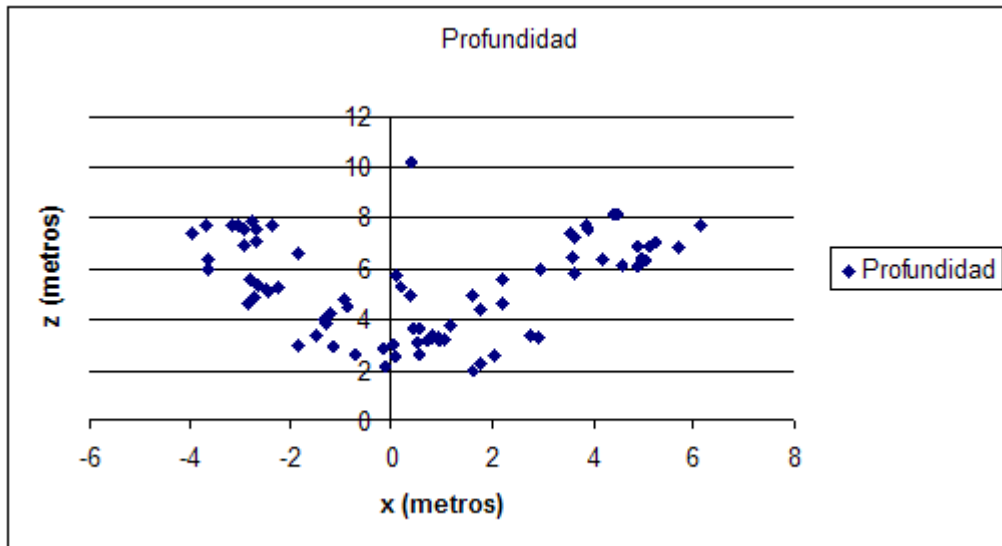


Figura 3.12: Marcas visuales detectadas en una imagen (a) indicadas con x . En (b) se ilustra el mapa de disparidad correspondiente a la imagen.

En la figura 3.13(a) se muestra otro ejemplo de marcas visuales detectadas; y en (b) se tiene la gráfica de las posiciones de las marcas visuales detectadas, en el plano XZ, que es paralelo al piso. Cabe mencionar que la exactitud en el cálculo de la posición depende de la calidad de la calibración de la cámara.



a)



b)

Figura 3.13: a) Imagen del lente izquierdo y marcas visuales detectadas. b) Posición de cada de una de las marcas en el plano X,Z, que es paralelo al piso.

3.5. Representación del ambiente e integración de marcas naturales

Recordemos que utilizamos tanto marcas de rango como visuales. Cada vez que el robot detecta una marca I_t^i ya sea de rango o visual, ésta se representa de la siguiente forma:

$$I_t^i = (r^i, \phi^i, s^i) \quad (3.5)$$

Donde:

- r^i , distancia al robot
- ϕ^i , ángulo relativo con respecto al frente del robot
- s^i , características de la marca

La distancia está representada en metros y el ángulo en radianes. Con esta representación se pueden usar varios tipos de marcas, el parámetro s^i determina implícitamente el tipo de marca, si es visual $s^i = \alpha_t^i$ (ver sección 3.4), si fuera una marca de rango entonces $s^i = (P, m)$, donde m representa el tipo de discontinuidad, izquierda o derecha; y P , la profundidad de la discontinuidad. En la figura 3.14 se muestra la localización de una marca relativa al robot, utilizando la representación. Cabe mencionar que para la posición 3D de las marcas visuales solo se contemplan los ejes X, Z del sistema coordinado de la cámara, que es paralelo al plano del piso, la razón es porque el robot solo se mueve en ese plano. Dicho de otra forma las marcas son proyectadas al plano del robot, haciendo $Y = 0$ para toda marcas visual. Al hacer $Y = 0$ se pierde información a nivel de representación del mapa de marcas, es decir, sino se hace 0 la componente Y se obtendrían mapas 3D de marcas visuales. Sin embargo, esto no afecta el proceso de localización porque el robot solo se mueve y se localiza en el plano X, Z .

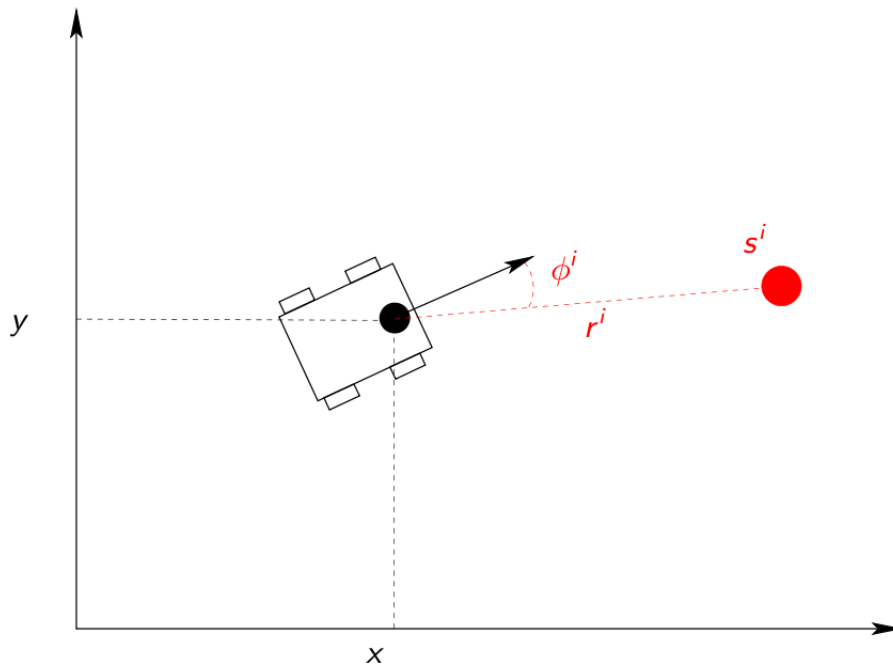


Figura 3.14: Localización de una marca relativa al robot. El círculo rojo es una marca detectada s^i , a una distancia r^i y ángulo ϕ^i respecto al frente del robot.

3.5.1. Representación final del ambiente

Para obtener una representación final del ambiente, primero se construye un mapa de celdas probabilista utilizando el láser y sonares mediante el método de filtro de partículas descrito en [1]. Después se enriquece dicho mapa con marcas visuales, la idea es realizar un recorrido por el ambiente para recolectar marcas visuales que tomaremos como referencias para realizar la localización global posterior. Partimos de un estado inicial del robot conocido, el robot realiza un recorrido por el ambiente localizándose localmente y al mismo tiempo detectando marcas visuales, en el recorrido, el robot es guiado por el usuario. Las marcas detectadas en el recorrido son proyectadas al plano del robot y son asignadas a su correspondiente celda en el mapa probabilista. En el algoritmo 3.1 se ilustra este proceso y en las figuras 3.15 y 3.16 se muestra el resultado de enriquecer un mapa de celdas probabilista (a), obteniendo como resultado el mapa enriquecido o mapa de marcas visuales (b).

Uno debería esperar que las marcas visuales deben estar en las celdas ocupadas cuya probabilidad de ocupación sea 1.0, es decir en zonas que representa paredes

o muebles que es donde comúnmente se detectan marcas, porque se encuentran objetos sobre ellos. Sin embargo, se observan marcas visuales en celdas cuya probabilidad de ocupación es de 0.5, que corresponde a zona no explorada. La razón es porque el campo de visión de la cámara rebasa los límites del ambiente de trabajo; es decir, los obstáculos detectados por el sensor láser. Las marcas que se encuentran sobre el espacio no ocupado corresponden a marcas detectadas en el techo del ambiente, a muebles con soportes delgados que el láser no detectó.

Algoritmo 3.1 Agregar información de marcas visuales al mapa construido a base de láser y sonares en [1].

1. Datos de entrada: mapa del ambiente modelado en rejillas de ocupación probabilista.
 2. Salida: mismo mapa con información adicional de marcas visuales.
 3. Desde una posición conocida del robot dentro del mapa.
 4. Iniciar un recorrido.
 - a) Hacer seguimiento de la posición.
 - b) Mientras se realiza el recorrido, recordemos que el conjunto C tiene como elementos las correspondencias de las marcas detectadas en el instante t y $t + \Delta t$. A C ya se le aplicó RANSAC.
 - 1) Para cada elemento (a_i, b_i) del conjunto C que corresponda a una relación de marcas visuales.
 - Tomar la primera componente, a_i , y calcular su posición en el mapa. Es decir a que celda, $c_{i,j}$, corresponde. La posición se calcula mediante la matriz de calibración de la cámara.
 - A la celda correspondiente, $c_{i,j}$ se le asigna la información de la marca visual si es que no es una marca repetida.
 5. Terminar recorrido.
-

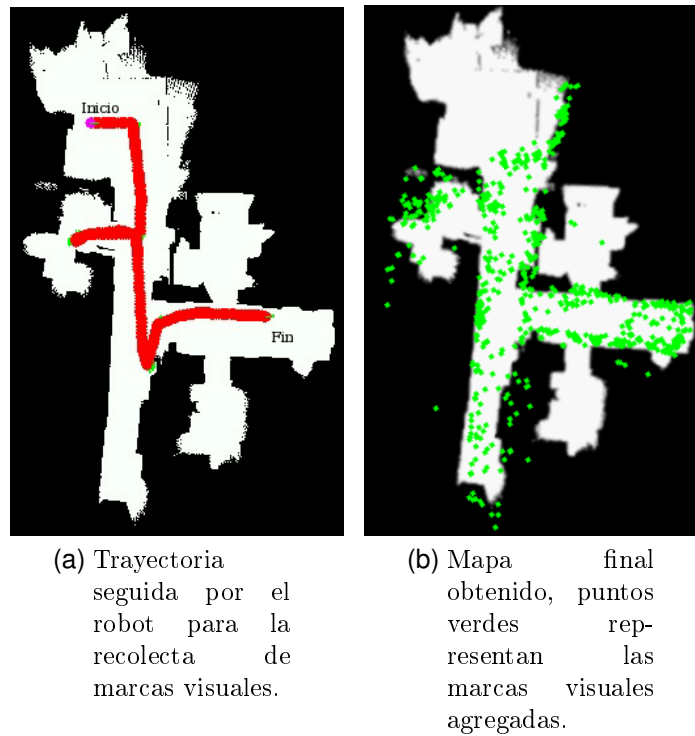


Figura 3.15: Ejemplo de agregar información de marcas visuales a un mapa construido en [1]. a) Se muestra el recorrido del robot marcada en rojo. b) Mapa resultante de agregar información de marcas visuales, donde los puntos verdes son las marcas visuales detectadas en el recorrido asignadas a su correspondiente celda dentro del mapa.

Capítulo 4

Localización

4.1. Introducción

Tener la capacidad para localizarse dentro de un ambiente no estacionario es un problema fundamental dentro del área de robótica móvil, así mismo es un requisito para realizar diversas tareas en las que el robot debe conocer su posición dentro del ambiente para poder ejecutarla, por ejemplo en tareas donde sea indispensable la navegación. Existen dos tipos de localización, local y global. En la primera el robot conoce su posición inicial y el objetivo es realizar un seguimiento de la posición, es decir, ir estimando en todo momento la nueva posición del robot, mientras éste se mueve. En la localización global el robot desconoce su posición inicial y su tarea es averiguarla, sin información *a priori* acerca de su posición dentro del mapa.

En este capítulo se introducen algoritmos para la localización local y global basados en marcas naturales para un robot móvil en ambientes interiores no estacionarios.

La idea general para resolver el problema del seguimiento de la posición, consiste en el apoyo de marcas naturales que toma como referencia. Mientras el robot navega, en cada instante se detecta un conjunto de marcas naturales y se halla una correspondencia con el conjunto de marcas anterior, luego con el método de RANSAC [12] eliminamos correspondencias erróneas. Finalmente, mediante la técnica de triangulación, se estima la nueva posición de robot. Este proceso se

describe detalladamente en la sección 4.2.

Para la parte de localización global inicialmente se tiene un modelo del ambiente basado en rejillas de ocupación. Después de realizar un recorrido por el ambiente con la finalidad de recabar marcas visuales. Ahora para localizar al robot globalmente, se obtienen las marcas visuales desde su posición y de manera análoga a la localización local, se estima la posición del robot utilizando las marcas observadas con respecto a las marcas del mapa. En la sección 4.3 se detalla el proceso.

4.2. Localización local

En el seguimiento de la posición, el robot estima su posición mientras navega apoyándose de marcas naturales que toma como referencia. El esquema general de este proceso se muestra en la figura 4.1. Se inicia el proceso de navegación, en el tiempo t se detecta el conjunto de marcas naturales, A_t , desde la posición conocida, $s_t = (x, y, \theta)$, el robot se mueve y después de un cierto intervalo de tiempo Δt se vuelve a extraer el conjunto de marcas naturales, $A_{t+\Delta t}$, en el tiempo $t + \Delta t$ desde la posición $s_{t+\Delta t}$. Se halla un conjunto C de correspondencias entre ambos conjuntos, por último se estima la posición y orientación $s_{t+\Delta t} = (x, y, \theta)$ respecto al mapa mediante la técnica de triangulación. Este proceso se repite mientras no se termine la navegación.

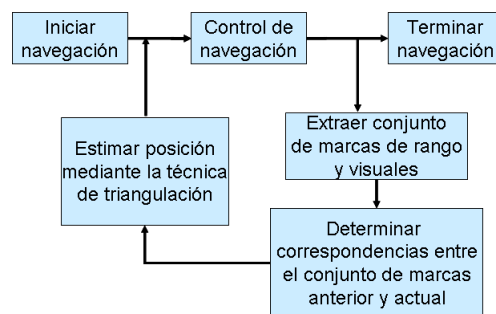


Figura 4.1: Esquema del proceso de localización local. Se obtiene el conjunto de marcas en cada instante y mientras haya movimiento, se establece una correspondencia entre el conjunto de marcas actual y el previo, y se estima la nueva posición del robot mediante la técnica de triangulación.

4.2.1. Correspondencia entre conjunto de marcas

Dados dos conjuntos de marcas naturales, A_t y $A_{t+\Delta t}$, el objetivo es encontrar la correspondencia $C = \{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}$ entre ambos conjuntos, en donde a_i es un elemento del conjunto A_t y b_i pertenece al conjunto $A_{t+\Delta t}$. Dicho de otra forma, a_i y b_i es la misma marca detectada desde la posición s_t y $s_{t+\Delta t}$ respectivamente. La dificultad del problema de correspondencia radica en que:

- Cada conjunto es tomado desde una ubicación diferente. La información con que se cuenta es, una estimación aproximada de la distancia recorrida por el robot proporcionada por el odómetro y para cada marca visual un vector de 128 elementos que la caracteriza.
- Puede haber marcas que ya no están presentes por haber quedado ocultas tras algún objeto o por el contrario puede haber nuevas marcas de nuevas regiones que no eran visibles desde la posición anterior.
- También existe la posibilidad de que algunas marcas hayan cambiado de posición, debido a un error de medición en los sensores o que en efecto hayan cambiado de posición al tratarse de un ambiente dinámico.

Consideremos que si establece una correspondencia equivocada, esto afectaría la nueva estimación de la posición del robot. Recordemos que estamos utilizando dos tipos de marcas, de rango y visuales, por tal motivo la correspondencia se hace entre marcas del mismo tipo. Este proceso se realiza en dos etapas:

1. Se establece una correspondencia de acuerdo a la estimación de desplazamiento del odómetro. Para las marcas visuales también se utiliza el vector característico de cada una de las marcas para hallar la correspondencia mediante la distancia euclidiana.
2. Después de la primera etapa tenemos una primera aproximación del conjunto C . Mediante el método RANSAC [12] y los atributos de posición de cada marca, hallamos una transformación $H : a_i \rightarrow b_i$ con un margen de error ϵ de tal forma que se obtenga el mayor número de pares (a_i, b_i) que cumplan con la transformación H . Como resultado tenemos un conjunto consistente de correspondencias.

A continuación se describe el proceso para determinar las correspondencias entre conjuntos de marcas de diferentes tipos.

Correspondencia entre marcas de rango

Sea A'_t el conjunto de marcas de rango detectadas en el instante t y $A'_{t+\Delta t}$ el conjunto de marcas de rango detectadas en el instante $t + \Delta t$. De acuerdo a la estimación del odómetro se calcula la posición (x_i, y_i) en donde se esperaba encontrar cada marca $a_i \in A'_t$. Para cada marca a_i , se busca la marca más cercana $b_j \in A'_{t+\Delta t}$ y se asocian, formando así la relación (a_i, b_j) .

La cercanía se define como la distancia Euclidiana y solamente se busca dentro de una región delimitada por un círculo alrededor de la posición esperada (x_i, y_i) con un radio proporcional al máximo error que puede arrojar la estimación del odómetro; si el odómetro fuera perfecto y nuestro sistema de detección de marcas también lo fuera, la marca a_i se encontraría justo en el lugar donde se esperaba, como sabemos que no lo es, se busca en la región mencionada.

En la figura 4.2(c) se ilustra este procedimiento: los cuadrados representan las marcas de rango en el tiempo t mientras que los círculos pequeños representan las marcas de rango en el tiempo $t + \Delta t$; los círculos más grandes representan el área donde se busca la correspondencia más cercana.

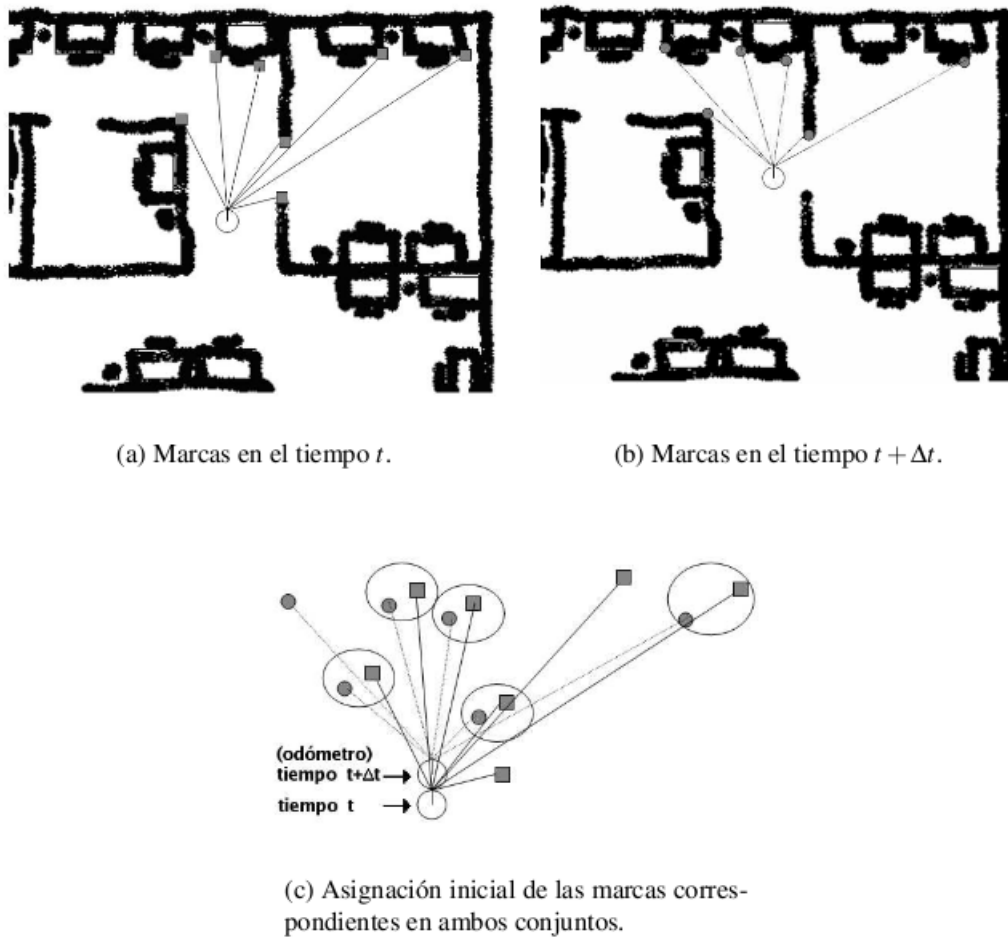


Figura 4.2: Correspondencia entre conjuntos de marcas de rango[2]. a) Posición conocida del robot en el tiempo t y marcas detectadas, representadas por cuadros; b) Posición del robot que se desea estimar en el tiempo $t + \Delta t$ y marcas detectadas, representadas por círculos; c) Correspondencias encontradas, los círculos grandes representan la región de búsqueda.

Correspondencia entre marcas visuales

La ventaja de utilizar marcas visuales es que cuentan con un mayor número de atributos (vector característico) que facilita diferenciarlos más fácilmente. Por lo que podemos determinar si un elemento a del conjunto A_t está presente en el conjunto $A_{t+\Delta t}$ utilizando el vector característico. Se busca a_i con su vector característico f_t^i más parecido en el conjunto $A_{t+\Delta t}$, si la distancia euclidiana entre f_t^i y $f_{t+\Delta t}^j$ es menor a un umbral, entonces la marca b_j corresponde a a_i .

El problema de correspondencia se reduce a una búsqueda de vecinos más cercanos. Un algoritmo básico es comparar el vector f_t^i de a_i con todos los vectores de los elementos de $A_{t+\Delta t}$. Sin embargo, si se tienen muchos elementos en ambos conjuntos esto tiene un alto costo computacional debido a la alta dimensión, por ejemplo, el tiempo computacional de búsqueda del vecino más cercano en árboles-Kd es de $O(2^D)$ con $D = \text{dimensión}$. Una solución consiste en hacer una búsqueda aproximada del vecino más cercano, el algoritmo encuentra uno de los vecinos más cercanos, pero no necesariamente el más cercano. Por ejemplo, el algoritmo *BestBinFirst* (BBF) propuesto por Beis y Lowe (1997)[14]. Es una variante de los árboles-kd, pero las casillas son buscadas por proximidad espacial más que por proximidad en el árbol, y se impone un límite de puntos visitados.

4.2.2. Árboles K-d

Un Árbol k-d (árbol k-dimensión) es una estructura de datos que organiza los puntos en un espacio Euclídeano de k dimensiones. Para construir un árbol K-d, básicamente se van organizando los vectores dimensión por dimensión formando grupos cada vez más pequeños, para nuestro caso tenemos vectores de 128 elementos que caracterizan cada marca SIFT.

Algoritmo 4.1 Resumen del algoritmo de construcción de árboles K-d

1. Empezar con la dimensión $d = 1$.
 2. Buscar la *mediana* en la componente d .
 - a) Formar un grupo de vectores que tienen su componente d menor que la mediana, y otro que la tiene mayor.
 3. Incrementar la dimensión $d = d + 1$.
 4. Para cada grupo que tenga más de un elemento, repetir etapas 2-3.
-

Un ejemplo de construcción de un árbol k-2, siguiendo el algoritmo 4.1 con el conjunto vectores $A = \{(5, 4), (9, 6), (7, 2), (8, 1), (4, 7), (2, 3)\}$, se muestra en la tabla 4.1. La búsqueda del vecino más cercano con d -dimensión, se hace en el

tiempo $O(2^d)$. En la práctica, la técnica no es utilizable para $d > 20$, existen muchos variantes de los árboles-kd, pero en dimensionalidad alta ningún algoritmo es capaz de encontrar el vecino más cercano más eficientemente que la búsqueda exhaustiva [17].

Tabla 4.1: Construcción de un árbol k-2, de acuerdo al algoritmo 4.1.

Operación	Resultado
1.- Empezamos con la dimensión $d = 1$.	$d = 1$. $A = \{(5, 4), (9, 6), (7, 2), (8, 1), (4, 7), (2, 3)\}$.
2.- Ordenamos A de forma ascendente de acuerdo a la componente d y buscamos el vector que tiene la componente d mediana.	$A = \{(2, 3), (4, 7), (5, 4), (7, 2), (8, 1), (9, 6)\}$. El elemento con la la mediana es $(7, 2)$.
3.- Formamos un grupo de vectores que tienen su componente $d = 1$ menor que la mediana y otro mayor que la mediana.	Grupo menor a la mediana: $A_1 = \{(2, 3), (4, 7), (5, 4)\}$ Grupo mayor a la mediana: $A_2 = \{(8, 1), (9, 6)\}$
4.- Incrementamos dimensión.	$d = d + 1 = 2$
5.- Repetimos pasos 2-4, para cada grupo que tenga más de un elemento.	Paso 2 para A_1 , con $d = 2$: $A_1 = \{(2, 3), (5, 4), (4, 7)\}$, mediana $(5, 4)$. Paso 3 para A_1 , con $d = 2$: $A_{11} = \{(2, 3)\}$ y $A_{12} = \{(4, 7)\}$. Paso 2 para A_2 , con $d = 2$: $A_2 = \{(8, 1), (9, 6)\}$, mediana $(9, 6)$. Paso 3 para A_2 , con $d = 2$: $A_{21} = \{(8, 1)\}$.
Árbol construido.	<pre> graph TD N72((7,2)) --- N54((5,4)) N72 --- N96((9,6)) N54 --- N23((2,3)) N54 --- N47((4,7)) N96 --- N81((8,1)) </pre> <p>The diagram shows a binary tree structure. The root node is (7,2) with a red 'x' next to it. The left child is (5,4) with a blue 'y' next to it. The right child is (9,6) with a blue 'y' next to it. The left child of (5,4) is (2,3) with a blue 'x' next to it. The right child of (5,4) is (4,7) with a blue 'x' next to it. The right child of (9,6) is (8,1) with a red 'x' next to it.</p>

4.2.3. Algoritmo *Best Bin First* (BBF)

Es una variante de los árboles k-d, la diferencia en la construcción radica en que la partición se realiza según la dimensión con mayor varianza y el umbral es el valor mediano, sin tener que ser un elemento. En figura 4.3 se ilustra un

ejemplo del proceso de construcción de un árbol mediante el algoritmo BBF.

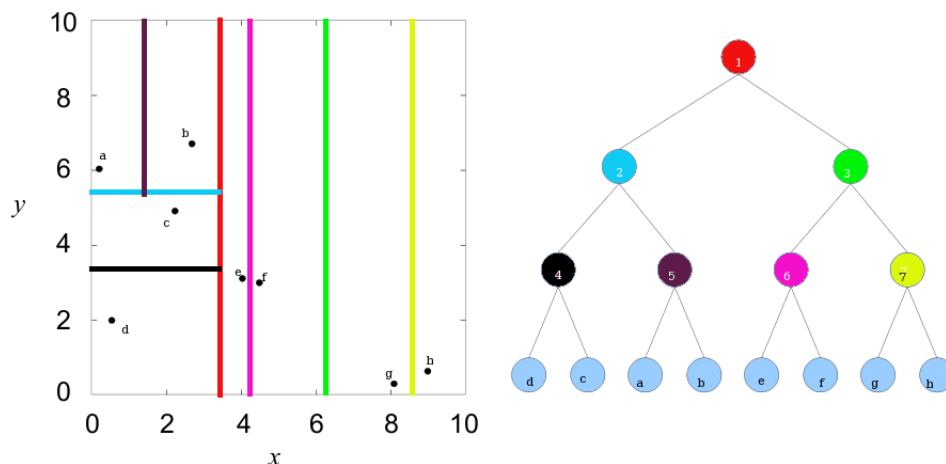


Figura 4.3: Proceso de construcción de un árbol mediante el algoritmo BBF.

Imagen izquierda, conjunto de datos, $A = \{a, b, \dots, h\}$, representados en el plano x, y ; líneas verticales, umbrales de las particiones respecto a la dimensión, $d = 1$ o primera componente de los vectores en A ; líneas horizontales, umbrales de las particiones considerando la dimensión, $d = 2$. Imagen derecha, árbol resultante. La enumeración de los nodos son simplemente etiquetas para identificar los nodos umbrales, los colores en los nodos representan sus correspondientes valores en el sistema cartesiano de la imagen izquierda, a excepción de los nodos terminales etiquetados con letras, que son los respectivos elemento del conjunto A .

1. Particionamos al conjunto de datos, $A = \{a, b, \dots, h\}$, de acuerdo a la dimensión con mayor varianza. Que en este caso es $d = 1$, $A_1 = \{a, d, c, b\}$ y $A_2 = \{e, f, g, h\}$. El umbral entre ambos conjuntos, es el valor medio entre el elemento b y e , que corresponde al nodo 1, su valor es ilustrado en la imagen izquierda de acuerdo al color del nodo (rojo).
2. Repetimos el paso 1 para el conjunto $A_1 = \{a, d, c, b\}$. La dimensión con mayor varianza es $d = 2$, particionamos el conjunto de acuerdo a d , $A_{11} = \{d, c\}$ y $A_{12} = \{a, b\}$. El umbral entre ambos conjuntos lo representa el nodo 2, que es el valor medio entre c y a .
3. El proceso se repite para todos los conjuntos con cardinalidad mayor a 1, resultantes de las particiones.

Para realizar una búsqueda aproximada de vecino más cercano en un árbol construido mediante el algoritmo BBF, seguimos el algoritmo 4.2. Por ejemplo, para

características SIFT (dimensión 128), con 100,000 vectores, limitando a 200 puntos terminales visitados, es aproximadamente 100 veces más rápido que una búsqueda exhaustiva con un 5% de correspondencias equivocadas, es decir, este algoritmo no garantiza encontrar el vecino más cercano [14].

Algoritmo 4.2 Algoritmo de búsqueda aproximada de vecino más cercano (BBF).

1. Entrada: vector, v .
2. Salida: vector vecino más cercano, c .
3. Iniciamos en el nodo raíz.
4. Mientras no se llegue a un nodo terminal:
 - a) Ir a nodo hijo, izquierdo o derecho de acuerdo al umbral del nodo actual.
 - b) Guardar la rama no visitada, clasificada por la distancia del nodo hijo elegido con el vector e respecto a su componente respectivo.
5. La distancia mínima actual, es igual a la distancia Euclidiana entre, $c = \text{nodo terminal}$, encontrado en paso 4 y el vector, v .
6. Saltar a la rama en espera no visitada (la más cercana al vector v). Eliminando las ramas más lejanas que la distancia mínima actual.
7. Repetir pasos 4-6, después de visitar un número predeterminado de nodos terminales.

De igual forma que en la correspondencia de marcas de rango, verificamos las correspondencias que se hallaron de acuerdo al vector de características, mediante la estimación del odómetro; obteniendo así una aproximación al conjunto de correspondencias $C = \{(a_i, b_j) \dots\}$. En la figura 4.4 se ilustran la correspondencia entre conjuntos de marcas visuales de cuadros consecutivos, utilizando el algoritmo BBF. Las líneas indican la correspondencia entre las marcas SIFT identificados en los extremos de cada línea.



(a)

Figura 4.4: Ejemplos de correspondencias entre conjuntos de marcas SIFT detectadas en el tiempo t y $t + \Delta t$. Imagen mitad superior t , imagen mitad inferior $t + \Delta t$.

La existencia de correspondencias erróneas repercute enormemente en la estimación de la nueva posición del robot. Esta mala relación se puede producir por varias causas:

- Error de medición en el sensor.
- Cambios en el ambiente.
- Por la técnica de búsqueda (algoritmo 4.2).

En la figura 4.5 se muestran dos ejemplos de malas correspondencias delimitadas por líneas rojas, entre el conjunto de marcas visuales detectadas en el instante

t , imagen mitad superior, y el conjunto detectado en el instante $t + \Delta t$, imagen mitad inferior. Cada extremo de cada línea es una marca SIFT detectada.

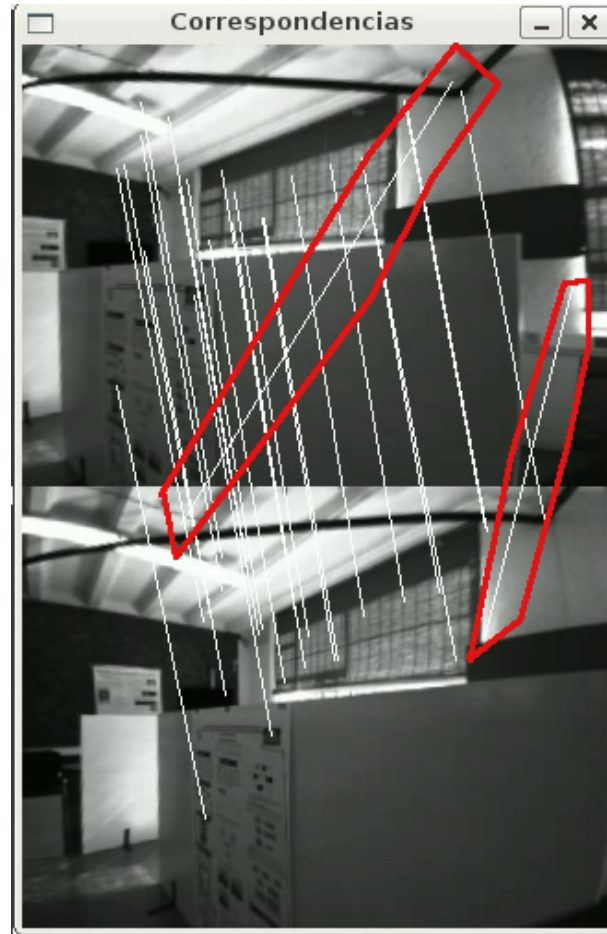


Figura 4.5: Correspondencia entre conjuntos de marcas visuales detectadas en cuadros tomados en el instante t y $t + \Delta t$, con 2 correspondencias erróneas. Las correspondencias erróneas están delimitadas por las líneas en color rojo.

Una alternativa para eliminar las correspondencias erróneas es verificar la consistencia geométrica [2] mediante la relación en cuanto a distancia que existe entre dichas marcas en el plano del robot. Es decir, la distancia que existe entre a_1 y a_2 debe ser equivalente a la distancia entre b_1 y b_2 y así sucesivamente. Cuando no son equivalentes se elimina dicho par de elementos del conjunto C . La desventaja de eliminar las correspondencias erróneas mediante consistencia geométrica es que no se sabe exactamente que correspondencia es la errónea, y al

eliminar ambas correspondencias existe una alta probabilidad de que se eliminen también correspondencias correctas. Para evitar esto se propone utilizar el método RANSAC [12] para eliminar el mayor número posible de malas relaciones o “outliers”.

Algoritmo 4.3 RANSAC.

1. Repetir N veces:
 - a) Escoger aleatoriamente k datos, donde k es el número mínimo de datos para ajustar el modelo.
 - b) Determinar los parámetros del modelo que se ajustan a los k datos.
 - c) Buscar todos los datos que son compatibles con el modelo estimado (*inliers*), dentro de un margen de error t .
 - d) Si el número de *inliers* es menor que d , entonces descartar el intento.
 - e) Sino, volver a ajustar el modelo con todos los *inliers* y estimar el error de ajuste.
2. Seleccionar el modelo con el menor error de ajuste.

La idea general del algoritmo del RANSAC se ilustra en el algoritmo 4.3. RANSAC se utiliza primeramente para eliminar correspondencias erróneas entre marcas visuales en el espacio de la imagen. Luego para correspondencia equivocadas de marcas de rango en el espacio del robot:

1. Los datos de entrada es el conjunto C' .
2. Escogemos 4 elementos aleatorios del conjunto C' y hallamos la transformación $H : a \rightarrow b$.
3. Buscamos cuantos otros puntos se ajustan la transformación H .
4. Si el número de elementos que ajustan es menor a d , repetimos el proceso con otros 4 puntos iniciales.
5. La mejor solución es la que tiene mayor soporte o consenso.

En la figura 4.6 se muestra el resultado de utilizar algoritmo RANSAC para eliminar correspondencias erróneas entre marcas visuales.



(a) Correspondencias erróneas eliminadas mediante RANSAC.

Figura 4.6: Ejemplo de aplicación de RANSAC. Imagen izquierda, correspondencias entre conjuntos de marcas visuales con algunas erróneas; imagen derecha, resultado de utilizar el algoritmo RANSAC para eliminar las correspondencias erróneas.

Después de utilizar RANSAC, tenemos el conjunto C de correspondencias, con poca probabilidad de contener correspondencias erróneas.

4.2.4. Estimación de la posición

Una vez que tenemos el conjunto consistente de correspondencias C , mediante triangulación podemos estimar la nueva posición del robot utilizando estas correspondencias de marcas como referencia. Se toman pares de elemento del conjunto C y mediante la técnica de triangulación obtenemos una hipótesis de la nueva posición del robot. Este proceso se ilustra en el algoritmo 4.4.

Entonces de acuerdo al algoritmo 4.4, se promedia un total de K hipótesis para obtener la nueva posición del robot en el tiempo $t + \Delta t$, donde:

$$\sum_{i=1}^{n-1} i = \frac{n(n+1)}{2} - n = K \quad (4.1)$$

Tabla 4.2: Notación para el algoritmo 4.4.

Notación	Descripción
$(\hat{x}_{Bi}, \hat{y}_{Bi})$	Hipótesis i de la posición del robot estimada en el plano x, y
p_i	peso para la hipótesis i
$(\bar{x}_{Bi}, \bar{y}_{Bi})$	Posición final estimada del robot

Algoritmo 4.4 Algoritmo para estimar la nueva posición del robot.

1. Datos de entrada: conjunto C .
 2. Salida: posición estimada del robot , $(\bar{x}_{Bi}, \bar{y}_{Bi})$.
 3. $P = 0$, suma de pesos.
 4. Si $|C| > 50$, elegir aleatoriamente sólo 50 elementos y eliminar de C los restantes.
 5. Desde $j = 1$ hasta $j = |C|$
 - a) Tomar el elemento j del conjunto C , $c_j = (a_j, b_j)$.
 - b) Desde $k = j$ hasta $k = |C|$.
 - 1) Tomar el elemento k del conjunto C , $c_k = (a_k, b_k)$.
 - 2) Estimar nueva posición del robot, $(\hat{x}_{Bi}, \hat{y}_{Bi})$, mediante triangulación. Tomando como referencias c_j y c_k .
 - 3) Se asigna un peso de acuerdo a la calidad de la posición estimada, $p_i = H(\hat{x}_{Bi}, \hat{y}_{Bi})$.
 - 4) $\bar{x}_{Bi} = \bar{x}_{Bi} + \hat{x}_{Bi} * p_i$.
 - 5) $\bar{y}_{Bi} = \bar{y}_{Bi} + \hat{y}_{Bi} * p_i$.
 - 6) $P = P + p_i$.
 6. Calculamos el promedio pesado para estimar la posición final del robot, $\bar{x}_{Bi} = \frac{\bar{x}_{Bi}}{P}$ y $\bar{y}_{Bi} = \frac{\bar{y}_{Bi}}{P}$.
-

con

$$n = |C|$$

Para asignar un peso de calidad de estimación a cada hipótesis nos basamos en el método propuesto en [3] donde se aborda formalmente el problema de la precisión, relacionado con el método de triangulación. Se determina la relación entre las marcas utilizadas y la calidad de la estimación obtenida. El análisis matemático determina que la variación de la precisión está dada por la ecuación $H = \text{sen}(\Theta)$ (ver figura 4.3). Donde H describe la relación entre las perturbaciones de la observación y la precisión de la estimación, Θ es el ángulo de separación entre las marcas, expresada en grados. Si el ángulo de separación, se acerca a 90° se le asigna un peso mayor de credibilidad debido a que la estimación obtenida se ve menos afectada por el ruido de los sensores. Por perturbación se refiere al ruido en los sensores.

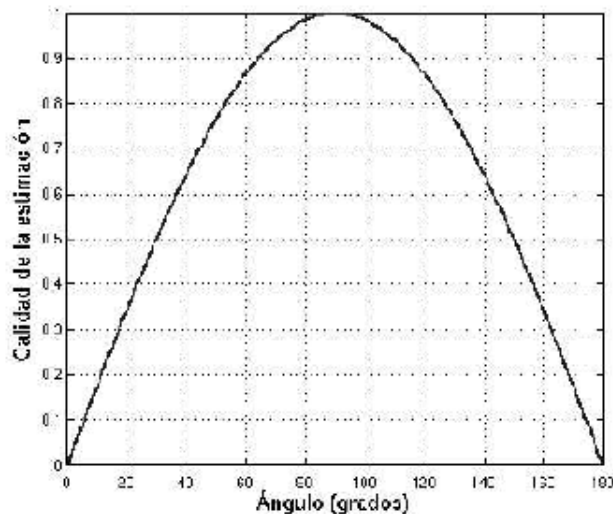


Figura 4.7: Calidad de estimación [3]. Si el ángulo de separación, entre las marcas que se toman como referencia para estimar la posición, tiende a 0° , la calidad de la estimación tiende a 0. Lo mismo ocurre cuando se acerca a 180° . Sin embargo cuando es un ángulo recto la calidad es 1, lo que indica que las estimaciones entre pares de marcas separadas por 90° , tendrán menos impacto por el ruido de los sensores en la detección de las marcas.

De esta forma a cada hipótesis se le asigna un peso $p_i = (\hat{x}_{Bi}, \hat{y}_{Bi})$ de acuerdo

a la calidad de estimación H .

4.2.5. Estimación de la orientación

Para estimar la orientación desde la nueva posición $B = (\bar{x}_{Bi}, \bar{y}_{Bi})$, podemos usar cualquier elemento del conjunto C , que determina las correspondencias entre el conjunto de marcas detectadas en el instante t y $t + \Delta t$. La diferencia de orientación entre la posición A y la posición B esta dada por :

$$\hat{\theta} = \beta - \beta' \quad (4.2)$$

Donde $\hat{\theta}$ es la cambio en la orientación del punto A al punto B. β es el ángulo con respecto a la orientación del robot de la marca m_k desde el punto B. β' es calculado utilizando las coordenadas de la marca m_i y la nueva posición del robot $(\bar{x}_{Bi}, \bar{y}_{Bi})$. De manera análoga a la estimación de la posición del robot, la estimación de la orientación del robot se obtiene del promedio de las diferencias en orientación que proporcionan las correspondencias entre el conjunto de marcas A_t y $A_{t+\Delta t}$.

$$\bar{\theta} = \frac{1}{N} \sum_{i=1}^N \hat{\theta} \quad (4.3)$$

Donde $\bar{\theta}$ es el promedio de las diferencias en orientación obtenidas de las correspondencias $c_i \in C$, N es la cardinalidad de $|C|$.

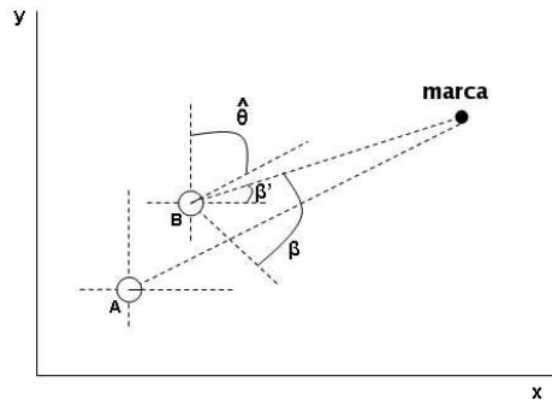


Figura 4.8: Estimación de la orientación del robot. A , posición del robot en el instante t . B , posición en el instante $t + \Delta t$. $marca$ detectada desde la posición A y B [2].

4.3. Localización global

El objetivo de la localización global es que el robot se localice dentro del mapa, sin alguna información *a priori* a cerca de su posición. Su único apoyo es un mapa del ambiente modelado en rejillas de ocupación probabilista, con la característica de que a este mapa se le ha agregado información de marcas visuales detectadas en el ambiente, en la sección 3.5.1 se detalla como se anexa esta información al mapa.

El esquema de este proceso de localización global se muestra en la figura 4.9. El robot inicialmente no conoce su ubicación dentro del mapa y su primer tarea es encontrar su posición.

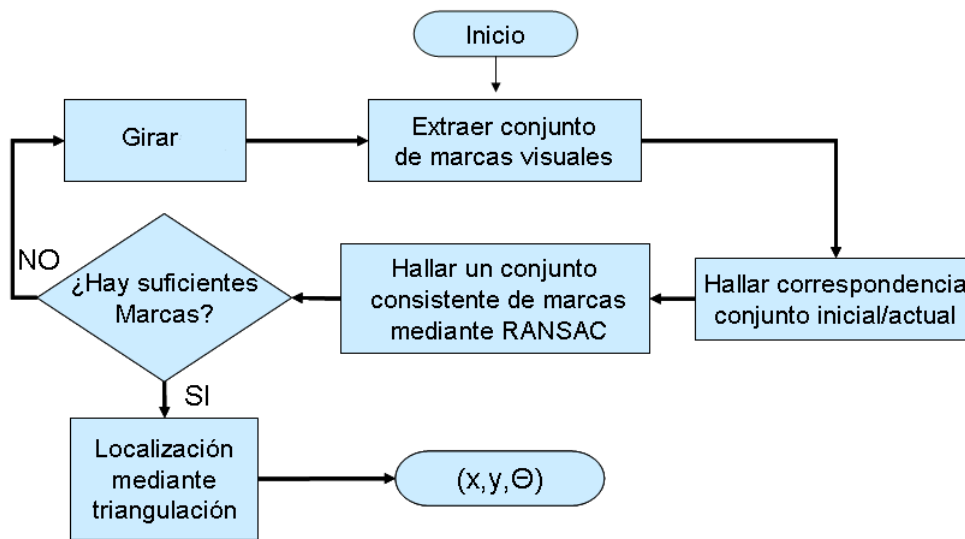


Figura 4.9: Esquema del proceso de localización global. Primeramente detecta las marcas visuales, halla la correspondencia con el conjunto de marcas visuales del mapa, si encuentra suficientes correspondencias, mediante triangulación estima su posición en el mapa. De lo contrario gira y se repite el proceso.

Se tiene un conjunto inicial A de marcas visuales del mapa que se recolectaron en el recorrido al momento de enriquecer el mapa. El robot gira sobre su propio eje extrayendo un conjunto B de marcas visuales. De manera análoga como en la localización local se halla la relación entre los conjuntos A, B , obteniendo un conjunto de correspondencias C . Se eliminan las correspondencias erróneas del conjunto C mediante el algoritmo RANSAC [12], y por último se estima la

posición del robot mediante triangulación utilizando ternas de elementos del conjunto C . En caso de que no haya suficientes marcas para localizarse, el robot gira nuevamente, el proceso termina cuando se halla localizado o halla completado un giro de 360° . El algoritmo 4.5 describe este proceso.

Algoritmo 4.5 Localización global a base de marcas visuales mediante triangulación.

1. Datos de entrada: A , conjunto de marcas visuales del mapa.
 2. Salida: posición del robot dentro del mapa.
 3. Detectar conjunto de marcas, B .
 4. Hallar el conjunto, C , de correspondencias en los conjuntos de marcas A, B .
 5. Eliminar correspondencias erróneas del conjunto C , utilizando el método RANSAC.
 6. Si $|C| < 30$ y el robot no ha girado completamente sobre su propio eje, girar e ir a paso 3. De lo contrario terminar ir a paso 7.
 7. Elegir ternas de elementos del conjunto C , mediante combinatoria sin repetición.
 - a) Para todas la ternas
 - 1) Utilizando la técnica de triangulación, estimar posición del robot.
 - b) Estimar posición final del robot, promediando todas las estimaciones.
 8. Terminar proceso, reportar estado del robot, *localizado o no localizado*.
-

4.4. Resumen

En este capítulo se introdujeron los algoritmos para estimar la posición del robot tanto en forma local como globalmente, mediante la técnica de triangulación tomando como referencia marcas naturales. Se describió el proceso para encontrar las correspondencias entre conjuntos de marcas de rango y conjunto de marcas visuales. Debido a la alta dimensionalidad del vector de características de las marcas visuales (SIFT), se utiliza el algoritmo de búsqueda del vecino mas

cercano BBF [14] que es una variante de los árboles k-d [15] para acelerar el proceso. Sin embargo, el algoritmo no garantiza encontrar el vecino más cercano o debido a cambios en el ambiente existe la probabilidad de que existan correspondencias erróneas. Para ello se propone usar el estimador robusto RANSAC[12] para descartar correspondencias erróneas. Se describieron los algoritmos para localización local y global que permite al robot localizarse aun ante cambios en el ambiente. En el siguiente capítulo se demuestra el funcionamiento de estos algoritmos en diferentes ambientes interiores.

Capítulo 5

Experimentos y resultados

En esta sección se describe una serie de experimentos realizados en cuatro ambientes diferentes bajo condiciones variables de iluminación y cambios en el ambiente, y se presentan estadísticas de cada una de las pruebas. Se presentan los mapas construidos con sensores de rango y enriquecidos con marcas visuales. En general se realizaron dos tipos de experimentos, seguimiento de la posición y localización global, con un total de aproximadamente 40 experimentos. La localización local se realizó en ambientes estáticos y cambiantes bajo las siguientes características:

1. Utilizando sólo marcas de rango.
2. Utilizando sólo marcas visuales.
3. Combinando marcas de rango y visuales.

Para la localización global se hicieron pruebas con el ambiente sin cambios y con cambios respecto al mapa inicial. Todos los experimentos se realizaron utilizando el robot PeopleBot que se muestra en la figura 5.2. El equipo de computo tiene las siguientes características, procesador *core 2 duo a 2.0 GHz* y 2 Gb en memoria *RAM*.

5.1. Ambientes de experimentación

Se utilizaron 4 ambientes diferentes con las siguientes características:

1. Interior del laboratorio de robótica del INAOE. Con una dimensión aproximada de 10.35m x 9.6m.
2. Interior del edificio Chavira del INAOE. Dimensiones aproximadas, 13m x 19.85m.
3. Pasillos construidos dentro del interior del lab. de robótica utilizando mamparas con una altura aproximada de 1.20m. Las dimensiones del ambiente son: 4m x 8.9m.
4. Ambiente construido en el interior del lab de robótica, simulando tres cuartos, un baño y un pasillo. Dimensiones del ambiente: 5.3m x 8m.

Los parámetros fijados para los distintos algoritmos utilizados en los distintos experimentos son los siguientes:

- Algoritmo BBF:
 - 200 terminales visitadas.
 - 0.49cm la distancia cuadrada mínima entre los 2 vecinos más cercanos (radio de búsqueda)
- Algoritmo SIFT:
 - Tamaño de imágenes 320x240 píxeles
 - 3 niveles en la pirámide de escala (octavas)
 - $\sigma = 1,6$ para la función Gaussiana (nivel de suavizado)
- Algoritmo RANSAC:
 - Margen de error para de calcular los *Inliers*: 15 cm utilizando sólo información de marcas visuales y de 7 cm con información de marcas visuales y de rango.

Los mapas respectivos se muestran en la figura 5.1.



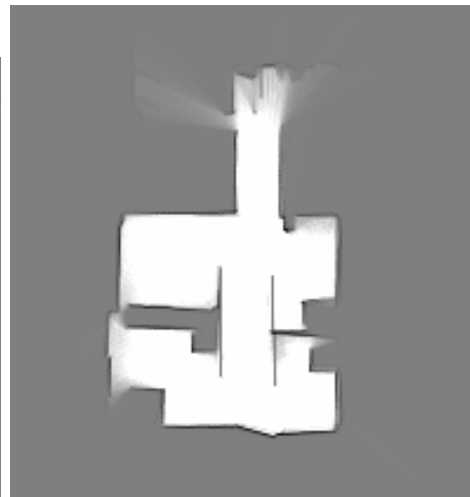
(a) Interior del laboratorio de robótica del INAOE.



(b) Interior del edificio Chavira del INAOE.



(c) Ambiente contruido simulando un pasillo.



(d) Ambiente construido en el interior del lab. de robótica.

Figura 5.1: Mapas construidos a base de sensores de rango [1] de los ambientes utilizados para la experimentación.

5.2. Sistema de visión

Se dotó de visión estéreo al robot móvil (figura 5.2) y se desarrolló un sistema bajo la arquitectura cliente/servidor que llamamos “Detector de marcas visuales”, cuya principal función es obtener imágenes en todo momento y detectar marcas visuales en dichas imágenes. El sistema atiende peticiones de imágenes y de marcas visuales. La razón de este diseño es para que otras aplicaciones puedan hacer uso de la cámara y realizar tareas diferentes a la localización.



Figura 5.2: Robot móvil PeopleBot dotado de un sistema de visión estéreo, un anillo de sonares y un sensor láser utilizando en la experimentación.

5.2.1. Experimento 1

El objetivo de este experimento es probar el funcionamiento y mostrar el rendimiento de nuestro sistema: “detector de marcas visuales”. Se realiza un seguimiento de la posición en el ambiente 1, pero solo mostramos los datos obtenidos por el sistema “detector de marcas visuales”. En la figura 5.3 se muestra la trayectoria del robot en el mapa, la trayectoria roja se estimó utilizando sólo

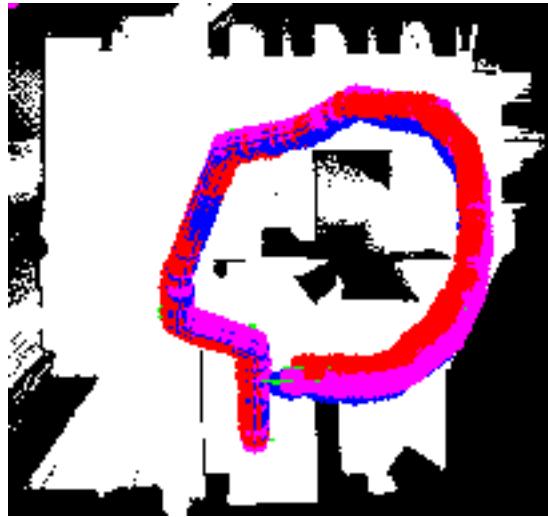


Figura 5.3: Recorrido del robot en el ambiente 1, para el experimento 1.

marcas de rango, la magenta utilizando sólo marcas visuales y la trayectoria azul se estimó utilizando ambos tipos de marcas.

El experimento consistió en realizar un recorrido por el ambiente 1 (figura. 5.3), el recorrido duró aproximadamente 4.5 minutos. Se obtuvieron los siguientes resultados: un total de 2,687 imágenes durante el recorrido, con un promedio de 100.29 marcas visuales detectadas por imagen y a su vez 61.11 marcas se pudo hallar su posición 3D en el ambiente. El tiempo promedio para detectar las marcas visuales en cada imagen y hallar su posición 3D en el ambiente fue de 0.134 segundos. En la figura 5.4 se muestra la gráfica de los datos obtenidos en el recorrido y en la figura 5.5 la gráfica de tiempo de procesamiento de cada imagen obtenida. Observamos que el tiempo de procesamiento depende del número de marcas detectadas, por tal razón las gráficas son semejantes. El número de marcas visuales detectadas está relacionada con el tipo de ambiente, si en el ambiente hay muchos objetos con textura (carteles, cuadros, muebles, etc), aumentará el número de marcas detectadas.

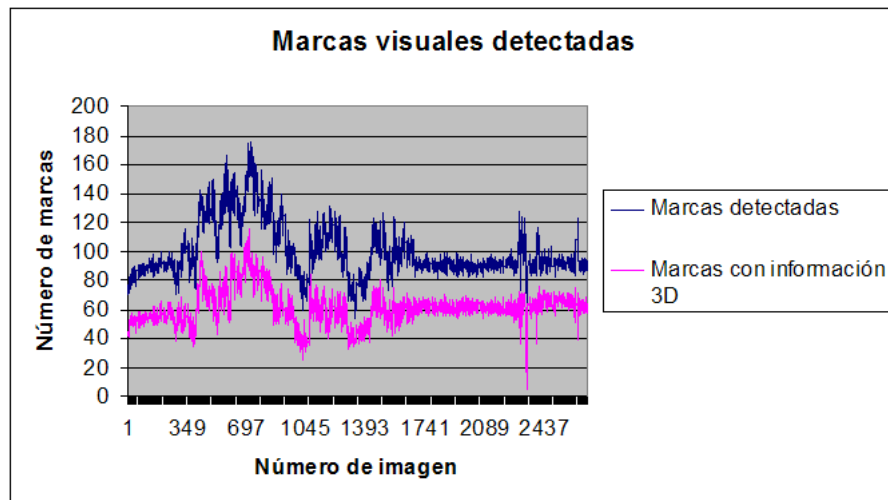


Figura 5.4: Número de marcas visuales detectadas en el recorrido del experimento 1 en el ambiente 1.

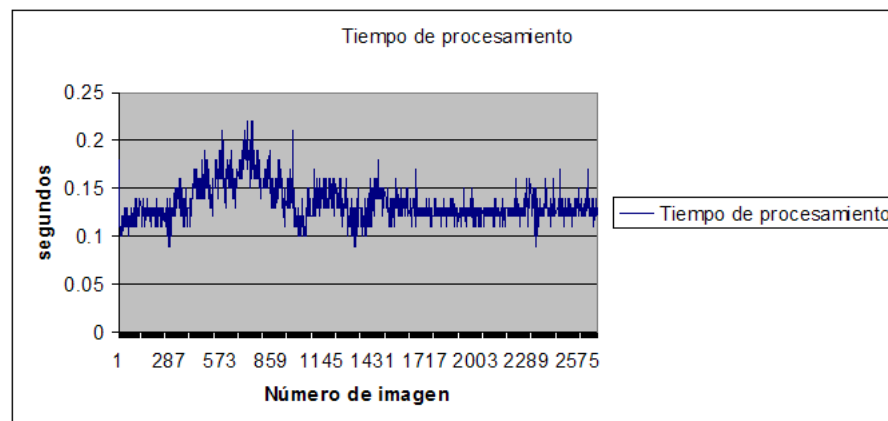


Figura 5.5: Tiempo de procesamiento para obtener las marcas visuales en cada imagen.

5.2.2. Construcción de mapa de marcas visuales

A continuación se muestran los resultados obtenidos después de hacer un recorrido por los ambientes mencionados al inicio de este capítulo. El procedimiento fue el siguiente:

1. La posición inicial del robot en el ambiente es conocida.
2. Iniciar un recorrido por el ambiente, el robot es comandado por el usuario.

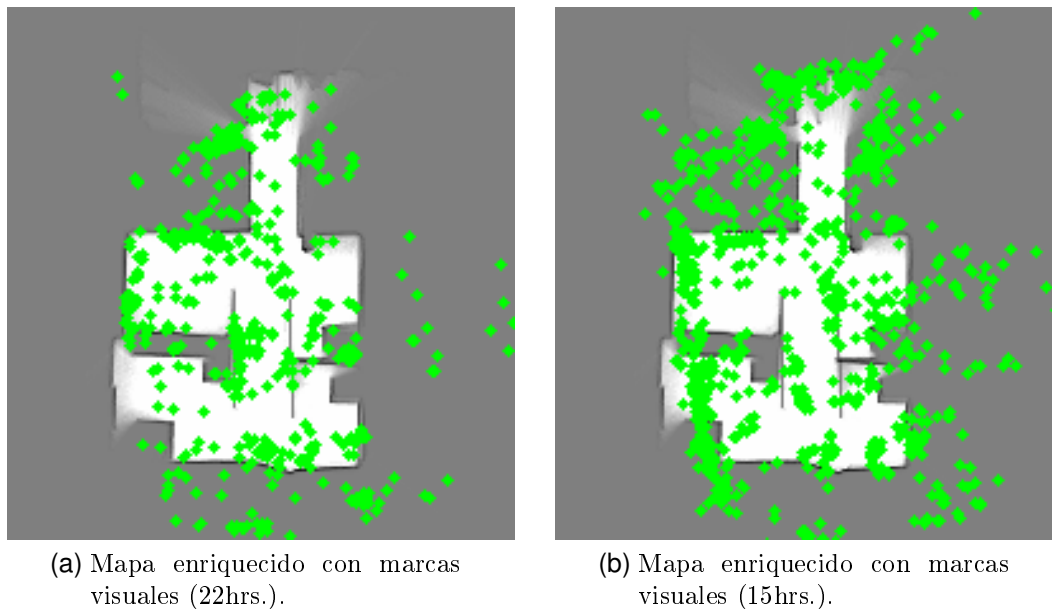
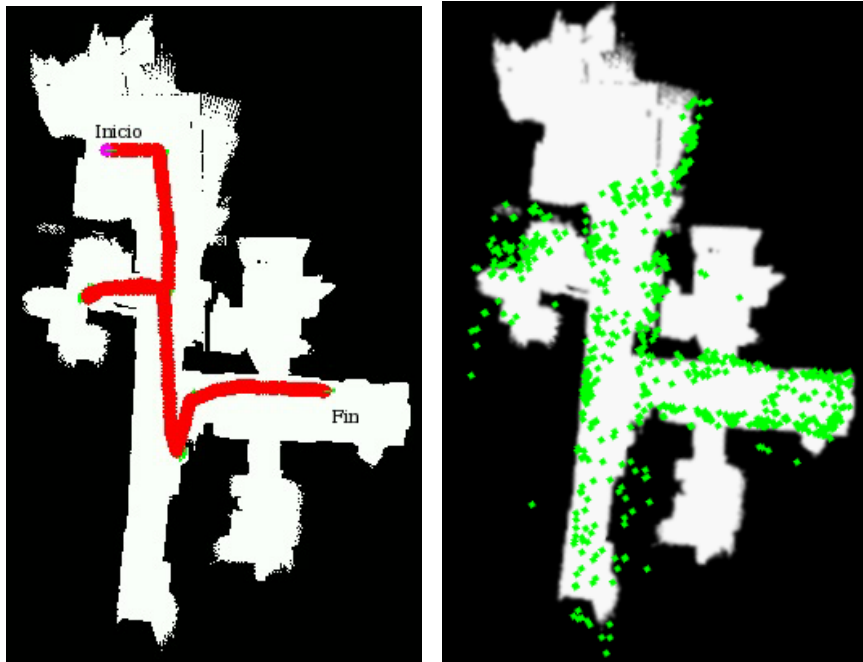


Figura 5.6: Mapa 4 con información visual, construidos en diferentes condiciones de iluminación.

3. Mientras se realiza el recorrido se va recabando información visual (Algoritmo 3.1).
4. Debido a que se realiza un seguimiento de la posición y las marcas que se consideran son marcas consistentes, es decir, mantienen su relación espacial (sección 5.2.2), ocurre que muchas marcas se repiten, por tal motivo se realiza un filtrado eliminando las marcas repetidas.

De esta forma se obtiene un mapa de marcas consistente; es decir, sólo se registraron aquellas marcas que permanecieron prácticamente en el mismo lugar a la hora de su detección.

El mapa mostrado en la figura 5.7(b) corresponde al interior del edificio Chavira del INAOE. Cabe mencionar que se construyó mientras había cambios en el ambiente, personas transitando, personas trabajando, etc. Esto lo podemos verificar en la figura 5.8.



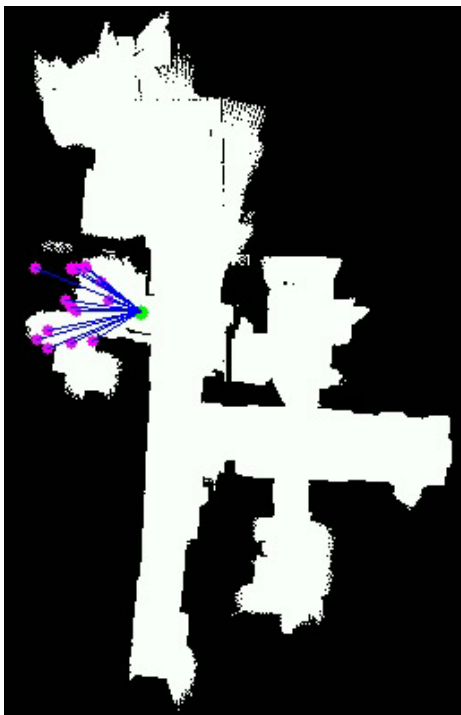
(a) Trayectoria seguida por el robot para la recolección de marcas visuales.

(b) Mapa final obtenido, puntos verdes representan las marcas visuales agregadas.

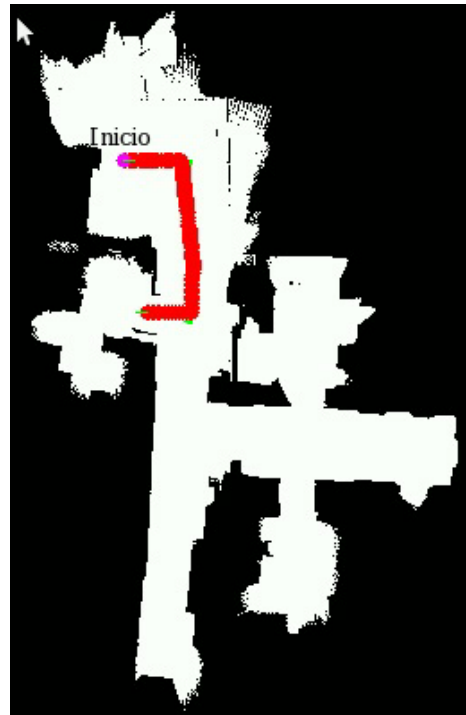
Figura 5.7: Mapa 2 con información visual. a) Trayectoria elegida, b) Mapa resultante.



(a) Imagen captura por la cámara y marcas visuales detectadas.



(b) Posición del robot respecto al mapa, círculo verde y posición respecto del mapa de las marcas detectadas en (a), círculos en color magenta. Sólo marcas que se consideraron consistentes.



(c) Trayectoria recorrida por el robot respecto del mapa hasta el momento.

Figura 5.8: Proceso de integración de marcas visuales al mapa 2, interior del edificio Chavira del INAOE.

Tabla 5.1: Integración de información visual.

Número de mapa	Imágenes procesadas	Marcas detectadas	Marcas sin repetición	Hora del experimento
2	422	8255	809	18hrs
3	375	11832	781	15hrs
4	633	17031	1083	16hrs

En la figura 5.6 mostramos el resultado de enriquecer el mapa 4 con marcas visuales en diferentes condiciones de iluminación, noche y tarde. Observamos que el mapa enriquecido en la tarde (15hrs) contiene un mayor número de marcas visuales que el mapa enriquecido en la noche (22hrs), esto es debido a que en la tarde hubo una mejor iluminación.

Del experimento 1 y el enriquecimiento de los mapas podemos concluir lo siguiente:

- El número de marcas detectadas depende de la iluminación y el tipo de ambiente, es decir, con mejores condiciones de iluminación y ambientes con objetos texturizados, mayor será el número de marcas detectadas. En el experimento 1 se muestra el rendimiento del “detector de marcas visuales”, el tiempo promedio de procesamiento fue de 0.134 seg. este tiempo es aceptable para el proceso global de localización.
- El proceso para enriquecer un mapa es básicamente hacer un seguimiento de posición e ir registrando aquellas marcas consistentes en cuanto a su relación espacial, es decir, se filtran aquellas marcas que se mueve, entonces el algoritmo para enriquecer el mapa con marcas visuales es tolerante a cambios en el ambiente mientras se realiza este proceso.

5.3. Seguimiento de la posición

Se realizó una serie de recorridos con el robot móvil en los diferentes ambientes bajo condiciones diferentes de iluminación y con las características mencionadas al inicio de este capítulo. El objetivo de los experimentos es evaluar las pruebas de localización local utilizando sólo marcas de rango, sólo marcas visuales y ambas. A continuación se describen algunos de los experimentos.

5.3.1. Experimento 2

Se realizó en el ambiente 1 aproximadamente a las 5 pm, se fue registrando el seguimiento de la posición utilizando sólo marcas de rango, sólo marcas visuales y combinada. El recorrido duró aproximadamente 5 minutos. La trayectoria recorrida por el robot en el plano x,z, que corresponde al piso del ambiente, se muestra en la figura 5.9 y respecto al mapa del ambiente en la figura 5.10. El ambiente permaneció estático mientras se realizó el recorrido. En la trayectoria de color azul el robot sólo tomó como referencias marcas de rango, en la de color magenta sólo marcas visuales y en la amarilla ambos tipos de marcas (figura 5.9).

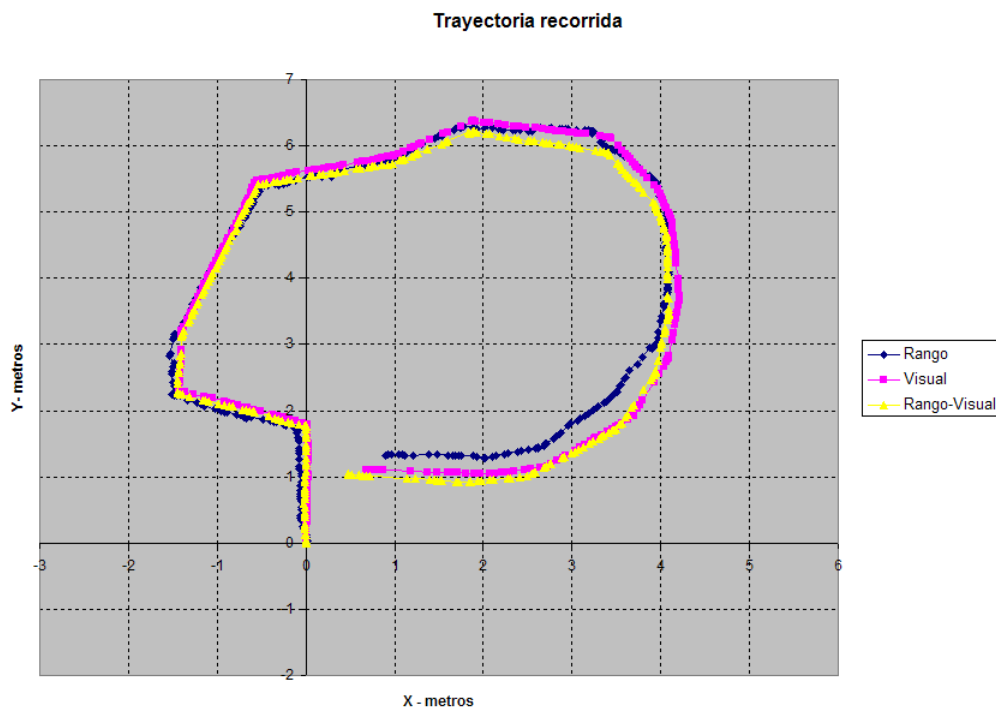


Figura 5.9: Trayectoria recorrida por el robot en el plano x,y para el experimento 2. La trayectoria estimada utilizando sólo marcas de rango (azul), cerca del final tiene un desfase de aproximadamente 15 cm. respecto a las otras estimaciones.

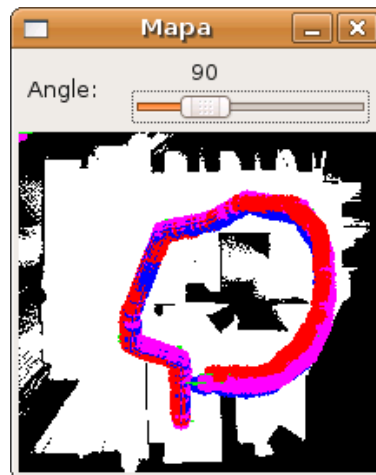


Figura 5.10: Trayectoria recorrida respecto al mapa del ambiente para el experimento 2.

La gráfica con el número de marcas consistentes en cada instante para estimar la nueva posición del robot se muestra en la figura 5.11 y la gráfica del tiempo de procesamiento en la figura 5.12. El tiempo promedio para estimar la nueva posición del robot en cada instante depende de cuantas marcas consistentes se tenga para realizar la estimación. Para este experimento el tiempo promedio fue de 0.090 segundos utilizando sólo marcas de rango, 0.155 segundos contemplando sólo marcas visuales y de 0.240 segundos utilizando ambas.

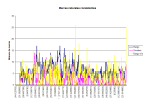


Figura 5.11: Número de marcas consistentes tomadas como referencia para estimar la posición del robot en cada instante.

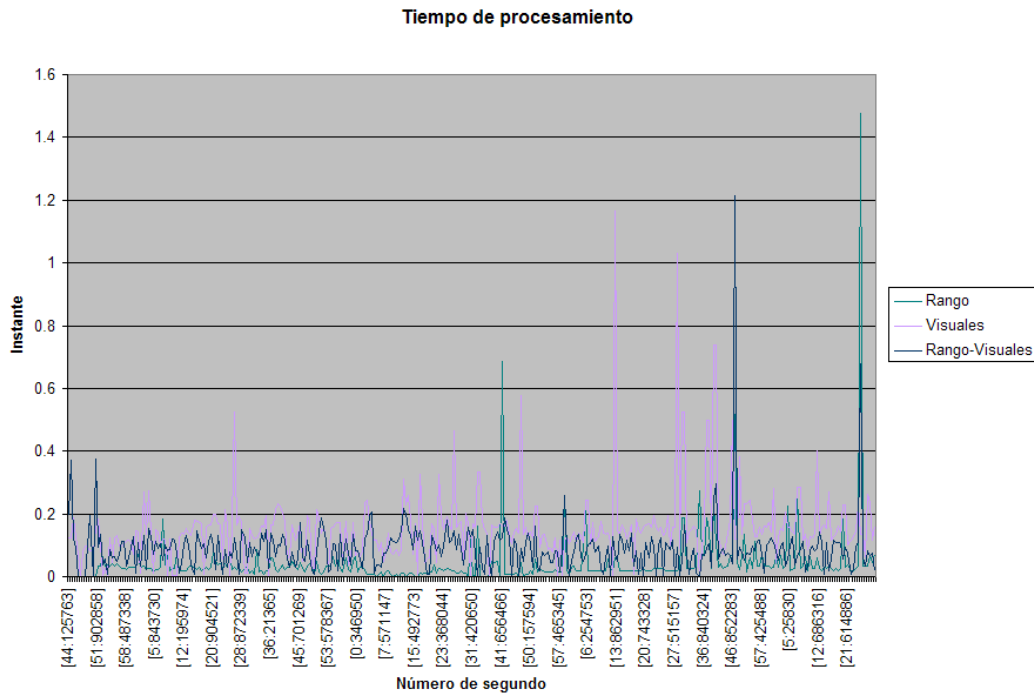


Figura 5.12: Tiempo de procesamiento para estimar la nueva posición del robot en cada instante.

5.3.2. Experimento 3

En este experimento se realizó en el ambiente 1. Se hizo un recorrido más corto, con la ventaja de que se gráfica la trayectoria real del robot obtenida manualmente. En la figura 5.14 y en la figura 5.3.2 se muestran las trayectorias estimadas respecto al plano x, z y mapa respectivamente, según la trayectoria de color rojo en la figura , estimada utilizando solo marcas de rango, el robot entra en colisión con un obstáculo, sin embargo, esto no ocurrió. La estimación tiene mayor error en esa parte y la razón es porque en ese lapso de tiempo no halló marcas de rango. Esto lo podemos corroborar observando la gráfica de marcas consistentes (figura 5.16). Aproximadamente a la mitad, tenemos que casi 8 segundos no se obtuvieron marcas de rango, entonces en esos 8 segundos el robot determinó su movimiento de acuerdo a la estimación del odómetro. En la figura 5.15 se muestra la gráfica de la orientación del robot durante el recorrido, cada color representa el tipo de marca que se usó para estimar la orientación. Observamos que son muy semejantes, los ligeros desfases son debido a que no están sincronizados, es decir,

en cada estimación no se necesariamente se trata de las mismas marcas visuales y de rango.

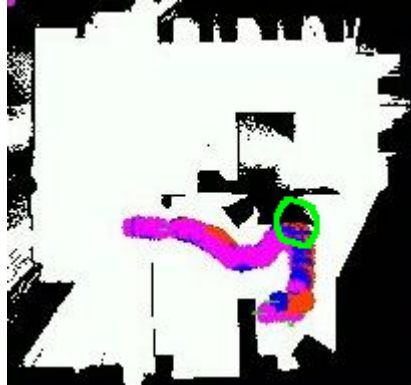


Figura 5.13: Trayectorias estimadas respecto el mapa del experimento 3. En parte encerra en el semicírculo hubo una colisión según la trayectoria estimada utilizando sólo marcas de rango.

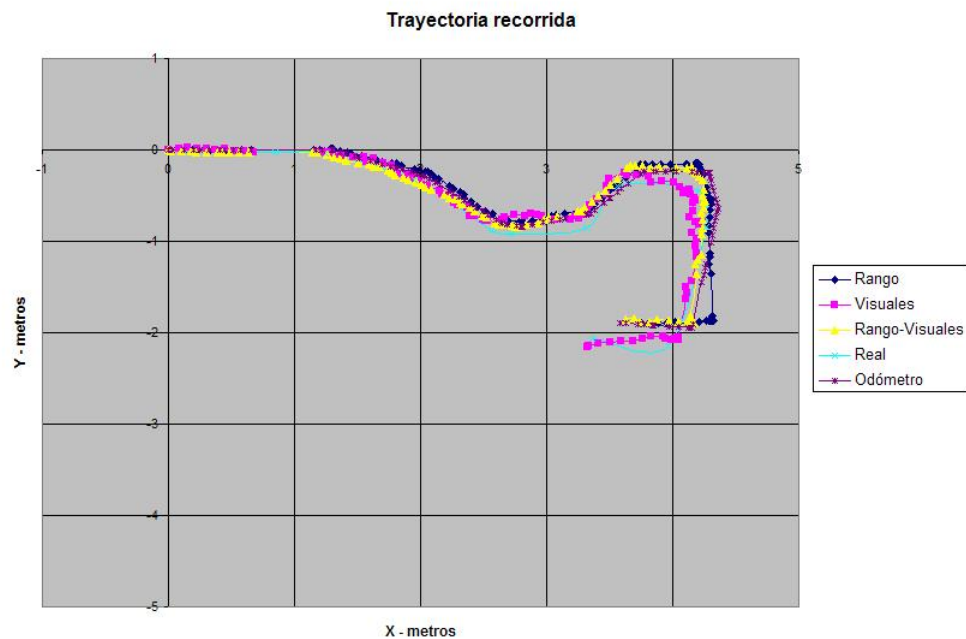


Figura 5.14: Trayectorias estimadas en el plano x, y del experimento 3. Para la trayectoria estimada en color azul marino sólo se utiliza marcas de rango, magenta sólo marcas visuales, amarillo ambas, azul celeste trayectoria real, púrpura trayectoria estima por el odómetro.

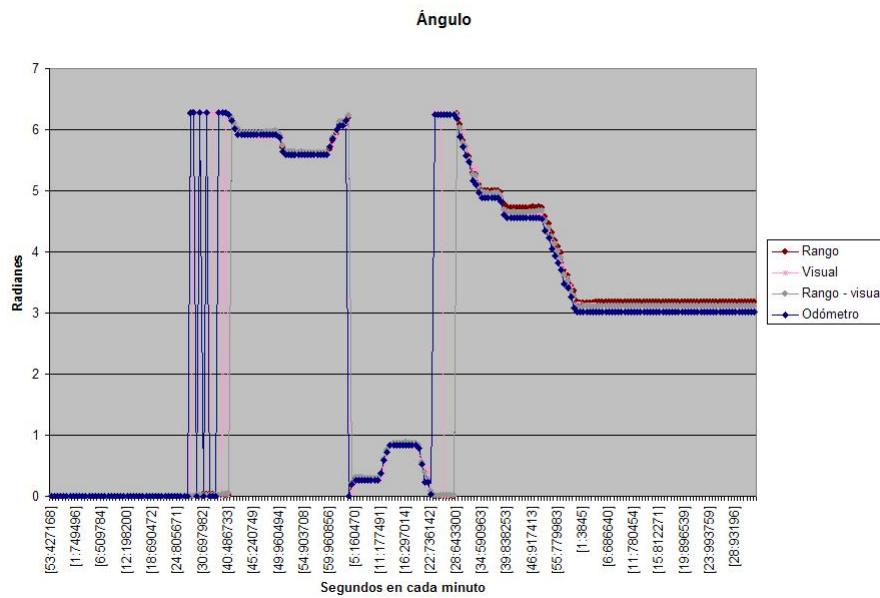


Figura 5.15: Orientación del robot durante el recorrido en el experimento 3. Ángulo expresado en radianes.

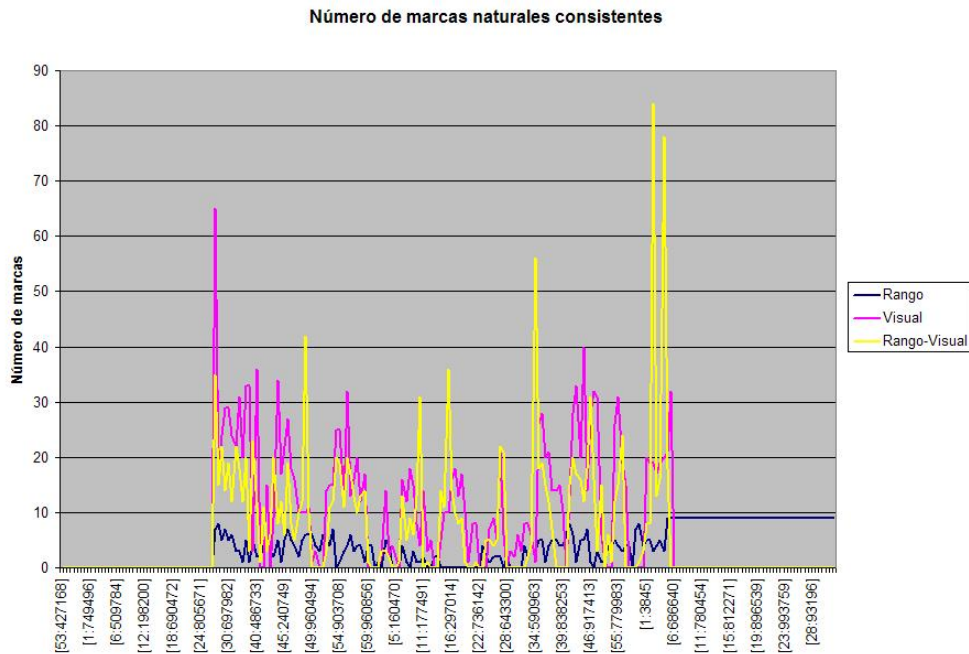


Figura 5.16: Marcas naturales consistentes para el experimento 3. Esto indica el número de marcas que se utilizaron en cada instante para estimar la posición del robot mediante triangulación. La gráfica no está sincronizada, es decir, son procesos independientes y las marcas utilizadas en el proceso que sólo considera marcas visuales no necesariamente son las mismas de el proceso que utiliza marcas de rango y visuales.

5.3.3. Experimento 4

Esta prueba se realizó en el ambiente 2. Es una prueba interesante, mientras el robot realizaba el recorrido por el ambiente hubo cambios considerables como personas transitando y movimiento de un póster frente al robot. El recorrido se hizo en los pasillos, cabe destacar que es un ambiente pobre en texturas, y al tratarse de pasillos se encontraron pocas marcas de rango.

Bajo estas condiciones el objetivo de este experimento es mostrar que en particular en pasillos, con frecuencia hay menos marcas de rango. El robot tiende a perderse si sólo se usan marcas de rango como referencia. Sin embargo, al combinar los tipos de marcas aumenta la robustez debido a que se complementan. No obstante, sigue existiendo la probabilidad de que el robot no encuentre marcas y se localice utilizando sólo el odómetro lo que implica que aumente la probabilidad

de una localización errónea debido a la imprecisión del mismo. En la figura 5.17, se muestran las trayectorias en el plano x, z y respecto al mapa en la figura 5.18 de acuerdo al tipo de marca utilizada. Notamos que utilizando sólo marcas de rango existe una desviación en la trayectoria respecto a las demás que se aproximan a la real, sin embargo, la forma de la trayectoria en general es muy parecida a las demás lo que deducimos que el error se produjo en la estimación de la orientación del robot. En la figura 5.19 corroboramos lo anterior mostrando que hubo un error acumulativo para el cálculo del ángulo del robot.

De la figura 5.20 obtenemos que en general el número de marcas de rango, casi en todo momento, está por debajo de 10, en promedio fue de 4.58 marcas, mientras que las marcas visuales en promedio fue de 7.3 y por último la estimación de la posición utilizando ambos tipos de marcas en promedio se tuvieron 14 marcas en cada instante. Esto es debido a que el recorrido se hizo en los pasillos, por consiguiente se encontraron pocas marcas de rango, además, de que en los pasillos se encontraron pocos objetos con textura. Recordemos que existen 3 procesos para estimar la posición del robot, uno utilizando sólo marcas de rango, otro marcas visuales y por último otro proceso utilizando ambos tipos de marcas. Cabe mencionar que no necesariamente en cada instante se tratan de las mismas marcas para todos los procesos.

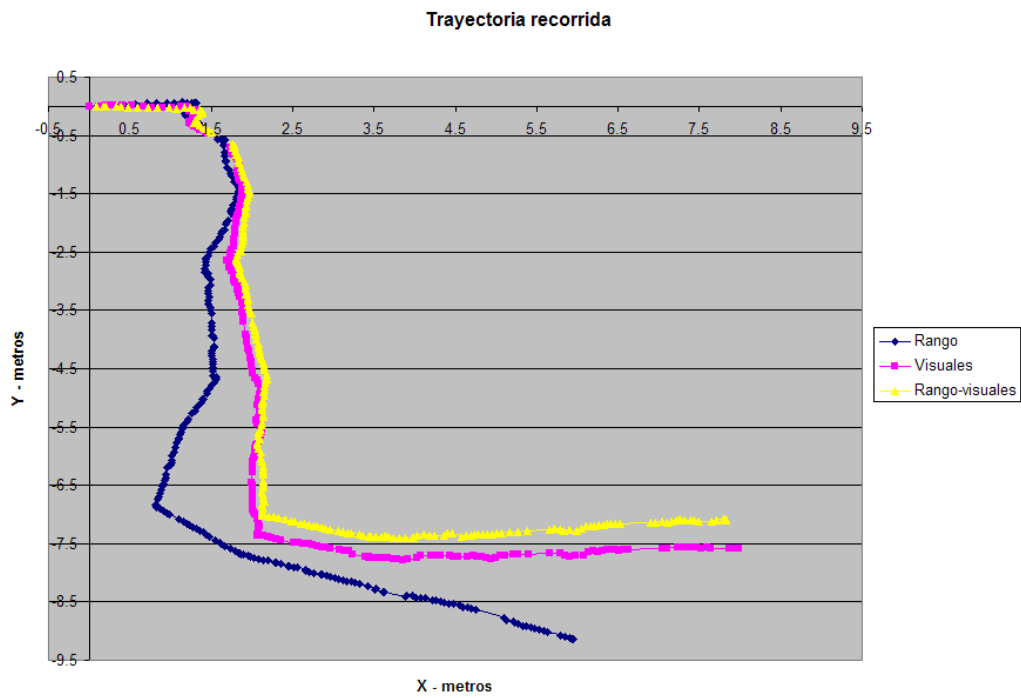


Figura 5.17: Trayectorias estimadas en el plano x, y en el experimento 4. Observamos que la trayectoria estimada usando solo marcas de rango (azul) tuvo un mayor desfase respecto a las demás que se acercan al real.

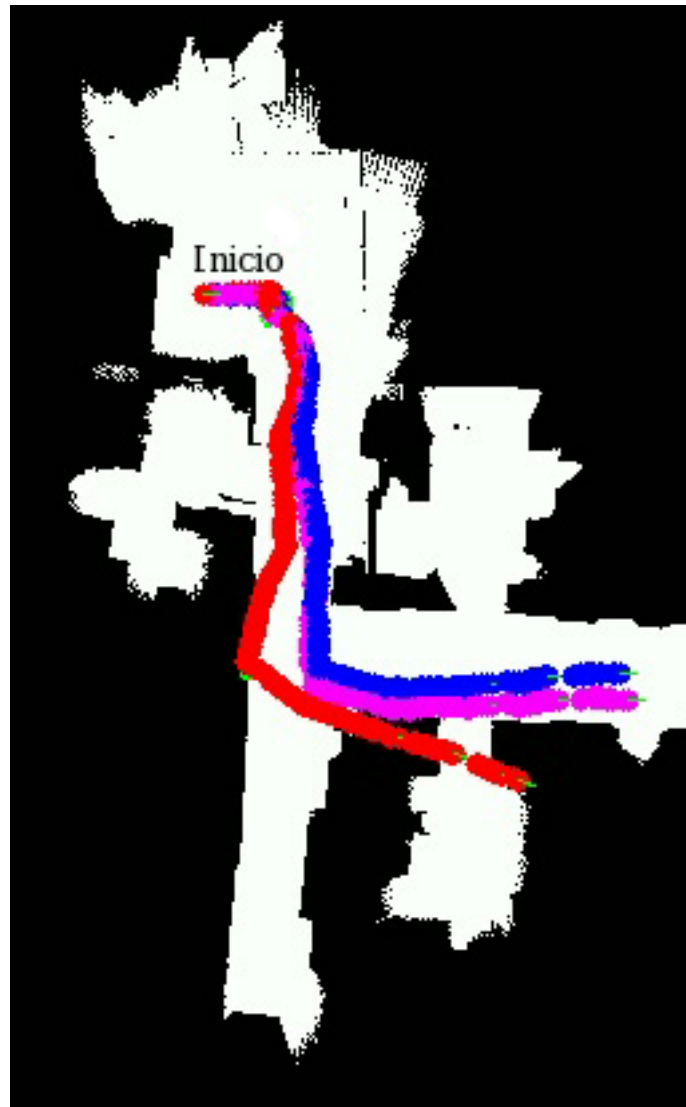


Figura 5.18: Trayectorias estimadas sobre el mapa del ambiente para el experimento 4. Color rojo, utilizando sólo marcas de rango; magenta, sólo marcas visuales y azul, ambos tipos de marcas. Utilizando sólo marcas de rango el robot se perdió.

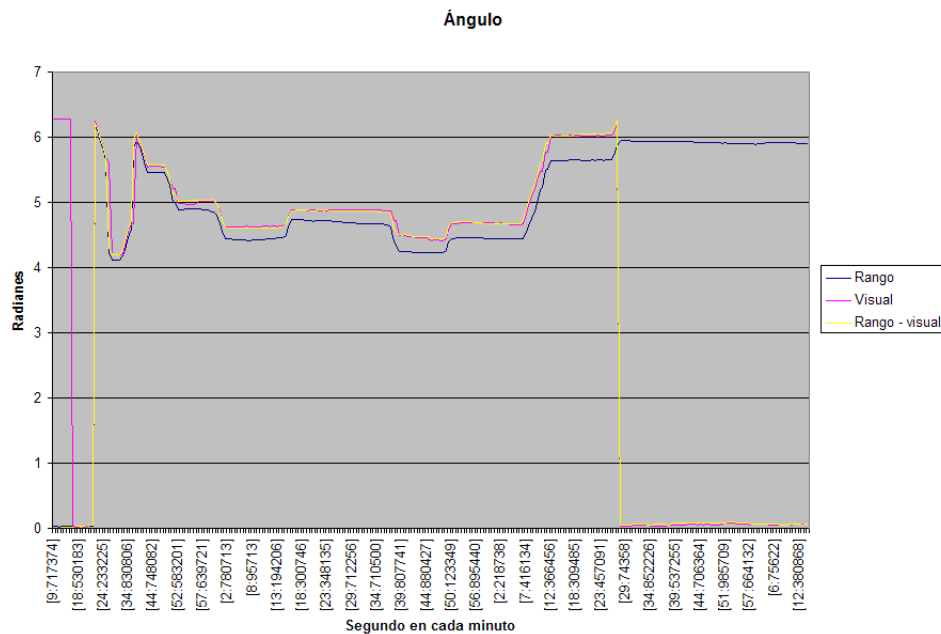


Figura 5.19: Valores de ángulos durante el recorrido en el experimento 4 según la estimación utilizando el tipo de marca.

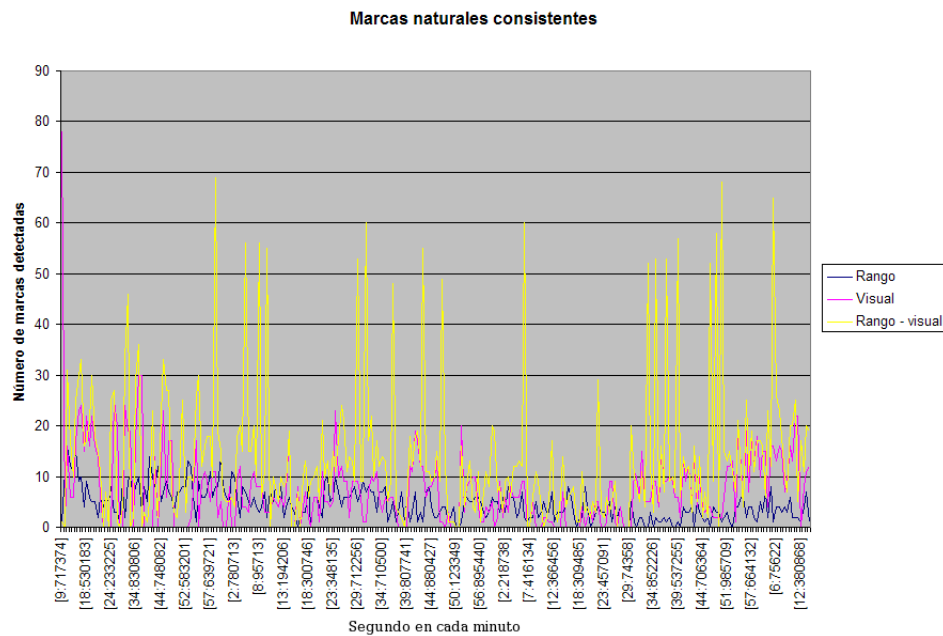


Figura 5.20: Número de marcas naturales consistentes en los diferentes casos de localización para el experimento 4. Usando sólo marcas de rango (azul), sólo visuales (magenta) y ambas (amarillas).

5.3.4. Resumen

Bajo las condiciones de experimentación mencionadas, mostramos que el algoritmo de seguimiento de la posición es robusto a cambios en el ambiente. Los tiempos para estimar la posición son aceptables, el combinar información de rango y visual resulta de gran utilidad ya que existe zonas donde no se detectan marcas de rango, pero si visuales o viceversa, entonces la probabilidad de quedarse sin referencia disminuye. Sin embargo, una desventaja del método propuesto es que no se recupera ante errores, el error es acumulativo. Se realizaron 25 experimentos de localización local, en los 25 recorridos, 19 trayectorias se acercaron más a la real combinando marcas de rango con visuales que sólo utilizar marcas de rango y visuales.

5.4. Localización global

Se realizaron dos tipos de experimentos para esta prueba, con cambios y sin cambios en el ambiente respecto al mapa inicial. Recordemos que primeramente se agrega información visual a los mapas construidos a base de sensores de rango, esto se hizo en dos tipos de iluminación, los resultados se muestran en la sección 5.2.2. Los cambios en el mapa básicamente fueron, cambios de posición de objetos, muebles, póster, etc.

5.4.1. Experimento 5

El experimento se hizo en el ambiente 4, en esta prueba no hubo cambios en el ambiente respecto al mapa inicial, en la figura 5.21 los círculos ilustran el proceso de triangulación. Se toman ternas de elementos de C , la intersección de esos círculos es una hipótesis de la ubicación del robot, el radio de los círculos corresponde a la distancia del robot a la marca detectada. Se elige la zona con mayor número de intersecciones, como solución final círculo azul, el error promedio fue de 10cm.

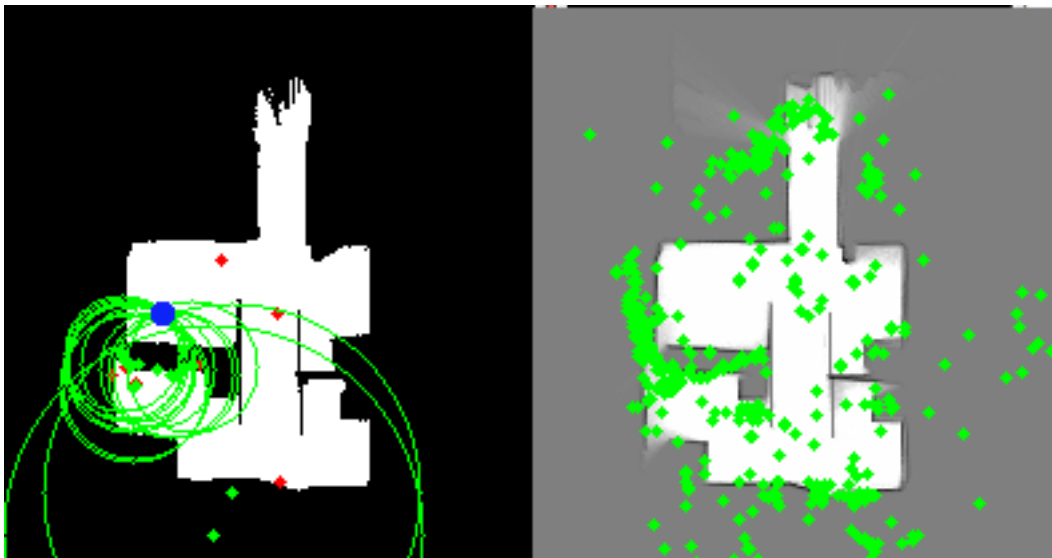


Figura 5.21: Resultado de aplicar el algoritmo de localización global. Imagen izquierda, puntos rojos y verdes representa el conjunto C de correspondencias entre el las marcas detectadas en esa posición y el conjunto de marcas del mapa; los verdes indican que son marcas consistentes, es decir, mantienen su estructura espacial entre ellas de acuerdo al algoritmo RANSAC representan el conjunto “inliers”; los rojos son correspondencias erróneas, conjunto “outliers”. Imagen derecha, mapa de marcas.

5.4.2. Experimento 6

Esta prueba se realizó en el ambiente 4 con cambios respecto al mapa inicial construido (figura 5.22 (c) imagen derecha). Los objetos que se cambiaron de posición fueron 3 revistas, su posición original se ilustra en la zona de color azul, y su nueva posición en la zona de color rosa. Las revistas las podemos ver en las imágenes (a) y (b). Ahora centremonos en la imagen (c), imagen izquierda, nuevamente los puntos rojos y verdes representan el conjunto C de correspondencias, los verdes indican el conjunto de “inliers” y los rojos el conjunto de “outliers” de acuerdo al algoritmo RANSAC [12]. Nótese que las marcas visuales enmarcas en rojo prácticamente coinciden con la zona rosa de la imagen derecha, es decir, las marcas visuales que se detectan en esa zona que corresponden a las revistas cambiadas de posición, no coinciden con el mapa de marcas original. El círculo azul indica la posición estimada del robot mediante triangulación, la localización fue correcta.

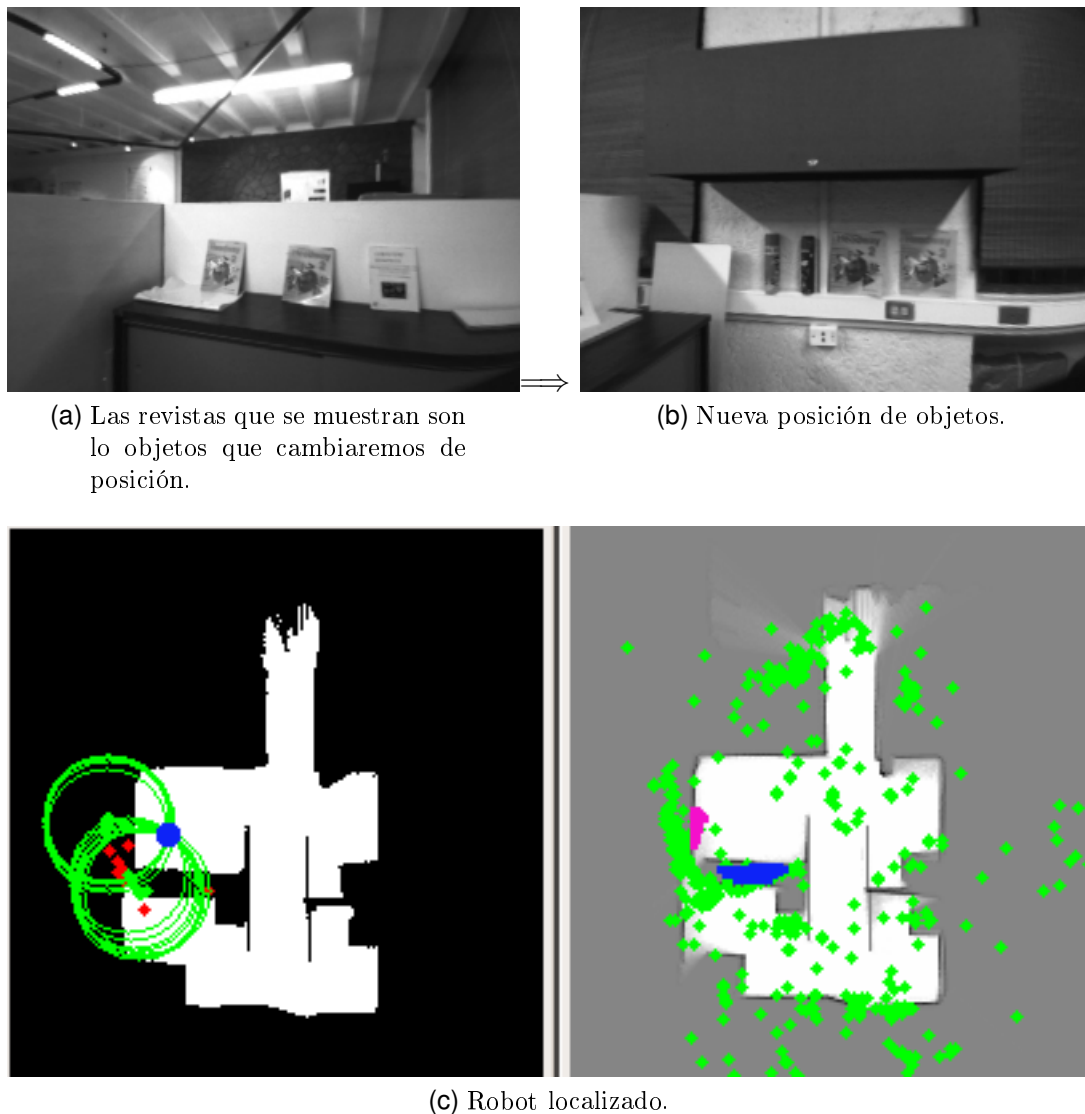


Figura 5.22: Localización global después haber cambios en el ambiente original.

El proceso de localización global dura como máximo aproximadamente 12 segundos que es el tiempo que se tarda en dar una vuelta entera sobre su propio eje. El algoritmo que se propone para la localización global es sencillo y rápido, pero tiene la desventaja de que si no se encuentran como mínimo una correspondencia de 30 elementos entre ambos conjuntos de marcas, se indica no localizado. Este dato se halló experimentalmente, por debajo de ese umbral la probabilidad de que se localice erróneamente es alta, además la probabilidad de fallar aumenta si existen fuertes cambios de iluminación, por ejemplo, que el mapa de marcas sea construido en la noche y se intente localizar en el día o viceversa. De los

15 experimentos de localización global con 5 en ambientes sin cambios y 10 con cambios, en 11 pruebas el robot se localizó satisfactoriamente 4 sin cambios y 7 con cambios. Otro dato encontrado de forma experimental es que algoritmo soporta aproximadamente hasta un 40 % de cambios en el ambiente, es decir, puede moverse el 40 % de las marcas visuales con respecto al mapa inicial.

5.5. Resumen

En este apartado se describió una serie de experimentos para evaluar los métodos propuestos. Se obtuvieron resultados prometedores, se superaron las expectativas debido a que en la localización local el método resultó ser robusto también para ambientes dinámicos. Inicialmente se tenía contemplado que a la hora de agregar información visual a los mapas construidos en [1] todo debía permanecer estático, sin embargo, se mostró que es tolerante a cambios. En cuanto al sistema “detector de marcas visuales” tiene un rendimiento aceptable en cuanto a la velocidad 0.134 segundos en promedio. Los mapas de marcas construidos en luz ambiente tienen mayor información comparado con los construidos a luz artificial, debido a que se detecta un mayor número de marcas visuales cuando se tiene una buena iluminación.

Capítulo 6

Conclusiones y trabajo futuro

La construcción de mapas y localización es una de las capacidades con las que debe contar el robot móvil para realizar tareas en donde tenga la necesidad de navegar por el ambiente. Estos problemas fueron abordados en los trabajos [2, 1] y [5, 7, 8] utilizando sólo un tipo de sensor, de rango y visual respectivamente. El objetivo que perseguimos fue tomar como base los trabajos [2, 1] y extender los algoritmos agregando información visual y de esta forma aumentar la robustez. Se introdujeron algoritmos para la localización local y global, integración de información y un sistema “detector de marcas visuales”. En los trabajos [5, 7, 8] utilizan un enfoque probabilista, construyen mapas 3D de marcas SIFT. Sin embargo, la diferencia sustancial con el enfoque propuesto en este trabajo de tesis es que se integra información de rango y visual. Además, mientras se construye el mapa de marcas puede haber cambios en el ambiente, después se soporta aproximadamente un 40 % de cambios de posición en las marcas respecto al mapa inicial. Mientras que en los trabajos mencionados esto no ocurre puesto que se asumen ambientes estáticos.

6.1. Conclusiones

Dentro del contexto probado y bajo condiciones mencionadas en los experimentos se alcanzaron los objetivos, los algoritmos propuestos son tolerantes a cambios en el ambiente, particularmente para la localización local es robusto también para ambientes dinámicos, en ese sentido se superaron las expectativas.

El método propuesto para agregar información visual de igual forma no necesariamente debe permanecer estático el ambiente. En general los algoritmos propuestos son robustos a cambios en el ambiente hasta un 40 % de cambios en el caso de la localización global. La idea fue utilizar los tipos de información de forma complementaria obteniendo así un sistema de localización tolerante a cambios que integra información visual y de rango para ambientes interiores. Los resultados obtenidos fueron satisfactorios no obstante sigue habiendo cuestiones que mejorar, por ejemplo, en el algoritmo de localización global tiene la desventaja de que si se cumple una rotación de 360° y no se encuentran correspondencias entre marcas detectadas y marcas del mapa el algoritmo falla. La solución puede ser, avanzar aleatoriamente y repetir el proceso. Agregando esto último y un método para actualizar el mapa cuando se detecten cambios en el ambiente se obtiene prácticamente un sistema de localización para ambientes dinámicos. Las principales contribuciones son:

- Un sistema “detector de marcas visuales” con un tiempo de procesamiento aceptable.
- Un algoritmo para enriquecer con información visual los mapas cosntruidos con los sensores de rango. Tolerante a cambios en el ambiente mientras se realiza este proceso.
- Un algoritmo de localización local utilizando marcas de rango y visuales, robusto para ambientes dinámicos.
- Un algoritmo de localización global tolerante a cambios en el ambiente respecto al mapa inicial.

6.2. Trabajo futuro

A continuación se enlista una serie de posibles mejoras:

1. Experimentalmente se detectó que al haber cambios importantes de iluminación al momento de enriquecer el mapa y realizar la localización global, la probabilidad de falla aumenta. Se propone implementar algoritmos de tratamiento de imágenes para disminuir esta cuestión.

2. En el método propuesto no se contempla la actualización del mapa una vez que se halla modificado el ambiente respecto al mapa inicial. En la sección del capítulo de experimentos se muestra que es posible detectar qué marcas se han movido y posiblemente qué objetos. Cabe mencionar que no todas las marcas del conjunto de “outliers” significa que se han movido puesto que puede tratarse también de un error de medición del sensor. Esta es otra cuestión, cómo actualizar los mapas en línea. Un primera idea es que sí las marcas pertenecientes al conjunto de marcas de “outliers” reinciden en su nueva posición un cierto numero de veces podemos actualizar la posición de dicha marca.
3. A pesar de que el método para la localización local es robusto para ambiente dinámicos, no lo es para la localización global, debido a que no se actualiza el mapa después de los cambios encontrados. Se propone extender el método para ambientes dinámicos.
4. Para hallar el conjunto consistente de relaciones entre conjuntos de marcas, se utiliza el método RANSAC para eliminar las relaciones erróneas. Sin embargo, sería interesante probar con el algoritmo *Graph Transformation Matching (GTM)* [16, 17] debido a que soporta un mayor porcentaje de “outliers” y es más rápido.
5. El método propuesto para la localización local es robusto para ambientes dinámicos, sin embargo, no se recupera ante errores. El error sigue siendo acumulativo, se propone extender el algoritmo para solucionar este problema. Un primer acercamiento es realizar un localización global periódicamente.

Referencias

- [1] Víctor Manuel Jáquez Leal. Construcción de mapas y localización simultánea con robots móviles. Tesis maestría, ITESM- Cuernavaca. 2005.
- [2] Sergio Francisco Hernández Alamilla. Navegación de un Robot Móvil en Ambientes Interiores Usando Marcas Naturales del Ambiente. Tesis de maestría, ITESM-Cuernavaca. 2005.
- [3] Kelly, A. Precision dilution in triangulation based mobile robot position estimation. Technical report, Carnegie Mellon University, Pittsburg, PA. 2003.
- [4] Lowe,D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* 60. 2004.
- [5] F.A. Moreno, J.L. Blanco, J. Gonzáles, V. Avéralo y J.A. Fernández-Madrigal. Un enfoque probabilístico para la estimación del desplazamiento de un robot móvil dotado de visión estéreo. XXVII Jornadas de automáticas. 2006.
- [6] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton & Olivier Stasse. MonoSLAM: Real-Time Single Camera SLAM. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 29, No. 6. 2007.
- [7] Stephen Se, David Lowe, Jim Little. Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features. In *Proceedings of the IEEE International Conference on Robotics and Automation. ICRA.* 2001.
- [8] Stephen Se, David Lowe, Jim Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics.* 2005.

-
- [9] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In AAAI National Conference on Artificial Intelligence, pages 593-598, 2002.
- [10] Kurt Konolige and David Beymer. SRI Small Vision System. Calibration Supplement to User's Manual Software version 4.4d May 2007
- [11] Kurt Konolige and David Beymer. SRI Small Vision System. User's Manual Software version 4.4d May 2007
- [12] Martin A. Fischler and Robert C. Bolles (June 1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM* 24: 381-395
- [13] Harris, C. and Stephens, M. A combined corner and edge detector. In 4th Alvey Vision Conference, Manchester, pp 147-151. 1998.
- [14] J S. Beis, D G. Lowe, Shape indexing using approximate nearest-neighbour search in high- dimensional spaces, *Proc. CVPR*, pp. 1000-1006 (1997).
- [15] J. L. Bentley, Multidimensional binary search trees used for associative searching, *Commun. ACM* 18, pp. 509-517 (1975).
- [16] W. Aguilar, Reconocimiento de objetos basado en la correspondencia estructural de características locales, Tesis de Maestría, UNAM (2006).
- [17] Yann Frauel, Curso de Visión Computacional, Posgrado en Ciencia e Ingeniería de la Computación, UNAM, México, <http://leibniz.iimas.unam.mx/~yann/CursoVision.html> (2008).
- [18] Edward Angel. *Interactive Computer Graphics*. Addison-wesley. 2000.
- [19] James Foley, Andries van Dam, Steven Feiner. *Introduction to computer graphics*. Addison-wesley. 1994.
- [20] Jianbo Shi and Carlo Tomasi. Good features to track. pp 593-600. 1994.
- [21] Herbert Bay, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features. *Proceedings of the ninth European Conference on Computer Vision*, May 2006.