BIBLIOTECA VIRTUAL
MIGUEL DE CERVANTES

Universidad de Alicante

Un enfoque bayesiano para la extracción de características y agrupamiento en visión artificial

Miguel Ángel Cazorla Quevedo

Tesis de Doctorado

Facultad: Escuela Politécnica Superior

Director: Dr. Francisco Escolano Ruiz

2000

UNIVERSIDAD DE ALICANTE

Tesis doctoral

Un enfoque bayesiano para la extracción de características y agrupamiento en visión artificial

Presentada por
Miguel Ángel Cazorla Quevedo
Dirigida por
Dr. Francisco Escolano Ruiz

Departamento de Ciencia de la Computación e Inteligencia Artificial

Mayo del 2000

Para ti, Amparo
Para vosotros, padres

También para ti, Fran
(en tu memoria)

No hay sistemas visuales perfectos. Para construir un sistema óptimo, es necesario especificar claramente sus tareas y el coste de no realizarlas. Además, para construir una máquina visual óptima, uno debe usar tanto conocimiento como sea posible acerca del mundo y de los recursos computacionales disponibles.

[Grzywacz y Escolano, 2000]

Agradecimientos

Una tesis doctoral no surge como un producto aislado. Suele ser la consecución de un camino que a veces se bifurca en bastantes ocasiones, muchas más de las que quisiéramos, para llegar a una meta final. Este camino se empezó a andar con los cursos de doctorado realizados en el Departamento de Tecnología Informática y Computación de la Universidad de Alicante. Durante los cursos empecé a trabajar con dos personas que desde entonces han sido una fuente de constante ánimo y crítica. Me refiero a mi director de tesis Francisco Escolano y a Domingo Gallardo. Junto a ellos, y más tarde con Otto Colomina, fuimos dando forma al contenido final de este trabajo. Fueron muchos días de *pizarrón* y *latiguillos*.

Las personas integrantes del grupo de investigación al que uno pertenece da soporte y ánimos para seguir adelante. En concreto, Ramón Rizo como director, Isabel Alfonso, Pilar Arques, Patricia Compañ, Faraón Llorens y Rosana Satorre. Los recursos, tanto humanos como materiales, de este grupo han servido para que esta tesis se desarrollara de la mejor manera posible.

En el verano de 1999 realicé una estancia en el Smith-Kettlewell Eye Research Institute (SKREI) de San Francisco. Quisiera dedicar unas líneas a agradecer la tremenda amabilidad con la que me acogieron. En especial, agradecer a Norberto Grzywacz todos los esfuerzos realizados para que mi estancia fuera lo más agradable posible. El final del camino de esta tesis empezó a verse en dicha estancia. Alan Yuille y James Coughlan compartieron conmigo sus trabajos actuales. El laboratorio de Alan fue el lugar de múltiples reuniones, en las cuales se discutieron y sacaron adelante muchas de las propuestas aquí presentadas. También a ellos deseo agradecer el tiempo dedicado a ampliar mis conocimientos (en principio bastante escasos) sobre teoría bayesiana. El trabajo de Scott Konishi sirvió de base para la formulación bayesiana de la evidencia de aristas. Scott me asesoró sobre la mejor forma de obtener las mejores instantáneas que luego utilizamos en los experimentos.

8

Prefacio

Dotar a un robot móvil de las capacidades visuales necesarias para sobrevivir en un entorno dado no es una tarea fácil. Dicha supervivencia implica desarrollar con éxito ciertas tareas como auto-localización, evitación de obstáculos, aproximación al objetivo, etc., y el éxito en su realización depende de la capacidad de inferir el estado del mundo y la ubicación del robot en el mismo. Para ello es imprescindible disponer de representaciones y estrategias capaces de procesar y asimilar de manera eficiente y robusta el elevado flujo de información captado por el robot durante su movimiento. Reconstruir exhaustivamente el mundo, tal como enfatiza el paradigma clásico [Marr, 1982], no es una opción aceptable debido a su elevado coste computacional. Durante la última década, el paradigma de la visión activa [Aloimonos *et al.*, 1988], [Bajcsy, 1988], [Ballard, 1991], [Krotkov, 1987], [Yuille y Blake, 1992] viene abordando ésta y otras cuestiones relacionadas con el diseño de sistemas visuales artificiales óptimos en el sentido en que se enfatiza el carácter selectivo del procesamiento visual y su íntimo acoplamiento con la tarea a desarrollar, lo cual conlleva un conocimiento básico acerca del entorno y de los recursos computacionales disponibles.

La inferencia bayesiana proporciona, en combinación con la teoría de la información, un marco conceptual atractivo para formalizar el problema de la inferencia visual [Knill y Richards, 1996]. Asimismo, este planteamiento permite expresar de forma cuantitativa y cualitativa la optimalidad del sistema visual en el sentido en que el paradigma de la visión activa lo requiere. Por ejemplo, dada una tarea concreta (auto-localización, evitación de obstáculos, aproximación al objetivo) el teorema de Bayes proporciona una regla para cuantificar la bondad de la inferencia realizada por los distintos módulos del sistema que intervienen: filtrado [Geman y Geman, 1984], extracción de características [Zerubia y Chellapa, 1993], segmentación [Zhu y Yuille, 1996b], estimación de profundidad [Belhumeur, 1996], matching y reconocimiento [Kittler, 1997]. Dicha regla integra una métrica para identificar la compatibilidad de una solución propuesta con la entrada visual (verosimilitud) con el conocimiento previo del entorno disponible (información a priori). Por otro lado la teoría de la información [Cover y Thomas, 1991] aporta métricas (como la entropía, la información mutua o la distancia de Kullback-Leibler) y principios (como el principio MDL o de longitud de codificación mínima) que permiten, entre otras cosas: reducir la redundancia en la entrada visual [Bartlett *et al.*, 1998], reducir la complejidad de los modelos de contorno [Figuereido *et al.*, 1997], búsqueda de caminos óptimos [Coughlan y Yuille, 1998], determinar las características necesarias para construir modelos de imagen [Zhu *et al.*, 1997], cuantificar la efectividad de los filtros en la

extracción de características [Konishi *et al.*, 1999], así como seleccionar los módulos estrictamente necesarios para resolver la tarea, en la medida en que estos aportan información de utilidad [Rimey y Brown, 1992].

Esta tesis incorpora elementos de inferencia bayesiana y teoría de la información, en los distintos módulos (filtrado y detección de aristas, detección de puntos esquina y uniones, agrupamiento de uniones mediante búsqueda de caminos) que intervienen en la obtención eficiente de una representación geométrica robusta y adecuada para inferir parámetros de posicionamiento, en particular la orientación relativa del robot con respecto al entorno.

Preface

Endowing a robot with the visual capabilities to survive in a partial known environment is not a trivial task. Surviving implies to develop successfully several tasks such as self-localization, obstacle avoidance, target approaching, and so on, and the success depends on the capability of inferring the state of the world and the robot position in it. To do this inference it is essential to have good representations and strategies for processing and assimilating in an efficient and robust way the huge flow of information collected by the robot while it is moving around. Exhaustive world reconstruction, as the classic paradigm emphasizes [Marr, 1982], is not a acceptable option due to its high computational cost. During the last decade, the active vision paradigm [Aloimonos *et al.*, 1988], [Bajcsy, 1988], [Ballard, 1991], [Krotkov, 1987], [Yuille y Blake, 1992] has been addressing this and another questions related to the design of optimal artificial visual systems, but optimal in the sense that emphasizes selective visual processing, close coupling between vision and task, which implies a good knowledge about the environment and the available resources.

Bayesian inference provides, in combination with information theory, an attractive conceptual frame to formalize the visual inference problem
[Knill y Richards, 1996]. This approach allows us to express in a quantitative and qualitative way the optimality of the visual system in the sense that the active vision paradigm requires. For instance, given a specific task (self-localization, obstacle avoidance, target approaching) the Bayes theorem provides a rule to quantify the goodness of a given inference performed by, for instance: filtering [Geman y Geman, 1984], feature extraction [Zerubia y Chellapa, 1993], segmentation [Zhu y Yuille, 1996b], depth estimation [Belhumeur, 1996], matching and recognition [Kittler, 1997]. Such a rule combines a metric to identify the compatibility of a proposed solution with the visual input (likelihood) and the prior knowledge. On the other hand, the information theory [Cover y Thomas, 1991] provides metrics (like entropy, mutual information and the Kullback-Leibler distance) and principles (like the MDL or minimum description length principle) which yields: reducing the redundancy of the visual input [Bartlett *et al.*, 1998], reducing the complexity of the contour models [Figuereido *et al.*, 1997], optimal path searching [Coughlan y Yuille, 1998], determining the necessary features to build image models [Zhu *et al.*, 1997], quantifying the effectiveness of the filters used in feature extraction [Konishi *et al.*, 1999], and selecting only those modules which are needed to solve a task by evaluating the utility of the information provide by these modules [Rimey y Brown, 1992].

This thesis includes elements of Bayesian inference and information theory

points in several visual modules (filtering and edge detection, corner and junctions detection, junction grouping by means of path searching) which are involved in obtaining an efficient and robust geometric representation which is adequated to infer positional parameters, like the robot relative orientation with respect to the environment.

Resumen

El marco de esta tesis se centra en la extracción de características y agrupamiento perceptual en visión. Nuestro propósito principal ha sido formular y experimentar con nuevos métodos para la extracción de características (principalmente aquellos relacionados con la identificación de puntos característicos y la clasificación de uniones) y el agrupamiento (realizando conexión entre uniones). El contexto de aplicación de estos métodos, la visión en robots autónomos, nos impone restricciones especiales sobre su aplicabilidad y así se observan los siguientes requerimientos: eficiencia, robustez y flexibilidad. Estos requerimientos están presentes en todas las técnicas aportadas en esta tesis: la clasificación de uniones se realiza mediante un método voraz que se basa en estadística robusta; el agrupamiento de uniones se realiza mediante un método de búsqueda de caminos con complejidad lineal media, el cual utiliza condiciones estadísticas de poda para restringir la búsqueda de caminos estables; por último, el agrupamiento de uniones encuentra la orientación relativa entre el robot y su entorno. Sin embargo, el rango de aplicación de los métodos propuestos no está restringido a inferir la orientación relativa del robot, y puede ser extendido a otras tareas como la segmentación de imágenes, estimación de profundidad o reconocimiento de objetos.

Esta tesis está estructurada en tres partes: clasificación de uniones, agrupamiento y cálculo de la orientación relativa. Los métodos presentados en cada parte son usados por las siguientes de acuerdo a una organización de complejidad incremental:

- En primer lugar, abordaremos el problema de la *clasificación de uniones*. Para ello nos apoyaremos en la detección de puntos característicos, es decir, de puntos donde la curvatura de los contornos es alta. También son puntos característicos aquellos puntos donde convergen varias aristas. Hemos comparado dos de los métodos de detección de dichos puntos, el operador de SUSAN y el de Nitzberg. Hemos determinado en qué situaciones utilizaremos uno y otro.

  Siendo las uniones estructuras geométricas que emanan de puntos característicos, definimos dichas uniones de forma paramétrica. Formulamos el problema de su clasificación en términos de la búsqueda de los parámetros que mejor se ajustan a la evidencia de la imagen. Dicha búsqueda se basa en la minimización de funciones de energía. Definimos dos métodos basados en inferencia bayesiana.

  Hemos realizado experimentos para comprobar la eficacia de estos méto-

dos, presentando resultados obtenidos al aplicarlos sobre imágenes tanto de interior como de exterior. Ambos métodos presentan buenos resultados en imágenes de interior. Sin embargo, cuando procesamos imágenes de exterior, y debido a la aparición de objetos no estructurados (árboles, carteles, etc.) el elevado número de puntos candidatos obtenidos genera un gran número de uniones ficticias. También hemos observado que los métodos propuestos son muy dependientes de la localización del centro de la unión. Malas localizaciones de dicho centro provocan errores en la clasificación de los límites angulares.

Hemos analizado la eficiencia y robustez de estos métodos propuestos frente a un método reciente basado en programación dinámica, Kona, obteniéndose un error medio y un tiempo de procesamiento menor cuando aplicamos nuestros operadores. Nuestro algoritmo de segmentación lineal detecta un menor número de falsos positivos, es decir, límites que en realidad no lo son. Sin embargo, nuestro algoritmo de modelización de aristas tiene un tiempo de computación menor y deja un menor número de límites sin detectar.

En algunas situaciones se detectan falsos positivos o límites inexistentes, como por ejemplo cuando se invaden dominios de otras regiones. Este caso se puede salvar recurriendo a la prolongación de la unión para buscar otra unión en la dirección que marca el límite de la sección angular. Esta propuesta es la que desarrollaremos en el siguiente punto de la tesis.

- El objetivo de la segunda parte de la tesis es definir *estrategias de conexión de uniones*, lo cual se plantea como la búsqueda de un camino, marcado por el contraste de la imagen. Dicha búsqueda se basa en un algoritmo $A*$ bayesiano, que nos permite centrarnos en la extracción del camino verdadero en una población de falsos caminos, en lugar de buscar el mejor camino en una población de caminos candidatos. Esta simplificación no asegura la optimalidad del resultado, pero permite obtener un coste lineal. El coste de un camino vendrá dado por la suma de dos componentes, la componente de intensidad y la geométrica. La componente de intensidad nos da una medida de la evidencia de arista. La geométrica nos permite modelar la rigidez de los caminos a explorar. Dentro de la componente de intensidad presentamos dos modelos a seguir. En uno de ellos utilizamos un filtro no lineal basado en el número de votos obtenidos para un segmento dado. Para ello se han tomado estadísticas previas, determinando la respuesta del filtro para un determinado operador de gradiente (en nuestro caso hemos utilizado SUSAN) cuando el segmento se encuentra encima de una arista y cuando se encuentra fuera de ésta. En el otro modelo, el de modelización de aristas, las estadísticas se han obtenido de un trabajo previo y modela-

mos tanto la componente de magnitud del gradiente como su orientación. Al igual que tenemos dos términos, consideramos dos umbrales, el de intensidad y el geométrico, que sirven para modelar la evidencia de gradiente y la rigidez del camino, respectivamente. El umbral de intensidad, nos modela la cantidad de evidencia que debe tener debajo un bloque de segmentos para considerarlo válido. Si aumentamos este umbral, sólo nos quedaremos con aquellos caminos que estén encima de un gradiente con mucha magnitud. Al disminuir el umbral permitimos que caminos con poca evidencia (magnitud de gradiente) sobrevivan a la poda. En cuanto al umbral de geométrico, cuando lo disminuimos, siempre dentro de los límites teóricos establecidos, permitimos caminos poco rígidos, mientras que si aumentamos el umbral sólo permitimos caminos con una variación de ángulo entre segmentos muy pequeña.

Una vez definidos los dos modelos, presentamos resultados obtenidos al aplicarlos sobre imágenes tanto de interior como de exterior. Al igual que en los métodos de clasificación de uniones, el método propuesto presenta buenos resultados en imágenes de interior, teniendo una respuesta no tan buena en entornos no estructurados.

El método propuesto es robusto frente a posibles errores en la localización de los límites angulares. Esto es debido a que inicializamos la búsqueda con un margen lo suficientemente grande para que se subsanen estos posibles errores. También es robusto a la no localización de un límite angular, ya que, dadas dos uniones existen dos formas de conectarlas: trazar caminos en ambas direcciones.

Hemos comprobado la ganancia en calidad de las uniones detectadas al aplicar el método de agrupamiento. Comprobamos que se produce una mejora notable en cuanto al número de límites de secciones angulares falsos eliminados. También mejora el error angular cometido en el cálculo de dichos límites. De esta forma estamos introduciendo una retroalimentación de un proceso de alto nivel en otro de un nivel inferior.

- Finalmente, abordamos la obtención del *ángulo de orientación de la cámara*. Partimos de un novedoso método de estimación que consiste en un proceso de votación basado en un criterio bayesiano, utiliza información bayesiana de pertenencia de un determinado píxel a una arista. Hemos realizado experimentos para comprobar su funcionamiento, proponiendo mejoras para solventar algunos problemas. Se ha comprobado que el tiempo de ejecución es excesivo para el el dominio de trabajo en el que estamos interesados, la robótica móvil. Por ello y haciendo uso de nuestro método de agrupamiento de uniones, hemos desarrollado un nuevo método que hace uso de la mis-

ma formulación bayesiana. Este método presenta una mejora sustancial en cuanto a tiempo de computación y el error medio cometido es prácticamente despreciable.

Finalizaremos esta tesis con las conclusiones y planteamientos futuros, fundamentalmente en las siguientes líneas: mejora de la localización, aplicación en tareas de segmentación, reconocimiento y estimación de profundidad, y mejora de los métodos de cálculo de orientación, fundamentalmente en su aplicación dinámica.

Abstract

This thesis focuses on feature extraction and perceptual grouping in computer vision. Our main purpose has been to formulate and test new methods for feature extraction (mainly those related to corner identification and junction classification) and grouping (through junction connection). The context of application of these methods, robot vision, imposes special constrains over their applicability and thus, the following requirements are observed: efficiency, robustness, and flexibility. These requirements are present in almost all the techniques presented in this thesis: Junction classification is performed by greedy method that relies on sound statistics; Junction grouping is performed by a path-searching method of linear complexity on average, which is based on pruning conditions to constrain the search of stable paths, and such conditions rely on statistical information; and junction grouping yields speeding up the voting scheme used to find the relative orientation between the robot and the environment. However, the range of application of the proposed method is not restricted to infer relative orientation, and can be extended to other tasks like image segmentation, depth estimation or object recognition.

This thesis is structured in three parts which cover junction classification, grouping and relative orientation, yielding a bottom-up exposition:

- *Junction Classification:* It relies on combining corner detectors and template matching. Corner detection is performed through two well known operators: SUSAN and Nitzberg. These operators provide an initial localization of the junction center. Given this localization, junction classification is posed in terms of finding the set of angular sections, or wedges, that explain as better as possible the underlying evidence in the image. We propose two greedy methods which relies on elements of Bayesian inference. These methods are tested with indoor and outdoor images in order to identify their robustness to noise and also to bad localizations of junctions centers. Experimental results show that these methods are saver than other methods recently proposed, like Kona, and their computational efficiency is even better. However, we have detected some problems due to the local extent of junction detection. These problems and those derived from bad center localization can be aliviated through local-to-global interactions, and these interactions are inferred by grouping processes.

- *Grouping:* Using classified junctions as starting elements, this stage is performed by finding connecting paths between wedge limits belonging to pairs of junctions, and these paths exist when there is sufficient contrast or edge

support below them. Given that corners are usually associated to points of high curvature in the image, it can be assumed that connecting paths must be smooth if they exist. Contrast support and smoothness are quantified by a cost function. As it can be assumed that there will be no more than one path between two junctions through a pair of wedge limits, such a path can be found by the Bayesian version of the well known A^* algorithm. This method recently proposed searches the true path, according to the cost function, in a population of false path, instead of the best paths among a population of possible paths. Such a simplification yields a computational average cost which is linear with the length of the path. We propose two versions of the cost functions and explore the possibilities of the method for the task at hand by selecting adequate thresholds which control the penalization introduced by the lack of support or smoothness. Penalized partial paths can be discarded by a pruning rule. We also modify this rule to provide stronger pruning although the admissibility of the algorithm is not guaranteed in this case. Grouping is completed by end-path conditions, and the robustness of the process is ensured by bidirectional search. Experimental results show that the grouping stage improves individual junction identification, and in this sense grouping gives feedback to a low level task. But grouping also provides a schematic representation which is useful to higher-level tasks like computing relative orientation from a single image.

- *Relative orientation:* Grouping results followed by straight line detection are very useful to compute the relative orientation between the observer and the environment. A recent proposal, which relies on bayesian inference, has proved that this task can be performed by using a single image, at least when horizontal viewing direction is assumed. This method uses perspective and assumes that lines follow a particular configuration known as Manhattan world. We have replaced the original pixel-based strategy by the edges resulting from grouping in the preceding stage, yielding robust and fast results.

This thesis is completed by final discussion and conclusions. We have identified several issues which can be improved in the future. Some of these issues are referred to the quality of the representations, and others refer to applying our grouping strategy in other tasks like segmentation or even object recognition.

Índice General

1 Introducción **27**
1.1 Visión orientada a tareas frente a visión reconstructiva 29
1.2 Descripción del contenido de la tesis 30

2 Detección y clasificación de uniones **35**
2.1 Métodos de detección de puntos característicos 35
2.1.1 Puntos característicos a partir de aristas 38
2.1.2 Puntos característicos a partir de la propia imagen 39
2.2 Comparativa de detectores de puntos característicos 41
2.2.1 El operador de Nitzberg 41
2.2.2 Descripción de SUSAN 42
2.2.3 Comparación de SUSAN y Nitzberg 46
2.3 Definición de uniones mediante Kona 50
2.4 Clasificación de uniones mediante segmentación 58
2.5 Clasificación de uniones mediante modelado de aristas 67
2.6 Discusión 76

3 Agrupamiento de uniones **79**
3.1 Agrupamiento de características 81
3.2 Modelado de caminos 82
3.2.1 Definición de la componente de intensidad 83
3.2.2 Definición de la componente geométrica 86
3.3 Seguimiento de caminos 89
3.3.1 Algoritmo y restricciones 89
3.3.2 Finalización del camino 91
3.4 Agrupamiento de uniones 94
3.4.1 Conexión punto a punto 101
3.5 Variación de las componentes geométrica y de intensidad 103
3.6 Clasificación de uniones 107
3.7 Discusión 107

4 Obtención del ángulo de orientación de la cámara 111
4.1 Técnicas de obtención de puntos de fuga . . . 112
4.2 Geometría tridimensional . . . 114
4.3 Modelo bayesiano inicial . . . 114
4.3.1 Estadísticas de las aristas . . . 114
4.3.2 Evidencia de un píxel . . . 115
4.3.3 Evidencia de todos los píxeles . . . 115
4.3.4 Resultados . . . 116
4.3.5 Etiquetado de los píxeles . . . 123
4.3.6 Conclusiones del método . . . 123
4.4 Utilización de la información de arista . . . 125
4.4.1 Obtención de caminos rectos . . . 125
4.4.2 Estadística de las aristas . . . 126
4.4.3 Cálculo del ángulo de orientación . . . 127
4.4.4 Resultados . . . 127
4.4.5 Etiquetado de los segmentos . . . 135
4.5 Discusión . . . 135

5 Conclusiones. Líneas de trabajo futuro 139

I Apéndices 143

A Inferencia bayesiana 145
A.1 Teorema de Bayes . . . 145
A.2 Principio de longitud de descripción mínima . . . 147
A.3 Hipótesis de máxima verosimilitud y error cuadrático medio . . . 148

B Búsquedas dentro de un rango 149
B.1 Búsqueda unidimensional . . . 150
B.2 Árboles de rango . . . 151

C Extracción de segmentos rectos 155

Índice de Figuras

1.1 Robot utilizado en algunos de los experimentos. 28
1.2 Proceso completo de aplicación de los distintos métodos propuestos. 30
1.3 Ejemplo de obtención de uniones en una imagen de gris. 31
1.4 Ejemplo de agrupamiento de uniones. 32
1.5 Cálculo del ángulo de orientación de la cámara. 33

2.1 Aplicación del filtro de Canny modificando la varianza. De izquierda a derecha y de arriba a abajo se han utilizado varianzas de 1, 2 y 3. . 37
2.2 Resultado de calcular los autovalores de una imagen. Arriba derecha: autovalor mayor (aristas), debajo izquierda: autovalor menor (esquinas), debajo derecha: puntos esquina detectados aplicando una umbralización y una búsqueda de máximo locales. 42
2.3 Área SUSAN. 43
2.4 Centro de gravedad. 45
2.5 t=10. 45
2.6 t=20. 45
2.7 t=30. 45
2.8 Puntos esquinas detectados por SUSAN. Arriba izquierda: t=10, arriba derecha: t=20, abajo: t=30. 46
2.9 Resultados al ir aumentando el parámetro de geometría (izquierda-derecha-abajo). 47
2.10 Ejemplo de ejecución del operador de SUSAN variando el ruido en la imagen. Izquierda: sin ruido, central: ruido gaussiano de varianza 20, derecha: ídem varianza 50. 48
2.11 Ejemplo de ejecución del operador de Nitberg variando el ruido en la imagen. Izquierda: sin ruido, central: ruido gaussiano de varianza 20, derecha: ídem varianza 50. 48
2.12 Comparación del error medio cometido por los operadores de SUSAN y Nitzberg. 49
2.13 Comparación del número de falsos positivos detectados por los operadores de SUSAN y Nitzberg. 49
2.14 Comparación de tiempos de ejecución frente al tamaño de la imagen de los operadores de SUSAN y Nitzberg. 50

2.15 Parámetros de la unión. 51
2.16 Valor acumulado de los radios en imágenes reales. 52
2.17 Ejemplo de cálculo de la discretización del vector angular. 54
2.18 Valor acumulado de los radios en imágenes sintéticas. 55
2.19 Efecto no deseado al utilizar un radio incorrecto. 56
2.20 Resultado de aplicar el método Kona. 56
2.21 Resultado de aplicar el método Kona. 57
2.22 Dinámica del límite de una sección angular. En este caso, $P(\mathcal{I}_{\theta_S}|\mu = 150, \sigma = 0) > P(\mathcal{I}_{\theta_S}|\mu = 120, \sigma = 5)$, por lo que la posición del límite se desplaza hacia la izquierda. 59
2.23 Dinámica del límite de una sección angular. 62
2.24 Algoritmo de segmentación lineal mediante competición de regiones. 63
2.25 Error producido en la discretización. 64
2.26 Resultado de aplicar el clasificador de uniones. 65
2.27 Resultado de aplicar el clasificador de uniones. 66
2.28 Arriba: Imagen original y el resultado de calcular el $\log(P_{on}/P_{off})$ en cada píxel. Abajo: Cálculo de los valores de orientación (izquierda) y magnitud (derecha). 68
2.29 Estadísticas de las aristas: P_{on} arriba, P_{off} abajo. 69
2.30 Gráfica del log de P_{on}/P_{off} (arriba) y de la función utilizada para P_{ang} (abajo). 70
2.31 Transformación de la imagen con el método de P_{on}/P_{off}. 72
2.32 Algoritmo de detección de uniones mediante modelado de aristas. . 73
2.33 Resultado de aplicar el método de modelado de aristas. 74
2.34 Resultado de aplicar el método de modelado de aristas. 75

3.1 Imagen original (izquierda) junto con el resultado de calcular el gradiente (derecha). 80
3.2 Estadísticas de las aristas: P_{on} y P_{off}, junto con el logaritmo del ratio entre las dos. 84
3.3 Las posibles configuraciones del filtro (izquierda), nomenclatura de los píxeles (derecha arriba) y los diez test realizados a lo largo de una determinada configuración (derecha abajo). 85
3.4 Funciones de distribución de probabilidad: P_{on} y P_{off}, junto con el logaritmo del ratio entre las dos. 87
3.5 Funciones de distribución de probabilidad: $P_{\triangle G}$ y U, junto con el logaritmo del ratio entre las dos 88
3.6 Ejemplo de árbol de búsqueda. El factor de ramificación Q es tres. . 89
3.7 Situaciones a detectar cuando nos quedamos sin caminos. 92
3.8 Ejemplo de inicialización, búsqueda y finalización de camino. 93

3.9 Algoritmo de búsqueda de camino. 95
3.10 Generación de caminos. 96
3.11 Algoritmo de agrupamiento. 96
3.12 Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro no lineal). 97
3.13 Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro de modelado de aristas). 98
3.14 Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro de modelado de aristas). 99
3.15 Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro de modelado de aristas). 100
3.16 Aplicación del método de agrupamiento exhaustivo. 102
3.17 Imagen utilizada en los experimentos de variación de los umbrales de corte. 103
3.18 Variación del umbral de corte de la componente de intensidad. 104
3.19 Variación del umbral de corte de la componente de intensidad y geométrica. 105
3.20 Variación del umbral de corte de la componente geométrica. 106
3.21 Mejora producida al introducir el agrupamiento de uniones. Las líneas azules son las uniones detectadas y las verdes son los caminos del agrupamiento. En la imagen de la izquierda podemos observar como algunas de las uniones tienen límites que no se corresponden con regiones reales. El proceso de agrupamiento elimina estos límites. En la imagen de la derecha (centro) una unión con forma de T termina clasificándose como una L. En la imagen de la derecha (arriba) la unión original es eliminada al no tener ningún límite soporte de arista. 109

4.1 Sistema de coordenadas en el mundo de Manhattan. 112
4.2 Esfera de acumulación para el cálculo del punto de fuga. 113
4.3 Algoritmo de obtención del ángulo de orientación de la cámara. . . . 116
4.4 Gráficas del MAP frente al ángulo de orientación. Estas gráficas se corresponden con las Figuras 4.5 y 4.9. 118
4.5 Resultado obtenido en imagen de exterior (1). El ángulo de orientación de la cámara obtenido fue de -18º. 119
4.6 Resultado obtenido en imagen de exterior (2). El ángulo de orientación de la cámara obtenido fue de 32º. 119
4.7 Resultado obtenido en imagen de exterior (3). El ángulo de orientación de la cámara obtenido fue de 20º. 120

4.8 Resultado obtenido en imagen de exterior (4). El ángulo de orientación de la cámara obtenido fue de 8º. 120
4.9 Resultado obtenido en imagen de interior (1). El ángulo de orientación de la cámara obtenido fue de 37º. 121
4.10 Resultado obtenido en imagen de interior (2). El ángulo de orientación de la cámara obtenido fue de -15º. 121
4.11 Resultado obtenido en imagen de interior (3). El ángulo de orientación de la cámara obtenido fue de -13º. 122
4.12 Resultado obtenido en imagen de interior (4). El ángulo de orientación de la cámara obtenido fue de 10º. 122
4.13 Puntos etiquetados con modelo 4 (derecha) y magnitud del gradiente (izquierda). 123
4.14 Puntos etiquetados con modelo 4 (derecha) y magnitud del gradiente (izquierda). 124
4.15 Resultado erróneo debido a la mala calidad de la imagen obtenida. . 125
4.16 Resultado erróneo ($\Psi = -15º$) utilizando la suma de la evidencia para cada modelo. 127
4.17 Resultado correcto ($\Psi = 40º$) utilizando el máximo valor de evidencia. 127
4.18 Función log posteriori. Izquierda: utilizando el máximo de la función. Derecha: utilizando la suma. 128
4.19 Algoritmo de obtención del ángulo de orientación de la cámara utilizando información de arista. 128
4.20 Resultado obtenido en imagen de exterior. El ángulo de orientación de la cámara obtenido fue de -15º. 129
4.21 Resultados obtenidos en imágenes de exterior. El ángulo de orientación de la cámara obtenido fue de 22º. 130
4.22 Resultados obtenidos en imágenes de exterior. El ángulo de orientación de la cámara obtenido fue de 9º. 131
4.23 Resultados obtenidos en imágenes de interior. El ángulo de orientación de la cámara obtenido fue de 34º. 132
4.24 Resultados obtenidos en imágenes de interior. El ángulo de orientación de la cámara obtenido fue de -13º. 133
4.25 Resultado obtenido en imagen de interior. El ángulo de orientación de la cámara obtenido fue de 13º. 134
4.26 Separación de aristas dependiendo de si están alineadas con el mundo de Manhattan o no. 136
4.27 Separación de aristas dependiendo de si están alineadas con el mundo de Manhattan o no. 137

B.1 Algoritmo de creación de un árbol binario. 150

B.2 Árbol binario creado indicando el nodo de partición. 151
B.3 Obtención de los puntos dentro de un rango. 152
B.4 Árbol de rango con el árbol binario asociado. 153
B.5 Algoritmo de consulta de un árbol binario. 154

C.1 Ejemplo de obtención de los segmentos rectos a partir del resultado del agrupamiento. 156

Capítulo 1

INTRODUCCIÓN

El marco de esta tesis se centra en la navegación autónoma de robots. Dentro de la navegación podemos definir un conjunto de tareas que el robot se puede plantear llevar a cabo. Se nos puede plantear una única tarea, o bien combinaciones de las tres. Estas tareas son:

- Localización: dado un entorno conocido y utilizando información del entorno, el objetivo es encontrar la posición del robot dentro del entorno.
- Construcción de mapas del entorno: el robot está situado en un entorno desconocido. El robot se mueve por él y obtiene una representación geométrica, que nos puede servir para resolver las otras dos tareas.
- Planificación de trayectorias: tenemos al robot en una posición concreta, el objetivo es guiarlo hasta otra posición sin que colisione con los obstáculos.

Clásicamente, estos problemas se han venido resolviendo utilizando la información proporcionada por sensores de ultrasonidos (sonar). Estos sensores devuelven medidas de distancia que nos indican dónde se encuentran los obstáculos. Sin embargo, estas medidas, además de locales, suelen ser ruidosas y fuente de errores. El objetivo de esta tesis es desarrollar un conjunto de técnicas y métodos de extracción y agrupamiento de características en visión artificial que permitan reducir algunos de los errores cometidos. Para que estos métodos se puedan utilizar en un sistema robótico deberán tener, entre otras, las siguientes características:

- El tiempo de ejecución deberá ser lo más pequeño posible. En un sistema móvil tenemos que ser capaces de dar respuesta en el menor tiempo posible debido al movimiento del propio móvil. La mayoría de métodos propuestos para robótica móvil que utilizan visión necesitan una gran cantidad de tiempo de procesamiento.

- Los métodos de extracción deben ser robustos. Tenemos que ser capaces de poder extraer dichas características ante pequeños cambios del entorno.

- También serán genéricas. Nos planteamos extraer características que sirvan para diversas tareas.

Un ejemplo de sistema que utiliza visión de forma eficiente es el desarrollado en la Universidad de Munich [Dickmanns, 1997]. Consiguen guiar a un vehículo por una autopista en tiempo real. También controlan el vuelo de un helicóptero. El sistema que extrae características de la imagen tiene una respuesta temporal de aproximadamente 0,03s. Sin embargo, las características obtenidas sólo sirven para el entorno en el que trabajan y bajo ciertas circunstancias externas, como puede ser el clima y las condiciones de iluminación derivadas de este. Actualmente, cuando se pretende desarrollar un sistema robótico que permita trabajar en tiempo real se suelen buscar características muy específicas al problema a tratar. Sin embargo, nuestra propuesta intenta ser más genérica, teniendo una merma en cuanto al tiempo de respuesta. Es decir, buscamos desarrollar un conjunto de métodos que permitan extraer características que puedan ser utilizadas por diversas tareas. Este planteamiento es clave en aproximaciones a la percepción dirigidas por la tarea.

Algunos de los algoritmos desarrollados en esta tesis se han probado utilizando un robot Pionner I, trabajando con el software Saphira. La Figura 1.1 muestra el robot utilizado.

Figura 1.1: Robot utilizado en algunos de los experimentos.

1.1 Visión orientada a tareas frente a visión reconstructiva

A principios de los años 80, Marr [Marr, 1982] definió uno de los primeros paradigmas sobre la visión artificial. En él establecía una división en tres niveles de la visión. El nivel más bajo se correspondía con una representación en dos dimensiones de la imagen. Para pasar al siguiente nivel aplicaríamos a la imagen una serie de operadores muy básicos que nos proporcionarían características muy locales de la imagen. Estos operadores pueden ser de detección de discontinuidades (aristas) en la imagen, detección de puntos de alta curvatura en aristas, estéreo, detección de movimiento, detección de texturas, etc. Aplicando estos operadores llegamos al segundo nivel. La representación de propiedades de las superficies visibles en un sistema centrado en el observador, tales como orientación de superficie, distancia a partir del observador, discontinuidades, etc. se denomina representación 2-1/2 D, que es una representación intermedia entre el segundo y tercer nivel. Por último llegamos al tercer nivel que se corresponde con una representación centrada en el objeto y en tres dimensiones.

Frente a este planteamiento surgió la visión orientada a tareas [Landy *et al.*, 1996] o visión activa [Yuille y Blake, 1992]. Si utilizamos un esquema de visión orientada a tareas, la tarea es la que define qué módulos u operadores utilizaremos y cómo se van a comunicar entre sí. Es la tarea la que se va a encargar de definir si utilizamos un detector de aristas en vez de uno de esquinas, si necesitamos una agrupación de aristas y en qué medida y qué representación es la más adecuada. Este nuevo esquema permite únicamente procesar aquellas zonas de la imagen útiles para la tarea, frente al esquema clásico que procesaba toda la imagen. También, el nuevo enfoque asume una limitación en los recursos del sistema. En esta tesis nos centraremos en la tarea de la localización visual en un entorno conocido. Esta tarea ha sido tratada en multitud de ocasiones con técnicas que utilizan el sonar como fuente de información. Podemos consultar [Gallardo, 1999] para una reciente propuesta utilizando información bayesiana. Sin embargo, la solución obtenida con estos métodos es sensible a los errores de odometría (medición de su propio movimiento). Los errores de odometría surgen por el ruido implícito en los sensores que nos miden el movimiento. El error más relevante es el que se comete en la detección del ángulo de giro del robot. Al margen de que la visión permite mayor rango perceptual, la extracción robusta y la utilización de características genéricas y representaciones esquemáticas permiten buenas localizaciones en tiempos aceptables. Este es el objetivo último de esta tesis.

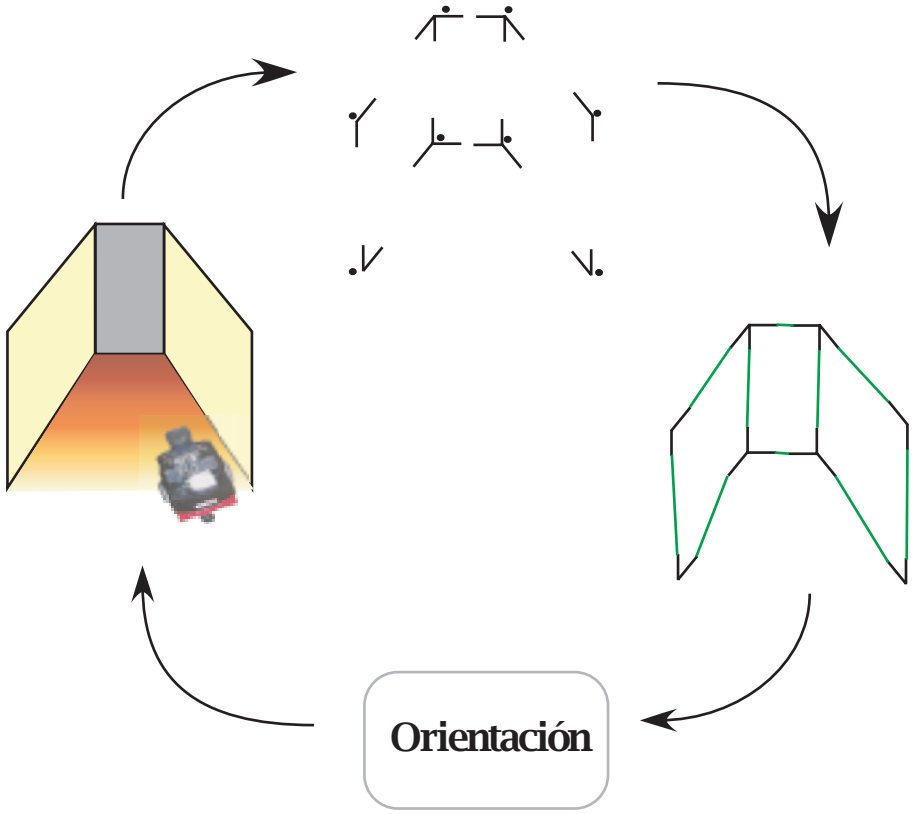

Figura 1.2: Proceso completo de aplicación de los distintos métodos propuestos.

1.2 Descripción del contenido de la tesis

Primero vamos a comentar cuál es el proceso de aplicación de los distintos métodos propuestos en esta tesis. Este proceso lo podemos contemplar en la Figura 1.2. Partimos de una imagen tomada por el robot. El primer paso consiste en detectar los puntos característicos de la imagen y en determinar si existe una unión en dicho punto. Al finalizar este primer paso disponemos de un conjunto de uniones, pero en algunas de ellas se van a plantear ciertos errores. Para solventar dichos errores y para obtener más características de la imagen, realizamos el segundo paso que consiste en el agrupamiento de las uniones. Disponemos ahora de un mapa con conexiones entre las uniones. Por último, utilizamos la información ob-

tenida para detectar el ángulo de orientación del robot. Este ángulo realimentará al robot, influyendo en su movimiento. A continuación comentamos la estructura general de la tesis.

Esta tesis está estructurada de la siguiente manera: en el Capítulo 2 trataremos sobre cómo podemos extraer características de muy bajo nivel, con el menor coste computacional posible. En nuestro caso nos hemos basado en las uniones. Las uniones son puntos de la imagen donde convergen dos o más aristas. En el caso de dos aristas también podemos hablar de esquinas. Hablaremos indistintamente de uniones o esquinas. Propondremos un esquema en dos fases. La primera consistirá en detectar el centro de la unión. En la siguiente fase encontraremos los límites de las secciones angulares de la unión. Un ejemplo de obtención de uniones se muestra en la Figura 1.3.

Figura 1.3: Ejemplo de obtención de uniones en una imagen de gris.

En el Capítulo 3 utilizaremos las características obtenidas en el capítulo anterior como base para realizar un agrupamiento de éstas. Veremos que la detección de uniones lleva implícita un cierto error tanto de detección de límites inexistentes como de localización de dichos límites. Con la agrupación de uniones pretendemos llevar a cabo dos tareas. La primera es intentar solventar dichos errores. La segunda tarea que nos marcamos como objetivo es la de encontrar un mapa planar mínimo que mejor describa la imagen y la relación entre las uniones. En la

Figura 1.4 se muestra un ejemplo de agrupamiento de uniones.

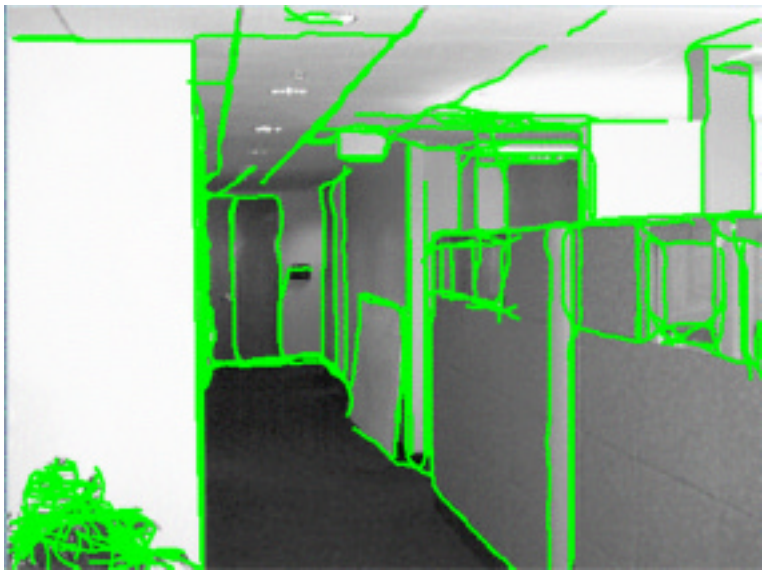

Figura 1.4: Ejemplo de agrupamiento de uniones.

Por último, el Capítulo 4 resuelve el problema de encontrar el ángulo de orientación de la cámara de una manera eficiente. En este capítulo presentaremos un novedoso método de detección de dicho ángulo utilizando información bayesiana. Este método utiliza la restricción de que nos encontramos en un mundo de Manhattan, donde se cumple que la mayoría de los objetos diseñados por el hombre se alinean en dicho mundo. Por ello encontramos únicamente tres puntos de fuga. A continuación presentaremos unas ciertas mejoras al método original y cómo podemos utilizar la información de las aristas para obtener dicho ángulo en un tiempo mucho menor. La Figura 1.5 muestra un ejemplo de cálculo del ángulo de orientación y del etiquetado de los píxeles según el mundo de Manhattan.

Finalizamos con un capítulo de conclusiones y posibles ampliaciones de los métodos propuestos en esta tesis.

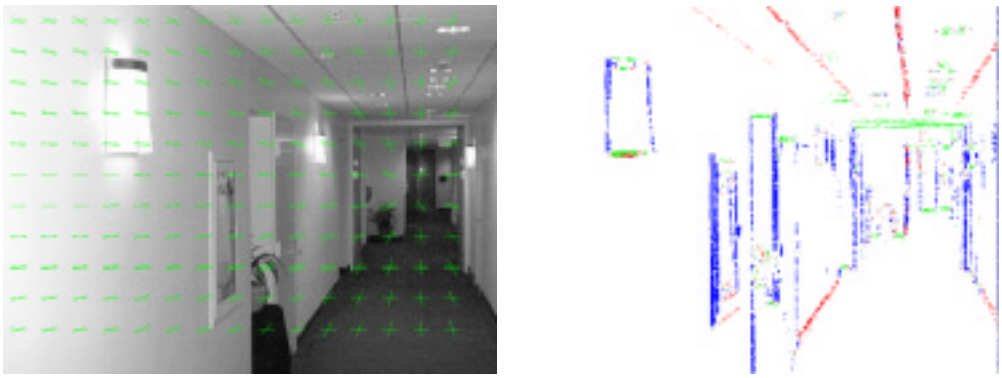

Figura 1.5: Cálculo del ángulo de orientación de la cámara.

Capítulo 2

Detección y clasificación de uniones

El primer paso en nuestra propuesta es la detección y clasificación de uniones. Este proceso lo podemos dividir en dos fases bien diferenciadas: localización del centro de la unión (detección) y localización de los límites de las secciones angulares que forman la unión (clasificación). La calidad de la fase de clasificación depende principalmente del error cometido en la fase de detección. Debemos, pues, utilizar un detector de puntos esquinas lo más preciso posible. En la primera parte de este capítulo realizaremos una revisión de los detectores de puntos característicos previamente definidos para a continuación realizar una comparativa en profundidad entre dos de estos métodos que proporcionan robustez y rapidez: el operador de SUSAN y el de Nitzberg.

En la Sección 2.3 especificamos el problema de la clasificación de uniones en términos de la minimización de una función de energía. Para ello nos basamos en el método Kona [Parida *et al.*, 1998] y proponemos dos nuevos métodos de clasificación. A continuación realizamos una comparación entre nuestros propuestos y Kona. Finalizamos este capítulo con una discusión y algunas posibles extensiones de este trabajo.

2.1 Métodos de detección de puntos característicos

Una imagen contiene una gran cantidad de datos la mayoría de los cuales proporciona muy poca información para interpretar la escena. Un sistema que incorpore visión artificial debe, en un primer paso, extraer de la forma más eficaz y

robusta posible determinadas características que nos proporcionen la máxima información posible. Estas características deben cumplir, entre otras, las siguientes condiciones:

- Su extracción a partir de la imagen no debe suponer un coste excesivo al sistema en el cual está integrado. El tiempo total de extracción debe ser lo más pequeño posible.
- Su localización debe ser muy precisa. El error cometido en la estimación de las características también debe ser lo más pequeño posible.
- Deben ser robustas y estables. Deberían permanecer a lo largo de una secuencia.
- Contendrán la máxima información posible de la escena, es decir, debemos ser capaces de extraer información de tipo geométrico a partir de ellas.

Las aristas, esquinas y uniones satisfacen estas condiciones. Aparecen de forma natural en la mayoría de escenarios tanto naturales como artificiales (construidos por el hombre). Sin embargo, la obtención de las aristas suele ser fuente de muchos errores. Pongamos por ejemplo el algoritmo de detección de aristas más comúnmente utilizado, el algoritmo de Canny [Canny, 1986]. Este algoritmo combina un operador diferencial con un filtro gaussiano. Variando el parámetro de varianza del filtro que utiliza este algoritmo, podemos tener el siguiente comportamiento (ver Figura 2.1): Cuando disminuimos la varianza aparecen un gran número de aristas, la mayoría de las cuales no aportan mucha información. La localización de las aristas en este caso es muy buena. Al aumentar la varianza nos quedamos con aquellas aristas que se encuentran en una zona con gran contraste. Sin embargo, la localización en este caso no es exacta y el tiempo de procesamiento aumenta. Este comportamiento sucede también en la mayoría de los filtros de detección de aristas. También el uso de derivadas de la imagen para el cálculo de aristas supone un error adicional por el ruido que se produce en el proceso de captación de la imagen, tal como se deduce del trabajo de [Micheli *et al.*, 1989].

Podemos definir una unión como la intersección de dos o más aristas en un determinado punto de la imagen. Debemos distinguir una unión de un punto esquina o *punto característico* (utilizaremos ambos términos indistintamente). Un punto esquina es un punto de la imagen donde el contorno de una arista tiene curvatura alta y/o el centro de una unión.

Desde el punto de vista computacional, se han propuesto una serie de enfoques para la detección de este tipo de características bidimensionales. Podemos clasificar estos enfoques en dos grupos principales:

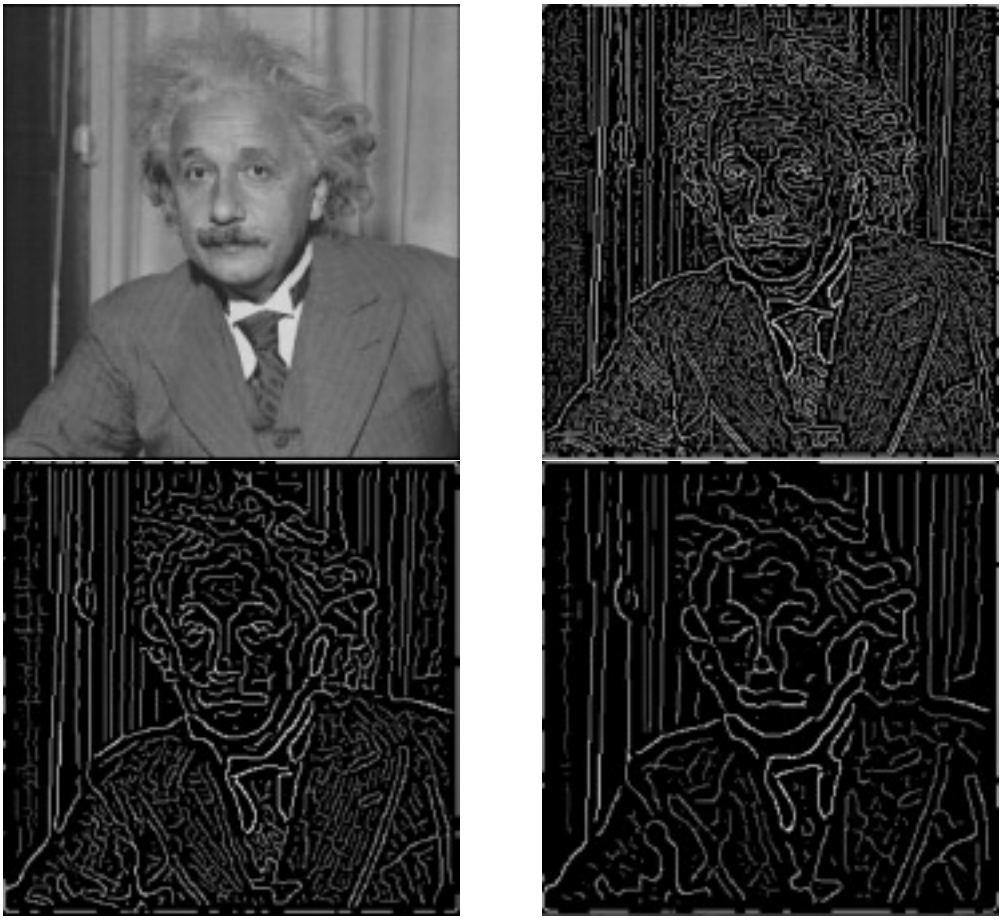

Figura 2.1: Aplicación del filtro de Canny modificando la varianza. De izquierda a derecha y de arriba a abajo se han utilizado varianzas de 1, 2 y 3.

1. Métodos que obtienen las aristas de la imagen mediante algún método de detección de aristas como el de Canny, para, a continuación, detectar puntos de cruce entre aristas o con un cambio sustancial en la dirección de la arista. Estos puntos se clasifican como puntos esquina o unión.

2. El resto de métodos trabajan directamente sobre imágenes de gris, es decir,

no se infieren a partir de la extracción de aristas.

Ciertos métodos clasifican las esquinas detectadas, encontrando los límites angulares donde se encuentran situadas las separaciones de las distintas regiones que confluyen en ellas . En las siguientes subsecciones se comentan algunos de estos métodos empleados tanto para la detección de esquinas como para su clasificación. Podemos consultar [Giraudon y Deriche, 1991] y [Smith y Brady, 1997] para una revisión de los métodos de detección de puntos esquina.

2.1.1 Puntos característicos a partir de aristas

Todos los métodos de obtención de esquinas y/o uniones dentro de este grupo tienen en común una primera fase de extracción de aristas. Algunos métodos de extracción de aristas pueden ser [Torre y Poggio, 1986], [Canny, 1986] o [Smith y Brady, 1997]. Una vez obtenida las aristas estas se representan mediante un código de cadena. Un código de cadena permite expresar, por ejemplo, el cambio de dirección de un píxel al siguiente o la discretización de la tangente para todos los puntos de la arista. Ahora el objetivo consiste en encontrar puntos de máxima curvatura dentro del código de cadena. Este puede ser llevado a cabo de dos posibles formas:

1. Calcular, en cada punto del código de cadena, el valor de la tangente utilizando las coordenadas de los píxeles [Asada y Brady, 1986], [Han *et al.*, 1989]. En [Medioni y Yasumoto, 1987] utilizan un B-spline cúbico para detectar los puntos esquina, ajustando este spline a los puntos de la imagen. También podemos detectar los puntos de máxima curvatura calculando las diferencias entre los códigos de cadena dentro de una ventana. En [Bell y Pau, 1990] se calcula el siguiente valor para todos los puntos del código cadena:

$$\frac{1}{SCAN/2} \sum_{i=1}^{SCAN/2} [codigo[i] - codigo[i + SCAN/2]]$$

   siendo SCAN el tamaño de la ventana. Si este valor supera un cierto umbral se considera que existe un punto de máxima curvatura dentro de la ventana

2. Estimar la tangente en un punto del código de cadena mediante las derivadas parciales de la imagen en ese punto [Deriche y Faugeras, 1990].

Existen propuestas que detectan los puntos esquinas haciendo uso del espacio escala [Rattarangsi y Chin, 1992], [Asada y Brady, 1986] y [Lindeberg y Li, 1997]. En general, la eficiencia de estos métodos dependen directamente de la calidad del

método empleado para la obtención de aristas: si este último no localiza correctamente los puntos de arista difícilmente podremos detectar puntos de esquina de forma exacta. Además se añade un tiempo extra de procesamiento que en ciertos sistemas puede ser prohibitivo.

2.1.2 Puntos característicos a partir de la propia imagen

Existen otros métodos que obtienen los puntos de máxima curvatura de una imagen utilizando directamente los valores de la imagen, sin realizar el paso previo de obtención de aristas. Estos métodos definen una medida que suelen denominar de "esquinidad" la cual se calcula para todos los puntos de la imagen. Cuando esta medida supera un cierto umbral se considera que el punto es una esquina. La mayoría de estos métodos utilizan operadores diferenciales.

Uno de los primeros métodos para la obtención de puntos característicos se describe en [Moravec, 1977]. Dada una ventana rectangular, este operador calcula el valor de cambio de la intensidad de la imagen cuando desplazamos ligeramente esta ventana. Una esquina viene caracterizada por elevados cambios de intensidad. Es altamente sensible a ruido.

En [Beaudet, 1978] se propone un operador que calcula el determinante de una matriz y queda definido como:

$$DET = I_{xx}I_{yy} - I_{xy}^2$$

donde I_{xy} es la derivada parcial de la imagen con respecto al eje x y después con respecto al eje y, y I_{xx} y I_{yy} es la segunda derivada con respecto a x e y, respectivamente. Este operador presenta ciertas características de curvatura interesantes. Las esquinas se obtienen aplicando este operador y realizando una umbralización. En un posterior trabajo [Deriche y Giraudon, 1993] se amplía este método incorporando conocimiento de arista y multiresolución y se realiza un estudio analítico del modelo de esquina. Posteriormente, en [Harris y Stephens, 1988] se continúa investigando sobre las propiedades del determinante antes mencionado. En este último trabajo se presenta un nuevo método de detección de puntos característicos que presenta robustez frente al ruido. La definición de este método se realiza de manera similar al operador de Moravec, pero calculando las derivadas de primer orden de la imagen. La localización de los puntos no es del todo exacta. Comentar que también este método sirve como base para la definición del operador de Nitzberg, comentado más adelante.

En [Kitchen y Rosenfeld, 1982] se presentan tres métodos que obtienen esquinas y que, de manera esquemática, se detallan a continuación:

1. Uso del producto del gradiente de intensidad y del gradiente de la dirección en un punto de la imagen como una medida de esquinidad.

2. Uso de la diferencia entre las direcciones de gradiente de puntos vecinos, los cuales son perpendiculares a la dirección gradiente del punto, como una medida de esquinidad.

3. En una vecindad 3×3 localizar 2 puntos A y B con valor de gris similar al punto C bajo consideración. La diferencia en la dirección entre los vectores CA y CB es una medida de esquinidad.

Una vez aplicada esta medida realizan una supresión de falsos candidatos para quedarse con un único punto en la vecindad. Un enfoque similar se sigue en [Zuniga y Haralick, 1983] donde se detectan esquinas en aquellos puntos donde se producen cambios significativos de gradiente a lo largo de las aristas.

En los trabajos de [Koenderink y Richards, 1988], [Mehrota y Nichani, 1990] y [Dreschler y Nagel, 1982] se definen operadores de cálculo de curvatura sobre una determinada vecindad haciendo uso de derivadas de primer, segundo y tercer orden de alguna función Gaussiana. En [Rangarajan *et al.*, 1989] utilizan 12 máscaras distintas para detectar distintas clases de esquinas. Hacen uso de un detector de aristas para eliminar falsos candidatos. En [Vasselle *et al.*, 1994] se introduce el concepto de espacio escala y determina la localización de esquinas estudiando las características de éstas en dicho espacio. En [Alvarez y Morales, 1997] también se basan en el concepto de espacio escala para determinar la localización de una esquina.

Por último, el enfoque de SUSAN [Smith y Brady, 1997] proporciona un método rápido, robusto frente al ruido (no utiliza derivadas de la imagen) y la localización de los puntos característicos presenta buenos resultados. Este algoritmo se describe en la Sección 2.2.2.

2.2 Comparativa de detectores de puntos característicos

Revisadas las propuestas previas, optamos por descomponer en dos subproblemas. La calidad de un detector de uniones viene determinado por la localización exacta del centro de la unión. Por ello, vamos a analizar dos métodos que presentan buenas características en cuanto a rapidez y error cometido. Describiremos ambos métodos, realizando a continuación una comparación entre ellos.

2.2.1 El operador de Nitzberg

En [Nitzberg *et al.*, 1993] se propone un operador que permite calcular la magnitud del gradiente y su orientación, así como la evidencia de presencia de puntos esquina. Para ello se utiliza la información de gradiente en una determinada vecindad y esto se combina con un operador de gradiente para reducir ruido. Lo primero que debemos hacer es calcular las derivadas parciales en x e y en toda la imagen. Estas se pueden estimar de la siguiente manera:

$$I_x(x,y) = (I(x+1,y) - I(x-1,y))/2$$
$$I_y(x,y) = (I(x,y+1) - I(x,y-1))/2$$
$$\nabla I(x,y) = \begin{pmatrix} I_x(x,y) \\ I_y(x,y) \end{pmatrix}$$

A continuación se define la siguiente matriz 2×2:

$$Q(\mathbf{x}) = \int d\mathbf{x}' \rho(\mathbf{x}-\mathbf{x}') \nabla I(\mathbf{x}') \nabla I(\mathbf{x}')^T = \int d\mathbf{x}' \rho(\mathbf{x}-\mathbf{x}') \begin{pmatrix} I_x^2(\mathbf{x}') & I_x I_y(\mathbf{x}') \\ I_x I_y(\mathbf{x}') & I_y^2(\mathbf{x}') \end{pmatrix} \tag{2.1}$$

donde $\nabla I(\mathbf{x})$ es un vector columna y $\rho(\mathbf{x})$ es una función de ponderación decreciente con valor máximo en 0.

La matriz $Q(\mathbf{x})$ tiene un autovalor mayor $\lambda_1(\mathbf{x})$, asociado al autovector $e_1(\mathbf{x})$ y un autovalor menor $\lambda_2(\mathbf{x})$. La magnitud del gradiente viene dada por $I_e(\mathbf{x}) = \lambda_1(\mathbf{x})$ y su orientación por la orientación del autovector asociado $\theta_I(\mathbf{x}) = e_1(\mathbf{x})$. Para el cálculo de los puntos esquina se utiliza el valor del segundo autovalor o autovalor menor, $I_c(\mathbf{x}) = \lambda_2(\mathbf{x})$.

La función de ponderación $\rho(\mathbf{x})$ nos permite suavizar en mayor o menor medida. En los experimentos, y por simplicidad, hemos elegido la siguiente función:

$$\rho(\mathbf{x}) = \begin{cases} \frac{1}{w^2} & \text{si} \quad -\frac{w}{2} \le x \le \frac{w}{2} \quad \text{y} \quad -\frac{w}{2} \le y \le \frac{w}{2} \\ 0 & \text{en otro caso.} \end{cases}$$

Figura 2.2: Resultado de calcular los autovalores de una imagen. Arriba derecha: autovalor mayor (aristas), debajo izquierda: autovalor menor (esquinas), debajo derecha: puntos esquina detectados aplicando una umbralización y una búsqueda de máximo locales.

fijándose $w = 4$. Para obtener los puntos esquina realizaremos una umbralización utilizando un valor prefijado, seguido de una supresión de mínimos locales. En la Figura 2.2 se muestra el resultado de calcular los autovalores de una imagen.

2.2.2 Descripción de SUSAN

En [Smith y Brady, 1997] se presenta un nuevo enfoque para el procesamiento de imágenes a nivel bajo. Se busca introducir un aspecto clave en el desarrollo de un sistema de visión artificial: rapidez. Este nuevo enfoque es guiado por la premisa de obtener un sistema que extraiga características de la imagen (aristas, esquinas, uniones) en el menor tiempo posible. Por supuesto, el error cometido en la localización de estas características debe ser mínimo. Para ello evitan, en la medida de lo posible, el uso de derivadas para la obtención de las características bidi-

mensionales. En [Micheli *et al.*, 1989] se demuestra que la diferenciación es un problema definido de forma ambigua (*ill-posed*). Esto es debido a que la diferenciación amplifica las componentes de alta frecuencia del ruido lo que implicará una pérdida de estabilidad o continuidad. El problema puede ser transformado en uno *well-posed* aplicando filtros paso bajo para eliminar las componentes de alta frecuencia, con la consecuente pérdida en la definición de los bordes.

Principio de SUSAN

El elemento principal de SUSAN es un filtro no lineal para la detección de las características. Este filtro calcula el área (área SUSAN) de la imagen (dentro del dominio definido por el filtro) que tiene el mismo nivel de gris que el centro. Si, por ejemplo, utilizamos un dominio circular como filtro y trabajamos sobre una imagen discretizada, el filtro acumula el número de puntos dentro del círculo con un nivel de gris similar al del centro. En la Figura 2.3 se puede observar cuál es el resultado de aplicar el filtro a toda la imagen. Los valores mostrados dentro del círculo indican el valor que proporciona el filtro en el cual se basa la detección de los puntos esquina. En la figura de la derecha se representa el resultado de calcular el área SUSAN para la parte superior izquierda del rectángulo (la figura está invertida). También podemos observar que las aristas en la imagen y las esquinas presentan mínimos locales que serán los que se deberán extraer según deseemos una característica u otra.

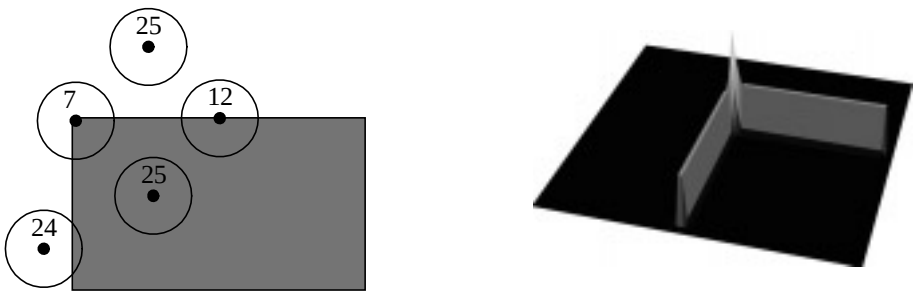

Figura 2.3: Área SUSAN.

El primer paso de cálculo del área SUSAN es idéntico tanto para la detección de aristas como para los puntos esquinas. Este paso consiste en calcular el área SUSAN, la cual se determina en cada punto $\mathbf{x}_0 = (x_0, y_0)$ de la imagen de la siguiente

forma:

$$n(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{D}} c(\mathbf{x},\mathbf{x}_0) \qquad \text{siendo} \qquad c(\mathbf{x},\mathbf{x}_0) = \begin{cases} 1 & \text{si} \quad |I(x,y) - I(x_0,y_0)| \leq t \\ 0 & \text{en otro caso} \end{cases}$$

donde $\mathcal{D}$ es el dominio del círculo, $I(x,y)$ es el brillo de la imagen en el punto $\mathbf{x} = (x,y)$ y t es el umbral que indica el contraste de brillo permitido. Si t es igual a cero estamos teniendo en cuenta únicamente aquellos puntos con el mismo nivel de gris que el centro. Al aumentar t aumentamos el contraste tolerado, lo cual permite absorber puntos de ruido. Esto influye notablemente en el número de puntos esquina detectados, pues al disminuir el valor de t aumentamos dicho número, al ser más sensible a ruido.

El detector de aristas de SUSAN

Dada el área SUSAN las aristas se detectan comparando su valor con un umbral geométrico g que en este caso se asigna a $3n_{max}/4$, donde n_{max} es el valor máximo que n puede tomar (el área correspondiente al dominio del filtro). El valor de la respuesta inicial es:

$$R(\mathbf{x}_0) = \begin{cases} g - n(\mathbf{x}_0) & \text{si} \quad n(\mathbf{x}_0) < g \\ 0 & \text{en otro caso} \end{cases}$$

Esta formulación nos proporciona una respuesta que es mayor cuanto menor sea el área SUSAN de la imagen. El valor de la normal de la arista suele ser necesaria en varias aplicaciones. Para calcular esta normal primero se calcula el centro de gravedad. El centro de gravedad es el centro del área SUSAN y se calcula de la siguiente manera:

$$\bar{\mathbf{x}}(\mathbf{x}_0) = \frac{\sum_{\mathbf{x}\in\mathcal{D}} \mathbf{x} c(\mathbf{x},\mathbf{x})}{\sum_{\mathbf{x}\in\mathcal{D}} c(\mathbf{x},\mathbf{x}_0)}$$

La normal será el vector cuyo origen se sitúe en el centro del filtro y cuyo extremo se encuentre en el centro de gravedad. También suele ser necesario un adelgazamiento de las aristas y esto se realiza en una dirección perpendicular a la arista (en la misma dirección de la normal de la arista). La Figura 2.4 muestra el centro de gravedad (marcado en rojo) en distintos puntos de la imagen.

En la Figura 2.5 y siguientes se muestra un ejemplo de detección de aristas con SUSAN. El tiempo de procesamiento no varía al variar los parámetros y es de 0.06s por imagen, nueve veces menor que el procesamiento con el filtro de Canny. En las figuras se puede observar el efecto de variar el parámetro t (contraste). Al aumentar el factor t disminuimos el número de aristas detectadas, permaneciendo sólo aquellas con mayor cambio de intensidad.

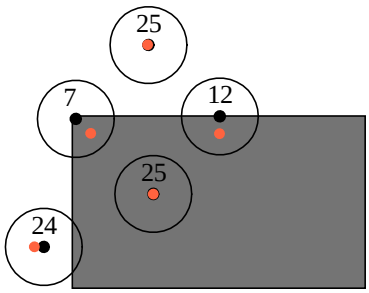

Figura 2.4: Centro de gravedad.

Figura 2.5: t=10.

Figura 2.6: t=20.

Figura 2.7: t=30.

El detector de esquinas de SUSAN

De manera idéntica al detector de aristas, el detector de esquinas calcula el área SUSAN de la imagen y también la compara con un umbral geométrico. Sin embargo, tal como se puede observar en la Figura 2.3, una esquina tendrá un valor SUSAN menor que las aristas por lo que este umbral se fija en $n_{max}/2$. Para eliminar falsos positivos se realizan los siguientes pasos:

1. Se calcula el centro de gravedad del área para el candidato. La distancia del centro de gravedad al centro del área determina si nos encontramos con un falso positivo: si esta distancia es corta, será un falso positivo.

2. Para que sea una esquina verdadera todos los puntos que descansan en una línea recta que une el centro del área con el centro de gravedad, deben ser parte del área SUSAN.

3. Aplican una supresión de mínimos locales, eliminando todos los mínimos

alrededor del candidato menor, dentro de una vecindad 5×5.

En las Figuras 2.8 y 2.9 se muestran algunos resultados de esquinas detectadas con SUSAN.

Figura 2.8: Puntos esquinas detectados por SUSAN. Arriba izquierda: t=10, arriba derecha: t=20, abajo: t=30.

2.2.3 Comparación de SUSAN y Nitzberg

Nuestro objetivo no es realizar una comparativa en profundidad, sino identificar efectos importantes que nos afecten a nuestro futuro trabajo. Se ha realizado un estudio comparativo del funcionamiento y la robustez de ambos operadores. Las Figuras 2.10 y 2.11 muestran resultados de aplicar el operador de SUSAN y de Nitzberg, respectivamente, variando el nivel de ruido en la imagen. En ambos experimentos se han seleccionado los mejores valores de los parámetros para ambos métodos.

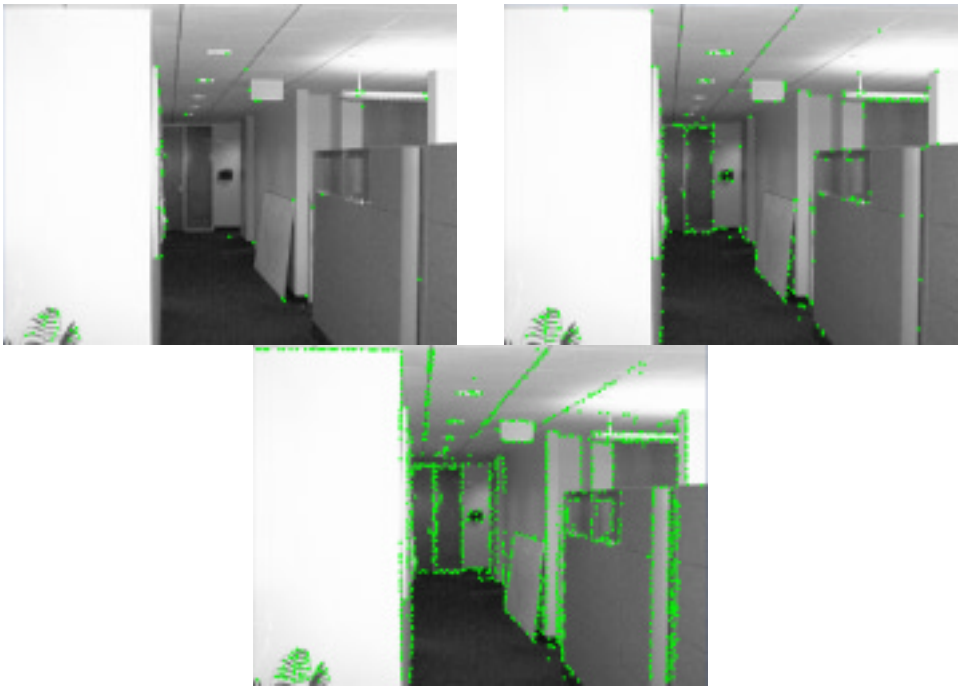

Figura 2.9: Resultados al ir aumentando el parámetro de geometría (izquierda-derecha-abajo).

La primera gráfica que se muestra es la de la Figura 2.12 y muestra el error medio cometido por ambos operadores frente a la presencia de ruido. El error se mide en número de píxeles y nos da la distancia del centro real al centro detectado por el operador. Para las estadísticas sólo se han tenido en cuenta aquellos puntos esquina detectados que son reales y que no están producidas por ruido.

A continuación se han obtenido estadísticas del número de falsos positivos y falsos negativos en función del ruido. Los falsos positivos son aquellos puntos característicos detectados por el operador, pero que en realidad no aparecen en la imagen. Los falsos negativos son aquellos puntos que aparecen en la imagen y sin embargo no son detectados. El número medio de falsos negativos es despreciable en ambos operadores: del orden de 2%. Un dato observado en este estudio es que al aumentar el ruido en la imagen, se detectan mejor los puntos característicos. De hecho, cuando se supera un umbral del 20% de ruido gaussiano ambos operadores son capaces de detectar el 100% de los puntos característicos en la imagen. En la

Figura 2.10: Ejemplo de ejecución del operador de SUSAN variando el ruido en la imagen. Izquierda: sin ruido, central: ruido gaussiano de varianza 20, derecha: ídem varianza 50.

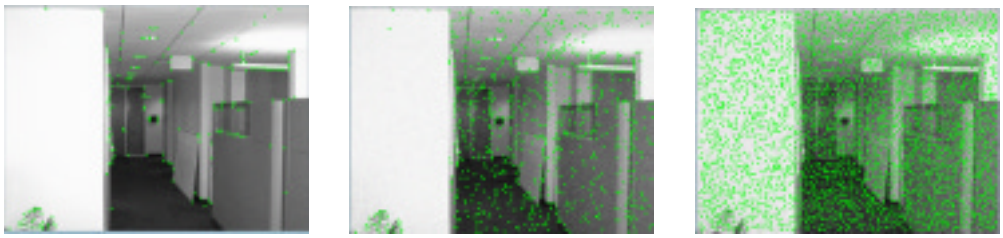

Figura 2.11: Ejemplo de ejecución del operador de Nitberg variando el ruido en la imagen. Izquierda: sin ruido, central: ruido gaussiano de varianza 20, derecha: ídem varianza 50.

Figura 2.13 se muestra el porcentaje del número de puntos candidatos detectados que son falsos positivos.

Por último, hemos tomado tiempos de la ejecución de ambos métodos aumentando el tamaño de la imagen. Nuestras imágenes originales tienen tamaño 640x480. Hemos ido reduciendo el tamaño de la imagen original por un factor de 5, 4, 3 y 2. En la Figura 2.14 se muestra el tiempo de CPU de ambos métodos. Aquí el tamaño 5 indica el tamaño original de las imágenes.

El operador de SUSAN es más rápido que el de Nitzberg cuando la imagen supera el tamaño de 320x240. Si nos fijamos en la robustez, ambos presentan las mismas características de robustez. Sin embargo, será necesario el uso de un filtro que elimine ruido, antes de aplicar cualquiera de estos filtros. El error medio cometido (distancia del centro real al detectado) ha sido de 1 píxel, valor que perfectamente se puede asumir en la mayoría de las aplicaciones de visión.

Sin embargo, el operador de Nitzberg no necesita ajustar tantos parámetros como el de SUSAN. La selección del parámetro de contraste en el operador de SUSAN es bastante crítico. Por todo esto, nos decantamos por el operador de Nitzberg como base para nuestro detector de uniones. Por otro lado, cuando la

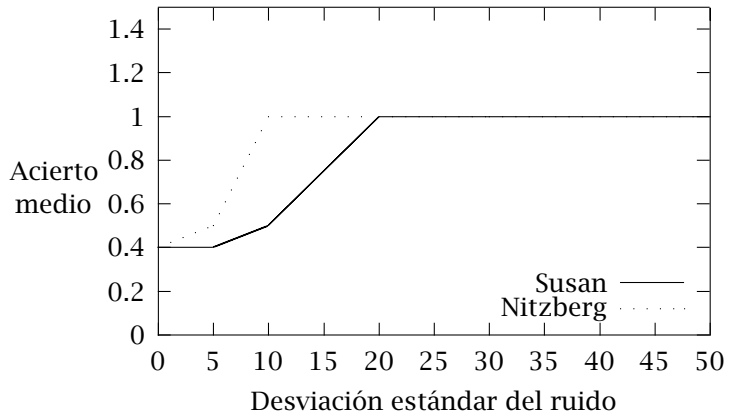

Figura 2.12: Comparación del error medio cometido por los operadores de SUSAN y Nitzberg.

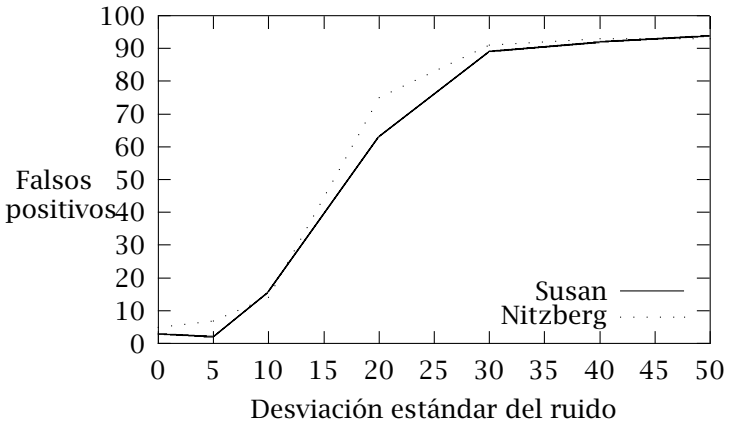

Figura 2.13: Comparación del número de falsos positivos detectados por los operadores de SUSAN y Nitzberg.

aplicación requiera el uso de imágenes de gran tamaño o definición, y también se busque rapidez, recurriremos al método SUSAN.

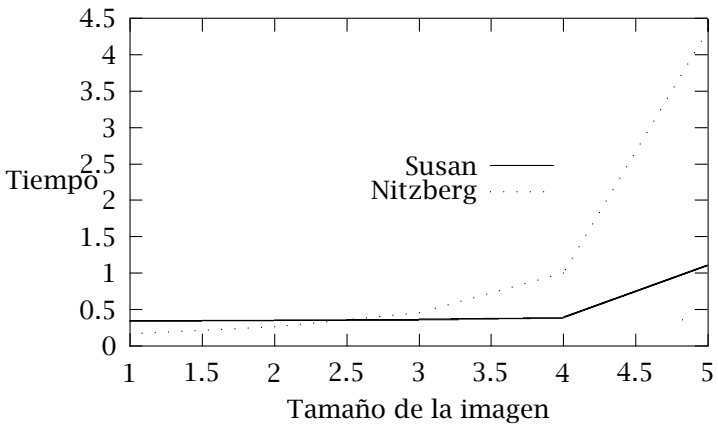

Figura 2.14: Comparación de tiempos de ejecución frente al tamaño de la imagen de los operadores de SUSAN y Nitzberg.

2.3 Definición de uniones mediante Kona

Para clasificar uniones hemos seleccionado el método Kona. Este método define una unión mediante un conjunto de parámetros. Se trata de encontrar los valores de los parámetros que mejor definan los datos de la imagen. La filosofía de trabajo es la misma que la de las plantillas deformables. También la clasificación de uniones se separa en dos fases: detección del centro y clasificación de los límites angulares, un enfoque que se ajusta a nuestra propuesta. Siguiendo la formulación paramétrica del método Kona [Parida *et al.*, 1998] una unión viene definida por (ver Figura 2.15):

- Centro de la unión: x_c, y_c.
- Radio exterior R_1 y radio interior R_0 de la unión. El radio interior es utilizado para evitar el ruido en la proximidad del centro de la unión.
- Número de secciones angulares de la unión: M.
- Los ángulos en los que se encuentran situados los límites de las secciones angulares: $\theta_1, \theta_2, \ldots, \theta_M$.
- Los valores en nivel de gris de las secciones angulares: $T_1, T_2, \ldots, T_M$.

Para encontrar los valores de los parámetros que definen una unión se utiliza una plantilla cuyo comportamiento se ajusta al principio de descripción mínima

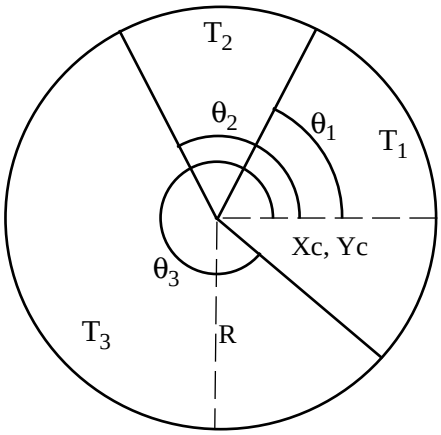

Figura 2.15: Parámetros de la unión.

longitud (MDL) intentándose encontrar el mínimo número de secciones angulares que mejor describen la unión. Para ello se define la siguiente función de energía:

$$E = \mathcal{D} + \lambda \mathcal{G}$$

donde $\mathcal{D}$ es una medida de la distancia de los datos a la plantilla usando la norma L^2:

$$\mathcal{D} = \int_{R_0}^{R_1} \int_0^{2\pi} |\mathcal{I}(r,\theta) - T(\theta)|^2 dr d\theta$$

$\mathcal{G}$ es una medida de la distancia del gradiente:

$$\mathcal{G} = \int_{R_0}^{R_1} \int_0^{2\pi} |\nabla \mathcal{I}(r,\theta) - \nabla T(\theta)|^2 dr d\theta$$

Separando las componentes angular y radial de $\mathcal{G}$ tenemos:

$$\mathcal{G} = \mathcal{A} e_\theta + \mathcal{R} e_r$$

$$\mathcal{A} = \int_{R_0}^{R_1} \int_0^{2\pi} \frac{1}{r^2} \left(\frac{\partial \mathcal{I}}{\partial \theta} - \frac{\partial T}{\partial \theta} \right)^2 dr d\theta$$

$$\mathcal{R} = \int_{R_0}^{R_1} \int_0^{2\pi} \left(\frac{\partial \mathcal{I}}{\partial r} \right)^2 dr d\theta$$

La función de energía global queda:

$$E = \lambda \mathcal{R} + (\mathcal{D} + \lambda \mathcal{A})$$

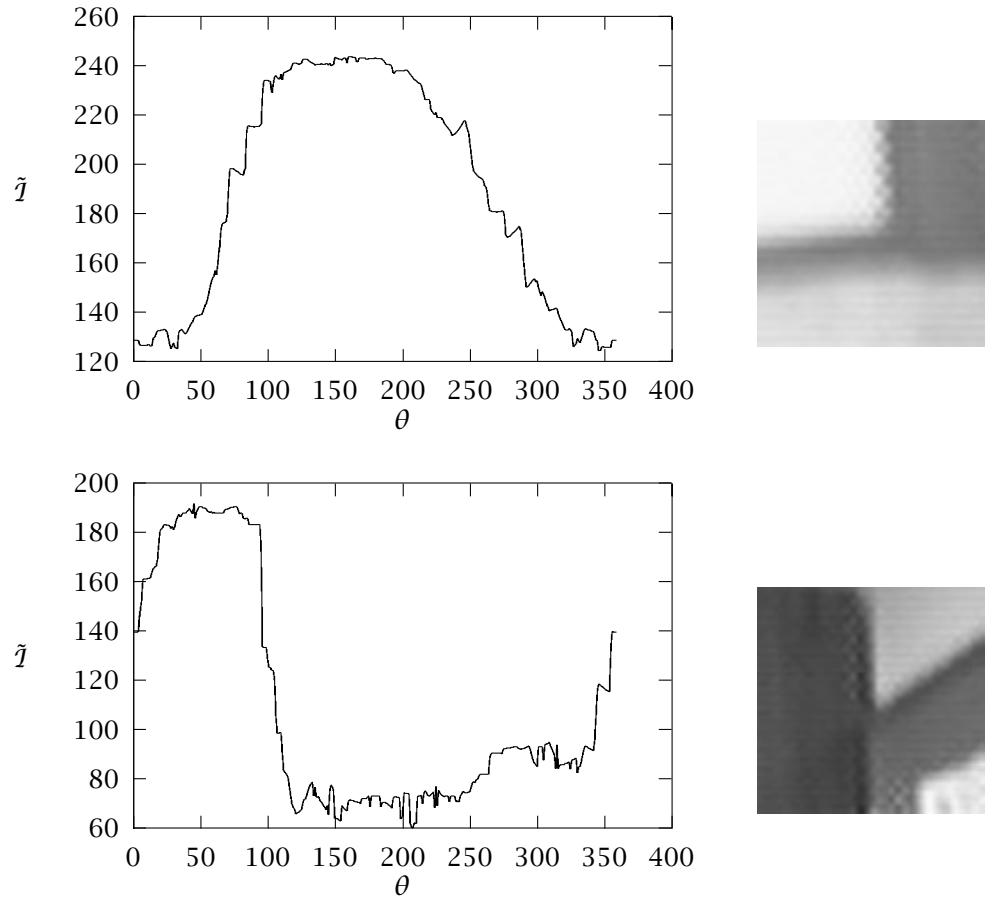

Figura 2.16: Valor acumulado de los radios en imágenes reales.

La función $\mathcal{R}$ se utiliza para localización del centro de la unión y para obtener los radios R_1 y R_0. Para calcular el resto de parámetros de la unión (número de secciones angulares, límites de dichas secciones y valores de gris) utilizan los otros dos términos de la función E:

$$E = \lambda\mathcal{R} + (\mathcal{D} + \lambda\mathcal{A}) = \lambda\mathcal{R} + \bar{E}$$

$$\bar{E} = \mathcal{D} + \lambda\mathcal{A} = \mathcal{F} + \mathcal{V}$$

donde

$$\mathcal{F} = \int_{R_0}^{R_1} \int_0^{2\pi} \left[\mathcal{I}^2(r,\theta) + \lambda \left(\frac{\partial \mathcal{I}(r,\theta)}{\partial \theta} \right)^2 \right] dr d\theta$$

$$\mathcal{V} = \int_{R_0}^{R_1} \int_0^{2\pi} \left[T^2(\theta) - 2T(\theta)\mathcal{I}(\theta) + \lambda \left(\frac{\partial \mathcal{T}^2}{\partial \theta} - 2\frac{\partial \mathcal{I}}{\partial \theta}\frac{\partial \mathcal{T}}{\partial \theta} \right) \right] dr d\theta$$

$\mathcal{F}$ no depende de los valores a estimar por lo que se descarta. Aproximan $\mathcal{V}$ de forma numérica. Para ello utilizan el valor de intensidad del ángulo θ, $\tilde{\mathcal{I}}_\theta$ que definimos a continuación.

Intensidad angular

Un primer paso es realizar un cambio de dominio. Pasamos de un dominio circular alrededor de un punto candidato (centro de la unión), a otro lineal en función del ángulo. Este cambio se realiza calculando el valor medio de los puntos de la imagen a lo largo del radio situado en un determinado ángulo θ, $\tilde{\mathcal{I}}_\theta$. Este valor se obtiene, de forma discreta, tal como ilustra la Figura 2.17. Cada intervalo (su longitud) se multiplica por el valor del punto de la imagen correspondiente:

$$\tilde{\mathcal{I}}_\theta = \frac{1}{(R_1 - R_0)} \sum_{i=1}^{N} l_i \cdot \mathcal{I}_{pixel_i}$$

donde $\mathcal{I}_{pixel_i}$ es la intensidad del punto de la imagen del correspondiente segmento l_i. Las Figuras 2.18 y 2.16 muestra el vector de acumulación angular $\tilde{\mathcal{I}}_\theta$ obtenido para las imágenes que aparecen en las mismas figuras.

Radios

Para obtener el radio R_1 (radio externo de la plantilla) podemos utilizar la siguiente función definida en [Parida *et al.*, 1997]:

$$\mathcal{R} = \int_{R_0}^{R_1} \int_0^{2\pi} \left(\frac{\partial I}{\partial r} \right) dr\, d\theta$$

Se calcula el valor de la función aplicada al punto candidato, incrementando el valor de R_1. Un valor preestablecido limita el valor de $\mathcal{R}$ y así se obtiene el valor R_1. R_0 suele ser una fracción de R_1. R_0 es importante debido a que cerca de un punto esquina se produce un cambio elevado en la dirección del gradiente y debido a la discretización se produce un elevado nivel de ruido. Normalmente se fija el valor de R_1 pues su cálculo resulta extremadamente costoso. Sin embargo,

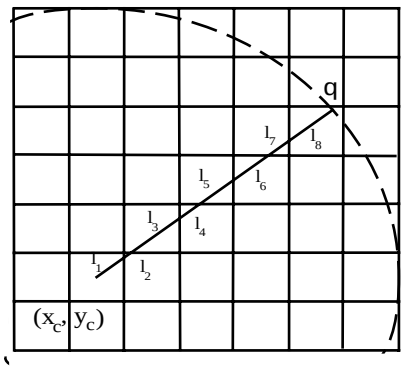

Figura 2.17: Ejemplo de cálculo de la discretización del vector angular.

estos valores pueden ser críticos, tal como se muestra en la Figura 2.19. En el caso de la imagen de la derecha, los valores utilizados fueron de $R_0 = 2, R_1 = 5$. En el caso de la izquierda, $R_0 = 0, R_1 = 10$. Como se puede observar, la imagen de la izquierda presenta peores resultados debido a que el dominio de la unión engloba otras regiones. Por ello aparecen límites angulares en posiciones donde teóricamente no deberían de aparecer. En nuestros experimentos hemos utilizado los valores de $R_0 = 2, R_1 = 5$.

Para localizar el centro de la unión se calcula el valor de $\mathcal{R}$ en toda la imagen y aquel punto con un mínimo local de la función (que supere un cierto umbral prefijado) es un candidato a unión.

Cálculo de las secciones angulares

Una vez identificados el centro y los radios, los únicos parámetros que quedan por encontrar son el número de secciones angulares y los ángulos donde se encuentran los límites de las secciones angulares. Los pasos a seguir para clasificar las uniones dentro de una imagen son:

1. Calcular los valores de intensidad de las secciones angulares. Se realiza mediante una media de los valores dentro de la sección angular:

$$T_l = \frac{\sum_{j=k_l}^{K_{l+1}} \tilde{I}(\theta_j)}{k_{l+1} - k_l}$$

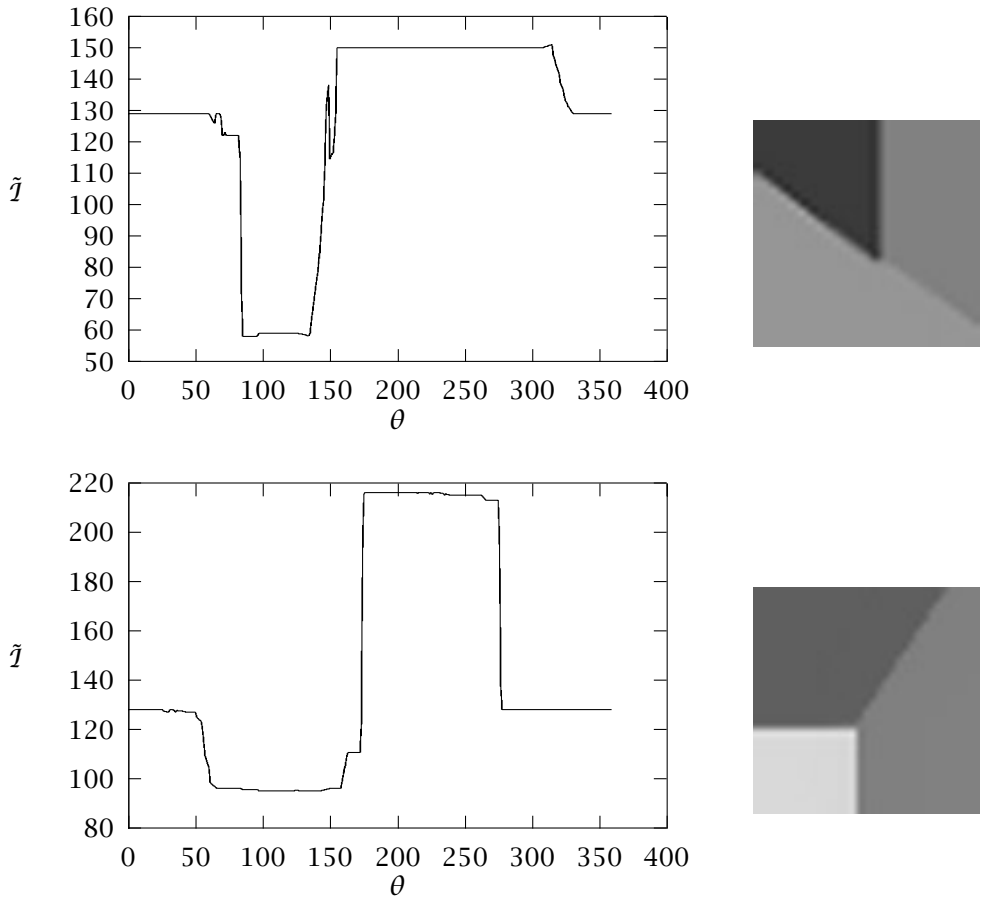

Figura 2.18: Valor acumulado de los radios en imágenes sintéticas.

2. Para calcular las posiciones de los límites de las secciones angulares se exploran todas las posibles situaciones de dichos límites. Para buscar las situaciones correctas se utiliza programación dinámica.

3. Se calcula el valor $\bar{E}$ de la siguiente manera:

$$\bar{E}^p = \sum_{l=1}^{p} \sum_{j=k_l}^{k_{l+1}} (\tilde{\mathcal{I}}(\theta_j) - T_l)^2$$

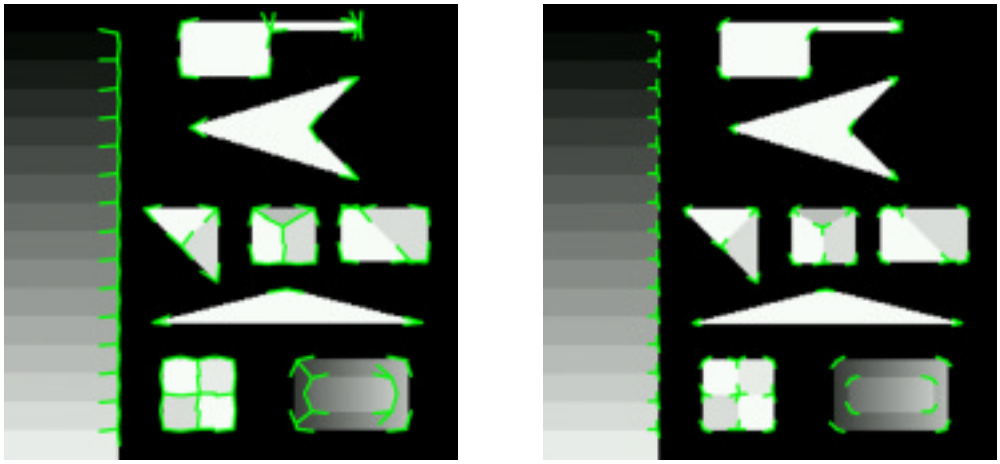

Figura 2.19: Efecto no deseado al utilizar un radio incorrecto.

que es una medida del error cometido con respecto a la imagen. Se procede a calcular desde $p = 1$ hasta que el cociente $r^n = \frac{\tilde{E}^{n+1}}{\tilde{E}^n}$ sea inferior a un cierto umbral prefijado. Aquel valor de n que hace que r^n descienda por debajo del umbral, nos determina el número de secciones angulares.

Las Figuras 2.20 y 2.21 nos muestran resultados de aplicar el método Kona. El tiempo medio de ejecución, utilizando el operador de SUSAN como inicialización, fue de 22 segundos para clasificar 200 uniones. En la siguiente sección se realiza un análisis más profundo de los resultados obtenidos con este operador.

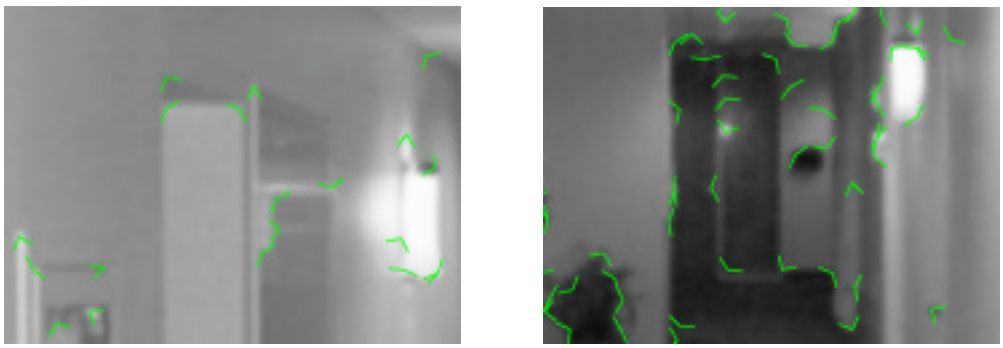

Figura 2.20: Resultado de aplicar el método Kona.

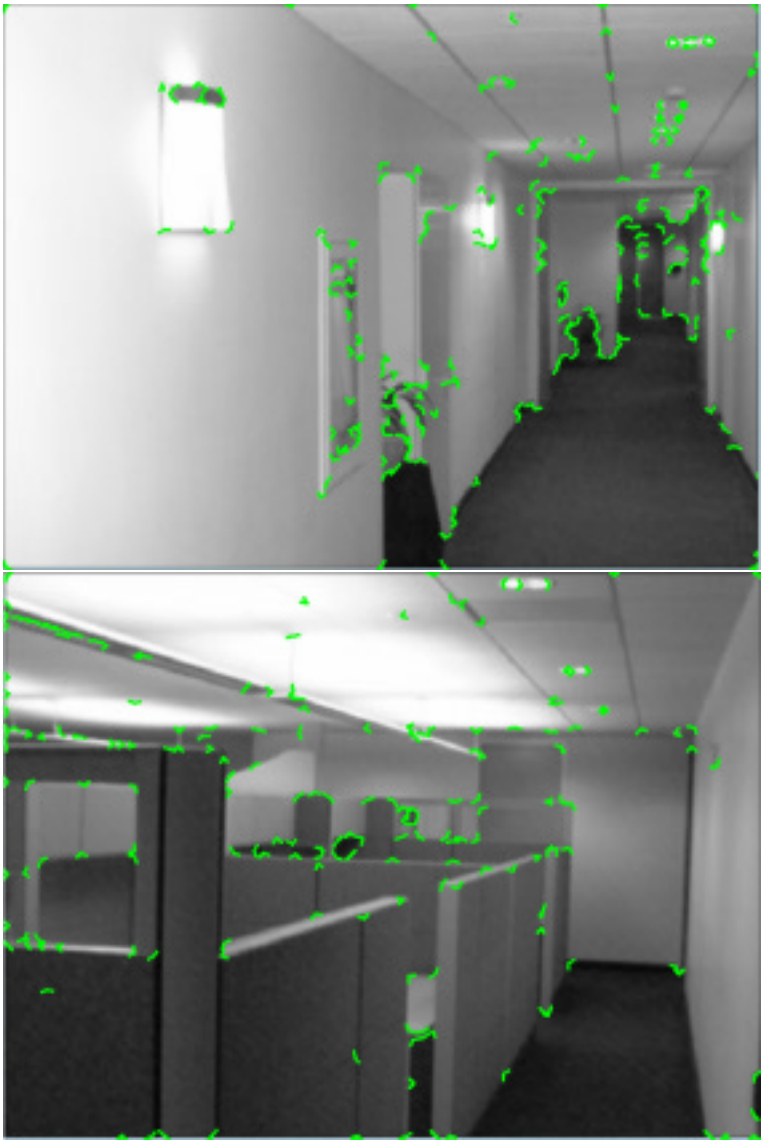

Figura 2.21: Resultado de aplicar el método Kona.

Discusión

Hemos realizado una experimentación previa con el operador Kona. Presenta una buena clasificación de las uniones con un error próximo a 15º. Sin embargo, la principal desventaja de este método es el alto coste temporal. Sustituyendo su detector del centro por el de SUSAN, mucho más rápido, el tiempo de ejecución por imagen fue de 22 segundos. Un tiempo inaceptable para el ámbito de aplicación de esta tesis, la robótica móvil. Por ello se hace necesario un nuevo método de clasificación de los límites angulares que reduzca lo más posible el tiempo de ejecución. En las siguientes secciones vamos a presentar dos nuevos métodos de clasificación de uniones. El primero consiste en aplicar el algoritmo *Region Competition* aplicado a una dimensión, de manera que la búsqueda de los límites de las secciones angulares se realice de forma voraz. En el segundo método explotaremos el modelado estocástico de aristas. Acabaremos este capítulo presentando resultados y comparativas entre ambos métodos.

2.4 Clasificación de uniones mediante segmentación

Este método sigue la misma filosofía que el algoritmo de *Region Competition* [Zhu y Yuille, 1996a], pero aplicado a segmentación en una sola dimensión. El método realiza una búsqueda de las secciones angulares que maximicen la homogeneidad de la intensidad angular. Una sección angular $S_i = \{\theta | \theta \in [\theta_{S_i}, \theta_{S_{i+1}}]\}$ se considera homogénea si sus valores de intensidad son consistentes con los generados por una familia de distribuciones de probabilidad preespecificadas $P(\hat{\mathcal{I}}_\theta | \alpha)$ donde α son los parámetros que codifican la distribución.

Vamos a definir una función de energía global aplicando un criterio MDL[1]:

$$\mathcal{E}(\Theta, \Gamma, \{\alpha_i\}) = \sum_{i=1}^{M} \left\{ -\log P(\{\hat{\mathcal{I}}_\theta : \theta \in R_i\} | \alpha_i) \right\} \tag{2.2}$$

donde M es el número de secciones angulares, $P(\{\hat{\mathcal{I}}_\theta : \theta \in S_i\} | \alpha_i)$ es la suma del coste de codificar cada valor $\hat{\mathcal{I}}_\theta$ dentro de la sección angular S_i de acuerdo a la distribución $P(\hat{\mathcal{I}}_\theta | \alpha_i)$ y $\Gamma = (\theta_{S_1}, \theta_{S_2}, \ldots, \theta_{S_M})$ son los límites de las secciones angulares. Asumiendo modelos de probabilidad independiente, tenemos que:

$$\log P(\{\hat{\mathcal{I}}_\theta : \theta \in S_i\} | \alpha_i) = \int_{S_i} \log P(\hat{\mathcal{I}}_\theta | \alpha_i) d\theta \tag{2.3}$$

[1] *Minimum Description Length*

y la función de energía global resultante es[2]:

$$\mathcal{E}(\Theta,\Gamma,\{\alpha_i\}) = \sum_{i=1}^{M}\left\{-\int_{\theta_{S_i}}^{\theta_{S_{i+1}}} \log P(\hat{\mathcal{I}}_\theta|\alpha_i)d\theta\right\} \tag{2.4}$$

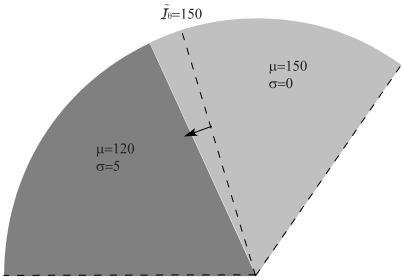

Figura 2.22: Dinámica del límite de una sección angular. En este caso, $P(\mathcal{I}_{\theta_S}|\mu = 150, \sigma = 0) > P(\mathcal{I}_{\theta_S}|\mu = 120, \sigma = 5)$, por lo que la posición del límite se desplaza hacia la izquierda.

Para encontrar el mínimo de esta función de energía se pueden utilizar algoritmos tales como el *simulated anneling* [Geman y Geman, 1984], pero inducen un alto coste computacional y nosotros buscamos un esquema que permita trabajar en tiempo real. Presentamos, pues, un algoritmo voraz dividido en dos fases:

1. Partimos de una segmentación en secciones previa. Para realizar esta segmentación calculamos las aristas utilizando el método de SUSAN adaptado a una dimensión, realizando una supresión de falsos mínimos. A continuación, tenemos que encontrar los $\alpha_i, i = 1, 2, \ldots, M$ que minimizan el coste de descripción de cada región.

   La estimación de los α_i's se puede realizar analíticamente y suponiendo que $P(\hat{\mathcal{I}}_\theta|\alpha_i)$ es una distribución Gaussiana, entonces $\alpha = (\mu, \sigma)$ es la media y la varianza.

2. En el segundo paso, calculados los $\{\alpha_i\}$, realizamos descenso por gradiente con respecto a los puntos $\theta \in \Gamma$. Para cualquier punto θ_{S_i} que pertenece a los límites de una sección angular tenemos:

$$\frac{d\theta_{S_i}}{dt} = -\frac{\partial\mathcal{E}(\Theta,\Gamma,\{\alpha_i\})}{\partial\theta_{S_i}} = \log\left\{\frac{P(\hat{\mathcal{I}}_{\theta_{S_i}}|\alpha_{i-1})}{P(\hat{\mathcal{I}}_{\theta_{S_i}}|\alpha_i)}\right\} \tag{2.5}$$

[2]En lo sucesivo, si $i = M$ entonces $S_{i+1} = S_1$.

Esta función obtenida es equivalente a la regla de decisión de clasificación bayesiana de dos categorías [Duda y Hart, 1973] (ver Apéndice A). El movimiento de θ_{S_i} se determina por el ratio anterior. Si $P(\hat{\mathcal{I}}_{\theta_{S_i}}|\alpha_i) > P(\hat{\mathcal{I}}_{\theta_{S_i}}|\alpha_{i-1})$, es decir, la intensidad en el punto se ajusta mejor a la distribución de la sección angular S_i que a la de S_{i-1}, entonces el límite de la sección angular se desplazará disminuyendo su valor radial. Las Figuras 2.22 y 2.23 muestran de forma gráfica qué ocurre en este caso.

En este punto se nos presentan algunos problemas que debemos solventar:

1. El movimiento del contorno depende de la probabilidad de $\hat{\mathcal{I}}_\theta$. Pequeños cambios en la intensidad pueden provocar que los límites de las secciones angulares no se clasifiquen correctamente. Por ello utilizaremos una discretización de $\pi/180$ para que dicha discretización no afecte demasiado .

2. Debemos permitir la unión de regiones cuando éstas tengan parámetros similares. Incorporamos ahora una nueva fase que permita reducir el número de regiones si con ello se disminuye la energía. Para ello, dadas dos regiones adyacentes, fusionaremos las dos regiones en una si la suma de la energía de ambas regiones es mayor o igual que la energía de la región resultado de la fusión. En la práctica si el cociente entre la suma de las energías y la energía de la región fusionada está por debajo de un cierto umbral, realizamos la unión.

En nuestro problema, vamos a trabajar con distribuciones Gaussianas. Esto se debe a que suponemos que dentro de una sección angular los valores de intensidad se pueden modelar mediante una distribución de este tipo. Esto se puede ampliar a otros tipos de distribuciones, que permitan trabajar, por ejemplo, con texturas. Por lo tanto, nuestra distribución va a estar modelada por una media y una varianza, $\alpha = (\mu, \sigma)$ y

$$P(\hat{\mathcal{I}}_\theta|(\mu,\sigma)) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(\hat{\mathcal{I}}_\theta - \mu)^2}{2\sigma^2}\right\} \tag{2.6}$$

Sustituyendo 2.6 en 2.5 obtenemos la siguiente expresión, que puede ser utilizada en vez del cálculo de logaritmos y exponenciales, sumamente costosos desde el punto de vista computacional:

$$\frac{d\theta_{S_i}}{dt} = \log(\frac{\sigma_i}{\sigma_{i-1}}) + \frac{(\hat{\mathcal{I}}_{\theta_{S_i}} - \mu_i)^2}{2\sigma_i^2} - \frac{(\hat{\mathcal{I}}_{\theta_{S_i}} - \mu_{i-1})^2}{2\sigma_{i-1}^2} \tag{2.7}$$

Vamos a introducir una serie de restricciones que deben cumplir las uniones para que sean consideradas como tal. Estas restricciones deberán ser tenidas en

cuenta a la hora de clasificar una unión y las tendremos en cuenta en los dos métodos propuestos. Las restricciones son tres:

No existencia de unión Si una unión se clasifica con menos de dos secciones angulares significa que no es tal unión. Podemos rechazar aquellas uniones que se clasifiquen con menos de dos límites angulares.

Esquina recta Cuando tengamos únicamente dos secciones angulares y la diferencia entre los ángulos de los límites de las secciones angulares sea igual a ± 180, se trata de una recta, no de una unión. En determinadas aplicaciones nos puede interesar eliminar estas falsas uniones.

Secciones muy pequeñas Cuando una sección angular no supera un cierto umbral de anchura, podemos unirla con otra mayor. Nuestro umbral es de $\pi/9$.

Otro problema que debemos resolver es el siguiente: cuando acumulamos el valor de intensidad angular, se calcula una media de los píxeles a lo largo del radio. Esta media puede inducir a error debido a la posible presencia de otra región dentro del dominio de la unión. La Figura 2.25 muestra este problema. Para solventarlo aplicamos la mediana, tal como se muestra en la figura. Por lo tanto, en vez de utilizar $\tilde{I}_\theta$ utilizaremos $\hat{I}_\theta$, la mediana a lo largo del radio.

En la Figura 2.24 se presenta el algoritmo de segmentación lineal mediante competición de regiones y las Figuras 2.26 y 2.27 muestran el resultado de aplicar este método de clasificación de uniones.

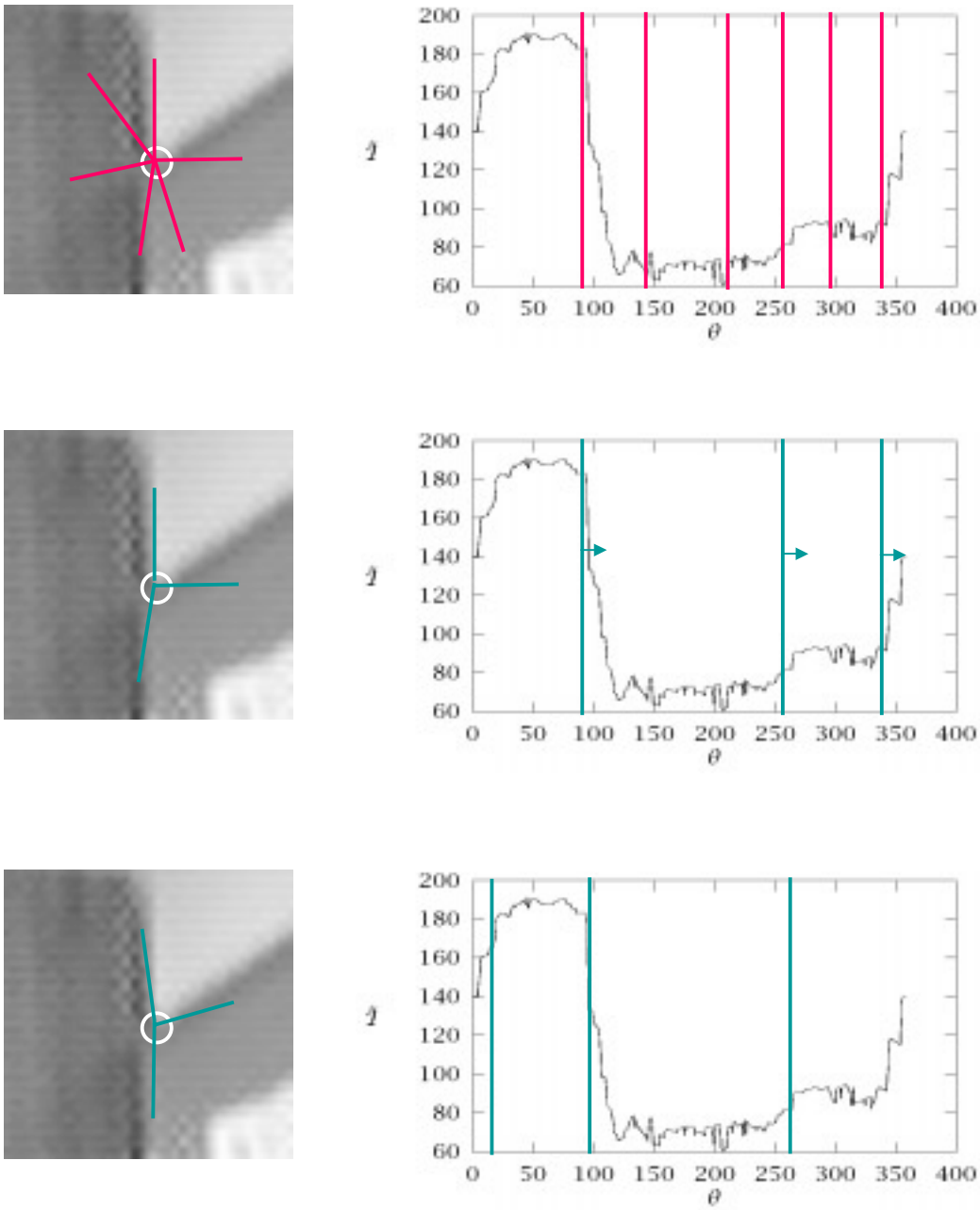

Figura 2.23: Dinámica del límite de una sección angular.

Algoritmo SEGLINEAL

Entrada: $\hat{\mathcal{I}}$: imagen de entrada, R_1, R_0: radios máximo y mínimo de la unión.
Salida: Uniones detectadas. Cada unión tendrá asociada los límites de las secciones angulares.

(a) Se aplica el detector de esquinas seleccionado. El resultado serán los posibles candidatos a unión.

(b) **Para cada** candidato a unión **hacer**:

i. Realizar la transformación del dominio circular a una dimensión.
ii. Inicializar la segmentación: calcular las aristas en una dimensión mediante SUSAN.
iii. Calcular los parámetros $\{\alpha_i\}$. En nuestro caso tenemos que encontrar la media y la varianza dentro de la sección angular.
iv. Aplicar la dinámica de Γ, minimizando la función de energía.
v. Ejecutar 2(b)iii, 2(b)iv iterativamente hasta que el movimiento de Γ converja. Entonces ir a 2(b)vi.
vi. Unir dos regiones adyacentes de tal manera que la unión provoque descenso de energía. Si se produce unión ir a 2(b)iii. Si el número de secciones es menor de dos, eliminar el candidato actual y pasar al siguiente.
vii. Si la unión no cumple las restricciones de no existencia de unión o esquina recta eliminar este candidato de la lista de uniones y pasar al siguiente candidato.

(c) **FinPara**

Figura 2.24: Algoritmo de segmentación lineal mediante competición de regiones.

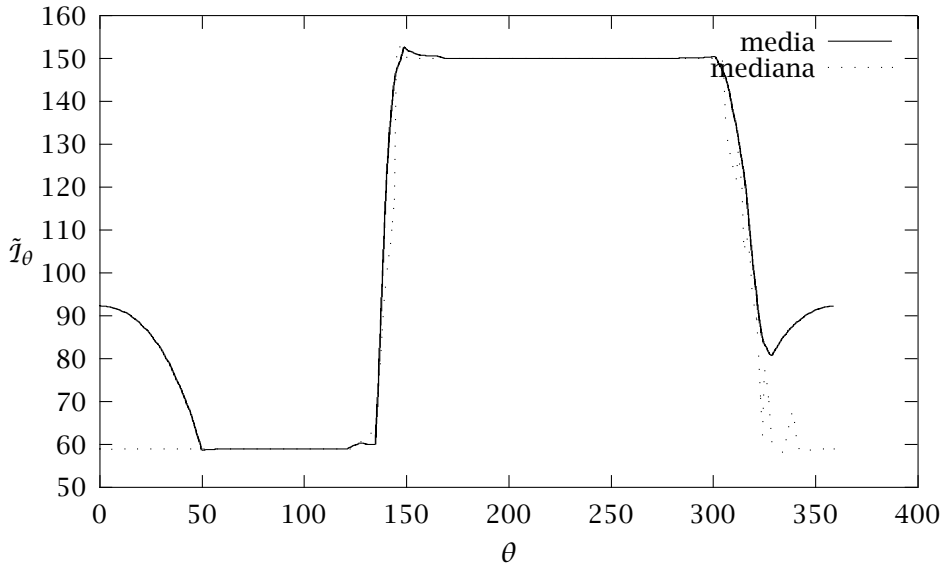

Figura 2.25: Error producido en la discretización.

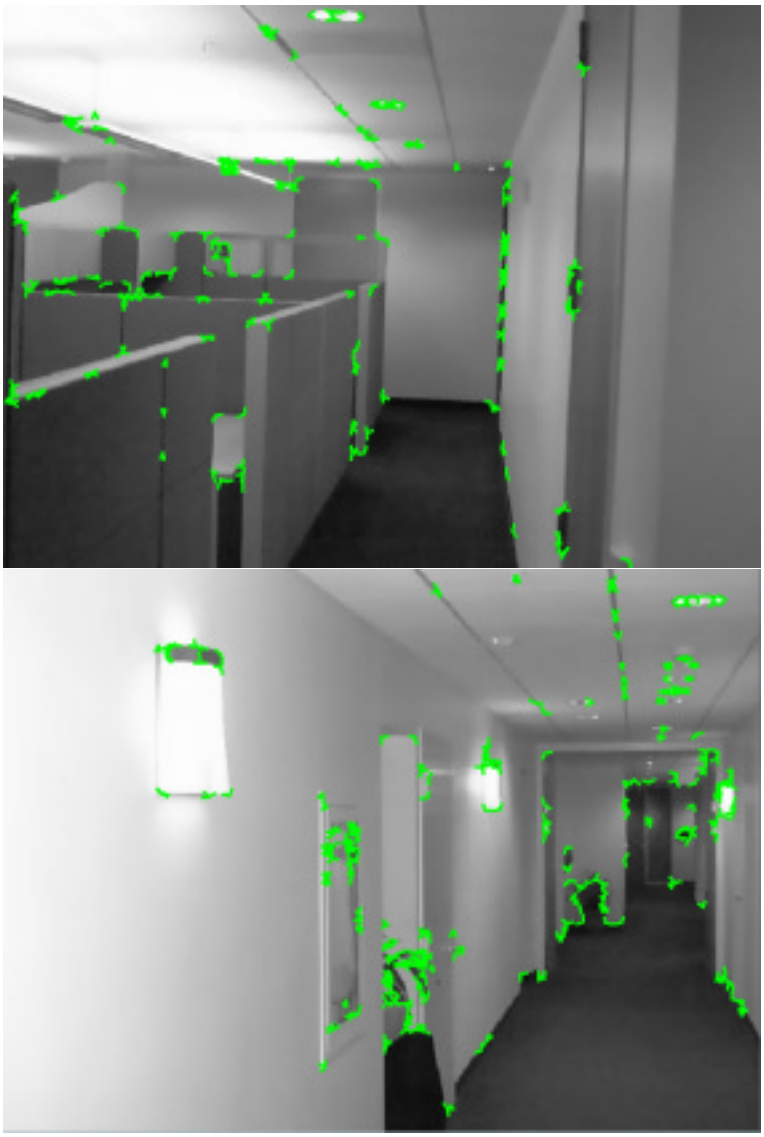

Figura 2.26: Resultado de aplicar el clasificador de uniones.

Figura 2.27: Resultado de aplicar el clasificador de uniones.

2.5 Clasificación de uniones mediante modelado de aristas

El método de segmentación lineal se basa en un criterio estadístico para medir la homogeneidad de regiones. Una alternativa, también bayesiana, a la homogeneidad de regiones consiste en medir la respuesta de los filtros de detección de aristas. Para ello nos basamos en recientes estudios [Konishi *et al.*, 1999] orientados a evaluar la calidad de dichos filtros.

Modelado de aristas

Supongamos un filtro local, como por ejemplo un filtro de gradiente. Si clasificamos previamente los píxeles de una imagen es posible extraer una caracterización estadística del filtro determinando la respuesta de éste cuando se encuentra en una arista real y cuando no lo está. Esta respuesta del filtro se suele cuantificar para que tome un número discreto de valores. En la Figura 2.29 se muestra las estadísticas obtenidas en el trabajo antes referenciado cuando el píxel se encontraba en una arista o fuera de ella, con una discretización de 20 valores. También, la Figura 2.30 muestra la función logaritmo del ratio entre estas dos funciones. Estas distribuciones nos proporcionan la probabilidad de que dada una respuesta, ésta corresponda a una arista o no. Denotaremos estas probabilidades como $P_{on}(E_{\vec{u}})$ y $P_{off}(E_{\vec{u}})$, respectivamente.

El filtro que se va a utilizar es muy simple: se trata de utilizar un filtro de gradiente seguido de lo derivada de una gaussiana con desviación estándar 1: $\vec{\nabla} G_{\sigma=1} * I$. Las imágenes resultantes una vez aplicado el filtro y calculados la magnitud y la orientación se muestra en la Figura 2.28. Este método se utilizará en éste y los siguientes capítulos como base para la formulación bayesiana propuesta.

Vamos a introducir el término de orientación de las aristas en las probabilidades. Tenemos $P_{on}(\vec{E}_{\vec{u}}|\beta)$ y $P_{off}(\vec{E}_{\vec{u}})$, donde $\vec{E}_{\vec{u}} = (E_{\vec{u}}, \phi_{\vec{u}})$, β es la orientación real de la arista y $\phi_{\vec{u}}$ es la dirección del gradiente medida en el punto $\vec{u}$. A continuación, realizan una suposición de factorización que $P_{on}(\vec{E}_{\vec{u}}|\beta) = P_{on}(E_{\vec{u}})P_{ang}(\phi_{\vec{u}} - \beta)$ y $P_{off}(\vec{E}_{\vec{u}}) = P_{off}(E_{\vec{u}})U(\phi_{\vec{u}})$. $P_{ang}(\cdot)$ codifica una medida de la distancia de la orientación de la arista $\sigma_{\vec{u}}$ a una orientación dada β. P_{ang} está basada en datos experimentales y tiene picos en 0 y π. La Figura 2.30 muestra el tipo de función utilizada. $U(\cdot) = 1/2\pi$ es una distribución uniforme.

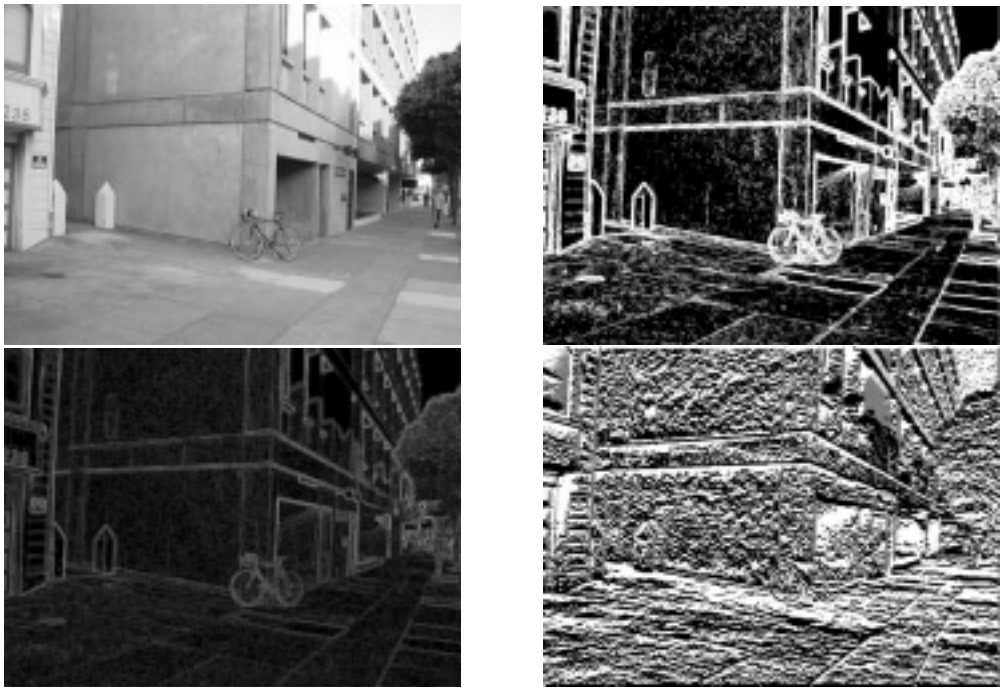

Figura 2.28: Arriba: Imagen original y el resultado de calcular el $\log(P_{on}/P_{off})$ en cada píxel. Abajo: Cálculo de los valores de orientación (izquierda) y magnitud (derecha).

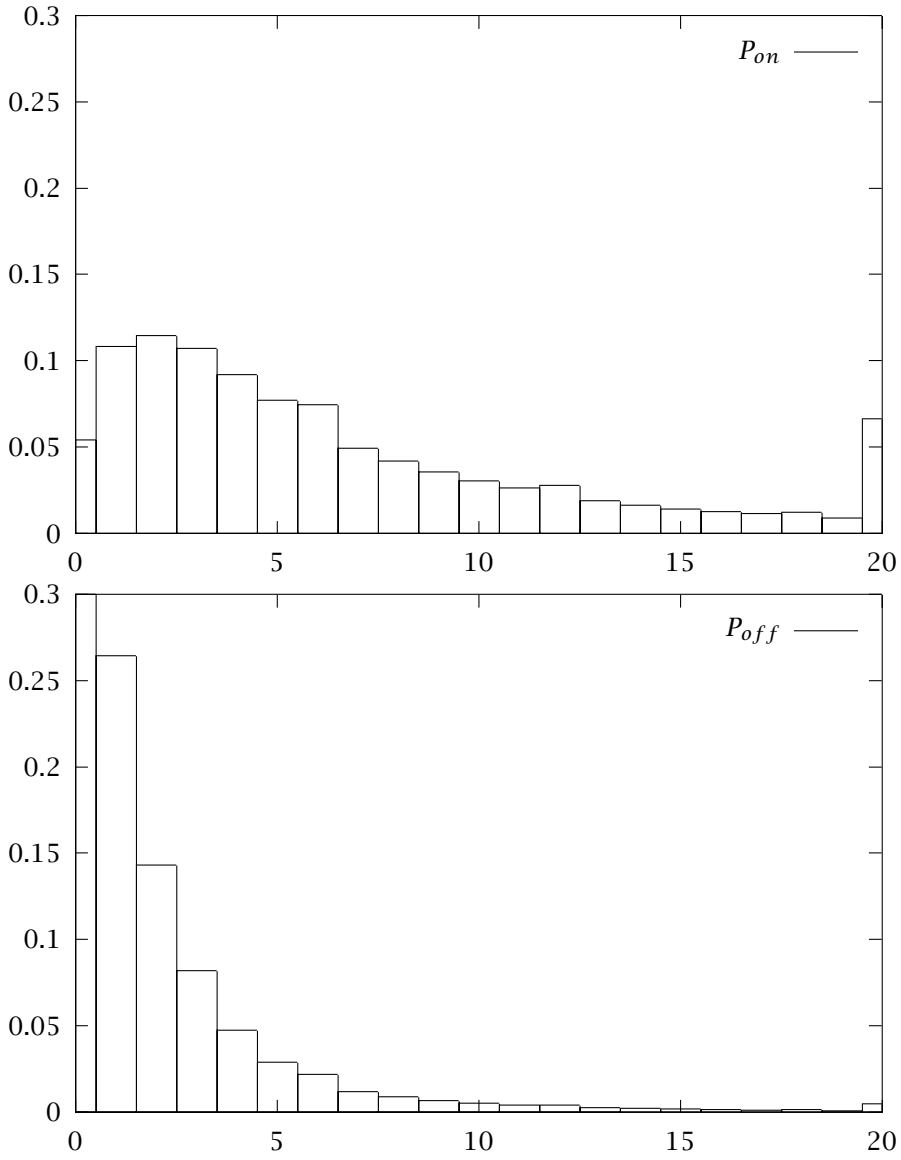

Figura 2.29: Estadísticas de las aristas: P_{on} arriba, P_{off} abajo.

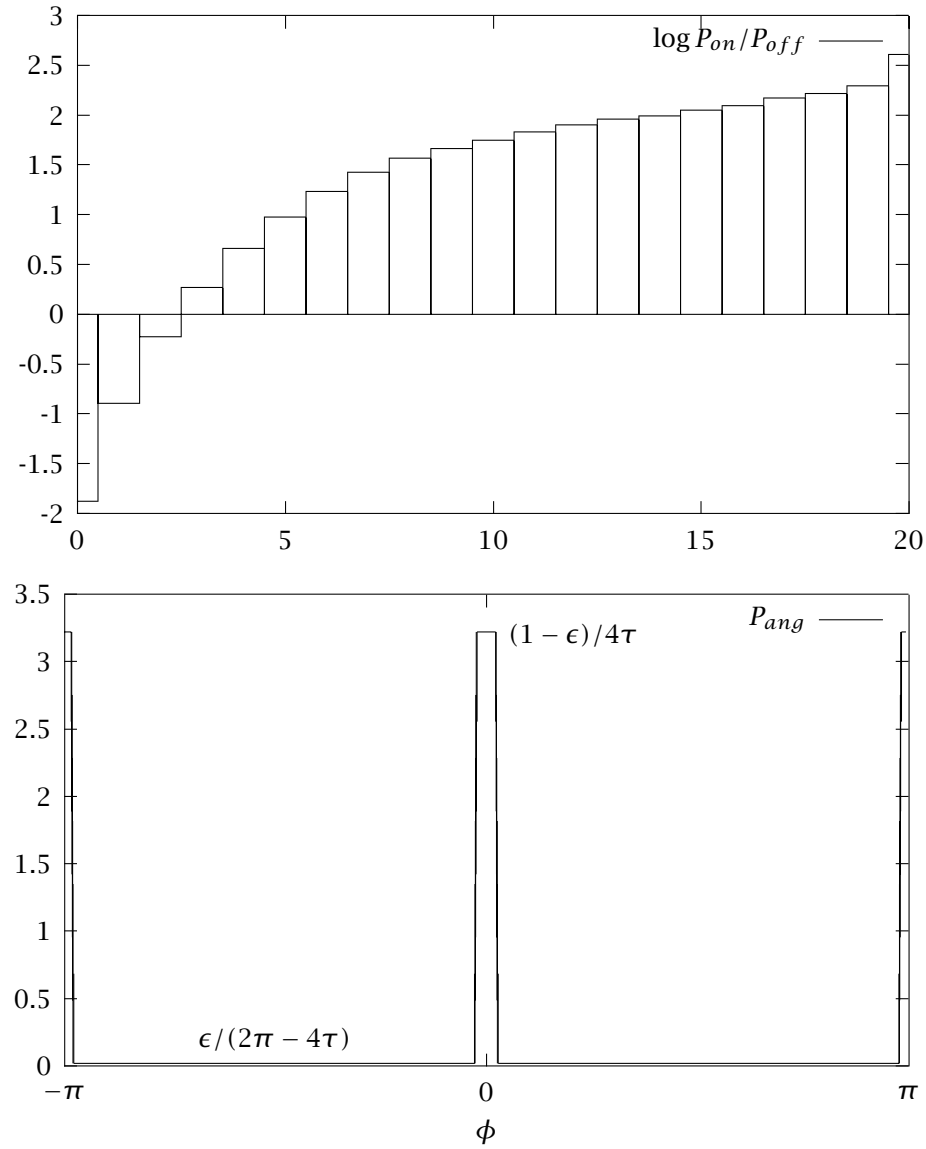

Figura 2.30: Gráfica del log de P_{on}/P_{off} (arriba) y de la función utilizada para P_{ang} (abajo).

Clasificación de uniones

Dado este modelo variamos la forma de obtener los vectores de acumulación, es decir, de realizar la transformación de dos dimensiones a una. Para ello haremos uso de la siguiente ecuación:

$$\hat{\mathcal{I}}_\theta = \frac{1}{(R_1 - R_0)} \sum_{i=1}^{N} l_i \cdot \log \frac{P_{on}(\vec{E}_{\vec{u}}|\beta)}{P_{off}(\vec{E}_{\vec{u}})} \tag{2.8}$$

donde $\vec{E}_{\vec{u}} = (E_{\vec{u}}, \sigma_{\vec{u}})$ siendo $E_{\vec{u}}$ la magnitud del gradiente y $\sigma_{\vec{u}}$ la orientación del mismo en el píxel $\vec{u}$, y siendo β el ángulo de la normal a θ.

Una vez hemos realizado la transformación a una dimensión, procedemos a marcar un umbral de corte, previamente definido, para detectar los picos de la función. Estos picos nos determinan los límites de las secciones angulares. Por último se realiza una búsqueda de máximo local para localizar las posiciones de dichos límites. El perfil obtenido después de la transformación se muestra en la Figura 2.31.

La Figura 2.32 muestra el algoritmo de detección de los límites angulares mediante la información de modelado de aristas. Las Figuras 2.33 y 2.34 muestran el resultado de aplicar este método de clasificación de uniones.

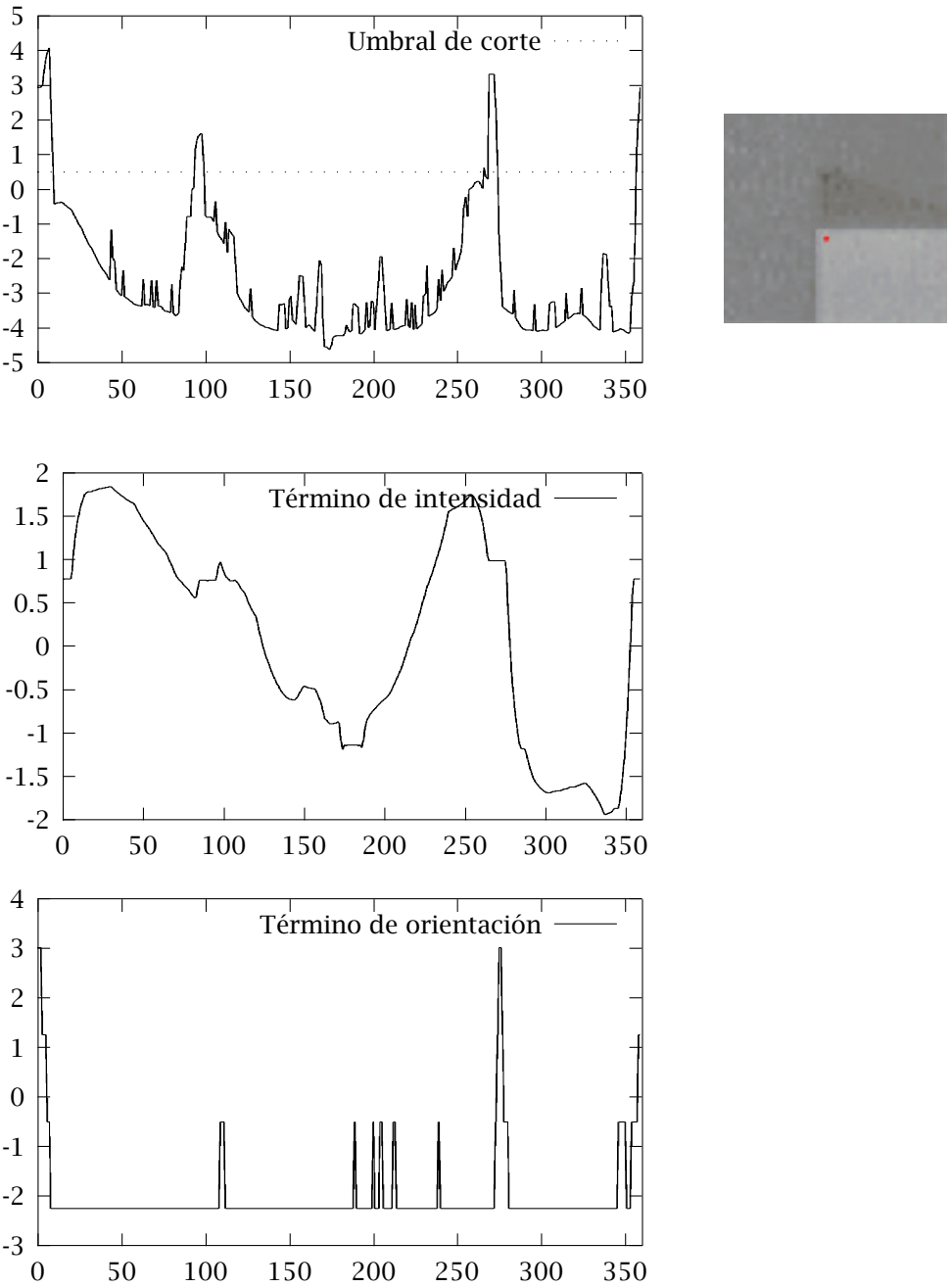

Figura 2.31: Transformación de la imagen con el método de P_{on}/P_{off}.

Algoritmo MODELARISTAS

Entrada: $\hat{\mathcal{I}}$: imagen de entrada, R_1, R_0: radios máximo y mínimo de la unión.
Salida: Uniones detectadas. Cada unión tendrá asociada los límites de las secciones angulares.

1. Se aplica el detector de uniones seleccionado. El resultado serán los posibles candidatos a ser unión.
2. Se aplica el filtro gausiano a la imagen de entrada. El resultado serán dos imágenes: la magnitud del gradiente y su orientación.
3. **Para cada** candidato a unión **hacer**:
   (a) Realizar la transformación del dominio circular a una dimensión. La transformación se realiza aplicando la Ecuación 2.8.
   (b) Aplicar un umbral de corte. Todos aquellos valores inferiores al umbral pasan a tener valor cero.
   (c) Buscar valores máximos locales. Estos máximos serán las posiciones de los límites angulares.
   (d) Si la unión o algunos de sus límites no cumple las restricciones de no existencia de unión o esquina recta eliminar la unión o el límite correspondiente.
4. **FinPara**

Figura 2.32: Algoritmo de detección de uniones mediante modelado de aristas.

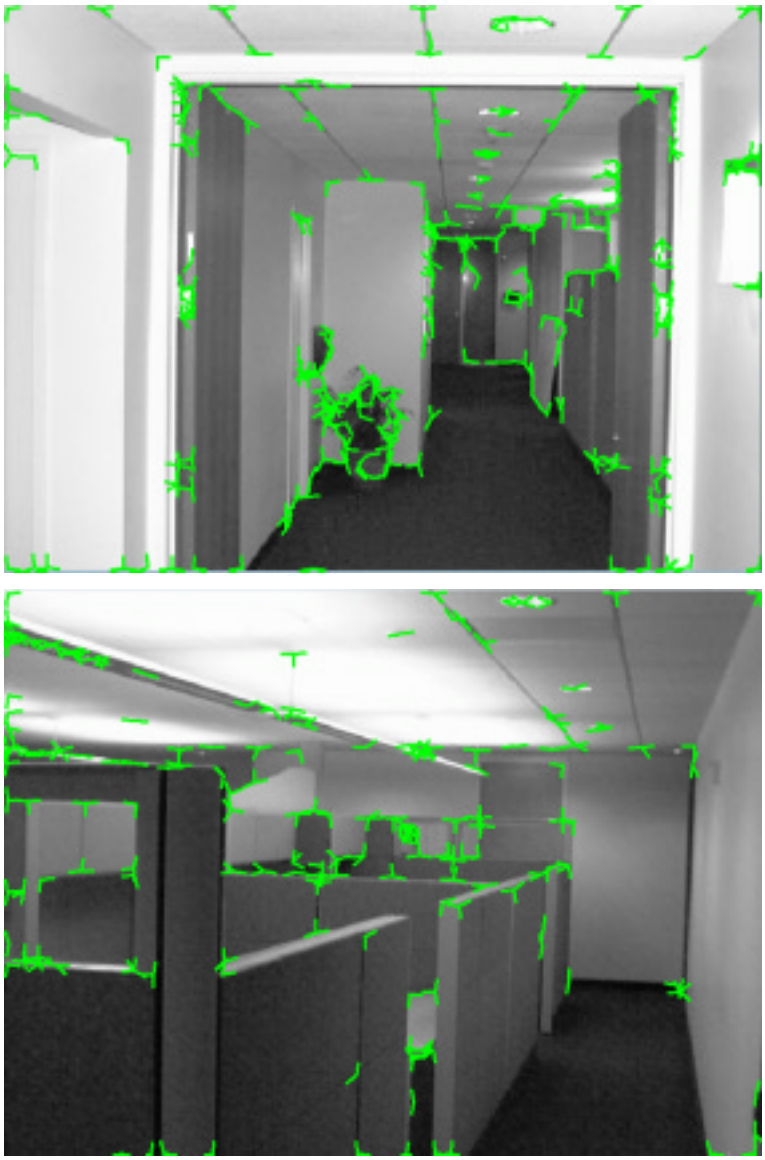

Figura 2.33: Resultado de aplicar el método de modelado de aristas.

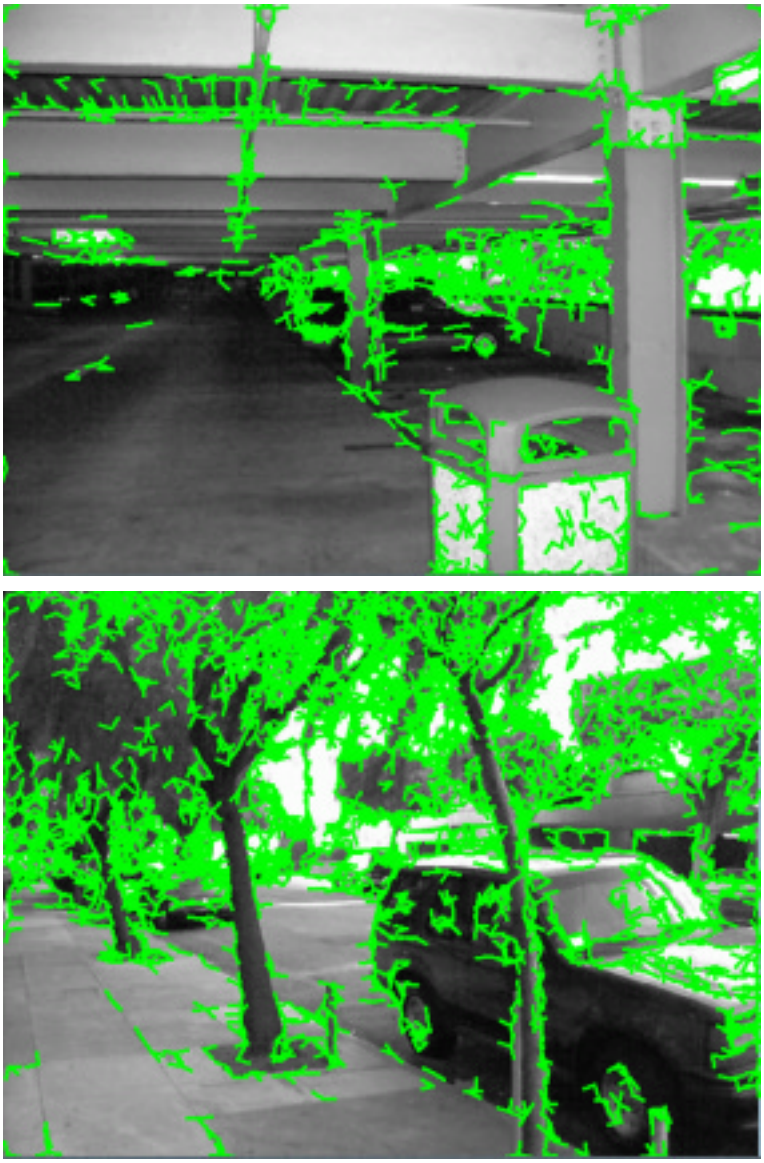

Figura 2.34: Resultado de aplicar el método de modelado de aristas.

2.6 Discusión

Hemos realizado comparativas entre los tres métodos presentados: el de Kona, el algoritmo de segmentación lineal y el de modelado de aristas. Los tres métodos se probaron en una secuencia de interior, en un escenario por donde se desplazaría un robot. La secuencia constaba de 18 instantáneas. También se probó en imágenes de exterior. Todas las imágenes tenían tamaño 640x480. Los algoritmos se implementaron en C++, utilizando la librería de procesamiento de imágenes Vista. Los experimentos se realizaron en un Pentium II 266 MHz bajo Linux.

En cuanto al error angular cometido en la obtención de los límites angulares, se obtiene un error medio de 15º para Kona, frente a los 10º obtenidos por los dos métodos propuestos. El algoritmo de modelado de aristas presenta un menor tiempo de procesamiento. La media del tiempo de computación fue de 4 segundos. A continuación, tenemos el algoritmo de segmentación lineal con 7 segundos. Por último, el método de Kona presentó un tiempo de procesamiento medio de 22 segundos.

El método de modelado de aristas es más sensible a la longitud del radio debido a su formulación basada en características locales. Necesita un radio mayor para acumular evidencia, lo que provoca errores como los descritos en la Sección 2.3. Sin embargo, este método presenta un número menor de límites no detectados frente a los otros dos. También presenta un mayor número de falsos positivos, límites que en realidad no lo son.

El algoritmo de segmentación lineal responde más eficientemente a la detección de los límites. Sin embargo, es más sensible a ruido.

Los tres métodos son poco robustos, y en la misma medida, al error por la mala localización del centro. Esta situación es bastante lógica al formularse los tres en el mismo dominio y con la misma transformación.

Resumiendo, en este capítulo hemos presentado dos nuevos métodos para la clasificación de uniones. Estos métodos hacen uso de operadores de detección de puntos característicos, es decir, de puntos donde la curvatura de los contornos es alta. También son puntos característicos aquellos puntos donde convergen varias aristas. Hemos comparado dos de los métodos de detección de dichos puntos, el operador de SUSAN y el de Nitzberg. Hemos determinado en qué situaciones utilizaremos uno y otro.

Definimos qué vamos a considerar una unión, presentando sus parámetros, que serán los que tendremos que encontrar. Para ello definimos los dos nuevos métodos mediante un esquema de minimización de función de energía. Ambas formulaciones tienen su fundamento en un esquema bayesiano.

Hemos realizado experimentos para comprobar la eficacia de los métodos propuestos, presentando resultados obtenidos al aplicarlos sobre imágenes tanto de

interior como de exterior. Los dos métodos presentan buenos resultados en imágenes de interior. Sin embargo, cuando procesamos imágenes de exterior, y debido a la aparición de objetos no estructurados (árboles, carteles, texturas, etc.) el número de uniones detectadas es muy elevado, produciéndose esto por la cantidad de puntos candidatos detectados por los operadores. También hemos observado que los métodos propuestos son muy dependientes de la localización del centro de la unión. Malas localizaciones de dicho centro provocan errores en la clasificación de los límites angulares.

Hemos analizado la eficiencia de estos métodos propuestos frente a Kona, obteniéndose un error y un tiempo de procesamiento menor cuando aplicamos nuestros operadores. El algoritmo de segmentación lineal detecta un menor número de falsos positivos. Sin embargo, el algoritmo de modelado de aristas tiene un tiempo de computación menor y deja un menor número de límites sin detectar.

En algunas situaciones se detectan límites inexistentes. Es el caso de cuando tenemos una región no perteneciente a la unión que se encuentra en el dominio de ésta. Este caso no se puede detectar mediante los métodos propuestos, y una clara ampliación es la prolongación de la unión para buscar otra unión en la dirección que marca el límite de la sección angular. Esta propuesta es la que desarrollaremos en el siguiente capítulo.

Capítulo 3

AGRUPAMIENTO DE UNIONES

Un segundo paso en nuestra propuesta es la creación de un mapa planar mínimo que describa la imagen y la relación entre las uniones. En esta parte tomaremos como entrada las uniones clasificadas con el detector de uniones descrito anteriormente. A partir de dichas uniones pretendemos crear una estructura geométrica que describa las relaciones entre éstas y las aristas presentes en la imagen. Por supuesto, cuanta más información (mayor número de uniones y de aristas entre estas) mejor definida estará la imagen, pero buscaremos el mínimo número de uniones y aristas necesario para realizar una descripción aceptable, en términos de su ajuste a la geometría de la imagen, con objeto de simplificar la complejidad de tareas subsiguientes. En este sentido, aplicaremos el criterio de longitud de descripción mínima.

El esquema clásico de agrupamiento utilizado en visión computacional [Lowe, 1985], [Trucco y Verri, 1998], [Ullman, 1996] consiste en agrupar aristas por proximidad y por similitud en orientación y longitud. Sin embargo, los elementos a agrupar en nuestra propuesta son las uniones. Pretendemos explotar la información de conexión que nos viene dada por el gradiente de la imagen.

Como ya hemos adelantado previamente, los problemas que nos vamos a encontrar una vez detectadas las uniones son los siguientes:

Errores de localización Pueden aparecer uniones que no se corresponden con ninguna unión real en la imagen. Debido a la presencia de ruido existe una posibilidad de detectar uniones en posiciones erróneas. También la discretización de la imagen juega un papel importante en la detección de uniones erróneas.

Errores de clasificación Algunas uniones son ficticias. Esto surge también debido a la discretización de la imagen: algunos objetos cuyos bordes son curvos provocan que se detecten uniones en dichos bordes. Realmente existe una arista debajo, pero el eliminar dicha unión no implica perder información.

Falsos negativos Es posible que no se detecten ciertas uniones. Puede ser debido a la no detección del candidato por parte del filtro de detección de puntos característicos o bien por que el algoritmo de clasificación de uniones desecha el candidato por tratarse de una arista terminal o por problemas derivados de la pérdida de gradiente.

El método a utilizar para el agrupamiento de uniones tiene que tener todo esto en cuenta y tratar de solventar las problemáticas presentadas.

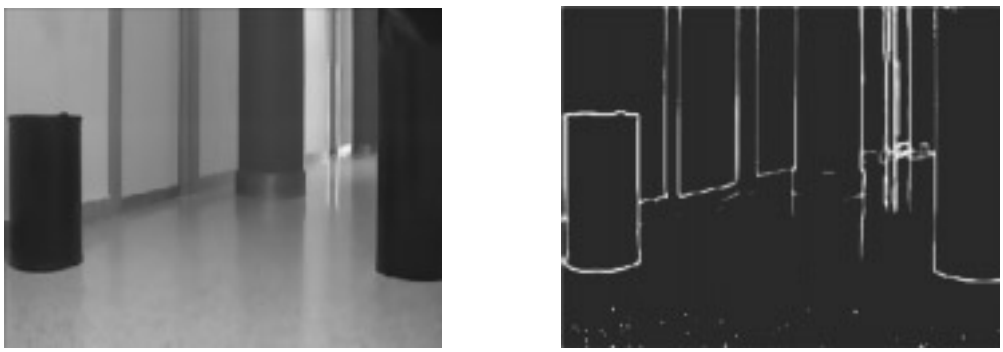

Figura 3.1: Imagen original (izquierda) junto con el resultado de calcular el gradiente (derecha).

Estamos interesados en encontrar *caminos que conecten uniones*, es decir, caminos a lo largo de aristas que conecten uniones, siempre y cuando dichas aristas existan. El resultado de calcular el gradiente de una imagen se muestra en la Figura 3.1. Vamos a enfocar el problema del agrupamiento como la búsqueda de un camino entre dos uniones. Si realmente existe una arista entre dos uniones, la magnitud del gradiente de la imagen nos proporcionará dicho camino.

En este capítulo vamos a presentar un método de búsqueda de caminos que utiliza la información del gradiente de la imagen para establecer el soporte de arista. Primero, veremos los métodos previos utilizados para el cálculo del ángulo de orientación en la Sección 3.1.

Como hemos comentado previamente, vamos a realizar la conexión entre dos uniones haciendo uso de la información de gradiente. Para la búsqueda de caminos entre uniones nos basaremos en la propuesta de [Yuille y Coughlan, 1999]

que extiende el trabajo previo de [Geman y Jedynak, 1996] para la detección y seguimiento de carreteras en imágenes tomadas desde satélite.

El resto del capítulo se organiza de la siguiente manera: la Sección 3.2 detalla cómo vamos a modelar un camino. En la Sección 3.3 explicaremos el algoritmo de seguimiento de caminos propuesto. Acabamos presentando el algoritmo completo con resultados y conclusiones del método.

3.1 Agrupamiento de características

El agrupamiento de características se puede definir como [Trucco y Verri, 1998]: *Dado un conjunto de características en una imagen, se trata de decidir que grupos de características forman parte del mismo objeto con mayor probabilidad, sin tener conocimiento de qué objetos estamos buscando.* Las características a utilizar pueden ser puntos, líneas, segmentos curvos, uniones, etc. El agrupamiento sirve de base en diversas tareas, como el reconocimiento de objetos [Ullman, 1996]. Un buen agrupamiento puede restringir las posibles interpretaciones de un objeto debido a que existe un número limitado de posibles combinaciones de dichas características como, por ejemplo, el caso de las uniones, tal como fue propuesto en [Waltz, 1975] y [Malik, 1987].

Se han propuesto diversas formas de llevar a cabo este agrupamiento de características. Un trabajo ya clásico es [Lowe, 1985], donde se discute el agrupamiento perceptual desde el punto de vista computacional y psicofísico. En [Sarkar y Boyer, 1993] se realiza una revisión de los distintos métodos sobre agrupamiento y organización perceptual. La mayoría de métodos realizan agrupamiento de aristas, utilizando criterios de proximidad, longitud de arista y orientación. Para el agrupamiento de uniones existe muy poco trabajo desarrollado. En [Brunstrom *et al.*, 1996] se propone un esquema de agrupamiento de uniones mediante seguimiento de caminos. Este método fija una dirección de búsqueda propuesta por una unión y utiliza un esquema de filtro de Kalman para seguir hasta la siguiente unión. La búsqueda se realiza utilizando un círculo que se va desplazando a saltos por el gradiente, buscando zonas con un gradiente similar. Los parámetros que rigen la búsqueda son la longitud de salto y el dominio del círculo.

En [Matas y Kittler, 1995] se propone un método que realiza detección, agrupamiento y etiquetado de las uniones en el mismo proceso. Para ello aplican un detector de aristas y utilizan los finales de aristas y las intersecciones entre ellas para realizar la detección y el etiquetado. Proponen un modelo de etiquetado de relajación probabilístico. El algoritmo presenta buenos resultados en imágenes con poco nivel de ruido y donde los contornos de los objetos están bien delimita-

dos. Además el tiempo de ejecución fue de 2 segundos con únicamente 90 aristas presentes.

3.2 Modelado de caminos

Definimos un camino P de longitud N como una colección de segmentos conectados consecutivos $p_1, p_2, \ldots, p_M$, donde M no es necesariamente constante para todos los caminos. El camino comienza en una unión con centro (x, y) y tiene un ángulo de salida θ, correspondiente a uno de las direcciones principales de la unión. Puesto que los puntos esquina vienen asociados a puntos de elevada curvatura, asumimos que la curvatura de los caminos entre uniones es suave. Para cada par de segmentos consecutivos definimos las variables $\alpha_1, \alpha_2, \ldots, \alpha_{M-1}$ como los ángulos que forman dichos segmentos. En concreto, α_i es el ángulo que forman los segmentos p_i y p_{i+1}. Utilizando el enfoque bayesiano de Yuille y Coughlan [Yuille y Coughlan, 1999], el camino óptimo P^* maximiza la siguiente función de energía:

$$\mathcal{E}_P(\{p_j, \alpha_j\}) = \sum_{j=1}^{N} \log\{\frac{P_{on}(f_{(p_j)})}{P_{off}(f_{(p_j)})}\} + \sum_{j=1}^{N-1} \log\{\frac{P_{\triangle G}(\alpha_{j+1} - \alpha_j)}{U(\alpha_{j+1} - \alpha_j)}\} \qquad (3.1)$$

El primer término de esta función se denomina *recompensa de intensidad* y está relacionado con la probabilidad de que el segmento se encuentre o no encima de una arista de la imagen. En la Sección 3.2.1 se definen dos posibles filtros a utilizar y la funciones de distribución obtenidas.

El segundo término es la *recompensa geométrica* y codifica la rigidez de los caminos. Variando esta componente podemos conseguir caminos más rígidos (rectas) o muy sinuosos. En la Sección 3.2.2 se define este término geométrico.

El logaritmo del ratio de las probabilidades representa el ratio de verosimilitud que minimiza errores de clasificación y tiene el siguiente comportamiento: si la función en el numerador tiene una valor superior al del denominador, para una misma instancia de la variable, el logaritmo devuelve un valor positivo. Sin embargo si ocurre al contrario, el valor del denominador es mayor, entonces devolverá un valor negativo.

3.2.1 Definición de la componente de intensidad

Para definir la componente de intensidad, haremos uso de dos filtros que aplicaremos a la imagen. El primero de ellos utiliza la información estadística descrita en la Sección 2.5. El segundo utiliza un filtro no lineal definido previamente en [Geman y Jedynak, 1996].

Filtro de modelado de aristas

Tal como se definió en la Sección 2.5 aplicaremos un filtro gausiano a la imagen: $\vec{\Delta}G_{\sigma=1} * I$. La respuesta de este filtro se modela mediante estadísticas obtenidas de imágenes. Estas estadísticas se obtuvieron en [Konishi *et al.*, 1999]. Dado un segmento p_i, el valor de P_{on} y P_{off} se obtiene de la siguiente manera:

$$P_{on}(p_i) = \sum_{\vec{u}\in p_i} P_{on}(E_{\vec{u}})P_{ang}(\phi_{\vec{u}} - \beta) \quad \text{y} \quad P_{off}(p_i) = \sum_{\vec{u}\in p_i} P_{off}(E_{\vec{u}})U(\phi_{\vec{u}}) \quad (3.2)$$

donde $\vec{u}$ son los puntos del segmento p_i, $E_{\vec{u}}$ es la magnitud del gradiente en el punto $\vec{u}$, β es el ángulo de la normal al segmento, $\sigma_{\vec{u}}$ es la orientación del gradiente en el punto $\vec{u}$ y $U(\cdot)$ es una distribución uniforme. La Figura 3.2 muestra las estadísticas obtenidas con este filtro y el logaritmo del cociente entre P_{on} y P_{off}.

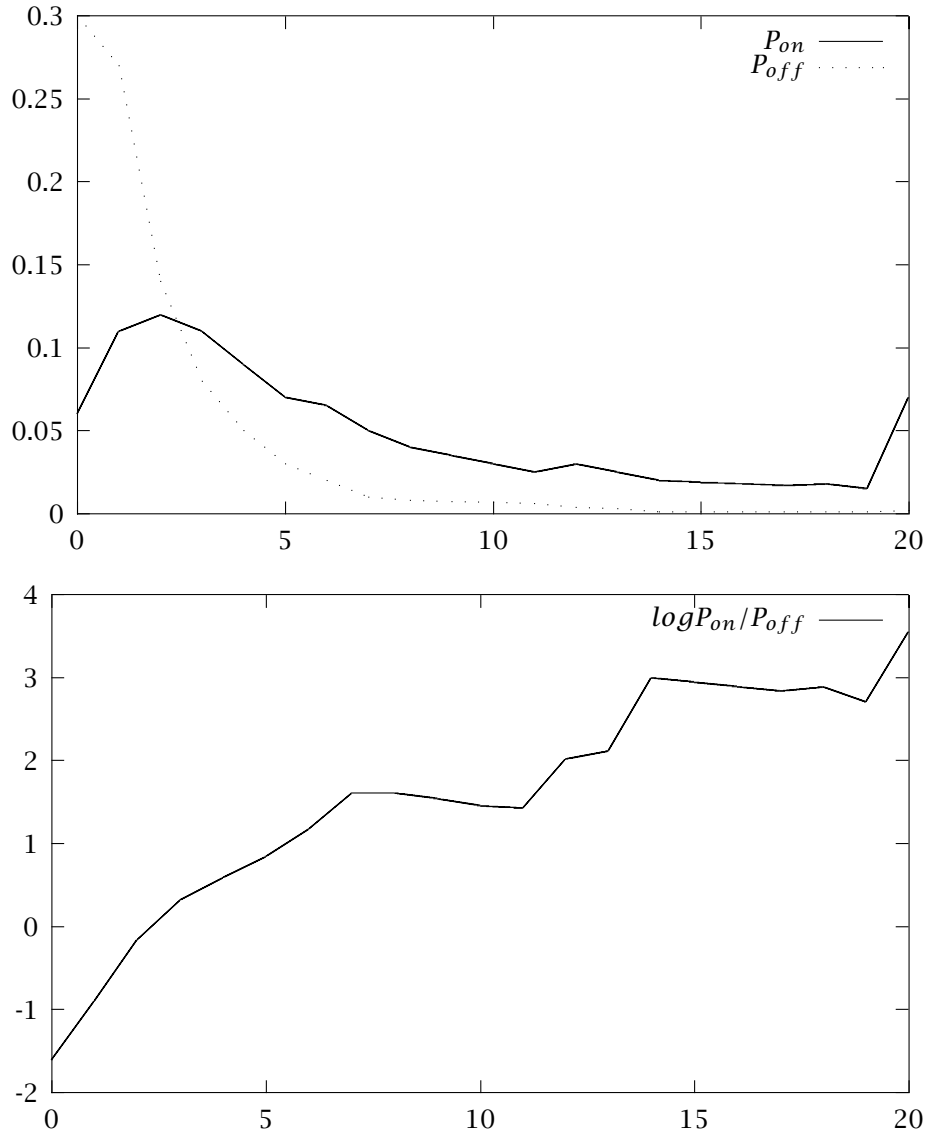

Figura 3.2: Estadísticas de las aristas: P_{on} y P_{off}, junto con el logaritmo del ratio entre las dos.

Filtro no lineal

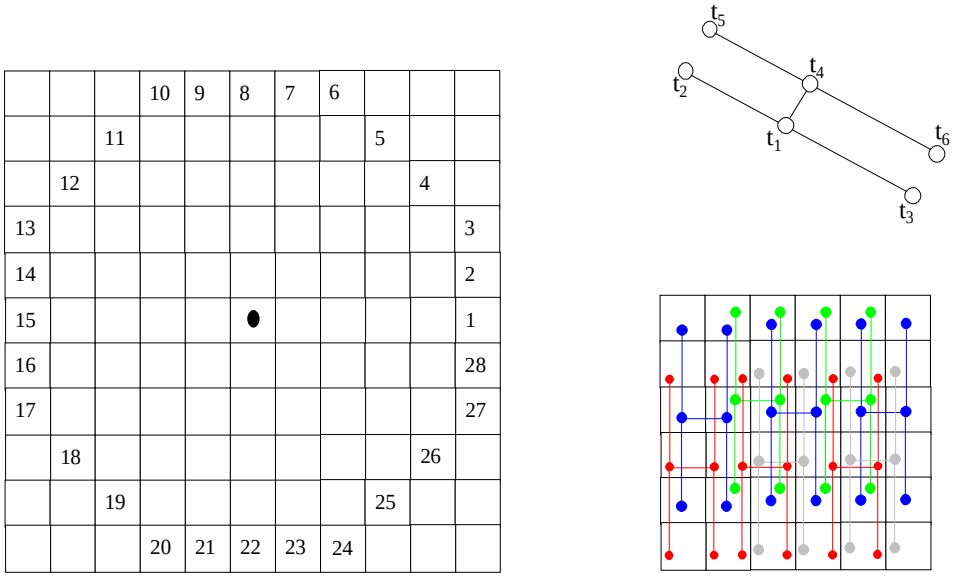

Figura 3.3: Las posibles configuraciones del filtro (izquierda), nomenclatura de los píxeles (derecha arriba) y los diez test realizados a lo largo de una determinada configuración (derecha abajo).

La probabilidad de que un segmento se encuentre encima de una arista se modela mediante una función de distribución $P(f_{(p_j)})$ sobre las respuestas de un filtro no lineal $f_{(p_j)} = \phi(|\nabla I(p_j)|)$, donde $|\nabla I(p_j)|$ es la magnitud del gradiente. La función de distribución $P(f_{(p_j)})$ se modela dependiendo de la posición relativa entre un segmento p_j y la arista, y se define en los siguientes términos:

$$P(f_{(p_j)}) = \begin{cases} P_{on}(f_{(p_j)}) & p_j \in P^* \\ P_{off}(f_{(p_j)}) & \text{en otro caso.} \end{cases} \tag{3.3}$$

donde $P_{on}(f_{(p_j)})$ y $P_{off}(f_{(p_j)})$ son las distribuciones de probabilidad de las respuestas de los segmentos encima y fuera de una arista. Estas funciones se obtienen empìricamente, calculando las estadísticas de las respuestas del filtro cuando localizamos el segmento encima de la arista y cuando lo localizamos fuera de ella. Para el diseño del filtro de intensidad nos hemos basado en el propuesto en [Geman y Jedynak, 1996], filtro que también utilizaron Yuille y Coughlan [Yuille y Coughlan, 1999] y que detallamos a continuación.

Primero aplicamos el operador de SUSAN para aristas sobre la imagen. Sobre el resultado aplicamos el siguiente test:

$$|I_{t_1} - I_{t_4}| < \min\{|I_{t_3} - I_{t_1}|, |I_{t_2} - I_{t_1}|, |I_{t_5} - I_{t_4}|, |I_{t_6} - I_{t_4}|\} \tag{3.4}$$

donde I_{t_i} indica el valor de intensidad del correspondiente píxel t_i (ver Figura 3.3). Este test lo realizamos por todos los píxeles a lo largo del segmento, hasta un total de 10. En la Figura 3.3 se muestran las distintas configuraciones utilizadas en la implementación así como un ejemplo de los filtros utilizados. Una vez realizados todos los test para un segmento, tendremos un número de respuestas afirmativas entre 0 y 10. Valores altos indicarán mayor presencia de arista. Valores bajos constatarán poca presencia de arista. Se han realizado 200 mediciones del filtro situando el segmento sobre una arista y otras 200 sobre fondo. Las funciones de distribución obtenidas, junto con el logaritmo del ratio entre las dos, se muestran en la Figura 3.4.

3.2.2 Definición de la componente geométrica

La recompensa geométrica permite codificar la rigidez del camino. En determinadas ocasiones podemos desear caminos completamente rectos y en otras podemos desear que la configuración del camino sea libre. Para ello tenemos la recompensa geométrica que está compuesta por dos funciones de distribución: $P_G(\alpha_{j+1} - \alpha_j)$ y $U(\alpha_{j+1} - \alpha_j)$. P_G modela una cadena de Markov de primer orden sobre las variables de orientación α_j. Esta función se define mediante una exponencial negativa:

$$P_{\triangle G}(\triangle\alpha_j) \propto \exp\{-\frac{C}{2A}|\triangle\alpha_j|\} \tag{3.5}$$

donde: $\triangle\ \alpha_j = \alpha_{j+1} - \alpha_j$, A es el máximo ángulo entre dos segmentos consecutivos y C modula la rigidez del camino. Por otro lado, $U(\alpha_{j+1} - \alpha_j)$ es la distribución uniforme de la variación angular y se introduce para mantener las componentes geométrica y de intensidad en el mismo rango. En la Figura 3.5 se muestran estas funciones.

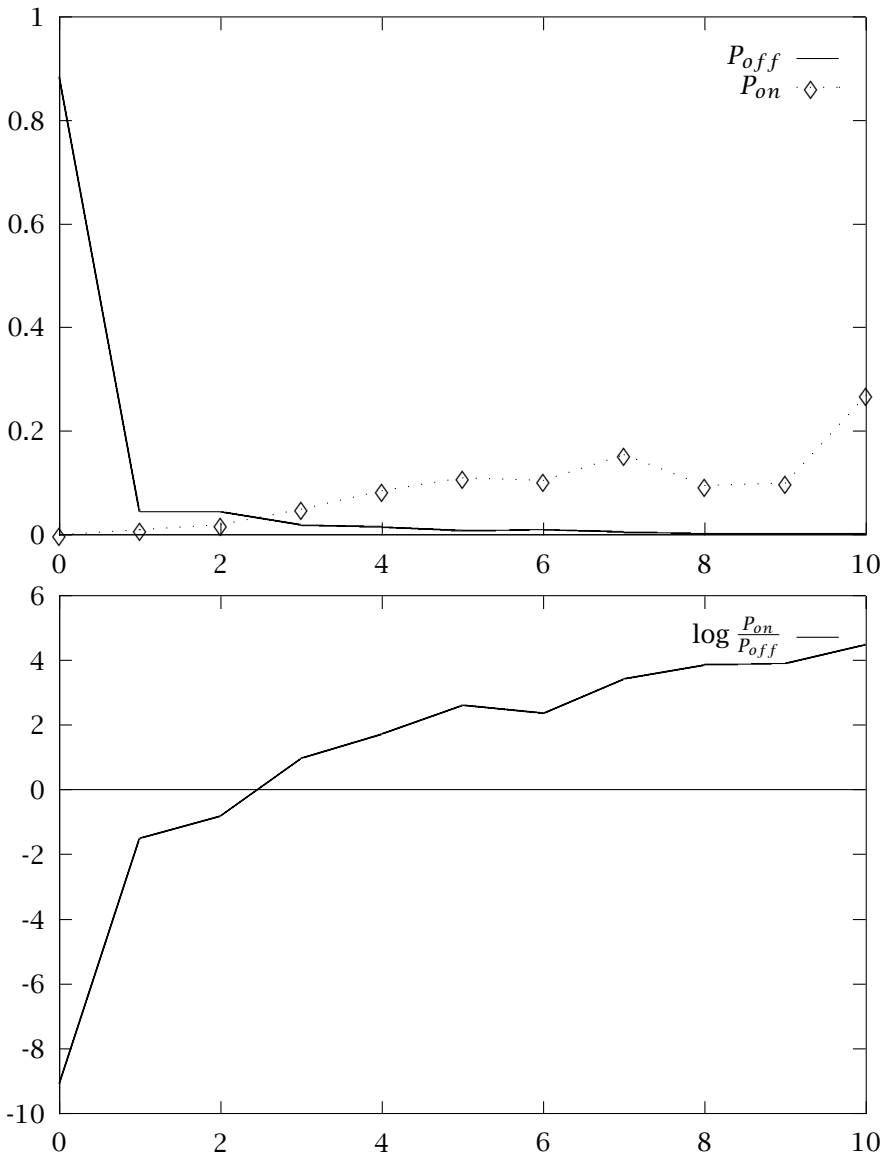

Figura 3.4: Funciones de distribución de probabilidad: P_{on} y P_{off}, junto con el logaritmo del ratio entre las dos.

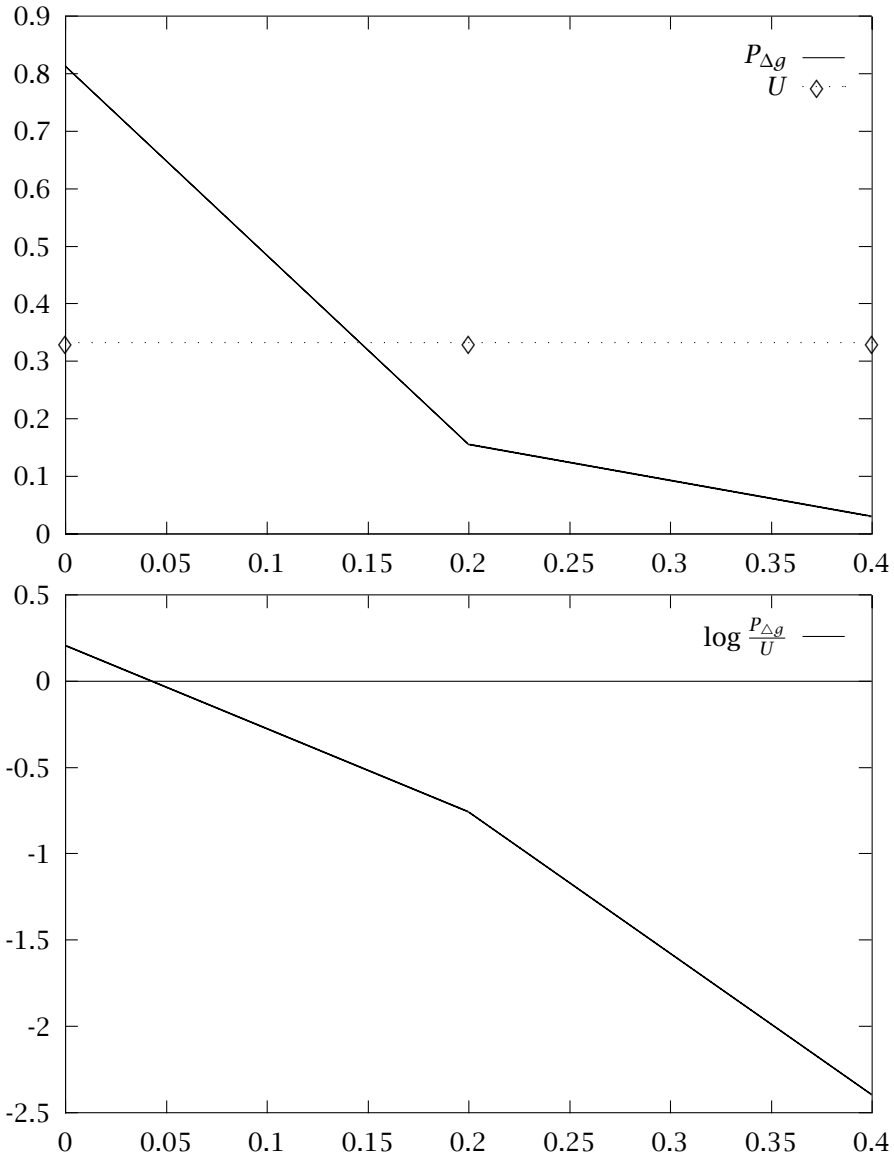

Figura 3.5: Funciones de distribución de probabilidad: $P_{\triangle G}$ y U, junto con el logaritmo del ratio entre las dos

.

3.3 Seguimiento de caminos

Encontrar caminos en escenas ruidosas y/o con fragmentación de aristas puede llegar a ser una tarea de difícil solución y esta tarea debe ser realizada en un corto intervalo de tiempo, especialmente cuando se imponen restricciones de tiempo real. Coughlan y Yuille [Coughlan y Yuille, 1998] recientemente han propuesto un método denominado A^* bayesiano que explota el conocimiento estadístico asociado a las componentes de intensidad y geometría. Este método surge de un análisis teórico previo [Yuille y Coughlan, 1997] sobre la conexión entre el algoritmo Twenty Question de Geman y Jedynak [Geman y Jedynak, 1996] y A^*, el algoritmo clásico utilizado en Inteligencia Artificial [Pearl, 1984]. A continuación se describe el enfoque genérico del algoritmo A^* bayesiano y su adaptación a la búsqueda de caminos entre uniones.

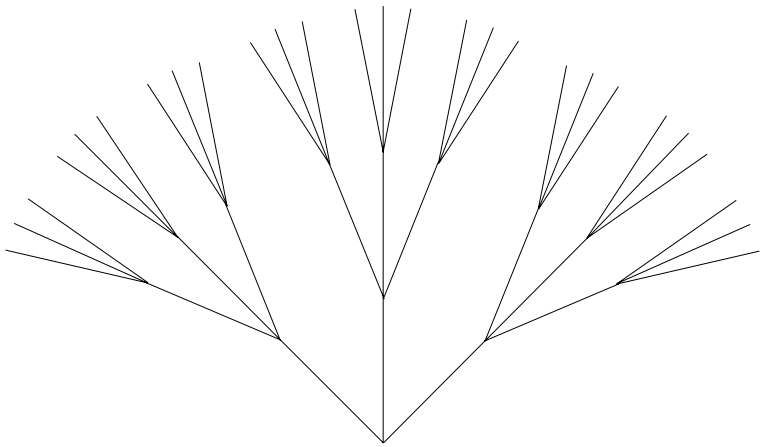

Figura 3.6: Ejemplo de árbol de búsqueda. El factor de ramificación Q es tres.

3.3.1 Algoritmo y restricciones

Dado el centro de una unión (x_0, y_0) y una orientación θ_0, el algoritmo explora un árbol cuya raíz es el centro de la unión y cada rama es un posible camino (ver Figura 3.6). Cada arista entre dos nodos se corresponde con un segmento. Cada segmento puede expandir Q sucesores, por ello tenemos Q^N caminos posibles y en consecuencia la búsqueda exhaustiva no es practicable. El algoritmo A^* bayesiano reduce la conducta conservadora del clásico A^* explotando el hecho de que

queremos detectar *un* camino en vez del encontrar el mejor de todos. En nuestro caso sabemos que puede existir un camino entre las uniones a unir. Como consecuencia de esto tenemos un único camino real frente a un gran número de caminos falsos. Podemos reducir la complejidad del problema eliminando (podando) caminos parciales cuya recompensa acumulada sea muy baja. El algoritmo evalúa las recompensas de intensidad y geométrica de los últimos N_0 segmentos del camino eliminando aquellos caminos cuya recompensa ponderada está por debajo de un cierto umbral, es decir:

$$\frac{1}{N_0}\sum_{j=N-N_0}^{N}\log\{\frac{P_{on}(p_j)}{P_{off}(p_j)}\} < T,\ o\ \frac{1}{N_0}\sum_{j=N-N_0}^{N}\log\{\frac{P_{\triangle G}(\triangle\alpha_j)}{U(\triangle\alpha_j)}\} < \hat{T} \tag{3.6}$$

donde T y $\hat{T}$ son los umbrales de intensidad y geométrico que modulan la conducta de poda del algoritmo. Estos parámetros establecen la recompensa (en los últimos segmentos) mínima que un camino necesita para sobrevivir y su valor no es arbitrario. Están relacionados con las distribuciones de probabilidad utilizadas para diseñar las recompensas de intensidad y geométricas. En [Coughlan y Yuille, 1998] se demuestra que estos valores umbral tienen que encontrarse en los siguientes intervalos:

$$-D(P_{off}||P_{on}) < T < D(P_{on}||P_{off}), -D(U_{\triangle G}||P_{\triangle G}) < \hat{T} < D(P_{\triangle G}||U_{\triangle G}) \tag{3.7}$$

donde D es la divergencia de Kullback-Leibler que nos mide lo diferentes que son dos distribuciones. Este distancia es siempre mayor o igual a cero, siendo cero cuando ambas distribuciones son la misma, y se define de la siguiente manera:

$$D(P_1||P_2) = \sum_{j=0}^{M} P_1(x_j)\log\frac{P_1(x_j)}{P_2(x_j)} \tag{3.8}$$

siendo M el número de elementos del dominio en el cual están definidas ambas distribuciones.

El algoritmo encuentra el mejor camino que sobrevive a la poda. El ratio de convergencia es $O(N)$. Los valores T y $\hat{T}$ se suelen escoger cercanos a los límites superiores, para que la poda sea mayor. Si las distribuciones P_{on} y P_{off} son muy similares, el algoritmo será conservativo y no podará demasiados caminos. El máximo rendimiento se consigue cuando ambas distribuciones son muy dispares. En definitiva, el grado de solape es inversamente proporcional a la potencia de discriminación del filtro de gradiente. El mismo razonamiento se sigue para $P_{\triangle G}$ and $U_{\triangle G}$.

La aplicación de este algoritmo en el contexto de agrupamiento de uniones motiva la extensión de la regla de poda. Consideramos la estabilidad de caminos

largos frente a los cortos. Los caminos más largos estarán más cerca del camino verdadero que los cortos debido a que han sobrevivido a un mayor número de podas. Por ello podemos eliminar aquellos caminos cuya longitud N_j sea menor que la del mejor camino hasta el momento N_{mejor} menos una constante, es decir, eliminaremos aquellos caminos que no cumplan:

$$N_j > N_{mejor} - zN_0$$

donde $z > 0$ es un parámetro que permite restringir o aumentar la poda. Valores altos de z provocan una mayor poda y un mayor riesgo de perder el camino y valores bajos conservan todos los caminos ralentizando el proceso. Dicho parámetro debe fijarse de forma experimental.

3.3.2 Finalización del camino

El algoritmo selecciona para expansión el mejor camino parcial hasta el momento y le aplica la regla de expansión, generando tres puntos de extensión. Consideramos que hemos alcanzado el final del camino cuando encontramos una unión en una pequeña vecindad alrededor del punto final del camino. Para comprobar si existe una determinada unión en una vecindad hemos utilizado *árboles de rango*, que son estructuras geométricas [de Berg *et al.*, 1997] que permiten realizar de forma eficiente consultas dentro de un rango. El coste de generar el árbol es $O(J \log J)$ donde J es el número de uniones detectadas. El coste de realizar una consulta es $O(K + \log J)$ donde K es el número de elementos devueltos. En el Apéndice B se detalla la construcción y consulta de esta estructura. Si el camino actual encuentra una unión, hemos alcanzado una unión, por lo que el algoritmo finaliza y conecta ambas uniones.

Por otro lado, la búsqueda puede finalizar sin encontrar ninguna unión. Esto sucede cuando todos los caminos parciales han sido eliminados. Uno a uno se van eliminando debido a la pérdida de alguna de las recompensas. Podemos distinguir entre dos situaciones:

- No existe camino real. Este situación se da cuando, por problemas del detector de uniones, algún límite de las secciones de la unión no se corresponde con una arista real. Esta situación se detecta por que el camino máximo alcanzado tiene un número de segmentos muy pequeño, muy cercano a N_0. En este caso consideramos que no hay soporte suficiente para declarar la presencia de camino. Este efecto se puede contemplar en la Figura 3.7 izquierda.
- Existe camino pero hemos llegado al final de él sin encontrar otra unión. Cuando el algoritmo de generación de camino alcance un punto en el cual

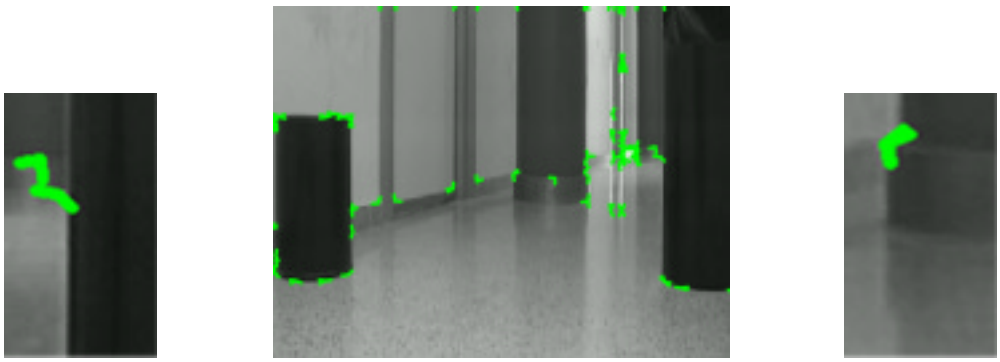

Figura 3.7: Situaciones a detectar cuando nos quedamos sin caminos.

la recompensa de arista en los últimos segmentos no supere el umbral se eliminarán todos los caminos, pues es posible que el detector de uniones no haya detectado la unión correspondiente. En este caso tendremos un camino lo suficientemente largo y almacenamos las coordenadas del punto final del camino como un posible candidato a ser unión. En la Figura 3.7 derecha vemos un ejemplo representativo de este caso. Este evento dispara la detección de uniones en el punto considerado para confirmar que no se trata de un falso negativo.

La Figura 3.8 muestra un ejemplo de generación de caminos. Partimos del límite de la sección angular de una unión y tomamos su ángulo de salida como inicialización del camino. A continuación comenzamos a explorar dicho camino aplicando las restricciones comentadas anteriormente. En el momento en que se encuentre una unión en una cierta vecindad alrededor del final del camino actual, podemos finalizar la búsqueda. Sólo mostramos el mejor camino hasta ese momento. Como se puede observar en la figura, hemos encontrado una unión y procedemos a marcar el límite de la unión entrante como visitado y conectamos ambas uniones. En la siguiente sección se comenta el algoritmo completo de agrupamiento de uniones.

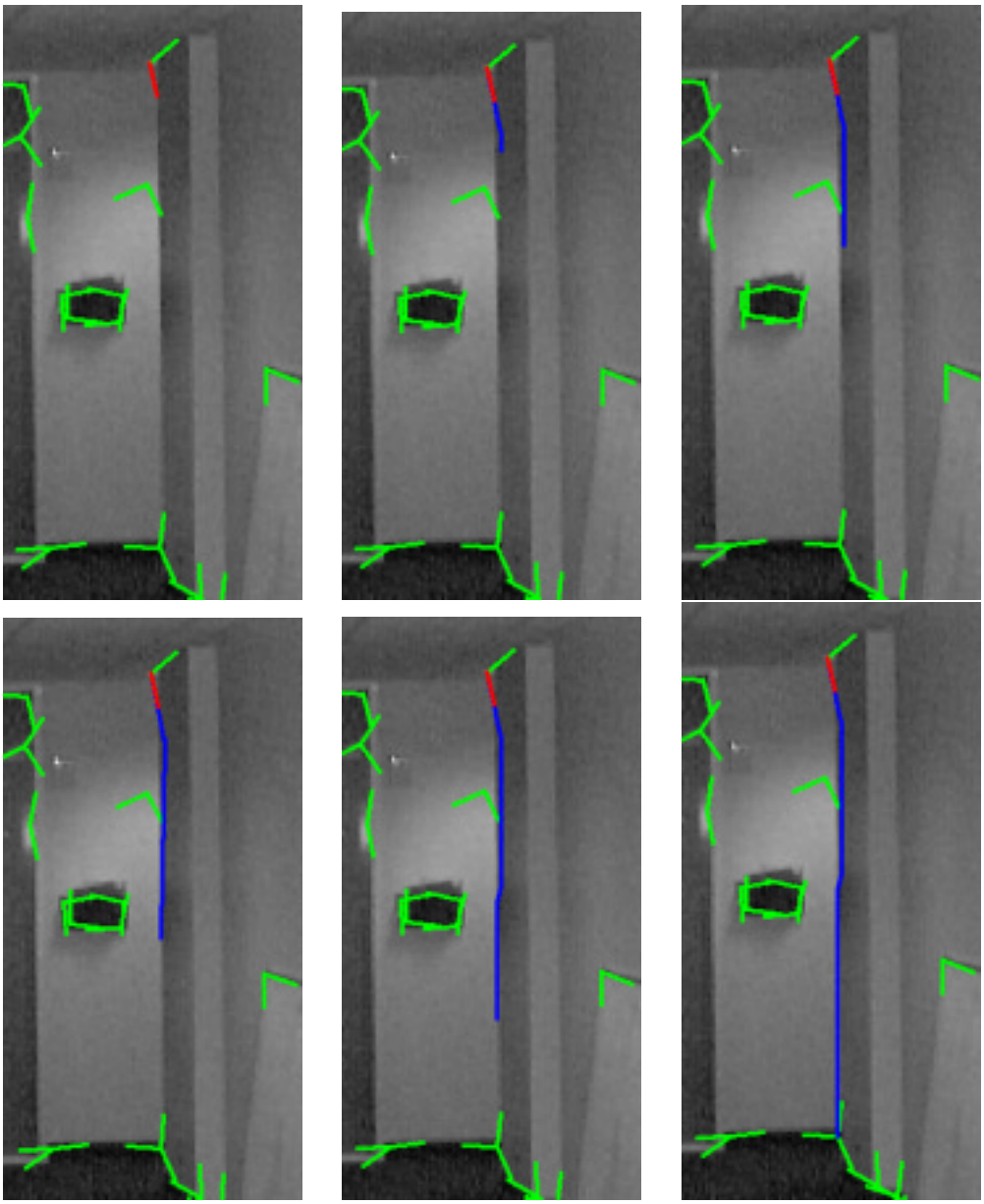

Figura 3.8: Ejemplo de inicialización, búsqueda y finalización de camino.

3.4 Agrupamiento de uniones

En esta sección detallamos el algoritmo de agrupamiento de uniones. El algoritmo toma como entrada el conjunto de uniones obtenidas con el detector de uniones propuesto anteriormente. Una vez procesadas todas las uniones devuelve un mapa de la escena, donde cada vértice del mapa es el centro de una unión y las aristas se corresponden con los caminos encontrados entre las uniones. Antes de detallar el algoritmo de agrupamiento pasamos a comentar el algoritmo de búsqueda de camino que será utilizado por el de agrupamiento para encontrar los caminos. El algoritmo se muestra en la Figura 3.9. El umbral N_T marca la longitud mínima que un camino debe tener para que sea considerado como tal.

En la Figura 3.10 se puede observar algunos caminos generados. El algoritmo mostrado en la Figura 3.11 es el que realiza el agrupamiento de las uniones visitando todas las uniones.

Usamos el etiquetado de los límites de las secciones angulares para evitar que se recorra el mismo camino dos veces: en un sentido y en el contrario. Esta redundancia proporciona robustez en el sentido de que si un camino no se obtiene en un sentido, se podría obtener en el contrario.

Este método genera una representación de nivel medio en forma de mapa o grafo de conectividad. Podemos usar la información contenida en los caminos para tareas de segmentación, seguimiento de características y reconocimiento. La Figura 3.12 muestra dos ejemplos de aplicación del algoritmo (siendo la componente de intensidad el filtro no lineal) a las uniones obtenidas con el detector de uniones propuesto. Otros ejemplos son los mostrados en las Figuras 3.13, 3.14 y 3.15, en las cuales se ha utilizado la componente de intensidad del filtro de modelado de aristas. Los parámetros utilizados en la implementación fueron:

- Factor de ramificación: $Q = 3$.
- Número de segmentos utilizados para comprobar umbral de recompensa: 5.
- Ángulo de generación de hijos: $\pm 0.2 = \pi/15$.
- Rigidez: 5.0.
- Divergencias calculadas: $-D(P_{off}||P_{on}) = -5.980$, $D(P_{on}||P_{off}) = 3.198$, $-D(U_{\triangle G}||P_{\triangle G}) = -0.752$, $D(P_{\triangle G}||U_{\triangle G}) = 0.535$.
- Umbrales utilizados para la poda: $T = 0.0$, $\hat{T} = 0.40$.
- Umbral angular $\theta_f = 30$.

Algoritmo de búsqueda de camino ALG_ABAYES

Entrada: x_c, y_c, θ centro y ángulo de salida del camino a buscar. $\mathcal{I}_e$ imagen de gradiente.
Salida: Verdadero si ha encontrado camino y falso si no lo ha encontrado. También devuelve el Camino y Union_detectada.

Generar Q caminos. Cada camino está compuesto de un segmento que parte del punto pasado como parámetro. El segmento está compuesto por dos puntos. El primero de ellos tendrá como punto inicial (x_c, y_c) y como final $(x_c + l_s * \cos\theta, x_c + l_s * \sin\theta)$. Los dos siguientes caminos se generan de la misma forma pero variando en $\pm\beta$ grados el ángulo de salida y así sucesivamente.

El algoritmo utiliza una lista de caminos LF. Al principio esta lista está compuesta por los Q caminos generados en el paso anterior. Cuando insertamos en esta lista lo hacemos de forma ordenada según el valor de E de cada camino, siendo el primer elemento de la lista el de menor E. De esta manera nos evitamos el reordenar la lista.
Mientras siempre **hacer** /* Bucle infinito */
Si LF está vacía **entonces**
Si longitud(Mejor_camino) $< N_T$ **entonces**
Devolver Falso.
Sino
Devolver Cierto, Mejor_camino. No hemos alcanzado ninguna unión. El punto de terminación de Mejor_camino es un posible candidato a unión.
FinSi
FinSi
Seleccionamos aquel elemento de LF con menor valor de E y lo eliminamos de la lista. Al estar ordenada de menor a mayor simplemente seleccionamos el primer elemento de la lista. Lo llamaremos Mcamino.
Si longitud(Mejor_camino) < longitud(Mcamino) **entonces**
Mejor_camino=Mcamino. Almacenamos una copia del mejor camino hasta el momento para el caso de que no encontremos ninguna unión y eliminemos todos los caminos.
FinSi
Eliminamos todos los caminos de LF cuya longitud sea inferior a la de Mcamino menos una constante.
Expandimos el camino actual. A partir del último segmento de Mcamino generamos Q caminos, de la misma forma que hemos generado los primeros caminos. Los insertaremos en LF siempre y cuando su recompensa geométrica y de intensidad en los últimos segmentos supere el umbral $\hat{T}$ y T, respectivamente.
FinMientras

Figura 3.9: Algoritmo de búsqueda de camino.

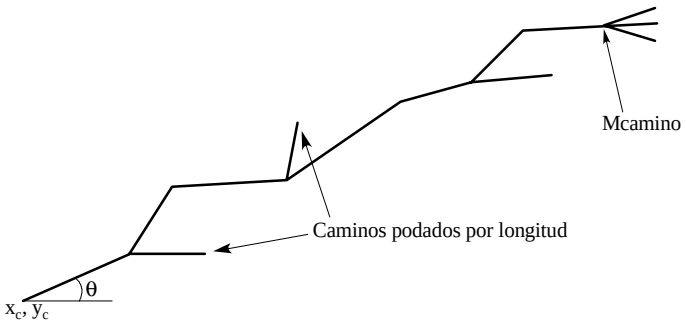

Figura 3.10: Generación de caminos.

Algoritmo de agrupamiento ALG_AGR

Entrada: $C = \{U_i : (x_c^i, y_c^i, \theta_1^i, \theta_2^i, \ldots, \theta_l^i)\}$ conjunto de uniones detectadas, con l no necesariamente el mismo para todo i. $\mathcal{I}_e$ imagen de gradiente.
Salida: Mapa planar detectado.

Para cada unión U de C **hacer**
  Para cada límite θ_i de las secciones angulares de U **hacer**
    Si θ_i ya ha sido visitado **entonces** continuar con el siguiente límite
    Sino
      Marcar θ_i como visitado
      Introducir las coordenadas del centro de la unión como un vértice del mapa.
      Si `Alg_ABayes` ($\mathcal{I}_e$, x_c, y_c, θ_i, `Camino, Union_encontrada`) **entonces**
        Incorporar `Camino` al mapa junto con las coordenadas del centro de `Union_encontrada`.
        Marcar como visitado aquel o aquellos límites de `Union_encontrada` cuya diferencia angular con respecto al ángulo de entrada del camino a la unión sea menor de θ_f.
      FinSi
    FinSi
  FinPara
FinPara

Figura 3.11: Algoritmo de agrupamiento.

Al igual que en el capítulo anterior, los experimentos se realizaron sobre secuencias de interior e imágenes de exterior. El tiempo medio de ejecución fue de 5 segundos, sin contar el tiempo de procesamiento de las uniones.

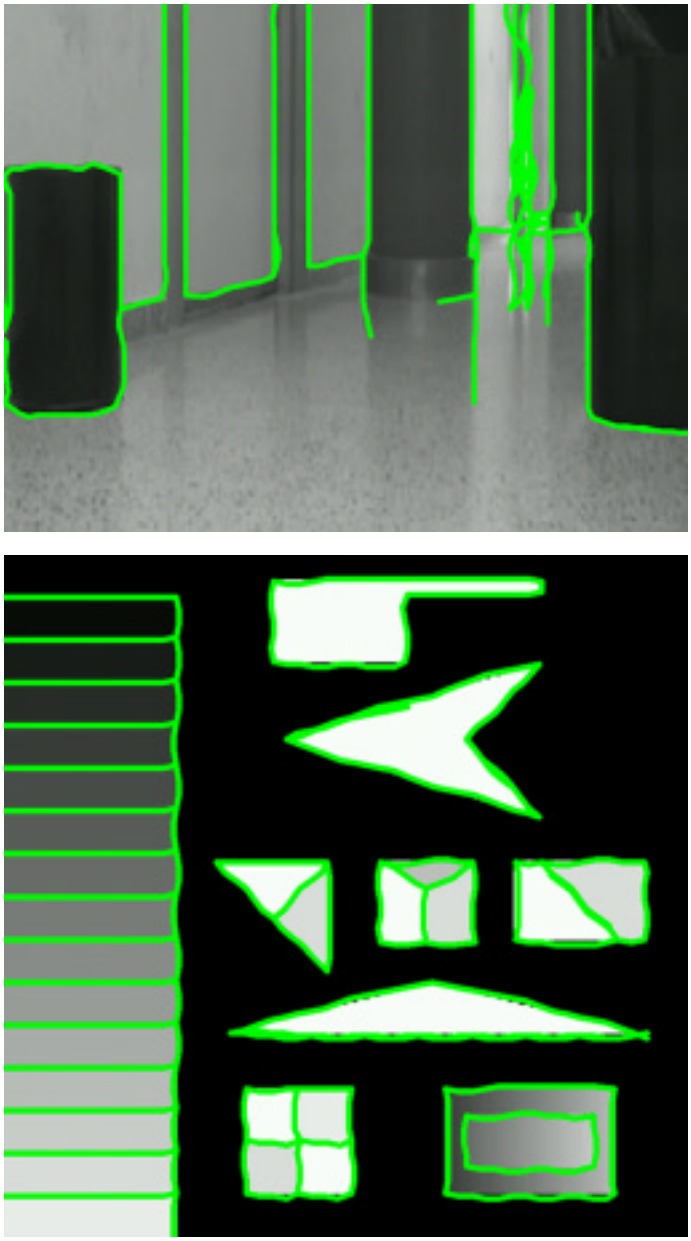

Figura 3.12: Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro no lineal).

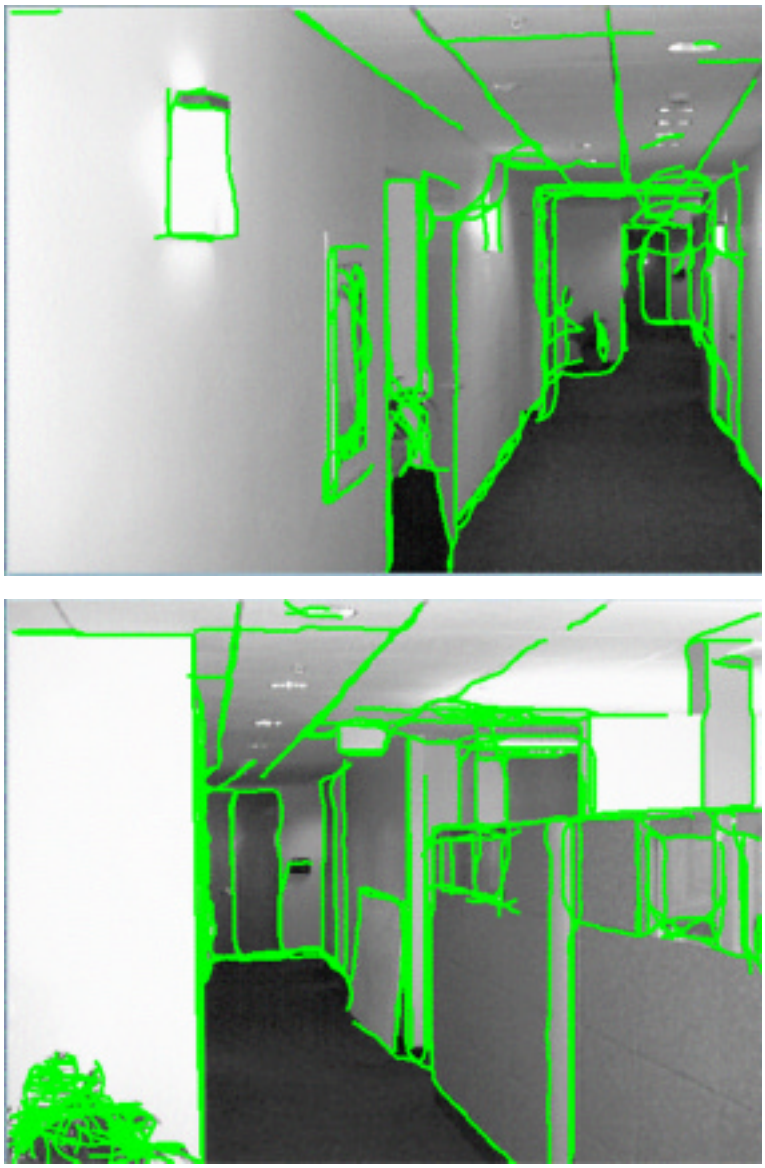

Figura 3.13: Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro de modelado de aristas).

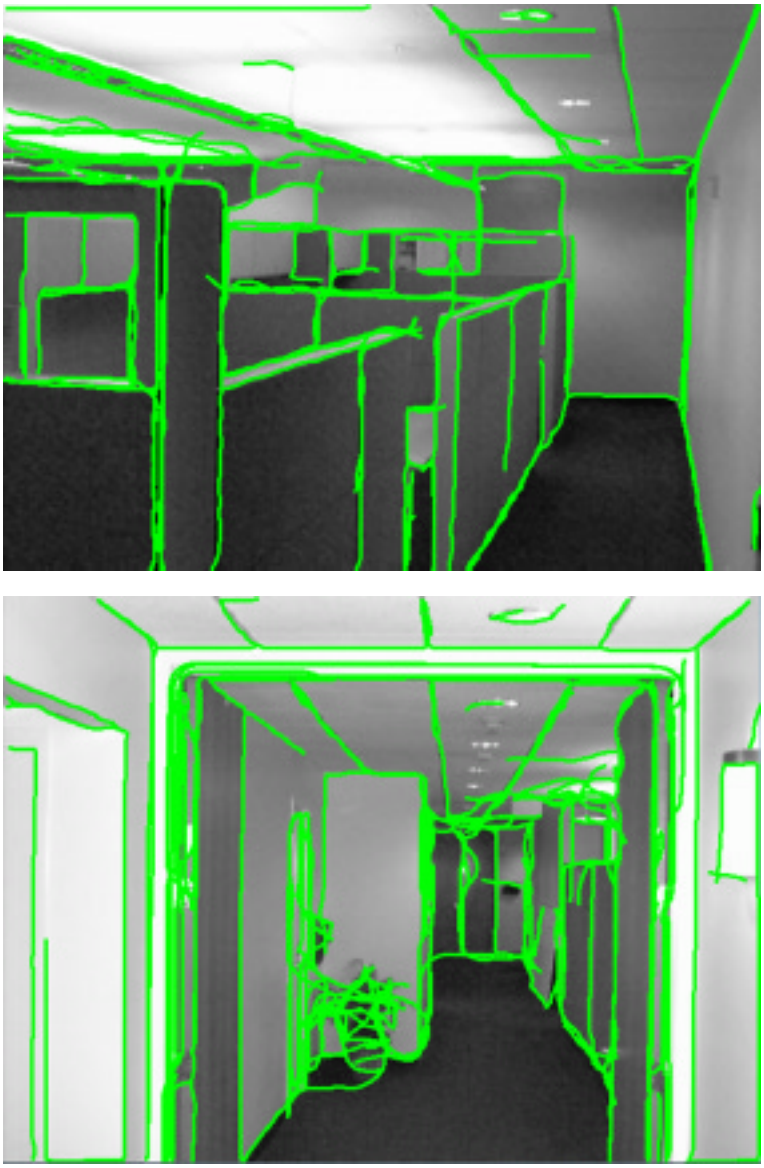

Figura 3.14: Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro de modelado de aristas).

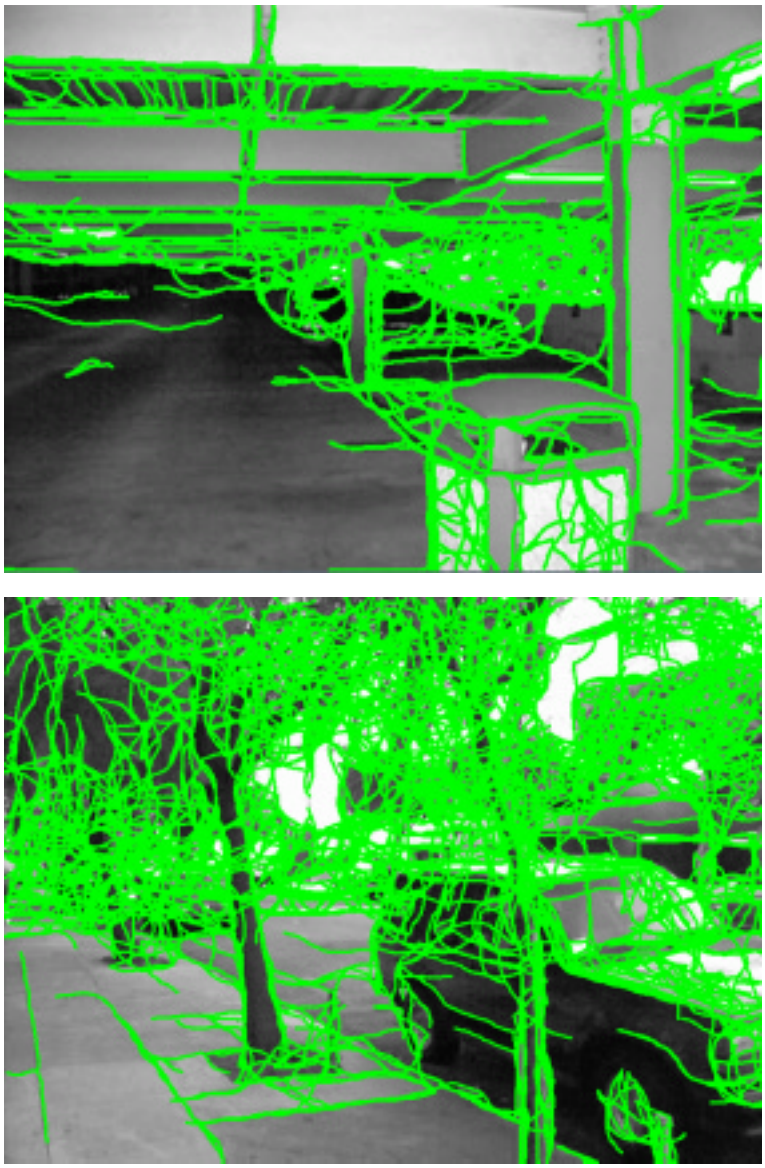

Figura 3.15: Ejemplos de aplicación del algoritmo de agrupamiento (componente de intensidad: filtro de modelado de aristas).

3.4.1 Conexión punto a punto

Comparamos ahora el método anterior con un método de búsqueda mediante fuerza bruta. Partimos de los puntos característicos obtenidos por cualquier detector (por ejemplo, SUSAN o Nitzberg). Este método consiste en ir visitando cada par de puntos característicos y comprobando si en el segmento definido por estos puntos existe evidencia de arista en la imagen. Para comprobar si existe evidencia vamos a utilizar la información estadística presentada en la Sección 2.5. Siendo r el segmento definido por dos puntos característicos y β el ángulo de la normal a ese segmento, nuestra función de energía viene dada por:

$$\mathcal{E}(r) = \int_r \log \frac{P_{on}(\vec{E}_{\vec{u}}|\beta)}{P_{off}(\vec{E}_{\vec{u}})} d\vec{u}$$

y en el caso discreto:

$$\mathcal{E}(r) = \sum_{\vec{u}\in r} \log \frac{P_{on}(\vec{E}_{\vec{u}}|\beta)}{P_{off}(\vec{E}_{\vec{u}})}$$

Esta función de energía nos permite cuantificar si existe evidencia de arista debajo del segmento definido por los puntos característicos. La determinación de si existe evidencia o no, viene dada por el criterio de corte detallado en la Sección 3.3 y reflejado en la Ecuación 3.6.

En la Figura 3.16 se muestra un resultado de aplicar este método. Como se puede observar existen segmentos que mantienen evidencia a pesar de no tener un soporte de arista en buena parte de su longitud. Esto es debido a que la parte que sí tiene soporte compensa la parte que no tiene. Esto provoca que se detecten demasiadas aristas entre puntos. Además, otro problema es que es poco flexible a pequeños cambios en la posición de los puntos característicos. Es decir, pequeños errores en la localización de los puntos esquina pueden provocar que perdamos la evidencia de la arista.

El tiempo medio de ejecución fue de 90 segundos. El tamaño de las imágenes utilizadas es de 640x480 y el número medio de puntos característicos detectados fue de 200.

Tanto el tiempo de ejecución como los problemas encontrados desaconsejan el uso de este método para el agrupamiento de uniones.

Figura 3.16: Aplicación del método de agrupamiento exhaustivo.

3.5 Variación de las componentes geométrica y de intensidad

Hemos realizado experimentos variando el umbral de corte de las componentes geométrica y de intensidad. La Figura 3.17 muestra la imagen utilizada para los experimentos. La Figura 3.18 muestra el resultado de variar la componente de intensidad. Para la imagen superior se utilizó un umbral de corte de 1 y para la inferior de 2. Como se puede observar cuando aumentamos el umbral de corte sólo sobreviven aquellos caminos donde la magnitud de gradiente es mayor. Al disminuir el umbral pasa lo contrario, permitiendo caminos con una magnitud de gradiente muy baja. El tiempo de procesamiento aumenta conforme disminuimos el umbral de corte y se debe a que se exploran muchos más caminos. La imagen

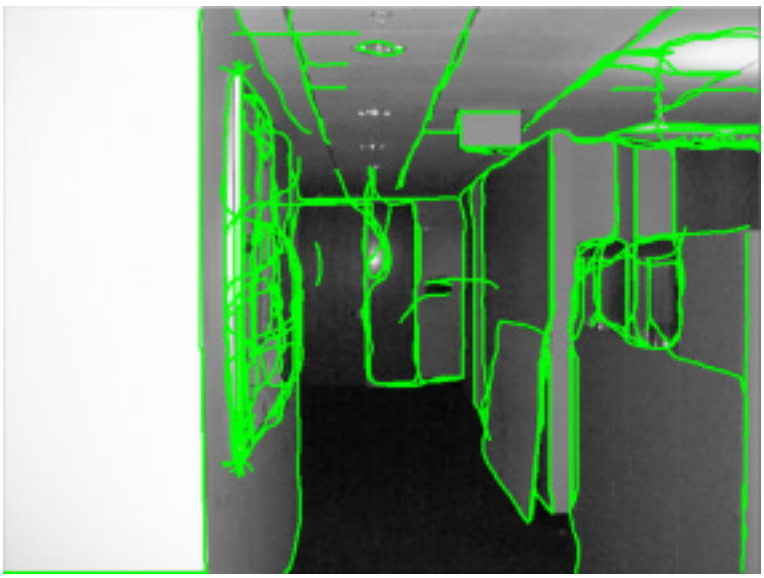

Figura 3.17: Imagen utilizada en los experimentos de variación de los umbrales de corte.

superior de la Figura 3.19 muestra el resultado de aplicar un umbral de -1. En la imagen inferior se muestra el resultado de variar la componente geométrica. En este caso hemos aplicado un umbral de -0.5. La Figura 3.20 muestra el resultado de aumentar el umbral de la componente geométrica a 0.5 (arriba) y 1 (abajo). Como se puede observar, conforme aumentamos la componente geométrica, sólo sobreviven aquellos caminos cuyo cambio de curvatura es poco significativo.

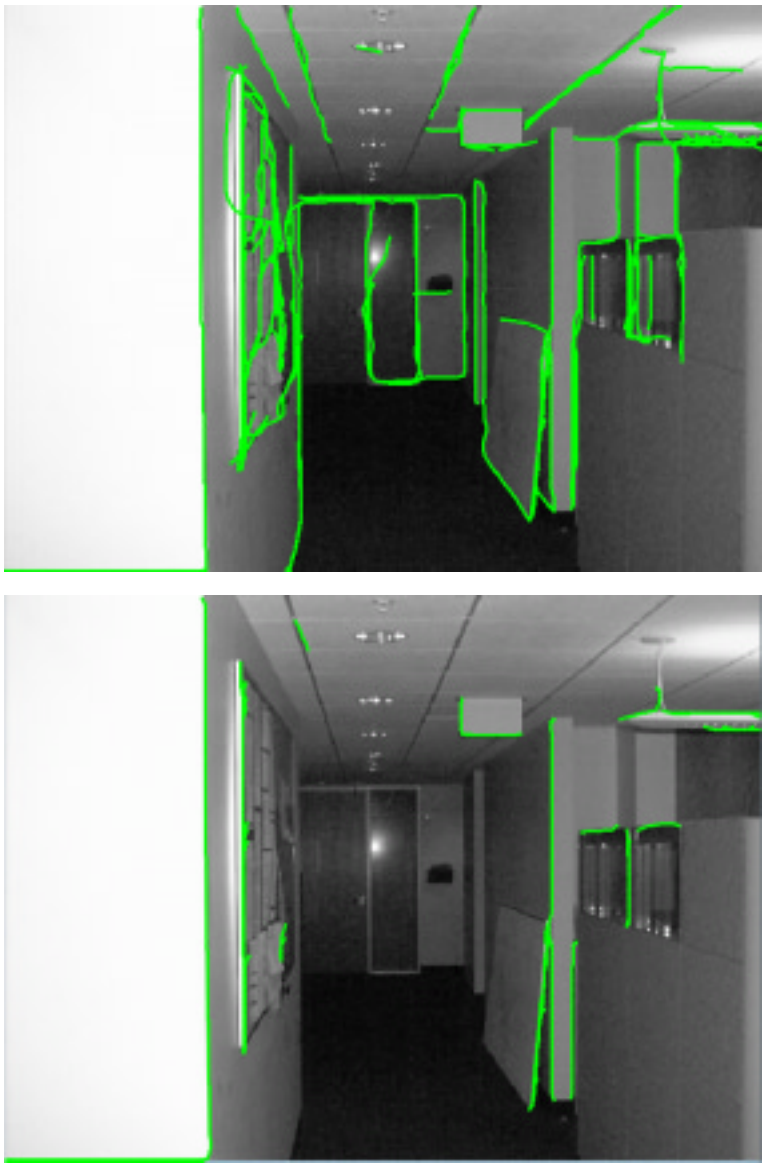

Figura 3.18: Variación del umbral de corte de la componente de intensidad.

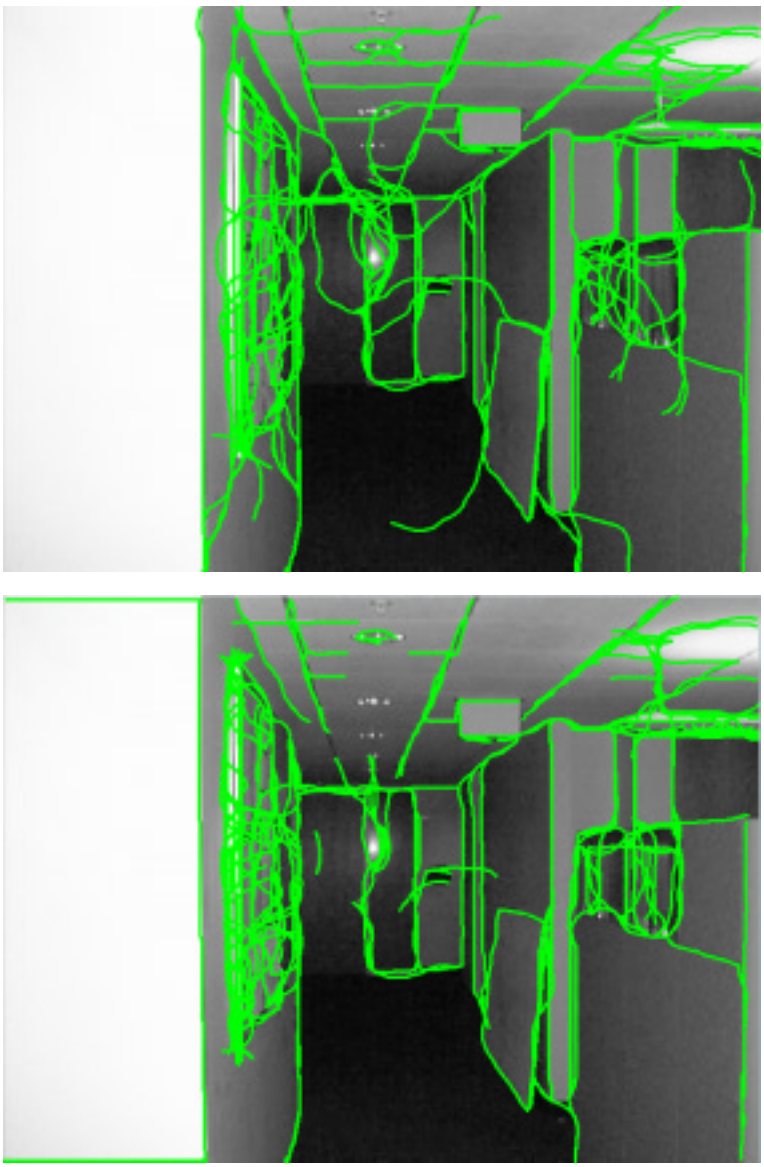

Figura 3.19: Variación del umbral de corte de la componente de intensidad y geométrica.

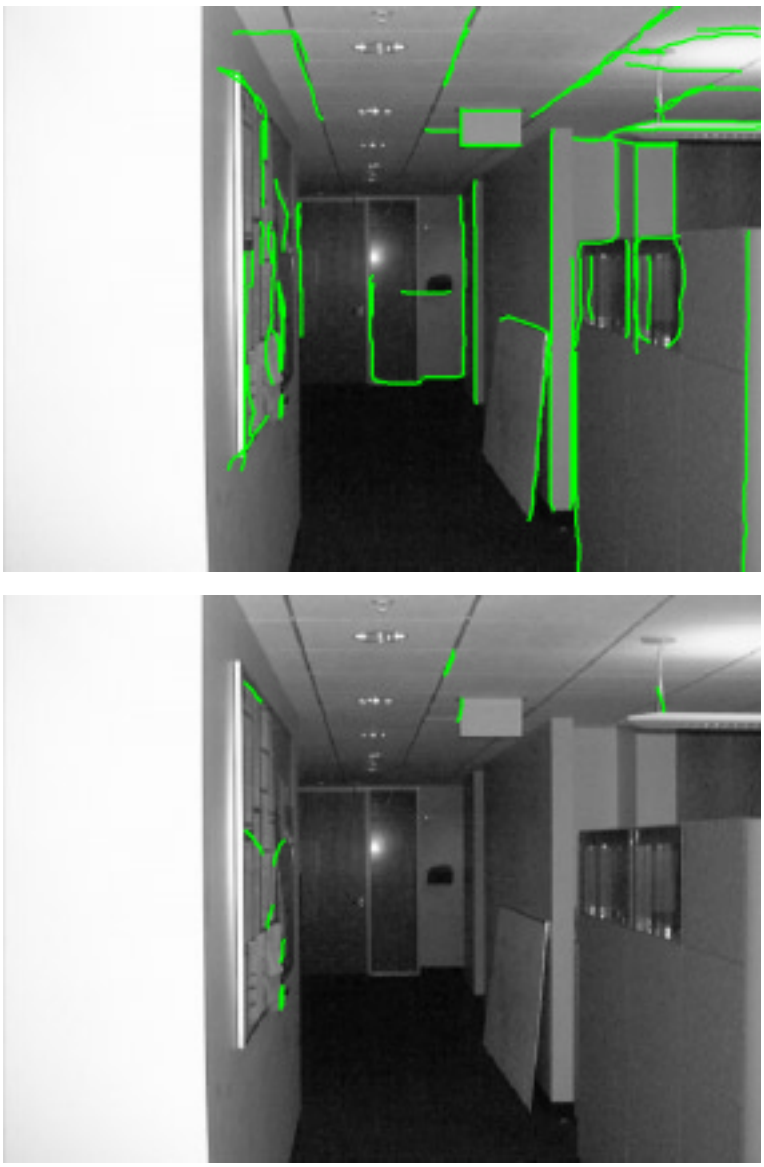

Figura 3.20: Variación del umbral de corte de la componente geométrica.

3.6 Clasificación de uniones

Como ya vimos en el capítulo anterior, los métodos de localización y clasificación de uniones son muy locales. Por ello, el error cometido es alto debido al ruido localizado en las proximidades de uniones y esquinas. El método de agrupamiento de uniones propuesto en este capítulo sirve para disminuir el efecto del ruido cometido en la clasificación de uniones. Con este método extendemos la unión en la dirección indicada por los límites de las secciones angulares. Al permitir una cierta flexibilidad en la dirección a seguir, estamos solventando posibles errores en la localización de dicho límite. También, si un camino no sobrevive en la primera fase de la búsqueda, desechamos ese camino. Lo que estamos haciendo es eliminar aquellos límites que, al extenderlos, no tengan suficiente evidencia de arista. En la Figura 3.21 mostramos varios ejemplos de cómo mejora el proceso de agrupamiento de uniones.

Con este proceso estamos mejorando los resultados obtenidos en la fase de detección y clasificación de uniones. Estamos produciendo una retroalimentación de un nivel superior a un nivel inferior para mejorar sus resultados.

3.7 Discusión

Este capítulo se ha planteado como una continuación de los métodos propuestos en el capítulo anterior. El objetivo a conseguir es la prolongación de las uniones detectadas para conectar uniones. La búsqueda de uniones se plantea como una búsqueda de camino, donde el camino lo marca el gradiente de la imagen. Se ha formulado dicha búsqueda en términos bayesianos, no buscando el mejor camino posible sino el camino verdadero en una población de caminos falsos. La búsqueda del camino se realiza con un algoritmo A* modificado, donde no se asegura la optimalidad del resultado, pero se mejora en cuanto al coste temporal. El coste de un camino vendrá dado por la suma de dos componentes, la componente de intensidad y la geométrica. La componente de intensidad nos da una medida de la evidencia de presencia de arista. La geométrica nos permite modelar la rigidez de los caminos a explorar. Dentro de la componente de intensidad presentamos dos modelos a seguir. En uno de ellos utilizamos un filtro no lineal basado en el número de votos que se realizan en un segmento dado. Para ello se han tomado estadísticas previas, determinando la respuesta del filtro para un determinado operador de gradiente (en nuestro caso hemos utilizado SUSAN) cuando el segmento se encuentra encima de una arista y cuando se encuentra fuera de ésta. En el otro modelo, el de modelado de aristas, las estadísticas se han obtenido a partir de otros autores y modelamos tanto la componente de magnitud del gradiente

como su orientación.

Una vez definidos los dos nuevos métodos, presentamos resultados obtenidos al aplicarlos sobre imágenes tanto de interior como de exterior. Igual que en los métodos de clasificación de uniones del capítulo anterior, el método propuesto presenta buenos resultados en imágenes de interior, teniendo una respuesta no tan buena en entornos no estructurados.

El método propuesto es robusto frente a posibles errores en la localización de los límites angulares. Esto es debido a que inicializamos la búsqueda con un margen lo suficientemente grande para que se subsanen estos posibles errores. También es robusto a la no localización de un límite angular, puesto que para un camino dado, tenemos dos posibles puntos de partida dadas dos uniones.

Tenemos dos umbrales, el geométrico y el de intensidad, que sirven para modelar la rigidez del camino y la evidencia de gradiente, respectivamente. Cuando disminuimos el umbral de gradiente, siempre dentro de los límites teóricos establecidos, permitimos caminos poco rígidos, mientras que si aumentamos el umbral sólo permitimos caminos con una variación de ángulo entre segmentos muy pequeña. En cuanto al umbral de intensidad, nos modela la cantidad de evidencia que debe tener debajo un bloque de segmentos para considerarlo válido. Si aumentamos este umbral, sólo nos quedaremos con aquellos caminos que estén encima de un gradiente con mucha magnitud. Al disminuir el umbral permitimos que con caminos con poca evidencia (magnitud de gradiente) sobrevivan a la poda.

Hemos comprobado la mejora que se produce al aplicar el método de agrupamiento frente a las uniones detectadas. Comprobamos que se produce una mejora notable en cuanto al número de límites de secciones angulares falsos que se eliminan. También mejora el error angular cometido en el cálculo de dichos límites. De esta forma estamos produciendo una retroalimentación de un proceso de un nivel a uno de un nivel inferior.

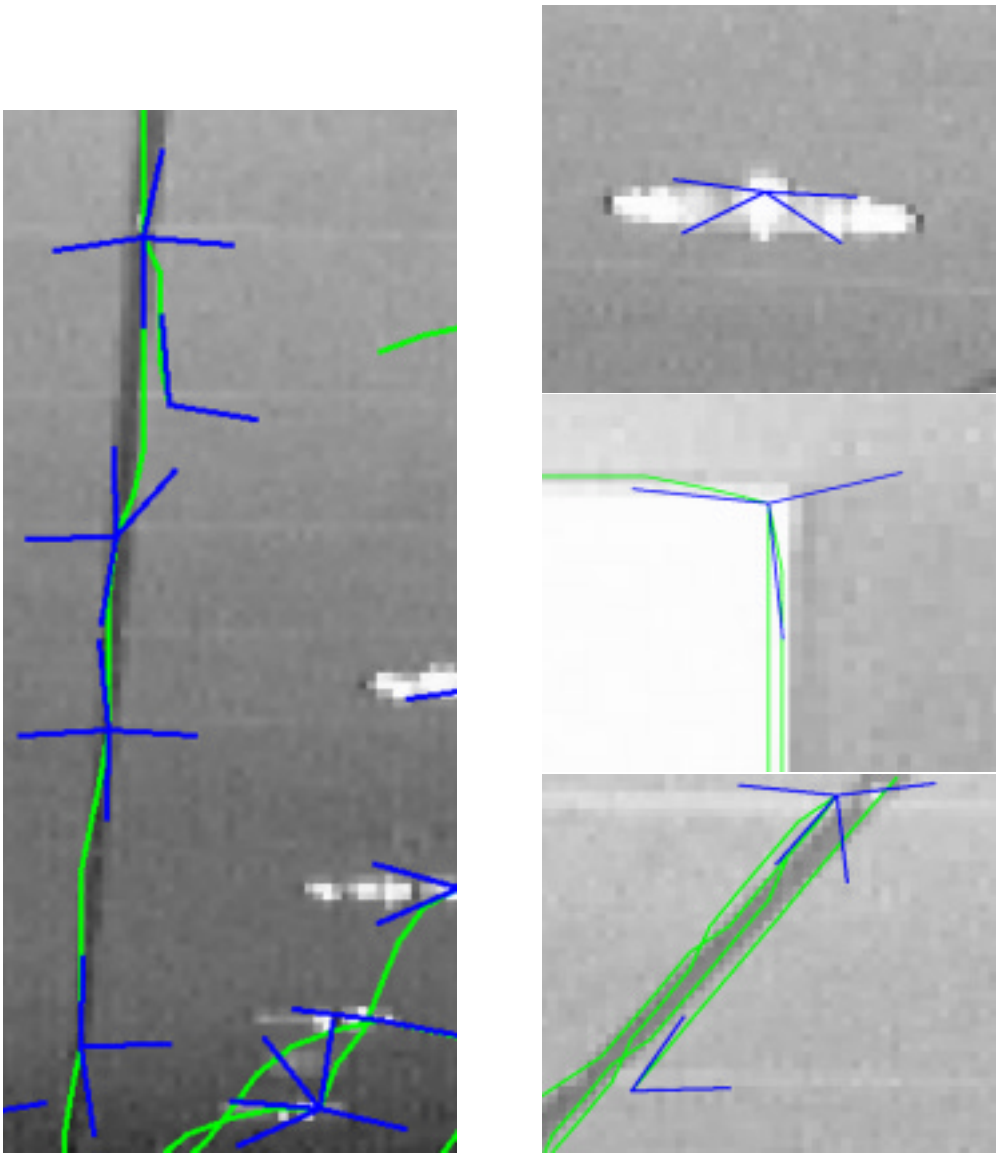

Figura 3.21: Mejora producida al introducir el agrupamiento de uniones. Las líneas azules son las uniones detectadas y las verdes son los caminos del agrupamiento. En la imagen de la izquierda podemos observar como algunas de las uniones tienen límites que no se corresponden con regiones reales. El proceso de agrupamiento elimina estos límites. En la imagen de la derecha (centro) una unión con forma de T termina clasificándose como una L. En la imagen de la derecha (arriba) la unión original es eliminada al no tener ningún límite soporte de arista.

Capítulo 4

Obtención del ángulo de orientación de la cámara

El cálculo del ángulo de orientación de la cámara es una tarea importante en el campo de la robótica móvil. Nos puede proporcionar información de orientación del robot cuando nos encontramos en una posición desconocida dentro de un entorno, conocido o no. En este capítulo vamos a presentar un nuevo método de obtención de dicho ángulo que se deriva del presentado en [Coughlan y Yuille, 1999]. Entre las ventajas de este método destacamos la utilización de una única imagen y su robustez ya que se basa en un esquema de inferencia bayesiana. Sin embargo, la eficiencia del método debe mejorarse para que éste sea aplicable a robótica móvil. Proponemos utilizar como base para la inferencia el agrupamiento de uniones descrito es capítulos anteriores ya que dicha estructura acelera la estimación. Para ello revisaremos los supuestos de aplicación del método bayesiano y lo situaremos en el contexto de las técnicas basadas en puntos de fuga.

El esquema bayesiano de obtención de la orientación relativa asume que el mundo se ajusta a la estructura del mundo de Manhattan. En el mundo de Manhattan tenemos un sistema de coordenadas en $\vec{i}, \vec{j}, \vec{k}$ (ver Figura 4.1). Si podemos determinar la posición relativa de la cámara en este sistema de coordenadas, podremos interpretar fácilmente la escena que estamos visualizando. Una característica propia de este tipo de mundo son los puntos de fuga (*vanishing points*).

Los puntos de fuga son aquellos puntos en el plano imagen donde convergen las proyecciones de las líneas paralelas del mundo. Son características invariantes a escala y rotación, por lo que pueden ser utilizadas para tareas de correspondencia entre imágenes [Moons *et al.*, 1996], calibración de la cámara o reconocimiento de objetos [Parodi y Piccioli, 1996, Shufelt, 1999a].

En un mundo estructurado tenemos tres puntos de fuga, uno por cada orientación posible de las aristas. Las aristas verticales tienen el punto de fuga en el infinito.

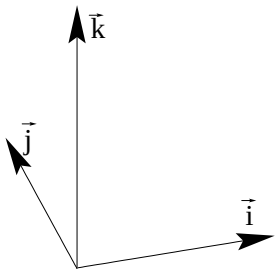

Figura 4.1: Sistema de coordenadas en el mundo de Manhattan.

En adelante vamos a suponer que nos encontramos en un entorno estructurado. Este tipo de suposición es perfectamente válida para la mayoría de entornos por los cuales un robot se va a mover: pasillos, habitaciones, naves industriales, etc. Sin embargo, también es perfectamente válido para entornos de exterior, siempre y cuando nos aparezcan estructuras construidas por el hombre tales como aceras, edificios, carreteras, etc.

Una última suposición es que la cámara sólo se puede orientar en el plano horizontal, es decir, gira alrededor del eje $\vec{k}$.

4.1 Técnicas de obtención de puntos de fuga

En esta sección vamos a presentar los primeros enfoques en la detección de puntos de fuga. Uno de los primeros trabajos en en este sentido es [Barnard, 1983]. En él se propone una metodología que ha establecido el paradigma para la detección de puntos de fuga más ampliamente extendido. Se supone conocida la distancia focal de la cámara, por lo que conocemos el centro de coordenadas. Se define entonces una esfera centrada en dicho centro, con radio unitario. Esta esfera se utiliza como espacio de orientaciones. Cada segmento de línea en la imagen tiene dos puntos, que junto con el centro de la esfera forman un plano, llamado plano

de interpretación. Cada plano intersecta con la esfera para formar un círculo (ver Figura 4.2). La esfera se utiliza como acumulador: zonas de la esfera donde inter-

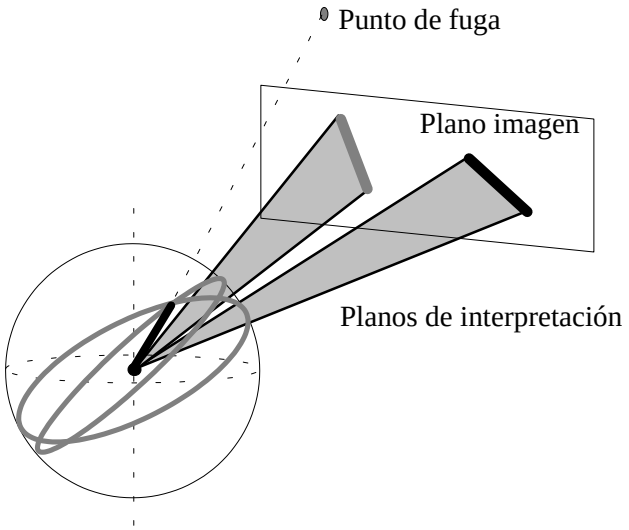

Figura 4.2: Esfera de acumulación para el cálculo del punto de fuga.

sectan círculos indican orientaciones comunes entre los segmentos que han generado los círculos. La idea de este método consiste en buscar zonas de la esfera más votadas, al estilo de la transformada de Hough. Este trabajo inicial fue extendido en [Magee y Aggarwal, 1984] y [Quan y Mohr, 1989], utilizando geometría esférica y transformada de Hough jerárquica, respectivamente. En general, la mayoría de los métodos propuestos asumen que los puntos iniciales y finales de los segmentos están bien calculados. Otros métodos propuestos basados en este enfoque son: [Brillault-O Mahoney, 1991], [McLean y Kotturi, 1995] y [Shufelt, 1999b]. Recientemente se ha propuesto un nuevo enfoque [Coughlan y Yuille, 1999] que no toma decisiones a priori sobre si un píxel pertenece o no a una arista, sino que se basa en un enfoque bayesiano. Además, explota las restricciones geométricas que se presentan en un mundo estructurado. Profundizamos este enfoque en este capítulo.

4.2 Geometría tridimensional

Vamos a definir y detallar algunos conceptos de geometría tridimensional que utilizaremos después en la definición del método. Definimos Ψ como el ángulo de orientación de la cámara dentro del mundo de Manhattan [Faugeras, 1993]: la cámara está orientada en la dirección $\cos\Psi\vec{i} - \sin\Psi\vec{j}$. Las coordenadas en el plano imagen $\vec{u} = (u, v)$ se relacionan con las coordenadas del mundo (x, y, z) mediante las ecuaciones:

$$u = \frac{f \cdot (-x\sin\Psi - y\cos\Psi)}{x\cos\Psi - y\sin\Psi}, \quad v = \frac{f \cdot z}{x\cos\Psi - y\sin\Psi} \tag{4.1}$$

donde f es la longitud focal de la cámara.

Los puntos de fuga de las líneas en las direcciones $\vec{i}$ y $\vec{j}$ se encuentran en los puntos $(-f\tan\Psi, 0)$ y $(f\cot\Psi, 0)$, respectivamente, en el plano imagen. Las líneas en la dirección $\vec{k}$ son todas verticales debido a la suposición de encontrarnos en una mundo estructurado.

Por último, vamos a relacionar el gradiente de intensidad obtenido en un punto de la imagen con el ángulo de orientación. Un punto de la imagen $\vec{u} = (u, v)$ con gradiente de intensidad $(\cos\theta, \sin\theta)$ es consistente con una línea $\vec{i}$ (es decir, apunta al punto de fuga), si $-v\tan\theta = u + f\tan\Psi$. Esta ecuación no cambia si añadimos $\pm\pi$ a θ, es decir, la polaridad de la arista no afecta a los cálculos. De forma similar tenemos $v\tan\theta = -u + f\cot\Psi$ para las líneas $\vec{j}$.

4.3 Modelo bayesiano inicial

Vamos a presentar el modelo bayesiano propuesto en [Coughlan y Yuille, 1999]. La diferencia de este modelo con respecto a otros enfoques propuestos con anterioridad es que no nos fuerza a decidir prematuramente qué píxeles están o no en una arista. Además, permitirá etiquetar posteriormente un píxel como perteneciente a una arista de los tres tipos que nos podemos encontrar en el mundo de Manhattan: $\vec{i}, \vec{j}, \vec{k}$.

4.3.1 Estadísticas de las aristas

En este trabajo no se utiliza un mapa de aristas binario, sino un modelo estadístico que aporta más información a la hora de etiquetar un píxel de la imagen como perteneciente a una arista. Para ello, se utiliza el trabajo desarrollado en [Konishi *et al.*, 1999]. El método y las estadísticas obtenidas se detallan en la Sección 2.5.

4.3.2 Evidencia de un píxel

Tenemos el valor del gradiente $\vec{E}_{\vec{u}}$ (magnitud y orientación) en un punto de la imagen $\vec{u}$. Este valor se explica mediante uno de los cinco modelos siguientes $m_{\vec{u}}$: $m_{\vec{u}} = 1, 2, 3$ para cada una de las posibles aristas generadas por líneas en las direcciones $\vec{i}$, $\vec{j}$ y $\vec{k}$, respectivamente, en la escena; $m_{\vec{u}} = 4$ significa que el gradiente ha sido generado por una arista aleatoria (no debida a una arista $\vec{i}, \vec{j}, \vec{k}$); y, por último, $m_{\vec{u}} = 5$ significando que el píxel no está en una arista. Las probabilidades a priori $P(m_{\vec{u}})$ de cada modelo las hemos estimado experimentalmente.

Se asume que la probabilidad del gradiente de la imagen $\vec{E}_{\vec{u}}$ tiene dos factores correspondientes a la componente de magnitud ($E_{\vec{u}}$) y orientación del gradiente ($\sigma_{\vec{u}}$):

$$P(\vec{E}_{\vec{u}}|m_{\vec{u}}, \Psi, \vec{u}) = P(E_{\vec{u}}|m_{\vec{u}})P(\phi_{\vec{u}}|m_{\vec{u}}, \Psi, \vec{u}) \tag{4.2}$$

donde $P(E_{\vec{u}}|m_{\vec{u}})$ es igual a $P_{off}(E_{\vec{u}})$ si $m_{\vec{u}} = 5$ o $P_{on}(E_{\vec{u}})$ si $m_{\vec{u}} \neq 5$. Por otro lado, $P(\phi_{\vec{u}}|m_{\vec{u}}, \Psi, \vec{u})$ es igual a $P_{ang}(\phi_{\vec{u}} - \theta(\Psi, m_{\vec{u}}, \vec{u}))$ si $m_{\vec{u}} = 1, 2, 3$ o $U(\phi_{\vec{u}})$ si $m_{\vec{u}} = 4, 5$. La orientación $\theta(\Psi, m_{\vec{u}}, \vec{u})$ es la normal determinada por la ecuación $-v \tan\theta = u + f \tan\Psi$ para líneas $\vec{i}$, $v \tan\theta = -u + f \cot\Psi$ para líneas $\vec{j}$, y $\theta = 0$ para líneas $\vec{k}$. Ψ es el ángulo de orientación determinado de la cámara.

En la propuesta inicial, en vez de determinar el modelo concreto al que pertenece un píxel, se acumula la evidencia para los cinco posibles modelos:

$$P(\vec{E}_{\vec{u}}|\Psi, \vec{u}) = \sum_{m_{\vec{u}}=1}^{5} P(\vec{E}_{\vec{u}}|m_{\vec{u}}, \Psi, \vec{u})P(m_{\vec{u}}) \tag{4.3}$$

De esta forma vamos a ser capaces de determinar la evidencia del ángulo de la cámara Ψ sin conocer a cuál de los cinco modelos pertenece el píxel.

4.3.3 Evidencia de todos los píxeles

Para determinar la evidencia sobre todos los píxeles de la imagen, denotados por $\{\vec{E}_{\vec{u}}\}$, se asume que los datos de la imagen son condicionalmente independientes en todos los píxeles, dada la dirección de orientación Ψ:

$$P(\{\vec{E}_{\vec{u}}\}|\Psi) = \prod_{\vec{u}} P(\vec{E}_{\vec{u}}|\Psi, \vec{u}) \tag{4.4}$$

Así, la distribución a posteriori sobre la dirección de orientación viene dada por el producto: $\prod_{\vec{u}} P(\vec{E}_{\vec{u}}|\Psi, \vec{u})P(\Psi)/Z$, donde Z es un factor de normalización y $P(\Psi)$ es la distribución a priori uniforme sobre el ángulo de orientación. Para

encontrar el MAP necesitamos maximizar el log del anterior término, ignorando Z que es independiente frente a Ψ:

$$\log[P(\{\vec{E}_{\vec{u}}\}|\Psi)P(\Psi)] = \log P(\Psi) + \sum_{\vec{u}} \log[\max_{m_{\vec{u}}} P(\vec{E}_{\vec{u}}|m_{\vec{u}}, \Psi, \vec{u})P(m_{\vec{u}})] \qquad (4.5)$$

El algoritmo propuesto en un principio evalúa numéricamente el log a posteriori para cada dirección de orientación Ψ en el rango -45^o a $+45^o$, tomada en incrementos de un grado (ver Figura 4.3). En la Figura 4.4 se muestra un resultado de evaluación de esta función. En ellas, podemos ver la existencia de un único máximo.

Algoritmo de obtención del ángulo de orientación de la cámara ALG_CA

Entrada: Imagen de entrada
Salida: Ángulo de orientación detectado.

Procesar la imagen para obtener el gradiente (magnitud y orientación).
Calcular el P_{on} y el P_{off} de cada píxel de la imagen.
Para cada punto de la imagen **hacer**
  Para cada ángulo posible ($[-45, 45]$) **hacer**
    Calcular la evidencia del píxel con el ángulo actual mediante la Ecuación 4.6.
    Acumular la evidencia para el ángulo en concreto.
  FinPara
FinPara
Devolver el ángulo con valor máximo de evidencia.

Figura 4.3: Algoritmo de obtención del ángulo de orientación de la cámara.

4.3.4 Resultados

Se ha aplicado el método de [Coughlan y Yuille, 1999] a varias imágenes. Las Figuras 4.5, 4.6, 4.7 y 4.8 muestran la aplicación del método en escenas de exterior; las Figuras 4.9 y 4.10 hacen lo propio con escenas de interior. Para cada figura se ha mostrado dos imágenes: la de la izquierda muestra hacia donde apuntan los puntos de fuga de las líneas $\vec{i}$ y $\vec{j}$ (las líneas $\vec{k}$ apuntan verticalmente al infinito); la de la derecha muestra el etiquetado final de los píxeles: rojo, modelo 1, verde, modelo 2, azul, modelo 3, negro, modelo 4, y blanco, modelo 5.

Las probabilidades a priori para cada modelo $P(m_{\vec{u}})$, se han determinado experimentalmente. Para estos experimentos hemos utilizado 0.2, 0.2, 0.2, 0.4, 0.9 para $m_{\vec{u}} = 1, 2, 3, 4, 5$, respectivamente. Sin embargo, cuando queremos detectar

los objetos no alineados con el mundo de Manhattan (ver Figuras 4.13 y 4.14), hemos aumentado la probabilidad del modelo 4 a 6.6. Para la función P_{ang} hemos sintonizado los siguientes valores: $\epsilon = 0.1$ y $\tau = 4^{o}$.

Las Figuras 4.5, 4.6, 4.7 y 4.8 muestran resultados del algoritmo con imágenes tomadas en el exterior. Las Figuras 4.9, 4.10, 4.11 y 4.12 corresponden a imágenes de interior. Podemos observar que no se ha tenido un cuidado especial en mantener la horizontalidad de la cámara para la obtención de estas imágenes. Sin embargo, el algoritmo responde bastante bien a pequeños cambios de orientación con respecto a la horizontal. El tiempo medio de ejecución fue de 90 segundos en un Pentium II a 266 Mhz trabajando con imágenes de 640x480.

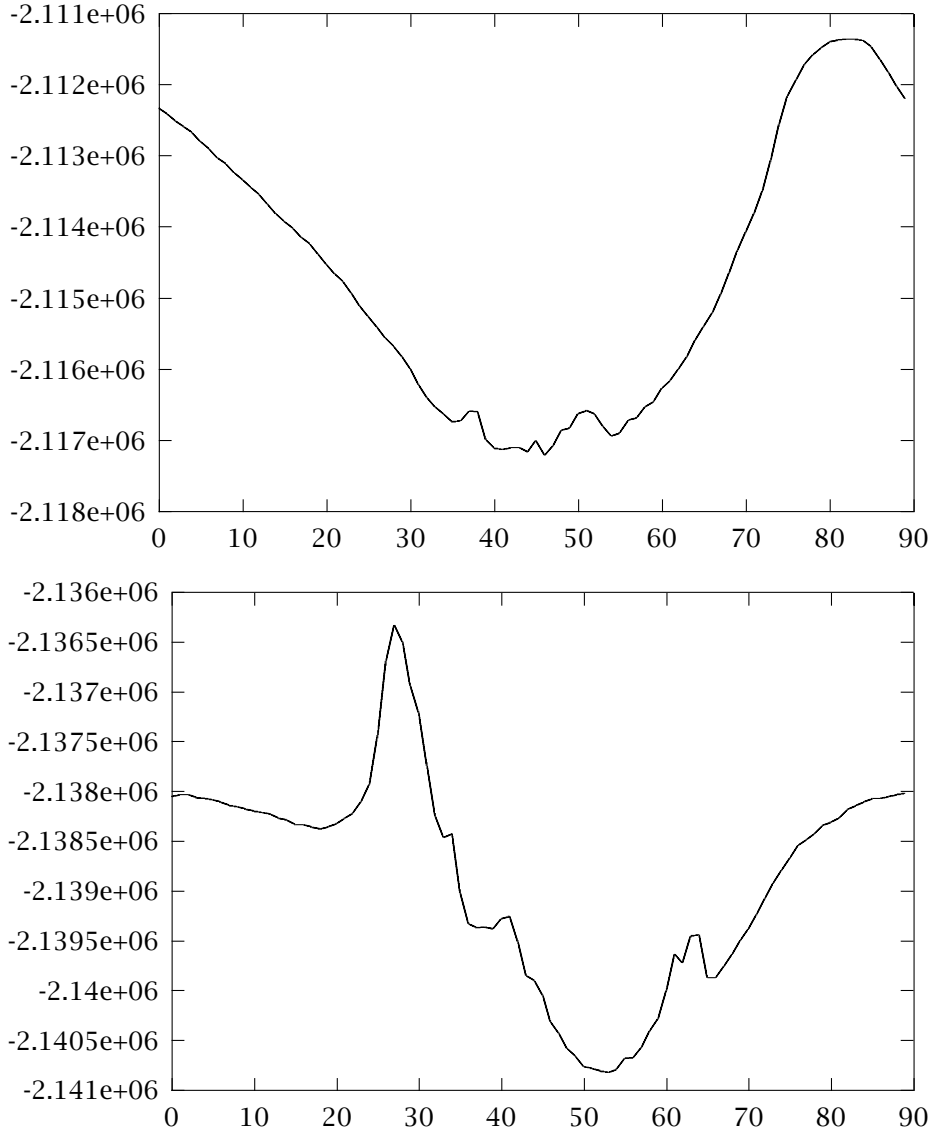

Figura 4.4: Gráficas del MAP frente al ángulo de orientación. Estas gráficas se corresponden con las Figuras 4.5 y 4.9.

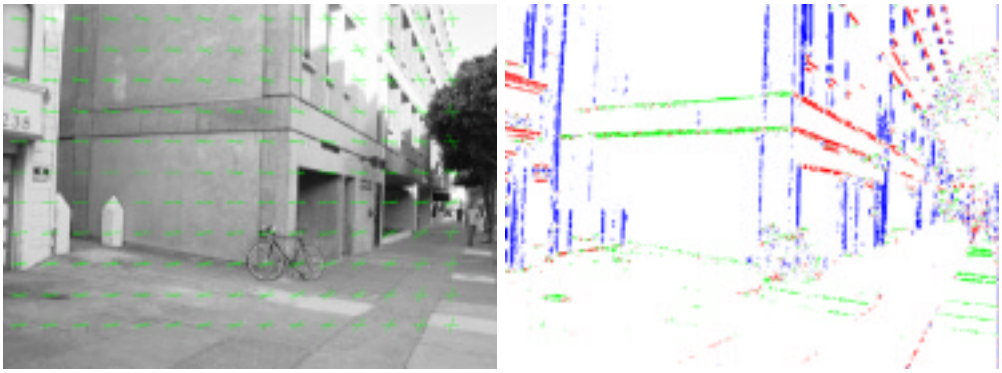

Figura 4.5: Resultado obtenido en imagen de exterior (1). El ángulo de orientación de la cámara obtenido fue de -18º.

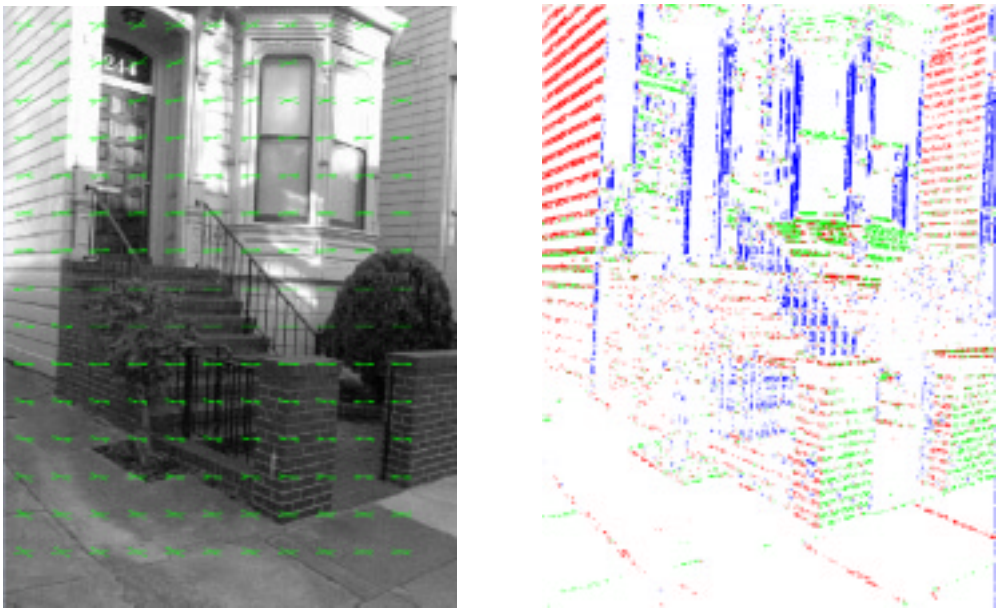

Figura 4.6: Resultado obtenido en imagen de exterior (2). El ángulo de orientación de la cámara obtenido fue de 32º.

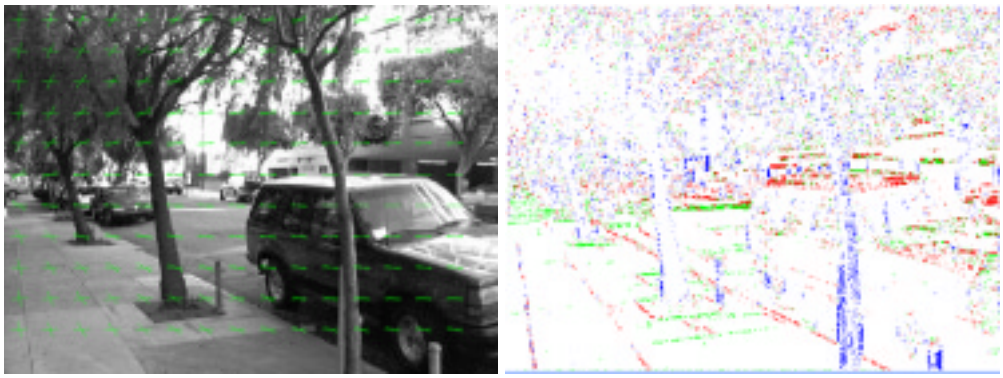

Figura 4.7: Resultado obtenido en imagen de exterior (3). El ángulo de orientación de la cámara obtenido fue de 20º.

Figura 4.8: Resultado obtenido en imagen de exterior (4). El ángulo de orientación de la cámara obtenido fue de 8º.

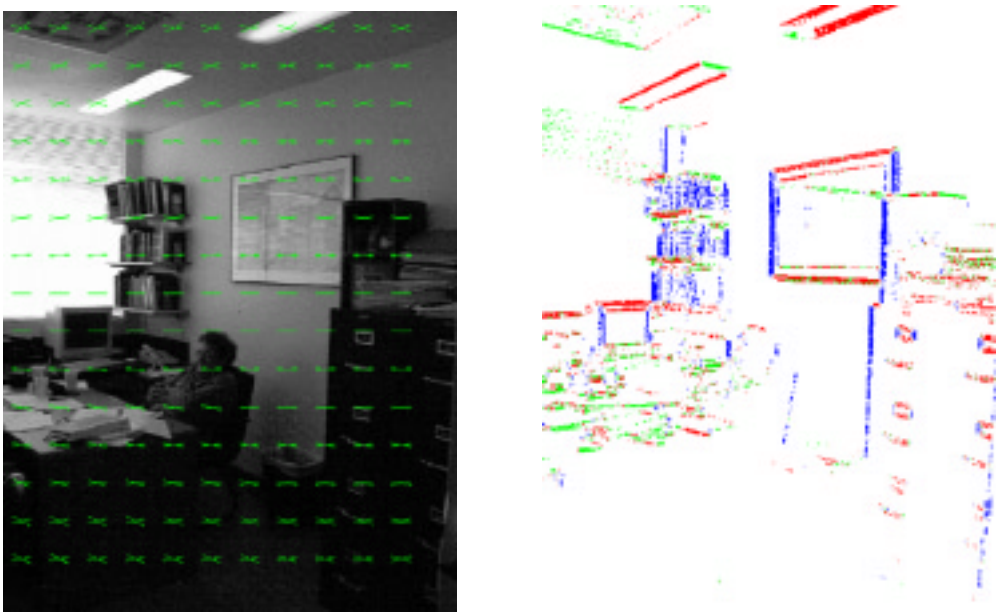

Figura 4.9: Resultado obtenido en imagen de interior (1). El ángulo de orientación de la cámara obtenido fue de 37º.

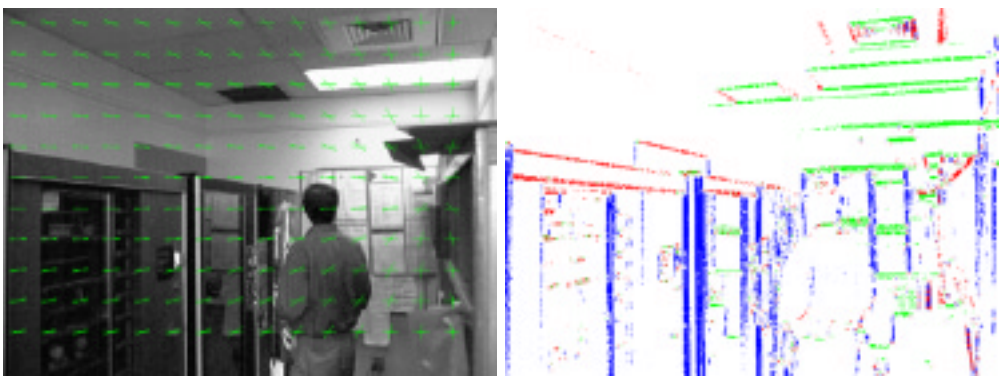

Figura 4.10: Resultado obtenido en imagen de interior (2). El ángulo de orientación de la cámara obtenido fue de -15º.

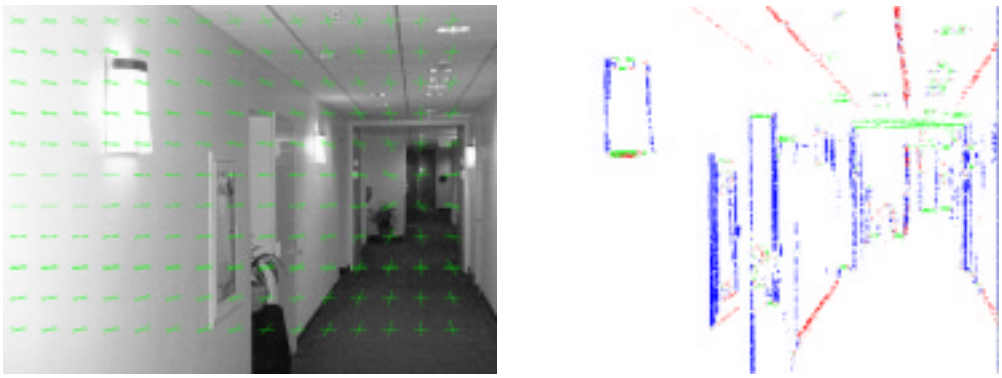

Figura 4.11: Resultado obtenido en imagen de interior (3). El ángulo de orientación de la cámara obtenido fue de -13º.

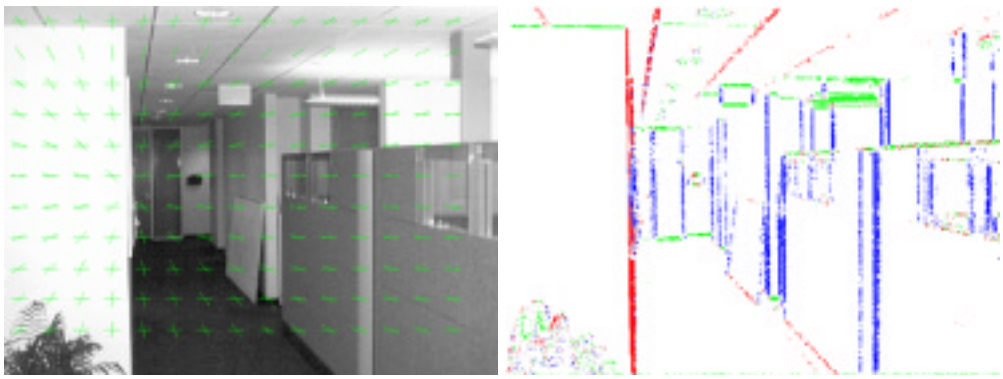

Figura 4.12: Resultado obtenido en imagen de interior (4). El ángulo de orientación de la cámara obtenido fue de 10º.

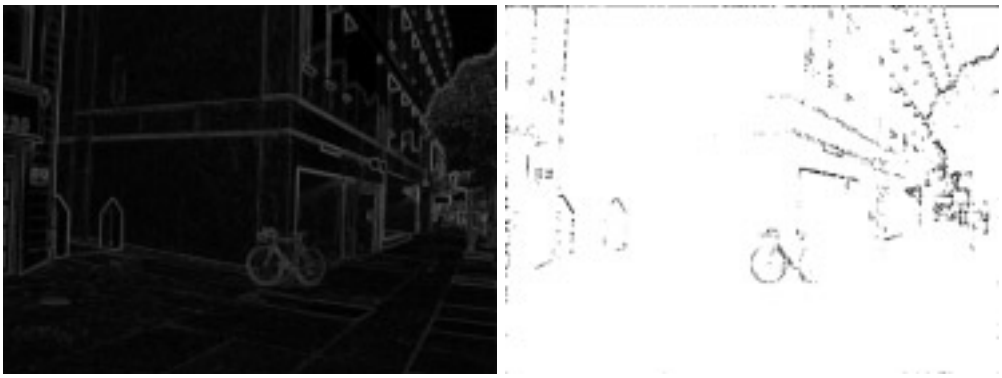

Figura 4.13: Puntos etiquetados con modelo 4 (derecha) y magnitud del gradiente (izquierda).

4.3.5 Etiquetado de los píxeles

Una aplicación inmediata de este método es la detección de objetos cuya forma no esté alineada con el mundo de Manhattan, es decir, la dirección de sus aristas no apuntan a los puntos de fuga. Estos puntos se corresponderán con el modelo 4. Para detectar estos puntos, una vez calculado el ángulo de orientación, podemos etiquetar cada píxel de la imagen estimando el modelo mediante el MAP de $P(m_{\vec{u}}|\vec{E}_{\vec{u}}, \Psi^*, \vec{u})$ para estimar el $m_{\vec{u}}$ en cada píxel $\vec{u}$. Para realizar este etiquetado se ha aumentado la probabilidad a priori del modelo 4 a 6.6. En las Figuras 4.13 y 4.14 se muestran resultados obtenidos en el etiquetado de estos píxeles.

4.3.6 Conclusiones del método

Este método nos permite calcular la orientación de la cámara a partir de la evidencia de los datos de la imagen. No es necesario tener un conocimiento específico del entorno en el que nos movemos, ni tomar decisiones precipitadas sobre la pertenencia de un píxel a un determinado tipo de línea. También nos permite detectar aquellos píxeles cuya forma no sea estructurada. De esta forma podremos aislar aquellos objetos, tales como personas, que se encuentren en nuestro entorno y que son obstáculos potenciales.

En cuanto a las limitaciones del método, como todo modelo bayesiano se ve afectado mucho por las funciones de distribución obtenidas. Para un entorno determinado, se puede hacer necesario tomar nuevas estadísticas de la respuesta de

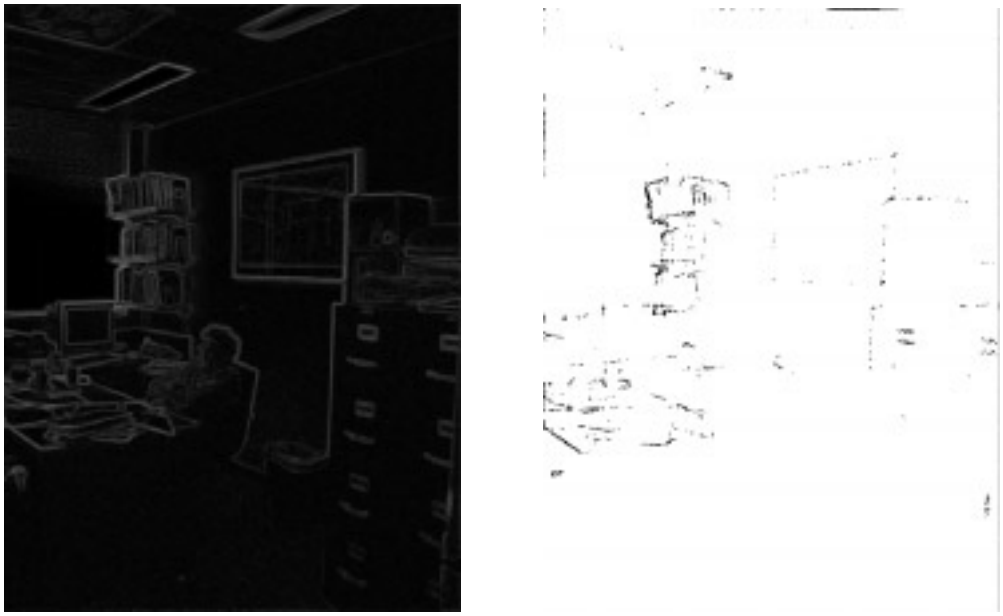

Figura 4.14: Puntos etiquetados con modelo 4 (derecha) y magnitud del gradiente (izquierda).

de un determinado filtro de aristas, así como ajustar las probabilidades a priori de cada modelo. Una vez ajustados estos datos, es posible que los datos de la imagen no nos proporcionen suficiente evidencia como para determinar con exactitud el ángulo de orientación de la cámara. Esto puede ser debido a que las aristas presentes en la imagen no nos proporcionen una correcta información. Por ejemplo, la imagen Figura 4.15 nos muestra un etiquetado erróneo debido a la no utilización del flash en la obtención de la imagen. Por ello, debemos ser especialmente cuidadosos en la obtención de las probabilidades que modelen nuestro entorno. Otra limitación es el tiempo de ejecución. El tiempo medio fue de 90 segundos. De esta forma, se hace muy difícil pensar en la implementación de este algoritmo en una plataforma móvil y trabajar en tiempo real.

En cuanto a posibles ampliaciones del método, una inmediata es incorporar las posibles orientaciones de la cámara en los dos ejes restantes. Esto no parece una tarea fácil a priori, pero permitiría una localización exacta de la orientación de la cámara. También, una posible ampliación es la utilización de mejores filtros de gradiente. En el trabajo antes mencionado [Konishi *et al.*, 1999] presentan una

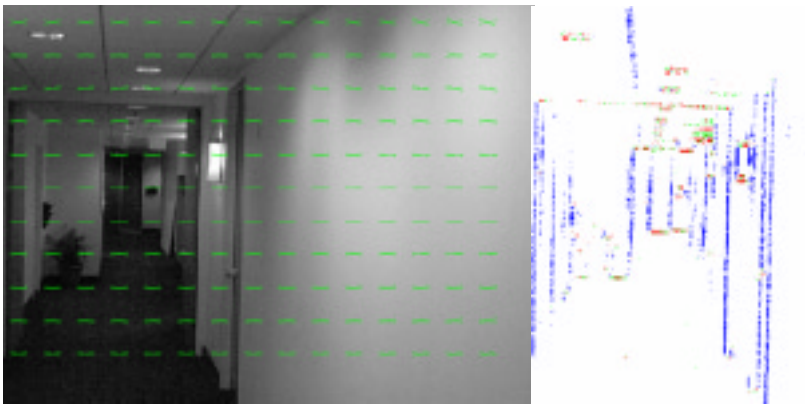

Figura 4.15: Resultado erróneo debido a la mala calidad de la imagen obtenida.

medida de la efectividad de los filtros de obtención de aristas. La utilización de filtros de color a diferentes escalas proporcionaría mejores resultados en cuanto a la localización y dirección exacta de las aristas. Sin embargo, el color es una característica muy dependiente del entorno en el que nos movemos. Los colores de un prado al aire libre no son los mismos que en el interior de un edificio de oficinas.

4.4 Obtención del ángulo de orientación de la cámara utilizando la información de arista

Una crítica inmediata al algoritmo de obtención del ángulo de orientación es el tiempo de ejecución elevado. Vamos a plantear un nuevo enfoque, pero haciendo uso de la formulación bayesiana planteada al principio del capítulo. Para ello utilizaremos el resultado de aplicar el método de agrupamiento de uniones desarrollado en el Capítulo 3.

4.4.1 Obtención de caminos rectos

La información obtenida como resultado de aplicar el algoritmo de agrupamiento, propuesto en el capítulo anterior, es un conjunto de vértices y caminos entre estos vértices. Para disminuir el tiempo de ejecución hemos convertido los caminos en segmentos rectos. Hemos utilizado el algoritmo propuesto en [Lowe, 1987] que se

detalla en el Apéndice C. Por lo tanto, después de aplicar dicho algoritmo tenemos un conjunto de segmentos rectos: C. Este conjunto será la entrada para el cálculo del ángulo de orientación.

4.4.2 Estadística de las aristas

La formulación bayesiana es la misma que la definida al principio de este capítulo. Lo que hemos variado son las funciones $P_{on}(E_{\vec{u}})$ y $P_{off}(E_{\vec{u}})$. Estas funciones tomarán los siguientes valores:

$$P_{on}(E_{\vec{u}}) = \begin{cases} 0.99 & \text{si} \quad E_{\vec{u}} \in C \\ 0.01 & \text{en otro caso} \end{cases}$$

$$P_{off}(E_{\vec{u}}) = \begin{cases} 0.99 & \text{si} \quad E_{\vec{u}} \notin C \\ 0.01 & \text{en otro caso} \end{cases}$$

También el ángulo de orientación en cada píxel ($\sigma_{\vec{u}}$) cambia, puesto que ahora no se calcula del gradiente, sino que cada píxel tiene asociado el ángulo de la normal del segmento correspondiente, caso de que el píxel pertenezca a una arista.

En la práctica, y en aras de mejorar la eficiencia, computamos sólo una vez por cada segmento el ángulo de la normal a dicho segmento y acumulamos el correspondiente valor multiplicando por su longitud, ya que dicha longitud da una idea de la confianza en la arista.

En la experimentación previa del algoritmo hemos detectado ciertos errores. En la Figura 4.16 se muestra un cálculo erróneo del ángulo de orientación, debido a que cerca del punto de fuga existen líneas verticales que pertenecen al modelo 1 y al modelo 3 a la vez. Por ello acumulan dos veces la misma evidencia debido a la suma de los valores que se produce en la Ecuación 4.3. Para solucionar este problema proponemos seleccionar el máximo valor de los cinco modelos, en vez de la suma. La Ecuación 4.3 quedaría:

$$P(\vec{E}_{\vec{u}}|\Psi,\vec{u}) = \max_{m_{\vec{u}}}\{P(\vec{E}_{\vec{u}}|m_{\vec{u}},\Psi,\vec{u})P(m_{\vec{u}})\} \tag{4.6}$$

La mejora al aplicar esta nueva ecuación se muestra en la Figura 4.17, en la cual el ángulo de orientación es calculado correctamente. En la Figura 4.18 se muestra el log de la probabilidad a posteriori. Cuando utilizamos la suma podemos observar un pico en -15 grados que es el máximo de la función. Esto es debido a la acumulación de evidencia por parte de dos modelos con ese ángulo: modelo 1 (rectas $\vec{i}$) y modelo 3 (rectas $\vec{k}$). Cuando utilizamos el máximo el pico se atenúa y podemos seleccionar el valor correcto.

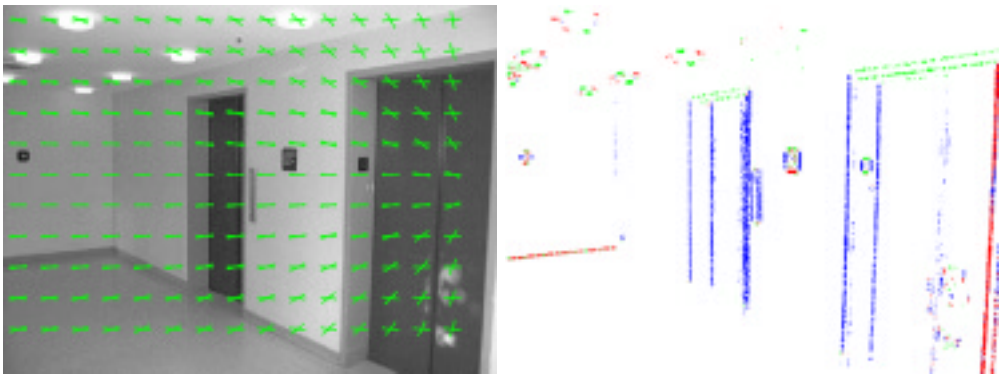

Figura 4.16: Resultado erróneo ($\Psi = -15º$) utilizando la suma de la evidencia para cada modelo.

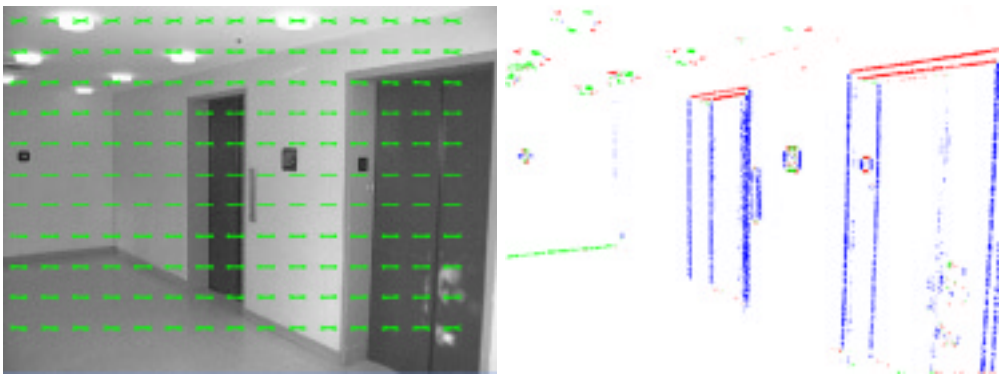

Figura 4.17: Resultado correcto ($\Psi = 40º$) utilizando el máximo valor de evidencia.

4.4.3 Cálculo del ángulo de orientación

Una vez comentadas las modificaciones efectuadas al algoritmo original, nuestra propuesta completa se muestra en la Figura 4.19.

4.4.4 Resultados

Se ha aplicado este método al mismo conjunto de imágenes que en el método anterior. Una vez aplicado el método, el error medio cometido con respecto al método anterior fue de 0.5 grados. En cuanto al tiempo de computación, hemos

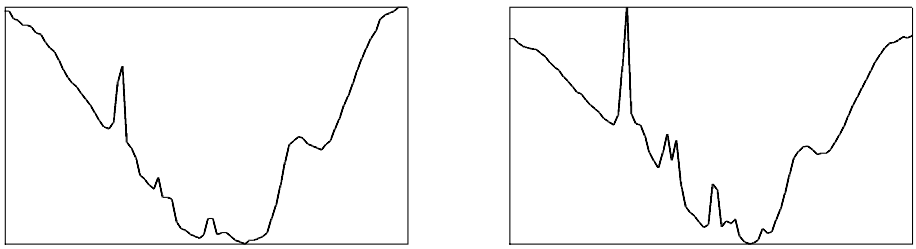

Figura 4.18: Función log posteriori. Izquierda: utilizando el máximo de la función. Derecha: utilizando la suma.

Algoritmo de obtención del ángulo de orientación de la cámara ALG_CAA

Entrada: Imagen de entrada
Salida: Ángulo de orientación detectado.

Aplicar el algoritmo de detección de uniones y el de agrupamiento de uniones.
Convertir los caminos en segmentos rectos.
Para cada segmento **hacer**
  Para cada ángulo posible ($[-45, 45]$) **hacer**
    Calcular la evidencia del segmento con el ángulo actual mediante la Ecuación 4.6 y utilizando el ángulo de la normal del segmento. Multiplicar la evidencia por la longitud del segmento.
    Acumular la evidencia para el ángulo en concreto.
  FinPara
FinPara
Devolver el ángulo con valor máximo de evidencia.

Figura 4.19: Algoritmo de obtención del ángulo de orientación de la cámara utilizando información de arista.

obtenido una media de 0.2 segundos, sin contar el tiempo de procesamiento de las aristas. Las Figuras 4.20 a 4.25 muestran ejemplos de aplicación del algoritmo propuesto. Las imágenes inferiores muestran el resultado del agrupamiento una vez aplicado el método para detectar segmentos rectos.

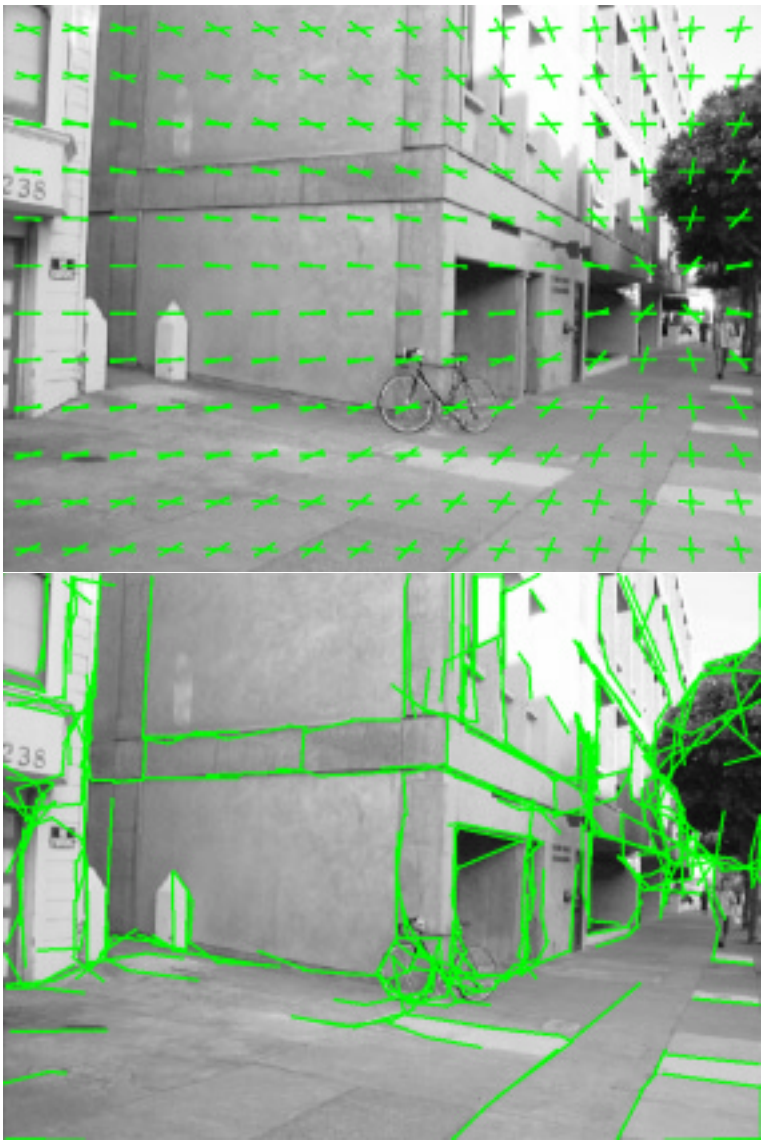

Figura 4.20: Resultado obtenido en imagen de exterior. El ángulo de orientación de la cámara obtenido fue de -15º.

Figura 4.21: Resultados obtenidos en imágenes de exterior. El ángulo de orientación de la cámara obtenido fue de 22º.

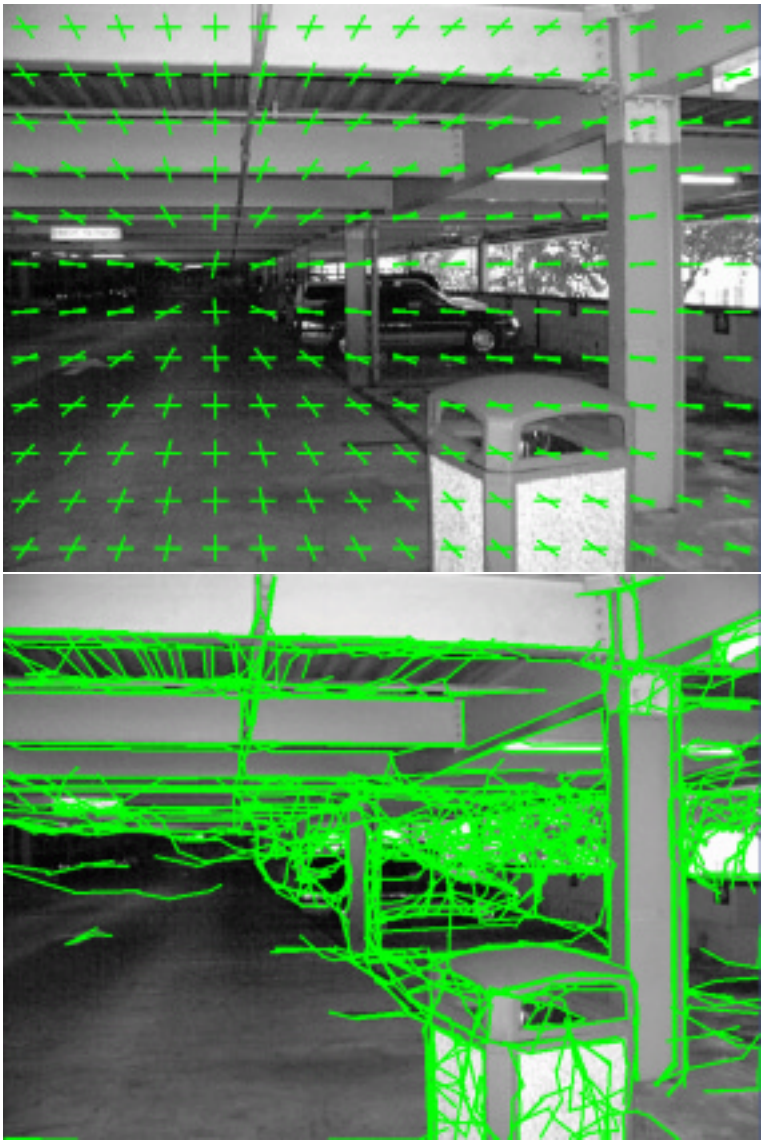

Figura 4.22: Resultados obtenidos en imágenes de exterior. El ángulo de orientación de la cámara obtenido fue de 9º.

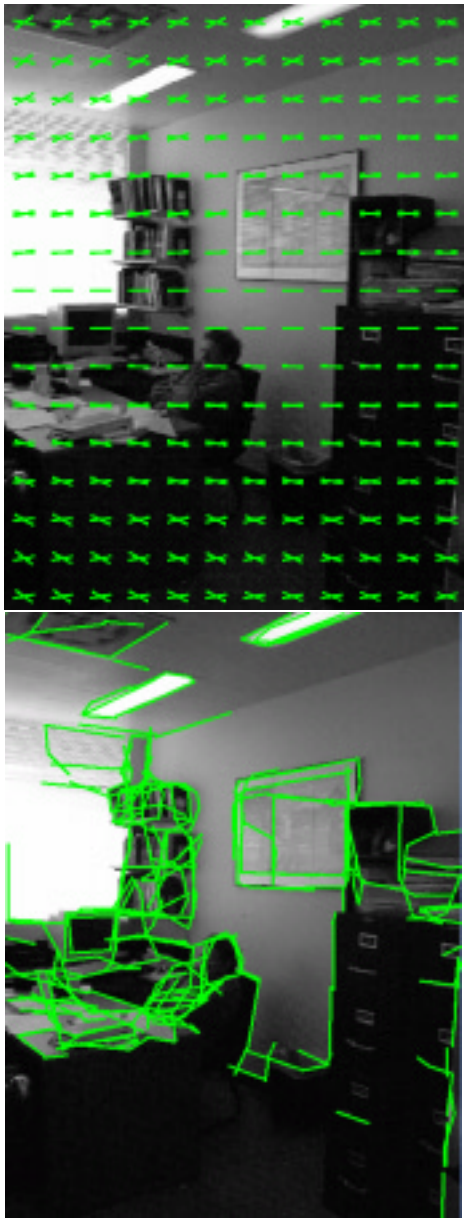

Figura 4.23: Resultados obtenidos en imágenes de interior. El ángulo de orientación de la cámara obtenido fue de 34º.

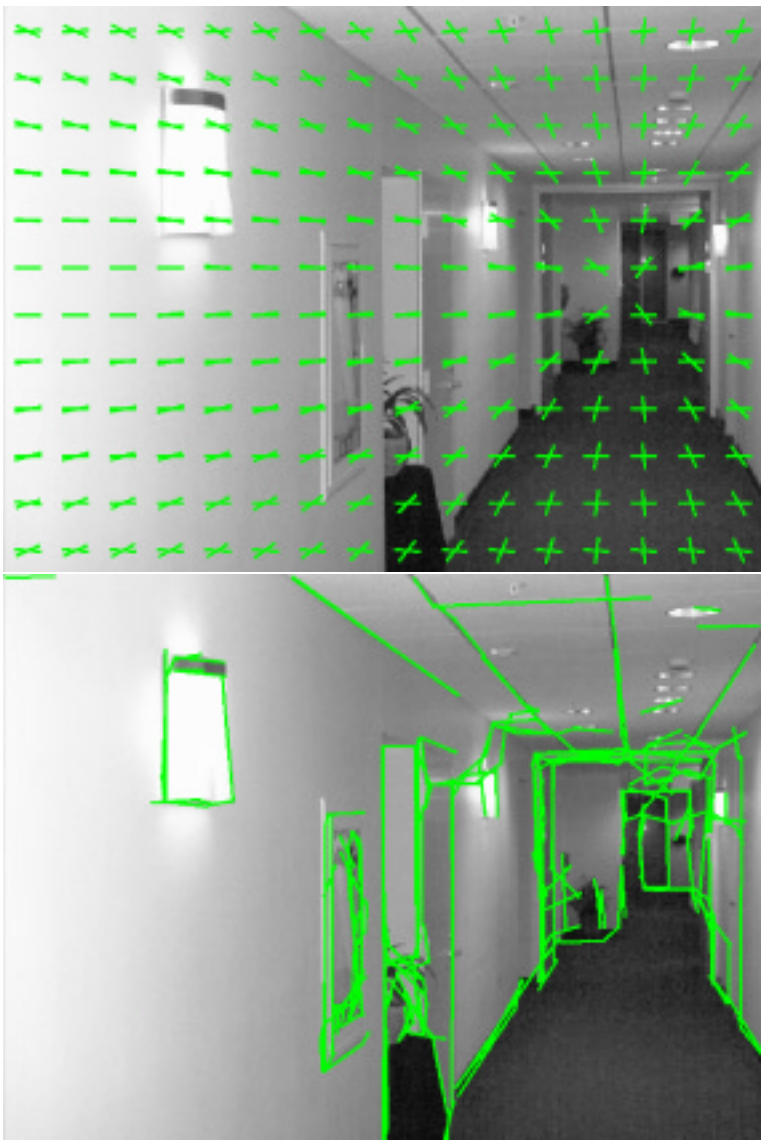

Figura 4.24: Resultados obtenidos en imágenes de interior. El ángulo de orientación de la cámara obtenido fue de -13º.

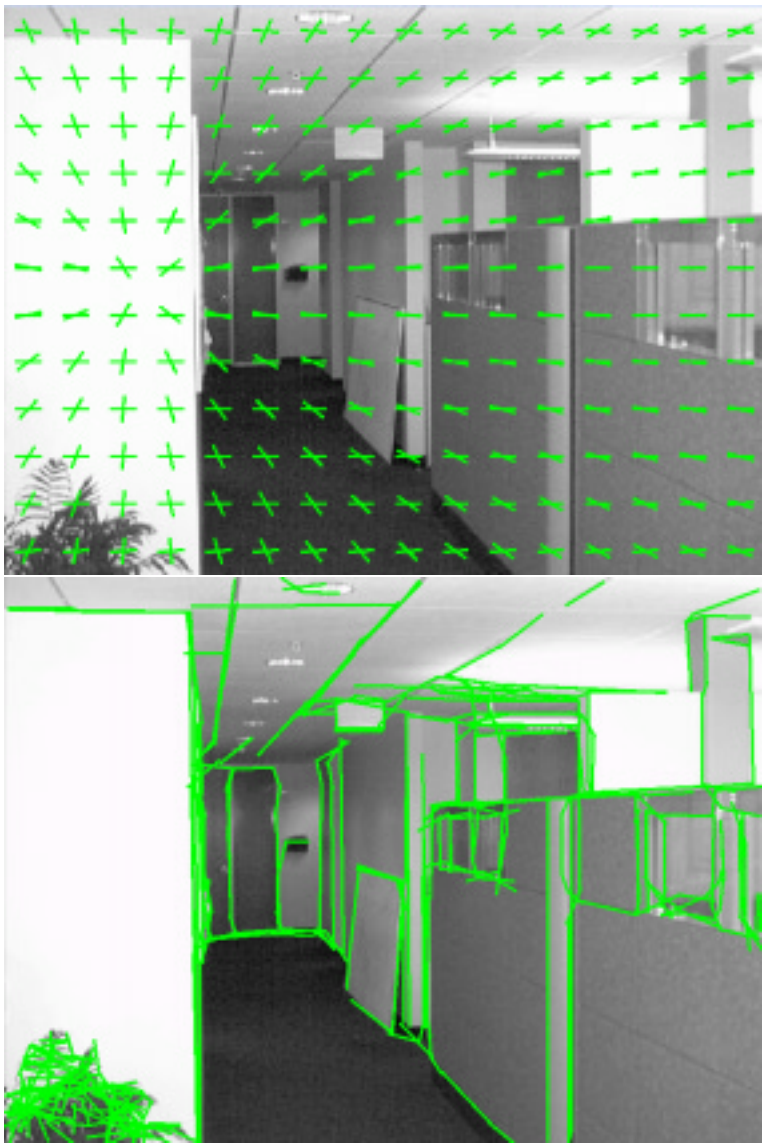

Figura 4.25: Resultado obtenido en imagen de interior. El ángulo de orientación de la cámara obtenido fue de 13º.

4.4.5 Etiquetado de los segmentos

Una posible aplicación del cálculo del ángulo de orientación es la eliminación de aquellas aristas que no se alineen con el mundo de Manhattan. En el trabajo de [Coughlan y Yuille, 1999] se obtiene el conjunto de puntos que no se etiquetan como pertenecientes a una de las tres orientaciones. En nuestro caso, hemos eliminado aquellas aristas que no cumplan las restricciones de Manhattan. En las Figuras 4.26 y 4.27 se muestra el resultado de eliminar estas aristas. Las imágenes de arriba muestran aquellas aristas no alineadas con el mundo de Manhattan. Las de debajo sí que lo están.

4.5 Discusión

El objetivo en este capítulo ha sido la obtención del ángulo de orientación de la cámara. Primero hemos presentado las características básicas de un novedoso método de detección de dicho ángulo que utiliza información bayesiana de pertenencia de un determinado píxel a una arista. Hemos realizado experimentos para comprobar su funcionamiento, proponiendo mejoras para solventar algunos problemas. Se ha comprobado que el tiempo de ejecución es excesivo para el el dominio de trabajo de esta aplicación, la robótica móvil. Por ello hemos adaptado este método para, siguiendo también una formulación bayesiana, tomar como fuente el resultado del agrupamiento descrito en el capítulo anterior. El nuevo método presenta una mejora sustancial en cuanto a tiempo de computación y el error medio cometido es prácticamente despreciable. Estos experimentos muestran que el agrupamiento de uniones puede ser una estructura útil para la simplificación de ciertas tareas en robótica móvil.

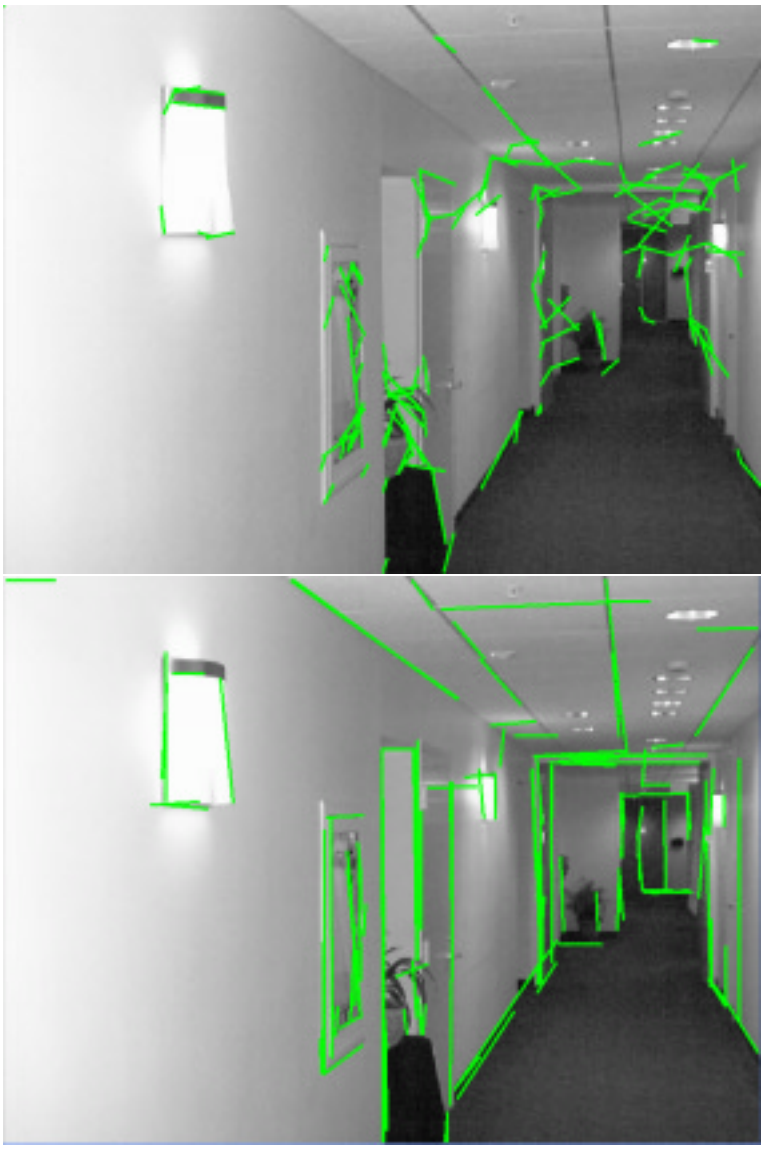

Figura 4.26: Separación de aristas dependiendo de si están alineadas con el mundo de Manhattan o no.

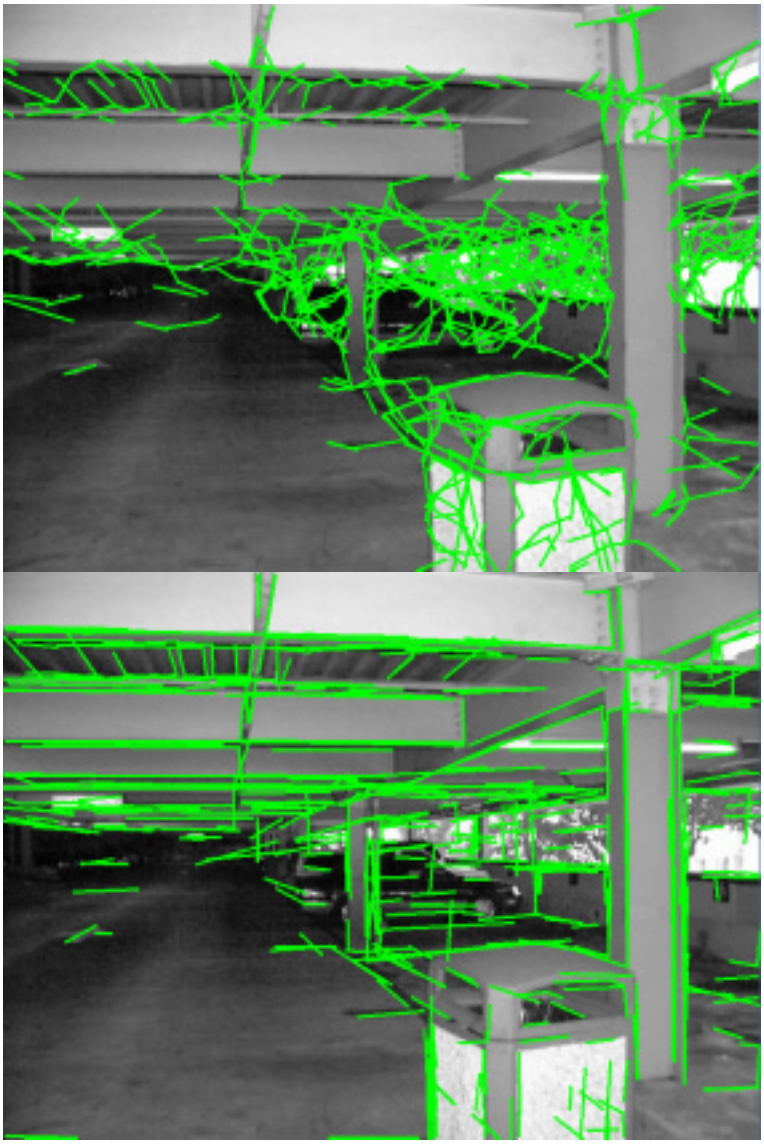

Figura 4.27: Separación de aristas dependiendo de si están alineadas con el mundo de Manhattan o no.

Capítulo 5

Conclusiones. Líneas de trabajo futuro

Este trabajo se centra dentro del campo de la robótica móvil. En concreto, se han presentado distintos métodos que permiten la extracción de características geométricas de la imagen. Estas características obtenidas servirán como elementos de entrada en un sistema robótico para resolver problemas perceptuales concretamente la detección del ángulo de orientación de la cámara. La resolución de este problema puede llegar a solventar los problemas derivados de los errores de odometría en las tareas de localización.

Todos los métodos presentados tienen una formulación bayesiana, redundando en la robustez de los mismos.

A continuación planteamos las distintas aportaciones que se realizan en este trabajo:

1. *Definición de un detector y clasificador de uniones.* Hemos presentado un novedoso sistema de detección y clasificación de uniones. El sistema tiene como característica principal un tiempo de computación muy bajo. Hemos realizado comparaciones entre nuestro detector y Kona, comprobándose la superioridad de nuestro método en cuanto a tiempo de computación y error medio cometido. También hemos realizado comparaciones entre dos de los métodos de detección de puntos característicos más eficientes y eficaces: el de SUSAN y el de Nitzberg.

2. *Definición de un esquema de agrupamiento de uniones.* Debido a los problemas planteados por el detector de uniones, hemos definido un esquema de agrupamiento de uniones que utiliza la información del gradiente de la imagen para realizar dicha agrupación. Este esquema es lo suficientemente flexible para poder modelar caminos con una cierta curvatura.

3. *Desarrollo de un método de detección del ángulo de orientación de la cámara.* Hemos estudiado un esquema novedoso de detección del ángulo de orientación de la cámara a partir de una imagen. Hemos propuesto mejoras al esquema original. Una de ellas es la utilización de la información proporcionada por el agrupamiento de uniones para obtener dicho ángulo. Comprobamos una mejora sustancial en cuanto a tiempo de cómputo del ángulo, que permite su utilización en un sistema robótico.

En cuanto a posibles ampliaciones de los métodos aquí presentados o líneas de trabajo futuro, podemos destacar las siguientes:

1. *Mejora en la localización del centro de las uniones.* Los clasificadores de uniones propuestos presentan una sensibilidad alta al error cometido en la localización del centro de la unión. Una posible mejora es la reducción de dicho error.

2. *Utilización de los extractores de características a tareas de segmentación y reconocimiento.* Con la información obtenida a partir del detector de uniones y del agrupamiento de uniones se puede plantear el definir un esquema para segmentar la imagen en regiones con características similares (nivel de gris similar, textura, etc.). También podemos utilizar esta información en tareas de reconocimiento de objetos, que puede ayudar al sistema robótico en el cual está integrado.

3. *Cálculo de la profundidad.* Haciendo uso de la correspondencia entre características, en nuestro caso tanto de uniones como del agrupamiento, en el tiempo podemos estimar la profundidad de los objetos en la escena.

4. *Disminución del tiempo de cálculo del ángulo de orientación.* El tiempo de cálculo del ángulo de orientación es elevado, teniendo en cuenta que el sistema se ha desarrollado pensando en su implantación en un robot. Una posible mejora se obtendría implementando el sistema en DSP's. Otra es incorporar el procesamiento temporal en la formulación. Se han propuesto distintos esquemas, que pasan desde el filtro de Kalman hasta el algoritmo Condensation. Este último ha tenido una gran aceptación en el campo de la visión robótica. Desde nuestro punto de vista, la aplicación de este esquema a nuestra propuesta es inmediata.

5. *Detección del resto de ángulos de orientación.* Incorporar el resto de ángulos de orientación a la formulación definida para poder obtener el ángulo de orientación total de la cámara.

Parte I

Apéndices

Apéndice A

INFERENCIA BAYESIANA

El razonamiento bayesiano proporciona un enfoque probabilístico a la inferencia. Está basado en la suposición de que las cantidad de interés son gobernadas por distribuciones de probabilidad y que se pueden tomar decisiones óptimas razonando sobre estas probabilidades junto con los datos obtenidos. Este enfoque está siendo utilizado en multitud de campos de investigación, de los que cabe destacar la robótica móvil y la visión computacional, ambas relacionadas con el contenido de esta tesis. En este apéndice queremos definir dos de las herramientas utilizadas en el desarrollo de esta tesis: el teorema de Bayes y el principio de longitud de descripción mínima.

A.1 Teorema de Bayes

A menudo nos surgen problemas en los cuales estamos interesados en determinar la mejor hipótesis h, dados los datos que hemos observado D. Una forma más correcta de expresar esto es decir que buscamos la hipótesis h más probable, dados los datos observados D más un conocimiento inicial sobre las probabilidades a priori de h. El teorema de Bayes nos proporciona un método directo para calcular estas probabilidades.

El teorema de Bayes se define con la siguiente ecuación:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Vamos a comentar el significado de cada término. $P(h)$ es el conocimiento inicial que tenemos sobre que la hipótesis h sea la correcta. Se le suele denominar la

probabilidad a priori de h. $P(D)$ se define de forma similar, pero esta vez sobre los datos D. $P(D|h)$ denota la probabilidad de observar los datos D dado que tenemos la hipótesis h. Se le suele denominar verosimilitud. Por último, $P(h|D)$ es la probabilidad a posteriori que la hipótesis h tiene, dados los datos observados D. En la mayoría de problemas donde se plantea la inferencia bayesiana, se parte de un conjunto de hipótesis H y se trata de encontrar la hipótesis más probable $h \in H$. De esta forma, a esta hipótesis más probable se le suele denominar hipótesis maximum a posteriori o MAP. Utilizando el teorema de Bayes, diremos que h_{MAP} es una hipótesis MAP de acuerdo a:

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} = \arg\max_{h \in H} P(D|h)P(h)$$

En el último paso hemos eliminado $P(D)$ porque es independiente de h.

En algunos casos todas las probabilidades en H son igualmente probables a priori ($P(h_i) = P(h_j), \forall h_i, h_j \in H$). En este caso sólo utilizaríamos el término de verosimilitud, $P(D|h)$, y podemos simplificar aún más la anterior ecuación:

$$h_{ML} \equiv \arg\max_{h \in H} P(D|h)$$

donde a la hipótesis h_{ML} se le suele nombrar como hipótesis de máxima verosimilitud (Maximum Likelihood).

Supongamos ahora que debemos elegir entre dos hipótesis, h_1 y h_2, dados los datos D. El criterio de elección para responder de forma eficiente sería seleccionar la hipótesis más probable. Es decir, aplicaríamos lo que se conoce como regla de decisión:

$$\text{si} \quad P(h_1|D) > P(h_2|D) \quad \text{elegir} \quad h_1, \quad \text{sino elegir} \quad h_2$$

Si aplicamos la regla de Bayes a cada término nos queda:

$$P(D|h_1)\frac{P(h_1)}{P(D)} > P(D|h_2)\frac{P(h_2)}{P(D)}$$

$$\underbrace{\frac{P(D|h_1)}{P(D|h_2)}}_{\text{ratio verosimilitud}} > \underbrace{\frac{P(h_2)}{P(h_1)}}_{\text{ratio a priori}}$$

Aplicando logaritmos a ambas partes nos queda:

$$\ln \frac{P(D|h_1)}{P(D|h_2)} > \ln \frac{P(h_2)}{P(h_1)}$$

En ausencia de información a priori todas las hipótesis son igualmente probables y el término de la derecha es $\ln 1 = 0$. La regla de decisión en ausencia de información a priori queda:

$$\text{si} \quad \ln \frac{P(D|h_1)}{P(D|h_2)} > 0 \quad \text{elegir} \quad h_1, \quad \text{sino elegir} \quad h_2$$

A.2 Principio de longitud de descripción mínima

El principio de longitud de descripción mínima (*minimum descripción length (MDL)*) puede ser resumido como "elegir la explicación más corta a los datos observados". Esta íntimamente relacionada con el criterio MAP antes comentado, incorporando conceptos básicos de teoría de la información. Retomando la definición de h_{MAP}:

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

y, de forma equivalente, expresando esta ecuación en términos de la maximización de $\log_2$:

$$h_{MAP} = \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h)$$

o, alternativamente, minimizando el negativo de esta cantidad:

$$h_{MAP} = \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h)$$

Esta última ecuación puede ser interpretada como que se prefieren hipótesis cortas. Cada uno de estos términos se puede entender como la longitud de descripción de las distribuciones bajo una codificación óptima. No vamos a entrar en comentar los términos de teoría de información. El principio MDL recomienda la elección de las hipótesis que minimizan estas dos longitudes de descripción. Así, este principio se puede definir como elegir la hipótesis h_{MDL} dada:

$$h_{MDL} = \arg\max_{h \in H} L_{C_1} P(D|h) + L_{C_2} P(h)$$

siendo L_{C_i} la longitud de descripción del mensaje i con respecto a C, que es el número de bits requeridos para codificar el mensaje i utilizando el código C. En el caso de que C_2 sea la codificación óptima de las hipótesis (h) y C_1 sea la codificación óptima de ($D|h$), entonces $h_{MDL} = h_{MAP}$.

A.3 Hipótesis de máxima verosimilitud y error cuadrático medio

También vamos a demostrar una equivalencia entre la hipótesis de máxima verosimilitud y el método que encuentra la hipótesis que minimiza el error cuadrático medio. Como ya hemos comentado:

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

Vamos a asumir que los datos $D = (d_1, d_2, \ldots, d_m)$ son independientes dado h, y así escribir podemos escribir $P(D|h)$ como el producto de los distintos $P(d_i|h)$:

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} P(d_i|h)$$

Suponiendo que las funciones de distribución son normales, con media μ y varianza σ^2, vamos a hacer coincidir la hipótesis h con la media μ. Sustituyendo en la ecuación anterior tenemos:

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-\mu)^2}$$

Siendo $\ln P$ una función monotónica de P, maximizar $\ln P$ equivale a maximizar P, y eliminando la varianza, que no depende de la hipótesis h, nos queda:

$$h_{ML}) = \arg\max_{h \in H} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - \mu)^2 = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - \mu)^2$$

Esta ecuación muestra que la hipótesis de máxima verosimilitud es la que minimiza la suma de los errores cuadráticos entre la hipótesis y los datos, siempre en los supuestos antes mencionados.

Apéndice B

Búsquedas dentro de un rango

Un problema clásico en geometría computacional es la búsqueda dentro de un rango. El problema se puede definir de la siguiente manera:

Tenemos un conjunto de puntos en el espacio n-dimensional. Se trata de encontrar de forma eficiente el conjunto de puntos cuyas coordenadas se encuentran situadas dentro de un rango. En el caso de dos dimensiones el rango especificado podría ser $[x_1 : x_2] \times [y_1 : y_2]$, indicando que las coordenadas de los puntos que se devuelven tienen que estar comprendidas entre dichos intervalos.

Se pueden encontrar varias propuestas en la literatura de tipos de datos para realizar estas búsquedas. En [de Berg *et al.*, 1997] se presentan algunas de las estructuras de datos más eficientes para realizar estas consultas, de las cuales se ha seleccionado la que presenta una menor complejidad en el tiempo de consulta. La estructura seleccionada es el árbol de rango (*Range Tree*) que es una ampliación del clásico árbol binario en una dimensión. En la siguiente sección se comenta la estructura del árbol binario y la forma de interrogar a dicho árbol, analizándose la complejidad de dicha estructura. En la Sección B.2 se detalla cómo se puede crear la estructura de árbol de rango y cómo se realiza la consulta de rango a dicha estructura.

B.1 Búsqueda unidimensional

Una búsqueda en una única dimensión es trivial: ordenamos todos los valores en un vector y simplemente realizamos una búsqueda lineal viendo qué valores se encuentran entre el rango especificado. Sin embargo aquí vamos a introducir una estructura de datos llamada *árbol binario* y que sirve de base para ampliar a mayores dimensiones. Cada nodo de este árbol tendrá la siguiente información: Un valor $\mathcal{V}$, un puntero a un subárbol izquierda $\mathcal{L}$ y otro a un subárbol derecha $\mathcal{R}$. Dado un conjunto de valores ordenados de menor a mayor, la creación del árbol binario se puede definir de forma recursiva tal como se muestra en la Figura B.1.

Algoritmo de creación de un árbol binario ALG_CAB

Entrada: Lista de valores ordenados L.
Salida: Nodo raíz del árbol creado T.

Si *Longitud*(L)=1 **entonces** `/*Un único elemento*/`
Crear un nuevo nodo N con $\mathcal{V} = Obtener_Elemento(L)$ y los punteros de los subárboles apuntando a un árbol vacío.
sino
Obtener el nodo cuyo valor es el central de la lista de valores (la mediana): v_{med}
Particionar la lista L en dos sublistas L_1 y L_2 de tal forma que L_1 contiene todos los elementos menores o iguales a v_{med} y L_2 los estrictamente mayores.
Crear un nuevo nodo N con $\mathcal{V} = v_{med}$. El puntero izquierdo $\mathcal{L}$ apuntará al árbol devuelto en la llamada a la función $Alg_CAB(L_1)$ y el derecho $\mathcal{R}$ ídem $Alg_CAB(L_2)$.
FinSi
Devolver N.

Figura B.1: Algoritmo de creación de un árbol binario.

Si tenemos los siguientes valores: 3, 5, 10, 15, 16, 20, 22, 23, 30, el resultado de la creación de este árbol se puede observar en la Figura B.2. Los valores almacenados en los nodos internos permiten guiar la búsqueda hacia los valores en las hojas. Todos los valores en el subárbol izquierdo de un determinado nodo contienen valores menores o iguales que el valor almacenado en dicho nodo, mientras que en el subárbol derecho estos valores son mayores.

El método de consulta parte de un determinado rango a buscar $[x_1 : x_2]$ y devuelve los puntos del árbol que se encuentran dentro de ese rango. El método se divide en dos fases: primero encontrar el nodo de partición para, a continuación, ir explorando a partir de este nodo e ir devolviendo los valores dentro del rango. Este nodo de partición (v_{part}) es aquel nodo del árbol cuyo valor es mayor que x_1 y menor que x_2, es decir se encuentra dentro del rango a buscar, y que además tiene

menor profundidad (más próximo a la raíz). En la Figura B.2 se muestra el nodo de partición para un rango de búsqueda de $[4:15]$. La Figura B.3 muestra el resultado de buscar un determinado rango en el árbol binario. Primero encontramos el nodo de partición. Después buscamos por el hijo izquierdo, quedándonos con los subárboles derechos que se encuentren dentro del rango, mientras que cuando buscamos por el hijo derecho nos quedamos con los subárboles izquierdos. En la Figura B.5 se detalla el método de consulta de rango.

La función Devolver_Subárbol() devuelve una lista con todos los puntos que pertenecen al árbol pasado como parámetro a la función.

La creación de este árbol se puede realizar con una complejidad temporal de $\bigcirc(n \log n)$ y una espacial de $\bigcirc(n)$. En cuanto a la complejidad de la consulta, siendo k el número de puntos devueltos se tiene una complejidad temporal de $\bigcirc(k + \log n)$.

B.2 Árboles de rango

Este tipo de árbol permite obtener, de manera eficiente, los puntos dentro de un espacio bidimensional cuyas coordenadas se encuentran dentro de un determinado rango. Si P es el conjunto de puntos totales, una consulta de rango

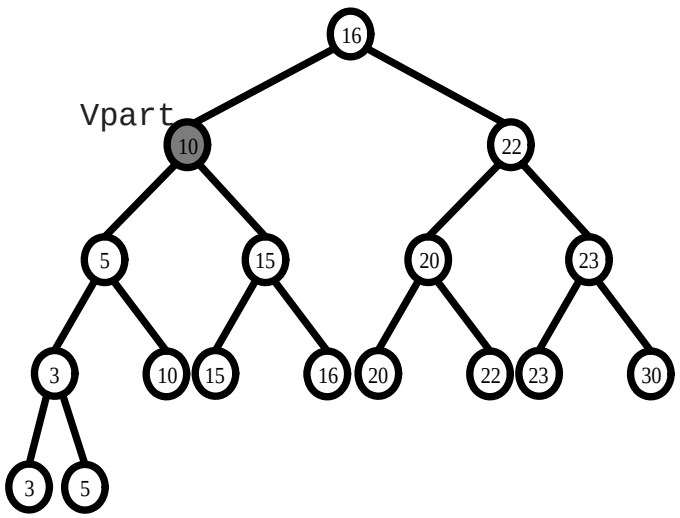

Figura B.2: Árbol binario creado indicando el nodo de partición.

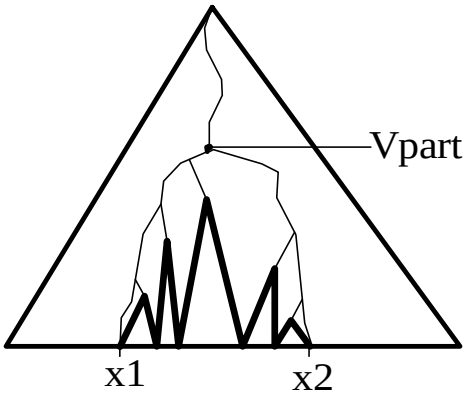

Figura B.3: Obtención de los puntos dentro de un rango.

($[x_1 : x_2] \times [y_1 : y_2]$) en dos dimensiones sobre P devolverá todos aquellos puntos que estén situados dentro del rectángulo definido por el rango de consulta. Un punto $p = (p_x, p_y)$ está situado dentro de dicho rectángulo si se cumple que:

$$p_x \in [x_1 : x_2] \quad \text{y} \quad p_y \in [y_1 : y_2]$$

Una consulta de este tipo se puede dividir en dos subconsultas, una para las coordenadas x y otra para las y. El árbol de rango es un árbol binario en el cual se han utilizado las coordenadas x de los puntos para su construcción. La consulta principal se realiza sobre estas coordenadas. Asociado a cada nodo de este árbol principal tenemos un árbol binario como el descrito en la anterior sección pero construido a partir de las coordenadas y de los puntos hijos de este nodo. La construcción de esta estructura es idéntica a la del árbol binario, excepto que el primer paso del algoritmo debe ser construir un árbol binario, asociado al nodo en cuestión, con las coordenadas y de los puntos (ver Figura B.4).

En cuanto a la consulta, el algoritmo también es muy similar al descrito para consulta en árboles binarios. Se realiza una consulta sobre las coordenadas x del árbol, y cuando se llamaba a la función Devolver_Subárbol ahora se debe realizar una consulta al árbol asociado con el rango de las y.

Si tenemos un conjunto P de n puntos, la creación de un árbol de rango se puede realizar con una complejidad temporal de $\bigcirc(n \log n)$ y una espacial de $\bigcirc(n \log n)$. En cuanto a la complejidad de la consulta, siendo k el número de puntos devueltos se tiene una complejidad temporal de $\bigcirc(k + \log^2 n)$.

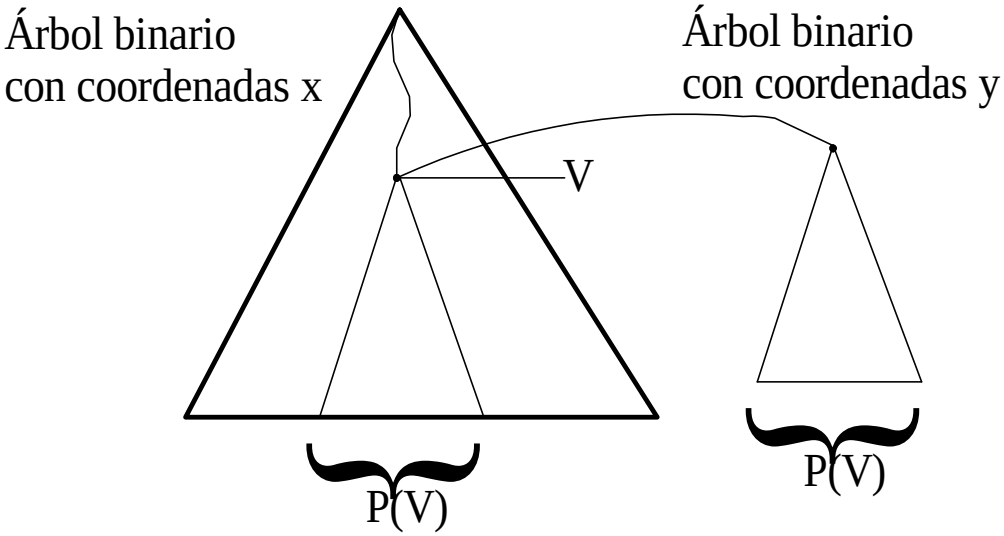

Figura B.4: Árbol de rango con el árbol binario asociado.

Algoritmo de consulta en un árbol binario ALG_COAB

Entrada: Un árbol $\mathcal{T}$ y un rango de búsqueda $[x_1 : x_2]$
Salida: Lista de puntos del árbol que se encuentran dentro del rango.

Crear una lista vacía $Lista$
v_{part}=raíz($\mathcal{T}$)
x_{part}=valor(v_{part})
mientras v_{part} no es hoja **y** ($x_2 \leq x_{part}$ **o** $x_1 > x_{part}$) **hacer**
    Si $x_2 \leq x_{part}$ **entonces**
        v_{part}=Hijo_izquierdo(v_{part})
    sino
        v_{part}=Hijo_derecho(v_{part})
    FinSi
    x_{part}=valor(v_{part})
FinMientras

```
Si v_part es hoja entonces
    Si (x_part ≥ x_1 y x_part ≤ x_2) entonces Insertar x_part en Lista FinSi
sino
    /* Primero se sigue el camino de la izquierda */
    v = v_part
    v=Hijo_izquierdo(v)
    mientras v no es hoja hacer
        x_v=valor(v)
        Si x_1 ≤ x_v entonces
            Concatenar (Lista, Devolver_Subárbol(Hijo_derecho(v)))
            v=Hijo_izquierdo(v)
        sino
            v=Hijo_derecho(v)
        FinSi
    FinMientras
    /* El nodo hoja también debe de ser chequeado */
    Si (x_v ≥ x_1 y x_v ≤ x_2) entonces Insertar x_v en Lista FinSi
    /* Ahora se continúa por la derecha */
    v = v_part
    v=Hijo_derecho(v)
    mientras v no es hoja hacer
        x_v=valor(v)
        Si x_2 ≥ x_v entonces
            Concatenar (Lista, Devolver_Subárbol(Hijo_izquierdo(v)))
            v=Hijo_derecho(v)
        sino
            v=Hijo_izquierdo(v)
        FinSi
    FinMientras
    /* El nodo hoja también debe de ser chequeado */
    Si (x_v ≥ x_1 y x_v ≤ x_2) entonces Insertar x_v en Lista FinSi
FinSi
Devolver Lista
```

Figura B.5: Algoritmo de consulta de un árbol binario.

Apéndice C

Extracción de segmentos rectos

En este apéndice vamos a detallar un algoritmo de extracción de segmentos rectos, a partir de un segmento de forma libre. El algoritmo es el siguiente:

Algoritmo de extracción de segmentos rectos ALG_EXT_SEGMENTOS

Entrada: Conjunto de segmentos a particionar.
Salida: Conjunto de segmentos particionados.

Para cada segmento del conjunto de entrada **hacer**
Encontrar un punto de ruptura. Este punto es aquel cuya distancia a la recta definida por los puntos inicial y final del segmento sea máxima.
Creamos dos segmentos resultado de particionar el segmento original por el punto de ruptura.
Cada segmento tendrá asociado un valor que viene determinado por el ratio longitud/desviación. Si un segmento es demasiado pequeño, su ratio es cero. Si el valor de un segmento es el mismo que el segmento que lo originó, no se realiza la partición. En caso contrario, calculamos recursivamente el valor del segmento.
Para los segmentos finales se ajusta el mejor segmento que pase por todos los puntos, mediante un esquema de mínimos cuadrados.
FinPara

La Figura C.1 muestra un ejemplo de aplicación de este método.

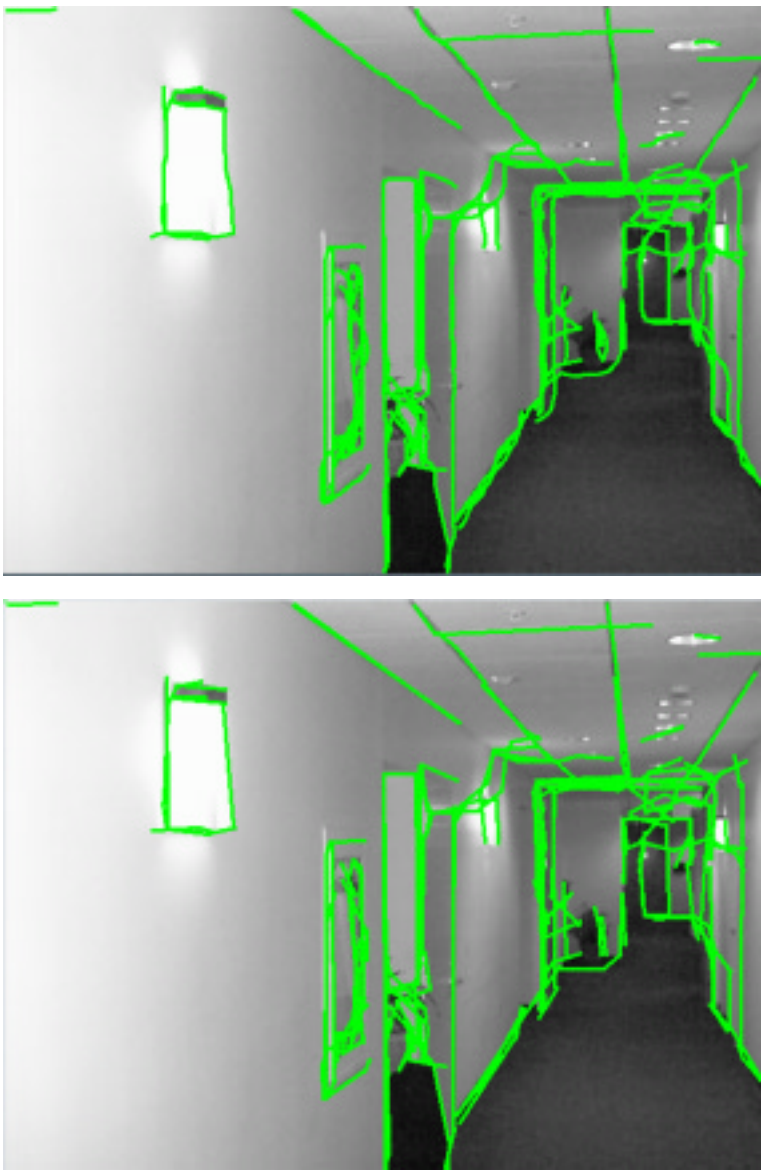

Figura C.1: Ejemplo de obtención de los segmentos rectos a partir del resultado del agrupamiento.

BIBLIOGRAFÍA

[Aloimonos *et al.*, 1988] Y. Aloimonos, I. Weiss, y A. Bandopadhay. Active vision. *International Journal of Computer Vision*, 2:333–356, 1988.

[Alvarez y Morales, 1997] L. Alvarez y F. Morales. Affine morphological multiscale analysis of corners and multiple junctions. *International Journal of Computer Vision*, 25(2):95–107, 1997.

[Asada y Brady, 1986] H. Asada y M. Brady. The curvature primal sketch. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):2–14, 1986.

[Bajcsy, 1988] R. Bajcsy. Active perception. In *Proc of the IEEE*, 1988.

[Ballard, 1991] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[Barnard, 1983] S. Barnard. Interpreting perspective images. *Artificial Intelligence*, 21, 1983.

[Bartlett *et al.*, 1998] M. Bartlett, H. Martin, y T.J. Sejnowski. Independent component representations for face recognition. In *Proc de la SPIE Symposium on Electric Imaging*, 1998.

[Beaudet, 1978] P.R. Beaudet. Rotational invariant image operators. In *Proc. of the 4th. International Conference on Pattern Recognition*, 1978.

[Belhumeur, 1996] P. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3), 1996.

[Bell y Pau, 1990] B. Bell y L.F. Pau. Contour tracking and corner detection in a logic programming environment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):913–917, September 1990.

[Brillault-O Mahoney, 1991] B. Brillault-O Mahoney. New method for vanishing pint detection. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54, 1991.

[Brunstrom *et al.*, 1996] K. Brunstrom, J. Eklundh, y T. Uhlin. Active fixation for scene exploration. *International Journal of Computer Vision*, 17(2), February 1996.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[Cazorla *et al.*, 1999a] M.A. Cazorla, F. Escolano, D. Gallardo, y O. Colomina. A competition-based deformable template for junctions detection. In *Proc de VIII SNRFAI*, 1999.

[Cazorla *et al.*, 1999b] M.A. Cazorla, F. Escolano, D. Gallardo, y R. Rizo. Bayesian models for finding and grouping junctions. In *Proc of the EMMCVPR 99*. Lectures Notes in Computer Science, 1999.

[Coughlan y Yuille, 1998] J. Coughlan y A.L. Yuille. Bayesian a* tree search with expected $o(n)$ convergence rates for road tracking. *Artificial Intelligence*, 1998.

[Coughlan y Yuille, 1999] J. Coughlan y A. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proc of the International Conference on Computer Vision*, 1999.

[Coughlan, 1998] J. Coughlan. *Efficient Search Algorithms for Finding Deformable Templates*. PhD thesis, Harvard University, Cambridge, Massachusetts, January 1998.

[Cover y Thomas, 1991] T. Cover y J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[de Berg *et al.*, 1997] M. de Berg, M. van Kreveld, M. Overmars, y O. Schwarkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997.

[Deriche y Faugeras, 1990] R. Deriche y O.D. Faugeras. 2d curve matching using high curvature points: Application to stereo vision. In *Proc. of the 10th. International Conference of Pattern Recognition*, pages 240–242, 1990.

[Deriche y Giraudon, 1993] R. Deriche y G. Giraudon. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124, 1993.

[Dickmanns, 1997] E.D. Dickmanns. Vehicles capable of dynamic vision. In *Proc de la 15 IJCAI*, 1997.

[Dreschler y Nagel, 1982] Dreschler y Nagel. On the selection of critical points and local curvature extrema of region boundaries for interframe matching. In *Proc of the International Conference of Pattern Recognition*, pages 542–544, 1982.

[Duda y Hart, 1973] R. O. Duda y P. E. Hart. *Pattern Classification and Scene Analisys*, page 20. John Wiley and Sons, 1973.

[Escolano, 1997] F. Escolano. *Plantillas deformables extendidas: modelización local robusta y caracterización*. PhD thesis, Universidad de Alicante, Mayo 1997.

[Faugeras, 1993] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, Cambridge, Massachusetts, 1993.

[Figuereido *et al.*, 1997] M.A. Figuereido, J. Leitao, y A.K. Jain. Adaptative parametrically deformable contours. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition Workshop*, 1997.

[Gallardo, 1999] D. Gallardo. *Aplicación del muestreo bayesiano en robots móviles: estrategias para localización y estimación de mapas del entorno*. PhD thesis, Universidad de Alicante, Mayo 1999.

[Geman y Geman, 1984] S. Geman y D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[Geman y Jedynak, 1996] D. Geman y B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE tran. on Pattern Analysis and Machine Intelligence*, 18(1), January 1996.

[Giraudon y Deriche, 1991] G. Giraudon y R. Deriche. On corner and vertex detection. In *Proc of the Conference on Computer Vision and Pattern Recognition*, pages 650–655, 1991.

[Grzywacz y Escolano, 2000] N. Grzywacz y F. Escolano. Visions of vision: Vision in brains and computers. MIT Press. Computational Neuroscience Series. En preparacion, 2000.

[Han *et al.*, 1989] M.H. Han, D. Jang, y J. Foster. Identification of corners points of two-dimensional images using a line search method. *Pattern Recognition*, 22(1):13–20, 1989.

[Harris y Stephens, 1988] C. G. Harris y M. Stephens. A combined corner and edge detection. In *Proc of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[Kitchen y Rosenfeld, 1982] L. Kitchen y A. Rosenfeld. Gray level corner detection. *Pattern Recognition Letters*, 1(2):95–102, 1982.

[Kittler, 1997] J. Kittler. Probabilistic relaxation: Potential, relationships and open problems. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition Workshop*, pages 393–408, 1997.

[Knill y Richards, 1996] D. Knill y W. Richards. *Perception as Bayesian inference*. Cambridge University Press, 1996.

[Koenderink y Richards, 1988] J.J. Koenderink y W. Richards. Two-dimensional curvature operators. *Journal of the Optical Society of America*, 5(7):1136–1141, July 1988.

[Konishi *et al.*, 1999] S. Konishi, J. Coughlan, A. Yuille, y C. Zhu. Fundamental bounds on edge detection: Edge cues. In *Proc of the International Conference on Computer Vision and Pattern Recognition*, 1999.

[Krotkov, 1987] E.P. Krotkov. *Exploratory Visual Sensing for Determining Spatial Layout with an Agile Stereo Camera System*. PhD thesis, Univ. of Pennsilvania, 1987.

[Landy *et al.*, 1996] M.S. Landy, T. Maloney, y M. Pavel, editors. *Exploratory Vision. The Active Eye*. Springer Verlag, 1996.

[Lindeberg y Li, 1997] T. Lindeberg y M. Li. Segmentation and clssification of edges using minimum description length approximation and complementary junction cues. *Computer Vision and Image Understanding*, 67(1):88–98, July 1997.

[Lowe, 1985] D.G. Lowe. *Perceptual Organization and Visual Recognition*. Kuwler, 1985.

[Lowe, 1987] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31, 1987.

[Magee y Aggarwal, 1984] M.J. Magee y J.K. Aggarwal. Determining vanishing points from perspective images. *Computer Vision, Graphics and Image Processing*, 26, 1984.

[Malik, 1987] J. Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1(1):73–104, 1987.

[Marr, 1982] D. Marr. *Vision*. Freeman, San Francisco, 1982.

[Matas y Kittler, 1995] J. Matas y J. Kittler. Contextual junctions finder. In J. Crowley y H. Christensen, editors, *Vision as process*. Springer, 1995.

[McLean y Kotturi, 1995] G.F. McLean y D. Kotturi. Vanishing point detection by line clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17, 1995.

[Medioni y Yasumoto, 1987] G. Medioni y Y. Yasumoto. Corner detection and curve representation using cubic B-splines. *Computer Vision, Graphics, and Image Processing*, 39:267–278, 1987.

[Mehrota y Nichani, 1990] R. Mehrota y S.Nichani. Corner detection. *Pattern Recognition*, 23(11):1223–1233, 1990.

[Micheli *et al.*, 1989] E. De Micheli, B. Caprile, P. Ottanelo, y V. Torre. Localization and noise in edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(10), 1989.

[Mitchell, 1997] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Moons *et al.*, 1996] T. Moons, L. Van Gool, M. Proesmans, y E. Pauwels. Affine reconstruction from perspective image pairs with a relative object-camera translation in between. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18, 1996.

[Moravec, 1977] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. of the International Joint Conference on Artificial Intelligence*, page 584, 1977.

[Mundy y Zisserman, 1992] J.L. Mundy y Z. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[Nitzberg *et al.*, 1993] M.Ñitzberg, D. Mumford, y T. Shiota. *Filtering, Segmentation and Depth*. Springer-Verlag, 1993.

[Parida *et al.*, 1997] L. Parida, D. Geiger, y R. Hummel. Kona: A multi-junction detector using minimum description length principle. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition Workshop*, pages 51–65, 1997.

[Parida *et al.*, 1998] L. Parida, D. Geiger, y R. Hummel. Junctions: Detection, classification, and reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(7), 1998.

[Parodi y Piccioli, 1996] P. Parodi y G. Piccioli. 3d shape reconstruction by using vanishing points. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18, 1996.

[Pearl, 1984] J. Pearl. *Heuristics. Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[Quan y Mohr, 1989] L. Quan y R. Mohr. Determining perspective structures using hierarchical hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9, 1989.

[Rangarajan *et al.*, 1989] K. Rangarajan, M. Shah, y D. Van Brackle. Optimal corner detector. *Computer Vision, Graphics, and Image Processing*, 48:230–245, 1989.

[Rattarangsi y Chin, 1992] A. Rattarangsi y R.T. Chin. Scale-based detection of corners of planar curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(4):430–449, 1992.

[Rimey y Brown, 1992] R. Rimey y C. Brown. Task-oriented vision with multiple bayes nets. In A. Yuille y A. Blake, editors, *Active Vision*. MIT Press, Cambridge, MA, 1992.

[Sarkar y Boyer, 1993] S. Sarkar y K.L. Boyer. Perceptual organization in computer vision: a review and a proposal for a classificatory structure. *IEEE Trans. on Systems, Man, and Cybernetics*, 23:382–399, 1993.

[Shufelt, 1999a] J.A. Shufelt. Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21, 1999.

[Shufelt, 1999b] J.A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21, 1999.

[Smith y Brady, 1997] S. M. Smith y J. M. Brady. Susan= a new approach to low level image processing. *Int. J. Comp. Vision*, 23(1):45–78, 1997.

[Torre y Poggio, 1986] V. Torre y T.A. Poggio. On edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(2):147–163, March 1986.

[Trucco y Verri, 1998] S. Trucco y A. Verri. *Introductory Techiques in Computer Vision*. Prentice-Hall, 1998.

[Ullman, 1996] S. Ullman. *High-Level Vision*. MIT Press, 1996.

[Vasselle *et al.*, 1994] B. Vasselle, G. Giraudon, y M. Berthod. Following corners on curves and surfaces in the scale space. In *Proc. of the European Conference on Computer Vision 94*, volume 800, pages 109–114, 1994.

[Waltz, 1975] D. Waltz. Understanding line drawing of scenes with shadows. In *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[Wandell, 1995] B.A. Wandell. *Fondations of Vision*. Sinauer Associates, Inc., 1995.

[Yuille y Blake, 1992] A.L. Yuille y A. Blake. *Active Vision*. MIT Press, 1992.

[Yuille y Coughlan, 1997] A.L. Yuille y J. Coughlan. Twenty questions, fouces of attention, and a*: A theorical comparison of optimization strategies. In *Proc of the EMMCVPR 99*. Lectures Notes in Computer Science, 1997.

[Yuille y Coughlan, 1999] A. Yuille y J. Coughlan. Visual search: Fundamental bounds, order parameters, and phase transitions. *IEEE tran. on Pattern Analysis and Machine Intelligence*, 1999.

[Zerubia y Chellapa, 1993] J. Zerubia y R. Chellapa. Mean field approximation using compound gauss markov random fileds for edge detection and image estimation. *IEEE Trans. on Neural Networks*, 8(4):703–709, 1993.

[Zhu *et al.*, 1997] S.C. Zhu, Y. Wu, y D. Mumford. Minimax entropy principle and its applications to texture modeling. *Neural Computation*, 9(8), 1997.

[Zhu y Yuille, 1996a] S. C. Zhu y A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.

[Zhu y Yuille, 1996b] S.C. Zhu y A.L. Yuille. Forms: A flexible object recognition and modelling system. *International Journal of Computer Vision*, 20(3):187–212, 1996.

[Zuniga y Haralick, 1983] O. A. Zuniga y R. M. Haralick. Corner detection using the facet model. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 30–37, 1983.