

Índice

Tabla de Ilustraciones.....	5
Agradecimientos	8
1. INTRODUCCIÓN.....	9
1.1. CONTEXTO GENERAL DEL PROYECTO	9
1.2. ALCANCE DEL PROYECTO FINAL DE CARRERA DENTRO DEL NMRS.....	10
1.3. OBJETIVOS DEL PROYECTO.....	11
1.4. EL FILTRO DE KALMAN	13
2. Estado del arte	15
2.1. CONSTRUCCIÓN DE MAPAS DINÁMICOS.....	15
2.1.1. Introducción	15
2.1.2. Fusión de la información.....	16
2.1.3. Escenarios y entorno	17
2.1.4. Enfoques y teorías.....	17
2.1.5. Enfoques dinámicos	23
2.1.6. Criterio de vivacidad VS. Criterio de robustez	27
2.1.7. Conclusiones	27
2.2. LOCALIZACIÓN Y MAPEADO SIMULTÁNEO	28
2.2.1. Introducción	28

2.2.2. Tipos fundamentales de SLAM.....	29
3. Tipos y estructuras de los mapas usados. Datos de partida.....	31
3.1. DESCRIPCIÓN GENERAL DE LA ESTRUCTURA MAPA.....	31
3.2. CONVERSIÓN DE COORDENADAS A CELDA EN EL MAPA	35
3.3. ESTRUCTURA MULTICAPA DE CELDAS.....	35
3.3.1. Altura	36
3.3.2. Gradientes	37
3.3.3. Confianza.....	38
3.3.4. Ocupación.....	41
3.3.5. Calidad GPS y calidad de la comunicación	42
3.4. DATOS DE PARTIDA	43
3.4.1. Mapa inicial del terreno en baja resolución (LowResMapGlobal).....	43
3.4.2. Dimensiones y características del robot	43
3.4.3. Aspecto del mapa local.....	45
3.4.4. Trayectoria del robot.....	46
3.4.5. Posición del mapa local en el interior del mapa global.....	47
3.4.6. Sistemas de referencia utilizados	48
4. ALGORITMOS DE RECONSTRUCCIÓN	50
4.1. INTRODUCCIÓN	50
4.2. ESQUEMA GENERAL DEL PROCESO DE RECONSTRUCCIÓN	50
4.2.1. Estimación del estado del robot.....	51
4.3. ESQUEMA GENERAL DEL MÓDULO	53
4.4. PROCESO: COMPROBACIÓN DEL RANGO DEL LÁSER	56

4.5. PROCESO: COMPROBACIÓN DE LA POSICIÓN DEL MAPA MÓVIL	59
4.6. PROCESO : CREACIÓN DEL HRM INICIAL A PARTIR DE LRM	61
4.7. PROCESO: SIMULACIONES DEL TERRENO Y DEL LÁSER	64
4.8. CALCULO DE LAS COORDENADAS CARTESIANAS DE LOS PUNTOS DE IMPACTO.....	66
4.9. ESTIMACIÓN DE ALTURAS Y GRADIENTES	69
4.10. ASIGNACIÓN DE CONFIANZAS	77
4.11. ASIGNACIÓN DE OCUPACIÓN.....	82
4.12. LIBERTAD E INFORMACIÓN DE PUNTOS DE NO IMPACTO	87
4.13. OPERACIONES SOBRE LOS MAPAS	95
4.13.1. Fusión de la información nueva con la información existente en alta resolución.	95
4.13.2. Conversión de alta a baja resolución.....	99
4.13.3. Conversión de baja a alta resolución.....	103
5. Localización visual del robot.....	105
5.1. INTRODUCCIÓN	105
5.2. ALGORITMO BASADO EN COMPARACIÓN DE MAPAS	108
5.3. ALGORITMO BASADO EN RECONSTRUCCIÓN A NIVEL SUBCELDA	116
5.4. MÉTODO SIFT	119
5.4.1. Detección de extremos en el espacio de escalas.....	121
5.4.2. Localización precisa de puntos característicos	126
5.4.3. Asignación de orientación.....	129
5.4.4. El descriptor local de la imagen.....	130

5.5. PRUEBAS REALIZADAS CON SLAM VISUAL A TRAVÉS DEL MÉTODO SIFT	134
5.5.1. Aplicación del método SIFT a las estructuras de mapas.....	136
5.5.2. Configuración de los parámetros del método SIFT	139
5.5.3. Ensayos realizados con imágenes	142
5.5.4. Ensayos realizados con mapas en condiciones operativas	146
5.5.5. Conclusiones sobre el método SIFT aplicado a la localización del robot.....	149
6. CONCLUSIONES	150
7. BIBLIOGRAFÍA	152
8. ANEXOS.....	154
EL FILTRO DE KALMAN DISCRETO	162
EL FILTRO DE KALMAN EXTENDIDO (EKF).....	168

Tabla de Ilustraciones

Figura 1. Esquema a alto nivel de las interfaces del proyecto NM-RS	11
Figura 2. Integración de los diferentes elementos del sistema de posicionamiento	12
Figura 6. Ejemplo de grid construido con el robot B21 (izqda.) y lecturas sonar instantáneas (dcha.) Adaptada de José Maria Cañas y Lía García, 2002. [5].....	16
Figura 7. Modelo borroso sonar para información de ocupación (izquierda) y de vacío (derecha). Adaptada de José Maria Cañas y Lía García, 2002. [5]	21
Figura 8. Modelos histográficos	23
Figura 9. Estado de ocupación en función de los pesos almacenados en la memoria de la cedilla (izda.). Latencia del enfoque por mayoría (dcha.). Adaptada de José María Cañas y Lía García, 2002. [5]	26
Figura 13. Mapa de celdas	31
Figura 14. Estructura general de los mapas	33
Figura 15. Ejemplos de situación del centro del mapa	34
Figura 16. Sistemas de referencia asociados al mapa.....	34
Figura 17. Reconstrucción de la altura de un terreno.....	37
Figura 18. Reconstrucción de los gradientes de un terreno	38
Figura 19. Efecto de la dispersión de los puntos de impacto.....	39
Figura 20. Mapa global con la trayectoria seguida por el robot	40
Figura 21. Reconstrucción global de la confianza	41
Figura 22. Reconstrucción global del grado de ocupación.....	42
Figura 23. Mapa global con la trayectoria del robot.....	46
Figura 24. Características principales del HRM y LRM	47
Figura 25. Desplazamiento del HRM	48
Figura 26. Sistemas de referencia del robot	49
Figura 27. Sentidos de giro de los ángulos de barrido del láser.....	49
Figura 28. Entradas y salidas del módulo 45 (Mapeado)	51
Figura 29. Esquema de la fusión sensorial y estimación del estado del robot.	52
Figura 30. Proceso de reconstrucción y mapeado.	55
Figura 31. Proceso para calcular el desplazamiento del mapa de Alta Resolución	57
Figura 32. Diferentes situaciones del robot dentro del Mapa de alta resolución.	58
Figura 33. Diferentes situaciones del robot dentro del mapa de Alta resolución.	58
Figura 34. Esquema del proceso de comprobación de la posición del mapa móvil	59
Figura 35. Situación relativa del mapa de alta resolución sobre el de baja.....	59
Figura 36. Situación relativa del mapa de alta resolución sobre el de baja.....	60
Figura 37. Creación del HRM inicial a partir del LRM.....	62
Figura 38. Conservación de los datos del HRM en la adquisición de nueva información.....	63
Figura 39. Ejemplo de entorno simulado con Matlab.....	65
Figura 40. Representación de puntos de colisión del láser.....	65
Figura 41. Criterio de signos en asignación de ángulos de barrido horizontal y vertical.....	66

Figura 43. Ejemplo de trayectoria con indicación de los puntos de reconstrucción.	72
Figura 44. Ejemplo de reconstrucción.....	73
Figura 46. Error de reconstrucción discretizado.	74
Figura 45. Discretización de la reconstrucción.....	74
Figura 47. Ejemplo de asignación de puntos de colisión para el cálculo de gradientes.	76
Figura 48. División en subceldas.	78
Figura 49. Coeficiente corrector en función de la distancia medida.	79
Figura 50. Proceso de asignación de confianzas	80
Figura 51. Asignación de confianzas	80
Figura 52. Reconstrucción de la ocupación de un terreno	83
Figura 53. Trayectoria simulada mediante puntos muy cercanos entre si	84
Figura 54. Ocupación teórica dados unos gradientes conocidos.....	85
Figura 55. Reconstrucción de la ocupación para la trayectoria de la Figura 51.....	85
Figura 56. Error en valor absoluto de la reconstrucción de la ocupación.....	86
Figura 57. Esquema del algoritmo de asignación de libertades.....	89
Figura 58. Reconstrucción de la ocupación y asignación de libertad para una trayectoria simple	93
Figura 59. Ocupación calculada de forma simple	94
Figura 60. Ocupación calculada con la información proveniente del algoritmo de asignación de libertades.....	95
Figura 61. Algoritmo de fusión de la información.....	98
Figura 62. Esquema de actualización del HRM	100
Figura 63. Ejemplo de actualización del HRM mediante media ponderada por las confianzas	101
Figura 64. Proceso de fusión del HRM con el LRM. Primero se redimensiona el HRM obteniendo un mapa en baja resolución de un tamaño local. Posteriormente dicho mapa se posiciona adecuadamente dentro de un mapa de dimensiones globales.....	102
Figura 65. Transmisión de información del HRM al LRM.	104
Figura 66. Comparación de mapas de alta resolución reconstruidos e importados desde mapas de baja resolución.	104
Figura 67. Esquema general de un proceso SLAM	107
Figura 68. Diagramas que muestran el proceso de localización en base a características del terreno	108
Figura 69. Localización del robot en el momento de comienzo de la localización visual	111
Figura 70. Desalineamiento de los mapas producidos por error en la estimación de la posición del borde de la celda	111
Figura 71. Mapas perfectamente alineados	112
Figura 72. Extracción de mapas	113
Figura 73. Comparación de errores en la estimación de la posición para dos métricas diferentes.	116
Figura 74. Puntos de referencia a nivel subcelda.	117
Figura 75. Puntos de referencia laterales.	117
Figura 76. Comparación de los algoritmos de nivel celda y subcelda.....	118
Figura 77. Extraccion de características en cada octava	122

Figura 78. Puntos vecinos a un punto característico	123
Figura 79. Representación del descriptor de puntos característicos	130
Figura 80. Porcentaje de puntos característicos encajados correctamente segun varia el tamaño del descriptor y el número de orientaciones.....	132
Figura 81. Eficiencia según el numero de puntos característicos almacenados en la base de datos.....	133
Figura 82. Esquema del offset utilizado en los cálculos de localización por comparación analítica de mapas.....	135
Figura 83. Ejemplo de posicionamiento conocida la distancia y el ángulo a un punto característico	135
Figura 84. Reconstrucción de un mapa local utilizando una única medida del láser	138
Figura 85. LRM con indicación de la zona reconstruida de forma local.....	138
Figura 86. Eliminación del área de la imagen que se escapa del error máximo admisible por el sistema de posicionamiento odometría-ins.....	139
Figura 87. Matching de imágenes con los datos del ensayo nº 1	142
Figura 88. Matching de imágenes utilizando parámetros mas restrictivos	143
Figura 89. Matching de imágenes utilizando una imagen girada y con áreas eliminadas según la configuración de parámetros restrictiva.....	144
Figura 90. Matching de imágenes utilizando una imagen girada y con áreas eliminadas utilizando los parámetros originales de Lowe	145
Figura 91. Matching de imágenes utilizando en a imagen izquierda un 75% de ruido uniforme. Imagen izquierda rotada y con áreas eliminadas.....	146
Figura 92. Método SIFT aplicado al reconocimiento de la posicion del mapa local en el interior del LRM.....	147
Figura 93. Método SIFT aplicado al reconocimiento de la posición del mapa local en el interior del LRM con la configuración original propuesta por Lowe.....	148
Figura 3. Ciclo del filtro de Kalman discreto.....	166
Figura 4: Ciclo completo del filtro de Kalman con ecuaciones de actualización.....	167
Figura 5. Ciclo completo del filtro de Kalman extendido con ecuaciones de actualización	169

Agradecimientos

A mi tutor, Jaime del Cerro por haber orientado y dirigido este proyecto con interés y paciencia.

A Antonio y Gonzalo por su ayuda día a día haciendo más fácil y agradable mi estancia en el departamento.

A SENER por apoyar en todo momento la realización del proyecto y facilitar ayuda e infraestructura.

A mi familia que con sus consejos y apoyo me han hecho sentirme siempre muy acompañado.

1. INTRODUCCIÓN

1.1. CONTEXTO GENERAL DEL PROYECTO

El presente proyecto se enmarca dentro de un proyecto internacional de la Agencia de Defensa Europea (EDA) categoría B denominado Sistemas Multirobot en Red, con el acrónimo NM-RS (Networked Multi-Robot Systems) cuyo principal objetivo es el desarrollo de un demostrador virtual que permita analizar la eficacia de la utilización de sistemas multi-robot en red frente al uso de robots individuales o uso de tropas en tareas del ámbito de la defensa.

Uno de los principales objetivos de la actividad de investigación y tecnología de la EDA es catalizar más colaboraciones en proyectos de investigación, cubriendo una amplia gama de rangos en las tecnologías. En general, estos trabajos colaborativos entran dentro de uno de dos tipos, según lo establecido en la Acción Común de la agencia [1]:

- Categoría A. Los proyectos o programas categoría A han sido propuestos por uno o más países miembros o por el jefe ejecutivo de la Agencia y cuentan con la participación de los 26 países miembros a menos que decidan no participar. Un ejemplo de colaboración de este tipo es el anterior programa de inversión común sobre protección de la fuerza.
- Categoría B. Los proyectos o programas categoría B se establecen por uno o más países miembros y en principio están abiertos a la participación de todos los países miembros. Sin embargo, en la práctica, los proyectos categoría B tienden a involucrar un número menor de países miembros porque estos decidan no entrar a formar parte.

El proyecto NM-RS está centrado en el desarrollo de tácticas y procedimientos para emplear nuevos sistemas robóticos en batallas, destinados a la ejecución de misiones durante periodos mucho más largos que sus homólogos tripulados. NM-RS proporcionará una plataforma de simulación de sistemas compuestos de varios robots que se mueven de forma autónoma supervisada, en entornos estructurados y no estructurados. Adicionalmente, este dispositivo de simulación proporcionará la capacidad para entrenar y controlar sistemas no tripulados en ambientes colaborativos.

Otra capacidad de la plataforma consiste en proporcionar al personal la capacidad para entrenarse con los sistemas robóticos mediante la experimentación con las interfaces de operación, como por ejemplo para determinar el número de sistemas no

tripulados que una tripulación de infantería puede controlar bajo una variedad de escenarios posibles. El software de simulación será usado en el futuro para controlar un equipo robótico.

Como objetivo del proyecto NM-RS, se incluye el desarrollo de algoritmos multi-robot para la realización de tareas cooperativas en presencia de diferentes niveles de comunicación entre los robots.

Estas técnicas innovadoras permitirán a los robots construir memorias de sus experiencias sobre el entorno, evaluar la utilidad de las alternativas de acciones cooperativas y entonces seleccionar acciones a realizar de tal forma que se incremente la probabilidad de que el objetivo global se cumpla, más allá de las decisiones individuales de cada robot [2].

1.2. ALCANCE DEL PROYECTO FINAL DE CARRERA DENTRO DEL NM-RS

El proyecto NM-RS tiene una arquitectura modular. Cada módulo tiene una funcionalidad específica, y comparte información con el resto de los módulos mediante unas interfaces minuciosamente diseñadas.

Dado el gran número de módulos existentes en el conjunto de la aplicación, únicamente se van a describir brevemente los módulos indispensables para la comprensión del presente proyecto final de carrera.

El alcance del proyecto final de carrera se encuentra ubicado como parte principal del módulo denominado “Navegación-DataMapping” (M44).

Los principales aspectos funcionales de este módulo consisten en realización de la fusión sensorial de los sistemas de navegación (Unidad inercial, GPS y odometría) y la utilización de la solución de navegación junto con un sistema de láser de barrido horizontal y vertical para la confección de mapas tridimensionales del entorno, esta última tarea prácticamente coincide con el alcance del proyecto final de carrera que aquí se presenta.

Por otro lado, otra tarea fundamental del módulo (igualmente incluida en el alcance del proyecto final de carrera) consiste en la utilización de la información de los mapas y los datos del láser para la realización de un sistema de localización para las situaciones en las que la precisión o disponibilidad del GPS no permita una correcta localización del robot.

El módulo M44 descrito requiere compartir información de otros módulos de la plataforma de simulación, en concreto con los que proporcionan información de los

sensores de navegación (GSP, odometría, etc.), el módulo que proporciona información del láser tridimensional y los datos físicos del robot.

Por último, el módulo M44 proporciona como salidas a otros módulos la siguiente información:

- Posición estimada del robot.
- Posición estimada realizando SLAM.
- Mapa de baja resolución actualizado.
- Mapa de alta resolución.
- Lista de objetos dinámicos detectados.

Una representación de alto nivel de las interfaces mencionadas podría ser la siguiente:

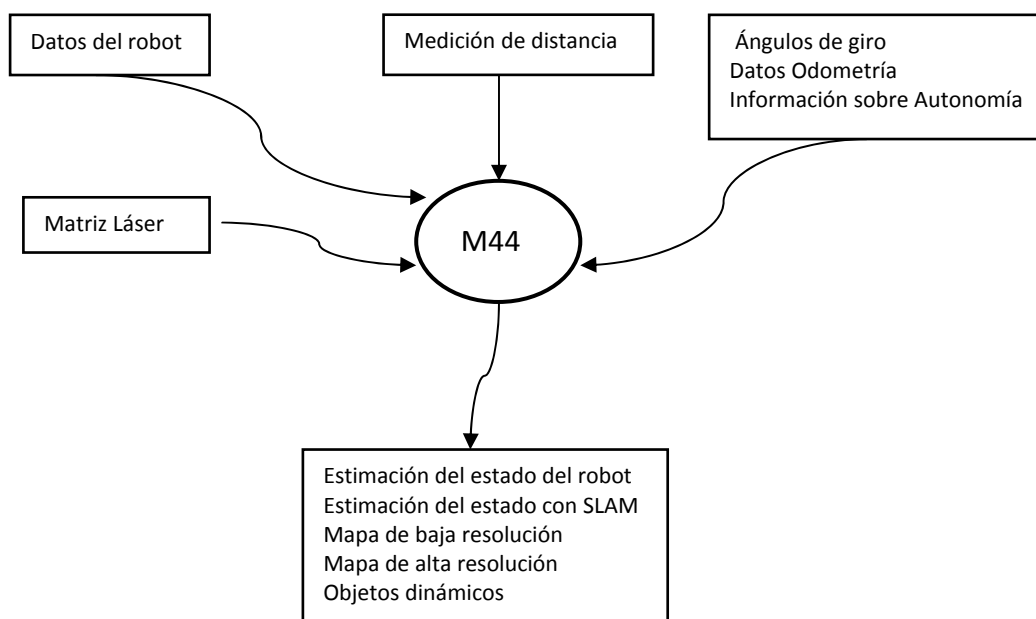


Figura 1. Esquema a alto nivel de las interfaces del proyecto NM-RS

1.3. OBJETIVOS DEL PROYECTO

Como análisis adicional a las interfaces ya mencionadas, es necesario comprender los niveles de fusión de información que se desarrollan para poder enmarcar el presente proyecto fin de carrera dentro de la globalidad del NM-RS. Los algoritmos utilizados reposan sobre un modelo de robótica probabilística donde se utiliza frecuentemente el filtro de Kalman como herramienta para poder estimar el estado final del robot a partir

de la estimación actual, las medidas observadas por los sensores y la entrada aplicada al sistema.

De esta forma, la parte principal de este proyecto utiliza como entradas información procedente de fusión de datos con filtro Kalman y a su vez proporcionara como salida otra información necesaria para una posterior fusión con otros filtro Kalman. Así, el presente proyecto es un intermediario entre dos bloques de robótica probabilística que la información de forma determinista.

De esta forma el proyecto se enmarca de la siguiente forma:

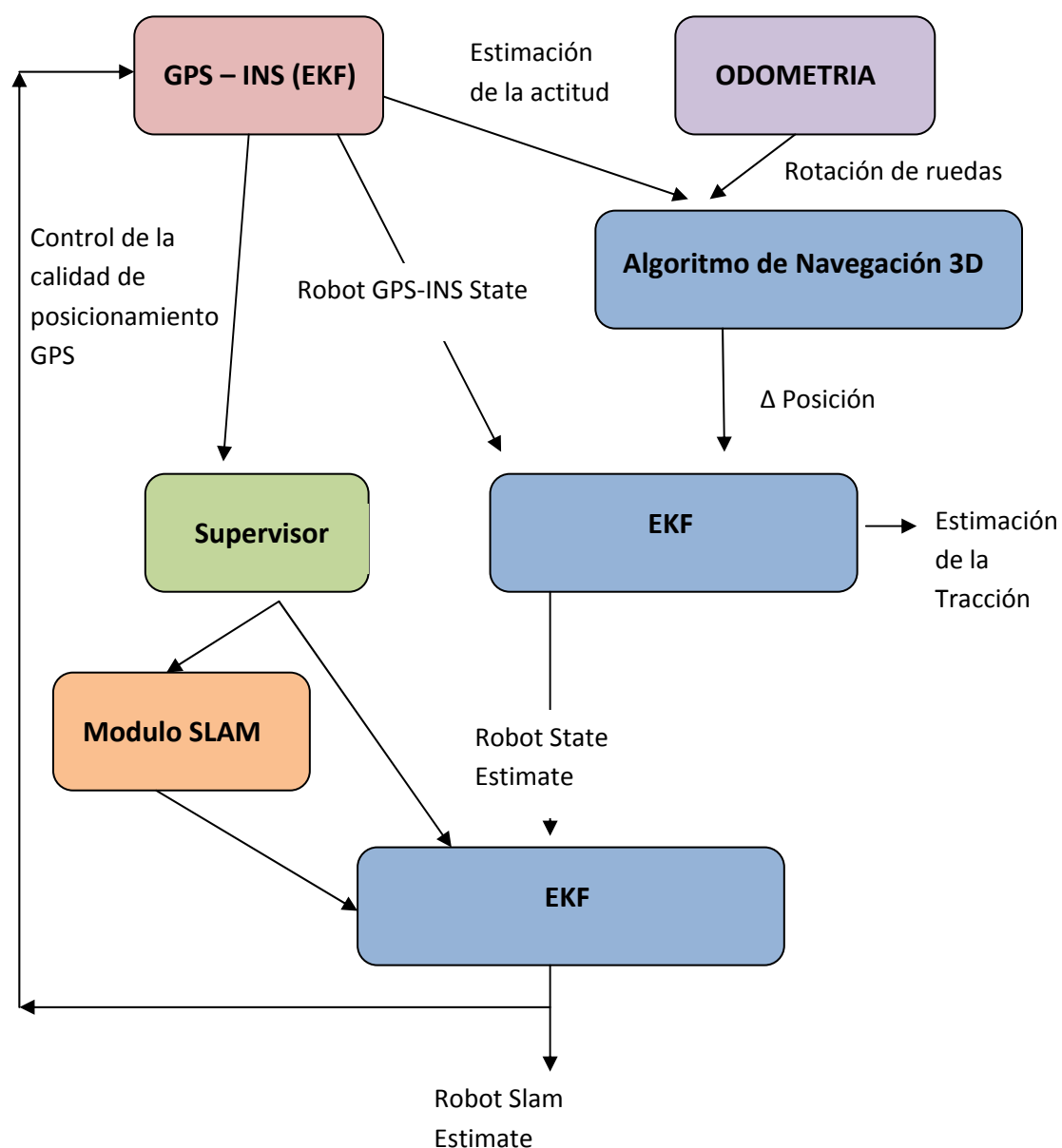


Figura 2. Integración de los diferentes elementos del sistema de posicionamiento

Una vez descrito el entorno de trabajo, se pasará a describir los objetivos del presente proyecto fin de carrera.

- Creación de un mapa local de alta resolución en modelo de celdas del terreno que incluye la siguiente información:
 - Modelo digital del terreno
 - Estimación de gradientes para determinar, atendiendo a los requisitos de cada robot, si la celda es transitable o no.
 - Nivel de comunicaciones y señal GPS en cada celda
 - Nivel de confianza sobre los valores almacenados en cada celda.
- Creación de un mapa general de baja resolución mediante la fusión de un mapa general recibido de un sistema central y la información que contiene el mapa local de alta resolución que elabora el propio robot.
- Desarrollo un algoritmo de SLAM (Simultaneous Location and Mapping) de forma que proporcione una estimación adicional de la posición del robot basado en la comparación de la información que tiene el robot del entorno (mapa local) y la información del láser. Esta tarea se desarrolla únicamente cuando la precisión de la estimación de posición que proviene del GPS es de muy baja calidad o nula. La estimación procedente del algoritmo de SLAM se incorpora como estimación a un al sistema de GNC (Guiado Navegación y Control) basado en sensores.

1.4. EL FILTRO DE KALMAN

La localización basada en el filtro de Kalman es la más extendida en la literatura y en implementaciones prácticas y debido a sus buenas propiedades, es apropiada para la mayoría de las aplicaciones siendo la que se ha utilizado para el desarrollo del presente proyecto.

El filtro de Kalman es un algoritmo recursivo óptimo para procesar información [3]. Combina la totalidad de la información disponible, ponderándola según su grado de incertidumbre, para realizar la estimación de las variables que definan el estado del sistema. El funcionamiento del filtro requiere el conocimiento de la dinámica del sistema, así como de los modelos estadísticos del ruido en las medidas de los sensores y de la incertidumbre inicial del modelo del sistema. Al tratarse de un

algoritmo recursivo, cada estimación se efectúa a partir de la anterior y de la nueva información disponible, sin que sea preciso almacenar todos los datos previos.

El filtro de Kalman permite minimizar el error en la estimación de las variables de interés cuando el modelo es lineal y la incertidumbre del sistema y de las medidas de los sensores es ruido blanco gaussiano. En esta situación, la función de densidad de probabilidad de cada variable a analizar condicionada a las medidas tomadas es tal que la media, la moda y la mediana coinciden, lo que evita cualquier posible conflicto a la hora de determinar cuál es la mejor estimación. Las hipótesis aceptadas pueden parecer altamente restrictivas pero hacen posible la resolución matemática del problema y se acercan bastante bien a la realidad en la mayoría de los casos. En otros sin embargo, han de contemplarse algunas variaciones y resulta de utilidad el llamado filtro extendido de Kalman (EKF) [4].

Para mayor información acerca de las ecuaciones y el algoritmo utilizado por el filtro Kalman consúltese el anexo IV dedicado al mismo. El algoritmo basado en el filtro de Kalman forma parte fundamental del sistema de posicionamiento del robot pero no ha sido objeto de desarrollo por el presente proyecto fin de carrera por lo que no se hará un comentario más extenso del mismo.

2. Estado del arte

2.1. *CONSTRUCCIÓN DE MAPAS DINÁMICOS*

2.1.1. Introducción

Una de las claves en robótica móvil es el conocimiento y la representación del entorno en que se mueve el robot, en concreto, representar la existencia de obstáculos con los cuales este puede chocar en su movimiento por el mundo.

Tradicionalmente, esa representación o mapa del entorno era introducida en el robot por el diseñador humano. El progreso hacia una mayor autonomía en los robots navegantes ha llevado a dotarlos de mecanismos que les permiten construir y mantener autónomamente esos mapas desde su propia información sensorial. La mayoría de los trabajos en representación del entorno para robots móviles dividen la información en dos partes que se tratan separadamente. Primeramente un mapa global refleja los obstáculos estáticos como paredes, armarios, etc. y permite planificación de largo plazo. En segundo lugar una representación instantánea, básicamente la última lectura sensorial, que permite reaccionar a obstáculos imprevistos. En esta exposición se comentaran varias técnicas de construcción de mapas con la idea de mantener una representación local al robot que capture el dinamismo de los objetos móviles a su alrededor. Dentro de los mapas métricos se puede distinguir entre dos tipos, el modelo de elementos geométricos y el de celdas de ocupación.

En el primero se dispone de unas primitivas de representación (puntos, esquinas, paredes, objetos, etc.) cuya posición se estima constantemente desde la información sensorial.

El segundo, que representa el espacio como un mallado regular de celdas cada una de las cuales contiene la creencia en que esa posición en el mundo esté ocupada o no. No necesita estructura en el entorno para conseguir una representación adecuada y facilita la fusión de datos sensoriales procedentes de sensores muy distintos. Adicionalmente, este segundo modelo representa explícitamente el espacio vacío, que resulta muy útil para la tarea de sortear obstáculos.

En la Figura 3, se puede observar un ejemplo de una rejilla alrededor del robot, las casillas oscuras indican la presencia de un obstáculo en ellas, y las claras de espacio vacío. Para comparar las distintas técnicas de construcción de mapas dinámicos se ha

utilizado la información proporcionada por los sensores sonar de un robot B21 comercial. Este tipo de sensor mide la distancia al objeto más próximo utilizando el eco de una onda ultrasónica y su empleo está muy difundido en la comunidad robótica. Las técnicas descritas también se han utilizado en un robot de exteriores casero, equipado con un sensor láser de proximidad, que proporciona medidas de ocupación mucho más fiables. En ambos casos las técnicas de construcción son exactamente las mismas, sólo se ha variado el modelo sensorial para reflejar las peculiaridades de cada sensor [5].

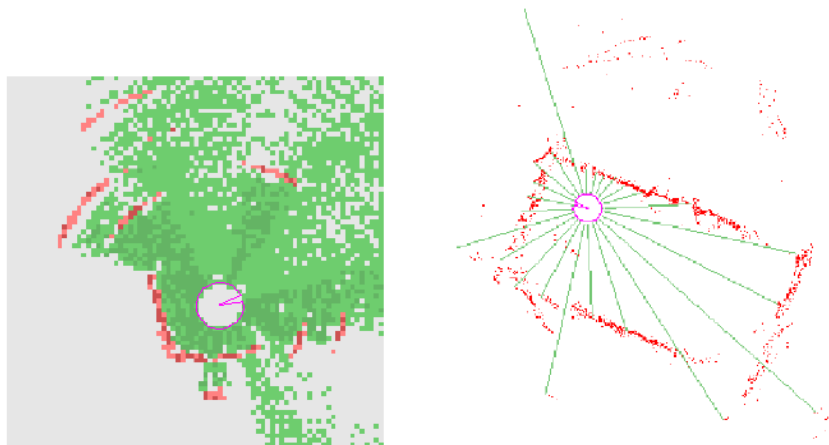


Figura 3. Ejemplo de grid construido con el robot B21 (izqda.) y lecturas sonar instantáneas (dcha.) Adaptada de José María Cañas y Lía García, 2002. [5]

2.1.2. Fusión de la información

Un robot autónomo percibe el estado de su entorno a través de sus sensores. La última lectura de todos sus sensores le proporciona una instantánea sobre el estado de sus alrededores, como muestra la Figura 3. Sin fusionar lecturas siempre se tendrá esta instantánea sensorial, continuamente refrescada. Debido a su simplicidad y a su vivacidad, esta representación ha sido utilizada en muchos casos para construir comportamientos reactivos sobre ella, por ejemplo el sorteo de obstáculos.

Se utiliza una rejilla para materializar la fusión de información procedente de múltiples lecturas sensoriales. Con una rejilla se puede recordar la información de zonas próximas o zonas que de repente quedan ocluidas por un obstáculo intermedio. La fusión también ayuda a depurar errores sensoriales y perfilar mejor el contorno de los obstáculos, compensando las lecturas erróneas con las medidas correctas,

previsiblemente más numerosas. Esta compensación es especialmente adecuada para los sensores sonar porque sus medidas son propensas a ruidos.

Finalmente, la fusión también permite la identificación de estímulos complejos, que no caben en una lectura instantánea. Permite acumular indicios, evidencias parciales. Esto resulta crucial cuando una instantánea sensorial por sí sola no es concluyente sobre la existencia de tal o cual estímulo. Un ejemplo podría ser el estímulo pared. La última instantánea sonar se ve afectada por la existencia de una pared próxima o no, pero desde esa instantánea es imposible distinguir si se trata de una pared o de cualquier otro obstáculo. Será la acumulación de varias lecturas, y el alineamiento de las celdillas ocupadas lo que permitirá concluir que el obstáculo concreto es una pared [5].

2.1.3. Escenarios y entorno

Normalmente se tendrá al robot midiendo continuamente el estado de su entorno a la vez que realiza sus maniobras. Se tendrá un flujo constante de lecturas sensoriales, no un conjunto finito y estático. La construcción en tiempo real de una representación razonable irá en paralelo a la toma de decisiones sobre esa misma representación.

Otro requisito importante es su carácter dinámico. El objetivo de los mapas del entorno que se generan es servir de base para todos los comportamientos del robot, incluido los reactivos. Por lo tanto deben reflejar con celeridad los cambios, tanto los nuevos obstáculos que aparecen como los nuevos huecos que surgen cuando los obstáculos se mueven.

Este escenario dinámico y en tiempo real es radicalmente distinto de la construcción de mapas estáticos, donde interesa reflejar los obstáculos fijos de determinado entorno y los mapas se pueden construir fuera de línea [5].

2.1.4. Enfoques y teorías

El problema de construcción y mantenimiento de celdas de ocupación ha sido ampliamente abordado en la literatura técnica específica. En general este problema se ha dividido en dos etapas. Primeramente se captura toda la información que proporciona una nueva lectura del sensor sobre la ocupación del espacio, siguiendo determinado modelo sensorial. En la segunda etapa esa información se utiliza para actualizar la creencia acumulada, materializando la fusión con otras medidas anteriores. La geometría de los modelos no se ha tenido en cuenta en la comparativa de esta exposición, que hace énfasis en el dinamismo de la regla de actualización.

En esta sección se ha agrupado los enfoques más representativos a la hora de representar la creencia de ocupación y de incorporar la información de nuevas

observaciones sensoriales: el modelo probabilístico bayesiano, la teoría de la evidencia, los conjuntos borrosos y el enfoque histográfico de Borenstein [5].

2.1.4.1. Enfoque probabilístico

El enfoque probabilístico es el más utilizado en la bibliografía. Se asume que cada celda del mapa puede tener únicamente uno de los dos estados: ocupada o vacía, que se tratará de estimar desde las observaciones sensoriales acumuladas. El conocimiento que el robot tiene en el instante t sobre la ocupación de la celdilla situada en (x, y) se refleja en la probabilidad de que la celdilla esté en alguno de los dos estados posibles condicionada a las observaciones que se han obtenido hasta ese momento. Así lo expresa la ecuación (1), donde $data(t - 1)$ supone el conjunto de observaciones acumuladas hasta el instante $t - 1$ y $obs(t)$ la observación actual. Cuando la probabilidad de ocupación es cercana a 0 entonces se está muy seguro que la celda está vacía. Por el contrario cuando es próxima a 1 entonces se tiene mucha confianza en que esa celda está ocupada. Inicialmente todas las casillas del grid tienen valor 0.5, reflejando el desconocimiento total.

En desarrollos probabilísticos recientes se parte del modelo sensor a posteriori, que marca la probabilidad de que la celdilla está ocupada o no dado tal o cual lectura del sonar $obs(t)$ es decir, $p(ocupada|obs(t))$.

Por ejemplo en [6] se utiliza un modelo sonar que vale $p(ocupada|obs(t)) = 0.4$ en las celdillas más cercanas al sensor que el radio observado y $p(ocupada|obs(t))=0.6$ en las celdillas más o menos coincidentes con ese radio. Para celdillas más distantes el modelo ofrece $p(ocupada|obs(t))=0.5$, que no aporta ninguna información en el enfoque probabilístico. Cuanto más se acerque a los extremos de probabilidad, 0 ó 1, más certidumbre aporta esa medida en un sentido u otro.

Actualización con regla de Bayes

A medida que el robot recibe nuevas observaciones sensoriales su información se va incorporando al mapa, actualizando las probabilidades almacenadas y haciéndolas evolucionar. Siguiendo el desarrollo de (1) y (2) se llega a la formulación incremental (3) de la regla de Bayes. Esta formulación maneja modelo a posteriori del sensor y ratios de probabilidad, definido como (2)

$p_{ocupada}(C_{(x,y)}, t) = p(ocupada/obs(t), data(t - 1))$	(1)
$p_{mapa} = p_{ocupada}/(1 - p_{ocupada})$	(2)

$p_{mapa}(C_{(x,y)}, t) = \frac{p_{obs}}{p_{a priori}} * p_{mapa}(C_{(x,y)}, t - 1)$	(3)
--	-----

Si una lectura sonar proporciona información sobre el estado de determinada celda $C_{(x,y)}$ el valor del modelo de sensor $p(ocupada|obs(t))$ en esa posición determina, a través de p_{obs} , si allí la probabilidad de ocupación sube o baja después de la nueva observación. El denominador $p_{a priori}$ simplemente normaliza la influencia de p_{obs} . Si $p(ocupada = obs(t)) = p(ocupada)$ entonces la observación no aporta ninguna información adicional sobre el conocimiento a priori y la probabilidad acumulada no cambia. Si $p(ocupada = obs(t)) > p(ocupada)$ entonces aumenta la probabilidad global en la ocupación de esa celdilla. Recíprocamente, la probabilidad acumulada disminuye cuando $p(ocupada = obs(t)) < p(ocupada)$.

Utilizar probabilidades permite tener un marco teórico fiable a la hora de realizar ciertas operaciones, cálculos e hipótesis con la información disponible. Otra ventaja sustancial es que (3) permite una formulación incremental, muy eficiente desde el punto de vista de tiempo y memoria requeridas en la actualización.

Uno de los inconvenientes de la actualización con regla de Bayes es que requiere que las distintas observaciones que se incorporan al mapa sean independientes, al menos en sentido markoviano. Esto no siempre se puede asegurar cuando se tiene un flujo continuo de mediciones. Otra desventaja es que no da medida alguna de confianza.

2.1.4.2. Teoría de la evidencia

La teoría de la evidencia se basa en la definición de *un campo de discernimiento* Θ , que es un conjunto de etiquetas que representan eventos mutuamente excluyentes. Tal y como se describe en [7], para la aplicación de mapas de ocupación las etiquetas interesantes son $\Theta = \{E, F\}$ porque las celdillas del grid pueden estar vacías, E , u ocupadas, F . Se define también una asignación básica de probabilidad como una función $m: \psi \rightarrow [0, 1]$ donde ψ es el conjunto de todos los subconjuntos posibles de Θ , en este caso $\psi = \{\emptyset, E, F, \{E, F\}\}$

El estado de cada celdilla se define asignando números de probabilidad a cada etiqueta en ψ , en este caso cuatro números. Sin embargo asumiendo $m_{mapa}(\emptyset) = 0$ y aplicando (4) basta almacenar dos de ellos, $m_{mapa}(E)$ y $m_{mapa}(F)$ para caracterizar el conocimiento sobre la ocupación de la celdilla en este enfoque.

El desconocimiento absoluto se refleja en $m_{mapa}(E) = 0$, $m_{mapa}(F) = 0$ y por lo tanto $m_{mapa}(E, F) = 1$. Cuando se esta seguro que una celdilla esta vacía entonces

$m_{mapa}(E) = 1$ y el resto se anula. Recíprocamente cuando se está seguro de que está ocupada $m_{mapa}(E) = 0$.

La geometría de sensor que emplea [7] es un cono de propagación. Para las celdillas dentro del arco el modelo viene dado por $(m_{sensor}(F) = \frac{1}{n}, m_{sensor}(E) = 0)$ donde n corresponde al número de celdas situadas en el arco. Para las celdillas en el interior del sector el modelo utilizado es $(m_{sensor}(F) = 0, m_{sensor}(E) = \rho)$.

Donde ρ es un factor constante de ajuste que iguala la masa total de evidencia asignada a las celdas vacías y a las ocupadas en cada lectura [5].

$$m_{celdilla(x,y)}(E) + m_{celdilla(x,y)}(F) + m_{celdilla(x,y)}(E, F) = 1 \quad (4)$$

Actualización con regla de Dempster-Shafer

La regla de Dempster-Shafer permite combinar evidencias sobre el evento A, $m_1(A)$ y $m_2(A)$, que este caso serían las asignaciones básicas de probabilidad acumuladas en cada celda del mapa para los eventos vacío E y ocupado F , y las proporcionadas por la última lectura sonar. Siguiendo el desarrollo se llega por ejemplo a (6).

$$m_{mapa}^t(E) = (m_{mapa}^{t-1} \oplus m_{sensor(t)})(E) \quad (5)$$

$$m_{mapa}^t(E) = \frac{m_{mapa}^{t-1}(E)m_{sensor(t)}(E) + m_{mapa}^{t-1}(E)m_{sensor(t)}(\{E, F\}) + m_{mapa}^{t-1}(\{E, F\})m_{sensor(t)}(E)}{1 - m_{mapa}^{t-1}(E)m_{sensor(t)}(F) - m_{mapa}^{t-1}(F)m_{sensor(t)}(E)} \quad (6)$$

Una ventaja de este enfoque es que contempla explícitamente la ambigüedad tanto en las medidas como en la creencia acumulada. El factor $m_{mapa(t)}(\{E, F\})$ representa la incertidumbre almacenada. También se representa la contradicción: una misma celdilla puede recoger a lo largo del tiempo tanto lecturas que indican que está ocupada $m_{mapa(t)}(E)$ como lecturas contradictorias que apuntan lo contrario $m_{mapa(t)}(F)$. Si queremos resumir la creencia en un único valor se necesita destilar esa creencia final convenientemente. En esa combinación irá implícitamente una compensación entre las evidencias de ocupación y vacío almacenadas en cada celdilla [5].

2.1.4.3. Enfoque borroso

En el enfoque borroso el mapa se almacena como dos conjuntos borrosos no complementarios: el de zonas vacías ε y el de zonas ocupadas o . Cada celda (x, y) del espacio pertenece en cierta medida a cada uno de los conjuntos y esa pertenencia es una función de pertenencia borrosa $\mu_e(x, y)$, $\mu_o(x, y)$.

La información de una lectura sonar k se captura igualmente con dos conjuntos borrosos ε^k y o^k que reflejan precisamente la evidencia de vacío y ocupación que aporta esa lectura k a las diferentes celdas del espacio. Por ejemplo en [5] se utilizan los modelos de la Figura 4.

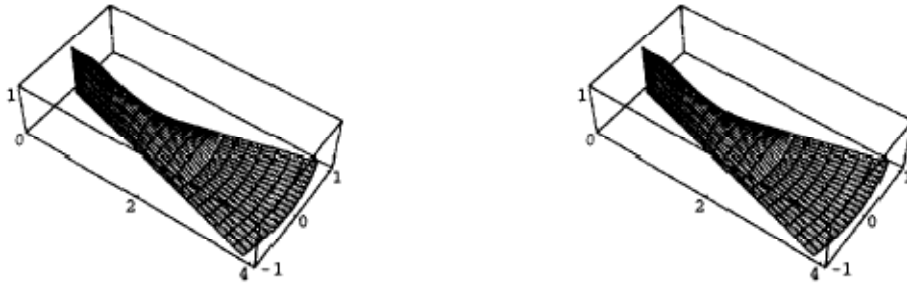


Figura 4. Modelo borroso sonar para información de ocupación (izquierda) y de vacío (derecha). Adaptada de José María Cañas y Lía García, 2002. [5]

Actualización con el operador borroso unión

Los conjuntos borrosos con las creencias globales se definen como la unión borrosa de las evidencias recogidas en cada lectura (7) (8). La operación de unión borrosa es asociativa, por ello estas ecuaciones (7) y (8) permiten una implementación incremental, eficiente desde el punto de vista práctico. En la formulación clásica [Poloni95] se han propuesto varios operadores de unión borrosa: producto algebraico (9), producto acotado (10), operador Dombi, operador Yager.

$$o = \bigcup_{i=1}^{i=k} o^i = (\bigcup_{i=1}^{i=k-1} o^i) \bigcup o^k \quad (4)$$

$$\varepsilon = \bigcup_{i=1}^{i=k} \varepsilon^i = (\bigcup_{i=1}^{i=k-1} \varepsilon^i) \bigcup \varepsilon^k \quad (5)$$

$(A \cup B)(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x)$	(6)
$(A \cup B)(x) = \min(1, \mu_A(x) + \mu_B(x))$	(7)

Una de las ventajas de la aproximación borrosa es que no necesita tantas asunciones teóricas como el desarrollo probabilístico y se tiene más libertad a la hora de diseñar el modelo sensorial y los operadores borrosos.

Las evidencias de ocupación y de vacío no son contradictorias en este enfoque. Precisamente por ello la aproximación borrosa exhibe una mayor robustez frente a medidas erróneas esporádicas que el enfoque probabilístico. Combinando los dos conjuntos borrosos globales se puede distinguir entre zonas ambiguas y zonas desconocidas, es decir, información contradictoria y ausencia de información [5].

2.1.4.4. Enfoque histográfico

El enfoque histográfico fue presentado por Johann Borenstein y Y. Koren. En él cada celda mantiene un valor de certidumbre CV indicando la confianza en la existencia de un obstáculo en esa posición, que se mueve entre $CV_{min} = 0$ y $CV_{max} = 15$. Para utilizar el mapa se suele binarizar la creencia de ocupación comparando el valor almacenado en cada celda con cierto umbral, por ejemplo 12. Sólo las casillas con evidencia superior se consideran realmente ocupadas.

Regla de actualización histográfica

La mezcla de información se hace empleando una regla aditiva heurística que suma el valor del modelo sensorial al acumulado en la celdilla.

$CV_{i,j}(t + 1) = CV_{i,j}(t) + \Delta(t)$	(8)
---	------

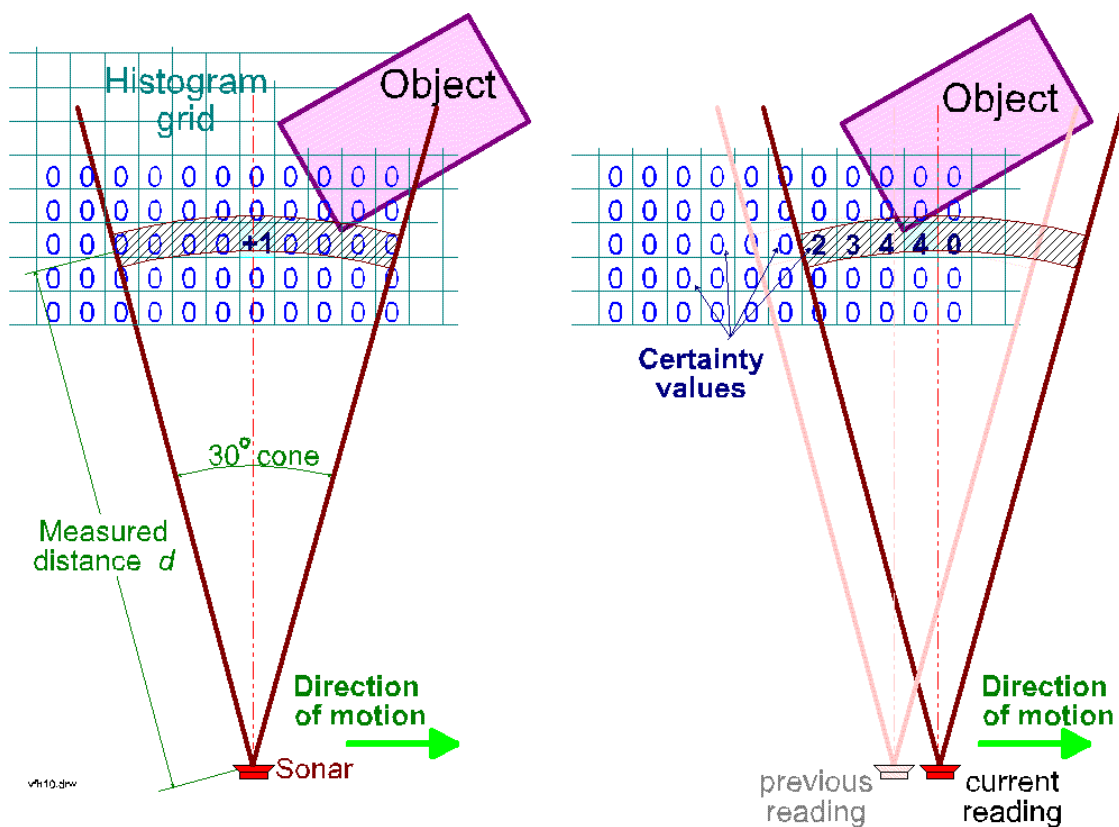


Figura 5. Modelos histográficos. Adaptado José María Cañas y Lía García, 2002. [5]

En el trabajo de Borenstein [9] sí hay un estudio explícito del carácter dinámico de la representación. La regla de actualización contempla la posibilidad de que la creencia pueda cambiar completamente de sentido con un número finito de observaciones sensoriales. Tantas veces como se necesite e independientemente de lo confiado que se estuviera en la creencia anterior. Se considera el número crítico de medidas necesarias para dar una creencia por firme. Ese valor marca la velocidad máxima de los obstáculos que puede reflejar el mapa tal y como está construido. Otra ventaja es que no necesita que las observaciones sensoriales sean independientes, se incorporan todas. Tampoco se hipotetiza como se distribuyen las medidas del sensor dada una configuración del mundo. Es la compensación entre unas y otras la que va conformando la distribución de probabilidad en el espacio.

De este enfoque derivan los desarrollos del presente proyecto, adaptándolo según las necesidades y simplificando determinados aspectos. Considerando la utilización de un laser en lugar de un cono ultrasónico.

2.1.5. Enfoques dinámicos

Ya se han comentado las aproximaciones más destacadas en la literatura técnica, ahora se describen dos procedimientos para mejorar el dinamismo en la representación que se consigue con los enfoques anteriores. El primer enfoque se

basa en el grid histográmico, pero reemplazando la regla de actualización aditiva por una ecuación diferencial. En la segunda aproximación cada celda lleva asociada una memoria de corto plazo, en la que se van almacenando las últimas medidas. Se decide el estado de la celda por mayoría sobre esa memoria.

El *estado de ocupación* de cada celdilla $C_{(x,y)}$ es una variable gradual, continua, que oscila entre un valor máximo E_{max} para señalar ocupación y un valor mínimo $E_{min} = -E_{max}$ para señalar certeza de vacío.

En ambos procedimientos se utiliza un modelo lobular que asigna un peso positivo $\Delta(t) = +1$ a las celdas situadas en el arco de ocupación y un peso negativo $\Delta(t) = -1$ a las situadas en el interior del lóbulo les asigna un valor negativo para indicar evidencia de vacío. Cuanto mayor es el valor absoluto mayor es la influencia de la medida en esa celdilla [5].

2.1.5.1. Actualización con ecuación diferencial

En este enfoque se actualiza el estado de ocupación de cada celdilla siguiendo la ecuación diferencial (12). El cambio en la creencia de ocupación seria un incremento o decremento dependiendo del signo de $\Delta(t)$. La amplitud del cambio depende de varios factores.

$ \begin{aligned} creencia(C_{(x,y)}, t) \\ &= creencia(C_{(x,y)}, t - 1) + \Delta(t) * saturacion(t) \\ &\quad * secuencia(t) * speed \end{aligned} $	(9)
$ saturacion = \begin{cases} \Delta(t) > 0 \mid E_{max} - creencia(x, y, t - 1) \mid \\ \Delta(t) < 0 \mid E_{min} - creencia(x, y, t - 1) \mid \end{cases} $	(10)

El *factor saturación* (13), acota el valor del incremento de tal modo que nunca se pasen los valores máximo y mínimo para la creencia. Este factor hace que afectando varias medidas con el mismo peso a la celda, las novedosas tengan más influencia, provoquen mas cambio en la creencia. Esto permite en la práctica cambios de opinión muy rápidos, para reflejar el posible movimiento de obstáculos. Además cuando se

está muy seguro de la ocupación de la celdilla, nuevas observaciones en este sentido apenas aportan información.

El parámetro *speed*, entre 0 y 1, constante para todo el grid, modula la velocidad de cambio de estado. Con el se parametriza el número de medidas necesarias para cambiar totalmente de creencia, menor cuanto mayor es *speed*.

Finalmente el *factor secuencia*, entre 0 y 1, refuerza el efecto de las medidas cuando estas aparecen seguidas, de tal manera que las medidas aisladas quedan aminoradas. Su valor se encuentra entre 0 y 1, y se calcula sobre una pequeña memoria de evidencias asociada a la celdilla. Si la última evidencia aparece en una secuencia de evidencias del mismo signo su influencia será mayor que si las anteriores son de signo contrario. En cierto modo este factor retarda el efecto de las sorpresas hasta que se van confirmando con una secuencia de lecturas en el mismo sentido.

La ecuación diferencial propuesta, por su propia naturaleza ofrece un alto dinamismo en la creencia y logra que las medidas recientes influyan en el estado de ocupación sistemáticamente más que las antiguas. Adicionalmente se ha incluido un *mecanismo de olvido* que periódicamente (1 segundo) multiplica la creencia de todas las celdillas por un *factor olvido* = 0.98. Este mecanismo acerca iterativamente la creencia de ocupación de todas las celdillas al estado de desconocimiento, creencia $(C_{(x,y)}, t) \simeq 0$ y fuerza a que estas se refresquen constantemente con nuevas observaciones.

2.1.5.2. Decisión por mayoría

En cada celdilla $C_{(x,y)}$ se almacena, con orden temporal, la información que aportan las últimas N medidas que afectan a la celdilla: $\Delta(t - 1)$, $\Delta(t - 2)$, ... $\Delta(t - N)$. Sumando los valores en memoria se tiene el peso acumulado. El peso acumulado $\sum_{i=1}^N peso(i)$ oscila entre $-N$ y $+N$. Con este peso acumulado se estima la ocupación de la celdilla siguiendo la función de la figura siguiente.

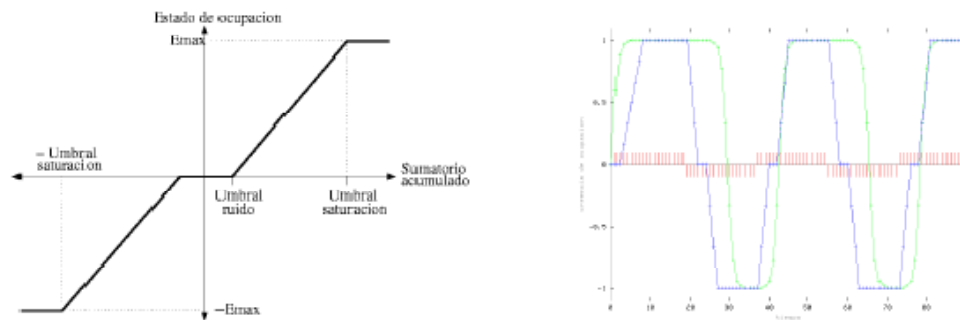


Figura 6. Estado de ocupación en función de los pesos almacenados en la memoria de la celdilla (izda.). Latencia del enfoque por mayoría (dcha.). Adaptada de José María Cañas y Lía García, 2002. [5]

En el eje vertical se ve el estado de ocupación final y en las abscisas el peso acumulado en la memoria de la celdilla. El umbral de ruido de la Figura 6 señala la cantidad mínima de evidencias necesarias para empezar a creer que la celdilla no esté libre. Este umbral inferior filtra las medidas erróneas espúreas, pues se necesita más de una medida para confirmar que la celdilla puede estar ocupada. Las celdillas realmente ocupadas o vacías lo superan sin problemas acumulando enseguida evidencias que lo respaldan.

A medida que se acumulan evidencias por encima de ese umbral, el estado de ocupación crece linealmente hasta alcanzar el valor máximo en el umbral de saturación. Este umbral superior introduce el fenómeno de saturación en la creencia. Esta saturación ecualiza las zonas del espacio por donde más y menos tiempo se ha movido el robot, con tal que las evidencias recogidas sean suficientes para concluir un estado u otro de la celdilla.

Sobre una misma celdilla pueden caer varias evidencias contradictorias, de ocupación y de vacío. Con este enfoque se observa claramente que unas lecturas compensan a otras. La idea aquí es que una lectura errónea no sesga la creencia frente a una mayoría de lecturas correctas. Para modular esa compensación el enfoque permite asignar distintos pesos a evidencias de ocupación, lejanas, cercanas, de vacío, etc.

Con este enfoque la inserción de observaciones nuevas refresca el contenido de la memoria local, de esta manera la creencia de ocupación está siempre actualizada. Adicionalmente se ha añadido un mecanismo artificial de olvido que periódicamente inserta observaciones neutras para eliminar observaciones antiguas. Gracias a esto la creencia acaba envejeciendo en ausencia de lecturas recientes [5].

2.1.6. Criterio de vivacidad VS. Criterio de robustez

En la literatura existen muchas comparativas pero pocas veces se ha evaluado explícitamente el comportamiento dinámico del algoritmo de construcción automática de mapas. Los enfoques más utilizados tienen un carácter estático subyacente en el cual no importa demasiado la velocidad en adquirir determinada creencia de ocupación, más bien su corrección. La compensación entre medidas persigue principalmente corregir algunas incertidumbres relativas al sensor.

En el caso de mapas dinámicos el estado de ocupación real puede cambiar con el tiempo, por lo tanto la regla de actualización además debe buscar compensar las lecturas antiguas con las recientes. Se desea que la creencia cambie rápidamente si las lecturas nuevas apuntan un cambio en el estado de ocupación actual, para reflejar con vivacidad los movimientos de los obstáculos. Se definen dos ratios que caracterizan el comportamiento dinámico de los algoritmos constructores de representación: *tiempo en incorporar obstáculo* (TIO) y *tiempo en incorporar hueco* (TIH), que miden precisamente el número de medidas necesarias para que la creencia de ocupación confirme la ocupación o el vacío. Para medirlos utilizaremos unas *secuencias de prueba* que corresponden a observaciones que atañen a una misma celdilla.

En contraposición al criterio de vivacidad tenemos el de robustez frente a lecturas inciertas. Esa compensación necesita cierta latencia para implementarse y una sola medida no modifica significativamente la creencia hasta que se confirma con nuevas observaciones. Entre estos dos criterios contrapuestos el algoritmo elegido establece un compromiso.

2.1.7. Conclusiones

Se han comentado las técnicas más populares de construcción y mantenimiento de mapas métricos en forma de rejilla. El enfoque probabilístico, la teoría de evidencia y el enfoque borroso resultan inválidos para representar características que puedan cambiar con el tiempo (por ejemplo la ocupación del espacio cuando hay obstáculos móviles). La principal razón es que tanto la regla de Bayes, la regla de Dempster-Shafer y el operador borroso de unión exhiben la *propiedad asociativa*: dada una secuencia de lecturas sensoriales, el estado final de las celdillas del grid es el mismo con independencia del orden en que se incorporen esas lecturas

El enfoque probabilístico bayesiano muestra una inercia proporcional a las evidencias acumuladas, lo que ralentiza en exceso su cambio de creencia. En general necesita tantas evidencias de ocupación como de desocupación para cambiar el sentido de su estimación. En la práctica ofrece un mayor dinamismo debido a una limitación práctica

que obliga a manejar valores de probabilidad en el intervalo $[\delta, 1 - \delta]$. Esta restricción desvirtúa todas las asunciones de probabilidad hechas y no forma parte explícita del formalismo bayesiano.

La teoría de la evidencia deriva en los mismos resultados que la probabilística una vez que se incorpora un reducido número de lecturas iniciales. Después de esas medidas la ambigüedad en la estimación $m_{mapa}(E, F)$ se anula y la evolución de ambos enfoques es similar. Las mismas restricciones se aplican.

El enfoque borroso clásico presenta un bloqueo inaceptable tras incorporar un pequeño número de lecturas sensoriales. No funciona bien si se tiene un flujo continuo de medidas. La razón de este bloqueo radica que el operador borroso de unión sea una función monótona creciente. Existen nuevas teorías dentro de este enfoque que proponen nuevos operadores borrosos que superan este bloqueo y lo acercan a una sencilla media aritmética.

Por el contrario el enfoque histográfico, de decisión por mayoría y el basado en ecuación diferencial sí refleja el dinamismo de la realidad. Todos ellos distinguen entre *secuencia_A* y *secuencia_B*, el estado final es distinto en ambos casos. En estos enfoques por muy seguro que se esté que tal celdilla está ocupada, basta un número relevante de lecturas en sentido contrario para cambiar radicalmente de creencia. Este dinamismo de representación es imprescindible para representar obstáculos móviles y resulta útil incluso con obstáculos estáticos si se arrastran errores de localización. El uso de técnicas estáticas obliga a mantener una localización absoluta precisa, para no mezclar evidencias de distintas celdillas.

El perfeccionamiento de los procedimientos descritos debe pasar por segmentar el grid dinámico de ocupación y a utilizar las técnicas dinámicas para detectar puertas sin localización absoluta fiable [5].

2.2. Localización y mapeado simultáneo

2.2.1. Introducción

Un robot móvil debe conocer donde se encuentra dentro de un entorno para poder navegar de forma autónoma e inteligente. La auto-localización y el conocimiento del emplazamiento de otros objetos requiere la existencia de un mapa y este requerimiento básico ha propiciado el desarrollo de los algoritmos de localización y mapeo simultáneos “SLAM” (Simultaneous Location and Mapping) durante las dos décadas pasadas, en los que el robot construye un mapa mientras explora el entorno.

La forma predominante de SLAM hasta día de hoy es SLAM estocástico introducido por Smith [10]. El SLAM estocástico tiene en cuenta de forma explícita los errores

introducidos por las medidas tomadas por los sensores: los errores en las medidas introducen incertidumbres en la estimación del emplazamiento de marcas, las cuales se tornan en incertidumbre en la localización del robot, por ello la estimación de la posición de las marcas y del robot son dependientes.

La mayoría de implementaciones prácticas del SLAM representan dichas incertidumbres y correlaciones mediante la función de densidad de probabilidad gaussiana (PDF), y propagan dichas incertidumbres mediante el filtro de Kalman extendido (EKF). Esta forma de SLAM es conocido como EKF-SLAM. Un problema con el EKF-SLAM es que requiere modelos de marcas basados en formas geométricas para tener en cuenta los datos medidos lo que limita las aproximaciones a entornos en los que dichos modelos no sean adecuados [11].

2.2.2. Tipos fundamentales de SLAM

Un tema importante dentro del EKF-SLAM es el diseño del modelo de observación. Las implementaciones actuales requieren como se ha dicho la utilización de formas geométricas para modelar la observación, tales como líneas, círculos, splines. Las medidas deben encajar en alguna de las categorías geométricas disponibles para ser clasificadas como un punto característico y los datos que no encajen con ninguna son desechados. El principal problema, es que tienden a ser específicos del entorno por el que se desplace el robot por ello un modelo que se comporte bien en un determinado entorno puede no funcionar en otro y desperdiciar una gran cantidad de información.

Una alternativa a los modelos de características analíticas es el procedimiento llamado “scan correlation”, basado en calcular la máxima probabilidad de alineamiento entre dos medidas en bruto de los sensores.

Dado una serie de medidas observadas, escáneres realizados por el robot o mapas locales y un mapa de referencia es posible que el robot se localice sin convertir sus medidas a ningún tipo de forma geométrica. Las observaciones son sencillamente alineadas con los datos del mapa de forma que se maximice las correlaciones de las medidas. El método scan correlation ha sido ampliamente usado cuando se posee un mapa a priori del entorno mediante el algoritmo iterated closet point (ICP) mediante rejillas de ocupación siendo el más popular de los métodos usados. Este tipo de algoritmos están basados en un método iterativo donde se calcula en primer lugar las correspondencias entre escaneos y posteriormente se trata de minimizar el error en la distancia para calcular el desplazamiento del sensor. Este proceso es repetido con una nueva estimación hasta la convergencia. Una característica común a la mayoría de versiones de ICP es el uso de la distancia euclídea para establecer correspondencias y aplicar mínimos cuadrados. Sin embargo, esta distancia no toma en cuenta el hecho de que los puntos lejanos al sensor pudieran encontrarse lejos

debido a rotaciones en el propio sensor. Para superar esto se ha propuesto el cálculo de dos tipos de correspondencias, una es la distancia euclídea y otra la distancia angular (para capturar la rotación del sensor). La ganancia en precisión es perdida en complejidad y convergencia ya que estos métodos realizan dos procesos de emparejado o “matching” y minimización para cada iteración. Se entiende que estos son los dos principales problemas para los algoritmos ICP: encontrar una forma adecuada de medir la cercanía y aplicar la minimización.

Dos métodos importantes han sido presentados para realizar el SLAM mediante scan correlation. El primero, usa la maximización de la esperanza (EM) para maximizar la correlación entre escaneos lo que resulta en un set de estimaciones sobre la posición del robot que producen un alineamiento óptimo entre todos los escaneos. El segundo método, llamado estimación consistente de la posición (CPE), acumula una historia seleccionada de escaneos y alineamientos formando una red.

El principal inconveniente con los métodos existentes para SLAM aplicando scan correlation es que no realizan la fusión de datos, y en lugar de ello se requiere de una serie de escaneos almacenada y que no son compatibles con la forma tradicional de la formulación de EKF-SLAM.

Otro método bastante novedoso denominado Scan-SLAM que intenta solventar las deficiencias de ambos métodos siendo idéntico al convencional EKF-SLAM excepto en la definición del modelo de marcas, que está comenzando a ser usado. En este modelo las marcas son definidos mediante una plantilla con los datos del sensor en bruto: los observados mediante un proceso de scan-matching. Este proceso da lugar a un modelo genérico de observación basado en la localización de un sistema de coordenadas local embebido en cada plantilla de marcas. Las plantillas también facilitan una estrategia de asociación de datos [11].

3. Tipos y estructuras de los mapas usados. Datos de partida.

De todas las formas comentadas en el estado del arte, en este proyecto se impuso la utilización de mapas de celdas con varios niveles. Se basan en dividir el terreno cuya información se quiere representar en celdas. Estas celdas tendrán asociado un valor numérico dependiendo de la información que representen. De esta forma cada celda representa un área de la superficie y se asigna un único valor numérico para toda la superficie. Por este motivo son consideradas con dos dimensiones y media, donde no se pueden representar objetos como túneles, puentes u objetos con voladizos.

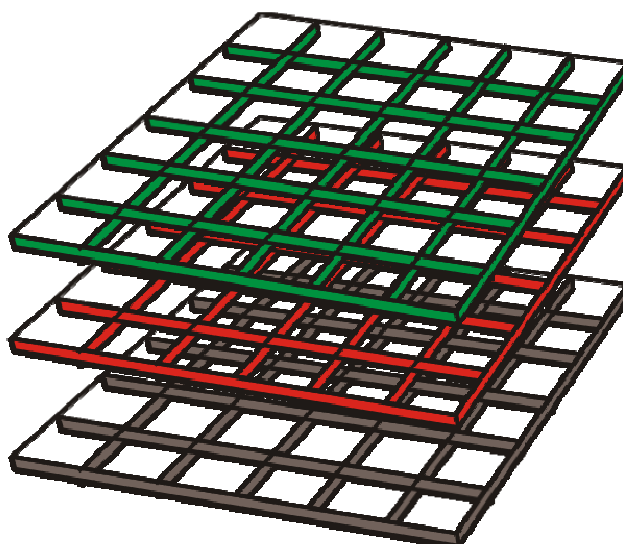


Figura 7. Mapa de celdas

Un mapa de celdas a varios niveles es una estructura de varias capas donde cada capa almacena un tipo distinto de información.

Esta estructura a varios niveles es parametrizable, permitiendo almacenar la información definiendo un tamaño global del entorno y un tamaño de la celda. Dependiendo de la extensión de terreno que se necesite abarcar se podrá definir el tamaño de la celda como se desee teniendo en cuenta que cuanto más grande sea ésta, menor precisión presenta.

3.1. DESCRIPCIÓN GENERAL DE LA ESTRUCTURA MAPA.

Como se ha justificado anteriormente, para almacenar diferentes tipos de información se ha definido una estructura de mapa de celdas en diferentes niveles.

Lógicamente, la información procesada y generada a medida que avanza la reconstrucción, y tanto para operaciones y cálculos intermedios como para la presentación final de los datos se utilizarán estructuras formadas por distintos tipos de datos, como números enteros, números en coma flotante o incluso estructuras.

Como requisito del proyecto se requiere la utilización de mapas discretos formados por celdas de tamaño determinado por las condiciones de cada ensayo que se desee realizar. Para tomar una idea del orden de magnitud de las dimensiones de cada celda, podremos tomar de 0.5 a 1.5 m las dimensiones del lado para las celdas de los mapas de alta resolución y de 2 a 10 m las dimensiones del lado de una celda de los mapas de baja resolución.

Todas las celdas de un mismo mapa presentan la misma dimensión. No se utilizan celdas irregulares ni una división en función del área de interés.

Tal y como se ha mencionado, existen fundamentalmente dos tipos de mapas, de alta resolución y de baja resolución. La única diferencia reside en el espacio físico real que ambos pueden representar. En el mapa de baja resolución se representa en principio la información correspondiente a todo el entorno de simulación del robot, mientras que en el mapa de alta se representa la información detallada de una zona del mapa cercana en distancia al robot.

Se han definido tres tipos de mapas:

- Low Resolution Map Global: mapa global inicial en baja resolución. Representa el conocimiento previo que se tiene del terreno y es la información de partida del robot. Según se va realizando la reconstrucción, este mapa aglutina las reconstrucciones parciales realizadas por los diferentes robots.
- Low Resolution Map: es una copia local (en cada robot) del anterior, que realiza el robot al inicio de su operación. Todas las actualizaciones que este haga sobre el terreno se realizan en el LowResMap.
- High Resolution Map: es el mapa local en alta resolución.

La estructura de forma general para ambos tipos de mapas es la que sigue a continuación:

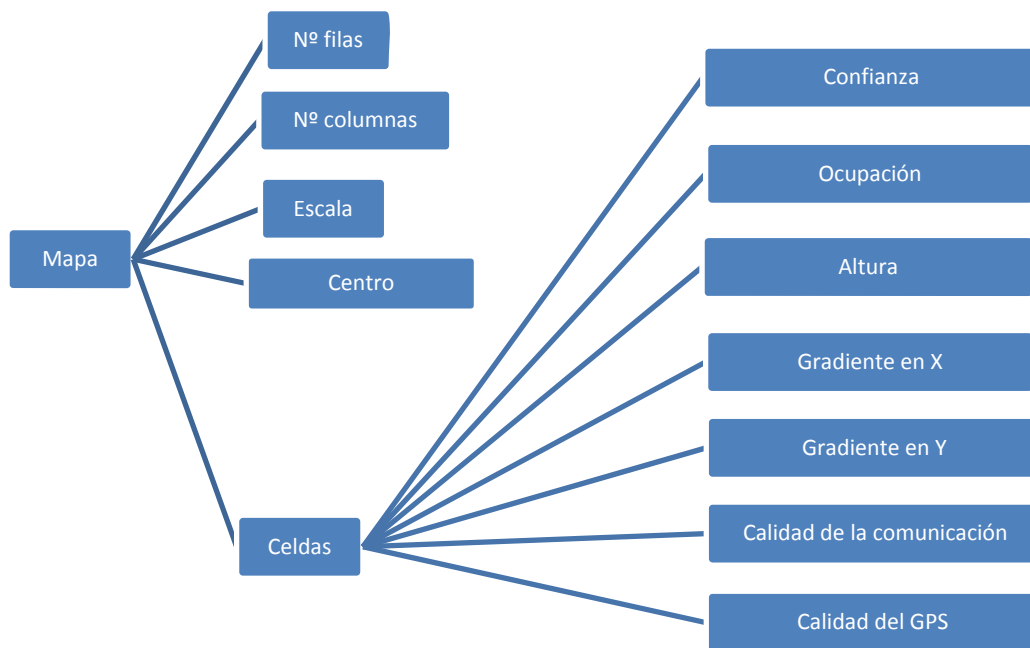


Figura 8. Estructura general de los mapas

A continuación se estudiara detenidamente la estructura de los mencionados mapas:

- Numero de filas y columnas: describe la dimensión de la matriz de celdas.
- Escala: tamaño en metros de la celda (son cuadradas)
- Centro: coordenadas UTM (*Universal Transverse Mercator*) del centro del mapa. Como punto de referencia de una celda se considera la esquina inferior izquierda de ésta. Para determinar la celda central, es necesario hacer distinción para los casos en los que el número de filas o columnas sea par o impar, para ello:

Siendo (i_c, j_c) los índices de la celda central del mapa, y llamando S_{Nm1} y S_{Nm2} al número de filas y columnas del mapa, se pueden presentar los siguientes casos:

Que S_{Nm1} sea un número par. La celda central estará en la fila $i_c = \frac{S_{Nm1}}{2} + 1$.

Que S_{Nm1} sea impar. La celda central estará en la fila $i_c = \frac{S_{Nm1}+1}{2}$.

Así tenemos los siguientes ejemplos:

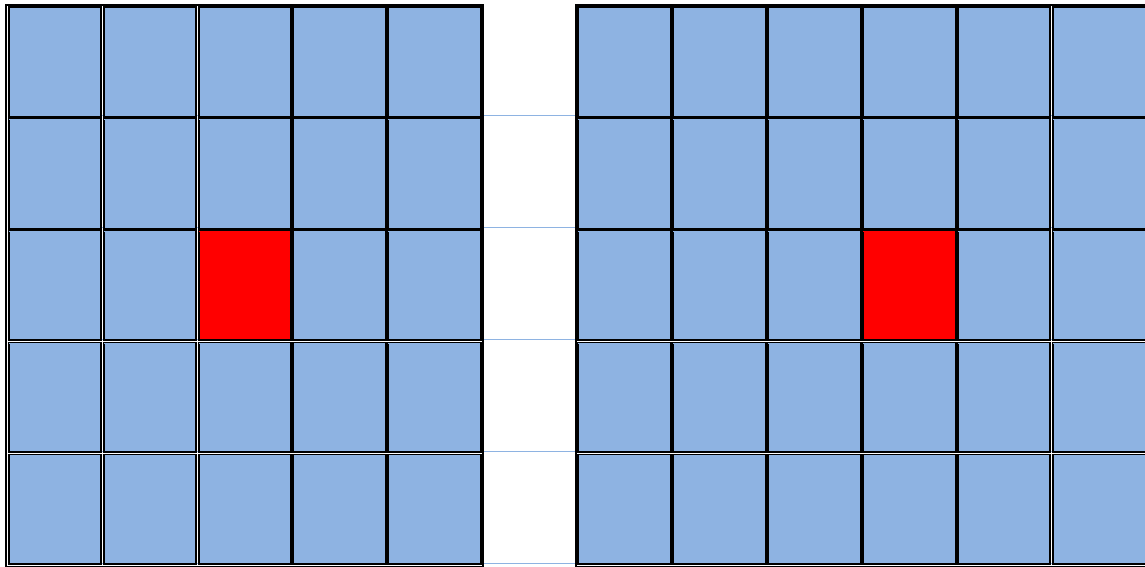


Figura 9. Ejemplos de situación del centro del mapa

Como ejemplo para ilustrar la idea se puede observar a continuación una matriz de 6 filas y 4 columnas con el centro situado en la celda (3,3).

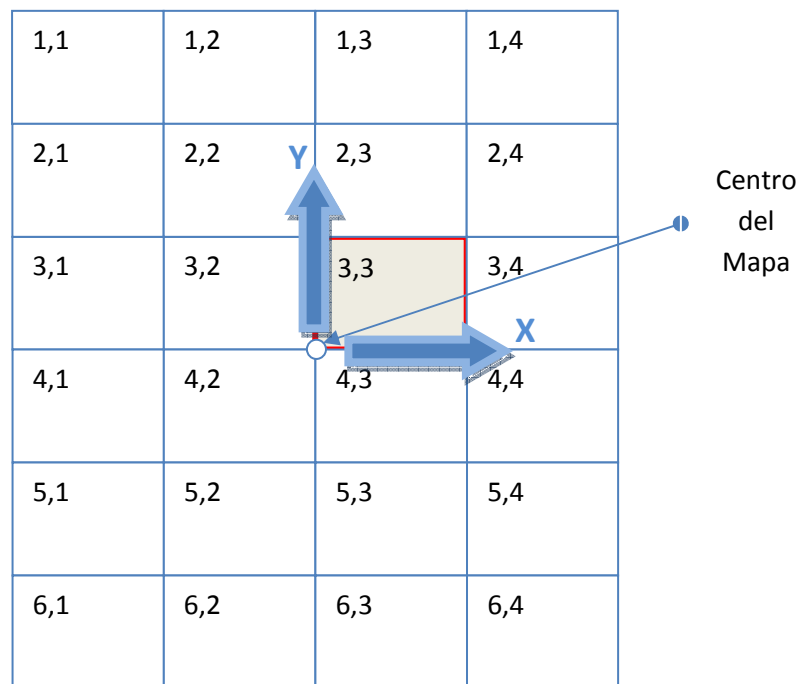


Figura 10. Sistemas de referencia asociados al mapa

- Capas: en los siguientes apartados se hará una extensa descripción de la funcionalidad de cada capa.

3.2. CONVERSIÓN DE COORDENADAS A CELDA EN EL MAPA

De forma general, y sin tener en cuenta el tipo de mapa, se utiliza un sencillo algoritmo para hacer corresponder a cada punto en su correspondiente celda del mapa. Aunque este proceso será detallado en los apartados dedicados a los algoritmos, para la correcta comprensión del funcionamiento de los mapas es necesario saber algún detalle sobre este proceso.

Considerando un punto del mapa (x_p, y_p) y un mapa definido por el centro cuya posición es (x_o, y_o) , y el tamaño de la celda definido por el parámetro “escala”, las distancias al centro vendrían definidas de la siguiente forma:

$\Delta x = x_p - x_o$	(11)
$\Delta y = y_p - y_o$	(12)

Los índices de la matriz de celdas “i, j” correspondiente a las distancias $(\Delta x, \Delta y)$ son:

$j = X_o + \text{int}\left(\frac{\Delta x}{\text{escala}}\right) \quad \forall \Delta x > 0$	(13)
$j = X_o + \text{int}\left(\frac{\Delta x}{\text{escala}} - 1\right) \quad \forall \Delta x < 0$	(14)
$i = Y_o + \text{int}\left(\frac{\Delta y}{\text{escala}}\right) \quad \forall \Delta y > 0$	(15)
$i = Y_o + \text{int}\left(\frac{\Delta y}{\text{escala}} + 1\right) \quad \forall \Delta y < 0$	(16)

3.3. ESTRUCTURA MULTICAPA DE CELDAS

A continuación se describen la información correspondiente a las diferentes capas del mapa de celdas.

Es importante considerar que el algoritmo que se ha desarrollado puede funcionar de forma robusta, partiendo de una información inicial imprecisa o incompleta o bien sin contar con ningún tipo de información de partida.

3.3.1. Altura

Uno de los objetivos principales de la reconstrucción del terreno es obtener una representación lo más real posible del terreno y de los obstáculos presentes en él. Al tratarse de un sistema multi-robot, todos ellos colaboran de una forma descentralizada o centralizada en la composición de un mapa global que incluye la fusión de la información que proviene de todos los robots participantes en la simulación (que han tenido capacidad de comunicarse con los demás o con el puesto central).

Por ello, cada robot se ocupa de actualizar el área del mapa correspondiente a la zona por la que transita, compartiendo las actualizaciones realizadas para un conocimiento común de todos los robots.

La reconstrucción de las elevaciones y depresiones del terreno es parte fundamental del proyecto, siendo datos de partida para el cálculo posterior de los gradientes y con ellos el nivel de ocupación o de accesibilidad del terreno. No se hace ninguna distinción entre terreno y otros obstáculos presentes como edificios, estructuras, vehículos, otros robots, etc.

Cada celda del mapa tiene asociado un valor numérico de la altura, que corresponderá básicamente con la altura media de todos los puntos observados en el interior de ella, ponderando dicha información con la estimación de la precisión en la observación de dicho punto de colisión.

Como ejemplo se puede observar el mapa que viene a continuación, resultado de la reconstrucción de un pequeño número de celdas en alta resolución:

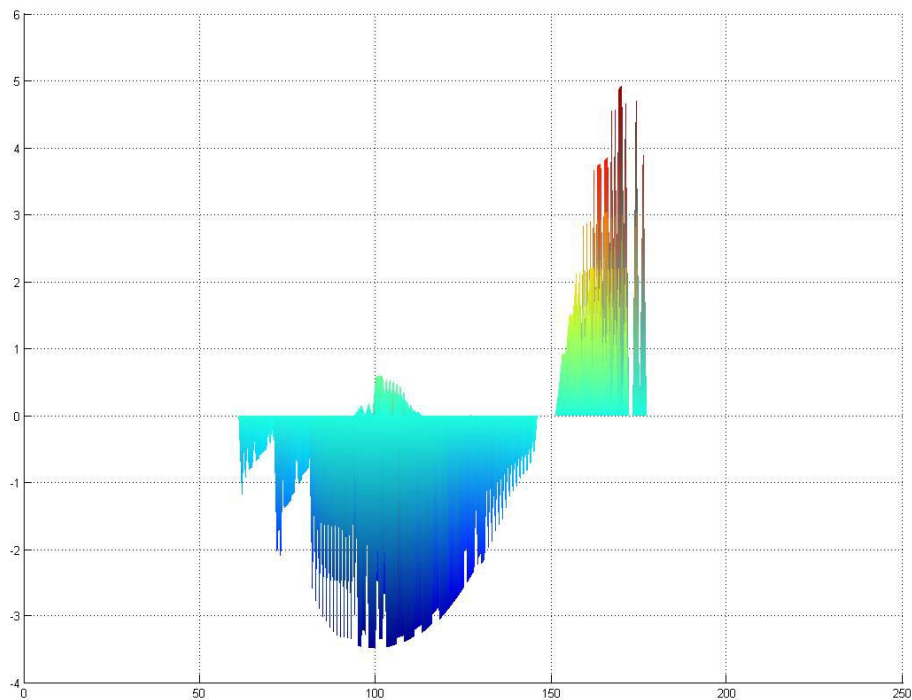


Figura 11. Reconstrucción de la altura de un terreno

En principio, el tipo de dato almacenado en cada celda es un número en coma flotante, de tal forma que es válido para valores reales de las alturas. Estas alturas son calculadas siempre en su valor absoluto, utilizando para ello el sistema fijo de referencia, es decir, tanto los mapas de alta resolución locales del robot como los mapas de baja resolución global tienen la misma referencia en cuanto a orientación y origen.

3.3.2. Gradientes

Las capas de gradientes tienen como objeto almacenar la estimación de las pendientes del terreno para cada celda según orientaciones de los ejes principales X e Y. Los gradientes son calculados siempre respecto a ejes fijos. Así, el gradiente según X mide la variación de la altura según avanzamos en el sentido marcado por el eje X fijo. En esta capa se almacena el valor de la tangente del ángulo que forma la superficie con la horizontal.

Es importante hacer notar que el gradiente es calculado utilizando para ello las alturas de los puntos de colisión que se corresponden a la misma celda, es decir, no se realiza un gradiente entre celdas adyacentes, sino en el interior de la misma.

Aunque en general los resultados no deben variar demasiado, se ha elegido esta forma de actuar por ser más precisa ya que representa el gradiente justo en la celda en la que se produce el cambio de altura, y si se hiciera un gradiente entre celdas, este se encontraría algo desplazado respecto de la variación de altura producida.

Como ejemplo de la capa de gradientes se puede observar la siguiente figura que ilustra los gradientes teóricos de un terreno en un mapa de baja resolución. Estos gradientes han sido calculados por el método de gradientes por filas y columnas que ya se explicará en el apartado correspondiente.

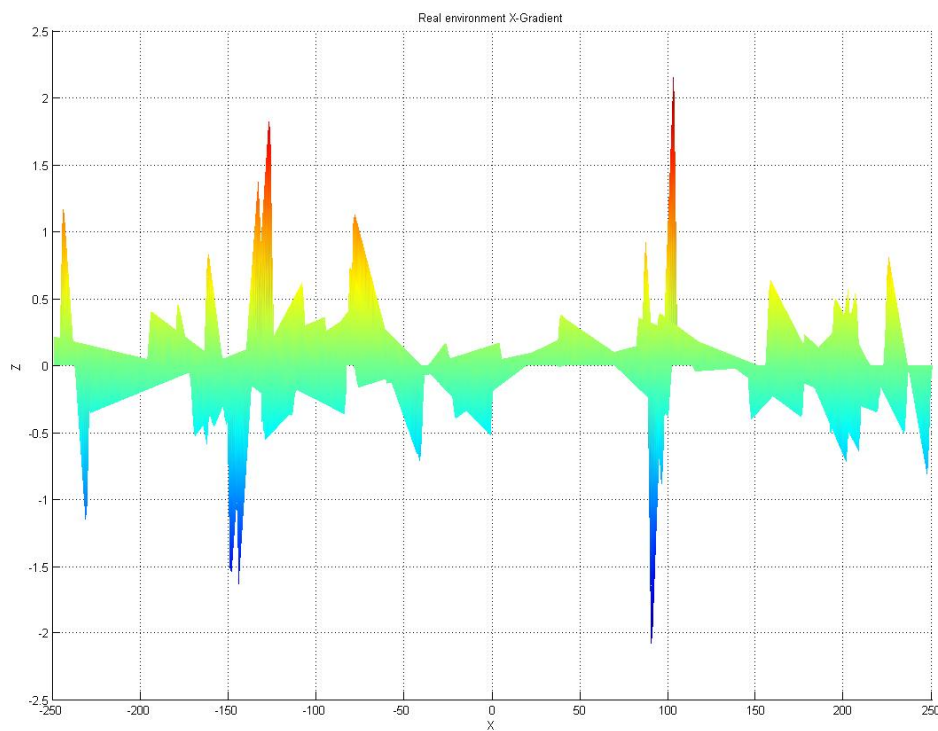


Figura 12. Reconstrucción de los gradientes de un terreno

3.3.3. Confianza

La capa de confianza se utiliza para tener una medida de lo fiables que son los datos almacenados en cada celda. Puesto que la reconstrucción se realiza a partir de las medidas de distancias obtenidas con un láser de barrido horizontal y vertical con incrementos de paso muy pequeños, se obtienen un gran número de puntos de colisión.

Cada punto de colisión es localizado en el mapa y asignado a su correspondiente celda, es posible por lo tanto que haya celdas que recojan un gran número de puntos de colisión y otras con un número pequeño de puntos, o incluso ningún impacto.

De esta forma, con la confianza solo se registra la información proporcionada por los puntos donde el láser ha impactado, ignorando el resto de puntos por donde el láser ha “sobrevolado”.

Otro efecto a considerar es que para una misma celda, puede ocurrir que los puntos estén muy juntos en una pequeña zona, o que por el contrario estén separados ocupando gran cantidad de la superficie de la misma.

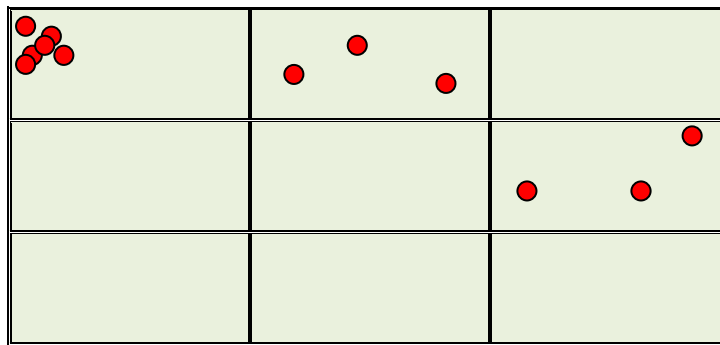


Figura 13. Efecto de la dispersión de los puntos de impacto

Es lógico plantearse qué aporta más información, si unas celdas donde el número de puntos sea reducido pero su dispersión sea grande como puede ocurrir en las celdas (1,2) y (2,3) de la Figura 13 si por el contrario aporta más información la situación de la celda (1,1) donde los puntos no presentan demasiada dispersión pero existe gran cantidad de puntos de impacto.

Por ese motivo, además del número de puntos de impacto sobre una celda hay que tener en cuenta la dispersión de esos puntos, así surgieron durante el desarrollo del software dos tipos de algoritmos que fueron implementados:

- Confianza basada en el número de puntos de impacto
- Confianza basada en la dispersión de los puntos de impacto.

De estos dos tipos de algoritmos investigados que se desarrollaran en apartados posteriores, el segundo de ellos resultó ser más eficaz y fue el elegido para la implementación definitiva.

Los valores numéricos utilizados para medir la confianza son valores reales en el intervalo $[0,1]$, de tal forma que se representa el desconocimiento absoluto del terreno con el valor de confianza 0, y el conocimiento suficiente con el valor 1, si bien internamente tiene una representación de niveles de 8 bits para lograr una mejor compactación de la información.

Las diversas misiones del proyecto NM-RS pueden implicar distintos niveles de conocimiento previo del terreno. Esto determina que la confianza inicial de los mapas puede ser variable pero con alta probabilidad será muy baja al inicio puesto que se considera que la información inicial de los mapas es muy pobre en comparación a la obtenida por el robot en la exploración. También existirán misiones donde la confianza inicial sobre el terreno sea totalmente nula, o simplemente no exista ninguna información, en esos casos se asignará el valor 0.

A medida que el robot progresa en la exploración del entorno y va recogiendo información sobre los puntos de impacto del láser, la confianza de cada celda va aumentando. Aunque como ya se explicará en la sección de algoritmos, cuando se produce un cambio en las medidas del terreno respecto a la información que había antes habrá momentos en los que la confianza disminuya.

A continuación se detalla el nivel de confianza adquirido durante el recorrido de una trayectoria de muestra por el robot. Debido a las limitaciones de simulación se ha tratado de hacer un recorrido a través del terreno poniendo especial detalle en las zonas más interesantes del mismo en cuanto al nivel de relieve se refiere y dejando otras zonas de menor importancia más carentes de información. Para ello se han establecido puntos para la posición del robot mas juntos cuando mayor es el nivel de detalle deseado y mas separados para cuando este nivel de detalle es menor.

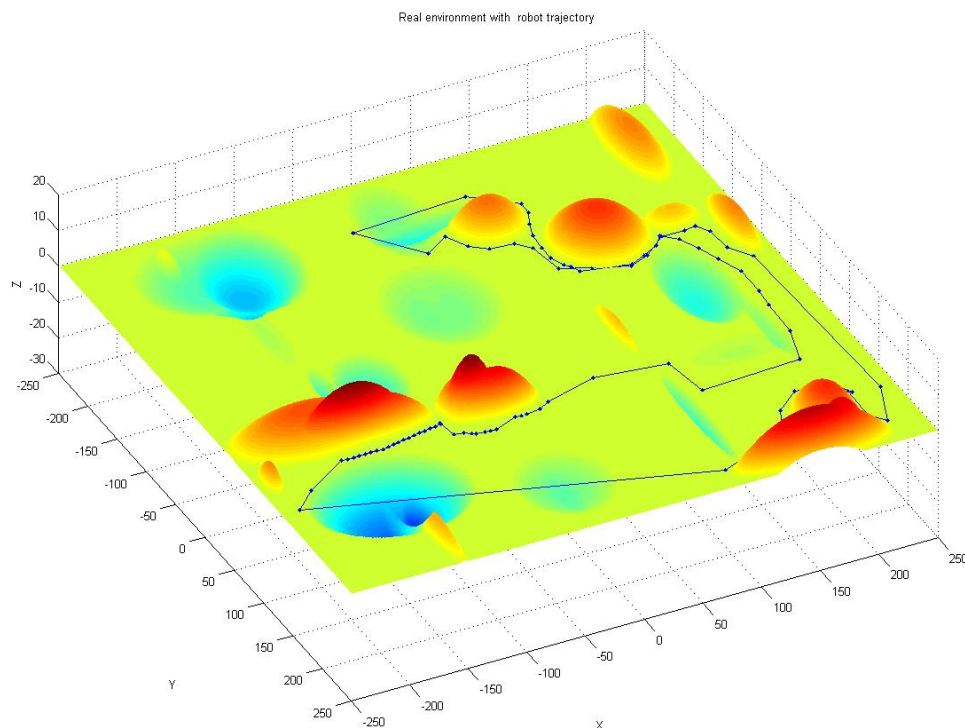


Figura 14. Mapa global con la trayectoria seguida por el robot

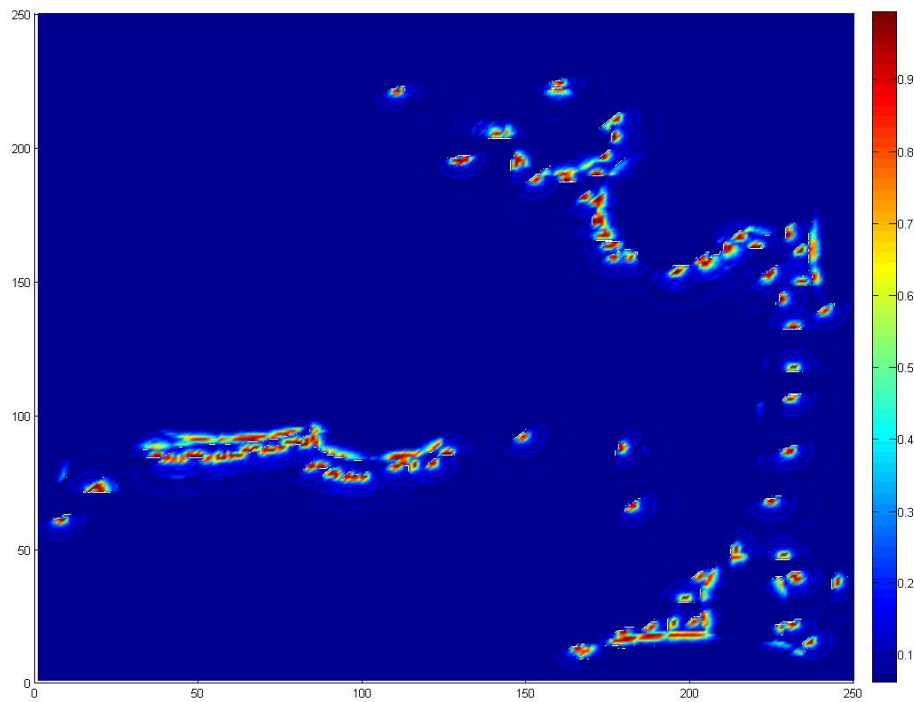


Figura 15. Reconstrucción global de la confianza

En rojo se visualiza las zonas de mayor confianza, mientras que en azul son las zonas más carentes de información.

3.3.4. Ocupación

El grado de ocupación de un terreno se define en relación al robot que transita por él y toma en consideración la capacidad del robot de moverse por superficies inclinadas. Si la pendiente es superior a la tolerada por las capacidades mecánicas del robot, se considerará el terreno como ocupado y si por el contrario el valor absoluto de la pendiente es admisible para el robot, se tomará como nivel de ocupación un valor numérico que subirá linealmente con el valor absoluto de la pendiente.

Si inicialmente se posee conocimiento previo de las alturas del terreno y por tanto de los gradientes del mismo, se podrán inferir los niveles de ocupación para cada celda de forma inmediata.

Este nivel de ocupación se utilizará en otros módulos del proyecto para planificar el camino que puede tomar el robot en su tránsito de un punto a otro, teniendo en cuenta de esa forma la facilidad de paso por cada celda.

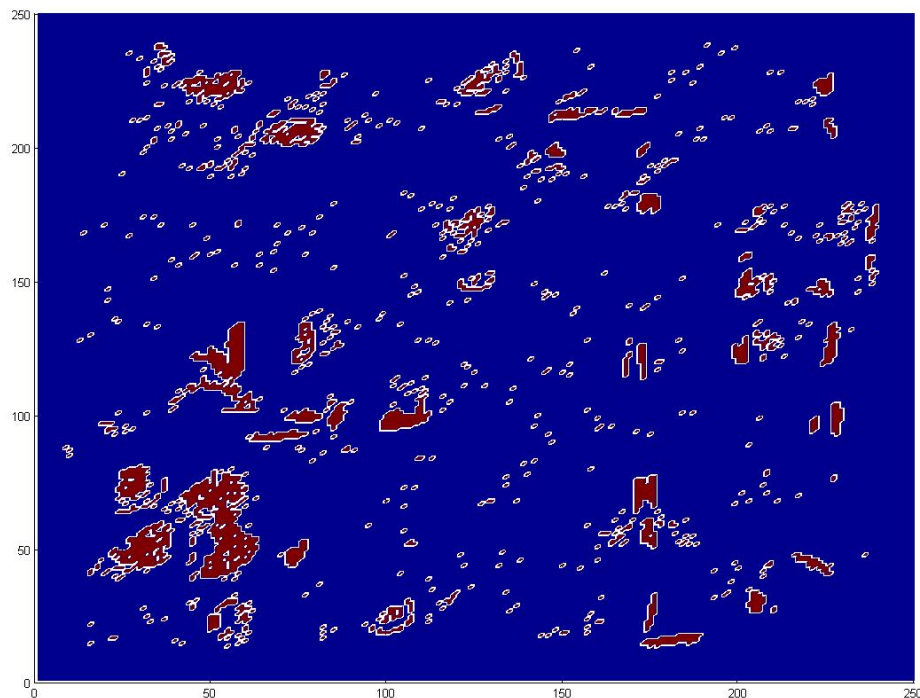


Figura 16. Reconstrucción global del grado de ocupación

En la Figura 16 se observa como visualizando únicamente el nivel de ocupación como libre u ocupado, tenemos las zonas en rojo que representan áreas ocupadas o zonas por donde el robot no puede desplazarse. Mientras que en azul se aprecia el terreno libre o terreno sin información.

3.3.5. Calidad GPS y calidad de la comunicación

El nivel de calidad del GPS determina varios parámetros del comportamiento del robot, como la conveniencia de realizar la reconstrucción o no realizarla, así como de la necesidad de ejecutar el sistema de SLAM.

Se han definido por requerimientos del proyecto tres niveles de comunicaciones GPS, que son, nivel 0 o ausencia de señal, nivel 1 o GPS no diferencial, y nivel 2 o GPS en modo diferencial. Estos niveles implican los siguientes comportamientos:

- Nivel 0: no se realiza la reconstrucción puesto que no se tiene información fiable para realizarla. Se activa el módulo de SLAM para proceder a la localización aproximada del robot.

- Niveles 1 y 2: en esta primera versión del software no se hacen diferencias entre ambos niveles y se procede a realizar la reconstrucción para ambos y a la no ejecución del módulo de SLAM.

El nivel de comunicaciones va asociado a la capacidad de interconexión de unos robots con otros o con una estación base. Este valor al igual que la calidad de recepción de la señal GPS es almacenado exclusivamente en la celda en la que se encuentra el robot en cada momento durante la ejecución de la misión.

3.4. DATOS DE PARTIDA

3.4.1. Mapa inicial del terreno en baja resolución (LowResMapGlobal)

Como se ha comentado anteriormente, en este mapa se almacena la información inicial y es en el que se realiza la fusión de la información procedente de todos los robots para ser transmitida de nuevo a todos los robots con la nueva información actualizada.

Como información de partida se dispone de un conocimiento previo del terreno sobre el cual se va a desarrollar la reconstrucción en forma de un mapa en baja resolución, con unas dimensiones y escala conocidas y determinadas en principio por los requisitos de un determinado ensayo que se vaya a realizar. Se llamará de aquí en adelante `LowResMapGlobal`.

La información previa contenida en este mapa puede ser nula y estar el mapa vacío o por el contrario, disponer de una información inicial cuya calidad no tiene porque ser muy alta, por lo que se asociará al mapa inicial un nivel de confianza bajo (ej. $C=1/16$ sobre 1).

3.4.2. Dimensiones y características del robot

Las dimensiones de los robots vienen impuestas según determinados modelos que van a ser usados para las pruebas. No obstante, éstos son heterogéneos en su tamaño y sistema de locomoción, por lo que cada robot generará mapas de alta resolución con tamaño de la rejilla adaptado al tamaño del propio robot.

3.4.2.1. Características de los sensores

Como ya se ha comentado, este software está realizado para robots equipados con sensor láser utilizados para medir distancias punto a punto y una integración de sensores GPS, inerciales fusionados con un filtro Kalman para obtener la primera

aproximación de la posición. Esta sección revisa las principales características del paquete de sensores requeridos para la navegación autónoma considerando algunos aspectos físicos (valores típicos) como el consumo de potencia, el tamaño, peso para poder tener una visión real del problema y de la solución considerada. Todos estos aspectos han tenido que ser simulados en MRS, y parte de ellos en Matlab. Los aspectos más importantes han sido la localización de los sensores respecto al centro de masas del robot, los ángulos barrido vertical y horizontal así como la resolución angular y resolución en la medida de la distancia del láser. A continuación se verá un ejemplo de valores típicos de sistemas comerciales y se explicará en profundidad las implicaciones de cada parámetro dentro de la simulación.

Sensores para SWR	Integración GPS/IMU/MAG	Láser	Ultrasonidos (valores unitarios)
Potencia	20 w	20 w	300 mw
Tamaño	800 cm^3	1000 cm^3	-
Peso	1 kg	6 kg	25 g
Rango	120 °/s	100 m Vertical: +- 45° paso 1 ° Horizontal: +-90° / paso 1°	0.5 – 10 m
Errores	1° Roll & Pitch 2° Yaw	± 0.02m	0.1 m

Como parámetros esenciales para realizar transformaciones de cambio de base se tiene que las dimensiones del láser y su posición relativa respecto el centro de masas del robot juegan un rol fundamental, ya que es necesario conocer esta información para convertir las coordenadas de un punto visto desde el sistema móvil ligado al láser a el sistema fijo ligado a un punto del mapa global. Así como el peso no es un parámetro de importancia en las simulaciones con Matlab puesto que no se han tenido en cuenta efectos dinámicos.

Otros parámetros fundamentales son los rangos de barrido vertical y horizontal que unidos a la resolución o el paso angular entre dos medidas consecutivas determinan las dimensiones de la matriz de salida del láser. Dicha matriz es uno de los elementos

más importantes en todo el desarrollo de este trabajo y se le dedicará un apartado específico. En cada ciclo de medidas, se obtiene como salida una matriz rectangular cuyos valores son las distancias del láser al punto de colisión para cada rayo caracterizado por los ángulos de barrido horizontal y vertical. De esta forma se obtiene una imagen reconocible del terreno detectado como si se visionara una imagen en un televisor. Las dimensiones de esta matriz llamando α al ángulo de barrido horizontal, σ al vertical y r_h y r_v a las resoluciones horizontal y vertical son:

$n^{\circ} \text{ filas} = \frac{(\sigma_{max} - \sigma_{min})}{r_v}$	(17)
$n^{\circ} \text{ columnas} = \frac{(\alpha_{max} - \alpha_{min})}{r_h}$	(18)

Donde el subíndice h indica horizontal y el subíndice v indica vertical.

3.4.2.2. Emplazamiento y orientación del robot

Se conoce a priori el estado inicial del robot, es decir sus coordenadas absolutas respecto al origen del mapa fijo (X_1 , Y_1 , Z_1) y la orientación del robot respecto al ejes fijos dados por los ángulos de Euler:

- **Alabeo o Roll (φ)**, ángulo que representa un giro respecto al eje X_1 fijo.
- **Cabeceo o Pitch(θ)**, ángulo que representa un giro respecto al eje Y_1 fijo.
- **Guiñada o Yaw(ψ)**, ángulo que representa un giro respecto al eje Z_1 fijo.

Inicialmente se partirá siempre con el robot situado en el centro del mapa local.

3.4.3. Aspecto del mapa local

Para realizar la reconstrucción se utilizará un mapa local que es el mapa que utiliza el robot en tiempo real para almacenar los cálculos realizados con las medidas tomadas por el láser. Este mapa denominado **HighResMap** es un mapa local en alta resolución. Las dimensiones, y escala serán parámetros que se elegirán previamente según los requisitos de cada ensayo. Para las pruebas en Matlab se ha venido utilizando unos valores de escala de 1 m. y con un número de filas y columnas de 240. El cambio de estos parámetros influye enormemente en la apreciación de los resultados, sobre todo de forma gráfica, puesto que cuanto más difieran las dimensiones del local respecto del global más difícil es apreciar los cambios en este al reconstruir. Por ese motivo se

eligieron estos valores, donde se hace primar más la visibilidad de los resultados que el realismo de las dimensiones consideradas. En otros casos más realistas es probable que la dimensión del mapa global sea mayor.

Este mapa aparecerá vacío inicialmente y comenzará a tener contenido a partir de la primera toma de datos del láser.

3.4.4. Trayectoria del robot

Se establecerá un camino que deberá seguir el robot por el mapa. Ese camino vendrá dado por una serie de puntos designados según sus coordenadas (x , y) sobre el mapa global, por lo que para la simulación en Matlab será un camino discreto basado en puntos. En cada lectura del láser se irá obteniendo la nueva posición del robot y las medidas del láser se calcularán en base a esa nueva posición.

A continuación se puede observar un ejemplo de una trayectoria elegida para el robot. En este caso el mapa global consta de una serie de montañas y de valles que el robot va bordeando. Se puede observar algún salto brusco en los movimientos del robot, este tipo de saltos solo tienen como finalidad el ahorro de tiempo de cálculo por zonas donde no se ha requerido excesivo detalle.

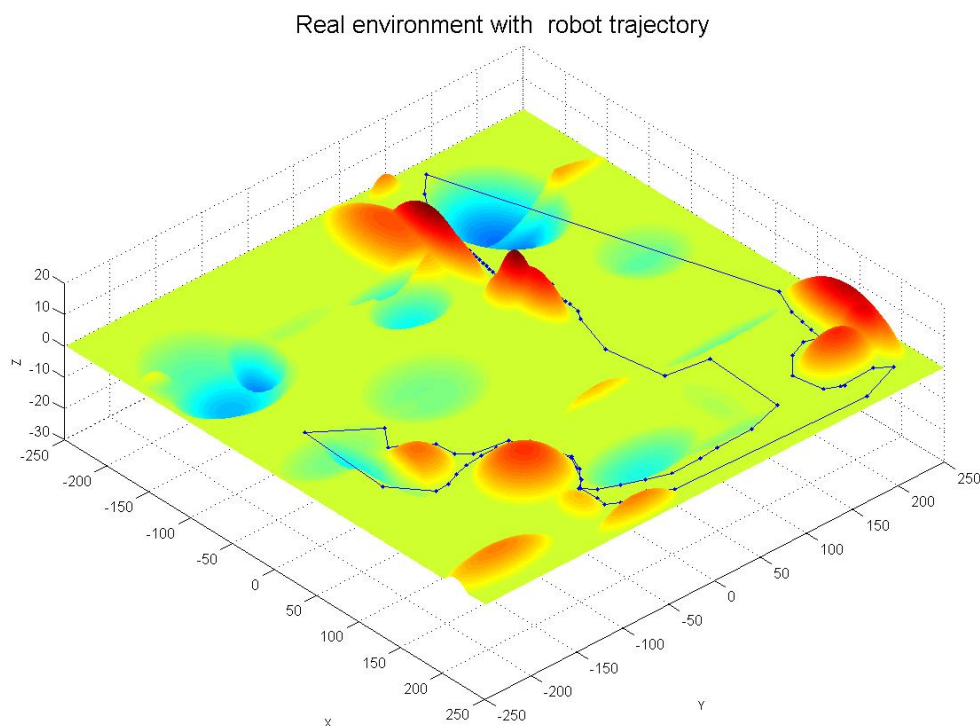


Figura 17. Mapa global con la trayectoria del robot

3.4.5. Posición del mapa local en el interior del mapa global

Se denominará c_local_global a la posición en cada instante del centro del mapa local ($HighResMap$) respecto al mapa global $LowResMapGlobal$. Llamaremos ciclo al proceso completo de reconstrucción y fusión de cada matriz de medidas del láser. Al inicio de cada ciclo se comprueba que el alcance del láser del robot está comprendido en el interior del mapa local y que no existan zonas de reconocimiento del láser que pudieran quedar fuera. Si el alcance del láser está dentro del rango del mapa local, no se desplaza el c_local_global , en caso contrario sí se desplazaría.

Debido al movimiento del robot por el terreno, esta comprobación se realiza en cada nuevo punto de desplazamiento. Al ser la simulación a base de desplazamientos discretos podría ocurrir que en un punto el láser se encontrará correctamente situado en el interior de los límites del $HighResMap$ y en el punto siguiente no ocurriera lo mismo.

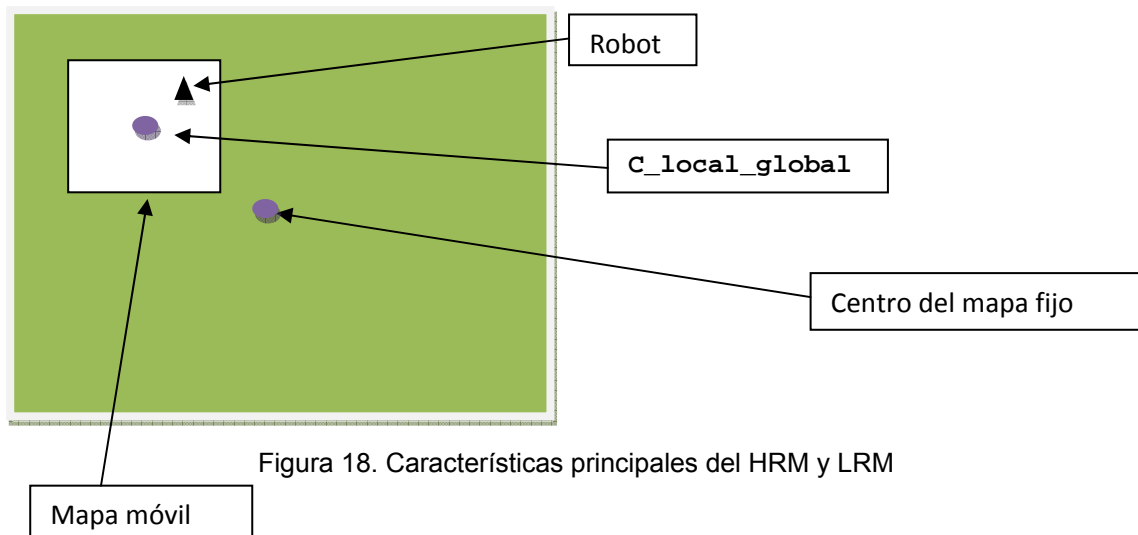


Figura 18. Características principales del HRM y LRM

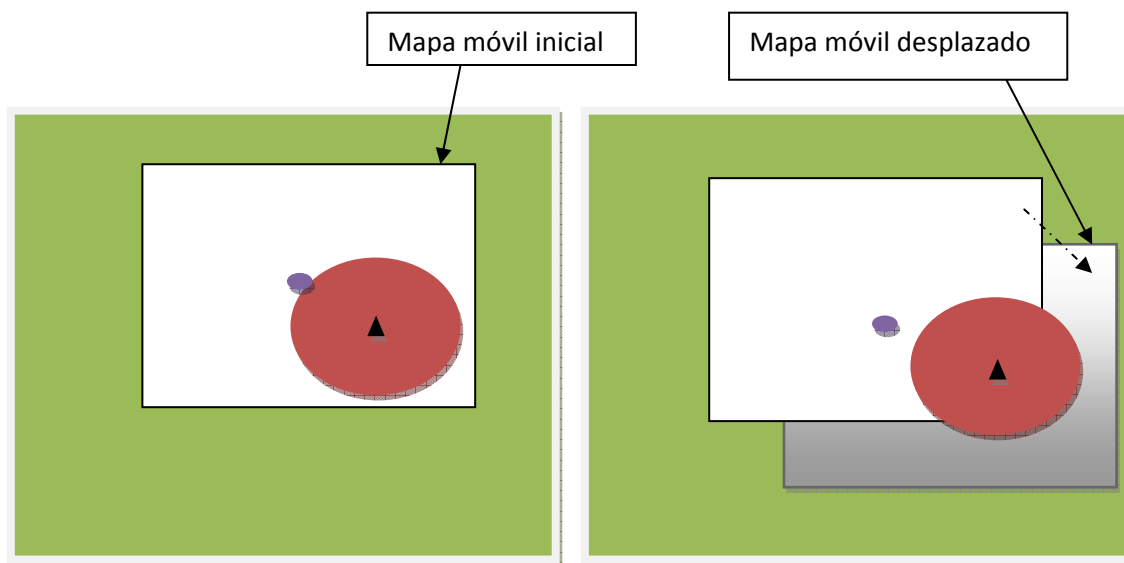


Figura 19. Desplazamiento del HRM

Como se puede observar en la figura 2 la situación es correcta puesto que el rango de alcance del láser se encuentra en el interior del mapa local, en esta situación no se desplazará c_local_global . Distinto es el caso de la figura 3 donde el láser sale del alcance del mapa local. En esta situación es preciso desplazar c_local_global y con ello el mapa local para que el láser este contenido en el mapa. Más adelante se explicará en detalle el proceso de desplazamiento de dicho mapa

3.4.6. Sistemas de referencia utilizados

Para la navegación autónoma y el reconocimiento del terreno se han usado 3 sistemas de coordenadas fundamentales. En primer lugar se encuentra el sistema fijo que es un sistema con origen en un punto determinado del mapa global, un punto determinado previamente. Por otro lado, y como sistemas de referencia móviles se tienen el sistema de referencia ligado al centro de masas del robot y solidario con él y el sistema de referencia ligado al láser y también solidario con él. Denominaremos sistema 1 al sistema fijo, sistema 2 al sistema móvil ligado al centro de masas del robot y sistema 3 al sistema ligado al láser. De esta forma tenemos:

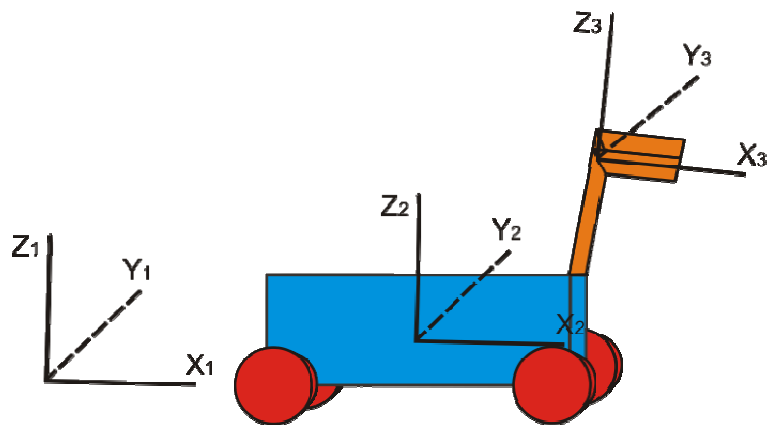


Figura 20. Sistemas de referencia del robot

Además se definirán ahora los sentidos del barrido horizontal (pan) y vertical del láser (tilt) en el sistema asociado al láser. Se tomará el eje Z_3 como origen del ángulo de barrido horizontal α y el sentido positivo de dicho ángulo será una rotación hacia X_3 . De forma similar se tomará el eje X_3 como el origen del ángulo de barrido vertical σ siendo el sentido positivo del mencionado ángulo un giro hacia el eje Z_3 en sentido negativo.

Es decir:

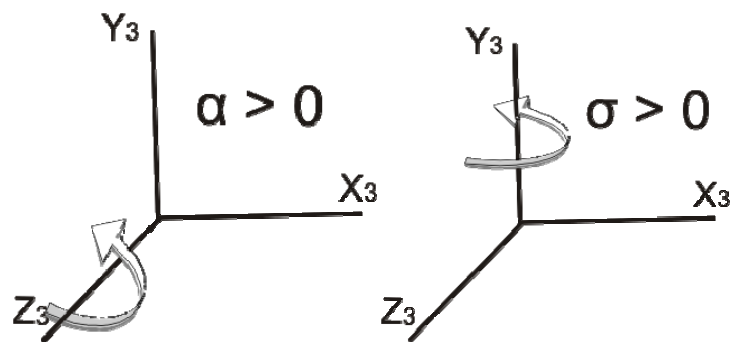


Figura 21. Sentidos de giro de los ángulos de barrido del láser.

4. ALGORITMOS DE RECONSTRUCCIÓN

4.1. *Introducción*

Se utiliza el término reconstrucción para indicar el procedimiento desarrollado para la obtención de un modelo digital del entorno, que es uno de los objetivos principales de este proyecto. Dicha reconstrucción del terreno explorado por el robot, mejora la información de la que se ocasionalmente se dispone a priori.

Para realizar este proceso se parte únicamente de la información obtenida por el láser, en forma de una matriz de medidas. A partir de estos datos y de la información sobre los ángulos de barrido horizontal y vertical que el láser proporciona en cada momento, se obtienen las coordenadas de los puntos en los que el láser colisiona referenciado al sistema 3 (Ver apartado anterior).

Posteriormente, tras un cambio de base se obtiene dicho punto de impacto del láser visto desde el sistema 1 (Ver apartado anterior). De esta forma, se podrá almacenar en la matriz del mapa reconstruido la información obtenida y estimar tanto la información global como la local.

Para la correcta localización en un mapa global geo-referenciado es necesario lógicamente conocer con precisión no sólo la posición del robot en éste en el momento de la adquisición de los datos sino también su actitud.

En términos prácticos, y dado que la estimación de la actitud tiene una gran disponibilidad (solo cuando el sensor inercial ser avería no proporciona estimaciones) se puede geo-referenciar cuando el GPS proporciona una información precisa, lo que ocurre cuando se dispone de una estimación de la posición bien en modo individual o diferencial (modos 1 y 2)

En caso de no disponer de una estimación de posición por parte del GPS, se procederá a la localización por diversos métodos de SLAM que se comentarán en el capítulo (pon aquí el número de capítulo).

4.2. *Esquema general del proceso de reconstrucción*

Como ya se ha comentado anteriormente el módulo M45 es llamado cada vez que se dispone de información nueva procedente del láser. La nueva información con la que se realiza la llamada al algoritmo, permite realizar los cálculos necesarios para reconstruir todos los puntos de impacto del láser. Se utiliza la metodología siguiente

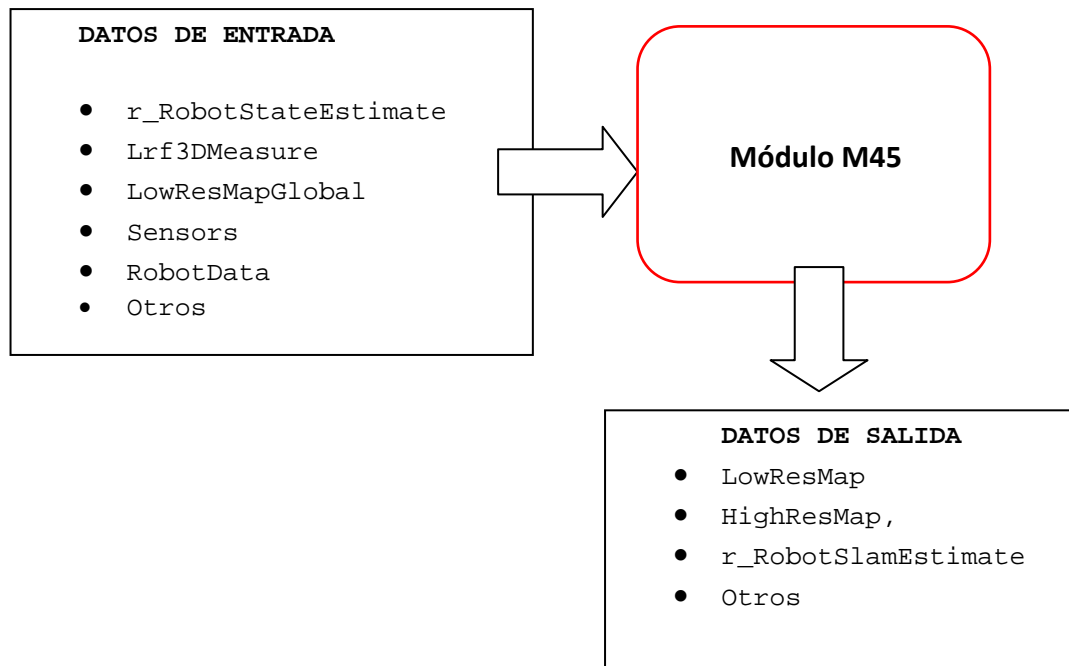


Figura 22. Entradas y salidas del módulo 45 (Mapeado)

4.2.1. Estimación del estado del robot

Para la obtención de la variable `r_RobotStateEstimate` de la que se obtiene la información de partida referente a la primera estimación de la posición y la orientación del robot se utilizan una serie de filtros de Kalman en cascada que fusionaran la información de los sensores inerciales, GPS y sistema de odometría del vehículo, como muestra la Figura 23 .

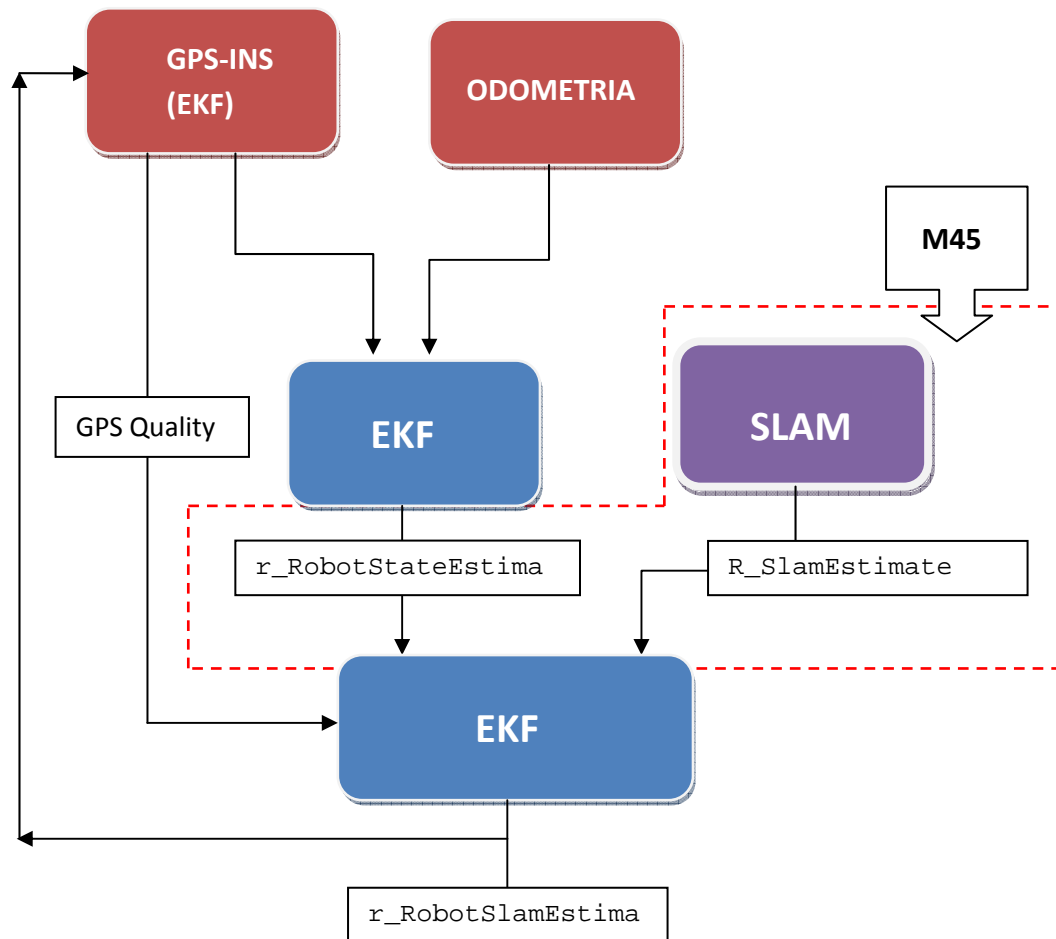


Figura 23. Esquema de la fusión sensorial y estimación del estado del robot.

Debido a que existen diferentes tipos de robots dentro del proyecto, cada uno utiliza diferentes modelos de odometría con sus correspondientes parámetros, de forma que se genera una estimación del incremento de la posición por módulos anteriores, de tal forma que pueda ser utilizada en la fusión de datos de una forma homogénea con independencia del vehículo utilizado.

Existe también un módulo encargado de fusionar la información proveniente del GPS y del sensor inercial para proporcionar una solución completa sobre posición, velocidad y orientación.

Como una de las más importantes contribuciones de este proyecto, se proporciona una estimación absoluta de la posición del robot utilizando la información del mapa previamente obtenido y las medidas del láser. Esta medida se introduce como estimación del estado en otro filtro de Kalman cuando no se dispone de estimación en la posición mediante GPS. Esta salida se utiliza como entrada como estimación del primer filtro de Kalman de la cascada.

4.3. ESQUEMA GENERAL DEL MÓDULO

El algoritmo de reconstrucción dispone de una etapa previa de configuración según una serie de parámetros como son el rango de movimiento horizontal y vertical del láser, la resolución de dicho movimiento y el alcance del láser.

También se procede a la inicialización del mapa de alta resolución a partir de las características que éste debe poseer, la posición del robot, el mapa global y la información sobre los sensores y dimensiones características del robot. Estos detalles se explicarán más detalladamente en apartados posteriores.

Una vez inicializados los mapas y las variables de forma conveniente, se procede a comprobar si existe señal GPS. Este es un punto fundamental en el mecanismo de operación de todo el módulo, puesto que determina si se va a proceder a la reconstrucción o si por el contrario se va a proceder a usar el módulo de Slam para obtener una mejor aproximación de la posición. Si se dispone de señal GPS se procederá a reconstruir utilizando la estimación de la posición, por el contrario, si no se dispone de señal GPS, se ejecutará el módulo Slam, que en la mayoría de los casos proveerá de una estimación de la posición mejorada.

Como herramienta de ayuda al desarrollo del algoritmo, se ha diseñado un entorno utilizando Matlab que permite la creación de entornos mediante la utilización de una superficie plana y añadir elementos positivos o negativos (montañas o valles) así como una simulación de un sensor láser que se puede orientar y situar de forma arbitraria en dicho entorno y éste proporciona la simulación del sensor real proporcionando una matriz de puntos de colisión según los parámetros de entrada deseados (rangos, incrementos, alcances y error en la estimación)

En primera instancia, se genera un mapa local auxiliar, utilizando para ello únicamente la información proveniente de una lectura del láser. Ese mapa auxiliar se fundirá posteriormente con la última estimación del mapa local de alta resolución (HighResMap) dando como resultado un nuevo mapa de alta resolución.

Dicho nuevo mapa (adaptado en sus dimensiones) se fusiona posteriormente con la última versión disponible de forma local en el robot del mapa global (LowResMapGlobal) , produciendo con ello la salida deseada del módulo de mapeado.

A continuación se describe con detalle cada fase del algoritmo utilizado.

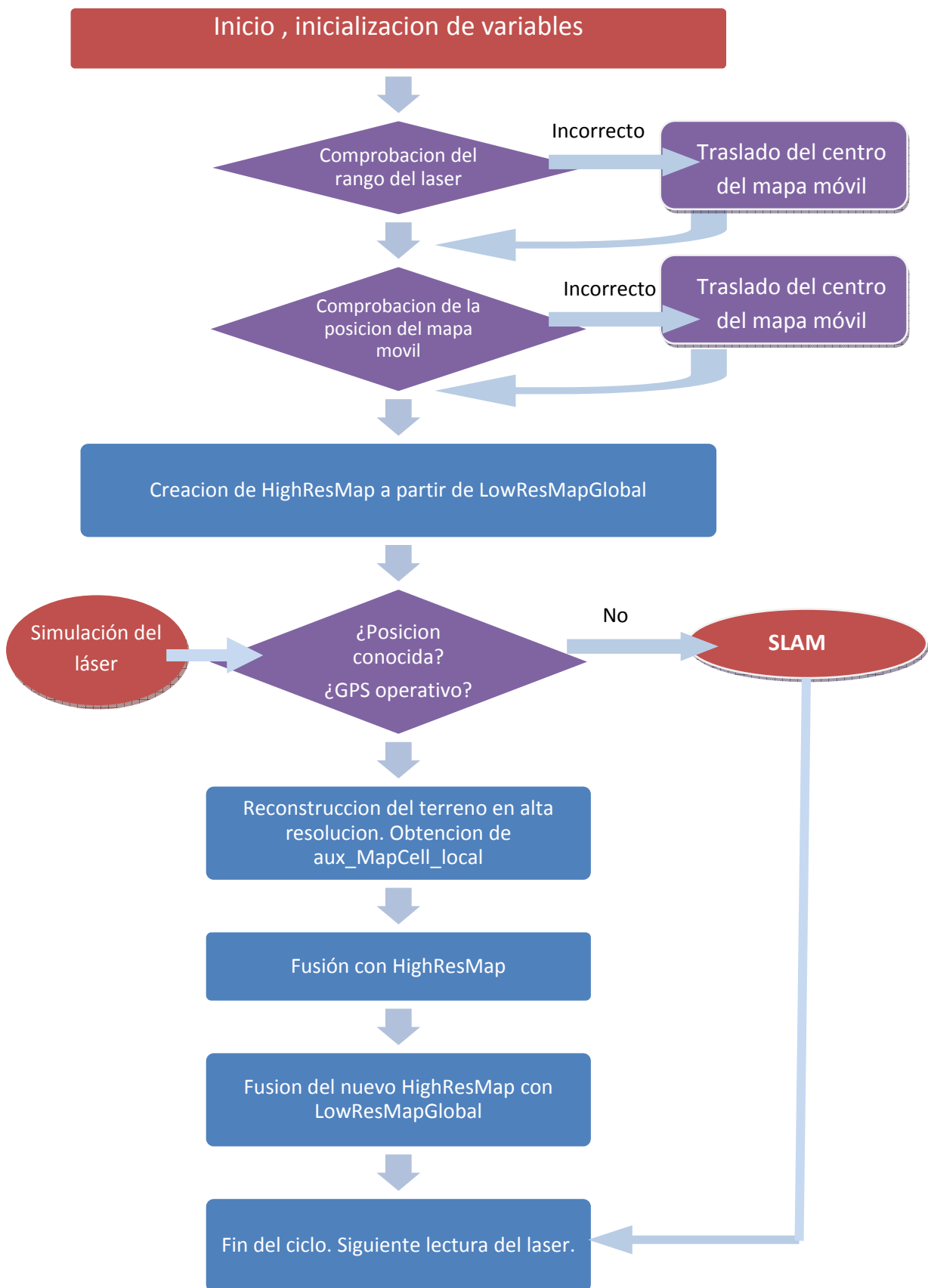


Figura 24. Proceso de reconstrucción y mapeado.

4.4. PROCESO: COMPROBACIÓN DEL RANGO DEL LÁSER

Como se ha descrito anteriormente, el sistema NMRS trabaja con dos tipos de mapa; el local de alta resolución (de una extensión pequeña, de orden de 2 veces el alcance del láser) y mapas de baja resolución (de la misma extensión que el entorno de simulación).

A su vez, existen mapas de baja resolución globales (que es el resultado de la fusión de la información de todos los robots existentes en la simulación) y mapas de baja resolución locales (que es la versión local de cada robot del mapa global modificado en las aportaciones realizadas a la reconstrucción por parte de dicho robot). La diferencia entre los mapas locales y globales será tanto mayor como mayor sea el tiempo transcurrido desde la transmisión de los datos de los robots y la fusión de la información procedente de varios de ellos y la retransmisión de nuevo a los robots participantes.

Con objeto de establecer la porción del mapa de baja resolución que se corresponde con la de alta resolución se realiza una serie de comprobaciones con el objeto de desplazar correctamente el mapa de alta resolución sobre el de baja.

Este desplazamiento del mapa de alta resolución debe realizarse cuando el desplazamiento del robot en el mapa de alta resolución motiva que la zona de alcance del láser se sale de éste. Esto puede producirse de forma habitual si las dimensiones que se eligen para el mapa local son pequeñas. Por ese motivo se produce un desplazamiento casi continuo del mapa local sobre el mapa global con una grabación y lectura de datos también continua. Para solucionar este problema se han diseñado unas comprobaciones previas que hay que realizar siempre.

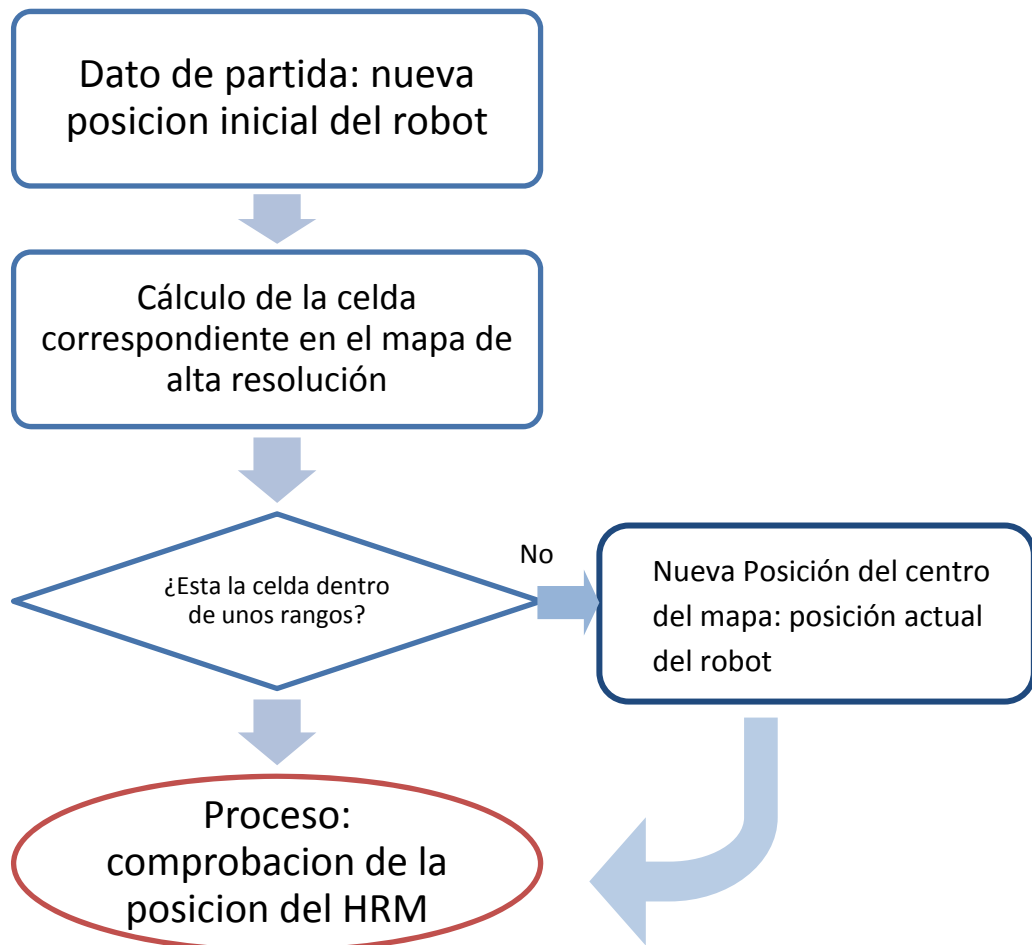


Figura 25. Proceso para calcular el desplazamiento del mapa de Alta Resolución

Al iniciar la simulación se parte con el punto central del mapa de alta resolución situado sobre el robot, por ello no existirá ningún problema relacionado con el hecho de que el punto de impacto del láser no se encuentre dentro del mapa. Pero en el movimiento del robot se debe comprobar periódicamente que el alcance máximo del láser se encuentra dentro de los límites del mapa.

Como se puede observar en el esquema anterior, se trata por tanto de calcular la celda que ocupa el robot en el mapa local y ver si la circunferencia correspondiente a todo el alcance de láser está contenida en el mapa.

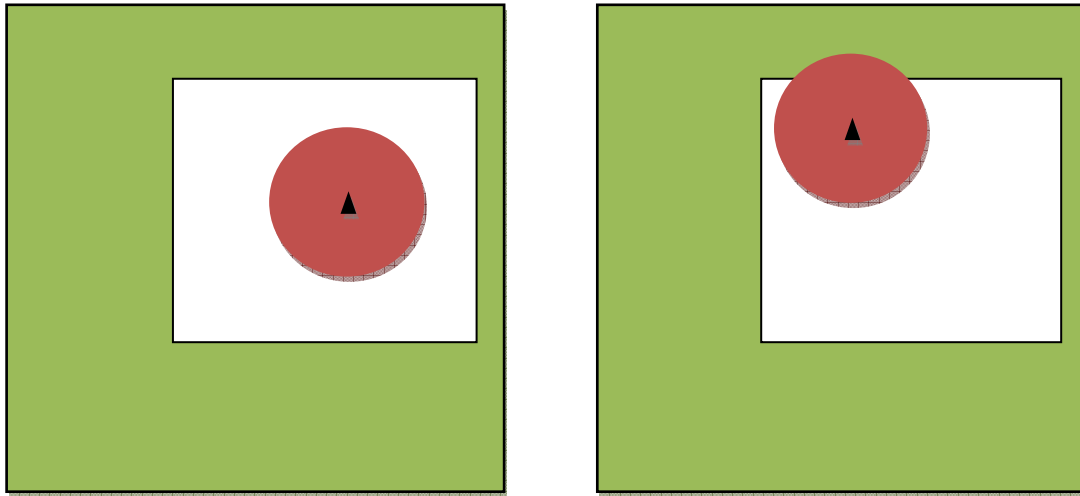


Figura 26. Diferentes situaciones del robot dentro del Mapa de alta resolución.

Como se puede observar en la Figura 26 (izq.), el rango de alcance del láser representado mediante un círculo, se encuentra dentro del área del mapa local, mientras que en la figura de la derecha, ocurre el caso contrario. Mientras que en el primer caso la comprobación sería correcta y no daría lugar a ninguna acción, en el segundo caso habría que recolocar el centro del mapa local de tal forma que ocupe ahora la posición actual del robot, tal y como indica la Figura 27:

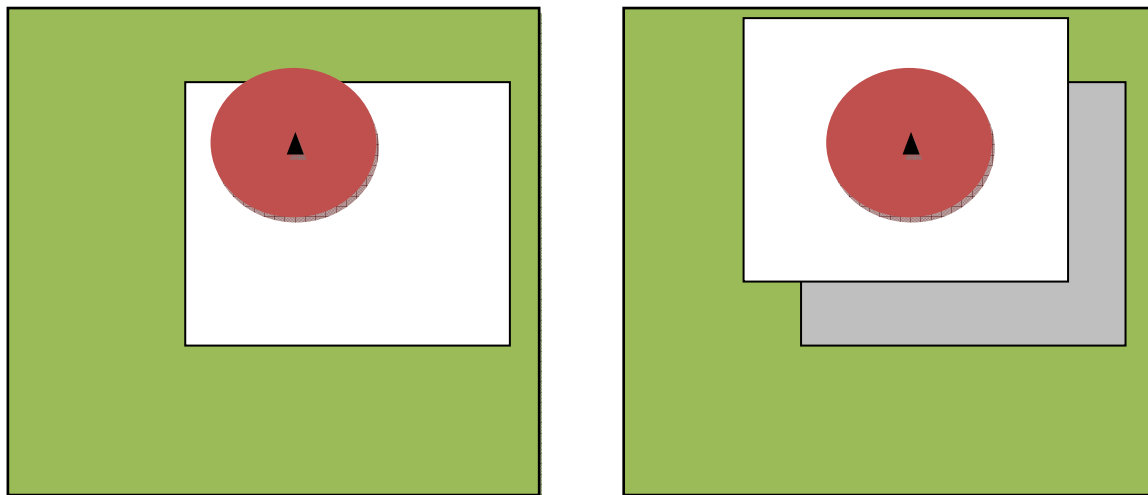


Figura 27. Diferentes situaciones del robot dentro del mapa de Alta resolución.

De esta forma la posición actual del robot vuelve a coincidir con el centro del mapa local.

4.5. PROCESO: COMPROBACIÓN DE LA POSICIÓN DEL MAPA MÓVIL

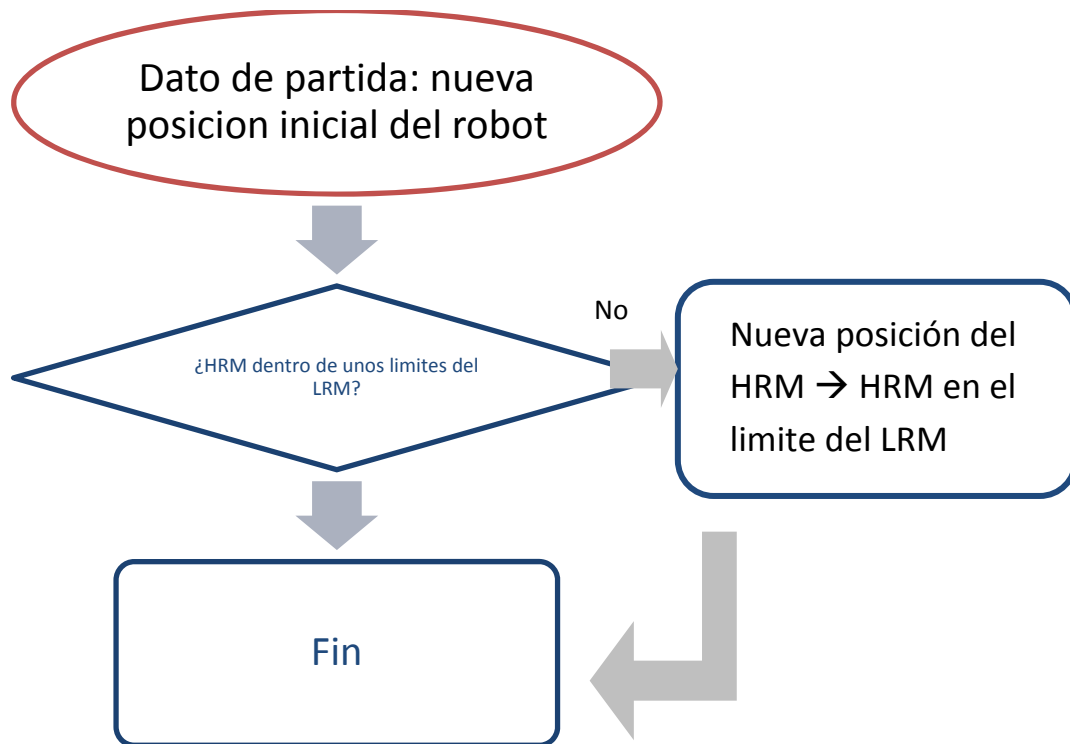


Figura 28. Esquema del proceso de comprobación de la posición del mapa móvil

Puede ocurrir que debido al movimiento del robot, o debido a la traslación efectuada en el proceso anterior el mapa local se encuentre en el extremo del mapa global. En ese caso se debe impedir que el sistema intente crear un nuevo HRM que se salga de los límites permitidos por el LRM. El criterio utilizado en estos casos consiste en superponer únicamente hasta los límites de la zona permitida aunque el robot no se encuentre centrado respecto al nuevo mapa de alta resolución creado.

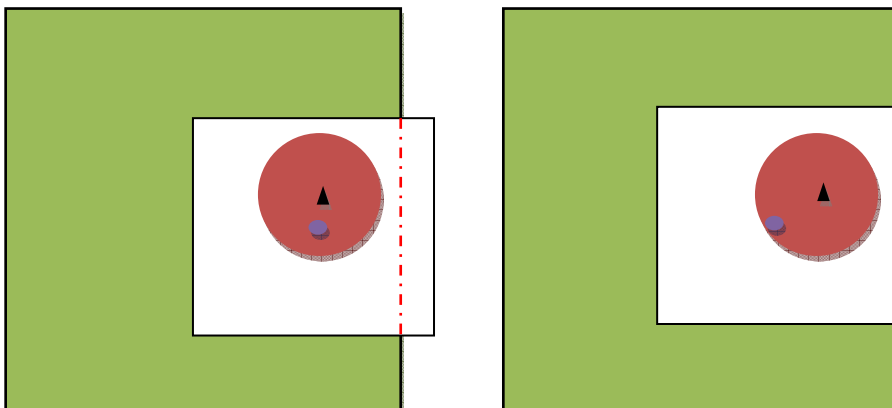


Figura 29. Situación relativa del mapa de alta resolución sobre el de baja.

Como se puede observar en la Figura 29, el mapa local se escapa de los límites permitidos. A esta situación se podría llegar únicamente cuando al desplazar el HRM mediante el proceso de comprobación del rango del láser, se diese un salto brusco en la posición del nuevo mapa, invadiendo la zona prohibida. Por ese motivo se efectúan los dos procesos de forma consecutiva, primero se procede a comprobar el rango del láser, se hacen los cambios oportunos y luego se comprueba que el HRM se encuentre en el interior del LRM.

A continuación (Figura 30) se muestra un caso particular de una situación que se podría dar en alguna ocasión. Se trata de medidas del láser que salen fuera del LRM. Esta situación puede darse si no se impide que el robot pase por cerca del borde del LRM. Si esto ocurre, es probable que el alcance del láser quede fuera de dicho mapa. En este caso se corrige la posición del centro del HRM, desplazándolo hacia la izquierda en este caso para que tenga cabida en el mapa global. Al realizar esta operación podría ocurrir que el alcance del láser estuviera ahora fuera del mapa local. En este caso se ignoraran las medidas que cumplan esa condición.

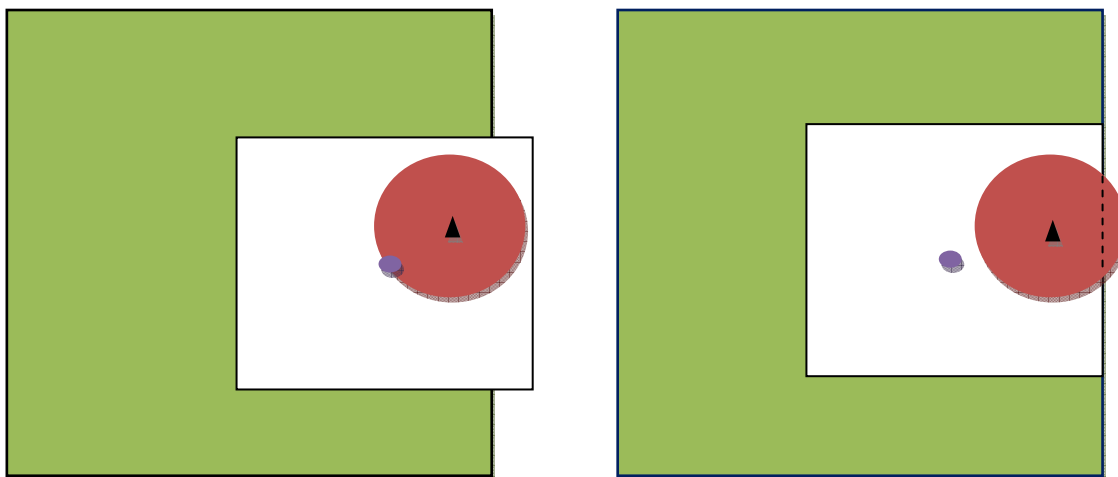


Figura 30. Situación relativa del mapa de alta resolución sobre el de baja.

Una vez realizados los dos procesos anteriores se obtiene una salida que se denominará $C_H = (X_H, Y_H)$ y que corresponde a las coordenadas del centro del HRM que se han ido calculando en base a todas las restricciones antes mencionadas.

El proceso de captura de datos desde LRM hacia HRM y el volcado de los mismos en sentido contrario será explicado detalladamente en apartados posteriores.

4.6. Proceso : Creación del HRM inicial a partir de LRM

Teniendo en cuenta que el tamaño de una celda de un mapa de baja resolución siempre debe ser múltiplo del tamaño de una celda de un mapa de alta resolución, los algoritmos para situar relativamente ambos mapas resultan sencillos. Así, una vez determinado el centro del HRM como se ha explicado en apartados anteriores, se calcularán las celdas del LRM que deben ser transferidas al mapa local. Para ello bastará con a partir de la celda del LRM correspondiente a la coordenada $C_H = (X_H, Y_H)$ sumarla o restarla la mitad de la dimensión del HRM adaptada por medio de la escala, las formulas a utilizar son las siguientes:

$n = \frac{\text{Escala LRM}}{\text{Escala HRM}}$	(22)
$I_o = I_g - \frac{S_{Nm1_{local}}}{2} \left(\frac{1}{n} \right) + 1$	(23)
$I_f = I_g + \frac{S_{Nm1_{local}}}{2} \left(\frac{1}{n} \right)$	(24)
$J_o = J_g - \frac{S_{Nm2_{local}}}{2} \left(\frac{1}{n} \right)$	(25)
$J_f = J_g + \frac{S_{Nm2_{local}}}{2} \left(\frac{1}{n} \right) - 1$	(26)

Donde:

- I_o, J_o : son la fila y columna iniciales respectivamente del mapa LRM.
- I_f, J_f : son la fila y columna finales.
- I_g, J_g : son la fila y columna del LRM correspondientes a C_H .
- $S_{Nm1_{local}}$ y $S_{Nm2_{local}}$ son el numero de filas y columnas del HRM.

Como se puede observar en el ejemplo que viene a continuación en la Figura 31, se tiene un LRM de dimensiones 8 x 8 cuya celda central viene sombreada y su coordenada identificativa se indica mediante un círculo en su esquina característica. Si se desea situar un mapa de alta resolución cuyas dimensiones equivalentes al LRM son una cuadrícula 4 x 4, al aplicar las ecuaciones anteriores resultaría:

$$[I_g, J_g] = [4, 4]$$

$$[I_o, J_o] = [3, 3]$$

$$[I_f, J_f] = [6, 5]$$

Lo que resulta en un mapa HRM también sombreado de las dimensiones deseadas (4 x 4), cuyo centro sigue siendo C_H . Pues bien, con esta metodología se calculan los índices de las celdas que delimitan el mapa de alta resolución dentro del LRM. Así, solo bastará a continuación copiar las celdas correspondientes en orden para crear un nuevo HRM.

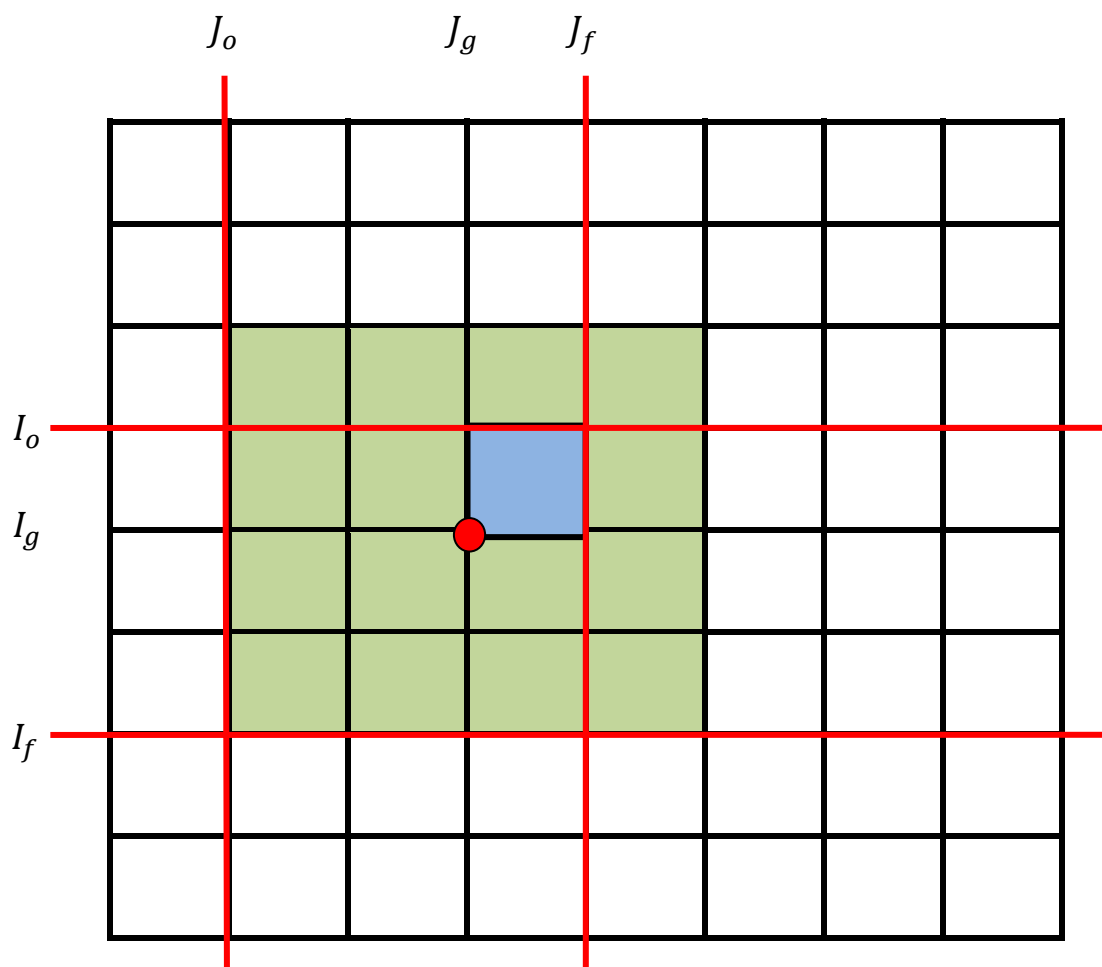


Figura 31. Creación del HRM inicial a partir del LRM

Para calcular el número de celdas de LRM equivalentes a otro número de celdas de HRM basta considerar que:

$$\begin{aligned} N^{\circ} \text{ celdas HRM} \cdot \text{Escala HRM} \\ = N^{\circ} \text{ celdas LRM equivalentes} \cdot \text{Escala LRM} \end{aligned} \quad (19)$$

Por ello el número de celdas del HRM equivalentes a una del LRM es:

$$N^{\circ} \text{ celdas HRM} = \frac{1 * \text{Escala LRM}}{\text{Escala HRM}} \quad (20)$$

Hay que tener en cuenta que en el desplazamiento del HRM suele ocurrir que parte del nuevo mapa coincide con el mapa antiguo, por lo que esa información coincidente no se introducirá en el HRM sino que se conservará del estado anterior. Dado que cada paso de alta a baja resolución implica una pérdida de información, es conveniente realizar este paso el menor número de veces. Equivalentemente a esto, es conveniente también utilizar la información existente de alta y no sobrescribirla con la información que se posee en baja ya que esta será siempre de peor calidad. Se puede ver claramente con un ejemplo grafico:

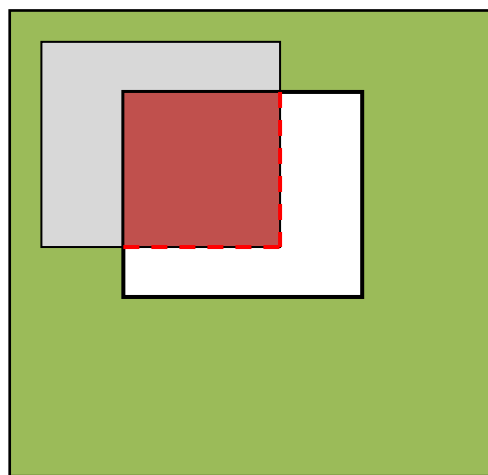


Figura 32. Conservación de los datos del HRM en la adquisición de nueva información

Como se puede ver al desplazar el HRM la zona coincidente sombreada más oscuro en el dibujo se debe conservar y no se deben importar nuevos datos de baja en esa área puesto que se perdería calidad de la información que se posee.

4.7. PROCESO: SIMULACIONES DEL TERRENO Y DEL LÁSER

Como se ha comentado anteriormente, con objeto de facilitar la depuración del software en Matlab, se desarrollaron una serie de aplicaciones con el objeto de tener todos los procesos disponibles antes de pasar el código al Microsoft Robotics Studio. Por ello, dos aplicaciones básicas fueron necesarias; una de ellas crea un terreno virtual y con ello un Low Resolution Map y la otra simulan el impacto del haz de rayos láser sobre dicho terreno.

Simulación del terreno

Este programa crea un mapa formado por la intersección de un terreno plano con unos determinados números de paraboloides positivos y negativos. Todo ello parametrizado de forma que se puede elegir el número de parábolas (Ver Figura 33). Aunque la localización de estas se produce de forma aleatoria en cada ejecución del software. También se pueden elegir las dimensiones totales del terreno. De esta forma se crearon numerosos mapas de prueba, mapas que vienen representados matemáticamente como una matriz cuyos valores reales representan la altura de cada celda.

Este programa resulto ser de gran utilidad para las simulaciones sucesivas dado que se podían crear mapas con los elementos característicos deseados.

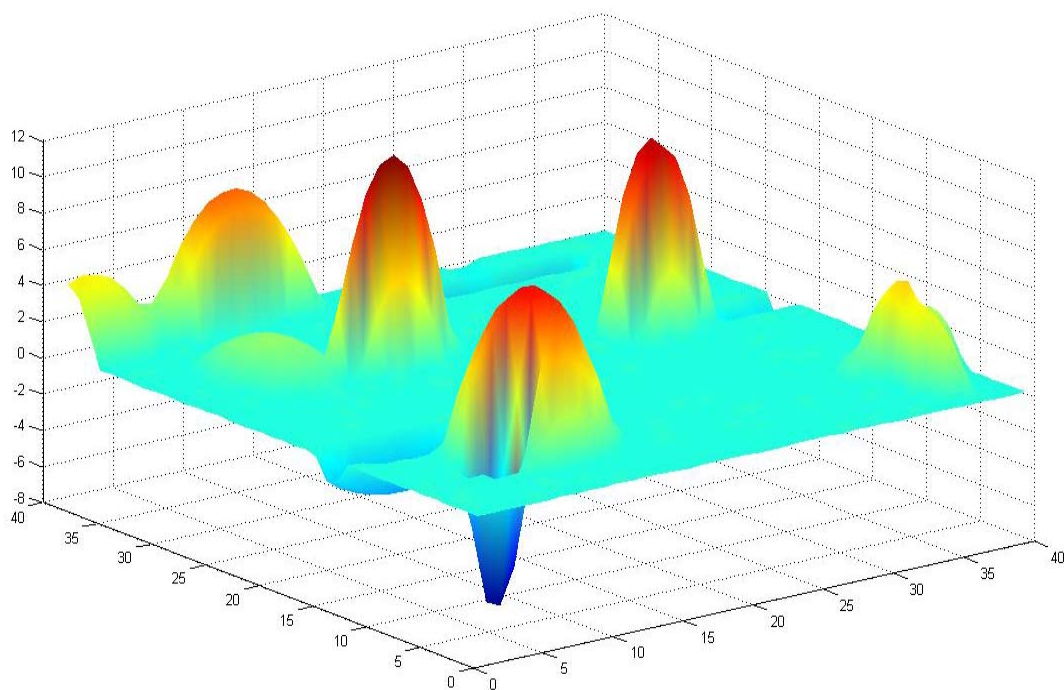


Figura 33. Ejemplo de entorno simulado con Matlab.

Simulación del láser

Para la simulación del láser se desarrollaron una serie de funciones con objeto de calcular las intersecciones de los rayos lanzados según unos ángulos determinados con los elementos del entorno (paraboloides o suelo) dentro del alcance máximo del láser. Con ello, se obtiene una serie de puntos (X, Y, Z) de colisión, donde la distancia de cada uno de los puntos de colisión al punto de partida (láser), se corresponde con el valor medido por el mismo.

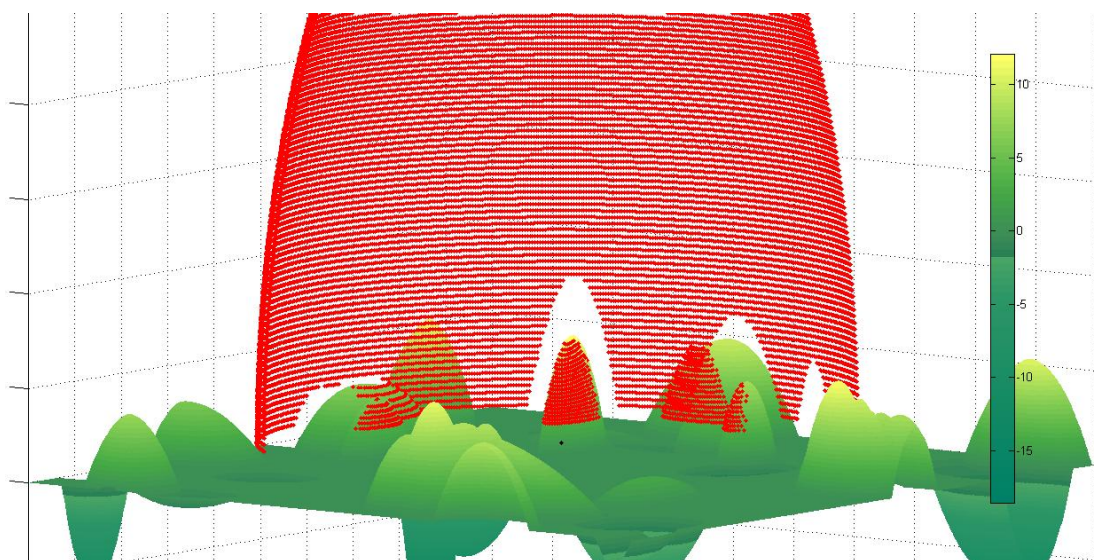


Figura 34. Representación de puntos de colisión del láser.

Como se puede observar en la Figura 34 el láser forma una semiesfera de puntos de impacto, aunque evidentemente donde el láser no impacta se lee el valor máximo correspondiente a su alcance máximo (típicamente 80 metros).

4.8. CALCULO DE LAS COORDENADAS CARTESIANAS DE LOS PUNTOS DE IMPACTO

En este apartado se va a describir el algoritmo para transformar la matriz de distancias entregada por el láser en los correspondientes puntos de colisión referenciados al sistema de coordenadas del mapa.

En primer lugar se describe la estructura de la matriz D de distancias proporcionada por el láser, denominada *Lrf3dMeasure*.

El sistema láser realiza dos barridos, uno horizontal que varía entre $(+\alpha, -\alpha)$ y uno vertical que varía entre $(+\sigma, -\sigma)$ según el siguiente criterio de signos.

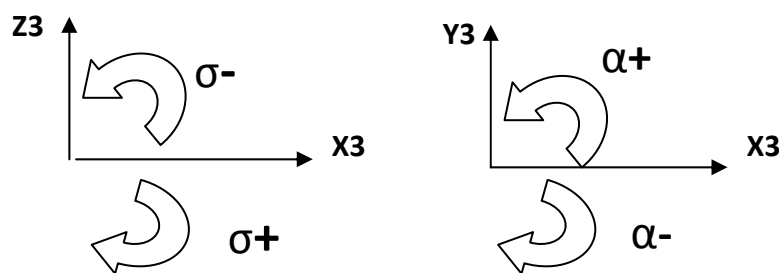


Figura 35. Criterio de signos en asignación de ángulos de barrido horizontal y vertical.

Teniendo en cuenta que *Lrf3dMeasure* representa una imagen en la que el valor de pixel describe la distancia al punto de impacto, tiene que la fila superior de la matriz corresponde al ángulo de barrido vertical más negativo y la fila inferior de la matriz corresponde al más positivo. Lo mismo se puede decir de la columna izquierda de la matriz que representaría el ángulo de barrido horizontal máximo mientras que la columna situada más a la derecha se representa el ángulo de barrido horizontal mínimo. De tal forma queda:

$$Lrf3dMeasure = \begin{pmatrix} \alpha_{MAX}, \sigma_{MIN} & \dots\dots\dots \alpha_{MIN}, \sigma_{MIN} \\ \alpha_{MAX}, \sigma_{MAX} & \dots\dots\dots \alpha_{MIN}, \sigma_{MAX} \end{pmatrix}$$

Figura 36. Variación de los barridos angulares vertical y horizontal en el interior de la matriz de medidas del laser.

Así, una vez que se posee la matriz con las medidas de las distancias a los puntos de colisión, se procede a recorrer dicha matriz elemento a elemento asociando a cada valor de distancia los correspondientes al ángulo α y σ según las siguientes expresiones:

$\alpha = \alpha_{MAX} - (j - 1) * RH$	(21)
$\sigma = \sigma_{MIN} + (i - 1) * RV$	(30)

Donde:

- α y σ son los ángulos del láser según las referencias establecidas.
- α_{MAX} es el ángulo de apertura horizontal máximo correspondiente a la mitad del ángulo total de barrido horizontal.
- σ_{MIN} es el ángulo de apertura vertical mínimo correspondiente a la mitad del ángulo total de barrido vertical.
- i, j son la fila y columna del elemento de la matriz del láser que se está procesando.
- RH, RV son las resoluciones angulares horizontal y vertical.

A continuación, se realiza la transformación de coordenadas esféricas a cartesianas según el sistema de referencia 3, centrado en la óptica del láser, según las siguientes expresiones:

$X_3 = d \cos(-\sigma) \cos(\alpha)$	(31)
$Y_3 = d \cos(-\sigma) \sin(\alpha)$	(32)
$Z_3 = d \sin(-\sigma)$	(33)

Posteriormente, se debe pasar desde el sistema de referencia 3 o sistema del láser hasta el sistema fijo para representar cada punto de impacto respecto al sistema de coordenadas fijo. Para ello, se cuenta con la información de los ángulos de Euler del sistema móvil (2) respecto al fijo (1) (actitud del robot) y se utiliza el cálculo matricial basado en transformaciones homogéneas.

En primer lugar se definen los ángulos de Euler que van a ser utilizados:

- Roll (φ), ángulo que representa un giro respecto al eje X_1 fijo.
- Pitch (θ), ángulo que representa un giro respecto al eje Y_1 fijo.
- Yaw (ψ), ángulo que representa un giro respecto al eje Z_1 fijo.

En segundo lugar se define el tipo de transformación utilizada:

$$\text{Vector sistema 1} = T_2^1 * R_2^1 * T_3^2 * \text{Vector sistema 3} \quad (34)$$

Estas operaciones son debidas a que para pasar del sistema 2 al 3 es necesaria una simple translación. Sin embargo, para pasar del 2 al sistema de referencia del mapa es necesaria una rotación y una posterior translación.

Estas operaciones en detalle serían:

$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & X_2 \\ 0 & 1 & 0 & Y_2 \\ 0 & 0 & 1 & Z_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} C(\psi) * C(\theta) & C(\psi) * S(\theta) * S(\varphi) - S(\psi) * C(\varphi) & C(\psi)S(\theta) * C(\varphi) + S(\psi) * S(\varphi) & 0 \\ S(\psi) * C(\theta) & S(\psi) * S(\theta) * S(\varphi) + C(\psi) * C(\varphi) & S(\psi) * S(\theta) * C(\varphi) - C(\psi) * S(\varphi) & 0 \\ -S(\theta) & C(\theta) * S(\varphi) & C(\theta) * C(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & X_3 \\ 0 & 1 & 0 & Y_3 \\ 0 & 0 & 1 & Z_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (35)$$

Una vez aplicada esta transformación matricial, se obtiene cada punto de impacto del láser respecto del sistema fijo y ya solo queda asignarlo a una determinada celda del mapa.

Las celdas se han numerado siguiendo los criterios de numeración de los elementos de una matriz. Como ejemplo consideraremos que el centro del mapa se encuentra en la casilla (3,3). En este caso, toda variación hacia la parte positiva del eje X irá aumentando el número de la columna, y toda variación en el sentido del eje Y positivo reducirá el número de fila de la matriz, tal y como describen las siguientes expresiones:

$j = X_c + \text{int}\left(\frac{\Delta x}{\text{Escala}}\right) \forall \Delta x > 0$	(36)
$j = X_c + \text{int}\left(\frac{\Delta x}{\text{Escala}} - 1\right) \forall \Delta x < 0$	(22)
$i = Y_c - \text{int}\left(\frac{\Delta y}{\text{Escala}}\right) \forall \Delta y > 0$	(23)
$i = Y_c - \text{int}\left(\frac{\Delta y}{\text{Escala}} - 1\right) \forall \Delta y < 0$	(24)

Donde:

- Δx es la distancia del punto de colisión al centro del mapa en dirección X_1 .
- Δy es la distancia del punto de colisión al centro del mapa en dirección Y_1 .
- Escala es la dimensión de la celda en metros.
- Los valores j e i representan en números enteros la columna y la fila respectivamente de cada celda.

4.9. ESTIMACIÓN DE ALTURAS Y GRADIENTES

Una vez obtenidos los puntos de colisión y la determinación de la celda a la que corresponden, éstos son almacenados en una lista de longitud variable para cada celda, dicha longitud dependerá del número de puntos de colisión que hayan “caído” en dicha celda.

El método para el cálculo de alturas y gradientes utilizado deriva de [12] y se basa en calcular el plano que mejor aproxima la nube de puntos que hay en cada celda. Básicamente aplica el concepto de regresión lineal plana, obtener el plano que aproxima la nube de puntos, la altura del centro de ese plano corresponderá a la altura media de los puntos lo que se tomará como altura de la celda.

Así, si la ecuación del plano se expresa de la forma:

$Z = a * X + b * Y + c$	(40)
-------------------------	------

Donde Z es la altura de la celda y X e Y son las coordenadas en el plano del mapa. De esta forma el gradiente según X será $G_x = \frac{\partial Z}{\partial X} = a$ y el gradiente según Y será a su vez $G_y = \frac{\partial Z}{\partial Y} = b$. Por lo que bastará calcular los coeficientes de dicha ecuación del plano para obtener los gradientes.

Se trata pues de buscar el hiperplano que mejor se ajuste, para ello, uno de los criterios que se pueden utilizar es minimizar la suma de errores o residuos al cuadrado.

Según el modelo de regresión múltiple expresado por:

$Y_i = \beta_0 + \beta_1 * X_{1i} + \beta_2 * X_{2i} + \dots + \beta_k * X_{ki} + u_i$	(41)
--	------

Donde $u_i \rightarrow N(0, \sigma^2)$ y $\beta_0, \beta_1, \beta_2, \dots, \beta_k, \sigma^2$ son parámetros desconocidos a estimar.

Expresándolo en notación matricial:

$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{21} & \dots & x_{k1} \\ 1 & x_{12} & x_{22} & \dots & x_{k2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{1n} & x_{2n} & \dots & x_{kn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$	(42)
$Y = X\beta + U$	(43)
$U \rightarrow N(0, \sigma^2 I)$	(44)

En el caso que nos ocupa, el vector de variables dependientes son el vector de alturas de cada celda, y el resto de variables independientes serían las coordenadas X e Y .

Para la estimación por mínimos cuadrados del sistema:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ 1 & \cdot & \cdot & \cdots & \cdot \\ 1 & \cdot & \cdot & \cdots & \cdot \\ 1 & \cdot & \cdot & \cdots & \cdot \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \cdot \\ \cdot \\ \cdot \\ \hat{\beta}_k \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ \cdot \\ e_n \end{bmatrix} \quad (45)$$

$$Y = X\hat{\beta} + e \quad (46)$$

Donde el vector e se define según (47)

$$\|e\|^2 = \sum_{i=1}^n e_i^2 \quad (47)$$

Para que $\|e\|^2$ sea mínimo, e tiene que ser perpendicular al espacio vectorial generado por las columnas de X .

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ 1 & \cdot & \cdot & \cdots & \cdot \\ 1 & \cdot & \cdot & \cdots & \cdot \\ 1 & \cdot & \cdot & \cdots & \cdot \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ \cdot \\ e_n \end{bmatrix} \quad (48)$$

Cumpléndose:

$$X^T e = 0 \quad (49)$$

$$X^T Y = X^T X \hat{\beta} + X^T e \quad (50)$$

$$X^T Y = X^T X \hat{\beta} \rightarrow \hat{\beta} = (X^T X)^{-1} X^T Y \quad (51)$$

Por lo tanto, los valores β_1 y β_2 corresponden directamente a los gradientes en sentido X y en sentido Y respectivamente.

Como ejemplo, la Figura 37 muestra una trayectoria con puntos de captura de datos o reconstrucción distanciados 50 m según coordenadas X e Y . En una situación real, el robot es capaz de realizar varias reconstrucciones por segundo, por ello se prevé que se obtenga información con puntos muy poco distanciados.

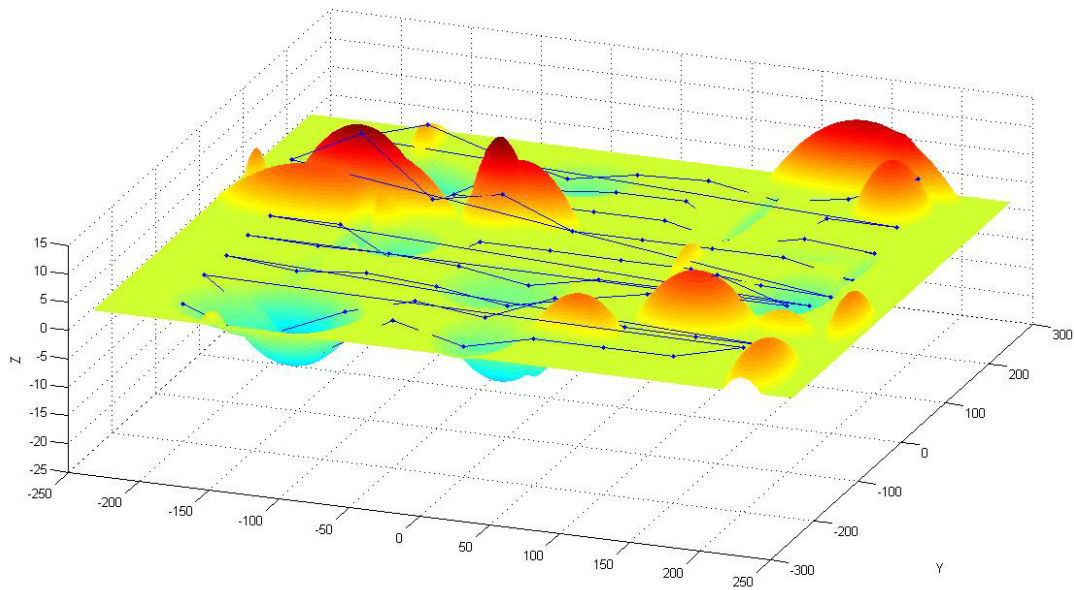


Figura 37. Ejemplo de trayectoria con indicación de los puntos de reconstrucción.

Los resultados de la reconstrucción de alturas para este caso se muestran en la siguiente figura:

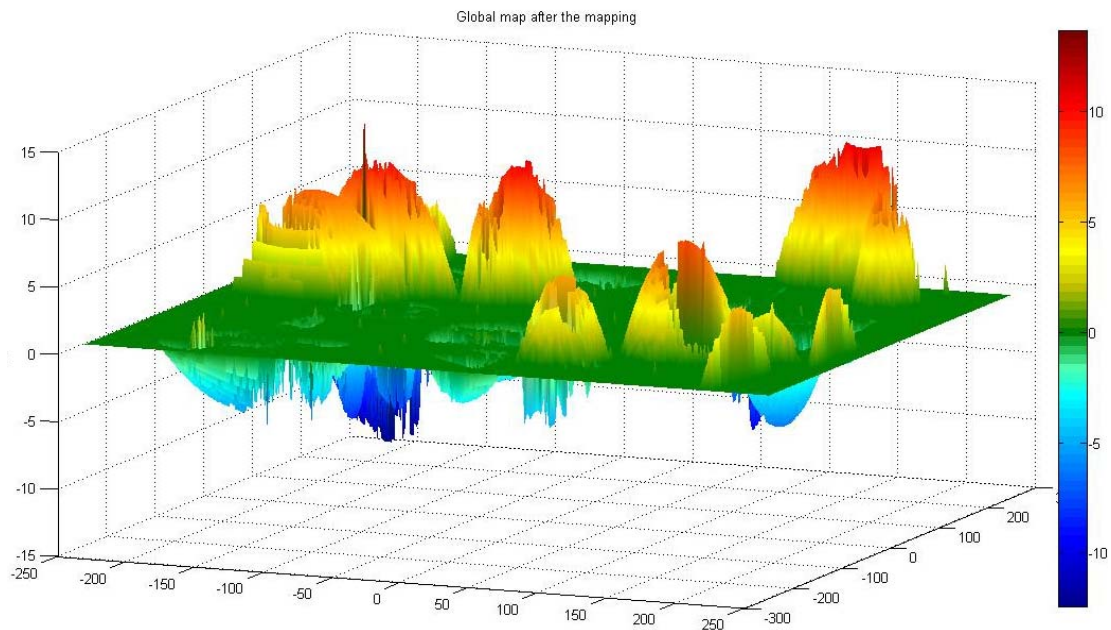


Figura 38. Ejemplo de reconstrucción.

A simple vista, se puede apreciar que son resultados bastante satisfactorios. Sin embargo es mejor apreciarlo desde un punto de vista de planta cuantificando los errores de la reconstrucción respecto al mapa teórico ideal.

Para ello, se han dividido los errores en intervalos discretizando los valores para una mejor observación (ver Figura 39). En ella, se puede apreciar como en los lugares cercanos a la posición del robot los errores son evidentemente muy bajos, del orden de 1 m y según aumente la distancia del punto al robot el error va creciendo. De hecho existen zonas como los picos de las montañas o los mínimos de los valles donde el láser no ha recogido información ninguna, en esos puntos el error es grande.

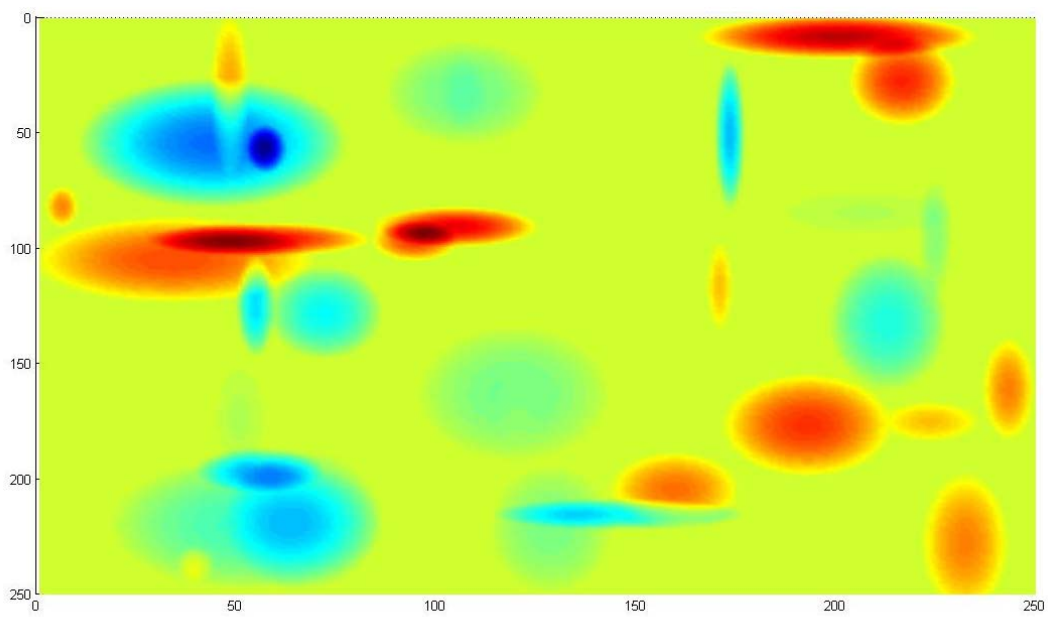


Figura 39. Discretización de la reconstrucción.

El error de reconstrucción correspondiente a la Figura 39 se representa en la Figura 40. En ella, se puede observar un error aceptable como ya se ha comentado en casi todos los lugares excepto en las zonas cuya visibilidad se ha visto muy reducida.

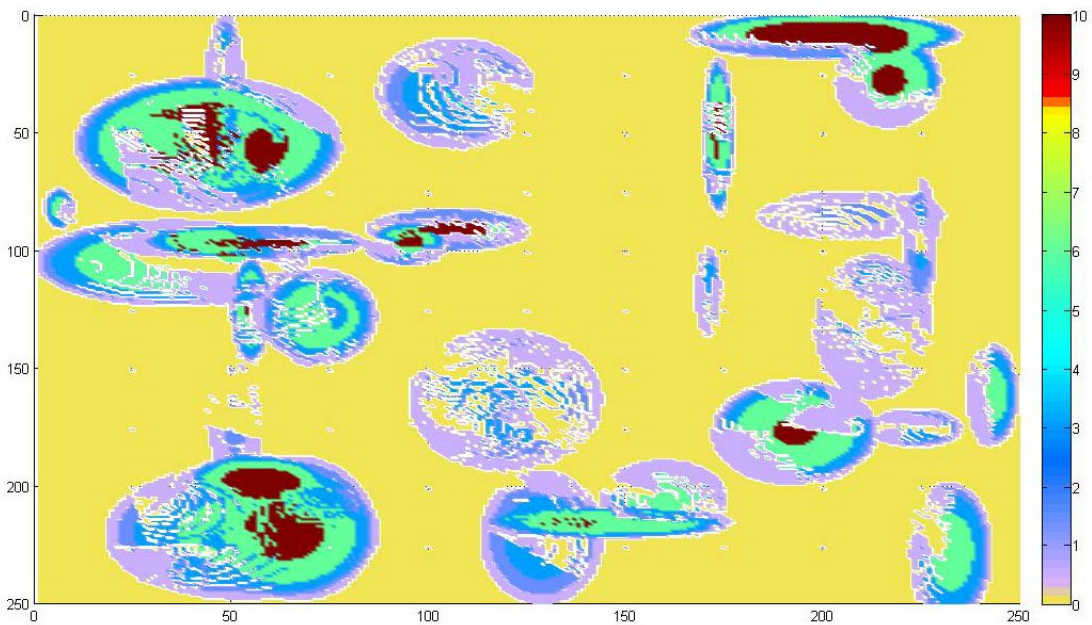


Figura 40. Error de reconstrucción discretizado.

Para el cálculo del hiperplano de regresión es necesario que al menos caigan 3 puntos en una misma celda. Dado que este algoritmo calcula un plano para cada celda por cada paso de simulación y luego todos los datos son reinicializados, es necesario que ocurra esa condición para una misma matriz de lectura del láser. Esto no siempre ocurre así y es habitual encontrar un gran número de celdas que no tienen 3 puntos de impacto. Por ello en esas celdas no se puede calcular el hiperplano de regresión y se han buscado otros métodos de cálculo del gradiente basados en la información de las celdas vecinas, consistente en el cálculo a nivel celdas utilizando las alturas, como muestran las siguientes expresiones:

$GX_{ij} = \frac{Z_{j+1} - Z_j}{escala}$	(52)
$GY_{ij} = \frac{Z_i - Z_{i-1}}{escala}$	(53)

Donde:

- GX_{ij} es el gradiente en dirección X del elemento (i, j) .
- GY_{ij} es el gradiente en dirección Y del elemento (i, j) .
- $escala$ es el valor del lado de la celda.

Este método denominado “cálculo del gradiente por filas y columnas” da resultados muy aproximados y es válido para realizar los cálculos sin mucha precisión, pero en cualquier caso ha sido el método que mejores resultados ha aportado si bien el valor del gradiente calculado no está centrado en la posición de la celda en la que se asigna. Es decir, el gradiente se encuentra desplazado.

La adaptación del método de cálculo de gradiente por filas y columnas consiste en utilizar cuatro celdas adyacentes de forma simultánea y calcular los gradientes en ambas direcciones utilizando los valores de las alturas de las cuatro celdas. Esto soluciona el problema cuando se tiene una acumulación de puntos cercanos pero con menos de 3 impactos por celda.

Como mejora del método anteriormente descrito, se procedió al cálculo de la regresión utilizando varias celdas a la vez y estimando el mismo valor para todas ellas tal y como se describe a continuación.

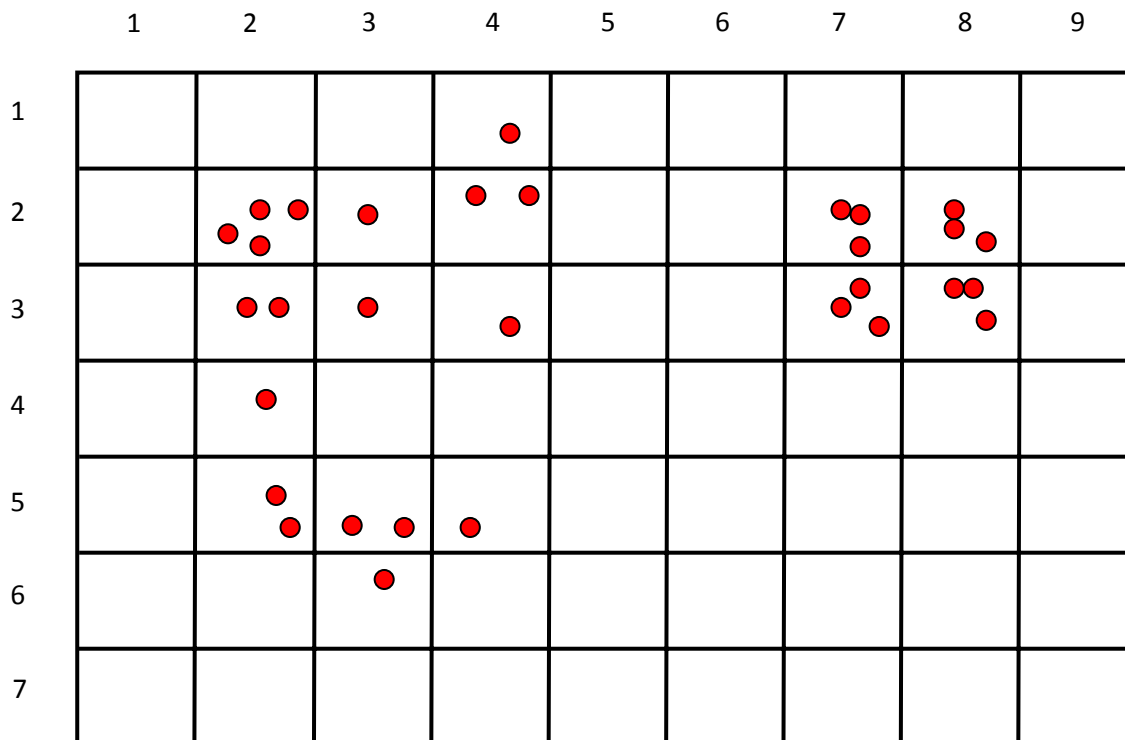


Figura 41. Ejemplo de asignación de puntos de colisión para el cálculo de gradientes.

En primer lugar se calculan los gradientes de todas las celdas donde el número de puntos de impacto es igual o superior a 3. Es decir, se calcularían los gradientes de las celdas (2, 2), (2, 7), (2, 8), (3, 7) y (3, 8).

En segundo lugar se procede a recorrer el mapa desde la celda (1, 1) en grupos de cuatro calculando los planos de regresión como si el grupo de cuatro celdas fuesen solo una y rellenando información solo para las celdas que no tengan ya calculado el gradiente. En este caso se calcularían los gradientes para los grupos de celdas de la forma siguiente:

- Celdas (1, 1), (1, 2), (2, 1) y (2, 2) donde todas tomarían el valor del gradiente calculado en (2, 2)
- Celdas (1, 3), (1, 4), (2, 3) y (2, 4) donde se calcularían los gradientes según X e Y y se tomarían igual para las cuatro celdas puesto que ninguna de ellas tiene más de dos puntos de impacto.
- Celdas (1, 7), (1, 8), (2, 7) y (2, 8). Como las celdas (2, 7) y (2, 8) ya tienen un valor calculado, este se mantiene pero la información de los 6 puntos de impacto se utiliza para calcular los gradientes de las otras dos celdas que no poseen información.

- Celdas (3, 1), (3, 2), (4, 1) y (4, 2) se calcularía un único gradiente común para todas.
- Celdas (3, 3), (3, 4), (4, 3) y (4, 4). No se posee información suficiente para el cálculo de un plano puesto que solo hay 2 puntos de impacto. Lo mismo ocurre para el siguiente grupo de cuatro celdas.
- Celdas (3, 7), (3, 8), (4, 7) y (4, 8). Se calcularía el gradiente con los 6 puntos de impacto y únicamente se actualizaría el valor de las celdas de la fila 4.
- Celdas (5, 1), (5, 2), (6,1) y (6, 2). No se posee información suficiente.
- Celdas (5, 3), (5, 4), (6, 3) y (6, 4). Se calcularían los gradientes con los 4 puntos de impacto y su valor se añadiría a las 4 celdas que forman la cuadrícula.

4.10. ASIGNACIÓN DE CONFIANZAS

Con el concepto “confianza” se desea reflejar el nivel de seguridad que se posee sobre la información que alberga una celda. Esta seguridad va relacionada íntimamente con el número de puntos de impacto del láser en su interior y el tiempo transcurrido desde su reconstrucción, ya que se trata de entornos con objetos dinámicos.

Debido a la forma de un entorno y la posición de un robot dentro del mismo, es posible que se produzcan un elevado número de puntos de impacto en algunas celdas y muy pocos o ninguno en otras muy cercanas. Adicionalmente, y dependiendo del tamaño de celdas utilizadas, es probable que los puntos de impacto dentro de una misma celda estén muy concentrados en una zona determinada y no haya ninguno en el resto del área. Por estas razones, la confianza en el valor de una celda va relacionada no sólo con el número de puntos de impacto que caigan sobre ella, sino también con su dispersión dentro de la celda.

Para evaluar la dispersión se han valorado diferentes alternativas, sobre todo desde el punto de vista de coste computacional, (tiempo de ejecución y requisitos de memoria). Entre ellas está calcular las coordenadas del punto medio de impacto y tomar la dispersión como la desviación típica y otros métodos estadísticos. Finalmente, el algoritmo más rápido y de menor requerimiento de memoria ha sido el algoritmo basado en la creación de subceldas.

Este método, consiste en dividir cada celda en “n” subceldas, tal y como muestra la Figura 42 donde la celda se ha dividido en 16 partes.










1 	2  	 3  	4
5	6	  7	 8
9	10	11	12
13	14	15	16

Figura 42. División en subceldas.

El algoritmo de asignación de confianzas calcula la subcelda asociada a cada punto de impacto mediante la lista de los puntos de impacto de una celda con sus coordenadas. A continuación, se cuentan las subceldas en las que al menos hay un punto de colisión. La confianza se calcula dividiendo el número de subceldas con puntos de colisión entre n con objeto de obtener un valor que varíe ente 0 y 1. El valor de n elegido durante las pruebas ha sido 16, este número deberá ser mayor cuanto mayor será el tamaño de la celda.

Como se ha visto, con este procedimiento se tiene en cuenta la distribución de los puntos de impacto y se calcula la confianza en base a dicha distribución, pero además de este criterio, se deben tener en cuenta otros parámetros como son las distancias del láser a los puntos de impacto.

Debido a los errores de medida del láser y más aun a los errores en la estimación de la orientación (alabeo, cabeceo y guiñada), los errores de medida en las distancias crecen al aumentar la distancia de colisión. Por ello, es útil penalizar la confianza de los puntos de impacto lejanos para proporcionar una incertidumbre sobre los mismos de tal forma que no tenga la misma credibilidad un impacto del láser a 5 metros que otro a 70 metros por ejemplo. Para ello se ha aplicado una función correctora de la siguiente forma:

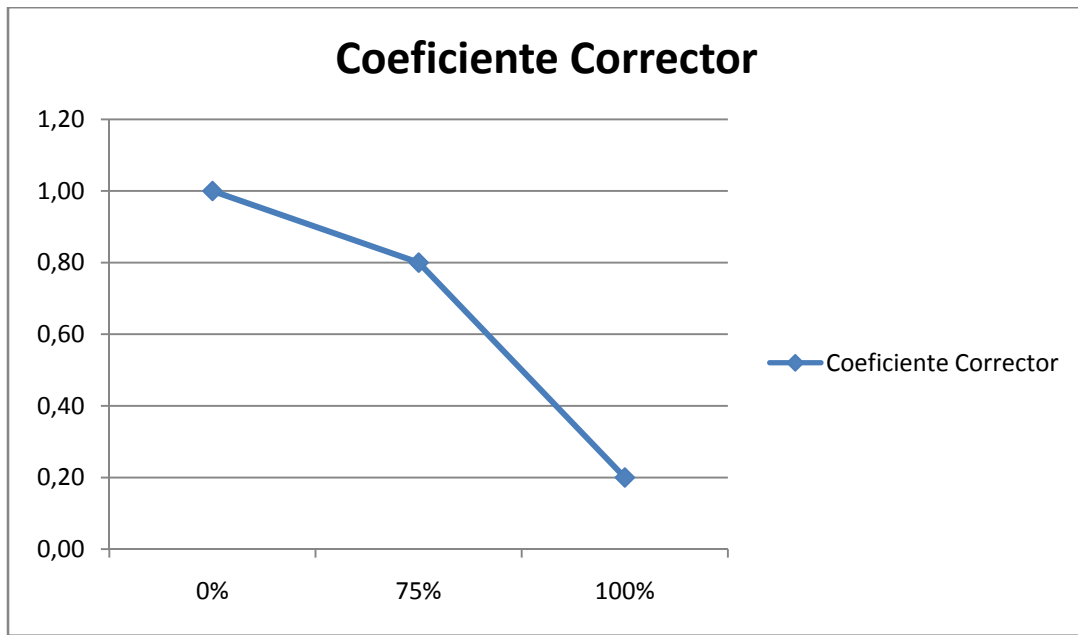


Figura 43. Coeficiente corrector en función de la distancia medida.

Así, para valores de distancia medida del láser pequeños (menores del 75% del alcance máximo) se aplica un factor poco penalizador y para medidas de distancia de valores elevados (mayores al 75% del alcance máximo) se aplica un factor penalizador más elevado. Esta forma de corrección está basada en que el error de medida del láser no depende de la distancia medida, pero sin embargo, puntos de colisión muy lejanos se ven claramente afectados por pequeños errores en la estimación de la actitud del robot.

El coeficiente corrector mostrado en la Figura 43, se puede expresar matemáticamente mediante las siguientes expresiones:

$$f = \frac{-0.2 D}{0.75 \text{ RangoMax}} + 1 \quad \text{si } D < 75\% \text{RangoMax} \quad (54)$$

$$f = \frac{-0.6(D - 0.75 \text{ RangoMax})}{0.20 \text{ RangoMax}} + 0.8 \quad \text{si } D > 75\% \text{RangoMax} \quad (55)$$

El diagrama de flujo de la asignación de confianzas sería por tanto el siguiente:

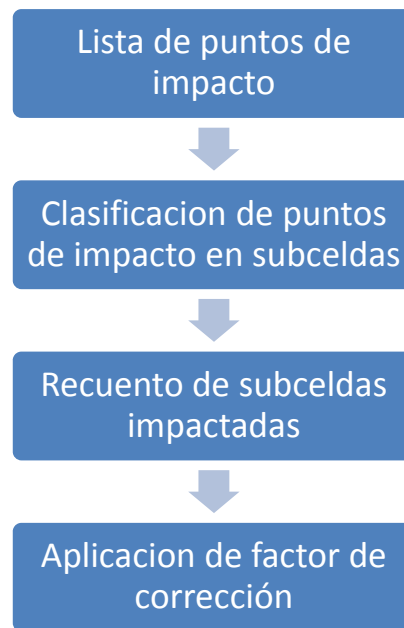


Figura 44. Proceso de asignación de confianzas

La Figura 45 muestra de la confianza calculada mediante la simulación de una trayectoria del robot por un camino montañoso (Figura 37):

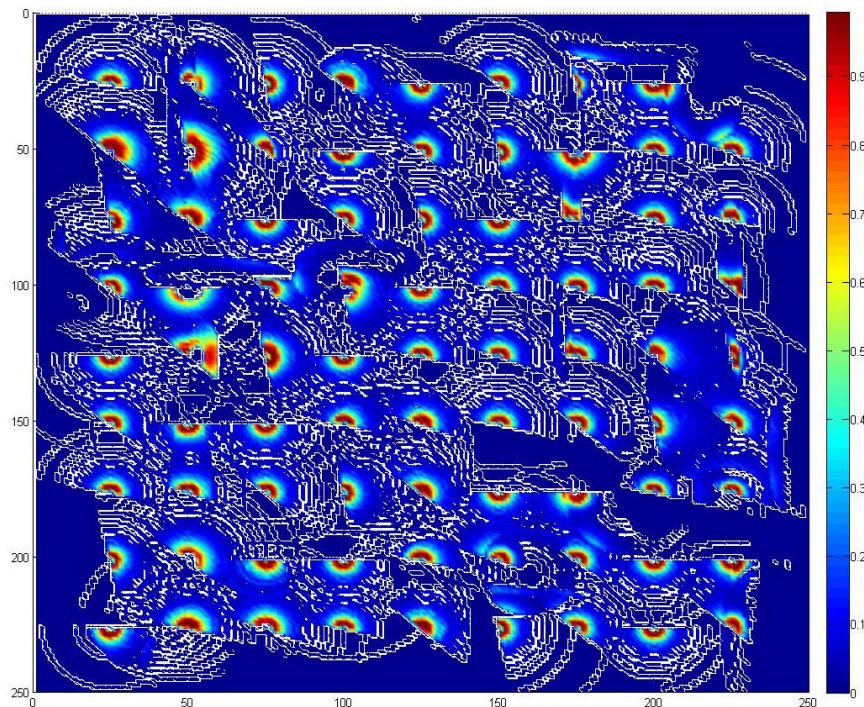


Figura 45. Asignación de confianzas

La confianza se muestra alta para las zonas cercanas a la posición de simulación del robot en cada paso y se muestran bajas cuanto más se aleja de dichas posiciones.

Habitualmente, en las siguientes reconstrucciones se puede obtener una confianza diferente para una celda. En apartados posteriores se describirán los procedimientos y algoritmos para la fusión de dicha información.

Otros métodos probados para estimar la confianza

Inicialmente, se desarrolló otro procedimiento para la asignación de confianzas, pero este último basado en subceldas es el que mejores resultados ha aportado.

El método inicial se basaba en un recuento global del número de puntos impactados en una celda, sin tener en cuenta la dispersión de los mismos. El recuento era acumulativo, de forma que se sumaban los puntos de impacto de toda la simulación, acumulando los de medidas sucesivas del láser y la confianza se variaba de forma exponencial con asíntota en 1. De forma esquemática se procedía de la siguiente forma:

Comienzo de la lectura de la matriz del láser, cálculo de los puntos de impacto para cada celda y suma del número de puntos totales.

Cálculo de la confianza mediante:

$C = 1 - e^{-n/\tau}$	(56)
-----------------------	------

Donde n corresponde al número de impactos sobre cada celda y τ la constante de crecimiento que es un parámetro a ajustar. Para calcular un valor aceptable de la constante de crecimiento de la confianza se estimaba el número mínimo de impactos necesarios para considerar una confianza máxima sobre la celda y de forma que para n igual a 3, la confianza fuese del 95 por ciento. Este método conlleva varias desventajas respecto al elegido definitivamente:

- La obligatoriedad de llevar una cuenta constante de los puntos de impacto.
- La suma acumulada de puntos de impacto implicaba usar funciones de conversión logarítmicas para calcular al final de cada ciclo el número de puntos de impacto acumulados para poder sumarlo al número de puntos de impacto

en la lectura nueva del láser. Ya que las confianzas al proceder de una exponencial del numero de impactos no se pueden sumar directamente, sino que había que calcular el número de puntos de impacto acumulados mediante la confianza acumulada, sumar con el nuevo número de impactos y volver a aplicar la exponencial.

- El cálculo de la constante de crecimiento de la exponencial resultaba algo arbitrario.
- No se tenía en cuenta la dispersión de los puntos.

4.11. Asignación de ocupación

La ocupación de una celda mide la dificultad que tiene el robot para viajar por ella, teniendo en cuenta su pendiente. El nivel de ocupación de la celda influye en el momento de la planificación del recorrido que el robot debe realizar, eligiendo por ejemplo el camino de menor dificultad entre varios que lleguen al mismo sitio.

Para calcular la pendiente de la celda se parte del gradiente de la misma, que representa la tangente del terreno según direcciones X e Y .

Como primera aproximación al cálculo de la ocupación se ha designado un valor de 1 (ocupada) a aquellas celdas cuyo módulo del gradiente sea de 0.3 (máxima pendiente que puede traspasar un robot en la simulación).

Para las celdas cuyo módulo del gradiente sea inferior, se calculará la ocupación mediante una función lineal que aumenta la ocupación desde módulo del gradiente nulo hasta módulo igual al valor máximo admisible.

Este método, a pesar de ser extremadamente sencillo es eficaz para su utilización en la planificación.

Se puede no obstante utilizar otros criterios, como basarse en el valor del gradiente en ambas direcciones y cuando alguna de ellas supere un valor crítico considerar la celda como ocupada. En cualquier caso es una decisión fuertemente ligada al algoritmo de planificación de trayectorias utilizado.

Como muestra, el ejemplo de la trayectoria ya mostrada en partes anteriores del presente proyecto.

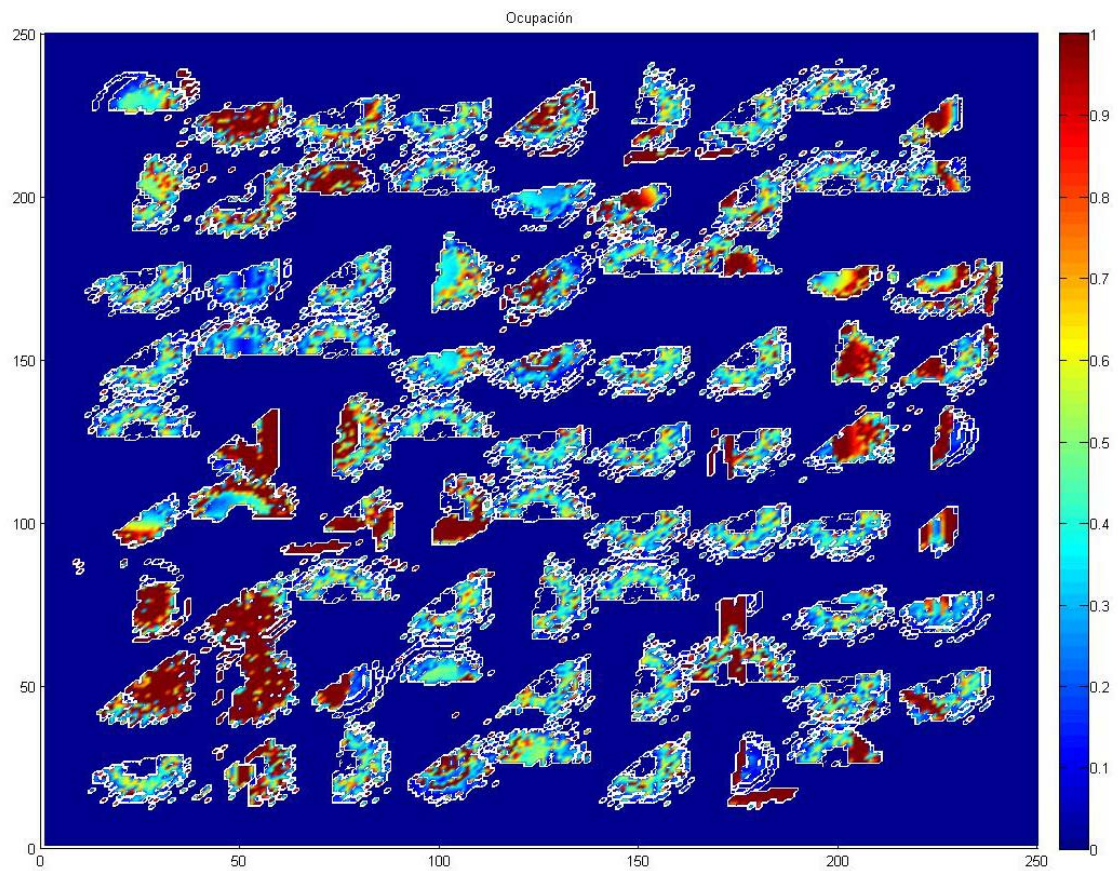


Figura 46. Reconstrucción de la ocupación de un terreno

Se puede observar el mapa de la

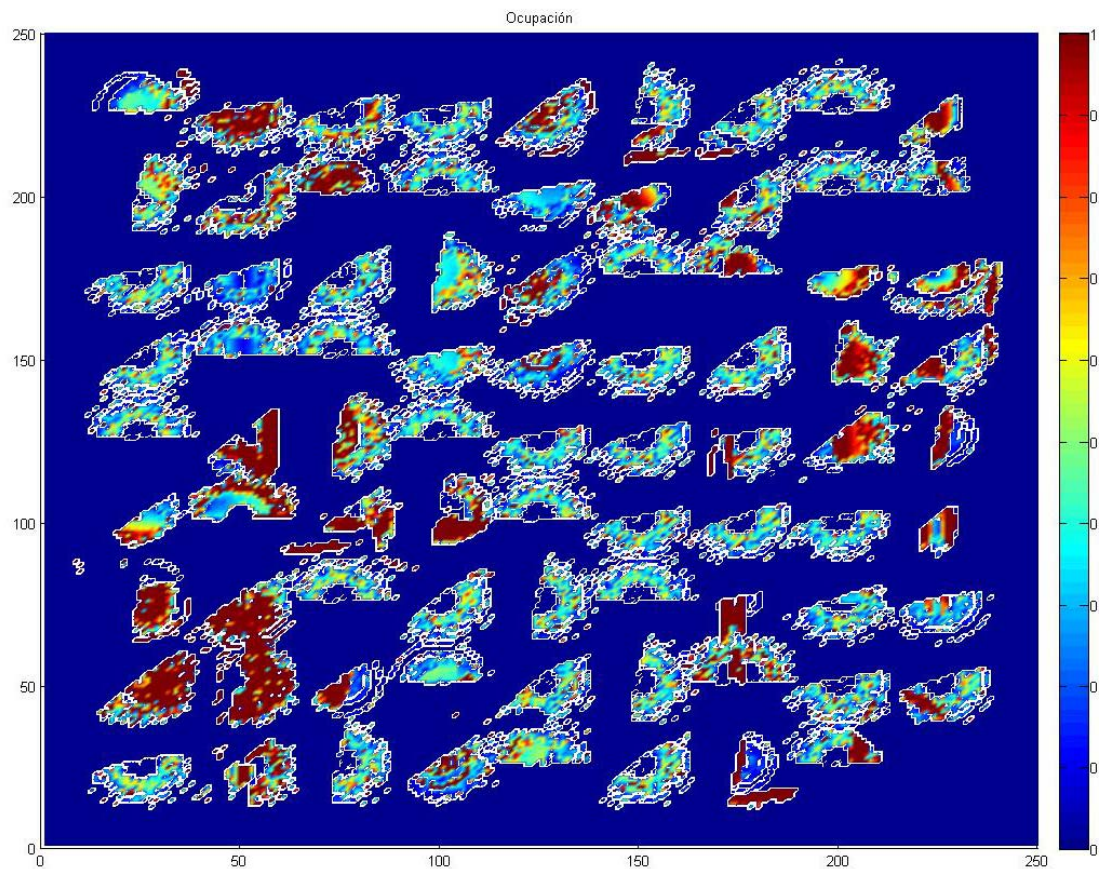


Figura 46 formado por semicírculos concéntricos distribuidos de manera más o menos uniforme. Dichos semicírculos corresponden a los impactos del láser sobre el terreno. El fondo indica las zonas de las que no se posee información sobre la pendiente del terreno y el módulo del gradiente viene coloreado según la escala situada a la derecha.

A continuación se van a mostrar unas pruebas sobre un recorrido nuevo realizado en una distancia corta y a pequeña distancia entre cada punto de lectura del láser (Figura 47). Se puede apreciar unos puntos azules entre una montaña grande, una pequeña y un valle también grande. La trayectoria tiene una longitud de 10 m y con una distancia de 0.5 m entre punto y punto. Como las distancias son grandes, solo merece la pena centrarse en las zonas donde el láser tiene alcance por lo que se han eliminado de las imágenes sobre la ocupación las zonas sin importancia. Aun así conviene centrarse en la parte central de la imagen que es donde se recopilado mayor información, por ello el error al comparar la reconstrucción con los resultados teóricos presenta allí la menor cuantía.

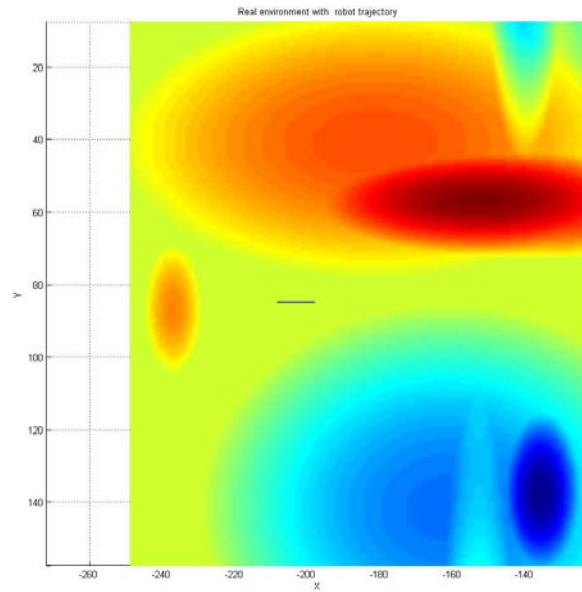


Figura 47. Trayectoria simulada mediante puntos muy cercanos entre si

En esta imagen (Figura 47), se puede observar la trayectoria rectilínea del robot, mientras que en la siguiente imagen se tiene la ocupación teórica en base al módulo del gradiente. Se puede observar como la ocupación es mínima en el suelo y en determinados puntos de las parábolas que componen el relieve alcanza el valor máximo. Es de esperar que en la zona donde el robot esta transitando los valores reconstruidos sean muy parecidos a los valores teóricos y con ello el error se minimice. Pero en cambio, en las zonas alejadas donde la visibilidad del láser es reducida el error puede ser mayor.

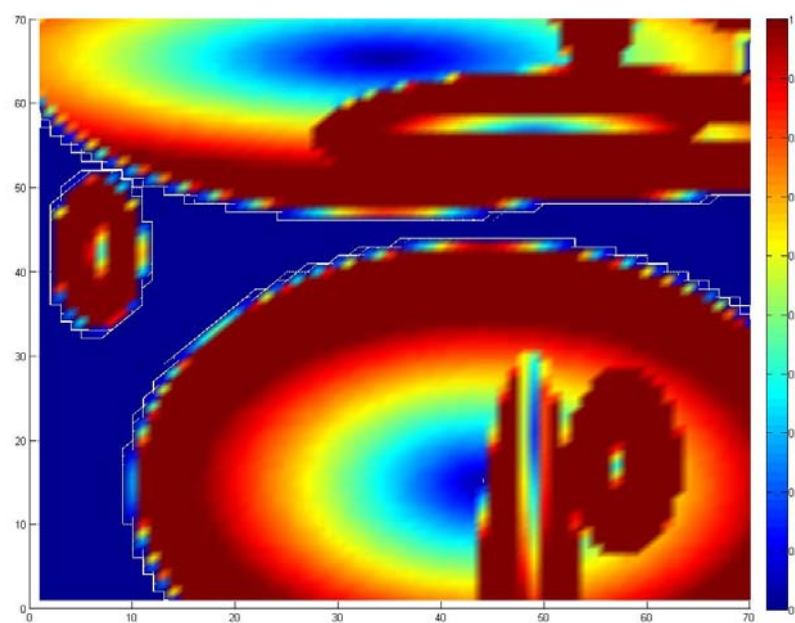


Figura 48. Ocupación teórica dados unos gradientes conocidos

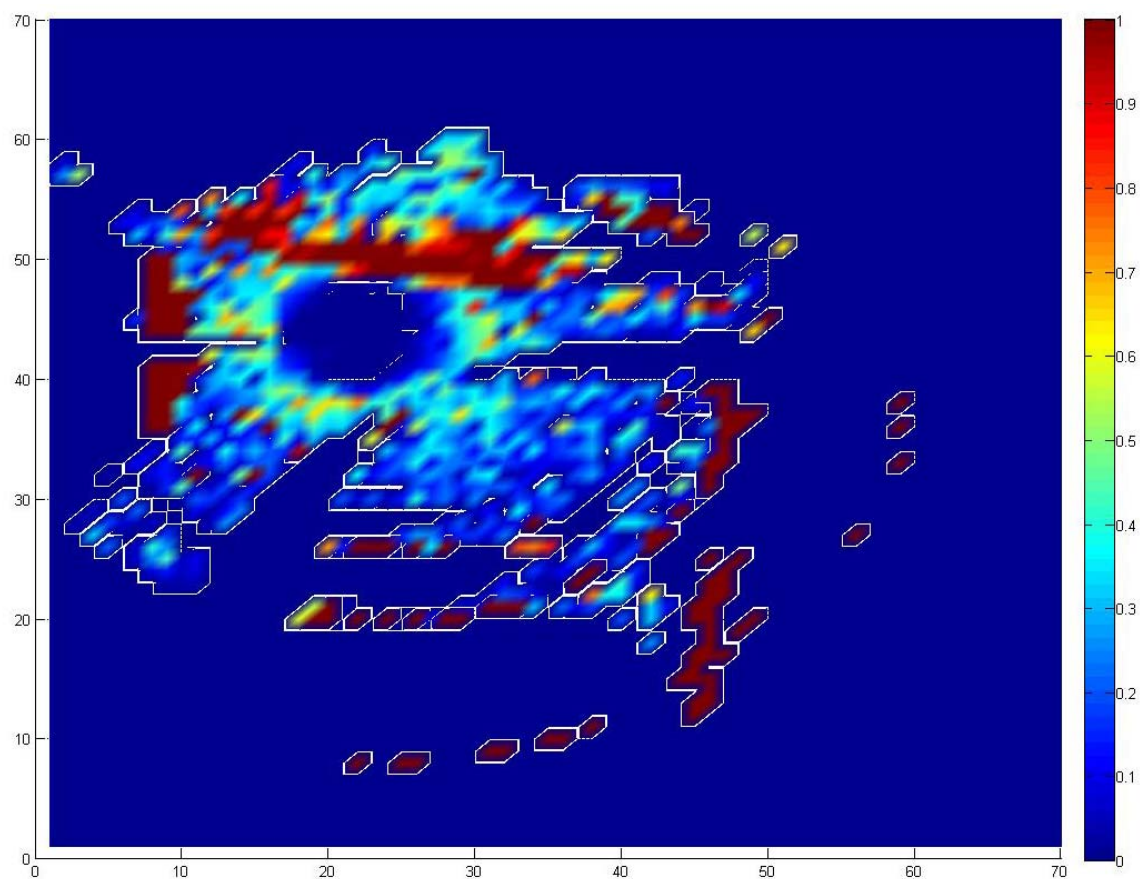


Figura 49. Reconstrucción de la ocupación para la trayectoria de la Figura 47

En la Figura 48 se observa la ocupación teórica calculada conociendo todos los gradientes del área analizada y en la Figura 49 se puede apreciar la ocupación calculada utilizando las alturas reconstruidas.

A continuación se puede observar el error obtenido al comparar ambos cálculos.

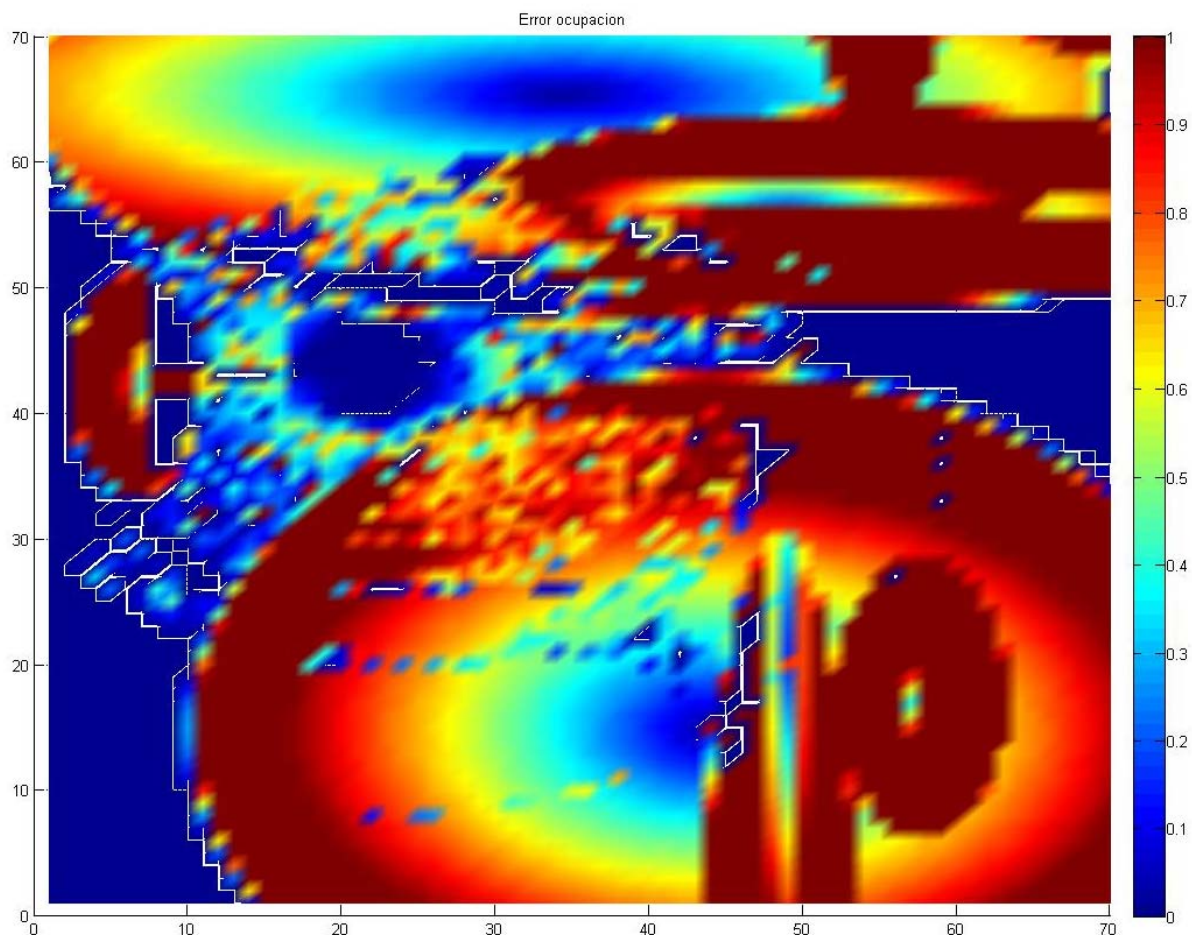


Figura 50. Error en valor absoluto de la reconstrucción de la ocupación

Como se puede observar, los resultados son buenos en las cercanías al robot. En las zonas lejanas pero con ocupación igual a la unidad el error es nulo, en cambio en las zonas donde la ocupación es inferior a la unidad hay mayor error. Esto se puede deber al método del cálculo de los gradientes ya que como se ha mencionado anteriormente, no se utiliza el mismo método para las celdas en las que hay puntos de colisión como en las que no los hay. No obstante, al tratarse de puntos alejados, la influencia del error es mucho menor ya que este se verá reducido cuando el robot se acerque a ellos.

4.12. LIBERTAD E INFORMACIÓN DE PUNTOS DE NO IMPACTO

Los algoritmos que aquí se presentan usan la información que pueden aportar los rayos del láser que pasan suficientemente cerca del suelo como para poder afirmar que el terreno por donde ha pasado esta libre de obstáculos.

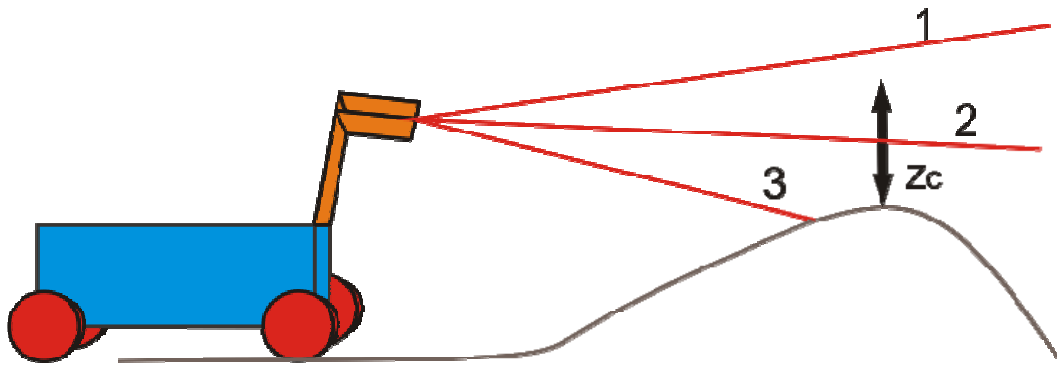


Figura 51. Representación de las posibles trayectorias de impacto del laser.

Cuando el láser del robot lanza un rayo láser pueden ocurrir fundamentalmente tres casos:

- El rayo no impacta sobre ningún objeto y alcanza la distancia máxima sin pasar en la cercanía de ninguna superficie.
- El rayo no impacta sobre ninguna superficie pero pasa por la cercanía de la superficie en su trayectoria.
- El rayo impacta sobre una superficie y evidentemente tiene que aproximarse en algún momento a dicha superficie.

Se llamara altura crítica () a la altura que se va a considerar cercana a la superficie. Esta altura puede estar relacionada con la altura del obstáculo móvil a detectar o con la altura del obstáculo fijo máxima admisible para ser superada por el robot.

Con la asignación de libertad o no ocupación se pretende obtener información de la mayor longitud posible del haz del láser, no únicamente del punto de impacto. Para tener una idea del aumento de información que podría suponer se puede estimar suponiendo que el robot lanza un rayo perfectamente paralelo al suelo, y que este rayo impacta con un objeto lejano a la distancia máxima de alcance del láser, suponemos 80 m. Si se tomara información del láser cada Δ de la longitud de la celda, se estaría

tomando puntos de información adicional para la reconstrucción del terreno. Con lo cual se aumenta considerablemente (De 1 a 400 veces) la información proporcionada por el láser. Este cálculo está realizado evidentemente para unas condiciones óptimas de disparo del láser en el que el rayo es permanentemente paralelo al suelo, eso no sucederá salvo contadas ocasiones pero es una buena forma de comprobar las capacidades máximas de este método para aprovechar la información normalmente no utilizada.

Metodología para la obtención de información adicional

Para la obtención de información adicional se propone un algoritmo basado en bucle que recorre el vector formado por el láser en incrementos de longitud parametrizados según la longitud de la celda. Se propone dividir la longitud de la celda entre 5 o 10 partes. De esta forma para punto del recorrido se puede calcular las coordenadas de dicho punto respecto al sistema fijo realizando las mismas transformaciones que se utilizaban para calcular el punto de impacto en la reconstrucción de las alturas.

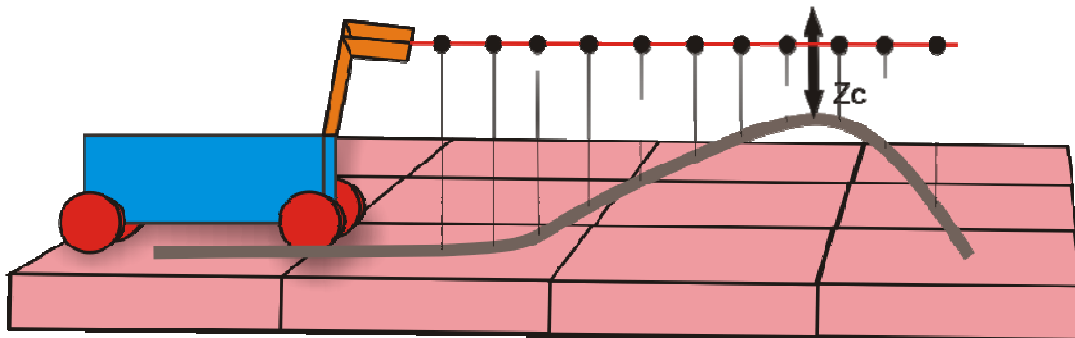


Figura 52. Concepto de altura crítica para la obtención de información adicional.

De esta forma se obtendrían las coordenadas de una serie de puntos resultantes de dividir el rayo en el número elegido de segmentos. Posteriormente se pasará a comparar la altura de cada punto con la altura que se posee previamente sobre la información del suelo. Conviene recordar que aunque el robot puede comenzar a reconstruir el terreno partiendo del mapa vacío, una funcionalidad importante del sistema es corregir un mapa que ya se tiene del terreno. Por ello es posible partir de un conocimiento más o menos fiable del terreno que permitirá realizar la comparación de alturas. La comparación de alturas trata de descartar todos los puntos cuya altura sea superior a la crítica puesto que se considerará que no aportan información al estar demasiado separados de la superficie. En el caso dibujado es posible que solo 3 o 4 puntos cumplieran la condición mencionada. En ese caso se utilizarán las

coordenadas de dichos puntos para dar a la celda correspondiente un aumento de la probabilidad de no ocupación.

El algoritmo que se utilizaría sería el siguiente:

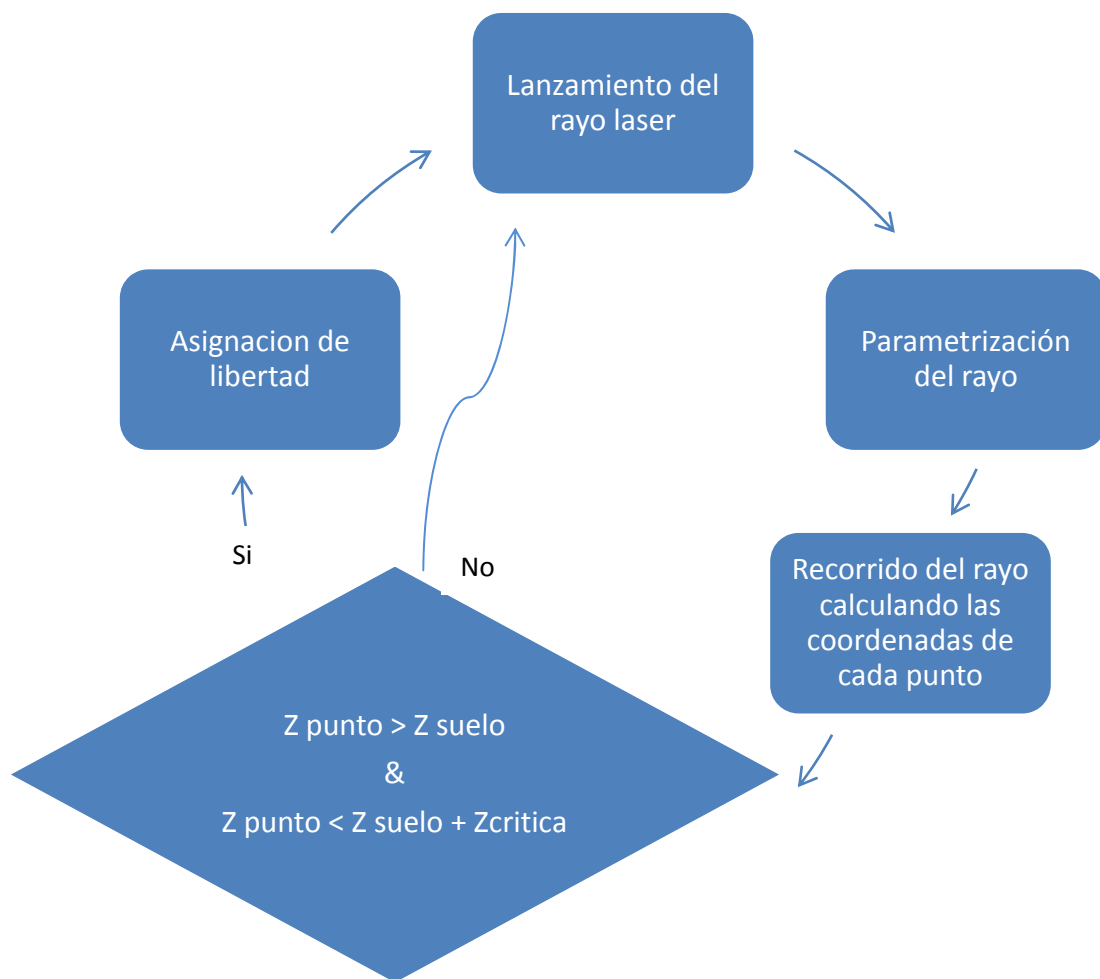


Figura 53. Esquema del algoritmo de asignación de libertades

Análisis del rendimiento en la ejecución

Si se analiza la ejecución del software corriendo esta característica de asignación de libertades se aprecia un notable aumento del tiempo de ejecución. Y son los subprogramas encargados de proporcionar esta funcionalidad los que consumen más tiempo. Por lo tanto, se puede afirmar que al añadir la funcionalidad de asignación de libertades aumenta considerablemente el tiempo de ejecución, mientras que es probable que la utilidad de los resultados conseguidos no aumente de la misma forma.

Función	Llamadas	Tiempo Total	Tiempo propio
SimDataMapping	1	944.382 s	3.319 s
SimLáser	5	524.629 s	180.254 s
M45DataMapping	5	419.061 s	6.248 s
Roots	2717715	341.375 s	341.375 s
FreedomAssignment	43383	324.968 s	95.573 s
Local2FixedCoord	1801875	136.363 s	136.363 s
Coordinates2Cell	1801890	98.985 s	59.986 s
QuadsNumber	10	68.961 s	35.119 s
MapCenterIndex	2377899	51.370 s	51.370 s
ConfidenceAssignment	5	33.841 s	0.561 s
Cell2Coordinates	576009	33.841 s	21.470 s
Heigh and Gradient Estimation	5	3.958 s	3.506 s
LowResMap fusión	5	3.755 s	3.272 s
Create LRM	5	2.119 s	2.119 s

Create HRM	5	1.652 s	1.636 s
Layers Fusion	10	0.982 s	0.966 s
FusionTemporal2Local	5	0.499 s	0 s
Spherical2Cartesian	43385	0.390 s	0.390 s

Subprogramas principales utilizados en la simulación del entorno:

- SimDataMapping: 3.319 s → 0.63 %
- SimLáser: 180.254 s → 34.33 %
- Roots: 341.375 → 65.03 %

Subprogramas principales utilizados para la funcionalidad real

- M45DataMapping: 1.52 %
- FreedomAssignment: 23.02 %
- Local2FixedCoord: 32.85 %
- Coordinates2Cell: 14.4 %
- QuadsNumber: 8.46 %
- MapCenterIndex: 12.37 %
- ConfidenceAssignment: 0 %
- Cell2Coordinates: 5.17 %
- Heigh and Gradient Estimation: 0.81 %
- LowResMap fusión: 0.78 %
- Create LRM: 0.51 %

- Create HRM: 0.39 %
- Layers fusión: 0 %
- FusionTemporal2Loca: 0 %
- Spherical2Cartesian: 0 %

Estas cifras indican cuantitativamente lo comentado anteriormente de forma cualitativa. La funcionalidad introducida viene representada por la función FreedomAssignment y se observa que consume el 23.02 % del tiempo siendo la segunda función que más consume en todo el proceso.

Interpretación de los resultados

¿Qué información aporta la llamada asignación de libertad? Suponiendo que se posee un conocimiento previo del terreno en forma de mapas, la libertad estaría dando una información adicional a los puntos de impacto del láser. Una información que consiste en que el láser ha podido atravesar dicha zona ya que ha pasado entre la altura del suelo y la crítica, por ese motivo se pueden sacar las siguientes conclusiones:

- El área atravesada por el láser y tomada como libre no ha sido ocupada por ningún objeto desde la creación del mapa tomado como referencia.
- A falta de información de la capa de gradientes, se puede utilizar la capa de libertades como complemento a la de ocupación. De esta forma donde no se tenga información del gradiente y si se posea de la libertad, se tomará 1 – *libertad* como el nivel de ocupación en dichas celdas.

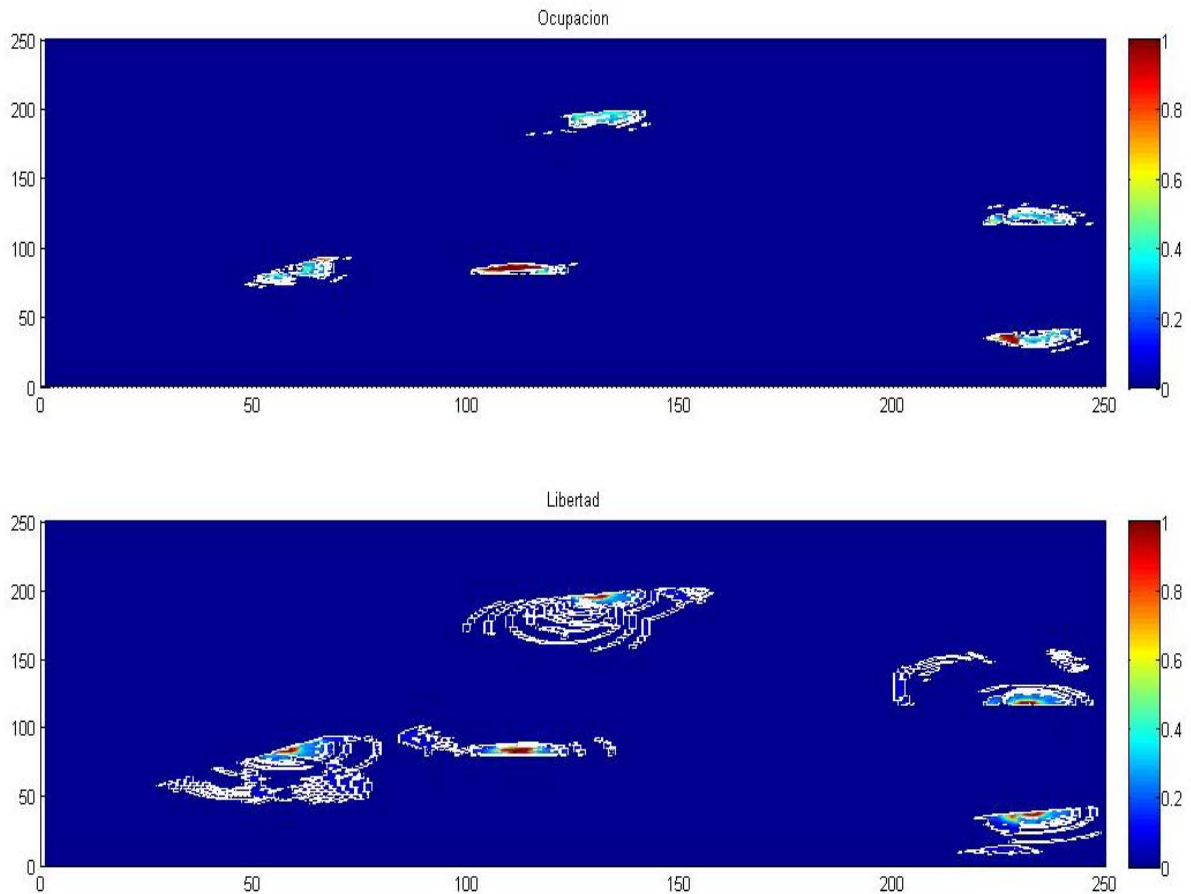


Figura 54. Reconstrucción de la ocupación y asignación de libertad para una trayectoria simple

Se puede apreciar que al tener en cuenta la capa de libertad la información obtenida del terreno es mucho mayor. Conviene por tanto no desperdiciar esa información y utilizarla siempre que no se tengan datos sobre el gradiente del terreno para calcular la ocupación real. La capa de libertad se convierte en una hipótesis de trabajo para el robot, representando una alternativa para cuando este debe planificar una trayectoria y son necesarios datos adicionales que no se poseen. No es representa una certeza absoluta sobre el grado de ocupación del terreno sino que es únicamente un punto de partida para comenzar el movimiento.

Uniendo la información de ambas capas en una sola se puede comprobar cómo la cantidad de información recopilada usando la capa de libertad es mucho mayor que sin usarla. En el ejemplo que se está analizando se pasa de información obtenida sobre la ocupación para 726 celdas a información sobre 2239 celdas lo que corresponde a un aumento del orden del 300 % en cantidad de información. Aunque no es totalmente cierto que una probabilidad muy baja de que la superficie este libre sea equivalente a que exista una probabilidad alta de que esté ocupada, a falta de

información si se puede utilizar esta. Gráficamente se pueden comprobar los resultados con una mayor percepción de las mejoras encontradas. Con todo esto se puede concluir que dependiendo de la potencia de cálculo disponible, es útil disponer de este nuevo algoritmo puesto que aumentando el tiempo de cálculo en un 23 %, el rendimiento en la obtención de información aumenta en un 300 %.

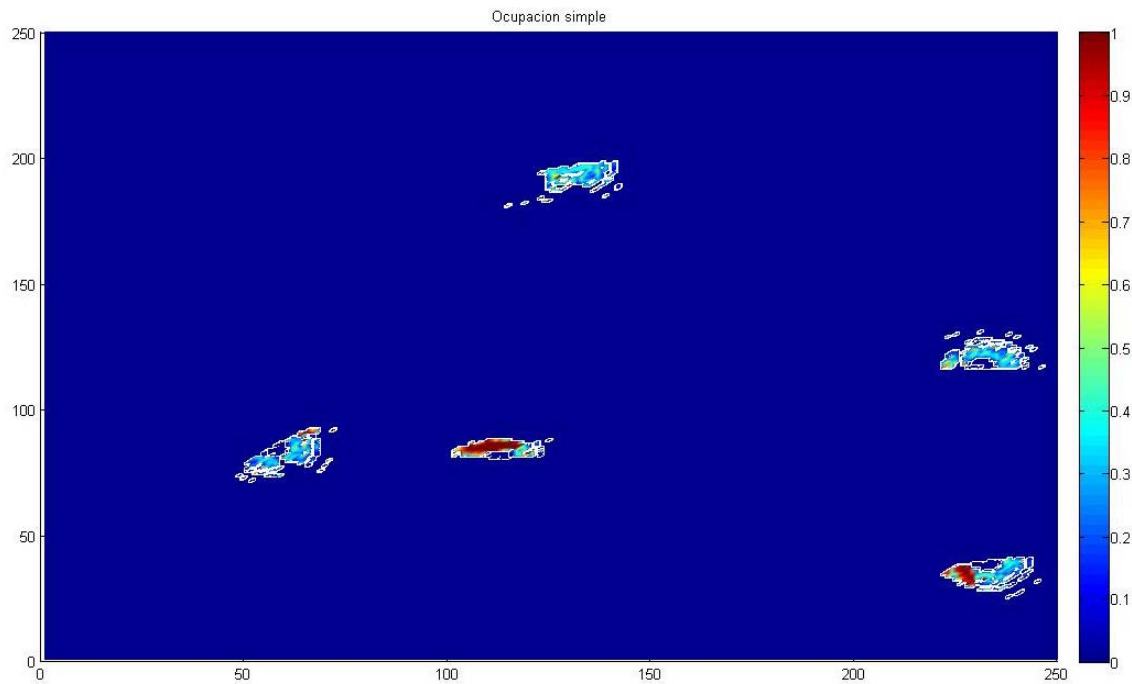


Figura 55. Ocupación calculada de forma simple

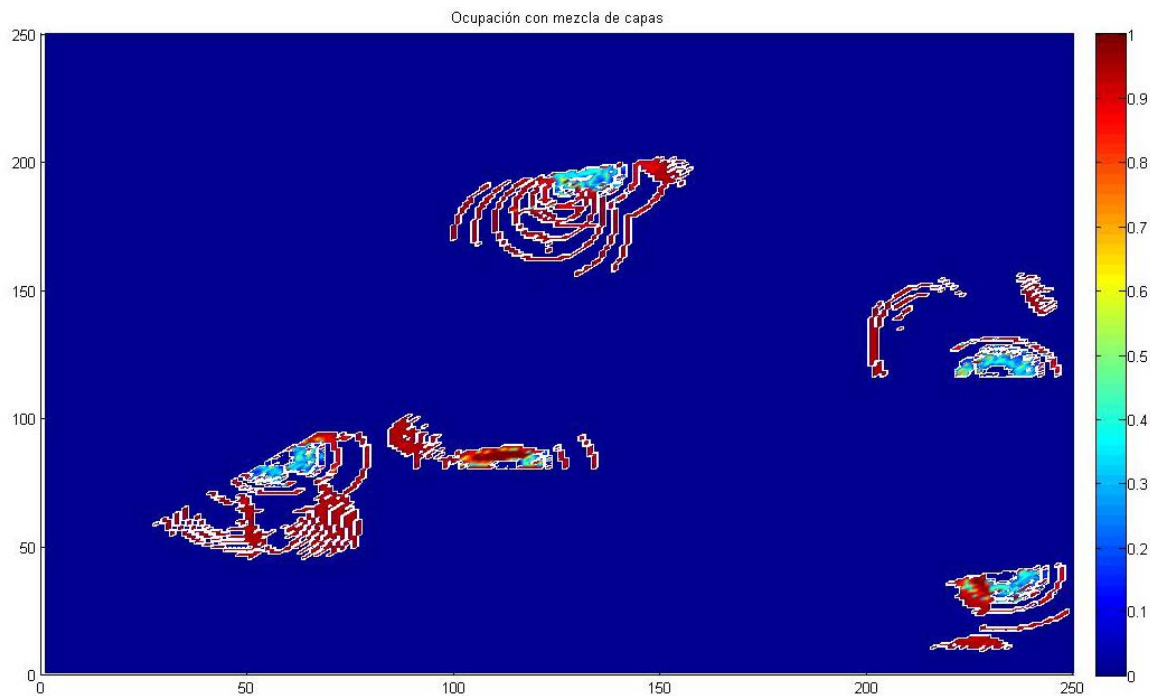


Figura 56. Ocupación calculada con la información proveniente del algoritmo de asignación de libertades

4.13. OPERACIONES SOBRE LOS MAPAS

En el proyecto final de carrera presentado, se pueden distinguir dos categorías de fusión de información:

- Fusión de la información existente en un mapa con la nueva información procedente de la reconstrucción con el láser.
- Fusión de la información procedente de varios robots que colaboran en la definición de un mapa global.

A continuación se detalla cada una de ellas.

4.13.1. Fusión de la información nueva con la información existente en alta resolución.

La información de alta resolución que se reconstruye con la última adquisición del láser debe ser fusionada con la información previa del entorno (también disponible con alta resolución). Este es uno de los principales procesos de todo el sistema de mapeado ya que la forma de fusión de los datos influye mucho en los resultados finales.

Como se está trabajando con mapas dinámicos, puede ocurrir que al observar una celda la información nueva sea similar a la información previa o sea totalmente

distinta, si por ejemplo algún objeto móvil ha invadido el área analizada. Por ello, el algoritmo no solo debe tener en cuenta la actualización de una celda por una mejor observación de la misma sino por un posible cambio en el contenido.

Como se ha definido anteriormente, se denomina *aux_MapCell_local* al mapa que se reconstruye contando como única información un escaneo del láser, este mapa temporal se va fusionando tras cada lectura del láser con el mapa de alta resolución (HRM) que va acumulando la reconstrucción dentro de la misma área.

El algoritmo propuesto comienza comprobando la matriz de celdas del mapa temporal *aux_MapCell_local*. Aquellas celdas que posean confianza no nula, contendrán información reconstruida con el último escaneo y será necesario fusionar dicha información con la posible ya existente para la misma celda en el mapa de alta resolución. En este proceso se pueden considerar dos casos diferenciados:

- Los datos son similares.
- Los datos son muy diferentes.

Cuando no se produce un cambio brusco en las estimaciones del terreno, sino que simplemente varía ligeramente la información recolectada en ese entorno, los datos fusionados son el resultado de aplicar una ponderación en función de las confianzas del dato del mapa de alta resolución y el mapa auxiliar.

$dato_{k+1} = \frac{dato_k * confianza_k + dato_{aux_MapCell_local} * confianza_{aux_MapCell_local}}{confianza_k + confianza_{medida}} \quad (25)$	
--	--

Para discernir si los datos son parecidos o no, se ha establecido un umbral de $\pm 10^\circ$ entre los gradientes. Éste, es un parámetro ajustable con la práctica y con las simulaciones posteriores que se han de hacer en Microsoft Robotics Studio.

Si por el contrario el gradiente del mapa auxiliar es claramente diferente del almacenado en el mapa se consideran tres posibilidades:

- Hay una diferencia importante en el valor de la confianza. Lo que significaría que uno de los dos valores es más fiable que el otro y se procedería a utilizar el valor más fiable de los dos.

- Hay un valor de confianza parecido y pequeño. En este caso, los dos valores parecen ser sensibles a la parte de la celda estimada por lo que el valor de mayor confianza será utilizado.
- Hay un valor de confianza parecido y elevado en ambos casos. En esta situación, lo más probable es que haya habido un cambio en el entorno y como tal, la nueva estimación sustituye a la antigua.

Aquí se produce otra disyuntiva, cuando se trata de distinguir que significa confianza mayor. Se está considerando que unos datos tienen confianza mayor que otros cuando se cumple: $Confianza_{medida} > (\frac{2}{16} + Confianza_k)$ lo que equivale a decir que hay al menos dos subceldas más de observación en las nuevas medidas que en las acumuladas.

Este caso se puede dar en la práctica cuando un robot detecte un cambio en la celda con una gran visibilidad del mismo y se puede considerar información muy fiable, por ello aunque provenga de una única medida se le dará la importancia necesaria y se actualizará el mapa.

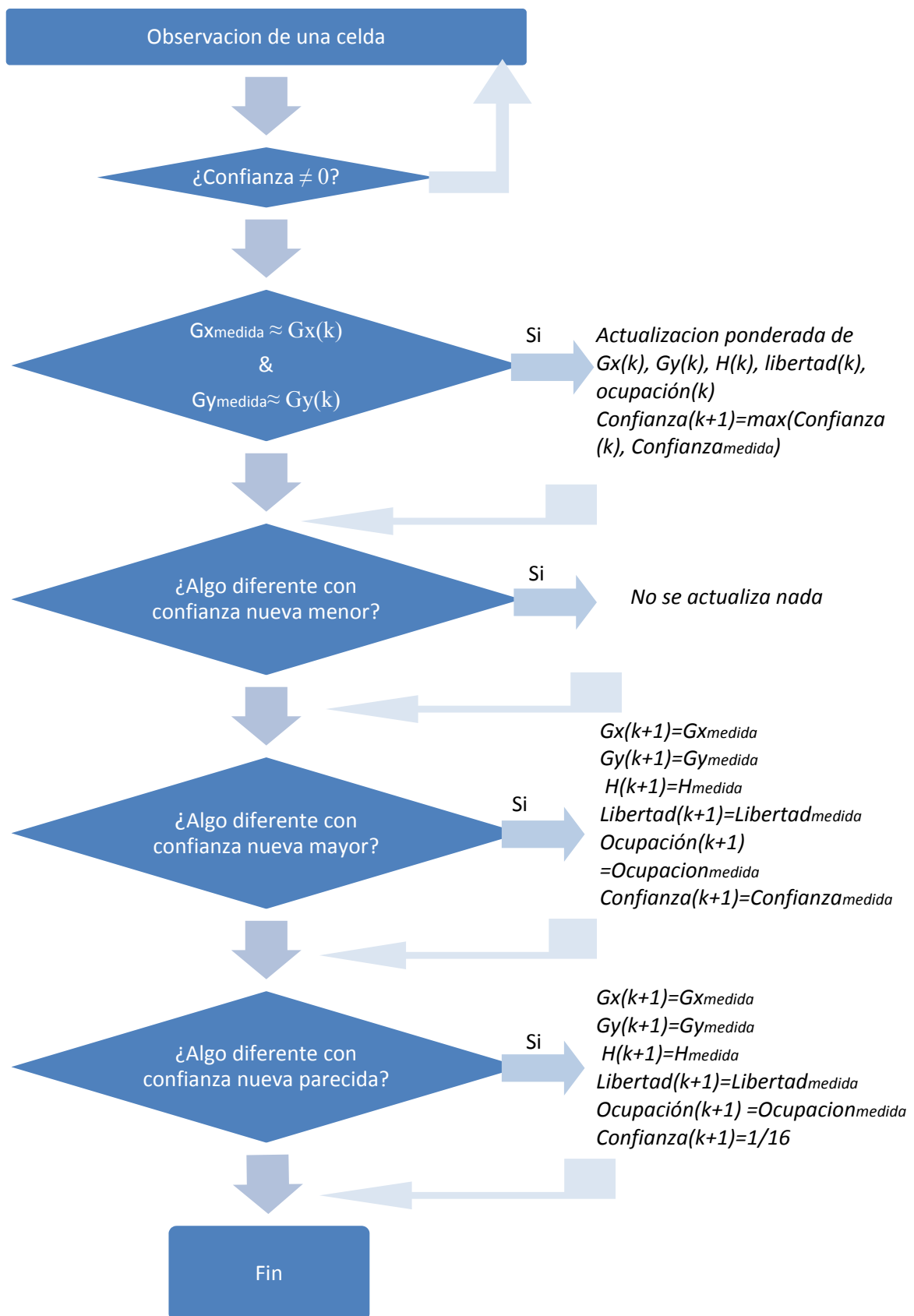


Figura 57. Algoritmo de fusión de la información

Se ha hablado mucho sobre la fusión de alturas, gradientes, ocupación, libertad, pero no se ha dicho nada sobre las capas de comunicación y de señal GPS. Para ellas no existe fusión, dado que el último valor será el válido puesto que la calidad en la recepción de la señal GPS y comunicación puede variar con el tiempo. Por lo tanto, para cuando una nueva medida sea incorporada al mapa, esta medida sustituirá a la anterior si existía y en el caso de que no fuera así se añadirá a las medidas existentes. Se hace notar que esta actualización solo se da en las celdas por las que transita el robot.

Con este algoritmo se ha abordado la fusión de la información temporal con la información acumulada, ambas en alta resolución. Posteriormente a esta fusión la información temporal será reemplazada por la nueva medida y el proceso se repetirá cíclicamente.

4.13.2. Conversión de alta a baja resolución

La información almacenada en el mapa de alta resolución es inmediatamente traspasada al mapa de baja resolución (LRM). Adicionalmente, cuando el robot se desplaza y el alcance de los sensores traspasa los límites del mapa de alta resolución, es necesario crear información de las nuevas celdas de alta resolución basándose en la información almacenada en las celdas de baja resolución contiguas al mapa anterior.

Considerando la primera conversión descrita, es necesario condensar la información de varias celdas del HRM para dar lugar a una única celda del LRM.

El número de celdas del HRM que se deben fundir para dar lugar a una del LRM es el siguiente:

$N^{\circ} \text{ celdas HRM} = \frac{N^{\circ} \text{ celdas LRM equivalentes} * \text{Escala LRM}}{\text{Escala HRM}}$	(26)
--	------

Este proceso ha sido dividido en dos partes, primero se realiza la fusión de las celdas necesarias del HRM para dar lugar a un nuevo LRM_{local} , mapa que representa un mapa local con la resolución adaptada a la del LRM. Posteriormente se situará el nuevo LRM_{local} sobre el LRM, centrando el local en el punto correcto del global se procederá a su fusión con el mismo algoritmo que se ha comentado anteriormente

para fusionar una medida temporal con las medidas acumuladas del HRM. El proceso es el siguiente:

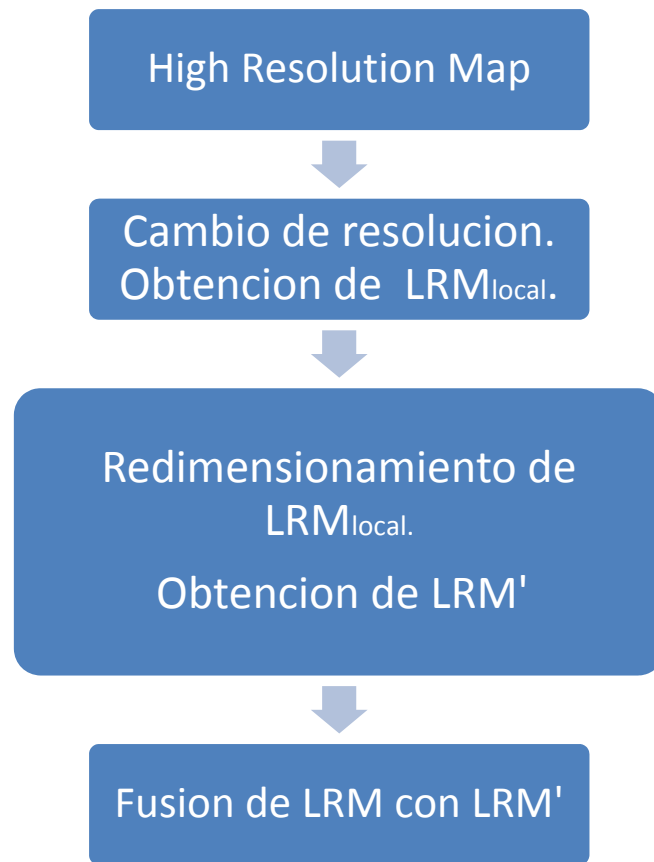


Figura 58. Esquema de actualización del HRM

Cambio de resolución

El proceso de cambio de resolución se realiza condensando el número adecuado de celdas del HRM mediante una media ponderada según la confianza de cada celda.

En este caso, se calcula la media ponderada de todos los elementos de la capa de información correspondientes a una celda de baja resolución con los elementos de la capa de confianza y resultan un determinado dato que se tomará como valor para obtener una celda del LRM_{local}. De esta forma, recorriendo la matriz HRM por grupos de elementos se va creando el nuevo mapa local con la nueva resolución.

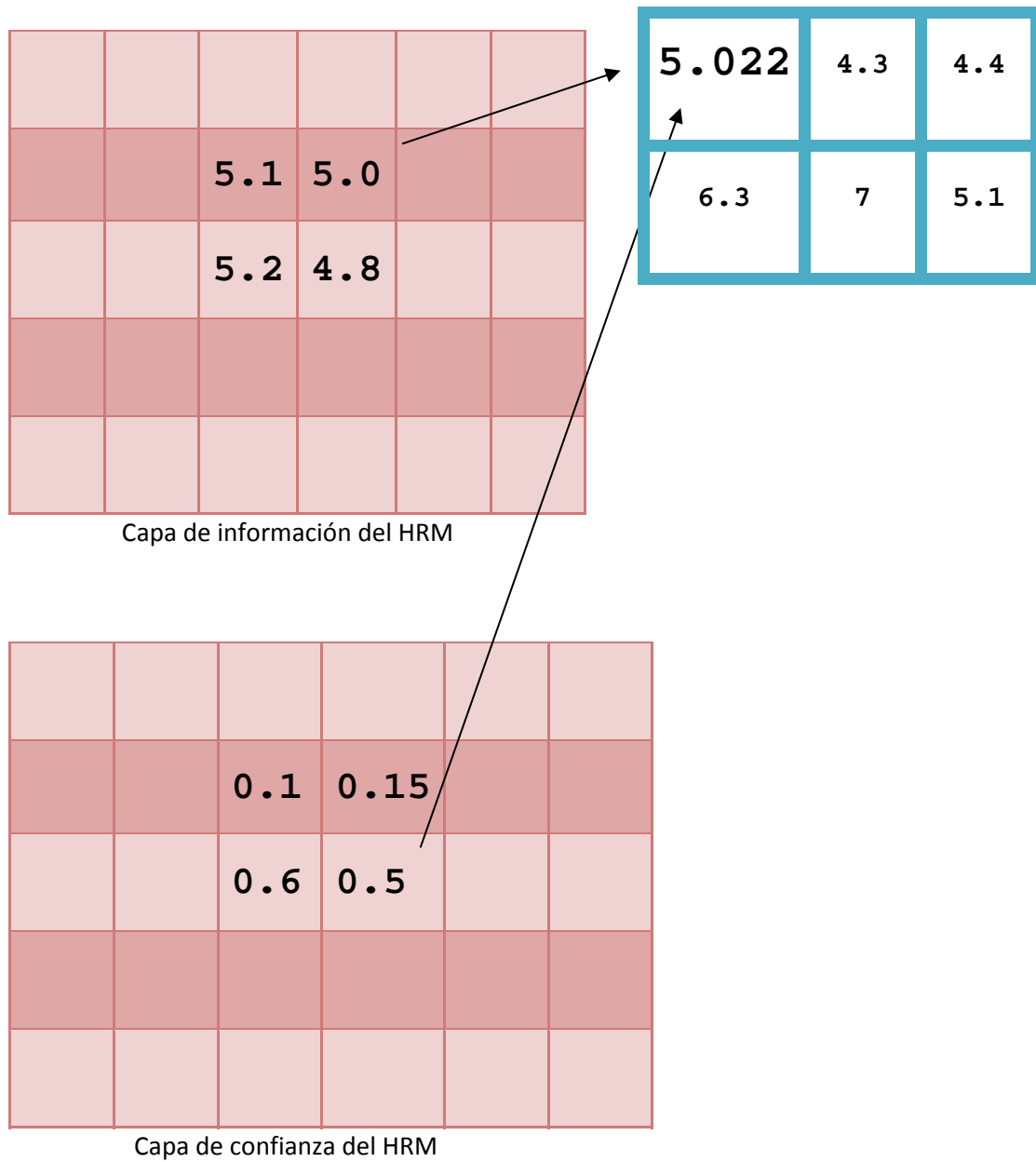


Figura 59. Ejemplo de actualización del HRM mediante media ponderada por las confianzas

La expresión que resume el algoritmo utilizado es:

$$Dato(i_k, j_k)_{LRM} = \frac{\sum_{i=i1}^{i=i2} \sum_{j=j1}^{j=j2} Confianza(i, j)_{HRM} \cdot Dato(i, j)_{HRM}}{\sum_{i=i1}^{i=i2} \sum_{j=j1}^{j=j2} Confianza(i, j)_{HRM}} \quad (27)$$

Ajuste de dimensión

Como segundo paso para la fusión al LRM está el redimensionamiento del LRM_{local} para obtener el LRM' . Para realizar esto se debe localizar el centro del HRM dentro del LRM y copiar la matriz obtenida LRM_{local} en el lugar correcto dentro de una matriz vacía y rellena de ceros.

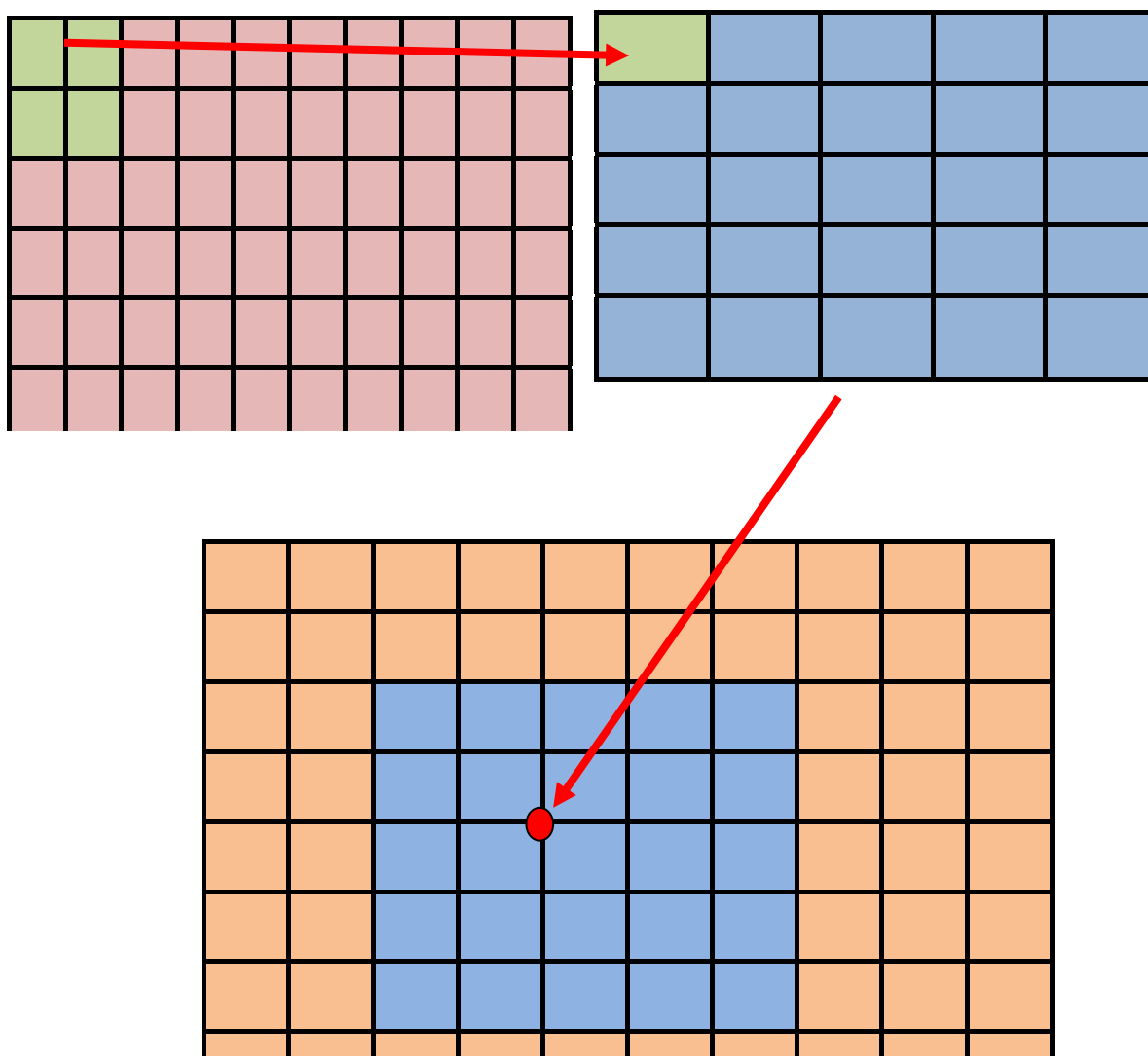


Figura 60. Proceso de fusión del HRM con el LRM. Primero se redimensiona el HRM obteniendo un mapa en baja resolución de un tamaño local. Posteriormente dicho mapa se posiciona adecuadamente dentro de un mapa de dimensiones globales.

Fusión de la información

Para la fusión de la información del nuevo mapa creado LRM' con el LRM que se encarga de registrar las medidas acumulativas se emplea el algoritmo de fusión

comentado anteriormente en el apartado dedicado a la fusión de datos del HRM. El algoritmo, basado en comparación de confianzas e innovaciones de los datos ha dado un buen resultado en el proceso de reconstrucción.

4.13.3. Conversión de baja a alta resolución

A medida que el robot se va desplazando por el mapa surge la necesidad de renovar el mapa local (HRM) porque su alcance se sale de las dimensiones de este. Es en esos momentos cuando cobra especial importancia el paso de información del LRM al HRM con el objeto de renovar la información local a medida que se produce el desplazamiento del robot.

Este proceso es parecido al proceso inverso que ya se ha comentado pero con algunas diferencias importantes.

En primer lugar, hay que detectar el momento oportuno para la toma de los datos pero eso ya se comentó en apartados anteriores y no se repetirá aquí.

Posteriormente, se produce una copia de información del LRM al HRM teniendo en cuenta la fórmula para adaptar el número de celdas. En este caso lo habitual es que una celda del mapa de baja corresponda a varias celdas del mapa de alta. En esta conversión de resolución lo único que se puede hacer es dar a las celdas del HRM correspondientes a una del LRM la misma información, puesto que no hay otra forma de dividir la información.

Este proceso de copia de información se hace únicamente para las nuevas celdas del mapa de alta resolución, respetando la información previa de las ya existentes. Para ello, se obtiene la información del mapa de baja resolución de la zona donde se sitúa el robot y se extenderá en el mapa de alta resolución. Este proceso se hace transmitiendo la misma información a todas las celdas correspondientes.

En la Figura 61, se pueden observar dos fragmentos de ambos mapas. Se puede apreciar como la información contenida en una celda del LRM se extiende a 4 celdas correspondientes del HRM. Esto ocurre así cuando la escala del LRM es el doble del HRM.

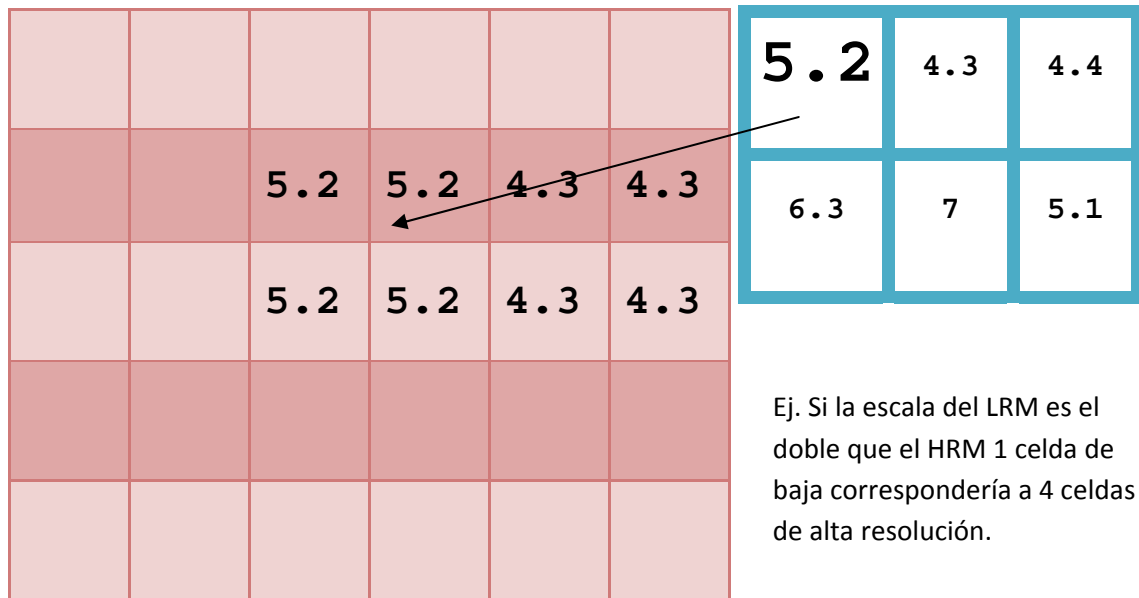


Figura 61. Transmisión de información del HRM al LRM.

La Figura 62 izquierda muestra un ejemplo de un mapa de alta resolución extraído de una determinada área del LRM según una vista del plano X-Z con objeto de ilustrar como el contorno del mapa local queda un poco deteriorado al haber sido capturado de un mapa de resolución menor. La figura de la derecha muestra el mapa real de alta resolución. Este efecto lógicamente se ve más acusado cuanto mayor sea la diferencia entre ambas escalas.

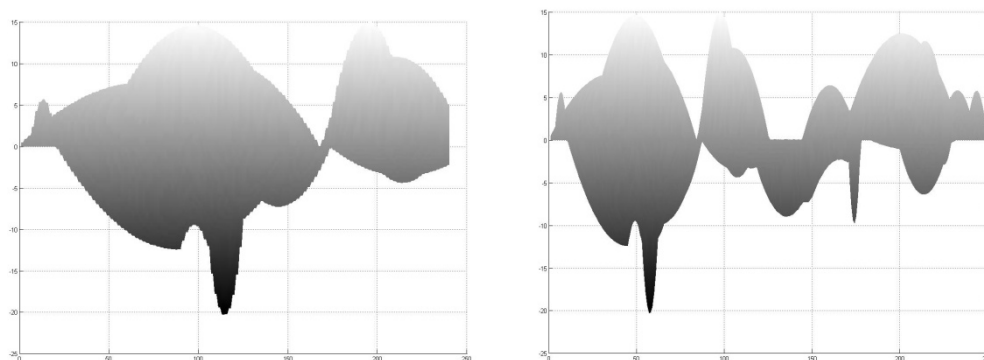


Figura 62. Comparación de mapas de alta resolución reconstruidos e importados desde mapas de baja resolución.

5. Localización visual del robot.

5.1. INTRODUCCIÓN

Si bien la calidad actual de los receptores GPS permiten un posicionamiento en exteriores con una precisión muy elevada, llegándose al orden del centímetro en tiempo real, el principal inconveniente que presentan se debe a que la recepción de la señal GPS no está garantizada en todo tiempo y lugar.

Con el objeto de localizarse cuando la señal GPS no está disponible en algún sitio, el robot combina diversas técnicas para obtener la estimación de su estado. En primer lugar, se dispone de la información de los sensores internos del robot como los sensores inerciales y la odometría. Esta información es fusionada mediante un filtro de Kalman y se obtiene una estimación del estado del robot (posición y actitud) que presenta como principal inconveniente la deriva con el tiempo ya que ambos procedimientos se basan en navegación incremental. Con el objetivo de establecer un sistema de posicionamiento que permita reducir las derivas de la estimación así obtenida se han desarrollado varios algoritmos que permiten la localización del robot dentro del mapa de alta resolución que está construyendo.

Los sistemas de localización y mapeado simultáneo conocidos por su nomenclatura inglesa como SLAM “Simultaneous location and mapping” son un campo muy activo en la investigación actual, sobre todo en interiores.

En el presente proyecto no se pretende realizar tal tipo de algoritmos, ya que se dispone de posicionamiento global y tanto el robot como el mapa quedan perfectamente referenciados, si bien las técnicas utilizadas derivan de las usadas en las aplicaciones SLAM estándar.

El objetivo del proceso de localización visual es usar la información del entorno del robot (lo que el robot detecta) para inferir cual es su localización dentro del mapa. Más detalladamente, se trata de usar la información que proporciona el láser, la cual permite una reconstrucción parcial del entorno y a continuación, “buscar” dicha información en un mapa donde los elementos reconstruidos están perfectamente referenciados e inferir así la posición del robot.

Habitualmente, un EKF (Filtro de Kalman Extendido) es la base de ésta técnica, que permite actualizar la estimación del estado del robot basándose en las características del entorno usadas como marcas naturales.

Las marcas naturales son características que pueden ser fácilmente reobservadas y distinguidas del resto del entorno, por lo que deben ser fácilmente reobservables permitiendo ser detectadas por ejemplo desde diferentes posiciones y diferentes ángulos. A su vez, deben tener como principal característica que sean distintas unas de otras, con objeto de no confundirlas. Evidentemente, todo lo que se tome como marca debe ser estacionario o bien conocer su posición en el momento de la detección. Así, un robot podría usar de marca a otro si conociese su posición en el momento en que éste es detectado.

Mediante un EKF se puede mantener un registro de la estimación de la posición y de su incertidumbre para todas las marcas y por lo tanto también para el propio robot.

A continuación se puede observar un esquema de un proceso general:

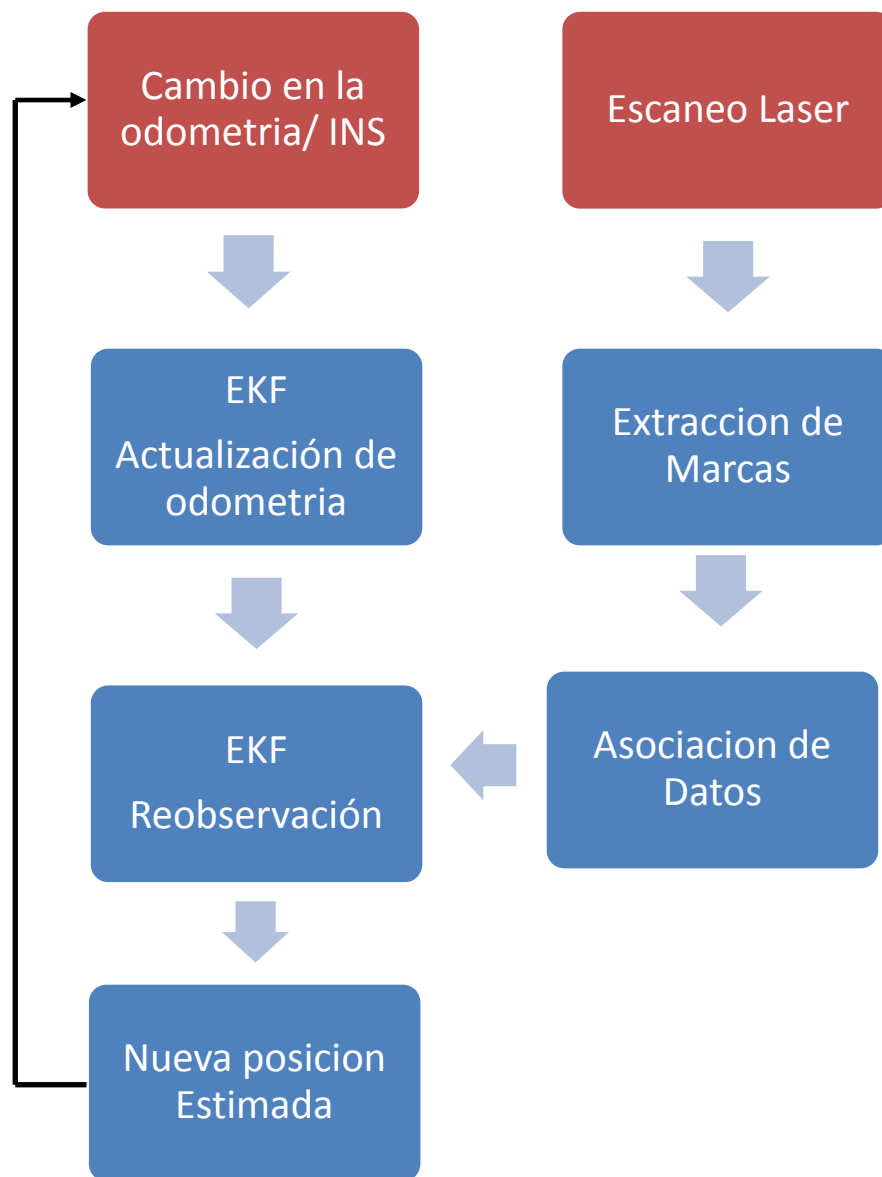


Figura 63. Esquema general de un proceso SLAM

Habitualmente, se puede utilizar una estructura de filtro de Kalman fuertemente acoplada, en la que tanto los estados procedentes de las estimaciones de la odometría o los sistemas inerciales es fusionada en un único filtro con la estimación de la posición de las marcas.

En el caso del proyecto final de carrera, no se ha utilizado esta opción y se ha utilizado una estructura de acoplamiento débil con filtros de Kalman en cascada, debido a que

existe un sistema de posicionamiento GPS que permite un posicionado muy preciso y se evita ejecutar las tareas de localización visual cuando éste está disponible.

Como indica la Figura 63, se produce una estimación del estado (posición y actitud) del robot que sirve como modelo para predecir el estado en el instante posterior. Como señal de medida se utiliza la posición del robot estimada a través del proceso de localización visual. En dicho proceso, las marcas re-observadas en la reconstrucción son localizadas en el mapa y estimar así la posición del robot.

Los siguientes diagramas intentaran explicar este proceso en más detalle:

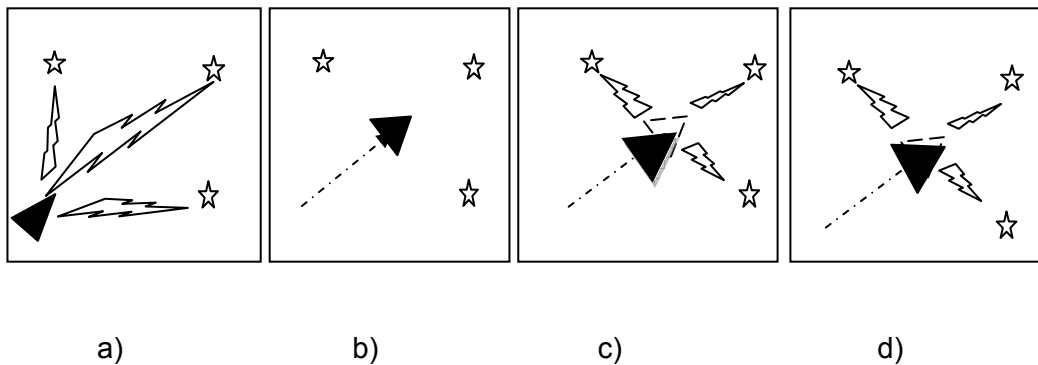


Figura 64. Diagramas que muestran el proceso de localización en base a características del terreno

El robot es representado en la Figura 64 por un triángulo de color negro, las estrellas representan marcas y los rayos representan las medidas iniciales usando las posiciones de las marcas. La figura b) muestra la nueva estimación de la posición del robot en base al sistema inercial y odometría (triángulo blanco) y la posición real alcanzada por el robot en negro. En la figura c) el robot vuelve a localizar las marcas y estimar su posición relativa a ellas y por lo tanto estimación de la posición absoluta que se introduce en el filtro para lograr una estimación de la posición final, indicada en la d).

En el proyecto final de carrera se han implementado diferentes clases de algoritmos de localización visual, que se detallan a continuación.

5.2. ALGORITMO BASADO EN COMPARACIÓN DE MAPAS

En lugar de la utilización de marcas, en este algoritmo se utiliza la reconstrucción local del terreno visionado en un único instante por el láser para generar un mapa parcial de alta resolución, a continuación se desplaza el mapa con reconstrucción parcial sobre el mapa de reconstrucción total hasta alcanzar la máxima similitud.

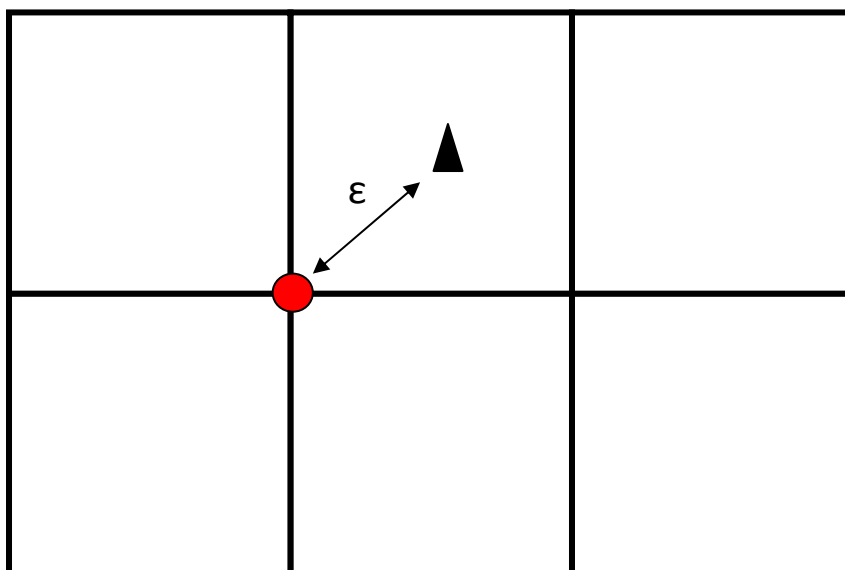
Para generar el índice de similitud se han utilizado las capas de alturas y la capa de gradientes de ambos mapas. Si bien la capa de gradientes es invariante al error de posición en altura (la altura es también estimada sin la ayuda del GPS) dispone de menor información que la capa de alturas.

El algoritmo de desplazamiento de un mapa sobre el otro tiene dos fases claramente diferenciadas. En primer lugar se realizan saltos a nivel celda, por lo que la salida de esta parte del algoritmo consistirá en la estimación de la celda en la que más probablemente se encuentre el robot. Con un coste computacional mayor, se realiza la estimación al nivel subcelda, consiste en reconstrucciones tomando como punto de partida subceldas y comparando la similitud, con lo cual la salida estimada es la subcelda en la que más probablemente se encuentra el robot.

Hipótesis de partida

En la primera parte del algoritmo de hay que hacer una serie de suposiciones para comenzar la búsqueda de la posición del robot. Estas suposiciones son:

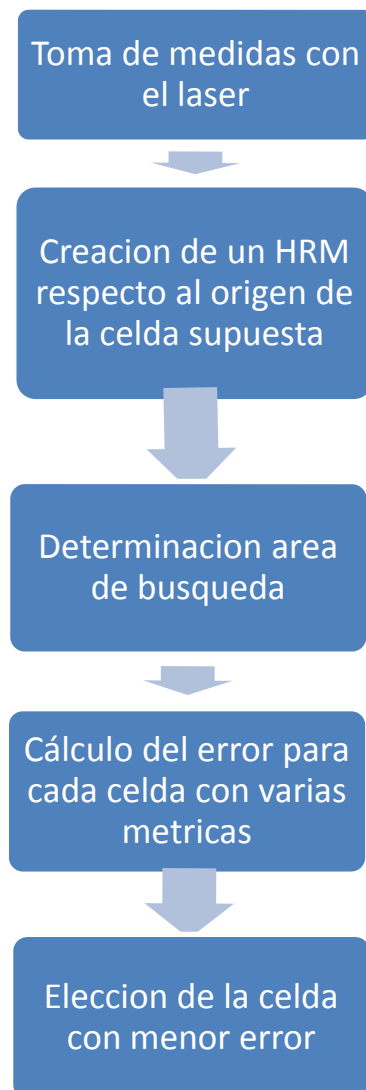
- No existe un error apreciable en la actitud del robot. Los ángulos de Euler proporcionados por el sistema son prácticamente exactos. Solo existe error en la posición y este error esta acotado. Esta hipótesis es razonable ya que la precisión en actitud de la unidad de medida inercial está por debajo del grado.
- Inicialmente se supondrá que la distancia del robot al origen de la celda es conocida y exacta. Esta suposición es necesaria para realizar la reconstrucción del mapa local respecto del origen de la celda en lugar de realizarlo desde el punto donde esté situado el robot. Más tarde se justificará el porqué de esta decisión.



Esta distancia denominada \mathcal{E} será la hipótesis de partida para el comienzo de la búsqueda ya que es la mejor aproximación que se posee al estado del robot.

- El error de posicionamiento máximo está acotado. Aunque la odometría tenga un error que es en principio acumulativo, a fusión con las medidas inerciales mediante el filtro Kalman mejora mucho las estimaciones y no es probable que el funcionamiento sin GPS se alargue demasiado en el tiempo. Este algoritmo está preparado para actuar en ambientes donde el GPS se pierde de forma ocasional por lo tanto el error estará acotado en un cuadrado de un determinado número de celdas. Por tanto no se buscará en todo el mapa de baja resolución sino que se busca en un área determinada, reduciendo con ello los tiempos de cómputo.

Esquema del algoritmo principal a nivel celda.



Los primeros pasos son similares a los ya comentados para el proceso de reconstrucción habitual del robot. En primer lugar se parte de la matriz de distancias del láser. Posteriormente se realizan las transformaciones matriciales habituales para la obtención de las coordenadas de cada punto de impacto. Con la única salvedad que aquí no se reconstruye respecto al origen del sistema fijo sino que se reconstruye respecto al origen de la celda donde se ha supuesto que se encuentra el robot.

La justificación de por qué se elige como punto de referencia el origen de la celda es la siguiente. Suponiendo que el estado real del robot sea el mostrado en la Figura 65, en la que el robot no se encuentra en la esquina de la celda (caso más habitual) y Y si se hiciera una reconstrucción cuyo origen fuera el propio robot, (bastante más simple de realizar en la práctica puesto que no se necesita conocer las posiciones absolutas del robot sino que únicamente con las distancias medidas por el láser y la estimación de actitud) el resultado sería el mostrado en la Figura 66, en la que los mapas a comparar quedarían muy desalineados.

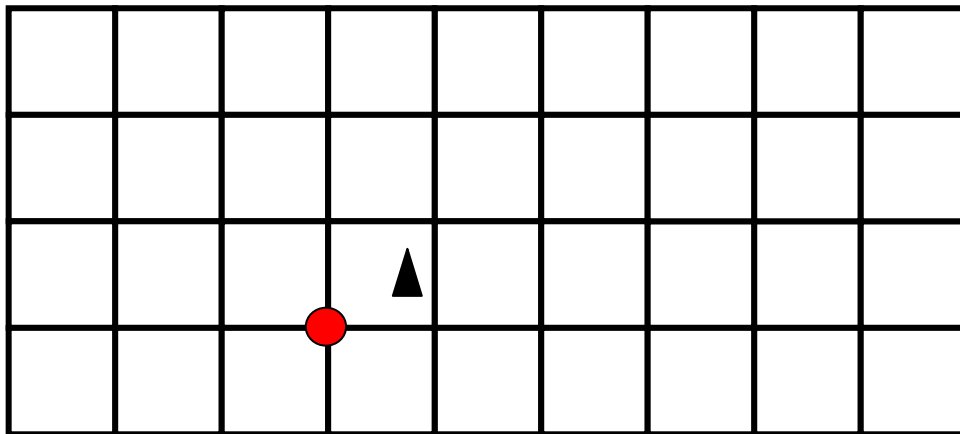


Figura 65. Localización del robot en el momento de comienzo de la localización visual

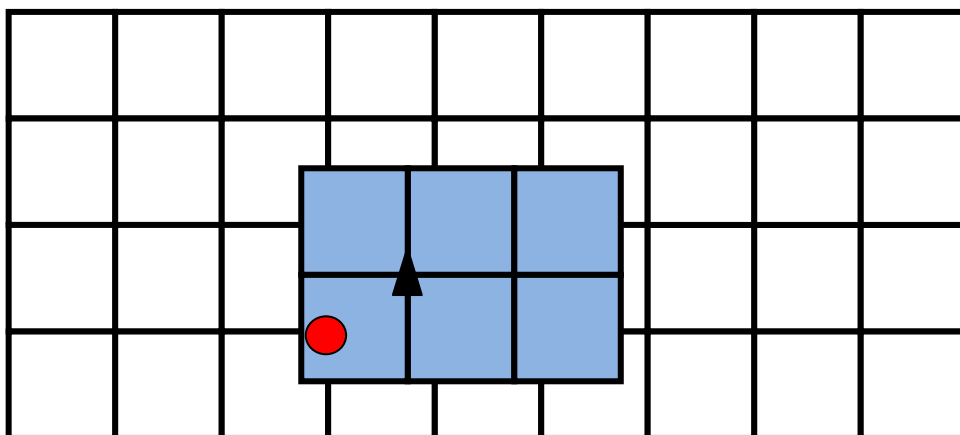


Figura 66. Desalineamiento de los mapas producidos por error en la estimación de la posición del borde de la celda

Por supuesto, el hecho de contar con una estimación inicial de la posición para tener en cuenta una estimación inicial de \mathcal{E} reduce mucho dicho desalineamiento.

La determinación del área de búsqueda se hace en base a los errores previsibles máximos que pueda tener la odometría. Habitualmente se ha estado tomando una distancia variable entre 3 y 10 metros de radio en los ensayos. Por ello, se delimitan las celdas del LRM que serán susceptibles de búsqueda.

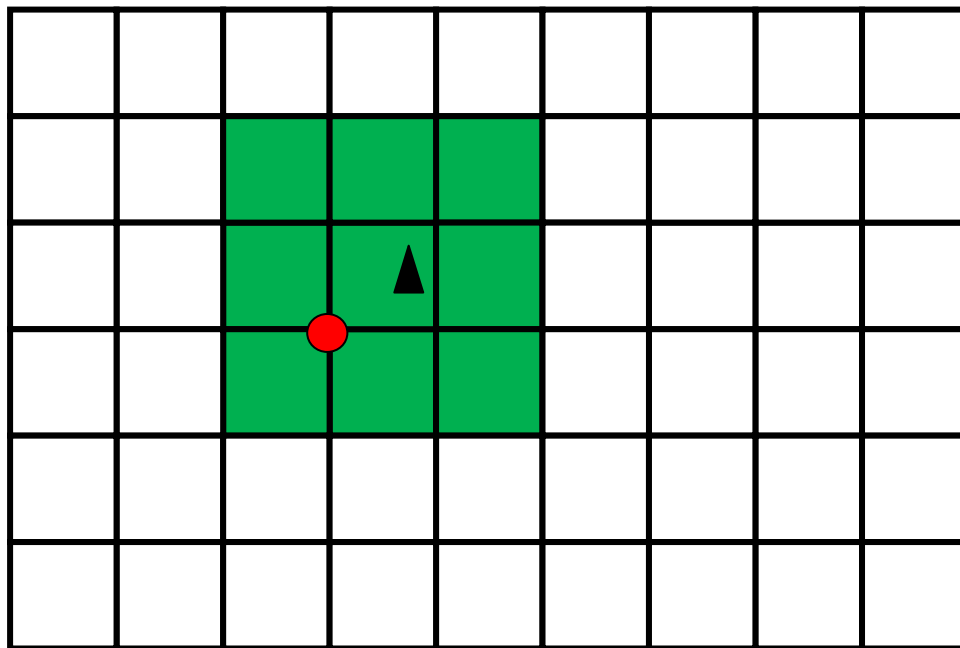


Figura 67. Mapas perfectamente alineados

Para la comparación de mapas, una vez seleccionadas las celdas candidatas a albergar al robot, se comienza extrayendo un mapa local centrado en el punto característico de cada una de ellas. De esta forma si se tienen 9 celdas del LRM candidatas se tendrán también 9 mapas de alta resolución extraídos para comparar con el mapa local realizado en los primeros pasos de la localización. El proceso de extracción del mapa local centrado en cada celda sigue el mismo procedimiento que el comentado en el apartado de extracción de un HRM ya comentado. Se trata de seleccionar el punto central del nuevo mapa, que en este caso será el punto característico de cada una de las 9 celdas. Posteriormente se delimita la nueva área de baja resolución a capturar y se transforma en un nuevo mapa de alta resolución. El proceso se repetirá cíclicamente con cada uno de las celdas candidatas. La Figura 68 ilustra de forma gráfica el proceso.

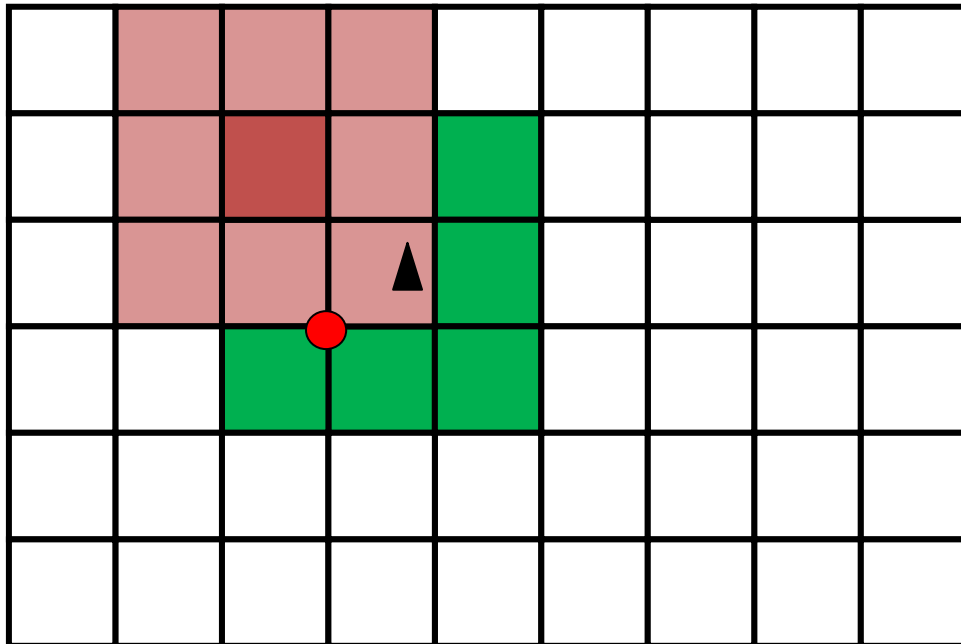


Figura 68. Extracción de mapas

El HRM asociado a la primera celda candidata sería el sombreado rojo. En este caso se ha mostrado con unas dimensiones de 3 x 3 celdas que coincide con la longitud de testeo, pero esto no tiene porque ser así. De esta forma, se tendrían 9 mapas de alta resolución, cada uno centrado en la celda correspondiente candidatos a ser comparados por diversas métricas.

Las métricas que han sido probadas son las siguientes:

- Módulo de las normas al cuadrado de los errores en el gradiente
- Norma al cuadrado de los errores en la altura
- Norma infinito aplicada a los gradientes
- Norma infinito aplicada a las alturas
- Combinación dos a dos

Cálculos realizados para la comparación de mapas

- a) Módulo de las normas al cuadrado de los errores en el gradiente

La fórmula aplicada para calcular esta métrica es la siguiente:

$\text{error gradiente} = \sqrt{e1^2 + e2^2}$	(28)
$e1 = \sum_{k=1}^n (GX\ local_k - GX\ candidato_k)^2$	(60)
$e2 = \sum_{k=1}^n (GY\ local_k - GY\ candidato_k)^2$	(61)

Donde:

- $GX\ local_k$ es un elemento k de la matriz de gradientes según la dirección X del mapa reconstruido respecto al punto característico de la celda donde se supone situado el robot.
- $GY\ local_k$ equivalente al anterior en la dirección Y.
- $GX\ candidato_k$ es un elemento k de la matriz de gradientes según dirección X del mapa local generado a partir de una determinada celda candidata.
- $GY\ candidato_k$ equivalente al anterior en dirección Y.

Esta métrica aplicada a todos los mapas candidatos centrados en cada una de las celdas candidatas produce como resultado de salida una celda óptima donde se ha producido el error mínimo.

Norma al cuadrado de los errores en la altura

$\text{error altura} = \sum_{k=1}^n (Z\ local_k - Z\ candidato_k)^2$	(62)
--	------

Donde:

- $Z\ local_k$ corresponde a la altura medida en un elemento k de la matriz de alturas del mapa reconstruido respecto al punto característico de la celda donde se supone situado el robot.
- $Z\ candidato_k$ corresponde a la altura medida en un elemento k de la matriz de alturas del mapa local generado a partir de una determinada celda candidata.

Igual que antes se aplica esta métrica a todos los mapas generados a partir de las celdas candidatas y se guarda la celda que presenta menor error. Esa celda será la óptima para esta métrica.

b) Norma infinito aplicada a los gradientes

$E_x = GX_{local} - GX_{candidato} $	(63)
$E_y = GY_{local} - GY_{candidato} $	(64)
$error\ gradiente_{\infty} = \ (E_x + E_y)\ _{\infty}$	(65)

Donde:

- GX_{local} y $GX_{candidato}$ son las matrices de gradientes según X de la reconstrucción local y la reconstrucción respecto a cada celda candidata respectivamente.
- $GY_{local} - GY_{candidato}$ igual que las anteriores para la dirección Y.
- $error\ gradiente$ es el elemento máximo de la suma de los valores absolutos de ambas restas.

c) Norma infinito aplicada a las alturas

$error\ altura_{\infty} = \ (Z\ local - Z\ candidato)\ _{\infty}$	(66)
---	------

d) Combinación de las métricas a y c

$error\ gradiente\ total = error\ gradiente * error\ gradiente_{\infty}$	(67)
--	------

e) Combinación de las métricas b y d

$error\ altura\ total = error\ altura * error\ altura_{\infty}$	(68)
---	------

Resultados obtenidos

La Figura 69 muestra la comparación de los errores en la localización del robot al describir una trayectoria arbitraria para un entorno generado en Matlab, utilizando las tanto las métricas basadas en el gradiente como en la altura.

Datos del ensayo:

- Escala LRM = 2 m.
- Escala HRM = 1 m.

- Estimación de error máximo de posición = 14 m de lado

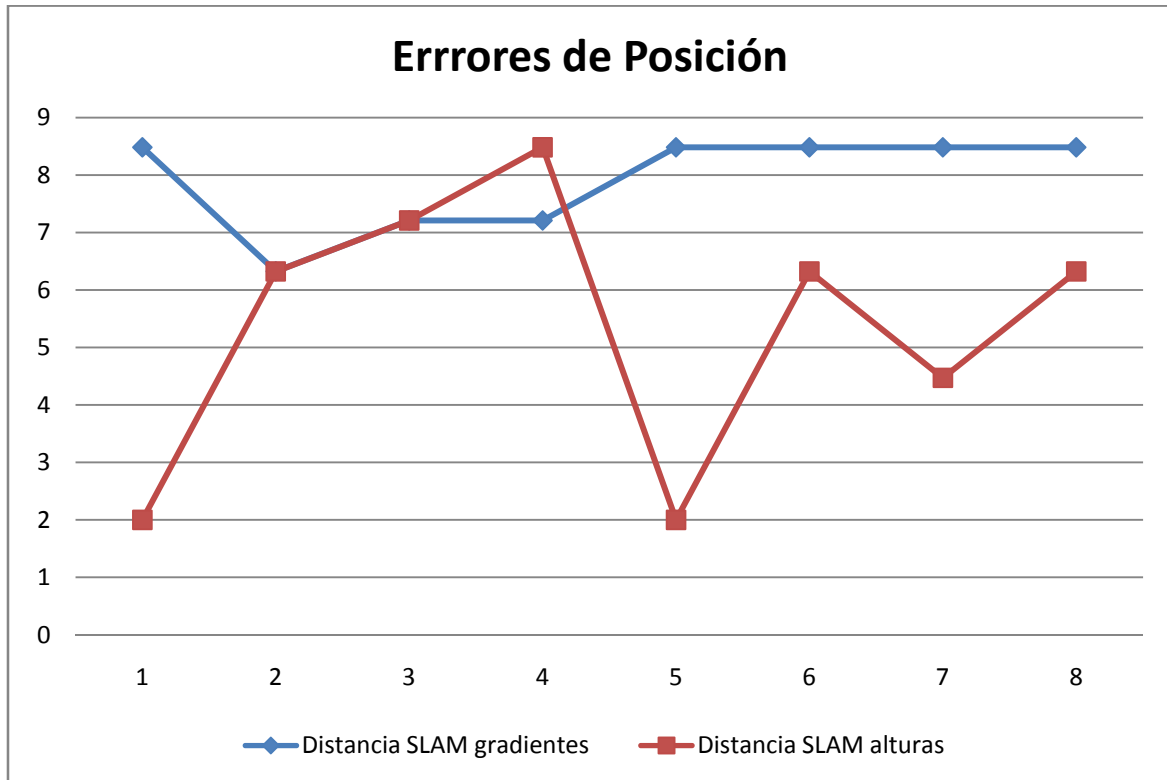


Figura 69. Comparación de errores en la estimación de la posición para dos métricas diferentes.

Para un estudio de una trayectoria con 8 puntos de reconstrucción, se observa que la métrica basada en gradientes produce peores resultados que la basada en la comparación de alturas aunque es más constante. No obstante, el error también depende de la calidad del mapa de alta resolución utilizada para la localización, ya que es más sensible al número de puntos de colisión la reconstrucción del gradiente que la de la altura.

5.3. Algoritmo basado en reconstrucción a nivel subcelda

Una vez encontrada la celda que posee el menor error de entre todas las celdas candidatas se puede proceder a ejecutar el algoritmo con precisión subcelda. Este proceso es similar al anterior pero se debe realizar una reconstrucción considerando como punto de referencia de las celdas una serie de puntos de referencia de subceldas tal y como indica la Figura 70.

El objetivo de este proceso es ajustar aun más la posición del robot por medio de pruebas eligiendo el punto cuya prueba resulte mejor.

Para ello se divide la celda tomada en el primer paso en 9 puntos:

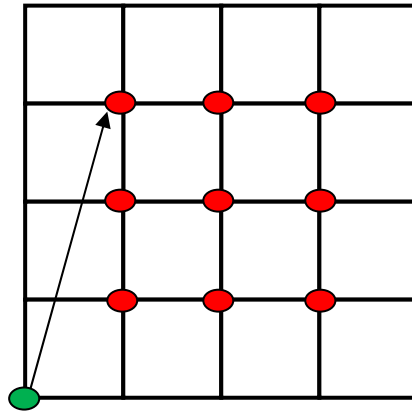


Figura 70. Puntos de referencia a nivel subcelda.

A continuación, se realizarán reconstrucciones de las medidas tomadas por la matriz del láser suponiendo situado al robot en cada uno de esos 9 puntos. Así, tomando como origen el punto característico de la celda, la distancia del origen a cada punto será la distancia que se introducirá en la matriz de traslación para representar la reconstrucción respecto a dicho punto característico. La justificación es idéntica al caso de SLAM principal.

Cuando se ha obtenido el punto de menor error pueden ocurrir dos casos distintos:

- Que el punto de menor error sea el central, luego se considerará que se ha llegado a un mínimo y se para la búsqueda dando ese punto como coordenadas del robot.
- Que el punto de menor error sea uno de los 8 periféricos. En ese caso se plantea una nueva búsqueda tratando de minimizar los errores hasta llegar a un punto de mínimo error.

Para el segundo caso se procede a buscar el punto de mínimo error entre los 3 puntos adyacentes a uno periférico.

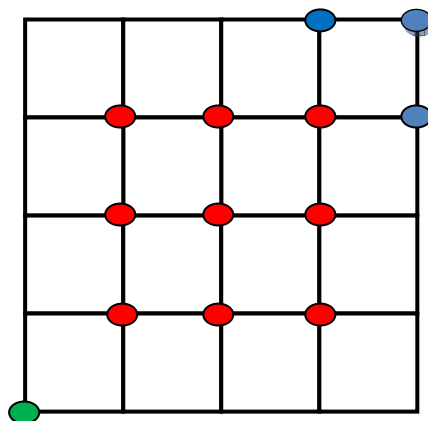


Figura 71. Puntos de referencia laterales.

Como se puede apreciar en la Figura 71, cada punto de la periferia posee 3 puntos de acceso inmediato. La nueva búsqueda se centrará en esos 3 nuevos puntos intentando minimizar con ello el error que se cometía anteriormente. Este proceso cíclico podría converger a un mínimo y hacer muy pequeño el error encontrando de esa forma la posición actual del robot. El proceso está basado en ir desplazando la búsqueda por la dirección de menor error. No se garantiza con ello que se llegue al mínimo absoluto puesto que este podría presentarse aislado, pero es una forma de intentar mejorar la estimación producida por el primer método.

En la práctica se depende mucho de las capacidades de cómputo, por ello no es conveniente dejar el proceso iterar constantemente, sino que es preferible elegir un número de iteraciones permitidas, y finalizando estas se dará por terminada la búsqueda y se escogerá el punto de menor error de los encontrados como mínimo.

Resultados obtenidos

En las siguientes pruebas se ha permitido una única iteración con lo que el desplazamiento del mínimo podría darse hasta el borde de la celda únicamente.

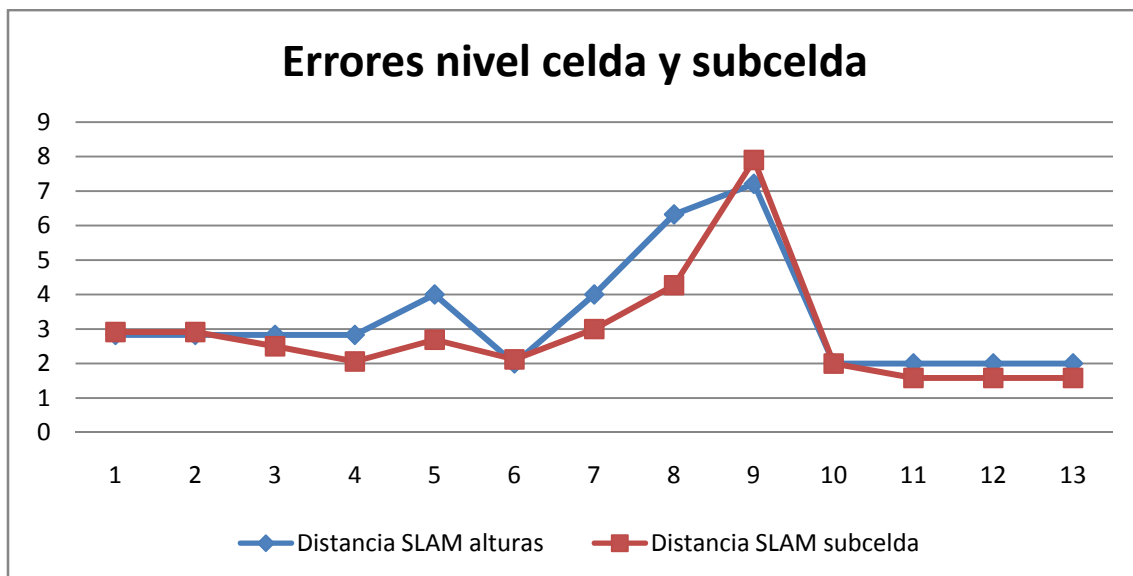


Figura 72. Comparación de los algoritmos de nivel celda y subcelda

Datos del ensayo:

- Escala LRM = 2 m.
- Escala HRM = 1 m.
- Tamaño del cuadrado de celdas candidatas = 14 m de lado

Se aprecia como con frecuencia el algoritmo de nivel subcelda mejora sensiblemente los resultados del nivel celda, por lo que la estimación de la posición del robot es mejor.

En general, se observa que cuanto más abrupto es el entorno sobre el que se ejecuta el algoritmo, mejores resultados se obtienen. Así, en un terreno prácticamente plano sin objetos de tamaño importante alrededor, el algoritmo de localización como es lógico pierde su eficacia. Este hecho no se puede considerar en si un problema, ya que la pérdida de señal GPS se produce habitualmente porque grandes objetos “hacen sombra” al receptor del robot y por lo tanto, deben existir objetos en las cercanías del robot.

5.4. MÉTODO SIFT

Introducción

La búsqueda de imágenes coincidentes es fundamental en multitud de problemas de visión por computador, incluyendo reconocimiento de objetos en un escena, resolver estructuras 3D de múltiples imágenes, correspondencia estéreo y otros. En este capítulo se describirán características de imágenes que tienen algunas propiedades que las hacen adecuadas para la coincidencia entre diferentes imágenes de un mismo objeto o escena. Las características son invariante ante el escalado y rotación y parcialmente invariante ante cambios en la iluminación y punto de vista de cámara 3D. Están muy bien localizadas tanto en el dominio espacial como en el dominio de la frecuencia reduciendo las probabilidades de oclusión o ruido. Un gran número de características pueden ser extraídas de imágenes típicas con algoritmos eficientes. De hecho las características son altamente distintivas, lo que permite a un único punto característico ser encontrado entre una gran base de datos de puntos característicos dando las bases para el reconocimiento de objetos dentro de una escena [13].

El coste de la extracción de estas características esta minimizado haciendo un desarrollo de filtros en cascada en los que las operaciones más costosas se desarrollan únicamente a los puntos que pasan el test inicial. Las principales etapas para generar el set de imágenes son:

- Detección de extremos en el espacio de escalas: la primera etapa en la búsqueda computacional busca sobre todas las escalas y emplazamiento de las imágenes. Esta implementada de forma eficiente mediante una diferencia de funciones gaussianas para identificar potenciales puntos de interés que son invariantes ante escala y orientación.
- Localización de puntos característicos: en cada emplazamiento de los candidatos, se busca un modelo detallado para determinar el emplazamiento y

la escala. Los puntos característicos son seleccionados en base a medidas de su estabilidad.

- Asignación de orientación: una o más orientaciones son asignadas a cada punto característico basadas en direcciones del gradiente de la imagen. Todas las operaciones siguientes son realizadas teniendo en cuenta que la imagen ha sido transformada respecto a su orientación, escala y emplazamiento inicial, dando de esa forma invarianza respecto a dichas transformaciones.
- Descripción de puntos característicos: los gradientes locales de la imagen son medidos a una escala seleccionada en una región centrada en cada punto característico. Son transformados en una representación que permite cambios en la forma y en la iluminación.

Esta técnica ha sido llamada Transformación de Características Invariantes a Escala (SIFT) ya que transforma los datos de una imagen a coordenadas invariantes a escalado relativas a sus puntos característicos.

Para emparejado y reconocimiento de imágenes, los puntos SIFT son primeramente extraídos de un set de imágenes de referencia y almacenados en una base de datos. Entonces, para una nueva imagen, se le extraen sus características de forma individual y se la compara con las características almacenadas en la base de datos encontrando puntos característicos candidatos basándose en la distancia euclídea.

Los descriptores de puntos característicos son altamente distintivos, lo que permite a un único punto característico ser encontrado con alta probabilidad dentro de una gran base de datos. Sin embargo, en una imagen recargada habrá muchos puntos característicos pertenecientes al fondo que no podrán ser marcados dentro de la base de datos haciendo crecer con ello los resultados falsos en lugar de los correctos. Las coincidencias correctas podrán ser filtradas del set completo identificando subconjuntos de puntos característicos que son congruentes con el emplazamiento del objeto, la escala y la orientación de la nueva imagen. La probabilidad de que varios puntos característicos coincidan en esos parámetros de casualidad es mucho menor que para un punto característico individual. La determinación de esos grupos consistentes puede ser realizada rápidamente mediante una eficiente tabla hash implementada gracias a la transformación de Hough generalizada [13].

Cada juego de 3 o más características que concuerdan en un objeto y su pose es estudiada de forma más profunda para su verificación detallada. Primeramente se realiza una estimación por mínimos cuadrados para afinar la pose del objeto. Cualquier otra característica de la imagen consistente con esa pose será identificada, y las demás serán descartadas. Finalmente una computación detallada es realizada en

base a la probabilidad de que un set particular de características indique la presencia de un objeto, dando la precisión de la búsqueda y un número de las coincidencias falsas probables. Los objetos que pasan todas estas pruebas podrán ser identificados correctamente con una alta probabilidad [13].

5.4.1. Detección de extremos en el espacio de escalas

Como se ha descrito anteriormente, se detectan puntos característicos usando una serie de filtros en cascada, lo que provee de eficientes algoritmos para identificar emplazamientos de candidatos que serán examinados profundamente.

La primera etapa de la detección de puntos característicos será identificar emplazamientos y escalas que puedan ser asignadas de forma repetitiva bajo diferentes vistas de un objeto. La detección de lugares que son invariantes ante cambios de escala de una de una imagen puede ser realizada buscando características estables a través de todas las posibles escalas, usando una función continua de escalado conocida como espacio de escalas.

La función de espacio de escalas es definida como una función $L(x, y, \sigma)$, que se produce de la convolucion de gaussianas de escala variable, $G(x, y, \sigma)$ con una imagen de entrada $I(x, y)$:

$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$	(29)
---	--------

Donde el * indica operador convolucion en x e y , además

$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$	(70)
---	--------

Para detectar de forma eficiente puntos característicos estables en el espacio de escala se ha propuesto la utilización de extremos de la función de diferencias gaussianas convolucionadas con la imagen, $D(x, y, \sigma)$, la cual puede ser calculada a partir de la diferencia de dos escalas sucesivas separadas por un factor k :

$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$	(71)
---	--------

Hay un gran número de razones para elegir esta función. En primer lugar es una función particularmente eficiente de computar, así como las imágenes suavizadas, L ,

necesarias para la búsqueda de características en el espacio de escalas y D puede ser calculada por una simple extracción de imágenes.

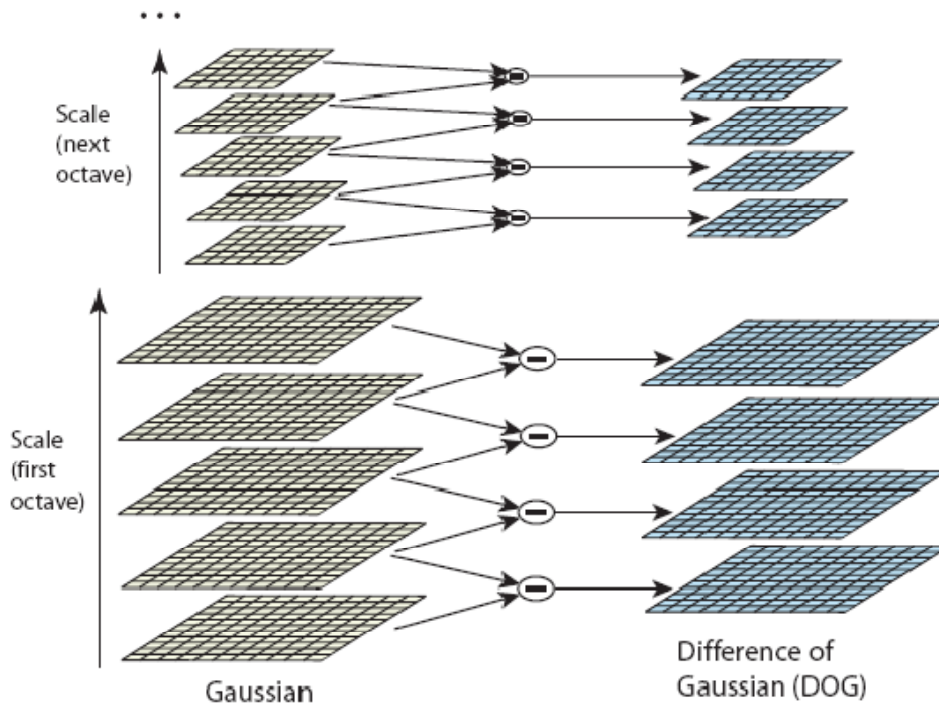


Figura 73. Extracción de características en cada octava. Adaptada de David G. Lowe [13].

Además, la diferencia de gaussianas provee de una aproximación cercana a la función normalizada en escala Laplaciana de Gaussiana, $\sigma^2 \nabla^2 G$, como fue estudiado por Linderberg (1994). Linderberg mostró que la normalización de la Laplaciana con el factor σ^2 es necesaria para una verdadera invarianza en escala. En comparaciones detalladas experimentales, Mikolajczyk (2002) encontró que el máximo y el mínimo de $\sigma^2 \nabla^2 G$ produce la mayor cantidad de características estables comparándolo con una gran cantidad de otras posibles funciones aplicadas a imágenes como el gradiente, Hessiana, o función de esquinas de Harris.

La relación entre D y $\sigma^2 \nabla^2 G$ puede ser fácilmente entendida gracias a la ecuación de difusión del calor siendo $t = \sigma^2$.

$\frac{dG}{d\sigma} = \sigma \nabla^2 G$	(72)
--	--------

Por ello, se puede ver que $\nabla^2 G$ puede ser computado mediante diferencias finitas aproximando $dG/d\sigma$, usando la diferencia de escalas cercanas en un factor $k\sigma$ y σ :

$\frac{dG}{d\sigma} = \sigma \nabla^2 G \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$	(73)
--	------

Y por ello,

$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G$	(30)
---	------

Esto muestra que cuando la diferencia de funciones gaussianas se aplica a diferentes escalas mediante un factor constante ya se incorpora el factor σ^2 de normalización de escala requerido por el laplaciano invariante ante escalas.

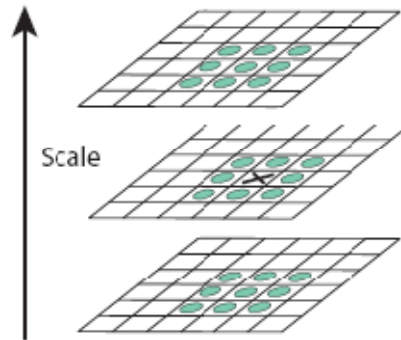


Figura 74. Puntos vecinos a un punto característico. Adaptada de David G. Lowe [13].

5.4.1.1. Detección de extremos locales

Para la detección de los máximos y mínimos locales de $D(x, y, \sigma)$, cada punto de muestra es comparado con sus ocho vecinos de la imagen y nueve vecinos en las escalas superior e inferior. Es seleccionado solo si es mayor que todos ellos o menor. El coste de la comprobación es razonablemente bajo comparado con el hecho de que la mayoría de los puntos de muestra serán eliminados en las primeras comprobaciones.

Una tarea importante es determinar la frecuencia de muestreo de la imagen y los dominios de escala que se necesitan para realizar la detección de extremos. Desafortunadamente, esto muestra que no hay un espaciado mínimo entre muestras para detectar todos los extremos ya que estos pueden estar arbitrariamente cerca. Por ello se debe elegir una solución que sea un compromiso entre eficiencia con completitud. De hecho, como se ha comprobado experimentalmente los extremos que están muy cercanos entre si son muy inestables ante pequeñas perturbaciones de la

imagen. Se pueden determinar las mejores aproximaciones experimentalmente estudiando un rango de frecuencias de muestreo y usando aquellas que proporcionan los resultados más satisfactorios [13].

5.4.1.2. Frecuencia de muestreo en la escala

La determinación experimental de la frecuencia de muestreo que maximiza la estabilidad se muestra en las figuras que siguen a continuación.

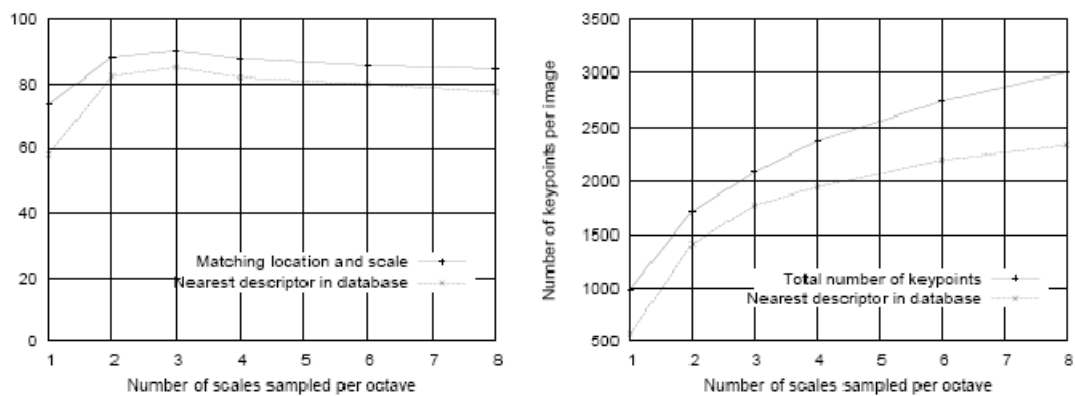


Figura 75. Determinación experimental de la frecuencia de muestreo en la escala. Adaptada de David G. Lowe [13].

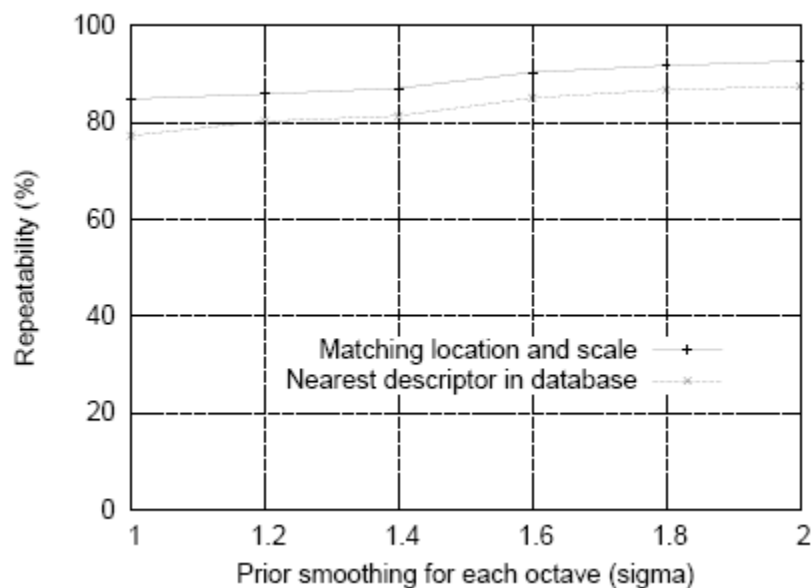


Figura 76. Repetitibilidad según la cantidad de suavizado a priori. Adaptada de David G. Lowe [13].

La Figura 76 muestra los resultados para una simulación usada para examinar el efecto de la variación del número de escalas por octava en la que la función de la imagen es muestreada para la detección de extremos. En este caso, cada imagen fue remuestreada seguida de una rotación de un ángulo aleatorio y escalada por una cantidad aleatoria entre 0.2 y 0.9 veces el tamaño original. Los puntos característicos de la resolución reducida fueron buscados en la imagen original de tal forma que todas las escalas de los puntos característicos estuvieran presentes en la imagen analizada. Además fue añadido un 1% de ruido a la imagen. La línea superior del primer gráfico muestra el porcentaje de puntos característicos que son detectados en un lugar y escala determinados en la imagen transformada. Para todos los ejemplos de este capítulo se definirá una escala de búsqueda con un factor de $\sqrt{2}$ respecto de la escala correcta y un lugar de búsqueda dentro de σ píxeles, donde σ es la escala de cada keypoint. La línea inferior en ese gráfico muestra el número de puntos característicos que han sido marcados correctamente dentro de una base de datos de 40.000 puntos característicos usando el procedimiento de marcado de vecinos más cercanos. Como este gráfico muestra, la mayor repetitividad se obtiene cuando se muestre a 3 escalas por octava.

Puede parecer sorprendente que la repetitividad no continua mejorando al aumentar el número de escalas muestreadas. La razón para ello es que son detectados muchos más extremos locales pero estos extremos son de media menos estables y por ello menos probables de ser detectados en la imagen transformada. Esto se muestra en el segundo gráfico que muestra el término medio de puntos característicos correctamente detectados para cada imagen. El número de puntos característicos crece cuando crece el número de escalas muestreadas y el número total de coincidencias correctas también crece. Se ha determinado que el número óptimo de escalas por octava que optimiza los resultados en base al coste computacional y eficiencia son 3 escalas por octava.

5.4.1.3. Frecuencia de muestreo en el dominio espacial

Así como se ha determinado la frecuencia de muestreo por octava del espacio de escalas se debe determinar la frecuencia de muestreo de la imagen relativa al suavizado de la misma.

Sabiendo que los extremos pueden presentarse arbitrariamente juntos debe existir un compromiso similar entre frecuencia y tasa de detección. La figura Figura 76 muestra la determinación experimental de la cantidad de suavizado a priori, σ , que se aplica a cada nivel de imagen antes de construir el espacio de escalas representativo de cada octava. De nuevo, la línea superior es la repetitividad de la detección de cada punto

característico, y los resultados muestran que la repetitividad continua incrementándose con σ . Sin embargo, hay un coste de usar una gran σ en términos de eficiencia, por eso se ha tomado habitualmente el valor de $\sigma=1.6$, el cual produce una repetitividad optima.

Efectivamente, si se pre-suaviza la imagen se están eliminando las altas frecuencias espaciales. Por ello, para hacer un uso total de la imagen esta puede ser expandida para crear más puntos de muestreo que los presentados de forma original. Habitualmente se doblará el tamaño de la imagen de entrada usando interpolación lineal para construir el primer nivel de la pirámide. Se asumirá que la imagen original tiene un nivel de suavizado previo de $\sigma=0.5$, y por ello la imagen doblada tendrá $\sigma=1$ relativo al nuevo espaciado de pixeles. Esto significa que un nivel adicional de suavizado es necesario a priori para crear la primera octava del espacio de escalas. El doblado de la imagen incrementa el número de puntos característicos estables por un factor de 4, pero no se han encontrado mejoras mayores aumentando la expansión.

5.4.2. Localización precisa de puntos característicos

Una vez que un punto característico candidato es encontrado comparando un pixel con sus vecinos, el siguiente paso es realizar una búsqueda detallada de los datos cercanos sobre el emplazamiento, escala y radio de las curvaturas principales. Esta información permite rechazar puntos de bajo contraste o que estén situados de forma pobre a lo largo de los bordes.

La implementación inicial de esta aproximación (Lowe, 1999) simplemente localiza puntos característicos en el lugar y escala de un punto central. Sin embargo, recientemente Brown ha desarrollado un método usando la expansión de Taylor de la función de espacio de escala, $D(x, y, \sigma)$, centrada de tal forma que el origen sea cada punto de muestreo:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (75)$$

Donde D y sus derivadas son evaluadas en el punto de muestreo y $x = (x, y, \sigma)^T$ es el offset desde dicho punto. La localización del extremo, \hat{x} , se determina tomando la derivada de esa función con respecto a x e igualándola a cero, dando

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (76)$$



Figura 77. Estas figuras representan las etapas en la detección de puntos característicos. La imagen superior izquierda de 233x189 píxeles es la imagen original. La imagen superior derecha representa la localización de 812 puntos característicos de los máximos y mínimos de la diferencia de funciones gaussianas. Los puntos característicos se muestran como vectores indicando escala, orientación y emplazamiento. La imagen inferior izquierda muestra el proceso después de aplicar un umbral de contraste mínimo permaneciendo 729 puntos característicos. Finalmente tras aplicar un umbral de radio de curvatura quedan 536 puntos característicos. Adaptada de David G. Lowe [13].

La función de valor de los extremos, $D(\hat{x})$, es muy útil para rechazar los extremos inestables con bajo contraste. Puede obtenerse de las anteriores ecuaciones, dando:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (77)$$

Habitualmente los extremos con un valor absoluto $|D(\hat{x})|$ menor de 0.03 son descartados (asumiendo que los valores de los píxeles en la imagen están en el rango $[0, 1]$).

5.4.2.1. Eliminación de los resultados de borde

Para estabilidad no es suficiente con rechazar puntos característicos de bajo contraste. La diferencia de gaussianas produce una fuerte respuesta a lo largo de los bordes incluso si el emplazamiento a lo largo del borde está mal determinado y por tanto es inestable ante pequeñas cantidades de ruido.

Un pico pobremente definido en la diferencia de gaussianas tendrá una gran curvatura principal a lo largo del borde pero una pequeña en dirección perpendicular. La curvatura principal puede ser calculada como una matriz Hessiana H de dimensiones 2×2 , calculada en el emplazamiento y escala del punto característico:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (78)$$

Las derivadas se estiman tomando diferencias entre los vecinos de los puntos de muestreo.

Los autovalores de H son proporcionales a las principales curvaturas de D . Se puede evitar calcular explícitamente los autovalores ya que solo interesa el cálculo de su cociente. Si se toma como α el autovalor mayor y β el autovalor menor. Entonces, si la suma de los autovalores es la traza de H y el producto es el determinante:

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \quad (31)$$

En el caso improbable de que el determinante sea negativo, las curvaturas tendrán diferentes signos por ello el punto será descartado como extremo. Si se toma r como el cociente entre el autovalor mayor y el menor, de forma que $\alpha = r\beta$, entonces:

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (80)$$

Lo cual depende solo del cociente entre los autovalores en lugar de sus valores individuales. La cantidad $(r + 1)^2/r$ es mínima cuando los dos autovalores son iguales

y se incrementa con r . Por ello para comprobar que el radio de las curvaturas principales es inferior a un determinado umbral, r , solo se necesita comprobar:

$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$	(81)
--	------

Estos cálculos son muy eficientes de realizar. Lo habitual es tomar $r = 10$, lo que elimina los puntos característicos que tienen un cociente entre las curvaturas principales mayor de 10.

5.4.3. Asignación de orientación

Asignando una orientación consistente a cada punto característico basada en las propiedades locales de la imagen se puede crear un buen descriptor de puntos característicos que represente la orientación relativa y produzca con ello una invarianza frente a la rotación de la imagen.

La escala de cada punto característico es usada para seleccionar la imagen suavizada de forma gaussiana más cercana en escala para que todos los cálculos se puedan realizar de forma invariante en escala. Para cada imagen de muestra, $L(x, y)$, a esa escala, la magnitud del gradiente $m(x, y)$, y orientación, $\theta(x, y)$, es precalculada usando diferencias de píxeles:

$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$	(82)
$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$	(83)

Se crea de esta forma un histograma de orientación de las orientaciones del gradiente de los puntos de muestra de la región de alrededor del punto característico. El histograma de orientación tiene 36 casillas cubriendo los 360 grados de todas las posibles orientaciones. Cada muestra añadida al histograma es pesado por la magnitud de su gradiente y mediante una ventana de pesos circular-gaussiana con un σ que es 1.5 veces la escala de cada punto característico.

Los picos en el histograma de orientación corresponden a direcciones dominantes de los gradientes locales. Se detecta con ello el máximo pico del histograma y junto con cualquier otro pico que este dentro del 80% del mayor se usa para crear la orientación del punto característico. Por ello, para emplazamientos como múltiples picos de

magnitud similar habrá múltiples puntos característicos creados en el mismo lugar y escala pero con diferentes orientaciones. Solo el 15% de los puntos característicos son asignados con múltiples orientaciones, pero esto contribuye de forma significativa a la estabilidad de la comparación. Finalmente una parábola es encajada entre los 3 valores del histograma más cercanos a cada pico para interpolar la posición del pico con mayor exactitud.

5.4.4. El descriptor local de la imagen

Las operaciones previas han asignado un emplazamiento a la imagen, escala, orientación a cada punto característico. Estos parámetros imponen un sistema de coordenadas local 2D repetible en el que describir la región local de la imagen y proporcionar con ello la invarianza ante esos parámetros. El siguiente paso es computar un descriptor para la imagen local que sea altamente descriptivo así como invariante ante posibles variaciones, como cambios en la iluminación o en el punto de vista.

Una aproximación obvia sería muestrear las intensidades locales de la imagen alrededor del punto característico a la escala adecuada y marcarlas usando una medida de correlación normalizada. Sin embargo, la simple correlación de muestras de imágenes es muy sensible a cambios causando pérdida de muestras. Una técnica mejor ha sido desarrollada basándose en la visión biológica, particularmente en las complejas neuronas del cortex visual primario.

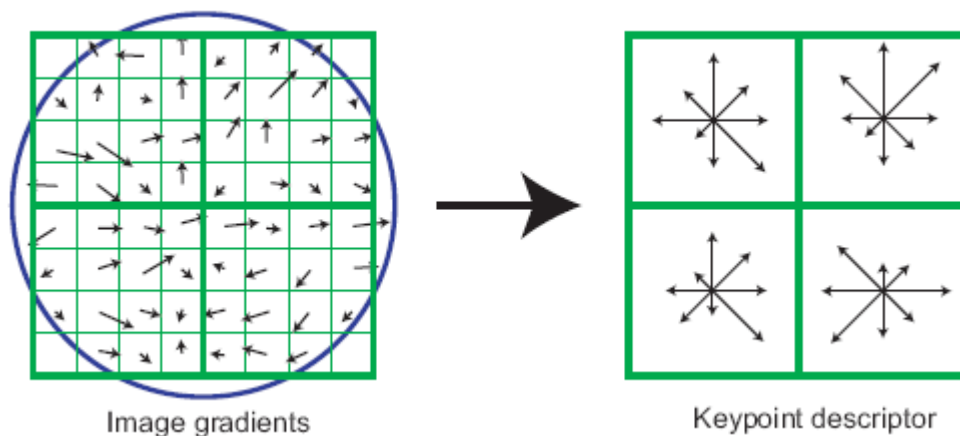


Figura 78. Representación del descriptor de puntos característicos. Adaptada de David G. Lowe [13].

Un descriptor de puntos característicos se crea primero calculando las magnitudes del gradiente y orientaciones de cada punto de prueba de la imagen en una región

alrededor del emplazamiento del punto característico como se muestra en la imagen izquierda. Son pesados mediante una ventana gaussiana indicada por el círculo. Estas muestras son acumuladas en un histograma sumando los contenidos de cada subregión 4x4, tal y como se muestra en la imagen derecha, con la longitud de cada flecha correspondiente a la suma de las magnitudes del gradiente para direcciones cercanas dentro de la región.

5.4.4.1. Representación del descriptor

Las figuras anteriores muestran el cálculo del descriptor de puntos característicos. Primero la magnitud y orientación del gradiente de la imagen son muestreadas en torno al emplazamiento del punto característico usando la escala del punto característico para seleccionar el nivel de suavizado gaussiano de la imagen. Para conseguir invarianza en la orientación, las coordenadas del descriptor y orientaciones del gradiente son rotadas relativamente a la orientación del punto característico. Para mayor eficiencia los gradientes son precomputados para todos los niveles de la pirámide descrita en la sección anterior. Este queda ilustrado con pequeñas flechas en cada punto de muestreo de la zona izquierda de la figura.

Una función de pesos gaussiana con σ igual a 1.5 veces la dimensión de la ventana del descriptor es usada para asignar un peso a la magnitud de cada punto de muestreo. Esto queda ilustrado con el círculo sobre la figura. El propósito de esta ventana gaussiana es evitar cambios súbitos en el descriptor con pequeños cambios en la posición de la ventana y dar menor énfasis a los gradientes que están lejos del centro del descriptor ya que están más afectados por errores de pérdida de información.

El descriptor de puntos característicos es mostrado en la parte derecha de la figura ¿?. Permite crear histogramas de 4x4 regiones de muestreo para cambios en las posiciones del gradiente. La figura muestra ocho direcciones para cada histograma de orientación, con la longitud de cada flecha correspondiente a la magnitud de dicha entrada del histograma. Una muestra del gradiente en la izquierda puede convertirse en 4 posiciones de muestreo mientras se sigue contribuyendo de la misma manera al histograma en la derecha, así se consigue el objetivo de permitir movimientos para mayor número de posiciones locales.

Es importante evitar todos los efectos de borde en los que el descriptor cambia súbitamente cuando se cambia el suavizado de una muestra para pasar de pertenecer de un histograma a otro o de una orientación a otra. Por ello, se usa la interpolación trilinear para distribuir el valor de cada muestra de gradiente en celdillas adyacentes en el histograma. En otras palabras, cada entrada en una celdilla es multiplicada por

un peso de $1 - d$ por cada dimensión, donde d es la distancia de la muestra dese el valor central de la celdilla medido en unidades de espaciado del histograma.

El descriptor se forma a partir de un vector que contiene los valores de todas las entradas de orientación del histograma, correspondiente a las longitudes de las flechas en la Figura 78. La figura muestra una matriz de histogramas de orientación 2x2. Los experimentos han demostrado que los mejores resultados se consiguen utilizando matrices de histogramas de orientación 4x4 con 8 celdillas de orientación en cada uno.

Finalmente, el vector de características es modificado para reducir los efectos de cambios en la iluminación.

5.4.4.2. Testeo del descriptor

Existen dos parámetros que pueden ser usados para variar la complejidad del descriptor: el numero de orientaciones, r , en los histogramas, y las dimensiones, n , de la matriz $n \times n$ de los histogramas de orientación. El tamaño del descriptor resultante es rn^2 . Cuando la complejidad del descriptor crece se puede discriminar mejor en una gran base de datos, pero también se vuelve más sensible a distorsiones de forma u oclusión.

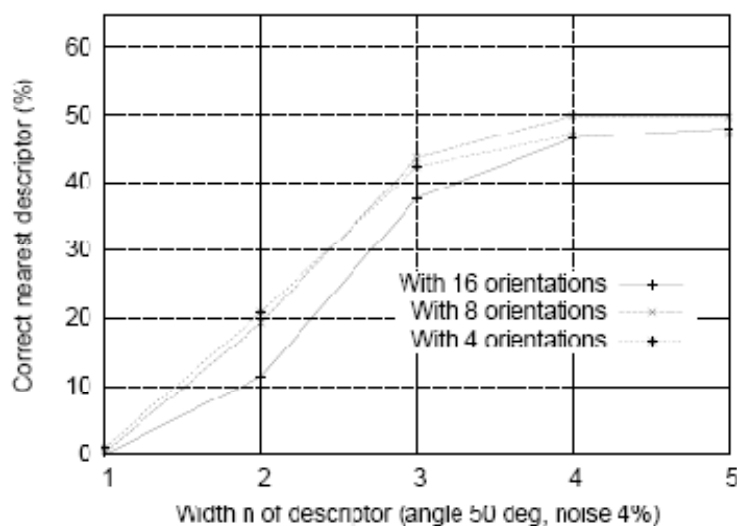


Figura 79. Porcentaje de puntos característicos encajados correctamente según varía el tamaño del descriptor y el número de orientaciones. Adaptada de David G. Lowe [13].

La Figura 79 muestra resultados experimentales en los que el número de orientaciones y el tamaño del descriptor varían. El grafico fue generado para un punto

de vista transformado en el cual una superficie plana es girada 50° alejándose del punto de vista frontal y añadiendo un 4% de ruido a la imagen. Esto se encuentra cerca de los límites de la búsqueda eficiente, y es precisamente en estos casos difíciles donde se requiere alta eficiencia en el descriptor. Los resultados muestran el porcentaje de puntos característicos que se encajan correctamente con el vecino más cercano entre una base de datos de 40.000 puntos característicos. El gráfico muestra que histogramas con una única orientación ($n=1$) son muy poco discriminantes, pero los resultados continúan mejorando hasta tener una matriz de 4×4 de histogramas de orientación con 8 orientaciones. Después de esto, añadir más orientaciones o un descriptor mayor pueden incluso empeorar los resultados haciéndolos más sensibles a distorsiones.

Búsqueda de correspondencias en grandes bases de datos

Una característica importante para la medida de la diferenciación de características es como varía la eficiencia de la búsqueda en función del número de elementos en la base de datos que son comparados.

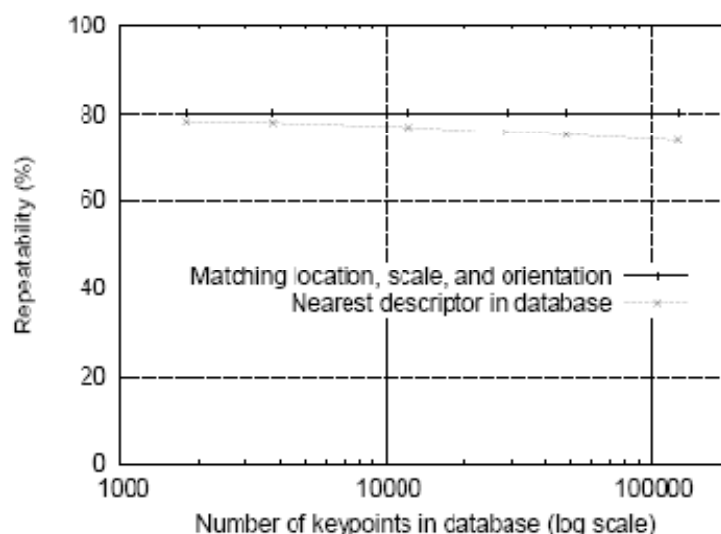


Figura 80. Eficiencia según el número de puntos característicos almacenados en la base de datos. Adaptada de David G. Lowe [13].

La Figura 80 muestra como la eficiencia varía en función del tamaño de la base de datos. La línea discontinua inferior muestra la porción de características de la imagen para las cuales el vecino más cercano de la base de datos fue marcado correctamente, mostrando el tamaño de la base de datos de forma logarítmica. El punto más a la izquierda está marcado únicamente frente a una única imagen, mientras que el punto del lado derecho esta marcado contra una base de datos de todas las características de 112 imágenes. Se puede apreciar como la eficiencia del marcado no decrece en función del número de elementos de distracción, por lo que

todo indica que el éxito de las operaciones surge aunque se aumente mucho el tamaño de la base de datos.

La línea continua muestra el porcentaje de puntos característicos que han sido identificados correctamente en la imagen transformada. La razón de que esta línea sea horizontal es que el test fue ejecutado sobre una base de datos repleta para cada valor, mientras que solo se variaba la porción de datos usada para los elementos distractores. Es de interés que el hueco entre las dos líneas sea pequeño, indicando que todos los fallos del emparejamiento se deben más a problemas con la función de localización inicial y asignación de orientación que problemas de diferenciación, incluso con tamaños muy grandes de bases de datos.

5.5. PRUEBAS REALIZADAS CON SLAM VISUAL A TRAVÉS DEL MÉTODO SIFT

El método SIFT es un método de reconocimiento de características de una imagen invariantes frente a cambios en la escala, rotaciones y parcialmente invariantes ante cambios en la iluminación y en el punto de vista tridimensional. Este algoritmo se puede aplicar para el reconocimiento de objetos dentro de una escena o a la localización de un determinado objeto dentro de otra imagen, o a una multitud de aplicaciones de la visión por computador.

Sin embargo, la aplicación a la localización de un robot del método SIFT es aparentemente más sencilla que la funcionalidad original puesto que los datos que se tienen de partida son dos mapas donde los cambios que se producen de una imagen respecto a la original son aparentemente menos significativos que en la aplicación para reconocimiento de imágenes.

La idea principal reside en la comparación de dos mapas mediante el algoritmo visual SIFT. Los dos mapas que se van a comparar extrayendo sus características serán el mapa global de baja resolución (LRM) y el mapa local que se reconstruirá tomando como referencia el propio robot (Mapa local auxiliar). Al igual que en las técnicas de localización por métodos de comparación numéricos que se han comentado en apartados anteriores se necesita la representación local del robot como punto de partida para la búsqueda. Esta vez no es necesario ni conveniente realizar la reconstrucción respecto al punto característico de la celda donde el robot cree estar como se realizó en el caso anterior.

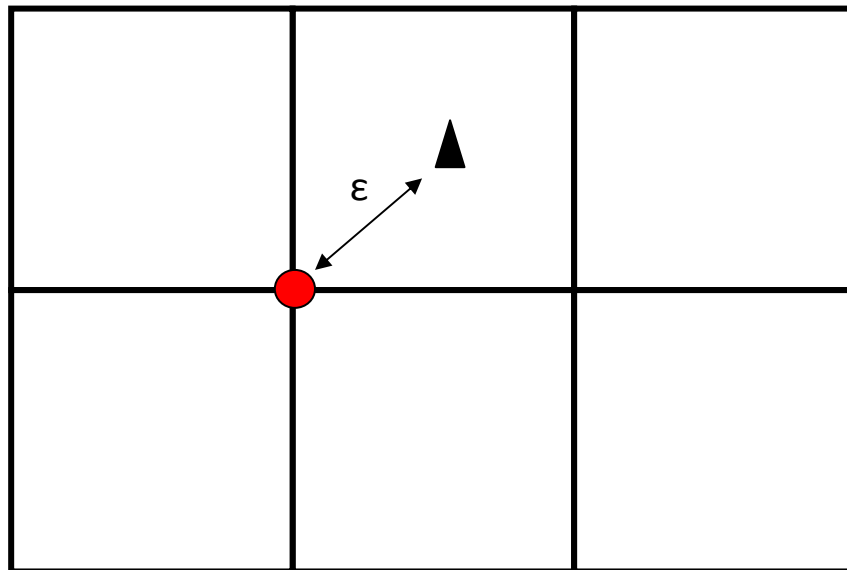


Figura 81. Esquema del offset utilizado en los cálculos de localización por comparación analítica de mapas

En este caso, no se van a superponer dos matrices para realizar la comparación, sino que se utilizarán otros tipos de algoritmos para identificar la posición del robot relativa a los puntos característicos encontrados en ambos mapas y así localizar el robot.

La idea básica para aplicar el método SIFT a la localización del robot se basa en encontrar una serie de puntos de la representación local en el interior del mapa global. Conseguido esto es sencillo posicionar al robot, puesto que al saber las distancias relativas del robot a un punto de su mapa local podría calcular su posición en el mapa global mediante la aplicación de la escala correspondiente y la equivalente distancia sobre el mapa global. Por ejemplo:

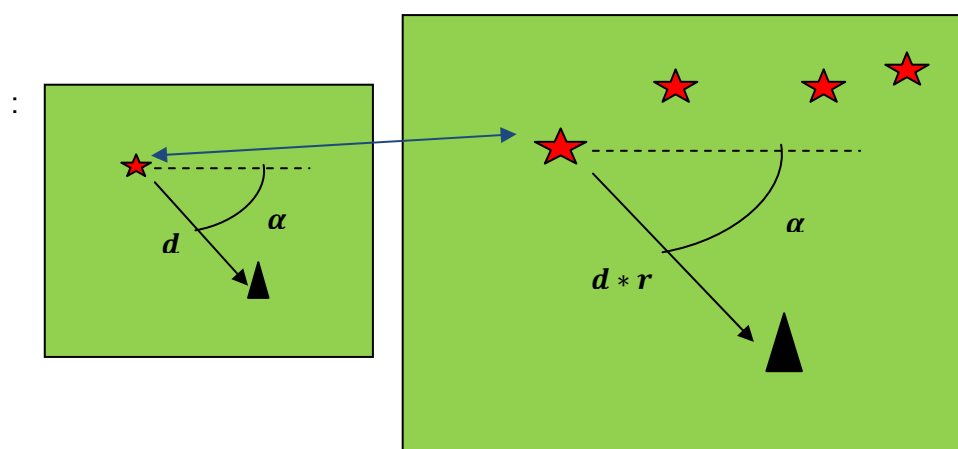


Figura 82. Ejemplo de posicionamiento conocida la distancia y el ángulo a un punto característico

Si se logra encontrar la correspondencia entre dos puntos característicos de ambos mapas mediante el método SIFT, conociendo la posición relativa del punto característico local respecto del robot es serán conocidos por tanto la distancia y el ángulo de dicho punto al robot. Conociendo la posición del mismo punto en el mapa global, basta realizar una transformación lineal mediante el cociente de escalas para hallar la posición actual del robot en el interior del mapa global.

Para mayor precisión se puede repetir el cálculo con todos los puntos característicos equivalentes encontrados y hallar el punto medio de todas las posiciones del robot obtenidas y tomar la media como posición estimada de salida del bloque de SLAM visual.

5.5.1. Aplicación del método SIFT a las estructuras de mapas

El método SIFT es un método de reconocimiento de características de imágenes, por tanto necesita una imagen como entrada. Una imagen es fundamentalmente una matriz de tres capas donde cada elemento corresponde a un pixel y cada capa lleva asociado un valor de intensidad de cada color, para los colores rojo, verde y azul (RGB).

Las matrices con las que se trabaja en el mapeado del robot llevan información sobre el terreno en forma de valores numéricos reales. Lo más lógico parece ser trabajar con las capas de alturas o gradientes porque representan una información que es fácilmente comparable con el mapa recibido como entrada a priori para el funcionamiento del robot. No tendría mucho sentido trabajar con la imagen de la señal GPS si puede cambiar con el tiempo con cierta frecuencia, o no es conocida hasta que el robot llega a un determinado punto. Tampoco parece lógico trabajar con la capa de confianza puesto que es el robot quien rellena esa capa en función de la forma en la que impacta el láser sobre el terreno. Por lo tanto interesa trabajar con información que se pueda obtener a priori con cierta precisión como la capa de alturas.

Como requisito de entrada para el algoritmo SIFT se requiere que la imagen sea una imagen en escala de grises normalizada, por lo tanto los valores de cada pixel tienen que ser números positivos en el intervalo [0, 1]. Es fácil normalizar los valores de la capa de alturas para que cumplan esos requisitos mediante las siguientes operaciones:

$Imagen = Imagen - \min (Imagen)$	(84)
-----------------------------------	-------

$Imagen = \frac{Imagen}{\max(Imagen)}$	(85)
--	-------

Tras estas operaciones la matriz queda normalizada y se ajusta perfectamente a las condiciones de entrada.

Hay que tener en cuenta que los mapas con los que se trabaja habitualmente en las simulaciones tienen unas dimensiones típicas del orden de 100x100 hasta 500x500, con lo que tomados como imágenes representan un tamaño que es relativamente pequeño para las imágenes habituales que se utilizan hoy en día. Por ese motivo esto va a producir que el algoritmo se ejecute de forma más rápida que la diseñada para el uso original. Esto es un punto a favor, pero hay otro muy importante en contra relacionado con el pequeño tamaño y detalle de la imagen que será discutido más adelante.

Al igual que ocurría con el algoritmo de celdas, realizado por medio de métricas y búsqueda del mínimo desarrollado en el apartado anterior, donde se reducía el ámbito de búsqueda del mapa, aquí ocurre lo mismo. Se debe reducir el área de búsqueda en el interior del LRM a un cuadrado donde la probabilidad de que se encuentre el robot sea muy elevada acotando las dimensiones con el error máximo esperable por parte de la fusión de la odometría con los sensores inerciales. Se habían utilizado unos valores de ± 7 metros en el apartado anterior en cuanto al error máximo admisible, por lo que el cuadrado a utilizar tiene 14 m de lado. Se utilizarán aquí otras dimensiones superiores por lo que el mapa local a buscar dentro del LRM podría tener la siguiente forma:

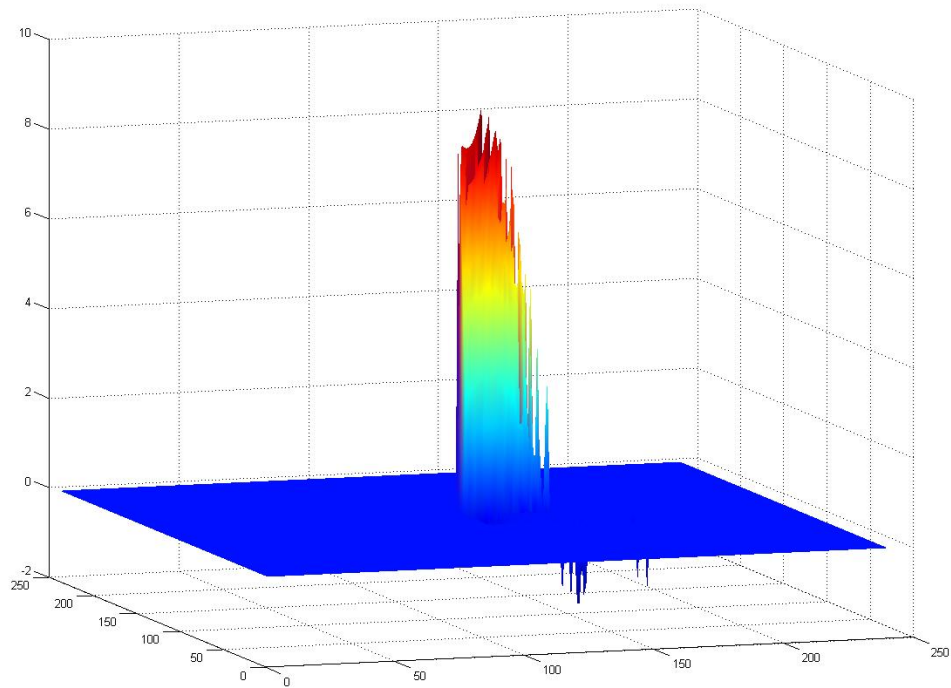


Figura 83. Reconstrucción de un mapa local utilizando una única medida del láser

Mientras que el LRM donde se va a realizar la búsqueda posee la siguiente estructura:

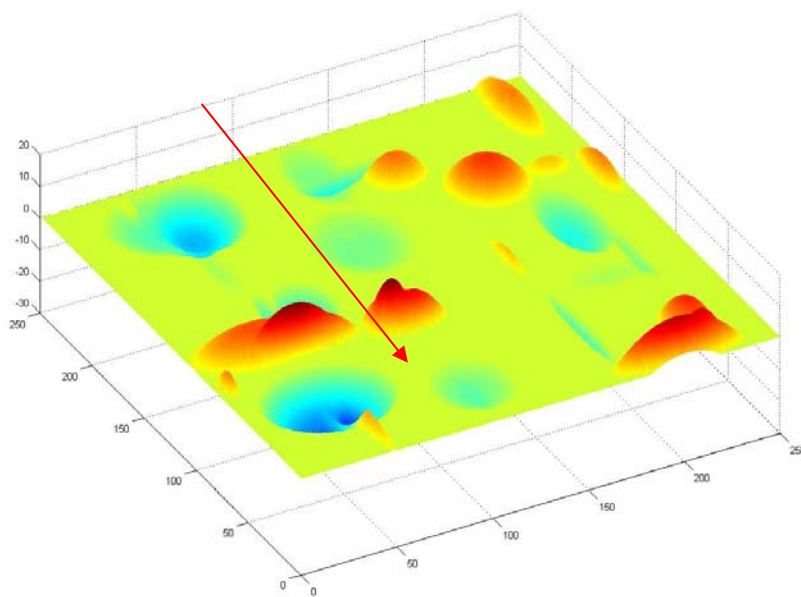


Figura 84. LRM con indicación de la zona reconstruida de forma local

En este caso, el robot se encuentra situado donde indica la punta de flecha por lo que la reconstrucción parcial realizada con una única medida del láser corresponde a esa área.

Debido a que el reconocimiento de imágenes debe funcionar en un área mayor que las técnicas de comparación con métricas se debe establecer un área de búsqueda mayor para que la imagen tenga suficiente información para la búsqueda de puntos característicos. Se puede establecer por ejemplo un área de 200x200 metros de búsqueda, quedando ambos mapas de la forma que se muestra a continuación:



Figura 85. Eliminación del área de la imagen que se escapa del error máximo admisible por el sistema de posicionamiento odometría-ins

Se aprecia como la imagen de la izquierda corresponde al mapa de baja resolución global donde se han eliminado las zonas alejadas de la posición del robot una distancia mayor de un cuadrado de 200 m de lado. Esto se hace para que habiendo suficiente información en la imagen no se interfiera en la búsqueda de puntos característicos con zonas de la imagen donde es imposible que el robot este emplazado.

5.5.2. Configuración de los parámetros del método SIFT

Para los ensayos realizados se ha utilizado una implementación del descriptor SIFT realizada por Andrea Vedaldi para la Universidad de California en Los Ángeles cuyo uso es libre con fines educativos y de investigación [14].

La implementación está diseñada para producir resultados compatibles con la versión de Lowe y está diseñada para el entorno de Matlab. Dicha implementación se divide en dos partes diferenciadas, SIFT detector y SIFT descriptor. El primero de ellos extrae de la imagen una colección de puntos característicos y el segundo se

encarga de hacer una descripción extensa del entorno de cada punto característico, consiguiendo mediante canonización que el descriptor sea invariante frente a traslaciones, rotaciones y escalado y siendo robusto frente a pequeñas distorsiones.

El software devuelve una matriz 4 x K conteniendo la referencia a cada punto y una matriz 128 x K conteniendo los descriptores para cada uno. Cada referencia está caracterizada por 4 números que están ordenados de la siguiente forma: (x1, x2) son los dos primeros valores que corresponden a las coordenadas del punto característico respecto a la esquina superior izquierda de la imagen, σ es la escala a la que ha sido hallado éste y por último θ que es su orientación. La esquina superior izquierda toma los valores (0, 0) y las coordenadas (x1, x2) pueden ser fraccionarias lo que dota al sistema de precisión sub-píxel. La escala σ es el nivel de suavizado en el que la característica ha sido encontrada. Este valor puede ser interpretado como el tamaño del entorno del punto característico, el cual es visualizado como un disco de tamaño 6σ . Cada descriptor es un vector que describe toscamente la apariencia del trozo de imagen correspondiente al entorno de cada punto. Típicamente este vector tiene una dimensión de 128, pero este valor puede ser cambiado.

Las posibilidades de configuración del software son múltiples, pero los valores por defecto han sido elegidos para emular la implementación original de Lowe.

El detector y descriptor SIFT han sido contruidos a partir de la función espacio de escalas gaussiana de la imagen $I(x)$. La función de espacio de escala gaussiana $G(x, \sigma)$ que ha sido ampliamente descrita en apartados anteriores representa la misma información que $I(x)$ pero a distintos niveles de escala muestreada de una forma particular para reducir la redundancia. El dominio de la variable σ es discretizado de forma logarítmica en O octavas. Cada octava es dividida posteriormente en S subniveles. La distinción entre octava y subnivel es importante puesto que a cada octava consecutiva los datos son espacialmente muestreados a la mitad. Las octavas y los subniveles son identificados por un índice de octava discreto y un subnivel s respectivamente. Las octavas o y los subniveles se rigen por la siguiente expresión:

$$\sigma(o, s) = \sigma_o 2^{o+s/S}, \quad o \in o_{min} + [0, \dots, O - 1], \quad s \in [0, \dots, S - 1]$$

Donde σ_o es la base del nivel de escalas. El software empleado admite los siguientes parámetros para las operaciones iniciales:

- Numero de octavas: O .
- Primera octava: índice de la primera octava o_{min} . Usualmente se toma como 0 o -1. Tomando o_{min} como -1 se consigue el efecto de doblar el tamaño de la imagen antes de computar la función de espacio de escala gaussiana.

- Numero de subniveles: S
- Suavizado base: σ_o
- Suavizado a priori nominal: σ_n . El algoritmo asume que la imagen de entrada esta previamente convolucionada $(g\sigma_n * I)(x)$ en oposici3n a $I(x)$ y se ajustan los c3lculos de forma adecuada. Usualmente se asume que $\sigma_n = 0.5$ p3xeles.

Otros par3metros importantes del detector son:

- Threshold: el umbral para los extremos encontrados. Extremos locales cuyo valor sea inferior a $|G(x, \sigma)|$ son rechazados.
- Edge Threshold: el umbral de borde. Si el extremo local est3 en un valle el algoritmo lo descarta por ser muy inestable. Los extremos est3n asociados con un valor proporcional a su nitidez y son rechazados si esa nitidez es inferior al umbral.

Par3metros de configuraci3n del descriptor:

- Magnificaci3n: el factor de magnificaci3n m est3 relacionado con el tama1o de cada celda espacial. Cada spatial bin del histograma tiene un tama1o $m\sigma$, donde σ es la escala del punto caracter3stico.
- Numero de celdas espaciales: este n3mero define la extensi3n y la dimensi3n del descriptor. La dimensi3n es igual a $NumSpatialBins^2 * NumOrientedBins$ y su extensi3n tiene un radio $NumSpatialBins * m\sigma/2$
- Numero de celdas de orientaci3n

Los valores para los par3metros que se han usado inicialmente son los originalmente propuestos por Lowe:

- $S = 3$
- $o_{min} = -1$
- $\sigma_o = 1.6 * 2^{1/S}$
- $\sigma_n = 0.5$
- $O = floor(\log_2 \min(M, N)) - o_{min} - 3$

- $Threshold = 0.04/(2 * S)$
- $Edge\ Threshold = 10$
- $Numero\ de\ celdas\ espaciales = 4$
- $Numero\ de\ celdas\ de\ orientación = 8$
- $Magnificación = 3$

5.5.3. Ensayos realizados con imágenes

Ensayo n°1

Con los parámetros recomendados por Lowe se obtienen demasiados puntos característicos como para poder ilustrar gráficamente los resultados, pero simplemente cambiando el número de celdas espaciales y de orientación, situándolos ambos a 16 celdas se obtiene un resultado muy bueno para dos imágenes de muestra.

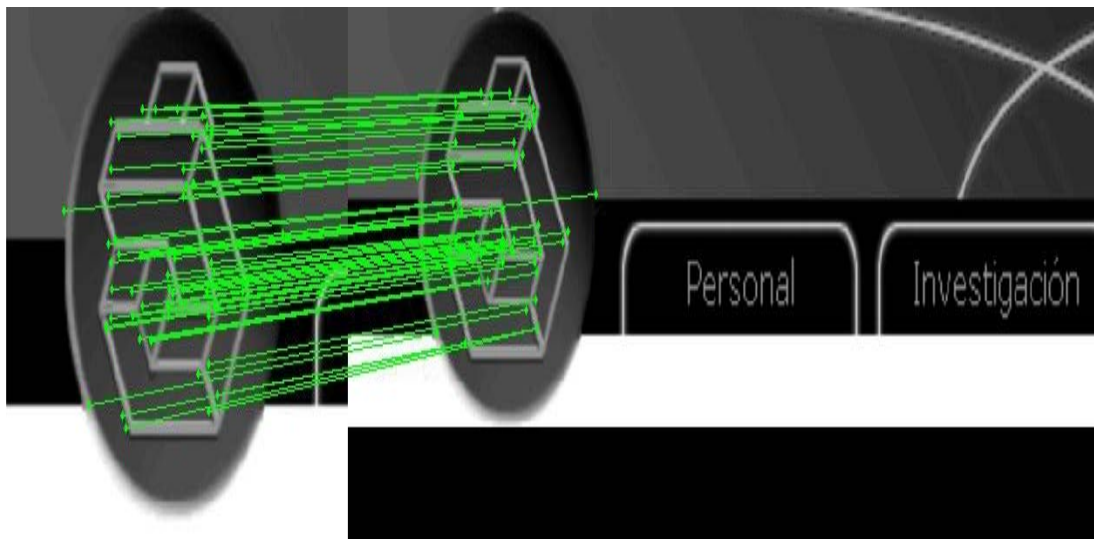


Figura 86. Búsqueda de equivalentes en imágenes con los datos del ensayo n° 1

Se puede observar como el sistema es capaz de detectar la imagen de la izquierda en la imagen de la derecha, estando ambas a distinta escala. Además estando la imagen derecha incluida en un fondo con otro contenido independiente del logotipo que se intenta reconocer. Es evidente que los resultados son muy buenos con este tipo de imágenes aunque no es un método infalible y se producen correspondencias falsas que deberían ser eliminadas por algún otro método.

Si se modifican los parámetros de tal forma que se haga más precisa la búsqueda de puntos característicos sacrificando con ello el número de concordancias encontradas pero mejorando la exactitud de la búsqueda se obtiene el siguiente resultado:

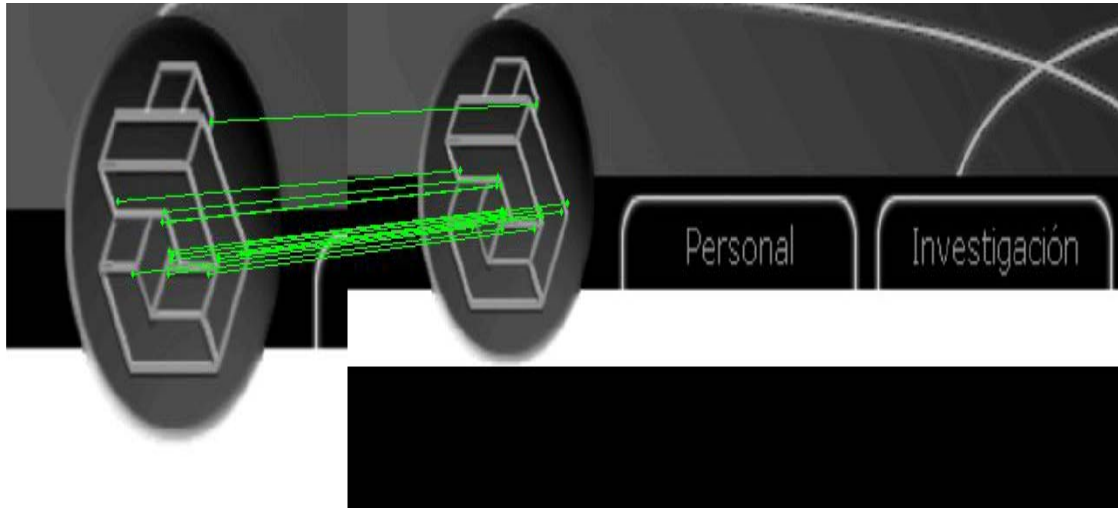


Figura 87. Búsqueda de equivalentes en imágenes utilizando parámetros más restrictivos

Siendo los parámetros modificados los siguientes:

- $S=5$
- Numero de celdas espaciales=20
- Numero de celdas de orientación=20
- Magnificación=5

Los resultados son muy buenos tras estas modificaciones pudiendo afirmar que el método es muy adecuado para este tipo de imágenes.

Los problemas que se han encontrado están relacionados fundamentalmente con la variación de los parámetros originales. Dependiendo de la complejidad y características de la imagen a analizar es conveniente modificar los parámetros originales de forma manual para aumentar la exactitud de los resultados. Aun no se ha dado con una forma automática de modificación de dichos parámetros, por lo que los ensayos basados en pruebas son actualmente las únicas alternativas razonables. Por ello surge la dificultad de implementación de este método para cualquier tipo de imágenes.

Ensayo nº 2

En este caso, situando los parámetros de la misma forma que en el ensayo nº1, es decir:

- $S=5$
- Numero de celdas espaciales=20
- Numero de celdas de orientación=20
- Magnificación=5

Se obtiene el siguiente resultado:

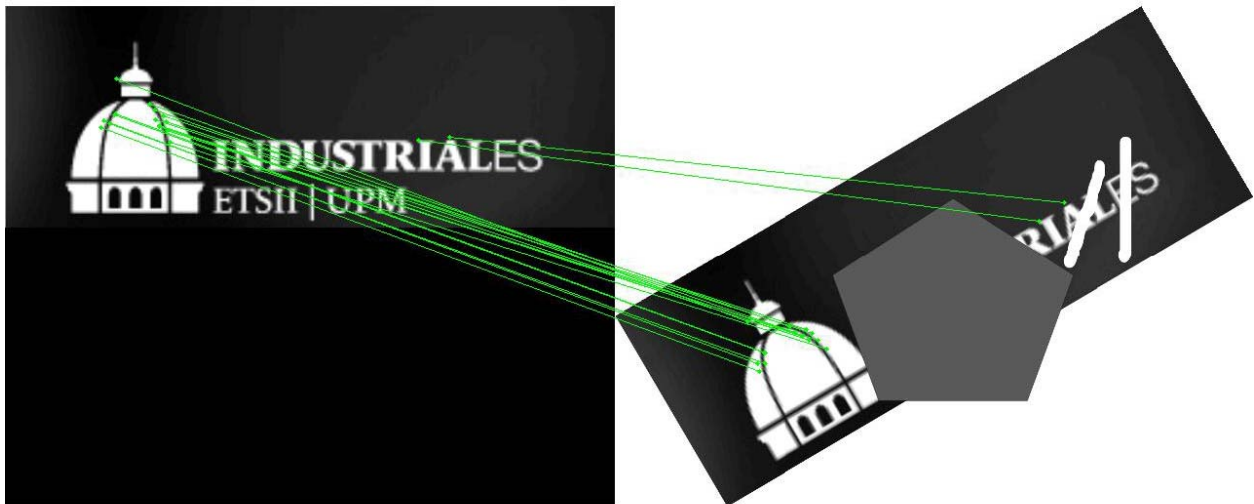


Figura 88. Búsqueda de equivalentes en imágenes utilizando una imagen girada y con áreas eliminadas según la configuración de parámetros restrictiva

Como se puede apreciar los resultados son increíblemente buenos y el algoritmo es capaz de distinguir puntos equivalentes aun cuando se produce una rotación de la imagen y una distorsión de ciertas zonas de la misma. Se han utilizado estos parámetros para la mejor representación grafica de los resultados, porque empleando la configuración inicial dada por Lowe, los resultados son más difíciles de verificar:

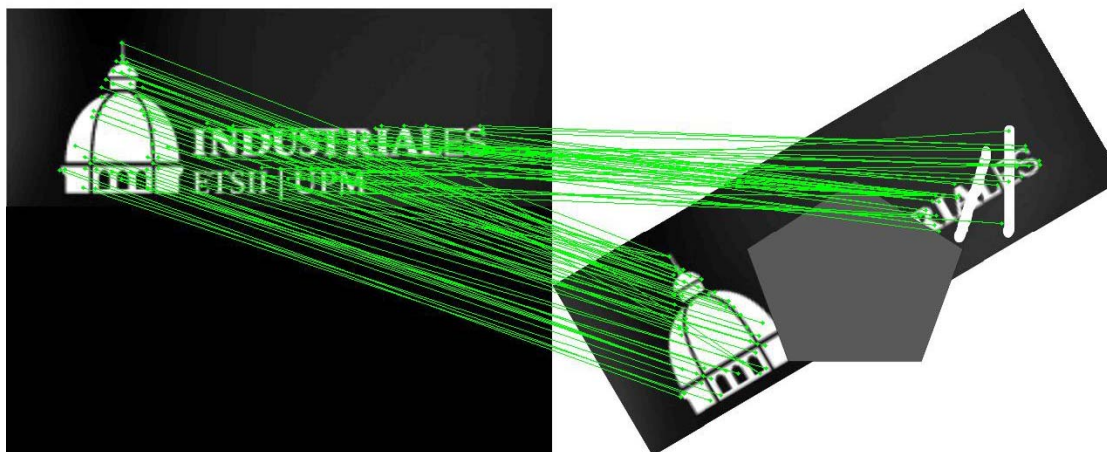


Figura 89. Búsqueda de equivalentes en imágenes utilizando una imagen girada y con áreas eliminadas utilizando los parámetros originales de Lowe

En este último caso se aprecia que se señalan como puntos característicos validos en la imagen derecha zonas de las dos líneas en blanco que no tienen correspondencia real con la imagen izquierda, por lo tanto el algoritmo falla en algunos puntos. Haciendo menos exigente los criterios de comparación del descriptor se producen casos de identificación falsos de puntos característicos. Pero con la configuración expuesta anteriormente no se tenían estos problemas.

En cualquier caso se puede concluir que ensayando la configuración adecuada de parámetros se puede conseguir que el reconocimiento de ciertos puntos característicos en una imagen sea prácticamente infalible.

Ensayo nº3

Utilizando la misma imagen que en el ensayo anterior se añade una proporción de 75% de ruido de distribución uniforme sobre la imagen original izquierda. La adición de ruido se realiza mediante un conocido programa de retoque fotográfico. Se sitúan los parámetros según la configuración más exigente, es decir, de nuevo:

- $S=5$
- Numero de celdas espaciales=20
- Numero de celdas de orientación=20
- Magnificación=5

Mientras que todos los demás permanecen idénticos a los recomendados por Lowe. Observando los resultados se aprecia que la cuantía de puntos característicos se reduce en gran medida sin embargo la precisión sigue siendo buena.

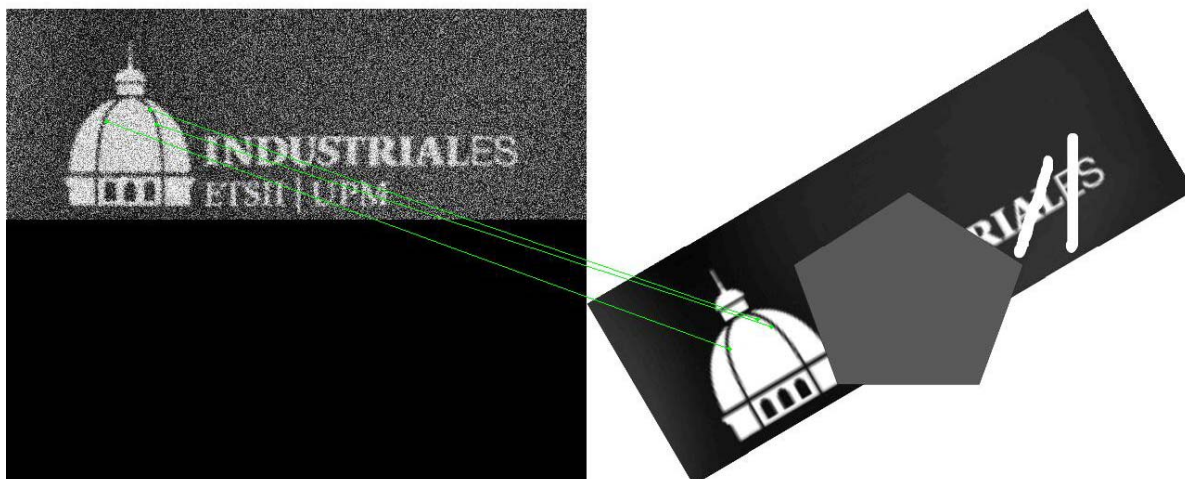


Figura 90. Búsqueda de equivalentes en imágenes utilizando en la imagen izquierda un 75% de ruido uniforme. Imagen izquierda rotada y con áreas eliminadas.

Los puntos característicos seleccionados en la imagen izquierda corresponden fielmente a los seleccionados en la imagen derecha, el algoritmo vuelve a dar buenos resultados con este tipo de imágenes aun empeorando en gran medida las condiciones de reconocimiento.

5.5.4. Ensayos realizados con mapas en condiciones operativas

Debido a los buenos resultados cosechados con imágenes cualesquiera obtenidas de la web era esperable que se mantuvieran los resultados con imágenes de contenido mucho más sencillo como son las matrices usadas para la reconstrucción en el proceso de mapeado.

En los siguientes ejemplos se van a usar mapas similares a los usados en el apartado inicial donde se explicaba los principios de aplicación del método SIFT a las estructuras de mapas. Utilizando la configuración dada por Lowe para la comparación de la reconstrucción de un mapa local con el LRM recortado se obtienen el siguiente resultado:

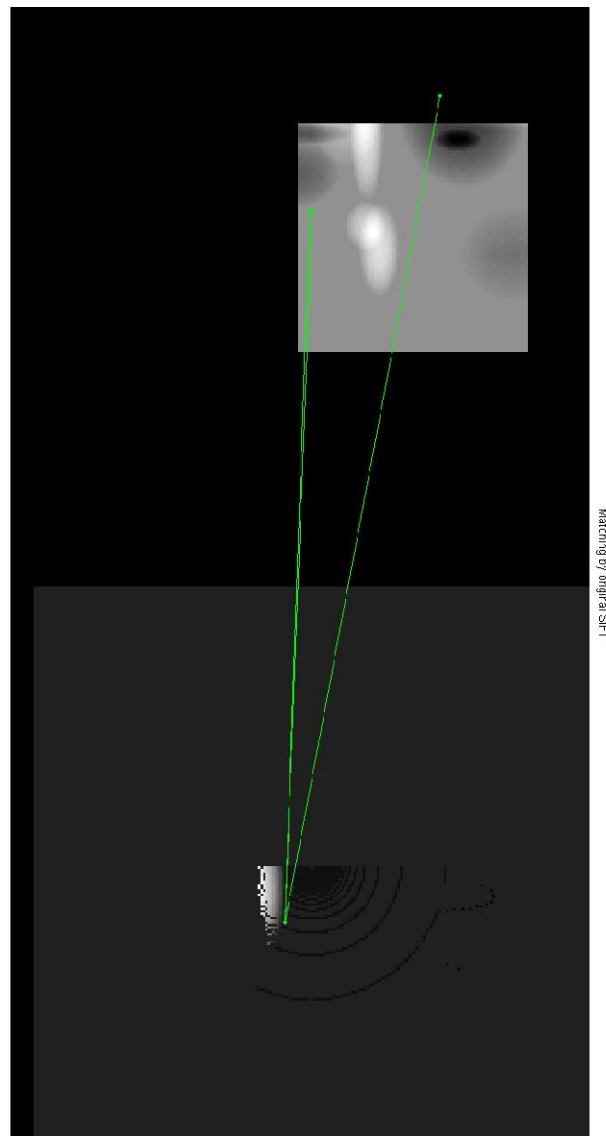


Figura 91. Método SIFT aplicado al reconocimiento de la posición del mapa local en el interior del LRM

Se puede ver claramente que los resultados son muy malos con este tipo de configuración. Mientras que en la imagen izquierda se detectan 2 puntos característicos, en la imagen derecha únicamente se detecta 1, por lo que se hacen corresponder ambos al mismo punto, pero ninguno de ellos está correctamente marcado.

Ante estos resultados desalentadores se optó por probar las configuraciones que habían producido éxito en las imágenes anteriores. Los resultados fueron aun más negativos pues no solo no se produjo ninguna coincidencia de puntos característicos sino que no se detectaron puntos característicos en ninguna de las imágenes.

Esto parece lógico si se tiene en cuenta que las configuraciones que habían producido buenos resultados en las anteriores pruebas con imágenes reales eran

configuraciones consideradas más restrictivas. Por lo tanto se decidió realizar pruebas con configuraciones menos restrictivas que la original de Lowe, modificando los parámetros de tal forma que produzcan más resultados aunque estos sean a priori de menor precisión. Para ello se optó por unos parámetros de los siguientes valores:

- $S=6$
- Numero de celdas espaciales=4
- Numero de celdas de orientación=4
- Magnificación=2

Los resultados continúan siendo inaceptables para los mapas, no detectándose nada más que un único punto característico en el mapa local que se hace corresponder a un gran número de ellos en el LRM. Aun así ninguno de los puntos seleccionados de la primera imagen corresponde realmente al de la imagen derecha. Solo hay uno de ellos que está muy próximo, y aun considerándolo como bueno la eficacia del método dista mucho de la conseguida con imágenes fotográficas reales.

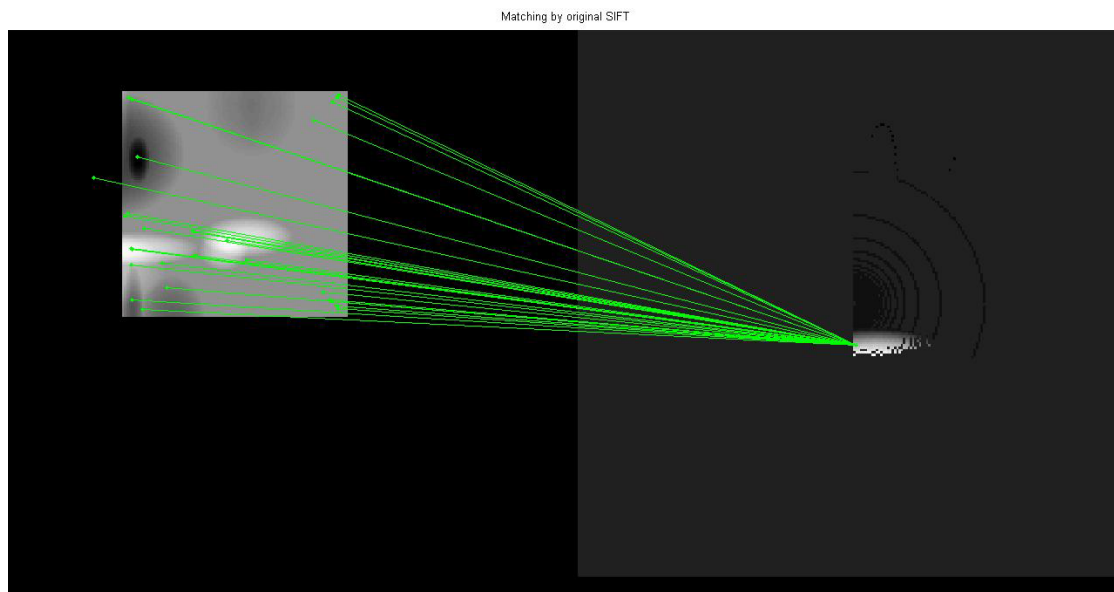


Figura 92. Método SIFT aplicado al reconocimiento de la posición del mapa local en el interior del LRM con la configuración original propuesta por Lowe

Los ensayos de este tipo han sido muy numerosos dando todos ellos un resultado muy negativo. Aun cuando se producía algún atisbo de correspondencia entre ambos mapas se encontró que al variar de mapa local se necesitaba cambiar la configuración de los parámetros personalizándolo para cada uno.

5.5.5. Conclusiones sobre el método SIFT aplicado a la localización del robot

La conclusión principal es que no se ha podido llevar a la práctica el método SIFT para la localización del robot utilizando la comparación de una reconstrucción local con el mapa global en baja resolución. Con los resultados tan positivos que se obtuvieron en los ensayos con fotografías reales parece lógico pensar que el método SIFT debe funcionar con imágenes como las de los mapas usados pero la realidad de los experimentos muestra otra cosa.

Las razones más probables para el fracaso de la aplicación de esta técnica parecen residir en la insuficiente información que proporciona la reconstrucción de una única lectura del láser para ser comparada por métodos gráficos. Hay que tener en cuenta que este método está pensado para la comparación de imagen con un número de píxeles mucho mayor al número de celdas que resultan con información tras una lectura del láser. Por ese motivo se considera que la información es demasiado pobre, ya que no proporciona un contraste suficiente entre las diferentes áreas de la zona reconstruida. Es probable también que la tipología de los mapas simulados basados en parábolas dificulte aun mas las tareas de reconocimiento puesto que una parábola vista en planta no tiene apenas elementos diferenciadores de contorno en su interior como para que el algoritmo pueda hallar elementos característicos.

Es previsible que para futuros desarrollos basados en el método SIFT para localización al conseguir mejorar el realismo en la simulación del entorno se consiga aumentar la concordancia de puntos característicos aumentando con ello las probabilidades de éxito del método en la localización. Además sería conveniente un intento de enriquecer el mapa local reconstruido de tal forma de poder acumular varias pasadas del láser con el robot estático de tal forma que se puede utilizar la información fusionada por todas ellas para aumentar la riqueza de la imagen y con ello el número de puntos característicos encontrados.

6. CONCLUSIONES

En el presente proyecto se han desarrollado algoritmos de reconstrucción preparados para su uso en un entorno simulado por un robot que posee un laser y GPS como elemento de localización principal. Además se han desarrollado herramientas para la simulación del entorno y las pruebas del mencionado algoritmo.

A continuación se describen las principales contribuciones del proyecto:

Algoritmos de reconstrucción

Durante el proyecto se han desarrollado algoritmos de reconstrucción para entornos exteriores con una calidad de posicionamiento muy elevada ya que se contaba con GPS. Se ha comprobado el correcto funcionamiento de los mismos para las condiciones de simulación y como los algoritmos de fusión de la información se comportan de forma exitosa con el sistema de mapas, así como el traslado del mapa local sobre el mapa global y las interacciones entre ellos.

Se han podido estimar los gradientes de un terreno de una forma adecuada para la planificación de la ruta por parte del robot, dando un nivel de ocupación adecuado para los requisitos del resto del proyecto NM-RS. Además se han utilizado unas técnicas llamadas de “asignación de libertad” para recabar información de los puntos de no impacto lo que proporciona un aumento de la información respecto las técnicas convencionales donde se desaprovechaban los datos de los mencionados puntos al no proporcionar información certera sobre el terreno y obstáculos.

Se ha desarrollado un método de medición de la confianza de la información aportada en una celda basado en la dispersión de los puntos sobre la misma lo que introduce una ventaja sobre algoritmos que solo tienen en cuenta el numero de impactos totales.

El método desarrollado de fusión de la información entre las medidas del laser y el HRM así como entre este y el LRM ha demostrado ser muy eficaz para la actualización de la información, con una rapidez de actualización aceptable y parametrizable lo que le dota de gran dinamismo. Además la no necesidad de almacenamiento indefinido de la información es un motivo importante para la implementación de dicho algoritmo.

Algoritmos de localización

Durante la realización de este proyecto se han probado dos métodos para ayudar al robot en la localización cuando se carece de señal GPS.

El método basado en comparación de mapas ha resultado ser una ayuda importante a la localización en ausencia de señal GPS aunque por otro lado es más impreciso de lo que se había esperado inicialmente. No obstante, se produce una mejora significativa en la comparación a nivel sub-celda respecto a la solución aportada en la primera aproximación. Se abre de esta forma un campo de trabajo adicional para siguientes desarrollos basados en la mejora de la localización para robots equipados con GPS cuando se produce la falta del mismo. Se recomienda hacer un énfasis mayor en la definición de las métricas para la comparación de mapas y los requisitos para dicha comparación definiendo un nuevo sistema que penalice de forma más precisa la comparación de mapas descentrados.

Por otro lado se ha probado un método aplicado a la visión por computador para la localización de robot en entornos exteriores (SIFT). Este método altamente potente con imágenes no ha dado un buen resultado para los mapas almacenados por el robot debido a la insuficiencia de detalle de dichos mapas. Es esperable que para entornos reales no se produzca tanta insuficiencia y por el contrario se dote de más realismo a los mapas de forma que se pueda configurar el sistema para la localización de puntos característicos de forma fiable. Es importante señalar la potencia de este método y la necesidad de una mayor investigación para su aplicación al posicionamiento de robots. Es probable que aumentando la información contenida en el mapa local realizado en una pasada del laser mediante la fusión de 2 o más pasadas con el robot estático se mejoren los resultados del algoritmo de localización visual.

7. BIBLIOGRAFÍA

- [1] Agencia de Defensa Europea <http://www.eda.europa.eu>
- [2] Documentación proyecto NM-RS
- [3] Peter S. Maybeck. "Stochastic Models, Estimation, and Control", volume 1. Academic Press, 1979.
- [4] Paloma de la Puente Yusti. "Sistema de navegación de un robot móvil". Proyecto Final de Carrera, DISAM 2007.
- [5] José María Cañas, Lía García. "Construcción de mapas dinámicos: comparativa", Universidad Rey Juan Carlos, enero 2002.
- [6] C. Peignot, F. Wawak, F. Matía y E.A. Puente, "Integration of heterogeneous world mapping techniques in the navigation system of an autonomous mobile robot", Computer Vision and Mobile Robotics, Workshop CVMR'98, Santorini (Greece), 1998.
- [7] D. Pagac, E. Nebot, H. Durrant-Whyte, "An evidential approach to probabilistic map-building", IEEE, Transactions on Robotics and Automation, Vol 14, Num 4, August 1998.
- [8] M. Poloni, G. Ulivi, M. Venditelli, "Fuzzy logic and autonomous vehicles: experiments in ultrasonic vision Fuzzy Sets and Systems 69", 1995.
- [9] J. Borenstein, Y. Koren, "Histogramic in-motion mapping for mobile robot obstacle avoidance", IEEE Journal of Robotics and Automation, Vol 7, No 4, 1991.
- [10] R. Smith, M. Self, P. Cheeseman, A stochastic map for uncertain spatial relationships, in: Fourth International Symposium of Robotics Research, 1987, pp. 467–474.
- [11] Juan Nieto_, Tim Bailey, Eduardo Nebot, "Recursive scan-matching SLAM." ARC Centre of Excellence for Autonomous Systems (CAS), The University of Sydney, NSW, Australia
- [12] László Neumann, Balázs Csébfalvi, Andreas König, Eduard Gröller, "Gradient Estimation in volume Data using 4D Linear regression". Institute of Computer Graphics, Vienna University of Technology. EUROGRAPHICS 2000.

[13] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", Computer Science Department, University of British Columbia, 2004.

[14] Andrea Vedaldi, "An implementation of SIFT detector and descriptor", University of California at Los angeles.

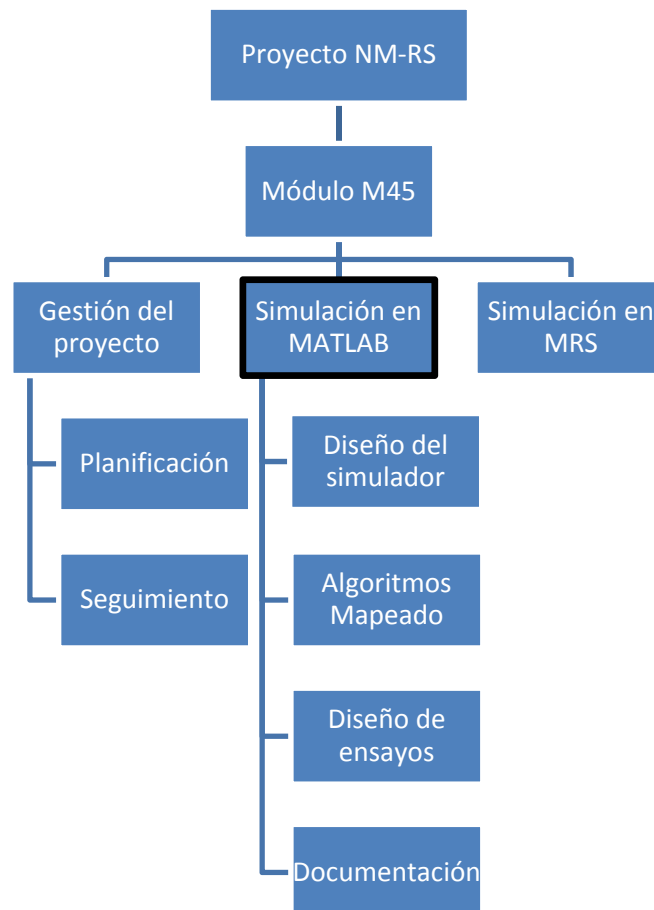
[15] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter", Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175.

8. ANEXOS

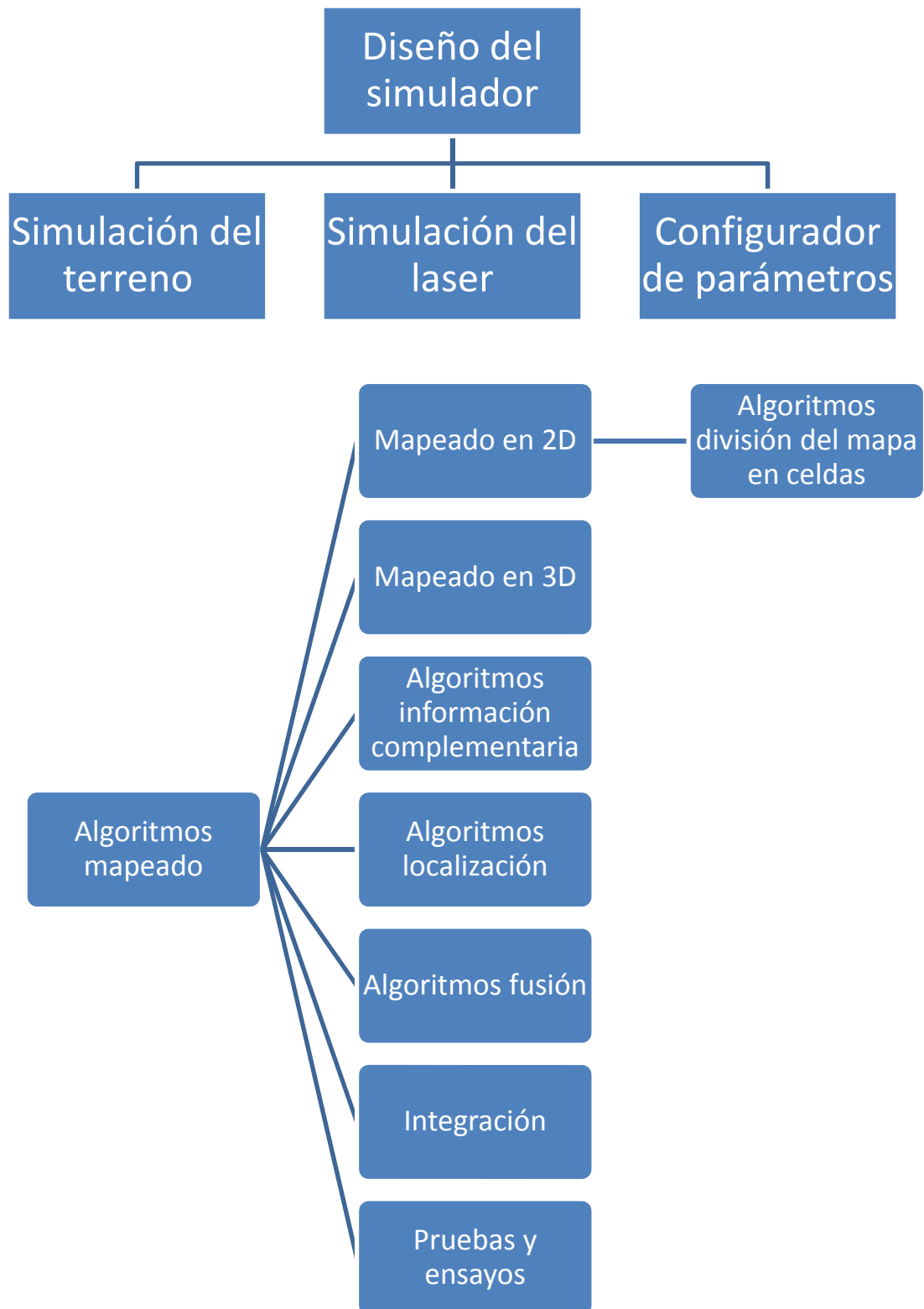
ANEXO I: ESTRUCTURA DE DESCOMPOSICIÓN DEL PROYECTO

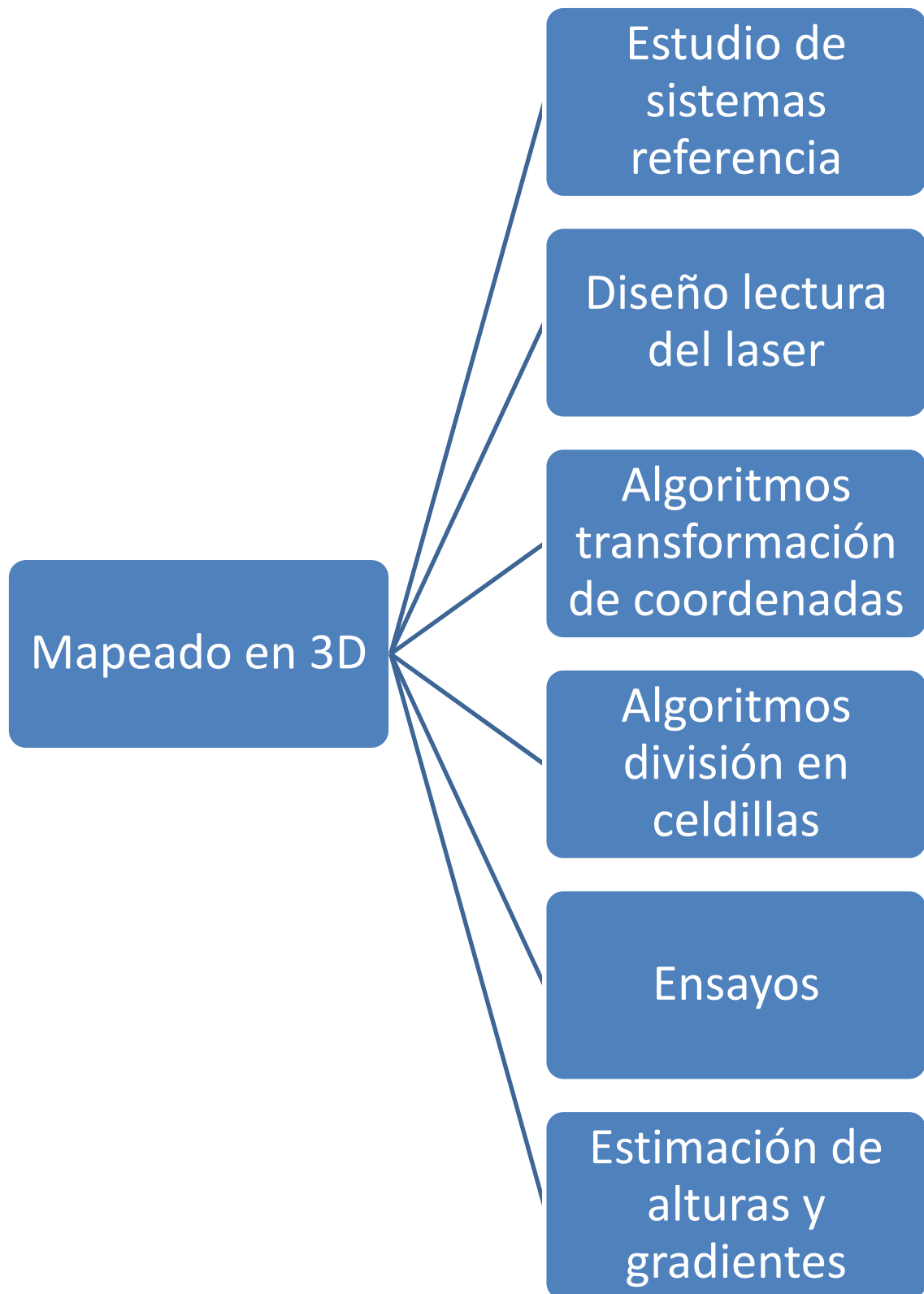
En la siguiente figura se muestra la descomposición del proyecto europeo NM-RS y la división en módulos del mismo. El proyecto fin de carrera presente se ha basado en el desarrollo de parte del módulo M45.

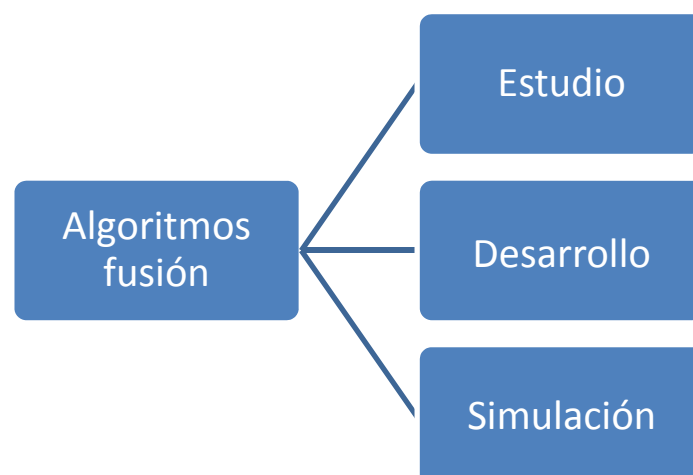
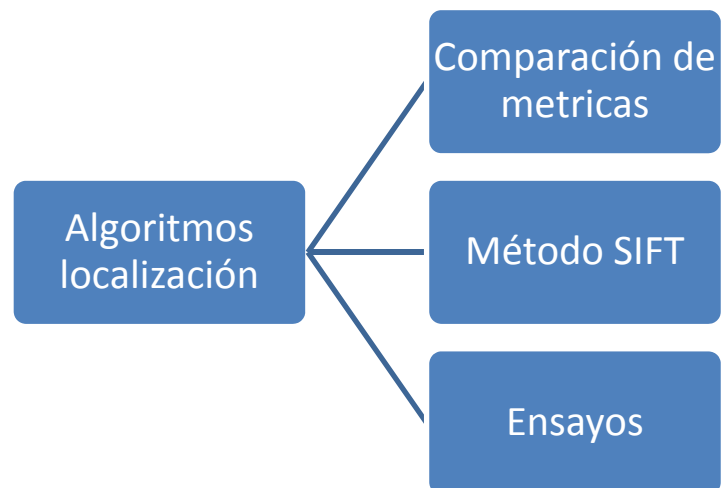
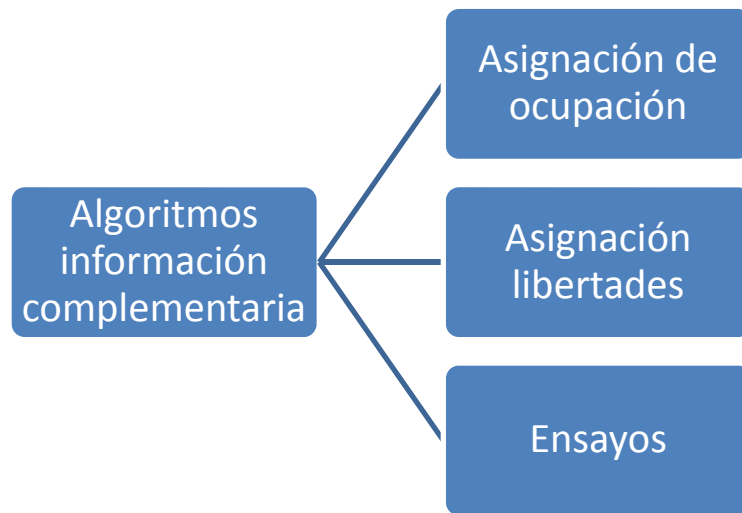
A continuación se detallan los elementos del módulo M45, destacando con borde negro los elementos desarrollados en el presente proyecto final de carrera.

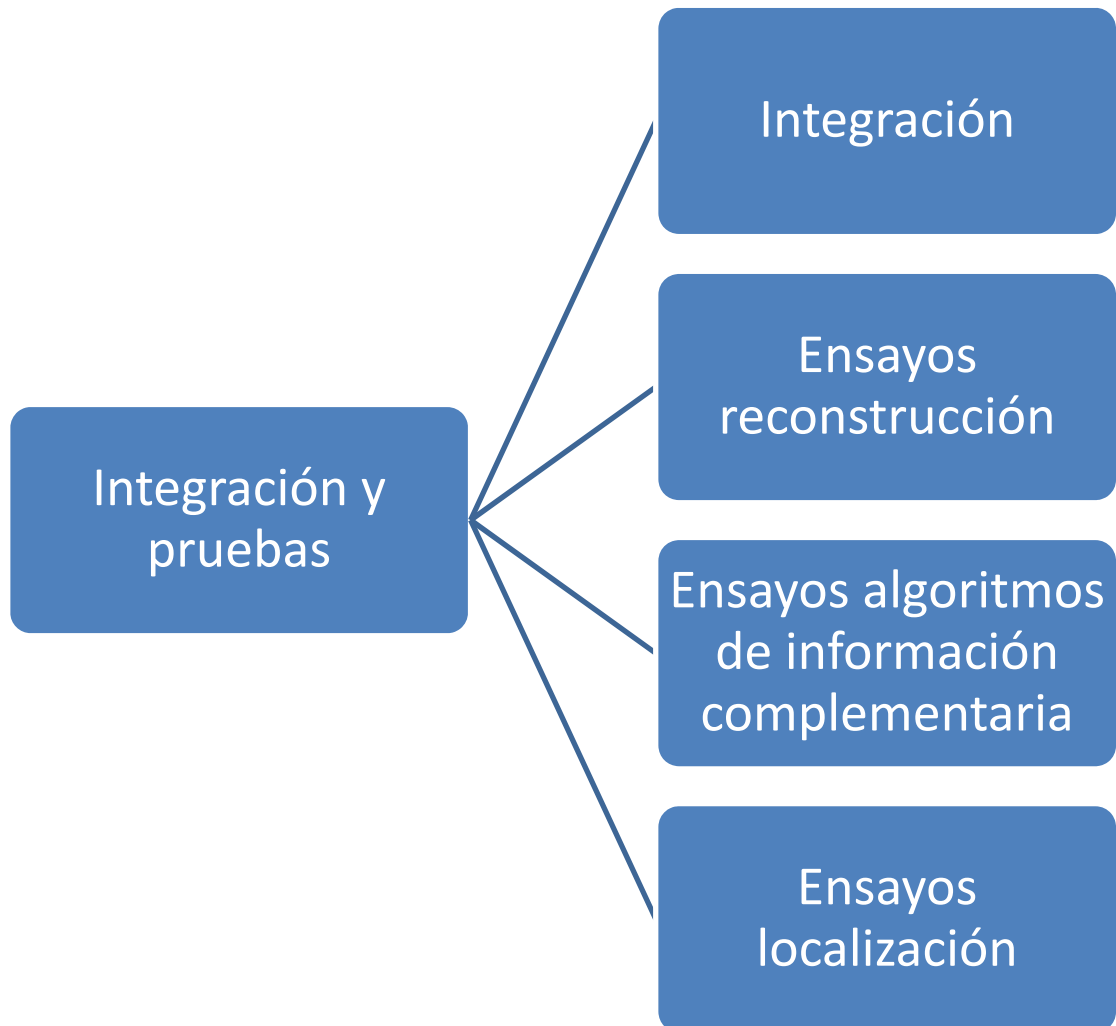


La gestión del proyecto consiste en planificar el proyecto al comienzo del mismo y realizar un seguimiento y control durante toda la duración del mismo. Este proyecto se ha realizado mediante reuniones enmarcadas dentro del proyecto NM-RS.

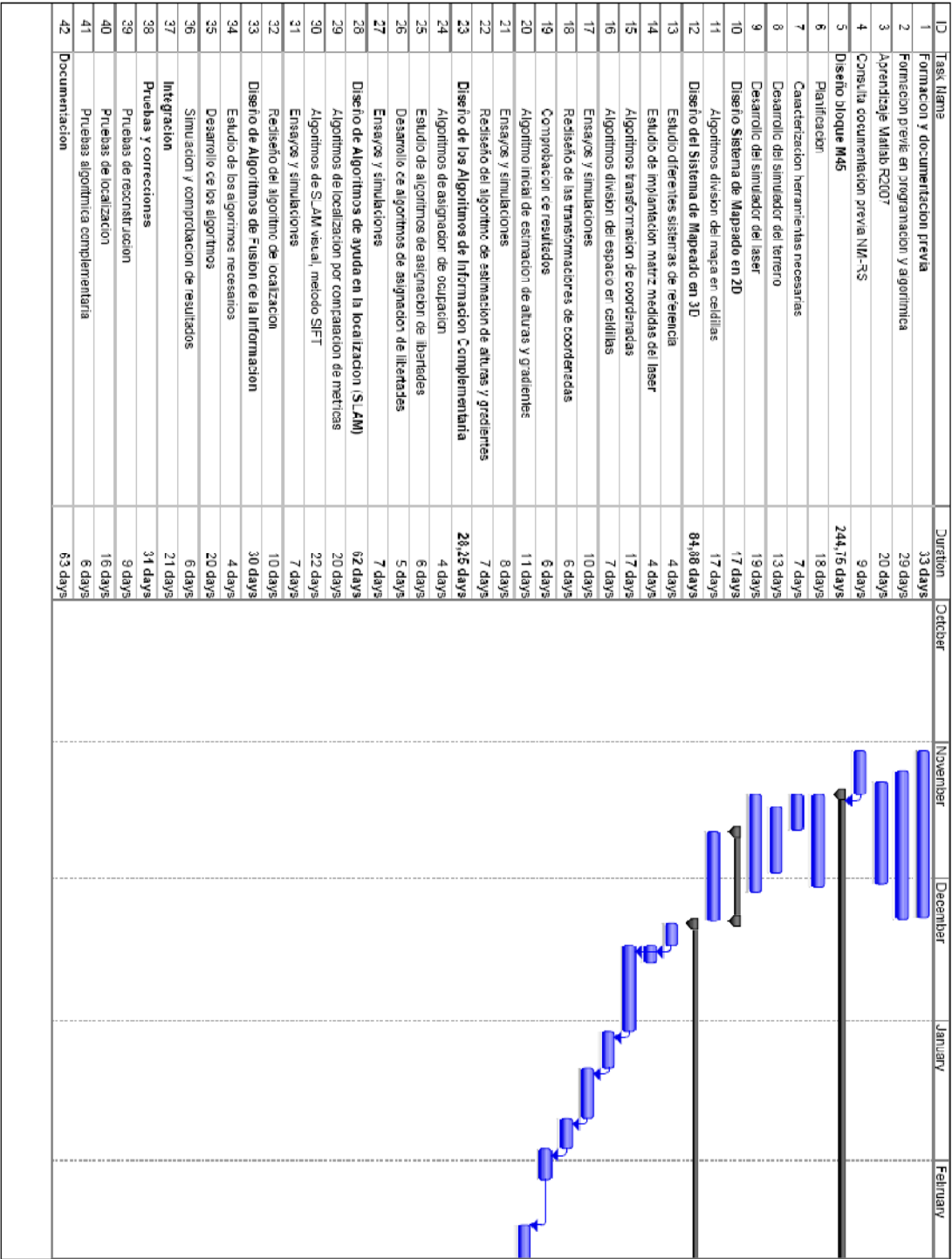


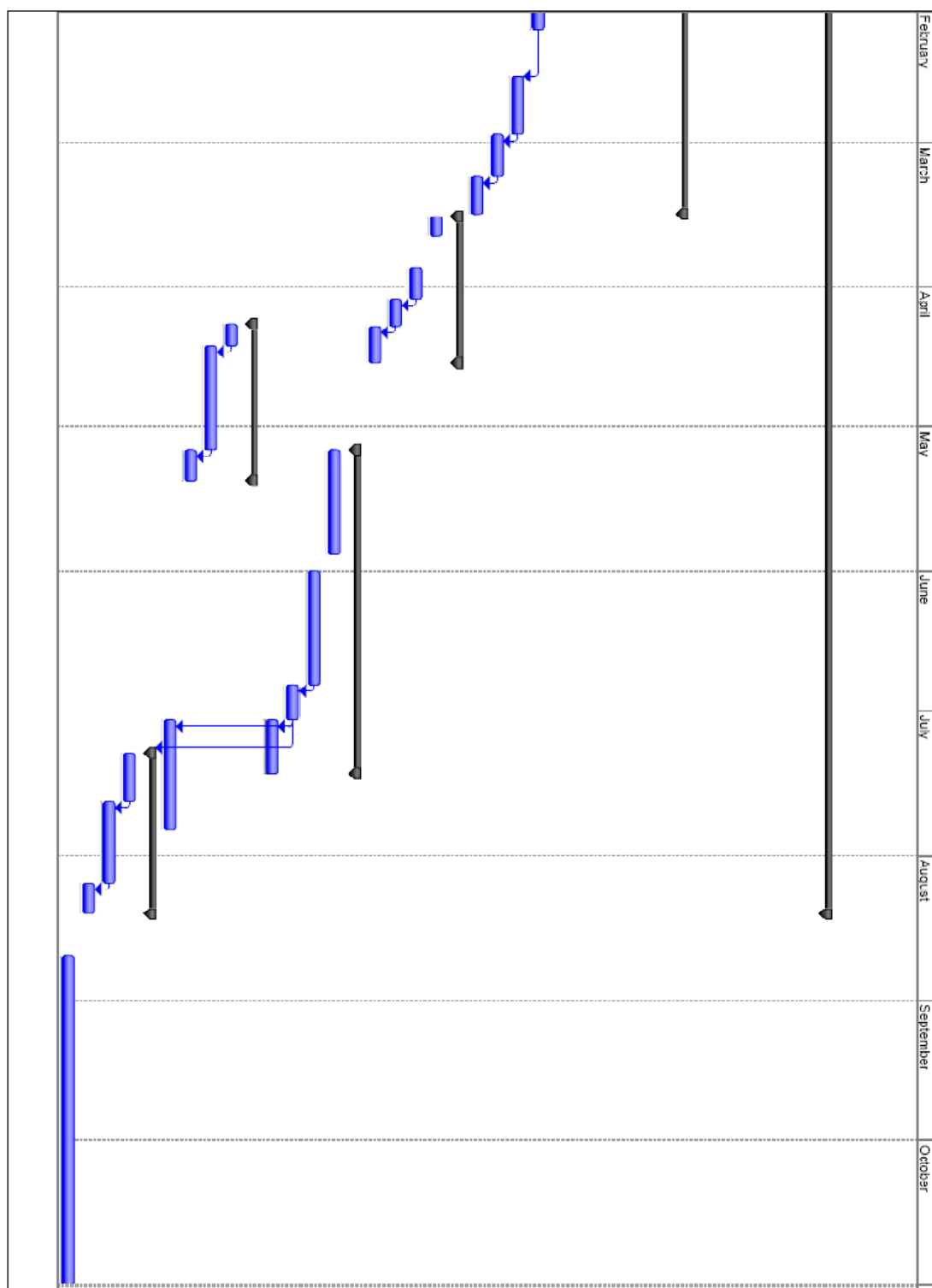






ANEXO II. PLANIFICACIÓN. DIAGRAMA DE GANTT.





ANEXO III. PRESUPUESTO

A continuación se detalla el presupuesto del proyecto realizado, hay que tener en cuenta que los cálculos son aproximados, para considerar que el desarrollo del software realizado no solo se amortizarán en el presente proyecto sino que se utilizará en otros proyectos.

Concepto	Valor
Inmuebles:	
Alquiler laboratorios	12.000,00 €
Mantenimiento y Servicios	3.000,00 €
Subtotal	15.000,00 €
Equipos hardware	
PCs	2.000,00 €
Comunicaciones	500,00 €
Mantenimiento	500,00 €
Subtotal	3.000,00 €
Software	
Licencias software	2.000,00 €
Subtotal	2.000,00 €
Personal	
Ingeniero 800 horas , 30 €/h	24.000,00 €
Director del proyecto, 150 horas, 60 €/h	9.000,00 €
Subtotal	33.000,00 €
Total del proyecto	53.000,00 €

ANEXO IV. EL FILTRO DE KALMAN

El Filtro de Kalman Discreto

En 1960, R.E. Kalman publicó un famoso documento describiendo una solución recursiva a el problema de filtrado de datos lineal y discreto. Desde aquel tiempo, debido en gran parte a los avances en computación digital, El filtro de Kalman ha sido objeto de numerosas investigaciones y aplicaciones, particularmente en el area de navegación asistida o autónoma.

El proceso a estimar

El filtro de Kalman direcciona el problema general de tratar de estimar el estado $x \in R^n$ de un proceso discreto en el tiempo que es gobernado por una ecuación diferencial lineal y estocástica :

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (86)$$

Con una medida $z \in R^n$ que es:

$$z_k = Hx_k + v_k \quad (87)$$

Las variables aleatorias w_k y v_k representan el ruido del proceso y de la medida respectivamente. Se asume que son independientes una de la otra, blancas y con una distribución de probabilidad normal

$$\begin{aligned} p(w) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned} \quad (88)$$

En la práctica, la matriz de covarianza del ruido Q y la matriz de covarianza de la medida R pueden cambiar a cada paso o medida, sin embargo aquí asumiremos que son constantes.

La matriz A $n \times n$ en la ecuación diferencial anterior relaciona el estado en el paso anterior $k-1$ con el estado en el estado actual k , en ausencia de ruido. Pero en la

práctica A puede cambiar con cada paso, pero aquí asumiremos que es constante. La matriz B de dimensiones $n \times l$ relaciona la entrada de control $u \in R^l$ con el estado x . La matriz H de dimensiones $m \times n$ en la ecuación de la medida relaciona el estado con la medida z_k . En la práctica H puede cambiar con cada paso o medida, pero aquí la consideraremos constante otra vez.

Los orígenes computacionales del filtro

Se define la variable $\hat{x}_k^- \in R^n$ (notese el signo menos) siendo es estado a priori al paso k mostrando el conocimiento del proceso a priori al paso k , y $\hat{x}_k \in R^n$ será el estado estimado posteriori en el paso k con la medida z_k . Se puede definir los errores de los estados a priori y a posteriori de la siguiente manera:

$$e_k^- \equiv x_k^- - \hat{x}_k^-, \text{ y } e_k \equiv x_k - \hat{x}_k \quad (89)$$

La covarianza del error estimado a priori es entonces:

$$P_k^- = E[e_k^- e_k^{-T}], \quad (90)$$

y la covarianza del error estimado a posteriori es:

$$P_k = E[e_k e_k^T]. \quad (91)$$

Derivando de las ecuaciones del filtro de Kalman, se comienza con el objetivo de encontrar una ecuación que calcule el estado estimado a posteriori \hat{x}_k como una combinación lineal de un estado a priori y una diferencia ponderada entre la medida actual z_k y la predicción de la medida Hx_k^- tal y como se muestra en (92).

$$\hat{x}_k \equiv \hat{x}_k^- + K(z_k - Hx_k^-) \quad (92)$$

La diferencia $(z_k - Hx_k^-)$ es llamada innovación en la medida, o residuo. El residuo refleja la discrepancia entre la predicción de la medida Hx_k^- y la medida actual z_k . Un residuo nulo indicaría que las dos están en completo acuerdo.

La matriz K de dimensiones $n \times n$ de (92) es elegida de tal forma que la ganancia K minimice la covarianza del error a posteriori (91). Esta minimización puede ser llevada a cabo sustituyendo (92) en la definición de e_k , sustituyendo todo ello en (91) desarrollando las operaciones indicadas, derivando el resultado respecto a K , igualando el resultado a cero y resolviendo para K .

$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$ $= \frac{P_k^- H^T}{H P_k^- H^T + R}$	(93)
--	------

Observando (93) se puede ver que cuando la covarianza del error se aproxima a cero, la ganancia K hace el residuo mas pesado. Específicamente:

$\lim_{R_k \rightarrow 0} K_k = H^{-1}$	(94)
---	------

Por otro lado, al aproximarse la covarianza del error estimado a priori P_k^- a cero, la ganancia K pondera el residuo con menos fuerza. Específicamente:

$\lim_{P_k^- \rightarrow 0} K_k = 0$	(95)
--------------------------------------	------

Otra forma de ver la aportación de K sería ver como según la covarianza de la medida del error R se aproxima a cero, la medida actual z_k es mas confiable, mientras la medida predicha $H\hat{x}_k^-$ es cada vez menos fiable. Por otro lado, así como la covarianza del error estimado a priori P_k^- se aproxima a cero la medida actual z_k es cada vez menos fiable, mientras la predicción de la medida $H\hat{x}_k^-$ es cada vez mas fiable.

Los orígenes probabilísticos del filtro

La justificación para (91) viene de la probabilidad para un estado estimado a priori \hat{x}_k^- condicionado por todas las medidas a priori z_k (Regla de Bayes). Por ahora es

suficiente con señalar que el filtro de Kalman mantiene los dos primeros momentos de la distribución del estado.

$E[x_k] = \hat{x}_k$ $E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k$	(96)
--	------

La estimación del estado a posteriori (91) refleja la media (el primer momento) de la distribución del estado, que tendrá distribución normal si las condiciones de (88) se verifican. La covarianza de la estimación del error a posterior (90) refleja la varianza de la distribución del estado, en otras palabras

$p(x_k z_k) \sim N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = N(\hat{x}_k, P_k)$	(97)
---	------

El algoritmo del Filtro de Kalman discreto

Esta sección comenzará describiendo de forma breve la operación de alto nivel de una forma de filtro de Kalman discreto. Después, se centrará en las ecuaciones específicas para este tipo de filtro.

El filtro de Kalman estima un proceso usando una forma de control realimentado: el filtro estima el estado de un proceso en un tiempo determinado y después obtiene realimentación en la forma de medidas con cierta cantidad de ruido. Por ello, las ecuaciones del filtro de Kalman forman dos grupos: ecuaciones actualizadas con el tiempo y ecuaciones actualizadas con la medida. Las ecuaciones actualizadas con el tiempo son responsables de proyectar hacia adelante (en el tiempo) el estado actual y la estimación de la covarianza del error para obtener una estimación a priori por el siguiente paso de tiempo. La actualización de las ecuaciones de la medida son responsables de la realimentación. Por ejemplo, por incorporar una nueva medida en una estimación a priori para obtener una estimación a posteriori mejorada.

Las ecuaciones actualizadas con el tiempo también pueden ser entendidas como unas ecuaciones predictivas, mientras que las ecuaciones de actualización de la medida pueden ser entendidas como unas ecuaciones de corrección. De hecho la estimación

final del algoritmo crea un algoritmo de predicción-corrección para resolver problemas numéricos.

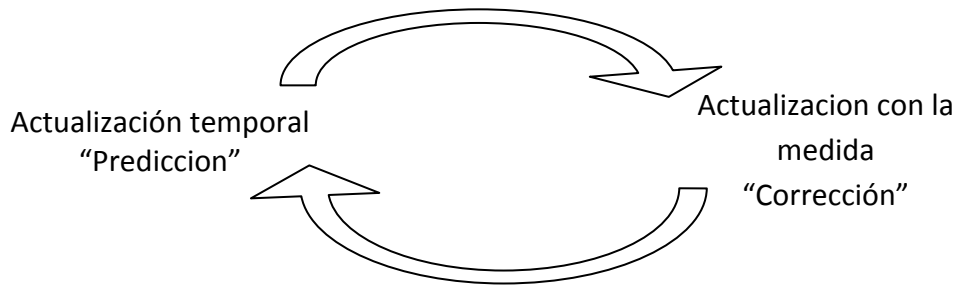


Figura 93. Ciclo del filtro de Kalman discreto

Las ecuaciones específicas para la actualización con el tiempo y con la medida son las siguientes:

Ecuaciones actualizadas con el tiempo para el filtro de Kalman discreto

$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$	(98)
---------------------------------------	------

$P_k^- = AP_{k-1}A^T + Q$	(99)
---------------------------	------

Notese de nuevo como las ecuaciones actualizadas con el tiempo proyectan las estimaciones del estado y la covarianza desde el paso $k-1$ hasta el paso k . A y B son desde (86), mientras que Q viene de (88).

Ecuaciones actualizadas con la medida para el filtro de Kalman discreto

$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$	(100)
---	-------

$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$	(101)
---	-------

$P_k = (1 - K_k H) P_k^-$	(102)
---------------------------	-------

La primera tarea durante la actualización de la medida es calcular la ganancia de Kalman, K_k . Notese que la ecuación dada aquí como (100) es la misma que en (93). El siguiente paso es actualizar la medida del proceso para obtener z_k y entonces generar una estimación del estado a posteriori incorporando la medida como se muestra en (101). De nuevo (101) es simplemente (92) repetida aquí para completitud. El paso final es obtener una estimación de la covarianza del error a posteriori via (102).

Después de cada par de actualización de tiempo y medida, el proceso es repetido con la estimación previa a posteriori usada para proyectar o predecir unos nuevos valores

estimados a priori. Esta característica recursiva es una de las mas atractivas del filtro Kalman, y que hacen del mismo que las implementaciones practicas sean mucho mas factibles que en otro tipo del filtros como el de Wiener que esta diseñado para operar con todos los datos directamente en cada estimacion . El filtro de Kalman en lugar de eso opera de forma recursiva desde el estado anterior.

Parámetros del filtro y ajuste

En la implementación actual del filtro, la covarianza del ruido de la medida R es usualmente medida de forma anterior al funcionamiento del filtro. Medir la covarianza del error de la medida es generalmente posible porque se necesita medir el proceso en cualquier caso por ello se deben tomar diversas muestras off-line para determinar la varianza de la medida.

La determinación de la la covarianza del ruido del proceso es generalmente mas difícil porque usualmente no se tienen la habilidad de observar el proceso que se esta estimando. A veces un modelo relativamente simple puede producir resultados aceptables si uno “introduce” cierto grado de incertidumbre en el proceso seleccionando Q adecuadamente. Ciertamente en ese caso se puede esperar que las medidas del proceso sean factibles.

En otro caso, tanto si se tiene o no base racional para la elección de los parámetros, a menudo se puede obtener un rendimiento varias veces superior ajustando los parámetros Q y R del filtro. El ajuste es realizado normalmente off-line, frecuentemente con la ayuda de otro filtro de Kalman en un proceso referido a la identificación del sistema.

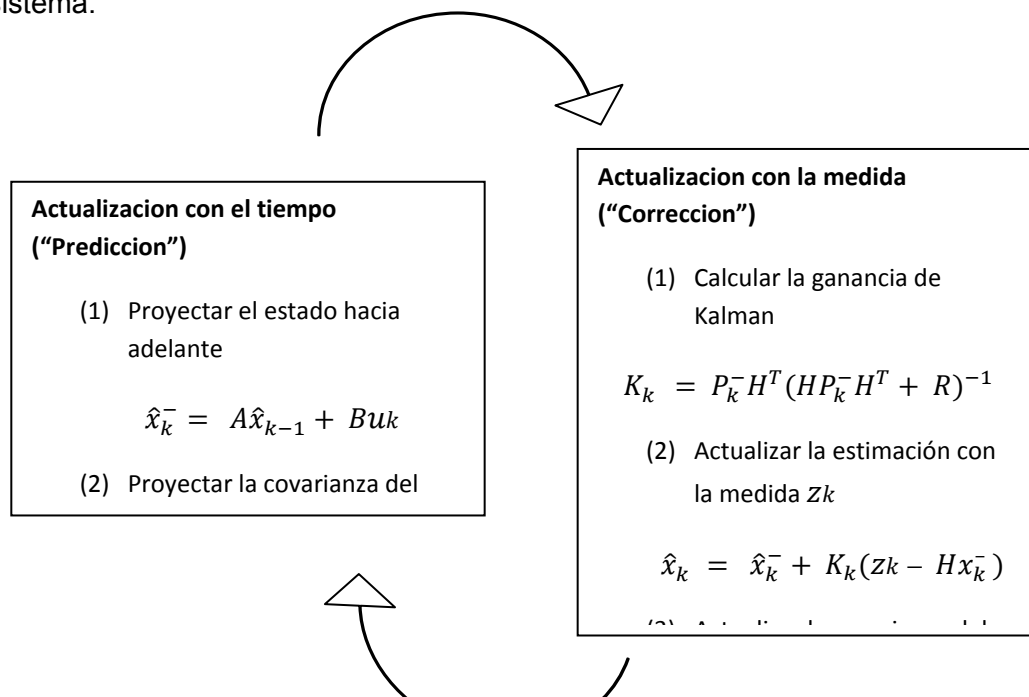


Figura 94: Ciclo completo del filtro de Kalman con ecuaciones de actualización

Para completar hay que dejar notar que bajo ciertas condiciones donde Q y R son de hecho constantes, la estimación de la covarianza del error P_k y la ganancia de Kalman K_k se estabilizarán rápidamente y entonces permanecerán constantes. En ese caso, estos parámetros pueden ser pre-calculados ejecutando el filtro off-line, o por ejemplo determinando el valor del estado estacionario de P_k .

Es frecuente sin embargo el caso de que el error en la medida no permanezca constante. Por ejemplo, cuando las balizas de avistamiento en seguidores optoelectronicos para paneles de techo, el ruido en las medidas de las balizas cercanas será mas pequeño que en las balizas lejanas. También, el ruido del proceso Q es a menudo cambiado de forma dinámica durante la operación del filtro, haciéndose Q_k para ajustarse a diferentes dinámicas. Por ejemplo, en el caso del seguimiento de la cabeza de un usuario de un entorno virtual 3D se puede reducir la magnitud de Q_k si el usuario parece moverse de forma lenta, e incrementar la magnitud si la dinámica empieza a cambiar rápidamente. En cuyo caso Q_k debe ser elegida tomando en cuenta ambas incertidumbres sobre las intenciones del usuario y la incertidumbre del modelo [15].

EL FILTRO DE KALMAN EXTENDIDO (EKF)

El proceso a estimar

Como se describe en secciones anteriores, el filtro de Kalman resuelve el problema de intentar estimar el estado $x \in R^n$ de un proceso discreto en el tiempo que es gobernado por una ecuación diferencial lineal y estocástica. Pero, ¿qué ocurre si el proceso a ser estimado es no lineal o si la relación de la medida con el proceso es no lineal?. Algunos de las mas interesantes y existosas aplicaciones del filtro de Kalman se encuentran en estas situaciones. Un filtro de Kalman linealizado en la media y covarianzas actuales es denominado el filtro de Kalman extendido o EKF.

Haciendo algo parecido a las series de Taylor, se puede linealizar la estimación alrededor del estado actual usando derivadas parciales del proceso y de las funciones de medida para calcular estimaciones incluso en relaciones no lineales. Por ello, se debe comenzar modificando algunas de las ecuaciones antes expuestas. Asumamos que en el proceso otra vez existe un vector de estado $x \in R^n$, pero que ahora el proceso es gobernado por una ecuación diferencial estocástica no lineal

$x_k = f(x_{k-1}, u_k, w_{k-1})$	(103)
----------------------------------	-------

con una medida $z \in R^m$ que es:

$z_k = h(x_k, v_k)$	(104)
---------------------	-------

Donde las variables aleatorias w_k y v_k vuelven a representar el ruido del proceso y de la medida como en (93) y (94). En este caso la función no lineal f en la ecuación diferencial (103) relaciona el estado en el estado anterior $k-1$ con el estado en el punto actual k . Esto incluye como parámetros cualquier función u_k y el ruido de media cero w_k . La función no lineal h en la ecuación de la medida (104) relaciona el estado x_k con la medida z_k .

En la práctica no se conocen los valores del ruido w_k y v_k en cada paso de ejecución. Sin embargo se pueden aproximar los vectores de estado y de medida sin ellos de la forma:

$\tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0)$	(105)
--	-------

$\tilde{z}_k = h(\hat{x}_k, 0)$	(106)
---------------------------------	-------

Donde \hat{x}_k , es alguna estimación a posteriori del estado.

Es importante notar que un fallo fundamental del EKF es que las distribuciones de varias variables aleatorias dejan de ser normales después de caer bajo transformaciones no lineales. El EKF es simplemente un estimador del estado que aproxima la optimalidad de la regla de Bayes por linealización.

Los orígenes computacionales del filtro

Para estimar un proceso con relaciones no lineales con la medida comenzaremos escribiendo unas nuevas ecuaciones de gobierno que linealizan un estimador sobre (104) y (106),

$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1}$	(107)
---	-------

$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_{k-1}) + Vv_{k-1}$	(108)
---	-------

Donde,

- x_k y z_k son los vectores del estado actual y de la medida
- \tilde{x}_k y \tilde{z}_k son los vectores de aproximación del estado y de la medida de (105) y (106)
- \hat{x}_k es la estimación a posteriori del estado en el paso k
- Las variables aleatorias w_k y v_k representan el ruido del proceso y el ruido de la medida como en (88) .
- A es la matriz Jacobiana de derivadas parciales de f respecto a x , esto es

$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, u_k, 0)$	(109)
--	-------

- W es la matriz Jacobiana de derivadas parciales de f respecto a w ,

$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\tilde{x}_k, u_k, 0)$	(110)
--	-------

- H es la matriz Jacobiana de las derivadas parciales de h respecto a x ,

$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0)$	(111)
---	-------

- V es la matriz Jacobiana de las derivadas parciales de h respecto a v

$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0)$	(112)
---	-------

Notese que por simplicidad en la notación no se usa el subíndice k con las Jacobianas A , W , H y V , incluso cuando estas sean diferentes en cada paso temporal.

Ahora se definirá una nueva notación para el error de predicción,

$\tilde{e}_k \equiv x_k - \tilde{x}_k ,$	(113)
--	-------

y el residuo de la medida,

$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k$	(114)
--	-------

Recuerdese que en la practica no se tiene acceso a x_k en (113), puesto que es la cantidad que se esta tratando de estimar. Por otro lado no se tiene acceso a z_k en (114) ya que es la medida actual que se utiliza para tratar de estimar x_k . Usando (113) y (114) se puede escribir las ecuaciones que gobiernan el proceso de error como

$\tilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k$	(115)
--	-------

$\tilde{e}_{z_k} \approx H(\tilde{e}_{x_k}) + \eta_k$	(116)
---	-------

Donde ε_k y η_k representan nuevas e independintes variables aleatorias de media cero y matrices de covarianza WQW^T y VQV^T , con Q y R como en (88) .

Notese que las ecuaciones (115) y (116) son lineales y recuerdan a las ecuaciones en diferencias del filtro de Kalman discreto. Esto hace que se piense en usar el residuo de la medida \tilde{e}_{z_k} en (116) como segundo filtro de Kalman para estimar el error de predicción \tilde{e}_{x_k} dado por (117). Este estimador, llamado \hat{e}_k , podría ser usado junto con (115) para obtener una estimación el estado a posteriori para el proceso original no lineal como,

$\hat{x}_k = \tilde{x}_k + \hat{e}_k .$	(117)
---	-------

Las variables aleatorias de (115) y (116) tienen aproximadamente las siguientes funciones probabilísticas de distribución:

$p(\tilde{e}_{x_k}) \sim N(0, E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T])$	(118)
--	-------

$p(\varepsilon_k) \sim N(0, W Q_k W^T)$	(119)
$p(\eta_k) \sim N(0, V R_k V^T)$	(120)

Haciendo esas aproximaciones y dando a \hat{e}_k el valor de cero, la ecuación del filtro de Kalman usada para estimar \hat{e}_k es

$\hat{e}_k = K_k \tilde{e}_{z_k}.$	(121)
------------------------------------	-------

Y sustituyendo (121) en (117) y haciendo uso de (116) se puede ver que no se necesita el segundo filtro de Kalman:

$\hat{x}_k = \tilde{x}_k + K_k \tilde{e}_{z_k} = \tilde{x}_k + K_k (z_k - \tilde{z}_k)$	(122)
---	-------

La ecuación (122) puede ser usada ahora como actualización con la medida en el filtro de Kalman extendido, con \tilde{x}_k y \tilde{z}_k viniendo de (107) y (108), y la ganancia de Kalman K_k viniendo de (94) con la sustitución apropiada para la covarianza del error de la medida.

El juego completo de ecuaciones para el EKF será mostrado a continuación. Notese que se ha sustituido \hat{x}_k^- por \tilde{x}_k para hacerlo consistente con el superíndice menos de la notación a priori, y ahora se añade el subíndice k a las Jacobianas A, W, H y V, para reforzar la idea de que son diferentes a cada paso de tiempo.

Ecuaciones actualizadas con el tiempo para el EKF

$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)$	(123)
$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$	(124)

Así como en el filtro de Kalman básico, las ecuaciones de actualización con el tiempo proyectan la estimación del estado y la covarianza desde el estado previo k-1 hasta el actual k. De nuevo f en (123) viene de (107), A_k y W_k son las Jacobianas del proceso en el paso k, y Q_k es la covarianza del ruido del proceso en el paso k.

Ecuaciones actualizadas con la medida para el EKF

$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$	(125)
$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0))$	(126)
$P_k = (1 - K_k H_k) P_k^-$	(127)

Así como en el filtro de Kalman básico y discreto las ecuaciones actualizadas con la medida corrigen las estimaciones del estado y la covarianza con la medida z_k . De nuevo h en (126) viene de (108), H_k y V_k son las Jacobianas de la medida en el paso k , y R_k es la covarianza del ruido de la medida en el paso k .

La forma de comportarse del EKF es la misma que la del filtro de Kalman lineal, por lo tanto las figuras que se utilizaron anteriormente para explicar el funcionamiento del filtro de Kalman básico se pueden inferir ahora para el EKF [15].

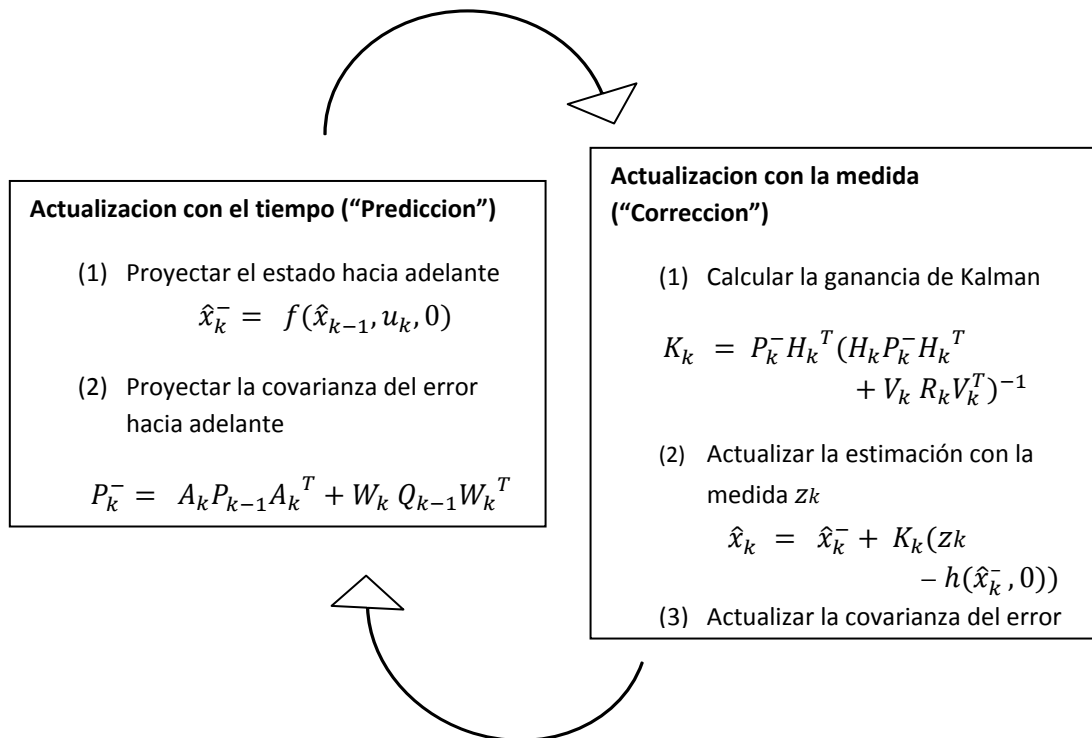


Figura 95. Ciclo completo del filtro de Kalman extendido con ecuaciones de actualización

Una característica importante del EKF es que la Jacobiana H_k en la ecuación de la ganancia de Kalman K_k sirve para corregir la propagación o magnificación solo de la componente más relevante de la información de la medida. Por ejemplo, si no hay una correspondencia uno a uno entre la medida z_k y el estado via h , la Jacobiana H_k afecta a la ganancia de Kalman de tal forma que esta solo magnifica la porción del residuo $z_k - h(\hat{x}_k^-, 0)$ que no afecta al estado. Si ocurriera que sobre todas las medidas no hubiera una correspondencia uno a uno entre la medida y el estado via h entonces se puede esperar que el filtro diverja rápidamente. En ese caso el proceso sería inobservable [15].

