UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"LOCALIZACIÓN DE UN ROBOT MÓVIL A TRAVÉS DE VISTAS CONOCIDAS "

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

PABLO FRANK BOLTON

DIRECTOR DE TESIS: DR. YANN FRAUEL

México, D.F. **2009.**

Resumen

La percepción visual – implementada a través de imágenes capturadas por una cámara CCD – es utilizada por un robot móvil para localizar su posición dentro de un ambiente conocido. El robot cuenta con una memoria *fotográfica* de algunas vistas del ambiente en cuestión, sin embargo, estas vistas constituyen una muestra de las posibles vistas que el robot pueda capturar durante un recorrido. Es entonces necesaria la implementación de varios algoritmos inteligentes que permitan al robot encontrar su posición relativa a las vistas conocidas.

Los problemas tratados en la resolución de la localización del robot van desde el procesamiento de las imágenes para su posterior comparación, hasta la implementación de rutinas de deducción de desplazamiento y resolución de ambigüedades de posición.

Se incorporan al modelo básico de localización módulos que ayudan a incrementar la precisión del resultado, la eficiencia del proceso general y el manejo de errores debidos al ruido en el sistema. Como aspectos novedosos en la integración de éstas técnicas, este trabajo incorpora los siguientes módulos:

1. Un método basado en grafos para la eliminación de correspondencias erróneas en la comparación de imágenes.

2. Un método de acumulación aplicado a la estimación tanto de posición espacial como de orientación.

3. La combinación, usando métodos de estimación probabilística, del método acumulativo con un método geométrico.

El problema de localización se trata bajo un escenario en el cual el robot se encarga de proporcionar a un visitante humano un recorrido informativo (una visita guiada) en un ambiente de laboratorio.

El sistema completo puede realizar recorridos exitosos dentro del ambiente conocido aún cuando cambie ligeramente el ambiente o se pierda durante el recorrido, ya que posee la capacidad de localizarse de nuevo y continuar con la ruta especificada.

Índice General

Índice de cuadros

Índice de figuras

Agradecimientos

Quiero agradecer a mi asesor de tesis, Yann Frauel, por guiarme y apoyarme en la investigación y elaboración de esta tesis. También deseo agradecer el apoyo (moral y técnico) de Wendy Aguilar y Montserrat Alvarado, que aportaron su propio trabajo a la implementación final del sistema.

Deseo agradecer a los doctores Maria Elena Martínez, Boris Escalante, Jesús Savage y Luis Enrique Sucar, por apoyarme con sus observaciones y comentarios y, en efecto, ayudarme a terminar esta tesis.

Por último, a mis amigos dentro y fuera del laboratorio que me apoyaron y me escucharon relatar mi trabajo una y otra vez.

Distrito Federal, México — Pablo Frank Bolton
Febrero 24, 2009

Capítulo 1

Introducción

La capacidad de percibir el ambiente es una de las características más importantes en un robot móvil, ya que de no hacerlo está sujeto a las instrucciones de un agente inteligente externo, o deambula sin noción de lo que sucede a su alrededor. Estos comportamientos podrían ser, de hecho, el objetivo del robot o elementos necesarios en su propósito final, sin embargo, en la mayoría de las aplicaciones modernas, poseer la capacidad de percibir el entorno brinda mayor información, y por lo tanto, mayor control sobre los procesos que se desean realizar.

Un escenario de experimentación comunmente usado en torno al tema de percepción visual artificial consiste en utilizar a un robot móvil como guía informativa durante un recorrido dentro de un ambiente conocido. En este escenario se pueden estudiar elementos de proceso motriz del robot, así como la inteligencia implementada para el procesamiento de las señales percibidas.

Los problemas principales dentro de este escenario son la navegación dentro del espacio seleccionado y el reconocimiento de elementos importantes del ambiente. La navegación – entendida como la serie de decisiones tomadas para realizar movimientos – depende del control que se tenga sobre el movimiento propio del robot (modulación

de actuadores, como lo serían ruedas y/o motores), y la interpretación de las señales percibidas a lo largo del trayecto. Así, se podría decir que la navegación depende en gran medida de la capacidad que tenga el robot de determinar su posición antes de cada movimiento. A este problema se le llama el problema de *localización* del robot móvil.

Existen dos grandes categorías dentro del problema de localización. Estos son el problema de *seguimiento de posición* (*position tracking*) y el problema de *localización global* (*global localization*). El primero parte del supuesto que el robot conoce su posición inicial, y debe realizar una estimación de posición tras cada movimiento (apoyado por la información de odometría de cada paso y las señales percibidas del entorno). El problema de localización global consiste en que el robot debe hallar su posición sin tener conocimiento alguno de su posición inicial. Sin importar cuál de los dos problemas de localización se intente resolver, tener la capacidad de percibir el entorno es una habilidad crítica que ha de poseer el robot móvil.

Una de las formas típicas de manejar la percepción de señales provenientes de la interacción del robot con el *mundo* ha sido el uso de mecanismos que ayuden a medir distancias desde el robot a su entorno. Dos de estas técnicas son el sonar y el láser. Estos dispositivos tienen la capacidad de identificar distancias mediante la medición del tiempo que tarda en viajar una señal de referencia que viaja desde el robot a su entorno y de regreso. En el caso del sonar, se envían ondas acústicas, y en el caso del laser, pulsos de luz. Estos dispositivos son muy útiles para aplicaciones básicas dónde sea necesaria información de distancias del entorno. Sin embargo, si el objetivo es más complejo o se desean implementar más aplicaciones aprovechando el mecanismo de percepción, éstas técnicas pueden resultar un tanto limitadas. Un

ejemplo de esto es intentar detectar objetos específicos durante la navegación. Aquí, el uso de imágenes permite al sistema especificar con mayor éxito las particularidades del objeto a ser reconocido, como lo sería la forma, los colores, la textura, y muchas otras características que bajo un esquema de detección por sonar, quedarían fuera del panorama.

Una alternativa de percepción del entorno es la llamada percepción visual del robot. Ésta consiste en la captación de imágenes digitalizadas a través de una cámara CCD, y su interpretación para obtener relaciones espaciales, reconocer objetos, movimiento, etc.

El problema de la imagen captada es saber *¿qué es qué?*, ya que la imagen es en realidad una matriz de valores discretos sin más significado que la interpretación que se le dé por parte de la inteligencia del robot. Una forma de obtener significado de las imágenes es a través de la comparación contra elementos conocidos, como lo serían vistas del ambiente o elementos individuales que puedan encontrarse en cualquier lugar dentro de éste. Estas vistas y objetos conocidos constituyen la *experiencia* o *memoria* visual del robot. El proceso de comparación en sí se detalla más adelante, pero se basa en la correlación entre características significativas de las imágenes. La comparación de imágenes es mejorada al incorporar un mecanismo de eliminación de correspondencias erróneas. Es con base en esta comparación que se logran definir relaciones de parecido y posición entre las vistas conocidas y las imágenes que se van obteniendo a lo largo del recorrido del robot. Se incorpora un módulo de estimación de pose por métodos acumulativos que permite aproximar la posición y orientación del robot. Esta aproximación se combina con una segunda estimación de localización a partir de un método geométrico de extrapolación de orientación. Éstas relaciones de

posición nos suministran posibles posiciones actuales del robot y por lo tanto posibles secuencias de éstas, lo que constituirían distintas rutas. Un análisis probabilístico de las secuencias de posiciones probables y las vistas del robot nos permite identificar la ruta más probable y por lo tanto definir la siguiente acción a realizar.

Objetivo: La meta final consiste en construir un sistema modular de localización visual para un robot móvil con cámara monocular que logre entregar un buen estimado comparativo de la posición y orientación. Éste objetivo se debe de alcanzar tanto para el problema de localización global como para el de rastreo de posición.

Resumen del sistema: El método seguido para alcanzar el objetivo mencionado consiste en utilizar imágenes de referencia contra las cuales será comparada la vista a localizar. Las vistas de referencia similares serán usadas para generar una estimación de posición por un método de acumulación. La orientación se estima tanto por acumulación como por un método geométrico que se basa en la relación proyectiva de las características de la vista a localizar con las de referencia. Para el caso en que se conozca la posición inicial del robot o una trayectoria seguida, la estimación visual se combina con la estimación de odometría para así generar una estimación de localización conjunta. Al final del proceso se cuenta con una estimación de posición y orientación del robot, sea ésta para la localización global o durante el proceso de seguimiento de posición.

Organización de la tesis: En el **Capítulo 2** se presenta el trabajo relacionado con esta línea de investigación. En el **Capítulo 3** se presenta el problema a tratar y la base de datos de imágenes. En el **Capítulo 4** se desarrolla el concepto de obtención de características esenciales y comparación entre imágenes. En el **Capítulo 5** se explica el método de acumulación por umbral de calidad para la obtención de

una posición aproximada. El **Capítulo 6** explica la restricción epipolar, utilizada para encontrar las posiciones relativas entre imágenes. En el **Capítulo 7** se explican los fundamentos probabilísticos y la técnica específica de localización basada en el filtro de Kalman, con la cuál se restringen las posibles posiciones del robot hasta lograr ubicarlo precisamente en el lugar correcto. En el **Capítulo 8** se presenta la técnica propuesta para la solución del problema de localización. En el **Capítulo 9** se presentan las pruebas realizadas por cada módulo y los resultados obtenidos. Finalmente, en el **Capítulo 10** se presentan las conclusiones.

Capítulo 2

Trabajo relacionado

Dados los avances en capacidad computacional que se han dado desde hace ya varios años, se ha vuelto posible el procesamiento de grandes cantidades de información en tiempos relativamente pequeños. Una de las aplicaciones que han sacado provecho de estos avances es el procesamiento digital de imágenes, siendo un uso de este procesamiento la comparación de imágenes y sus aplicaciones. Algunas de estas aplicaciones son la localización, el seguimiento y la identificación de objetos.

Existen trabajos en cuyo escenario de localización no se conoce el ambiente, y están enfocados a localizarse dentro del espacio, así como crear el mapa del ambiente [13, 17]. A diferencia de los trabajos mencionados, el presentee trabajo está enfocado a la localización de un robot móvil aprovechando la comparación de imágenes capturadas por éste, con una base de datos de imágenes de un ambiente conocido. La comparación de imágenes no puede hacerse realistamente píxel a píxel, ya que la carga computacional sería inmensa. Es por esto que se ha cambiado el enfoque y la escala de comparación de tal forma que se identifiquen elementos esenciales dentro de las imágenes, y sea mediante la relación entre estos que se defina aquella entre las imágenes. Se han propuesto varios métodos para encontrar correspondencias entre

estos elementos específicos de las imágenes. Uno de los acercamientos más socorridos es el de identificar puntos críticos en las imágenes y encontrar las correspondencias entre estos. Los primeros métodos en hacer esto se basaron en la detección de esquinas como puntos críticos [20]. Después, estas técnicas se fueron refinando en torno a la necesidad de que estos identificadores fueran invariantes a cambios en escala, rotación e iluminación, así como la presencia de ruido. Uno de los algoritmos más robustos en torno a estas necesidades es la transformada de características invariante a escala (cuyo nombre original es "Scale Invariant Feature Transform" ó SIFT) [32]. En el trabajo de Lowe, se describe cómo encontrar numerosos puntos críticos a comparar entre dos imágenes, y así encontrar las correspondencias entre ellas usando el algoritmo de mejor conjunto primero o BBF (Best-Bin-First)[6]. En el mismo trabajo se recomienda incluir, como etapa siguiente, un algoritmo para eliminar las correspondencias erróneas. Por otro lado, un algoritmo de este tipo ha sido desarrollado por Aguilar et al. [1] bajo el nombre Correspondencias por Transformada de Grafos o GTM (Graph Transformation Matching), lo cuál permitiría contar con una relación entre imágenes basado en correspondencias correctas entre sus puntos críticos. Un método alternativo para la eliminación de correspondencias erróneas es el mecanismo de consenso de muestras aleatorias (cuyo nombre original es "Random sample consensus" ó RANSAC) [15]. RANSAC localiza más correspondencias, pero no elimina tantas correspondencias erróneas como GTM [2].

El siguiente paso en la localización de un robot móvil, basado en comparación, sería relacionar la vista actual con aquella(s) conocida(s) a la(s) que más se parezca y encontrar su posición relativa. Para este paso es recomendable contar con varias vistas (visión estéreo o de tres vistas) para así poder precisar mejor la posición del

robot. Una técnica para lograr esto con tres vistas se presenta en [42], donde se trabaja con tres imágenes de la misma escena (sistema Triclops de visión estéreo). Uno de los retos en el presente trabajo es lograr localizar el robot basado en una vista capturada con una cámara monocular (una sola imagen por vista). La localización de un robot basado en una sola cámara se presenta en [8], donde se apoya la detección de características con medidas de distancia con un láser. Bajo la metodología elegida para el presente trabajo, se comparará la vista desconocida con una base de datos de imágenes calibradas tomadas en el departamento de Ciencias del la Computación del IIMAS [45]. Existen varios trabajos que utilizan imágenes de referencia para determinar la posición del robot. En [37] se hace la localización de un robot en un pasillo usando, como referencia, una serie de imágenes omnidireccionales. En [50] se usan imágenes de referencia y el método de Ray casting [41] para aproximar la posición del robot.

En trabajos anteriores [3, 43, 27, 5] se ha demostrado que el desplazamiento entre dos imágenes puede ser calculado por medio de la geometría epipolar, donde se determinó que la precisión de esta estimación depende del ambiente y el extractor de características elegido [19]. Una alternativa de posicionamiento consiste en encontrar un punto representativo (un centroide) de las poses de las vistas que se consideren "cercanas" a la vista actual.

La estimación de posición del robot se irá refinando conforme éste avance en el ambiente y vaya obteniendo nuevas vistas. Esto se puede hacer basado en modelos probabilísticos de la posición del robot, que a su vez se apoyan en las comparaciones de vistas y la odometría del trayecto. Existen varias metodologías para modelar el

movimiento basado en odometría, así como su incorporación con un modelo de percepción visual [10, 24].

Otro reto de este trabajo es lograr realizar la localización del robot en tiempo real. Para lograrlo, es necesario hacer que cada módulo y su interconexión sea lo más eficiente y rápido posible. A continuación se discutirán los módulos de operación y las técnicas usadas para aumentar su eficiencia.

Capítulo 3

Autolocalización de un robot móvil

En este capítulo se plantea el problema de localización, se presentan algunos antecedentes del problema y se explica el escenario de investigación que se diseñó para poner en prueba las capacidades de localización de un robot móvil. Dentro de éste se encuentra el contexto del experimento, los elementos que interactúan dentro de él, y la serie de pasos para resolver el problema.

3.1. Planteamiento del Problema

El problema consiste en lograr que un robot utilice información visual para identificar su posición dentro de un ambiente conocido. Para lograr esto se cuenta con una base de datos de información que reúne la experiencia visual del robot. La base de datos de imágenes (o BDimag) sirve como la memoria de referencia del robot, ya que así consta con una serie de vistas con ubicaciones conocidas (coordenadas de posición y orientación definidas). Se define la *pose* del robot como la coordenada conjunta de posición y orientación. El robot debe de poder comparar lo que está viendo en el momento de navegar (la *vista actual*) contra su memoria, y decir a cuáles de

las imágenes que tiene en la BDimag se le parecen más. Es importante hacer notar que la similitud entre dos imágenes puede deberse a que los puntos de vista que las originaron fueron tomados desde posiciones cercanas. Esto se traduce a que las coordenadas relacionadas con cada imagen candidato son posibles poses cercanas al estado real del robot.

En este punto se extrae la pose real a partir de poses relativas usando dos métodos, para ser combinados posteriormente: El primer método, basado en agrupación de puntos en cúmulos, se basa en el hecho de que se cuenta con una serie de poses posibles en las que pueda estar el robot. Estos puntos estarán ordenados en cúmulos, de los cuales se puede obtener un centroide. En caso que se tenga un sólo cúmulo, su centroide se toma como la pose real del robot. De haber más de uno, se altera ligeramente la posición (por ejemplo un giro de 30°), se captura una imagen nueva y se repite el ciclo (comparando ahora contra los candidatos de la etapa anterior). Esta lógica aplica tanto a la posición como a la orientación del robot y se detalla en el capítulo 5. En el segundo método, se aprovechan las restricciones de la geometría epipolar aplicadas al par de imágenes parecidas (la vista actual y cada candidato). Esto se explica a detalle en el capítulo 6.

Al final, las estimaciones de cada método se combinan para generar una estimación visual final. Ésta, a su ves podrá combinarse con el modelo tradicional de seguimiento por odometría (en caso de conocerse la posición inicial) y así precisar la posición del robot durante una serie de movimientos.

El proceso completo puede verse en la figura 3.1. Los elementos mostrados en la figura se explicarán en los siguientes capítulos.

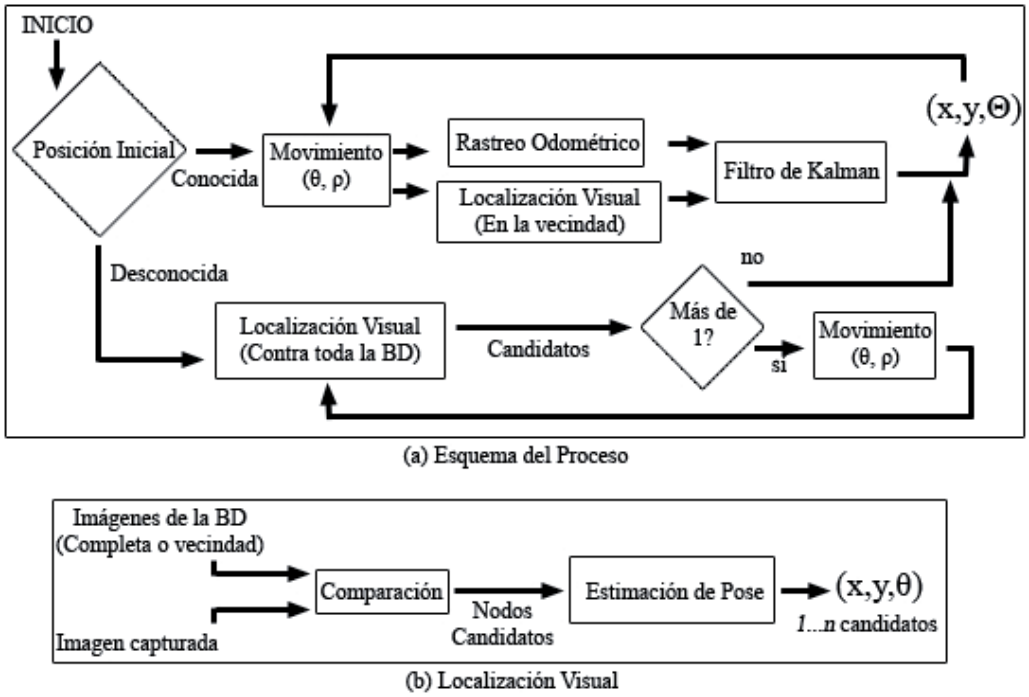

Figura 3.1: Esquema del proceso de localización. En (a) se observa el proceso general donde se puede seguir uno de dos caminos: el rastreo de posición (en caso de conocerse la posición inicial) o la localización global (si no se conoce la posición inicial). En el primer caso se realiza un movimiento seguido de un par de estimaciones que han de combinarse para generar la estimación final (capítulo 7). En (b) se observa el detalle del proceso de localización visual, el cual se basa en una fase preliminar de comparación (capítulo 4) seguida de una de estimación de pose (capítulos 5 y 6).

3.2. La base de datos de imágenes

El contexto temático de localización en el cual un robot móvil proporciona una visita guiada dentro de un ambiente conocido permite incluir dentro de la temática del escenario la necesidad intrínseca de localización y seguimiento de trayectorias. Asimismo, se tiene la opción natural de utilizar los objetos de demostración (aquello que va mostrando el robot, como lo serían pinturas en una galería, o posters en un

laboratorio) como indicadores de posición (también llamados *landmarks*). En este trabajo, no se utilizan *landmarks*, sino que se cuenta con una base de datos de imágenes del espacio en el que el robot dará la visita.

Para este escenario de localización se quiere que el robot móvil cuente con una especie de *experiencia previa* del espacio en el que habrá de ubicarse. Esta experiencia previa toma la forma de *memoria visual*, implementada a través de una base de datos de imágenes del espacio en cuestión. La base de datos de imágenes se compone de vistas tomadas desde diferentes posiciones, o nodos definidos *a priori* dentro de un espacio definido. Desde cada nodo se toman ocho imágenes (una cada 45°) para así cubrir todo el espacio desde ese punto de vista. Es importante mencionar que como estos nodos serán usados como posiciones de referencia, se registra, junto con cada imagen, la posición y orientación en la cual se encontraba el robot al capturar cada vista.

Capítulo 4

Comparación de imágenes

En este capítulo se explica la importancia que tiene la habilidad de comparar imágenes así como el proceso para identificar puntos críticos y, con base en ellos, efectuar la comparación.

4.1. Introducción

En este trabajo se intenta interpretar y usar información a partir de señales luminosas captadas por una cámara. Este escenario se apega a la teoría de David Marr, el cual afirma que "Visión es un proceso que produce, a partir de imágenes del mundo externo, una descripción que es útil para el observador y que está libre de información irrelevante" [35]. Este acercamiento a la visión como un procesamiento de señales es medular en la línea de investigación de teoría de cognición llamada "neurociencia computacional", donde se conjuntan varias disciplinas para intentar describir funcionalmente la forma en que el cerebro humano (en particular las neuronas) logran captar, almacenar, usar y compartir información.

Parte del procesamiento realizado depende de la función que se le dará a la información ya que aquello eliminado o resaltado depende de la aplicación. A lo que se refiere Marr como *útil* e *irrelevante* depende del objetivo. En lo que se refiere a este escenario de localización a través de visión computacional, aquello que es útil es lo que nos permite comparar una nueva vista con la experiencia visual del robot. Entonces, el siguiente paso en este procesamiento de información es la comparación de imágenes y la obtención de una medida de su parecido .

4.2. Detección de características

Las técnicas para comparar imágenes difieren en el tipo de los elementos usados para su caracterización. Estos elementos son llamados *características principales*.

Hay dos tipos básicos de características de una imagen: características *globales* y características *locales*. Las características *globales* constituyen una representación de la imagen completa, mientras que las *locales* son partes o regiones importantes de la imagen. En este trabajo se aprovechan las similitudes y diferencias entre partes específicas de cada imagen, por lo que es preferible la obtención de características locales sobre las globales.

Algunas propiedades deseables en las características locales de una imagen son las siguientes:

1. **Localizadas** : Que las características estén relacionadas con un área específica y definida de la imagen, como lo son puntos, líneas y áreas conexas.

2. **Significativas** : Que las características representen partes llamativas o importantes de la escena, como lo serían regiones distintivas y bordes de la imagen.

3. **Robustas** : Que las características sean detectables bajo diferentes condiciones de la misma escena, como lo serían rotaciones de la imagen y cambios de punto de vista y de iluminación.

Algunas técnicas de detección de características locales se describen a continuación.

4.2.1. Detección de bordes [18]

Un tipo de detección de características muy usado es la detección de bordes, los cuales son importantes ya que suelen marcar los puntos de separación de objetos y regiones. Se entiende como borde a una serie de puntos que se encuentran en el umbral de dos áreas con un cambio drástico de intensidad. Esta región cambio puede verse como un escalón de pendiente elevada (de subida o bajada).

Dos técnicas para evaluar este cambio son la obtención de los puntos críticos de la primera derivada, y los llamados "cruces por cero" al utilizar la segunda derivada. Al primer caso se le denomina la obtención del máximo *gradiente*, que corresponde a la dirección de la máxima razón de cambio de intensidad de un campo escalar. Para imágenes, el campo escalar corresponde a la matriz de valores que representa a la imagen. La segunda técnica es denominada la obtención del *Laplaciano* del campo escalar.

Ambas técnicas se pueden implementar por medio de filtros matriciales aplicados a las imágenes. Un par de estos filtros son la serie de filtros de Prewitt y de Sobel.

Es importante mencionar la gran sensibilidad al ruido que presentan estas detecciones. Es por esto que se suelen realizar en conjunción con el filtrado Gaussiano de la imagen. El filtrado Gaussiano de una imagen produce una versión *borrosa* de

ésta, lo que ayuda a eliminar los cambios de alta frecuencia, muchos de los cuales son ruido. De esta conjunción de filtrados se obtienen las técnicas de *LoG* (Laplaciano de una Gaussiana) y de *DoG* (Diferencia de Gaussianas), siendo la segunda una muy buena aproximación de la primera. Los cruces por cero de una *LoG* permite encontrar bordes, mientras que los extremos de ésta ayudan a identificar manchas.

Uno de los mejores detectores es el llamado detector de bordes de Canny. En éste se incorporan la supresión de ruido por filtrado Gaussiano y la detección de bordes con filtros de Sobel. En esta técnica se añade un proceso de refinamiento de bordes, en el cual se adelgazan las regiones detectadas y se conectan con otras que representen el mismo borde. Una serie de extracciones de bordes se muestra en la figura 4.1

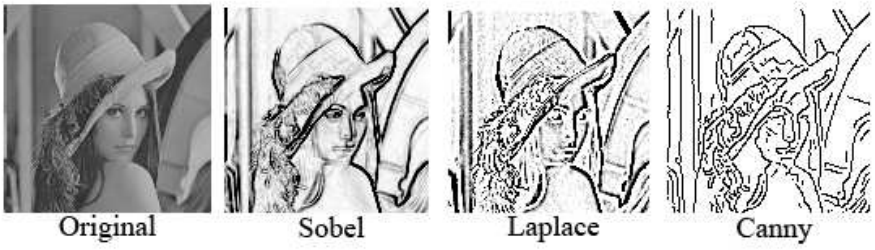

Figura 4.1: Extractores de bordes. En las imágenes se puede ver el efecto de aplicar cada uno de los extractores de bordes a la imagen original de Lena.

4.2.2. Detección de líneas y figuras geométricas

Otro método de detección de características consiste en buscar figuras geométricas específicas dentro de cúmulos de puntos en la imagen. Un ejemplo de esta técnica es la transformada de Hough [14]. La idea de esta técnica es definir la parametrización de una figura buscada y, mediante un sistema de votación de puntos (pixeles) participantes, localizar aquellas zonas de la imagen donde más elementos pertenecen a la misma figura. Las formas típicas a detectar bajo esta técnica son líneas y círculos,

como se puede ver en la figura 4.2, basada en el trabajo de Duda y Hart[14] donde se puede observar la detección de las líneas de la imagen de un cubo.

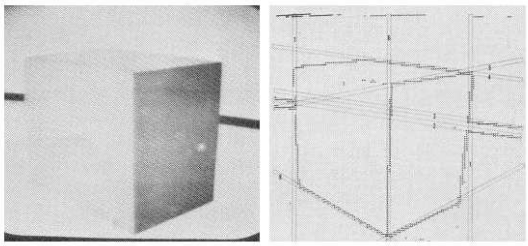

Figura 4.2: Extractor de líneas de Hough, Obtenido de [14]

4.2.3. Detección de esquinas

Los detectores de esquinas se enfocan en resaltar aquellos puntos que tienen cambios significativos de intensidad en todas direcciones, por lo que se podría decir que identifica *puntos* críticos. El detector de esquinas de Harris [20] hace esto al medir los cambios de intensidad al deslizar una ventana sobre la imagen. Los cambios son drásticos alrededor de puntos de interés. Un ejemplo se puede ver en la figura 4.3

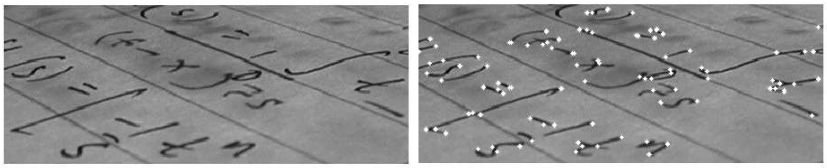

Figura 4.3: Detector de esquinas de Harris, Obtenido de [47]

Como se mencionó al principio de esta sección es importante tomar en cuenta que estos puntos se encuentran por las características de la escena, así que un cambio en punto de vista o la rotación de la imagen debería de generar los mismos puntos o

puntos muy similares. Pese a que son bastante robustos, más adelante se explicará que estos detectores son muy sensibles a los cambios de iluminación.

4.2.4. Teoría espacio-escala [49, 30]

Como ya se mencionó anteriormente, la aplicación de un filtro Gaussiano sirve el propósito de *borronear* la imagen, removiendo así los cambios drásticos de intensidad. Esto se debe a que el filtro Gaussiano causa que se transforme el valor de intensidad de cada pixel al mezclarlo con el de los pixeles cercanos en un radio definido. Esta mezcla se realiza tomando el promedio ponderado de valores de intensidad, donde esta ponderación sigue una distribución normal centrada en el pixel al cual se le está realizando la transformación. En términos técnicos, este proceso se realiza obteniendo la convolución entre la imagen y una matriz Gaussiana, la cual se construye con el núcleo de la ecuación 4.2.1. La operación de convolución que regresa la familia de imágenes L definidas por la escala t se muestra en la ecuación 4.2.2, donde $f(x, y)$ representa la imagen original y la escala se define como $t = \sigma^2$, que forma parte del exponente en la ecuación 4.2.1 y se llama la *varianza* de la distribución.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{4.2.1}$$

$$L(x, y; t) = G(x, y; t) * f(x, y) \tag{4.2.2}$$

El resultado de aplicar esta transformación a la imagen cambia la escala a la que se encuentran los cambios más drásticos de intensidad (frecuencias altas). Mientras mayor sea el tamaño del núcleo del filtro Gaussiano, mayor será el emborronamiento y

menores serán las frecuencias máximas de la imagen resultante. El tamaño del núcleo se define por la escala t, y lo que provoca es que ya no se detecten detalles de tamaño menor a $\sqrt{t}$. Este paso puede repetirse varias veces para generar una serie de imágenes a diferentes escalas. A este proceso se le llama la generación del *espacio-escala*.

Otra forma de entender la familia de funciones L es verla como la solución a la ecuación de difusión de calor mostrada en la ecuación 4.2.3.

$$\frac{\partial L}{\partial t} = \frac{1}{2}\left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}\right) \tag{4.2.3}$$

Bajo esta interpretación, las intensidades luminosas de la imagen original se piensan como una distribución de temperatura, y las imágenes L son el resultado de la difusión de temperatura de esta distribución inicial a lo largo del tiempo.

El efecto de una serie de filtrados gaussianos se muestra en la figura 4.4.

Figura 4.4: Efecto de filtrado gaussiano a diferentes escalas. Obtenida de [48]

El núcleo usado es el Gaussiano ya que otros filtros paso-bajas (que eliminan frecuencias altas), como el filtro promediador, pueden provocar la aparición de estructuras nuevas en la imagen. Esto sucede ya que se añade información de los pixeles circundantes al pixel transformado. El núcleo Gaussiano, en cambio, constituye una base canónica para la generación lineal de un espacio-escala y es por tanto el mejor

para este tipo de aplicaciones [26, 30].

La utilidad de este espacio-escala radica en el hecho de que a diferentes escalas los objetos presentan diferentes formas o estructuras características, las cuales pueden servir para su reconocimiento bajo distintos ambientes. Las técnicas de detección de características descritas anteriormente trabajan todas en una sola escala. Gracias a la obtención del espacio-escala de una imagen, se pueden detectar características importantes a diferentes niveles de detalle. Una técnica de detección de características que aprovecha este desarrollo se explica en la sección 4.4.

Esta idea se puede extender para obtener una *pirámide de imágenes*. En una pirámide de imágenes, se acomoda una pila de imágenes de resolución decreciente, siendo la base aquella con la máxima resolución y la punta aquella con la menor resolución. La estructura es piramidal ya que después de cada etapa de suavizado, la imagen se submuestrea para obtener el siguiente nivel. En el caso de una pirámide Gaussiana, el suavizado se realiza con el núcleo Gaussiano mostrado en la ecuación 4.2.1. La ecuación 4.2.4 muestra la forma de obtener el nivel $j-1$ (menor resolución) a partir del nivel j (mayor resolución). El término $S^{\downarrow}$ se refiere al submuestreo, G_t representa el núcleo Gaussiano a la escala t, y la $I = f(x, y)$, que es la imagen original.

$$P_{Gaussiana}(I)_{j-1} = S^{\downarrow}\left(G_t \otimes P_{Gaussiana}(I)_j\right) \tag{4.2.4}$$

Una representación gráfica de una pirámide Gaussiana se muestra en la figura 4.5.

Se puede decir que un nivel de detalle grueso (menor resolución) *aproxima* al siguiente nivel (mayor resolución) ya que al añadir un nivel de *detalle* se pasa de uno al otro. Así, si se aprovecha un operador de sobremuestreo $S^{\uparrow}$, se puede hacer que un nivel de detalle grueso quede del tamaño del nivel anterior, siendo entonces la

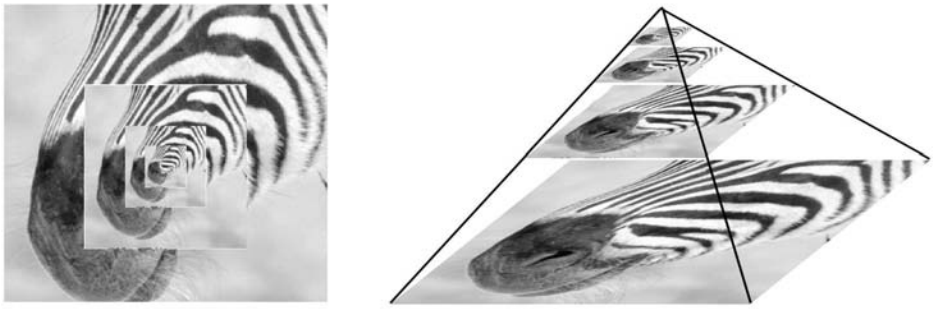

Figura 4.5: Imágenes escaladas y pirámide Gaussiana

diferencia entre ambos el nivel de detalle que caracteriza el salto de escala. A una pirámide que obtiene esta serie de diferencias se le llama una *pirámide Laplaciana*. La ecuación 4.2.5 muestra la forma de obtener el nivel $j-1$ a partir del nivel j.

$$P_{Laplaciana}(I)_{j-1} = P_{Gaussiana}(I)_j - S^{\uparrow}\left(P_{Gaussiana}(I)_{j-1}\right) \tag{4.2.5}$$

4.3. Descriptores

La detección de características resalta aquellos elementos importantes de la imagen, sin embargo, ¿cómo es que estos elementos se distinguen unos de otros? Se debe de poder caracterizar estos puntos de interés de tal forma que su descripción sea invariante a la escala de la imagen o a una rotación de la misma. Es importante que resista cambios en la iluminación de la escena, aunque esto último, como se explicará más adelante, es una cuestión que resulta un tanto más difícil. Esta caracterización de puntos de interés permite realizar comparaciones entre estos y así encontrar similitudes entre las imágenes que los contienen. Estas descripciones se construyen en forma de un vector que caracteriza los puntos importantes de la imagen. A este vector se le

suele llamar el *descriptor* del punto.

En su trabajo, Mikolajczyk y Schmid [38] evalúan una serie de descriptores locales. A continuación se mencionan algunas de estas técnicas.

Hay varias formas de construir este descriptor. La forma más simple es guardar los valores de los pixeles que rodean al punto dentro de un área o ventana definida. A esta técnica se le llama *parche de imagen*. Esta técnica resulta muy costosa en términos del tamaño de descriptor contra la precisión obtenida ya que es muy sensible a cambios de iluminación y punto de vista.

Otra forma es obtener el histograma de luminosidad de la ventana dentro de la que se encuentra el punto de interés. Esto ayuda en términos de tamaño del descriptor e invarianza a rotación ya que el histograma no registra posiciones específicas, sólo distribucion de iluminaciones. El problema es que sigue siendo muy sensible a cambios de iluminación y es bastante general (no permite una clara distinción entre puntos).

Otro método consiste en determinar, dentro de una ventana de la imagen, los *momentos generalizados* [38], que es un análisis estadístico de las relaciones de luminosidad de la imagen. Pese que es fácil de calcular, para que sea suficientemente preciso se necesita un tamaño grande de descriptor y debido a su naturaleza mantiene una alta sensibilidad a cambios de iluminación.

Un buen descriptor basado en la forma de los objetos es el llamado *contexto de forma* [7]. Se basa en la relación entre puntos del contorno de un objeto. Este descriptor puede adaptarse para lograr invarianza a escala y rotación.

Otras técnicas, llamadas *regiones afín-invariantes e imágenes spin* [29] se basan en la obtención del histograma de zonas circulares concéntricas de la región de interés, la cual es previamente normalizada en términos de sus valores característicos. Para

esta normalización, se le aplica una transformación afín a la región para que los valores característicos se igualen. A este paso se le conoce como una normalización afín. Este descriptor logra invarianza a transformaciones afines y rotaciones, y es uno de los pocos que puede adaptarse para lograr cierto nivel de invarianza a cambios de iluminación.

Como puede verse, estas técnicas incorporan diferentes acercamientos a la descripción de la región o "circunstancia" de los puntos de interés. El problema de la invarianza a cambios de iluminación se mantiene en todos estos descriptores y es uno de los problemas más difíciles de sortear. Esto se debe a que la representación de una misma escena bajo dos condiciones de iluminación diferentes puede resultar considerablemente disímil. No así para transformaciones como rotación o cambio de escala ya que éstas no alteran significativamente el contexto de los puntos de interés. Los cambios de escala y rotación mantienen las relaciones entre los puntos, sin embargo, al cambiar la iluminación, se altera la estructura medular del mecanismo para representar una escena, lo que hace que la comparación entre los puntos sea mucho más complicada.

A continuación se explica una técnica que detecta características principales que incorpora algunos de los conceptos explicados anteriormente.

4.4. Transformada de características invariante a escala

La Transformada de características invariante a escala (SIFT [32], por sus siglas en inglés) es un sistema completo de detección y caracterización de puntos críticos. Esto quiere decir que cumple la función de detección de características y la elaboración de

un descriptor que las distingue.

SIFT realiza esta operación siguiendo los siguientes pasos:

1. Detección de extremos en el espacio-escala
2. Localización de puntos críticos
3. Asignación de orientación
4. Descriptor del punto crítico

A continuación se explica a detalle cada uno de los pasos para generar una serie de puntos característicos definidos a través de sus descriptores.

4.4.1. Detección de extremos en el espacio-escala

La detección de puntos críticos comienza por detectar una serie de candidatos que se distingan dentro de su región. Esta "distinción" quiere decir ser un máximo o un mínimo local en términos de luminosidad (ser un detalle llamativo). Como se vio en la sección 4.2.1, se pueden encontrar estos máximos y mínimos al utilizar un filtrado tipo *LoG*. Este filtrado puede aproximarse obteniendo las diferencias entre dos niveles diferentes de resolución del espacio-escala de la imagen (conocido en inglés como *scale-space*). Al restar la imagen de menor resolución a la de mayor resolución, lo que resulta son los detalles. Esto es el filtrado llamado *DoG*. Si se encuentran estas diferencias para cada par de niveles contiguos del espacio-escala de la imagen, se genera una estructura parecida a la de la figura 4.6.

Este proceso se aplica a cada nivel de submuestreo de la imagen, llamado *octava*. En cada nivel de las imágenes *DoG* se resaltan los detalles más significativos. El siguiente paso consiste en comparar estos puntos entre diferentes niveles y así encontrar

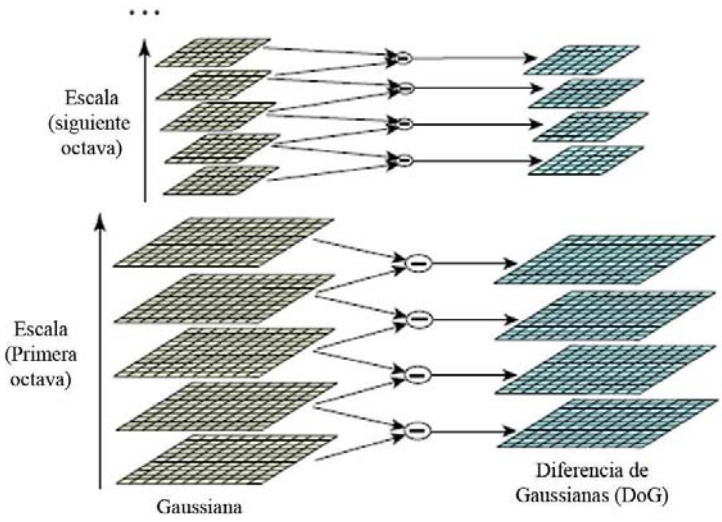

Figura 4.6: Obtención del espacio-escala Gaussiano y DoG. Obtenida de [32]

los extremos en escala. Esto se logra al comparar cada pixel con sus 26 vecinos (los 8 en su misma escala y los 18 de la escala superior e inferior). Aquellos pixeles que sean un máximo o un mínimo de esta región cúbica se marcan como candidatos a punto crítico.

La cantidad de puntos detectados de esta forma depende de las frecuencias de muestreo en espacio y en escala. Esto quiere decir que se debe decidir el espaciamiento entre escalas de resolución y la frecuencia de submuestreo para el espacio-escala. Estos valores se escogen dependiendo del poder computacional con que se cuente y los requisitos de tiempo de operación. Sin embargo, un descriptor de buena calidad puede generarse con un pequeño subconjunto de estos candidatos.

4.4.2. Localización de puntos críticos

El siguiente paso consiste en realizar un análisis detallado de los candidatos en relación a los datos cercanos en términos de posición, escala y razón de curvaturas

principales. Esto ayuda a determinar qué puntos destacan realmente y cuáles tienen bajo contraste o están mal colocados.

Un punto importante es que no se mantiene el punto candidato exactamente, sino que se encuentra la posición interpolada del extremo (que puede ser entre escalas). También se eliminan aquellos candidatos que se encuentren sobre bordes, ya que resaltan con la operación de DoG pero son muy sensibles al ruido.

4.4.3. Asignación de orientación

Se desea construir un descriptor invariante a rotaciones. La forma en que esto se logra en SIFT es basar el descriptor en una orientación consistente con las características locales del punto crítico. Esta orientación se determina a partir de la orientación y magnitud del gradiente. La base del gradiente es la imagen L (ecuación 4.2.2) más cercana a la escala del punto crítico (ya que es un extremo interpolado).

Se obtiene un histograma de orientaciones de gradiente de puntos en una región cercana al punto crítico. Cada elemento añadido al histograma (de 36 posibles categorías representando los 360 grados) es ponderado por su magnitud y una ventana gaussiana de $\sigma = 1.5$ veces la escala del punto crítico. A continuación se asigna una orientación al punto crítico relativa al máximo pico del histograma y a aquellos picos dentro del 80 % del máximo. Esto quiere decir que puede haber más de una orientación por punto crítico (esto sucede para aproximadamente 15 % de los puntos críticos). Finalmente, para mejorar la precisión, se interpola una parábola a los 3 valores del histograma más cercanos a cada pico.

4.4.4. Descriptor del punto crítico

Hasta este momento se cuenta con puntos críticos con una posición en el espacio, una escala y una orientación representativa. Ahora se necesita construir el descriptor de tal manera que añada cierta invarianza a otra posibles alteraciones, como lo son el punto de vista y cambios de iluminación. El descriptor se obtiene de la siguiente forma:

1. Se encuentran la magnitud y orientación del gradiente de puntos muestra en una región cercana a cada punto crítico.

2. Estos valores se ponderan con una ventana Gaussiana

3. Estas muestras son acumuladas en un histograma de orientaciones dividido en 4×4 sub-regiones.

Este proceso se puede ver en la figura 4.7, donde la longitud de cada flecha se obtiene al sumar las magnitudes del gradiente cercanas a esa dirección dentro de la región. En la figura se muestran 2×2 sub-regiones, pero el algoritmo trabaja en realidad con 4×4.

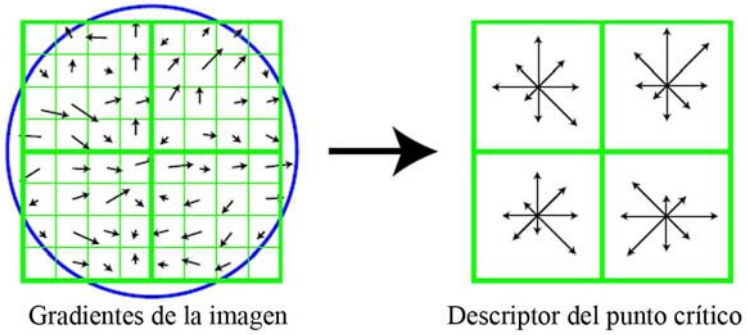

Figura 4.7: Obtención del descriptor del punto crítico. Obtenida de [32]

El resultado es un histograma que contiene $4 \times 4 \times 8 = 128$ valores. Para reducir efectos en cambio de iluminación, el vector está normalizado a longitud unitaria. Esto permite la invarianza ante cambios afines de iluminación, que corresponde a cambios de contraste (multiplicar cada pixel por una constante) y brillo (sumar a cada pixel un valor constante). Sin embargo, pueden haber alteraciones no-afines de iluminación debido a la naturaleza tridimensional de la escena. Esto se combate reduciendo el efecto de valores altos de magnitud de gradiente (que son los más afectados por este tipo de alteraciones). El resultado es un descriptor invariante a escala, rotación y transformaciones afines, y bastante resistente ante cambios de iluminación.

4.4.5. Comparación

En este punto es posible realizar la comparación entre dos imágenes (también llamado *registro* de imágenes). Esta comparación se basará en los puntos característicos contenidos en cada imagen y definidos a través de sus descriptores. Lo que se hace es encontrar, para cada punto crítico de una imagen, aquel más parecido en la otra imagen. Hay varios métodos para resolver este problema, dentro de los cuales se encuentran métodos exhaustivos (todos contra todos), métodos basados en indexado (*geometric hashing* [28]), métodos basados en árboles (BBF [6]) y más. En el presente trabajo se realiza la comparación de los descriptores de forma exhaustiva, lo que quiere decir que se compara cada punto contra cada punto del par de imágenes para encontrar las correspondencias. Este método es computacionalmente costoso pero permite eliminar del análisis errores debidos a la elección del método de registro.

En el trabajo de Beis y Lowe [6] se describe el método BBF o Best-Bin-First (mejor cubo primero) que sigue una técnica de vecino más cercano donde la medida de distancia es la distancia euclidiana entre los descriptores de cada punto. El vecino

más cercano se busca solamente entre un grupo de puntos que están clasificados de acuerdo a un criterio de similitud. Se aprovecha una estructura de montículo o *heap* para clasificar los puntos y así acelerar la búsqueda. Este método está incorporado en el trabajo de Lowe [32] como método recomendado en la descripción de SIFT aplicada a detectar correspondencias.

Después de este paso se tiene una serie de correspondencias entre imágenes. Sin embargo, es todavía posible que existan correspondencias erróneas. A éstas se les suele llamar *falsas correspondencias* (*outliers* en inglés) y se deben a las similitudes entre las características de la región entre los puntos críticos. A continuación se presenta una técnica para eliminar estas correspondencias erróneas.

4.5. Correspondencias por transformación de grafos

Como se mencionó en la sección anterior, al final del emparejamiento por vecino más cercano se pueden tener correspondencias erróneas entre imágenes. Esto puede causar fuertes errores debidos a una mala interpretación de la información que se extrae de esta correspondencia. Es de vital importancia eliminar las falsas correspondencias y mantener las correspondencias correctas.

En [2] se propone una técnica basada en grafos para la eliminación de las falsas correspondencias. Esta técnica, llamada correspondencias por transformación de grafos (cuyo nombre original es "Graph Transformation Matching" ó GTM) se basa en información de estructuras locales de la imagen para seleccionar las correspondencias correctas.

El principio de GTM es forzar relaciones espaciales coherentes de las correspondencias finales entre las dos imágenes. Esto se logra eliminando iterativamente las

correspondencias que perturban las relaciones de vecindad entre los puntos de cada imagen. Las relaciones de vecindad se establecen al relacionar puntos mediante un grafo no dirigido. Esta gráfica, llamada *gráfica mediana de los K_g vecinos más cercanos*, define vértices que validan, o definen, estructuras de vecindad entre puntos (con K_g definiendo la conectividad mínima del grafo). Si vértices vecinos en una imagen resultan no vecinos en otra, se altera la relación de vecindad, y por lo tanto, se detecta una falsa correspondencia. Al eliminarse el vértice problemático se vuelve a comparar las vecindades entre vértices de cada grafo de la imagen hasta que las vecindades sean similares. El funcionamiento de este algoritmo puede observarse para el par de grafos Gp y Gp' en la figura 4.8. Se puede observar cómo, después de algunas iteraciones, se eliminan las diferencias entre vecindades y se llega a un grafo común. Este grafo final establece relaciones de vecindad similares para los puntos-correspondencia de cada imagen.

Comparando GTM con el método de eliminación de correspondencias erróneas conocido como RANSAC[15], GTM produce una menor cantidad de correspondencias finales (a partir del conjunto inicial), pero elimina más características erróneas [2]. Esto es ideal para la presente aplicación, ya que como se verá en el capítulo 6, la estimación de posición geométrica es muy sensible a ruido entendido como errores de correspondencia.

4.6. Secuencia de comparación

En este trabajo, como en [32], el objetivo es encontrar la imagen más parecida a una vista actual dentro de una base de datos de varias imágenes. Los pasos descritos anteriormente pueden generalizarse para efectuar una secuencia de comparación entre

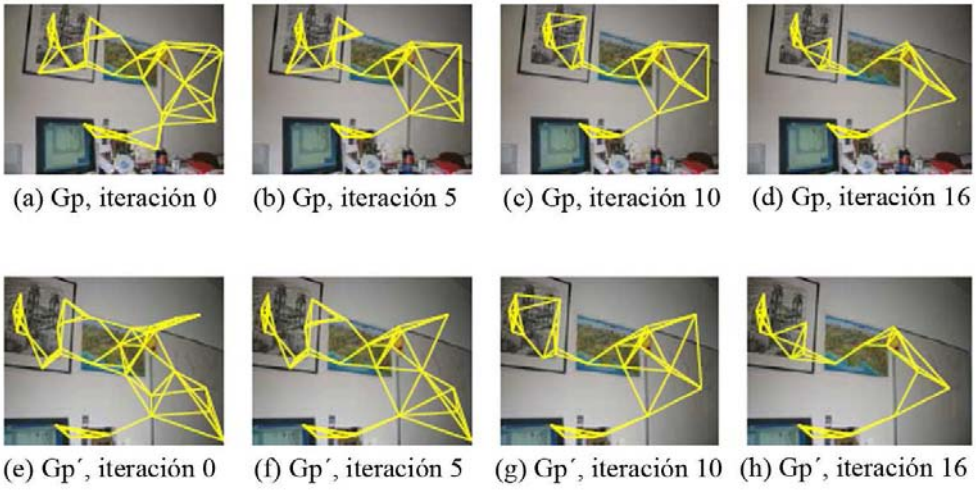

Figura 4.8: Funcionamiento de GTM. Se muestra la comparación de dos vistas. La primera, replicada a lo largo de las imágenes (a)–(d) y la segunda se muestra replicada a lo largo de las imágenes (e)–(h). Como puede observarse, se forma un grafo inicial con los puntos críticos detectados en cada una, correspondiente a las imágenes (a) y (e) respectivamente. A continuación, el algoritmo GTM se encarga de eliminar los puntos de cada grafo que perturban la relación de vecindad entre los grafos (aquellos puntos que los hacen diferentes). Esto se hace hasta que las relaciones de vecindad sean iguales, ilustrado en las imágenes (d) y (h) respectivamente. Obtenida de [2]

una vista y una base de datos de imágenes (BDimag, descrita en el capítulo 1). Para poder realizar esta secuencia de comparación se necesita antes tener una base de datos de descriptores a comparar. Para obtener esta base de datos de puntos críticos (que llamaremos BDkey) se realizan los siguientes pasos:

1. Transformar cada imagen de la BDimag a formato “pgm” (formato *portable graymap*), ya que es el formato sobre el cual opera SIFT.

2. Usar el algoritmo SIFT para obtener los puntos críticos de cada imagen y un archivo (tipo “.key”) que contenga sus descriptores.

3. Almacenar la información de cada archivo “.key” en formato binario para que

sea más compacto.

La secuencia de comparación se describe a continuación:

1. Obtener el archivo ".key" relacionado con la imagen de la vista actual.

2. Encontrar las correspondencias iniciales entre "vista.key" y cada archivo de la BDkey. Esto corresponde a comparar la imagen actual contra todas las imágenes de la base de datos de referencia. El resultado de este paso es una serie de candidatos, cada uno con una serie de correspondencias con "vista.key".

3. Usar el algoritmo GTM para eliminar las falsas correspondencias del grupo de correspondencias de cada candidato.

Al final de estos pasos se cuenta con los candidatos más "parecidos" a la vista actual del robot. Esto se define a través del número de correspondencias (que deben superar un número mínimo) halladas entre la vista actual y las imágenes de referencia. Dado que cada imagen candidata tiene unas coordenadas asociadas, también se cuenta con una serie de posiciones cerca de las cuales puede estar la posición real del robot. Las secciones siguientes explican las técnicas usadas para la localización del robot a partir de las coordenadas de los candidatos y las correspondencias con la vista actual del robot.

Capítulo 5

Acumulación por umbral de calidad

En este capítulo se explica una de las alternativas probadas para generar una aproximación de posición real a partir de una serie de puntos candidatos. Se explica brevemente el acercamiento a partir de la selección de cúmulos y el método específico seleccionado.

5.1. Introducción

En el escenario presentado en este trabajo, hay una etapa muy importante que consiste en determinar la primera aproximación a la posición real del robot. Lo que se ha logrado hasta este punto es generar una serie de candidatos (con sus coordenadas asociadas) que se parecen a la imagen que representa la vista actual del robot. El parecido se manifiesta en términos de número de correspondencias correctas (sin falsas correspondencias luego de GTM). En el siguiente capítulo se explica una técnica con la cual se puede determinar, bajo un contexto geométrico, la pose del robot. En ésta técnica, llamada *localización por geometría epipolar* se aprovechan

las características cualitativas de las correspondencias. Sin embargo, para que esto se pueda lograr, es vital que se satisfagan ciertos requerimientos mínimos, los cuales no siempre se cumplen. Es por esto que se debe de contar con un método alternativo de posicionamiento.

Este método alternativo tiene que ser capaz de generar una buena aproximación a partir de la información con la que se cuente. Esta información es el grupo de candidatos "cercanos", la cantidad de correspondencias entre la vista actual con cada candidato y la ubicación de estos. Una idea es dividir los candidatos en grupos cercanos (que se encuentren en la misma zona) y utilizar el centroide de estos grupos como posiciones probables del robot. Esta idea se basa en un método de agrupamiento de puntos en cúmulos.

La agrupación de puntos en cúmulos, también llamado *acumulación* o *clustering*, consiste en dividir la información en grupos con características similares. La necesidad de hacer acumulación proviene de la necesidad de simplificar grandes cantidades de información o tener la capacidad de representarlas a través de sus características más importantes; aquellas que distinguen similitudes y diferencias críticas entre los distintos tipos de datos. Esta metodología suele usarse en minería de datos, reconocimiento de patrones, aprendizaje automático, segmentación y varias otras disciplinas. A continuación se describirán brevemente algunos tipos de acumulación y sus aplicaciones.

5.2. Tipos de acumulación

En el trabajo de Berkhin [9] se presenta una breve descripción de las técnicas de acumulación más representativas. Normalmente, se dividen en tres tipos generales:

1. Acumulación Jerárquica,

2. Acumulación por particiones y

3. Métodos alternativos

Los métodos basados en acumulación Jerárquica se basan en la agrupación actual de cúmulos para determinar la del siguiente paso. Existen dos tipos básicos: aglomerativos o divisivos. Los primeros se ocupan de unir grupos encontrados en la etapa anterior para así definir menos cúmulos, mientras que los segundos se ocupan de partirlos para definir más. Un ejemplo de este tipo de acumulación es el método aglomerativo de acumulación por vecino más cercano (*single linkage clustering*).

Los métodos de acumulación por particiones determinan el número de cúmulos *a priori* y se ocupan en definirlos (asignar puntos a cada uno). Los algoritmos *K-means* y *Fuzzy c-means* son ejemplos de este tipo de acumulación.

Bajo la categoría de métodos alternativos se agrupan aquellos métodos que no pueden clasificarse directamente en las dos categorías mencionadas ya que se basan en modificaciones de los tipos principales o usan una metodología completamente distinta. Algunos de estos algoritmos están basados en generación de claves (*hashing*), teoría de grafos, métodos estadísticos e incluso en reglas heurísticas. Una descripción más detallada de estos puede verse en [9].

Normalmente, se utiliza una medida de distancia para determinar la pertenencia o exclusión de un dato en relación a un cúmulo. Esta medida de distancia sirve para determinar el parecido entre los miembros de cada cúmulo. Algunas de las medidas típicas son la distancia Euclidiana, la distancia Manhattan y la distancia de Hamming.

Las circunstancias específicas de este trabajo requieren un algoritmo que pueda definir un número de cúmulos que represente adecuadamente el espacio de candidatos. Este número de cúmulos puede variar dependiendo en la cantidad y distribución inicial

de candidatos. El algoritmo seleccionado se explica a continuación.

5.3. Acumulación por umbral de calidad

Acumulación por umbral de calidad (QTC, por sus siglas en inglés) es un método de acumulación desarrollado por Heyer et al [23] para agrupar genes por sus características. Este método tiene la ventaja de que los cúmulos se van formando dinámicamente (no es un número fijo) y siempre da el mismo resultado. Un esquema del algoritmo se muestra en 5.1.

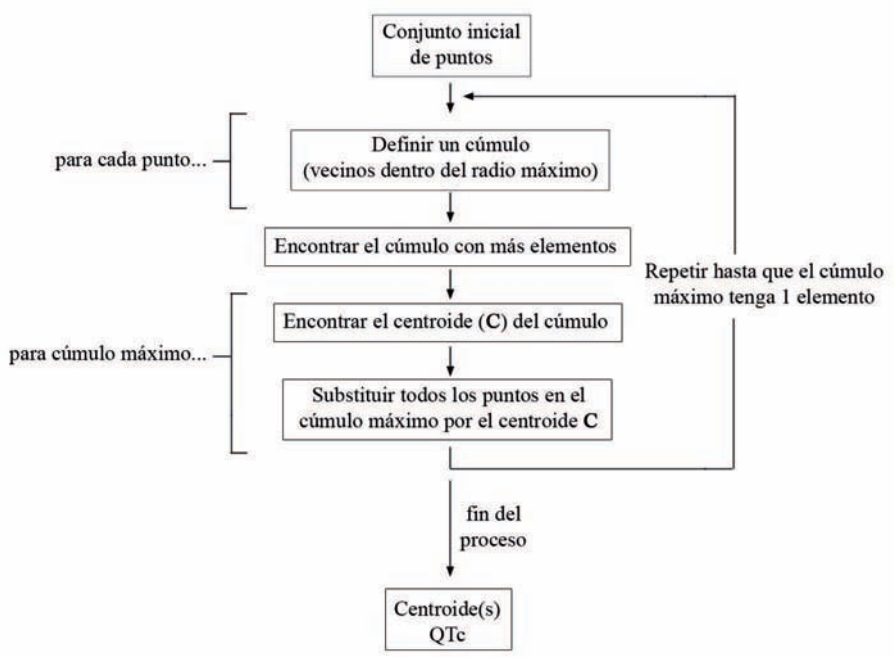

Figura 5.1: Esquema del proceso de QTc

Se parte de un conjunto de poses descritas por un vector dentro del plano coordenado del experimento. Para el problema tratado en esta tesis, el vector que describe

cada pose o nodo es de la forma $n_i = (x_i, y_i, \theta_i)^T$, donde el par x_i, y_i denota la coordenada de posición del nodo y θ_i su orientación. Estos nodos son, en realidad, las poses que representan a cada una de las vistas candidatas extraídas de las imágenes de referencia. Cada candidato tiene un peso asociado que representa la importancia del candidato dentro del cúmulo. A mayor peso, mayor importancia, lo que significa que un nodo con un peso elevado es más representativo que uno con peso bajo. Para el problema tratado aquí, el peso representa "parecido" (definido por el número normalizado de correspondencias correctas). A mayor peso, hay un mayor parecido entre la vista candidata y la vista actual del robot.

Los pasos del algoritmo se detallan a continuación:

1. Se elige un diámetro máximo para un cúmulo. Éste dependerá de la métrica usada y la precisión que se desee obtener. La forma en que varía el algoritmo dependiendo del radio se explica más adelante.

2. Construir un cúmulo candidato para cada punto. Esto se hace al incluir al vecino más cercano al punto central hasta que el siguiente vecino más cercano se encuentre fuera del radio máximo definido en el punto anterior.

3. El cúmulo que contenga la mayor cantidad de puntos se define como el cúmulo máximo. Si este cúmulo máximo tiene más de un punto, entonces se procede con el algoritmo. De lo contrario, el (los) punto(s) restante(s) constituyen el resultado del algoritmo.

   Se encuentra el centroide de este cúmulo máximo. El centroide se basa en el *centro de masa* o *promedio ponderado* de los puntos dentro del cúmulo. El centroide

se obtiene al aplicar la siguiente ecuación:

$$C = \begin{bmatrix} x_C \\ y_C \\ \theta_C \end{bmatrix} = \begin{bmatrix} \frac{1}{P_T} \sum_{i=0}^{m} x_i P_i \\ \frac{1}{P_T} \sum_{i=0}^{m} y_i P_i \\ \frac{1}{P_T} \sum_{i=0}^{m} \theta_i P_i \end{bmatrix}$$

donde P_i representa el *peso* del candidato i y P_T es el peso total del cúmulo actual. El centroide es el punto que representa de mejor forma a la totalidad de puntos dentro del cúmulo.

4. Se sustituyen los puntos del cúmulo máximo por el centroide encontrado en el paso anterior.

5. Repetir recursivamente con el grupo reducido de puntos.

Al final de este algoritmo, se podría tener uno o varios puntos que representan el conjunto original. Si se tiene varios puntos quiere decir que, dado el radio máximo elegido, el conjunto original de puntos se encuentra en la misma "zona" y puede representarse con un solo centroide. Éste se puede usar como la estimación de pose del robot.

Si se tiene más de un punto al final del algoritmo, quiere decir que para el radio máximo predefinido, existe más de una "zona" relevante y por lo tanto, más de una posible pose a considerar. Si éste es el caso, se debe tener cuidado con la siguiente acción del robot ya que la determinación de su posición es ambigua y cualquier acción subsecuente es riesgosa. En general, es recomendable realizar un giro sin avance (de aproximadamente 30°) para repetir el algoritmo con nuevos candidatos. Esto se explica detalladamente en el capítulo 7.

El centroide o centroides finales tienen dos partes importantes: las coordenadas de posición finales x_C, y_C y la orientación promedio θ_C.

Como se puede ver por el proceso antes descrito, la decisión sobre la magnitud del radio máximo es vital para definir el concepto de "zona" mencionado arriba. Por ejemplo, en un escenario donde se tienen varios cuartos unidos por un pasillo, es conveniente definir un radio máximo tal que permita distinguir elementos que no se encuentran en la misma región (cuartos o pasillo). Si se elige un radio máximo demasiado pequeño, es relativamente fácil, pese a una probable cercanía entre las poses de los candidatos, que se defina más de una "zona" y por lo tanto más de una pose estimada (que significa ambigüedad). Por el otro lado, si se define un radio máximo demasiado grande, se podrían "promediar" candidatos que se encuentran el lugares claramente distintos (como vistas que estén en cuartos diferentes pero que se parecen). La elección del radio dependerá del ambiente mismo y de la precisión final deseada.

Para este trabajo se eligió una medida de distancia Euclidiana, ya que en este contexto es la que brinda la medida más precisa. Esto se debe a que la distancia se mide en línea recta y no por el número de saltos verticales u horizontales de nodos en una cuadrícula (Manhattan), lo que ocasiona diferencias fuertes entre nodos que se encuentran en la misma línea y otros en posiciones diagonales. También es importante hacer notar que no se utiliza una métrica como la distancia de Mahalanobis, que está basada en la correlación entre las características de cada candidato (las correspondencias con la vista actual), porque éstas pueden depender de elementos no relacionados de la escena o diferentes estructuras de la imagen. QTc con una métrica Euclidiana representa un método sencillo y rápido para una primera aproximación de la posición del robot.

Capítulo 6

Geometría Epipolar

En este capítulo se explican los conceptos básicos utilizados para aproximar, geométricamente, la posición del robot en relación a uno o varios candidatos cercanos. Esta metodología constituye la segunda forma de aproximación, alternativa a la explicada en el capítulo anterior.

6.1. Introducción

Para poder hacer una estimación geométrica de la posición del robot se tienen que cumplir una serie de condiciones mínimas en la calidad de la información recopilada. Esto esencialmente quiere decir que se debe de contar con información suficiente para el uso de modelos de mayor exactitud como el que se explica a continuación. Es importante mencionar que ésta técnica se plantea como alternativa a la explicada en el capítulo anterior, donde el único requisito es que exista al menos un candidato (definido así por la aparición de el número mínimo de correspondencias correctas). Para esta técnica, el requisito necesario es no solo que se cumpla un requisito de cantidad de correspondencias sino también de la calidad de las mismas. El método se

explica a continuación.

El problema es determinar la posición de un robot a través de la relación entre lo que ve el robot (la vista actual) y las vistas conocidas más cercanas (cuyas posiciones y orientaciones son conocidas). Otra forma de ver este problema es pensar que la posición real del robot es un punto al que se llega por un movimiento desde una posición conocida. Si este movimiento se puede estimar, entonces la posición final puede determinarse.

La estimación de movimiento a través de imágenes se basa en la extracción de información acerca del espacio capturado. Se debe determinar el cambio espacial de acuerdo a lo percibido en dos momentos diferentes, representados por imágenes. Es importante entender que en este escenario la información con la que contamos está modelada como un espacio proyectivo (la proyección del mundo en el plano de la imagen). En este espacio las medidas de ángulos y distancias pueden no tener relación con lo que sucede en el espacio euclidiano (3-D). No obstante, dentro de este espacio todavía se cuenta con mucha información (como coplanaridad, colinearidad y relaciones cuantitativas), la cual se puede aprovechar para extrapolar características del ambiente euclidiano. Si aparte de esta información se tiene una relación entre el espacio proyectivo y el sistema de coordenadas de la cámara en el espacio euclidiano, se pueden extrapolar datos de ángulos y medidas. Éste es el caso de estimación de movimiento con cámaras calibradas.

Como se explica en el trabajo de Luong y Faugeras [33], hay dos acercamientos para la obtención de esta información. La primera consiste en encontrar una relación entre las coordenadas en pixeles y las del mundo 3-D para contar con un modelo de relación espacial. Para esto, se tiene que cuidar que la cámara que captura las

imágenes esté bien calibrada, lo que fija un modelo de proyección entre los pixeles y el mundo. Así, se pueden encontrar relaciones espaciales a través de mediciones con pixeles. El problema con éste método es que la calibración de la cámara, especialmente en sistemas de visión activos, no puede garantizarse.

Un segundo método consiste en aprovechar ciertas características de espacio que tienen la misma forma de proyectarse en una imagen. A estas características se les llama *invariantes proyectivas* y definen una relación geométrica entre las diferentes proyecciones de una misma escena. Este acercamiento no necesita que las cámaras estén calibradas, por lo que un número menor de parámetros se deben de definir para determinar la relación proyectiva entre dos imágenes. Sin embargo, la información extraída debe de procesarse aún más para extraer significado útil en el espacio euclidiano.

La información de relación geométrica se encuentra totalmente definida en la llamada *matriz esencial*, donde se encuentran encapsulados los parámetros de la geometría del par de imágenes. A la geometría propia a dos vistas representada por esta matriz se le llama *geometría epipolar*.

6.2. La restricción epipolar

El escenario de relación geométrica entre dos vistas de una misma escena se representa en la figura 6.1.

Se explica éste escenario siguiendo la línea de Trucco y Verri en [44]. Se explica el escenario y los elementos principales de los sistemas ópticos del par de imágenes de la escena a descubrir. Se parte de los centros ópticos O_i y O_d de las dos cámaras tipo *pinhole*, y los planos de proyección correspondientes π_i y π_d (también llamados planos

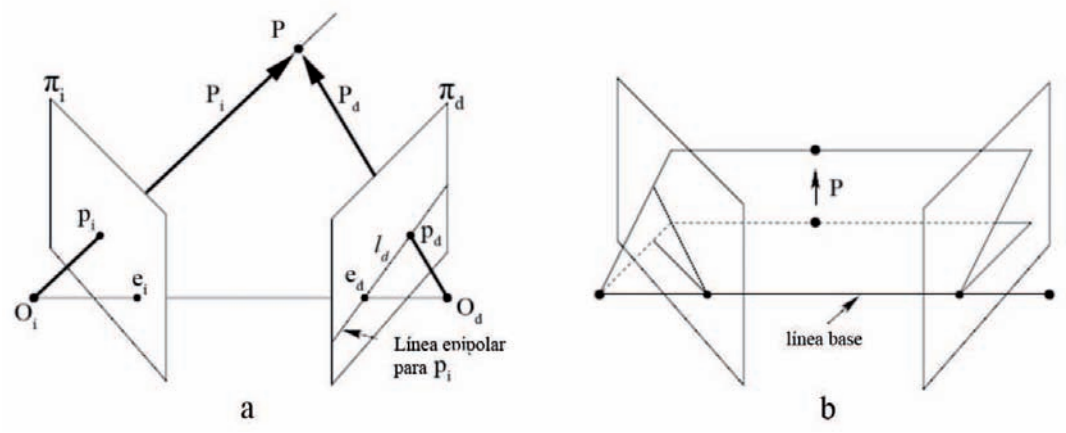

Figura 6.1: Geometría Epipolar. En (a) se observan dos vistas de una misma escena, representadas por un par de planos de imagen π_i y π_d y el punto de la escena **P**. Los epipolos corresponden a las proyecciones de cada centro óptico en el plano de la imagen complementario. El plano epipolar es aquél formado por los centros ópticos y el punto **P**. El trabajo de recuperación de la escena se hace a partir de las proyecciones de cada punto, las cuales serán puntos correspondientes en cada plano de imagen. En (b) se observa el efecto de cambiar de posición el punto a proyectar. Como puede verse, las líneas epipolares se mueven, pero los epipolos se mantienen fijos. Basada en las figuras 9.1 y 9.2 de [22] y en la figura 7.6 de [44].

de imagen). Esto se refiere al par de imágenes obtenidas de la misma escena, sólo que para explicarlo se tratarán aquí como dos cámaras distintas (pese a que pueda ser la misma). Las distancias focales son f_i y f_d respectivamente. Si se tiene un punto en el espacio P, el plano que se define por los centros ópticos y este punto se denomina el *plano epipolar*. La línea que une los centros ópticos se llama la *línea base*.

Los sistemas de referencia definidos por las dos cámaras se relacionan a través de los *parámetros extrínsecos*. Estos definen una transformación rígida en el espacio 3-D a través de un *vector de traslación* $\mathbf{T} = (O_d - O_i)$ y un *matriz de rotación* $\mathbf{R}$. Si el punto P se define por los vectores $P_i = [X_i, Y_i, Z_i]$ y $P_d = [X_d, Y_d, Z_d]$ en cada sistema de referencia, entonces la relación entre estos está dada por:

$$P_d = \mathbf{R}(P_i - \mathbf{T}) \tag{6.2.1}$$

Las proyecciones de P en π_i y π_d son p_i y p_d respectivamente, y se derivan de las ecuaciones de proyección de perspectiva siguientes:

$$\begin{aligned} p_i &= \frac{f_i}{Z_i} P_i \\ p_d &= \frac{f_d}{Z_d} P_d \end{aligned} \tag{6.2.2}$$

Las proyecciones de cada centro óptico en el plano de proyección de la otra cámara se llaman *epipolos* (e_d y e_i), los cuales también se definen como el punto de intersección de cada plano proyectivo con la línea base.

Por último, se define a la intersección del plano epipolar con cada plano de imagen como una *línea epipolar*. La relación entre estos elementos define una restricción geométrica llamada la *restricción epipolar*.

La restricción epipolar relaciona la proyección de un punto en un plano de imagen con una línea en el plano de la imagen de la otra cámara. En la figura 6.1 (a) se puede ver esta relación entre el punto p_i del plano π_i y la línea l_d en el plano π_d. Esta relación indica que dado que la proyección de P en el plano de la imagen de la cámara O_i es el punto p_i, entonces el punto P podría encontrarse en cualquier sitio de la recta que une O_i con p_i, cuya proyección sobre el plano de la imagen π_d forma a la línea epipolar l_d.

6.2.1. La matriz esencial

Siguiendo el desarrollo de Truco y Verri[44], la ecuación del plano epipolar puede escribirse como la condición de coplanaridad de los vectores P_d, $\mathbf{T}$ y $P_i - \mathbf{T}$, la cual queda como:

$$(P_i - \mathbf{T})^\top \mathbf{T} \times P_i = 0 \tag{6.2.3}$$

donde $\mathbf{T}$ representa el vector de la traslación entre un centro óptico y el otro. Si esta ecuación se combina con 6.2.1, se obtiene:

$$(\mathbf{R}^\top P_d)^\top \mathbf{T} \times P_i = 0 \tag{6.2.4}$$

El producto cruz de vectores puede escribirse como una multiplicación por una matriz de rango incompleto, por lo que

$$\mathbf{T} \times P_i = \mathbf{S} P_i$$

donde

$$\mathbf{S} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{6.2.5}$$

entonces, 6.2.4 se convierte en:

$$
\begin{aligned}
(\mathbf{R}^\top P_d)^\top \mathbf{T} \times P_i &= 0 \\
(\mathbf{R}^\top P_d)^\top \mathbf{S} P_i &= 0 \\
P_d^\top E P_i &= 0
\end{aligned} \tag{6.2.6}
$$

con

$$E = \mathbf{RS} \tag{6.2.7}$$

donde la matriz E de 3×3 es la matriz esencial. Se pueden combinar 6.2.2 y 6.2.6 para obtener una relación entre puntos proyectados:

$$p_d^\top E p_i = 0 \tag{6.2.8}$$

lo que relaciona una proyección con otra a través de E. Otra cosa que se extrae de esto es que, dado que el punto p_i es la proyección del punto P en π_i, entonces, el vector definido por $P_i = O_i P$, que pasa por p_i, tiene una proyección en el plano π_d, la cual es la línea epipolar l_d. Esto quiere decir que como p_i está en el vector P_i, entonces su proyección se encuentra sobre la línea l_d. Algebráicamente, esto se traduce en

$$l_d = E p_i \tag{6.2.9}$$

Así, combinando la información de 6.2.8 y 6.2.9, se llega a la restricción epipolar, la cual dice:

> *Puntos proyectados correspondientes deben encontrarse sobre líneas epipolares conjugadas.*

La matriz E establece una relación entre la restricción epipolar y los parámetros extrínsecos que relacionan el par de sistemas ópticos.

Aquí es importante recapitular la lógica de la metodología. Si se encuentra la matriz E, se pueden encontrar los parámetros extrínsecos $\mathbf{R}$ y $\mathbf{T}$ y por lo tanto la relación entre las cámaras O_i y O_d, cosa que se traduce en encontrar la posición relativa de un punto de vista respecto a otro, que es lo que queremos lograr para localizar al robot.

6.2.2. La matriz fundamental

En el caso en que no se tenga una cámara calibrada también es posible encontrar un mapeo entre puntos y líneas epipolares. En este caso, los puntos proyectados sobre el plano de la cámara solo tienen coordenadas en pixeles (ya que no se conoce la relación con el sistema 3-D). Estos puntos $\bar{p}_i$ y $\bar{p}_d$ se relacionan con los puntos con coordenadas de cámara p_i y p_d de la siguiente manera:

$$p_i = M_i^{-1}\bar{p}_i \tag{6.2.10}$$
$$p_d = M_d^{-1}\bar{p}_d$$

donde M_i^{-1} y M_d^{-1} son las matrices de los parámetros intrínsecos. Éstas matrices relacionan las coordenadas de cámara con las coordenadas de pixel, incorporando los parámetros de perspectiva, de transformación de coordenadas y de distorsión de la imagen. El determinar correctamente éstas matrices constituye el proceso de calibración de una cámara.

Sustituyendo 6.2.10 en 6.2.8 tenemos

$$\bar{p}_d^{\top} F \bar{p}_i = 0 \tag{6.2.11}$$

donde F de 3×3 es llamada la *matriz fundamental* y es

$$F = M_d^{-\top} E M_i^{-1} \tag{6.2.12}$$

y se cumple que

$$\bar{l}_d = F \bar{p}_i \tag{6.2.13}$$

En este caso, la reconstrucción de la escena se puede lograr hasta una transformación proyectiva, lo que quiere decir que la información extraída proviene de las proyecciones del escenario en 3-D. Como se menciona al principio de este capítulo, esto es suficiente para extraer características de la escena como colinearidad y coplanaridad. Algunas de las aplicaciones de esta reconstrucción para robots móviles se resumen en [27, 43, 5].

Las propiedades básicas de las matrices E y F son:

La matriz esencial E:

1. Contiene solamente la información de los parámetros extrínsecos
2. Es de rango 2, ya que S en 6.2.7 tiene rango 2 y R es de rango completo.
3. Los dos valores singulares diferentes de cero son iguales

La matriz fundamental F:

1. Contiene información de los parámetros intrínsicos y extrínsecos

2. Es de rango 2, ya que M_d y M_i tienen rango completo y E es de rango 2.

Para el caso tratado en este trabajo, se cuenta con una cámara calibrada. Esto quiere decir que puntos de correspondencias $\bar{p}_i = [\bar{u}_i, \bar{v}_i]$ y $\bar{p}_d = [\bar{u}_d, \bar{v}_d]$ con coordenadas en pixeles pueden pasarse a coordenadas de cámara $p_i = [u_i, v_i]$ y $p_d = [u_d, v_d]$ (usando las matrices de calibración). Así, la reconstrucción de la escena se puede realizar aprovechando la matriz esencial.

El método directo para recuperar la geometría del espacio y el que se usará en este trabajo, depende del desarrollo de la ecuación 6.2.8, donde se establece que hay una relación entre p_i y p_d a través de E. A este método se le conoce como el algoritmo de los 8 puntos.

6.3. El algoritmo de los 8 puntos

En [31], Longuet-Higgins presentó un método lineal basado en relación de correspondencias entre un par de imágenes para obtener la matriz esencial. Esta matriz representa toda la información del escenario epipolar, extraído a partir de un par de imágenes obtenidas por unas cámaras calibradas. Éste método también se puede usar para determinar la matriz fundamental [21, 44].

Partiendo de la ecuación 6.2.8, donde $p_d \rightarrow p_i$ representa un par de puntos correspondientes, si se tienen suficientes correspondencias entre puntos ($n \geq 8$), se puede construir un sistema de ecuaciones lineales cuya solución no trivial es E.

Si se expresan los puntos $\mathbf{p}_i = (u_i, v_i, 1)^\top$ y $\mathbf{p}_d = (u_d, v_d, 1)^\top$ (representación homogénea), cada par de correspondencias generan una ecuación lineal con elementos de la matriz E. En 6.3.2 se puede ver la ecuación correspondiente a un par de puntos p_i y p_d desarrollando la ecuación 6.2.8:

$$p_d^T E p_i = [u_d, v_d, 1] \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0 \tag{6.3.1}$$

y desarrollando:

$$u_i u_d E_{11} + u_i v_d E_{21} + u_i E_{31} + v_i u_d E_{12} + v_i v_d E_{22} + v_i E_{32} + u_d E_{13} + v_d E_{23} + E_{33} = 0$$

la cual puede expresarse como

$$(u_i u_d, u_i v_d, u_i, v_i u_d, v_i v_d, v_i, u_d, v_d, 1)\mathbf{e} = 0 \tag{6.3.2}$$

y si se tienen m correspondencias, se puede definir el sistema de ecuaciones lineales como

$$A\mathbf{e} = 0 \tag{6.3.3}$$

donde $\mathbf{e}$ es un vector de 9 elementos que contiene los elementos de E y A es la llamada *matriz ecuación* y es una matriz con rango máximo de 8. A se define como

$$\mathbf{A} = \begin{pmatrix} u_{i1}u_{d1} & u_{i1}v_{d1} & u_{i1} & v_{i1}u_{d1} & v_{i1}v_{d1} & v_{i1} & u_{d1} & v_{d1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{im}u_{dm} & u_{im}v_{dm} & u_{im} & v_{im}u_{dm} & v_{im}v_{dm} & v_{im} & u_{dm} & v_{dm} & 1 \end{pmatrix} \tag{6.3.4}$$

Dada la naturaleza de A pueden darse tres casos. El primero se denomina el caso mínimo y sucede cuando A tiene rango menor a 8. En este caso,es necesario usar como apoyo restricciones no lineales. La solución a estos casos se puede ver en [22].

En el siguiente caso se tienen exactamente 8 correspondencias. Entonces A tendría rango 8. En este caso se aprovecha el hecho de que se puede fijar el factor de escala para especificar la novena incógnita. Esto se debe a que ya que nos encontramos en un ambiente proyectivo, el sentido de la escala se pierde, lo que quiere decir que las relaciones entre los elementos de la escena están definidas pero sus cantidades pueden fijarse a diferentes escalas. Lo que se dice aquí es que se define una escala arbitraria y se continúa con la solución. Así, se tendrían 9 incógnitas (una de las cuales es el valor de escala que se fija en $||\mathbf{e}|| = 1$) y 9 ecuaciones lineales, por lo que la solución es única (se puede obtener por métodos lineales).

El último caso se da cuando se tienen más de 8 correspondencias. Aquí, el sistema está sobre-determinado y no se puede encontrar una solución no trivial a 6.3.3. Se debe recurrir a un método de mínimos cuadrados que minimice $||A\mathbf{e}||$ sujeto a la restricción $||\mathbf{e}|| = 1$. Un método para obtener el vector $\mathbf{e}$ bajo estas circunstancias se llama *descomposición en valores singulares* (SVD[4, 40], por sus siglas en inglés).

Antes de iniciar éste método, es importante llevar a cabo la normalización de los puntos, ya que debido a la variación de los valores de coordenadas, se pueden presentar inestabilidades numéricas que, a su vez, provoquen errores en la obtención de E. Un método sencillo par evitar este problema es traducir las primeras dos coordenadas de cada punto al centroide de cada conjunto de datos y escalar la norma de cada punto para que la norma promedio de cada conjunto sea igual a uno. Así, se obtiene la matriz E, y luego si se le aplica una de-normalización se encuentra el valor real de E.

Después de la normalización se sigue con la SVD de A. Esta descomposición ayuda a expresar cualquier matriz rectangular de $m \times n$ en términos de 3 matrices:

$$A = UDV^{\top} \tag{6.3.5}$$

donde las columnas de la matriz U de $m \times m$ y la matriz V de $n \times n$ son vectores unitarios mutuamente ortogonales. La matriz D de $m \times n$ es diagonal; los elementos σ_i de la diagonal son llamados *valores singulares* y cumplen que $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$.

Resulta que la minimización de $||A\mathbf{e}||$ se da cuando $\mathbf{e}$ equivale al vector unitario de V correspondiente al menor eigenvalor de D (ver [44], apéndice A.6).

$$\mathbf{e} = (V_{min\ 1}, V_{min\ 2}, V_{min\ 3}, V_{min\ 4}, V_{min\ 5}, V_{min\ 6}, V_{min\ 7}, V_{min\ 8}, V_{min\ 9})$$

y por lo tanto

$$E = \begin{bmatrix} V_{min\ 1} & V_{min\ 2} & V_{min\ 3} \\ V_{min\ 4} & V_{min\ 5} & V_{min\ 6} \\ V_{min\ 7} & V_{min\ 8} & V_{min\ 9} \end{bmatrix} \tag{6.3.6}$$

Después de de-normalizar es vital forzar la restricción propia de ésta matriz, llamada la *restricción de singularidad*, en la que $det(E) = 0$, el rango de $E = 2$ y que los dos valores singulares de E sean iguales. Esta restricción debe de forzarse ya que debido a imperfecciones de medición y relación entre correspondencias esto puede no presentarse en la E obtenida. Así, lo que se hace es obtener la SVD de la matriz fundamental *no restringida* que llamaremos $\tilde{E}$:

$$\tilde{E} = \tilde{U}\tilde{D}\tilde{V}^{\top} \tag{6.3.7}$$

Si como se espera, los dos valores singulares de $\tilde{D}$ son diferentes y el menor valor de $\tilde{D}$ es diferente de 0, se cámbia para que sean $\sigma_1 = \sigma_2 = 1$ y $\sigma_3 = 0$ y generar $\hat{D}$:

$$\hat{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.3.8}$$

Se *regresa* a E reconstruyéndola a partir de su SVD corregida (cambiando $\tilde{D}$ por $\hat{D}$ y así forzar la restricción de singularidad)

$$E = \tilde{U}\hat{D}\tilde{V}^{\top} \tag{6.3.9}$$

6.4. Extracción de traslación y rotación

El problema es que no se conoce la geometría del espacio euclidiano del escenario y queremos recuperarla. A diferencia de técnicas de triangulación, aquí no se conoce la línea base del sistema y por lo tanto no se puede recuperar la escala verdadera de la escena a menos que se conozca la distancia real entre dos puntos de la escena *observada*.

Los pasos para recuperar la traslación $\mathbf{T}$ y la rotación $\mathbf{R}$ se extraen de [44] y se explican a continuación.

Primero, debemos establecer una relación entre E y $(\mathbf{T}, \mathbf{R})$, la cual ya tenemos en la ecuación 6.2.7. Antes de seguir, se busca una normalización de E de la siguiente forma. Se define la matriz

$$E^{\top}E = \mathbf{S}^{\top}\mathbf{R}^{\top}\mathbf{S}\,\mathbf{R} = \mathbf{S}^{\top}\mathbf{S} \tag{6.4.1}$$

o

$$E^\top E = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_y T_x & T_z^2 + T_x^2 & -T_y T_z \\ -T_z T_x & -T_z T_y & T_x^2 + T_y^2 \end{bmatrix} \tag{6.4.2}$$

de 6.4.2 se obtiene la traza de $E^\top E$:

$$Tr(E^\top E) = 2||\mathbf{T}||^2 \tag{6.4.3}$$

ya que si se divide a cada elemento de la matriz esencial entre

$$N = \sqrt{\frac{Tr(E^\top E)}{2}} \tag{6.4.4}$$

equivaldría a normalizar la longitud del vector de traslación $\mathbf{T}$ a 1. Usando esta normalización, la ecuación 6.4.2 queda como

$$\hat{E}^\top \hat{E} = \begin{bmatrix} 1 - \hat{T}_x^2 & -\hat{T}_x \hat{T}_y & -\hat{T}_x \hat{T}_z \\ -\hat{T}_y \hat{T}_x & 1 - \hat{T}_y^2 & -\hat{T}_y \hat{T}_z \\ -\hat{T}_z \hat{T}_x & -\hat{T}_z \hat{T}_y & 1 - \hat{T}_z^2 \end{bmatrix} \tag{6.4.5}$$

con $\hat{E}$ la matriz esencial normalizada y $\hat{\mathbf{T}} = \mathbf{T}/||\mathbf{T}||$. Se puede aprovechar cualquier columna o vector de ésta matriz para obtener los componentes de $\hat{\mathbf{T}}$.

La matriz de rotación puede extraerse algebraicamente de la siguiente forma. Se define $w_i = \hat{E}_i \times \hat{\mathbf{T}}$ con $i = 1, 2, 3$ y $\hat{E}_i$ los tres renglones de $\hat{E}$, pensados como vectores en el espacio 3-D. Si se piensa que $\mathbf{R}_i$ son los renglones de la matriz de rotación (pensados como vectores en 3-D), entonces

$$\mathbf{R}_i = \mathbf{w}_i + \mathbf{w}_j \times \mathbf{w}_k \tag{6.4.6}$$

con la tripleta (i, j, k) tomando valores de todas las permutaciones cíclicas de $(1, 2, 3)$.

Sin embargo, dada la construcción de $\hat{E}^\top \hat{E}$ en términos cuadráticos de $\hat{\mathbf{T}}$, los componentes recuperados pueden diferir en signo de los componentes reales. Así, se pueden obtener cuatro posibles combinaciones del par $(\hat{\mathbf{T}}, \mathbf{R})$. No obstante, al hacer la reconstrucción 3-D del escenario, a partir de los cuatro pares posibles, se descubre uno sólo como la estimación correcta. Esto último nos aclararía la dirección de la traslación, mas no la escala de la misma.

Un elemento importante en la precisión de la aproximación por geometría epipolar es que es muy sensible al ruido, el cual se presenta en forma de correspondencias erróneas naturalmente presentes en información adquirida de sistemas reales de visión [19]. En el trabajo de Haralick se plantea la opción de incrementar la cantidad de correspondencias como método de aminorar el peso de los errores por ruido. Esta posibilidad se explora en el capítulo 9.

En resumen, se parte de una serie de correspondencias entre dos imágenes (en coordenadas de pixeles), las cuales son transformadas (mediante las matrices de calibración), a coordenadas de cámara. Se recupera la geometría epipolar a través de la estimación de la matriz esencial $\mathbf{E}$, la cual se descompone para encontrar el par $(\hat{\mathbf{T}}, \mathbf{R})$. Este par define una transformación rígida en el espacio desde el centro óptico de una cámara al otro. De esta forma, se puede saber la posición y orientación relativa en el espacio entre dos vistas. Es importante mencionar que la estimación de posición es una aproximación de dirección mas no de escala, por lo que para aproximar la posición del robot es preferible el uso de la técnica descrita en el capítulo 5. La orientación, en cambio, puede aproximarse correctamente usando geometría epipolar así como la técnica de QTC. La incorporación de estas aproximaciones se detalla en

el capítulo 8.

Otro punto importante es que es posible utilizar métodos de predicción probabilística para generar una mejor estimación de pose. Esto se plantea en el siguiente capítulo.

Capítulo 7

Localización Probabilística

Aquí se explica el contexto teórico detrás de la predicción de la posición del robot a partir de una posición estimada y un movimiento. Se detallan los antecedentes probabilísticos y se explica la relevancia de esta metodología al problema tratado.

7.1. Introducción

El modelado de un sistema físico debe de tomar en cuenta que éste no es perfecto. Éste es el caso para el problema tratado en esta tesis. Para obtener un modelo más general, se debe incorporar la incertidumbre provocada por errores de medición y tomar en cuenta el ruido en el sistema y la posibilidad de contar con varias aproximaciones a la solución real.

Para el caso de la localización de un robot por vistas conocidas, las fallas pueden provenir de errores de captura o interpretación de imágenes (como sucede si se tienen superficies reflejantes u obstáculos nuevos), pero también de los errores de registro de movimiento. Esto último quiere decir que debido a imperfecciones en el sistema de movimiento del robot, un movimiento planeado no se refleje en el movimiento

realizado, lo que se traduce en un error de localización. También es posible que se tengan más de una posición candidata para la posición inicial.

Se debe de contar con un método para estimar la posición del robot después de una fase compuesta de percepción y movimiento. El método utilizado en este trabajo se basa en la estimación probabilística, en donde se asignan distintos valores de probabilidad a las posibles ubicaciones del robot. Estos valores deben refinarse con el progreso del sistema para encontrar un punto de mayor probabilidad y así definir la posición más próxima posible a la posición real del robot. Los fundamentos de esta teoría se presentan a continuación.

7.2. Filtros de Kalman [25, 36]

El filtro de Kalman [25] es un algoritmo recursivo de estimación del estado de un sistema dinámico. Es recursivo debido a que se basa en el resultado de la estimación del estado anterior aunado a una medición actual.

El filtro de Kalman se ocupa de combinar diferentes mediciones para encontrar la aproximación óptima al estado real del sistema. El filtro hace esto en dos fases:

1. **Predicción de estado**: donde se estima el estado del sistema en el siguiente instante, y

2. **Corrección**: donde se realiza una medición del estado actual, la cual se incorpora a la predicción.

El filtro de Kalman se basa en dos modelos, que representan la obtención de las estimaciones necesarias para el par de fases mencionadas arriba. Estos son el modelo

de transición de estado y el modelo de medición. A continuación se describe el filtro de Kalman aprovechando la notación presentada en [24].

Para éste trabajo se define un estado del sistema en el instante t como $s(t) = (x_t, y_t, \theta_t)^T$, donde (x_t, y_t) define la posición del robot y (θ_t) su orientación. La transición del estado se modela mediante una función llamada la función de transición de estado, que opera la matriz de transición Φ con el estado actual e incorpora la incertidumbre modelada como ruido del estado. Para el filtro de Kalman mostrado en esta sección, esta función es una función lineal. El modelo de transición de estado $s(t+1)$ es el siguiente:

$$s(t+1) = \Phi s(t) + \omega(t+1) \tag{7.2.1}$$

donde $\omega(t+1)$ representa la incertidumbre de la predicción de estado, y se representa por la esperanza del ruido $E[\omega(t+1)] = 0$ y varianza de éste $E[\omega(t+1)\omega(t+1)^T] \equiv Q(t)$.

El mapeo del estado actual al modelo de medición es $s(t) \rightarrow m(t)$, lo que quiere decir que se pasa de "coordenadas de estado" a sus "coordenadas de medición" respectivas. Para el caso de este trabajo, ambas son iguales dado que la medición visual arroja una coordenada de la forma $(x_t, y_t, \theta_t)^T$. La función de medición se utiliza para obtener el estado siguiente (bajo el modelo de medición) y es el siguiente:

$$m(t+1) = Ms(t+1) + \gamma(t+1) \tag{7.2.2}$$

donde M representa la matriz de medición (que mapea $s(t) \rightarrow m(t)$) y γ representa la incertidumbre de la observación. También es importante recordar que la medición en

sí cuenta con un factor de incertidumbre debido a errores en el sistema de medición. Ésta incertidumbre también se modela con una media en cero y una varianza definida por $E[\gamma(t+1)\gamma(t+1)^T] \equiv R(t)$.

El filtro de Kalman modela el estado del filtro con dos variables:

1. $\hat{s}(t|t)$, que es la estimación del estado en el momento t dadas las observaciones hasta, e incluyendo el instante t, y

2. $P(t|t)$, que es la matriz de covarianza del error de estimación, que representa la incertidumbre de la estimación.

En una dimensión, estos elementos son la media de la predicción y la varianza de la misma.

El filtro de Kalman se desarrolla de la siguiente forma: Primero, se realiza la predicción del estado siguiente usando la función de transición de estado:

$$\hat{s}_p(t+1|t) = \Phi \hat{s}_p(t|t) \tag{7.2.3}$$

donde la multiplicación de la matriz de transición de estado Φ con la medida del estado en el tiempo t representa la aplicación de la función de transición al estado actual. Ahora se obtiene la predicción de la incertidumbre en el tiempo $t+1$:

$$P_p(t+1|t) = \Phi P_p(t|t)\Phi^T + Q(t) \tag{7.2.4}$$

que se obtiene examinando la media del error cuadrático entre la estimación y la posición "real" $s(t+1)$:

$$E[(\hat{s}_p(t+1|t) - s(t+1))(\hat{s}_p(t+1|t) - s(t+1))^T] \tag{7.2.5}$$

Ahora que se cuenta con la estimación de estado por predicción, definido por $\hat{s}_p(t+1|t)$ y $P_p(t+1|t)$, se incorpora la estimación por medición definida por $m_m(t+1)$ y $P_m(t+1)$. Se obtiene el error entre la medición $m(k+1)$ y la predicción del estado $\hat{s}_p(t+1|t)$:

$$e(t+1) = m(t+1) - M\hat{s}_p(t+1|t) \tag{7.2.6}$$

donde la matriz M mapea el estado $\hat{s}_p(t+1|t)$ al modelo de medición para así encontrar la diferencia entre lo medido y la predicción de la medición.

Ahora, se debe de incorporar de alguna manera el error entre la medición y la predicción de medición. En un filtro de Kalman lineal, se encuentra la estimación del nuevo estado (ya con la medición incorporada) de la siguiente forma:

$$\hat{s}(t+1|t+1) = \hat{s}_p(t+1|t) + K(t+1)e(t+1) \tag{7.2.7}$$

donde $K(t+1)$ representa la *ganancia de Kalman* y define la importancia de cada estimación (predicción y medición) en la estimación final. La ganancia de Kalman se define de tal forma que minimice la ecuación de covarianza del error mostrada en 7.2.5 (a partir de una estimación por mínimos cuadrados del error entre la estimación y la posición "real")[24, 46]. Esto se logra planteando la ganancia de Kalman de la siguiente forma:

$$K(t+1) = P(t+1|t)M^T(t+1)[M(t+1)P(t+1|t)M^T(t+1) + R(t+1)]^{-1} \quad (7.2.8)$$

y se puede obtener la matriz de covarianza del nuevo estado:

$$P(t+1|t+1) = [I - K(t+1)M(t+1)]P(t+1|t) \quad (7.2.9)$$

A continuación, en la figura 7.1, se desarrolla un ejemplo de filtrado de Kalman para estimación de posición desarrollado en [36].

En las figuras 7.1(a) y 7.1(b) se ilustran 2 estimaciones de posición con incertidumbre. En la primera, la función $f_{x(t_1)|z(t_1)}(x|z_1)$ muestra la probabilidad de encontrarse en la posición x dada una medición Z_1. Fue tomada en el tiempo t_1 y se representa como una gaussiana centrada en Z_1. Esto quiere decir que la medición indica que la posición es Z_1 con alta probabilidad (el pico de la gaussiana), pero debido a errores en el proceso de medición, la posición real podría estar desplazada hacia los lados de Z_1 con probabilidad descendente. La precisión de la medida se observa en el ancho de la gaussiana, la cual está determinada por la desviación σ_{Z_1}. En este punto, la posición estimada es $\hat{x}(t_1) = Z_1$. Esta gaussiana puede representar la posición estimada después de aplicar la función de transición de estado.

La segunda distribución de probabilidad es $f_{x(t_2)|z(t_2)}(x|z_2)$, que es la que fue tomada en el tiempo t_2 y está centrada en Z_2 con una desviación de σ_{Z_2}. Como se puede ver, la medida de t_2 es más precisa, no obstante, no debe desecharse la medida de t_1, sino que se debe de aprovechar la información que ésta añade a t_2. La combinación de éstas medidas se hace de la siguiente forma:

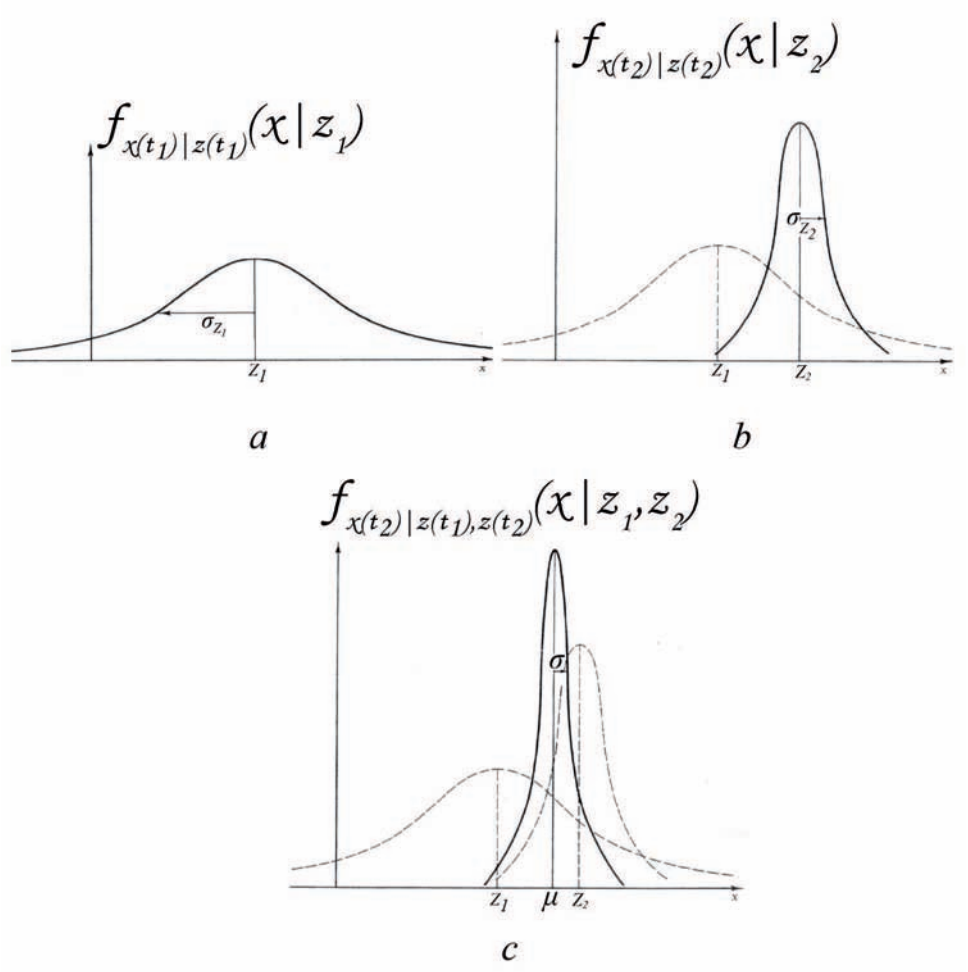

Figura 7.1: Mediciones con incertidumbre y algoritmo de Kalman. En (c) se integran las mediciones mostradas en (a) y (b). Basado en las figuras 1.4 - 1.6 de [36]

$$\mu = [\frac{\sigma_{Z_2}^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2}]Z_1 + [\frac{\sigma_{Z_1}^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2}]Z_2 \tag{7.2.10}$$

$$\frac{1}{\sigma^2} = (\frac{1}{\sigma_{Z_1}^2}) + (\frac{1}{\sigma_{Z_2}^2}) \tag{7.2.11}$$

Esto quiere decir que se hace el promedio entre las medias de cada gaussiana, ponderado por las varianzas relativas a cada medición. Así, si $\sigma_{Z_1}^2 = \sigma_{Z_2}^2$, μ queda en medio. Por otro lado, si alguna de las dos varianzas es mayor a la otra, la media relativa a ésta varianza tendrá un menor peso en el promedio ponderado. Esto tiene sentido ya que esta medida de mayor varianza es más incierta y por lo tanto debería afectar poco al resultado de la ponderación.

Como puede verse en la ecuación 7.2.11, la varianza resultante es menor que la de las varianzas independientes de Z_1 y Z_2. Así, la ponderación de la medida actual con el estado anterior resulta en una estimación más precisa del estado actual.

Para el ejemplo de la figura 7.1(c), la posición estimada es $\hat{s}(t_2) = \mu$ y la varianza es σ.

Desarrollando las ecuaciones 7.2.10 y 7.2.11, tenemos que:

$$\hat{s}(t_2) = [\frac{\sigma_{Z_2}^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2}]Z_1 + [\frac{\sigma_{Z_1}^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2}]Z_2 \tag{7.2.12}$$

$$= Z_1 + [\frac{\sigma_{Z_1}^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2}][Z_2 - Z_1] \tag{7.2.13}$$

o en su forma final:

$$\hat{s}(t_2) = \hat{s}(t_1) + K(t_2)[Z_2 - \hat{s}(t_1)] \tag{7.2.14}$$

donde

$$K(t_2) = \frac{\sigma_{Z_1}^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2} \tag{7.2.15}$$

Como se explica en [36], la ecuación 7.2.14 quiere decir que la mejor predicción de $\hat{s}(t_2)$ es la predicción del nuevo estado más una corrección basada en una nueva medida. Este ejemplo nos permite observar la el razonamiento detrás del filtro de Kalman y la obtención de la ganancia del mismo.

La función de transición de estado para el problema tratado en este trabajo es una función que predice una posición dada la posición anterior y un "movimiento". Al incorporar el aspecto dinámico, no sólo se cuenta con la incertidumbre de la observación (que constituye la medición de la fase de corrección del filtro), sino que se debe tomar en cuenta la incertidumbre de la precisión del registro del movimiento en el modelo. Una forma de llevar el registro de la posición por una serie de movimientos es la incorporación de un modelo de predicción por odometría.

7.3. Modelo de estimación de pose usando odometría

Se le llama odometría al registro de pose de un objeto a partir de las acciones que éste realize. Para el caso de un robot móvil, estas acciones son girar y avanzar. En versiones más complejas [11], la odometría se puede tomar desde las acciones de aceleración y modulación de velocidad por cada actuador del robot, como lo serían las ruedas.

La pregunta importante es la siguiente:

¿Porqué necesitamos aproximar la pose del robot si contamos con una serie de movimientos registrados?

La respuesta es que la serie de movimientos planeados es generalmente diferente a la serie de movimientos realizados. La forma de estimar la pose del robot es contar con un modelo de aproximación que tome en cuenta los posibles errores de movimiento. Este modelo de aproximación debe tomar en cuenta – para el escenario en que se modela la odometría a través de giros y avances – la forma en que varía el avance dada la distancia recorrida así como la variación angular después de un giro planeado.

La forma de modelar la incertidumbre del movimiento es plantear la posición espacial como una gaussiana que representa el espacio en el que pueda estar el robot (con una probabilidad máxima centrada en la media). Un ejemplo de movimiento en una dimensión se muestra en la figura 7.2.

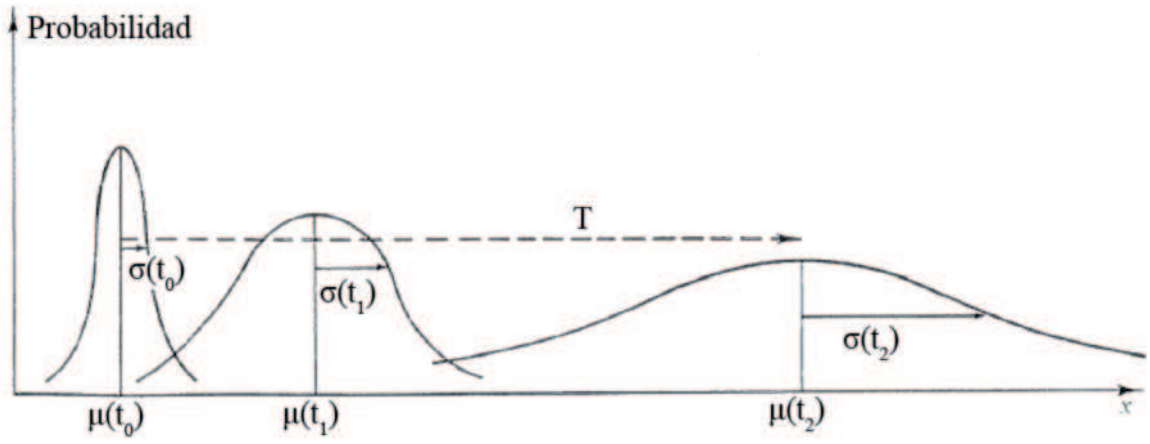

Figura 7.2: Propagación de incertidumbre en una dimensión. Conforme se avanza la incertidumbre aumenta. Esto se debe a que la distancia recorrida no es exacta, sino que es una variable aleatoria descrita por una gaussiana (centrada en μ y con una varianza σ^2). Mientras más se avance, más incierta es la posición real y por lo tanto mayor será la varianza. Basado en la figura 1.7 de [36].

La forma de representar esta propagación de incertidumbre de manera algebraica se explica a continuación. En [36] se muestra un ejemplo de propagación en una dimensión (línea recta). La forma de obtener la media $\mu(t_1)$ y la varianza $\sigma^2(t_1)$ dados $\mu(t_0)$ y $\sigma^2(t_0)$ es la siguiente:

$$\begin{aligned} \mu(t_1) &= \mu(t_0) + T \\ \sigma^2(t_1) &= \sigma^2(t_0) + \sigma_T^2 \end{aligned} \tag{7.3.1}$$

Lo que significa que si se conocen la media y varianza del momento t_0 y la traslación realizada T, así como la varianza de ésta (σ_T^2), la gaussiana de la posición en el tiempo t_1 se define por la media desplazada (sumando la traslación T) y la suma de la varianza anterior con la varianza de la traslación (que tiene sentido, ya que incorpora la incertidumbre del paso anterior con aquella del movimiento realizado).

El problema de aplicar este modelo a nuestro problema de localización es que los movimientos se realizan sobre dos dimensiones, por lo que hay que considerar que entre cada traslación hay un giro. ¿Qué sucede para un escenario donde exista movimiento en dos dimensiones?. En dos dimensiones también se puede representar la incertidumbre con una gaussiana. Una gaussiana de este tipo se muestra en la figura 7.3. μ marca la posición más probable, que se encuentra en la media probabilística de la posición. σ_a denota la desviación estándar en la dirección de máxima variabilidad, mientras que σ_b lo hace para la dirección de menor variabilidad.

Existen varios modelos adaptados a ésta propagación bidimensional [11, 24]. El objetivo de estos modelos es obtener una buena estimación de posición para incorporar como medida en el filtro de Kalman.

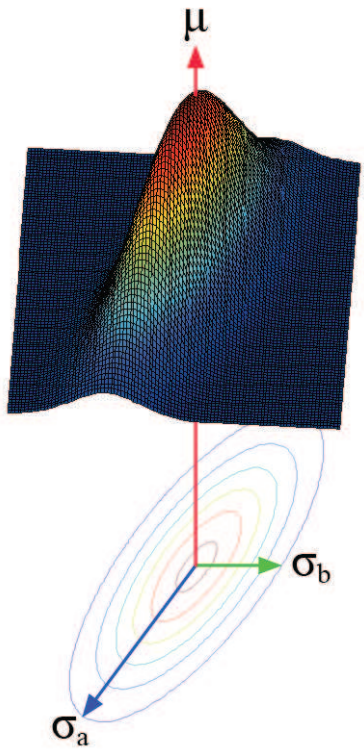

Figura 7.3: Gaussiana de dos dimensiones y sus componentes. La altura de la gaussiana representa la probabilidad de que la posición real se encuentre en ese punto. La posición se especifica por una coordenada sobre dos ejes, por lo que es una gaussiana sobre dos dimensiones. Sobre cada eje se nota una varianza diferente, y la media se fija sobre la posición más probable en el plano.

La cuestión principal es que el filtro de Kalman está asociado a un modelo lineal (una dimensión) y no es directamente aplicable a modelos con *no-linealidades* asociadas al proceso de movimiento y/o a la observación. Para esto se tiene el filtro extendido de Kalman o *extended kalman filter* (EKF), donde la restricción no es que los modelos de transición y de observación sean lineales sino que sean *diferenciables*. La diferencia entre los acercamientos a localización usando EKF se encuentra en la forma en que se expresan los modelos de observación y traslación. En esta tesis se aplica el acercamiento encontrado en [24] que pese a ser una "simplificación del fenómeno

físico" [11] resulta suficiente para el nivel de precisión de localización manejado en éste trabajo. Este modelo se explica a continuación.

7.4. Localización espacial usando el filtro extendido de Kalman

Como se explicó en la sección anterior, si se tienen dos mediciones de posición, éstas pueden combinarse para generar una estimación más precisa. En este caso, las dos mediciones son la estimación visual de pose y la estimación de pose por odometría. Para esta última se usará el modelo de traslación propuesto en [24].

En este modelo se parte del hecho de que se conoce una estimación de la posición actual en términos de la media de posición por odometría μ_o y su matriz de varianza-covarianza Σ_o, que tienen la siguiente forma:

$$\mu_o(t) = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \tag{7.4.1}$$

$$\Sigma_o = \begin{bmatrix} Var(x) & Cov(xy) & Cov(x\theta) \\ Cov(yx) & Var(y) & Cov(y\theta) \\ Cov(\theta x) & Cov(\theta y) & Var(\theta) \end{bmatrix}$$

donde $Var(a) = \sigma_a^2$ es la *varianza* de a y $Cov(ab) = Cov(ba)$ se usa para denotar la *covarianza* de a y b.

Las predicciones de pose por odometría se realizan para un movimiento compuesto de un avance seguido de un giro. La predicción de media de pose $\mu_o(t+1)$ se calcula aplicando la función de transferencia de estado $f(\mu_o(t), T)$ de la siguiente forma:

$$\mu_o(t+1) = f(\mu_o(t), T) = \mu_o(t) + T = \begin{bmatrix} x_t - \rho\cos\theta_t \\ y_t + \rho\sin\theta_t \\ \theta_t + \Delta\theta \end{bmatrix} \tag{7.4.2}$$

Estas ecuaciones (en particular la naturaleza de la transición T) se derivan de la geometría de un movimiento dentro del sistema coordenado definido para este trabajo, el cual puede verse en la figura 7.4.

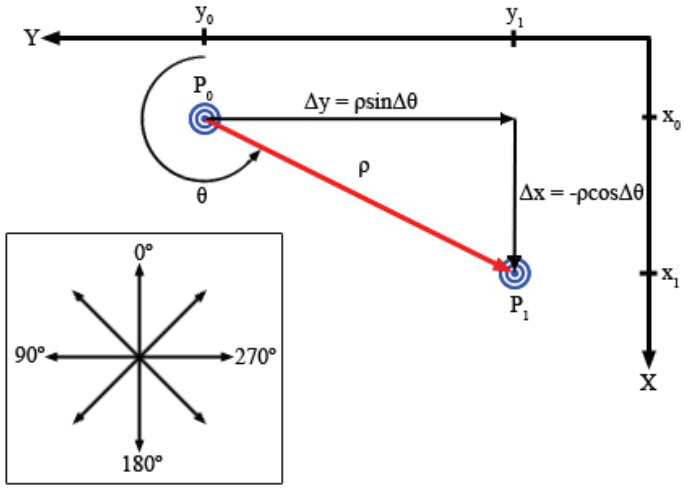

Figura 7.4: Modelo de movimiento en el sistema coordenado del experimento. El origen se sitúa en la esquina superior derecha del escenario y se inicia la orientación en la dirección negativa del eje **x**. Esto se hizo de esta manera para mantener la notación usada en [45], en la cual se basa el presente trabajo.

Dado que la función de transición de estado $f(\mu_o(t), T)$ no es una función lineal (el estado siguiente no depende linealmente del estado anterior), es necesario adaptar este escenario para poder aplicar el filtro de Kalman, que depende de un modelo de transición de estado lineal. Una solución propuesta en [12] es obtener una aproximación lineal de esta función alrededor de un punto de operación, lo cual se logra obteniendo el jacobiano de f. Si se expresa la función f como:

$$f(\mu_o(t), T) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} x_t - \rho\cos\theta \\ y_t + \rho\sin\theta \\ \theta_t + \Delta\theta \end{bmatrix} \tag{7.4.3}$$

entonces el jacobiano de f evaluado en el punto $\hat{s}$ queda como:

$$\Phi = \left[\frac{\delta f}{\delta s}(\hat{s})\right] = \begin{bmatrix} \frac{\delta f_1}{\delta x} & \frac{\delta f_1}{\delta y} & \frac{\delta f_1}{\delta \theta} \\ \frac{\delta f_2}{\delta x} & \frac{\delta f_2}{\delta y} & \frac{\delta f_2}{\delta \theta} \\ \frac{\delta f_3}{\delta x} & \frac{\delta f_3}{\delta y} & \frac{\delta f_3}{\delta \theta} \end{bmatrix} \tag{7.4.4}$$

$$= \begin{bmatrix} \frac{\delta(x_t - \rho\cos\theta)}{\delta x} & \frac{\delta(x_t - \rho\cos\theta)}{\delta y} & \frac{\delta(x_t - \rho\cos\theta)}{\delta \theta} \\ \frac{\delta(y_t + \rho\sin\theta)}{\delta x} & \frac{\delta(y_t + \rho\sin\theta)}{\delta y} & \frac{\delta(y_t + \rho\sin\theta)}{\delta \theta} \\ \frac{\delta(\theta_t + \Delta\theta)}{\delta x} & \frac{\delta(\theta_t + \Delta\theta)}{\delta y} & \frac{\delta(\theta_t + \Delta\theta)}{\delta \theta} \end{bmatrix} \tag{7.4.5}$$

$$= \begin{bmatrix} 1 & 0 & \rho\sin\theta \\ 0 & 1 & \rho\cos\theta \\ 0 & 0 & 1 \end{bmatrix} \tag{7.4.6}$$

Esto quiere decir que se obtiene la derivada de f respecto al vector $s = (x, y, \theta)^T$ cerca de un punto (evaluada en un punto). Si se piensa en dos dimensiones, esto se asemeja a usar una recta en vez de una curva. La recta se definiría como la recta tangente a un punto de la curva. Ésta recta tangente es la recta que toca el punto y que tiene una pendiente igual a la derivada de la función que describe la curva en el punto en cuestión. El uso del jacobiano se debe a que nos encontramos en un ambiente multidimensional donde estamos sacando la derivada respecto a un vector y no a una sola variable.

Esto también se puede hacer para el modelo de medición. Partiendo de una función no lineal $g(s)$, la linearización se define:

$$M = \left[\frac{\delta g}{\delta s}(\hat{s}(t+1|t))\right] \tag{7.4.7}$$

Ya teniendo esta linearización de f es posible utilizar el EKF basado en el filtrado de Kalman explicado en la sección 7.2. Los pasos de la localización con EKF se describen a continuación:

1. **Inicio**: Se cuenta con la información del punto de inicio del robot: $\hat{s}(0|0) = (x_0, y_0, \theta_0)^T$ y $P(0|0)$.

2. **Recursión**:

   a) **movimiento**: El robot se desplaza físicamente a otro punto siguiendo una traslación T y una rotación θ.

   b) **Predicción sin medida**: Se calcula la mejor estimación usando el modelo de odometría.

$$s(t+1|t) = f(\hat{s}(t|t), T) \tag{7.4.8}$$

   c) **Cálculo de** Φ:

$$\Phi = \left[\frac{\delta f}{\delta s}(\hat{s}(t|t))\right] \tag{7.4.9}$$

   d) **Obtención de la matriz de covarianza predicha**:

$$P(t+1|t) = \Phi P(t|t)\Phi^T + Q(t) \tag{7.4.10}$$

e) **Hacer una medición del ambiente**: Determinar el valor de la medición del ambiente en la nueva posición $v(s)$ y la función de medición correspondiente g.

f) **Cálculo de** M:

$$M = \left[\frac{\delta g}{\delta s}(\hat{s}(t+1|t))\right] \tag{7.4.11}$$

g) **Obtención de la ganancia de Kalman**:

$$K(t+1) = P(t+1|t)M^T(t+1)[M(t+1)P(t+1|t)M^T(t+1) + R(t+1)]^{-1} \tag{7.4.12}$$

h) **Obtención de la matriz de covarianza del estado siguiente**:

$$P(t+1|t+1) = [I - K(t+1)M(t+1)]P(t+1|t) \tag{7.4.13}$$

i) **Cálculo de la estimación del estado siguiente**:

$$\hat{s}(t+1|t+1) = \hat{s}(t+1|t) + K(t+1)[v - g(\hat{s}(t+1|t))] \tag{7.4.14}$$

La naturaleza bidimensional del escenario y el hecho de que se pueden realizar giros entre fases de avance, implica tomar en cuenta ciertos detalles importantes. Cuando se realiza un giro, el estado del robot es el mismo salvo por la coordenada de orientación, la cual es ahora $\theta + \Delta\theta$. Si a continuación se realiza un avance, la gaussiana que representa el estado anterior se debe unir con aquella que representa el movimiento, que se define por T y $Q(t)$.

Para el caso de este trabajo, cuando se realiza un avance, la determinación de la matriz de varianza–covarianza se determina para la dirección de movimiento. A ésta se le denomina $\hat{Q}(t)$ y se obtiene de la siguiente forma.

$$\hat{Q}(t) = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

donde los valores de la diagonal se calculan en línea (durante la serie de movimientos) como funciones de la distancia avanzada ρ.

Si la dirección de movimiento tiene una orientación diferente a la orientación final del estado anterior, es importante hacer la suma de la ecuación 7.4.10 en el mismo contexto. Esto se refiere a que se debe de expresar $\hat{Q}(t)$ en el mismo sistema de referencia en que se encuentre $P(t|t)$. Para esto, es necesario re-expresar $\hat{Q}(t)$ al hacer una rotación que corrija la diferencia entre los ejes coordenados de movimiento y del estado.

$$Q(t) = R\hat{Q}(t)R^T \tag{7.4.15}$$

Ahora, ¿cómo varía la gaussiana que representa el estado (pose) del robot en $2D$ dados una serie de avances y un giros?: Después de un avance, la variación a lo largo de la dirección de movimiento aumenta, ya que se incorpora la incertidumbre del movimiento a la de la posición anterior. La variación a lo largo de la dirección perpendicular al movimiento también aumenta debido a la incertidumbre de la orientación anterior aunada al giro en el movimiento. Esto puede observarse en la figura 7.5.

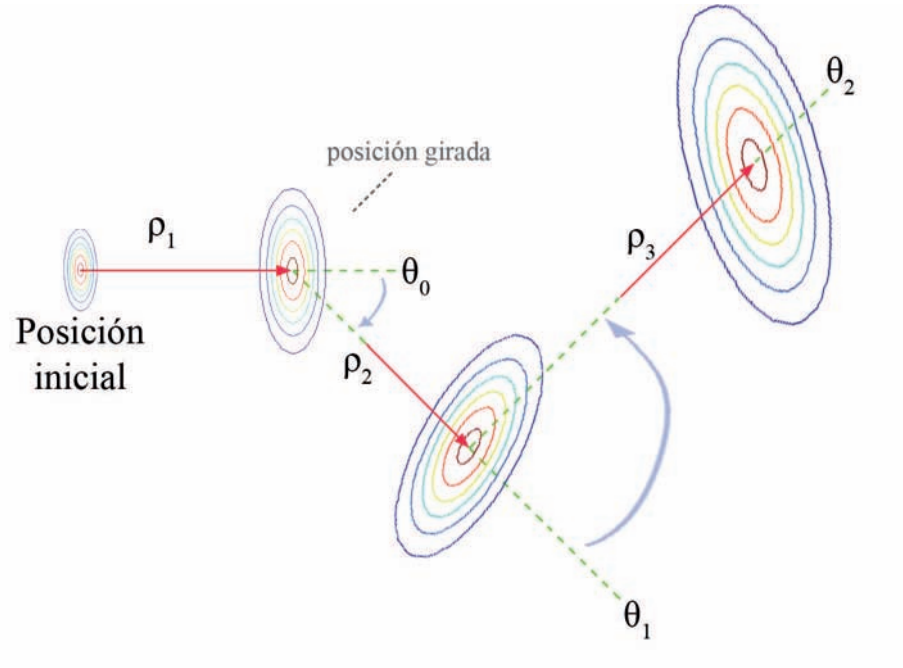

Figura 7.5: Propagación de la incertidumbre. Como se explicó en 7.2, el movimiento y giro que hace el robot se tratan como variables aleatorias, por lo que la incertidumbre de posición aumenta también en dos dimensiones. Conforme se avanza, la posición es más incierta y por lo tanto las varianzas sobre cada eje (movimiento y el perpendicular al movimiento) aumentan. La integración entre las gaussianas de movimiento (que incorpora la gaussiana del estado anterior) y visión (del estado actual) provoca que la gaussiana final se encuentre ligeramente rotada hacia la orientación anterior.

En la figura 7.5, se puede observar que las gaussianas no se encuentran dispuestas con la dirección de máxima variabilidad en un ángulo perpendicular a la dirección del movimiento. Esto se debe a que no sólo se representa la gaussiana de incertidumbre del movimiento, sino la combinación de ésta con la gaussiana de posición anterior. Esto hace que la gaussiana quede ligeramente girada hacia la posición anterior.

Es muy importante mencionar que el error de localización visual se mantiene relativamente constante y por lo tanto la localización final obtenida con el filtro de Kalman no excederá un umbral de incertidumbre. El filtro de Kalman genera una

mejor estimación que las dos en las que se basa (predicción y medición), por lo que la precisión de estimación de estado siguiente no será mayor que la de la de medición, y como esta última tiene una precisión constante, la estimación de posición usando EKF también. En la figura 7.5 se puede ver que la gaussiana incrementa de tamaño. Esto es representativo del algoritmo solo en los primeros movimientos donde la incertidumbre de localización visual excede la de la estimación por odometría. En el punto en que la incertidumbre por estimación odométrica exceda a la de localización visual, esta última actuará como un punto límite de incertidumbre. A partir de éste punto, la precisión de lo localización se mantendrá muy cerca a la definida por la localización visual, sin excederla jamás.

Para el caso en que se tenga más de una pose candidata, la incertidumbre de cada una de ellas se mantendrá por debajo del umbral mencionado arriba y pueden unirse (promediando) en dado caso de que se encuentren suficientemente cerca. En general, aquellas posiciones candidatas que se generen debido a la existencia de ambigüedad visual desaparecerán después de pocos movimientos. Los nuevos movimientos incorporarán elementos visuales que aumentarán o disminuirán el peso relativo del candidato respecto a los demás, lo que a su vez provoca que el candidato sobreviva o desaparezca.

La incorporación de los módulos explicados hasta ahora se presenta en el siguiente capítulo.

Capítulo 8

Técnica Propuesta

En este capítulo se explica la forma en que se combinan las técnicas antes descritas para conformar un sistema modular que resuelva el problema de localización por comparación de vistas conocidas.

8.1. Introducción

Los procesos descritos en las secciones anteriores sirven para comparar imágenes, generar posiciones candidatas y refinar la localización con un modelo probabilístico. Ahora, se describe la forma en que se han de integrar estas técnicas para que el robot GOLEM pueda realizar exitosamente los dos tipos de localización: localización global y seguimiento.

Los módulos se basan fuertemente en los conceptos descritos en los capítulos anteriores. Sin embargo, ciertas modificaciones y adaptaciones se llevaron a cabo para aprovechar los conceptos y mecanismos útiles y desechar los procesos innecesarios. A continuación se discute la plataforma de implementación, así como los distintos puntos de conexión entre módulos. En la figura 8.1 se muestra un esquema del sistema

implementado en su totalidad.

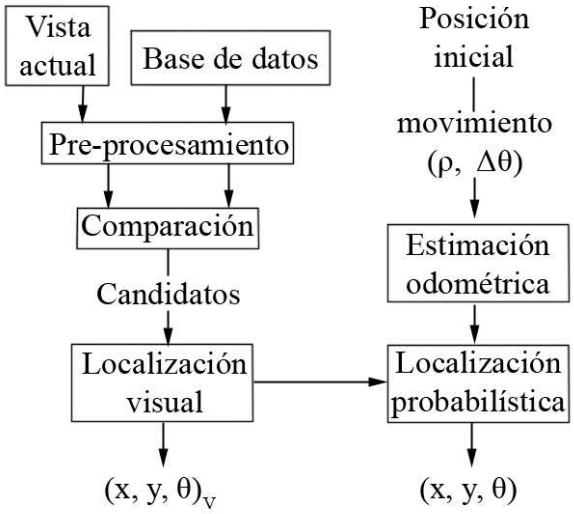

Figura 8.1: Esquema del sistema. Se muestran los pasos para generar una estimación visual de posición $(x, y, \theta)_v$ (para localización global) y la forma de combinarla con una estimación por odometría para generar la estimación final (x, y, θ) (para seguimiento de posición).

El sistema puede efectuar dos estimaciones: la primera, en la que no se conoce la posición inicial, es la localización global del robot; la segunda es rastrear la posición del robot a través de una serie de movimientos. En la figura 8.1 se puede apreciar ambos caminos. En el lado izquierdo se muestran los pasos generales para realizar una aproximación por localización visual, la cual es usada para generar una estimación de la posición del robot (la cual se representa por la terna (x, y, θ)) de la izquierda. Si a esta línea se le añade la línea de pasos de la columna de la derecha, se cuenta con el sistema completo de rastreo de posición (el cuál incorpora los pasos para localización global). Para el primer problema, los pasos son los siguientes:

1. **Pre-procesamiento**: Para todas las imágenes a ser procesadas debe aplicarse

un proceso de limpieza y detección de características.

2. **Comparación**: Se comparan las imágenes pertinentes de la base de datos contra la vista actual y se generan los nodos candidatos (parecidos) a ésta.

3. **Localización visual**: Se utiliza la secuencia de estimación de QTC seguida de geometría epipolar para generar una terna (x, y, θ).

Para el caso de realizarse un seguimiento de posición, el procedimiento aprovecha lo descrito anteriormente de la siguiente forma:

1. **Movimiento**: A partir de la posición inicial se realiza un avance y un giro.

2. **Estimación odométrica**: A partir del movimiento y la posición inicial se genera una pose probable de posición del robot.

3. **Localización probabilística**: La estimación por odometría se combina con aquella por localización visual para generar una terna (x, y, θ) de posición rastreada.

A continuación se explican cada uno de estos procesos.

8.2. Implementación

Todos los módulos se programaron en lenguaje *C*, siguiendo el paquete SIFT(C/Matlab) desarrollado por David Lowe. La implementación de los módulos y su interacción se describe a continuación.

8.2.1. Limpieza de imágenes

El primer paso consistió en realizar un procesamiento de limpieza sobre las imágenes de la base de datos ya que, debido a ciertos errores en el equipo al momento de su obtención, éstas tenían formas onduladas repetitivas que deformaban la imagen. El proceso de limpieza se inicia pasando a la imagen original por un filtro paso-bajas para eliminar el ruido de alta frecuencia presente en las *olas* de ruido. La imagen resultante I pasa entonces por un proceso de *enmascaramiento de imagen borrosa* (*unsharp masking*) [18], en el cual se realiza la siguiente acción sobre la imagen I:

$$I_{UM} = I + (I - I_{fpb})k_n \tag{8.2.1}$$

lo que quiere decir que primero se obtiene una versión borrosa de I (donde quedan las bajas frecuencias y se eliminan las altas) llamado I_{fpb} (I con filtrado paso bajas). A continuación se obtiene la diferencia entre la imagen original y la versión borrosa $(I - I_{fpb})$, dejando sólo los detalles de alta frecuencia. Por último, se suman estos detalles a la imagen original, multiplicados por una constante de "nitidez" k_n (que suele ser de entre 1 y 3) que representa el grado de acentuación de los detalles. En la figura 8.2 se muestra un ejemplo de la imagen original (donde pueden apreciarse las *olas* mencionadas y un par de líneas que están agregadas a la imagen por defectos en la su captura) y en la imagen 8.3 el proceso para obtener la imagen filtrada. En (a) se observa la imagen original, en (b) la imagen pasada por un filtro paso-bajas y en (c) la imagen después del filtrado de enmascaramiento de imagen borrosa.

Pese a que a simple vista el cambio no es muy notorio, la imagen resultante I_{UM} tiene un resalte de nitidez, y la diferencia con la imagen original de la base de datos es que se han eliminado las *olas* de ruido sin eliminar detalles como bordes y áreas

Figura 8.2: Imagen original con errores

de alto contraste. En pruebas de comparación de imágenes, se encontró que tras esta precaución, se encontraba una mayor cantidad de correspondencias (de alrededor de 30 % para la cantidad límite de correspondencias para utilizar el algoritmo de los ocho puntos).

8.2.2. SIFT y GTM

Después de la limpieza, se cambió el formato de cada imagen para terminar con la base de datos en formato ".pgm"(escala de grises), que es el formato de entrada del programa de SIFT. Así, se procesó la base de datos para tener otra base de datos equivalente con los *descriptores* de cada imagen.

En el momento de comparar una vista actual de GOLEM, lo primero es obtener el descriptor equivalente mediante su procesamiento a través del programa SIFT. En este momento se usa el proceso de comparación de descriptores contra los descriptores

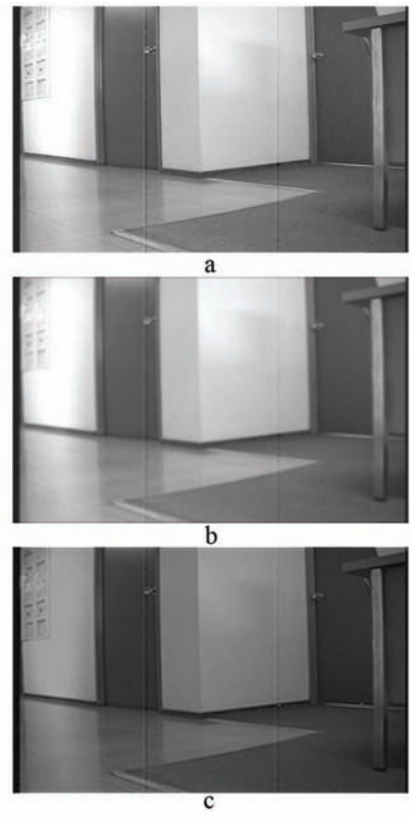

Figura 8.3: Limpieza por enmascaramiento de imagen borrosa. En (a) se muestra la vista original. En (b) se muestra el resultado de pasar la imagen por un filtrado gaussiano. Aquí se eliminan muchos de los artefactos y errores añadidos a la imagen por el mecanismo de adquisición de la imagen. En (c) se muestra el resultado de realizar el enmascaramiento de imagen borrosa, efectivamente aumentando la nitidez de las características de la escena.

de cada uno de los candidatos, que en el caso de localización global es toda la base de datos de descriptores, y en el caso de localización local es contra aquellos descriptores pertenecientes a los candidatos cercanos. En este punto se puede decidir la "precisión" de la comparación. Esta disyuntiva se debe a que los puntos críticos a comparar se encuentran distribuidos no sólo en el espacio de la imagen sino en la escala. Así, si se desea mayor precisión se incluyen más escalas, con la desventaja de que se incurre en un mayor tiempo de procesamiento. Para menor precisión y mayor velocidad, el

recurso que se tomó fue comparar sólo usando las escalas “altas”, donde se encuentra encapsulada la información general de las imágenes.

El resultado de este paso es una estructura que contiene, para cada par de imágenes con similitudes, una lista de cuartetas que representan las coordenadas en pixeles de sus correspondencias. El formato de cada cuarteta es (r_1, c_1, r_2, c_2) (renglón y columna de la primera imagen contra renglón y columna de la segunda). Esta misma lista se suministra al algoritmo de GTM que elimina las falsas correspondencias. Un ejemplo de esto se ve en la figura 8.4.

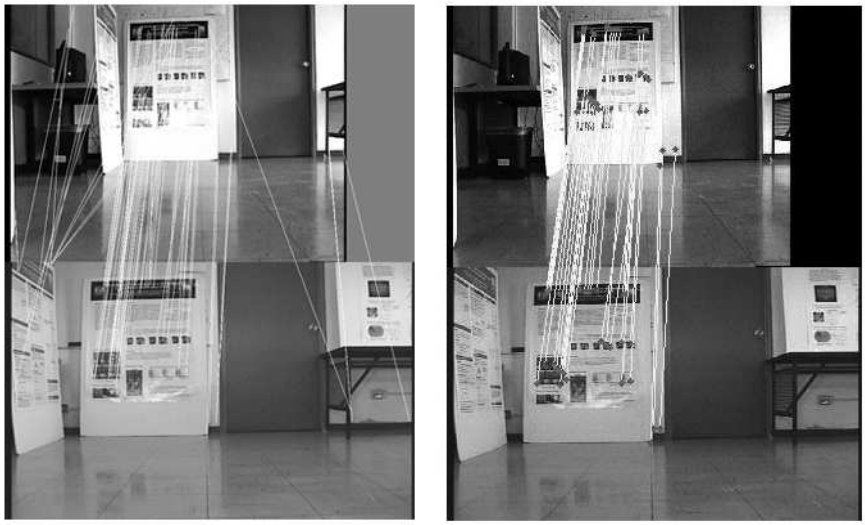

Figura 8.4: Obtención de correspondencias. an (a) se observan las correspondencias obtenidas iniciales y en (b) aquellas que quedan después del procesamiento con GTM.

8.2.3. Localización visual

En este punto la pregunta es si se cuenta con suficientes correspondencias para aprovechar la geometría epipolar (GE) o si la estimación se realizará puramente con

QTC. Como se comentó en el capítulo 6, para obtener una precisión equivalente a la de QTC, el algoritmo de localización por geometría epipolar necesita varios cientos de correspondencias [19]. Esta situación se presenta muy pocas veces a lo largo del recorrido del robot. Esto se debió a la escasa cantidad de características en el escenario del experimento. Para realizar pruebas en un ambiente con más características se probó un ambiente alternativo que se explicará en el siguiente capítulo.

Como se comentó en el capítulo 6, la traslación estimada es hasta un factor de escala, por lo que ésta debe de aproximarse por otro método. Una posibilidad es el uso exclusivo de QTC como método de aproximación de posición, lo cual se explicó en el capítulo 5. La otra posibilidad es utilizar varias imágenes de referencia para así obtener –usando la técnica de geometría epipolar– varios vectores de traslación $\mathbf{T}$. Estos pueden sobreponerse y así ubicar una zona de intersección donde la posición del robot esté lo más cercana posible a cada una de las predicciones de traslación. Un factor de riesgo al usar este método es que pequeñas alteraciones en los vectores $\mathbf{T}$ causan cambios drásticos en la zona de intersección (la cual no siembre se encuentra en una zona particular, sino en varios puntos lejanos entre sí). Este factor llevó a la decisión de utilizar solamente la técnica de QTC para la estimación de posición.

La estimación de orientación, en cambio, puede ser aproximada tanto por QTC como por GE (en caso de contar con suficientes correspondencias mínimas). Así, se generan dos orientaciones posibles y se promedian para generar una aproximación que combine ambas técnicas.

Para el caso en que no pueda estimarse la pose de una vista, se realiza una fase de movimiento seguro (un movimiento que no arriesgue la integridad del robot) seguida de una fase de localización. Para este escenario se decidió realizar un giro de 30º

seguido de otra captura de imagen y la repetición del proceso de localización visual. En caso de cubrir 360º con giros, el proceso se detiene y se declara un error de localización.

8.2.4. Localización probabilística

En éste momento, contamos con una o varias posiciones probables (ya que QTC puede entregar más de un centroide general). Se sigue la idea del filtro de Kalman de especificar la posición por fases de "predicción de movimiento" y "corrección por medición".

En este punto es importante recuperar la diferencia entre el problema de localización global y localización local. En el primero, no se conoce una posición inicial y en el segundo se cuenta con una o varias posiciones iniciales probables. Para la localización global, la fase de comparación y generación de candidatos explicada en la sección 8.2.3 se hace sobre toda la base de datos de imágenes. En cambio, para la localización local, la comparación se realiza sobre la vecindad de cada una de las posiciones iniciales donde probablemente se encuentra el robot. El proceso descrito a continuación se aplica a la unión de dos estimaciones, la visual y la odométrica. De no contar con una de éstas, el filtrado de Kalman es inaplicable y se utiliza como predicción final aquella estimación con la que se cuente. Este sería el caso para el problema de localización global, donde no se tendría una estimación odométrica por falta de posición inicial a partir de la cual realizar los movimientos. También es importante mencionar que en el caso en que no se pueda generar una estimación por localización visual, la única presente será la odométrica y por lo tanto será usada como la estimación final de pose. Así, el siguiente proceso es aplicable a la situación donde se

tenga tanto una estimación visual como una odométrica, quedando entonces restringido al problema de localización local. No obstante, luego de una primera aproximación en el caso de localización global, se pasa inmediatamente al problema local y por lo tanto se puede realizar una estimación probabilística.

Luego de un movimiento se calcula, usando el modelo de odometría descrito en el capítulo anterior, la pose probable del robot. A continuación se captura una imagen de la vista actual del robot para ser usada en la fase localización visual descrita arriba. La posición que brinde esta fase constituye la estimación por medición. Ésta se utiliza para corregir (adaptar) la posición estimada por odometría. El resultado es una estimación de pose que incorpora las dos estimaciones (predicción y medición) y que es más precisa que cada una de forma independiente.

Si en la fase de localización visual se cuenta con más de un candidato, o si se parte de varias posibles posiciones iniciales, se hace una estimación por filtrado de Kalman por cada una de los posibles candidatos. Este caso se presenta cuando no se desea usar el super-cúmulo o el promedio de los cúmulos finales de la fase de localización visual. La naturaleza del proceso es tal que aquellas poses con una gran diferencia entre predicción de pose y medición de pose tendrán un peso de probabilidad cada vez menor. Se fija un umbral de probabilidad de tal forma que si un candidato tiene un menor peso que éste, se le considera inconsecuente y se elimina del grupo de posibles candidatos.

Un esquema completo de los pasos de localización, que incluye cada módulo mencionado, se muestra en la imagen 8.5. En primer lugar, se genera una estimación de posición por odometría para cada candidato de la fase actual. A continuación se hace la comparación entre la vista actual y la base de datos de imágenes. Esto nos permite

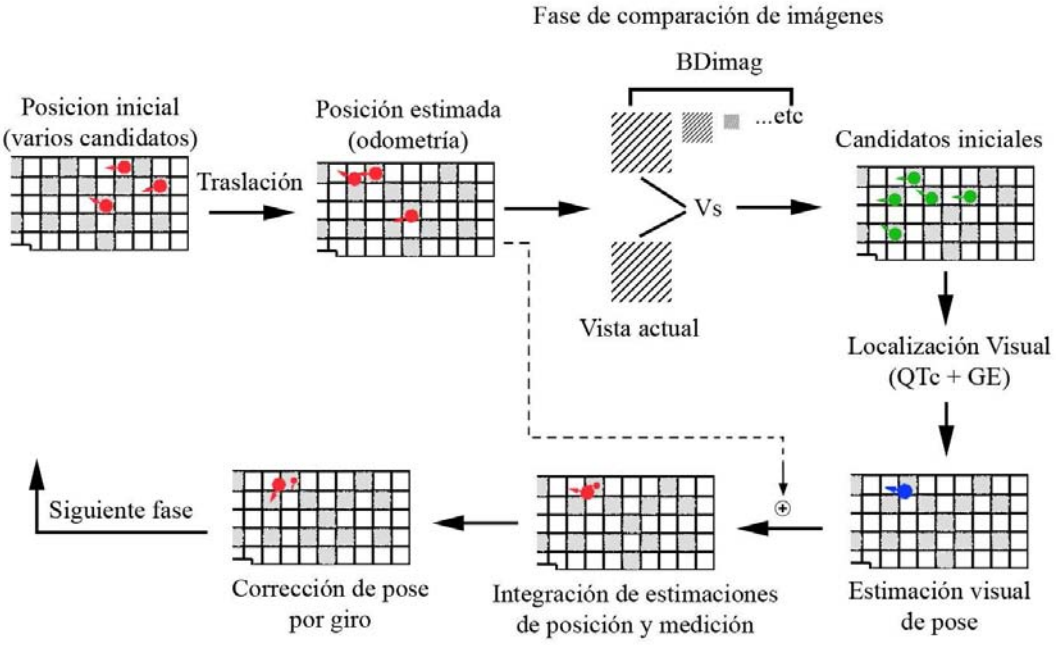

Figura 8.5: Esquema del proceso implementado. Este proceso se aplica al problema de localización local (en este caso con varias posiciones iniciales probables). La secuencia es la siguiente: a partir del conjunto de posiciones iniciales se realiza un movimiento. Se obtienen las posiciones *movidas* desde este punto se generan dos estimaciones, una odométrica y otra visual (siendo ésta la que se detalla con los pasos de comparación y localización visual). Las dos estimaciones se integran usando el filtro de Kalman. Al final se incluye el giro final del que se compone el movimiento.

generar una serie de candidatos situados sobre las posiciones de nodos y orientaciones que registraron correspondencias con la vista del robot. Si ésta es la primera fase de comparación (localización global), se busca contra toda la base de datos, de lo contrario, la búsqueda se hace en la vecindad de cada candidato.

En el siguiente paso se realiza una estimación de pose por medición. La pose estimada hasta ahora, y que se basa en la la fase anterior se incorpora a la medición y se genera una serie de candidatos probables. En la figura, se muestran los dos candidatos cuyo peso de probabilidad superó el umbral mínimo (el candidato de mayor

tamaño representa mayor probabilidad, mientras que el segundo – el pequeño – es un candidato con baja probabilidad pero por encima del umbral). Hay que recordar que aunado a cada media de posición se encuentra una matriz de varianza – covarianza que especifica a la gaussiana que representa la precisión de la estimación. Luego de un giro, ésta matriz, y la gaussiana correspondiente se modifica para incluir la incertidumbre debida al giro. Es importante recordar que pese al giro, la posición del robot no cambia, solo su orientación. Esto es relevante ya que no se añade la incertidumbre hasta el momento de realizar el siguiente avance.

8.3. Navegación

Pese a que éste no es un objetivo principal de este trabajo, una consecuencia del método de localización es la habilidad de realizar una navegación (sin obstáculos) dentro del escenario conocido.

En el caso en que se tenga una ruta planeada o un destino general, los movimientos realizados en la fase de movimiento pueden ser influenciados por la ruta a seguir. Esto quiere decir que los giros y avances se escogerán de acuerdo al siguiente mejor acercamiento al destino.

Una posibilidad para el diseño de la ruta, es escoger la serie de nodos que conecta la posición actual a la posición destino, y seguirlos combinando fases de localización y fases de movimiento por odometría. La forma de escoger los nodos sería usando el algoritmo de Dijkstra, donde se detectan la serie de nodos en un grafo conectado para llegar de un nodo inicial a uno final con el menor costo posible. Aquí, el costo se asigna por las distancias entre nodos. Las fases de localización podrán ser, dependiendo del escenario, de localización global o de seguimiento, en cuyo caso sólo se irán propagando

las posiciones probables con actualizaciones de localización probabilística. Así, la navegación dentro del espacio conocido se puede basar en aproximaciones sucesivas a una serie de nodos conocidos. Este escenario puede resolverse, incluso, bajo el esquema sencillo de localización probabilística presentado en este trabajo.

Capítulo 9

Pruebas y Resultados

En este capítulo se explican los experimentos realizados y los escenarios específicos en los que estos se desarrollaron. También se hace un recuento de los resultados obtenidos en cada módulo del procesamiento así como sus parámetros operativos.

9.1. Introducción

La serie de experimentos realizados se basó en una arquitectura modular, por lo que se probó la efectividad de cada módulo por separado para luego combinarlos de forma óptima. La serie de pruebas modulares fueron las siguientes:

1. **Comparación de imágenes**
2. **Búsqueda de candidatos**
3. **Estimación de posición por QTC**
4. **Estimación de posición por geometría epipolar**
5. **Estimación final de pose**

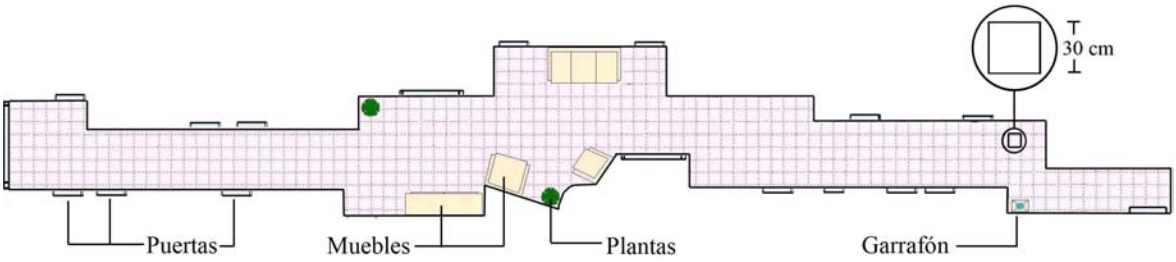

Figura 9.1: Plano del pasillo

6. **Rastreo de posición**

Antes de exponer las pruebas y resultados se explican los escenarios en los cuales se probó el sistema.

9.2. Escenario original de pruebas

Se eligió como espacio de localización el pasillo del cuarto piso del edificio del IIMAS, en la UNAM. Un plano de este espacio se muestra en la figura 9.1. También se desarrolló una base de datos alternativa, que abarca uno de los laboratorios y parte del pasillo mencionado arriba. Se explica la base de datos basada en el pasillo a continuación.

Las imágenes que funcionan como referencia se capturaron en [45] y se tomaron de tal forma que cubrieran la mayor cantidad de vistas posibles del espacio. El área total del pasillo se segmentó para formar una cuadrícula de posiciones de referencia. Cada cuadrado tiene 30 centímetros de lado y corresponde a una baldosa del piso del departamento. Se eligieron algunos cuadros especiales o *nodos*, desde donde se capturaron las imágenes.

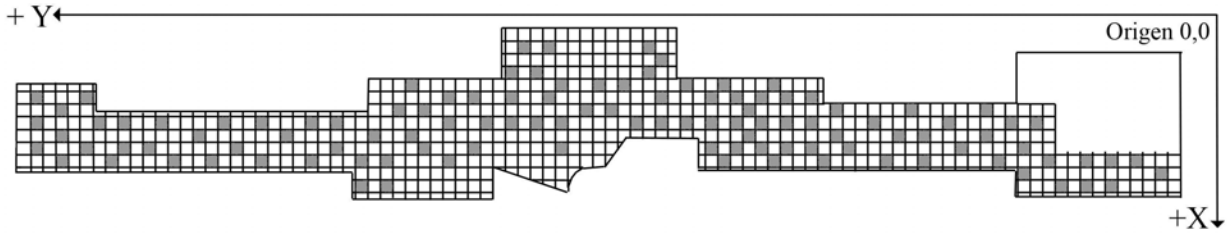

Figura 9.2: Nodos y sistema coordenado del espacio. Los cuadros sombreados corresponden a los nodos desde donde se obtienen las ocho vistas.

En la figura 9.2 se puede observar la disposición de los nodos (zonas sombreadas) en la cuadrícula, y el esquema de referencia utilizado. A cada nodo se le asigna un par de coordenadas (x, y) en relación al origen de coordenadas mostrado en la figura 9.2.

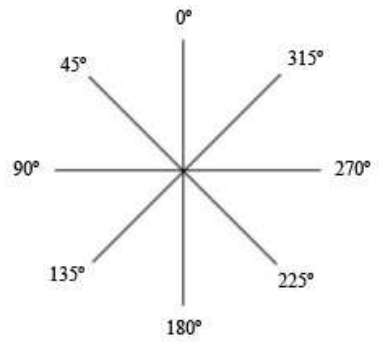

Figura 9.3: Sistema orientado

Desde cada nodo se tomaron ocho puntos de vista, indicados por el sistema orientado mostrado en la figura 9.3. Como puede verse, los puntos de vista cubren los 360º alrededor del nodo, partiendo desde la dirección indicada por 0º y girando en contra de las manecillas del reloj 45º hasta aquella indicada por 315º. Cabe mencionar que la cámara utilizada por el robot móvil tiene un *campo visual* de 50º, por lo que el espaciamiento entre puntos de vista es adecuado. El ángulo de orientación del robot

constituye la tercer coordenada de la pose del robot (x, y, θ).

9.3. GOLEM

El robot, llamado GOLEM es un robot móvil tipo "Magellan Pro Compact" que funciona bajo la infraestructura iROBOT mobility [34, 39]. Este robot puede verse en la figura 9.4

Figura 9.4: Robot Golem

GOLEM cuenta con 16 sonares y sensores infrarojos y de contacto dispuestos alrededor de su chasis. Se mueve por la propulsión de dos ruedas motorizadas laterales y se mantiene equilibrado gracias a un par de ruedas no motorizadas ubicadas al frente y detrás del robot. También tiene una cámara tipo CCD montada en la parte superior del robot. GOLEM cuenta con una conexión inalámbriaca mediante la cual recibe comandos de movimiento y a través de la cual éste envía las imágenes capturadas por la cámara CCD.

El robot tiene 40.6 centímetros de radio y la cámara está acomodada aproximadamente a 42 centímetros de altura, con un eje óptico aproximadamente horizontal.

9.4. Comparación de imágenes

Los experimentos realizados para este módulo se apuntaron a encontrar la mayor cantidad de características y correspondencias basadas en el descriptor SIFT. Los diferentes escenarios de comparación se diseñaron sobre el espacio descrito en la figura 9.1. Las variables de análisis fueron la cantidad de iluminación, el tipo de escena observada y la precisión de comparación.

En esta serie de experimentos se encontró que el escenario utilizado era inadecuado para el experimento. Las características detectadas en el escenario descrito en la figura 9.1 se conforman, en su mayoría, a detalles de textura e iluminación de la escena. La textura de la mayoría del escenario se conforma de patrones rugosos de un mismo color definidos por las sombras generadas por diferentes focos luminosos (muros blancos y rugosos). Al variar la iluminación o el ángulo de observación, las características obtenidas de este tipo de texturas varía considerablemente. Esto provoca que estas características sean particulares de un punto de vista (y contexto de iluminación) muy específico y por lo tanto resultan inaplicables en un proceso de comparación. Esto puede observarse en 9.5.

En la figura 9.5(a) se presenta una vista tomada durante el recorrido del robot, y en 9.5(b) se muestra la imagen de referencia más parecida dentro de la base de datos. Como puede verse, la mayoría de las características a comparar se encuentran sobre la textura de la pared del pasillo. El resultado de comparar este par de imágenes se muestra en 9.6. Como puede verse, sólo se detectan siete correspondencias iniciales (luego de GTM quedan tres). Este ejemplo es típico del primer escenario, en el cual existe una baja detección de correspondencias y por lo tanto menos información de la escena. Esto, como se mencionó en el capítulo 6, hace más sensible la técnica de

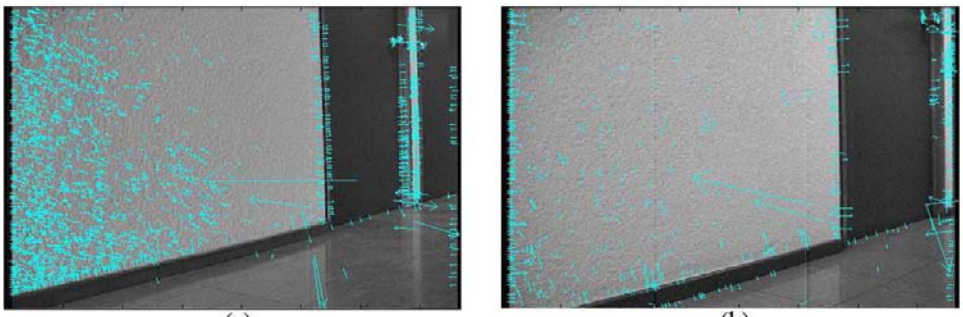

Figura 9.5: Tipos de puntos característicos en el escenario 1. Se muestran los puntos característicos de una vista de recorrido (a) y la imagen de referencia más parecida (b) dentro del escenario 1.

geometría epipolar al ruido y por lo tanto menos confiable.

Se decidió crear una base de datos alternativa. Este escenario, denominado el escenario 2 se compuso de un laboratorio y parte del pasillo descrito en la figura 9.1. Este nuevo escenario se muestra en la figura 9.7.

Para asegurar la existencia de más características relevantes (no tan dependientes de la iluminación y el punto de vista), se rodeó el espacio del laboratorio con posters de proyectos del departamento de computación del IIMAS. El procesamiento SIFT rindió muchas más características relevantes que en el escenario 1. Incluso al variar considerablemente la iluminación, la cantidad de características relevantes detectadas se mantuvo muy por encima de los resultados obtenidos anteriormente. Asimismo, la cantidad de correspondencias obtenidas aumentó debido a que está directamente relacionado con la cantidad de elementos a comparar. En 9.9(a) y (b) se puede observar el tipo de características detectadas en el escenario 2 para el par de imágenes similares (vista de recorrido y referencia más parecida).

Las correspondencias obtenidas para el par de imágenes del escenario 2 se observa

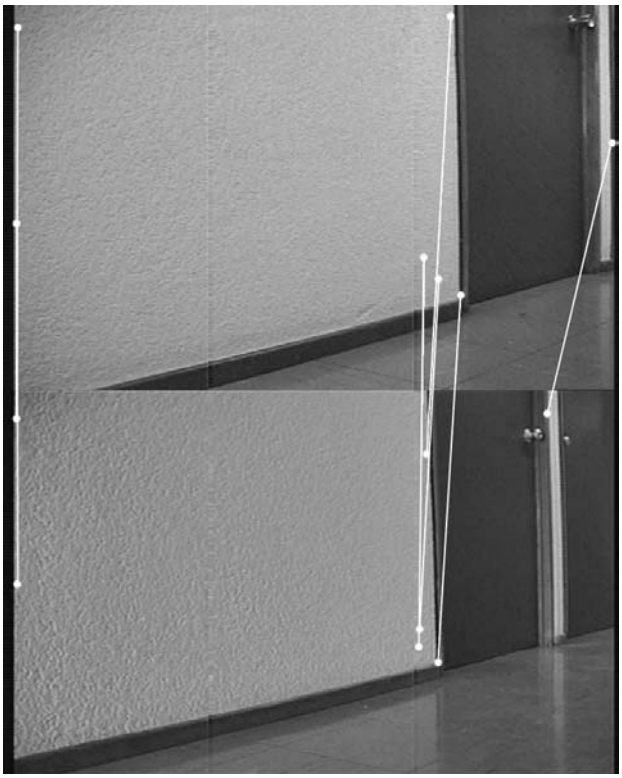

Figura 9.6: Correspondencias en el escenario 1

en 9.9. En este caso se obtuvieron 235 correspondencias iniciales y 159 después de GTM. Este tipo de resultados provee información más cuantiosa y robusta con la cual trabajar.

En el escenario 1, el promedio de correspondencias correctas para imágenes muy parecidas era del orden de 10. Éste mismo número para el escenario 2 se incrementó a más de 100 correspondencias. Este escenario no es mejor, sino más rico en características relevantes, y por lo tanto en correspondencias. Fue por esto que los siguientes experimentos se realizaron en este contexto.

Otro punto a mencionar es que el método de comparación de imágenes seleccionado fue el método exhaustivo. En general, este método es el más lento pero también el

Figura 9.7: Plano del Laboratorio

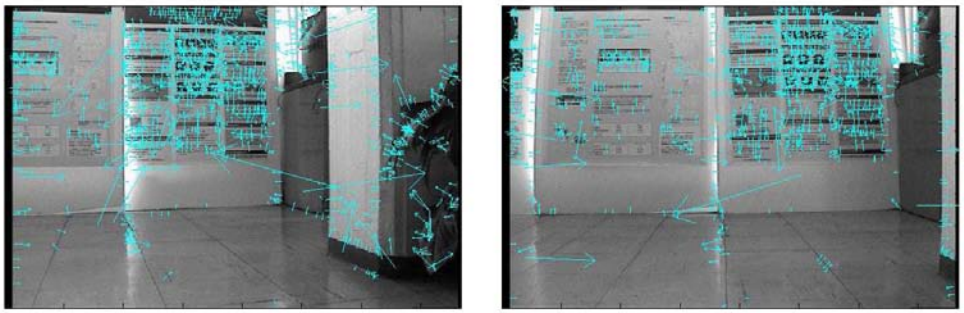

Figura 9.8: Tipos de puntos característicos en el escenario 2. Se muestran los puntos característicos de una vista de recorrido (a) y la imagen de referencia más parecida (b), dentro del escenario 2.

más preciso. Si este método se sustituye por el sugerido por Lowe en [32] (BBF), la velocidad de comparación incrementaría considerablemente. El costo de este cambio, según lo reportado por Lowe para cantidades similares de correspondencias, es de una pérdida de precisión de 5 %, lo cual no alteraría considerablemente los resultados de estimación basada en cúmulos pero sí representaría una sensibilidad adicional a la ya endeble técnica de estimación por geometría epipolar.

Todas las pruebas se realizaron en el interior de un laboratorio usando luz artificial

Figura 9.9: Correspondencias en el escenario 2

de neón. Las imágenes de la base de datos de referencia se tomaron en un espacio relativamente corto de tiempo (una hora y media) y por lo tanto aprovechan la luz de día con aproximadamente el mismo brillo. Las imágenes de prueba, en cambio, se tomaron con luz artificial encendida pero con luz de día variable. Se intentó mantener la esencia de la escena intacta en lo que respecta a los objetos como sillas y posters, pero pequeños cambios fueron inevitables. Estos experimentos, sus parámetros operativos y los resultados obtenidos en cada uno se describen a continuación.

9.5. Búsqueda de candidatos

Las variables analizadas en este módulo fueron la cantidad de características a relacionar y la escala a la cual se realizó la comparación.

Como se explicó en el capítulo anterior, la precisión de comparación se refiere a la cantidad de características usadas en la comparación, la cual depende en las escalas incluidas en la búsqueda. Se realizaron varias corridas de comparación de vistas contra toda la base de datos y se determinó que, para generar rápidamente una serie de candidatos, era conveniente comparar con precisión media. Una vez se tenga un grupo reducido de candidatos es posible continuar con una mayor precisión. En el cuadro 9.1 se muestra el tiempo de comparación de la imagen 9.9(a) contra la base de datos de referencia del laboratorio. Se realizó una corrida con un nivel distinto de precisión (determinado por el umbral *umb*). Una comparación con precisión de $umb = x$ se refiere a tomar en cuenta sólo los puntos característicos a escalas iguales o mayores a x, donde x es la escala relativa a la imagen original. Es importante recordar que dado el proceso de obtención de características descrito en el capítulo 4, existen puntos característicos a escalas menores a $umb = 1$ (que son los "detalles de la escena") y mayores a $umb = 1$ (que son los puntos característicos de frecuencias bajas).

Cuadro 9.1: Localización por escalas

umb	num corr	pose real	pose est	$\mid e\rho \mid$ (cm)	$\mid e\theta \mid$ (gra)	t (s)
0	2241	195,325,255º	192,350,284º	25.2 cm	29º	568 s
1	2052	195,325,255º	193,350,279º	25.1 cm	24º	397 s
2	867	195,325,255º	187,333,269º	11.3 cm	14º	28 s
3	436	195,325,255º	185,322,260º	10.4 cm	5º	6 s
4	224	195,325,255º	172,284,270º	47.0 cm	15º	2 s

En el cuadro 9.1, *umb* es la escala probada; *num corr* es el total de correspondencias halladas; *pose real* y *pose est* se refieren a la pose de la vista comparada y la posición estimada por el sistema respectivamente (estas poses se representan por tres números que son sus coordenadas de x, y y θ); $\mid e\rho \mid$ se refiere al error de localización espacial (en centímetros) y $\mid e\theta \mid$ al error angular (en grados); t representa el tiempo de ejecución de la comparación (en segundos). En la figura 9.10 se puede ver la representación gráfica de los datos de el cuadro 9.1.

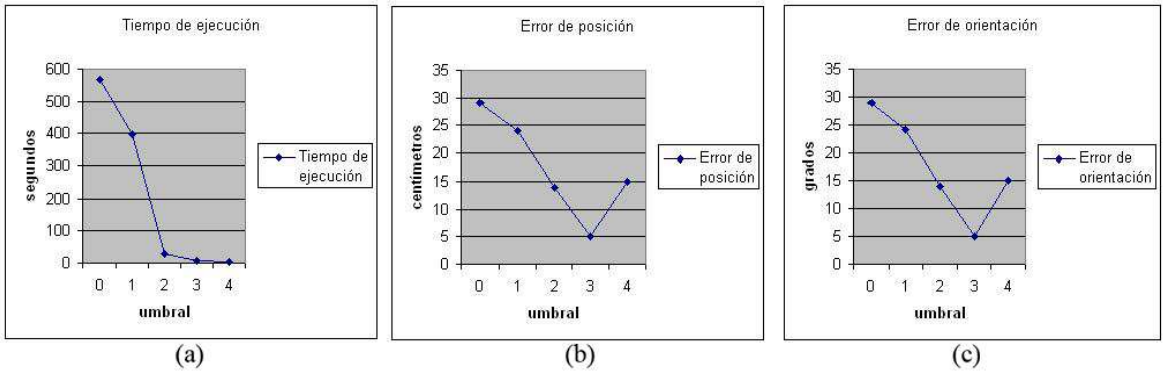

Figura 9.10: Localización por escala por escala. En (a) se muestra el tiempo de procesamiento de la localización por umbral de búsqueda para una imagen. En (b) se muestra el error entre la posición real y la estimada por escala de búsqueda. En (c) se muestra el error de orientación entre la posición real y la estimada por escala de búsqueda.

Como puede observarse en la figura 9.10 (a), el tiempo de ejecución del proceso de comparación disminuye drásticamente con el ajuste del umbral de escala (a lo que se le llama precisión de comparación). Esto se deba a que la mayor cantidad de características de una imagen (más del 95 %) se encontraba en las escalas bajas, que son los detalles encontrados en las altas frecuencias. También puede observarse, en 9.10 (b) y 9.10 (c) que el error de localización no varía considerablemente, siendo incluso más preciso alrededor de una escala de comparación $umb = 3$. Esto se debe

a que las características más robustas o relevantes son, en su mayoría, aquellas que se encuentran ligadas a los componentes físicos de la escena y no a los detalles de textura (que son muy dependientes de la iluminación). En cambio, para comparaciones precisas –umb en el intervalo $[0-2)$ – se incluyen correspondencias de detalles similares pero no idénticos que se dan a escalas muy bajas. Esto provoca la inclusión al conjunto de candidatos a vistas que de otro modo no serían consideradas "parecidas". Esta inclusión de candidatos marginalmente parecidos altera la obtención de la estimación de posición (en muchos casos de manera negativa) al agregar candidatos. Es entonces prudente hacer un primer acercamiento con una comparación a precisión media ($umb = 3$) con el propósito de generar rápidamente candidatos parecidos. Esta comparación se realiza sobre las escalas altas que representan los detalles gruesos y características de baja frecuencia. La ventaja de realizar este acercamiento previo es, como se puede ver en el cuadro 9.1, un incremento de velocidad de cerca del 400 % del tiempo de comparación sin umbralización de escalas. La circunstancia mostrada en el cuadro 9.1 y la figura 9.10 es representativa del resto de las imágenes (no solo de esa vista en particular).

Debido a lo anterior, la búsqueda de candidatos se realizaba primero sobre las bajas frecuencias a una alta velocidad, y se refinaba en una segunda pasada (solo sobre los candidatos identificados para las bajas escalas) realizando la comparación sobre todas las escalas. Esos pasos aseguraron un incremento significativo en la velocidad de procesamiento a cambio de una pérdida mínima de precisión.

9.6. Estimación de posición por QTC

Como se comentó anteriormente, el propósito de QTC es detectar centroides de probabilidad de presencia del robot dentro de un cúmulo de candidatos. Este proceso se basa en la posición de cada candidato, así como el peso de éste. El peso de un candidato es la cantidad de correspondencias que presenta en relación a la vista actual. El peso se plantea como la cantidad de correspondencias normalizada.

Para estos experimentos se capturaron cuarenta imágenes de prueba desde posiciones elegidas aleatoriamente dentro del escenario de localización. La pose de cada vista de prueba se registró con cuidado con el propósito de ser usada como referencia de precisión. Las pruebas realizadas se muestran en el cuadro 9.2.

En el cuadro 9.2, *posición* es el número de la prueba; *pose real* es y *pose est* se refieren a la pose de la vista comparada y la posición estimada por el sistema respectivamente; $\mid e\rho \mid$ se refiere al valor absoluto del error de localización espacial (en centímetros) y $\mid e\theta \mid$ al valor absoluto del error angular (en grados); t representa el tiempo de ejecución de la comparación (en segundos) y *loc* es una bandera que denota el tipo de localización lograda. Para $loc = 0$, el sistema no pudo encontrar la posición del robot. Para $loc = 1$, el sistema encontró un solo centroide como resultado de QTC. Para $loc \in (0, 1)$ (sin incluir 0 y 1), se encontró más de un centroide y la posición final se obtuvo a partir de una combinación del mayor centroide y los otros candidatos. Esta posición se estima como un super-centroide a partir de los pesos de cada centroide final. El valor de *loc* se define a partir de la similitud y cercanía de centroides, siendo un valor cercano a 1 si son éstos muy parecidos y cercano a 0 si hay diferencias de orientación y posición. La penalización de puntaje es proporcional a la diferencia entre el centroide principal y los demás candidatos.

Cuadro 9.2: Pruebas de Localización con QTC

posición	pose real	pose est	$\mid e\rho \mid$ (cm)	$\mid e\theta \mid$ (gra)	t (s)	loc
1	91,86,45º	–	–	–	0	0
2	85,80,306º	60,60,270º	32.0 cm	36º	7 s	1
3	232,140,162º	240,90,180º	50.6 cm	18º	7 s	1
4	169,176,284º	219,141,313º	61.0 cm	29º	29 s	1
5	152,174,72º	153,133,90º	41.0 cm	18º	13 s	1
6	182,116,176º	240,90,180º	63.6 cm	4º	14 s	1
7	144,118,293º	143,104,302º	14.0 cm	9º	23 s	0.97
8	102,101,18º	–	–	–	0 s	0
9	240,90,81º	213,81,90º	28.5 cm	9º	19 s	1
10	170,135,95º	200,107,90º	41.0 cm	5º	12 s	1
11	185,439,158º	–	–	–	0 s	0
12	185,375,0º	182,376,0º	3.2 cm	0º	8 s	0.89
13	259,429,9º	292,361,14º	75.6 cm	5º	35 s	0.9
14	205,368,167º	224,316,174º	55.4 cm	7º	32 s	0.93
15	260,410,194º	276,340,180º	71.8 cm	14º	25 s	0.96
16	96,385,198º	164,380,188º	68.2 cm	10º	30 s	0.96
17	89,439,72º	60, 450,90º	31.0 cm	18º	13 s	1
18	52,373,144º	150,390,135º	99.5 cm	9º	10 s	1
19	195,325,257º	160,350,255º	43.0 cm	2º	42 s	0.99
20	102,340,203º	268,451,225º	199.7 cm	22º	16 s	0.65
21	369,107,185º	–	–	–	5 s	0
22	397,179,338º	419,181,325º	22.1 cm	13º	11 s	1
23	417,168,180º	510,150,135º	94.7 cm	45º	32 s	1
24	475,155,318º	510,156,315º	35.0 cm	3º	24 s	1
25	389,108,153º	–	–	–	4 s	0
26	470,115,149º	–	–	–	4 s	0
27	460,151,320º	487,153,314º	27.1 cm	6º	24 s	0.94
28	539,173,279º	540,210,270º	37.0 cm	9º	36 s	1
29	402,173,99º	381,193,98º	29.0 cm	1º	32 s	0.74
30	374,137,135º	404,135,135º	30.1 cm	0º	49 s	1
31	553,323,77º	–	–	–	4 s	0
32	414,412,50º	412,361,85º	51.0 cm	35	37 s	0.99
33	387,409,108º	–	–	–	2 s	0
34	322,440,59º	–	–	–	2 s	0
35	404,316,293º	395,329,282º	15.8 cm	11º	30 s	0.99
36	410,403,9º	388,308,17º	97.5 cm	8º	27 s	0.79
37	558,419,356º	493,353,0º	92.6 cm	4º	31 s	0.62
38	367,315,117º	417,350,115º	61.0 cm	2º	12 s	1
39	330,240,14º	430,291,17º	112.3 cm	3º	25 s	0.89
40	330,390,225º	325,402,231º	13.0 cm	6º	39 s	0.94

De las 40 posiciones iniciales, 31 fueron localizadas al primer intento (aquellas que en el cuadro 9.2 están marcadas con un valor de *loc* diferente de 0). El valor de *loc* se utilizó como un indicador de certeza de localización y se determinó que para valores menores a $loc = 0.70$ era conveniente realizar un ciclo de verificación que consistía en girar al robot 30º y repetir la búsqueda localmente (en la vecindad de la posición estimada). Esto mejoró la estimación de posición para las posiciones con promedio penalizado (con combinación de centroides altamente diferenciados). Este fue el caso de las posiciones 20 y 37 mostradas en el cuadro 9.2. Esta verificación se muestra en el cuadro 9.3.

Cuadro 9.3: Verificación de posición

posición	giro	pose real	pose est	|e ρ|(cm)	|e θ|(gra)	t(s)	loc
20	original	102,340,203º	268,451,225º	199.6 cm	22º	16 s	0.65
20b	30º	102,340,233º	136,324,243º	37.6 cm	10º	50 s	0.95
37	original	558,419,356º	493,353,0º	92.6 cm	4º	31 s	0.62
37b	30º	558,419,26º	569,436,18º	20.2 cm	8º	16 s	1

La precisión de estimación de posición se midió como el error promedio de posición de media de posición $\bar{\rho}_{eQTC}$ de la siguiente forma:

$$\bar{\rho}_{eQTC} = \frac{1}{40} \sum_{i=1}^{40} \mid e\rho_i \mid = 47.18cm \tag{9.6.1}$$

con una desviación estándar $\sigma(\rho)_{QTC}$ calculada como:

$$\sigma(\rho)_{QTC} = \sqrt{\frac{1}{40} \sum_{i=1}^{40} (\mid e\rho_i \mid - \bar{\rho}_{eQTC})^2} = 27.36cm \tag{9.6.2}$$

La precisión de QTC para la estimación de relación angular se midió en términos del promedio de error de estimación de orientación $\bar{\theta}_e$ de la siguiente manera:

$$\bar{\theta}_{eQTC} = \frac{1}{40}\sum_{i=1}^{40} \mid e\theta_i \mid = 11.1^{\circ} \tag{9.6.3}$$

con una desviación estándar $\sigma(\theta)$ calculada como:

$$\sigma(\theta)_{QTC} = \sqrt{\frac{1}{40}\sum_{i=1}^{40}(\mid e\theta_i \mid -\bar{\theta}_{eQTC})^2} = 10.19^{\circ} \tag{9.6.4}$$

Estos valores se obtuvieron considerando las estimaciones de posición y orientación verificadas. Esto es después de las fases de giro y localización para las vistas no inmediatamente localizadas para aquellas con un valor de *loc* menor al umbral anteriormente mencionado.

9.7. Estimación de orientación por geometría epipolar

Para estos experimentos se utilizaron las mismas vistas de prueba usadas para la estimación de pose por QTC. En el cuadro 9.4 se muestran los resultados de estas pruebas.

En el cuadro 9.4, *posición* es el número de la prueba; θ *real* es la orientación real de la vista comparada; θ *est* es la orientación estimada por la geometría epipolar; y la posición estimada por el sistema respectivamente $\mid e\theta \mid$ se refiere al error de angular (en valor absoluto). La nota $n < 8$ se refiere a la imposibilidad de generar una estimación de orientación debido a falta de correspondencias.

La precisión de geometría epipolar para la estimación de relación angular se midió en términos del promedio de error de estimación de orientación de la siguiente manera:

Cuadro 9.4: Pruebas de estimación de orientación con geometría epipolar

posición	θ real	θ est	$\mid e\theta \mid$(gra)
1	45º	–	–
2	306º	279.9º	26.1º
3	162º	220.8º	58.8º
4	284º	291.6º	7.6º
5	72º	56.6º	15.4º
6	176º	163.1º	12.9º
7	293º	295.2º	2.2º
8	18º	–	–
9	81º	88.3º	7.3º
10	95º	87.8º	7.2º
11	158º	–	–
12	0º	16.7º	16.7º
13	9º	22.3º	13.3º
14	167º	186.1º	19.1º
15	194º	177.7º	16.3º
16	198º	196.1º	1.9º
17	72º	79.2º	7.2º
18	144º	n $<$ 8	–
19	257º	247.7º	9.3º
20	203º	207º	4º
21	185º	–	–
22	338º	324.3º	13.7º
23	180º	161.6º	18.4º
24	318º	293.7º	24.3º
25	153º	–	–
26	149º	–	–
27	320º	286.8º	33.2º
28	279º	280.2º	1.2º
29	99º	114.9º	15.9º
30	135º	n $<$ 8	–
31	77º	–	–
32	50º	57.8º	7.8º
33	108º	–	–
34	59º	–	–
35	293º	295.1º	2.1º
36	9º	1.5º	10.5º
37	356º	6.8º	10.8º
38	117º	117.3º	0.3º
39	14º	10.4º	3.6º
40	225º	244.6º	19.6º

$$\bar{\theta}_{eGE} = \frac{1}{40}\Sigma_1^{40} \mid e\theta_i \mid = 13.34^{\circ} \tag{9.7.1}$$

con una desviación estándar $\sigma(\theta)_{GE}$ calculada como:

$$\sigma(\theta)_{GE} = \sqrt{\frac{1}{40}\sum_{i=1}^{40}(\mid e\theta_i \mid -\bar{\theta}_{eGE})^2} = 11.89^{\circ} \tag{9.7.2}$$

Es importante hacer notar que a este resultado se incorporan las estimaciones de las pruebas 20 y 37 que como se mencionó en la sección anterior se sabían inexactas. Ya con la verificación, se altera el promedio de error de orientación a $bar\theta_{eGE} = 12.21$ con $\sigma(\theta)_{GE} = 10.81$. También se tienen las estimaciones altamente erróneas de las pruebas 2 y 3 que son vistas cerradas con un solo candidato de referencia (y con una cantidad relativamente baja de correspondencias). Estos detalles pueden usarse para forzar fases de verificación que mejoren la calidad de la estimación. Las vistas 2 y 3 se muestran en la figura 9.11.

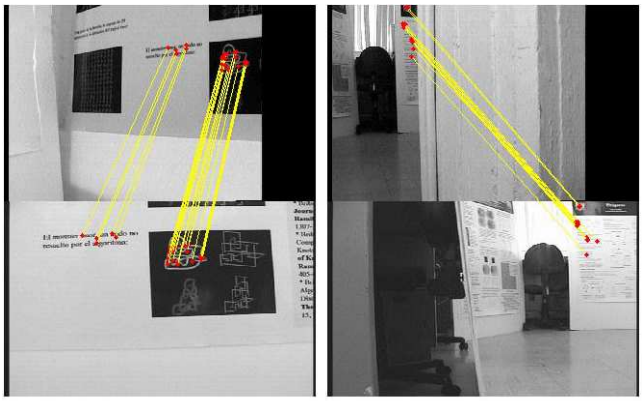

Figura 9.11: Vistas Cerradas de prueba. En ambas imágenes, la vista de arriba es la vista de prueba y la de abajo es la vista de referencia. En (a), la vista es una esquina del laboratorio. En (b) la mayoría de la escena está ocluida por un pilar

En nueve pruebas (posiciones 1, 8, 11, 21, 25, 26, 31, 33 y 34) no existe estimación

debido a que para la posición de la vista no se generó ningún candidato (no hubieron correspondencias suficientes para generar una estimación de pose).

9.8. Estimación final de pose

La pose final propuesta se conforma de la posición (x, y) estimada por el proceso de QTC y una combinación de las orientaciones estimadas de QTC y por geometría epipolar. La combinación fue el promedio de las dos estimaciones, lo cual se muestra en el cuadro 9.5.

Para esta estimación se incorporan las pruebas iniciales y las verificaciones de aquellas que lo necesitaron, como fue el caso de las posiciones no localizadas a la primera vuelta y que requirieron fases de giro y localización (posiciones 1, 8, 11, 21, 25, 26, 31, 33 y 34) y aquellas identificadas como malas estimaciones por el valor inicial de *loc* (posiciones 20 y 37). La precisión de la estimación se calculó de la misma manera que en las pruebas anteriores:

$$\bar{\rho_e} = \frac{1}{40}\sum_{i=1}^{40} \mid e\rho_i \mid = 43.13cm \tag{9.8.1}$$

con una desviación estándar $\sigma(\rho)$ calculada como:

$$\sigma(\rho) = \sqrt{\frac{1}{40}\sum_{i=1}^{40}(\mid e\rho_i \mid -\bar{\rho_e})^2} = 25.79cm \tag{9.8.2}$$

$$\bar{\theta_e} = \frac{1}{40}\Sigma_1^{40} \mid e\theta_i \mid = 9.92^{\mathrm{o}} \tag{9.8.3}$$

con una desviación estándar $\sigma(\theta)$ calculada como:

Cuadro 9.5: Estimación Final

posición	pose real	pose est	$\mid e\rho \mid$(cm)	$\mid e\theta \mid$(gra)
1	91,86,105º	60,60,96.1º	40.5 cm	8.9º
2	85,80,306º	60,60,274.9º	32.0 cm	31.1º
3	232,140,162º	240,90,200.4º	50.6 cm	38.4º
4	169,176,284º	219,141,302.3º	61.0 cm	18.3º
5	152,174,72º	153,133,73.3º	41.0 cm	1.3º
6	182,116,176º	240,90,171.6º	63.6 cm	4.4º
7	144,118,293º	143,104,298.6º	14.0 cm	5.6º
8	102,101,108º	108,125,101.1º	37.8 cm	6.9º
9	240,90,81º	213,81,89.2º	28.5 cm	8.2º
10	170,135,95º	200,107,88.9º	41.0 cm	6.1º
11	185,439,188º	200,425,193º	20.5 cm	5º
12	185,375,0º	182,376,8.3º	3.2 cm	8.3º
13	259,429,9º	292,361,18.2º	75.6 cm	9.2º
14	205,368,167º	224,316,180.1º	55.4 cm	13.1º
15	260,410,194º	276,340,178.7º	71.8 cm	15.3º
16	96,385,198º	164,380,192.1º	68.2 cm	5.9º
17	89,439,72º	60, 450,84.6º	31.0 cm	12.6º
18	52,373,144º	150,390,135º	99.5 cm	9º
19	195,325,257º	160,350,251.4º	43.0 cm	5.6º
20	102,340,233º	136,324,245.4º	37.6 cm	12.4 º
21	369,107,275º	402,120,280.4º	35.5 cm	5.4º
22	397,179,338º	419,181,324.9º	22.1 cm	13.1º
23	417,168,180º	510,150,148.3º	94.7 cm	31.7º
24	475,155,318º	510,156,304.3º	35.0 cm	13.6º
25	389,108,183º	420,120,184º	33.2 cm	1º
26	470,115,179º	430,129,193.1º	42.4 cm	14.1º
27	460,151,320º	487,153,300.4º	27.1 cm	19.6º
28	539,173,279º	540,210,275.1º	37.0 cm	3.9º
29	402,173,99º	381,193,106.5º	29.0 cm	7.5º
30	374,137,135º	404,135,135º	30.1 cm	0º
31	553,323,107º	540,300,102º	26.4 cm	5º
32	414,412,50º	412,361,71.4º	51.0 cm	21.4º
33	387,409,138º	390,390,139.8º	19.2 cm	1.8º
34	322,440,59º	325,446,130.6º	6.7 cm	11.6º
35	404,316,293º	395,329,288.6º	15.8 cm	4.4º
36	410,403,9º	388,308,7.8º	97.5 cm	1.2º
37	558,419,26º	569,436,24.9º	20.2 cm	1.1º
38	367,315,117º	417,350,116.2º	61.0 cm	0.8º
39	330,240,14º	430,291,13.7º	112.3 cm	0.3º
40	330,390,225º	325,402,237.8º	13.0 cm	12.8º

$$\sigma(\theta) = \sqrt{\frac{1}{40}\sum_{i=1}^{40}(\mid e\theta_i \mid -\bar{\theta}_e)^2} = 8.82^{\text{o}} \tag{9.8.4}$$

A partir de esta información se generó el modelo de incertidumbre del proceso de localización visual. Éste se representa con una matriz de varianza–covarianza Σ_v que se define por el valor de error promedio de localización de la siguiente manera:

$$\Sigma_v = \begin{bmatrix} \bar{\rho_e}^2 & 0 & 0 \\ 0 & \bar{\rho_e}^2 & 0 \\ 0 & 0 & \bar{\theta_e}^2 \end{bmatrix}$$

Esta matriz solo cuenta con elementos en la diagonal principal ya que esta matriz es la versión diagonalizada (mostrando componentes principales) de la gaussiana que representa la incertidumbre de localización visual.

9.9. Rastreo de posición

Para evaluar la eficiencia del modelo de odometría y la corrección probabilística se realizaron una serie de movimientos para comparar la posición real del robot con la predicción de ésta.

Primero se calcularon los valores de σ_x, σ_y y σ_Θ necesarios para el cálculo del nuevo estado en el modelo de Kalman. Estos valores se usan para formar la matriz de varianza-covarianza que representa la incertidumbre del modelo de odometría. Estos valores se plantearon como funciones de la distancia recorrida rho y, en el caso de σ_Θ, el giro realizado en la fase inmediatamente anterior.

Los valores de σ_x, σ_y y σ_Θ se calcularon estadísticamente realizando series de movimientos y anotando las desviaciones sobre los ejes de movimiento. A continuación se acopló una función a los puntos medidos para así contar con una predicción de estas desviaciones como función de la distancia recorrida y el giro inmediatamente anterior. Estas funciones quedaron así:

$$\sigma_x = \frac{1}{600}\rho^{1.6} \tag{9.9.1}$$

$$\sigma_y = \frac{1}{7}\rho^{0.6} \tag{9.9.2}$$

$$\sigma_\theta = \frac{1}{100}\theta_{t-1} \tag{9.9.3}$$

que da como lugar a la matriz de varianza-covarianza de odometría:

$$\Sigma_o = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

También se acopló una función para la predicción de la desviación del robot, la cual se presentaba como un ligero giro a la derecha y un ajuste a la distancia avanzada (debido a una rueda defectuosa). Esta desviación se denominó $desvX$ y se calculó como el promedio de desviación perpendicular al movimiento (siempre a la derecha). Éste valor se definió en 10 % del valor de ρ, o lo que es lo mismo:

$$desvX = (0.11) \times \rho \tag{9.9.4}$$

y para el caso del ajuste de distancia recorrida:

$$\hat{\rho} = (0.96) \times \rho \tag{9.9.5}$$

También se calculó la desviación angular $desv\theta$ a partir de una serie de giros de prueba. El valor de esta desviación (también debido a la rueda defectuosa) quedó como:

$$desv\theta = (0.97) \times \Theta \tag{9.9.6}$$

A partir de esta desviación, se ajustó la posición estimada por el movimiento planeado (que de otra forma estaría centrada en la posición ideal). Este ajuste se aproximó haciendo que el avance del robot se realizara sobre un vector recto ligeramente desviado a la derecha. El ángulo de desviación dependía de dos factores: La desviación angular $desv\theta$ y la desviación a la derecha $desvX$. La desviación angular pertinente a cada avance era la inmediata anterior (debido a la secuencia de movimiento compuesta de un avance seguida de un giro). Así, el vector de avance ajustado se realizó con una magnitud igual a $\hat{\rho}$ y un ángulo definido por:

$$\Delta\hat{\theta}_t = \arctan \frac{desvX}{rho} - desv\theta \tag{9.9.7}$$

y así se ajusta el vector de movimiento $\vec{u}$ como:

$$\vec{u} = [\hat{\rho} \;\; \Delta\hat{\theta}_t] \tag{9.9.8}$$

El ajuste a la predicción de la posición final del robot tras un movimiento planeado se muestra en la figura 9.12.

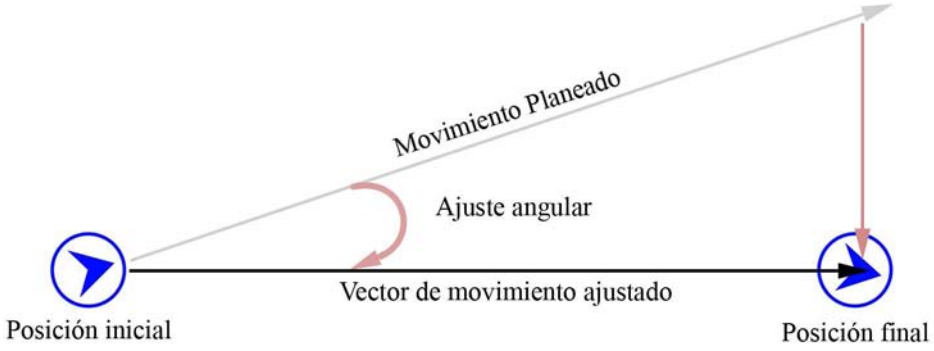

Figura 9.12: Ajuste de movimiento. A partir del movimiento planeado $\vec{u} = [\rho \ \ \Delta\theta]$ se estima la posición de llegada considerando el desvío de traslación a la derecha y el ajuste angular al giro inmediato anterior.

A continuación se realizaron una serie de movimientos sin corrección para probar la predicción de posición del modelo de odometría. Se comparó una gaussiana G_{real} (adaptada a la distribución de posiciones reales) con la gaussiana prevista por el modelo de odometría G_{odo}. El modelo resultó adecuado para la escala del experimento y la precisión buscada ya que la posición del robot se encontró consistentemente dentro del área prevista por el modelo de odometría. Esto se midió al registrar la posición real después de una serie de movimientos y verificar que ésta se encontraba dentro del área de incertidumbre determinada por el modelo de odometría.

Como paso final, se realizaron los mismos movimientos que antes pero con correcciones por filtrado de Kalman. Para cada movimiento se obtuvo la pose estimada por localización visual (explicada en la sección anterior) y la posición estimada por el modelo de odometría. La posición estimada final consiste en el resultado del filtrado de Kalman de estas dos estimaciones. Un ejemplo de la serie de movimientos realizados y las posiciones predichas se muestra en el cuadro 9.6. La serie de posiciones

reales y estimadas por movimiento se muestra en la figura 9.13.

Cuadro 9.6: Estimación probabilística de pose

posición	pose real	pose est	$\mid e\rho \mid$	$\mid e\theta \mid$	$\vec{u}$
1	185,375,270	185,375,270	0 cm	0º	240 cm, 0º
2	201,145,263	211.5,145.9,263.7	10.54 cm	0.7º	0 cm, 180º
3	190,175,78	211.5,146.1,78.1	36.02 cm	0.1º	195 cm, 90º
4	141,333,164	150.7,325.1,159.5	12.51 cm	4.5º	300 cm, 225º
5	439,431,29	415.3,454.8,12.1	33.58 cm	16.9º	0 cm, -45º
6	430,421,346	421.1,446.2,331.8	26.73 cm	14.2º	–

La posición del robot se especificó dentro de un radio de error menor al de predicción por odometría y al de la medición por QTC, que es la esencia del filtro de Kalman. Este error varió dependiendo del desplazamiento realizado (o serie de movimientos) y el área del laboratorio donde se capturó la vista actual (debido a la cantidad de candidatos). Las estimaciones están dentro del radio de predicción de la localización visual explicada en la sección anterior y que se representa gráficamente por una gaussiana de dos dimensiones determinada por la matriz de varianza-covarianza de la localización visual. También se encuentran dentro del área de predicción del modelo odométrico, también definido a través de su propia matriz de varianza–covarianza. El área final de incertidumbre es menor a ambas gaussianas y estará limitado en extensión por la más precisa de las dos. En el caso de movimientos cortos, el modelo de predicción por odometría cuenta con un área de incertidumbre menor a la de localización visual. Sin embargo, si los movimientos son grandes, el límite superior de incertidumbre será determinado por la incertidumbre de la localización visual, la cual será más precisa.

La distancia a la cual ambos modelos de estimación son equivalentes se obtiene de la siguiente forma. Se encuentra el punto donde el valor máximo de Σ_v sea igual

al valor máximo de Σ_o (que es siempre en la dirección perpendicular al movimiento). Esto se puede poner en términos del avance ρ como se observa en la ecuación 9.9.1, donde $\sigma_x = \frac{1}{600}\rho^{1.6}$ es el valor máximo de Σ_o. Para el caso de Σ_v, el valor máximo, como se puede ver en la ecuación 9.8.5 es $\bar{\rho_e}^2$. Así, se tiene que el punto de igualdad de circunstancias (para el eje de mayor incertidumbre) es:

$$\bar{\rho_e}^2 = \sigma_x = \frac{1}{600}\rho^{1.6} \tag{9.9.9}$$

y despejando el valor de ρ para que suceda este equilibrio:

$$\rho_{eq} = (\bar{\rho_e} \times 600)^{0.625} = 572.9cm \tag{9.9.10}$$

Esto quiere decir que si el robot viaja más de $572.9cm$, la estimación visual tendrá una mayor precisión y por lo tanto marcará el límite superior a la incertidumbre del modelo completo. En la figura 9.14 se muestra la posición estimada del robot para la posición 6 de el cuadro 9.6. En 9.14(a) se observa la gaussiana que representa la incertidumbre de la localización que utiliza el filtro de Kalman y las estimaciones de odometría y localización visual. En 9.14(b) se observa la gaussiana de la posición estimada solo por el modelo de odometría. Queda reflejado el hecho de que la segunda estimación es menos precisa. Esto se debe a que, en total, ha recorrido más de $700cm$ de desplazamientos y 540^o de giros, lo cual constituye, para ese modelo de estimación, una predicción menos precisa que la efectuada, en promedio, por la localización visual.

Para los casos en que solo se contara con una estimación (como cuando no se logra una estimación visual), la posición estimada final se fija usando aquella estimación con la que se cuente. En el caso de no contar con ambas estimaciones, el sistema propone un giro y el proceso se reanuda.

Las observaciones finales sobre el funcionamiento del sistema se presentan en el siguiente capítulo.

9.10. Discusión

En lo que respecta al alcance del presente trabajo – localizar al robot globalmente y rastrear su posición – los resultados obtenidos fueron exitosos. En ambos casos el sistema presenta una estimación de pose *cercana* a la real. La precisión del sistema para cada problema es relativa al objetivo y a los resultados de los trabajos relacionados. El objetivo deseado para la localización global en este trabajo puede plantearse como "Localizar al robot dentro de un área que sea lo más estrecha posible en relación al área total en términos de posición y orientación". En este sentido, tomando en cuenta que el área total $\mathbf{A}_T$ del laboratorio en el que se realizaron las pruebas es de

$$\mathbf{A}_T = base \times altura = 5.7m \times 6m = 34.2m^2 \tag{9.10.1}$$

y el área promedio de incertidumbre de localización global $\mathbf{A}_{LG}$ (obtenida a partir del radio de incertidumbre calculado en la sección 9.6) fue de

$$\mathbf{A}_{LG} = \pi \times (\bar{\rho}_e)^2 = \pi \times (0.43)^2 = 0.58m^2 \tag{9.10.2}$$

entonces el área relativa de incertidumbre de localización global $\mathbf{AR}_{LG}$ es

$$\mathbf{AR}_{LG} = \frac{\mathbf{A}_{LG}}{\mathbf{A}_T} \times 100 = \frac{0.58}{34.2} \times 100 = 1.7\,\% \tag{9.10.3}$$

o lo que es equivalente, el robot se localiza con un 98.3 % de precisión relativa al área de trabajo.

Para el caso de la orientación estimada, el promedio de error calculado es de 9.92^{o}. El error relativo de orientación respecto al total $\mathbf{E}\theta_{LG}$ sería de

$$\mathbf{E}\theta_{LG} = \frac{9.92^{\text{o}}}{360^{\text{o}}} \times 100 = 2.76\,\% \tag{9.10.4}$$

o lo que es equivalente a una precisión relativa de $97.24\,\%$. Ambos resultados (posición y orientación) son representativos de una precisión alta.

El mismo método se puede aprovechar para calcular la precisión relativa de la estimación bajo rastreo de posición. En éste caso, como se explica en la sección 9.9, el límite máximo de incertidumbre se fija por la incertidumbre de la mejor de las dos estimaciones a integrarse. En el peor caso (que sucede cuando la odometría es peor que la localización global) es exactamente el valor obtenido por la localización global, que es igual a $98.3\,\%$. Para el experimento mostrado en el cuadro 9.6, el error promedio de posición (sin contar la primera) es de $24cm$ ó $0.24m$, lo que equivale a un área de $0.18m^2$. Así, el área de incertidumbre relativa de posición para rastreo de posición $\mathbf{AR}_{RP}$ sería de

$$\mathbf{AR}_{RP} = \frac{0.18}{34.2} \times 100 = 0.53\,\% \tag{9.10.5}$$

o lo que equivale a un $99.47\,\%$ de precisión relativa. Para la orientación, el promedio del experimento due de 7.3^{o}, por lo que la incertidumbre relativa de orientación para rastreo de posición $\mathbf{E}\theta_{RP}$ sería de

$$\mathbf{E}\theta_{RP} = \frac{7.3^{\text{o}}}{360^{\text{o}}} \times 100 = 2.03\,\% \tag{9.10.6}$$

que equivale a una precisión relativa de $97.97\,\%$.

Para el análisis de precisión relativa a otros trabajos de localización visual se debe realizar una comparación de los resultados tomando en cuenta las condiciones iniciales del escenario y las limitaciones de cada acercamiento. El contexto de comparación es el de la localización en un ambiente de laboratorio usando una sola cámara. En [8] se obtiene, para el problema de rastreo de posición, un error promedio de posición de $39cm$ y un error promedio angular de 4.5^{o}. Estos valores presentan una precisión ligeramente mayor a la obtenida en el presente trabajo. No obstante, se tiene que considerar que la base de obtención de pose encontrada en [8] es un mapa de características definido con medición de distancias con un laser. En [37], la localización se realiza en un pasillo con imágenes omnidireccionales. En este trabajo, la ambigüedad entre posiciones es alta, por lo que la localización es realizada usando el modelo de seguimiento de posición, el cual alcanza una precisión en la cual el error de posición es del orden de $10cm$. La precisión es mayor que en el presente trabajo, pero bajo circunstancias diferentes ya que cada imagen omnidireccional contiene más información visual sobre la circunstancia del punto de vista y el movimiento del robot es a lo largo de un pasillo. En [50] se utilizan imágenes de referencia y Ray casting [41] para localizar al robot en un ambiente de laboratorio durante un recorrido (rastreo de posición). Durante el recorrido varían los errores de posición espacial y angular. El promedio general reportado en las pruebas es del orden de $40cm$ en error de posición y 5^{o} en el error angular. Como puede verse, estos resultados son muy parecidos a los obtenidos en el presente trabajo.

En conclusión, dentro del escenario de localización en interiores con una sola cámara, los resultados obtenidos en ésta tesis, en términos de precisión, se encuentran en el orden de resultados obtenidos en trabajos similares. En términos de velocidad de

procesamiento, el sistema todavía no se encuentra integrado y depurado totalmente, por lo que los tiempos de ejecución son mucho mayores a los reportados por los otros trabajos. Otras limitaciones a tener en cuenta son las siguientes. Para aprovechar la geometría epipolar, conviene tener un ambiente con una cantidad elevada de las características usadas para comparar imágenes (en éste caso características SIFT). También, el sistema sigue siendo altamente susceptible a cambios en la iluminación, cosa que sucede con casi todos los métodos basados en detección de características donde no esté implementado un sistema de invarianza a cambios de iluminación.

Pese a estas limitaciones, y como se discute en esta sección, el sistema es efectivo y eficiente en términos de precisión.

Figura 9.13: Pruebas de rastreo de posición. En (a) se observan la posición inicial real y estimada sobrepuestas. Esto se debe a que se conoce la posición inicial. En la serie de imágenes (b)–(g) se observan las posiciones reales y estimadas descritas en el cuadro 9.6. Las posiciones reales tienen una cruz dentro y las estimadas son aquellas representadas con un círculo rodeando una cabeza de flecha.

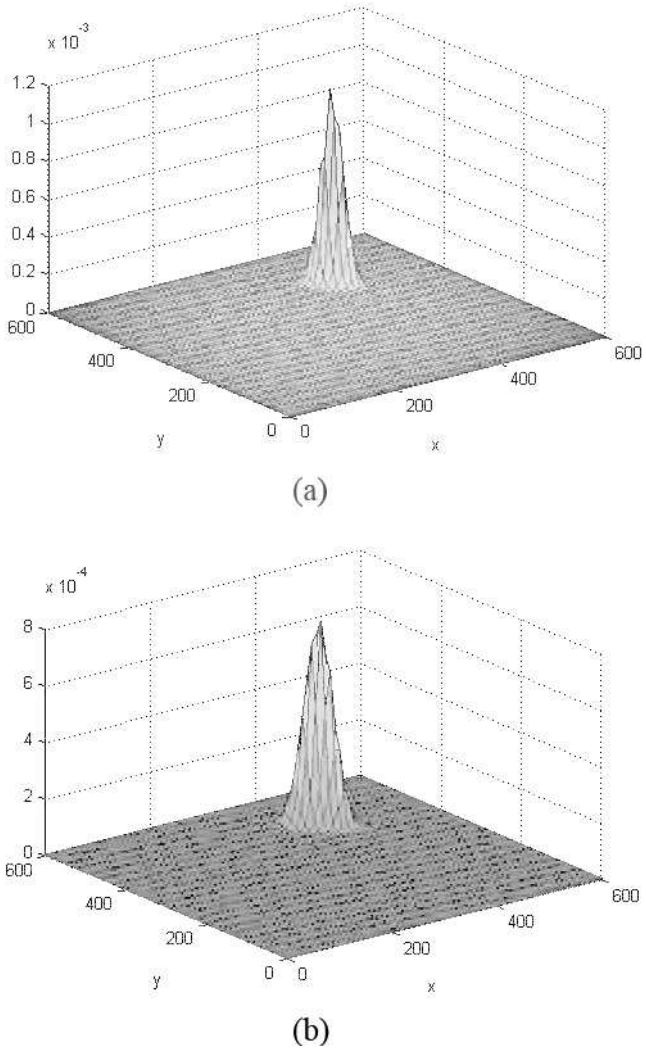

Figura 9.14: Incertidumbre con y sin localización visual. En (a) se muestra la gaussiana que representa la incertidumbre de la posición estimada con el filtrado de Kalman para la última posición del trayecto expuesto en la tabla 9.6. En (b) se muestra la gaussiana de la posición estimada solo usando odometría para la misma posición. Como puede verse, para esta longitud de trayecto, la gaussiana mostrada en (b) tiene una mayor varianza que la de (a). En otras palabras, La estimación de (b) es más incierta que la de (a).

Capítulo 10

Conclusiones

En este trabajo se realizó la integración de varios módulos con el objetivo de encontrar la pose de un robot en un escenario del cual se conocen algunas vistas. Los módulos probados fueron:

1. Detección de correspondencias bajo el descriptor SIFT
2. Localización de pose por recuperación de la geometría epipolar de la escena.
3. Localización de pose por aplicación del algoritmo QTc.
4. Predicción de posición usando odometría.
5. Localización por la incorporación de la predicción de posición y la medición de esta bajo el esquema del filtrado de Kalman.

En lo que respecta al método elegido para comparar imágenes, SIFT se confirmó como un robusto sistema de detección de características. Sin embargo, el número y tipo de características detectadas, en conjunción con la comparación exhaustiva entre imágenes resulta en un lento proceso de comparación. La inclusión del sistema de umbralización de comparación resultó en una aceleración de alrededor del 400 % de los

tiempos iniciales. Aún así, no se probó el sistema on el método de primer cubo primero (best bin first), que reporta fuertes incrementos en la velocidad de comparación [32].

La inclusión del algoritmo de GTM a la comparación de imágenes ayudó a eliminar la gran mayoría de ruido del sistema de medición (visual). Esto hace posible el uso de técnicas como la de geometría epipolar.

El método de acumulación denominado QTC resultó ser una buena aproximación inicial a la posición del robot. Este método se relaciona de manera natural con el concepto de nodos de referencia bajo el cual se propuso la resolución de este trabajo. Las ideas de centroides de posición a partir de las vistas candidatas permitió generar aproximaciones tanto de posición como dé orientación con resultados similares a los reportados por otros trabajos en el área. El método tiene el inconveniente de depender absolutamente de la base de datos de imágenes del ambiente de localización. La adquisición de las imágenes puede tomar mucho tiempo y requiere precisión al generarse. La velocidad de localización bajo éste método es directamente proporcional a la cantidad de imágenes de referencia con las que se cuente. Para el escenario en el que se realizaron las pruebas, la cantidad de imágenes a ser comparadas para el problema de localización global fue de 328. El tiempo promedio de comparación para la localización global fue de 24 segundos (incluyendo el conjunto de pruebas que se tuvo que verificar). Para el caso de rastreo de posición, contando con la posición inicial, el tiempo promedio fue de 4 segundos. Este tiempo de procesamiento es alto si se desea alcanzar la localización en tiempo real. No obstante, el sistema tiene múltiples puntos donde puede lograrse una aceleración como lo son la calidad de comparación, la discriminación de candidatos cercanos (por la definición de la distancia de vecindad) y los parámetros mismos de la obtención de características SIFT.

La metodología conocida como localización por geometría epipolar resultó, dentro del escenario de pruebas, demasiado inexacta y susceptible a ruido como para generar una posición (x, y) estable. Sin embargo, aprovechando el concepto de candidatos parecidos de QTC, se logró generar una estimación de orientación muy cercanas a las de QTC y que ayudaron a la verificación de las estimaciones del método acumulativo. Debido a la forma en que se implementó la geometría epipolar, éste depende de una primera aproximación por parte de QTC y por lo tanto añade su tiempo de operación al tiempo de la estimación final. No obstante, este método se basa en operaciones con matrices que pueden realizarse en tiempo real y por lo tanto su añadido de tiempo es mínimo (de 1 a 3 segundos).

La metodología presentada resuelve los dos problemas de localización: localización global y rastreo de posición. Los módulos se pueden modificar para aumentar la precisión de localización. Esto es práctico si se cuenta con una buena capacidad de procesamiento ya que al aumentar la precisión se incurre en un aumento de carga de procesamiento y por lo tanto en mayor tiempo de operación. Es importante mencionar que no se ha probado el sistema en un ambiente dinámico con objetos o personas moviéndose en el campo de visión del robot.

En conclusión, se presenta un método modular que resuelve los problemas de localización global y seguimiento de posición sujeto a la existencia de una base de datos de imágenes. La metodología se basa en un sistema imperfecto y limitado (un robot con alta incertidumbre en el movimiento y un sistema monocular de visión) y logra generar estimaciones de buena calidad aprovechando la incorporación de varias técnicas de procesamiento de señales y análisis de patrones, como lo son GTM y QTC. En total, se cuenta con un sistema que incorpora varias técnicas que se apoyan entre sí para

generar una buena aproximación de la posición del robot, y que pueden ajustarse (alterando los parámetros intrínsecos a cada módulo) para adaptarse a escenarios de distinta índole.

10.1. Trabajo futuro

Para el trabajo futuro de este proyecto se intentará evitar la necesidad de realizar una localización métrica del robot y migrar a una localización topológica. Se espera utilizar aprendizaje automático para la especificación de los módulos de localización y las técnicas específicas para completar cada tarea. También se estudiará la opción de refinar la estimación de traslación por geometría epipolar (con la opción de utilizar un sistema estéreo) y así poder usarla en la estimación de localización espacial del robot. Otro punto a investigar es la mejor forma de combinar las dos técnicas de localización visual, siendo que se mantenga el método de filtrado de Kalman o se utilicen métodos alternativos basados en filtros de partículas como en [8, 16, 50]. También es importante mencionar que la técnica de asignación de confiabilidad para las predicciones de pose en el módulo de QTC es específico al robot y escenario probados. Este dispositivo es muy útil para detectar la necesidad de verificación pero es necesario definirlo de manera más robusta y dinámica.

10.2. Contribución

La principal contribución del presente trabajo es la integración de los distintos módulos utilizados con el fin de localizar al robot. Asimismo, se implementaron algunos elementos novedosos. En la fase de comparación entre imágenes se aplicó un

método de umbralización sobre las escalas de los descriptores SIFT para acelerar el proceso de generación de candidatos cercanos.

El método de GTM fue implementado con excelentes resultados para eliminar correspondencias erróneas. Como se comentó en el capítulo 4, este algoritmo es una excelente alternativa de eliminación de correspondencias erróneas ya que reporta un aumento en términos de la eliminación de éste tipo de errores respecto a métodos como RANSAC. Este hecho es muy importante si las correspondencias obtenidas se desean utilizar en conjunto con métodos geométricos de localización.

En la fase de aproximación de posición se integró el método de acumulación QTC con el método de geometría epipolar. La virtud de esta combinación radica en el hecho de que las estimaciones son complementarias. El método de geometría epipolar es más efectivo si se ponderan varias aproximaciones de orientación a partir de varias imágenes de referencia. Mientras tanto, QTC presenta estas vistas de referencia en conjunto con un peso de ponderación asociado basado en la cantidad de correspondencias. Otro forma en que estos métodos se complementan radica en la naturaleza de las estimaciones de orientación. QTC obtiene el promedio ponderado de las orientaciones de los candidatos, pero no contiene ninguna información de relación angular entre la vista actual y la de los candidatos. No obstante, esta relación angular está presente en las estimaciones obtenidas por geometría epipolar.

Todos los módulos pueden adaptarse a diferentes ambientes de localización. En la fase de comparación se puede detectar y fijar la mejor escala de comparación, así como los parámetro del grafo de vecindad del método de GTM. En la fase de localización, se pueden variar los parámetros que definen las particularidades de la acumulación. En el caso del filtro de Kalman, el modelo de odometría puede cambiarse para incorporar

las características de un ambiente diferente o incluso de un movimiento distinto (como sucedería si se alteran las características mecánicas del robot en cuestión).

En conclusión, se presenta un sistema modular altamente configurable que integra algoritmos novedosos así como métodos estándar de procesamiento de información, y que resuelven efectivamente los problemas de localización global y rastreo de posición.

Bibliografía

[1] W. Aguilar, *Reconocimiento de objetos basado en la correspondencia estructural de características locales*, Master's thesis, Universidad Nacional Autónoma de México, 2006.

[2] W. Aguilar, Y. Frauel, F. Escolano, M.E. Martinez-Perez, A. Espinoza-Romero, y M.A. Lozano, *A robust graph transformation matching for non-rigid registration*, Image and Vision Computing.

[3] M. Alvarado, *Estimación del movimiento propio a partir de una serie de imágenes*, Master's thesis, Universidad Nacional Autónoma de México, 2007.

[4] K.E. Atkinson, *An introduction to numerical analysis*, second ed., JohnWiley and Sons, 1989.

[5] P. A. Beardsley, A. Zisserman, y D. W. Murray, *Navigation using affine structure from motion*, Proc 3rd European Conf on Computer Vision, Stockholm, Lecture Notes in Computer Science, Springer, 1994, pp. 85–96.

[6] J. Beis y D. Lowe, *Shape indexing using approximate nearest-neighbour search in highdimensional spaces*, Conference on Computer Vision and Pattern Recognition (Puerto Rico), 1997, pp. 1000–1006.

[7] S. Belongie, J. Malik, y J. Puzicha, *Shape matching and object recognition using shape contexts*, IEEE Transactions on Pattern Analysis and Machine Inteligence **24** (2002), 509–522.

[8] M. Bennewitz, C. Stachniss, W. Burgard, y S. Behnke, *Metric localization with scale-invariant visual featues using a single camera*, Proc of European Robotics Symposium (EUROS 2006), 1997, pp. 143–157.

[9] P. Berkhin, *Survey of clustering data mining techniques*, Tech. report, Accrue Software, San Jose, CA, 2002.

[10] F. Chenavier y J. Crowley, *Position estimation for a mobile robot using vision and odometry*, Proc of IEEE International Conference on Robotics and Automation, 1992, pp. 2588–2593.

[11] K. S. Chong y L. Kleeman, *Accurate odometry and error modelling for a mobile robot*, In IEEE International Conference on Robotics and Automation, 1997, pp. 2783–2788.

[12] C. K. Chui y G. Chen, *Kalman filtering with real-time applications*, Springer-Verlag New York, Inc., 1987.

[13] A.J. Davison, W.W. Mayol, y D.W. Murray, *Real-time localisation and mapping with wearable active vision*, ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (Washington, DC, USA), IEEE Computer Society, 2003, p. 18.

[14] R. Duda y P. Hart, *Use of the hough transformation to detect lines and curves in pictures*, Communications of the ACM **15** (1972), 11–15.

[15] M. A. Fischler y R. C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Commun. ACM **24** (1981), no. 6, 381–395.

[16] D. Fox, W. Burgard, F. Dellaert, y S. Thrun, *Monte carlo localization: Efficient position estimation for mobile robots*, In National Conference on Artificial Intelligence, 1999, pp. 343–349.

[17] S. Frintrop, *Visual robot localization and mapping based on attentional landmarks*, KI '07: Proceedings of the 30th annual German conference on Advances in Artificial Intelligence (Berlin, Heidelberg), Springer-Verlag, 2007, pp. 456–459.

[18] R. Gonzales y R. Woods, *Digital image processing*, second ed., Prentice Hall, 2002.

[19] R. M. Haralick, H. Joo, X. Zhuang, V.G. Vaidya, y M.B. Kim, *Pose estimation from corresponding point data*, IEEE Transactions on Systems, Man and Cybernetics **19** (1989), 1426–1446.

[20] C. Harris y M. Stephens, *A combined corner and edge detector*, In Fourth Alvey Vision Conference (Manchester, UK), 1988, pp. 147–151.

[21] R. I. Hartley, *In defence of the 8-point algorithm*, ICCV '95: Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, 1995, p. 1064.

[22] R. I. Hartley y A. Zisserman, *Multiple view geometry in computer vision*, second ed., Cambridge University Press, ISBN: 0521540518, 2004.

[23] L.J. Heyer, S. Kruglyak, y S. Yooseph, *Exploring expression data: identification and analysis of coexpressed genes*, Genome Research **9** (1999), 1106–1115.

[24] M. Jenkin, E. Milos, P. Jasiobedzki, N. Bains, y K. Tran, *Global navigation for ark*, In IEEE/RSJ International Conference on Inteligent Robots and Systems, 1993, pp. 2165–2171.

[25] R. Kalman, *A new approach to linear filtering and prediction problems*, Journal of Basic Engineering **82** (1960), 35–45.

[26] J. Koenderink, *The structure of images*, Biological Cybernetics **50** (1984), 363–370.

[27] L. Robert, C. Zeller, O. Faugeras, y M. Hebert, *Applications of non-metric vision to some visually guided robotics tasks*, 1995.

[28] Y. Lamdan y H.J. Wolfson, *Geometric hashing: A general and efficient model-based recognition scheme*, Computer Vision., Second International Conference on (1988), 238–249.

[29] S. Lazebnik, C. Schmid, y J. Ponce, *Sparse texture representation using affine-invariant neighborhoods*, CVPR, 2003, pp. 319–324.

[30] T. Lindeberg, *Scale-space theory: A basic tool for analysing structures at different scales*, Journal of Applied Statistics **21** (1994), 224–270.

[31] Longuet-Higgins H.C., *A computer algorithm for reconstructing a scene from two projections*, Nature **293** (1981), 133–135.

[32] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, Int. J. Comput. Vision **60** (2004), no. 2, 91–110.

[33] Q. Luong y O. Faugeras, *The fundamental matrix: Theory, algorithms, and stability analysis*, (1995).

[34] Magellan Pro iROBOT, *Cse it service catalog*, World Wide Web, https://wiki.cse.buffalo.edu/services/content/magellan-pro-irobot, 2008.

[35] D. Marr, *Early processing of visual information*, Philosofical Transactions of the Royal Society of London **275** (1976), 483–519.

[36] P.S. Maybeck, *Stochastic models, estimation, and control*, first ed., Academic Press, 1979.

[37] E. Menegatti, M. Zoccarato, y H. Ishiguro, *Image-based monte-carlo localisation with omnidirectional images*, Proceedings Conference of the Italian Association of Artificial Intelligence **48** (2004), 17–30.

[38] K. Mikolajczyk y C. Schmid, *A performance evaluation of local descriptors*, IEEE Transactions on Pattern Analysis and Machine Inteligence **27** (2005), 1615–1630.

[39] L. Pineda, *El proyecto dime y el robot conversacional golem: Una experiencia multidisciplinaria entre la computación y la lingüística*, World Wide Web, http://leibniz.iimas.unam.mx/ luis/golem/.

[40] W.H. Press, B.P. Flannery, S.A. Teukolsky, y W.T. Vetterling, *Numerical recipes in c: The art of scientific computing*, Cambridge University Press, 1988.

[41] S.D. Roth, *Ray Casting for Modeling Solids*, Computer Graphics and Image Processing **18** (1982), no. 2, 109–144.

[42] S. Se y D. Lowe, *Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks*, (2002), no. 21.

[43] A. Shashua, *Projective structure from uncalibrated images: Structure-from-motion and recognition*, PAMI **16** (1994), no. 8, 778–790.

[44] E. Trucco y A Verri, *Introductory techniques for 3-d computer vision*, Prentice-Hall, 1998.

[45] I. Vázquez, *Autolocalización de un robot móvil por medio de visión computacional en espacios cerrados*, 2005, Tesis de Licenciatura.

[46] G. Welch y G. Bishop, *An introduction to the kalman filter*, Tech. report, Chapel Hill, NC, USA, 1995.

[47] Wikipedia, *Corner detection*, World Wide Web, http://en.wikipedia.org/wiki/Corner-detection, 2008.

[48] Wikipedia, *Scale space*, World Wide Web, http://en.wikipedia.org/wiki/Scale-space, 2008.

[49] A.P. Witkin, *Scale-space filtering*, IJCAI, 1983, pp. 1019–1022.

[50] J. Wolf, W. Burgard, y H. Burkhardt, *Robust vision-based localization by combining an image-retrieval system with monte carlo localization*, Robotics, IEEE Transactions on **21** (2005), no. 2, 208–216.